

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**

**Факультет прикладної математики та інформатики**

(повне найменування назва факультету)

**Кафедра інформаційних систем**

(повна назва кафедри)

## **ДИПЛОМНА РОБОТА**

**ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯ КОНВЕЄРНИМ  
ВИРОБНИЦТВОМ НА ПІДПРИЄМСТВІ**

**Виконав(ла): студент(ка) групи ПМІ-44  
спеціальності 122 – комп'ютерні науки**  
(шифр і назва спеціальності)

**Ляхович Я.В.**

(підпис)

(прізвище та ініціали)

**Керівник**

(підпис)

**Соколовський Я.І.**

(прізвище та ініціали)

**Рецензент**

(підпис)

(прізвище та ініціали)

2023

# ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Кафедра інформаційних систем

Освітньо-кваліфікаційний рівень бакалавр

Факультет прикладної математики та інформатики

Спеціальність 122 – комп'ютерні науки

«ЗАТВЕРДЖУЮ»

Зав. кафедрою проф. Шинкаренко Г.А.

« 5 » вересня 2022 р.

## ЗАВДАННЯ

### НА ДИПЛОМНУ РОБОТУ СТУДЕНТА

Ляхович Ярини Володимирівної

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інформаційної системи управління виробництвом компанії з виготовлення меблів

керівник роботи Соколовський Я.І.

затверджена на Вченій раді факультету від « 13 » вересня 2022 р. №

2. Строк подання студентом роботи 13.06.2023 р.

3. Вихідні дані до роботи \_\_\_\_\_

Література та інтернет ресурси за тематикою роботи

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

1. Огляд сучасного стану проблеми

2. Модель даних. Обрані методи та технології

3. Програмна реалізація

4. Результати апробації

5. Висновки та підсумки

5. Перелік графічного матеріалу (з докладним зазначенням обов'язкових креслень) \_\_\_\_\_

Діаграми, скріншоти

Презентація дипломної роботи



## ТЕХНІЧНЕ ЗАВДАННЯ

**Призначення системи:** Головним завданням є розроблення інформаційної системи управління виробництвом компанії з виготовлення меблів, доступ до якої буде здійснюватися за допомогою мережі Інтернет. Система повинна забезпечувати зберігання всієї інформації в єдиній глобальній базі даних, з якою працює програмне забезпечення, що дозволяє здійснювати внесення, зберігання, обробку даних та отримання передбачених звітів.

### **Основні функції системи:**

- формування та збереження документів – замовлень;
- формування та збереження документів з переліком та описом деталей і їх технологічним маршрутом виготовлення для кожного виробу компанії;
- можливість роздрукувати дані документи;
- можливість редагування інформації замовлень, виробів, деталей та їх складових(наприклад, списку доступних матеріалів та кольорів);
- доступ для роботи в системі повинен здійснюватися шляхом введення індивідуального імені та пароля користувача, що відповідають рівню доступу до інформації.

**Інтерфейс системи:** Зовнішній вигляд інформаційної системи має бути логічним і лаконічним, позбавлений важких елементів, відповідати сучасним тенденціям сучасного дизайну та максимально подібним в різних версіях веб-браузерів.

## АНОТАЦІЯ

Бакалаврська дипломна робота (проект): пояснювальна записка: 66 стор., 6 рис., 3 табл., 5 додатків, 26 джерел.

Дипломна робота присвячена проблемі проектуванні та реалізації веб-орієнтованій інформаційній системі для компанії з виготовлення меблів. з допомогою сучасних комп'ютерних технологій таких як ASP.NET та ADO.NET.

Розроблена інформаційна система здатна реалізувати поставлені перед нею задачі, містить максимально дружелюбний інтерфейс користувача і весь необхідний комплекс інструментів для спрощення та автоматизація технічного документообігу підприємства.

ВЕБ, ІНФОРМАЦІЙНА СИСТЕМА, HTML, CSS, JAVASCRIPT, JQUERY, BOOTSTRAP, C#, .NET FRAMEWORK, ASP.NET, ADO.NET

## ABSTRACT

The bachelor project's explanatory note includes 66 pages, 6 images, 5 tables, 26 sources.

The bachelor project is dedicated to the problem of designing and implementation web-based information system for the company which produces furniture using modern computer technologies such as ASP.NET and ADO.NET.

The developed information system is able to realize the goals, contains the most friendly user interface and all the necessary set of tools to simplify and automate the technical document of the company.

WEB, INFORMATION SYSTEMS, HTML, CSS, JAVASCRIPT, JQUERY, BOOTSTRAP, C #, . NET FRAMEWORK, ASP.NET, ADO.NET

## ЗМІСТ

АНОТАЦІЯ.....	5
ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	7
ВСТУП .....	8
РОЗДІЛ 1 СТАН ПРОБЛЕМНОЇ ОБЛАСТІ.....	9
1.1 Роль інформаційних систем .....	9
1.2 Аналіз завдання .....	13
РОЗДІЛ 2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....	15
2.1 Бази даних .....	15
2.2 Інтерфейс користувача.....	17
2.3 Серверна сторона інформаційної системи.....	21
РОЗДІЛ 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ .....	24
3.1 Реляційна алгебра.....	24
3.2 Реляційне числення.....	31
РОЗДІЛ 4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ .....	35
4.1 Вимоги до ПЗ.....	35
4.2 Проектування та реалізація бази даних.....	36
4.3 Проектування інтерфейсу користувача.....	42
4.4 Реалізація логіки функціонування інформаційної системи .....	55
ВИСНОВКИ .....	69
ПЕРЕЛІК ПОСИЛАНЬ.....	70
Додаток А Настанови користувача .....	72
Додаток Б Програмний код.....	76

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І  
ТЕРМІНІВ

CLR – Common Language Runtime

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

IIS - Internet Information Services

OS – operating system

UI – User Interface

БД – база даних

ІС – інформаційна система

ПК – персональний комп'ютер

ПЗ – програмне забезпечення

## ВСТУП

В сучасному суспільстві важко уявити функціонування підприємства (організації) без використання інформаційних комп'ютерних технологій. Це спричинено в першу чергу потребою швидко і якісно обробляти інформацію, забезпечувати оперативний документообіг між філіалами, швидкої передачі інформації, тощо. Паперові носії інформації по своїй природі не здатні забезпечити ці потреби. І власне тому в сучасному суспільстві велике значення надається зберіганню будь яких даних в електронному варіанті. Такий метод зберігання інформації відкриває наступні можливості:

- мінімізація витрат на збереження необмежених об'ємів інформації
- зручний і швидкий пошук необхідних даних
- передача цих даних на великі відстані
- автоматизація обробки будь якої інформації
- зручний доступ до інформації з будь якої точки світу

Найбільшим і найбільш розвинутим середовищем збереження і отримання інформації є всесвітні мережа інтернет. Збереження і обробка інформації в цій мережі відбувається за допомогою так званих “веб-орієнтованих” інформаційних систем. До таких систем можна віднести як і просту веб-сторінку, метою якої є проста публікація даних про підприємство, так і складну систему для управління підприємством, що базується на веб-технологіях.

З цих причин можна вважати тему даного дипломного проекту актуальною.

**Об'єктом дослідження** виступають інформаційні системи.

**Предметом дослідження** є проектування та програмна реалізація веб-орієнтованої інформаційної системи управління технічним документообігом для компанії з виготовлення меблів з використання .Net технологій, а саме ASP.Net та ADO.Net.

**Мета роботи** - розробити повнофункціональну інформаційну систему, яка б відповідала поставленим завданням і яку б можна було впровадити в виробництво.



## РОЗДІЛ 1 СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

### 1.1 Роль інформаційних систем

Раніше обчислювальна техніка використовувалась головним чином для виконання тих чи інших складних математичних розрахунків. Але навряд чи деяке велике підприємство мало на тоді автоматизовані системи ведення документообігу чи виконання певних бухгалтерських операцій. Це було пов'язано по-перше з малою потужністю обчислювальної техніки та її великою вартістю; по-друге, що, на мій погляд є головним, відсутністю технологій розробки таких програмних систем. У чому ж складність розробки систем такого типу? Це їх об'єм, а складність самої предметної області призводить до того, що такі системи не тільки дуже важко розробляти, але практично неможливо супроводжувати, модифікувати, розвивати, а також масштабувати, а негнучка система прогресивному підприємству непотрібна. Такі стадії життєвого циклу системи можливо реалізувати лише тоді, коли цей процес буде автоматизованим, тобто будуть розроблені відповідні технології.

Завдяки розвитку відповідних інформаційних технологій в наш час все більше і більше підприємств все ж таки переходять на автоматизовані системи. Перший поштовх до цього дало виникнення технології клієнт-сервер, коли вже з'явився централізований сервер, на якому були зосереджені всі основні данні та виконувались деякі операції. Але розробка систем такого типу потребувала вже доволі значних знань в області інформаційних технологій. Хоча час, витрачений на вивчення технології та її застосування при розробці системи, безперечно окупувався.

На зміну технології клієнт-сервер прийшли, так звані, Web-технології, які мають переваги перед технологією клієнт-сервер з одного боку, з іншого все ж таки було очевидно, що їх необхідно продовжувати розвивати. В наш час ця технологія

знаходиться на досить зрілому етапі розвитку, що відкриває такі можливості, про які раніше навіть фахівці в сфері інформаційних технологій навряд чи здогадувались.

Сьогодні перед підприємствами, установами, організаціями дуже гостро стають проблеми автоматизації управління. Це пов'язано з кількома причинами. По-перше, зменшується рентабельність бізнесу організацій і підприємств, загострюється ринкова конкуренція. По-друге, змінюється співвідношення кількості торгових і виробничих компаній. По-третє, зростання компаній (кількості працівників, поява регіональних філій, віддалених офісів та інше), розширення асортименту товарів, кількості клієнтів і постачальників, також сприяють підвищенню складності бізнесів-процесів і стимулюють керівників до пошуку інструментів, необхідних до управління. Четверта причина пов'язана з тим, що із завданнями впровадження автоматизованих систем управління (АСУ) обов'язково стикаються вітчизняні підприємства, які бажають отримати закордонні інвестиції. Часто наявність АСУ є однією з вимог інвестора. П'ятою причиною є вихід вітчизняних фірм на закордонні ринки, на які дуже жорсткі конкуренція і вимоги до якості товарів.

Залежно від конкретної області застосування інформаційні системи можуть дуже сильно відрізнятись за своїми функціями, архітектурі, реалізації. Однак можна виділити, принаймні, дві властивості, які є загальними для всіх інформаційних систем.

По-перше, будь-яка інформаційна система призначена для збору, зберігання і обробки інформації. Тому в основі будь-якої інформаційної системи лежить середу зберігання і доступу до даних. Середу повинна забезпечувати рівень надійності зберігання та ефективність доступу, які відповідають області застосування інформаційної системи.

По-друге, інформаційні системи орієнтуються на кінцевого користувача, наприклад, банківського клерка. Такі користувачі можуть бути дуже далекі від світу комп'ютерів. Для них термінал, персональний комп'ютер або робоча станція являють собою всього лише знаряддя їх власної професійної діяльності. Тому інформаційна

система повинна мати простим, зручним, легко освоюваним інтерфейсом, який повинен надати кінцевому користувачеві всі необхідні для його роботи функції, але в той же час не дати йому можливість виконувати будь зайві дії. Іноді цей інтерфейс може бути графічним з меню, кнопками, підказками і т.д. У будь-якому випадку наявність розвинених інтерфейсних засобів є обов'язковим для будь-якої сучасної інформаційної системи.

Система з Web-інтерфейсом, незалежно від архітектури, спочатку є розподіленою і надає можливість спільної паралельної роботи з інформацією. Повна вартість Web-системи (включаючи витрати на утримання), як правило, виявляється нижче за рахунок централізованого оновлення, відсутності необхідності конфігурації та обслуговування користувача робочих місць. Вся робота з системою відбувається через Web-браузер, який входить в набір стандартних програм будь-якої операційної системи. Ні до програмної, ні до апаратної частини робочого місця користувача не пред'являється ніяких вимог, окрім як організувати мережеве з'єднання з сервером і забезпечити роботу Web-браузера (технологія тонкого клієнта). Сучасний Web-інтерфейс забезпечує прийнятний для інформаційних систем рівень швидкодії та зручності використання.

Реалізація Web-додатків на платформі .NET Framework (технологія ASP.NET):

- скорочує термін розробки за рахунок використання сучасних візуальних засобів і багатого набору базових компонентів;
- озброює його найбільш використовуваними технологіями доступу до даних і службам (ADO.NET, робота з XML, Web-Service), методами роботи з файлами різного формату (включаючи закриті формати MS Office);
- захищає вихідний код системи від небажаного втручання за допомогою його компіляції, здійснює моніторинг та захист від хакерських атак;
- не поступається, а в деяких випадках і перевершує аналогічні рішення за показниками швидкодії;
- надає можливість захисту даних шляхом організації шифрованого каналу або обмеження доступу за адресами користувачів.

Інформаційні Web-системи отримують особливу перевагу там, де є розподіленість мережі користувачів системи. Це може бути філіальна структура організації, наявність віддалених співробітників або партнерів.

Так само Web-система виявляється більш вигідною у разі потреби мобільного доступу, наприклад коли користувач працює з різних ПК, або йому необхідно мати доступ до системи зі свого портативного комп'ютера, але не бути територіально прив'язаним. У даному випадку вигода полягає у відсутності необхідності установки клієнтського програмного забезпечення та організації доступу до системи (за винятком доступу до інтернету).

Якщо Ви стоїте перед вибором, чи підходить інформаційна система, заснована на технологіях Web, саме Вам, Ви порівняєте свої вимоги зі списком вимог яким Web технології задовольняють найкраще:

- потрібно створити єдиний інформаційний простір для віддалених офісів компанії, клієнтів або партнерів (облікові системи, складські системи, електронні вітрини, інформаційні портали);
- робочі місця користувачів оснащені різними платформами (Linux, MacOS, Windows);
- відсутність необхідності в установці і супроводі клієнтських додатків, а також витрат на покупку ліцензій для програмного забезпечення, яке б здійснювало підтримку роботи front-end додатків.

Однак, треба мати на увазі, що є ряд вимог, які незалежно від платформи роблять реалізацію проекту на основі Web-технологій абсолютно неприйнятною:

- наявність високо інтерактивного інтерфейсу (інтерактивні ігри);
- необхідність роботи на стороні клієнта з додатковим обладнанням (пряме отримання зображень з цифрової камери);
- обробка даних без завантаження на сервер;
- реалізація елементів інтерфейсу поза вікна браузера (іконка в системному треї);
- робота з системою у відсутності підключення до сервера.

Сучасна тенденція розвитку програмних систем така, що частка мережеских віддалених, в тому числі і Web-орієнтованих рішень постійно збільшується. Це зумовлено насамперед меншою вартістю експлуатації таких систем, поліпшенням якості каналів зв'язку і зниженням їх вартості.

Web-технології так само не стояли на місці і набули більш багаті можливості побудови інтерактивного інтерфейсу, сучасні середовища швидкої розробки Web-додатків, можливість взаємодіяти з уже існуючими системами і сервісами.

Звичайно, якою б досконалою була технологія, одна вона не в змозі вирішити 100% завдань, і Web тут не виняток. Але якщо ж Web-рішення задовольняє вимогам Вашого проекту, будьте впевнені - ця технологія Вас не розчарує.

Хоча на ринку зараз існує ряд продуктів, покликаних автоматизувати процес управління підприємством, об'єднаних загальним недоліком - всі вони, прагнучи охопити максимальний спектр підприємств, володіють надмірністю функцій і разом з тим не реалізують деяких специфічних функцій, необхідних виключно невеликим підприємствам. Тому було вирішено реалізувати систему, яка б відповідала поставленим вимогам і задовольняла потребам типового малого підприємства.

## 1.2 Аналіз завдання

Завданням є розроблення веб-орієнтованої інформаційної системи управління технічного документообігу для компанії з виготовлення меблів.

Автоматизована інформаційна система – це взаємозв'язана сукупність даних, обладнання, програмних засобів, персоналу, стандартних процедур, які призначені для збору, обробки, розподілу, зберігання, представлення інформації згідно з вимогами, які впливають з цілей організації. Сьогодні, у вік інформації, практично кожна інформаційна система використовує комп'ютерні технології, і тому надалі під інформаційними системами будемо розуміти саме автоматизовані.

Під веб-орієнтованою інформаційною системою розуміємо автоматизовану інформаційну систему з веб-інтерфейсом.

Веб-інтерфейс – це сукупність засобів, за допомогою яких користувач взаємодіє з веб-сайтом або веб-застосунком через браузер. Веб-інтерфейси отримали широке поширення у зв'язку із зростанням популярності всесвітньої павутини і відповідно повсюдного розповсюдження веб-браузерів.

Одним з основних вимог до веб-інтерфейсів є їхній однаковий зовнішній вигляд і однакова функціональність при роботі в різних браузерах.

Основними функціями, які повинна виконувати інформаційна система є:

- формування та збереження документів – замовлень;
- формування та збереження документів з переліком та описом деталей і їх технологічним маршрутом виготовлення для кожного виробу компанії;
- можливість роздрукувати дані документи;
- можливість редагування інформації замовлень, виробів, деталей та їх складових(наприклад, списку доступних матеріалів та кольорів).

Аналізуючи завдання можемо зробити наступні висновки, що розроблювана інформаційна система спрямована на внутрішнє використання компанії її менеджерами та технологами. Виходячи з цього система повинна реалізовувати не лише описані вище функції і зручний інтерфейс для роботи з ними, а й мати модуль автентифікації та авторизації.

Модуль автентифікації та авторизації забезпечить безпеку інформаційної системи, попередить несанкціонований доступ і присіче її використання іншими компаніями.

Також інформаційній системі необхідна база даних, де б могла зберігатися інформація про замовлення, виробу, деталі, матеріали, користувачів, тощо.

## РОЗДІЛ 2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1 Бази даних

Основною частиною ІС на яку опирається її робота є бази даних.

База даних – впорядкований набір логічно взаємопов'язаних даних, що використовуються спільно та призначені для задоволення інформаційних потреб користувачів.

Головне завдання БД - гарантоване збереження значних обсягів інформації та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином, БД складається з двох частин: збереженої інформації та системи керування нею.

Для проектування бази даних використаємо MySQL Server, а для роботи з базою даних SQL та MySQL Workbench.

SQL – декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних і її модифікації, системи контролю за доступом до бази даних. Сам по собі SQL не є ні системою керування базами даних, ні окремим програмним продуктом. Не будучи мовою програмування в тому розумінні, як C або Pascal, SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати в якості інструкцій для керування даними. Стандарт SQL, крім того, вміщує функції для визначення зміни, перевірки і захисту даних.

MySQL Server – компактний багатонитевий сервер баз даних. Характеризується великою швидкістю, стійкістю і простотою використання.

Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;

- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL – одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL Workbench – інструмент для візуального проектування баз даних, що інтегрує проектування, моделювання, створення й експлуатацію БД в єдине безшовне оточення для системи баз даних MySQL. Є наступником DBDesigner 4 з FabForce.

Можливості програми:

- Дозволяє наочно представити модель бази даних в графічному вигляді;
- Наочний і функціональний механізм установки зв'язків між таблицями, в тому числі «багато до багатьох» із створенням таблиці зв'язків;
- Reverse Engineering — відновлення структури таблиць з вже існуючої на сервері БД (зв'язки відновлюються в InnoDB, при використанні MyISAM зв'язки необхідно встановлювати вручну);
- Зручний редактор SQL запитів, що дозволяє відразу ж відправляти їх серверові і отримати відповідь у вигляді таблиці;
- Можливість редагування даних у таблиці в візуальному режимі.



## 2.2 Інтерфейс користувача

Інтерфейс користувача – сукупність засобів для обробки та відображення інформації, максимально пристосованих для зручності користувача; у графічних системах інтерфейс користувача реалізовується багатовіконним режимом, змінами кольору, розміру, видимості (прозорість, напівпрозорість, невидимість) вікон, їхнім розташуванням, сортуванням елементів вікон, гнучкими налаштуваннями як самих вікон, так і окремих їхніх елементів, доступністю багатокористувацьких налаштувань.

Різновидом інтерфейсів користувача є веб-інтерфейси. Веб-інтерфейси зручні тим, що дають можливість вести спільну роботу співробітникам, які не перебувають в одному офісі (наприклад, веб-інтерфейси часто використовуються для заповнення різних баз даних або публікації матеріалів в інтернет – ЗМІ).

Хорошим прикладом використання веб-інтерфейсу є Вікіпедія: практично весь вміст вільної всесвітньої енциклопедії створений і доданий на сторінки сайту за допомогою веб-інтерфейсу.

Веб-інтерфейс дає можливість універсального віддаленого доступу до служб та пристроїв, у цьому технології практично нема альтернатив. Але водночас, оскільки такий інтерфейс доступний усім, постають серйозні питання забезпечення безпеки, зокрема автентифікація та авторизація користувачів, шифрування переданих даних від сторонніх очей, модерація вмісту тощо.

Основними технологіями створення веб-інтерфейсів є:

- HTML
- CSS
- JavaScript

HTML (англ. HyperText Markup Language — Мова розмітки гіпертекстових документів) – стандартна моварозмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

HTML є похідною мовою від SGML, успадкувавши від неї визначення типу документу та ідеологію структурної розмітки тексту.

HTML впроваджує засоби для:

- створення структурованого документу шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- отримання інформації із Всесвітньої мережі через гіперпосилання;
- створення інтерактивних форм;
- включення зображень, звуку, відео, та інших об'єктів до тексту.

Каскадні таблиці стилів (англ. Cascading Style Sheets або скорочено CSS) - спеціальна мова, що використовується для відображення сторінок, написаних мовами розмітки даних.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори тощо. CSS також дозволяє адаптувати контент до різних умов відображення (на екрані

монітора, мобільного пристрою (КПК), у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерів та ін.).

JavaScript – назва реалізації стандарту мови програмування ECMAScript компанії Netscape, базується на принципах прототипного програмування. Найпоширеніше і найвідоміше застосування мови – написання сценаріїв для веб-сторінок, але, також, використовується для впровадження сценаріїв керування об'єктами вбудованими в інші програми.

JavaScript розроблений у компанії Netscape. На сьогоднішній день підтримується більшістю браузерів. Текст програми включається безпосередньо в HTML-документ і інтерпретується самим браузером (точніше, вбудованим у браузер рушієм JavaScript). Застосовується в основному для часткової автоматизації обробки і маніпуляції даними, які використовує сторінка.

Також для виконання однієї з основних вимог до веб-інтерфейсів - однакового зовнішнього вигляду і однакової функціональності при роботі в різних браузерах варто використовувати бібліотеки для javascript'a та css'a.

Бібліотеками, які використовуватимуться є:

- jQuery;
- Bootstrap.

jQuery – популярна JavaScript-бібліотека з відкритим сирцевим кодом. Вона була представлена у січні 2006 року у BarCamp NYC Джоном Ресігом (John Resig). Згідно з дослідженнями організації W3Techs, JQuery використовується понад половиною від мільйона найвідвідуваніших сайтів. jQuery є найпопулярнішою бібліотекою JavaScript, яка посилено використовується на сьогоднішній день.

jQuery є вільним програмним забезпеченням під ліцензією MIT.

Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій, і розробки AJAX-застосунків. jQuery також надає можливості для розробників, для створення плагінів у верхній частині бібліотеки JavaScript. Використовуючи ці об'єкти, розробники можуть створювати абстракції для низькорівневої взаємодії та

створювати анімацію для ефектів високого рівня. Це сприяє створенню потужних і динамічних веб-сторінок.

Bootstrap – це набір інструментів від Twitter (відноситься до класу інструментів: CSS-фреймворк), створений для полегшення розробки web застосунків та сайтів. Він включає CSS та HTML для типографії, форм, кнопок, таблиць, сіток, навігації тощо, а також додаткові розширення JavaScript.

Репозиторій з фреймворком є одним з найбільш популярних на GitHub і, серед інших, його використовують NASA і MSNBC.

Bootstrap використовує найсучасніші напрацювання в області CSS та HTML, тому необхідно бути уважним при підтримці старих браузерів.

Основні інструменти Bootstrap:

- сітки – наперед задані розміри колонок, які можна відразу ж використовувати, наприклад ширина колонки 90px відноситься до класу `.span2`, який ми можемо використовувати в CSS описі документа;
- шаблони – фіксований або гумовий шаблон документа;
- типографіка – опис шрифтів, визначення деяких класів для шрифтів таких як код, цитати тощо;
- медіа – представляє певне управління зображеннями та відео;
- таблиці – засоби оформлення таблиць, дозволяє додавати функціональність сортування;
- форми – класи для оформлення не тільки форм але і деяких подій;
- навігація – класи оформлення для табів, вкладок, сторінок, меню і тулбара;
- алерт – оформлення діалогових вікон, підказок і спливаючих вікон.

## 2.3 Серверна сторона інформаційної системи

Серверна частина інформаційної системи, а саме логіка функціонування реалізується з використання технологій ASP.NET та ADO.NET і мови програмування C#.

ASP.NET – технологія створення веб-застосунків і веб-сервісів від компанії Майкрософт. У цей час останньою версією цієї технології є ASP.NET 7 .

ASP.NET має перевагу у швидкості в порівнянні зі скриптовими технологіями, тому що при першому зверненні код компілюється і поміщається в спеціальний кеш, і згодом тільки виконується, не вимагаючи витрат часу на парсинг, оптимізацію, і т. д.

Коли платформа ASP.NET була випущена вперше, від попередніх продуктів Microsoft і конкуруючих платформ її відрізняли сім ключових фактів.

Перший факт: ASP.NET інтегрується с. NET Framework.

Другий факт: код ASP.NET компілюється, а не інтерпретується.

Третій факт: в ASP.NET підтримується безліч мов програмування.

Четвертий факт: ASP.NET обслуговується середовищем CLR.

Шостий факт: ASP.NET підтримує всі браузерери.

Сьомий факт: ASP.NET дозволяє легко виконувати розгортання і конфігурацію.

Microsoft .NET (читається дот-нет) –є сумісність служб, написаних різними мовами. Хоча ця можливість рекламується Microsoft як перевага .NET, платформа Java має таку саму можливість. .NET поділяється на дві основні частини – середовище виконання (по суті віртуальна машина) та інструментарій розробки.

Середовище розробки .NET створює байт-код, призначений для виконання віртуальною машиною. Вхідна мова цієї машини в .NET називається CIL (Common Intermediate Language), також відома як MSIL (Microsoft Intermediate Language), або просто IL. Застосування байт-кода дозволяє отримати крос-платформовість на рівні скомпільованого проекту (в термінах .NET: збірка), а не на рівні сирцевого тексту,

як, наприклад, в C. Перед запуском збірки в середовищі виконання (CLR) байт-код перетворюється вбудованим в середовище JIT-компілятором (just in time, компіляція на льоту) в машинні коди цільового процесора. Середовище CLR є реалізацією специфікації CLI (англ. Common Language Infrastructure), специфікації загальномовної інфраструктури, компанією Microsoft.

Віртуальна машина CLR дозволяє програмістам забути про багато деталей про конкретний процесор, на якому виконуватиметься програма. CLR також забезпечує такі важливі служби як:

- управління пам'яттю
- управління потоками
- обробка винятків
- збірка сміття
- безпека виконання

ADO.NET (ActiveX Data Objects .NET) — це набір бібліотек, що поставляється з Microsoft .NET Framework і призначений для взаємодії з різними сховищами даних з .NET-застосунків. Бібліотеки ADO.NET включають класи для приєднання до джерела даних, виконання запитів і обробки їхніх результатів. Крім того, ADO.NET можна використовувати в якості надійного, ієрархічно організованого, відокремленого кешу даних для автономної роботи з даними.

ADO.NET володіє багатьма перевагами порівняно з іншими технологіями доступу до даних. Основні з них це:

- Підтримка XML. ADO також підтримує XML, але не буде так само ефективно обробляти XML-дані, як це робить ADO.NET, оскільки ADO.NET створювався з врахуванням XML, а ADO-ні.
- Простота модифікації. Протягом терміну служби системи в неї можна вносити незначні зміни, однак спроби провести архітектурні зміни трапляються рідко, через виняткову складність завдання.
- Простота програмування. Компоненти даних ADO.NET в Visual Studio інкапсулюють функціональні можливості доступу до даних різними

способами, що допомагає розробляти програмні продукти значно швидше і з меншою кількістю помилок.

- **Продуктивність.** Для невідключених застосунків набори даних ADO.NET дають вигоду в продуктивності в порівнянні з невідключеними наборами записів ADO. Передача невідключеного набору записів між рівнями за допомогою COM-упаковки може призвести до великої витрати обчислювальних ресурсів, тому що значення в наборі записів перетворюються до типів даних, відомих COM. У ADO.NET таке перетворення типів даних не потрібно.

Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Перейнявши багато що від своїх попередників – мов C++, Delphi, Модула і Smalltalk – C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад множинне спадкування класів (на відміну від C++).

C# розроблялась як мова програмування прикладного рівня для CLR і тому вона залежить, перш за все, від можливостей самої CLR. Це стосується, перш за все, системи типів C#. Присутність або відсутність тих або інших виразних особливостей мови диктується тим, чи може конкретна мовна особливість бути трансльована у відповідні конструкції CLR. Так, з розвитком CLR від версії 1.1 до 2.0 значно збагатився і сам C#; подібної взаємодії слід чекати і надалі. (Проте ця закономірність буде порушена з виходом C# 3.0, що є розширеннями мови, що не спираються на розширення платформи .NET.) CLR надає C#, як і всім іншим .NET-орієнтованим мовам, багато можливостей, яких позбавлені «класичні» мови програмування. Наприклад, збірка сміття не реалізована в самому C#, а проводиться CLR для програм, написаних на C# точно так, як і це робиться для програм на VB.NET, J# тощо.

## РОЗДІЛ 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

### 3.1 Реляційна алгебра

В маніпуляційній складовій визначаються два базові механізми маніпулювання реляційними даними: основана на теорії множин реляційна алгебра і реляційне числення, що базується на математичній логіці (вірніше, на численні предикатів першого порядку). Всі ці механізми володіють однією важливою властивістю: вони замкнені відносно поняття відношення. Це означає, що вирази реляційної алгебри і формули реляційного числення визначаються над відношеннями реляційни БД і результатами обчислень також є відношення. Як наслідок, будь-який вираз або формула можуть інтерпретуватися як відношення, що дозволяє використовувати їх в інших виразах або формулах. Як ми побачимо, алгебра і числення володіє значною виразною потужністю: дуже складні запити до бази даних можуть бути виражені за допомогою одного виразу реляційної алгебри або однією формули реляційного числення. З цієї причини саме ці механізми включені до реляційної моделі. Відомо, що механізми реляційної алгебри і реляційного числення еквівалентні, тобто для будь-якого допустимого виразу реляційної алгебри можна побудувати еквівалентну (тобто ту, яка дає той самий результат) формулу реляційного числення і навпаки. Через це, в конкретній ситуації для перевірки степені реляційності деякої мови БД можна використовувати будь-який з цих механізмів. Зауважимо, що дуже рідко алгебра або числення приймаються в якості повної основи деякої мови БД. Звичайна (як, наприклад, у випадку мови SQL) мова основана на деякій суміші алгебраїчних і логічних конструкцій. Тому, знання алгебраїчних і логічних основ мов баз даних часто є корисним на практиці.

Основна ідея реляційної алгебри полягає в тому, що оскільки відношення є множинами, засоби маніпулювання відношеннями можуть базуватися на



традиційних теоретико-множинних операціях, додатково до деяких спеціальних операцій, специфічних для баз даних. Набір основних алгебраїчних операцій, запропонований Коддом, складається з восьми операцій, які діляться на два класи, – теоретико-множинні операції і спеціальні реляційні операції. До складу теоретико-множинних операцій входять операції:

- об'єднання відношень;
- перетин відношень;
- взяття різниці відношень;
- Декартовий добуток відношень.

Спеціальні реляційні операції включають:

- обмеження відношення;
- проекція відношення;
- з'єднання відношень;
- ділення відношень.

Крім того, до складу алгебри входить операція присвоювання, що дозволяє зберігати в базі даних результати обчислення алгебраїчних виразів, і операція перейменування атрибутів, яка дає можливість коректно сформулювати заголовок (схему) результуючого відношення.

Загальна інтерпретація реляційних операцій

Якщо не вдаватися в деякі тонкощі, майже всі операції з запропонованих вище володіють очевидною і простою інтерпретацією:

- Операція перейменування дає відношення, тіло якого співпадає з тілом операнда, але імена атрибутів змінені.
- Операція присвоювання дозволяє зберегти результат обчислення реляційного виразу в існуючому відношенні БД. Оскільки результатом будь-якої реляційної операції (крім операції присвоювання) є деяке відношення, можна відобразити реляційні вирази, в яких замість відношення-операнда деякої реляційної операції знаходиться вкладений реляційний вираз.

Замкненість реляційної алгебри

Кожне відношення характеризується схемою (або заголовком) і набором кортежів (або тілом). Тому, якщо реалізовувати алгебру, операції якої замкнені відносно поняття відношення, кожна операція повинна породжувати відношення в повній мірі, тобто воно повинно володіти і тілом, і заголовком. Лише в цьому випадку буде можливість будувати вкладені вирази. Заголовок відношення є множиною пар  $\langle \text{ім'я-атрибута}, \text{ім'я-домена} \rangle$ . Якщо зробити загальний огляд реляційних операцій, можна побачити, що домени атрибутів результуючого відношення однозначно визначаються доменами відношень-результатів.

#### Особливості теоретико-множинних операцій реляційної алгебри

Хоча в основі теоретико-множинної частини реляційної алгебри лежить класична теорія множин, відповідні операції реляційної алгебри володіють деякими особливостями.

Почнемо з операції об'єднання (вертикальне) (все, що ми говоритимемо з приводу об'єднання переноситься на операції перетину і різниці).

- При виконанні операції об'єднання двох відношень одержується відношення, що містять всі кортежі, що входять хоча б в одне з відношень-операндів.

- Операція перетину двох відношень дозволяє одержати відношення, що включає всі кортежі, які входять до обох відношень-операндів.

- Відношення, що є різницею двох відношень, включає всі кортежі, що входять у відношення – перший операнд; такі, що жоден з них не входить у відношення, яке є другим операндом.

Результат операцій: відношення. При виконанні цієї операції необхідно пам'ятати про сумісність відношень по об'єднанню: два відношення сумісні по об'єднанню в тому і лише в тому випадку, коли володіють однаковими заголовками. Більш точно, це означає, що в заголовках обох відношень міститься один і той самий набір імен атрибутів, і одночасно атрибути визначені на одному і тому самому домені. Якщо два атрибути сумісні по об'єднанню, то при звичайному виконанні над ними операцій об'єднання:

SELECT \* FROM A UNION SELECT \* FROM B,

перетину

SELECT \* FROM A INTERSECT SELECT \* FROM B,

і взяття різниці

SELECT \* FROM A EXCEPT SELECT \* FROM B

в мові SQL результатом операції є відношенням є відношення з коректно визначеним заголовком, що співпадає з заголовком кожного з відношень операндів. Зауважимо, що залучення до складу операцій реляційної алгебри трьох операцій об'єднання, перетину і різниці є надлишковим, оскільки будь-яка з цих операцій виражається через дві інші. Тим не менш, Кодд в свій час вирішив включити всі три операції, виходячи з інтуїтивних потреб потенціального користувача системи реляційних БД, далекого від математики.

Операція взяття Декартового (або прямого) добутку двох відношень.

- При виконанні прямого добутку двох відношень одержується відношення, кортежі якого є конкатенацією (сцепленням) кортежів першого і другого операндів (A CROSS JOIN B).

В реляційній алгебрі виконується спеціалізована форма операції взяття прямого добутку – розширений прямий добуток відношень. При взятті розширеного прямого добутку двох відношень елементом результуючого відношення є кортеж, що є конкатенацією (або злиттям) одного кортежа першого відношення і одного кортежа другого відношення. Для того щоб одержати результуюче відношення з коректно сформованим заголовком потрібно притримуватись поняття сумісності по взяттю розширеного прямого добутку. Два відношення сумісні при цьому в тому і лише в тому випадку, якщо множини імен атрибутів цих відношень не перетинаються, тобто немає атрибутів з однаковими назвами. Будь-які два відношення можна зробити сумісними по взяттю прямого добутку шляхом застосування операції перейменування до одного з цих відношень. Слід зауважити, що операція взяття прямого добутку не є виправданою на практиці. По-перше, потужність її результату досить велика навіть при допустимих потужностях

операндів, а по-друге, результат операції не є більш інформативним, ніж взяті в сукупності операнди.

- Вибірка (обмеження). Результатом обмеження відношення за деякою умовою є відношення, що містить кортежі відношення-операнда, яке задовольняє цій умові.

Нехай маємо відношення  $A$  з атрибутами  $X$  і  $Y$ ., а символ  $\Theta$  означає будь-який оператор порівняння ( $=, \neq, >, \geq$  та ін.), такий що умова  $X \Theta Y$  при заданих  $X$  та  $Y$  дає значення істина або хибність. Тоді  $\Theta$ -вибіркою з відношення  $A$  по атрибутам  $X$  та  $Y$  називається відношення, що має той самий заголовок, що і відношення  $A$  і тіло, що містить множину всіх кортежів  $t$  відношення  $A$ , для яких перевірка умови  $X \Theta Y$  дає значення істина. Умова найчастіше задається за допомогою ключового слова **WHERE**.

```
SELECT *  
FROM S  
WHERE S#='S1';
```

З використанням цих визначень можна використовувати операції обмеження, в яких умова обмеження є довільною булевим виразом, складеним з простих умов з використанням логічних зв'язок **AND**, **OR** і **AND** та дужок. На інтуїтивному рівні операцію обмеження краще за все уявити як взяття деякої «горизонтальної» вирізки з відношення-операнда.

Проекція.

- При виконанні проекції відношення на заданий набір його атрибутів одержується відношення, кортежі якого одержуються шляхом взяття відповідних значень з заданих стовпців кортежів відношення-операнда.

Нехай задане відношення  $A$  з атрибутами  $X, Y, \dots, Z$ . Тоді проекцією відношення  $A$  за атрибутами  $X, Y, \dots, Z$  ( $A \{X, Y, \dots, Z\}$ ) називається відношення, що задовольняє наступним вимогам.

– Його заголовок одержується з заголовку  $A$  шляхом видалення з нього всіх атрибутів, що не входять до множини  $A \{X, Y, \dots, Z\}$ .

– Його тіло містить множину всіх кортежів виду  $\{X:x, Y:y, \dots, Z:z\}$ , таких, що  $x, y, z$  у відношенні  $A$  є значеннями відповідних атрибутів (`SELECT DISINCT X, Y, ... Z FROM A`).

```
SELECT S.S#, S.SNAME
```

```
FROM S
```

Таким чином, за допомогою проекції можна створити вертикальну підмножину заданого відношення.

Спеціальні реляційні операції

Операція з'єднання відношень (горизонтальна)

- При з'єднанні двох відношень за деякою умовою утворюється результуюче відношення, кортежі якого є конкатенацією кортежів першого і другого відношень і задовольняють цій умові.

Загальна операція з'єднання (яку також називають з'єднанням за умовою) потребує наявності двох операндів – відношень, що з'єднуються і третього операнда – простої умови. Нехай з'єднуються відношення  $A$  і  $B$ , що не мають спільних імен атрибутів. Тоді  $\Theta$ -з'єднання відношення  $A$  за атрибутом  $X$  з відношенням  $B$  за атрибутом  $Y$  називається результат наступного виразу

```
(A JOIN B) WHERE X $\Theta$ Y;
```

```
S JOIN P USING CITY
```

Є часткові випадки з'єднання – просте і еквіз'єднання. Операція природного з'єднання застосовується до пари відношень  $A$  і  $B$ , що володіють спільним атрибутом  $c$  (тобто, атрибутом з одним і тим самим іменем і визначеним на тому самому домені). Нехай відношення  $A$  і  $B$  мають заголовки

$$\{X_1, X_2, \dots, X_m, Y_1, Y_2, \dots, Y_n\}$$

та

$$\{Y_1, Y_2, \dots, Y_n, Z_1, Z_2, \dots, Z_p\}$$

відповідно, тобто атрибути  $Y_1, Y_2, \dots, Y_n$  (і лише вони) – спільні для двох цих відношень,  $X_1, X_2, \dots, X_m$  – решта атрибутів відношення  $A$  і  $Z_1, Z_2, \dots, Z_p$  – решта атрибутів для  $B$ . далі ми будемо їх називати як  $X, Y, Z$ . Тоді операція природного

з'єднання відношень  $A$  і  $B$  називається відношення з заголовком  $\{X, Y, Z\}$  і тілом, що містить множину всіх кортежів вигляду  $\{X:x, Y:y, Z:z\}$ . Таке з'єднання необов'язково виконується по значенню первинного і відповідного зовнішнього ключа.

### S NATURAL JOIN P

Операція з'єднання називається операцією еквіз'єднання, якщо умова з'єднання має вигляд  $(a = b)$ , де  $a$  і  $b$  – атрибути різних операндів з'єднання. Цей випадок важливий тому, що (а) він часто зустрічається на практиці, і (б) для нього існують ефективні алгоритми реалізації.

### Операція ділення відношень

- У операції реляційного ділення два операнди – бінарне і унарне відношення. Результуюче відношення складається з одноатрибутних кортежів, що містять значення першого атрибута кортежів першого операнда таких, що множина значень другого атрибута (при фіксованому значенні першого атрибута) співпадає з множиною значень другого операнда.

Ця операція є найменш очевидною з усіх операцій реляційної алгебри і тому потребує детального пояснення. Нехай задані два відношення –  $A$  з заголовком  $\{a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m\}$  і  $B$  з заголовком  $\{b_1, b_2, \dots, b_m\}$ . Будемо вважати, що атрибут  $b_i$  відношення  $A$  і атрибут  $b_i$  відношення  $B$  не лише володіють одним і тим самим іменем, але й визначені на одному і тому самому домені. Назвемо множину атрибутів  $\{a_j\}$  складовим атрибутом  $a$ , а множину атрибутів  $\{b_j\}$  – складовим атрибутом  $b$ . Після цього будемо говорити про реляційне ділення бінарного відношення  $A$  ( $a, b$ ) на унарне відношення  $B(b)$ . Результатом ділення  $A$  на  $B$  є унарне відношення  $S(a)$ , що складається з кортежів  $v$  таких, що у відношенні  $A$  є кортежі  $\langle v, w \rangle$  такі, що множина значень  $\{w\}$  включає множину значень атрибута  $b$  у відношенні  $B$ . Припустимо, що в базі даних робітників підтримуються два відношення: РОБІТНИКИ (ВІД\_НОМЕР, ІМЯ) і ІМЕНА (ІМЯ), причому унарне відношення ІМЕНА містить всі прізвища, які мають робітники організації. Тоді після виконання операції реляційного ділення відношення РОБІТНИКИ на

відношенне ІМЕНА буде отримане унарне відношення, що міститиме номери відділів, робітники яких мають всі можливі в цій організації імена.

### 3.2 Реляційне числення

Розглянемо запит «Вибрати номери постачальників і назви міст, в яких знаходяться постачальники деталі з номером P2». Якщо б для формулювання такого запиту використовувалася реляційна алгебра, ми б отримали алгебраїчний вираз, який читався б, наприклад, наступним чином:

- виконати з'єднання відношення S і SP за атрибутом S#;
- вибрати з результату цього з'єднання кортежі з номером деталі 'P2';
- спроектувати результат попередньої операції за атрибутами S# і CITY

Ми чітко сформулювали послідовність кроків виконання запиту, кожен з яких відповідає одній реляційній операції. Якщо ж сформулювати той самий запит з використанням реляційного числення, про яке йтиметься зараз, то ми одержимо формулу, яку можна було б прочитати, наприклад, наступним чином: Видати S# і CITY для робітників таких, що постачають деталь з номером P2. У другому способі формулювання ми вказуємо лише характеристики результуючого відношення, але нічого не говоримо про спосіб його формування. В цьому випадку система повинна сама вирішити, які операції і в якому порядку потрібно виконати над відношеннями. Зазвичай говорять, що алгебраїчне формулювання є процедурним, тобто таким, що задає правила виконання запиту, а логічне – описовим (або декларативним), оскільки воно лишень описує властивості бажаного результату. Як ми зазначали, насправді ці два механізми еквівалентні і існують не дуже складні правила перетворення одного формалізму в інший.

Кортежні змінні і правильно побудовані формули.

Реляційне числення є прикладною гілкою формального механізму числення предикатів першого порядку. Базисними поняттями числення є поняття змінної з

визначеною для неї областю допустимих значень і поняття правильно побудовано формули, яка спирається на змінні, предикати і квантори. В залежності від того, що є областю визначення змінної, розрізняють числення кортежів і числення доменів.

В численні кортежів областями визначення змінних є відношення бази даних, тобто допустимим значенням кожної змінної є кортеж деякого відношення.

В численні доменів областями визначення змінних є домени, на яких визначені атрибути відношень бази даних, тобто допустимим значенням кожної змінної є значення деякого домена.

Для визначення кортежної змінної, областю визначення якої є деяке базове відношення потрібно вжити конструкцію (наприклад використану в операторі SELECT):

```
Select PX.COLOR, PX.CITY
FROM P AS PX
WHERE PX.CITY<>'Paris'
AND PX.WEIGHT>10.0;
```

З цього визначення випливає, що в будь-який момент часу змінна PX представлятиме змінну кортежа, областю значень якої буде поточне значення відношення P. При використанні кортежних змінних в формулах можна посилатися на значення атрибута змінної (це аналогічно тому, як, наприклад, при програмуванні на мові Сі можна посилатися на значення поля структурної змінної).

```
PX.CITY
```

В мові SQL допускається неявне звернення до змінних кортежів, що дозволяє переписати наш запит у наступному вигляді:

```
Select P.COLOR, P.CITY
FROM P
WHERE P.CITY<>'Paris'
AND P.WEIGHT>10.0;
```

Основа ідея полягає в тому, щоб дозволити використовувати ім'я відношення для позначення неявної змінної кортежа, областю значення якої буде дана таблиця.



Правильно побудовані формули (WFF – Well-Formed Formula) служать для виразу умов, що накладаються на кортежні змінні. Основою WFF є прості порівняння, які є операціями порівняння скалярних значень (значень атрибутів змінних або літерально заданих констант). Наприклад, конструкція

«A.CITY=B.CITY»

є простим порівнянням. Більш складні варіанти WFF будуються за допомогою логічних зв'язок NOT, AND, OR. Нарешті, допускається побудова WFF за допомогою кванторів. Якщо form – це WFF, в якій бере участь змінна var, то конструкції EXISTS var (form) і FORALL var (form) є wff. Змінні, що входять до WFF, можуть бути вільними або зв'язаними. Всі змінні, що входять до WFF, при побудові якого не використовується квантор, є вільними. Вказати імена постачальників деталі з номером 'P2':

```
SELECT DISTINCT S.SNAME
FROM S
WHERE EXISTS
(SELECT*
FROM SP
WHERE SP.S#=S.S#
AND SP.P#='P2');
```

SQL EXISTS (SELECT ... FROM) буде мати значення істина тоді і лише тоді, коли результат обчислення виразу SELECT ... FROM буде непустим.

SQL не містить деякої безпосередньої підтримки універсального квантора FORALL; відповідно запити типу «ДЛЯ ВСІХ» звичайно виражаються через заперечення квантора існування, як у наступному прикладі вибрати імена постачальників, які не постачають деталь з номером 'P2':

```
SELECT DISTINCT S.NAME
FROM S
WHERE NOT EXISTS
(SELECT *
```

```
FROM SP
SP.S#=S.S#,
AND SP.P#='P2');
```

В численні доменів областю визначення змінних є не відношення, а домени. По відношенню до БД Постачання деталей можна говорити, наприклад, про домени-змінні S# (значення – допустимі номери постачальників) або SNAME (значення – допустимі імена постачальників). Основною формальною відмінністю числення доменів від числення кортежів є наявність додаткового набору предикатів, що дозволяють виражати так звані умови входження. Якщо R – це n-арне відношення з атрибутами  $a_1, a_2, \dots, a_n$ , тоді умова членства має вигляд  $R(a_{i1}:v_{i1}, a_{i2}:v_{i2}, \dots, a_{im}:v_{im})$  ( $m \leq n$ ), де  $v_{ij}$  – це або літеральна константа, або ім'я доменної змінної. Умова членства приймає значення true в тому і лише тому випадку, якщо у відношенні R існує кортеж, що містить вказані значення вказаних атрибутів.

## РОЗДІЛ 4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до ПЗ

Програмне забезпечення необхідне для реалізації завдання, яке буде використовуватися, наведене в таблиці 4.1.1

Таблиця 4.1.1

<b>Програмне забезпечення</b>	<b>Версія</b>
Microsoft Visual Studio	2019
MySQL Server	5.5
MySQL Workbench	6.06
MySQL Connector Net	7.0
Adobe Photoshop	14

Вибір даного програмного забезпечення продиктований технологіями, які використовуватимуться для проектування та реалізації інформаційної системи, а саме:

- ASP.NET;
- ADO.NET;
- MySQL.

А програмне забезпечення необхідне для повноцінного функціонування інформаційної системи наводиться в таблиці 4.1.2.

Таблиця 4.1.2

<b>Програмне забезпечення</b>	<b>Версія</b>
Windows OS	10
Opera / Firefox / Chrome / IE	21 / 30 / 35 / 11

## 4.2 Проектування та реалізація бази даних

Для проектування баз даних було розроблено чи мало правил, настанов та рекомендацій щодо структури таблиць, їх взаємозв'язків, тощо. Їх ціллю зменшити розмір БД та підвищити безпеку та цілісність інформації.

Проте в сьогоdnішніх умовах, коли постійна пам'ять практично безплатна, а доступ до інтернету носить постійний характер, на першу роль виходять інші характеристики, а саме швидкість обробки інформації інформаційною системою та спрощення процесів супроводу та оновлення інформаційних систем.

Тому інформаційній системі для забезпечення передбаченого функціоналу необхідна база даних, яка включатиме такі таблиці:

- таблиця кольорів – для зберігання можливих кольорів виробів;
- таблиця матеріалів – для зберігання можливих матеріалів деталей;
- таблиця замовлень – для зберігання інформації про дату замовлення, замовника, час виконання;
- таблиця взаємозв'язку замовлень та виробів – для формування таблиць виробів для кожного із замовлень;
- таблиця деталей – для зберігання характеристик деталей з яких складаються вироби;
- таблиця виробів – для зберігання характеристик виробів;
- таблиця взаємозв'язку деталей та виробів – для формування таблиць деталей для кожного із виробів;
- таблиця технологічних процесів – для зберігання технологічних карт для кожної деталі;
- таблиця користувачів ІС – для зберігання логінів та паролів персоналу підприємства;
- таблиця верстатів – для зберігання інформації про верстатів

- таблиця операцій верстатів - для зберігання інформації про операції, що можна здійснювати на кожному із верстатів;

Діаграма, що зображена на рисунку 4.2.1 відображує структуру та взаємозв'язок спроектованих таблиць для бази даних інформаційної системи.

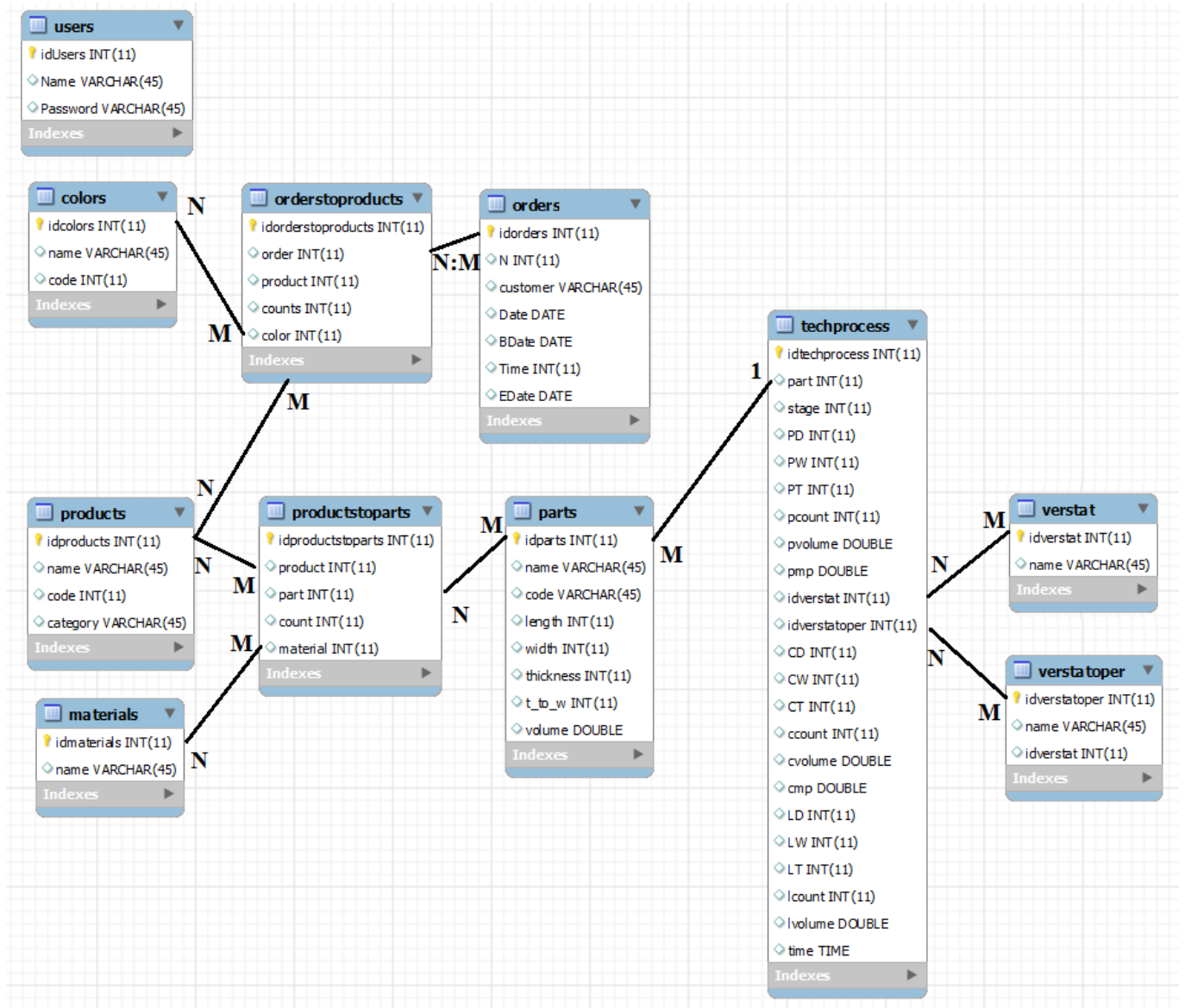


Рисунок 4.2.1 – Діаграма БД

Створення таблиці кольорів відбувається за допомогою наступного розроблено SQL коду:

```
CREATE TABLE `colors` (
  `idcolors` int(11) NOT NULL,
  `name` varchar(45) DEFAULT NULL,
```

```
`code` int(11) DEFAULT NULL,  
PRIMARY KEY (`idcolors`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Створення таблиці матеріалів відбувається за допомогою наступного розроблено SQL коду:

```
CREATE TABLE `materials` (  
`idmaterials` int(11) NOT NULL,  
`name` varchar(45) DEFAULT NULL,  
PRIMARY KEY (`idmaterials`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Створення таблиці замовлень відбувається за допомогою наступного розроблено SQL коду:

```
CREATE TABLE `orders` (  
`idorders` int(11) NOT NULL AUTO_INCREMENT,  
`N` int(11) DEFAULT NULL,  
`customer` varchar(45) DEFAULT NULL,  
`Date` date DEFAULT NULL,  
`BDate` date DEFAULT NULL,  
`Time` int(11) DEFAULT NULL,  
`EDate` date DEFAULT NULL,  
PRIMARY KEY (`idorders`)  
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```

Створення таблиці взаємозв'язку замовлень та виробів відбувається за допомогою наступного розроблено SQL коду:

```
CREATE TABLE `orderstoproducts` (  
`idorderstoproducts` int(11) NOT NULL AUTO_INCREMENT,  
`order` int(11) DEFAULT NULL,  
`product` int(11) DEFAULT NULL,  
`counts` int(11) DEFAULT NULL,
```

```
`color` int(11) DEFAULT NULL,  
PRIMARY KEY (`idorderstoproducts`)  
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8;
```

Створення таблиці деталей відбувається за допомогою наступного розроблено SQL коду:

```
CREATE TABLE `parts` (  
  `idparts` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(45) DEFAULT NULL,  
  `code` varchar(45) DEFAULT NULL,  
  `length` int(11) DEFAULT NULL,  
  `width` int(11) DEFAULT NULL,  
  `thickness` int(11) DEFAULT NULL,  
  `t_to_w` int(11) DEFAULT NULL,  
  `volume` double DEFAULT NULL,  
  PRIMARY KEY (`idparts`)  
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8;
```

Створення таблиці виробів відбувається за допомогою наступного розроблено SQL коду:

```
CREATE TABLE `products` (  
  `idproducts` int(11) NOT NULL,  
  `name` varchar(45) DEFAULT NULL,  
  `code` int(11) DEFAULT NULL,  
  `category` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`idproducts`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Створення таблиці взаємозв'язку деталей та виробів відбувається за допомогою наступного розроблено SQL коду:

```
CREATE TABLE `productstoparts` (  
  `idproductstoparts` int(11) NOT NULL AUTO_INCREMENT,
```

```

`product` int(11) DEFAULT NULL,
`part` int(11) DEFAULT NULL,
`count` int(11) DEFAULT NULL,
`material` int(11) DEFAULT NULL,
PRIMARY KEY (`idproductstoparts`)
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8;

```

Створення таблиці технологічних процесів відбувається за допомогою наступного розроблено SQL коду:

```

CREATE TABLE `techprocess` (
`idtechprocess` int(11) NOT NULL AUTO_INCREMENT,
`part` int(11) DEFAULT NULL,
`stage` int(11) DEFAULT NULL,
`PD` int(11) DEFAULT NULL,
`PW` int(11) DEFAULT NULL,
`PT` int(11) DEFAULT NULL,
`pcount` int(11) DEFAULT NULL,
`pvolume` double DEFAULT NULL,
`pmp` double DEFAULT NULL,
`idverstat` int(11) DEFAULT NULL,
`idverstatoper` int(11) DEFAULT NULL,
`CD` int(11) DEFAULT NULL,
`CW` int(11) DEFAULT NULL,
`CT` int(11) DEFAULT NULL,
`ccount` int(11) DEFAULT NULL,
`cvolume` double DEFAULT NULL,
`cmp` double DEFAULT NULL,
`LD` int(11) DEFAULT NULL,
`LW` int(11) DEFAULT NULL,
`LT` int(11) DEFAULT NULL,

```



```

`lcount` int(11) DEFAULT NULL,
`lvolume` double DEFAULT NULL,
`time` time DEFAULT NULL,
PRIMARY KEY (`idtechprocess`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;

```

Створення таблиці користувачів ІС відбувається за допомогою наступного розроблено SQL коду:

```

CREATE TABLE `users` (
`idUser` int(11) NOT NULL,
`Name` varchar(45) DEFAULT NULL,
`Password` varchar(45) DEFAULT NULL,
PRIMARY KEY (`idUser`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Створення таблиці верстатів відбувається за допомогою наступного розроблено SQL коду:

```

CREATE TABLE `verstat` (
`idverstat` int(11) NOT NULL,
`name` varchar(45) DEFAULT NULL,
PRIMARY KEY (`idverstat`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Створення таблиці операцій верстатів відбувається за допомогою наступного розроблено SQL коду:

```

CREATE TABLE `verstatoper` (
`idverstatoper` int(11) NOT NULL,
`name` varchar(45) DEFAULT NULL,
`idverstat` int(11) DEFAULT NULL,
PRIMARY KEY (`idverstatoper`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

### 4.3 Проектування інтерфейсу користувача

Також для проектування інтерфейсу використовуємо ASP.NET, хоча він включає перераховані вище технології (HTML, CSS, JavaScript) для проектування веб-інтерфейсів користувача і використовує їх, для нас важливі його веб-елементи управління, за допомогою яких на клієнтській стороні можна відобразити складні елементи інтерфейсу, а на серверній стороні реалізувати логіку їх роботи.

Виходячи із завдання було спроектований інтерфейс, який включає наступні веб-сторінки:

- логування – для входу в ІС;
- програми виробництва – для формування, перегляду та редагування замовлень;
- виробів – для формування, перегляду та редагуванню характеристик виробів та їх технологічних маршрутів;
- засобів виробництва - для редагування інформації про наявні верстати та їх функціонал;
- налаштування – для редагування інформації про компоненти для виробництва, таких як база кольорів виробів чи матеріалів деталей;
- звітів – для роздруку чи скачування звітів в форматах .docx, .excel, .pdf;

На рисунку 4.3.1 представлена веб-сторінка, яка реалізує інтерфейс логування.

Основою інтерфейсу логування є елемент `<asp:Login>`, функціональне призначення якого буде наведено в наступному розділі, а зараз важливим є його графічний інтерфейс.

Стиль елементів інтерфейсу визначається в файлі `signin.css`:

```
body {  
  padding-top: 40px;  
  padding-bottom: 40px;  
  background-color: #eee;  
}  
  
.form-signin {  
  max-width: 330px;  
  padding: 15px;  
  margin: 0 auto;
```

```

}
.form-signin .form-signin-heading,
.form-signin .checkbox {
  margin-bottom: 10px;
}
.form-signin .checkbox {
  font-weight: normal;
}
.form-signin .form-control {
  position: relative;
  height: auto;
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
  padding: 10px;
  font-size: 16px;
}
.form-signin .form-control:focus {
  z-index: 2;
}
.form-signin input[type="email"] {
  margin-bottom: -1px;
  border-bottom-right-radius: 0;
  border-bottom-left-radius: 0;
}
.form-signin input[type="password"] {
  margin-bottom: 10px;
  border-top-left-radius: 0;
  border-top-right-radius: 0;
}
}

```

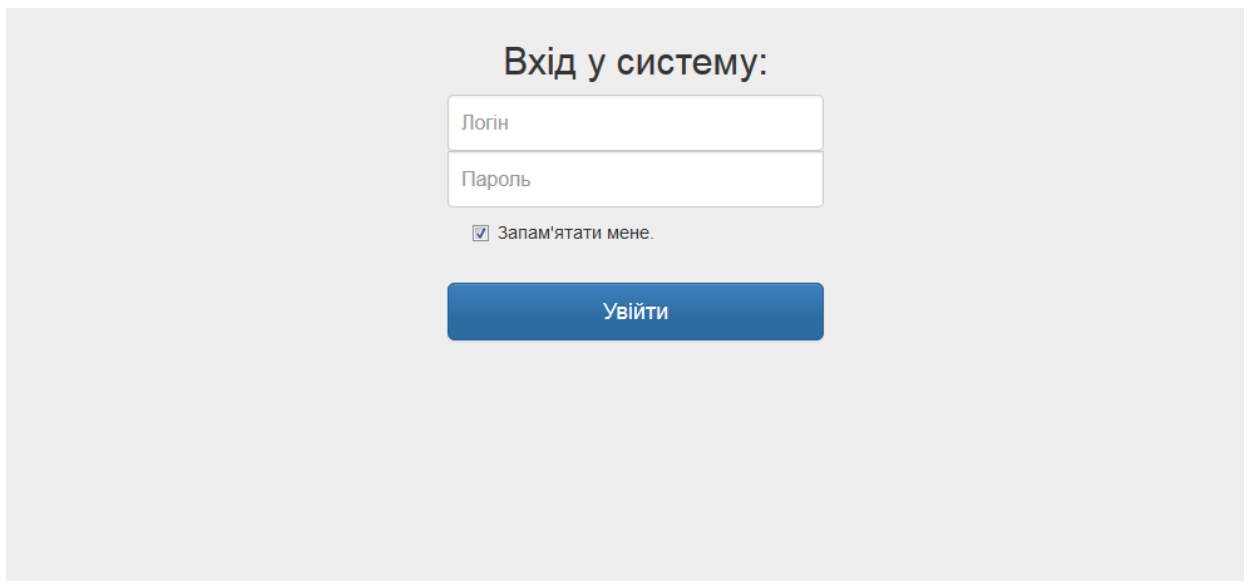


Рисунок 4.3.1 Інтерфейс логування

А наступний код є реалізацію такого інтерфейсу:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="login.aspx.cs" Inherits="login" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<link href="../../../css/jquery-ui-1.10.4.custom.css" rel="stylesheet" />
<link href="../../../css/bootstrap.css" rel="stylesheet" />
<link href="../../../css/bootstrap-theme.css" rel="stylesheet" />

```

```

<link href="../../css/additional.css" rel="stylesheet" />
<script type="text/javascript" src="../../js/jquery-1.10.2.js"></script>
<script type="text/javascript" src="../../js/jquery-ui-1.10.4.js"></script>
<script type="text/javascript" src="../../js/bootstrap.js"></script>
<link href="../../css/signin.css" rel="stylesheet" />
</head>
<body>
<div class="container">
<form runat="server" class="form-signin">
<asp:Login ID="Login1" runat="server" OnAuthenticate="Login1_Authenticate" Width="300px"
FailureText="Ваш логін чи пароль не вірний.">
<LayoutTemplate>
<h2 class="form-signin-heading" style="text-align: center">Вхід у систему:</h2>
<asp:TextBox ID="UserName" runat="server" CssClass="form-control" placeholder="Логін"
autofocus></asp:TextBox>
<asp:TextBox ID="Password" runat="server" CssClass="form-control" TextMode="Password"
placeholder="Пароль"></asp:TextBox>
<label class="checkbox"><asp:CheckBox ID="RememberMe" CssClass="checkbox" runat="server"
Text="Запам'ятати мене."/></label>
<label style="text-align:center; color:red;"><asp:Literal ID="FailureText" runat="server"
EnableViewState="False"></asp:Literal></label>
<asp:Button ID="LoginButton" runat="server" CommandName="Login" CssClass="btn btn-lg btn-primary
btn-block" Text="Увійти" ValidationGroup="Login1" />
</LayoutTemplate>
</asp:Login>
</form>
</div>
</body>
</html>

```

Для всіх сторінок ІС, окрім спільне використання бібліотек Bootstrap та jQuery, є спільне використання файлу стилів – additional.css.

Він містить класи, що описують вигляд текстових полів, кнопок, таблиць, полів, відступів, тощо.

Вмістом його є наступний код:

```

body
{
    background-color:#eee
}
.LBBorder
{
    border-color: #bce8f1;
}
.IMG
{
    padding: 5px;
    padding-left: 15%;
}
.MARGIN
{
    padding-left: 25%;
}
.MTABLE
{
    text-align:center;
}
.M
{

```

```

{
width:24px;
height:24px;
background:url(../images/dnf3.png);
background-size: 24px 24px;
background-repeat: no-repeat;
position:relative;
top: 5px;
right: 5px;
}
.print
{
width:24px;
height:24px;
background:url(../images/print.png);
background-size: 24px 24px;
background-repeat: no-repeat;
position:relative;
top: -5px;
right: 5px;
}
.save
{
width:24px;
height:24px;
background:url(../images/save.png);
background-size: 24px 24px;
background-repeat: no-repeat;
position:relative;
top: 5px;
right: 5px;
}
.mybtn
{
width: 100%;
height: 36px;
font-size: 18px;
}
.mybtn2
{
width: 75px;
height: 75px;
font-size: 18px;
margin: 15px;
border: 1px solid black;
}
.report_view
{
border: 2px solid black;
}
.marg
{
padding: 10px;
}
.nav > li > a.cl:hover {
background-color: #d9edf7;
}
th
{
background-color: #f5f5f5;
}
.TFooter

```

```

{
    background-color: #f5f5f5;
}
.btn-danger {
    background-image: -webkit-linear-gradient(top, #f36a66 0%, #d0403c 100%);
    background-image: linear-gradient(to bottom, #f36a66 0%, #d0403c 100%);
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffd9534f',
endColorstr='#ffc12e2a', GradientType=0);
    filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);
    background-repeat: repeat-x;
    border-color: #b92c28;
}
.btn-danger:hover,
.btn-danger:focus
{
    background-color: #d0403c;
    background-position: 0 -15px;
}

```

Це не всі стилі використанні на сторінка ІС, багато є вбудованими за допомогою атрибута style для забезпечення 100% перевизначення стилів для елементів, або спрощення супроводу коду.

Особливим інтерфейсу програми виробництва є використання “важких” елементів ASP.Net, а саме: <asp:GridView>, <asp:ListBox>, <asp:DropDownList>. І ряду простіших, таких як <asp:TextBox> чи <asp:LinkButton>.

Завдання даного інтерфейсу є забезпечення можливості формування та редагування інформації про замовлення та роздрук такої інформації.

Наступний код є реалізацією такого інтерфейсу:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="Default3" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<link href=" ../css/jquery-ui-1.10.4.custom.css" rel="stylesheet" />
<link href=" ../css/bootstrap.css" rel="stylesheet" />
<link href=" ../css/bootstrap-theme.css" rel="stylesheet" />
<link href=" ../css/additional.css" rel="stylesheet" />
<script type="text/javascript" src=" ../js/jquery-1.10.2.js"></script>
<script type="text/javascript" src=" ../js/jquery-ui-1.10.4.js"></script>
<script type="text/javascript" src=" ../js/bootstrap.js"></script>
</head>
<body>
<form runat="server">
<div class="panel panel-info" style="margin: 2% auto; width: 1280px">
<div class="panel-heading">
<ul class="nav nav-pills nav-justified">
<li><a href="Default.aspx" class="cl"></a></li>
<li class="active"><a href="Default.aspx">Програма виробництва</a></li>
<li><a href="Products.aspx">Вироби</a></li>
<li><a href="Meansofproduction.aspx">Засоби виробництва</a></li>
<li><a href="Options.aspx">Налаштування</a></li>
<li><a href="Reports.aspx">Звіти</a></li>

```

```

<li><div style="float:right;"><asp:Button ID="Button4" runat="server" Text="X" CssClass="btn btn-
danger" OnClick="Signout_Click" /></div></li>
</ul>
</div>
<div class="panel-body">
<table style="margin:0 auto;">
<tr>
<td rowspan="3" style="padding:5px; vertical-align:top;">
<asp:Label ID="Label9" runat="server" Text="Документи:"></asp:Label>
<br />
<asp:ListBox ID="ListBox1" runat="server" CssClass="LBBorder" AutoPostBack="true"
OnSelectedIndexChanged="ListBox1_SelectedIndexChanged" ></asp:ListBox>
</td>
<td style="padding:15px;">
<table style="margin:0 auto;">
<tr><td style="text-align:right; padding:5px;"><asp:Label ID="Label11" runat="server" Text="Дата
"></asp:Label></td><td><asp:TextBox ID="TextBox1" runat="server" Enabled="false" CssClass="form-
control disabled" Height="30px"></asp:TextBox></td></tr>
<tr><td style="text-align:right; padding:5px;"><asp:Label ID="Label12" runat="server"
Text="Замовлення № "></asp:Label></td><td><asp:TextBox ID="TextBox2" runat="server"
Enabled="false" CssClass="form-control disabled" Height="30px"></asp:TextBox></td></tr>
<tr><td style="text-align:right; padding:5px;"><asp:Label ID="Label13" runat="server"
Text="Замовник "></asp:Label></td><td><asp:TextBox ID="TextBox3" runat="server" CssClass="form-
control" Height="30px"></asp:TextBox></td></tr>
<asp:TextBox ID="Label10" runat="server" Text='<# Bind("counts") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="Label11" runat="server" Text='<# Bind("counts") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="Label16" runat="server" Enabled="true" CssClass="textbox-little
MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Настройки">
<HeaderStyle CssClass="MTABLE M4" />
<FooterStyle CssClass="TFooter" />
<itemtemplate>
<asp:linkbutton id="btnEdit" runat="server" causesvalidation="false" commandname="Edit" text="">
<asp:Image ID="Image1" runat="server" ImageUrl="~/images/edit.png"
CssClass="IMG"/></asp:linkbutton>
<asp:linkbutton id="btnDelete" runat="server" causesvalidation="false" commandname="Delete"
text="" >
<asp:Image ID="Image2" runat="server" ImageUrl="~/images/delete.png"
CssClass="IMG"/></asp:linkbutton>
</itemtemplate>
<edititemtemplate>
<asp:linkbutton id="btnUpdate" runat="server" commandname="Update" text="" >
<asp:Image ID="Image3" runat="server" ImageUrl="~/images/update.png"
CssClass="IMG"/></asp:linkbutton>
<asp:linkbutton id="btnCancel" runat="server" causesvalidation="false" commandname="Cancel"
text="" >
<asp:Image ID="Image1" runat="server" ImageUrl="~/images/cancel.png"
CssClass="IMG"/></asp:linkbutton>
</edititemtemplate>
<footertemplate>
<asp:linkbutton id="btnInsert" runat="server" commandname="FooterInsert" text=""
CssClass="MARGIN">
<asp:Image ID="Image1" runat="server" ImageUrl="~/images/insert.png" CssClass="IMG"
/></asp:linkbutton>
</footertemplate>
</asp:TemplateField>

```

```

</Columns>
<EmptyDataTemplate>
No records
</EmptyDataTemplate>
</asp:GridView>
</td>
</tr>
<tr>
<td>
<table class="table-center">
<tr>

```

Наступним інтерфейсом, який має унікальну в межах інформаційної системи, структури є веб-сторінка, яка реалізує інтерфейс роботи з виробима і який зображений на рисунку 4.3.3 та 4.3.4.

Особливістю даного інтерфейсу є використання модального вікна та елементів `<asp:ScriptManager>` і `<asp:UpdatePanel>`.

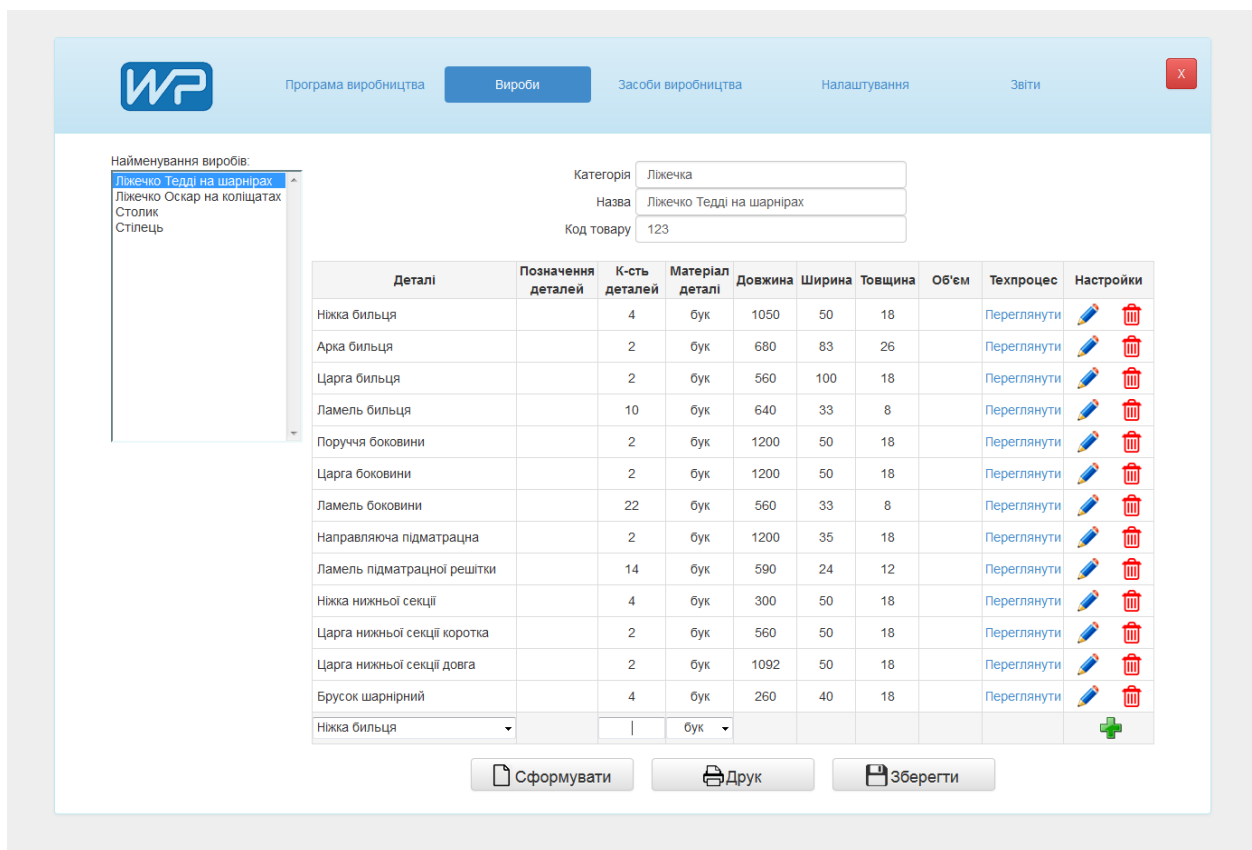


Рис. 4.3.3 Інтерфейс роботи з виробами



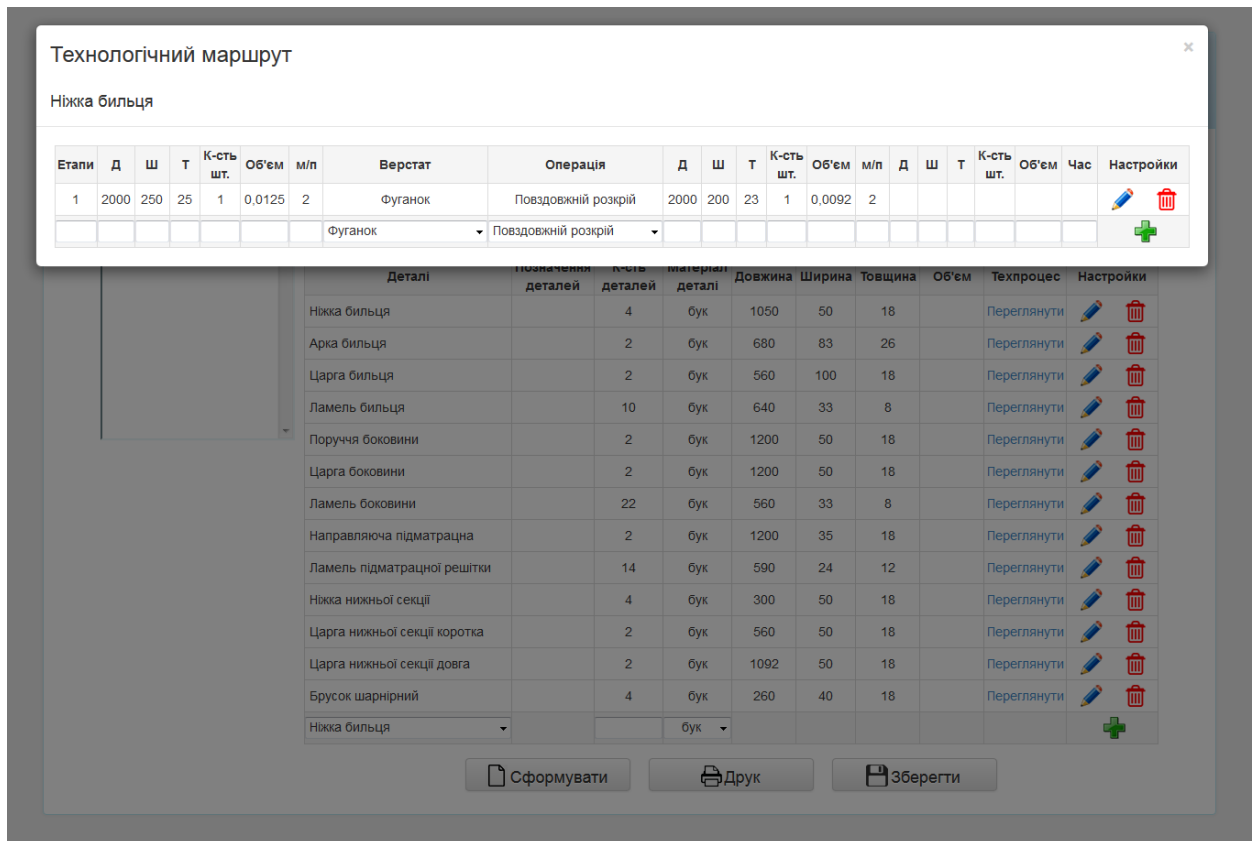


Рис. 4.3.4 Інтерфейс роботи з технологічним маршрутом деталей

Наступний код є реалізацією даного інтерфейсу:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Products.aspx.cs" Inherits="Products" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<link href=" ../css/jquery-ui-1.10.4.custom.css" rel="stylesheet" />
<link href=" ../css/bootstrap.css" rel="stylesheet" />
<link href=" ../css/bootstrap-theme.css" rel="stylesheet" />
<link href=" ../css/additional.css" rel="stylesheet" />
<script type="text/javascript" src=" ../js/jquery-1.10.2.js"></script>
<script type="text/javascript" src=" ../js/jquery-ui-1.10.4.js"></script>
<script type="text/javascript" src=" ../js/bootstrap.js"></script>
</head>
<body>
<form id="form1" runat="server">
<div>
<div class="panel panel-info" style="margin: 2% auto; width: 1280px">
<div class="panel-heading">
<ul class="nav nav-pills nav-justified">
<li><a href="Default.aspx" class="cl"></a></li>
<li><a href="Default.aspx">Програма виробництва</a></li>
<li class="active"><a href="Products.aspx">Вироби</a></li>
<li><a href="Meansofproduction.aspx">Засоби виробництва</a></li>
<li><a href="Options.aspx">Налаштування</a></li>
<li><a href="Reports.aspx">Звіти</a></li>
<li><div style="float:right;"><asp:Button ID="Button4" runat="server" Text="X" CssClass="btn btn-danger"
OnClick="Signout_Click"/></div></li>
</ul>
</div>
<div class="panel-body">
```

```

</ItemTemplate>
<FooterTemplate>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Ширина" ItemStyle-CssClass="MCELL" ItemStyle-Width="65px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L61" runat="server" Text='<%# Bind("width") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Товщина" ItemStyle-CssClass="MCELL" ItemStyle-Width="70px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L71" runat="server" Text='<%# Bind("thickness") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Об'єм" ItemStyle-CssClass="MCELL" ItemStyle-Width="70px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L81" runat="server" Text='<%# Bind("volume") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Техпроцес" ItemStyle-CssClass="MCELL" ItemStyle-Width="90px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<ItemTemplate>
<asp:LinkButton ID="LinkButton1" runat="server" CommandName="detail" CommandArgument='<%#
(((GridViewRow) Container).RowIndex.ToString() + "_" + Eval("name"))
>'>Переглянути</asp:LinkButton>
<td class="M6 marg"><asp:linkbutton id="Linkbutton1" runat="server"
OnClick="Signout_Click"><button class="btn btn-default mybtn"
onclick="javascript:__doPostBack('Linkbutton1','')">Сформувати</button></asp:linkbutton></td>
<td class="M6 marg"><asp:linkbutton id="Linkbutton2" runat="server"
OnClick="Signout_Click"><button class="btn btn-default mybtn"
onclick="javascript:__doPostBack('Linkbutton2','')">Друк</button></asp:linkbutton></td>
<td class="M6 marg"><asp:linkbutton id="Linkbutton3" runat="server"
OnClick="Signout_Click"><button class="btn btn-default mybtn"
onclick="javascript:__doPostBack('Linkbutton3','')">Зберегти</button></asp:linkbutton></td>
<div class="modal fade" id="currentdetail" tabindex="-1" role="dialog" aria-
labelledby="myModalLabel" aria-hidden="true">
<div class="modal-dialog modal-lg" style="width:1280px;">
<div class="modal-content">
<asp:UpdatePanel ID="UpdatePanel2" runat="server">
<ContentTemplate>
<div class="modal-header">

```

```

<button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
<h3 class="modal-title" id="myModalLabel">Технологічний маршрут</h3>
<br />
<h4 class="modal-title" ><asp:Label ID="Label4" runat="server" Text="Label"></asp:Label></h4>
<asp:HiddenField ID="HiddenField1" runat="server" />
</div>
<div class="modal-body">
<asp:GridView ID="GridView2" runat="server" AllowPaging="false" CssClass="table-bordered table-
hover table-center" AutoGenerateColumns="False" ShowFooter="true" ShowHeaderWhenEmpty="True"
OnRowEditing="GridView2_RowEditing" OnRowCancelingEdit="GridView2_RowCancelingEdit"
OnRowDeleting="GridView2_RowDeleting" OnRowUpdating="GridView2_RowUpdating"
OnRowCommand="GridView2_RowCommand">
<Columns>
<asp:TemplateField HeaderText="Етапи" ItemStyle-CssClass="M" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB11" runat="server" Text='<%# Bind("stage") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L11" runat="server" Text='<%# Bind("stage") %>' ></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB12" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Д" ItemStyle-CssClass="MCELL M" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB21" runat="server" Text='<%# Bind("PD") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Операція" ItemStyle-CssClass="MCELL" ItemStyle-Width="300px"
ItemStyle-HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:DropDownList ID="DDL221" runat="server" CssClass="textbox-little"></asp:DropDownList>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L211" runat="server" Text='<%# Bind("name1") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:DropDownList ID="DDL212" runat="server" CssClass="textbox-little"></asp:DropDownList>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Д" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px" ItemStyle-

<EditItemTemplate>
<asp:TextBox ID="TB161" runat="server" Text='<%# Bind("LW") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L161" runat="server" Text='<%# Bind("LW") %>'></asp:Label>
</ItemTemplate>

```

```

<FooterTemplate>
<asp:TextBox ID="TB162" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="T" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB171" runat="server" Text='<%# Bind("LT") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L171" runat="server" Text='<%# Bind("LT") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB172" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="К-сть шт." ItemStyle-CssClass="MCELL" ItemStyle-Width="50px"
ItemStyle-HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB181" runat="server" Text='<%# Bind("lcount") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L181" runat="server" Text='<%# Bind("lcount") %>'></asp:Label>
</Columns>
<EmptyDataTemplate>
No records
</EmptyDataTemplate>
</asp:GridView>
</div>
</ContentTemplate>
<Triggers>
<asp:AsyncPostBackTrigger ControlID="GridView1" EventName="RowCommand" />
</Triggers>
</asp:UpdatePanel>
</div>
</div>
</div>
</div>
</form>
</body>
</html>

```

Ще одним унікальним інтерфейсом є інтерфейс, який реалізує можливість роздруковувати і скачувати звіти. Також для представлення звітів на веб-сторінці використовує елемент <rsweb:ReportViewer>.

Наступний код є реалізацією даного інтерфейсу:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Reports.aspx.cs" Inherits="Reports" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<link href=" ../css/jquery-ui-1.10.4.custom.css" rel="stylesheet" />
<link href=" ../css/bootstrap.css" rel="stylesheet" />

```

```

<link href="../css/bootstrap-theme.css" rel="stylesheet" />
<link href="../css/additional.css" rel="stylesheet" />
<script type="text/javascript" src="../js/jquery-1.10.2.js"></script>
<script type="text/javascript" src="../js/jquery-ui-1.10.4.js"></script>
<script type="text/javascript" src="../js/bootstrap.js"></script>
<script type="text/javascript" src="../js/print.js"></script>
</head>
<body style="background-color:#eee">
<form id="form1" runat="server">
<div>
<div class="panel panel-info" style="margin: 2% auto; width: 1280px">
<div class="panel-heading">
<ul class="nav nav-pills nav-justified">
<li><a href="Default.aspx" class="cl"></a></li>
<li><a href="Default.aspx">Програма виробництва</a></li>
<li><a href="Products.aspx">Вироби</a></li>
<li><a href="Meansofproduction.aspx">Засоби виробництва</a></li>
<li><a href="Options.aspx">Налаштування</a></li>
<li class="active"><a href="Reports.aspx">Звіти</a></li>
<li><div style="float:right;"><asp:Button ID="Button4" runat="server" Text="X" CssClass="btn btn-
danger" OnClick="Signout_Click"/></div></li>
</ul>
</div>
<div class="panel-body">
<asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
<table style="width:70%; margin: 0 auto;">
<tr>
<td style="vertical-align:top; padding: 3px;">
<span>Категорія:</span>
<asp:DropDownList ID="DropDownList1" runat="server" Width="150px"></asp:DropDownList>
<span>Звіт:</span>
<asp:DropDownList ID="DropDownList2" runat="server" Width="150px"></asp:DropDownList>
</td>
<td style="vertical-align:top; padding: 3px;">
<span>Вміст:</span>
<rsweb:ReportViewer ID="rv1" runat="server" CssClass="report_view" Font-Names="Verdana"
Height="650px" Width="800px" Font-Size="8pt" ShowToolBar="false" WaitMessageFont-Names="Verdana"
WaitMessageFont-Size="14pt" >
<LocalReport ReportPath="Reports/Report.rdlc">
<DataSources>
<rsweb:ReportDataSource DataSourceId="ObjectDataSource1" Name="DataSet1" />
</DataSources>
</LocalReport>
</rsweb:ReportViewer>
<asp:ObjectDataSource ID="ObjectDataSource1" runat="server" DeleteMethod="Delete"
InsertMethod="Insert" OldValuesParameterFormatString="original_{0}" SelectMethod="GetData"
TypeName="woodDataSetTableAdapters.orderstoproductsTableAdapter" UpdateMethod="Update" >
<DeleteParameters>
<asp:Parameter Name="p1" Type="Int32" />
</DeleteParameters>
<InsertParameters>
<asp:Parameter Name="p1" Type="Int32" />
<asp:Parameter Name="p2" Type="Int32" />
<asp:Parameter Name="p3" Type="Int32" />
<asp:Parameter Name="p4" Type="Int32" />
</InsertParameters>
</asp:ObjectDataSource>
</td>
<td style="vertical-align:top; padding: 3px;">
<span>Опції:</span>
<asp:linkbutton id="Linkbutton1" runat="server" OnClick="Signout_Click"><button class="btn btn-
warning mybtn2" onclick="javascript: __doPostBack('Linkbutton1','')"></button></asp:linkbutton>

```

```

<asp:linkbutton id="Linkbutton2" runat="server" OnClick="Signout_Click"><button class="btn btn-
primary mybtn2" onclick="javascript:__doPostBack('Linkbutton2','')"></button></asp:linkbutton>
<asp:linkbutton id="Linkbutton3" runat="server" OnClick="Signout_Click"><button class="btn btn-
success mybtn2" onclick="javascript:__doPostBack('Linkbutton3','')"></button></asp:linkbutton>
<asp:linkbutton id="Linkbutton4" runat="server" OnClick="Signout_Click"><button class="btn btn-
danger mybtn2" onclick="javascript:__doPostBack('Linkbutton4','')"></button></asp:linkbutton>
<br />
</td>
</tr>
</table>
<script>
$('#printreport').click(function () {
printReport('rv1');
});
</script>
</div>
</div>
</div>
</form>
</body>
</html>

```

Також представлення даного інтерфейсу можна побачити на рисунку 4.3.5.

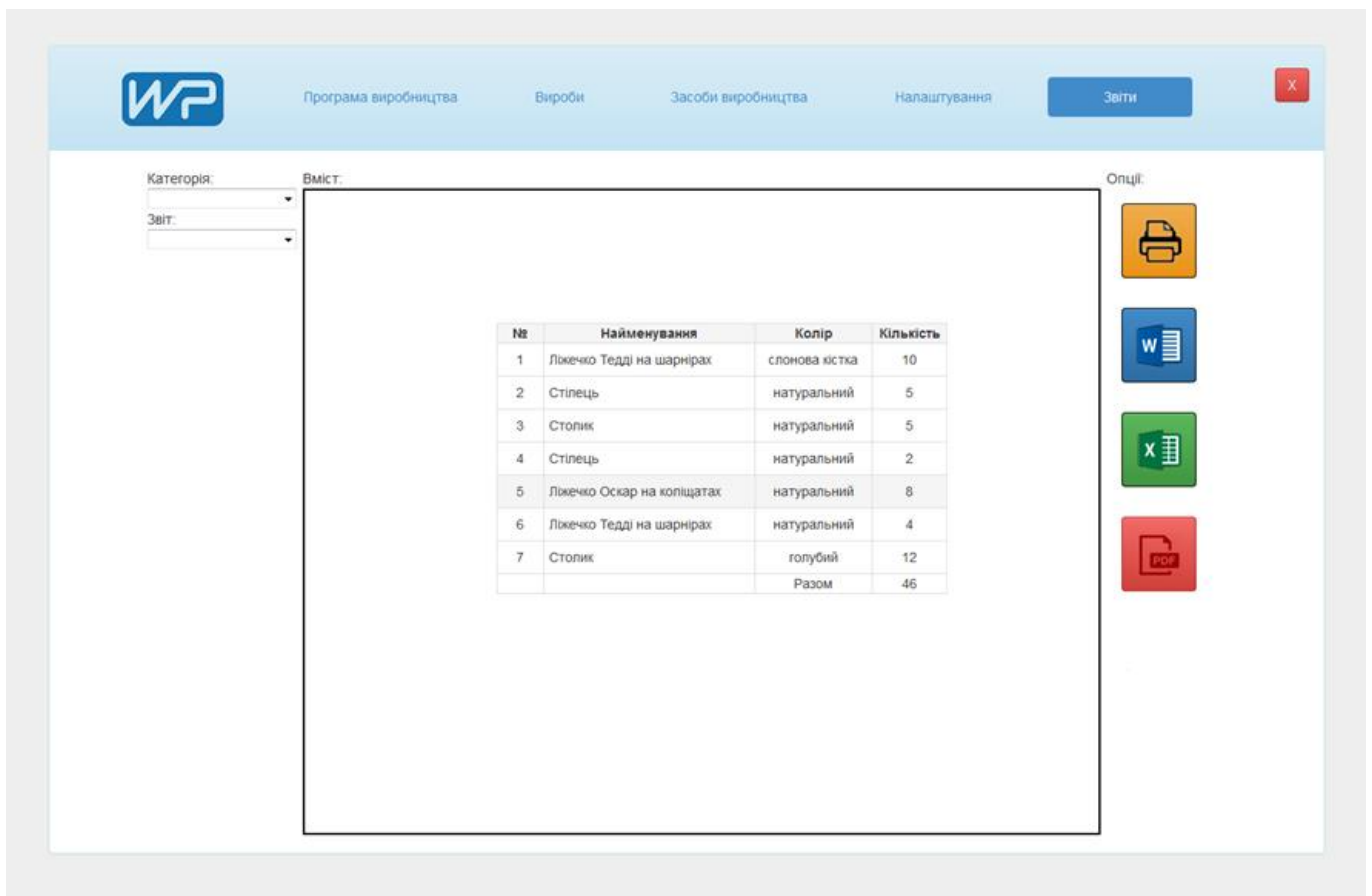


Рис. 4.3.5 Інтерфейс для роботи з звітами

## 4.4 Реалізація логіки функціонування інформаційної системи

Для реалізації логіки функціонування інформаційної системи, а саме доступу до БД, нам також буде необхідне додатко ПО:

- MySQL Connector Net;
- MySQL for Visual Studio;

Дане ПО забезпечує необхідні класи та інтерфейси для роботи з БД.

Реалізація обробки подій для інтерфейсу логуювання включає наступний код:

```
public partial class login : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Login1_Authenticate(object sender, AuthenticateEventArgs e)
    {
        if (login.LogIn(Login1.UserName, Login1.Password))
        {
            System.Web.Security.FormsAuthentication.RedirectFromLoginPage(Login1.UserName, true);
        }
    }
    protected static bool LogIn(string User, string Password)
    {
        MySqlConnection conn;
        DataTable data;
        MySqlDataAdapter da;
        MySqlCommandBuilder cb;

        String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false");
        conn = new MySqlConnection(connStr);
        conn.Open();
        conn.ChangeDatabase("wood");

        data = new DataTable();
        da = new MySqlDataAdapter("SELECT * FROM users", conn);
        da.Fill(data);
        for (int i = 0; i < data.Rows.Count; i++)
        {
            if (data.Rows[i][1].ToString() == User && data.Rows[i][2].ToString() == Password)
                return true;
        }
        return false;

        conn.Close();
    }
}
```

Реалізація логуювання відбувається на основі механізму логуювання форм та класу FormsAuthentication.

Реалізація обробки подій для інтерфейсу програми виробництва включає наступний код:

```
public partial class Default3 : System.Web.UI.Page
{
    protected void Signout_Click(object sender, EventArgs e)
    {
        FormsAuthentication.SignOut();
        Response.Redirect("login.aspx");
    }
    protected void ListBox_Fill()
    {
        MySqlConnection conn;
        DataTable data;
        MySqlDataAdapter da;
        MySqlCommandBuilder cb;

        String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false");
        conn = new MySqlConnection(connStr);
        conn.Open();
        conn.ChangeDatabase("wood");

        data = new DataTable();
        da = new MySqlDataAdapter("SELECT N, Date FROM orders", conn);
        da.Fill(data);
        ListBox1.Items.Clear();
        for (int i = data.Rows.Count - 1; i >= 0; i--)
        {
            ListBox1.Items.Add("№" + data.Rows[i][0].ToString() + " від " +
data.Rows[i][1].ToString().Split(' ')[0]);
        }
        conn.Close();
    }
    protected void Grid_Fill()
    {
        MySqlConnection conn;
        DataTable data;
        MySqlDataAdapter da;
        MySqlCommandBuilder cb;

        String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false");
        conn = new MySqlConnection(connStr);
        conn.Open();
        conn.ChangeDatabase("wood");

        da = new MySqlDataAdapter("SELECT N, Date FROM orders", conn);
        data = new DataTable();
        da.Fill(data);
        string N = "";
        string temp;
        foreach (DataRow row in data.Rows)
        {
            temp = "№" + row[0].ToString() + " від " + row[1].ToString().Split(' ')[0];
            if (temp == ListBox1.SelectedValue)
            {
                N = row["N"].ToString();
                break;
            }
        }
        da = new MySqlDataAdapter("SELECT * FROM orders WHERE N = " + N, conn);
    }
}
```



```

data = new DataTable();
da.Fill(data);
TextBox1.Text = data.Rows[0][3].ToString().Split(' ')[0];
TextBox2.Text = data.Rows[0][1].ToString();
TextBox3.Text = data.Rows[0][2].ToString();
TextBox4.Text = data.Rows[0][4].ToString().Split(' ')[0];
TextBox5.Text = data.Rows[0][5].ToString();
TextBox6.Text = data.Rows[0][6].ToString().Split(' ')[0];

DataTable products, colors;
products = new DataTable();
colors = new DataTable();

DropDownList ddl1 = (DropDownList)(GridView1.FooterRow.FindControl("Label14"));
DDL_Fill_Products(ddl1);
DropDownList ddl2 = (DropDownList)(GridView1.FooterRow.FindControl("Label15"));
DDL_Fill_Colors(ddl2);

conn.Close();
}
protected void DDL_Fill_Products(DropDownList ddl)
{
    MySqlConnection conn;
    DataTable data;
    MySqlDataAdapter da;
    MySqlCommandBuilder cb;

    String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false");
    conn = new MySqlConnection(connStr);
    conn.Open();
    conn.ChangeDatabase("wood");

    data = new DataTable();
    da = new MySqlDataAdapter("SELECT products.name FROM products", conn);
    da.Fill(data);
    foreach (DataRow row in data.Rows)
        ddl.Items.Add(row[0].ToString());
    conn.Close();
}
protected void DDL_Fill_Colors(DropDownList ddl)
{
    MySqlConnection conn;
    DataTable data;
    MySqlDataAdapter da;
    MySqlCommandBuilder cb;

    String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false");
    conn = new MySqlConnection(connStr);
    conn.Open();
    conn.ChangeDatabase("wood");

    data = new DataTable();
    da = new MySqlDataAdapter("SELECT colors.name FROM colors", conn);
    da.Fill(data);
    foreach (DataRow row in data.Rows)
        ddl.Items.Add(row[0].ToString());
    conn.Close();
}
protected void Page_Load(object sender, EventArgs e)

```

```

{
    if (!Page.IsPostBack)
        ListBox_Fill();
    ListBox1.Height = 300;
}
}
protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    MySqlConnection conn;
    DataTable data;
    MySqlDataAdapter da;
    MySqlCommandBuilder cb;

    String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false; charset=utf8");
    conn = new MySqlConnection(connStr);
    conn.Open();
    conn.ChangeDatabase("wood");

    string N = "";
    N = TextBox2.Text;
    da = new MySqlDataAdapter("SELECT * FROM orderstoproducts WHERE `order` =" + N, conn);
    data = new DataTable();
    da.Fill(data);

    MySqlCommand cmd = new MySqlCommand("DELETE FROM orderstoproducts WHERE
idorderstoproducts = "+data.Rows[e.RowIndex][0]+";", conn);
    cmd.ExecuteNonQuery();
    conn.Close();
    Grid_Fill();
}
protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    MySqlConnection conn;
    DataTable data;
    MySqlDataAdapter da;
    MySqlCommandBuilder cb;

    String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false; charset=utf8");
    conn = new MySqlConnection(connStr);
    conn.Open();
    conn.ChangeDatabase("wood");

    string N = "";
    N = TextBox2.Text;

    da = new MySqlDataAdapter("SELECT * FROM orderstoproducts WHERE `order` =" + N, conn);
    data = new DataTable();
    da.Fill(data);

    MySqlCommand cmd = new MySqlCommand("UPDATE orderstoproducts SET product=" +
(((DropDownList)GridView1.Rows[GridView1.EditIndex].FindControl("Label14")).SelectedIndex + 1) +
", color=" +
(((DropDownList)GridView1.Rows[GridView1.EditIndex].FindControl("Label17")).SelectedIndex + 1) +
", counts = " + ((TextBox)GridView1.Rows[GridView1.EditIndex].FindControl("Label10")).Text +
"WHERE idorderstoproducts = "+data.Rows[GridView1.EditIndex][0].ToString()+";", conn);
    cmd.ExecuteNonQuery();
    conn.Close();
    GridView1.EditIndex = -1;
    Grid_Fill();
}
}

```

```

protected void ListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    Grid_Fill();
}
protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
{
    if (e.CommandName.Equals("FooterInsert"))
    {
        MySqlConnection conn;
        DataTable data;
        MySqlDataAdapter da;
        MySqlCommandBuilder cb;

        String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false; charset=utf8");
        conn = new MySqlConnection(connStr);
        conn.Open();
        conn.ChangeDatabase("wood");

        string N = "";
        N = TextBox2.Text;
        MySqlCommand cmd = new MySqlCommand("INSERT INTO orderstoproducts
(`order`,product,color,counts) VALUES (" + N + "," +
(((DropDownList)GridView1.FooterRow.FindControl("Label14")).SelectedIndex+1) + "," +
(((DropDownList)GridView1.FooterRow.FindControl("Label15")).SelectedIndex+1) + "," +
((TextBox)GridView1.FooterRow.FindControl("Label16")).Text + ");", conn);
        cmd.ExecuteNonQuery();
        conn.Close();
        Grid_Fill();
    }
}
protected void Button1_Click(object sender, EventArgs e)
{
    foreach (DataRow row in data.Rows)
        sum += Convert.ToInt32(row[3].ToString());
    string[] str = new string[4];
    str[0] = null;
    str[1] = "Разом";
    str[2] = sum.ToString();
    str[3] = null;
    data.Rows.Add(str);

    GridView1.DataSource = data;
    GridView1.DataBind();
    GridView1.Rows[data.Rows.Count - 1].Cells[1].ColumnSpan = 2;
    GridView1.Rows[data.Rows.Count - 1].Cells[1].Style.Add("text-align", "right");
    GridView1.Rows[data.Rows.Count - 1].Cells[1].Style.Add("padding-right", "30px");
    GridView1.Rows[data.Rows.Count - 1].Cells[data.Columns.Count -
1].FindControl("btnEdit").Visible = false;
    GridView1.Rows[data.Rows.Count - 1].Cells[data.Columns.Count -
1].FindControl("btnDelete").Visible = false;

}
protected void Button3_Click(object sender, EventArgs e)
{
    if (ListBox1.SelectedIndex != -1)
    {
        MySqlConnection conn;
        DataTable data;
        MySqlDataAdapter da;
        MySqlCommandBuilder cb;

```

```

        String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false; charset=utf8");
        conn = new MySqlConnection(connStr);
        conn.Open();
        conn.ChangeDatabase("wood");

        string N = "";
        N = TextBox2.Text;
        string BDate = (TextBox4.Text.Substring(6, 4) + "-" + TextBox4.Text.Substring(3, 2) +
        "-" + TextBox4.Text.Substring(0, 2));

        MySqlCommand cmd = new MySqlCommand("set names 'utf8'; UPDATE orders SET customer='"
+ TextBox3.Text + "', BDate='" + BDate + "' WHERE N=" + N.ToString() + "';", conn);
        cmd.ExecuteNonQuery();

        da = new MySqlDataAdapter("set names 'utf8'; SELECT * FROM orders WHERE N = " + N,
conn);

        data = new DataTable();
        da.Fill(data);
        conn.Close();
        TextBox1.Text = data.Rows[0][3].ToString().Split(' ')[0];
        TextBox2.Text = data.Rows[0][1].ToString();
        TextBox3.Text = data.Rows[0][2].ToString();
        TextBox4.Text = data.Rows[0][4].ToString().Split(' ')[0];
        TextBox5.Text = data.Rows[0][5].ToString();
        TextBox6.Text = data.Rows[0][6].ToString().Split(' ')[0];
    }
}
}

```

Реалізація обробки подій для інтерфейсу з виробами та технологічним маршрутом деталей включає наступний код:

```

public partial class Products : System.Web.UI.Page
{
    protected void Signout_Click(object sender, EventArgs e)
    {
        FormsAuthentication.SignOut();
        Response.Redirect("login.aspx");
    }
    protected void ListBox_Fill()
    {
        MySqlConnection conn;
        DataTable data;
        MySqlDataAdapter da;
        MySqlCommandBuilder cb;

        String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false");
        conn = new MySqlConnection(connStr);
        conn.Open();
        conn.ChangeDatabase("wood");

        data = new DataTable();
        da = new MySqlDataAdapter("SELECT name FROM products", conn);
        da.Fill(data);
        ListBox1.Items.Clear();
        for (int i = 0; i < data.Rows.Count; i++)
        {
            ListBox1.Items.Add(data.Rows[i][0].ToString());
        }
        conn.Close();
    }
}

```

```

}

protected void DDL_Fill_Materials(DropDownList ddl)
{
    MySqlConnection conn;
    DataTable data;
    MySqlDataAdapter da;
    MySqlCommandBuilder cb;

    String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false");
    conn = new MySqlConnection(connStr);
    conn.Open();
    conn.ChangeDatabase("wood");

    data = new DataTable();
    da = new MySqlDataAdapter("SELECT name FROM materials", conn);
    da.Fill(data);
    foreach (DataRow row in data.Rows)
        ddl.Items.Add(row[0].ToString());
    conn.Close();
}
protected void DDL_Fill_Verstat(DropDownList ddl)
{
    MySqlConnection conn;
    DataTable data;
    MySqlDataAdapter da;
    MySqlCommandBuilder cb;

    String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false");
    conn = new MySqlConnection(connStr);
    conn.Open();
    conn.ChangeDatabase("wood");

    data = new DataTable();
    da = new MySqlDataAdapter("SELECT name FROM verstat", conn);
    da.Fill(data);
    foreach (DataRow row in data.Rows)
        ddl.Items.Add(row[0].ToString());
    conn.Close();
}
protected void DDL_Fill_VerstatOper(DropDownList ddl)
{
    MySqlConnection conn;
    DataTable data;
    MySqlDataAdapter da;
    MySqlCommandBuilder cb;

    String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false");
    conn = new MySqlConnection(connStr);
    conn.Open();
    conn.ChangeDatabase("wood");

    data = new DataTable();
    da = new MySqlDataAdapter("SELECT name FROM verstatoper", conn);
    da.Fill(data);
    foreach (DataRow row in data.Rows)
        ddl.Items.Add(row[0].ToString());
    conn.Close();
}
protected void Page_Load(object sender, EventArgs e)

```

```

{
    if (!Page.IsPostBack)
        ListBox_Fill();
    ListBox1.Height = 300;
}
}
data = new DataTable();
da = new MySqlDataAdapter("SELECT parts.name, parts.code, productstoparts.count,
materials.name, parts.length, parts.width, parts.thickness, parts.volume "
    +"FROM productstoparts, products, parts, materials "
    +"WHERE productstoparts.product = " + N
    +" AND productstoparts.product = products.idproducts "
    +"AND productstoparts.part = parts.idparts "
    +"AND productstoparts.material = materials.idmaterials "
    +"ORDER BY productstoparts.idproductstoparts", conn);

da.Fill(data);

DataTable parts, materials;
parts = new DataTable();
materials = new DataTable();

if (data.Rows.Count != 0)
{
    DropDownList ddl1 = (DropDownList)(GridView1.FooterRow.FindControl("DDL12"));
    DDL_Fill_Parts(ddl1);
    DropDownList ddl2 = (DropDownList)(GridView1.FooterRow.FindControl("DDL42"));
    DDL_Fill_Materials(ddl2);
}
conn.Close();
}
protected void Grid_Fill2(int index)
{
    MySqlConnection conn;
    DataTable data;
    MySqlDataAdapter da;
    MySqlCommandBuilder cb;

    String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false; charset=utf8");
    conn = new MySqlConnection(connStr);
    conn.Open();
    conn.ChangeDatabase("wood");

    da = new MySqlDataAdapter("SELECT * FROM parts", conn);
    data = new DataTable();
    da.Fill(data);
    string N = "";
    string temp;
    foreach (DataRow row in data.Rows)
    {
        temp = Label4.Text;
        if (temp == row[1].ToString())
        {
            N = row[0].ToString();
            break;
        }
    }
    data = new DataTable();
    da = new MySqlDataAdapter("SELECT techprocess.stage, techprocess.PD, techprocess.PW,
techprocess.PT, "
    +"techprocess.pcount, techprocess.pvolume, techprocess.pmp, verstat.name, "
    +"verstatoper.name, techprocess.CD, techprocess.CW, techprocess.CT, "
    +"techprocess.ccount, techprocess.cvolume, techprocess.cmp, techprocess.LD, "

```

```

        +"techprocess.LW, techprocess.LT, "
        +"techprocess.lcount, techprocess.lvolume, techprocess.time "
        +"FROM techprocess, parts, verstat, verstatoper "
        +"WHERE techprocess.part = parts.idparts "
        +"AND techprocess.idverstat = verstat.idverstat "
        +"AND techprocess.idverstatoper = verstatoper.idverstatoper "
        +"AND techprocess.part =" + N
        +" ORDER BY techprocess.stage;", conn);

if (data.Rows.Count == 0)
{
    data.Rows.Add(str);
    flag = true;
}
GridView2.DataSource = data;
GridView2.DataBind();
if (data.Rows.Count == 1 && flag)
{
    GridView2.Rows[data.Rows.Count - 1].Cells[21].FindControl("btnEdit").Visible = false;
    GridView2.Rows[data.Rows.Count - 1].Cells[21].FindControl("btnDelete").Visible =
false;
    GridView2.Rows[data.Rows.Count - 1].Cells[0].Style.Add("height", "30px");
}
DropDownList ddl1 = (DropDownList)(GridView2.FooterRow.FindControl("DDL82"));
DDL_Fill_Verstat(ddl1);
DropDownList ddl2 = (DropDownList)(GridView2.FooterRow.FindControl("DDL212"));
DDL_Fill_VerstatOper(ddl2);
conn.Close();
}

da = new MySqlDataAdapter("SELECT * FROM productstoparts WHERE product =" + N, conn);
data = new DataTable();
da.Fill(data);

MySqlCommand cmd = new MySqlCommand("DELETE FROM productstoparts WHERE idproductstoparts
= " + data.Rows[e.RowIndex][0] + ";", conn);
cmd.ExecuteNonQuery();
conn.Close();
Grid_Fill();
}
protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    MySqlConnection conn;
    DataTable data;
    MySqlDataAdapter da;
    MySqlCommandBuilder cb;

    String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false; charset=utf8");
    conn = new MySqlConnection(connStr);
    conn.Open();
    conn.ChangeDatabase("wood");

    da = new MySqlDataAdapter("SELECT * FROM products", conn);
    data = new DataTable();
    da.Fill(data);
    string N = "";
    string temp;
    foreach (DataRow row in data.Rows)
    {
        temp = row[1].ToString();
        if (temp == ListBox1.SelectedValue)
        {

```

```

        N = row[0].ToString();
        break;
    }
}

da = new MySqlDataAdapter("SELECT * FROM productstoparts WHERE product =" + N, conn);
data = new DataTable();
da.Fill(data);

MySqlCommand cmd = new MySqlCommand("UPDATE productstoparts SET product=" + N + ", part="
+ (((DropDownList)GridView1.Rows[GridView1.EditIndex].FindControl("DDL11")).SelectedIndex + 1) +
", count = " + ((TextBox)GridView1.Rows[GridView1.EditIndex].FindControl("TB31")).Text +
",material = " +
(((DropDownList)GridView1.Rows[GridView1.EditIndex].FindControl("DDL41")).SelectedIndex + 1) + "
WHERE idproductstoparts = " + data.Rows[GridView1.EditIndex][0].ToString() + ";", conn);
cmd.ExecuteNonQuery();
conn.Close();
GridView1.EditIndex = -1;
Grid_Fill();
}
protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
{
protected void GridView2_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    MySqlConnection conn;
    DataTable data;
    MySqlDataAdapter da;
    MySqlCommandBuilder cb;

    String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false; charset=utf8");
    conn = new MySqlConnection(connStr);
    conn.Open();
    conn.ChangeDatabase("wood");

    da = new MySqlDataAdapter("SELECT * FROM parts", conn);
    data = new DataTable();
    da.Fill(data);
    string N = "";
    string temp;
    foreach (DataRow row in data.Rows)
    {
        temp = Label4.Text;
if (temp == row[1].ToString())
        {
            N = row[0].ToString();
            break;
        }
    }
    da = new MySqlDataAdapter("SELECT * FROM techprocess WHERE part = " + N + " ORDER BY
techprocess.stage", conn);
    data = new DataTable();
    da.Fill(data);

    MySqlCommand cmd = new MySqlCommand("DELETE FROM techprocess WHERE idtechprocess = " +
data.Rows[e.RowIndex][0] + ";", conn);
    cmd.ExecuteNonQuery();
    conn.Close();
    Grid_Fill2(0);
}
protected void GridView2_RowUpdating(object sender, GridViewUpdateEventArgs e)
{

```



```

MySQLConnection conn;
DataTable data;
MySQLDataAdapter da;
MySQLCommandBuilder cb;

String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false; charset=utf8");
conn = new MySqlConnection(connStr);
conn.Open();
conn.ChangeDatabase("wood");

da = new MySQLDataAdapter("SELECT * FROM parts", conn);
data = new DataTable();
da.Fill(data);
string N = "";
string temp;
foreach (DataRow row in data.Rows)
{
    temp = Label14.Text;
    if (temp == row[1].ToString())
    {
        N = row[0].ToString();
        break;
    }
}

da = new MySQLDataAdapter("SELECT * FROM techprocess WHERE part =" + N + " ORDER BY
techprocess.stage", conn);
data = new DataTable();
da.Fill(data);

MySQLCommand cmd = new MySQLCommand("UPDATE techprocess SET " +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB11")).Text +
    " stage = " +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB21")).Text +
    ", PD = " +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB31")).Text +
    ", PW = " +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB41")).Text +
    ", PT = " +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB41")).Text +
    ", pcount = " +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB51")).Text +
    ", pvolume = " +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB61")).Text +
    ", pmp = " +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB71")).Text +
    ", idverstat = " +
    ", LD = " +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB151")).Text +
    ", LW = " +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB161")).Text +
    ", LT = " +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB171")).Text +
    ", lcount = " +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB181")).Text +
    ", lvolume = " +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB191")).Text +
    ", time = '" +
((TextBox)GridView2.Rows[GridView2.EditIndex].FindControl("TB201")).Text +
    "' WHERE idtechprocess = " +
data.Rows[GridView2.EditIndex][0].ToString() + ";", conn);
cmd.ExecuteNonQuery();
conn.Close();

```

```

        GridView2.EditIndex = -1;
        Grid_Fill2(0);
    }
    protected void GridView2_RowCommand(object sender, GridViewCommandEventArgs e)
    {
        if (e.CommandName.Equals("FooterInsert"))
        {
            MySqlConnection conn;
            DataTable data;
            MySqlDataAdapter da;
            MySqlCommandBuilder cb;

            String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false; charset=utf8");
            conn = new MySqlConnection(connStr);
            conn.Open();
            conn.ChangeDatabase("wood");

            da = new MySqlDataAdapter("SELECT * FROM parts", conn);
            data = new DataTable();
            da.Fill(data);
            string N = "";
            string temp;
            foreach (DataRow row in data.Rows)
            {
                temp = Label4.Text;
                if (temp == row[1].ToString())
                {
                    N = row[0].ToString();
                    break;
                }
            }
            string q = ((TextBox)GridView2.FooterRow.FindControl("TB12")).Text;
            string q2 = ((TextBox)GridView2.FooterRow.FindControl("TB22")).Text;
            string q3 = ((TextBox)GridView2.FooterRow.FindControl("TB32")).Text;
            string q4 = ((TextBox)GridView2.FooterRow.FindControl("TB42")).Text;
            string q5 = ((TextBox)GridView2.FooterRow.FindControl("TB52")).Text;
            string q6 = ((TextBox)GridView2.FooterRow.FindControl("TB62")).Text;
            string q7 = ((TextBox)GridView2.FooterRow.FindControl("TB72")).Text;
            string q8 = (((DropDownList)GridView2.FooterRow.FindControl("DDL82")).SelectedIndex +
1).ToString();

            MySqlCommand cmd = new MySqlCommand("INSERT INTO `wood`.`techprocess` (`part`,
`stage`, `PD`, `PW`, `PT`, `pcount`, `pvolume`, `pmp`, `idverstat`, `idverstatoper`, `CD`, `CW`,
`CT`, `ccount`, `cvolume`, `cmp`, `LD`, `LW`, `LT`, `lcount`, `lvolume`, `time`)
+ " VALUES (" + N + ","
+
((TextBox)GridView2.FooterRow.FindControl("TB12")).Text + ","
+
((TextBox)GridView2.FooterRow.FindControl("TB22")).Text + ","
+
((TextBox)GridView2.FooterRow.FindControl("TB32")).Text + ","
+
((TextBox)GridView2.FooterRow.FindControl("TB42")).Text + ","
+
((TextBox)GridView2.FooterRow.FindControl("TB52")).Text + ","
+
((TextBox)GridView2.FooterRow.FindControl("TB62")).Text + ","
+
((TextBox)GridView2.FooterRow.FindControl("TB72")).Text + ","
+
(((DropDownList)GridView2.FooterRow.FindControl("DDL82")).SelectedIndex + 1) + ","

```

```

        cmd.ExecuteNonQuery();
        conn.Close();
        Grid_Fill2(0);
    }
}

```

Реалізація обробки подій для інтерфейсу для роботи з звітами включає наступний код:

```

public partial class Reports : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Signout_Click(object sender, EventArgs e)
    {
        FormsAuthentication.SignOut();
        Response.Redirect("login.aspx");
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        MySqlConnection conn;
        DataTable data;
        MySqlDataAdapter da;
        MySqlCommandBuilder cb;

        String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false");
        conn = new MySqlConnection(connStr);
        conn.Open();
        conn.ChangeDatabase("wood");
        da = new MySqlDataAdapter("SELECT N, Date FROM orders", conn);
        data = new DataTable();
        da.Fill(data);
        string N = "3";

        data = new DataTable();
        da = new MySqlDataAdapter("SELECT orders.N, products.name, colors.name,
orderstoproducts.counts FROM orders, products, colors, orderstoproducts WHERE
orderstoproducts.order = " + N.ToString() + " AND orderstoproducts.order = orders.idorders AND
orderstoproducts.product = products.idproducts AND orderstoproducts.color = colors.idcolors ORDER
BY orderstoproducts.idorderstoproducts", conn);
        da.Fill(data);
        for (int i = 0; i < data.Rows.Count; i++)
        {
            data.Rows[i][0] = i + 1;
        }
        int sum = 0;
        foreach (DataRow row in data.Rows)
            sum += Convert.ToInt32(row[3].ToString());
        string[] str = new string[4];
        str[0] = null;
        str[1] = "Разом";
        str[2] = sum.ToString();
        str[3] = null;
        data.Rows.Add(str);

        ReportViewer1.Reset();
        ReportDataSource rds = new ReportDataSource("DataSet1", data);
    }
}

```

```
ReportViewer1.LocalReport.DataSources.Add(rds);
ReportViewer1.LocalReport.ReportPath = "Report.rdlc";
ReportViewer1.LocalReport.Refresh();

ReportViewer1.LocalReport.Render()

conn.Close();
}
}
```

Реалізація спирається на роботі з об'єктом ReportViewer1, який представляє собою контейнер для файлів .rdlc, які містять шаблони звітів.

## ВИСНОВКИ

В результаті розробки веб-орієнтованої інформаційної системи управління технічним документообігом для компанії з виготовлення меблів було самостійно розглянуто та освоєно технології у веб-програмуванні, а саме:

- ASP.NET
- ADO.NET
- Bootstrap
- jQuery

Розроблена інформаційна система здатна реалізувати поставлені перед нею задачі, містить максимально дружелюбний інтерфейс користувача і весь необхідний комплекс інструментів для спрощення та автоматизація технічного документообігу підприємства.

Хоча ІС відповідає завданню і готова до використання, проте нічого досконалого не існує, тому рекомендується поступове і в тестовому режимі впровадження таких систем.

Напрямами для вдосконалення можуть бути -інтерфейс користувача та розширення функціоналу ІС. У випадку інтерфейсу користувача, покращення мають базуватися безпосередньо на відгуках персоналу, який буде працювати з системою. А у випадку розширення функціоналу – на розвитку підприємства і його технічної бази.

## ПЕРЕЛІК ПОСИЛАНЬ

1. В. Васвані. MySQL: використання і адміністрування - М.: «Пітер», 2011. - 368 с.
2. Стів Суерінг, Тім Конверс, Джойс Парк. PHP і MySQL. Біблія програміста, 2-е видання - М.: «Діалектика», 2010. - 912 с.
3. Роберт Шелдон, Джоффри Мой. MySQL 5: базовий курс - М.: «Діалектика», 2007. - 880 с.
4. Кузнєцов Максим, Сімдянов Ігор. MySQL на прикладах. - Спб.: «БХВ-Петербург», 2008. - 952 с.
5. Тереза Нейл, Білл Скотт Проектування веб-інтерфейсів - Москва, Символ-Плюс, 2010. - 352 с.
6. Дж. Берд Веб-дизайн. Керівництво розробника - П.: «Пітер», 2012. - 224 с.
7. Ітан Маркотт Чуйний веб-дизайн - М.: Манн, Іванов та Фербер, 2012. - 159 с.
8. Е. Фрімен, Е. Фрімен Вивчаємо HTML, XHTML і CSS - П.: «Пітер», 2010. - 656 с.
9. Пітер Лабберс, Брайан Олберс, Френк Салім. HTML5 для професіоналів: потужні інструменти для розробки сучасних веб-додатків - М.: «Вільямс», 2011. - 272 с.
10. Стівен Шафер. HTML, XHTML і CSS. Біблія користувача, 5-е видання - М.: «Діалектика», 2010. - 656 с.
11. Крістофер Шмітт. CSS. Рецепти програмування - СПб.: БХВ-Петербург, 2007. - 592 с.
12. Адамс Кріс Адміністрування сервера IIS 7. - М.: Біном, 2010. - 362 с.
13. Метью Мак-Дональд, Адам Фрімен, Маріо Шпуста Microsoft ASP.NET 4.0 із прикладами на C # 2010 для професіоналів, 4-е видання - М.: «Вільямс», 2011. - 1424 с.

14. Даніель Арсеновскі Рефакторинг в С # і ASP.NET для професіоналів - М.: «Діалектика», 2009. - 528 с.
15. Роб Камерон, Дейл Михалик ASP.NET 3.5, компоненти AJAX і серверні елементи управління для професіоналів - М.: «Вільямс», 2009. - 608 с.
16. Сахіл Малік Microsoft ADO.NET 2.0 для професіоналів - М.: «Вільямс», 2006. - 560 с.
17. Девід Сєпа - Microsoft ADO.NET: "Російська Редакція", 2003. - 120 с.
18. Крістіан Нейгел С # 5.0 і платформа. NET 4.5 для професіоналів - М.: «Діалектика», 2013. - 1440 с.
19. А. Хейлсберг, М. Торгерсен, С. Вілтамут, П. Голд Мова програмування С # Класика Computers Science. 4-е видання - СПб.: «Пітер», 2012. - 784 с.
20. Е. Стілмен, Дж. Грін Вивчаємо С #. 2-е видання - СПб.: «Пітер», 2012. - 704 с.
21. Ендрю Троелсен Мова програмування С # 5.0 і платформа. NET 4.5, 6-е видання - М.: «Вільямс», 2013. - 1312 с.
22. Джозеф Албахарі, Бен Албахарі С # 5.0. Довідник. Повний опис мови - М.: «Вільямс», 2013. - 1008 с.
23. Герберт Шилдт. С # 4.0: повне керівництво - М.: «Вільямс», 2010. - С. 1056.
24. Джон Скит. С #: програмування для професіоналів, 2-е видання - М.: «Вільямс», 2011. - 544 с.
25. Крістіан Нейгел, Карлі Уотсон та ін Visual С # 2010: повний курс - М.: Діалектика, 2010 – 369 с.
26. Джеффри Ріхтер CLR via С #. Програмування на платформі. NET Framework 4.0 мовою С #. 3-е изд - СПб.: Питер, 2011 – 1308 с.

## Додаток А

### Настанови користувача

#### А.1 Призначення інформаційної системи

Головним завданням є розроблення інформаційної системи управління технічним документообігом малого підприємства, створення та організація бази даних замовлень та продуктів компанії, доступ до якої буде здійснюватися за допомогою мережі Інтернет. Система забезпечує зберігання всієї інформації в єдиній глобальній базі даних, з якою працює програмне забезпечення, що дозволяє здійснювати внесення, зберігання, обробку даних і отримання передбачених звітів.

#### А.2 Початок роботи з інформаційною системою

Для початку роботи з системою необхідно:

1. Відкрити веб-браузер;
2. В адресному рядку веб-браузера вести веб-адрес компанії (наприклад, woodplannig.com);
3. На сторінці, що відкрилася слід ввести отриманий від системного адміністратора підприємства логін та пароль;
4. Після загрузки головної сторінки можна починати роботу з системою.

#### А.3 Інструкція користувача

Для роботи із замовленнями слід перейти на сторінку “Програма виробництва”.

Для створення нового замовлення необхідно:

1. Нажати на кнопку “Сформувати”;
2. Ввести необхідну інформацію в поля(замовник, дата початку виконання, термін виконання, тощо);



3. Після введення даних, слід надати на кнопку “Зберегти”;
4. Після збереження, нове замовлення появиться в списку замовлень.

Для редагування інформації про замовлення необхідно:

1. Змінити необхідну інформації в полях з відповідною назвою;
2. Нажати на кнопку “Зберегти”.

Для додавання виробу до переліку замовлених необхідно:

1. В останньому рядку таблиці з переліком виробів в замовленні, необхідно вибрати необхідний виріб, його характеристики та вибрати кількість;
2. Щоб додати, слід натиснути на кнопку з зображенням зеленого плюса, що знаходиться в тому ж рядку;

Для видалення виробу з переліку замовлених необхідно:

1. У відповідному рядку таблиці з переліком замовлених виробів слід натиснути на кнопку із зображення червоного смітника.

Для редагування виробу з переліку замовлених необхідно:

1. У відповідному рядку таблиці з переліком замовлених виробів слід натиснути на кнопку із зображення синього олівця.
2. Внести редакції у відповідні поля замовленого виробу;
3. Для підтвердження редагування, у відповідному рядку таблиці з переліком замовлених виробів слід натиснути на кнопку із зображення зелених стрілок.
4. Для відміни редагування, у відповідному рядку таблиці з переліком замовлених виробів слід натиснути на кнопку із зображення червоного знаку стоп.

Для роздруку інформації про замовлення необхідно:

- Або натиснути на кнопку “Друк”;
- Або перейти на веб-сторінку “Звіти”;

Для роботи із виробами, деталями та їх технологічними маршрутами слід перейти на сторінку “Вироби”.

Для створення нового виробу необхідно:

1. Нажати на кнопку “Сформувати”;
2. Ввести необхідну інформацію в поля(категорія, назва, код);
3. Після введення даних, слід надати на кнопку “Зберегти”;
4. Після збереження, новий виріб з’явиться в списку виробів.

Для редагування інформації про виріб необхідно:

1. Змінити необхідну інформації в полях з відповідною назвою;
2. Нажати на кнопку “Зберегти”.

Для додавання деталі до переліку необхідних у виробництві виробу необхідно:

1. В останньому рядку таблиці з переліком деталей, необхідно вибрати необхідну деталь та її характеристики;
2. Щоб додати, слід натиснути на кнопку з зображенням зеленого плюса, що знаходиться в тому ж рядку;

Для видалення деталі з переліку необхідних у виробництві виробу необхідно:

1. У відповідному рядку таблиці з переліком деталей слід натиснути на кнопку із зображення червоного смітника.

Для редагування деталі з переліку необхідних у виробництві виробу необхідно:

1. У відповідному рядку таблиці з переліком деталей слід натиснути на кнопку із зображення синього олівця.
2. Внести редакції у відповідні поля деталі;
3. Для підтвердження редагування, у відповідному рядку таблиці з переліком деталей слід натиснути на кнопку із зображення зелених стрілок.
4. Для відміни редагування, у відповідному рядку таблиці з переліком деталей слід натиснути на кнопку із зображення червоного знаку стоп.

Для перегляду технологічного маршруту необхідно натиснути на посилання “Переглянути” у колонці “Техпроцес” відповідного рядка таблиці з деталями.

Робота з даною інформацією виконується аналогічним чином до роботи з виробами.

Для роздруку інформації про замовлення необхідно:

- Або натиснути на кнопку “Друк”;
- Або перейти на веб-сторінку “Звіти”;



## Додаток Б

### Програмний код

#### Б.1 Модуль “Програма виробництва”

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="Default3" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<link href="../../css/jquery-ui-1.10.4.custom.css" rel="stylesheet" />
<link href="../../css/bootstrap.css" rel="stylesheet" />
<link href="../../css/bootstrap-theme.css" rel="stylesheet" />
<link href="../../css/additional.css" rel="stylesheet" />
<script type="text/javascript" src="../../js/jquery-1.10.2.js"></script>
<script type="text/javascript" src="../../js/jquery-ui-1.10.4.js"></script>
<script type="text/javascript" src="../../js/bootstrap.js"></script>
</head>
<body>
<form runat="server">
<div class="panel panel-info" style="margin: 2% auto; width: 1280px">
<div class="panel-heading">
<ul class="nav nav-pills nav-justified">
<li><a href="Default.aspx" class="cl"></a></li>
<li class="active"><a href="Default.aspx">Програма виробництва</a></li>
<li><a href="Products.aspx">Вироби</a></li>
<li><a href="Meansofproduction.aspx">Засоби виробництва</a></li>
<li><a href="Options.aspx">Налаштування</a></li>
<li><a href="Reports.aspx">Звіти</a></li>
<li><div style="float:right;"><asp:Button ID="Button4" runat="server" Text="X" CssClass="btn btn-
danger" OnClick="Signout_Click" /></div></li>
</ul>
</div>
<div class="panel-body">
<table style="margin:0 auto;">
<tr>
<td rowspan="3" style="padding:5px; vertical-align:top;">
<asp:Label ID="Label9" runat="server" Text="Документи:"></asp:Label>
<br />
<asp:ListBox ID="ListBox1" runat="server" CssClass="LBBorder" AutoPostBack="true"
OnSelectedIndexChanged="ListBox1_SelectedIndexChanged" ></asp:ListBox>
</td>
<td style="padding:15px;">
<table style="margin:0 auto;">
<tr><td style="text-align:right; padding:5px;"><asp:Label ID="Label11" runat="server" Text="Дата
"></asp:Label></td><td><asp:TextBox ID="TextBox1" runat="server" Enabled="false" CssClass="form-
control disabled" Height="30px"></asp:TextBox></td></tr>
<tr><td style="text-align:right; padding:5px;"><asp:Label ID="Label12" runat="server"
Text="Замовлення № "></asp:Label></td><td><asp:TextBox ID="TextBox2" runat="server"
Enabled="false" CssClass="form-control disabled" Height="30px"></asp:TextBox></td></tr>
<tr><td style="text-align:right; padding:5px;"><asp:Label ID="Label13" runat="server"
Text="Замовник "></asp:Label></td><td><asp:TextBox ID="TextBox3" runat="server" CssClass="form-
control" Height="30px"></asp:TextBox></td></tr>
<tr><td style="text-align:right; padding:5px;"><asp:Label ID="Label14" runat="server" Text="Дата
початку виконання "></asp:Label></td><td><asp:TextBox ID="TextBox4" runat="server"
CssClass="form-control" Height="30px"></asp:TextBox></td></tr>
</table>
</td>
</tr>
</table>
</div>
</form>
```

```

<tr><td style="text-align:right; padding:5px;"><asp:Label ID="Label5" runat="server" Text="Термін
виконання "></asp:Label></td><td><asp:TextBox ID="TextBox5" runat="server" Enabled="false"
CssClass="form-control disabled" Height="30px"></asp:TextBox></td></tr>
<tr><td style="text-align:right; padding:5px;"><asp:Label ID="Label6" runat="server" Text="Дата
завершення виконання "></asp:Label></td><td><asp:TextBox ID="TextBox6" runat="server"
Enabled="false" CssClass="form-control disabled" Height="30px"></asp:TextBox></td></tr>
</table>
</td>
</tr>
<tr>
<td style="padding: 5px;">
<asp:GridView ID="GridView1" runat="server" AllowPaging="false" CssClass="table-bordered table-
hover table-center" AutoGenerateColumns="False" ShowFooter="true" ShowHeaderWhenEmpty="True"
OnRowEditing="GridView1_RowEditing" OnRowCancelingEdit="GridView1_RowCancelingEdit"
OnRowDeleting="GridView1_RowDeleting" OnRowUpdating="GridView1_RowUpdating"
OnRowCommand="GridView1_RowCommand">
<Columns>
<asp:TemplateField HeaderText="№" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:Label ID="Label1" runat="server" Text='<%# Bind("N") %>'></asp:Label>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="Label2" runat="server" Text='<%# Bind("N") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Найменування" ItemStyle-Width="225px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:DropDownList ID="Label4" runat="server" CssClass="textbox-little"></asp:DropDownList>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="Label5" runat="server" Text='<%# Bind("name") %>' CssClass="LPad"></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:DropDownList ID="Label14" runat="server" CssClass="textbox-little"></asp:DropDownList>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Коліп" ItemStyle-CssClass="MCELL" ItemStyle-Width="125px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:DropDownList ID="Label7" runat="server" CssClass="textbox-little MCELL"></asp:DropDownList>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="Label8" runat="server" Text='<%# Bind("name1") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:DropDownList ID="Label15" runat="server" CssClass="textbox-little MCELL"></asp:DropDownList>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Кількість" ItemStyle-CssClass="MCELL" ItemStyle-Width="80px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="Label10" runat="server" Text='<%# Bind("counts") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>

```

```

<asp:Label ID="Label11" runat="server" Text='<%# Bind("counts") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="Label16" runat="server" Enabled="true" CssClass="textbox-little
MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Настройки">
<HeaderStyle CssClass="MTABLE M4" />
<FooterStyle CssClass="TFooter" />
<itemtemplate>
<asp:linkbutton id="btnEdit" runat="server" causesvalidation="false" commandname="Edit" text="">
<asp:Image ID="Image1" runat="server" ImageUrl="~/images/edit.png"
CssClass="IMG"/></asp:linkbutton>
<asp:linkbutton id="btnDelete" runat="server" causesvalidation="false" commandname="Delete"
text="" >
<asp:Image ID="Image2" runat="server" ImageUrl="~/images/delete.png"
CssClass="IMG"/></asp:linkbutton>
</itemtemplate>
<edititemtemplate>
<asp:linkbutton id="btnUpdate" runat="server" commandname="Update" text="" >
<asp:Image ID="Image3" runat="server" ImageUrl="~/images/update.png"
CssClass="IMG"/></asp:linkbutton>
<asp:linkbutton id="btnCancel" runat="server" causesvalidation="false" commandname="Cancel"
text="" >
<asp:Image ID="Image1" runat="server" ImageUrl="~/images/cancel.png"
CssClass="IMG"/></asp:linkbutton>
</edititemtemplate>
</footertemplate>
<asp:linkbutton id="btnInsert" runat="server" commandname="FooterInsert" text=""
CssClass="MARGIN">
<asp:Image ID="Image1" runat="server" ImageUrl="~/images/insert.png" CssClass="IMG"
/></asp:linkbutton>
</footertemplate>
</asp:TemplateField>
</Columns>
<EmptyDataTemplate>
No records
</EmptyDataTemplate>
</asp:GridView>
</td>
</tr>
<tr>
<td>
<table class="table-center">
<tr>
<td class="M6 marg"><asp:linkbutton id="Linkbutton1" runat="server"
OnClick="Signout_Click"><button class="btn btn-default mybtn"
onclick="javascript: __doPostBack('Linkbutton1','')">Сформувати</button></asp:linkbutton></td>
<td class="M6 marg"><asp:linkbutton id="Linkbutton2" runat="server"
OnClick="Signout_Click"><button class="btn btn-default mybtn"
onclick="javascript: __doPostBack('Linkbutton2','')">Друк</button></asp:linkbutton></td>
<td class="M6 marg"><asp:linkbutton id="Linkbutton3" runat="server"
OnClick="Signout_Click"><button class="btn btn-default mybtn"
onclick="javascript: __doPostBack('Linkbutton3','')">Зберігти</button></asp:linkbutton></td>
</tr>
</table>
</td>
</tr>
</table>

```

```

</div>
</div>
</form>
</body>
</html>

```

## Б.2 Модуль “Вироби”

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Products.aspx.cs" Inherits="Products" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<link href=" ../css/jquery-ui-1.10.4.custom.css" rel="stylesheet" />
<link href=" ../css/bootstrap.css" rel="stylesheet" />
<link href=" ../css/bootstrap-theme.css" rel="stylesheet" />
<link href=" ../css/additional.css" rel="stylesheet" />
<script type="text/javascript" src=" ../js/jquery-1.10.2.js"></script>
<script type="text/javascript" src=" ../js/jquery-ui-1.10.4.js"></script>
<script type="text/javascript" src=" ../js/bootstrap.js"></script>
</head>
<body>
<form id="form1" runat="server">
<div>
<div class="panel panel-info" style="margin: 2% auto; width: 1280px">
<div class="panel-heading">
<ul class="nav nav-pills nav-justified">
<li><a href="Default.aspx" class="cl"></a></li>
<li><a href="Default.aspx">Програма виробництва</a></li>
<li class="active"><a href="Products.aspx">Вироби</a></li>
<li><a href="Meansofproduction.aspx">Засоби виробництва</a></li>
<li><a href="Options.aspx">Налаштування</a></li>
<li><a href="Reports.aspx">Звіти</a></li>
<li><div style="float:right;"><asp:Button ID="Button4" runat="server" Text="X" CssClass="btn btn-
danger" OnClick="Signout_Click"/></div></li>
</ul>
</div>
<div class="panel-body">
<table style="margin:0 auto;">
<tr>
<td rowspan="3" style="padding:5px; vertical-align:top;">
<asp:Label ID="Label9" runat="server" Text="Найменування виробів:"></asp:Label>
<br />
<asp:ListBox ID="ListBox1" runat="server" CssClass="LBBorder" AutoPostBack="true"
OnSelectedIndexChanged="ListBox1_SelectedIndexChanged" ></asp:ListBox>
</td>
<td style="padding:15px;">
<table style="margin:0 auto;">
<tr><td style="text-align:right; padding:5px;"><asp:Label ID="Label11" runat="server"
Text="Категорія "></asp:Label></td><td><asp:TextBox ID="TextBox1" runat="server" Enabled="true"
CssClass="form-control" Height="30px" Width="300px"></asp:TextBox></td></tr>
<tr><td style="text-align:right; padding:5px;"><asp:Label ID="Label12" runat="server" Text="Назва
"></asp:Label></td><td><asp:TextBox ID="TextBox2" runat="server" Enabled="true" CssClass="form-
control" Height="30px" Width="300px"></asp:TextBox></td></tr>
<tr><td style="text-align:right; padding:5px;"><asp:Label ID="Label13" runat="server" Text="Код
товару "></asp:Label></td><td><asp:TextBox ID="TextBox3" runat="server" CssClass="form-control"
Height="30px" Width="300px"></asp:TextBox></td></tr>
</table>
</td>
</tr>

```

```

<tr>
<td style="padding: 5px;">
<asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
<ContentTemplate>
<asp:GridView ID="GridView1" runat="server" AllowPaging="false" CssClass="table-bordered table-
hover table-center" AutoGenerateColumns="False" ShowFooter="true" ShowHeaderWhenEmpty="True"
OnRowEditing="GridView1_RowEditing" OnRowCancelingEdit="GridView1_RowCancelingEdit"
OnRowDeleting="GridView1_RowDeleting" OnRowUpdating="GridView1_RowUpdating"
OnRowCommand="GridView1_RowCommand">
<Columns>
<asp:TemplateField HeaderText="Детали" ItemStyle-CssClass="M">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:DropDownList ID="DDL11" runat="server" ></asp:DropDownList>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L11" runat="server" Text='<%# Bind("name") %>' CssClass="LPad"></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:DropDownList ID="DDL12" runat="server" CssClass="textbox-little"></asp:DropDownList>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Позначення деталей" ItemStyle-CssClass="MCELL M" ItemStyle-
Width="90px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L21" runat="server" Text='<%# Bind("code") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="К-сть деталей" ItemStyle-CssClass="MCELL" ItemStyle-Width="75px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB31" runat="server" Text='<%# Bind("count") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L31" runat="server" Text='<%# Bind("count") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB32" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Матеріал деталі" ItemStyle-CssClass="MCELL" ItemStyle-
Width="75px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:DropDownList ID="DDL41" runat="server" CssClass="textbox-little MCELL"></asp:DropDownList>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L41" runat="server" Text='<%# Bind("name1") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:DropDownList ID="DDL42" runat="server" CssClass="textbox-little MCELL"></asp:DropDownList>
</FooterTemplate>

```



```

</asp:TemplateField>
<asp:TemplateField HeaderText="Довжина" ItemStyle-CssClass="MCELL" ItemStyle-Width="70px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L51" runat="server" Text='<%# Bind("length") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Ширина" ItemStyle-CssClass="MCELL" ItemStyle-Width="65px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L61" runat="server" Text='<%# Bind("width") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Товщина" ItemStyle-CssClass="MCELL" ItemStyle-Width="70px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L71" runat="server" Text='<%# Bind("thickness") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="06'ем" ItemStyle-CssClass="MCELL" ItemStyle-Width="70px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L81" runat="server" Text='<%# Bind("volume") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Техпроцес" ItemStyle-CssClass="MCELL" ItemStyle-Width="90px">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<ItemTemplate>
<asp:LinkButton ID="LinkButton1" runat="server" CommandName="detail" CommandArgument='<%#
(((GridViewRow) Container).RowIndex.ToString() + "_" + Eval("name"))
%>'>Переглянути</asp:LinkButton>
</ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Настройки">
<HeaderStyle CssClass="MTABLE M4" />
<FooterStyle CssClass="TFooter" />
<itemtemplate>
<asp:linkbutton id="btnEdit" runat="server" causesvalidation="false" commandname="Edit" text="">
<asp:Image ID="Image1" runat="server" ImageUrl="~/images/edit.png"
CssClass="IMG"/></asp:linkbutton>
<asp:linkbutton id="btnDelete" runat="server" causesvalidation="false" commandname="Delete"
text="" >

```

```

<asp:Image ID="Image2" runat="server" ImageUrl="~/images/delete.png"
CssClass="IMG"/></asp:linkbutton>
</itemtemplate>
<edititemtemplate>
<asp:linkbutton id="btnUpdate" runat="server" commandname="Update" text="" >
<asp:Image ID="Image3" runat="server" ImageUrl="~/images/update.png"
CssClass="IMG"/></asp:linkbutton>
<asp:linkbutton id="btnCancel" runat="server" causesvalidation="false" commandname="Cancel"
text="" >
<asp:Image ID="Image1" runat="server" ImageUrl="~/images/cancel.png"
CssClass="IMG"/></asp:linkbutton>
</edititemtemplate>
<footertemplate>
<asp:linkbutton id="btnInsert" runat="server" commandname="FooterInsert" text=""
CssClass="MARGIN">
<asp:Image ID="Image1" runat="server" ImageUrl="~/images/insert.png" CssClass="IMG"
/></asp:linkbutton>
</footertemplate>
</asp:TemplateField>
</Columns>
<EmptyDataTemplate>
No records
</EmptyDataTemplate>
</asp:GridView>
</ContentTemplate>
</asp:UpdatePanel>
</td>
</tr>
<tr>
<td>
<table class="table-center">
<tr>
<td class="M6 marg"><asp:linkbutton id="Linkbutton1" runat="server"
OnClick="Signout_Click"><button class="btn btn-default mybtn"
onclick="javascript:__doPostBack('Linkbutton1','')">Сформувати</button></asp:linkbutton></td>
<td class="M6 marg"><asp:linkbutton id="Linkbutton2" runat="server"
OnClick="Signout_Click"><button class="btn btn-default mybtn"
onclick="javascript:__doPostBack('Linkbutton2','')">Друк</button></asp:linkbutton></td>
<td class="M6 marg"><asp:linkbutton id="Linkbutton3" runat="server"
OnClick="Signout_Click"><button class="btn btn-default mybtn"
onclick="javascript:__doPostBack('Linkbutton3','')">Зберегти</button></asp:linkbutton></td>
</tr>
</table>
</td>
</tr>
</table>
</div>
</div>
<div class="modal fade" id="currentdetail" tabindex="-1" role="dialog" aria-
labelledby="myModalLabel" aria-hidden="true">
<div class="modal-dialog modal-lg" style="width:1280px;">
<div class="modal-content">
<asp:UpdatePanel ID="UpdatePanel2" runat="server">
<ContentTemplate>
<div class="modal-header">
<button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
<h3 class="modal-title" id="myModalLabel">Технологічний маршрут</h3>
<br />
<h4 class="modal-title" ><asp:Label ID="Label4" runat="server" Text="Label"></asp:Label></h4>
<asp:HiddenField ID="HiddenField1" runat="server" />

```

```

</div>
<div class="modal-body">
<asp:GridView ID="GridView2" runat="server" AllowPaging="false" CssClass="table-bordered table-
hover table-center" AutoGenerateColumns="False" ShowFooter="true" ShowHeaderWhenEmpty="True"
OnRowEditing="GridView2_RowEditing" OnRowCancelingEdit="GridView2_RowCancelingEdit"
OnRowDeleting="GridView2_RowDeleting" OnRowUpdating="GridView2_RowUpdating"
OnRowCommand="GridView2_RowCommand">
<Columns>
<asp:TemplateField HeaderText="Этапи" ItemStyle-CssClass="M" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB11" runat="server" Text='<%# Bind("stage") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L11" runat="server" Text='<%# Bind("stage") %>' ></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB12" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Д" ItemStyle-CssClass="MCELL M" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB21" runat="server" Text='<%# Bind("PD") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L21" runat="server" Text='<%# Bind("PD") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB22" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Ш" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB31" runat="server" Text='<%# Bind("PW") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L31" runat="server" Text='<%# Bind("PW") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB32" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Т" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB41" runat="server" Text='<%# Bind("PT") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L41" runat="server" Text='<%# Bind("PT") %>'></asp:Label>

```

```

</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB42" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="К-сть шт." ItemStyle-CssClass="MCELL" ItemStyle-Width="50px"
ItemStyle-HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB51" runat="server" Text='<%# Bind("pcount") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L51" runat="server" Text='<%# Bind("pcount") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB52" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="06'ем" ItemStyle-CssClass="MCELL" ItemStyle-Width="60px"
ItemStyle-HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB61" runat="server" Text='<%# Bind("pvolume") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L61" runat="server" Text='<%# Bind("pvolume") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB62" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="м/п" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB71" runat="server" Text='<%# Bind("pmp") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L71" runat="server" Text='<%# Bind("pmp") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB72" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Верстат" ItemStyle-CssClass="MCELL" ItemStyle-Width="300px"
ItemStyle-HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:DropDownList ID="DDL81" runat="server" CssClass="textbox-little"></asp:DropDownList>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L81" runat="server" Text='<%# Bind("name") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:DropDownList ID="DDL82" runat="server" CssClass="textbox-little"></asp:DropDownList>
</FooterTemplate>

```

```

</asp:TemplateField>
<asp:TemplateField HeaderText="Операция" ItemStyle-CssClass="MCELL" ItemStyle-Width="300px"
ItemStyle-HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:DropDownList ID="DDL221" runat="server" CssClass="textbox-little"></asp:DropDownList>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L211" runat="server" Text='<%# Bind("name1") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:DropDownList ID="DDL212" runat="server" CssClass="textbox-little"></asp:DropDownList>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Д" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB91" runat="server" Text='<%# Bind("CD") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L91" runat="server" Text='<%# Bind("CD") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB92" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Ш" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB101" runat="server" Text='<%# Bind("CW") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L101" runat="server" Text='<%# Bind("CW") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB102" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Т" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB111" runat="server" Text='<%# Bind("CT") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L111" runat="server" Text='<%# Bind("CT") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB112" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="К-сть шт." ItemStyle-CssClass="MCELL" ItemStyle-Width="50px"
ItemStyle-HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />

```

```

<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB121" runat="server" Text='<%# Bind("ccount") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L121" runat="server" Text='<%# Bind("ccount") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB122" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="O6'εM" ItemStyle-CssClass="MCELL" ItemStyle-Width="60px"
ItemStyle-HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB131" runat="server" Text='<%# Bind("cvolume") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L131" runat="server" Text='<%# Bind("cvolume") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB132" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="M/n" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB141" runat="server" Text='<%# Bind("cmp") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L141" runat="server" Text='<%# Bind("cmp") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB142" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Д" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB151" runat="server" Text='<%# Bind("LD") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L151" runat="server" Text='<%# Bind("LD") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB152" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Ш" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>

```

```

<asp:TextBox ID="TB161" runat="server" Text='<%# Bind("LW") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L161" runat="server" Text='<%# Bind("LW") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB162" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="T" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB171" runat="server" Text='<%# Bind("LT") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L171" runat="server" Text='<%# Bind("LT") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB172" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="K-сть шт." ItemStyle-CssClass="MCELL" ItemStyle-Width="50px"
ItemStyle-HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB181" runat="server" Text='<%# Bind("lcount") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L181" runat="server" Text='<%# Bind("lcount") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB182" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="06'ем" ItemStyle-CssClass="MCELL" ItemStyle-Width="60px"
ItemStyle-HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB191" runat="server" Text='<%# Bind("lvolume") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>
<ItemTemplate>
<asp:Label ID="L191" runat="server" Text='<%# Bind("lvolume") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB192" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="4ac" ItemStyle-CssClass="MCELL" ItemStyle-Width="50px" ItemStyle-
HorizontalAlign="Center">
<HeaderStyle CssClass="MTABLE " />
<FooterStyle CssClass="TFooter" />
<EditItemTemplate>
<asp:TextBox ID="TB201" runat="server" Text='<%# Bind("time") %>' CssClass="textbox-little
MCELL"></asp:TextBox>
</EditItemTemplate>

```

```

<ItemTemplate>
<asp:Label ID="L201" runat="server" Text='<%# Bind("time") %>'></asp:Label>
</ItemTemplate>
<FooterTemplate>
<asp:TextBox ID="TB202" runat="server" CssClass="textbox-little MCELL"></asp:TextBox>
</FooterTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Настройки">
<HeaderStyle CssClass="MTABLE M4" />
<FooterStyle CssClass="TFooter" />
<itemtemplate>
<asp:linkbutton id="btnEdit" runat="server" causesvalidation="false" commandname="Edit" text="">
<asp:Image ID="Image1" runat="server" ImageUrl="~/images/edit.png"
CssClass="IMG"/></asp:linkbutton>
<asp:linkbutton id="btnDelete" runat="server" causesvalidation="false" commandname="Delete"
text="" >
<asp:Image ID="Image2" runat="server" ImageUrl="~/images/delete.png"
CssClass="IMG"/></asp:linkbutton>
</itemtemplate>
<edititemtemplate>
<asp:linkbutton id="btnUpdate" runat="server" commandname="Update" text="" >
<asp:Image ID="Image3" runat="server" ImageUrl="~/images/update.png"
CssClass="IMG"/></asp:linkbutton>
<asp:linkbutton id="btnCancel" runat="server" causesvalidation="false" commandname="Cancel"
text="" >
<asp:Image ID="Image4" runat="server" ImageUrl="~/images/cancel.png"
CssClass="IMG"/></asp:linkbutton>
</edititemtemplate>
<footertemplate>
<asp:linkbutton id="btnInsert" runat="server" commandname="FooterInsert" text=""
CssClass="MARGIN">
<asp:Image ID="Image5" runat="server" ImageUrl="~/images/insert.png" CssClass="IMG"
/></asp:linkbutton>
</footertemplate>
</asp:TemplateField>
</Columns>
<EmptyDataTemplate>
No records
</EmptyDataTemplate>
</asp:GridView>
</div>
</ContentTemplate>
<Triggers>
<asp:AsyncPostBackTrigger ControlID="GridView1" EventName="RowCommand" />
</Triggers>
</asp:UpdatePanel>
</div>
</div>
</div>
</div>
</form>
</body>
</html>

```

### Б.3 Модуль “Логування”

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="login.aspx.cs" Inherits="login" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">

```



```

<title></title>
<link href="../css/jquery-ui-1.10.4.custom.css" rel="stylesheet" />
<link href="../css/bootstrap.css" rel="stylesheet" />
<link href="../css/bootstrap-theme.css" rel="stylesheet" />
<link href="../css/additional.css" rel="stylesheet" />
<script type="text/javascript" src="../js/jquery-1.10.2.js"></script>
<script type="text/javascript" src="../js/jquery-ui-1.10.4.js"></script>
<script type="text/javascript" src="../js/bootstrap.js"></script>
<link href="../css/signin.css" rel="stylesheet" />
</head>
<body>
<div class="container">
<form runat="server" class="form-signin">
<asp:Login ID="Login1" runat="server" OnAuthenticate="Login1_Authenticate" Width="300px"
FailureText="Ваш логін чи пароль не вірний.">
<LayoutTemplate>
<h2 class="form-signin-heading" style="text-align: center;">Вхід у систему:</h2>
<asp:TextBox ID="UserName" runat="server" CssClass="form-control" placeholder="Логін"
autofocus></asp:TextBox>
<asp:TextBox ID="Password" runat="server" CssClass="form-control" TextMode="Password"
placeholder="Пароль"></asp:TextBox>
<label class="checkbox"><asp:CheckBox ID="RememberMe" CssClass="checkbox" runat="server"
Text="Запам'ятати мене."/></label>
<label style="text-align:center; color:red;"><asp:Literal ID="FailureText" runat="server"
EnableViewState="False"></asp:Literal></label>
<asp:Button ID="LoginButton" runat="server" CommandName="Login" CssClass="btn btn-lg btn-primary
btn-block" Text="Увійти" ValidationGroup="Login1" />
</LayoutTemplate>
</asp:Login>
</form>
</div>
</body>
</html>

```

```

public partial class login : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Login1_Authenticate(object sender, AuthenticateEventArgs e)
    {
        if (login.LogIn(Login1.UserName, Login1.Password))
        {
            System.Web.Security.FormsAuthentication.RedirectFromLoginPage(Login1.UserName, true);
        }
    }
    protected static bool LogIn(string User, string Password)
    {
        MySqlConnection conn;
        DataTable data;
        MySqlDataAdapter da;
        MySqlCommandBuilder cb;

        String connStr = String.Format("server=localhost;user id=root; password=root;
database=mysql; pooling=false");
        conn = new MySqlConnection(connStr);
        conn.Open();
        conn.ChangeDatabase("wood");
    }
}

```

```

data = new DataTable();
da = new MySqlDataAdapter("SELECT * FROM users", conn);
da.Fill(data);
for (int i = 0; i < data.Rows.Count; i++)
{
    if (data.Rows[i][1].ToString() == User && data.Rows[i][2].ToString() == Password)
        return true;
}
return false;

conn.Close();
}
}

```

#### Б.4 База даних

```

CREATE TABLE `colors` (
`idcolors` int(11) NOT NULL,
`name` varchar(45) DEFAULT NULL,
`code` int(11) DEFAULT NULL,
PRIMARY KEY (`idcolors`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `materials` (
`idmaterials` int(11) NOT NULL,
`name` varchar(45) DEFAULT NULL,
PRIMARY KEY (`idmaterials`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `orders` (
`idorders` int(11) NOT NULL AUTO_INCREMENT,
`N` int(11) DEFAULT NULL,
`customer` varchar(45) DEFAULT NULL,
`Date` date DEFAULT NULL,
`BDate` date DEFAULT NULL,
`Time` int(11) DEFAULT NULL,
`EDate` date DEFAULT NULL,
PRIMARY KEY (`idorders`)

```

```
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `orderstoproducts` (
```

```
`idorderstoproducts` int(11) NOT NULL AUTO_INCREMENT,
```

```
`order` int(11) DEFAULT NULL,
```

```
`product` int(11) DEFAULT NULL,
```

```
`counts` int(11) DEFAULT NULL,
```

```
`color` int(11) DEFAULT NULL,
```

```
PRIMARY KEY (`idorderstoproducts`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `parts` (
```

```
`idparts` int(11) NOT NULL AUTO_INCREMENT,
```

```
`name` varchar(45) DEFAULT NULL,
```

```
`code` varchar(45) DEFAULT NULL,
```

```
`length` int(11) DEFAULT NULL,
```

```
`width` int(11) DEFAULT NULL,
```

```
`thickness` int(11) DEFAULT NULL,
```

```
`t_to_w` int(11) DEFAULT NULL,
```

```
`volume` double DEFAULT NULL,
```

```
PRIMARY KEY (`idparts`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `products` (
```

```
`idproducts` int(11) NOT NULL,
```

```
`name` varchar(45) DEFAULT NULL,
```

```
`code` int(11) DEFAULT NULL,
```

```
`category` varchar(45) DEFAULT NULL,
```

```
PRIMARY KEY (`idproducts`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `productstoparts` (
```

```
`idproductstoparts` int(11) NOT NULL AUTO_INCREMENT,
```

```

`product` int(11) DEFAULT NULL,
`part` int(11) DEFAULT NULL,
`count` int(11) DEFAULT NULL,
`material` int(11) DEFAULT NULL,
PRIMARY KEY (`idproductstoparts`)
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8;

CREATE TABLE `techprocess` (
`idtechprocess` int(11) NOT NULL AUTO_INCREMENT,
`part` int(11) DEFAULT NULL,
`stage` int(11) DEFAULT NULL,
`PD` int(11) DEFAULT NULL,
`PW` int(11) DEFAULT NULL,
`PT` int(11) DEFAULT NULL,
`pcount` int(11) DEFAULT NULL,
`pvolume` double DEFAULT NULL,
`pmp` double DEFAULT NULL,
`idverstat` int(11) DEFAULT NULL,
`idverstatoper` int(11) DEFAULT NULL,
`CD` int(11) DEFAULT NULL,
`CW` int(11) DEFAULT NULL,
`CT` int(11) DEFAULT NULL,
`ccount` int(11) DEFAULT NULL,
`cvolume` double DEFAULT NULL,
`cmp` double DEFAULT NULL,
`LD` int(11) DEFAULT NULL,
`LW` int(11) DEFAULT NULL,
`LT` int(11) DEFAULT NULL,
`lcount` int(11) DEFAULT NULL,
`lvolume` double DEFAULT NULL,

```

```
`time` time DEFAULT NULL,  
PRIMARY KEY (`idtechprocess`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `users` (  
`idUsers` int(11) NOT NULL,  
`Name` varchar(45) DEFAULT NULL,  
`Password` varchar(45) DEFAULT NULL,  
PRIMARY KEY (`idUsers`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `verstat` (  
`idverstat` int(11) NOT NULL,  
`name` varchar(45) DEFAULT NULL,  
PRIMARY KEY (`idverstat`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `verstatoper` (  
`idverstatoper` int(11) NOT NULL,  
`name` varchar(45) DEFAULT NULL,  
`idverstat` int(11) DEFAULT NULL,  
PRIMARY KEY (`idverstatoper`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

## Б.5 Стили модуля “Логуювання”

```
body {  
padding-top: 40px;  
padding-bottom: 40px;  
background-color: #eee;  
}  
  
.form-signin {  
max-width: 330px;  
padding: 15px;  
margin: 0 auto;  
}  
.form-signin .form-signin-heading,  
.form-signin .checkbox {  
margin-bottom: 10px;  
}  
.form-signin .checkbox {
```

```

    font-weight: normal;
}
.form-signin .form-control {
    position: relative;
    height: auto;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
    padding: 10px;
    font-size: 16px;
}
.form-signin .form-control:focus {
    z-index: 2;
}
.form-signin input[type="email"] {
    margin-bottom: -1px;
    border-bottom-right-radius: 0;
    border-bottom-left-radius: 0;
}
.form-signin input[type="password"] {
    margin-bottom: 10px;
    border-top-left-radius: 0;
    border-top-right-radius: 0;
}

```

## Б.6 Стилї інформаційної системи

```

body
{
    background-color: #eee
}
.LBBorder
{
    border-color: #bce8f1;
}
.IMG
{
    padding: 5px;
    padding-left: 15%;
}
.MARGIN
{
    padding-left: 25%;
}
.MTABLE
{
    text-align: center;
}
.M
{
    min-width: 30px;
}
.M2
{
    min-width: 50px;
}
.M3
{
    min-width: 75px;
}
.M4

```

```

{
    min-width: 100px;
}
.M5
{
    min-width: 150px;
}
.M6
{
    min-width: 200px;
}
.MCELL
{
    text-align:center;
}
.table-center
{
    margin:0 auto;
}
.texbox-little
{
    width: 100%;
    height: 24px;
}
.LPad
{
    padding-left: 5px;
}
.new_file
{
    width:24px;
    height:24px;
    background:url(../images/dnf3.png);
    background-size: 24px 24px;
    background-repeat: no-repeat;
    position:relative;
    top: 5px;
    right: 5px;
}
.print
{
    width:24px;
    height:24px;
    background:url(../images/print.png);
    background-size: 24px 24px;
    background-repeat: no-repeat;
    position:relative;
    top: -5px;
    right: 5px;
}
.save
{
    width:24px;
    height:24px;
    background:url(../images/save.png);
    background-size: 24px 24px;
    background-repeat: no-repeat;
    position:relative;
    top: 5px;
    right: 5px;
}
.mybtn
{

```

```

    width: 100%;
    height: 36px;
    font-size: 18px;
}
.mybtn2
{
    width: 75px;
    height: 75px;
    font-size: 18px;
    margin: 15px;
    border: 1px solid black;
}
.report_view
{
    border: 2px solid black;
}
.marg
{
    padding: 10px;
}
.nav > li > a.cl:hover {
    background-color: #d9edf7;
}
th
{
    background-color: #f5f5f5;
}
.TFooter
{
    background-color: #f5f5f5;
}
.btn-danger {
    background-image: -webkit-linear-gradient(top, #f36a66 0%, #d0403c 100%);
    background-image: linear-gradient(to bottom, #f36a66 0%, #d0403c 100%);
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffd9534f',
endColorstr='#ffc12e2a', GradientType=0);
    filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);
    background-repeat: repeat-x;
    border-color: #b92c28;
}
.btn-danger:hover,
.btn-danger:focus
{
    background-color: #d0403c;
    background-position: 0 -15px;
}

```