

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА
ФРАНКА**

Факультет прикладної математики та інформатики
(повне найменування назва факультету)

Кафедра дискретного аналізу та інтелектуальних систем
(повна назва кафедри)

ДИПЛОМНА РОБОТА

**СТВОРЕННЯ SAAS ПЛАТФОРМИ ДЛЯ АВТОМАТИЗАЦІЇ ПРОЦЕСУ
УКЛАДАННЯ УГОД**

Виконав: студент IV курсу, групи ПМі-43с
напряму підготовки (спеціальності)

122 «Комп'ютерні науки»

(шифр і назва спеціальності)

Петришин М.Ю.

(підпис)

(прізвище та ініціали)

Керівник

Васильків Т.І.

(підпис)

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет Прикладної математики та інформатики

Кафедра Дискретного аналізу та інтелектуальних систем

Спеціальність 122 «Комп'ютерні науки»

(шифр і назва)

«ЗАТВЕРДЖУЮ»

Завідувач кафедри _____

"31" серпня 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Петришину Максиму Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Створення SaaS платформи для автоматизації процесу укладання угод

керівник роботи асистент кафедри дискретного аналізу та інтелектуальних систем Васильків Тетяна Ігорівна

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені Вченою радою факультету від "13" вересня 2022 року № 15

2. Строк подання студентом роботи 13.06.2023р.

3. Вихідні дані до роботи Середовище розробки Visual Studio, Node Js, .NET, фреймворк React, база даних PostgreSQL, векторний сервіс Figma, UI бібліотека Material, React бібліотеки Signature, Google Login та Router, бібліотека Redux, бібліотека Redux Saga

4. Зміст дипломної роботи (перелік питань, які потрібно розробити) Розробити додаток для автоматизації підписання угод. Розібратись у тонкощах взаємодії React з сервісами Google

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Діаграма класів, архітектура серверної частини, архітектура клієнтської частини, знімки з комп'ютера та мобільного пристрою, для демонстрації додатку

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання **31 серпня 2022 р.**

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Створення проекту для клієнтської частини додатку.	14.10.2022	
2.	Дослідження бібліотек для підписання. PDF Tron / Web-viewer.	18.10.2022	
3.	Створення проекту для серверної частини додатку.	27.10.2022	
4.	Написання кінцевих точок для користувачів.	05.11.2022	
5.	Додавання нових кінцевих точок для угод. Налаштування контролерів. Додавання автентифікації.	14.11.2022	
6.	Тестування серверної частини та її взаємодії з клієнтом.	29.11.2022	
7.	Підключення Google Auth. Налаштування автентифікації додатку з іншими сервісами.	12.02.2023	
8.	Написання стилів для розмітки додатку.	20.02.2023	
9.	Рефакторинг архітектури клієнта. Додавання ключових точок для сервісів Google Drive та Google Docs.	21.03.2023	
10.	Додавання сторінки «Мій кабінет». Підключення Google API .	10.04.2023	
11.	Додавання роутингу в «Мій кабінет». Рефакторинг Action creators у Redux рівні.	27.04.2023	
12.	Дослідження зчитування Google документу та його шаблонів.	30.04.2023	
13.	Додавання сторінки угод користувача та сторінки профілю.	03.05.2023	
14.	Покращення сторінки профілю та підписання угоди.	07.05.2023	
15.	Додавання можливості підписання угоди, редагування GDocs.	08.05.2023	
16.	Створення сторінки угоди.	10.05.2023	
17.	Додавання сторінок «Про нас», «Контакти», «Часті запитання»	13.05.2023	
18.	Редизайн сторінки угоди. Рефакторинг Redux рівня.	15.05.2023	

Студент _____
(підпис)

Петришин М.Ю.
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Васильків Т. І.
(прізвище та ініціали)

СУТЬ ДИПЛОМНОЇ РОБОТИ

Робота складається із вступу, двох розділів, висновків та джерел. Обсяг дипломної роботи: 37 сторінок та 47 рисунків. Список використаних джерел складається з 14 найменувань.

Мета даної роботи – розробка та дослідження процесу автоматизації підписання угод завдяки бібліотеці React та сервісам Google. Вирішення подібної проблеми є доцільним, оскільки багато бізнес-процесів переходять у онлайн через зв'язку із введенням воєнного стану у нашій країні. Ефективні інструменти для підтримки та управління цими процесами відіграють велику роль у підвищенні продуктивності даної сфери. Моя робота надає користувачам можливість зручно та швидко керувати підписанням угод. Завдяки цьому інструменту, час та витрати, пов'язані з обробкою паперових документів та фізичним підписанням, значно зменшуються. Замість традиційного підписання документів, мій додаток дозволяє створювати, розсилати та підписувати угоди електронно, з будь-якого місця та в будь-який час, просто за допомогою Інтернету та пристрою, що підтримує доступ до мережі.

У результаті - витрати на друкування, поштові відправлення та зберігання фізичних копій документів значно зменшуються. Це не лише економить кошти, але й дозволяє використовувати час більш ефективно.

Також не варто забувати про універсальність, оскільки така платформа може бути використана в будь-якій сфері бізнесу, де виникає потреба у підписанні угод. Вона може забезпечити ефективне керування підписанням документів незалежно від того, чи це навчальний заклад, фінансова установа, логістична компанія або будь-яка інша організація.

ЗМІСТ

ЗМІСТ	4
ВСТУП.....	5
ПОСТАНОВКА ЗАДАЧІ.....	7
РОЗДІЛ 1. АРХІТЕКТУРА СИСТЕМИ	8
1.1 База даних	8
1.2 Серверна частина	9
1.3 Клієнтська частина	11
РОЗДІЛ 2. ДЕМОНСТРАЦІЯ РОЗРОБЛЕНОГО ДОДАТКУ	13
2.1 Головна сторінка сайту	13
2.2 Сторінка «Про нас»	18
2.3 Сторінка «Контакти»	19
2.4 Сторінка «Часті запитання»	20
2.5 Сторінка «Мій кабінет»	21
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	36

ВСТУП

Криза, що викликана пандемією COVID-19 сильно вплинула на сучасні економічні процеси і підірвала економіки країн світу. Кризові явища останніх років у світовій економіці проявлялися: зниженні більшості індексів фондових бірж світу; зменшенні вартості нафти та продукції металургії; скорочення обсягів промислового виробництва; значному зменшенні доходів компаній у сферах авіаперевезень, морського транспорту, торгівлі та розваг, туризму; скороченні доходів домогосподарств; збільшенні прибутків компаній у сферах онлайн послуг (ігор, освіти тощо); підвищенні попиту інвесторів на активи з низьким ризиком. Експертами передбачалося суттєве скорочення темпів глобального економічного зростання до кінця 2025 року, а глобальні економічні втрати ймовірно мали б досягнути 5 трлн доларів. [1] Наша держава не є винятком. Протягом 2020 р. економіка України скоротилась на 7,2 %, а з 2021 р. до 2025 р. очікувалось щорічне зростання на рівні 3-4 %. Враховуючи цей факт, додамо ще військові дії, які розв'язала росія проти України у 2022 році. Все це сильно ослабило і економічний стан нашої держави. І лише макроекономічна допомога союзників дозволяє державі виконувати свої зобов'язання перед кредиторами, населенням і вижити у такі нелегкі часи.

Виникає необхідність виваженої політики з боку уряду та органів місцевого самоврядування; прийняття ними рішень, що не лише враховують ефект карантину і війни, а й спрямовані на формування інноваційної, конкурентоспроможної та соціально орієнтованої економіки в плані підписання угод та їх автоматизації.

Подальші дослідження процесів післявоєнного відновлення повинні бути спрямовані на вивчення соціальних та економічних наслідків війни, шляхів мінімізації її негативних ефектів, пошуку нових механізмів організації суспільних процесів та стабілізації економіки в умовах війни. І оформлення угод способом автоматизації є невеликим практичним кроком для цього. Усі вищеописані проблеми породжують величезну потребу у інструментах, які можуть допомогти підтримувати ці процеси.

Проект **AgrConclude** присвячений автоматизації підписання угод, що є актуальною та важливою темою в епоху цифрових технологій. Даний додаток дозволяє користувачам з легкістю та ефективністю керувати підписанням угод, скорочуючи час та витрати на їх обробку. Крім того, даний інструмент може бути використаний в будь-якій сфері бізнесу, що робить його незамінним для розвитку сучасного підприємництва.

ПОСТАНОВКА ЗАДАЧІ

Розробити SaaS(Software as a service) платформу для автоматизації процесу укладання угод.

Під час виконання курсової роботи необхідно виконати такі завдання:

- розробити додаток для простого та швидкого підписання угод
- зробити зручний та інтуїтивний інтерфейс
- надати можливість перегляду власних угод
- надати можливість створювати, відхиляти угоди, переглядати користувачів та їх угоди.
- оптимізувати роботу програми, не підгружаючи зайвий контент
- мова інтерфейсу – англійська
- зручність експлуатації

РОЗДІЛ 1. АРХІТЕКТУРА СИСТЕМИ

1.1 База даних

Для розробки додатку було використано таку реляційну базу даних (дальше БД) як **PostgreSQL**. Це популярна кросплатформена технологія, яка може бути запущена на будь-якій машині, на відміну від SQL Server.

За весь зв'язок з базою даних відповідає Entity Framework Core у Infrastructure рівні серверу (див. 1.2).

Приклад класів додатку, навколо яких будуватиметься процес зображено на рисунку 1.1.

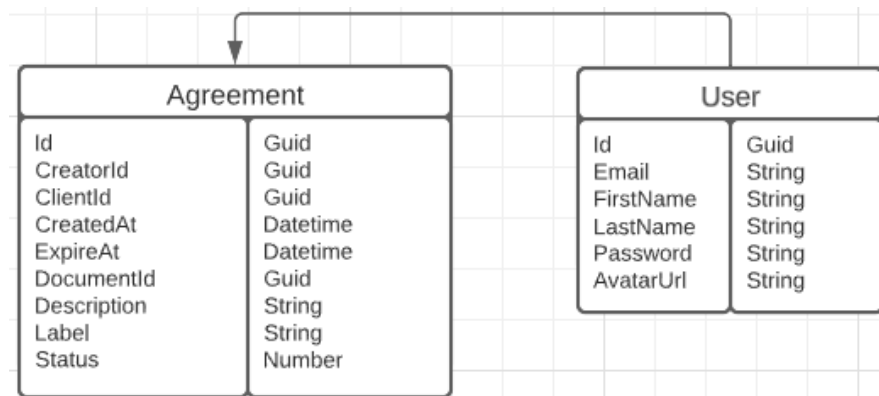


Рисунок 1.1 – Діаграма класів

- **User** – для збереження користувачів у базі даних.
 1. Id – унікальний ідентифікатор користувача.
 2. AvatarUrl – посилання на аватар користувача у Google
 3. Password, Email, Name – детальніша інформація про користувача.
- **Agreement** – для збереження угод у базі даних
 1. Id – унікальний ідентифікатор угоди.
 2. CreatorId – унікальний ідентифікатор творця угоди.
 3. ClientId – унікальний ідентифікатор отримувача.
 4. CreatedAt, ExpireAt – дата створення угоди та її тривалість
 5. DocumentId – унікальний ідентифікатор документа, над яким укладається угода.
 6. Description, Label, Status – детальніша інформація про угоду.

1.2 Серверна частина

ASP .NET Core – це фреймворк, за допомогою якого ми можемо створити хост для нашої програми. Він може обробляти HTTP запити, які приходять з веб-серверів (Kestrel або IIS Express).

Окрім ASP .NET Core існує ще інший фреймворк для .NET'у – ASP .NET. Він є більш старим та не крос-платформеним – його можна запускати лише на операційній системі Windows, та працювати він буде виключно з IIS Express.

О 2021 році було розроблено фреймворк ASP .NET Core, на новій версії .NET'у. Він уже може працювати з Kestrel та іншими веб-серверами, що є дуже гнучким та дає можливість контейнеризувати наші додатки в Linux контейнери та запускати наші сервери на інших операційних системах.

Щоб звертатись до бази даних Postgres (див 1.1) використано Entity Framework Core, який є базовим для PostgreSQL. Фреймворк розширює логіку, яка вміє працювати з цією БД. Цим розширенням слугують міграції – вони створені для того, аби дані з нашої бази завжди були послідовними, тобто була змогу переключатись між нашими версіями сутностей.

Архітектура серверної частини побудована наступним чином:

- Application – рівень, який відповідає за всю бізнес-логіку, що буде відбуватись у нашому проекті, та безпосереднє звернення до інтерфейсів, що працюють з БД.
- Infrastructure – рівень, який відповідає за безпосереднє звернення до сутностей бази даних. Тут також зберігаються міграції – версії сутностей. Цей рівень є абстрагованим та «знає» лише про БД.
- API – рівень, що дає клієнтам, які взаємодіють з сервером, доступ до публічних кінцевих точок, за допомогою яких ми можемо звертатись до нашого сервера. Також тут відбувається валідація моделей, які є в Application рівні.
- Domain – рівень, де зберігається все, з чим наш сервер буде взаємодіяти – інтерфейси, які реалізовує Infrastructure рівень і які використовує Application рівень

В Infrastructure рівні для доступу до наших сутностей і ресурсу я використовую Entity Framework Core, оскільки він допомагає генерувати таблиці і записи БД, базуючись на сутностях.

Тип такої архітектури називається Onion Architecture. Вона зображена на рисунку 1.2.

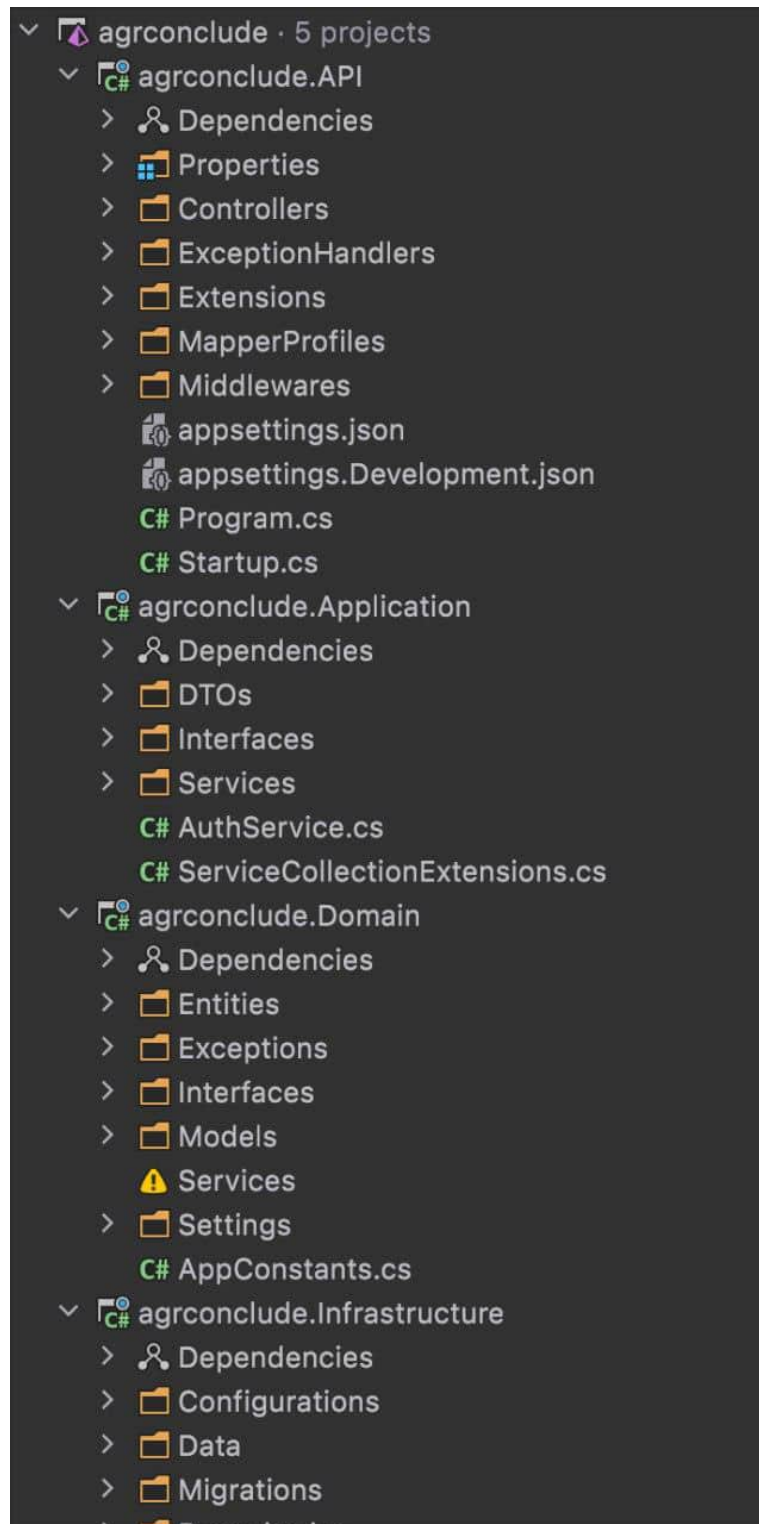


Рисунок 1.2 – Архітектура серверної частини додатку

1.3 Клієнтська частина

Для побудови та розробки клієнтської частини додатку було використано:

- React — це JavaScript бібліотека, для побудови односторінкових застосунків (Single Page Application). Вона створена для вирішення проблеми оновлення сторінок шляхом організації їх часткового монтування.
- Redux — відкрита JavaScript бібліотека, призначена для управління станом програм JS.
- Redux Saga — це middleware для Redux, ціль якої - зробити асинхронні операції ефективними до виконання, легко виконуваними і такими, які можна тестувати
- Formik — бібліотека, призначена для зручної побудови форм.
- Yup — бібліотека, призначена для швидкої валідації введених даних.
- React Signature — бібліотека, призначена для створення підпису
- Material UI — бібліотека для використання готових компонент з вбудованими CSS стилями, HTML розміткою та логікою, написаною на JS.
- React-google-login — бібліотека для авторизації за допомогою Google
- React-router-dom — бібліотека для навігації між Реакт-компонентами, яка організовується без перезавантаження веб-сторінки.
- Axios — це JavaScript бібліотека для виконання HTTP запитів в Node.js з автоматичним перетворенням JSON даних.
- Google Drive API — це набір інструментів, який дозволяє створювати, зчитувати, оновлювати та взаємодіяти з усіма файлами, що містяться у Google Drive користувача за допомогою HTTP запитів.
- Google Docs API — це набір інструментів, який дозволяє модифікувати та створювати документи у Google Docs за допомогою HTTP запитів.

Архітектура клієнтської частини побудована наступним чином:

- components – папка з елементами інтерфейсу
- layouts – папка для усіх сторінок сайту
- resources – папка для зображень
- services – папка, де зберігаються константи, шляхи на сторінки та корисні функції
- store – папка Redux для менеджменту віддаленого стану додатку
- styles – стилі

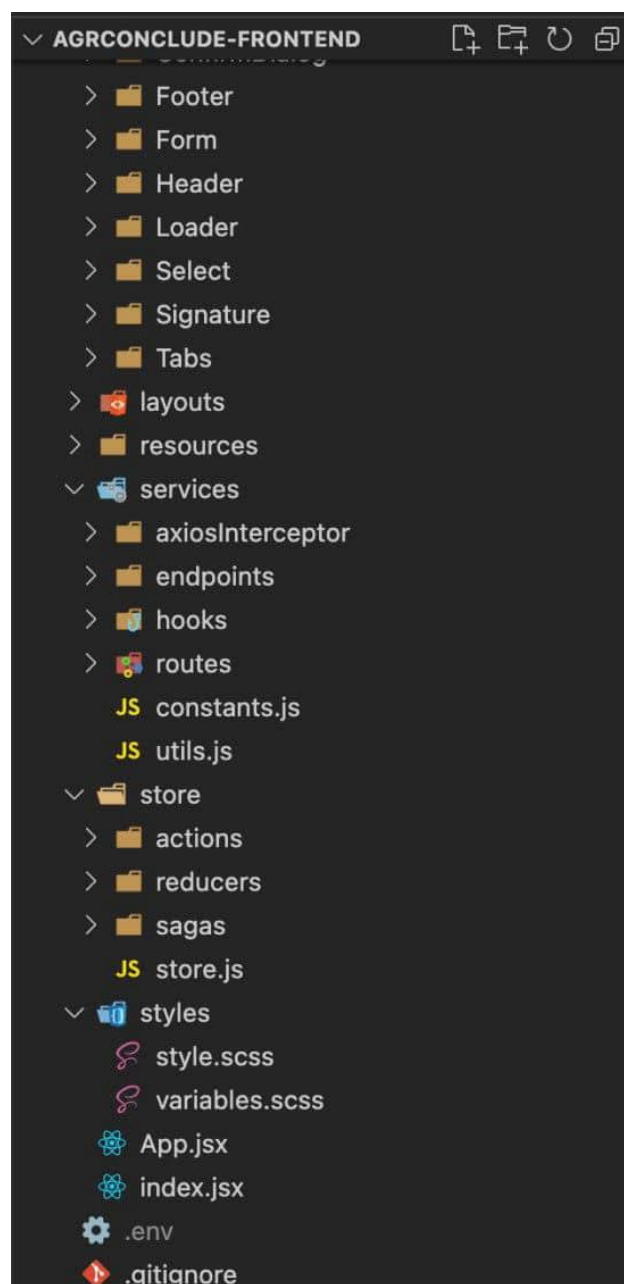


Рисунок 1.3 – Архітектура клієнтської частини додатку

РОЗДІЛ 2. ДЕМОНСТРАЦІЯ РОЗРОБЛЕНОГО ДОДАТКУ

2.1 Головна сторінка сайту

Увійшовши на сайт, нас зустрічає його головна сторінка (рис. 2.1). На ній ми зможемо ознайомитись з усією потрібною нам інформацією, безпосередньо зареєструватись в додатку та перейти на інші сторінки, використавши навігацію.

Головна сторінка включає в себе три секції – «Вхід», «Про нас», «Контакти»



Рисунок 2.1 – Головна сторінка. Секція входу

Мобільний вигляд секції входу зображено на рисунку 2.2.

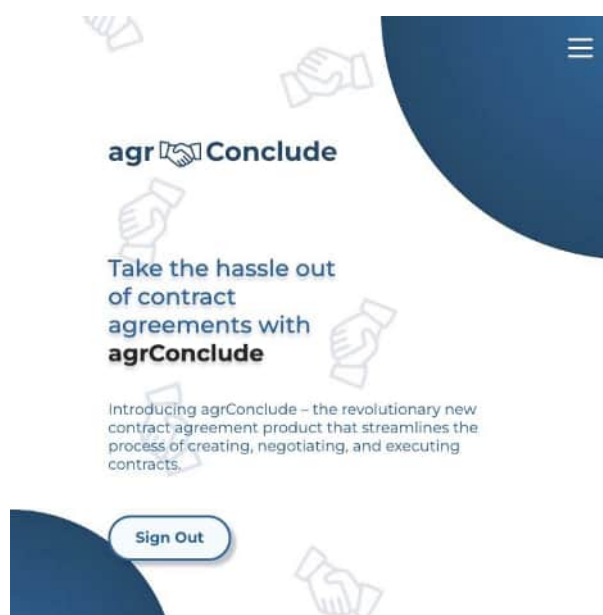


Рисунок 2.2 – Головна сторінка на мобільних пристроях. Секція входу

Секція «Про нас» (рис. 2.3) включає в себе коротку інформацію про проект та містить посилання, перейшовши по якому ми зможемо отримати більш детальну інформацію на однойменній сторінці (див. 2.2).

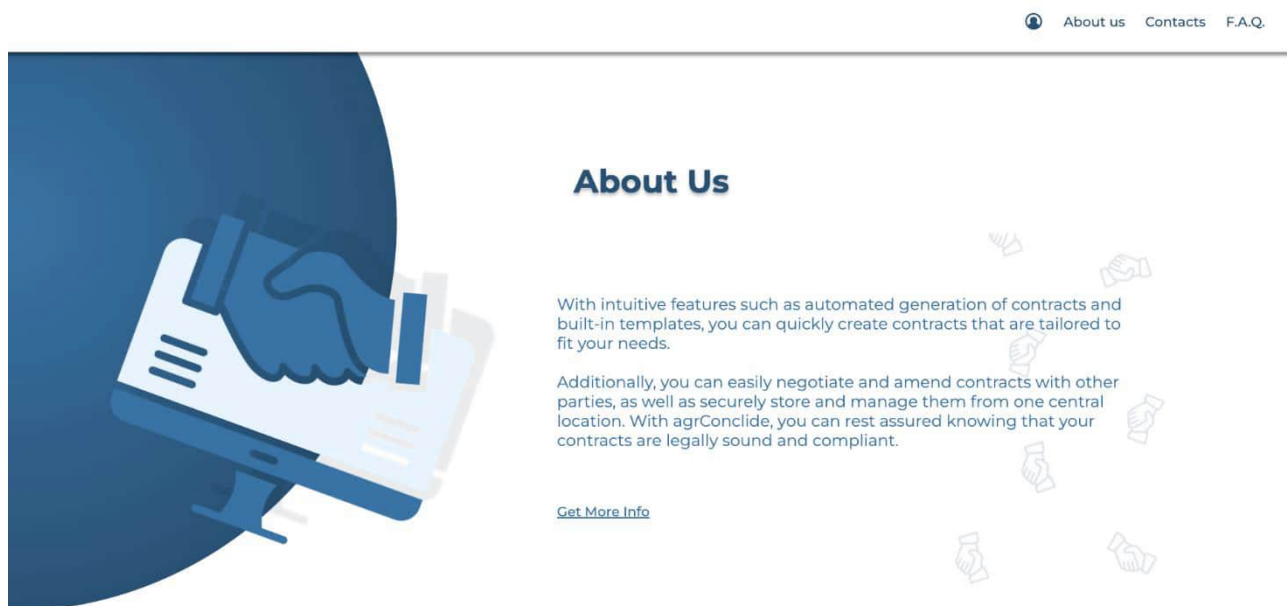


Рисунок 2.3 – Головна сторінка. Секція «Про нас»

Мобільний вигляд секції «Про нас» зображено на рисунку 2.4.

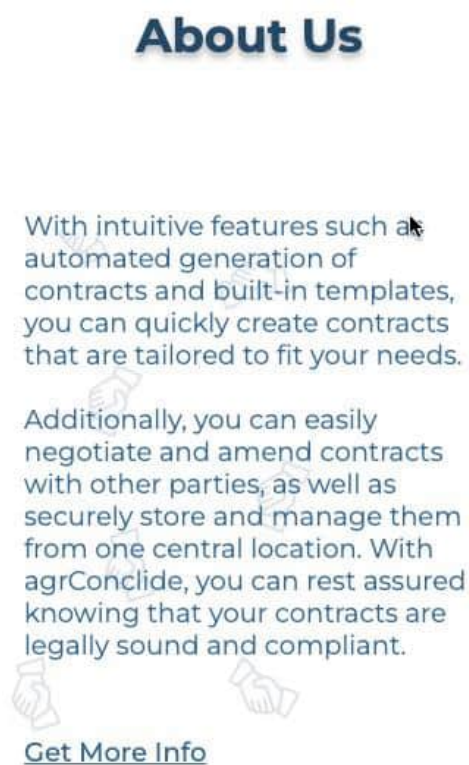


Рисунок 2.4 – Головна сторінка на мобільних пристроях. Секція «Про нас»

Секція «Контакти» (рис. 2.5) включає в себе посилання на соціальні мережі, інформативне повідомлення стосовно безпеки та містить посилання, перейшовши по якому ми зможемо отримати більш детальну інформацію на однойменній сторінці (див. 2.3).

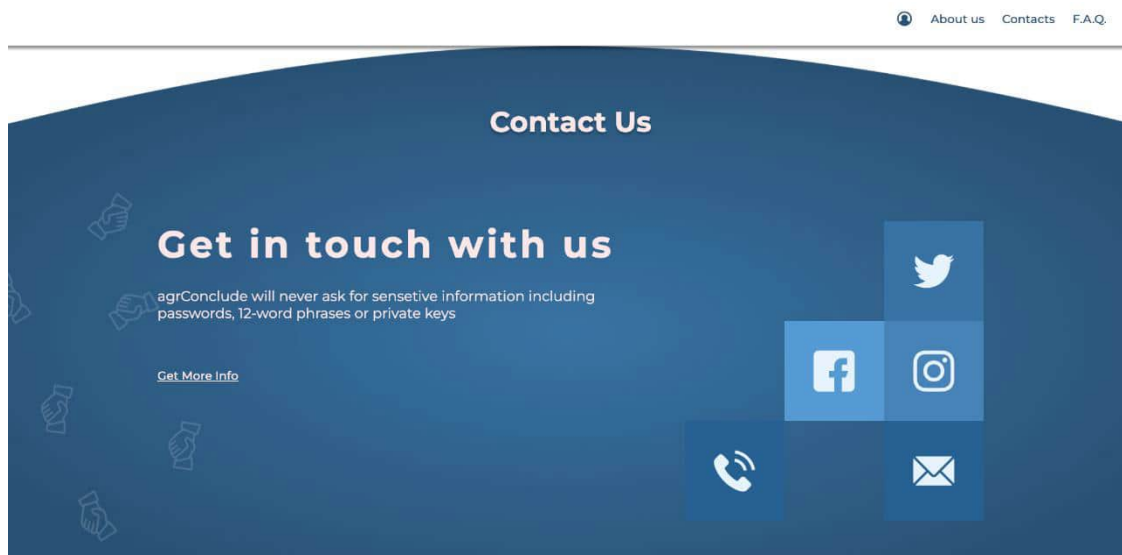


Рисунок 2.5 – Головна сторінка. Секція «Контакти»

Мобільний вигляд секції «Контакти» зображено на рисунку 2.6.

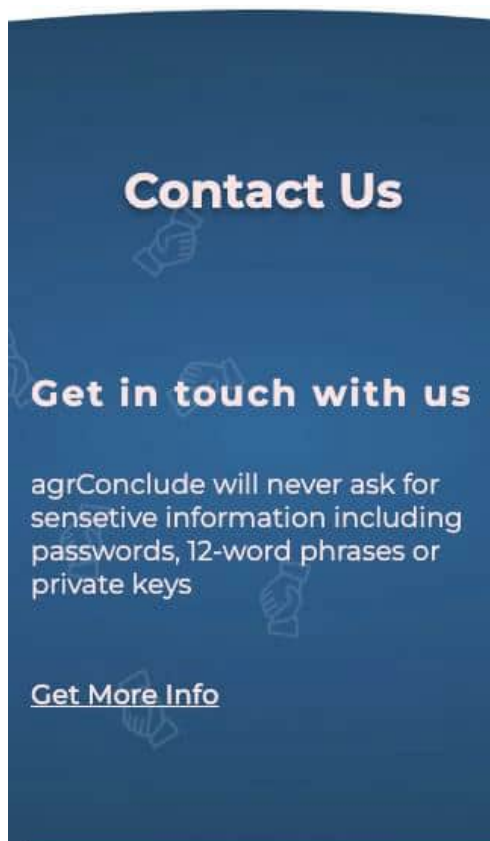


Рисунок 2.6 – Головна сторінка на мобільних пристроях.. Секція «Контакти»

Після успішної авторизації користувача деякі частини сайту, зокрема меню сайту або кнопка входу, набувають іншого вигляду.

У меню сайту з'являється кнопка для входу в особистий кабінет користувача, де він зможе створювати угоди та надсилати їх іншим (рис 2.7).

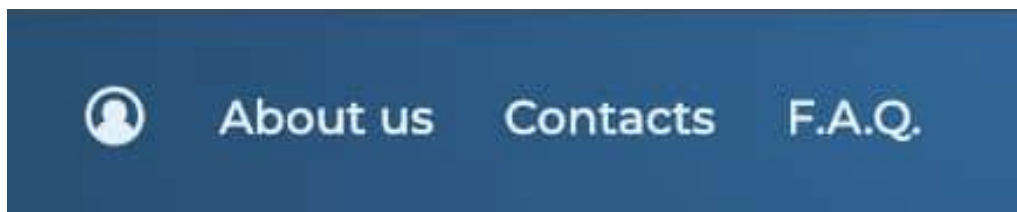


Рисунок 2.7 – Меню на головній сторінці при авторизованому користувачеві

Кнопка входу міняється на кнопку виходу (рис 2.8). Натиснувши на неї – система вийде з аккаунту користувача.

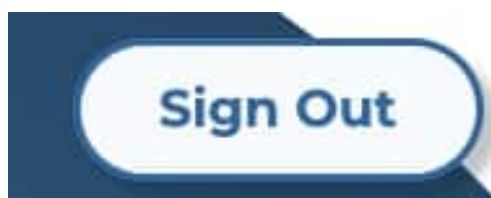


Рисунок 2.8 – Кнопка на головній сторінці при авторизованому користувачеві

На головній сторінці навігаційне меню має тло синього кольору (рис 2.9), а на інших сторінках – білого.



Рисунок 2.9 – Вигляд меню на головній сторінці

Також при скролі головної сторінки меню ставатиме фіксованим, а його тло набуде білого кольору (рис 2.10) з повільної анімацією.



Рисунок 2.10 – Вигляд меню на інших сторінках

На мобільних пристроях відображення меню у шапці профілю стає неможливим, тому натомість я вирішив додати «гамбургер» меню (рис 2.11).

Це меню являє собою невелику кнопку, при натисканні на яку безпосередньо відкривається меню. Воно має аналогічну структуру та логіку, лише відрізняється вигляд.

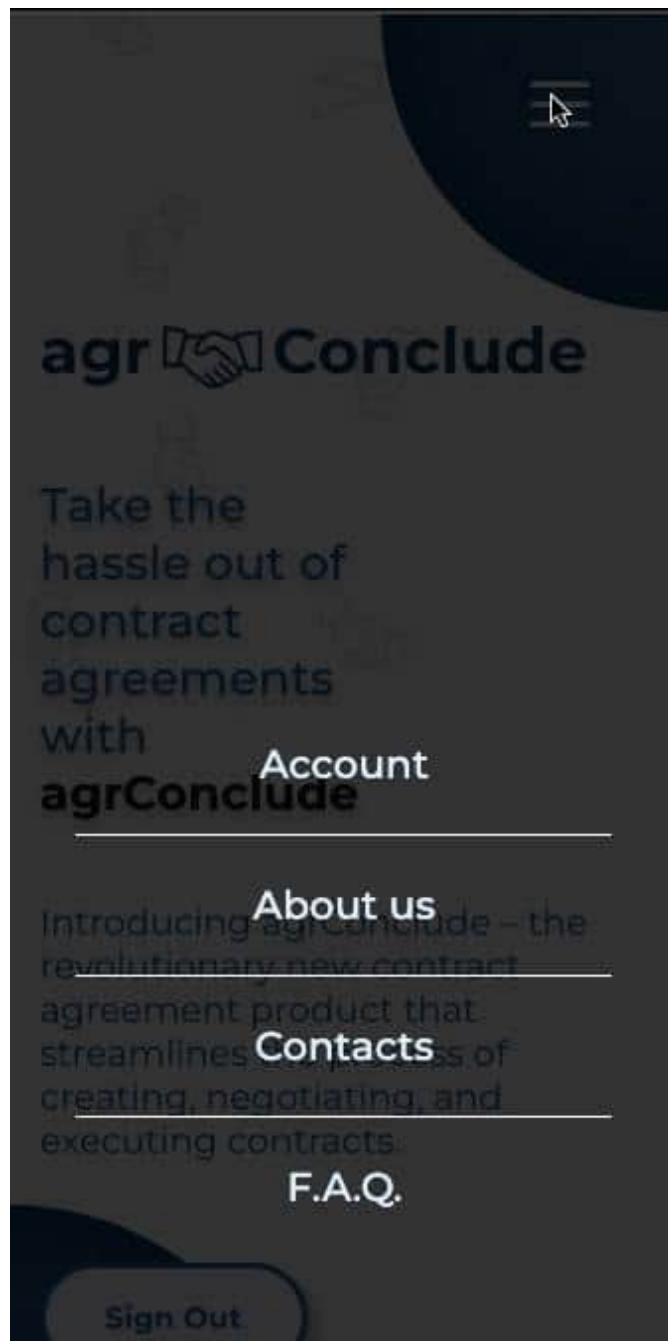


Рисунок 2.11 – Вигляд меню на мобільних пристроях

2.2 Сторінка «Про нас»

Використавши навігацію з головної сторінки (див 2.1) або з меню (рис. 2.7) ми маємо змогу перейти на сторінку «Про нас» (рис. 2.12).

Тут можна дізнатись різну інформацію про додаток:

- історію створення
- успішні результати
- з ким налагоджені процеси співпраці
- основний відсоток користувачів



Рисунок 2.12 – Сторінка «Про нас»

Мобільний вигляд сторінки «Про нас» зображено на рисунку 2.13.

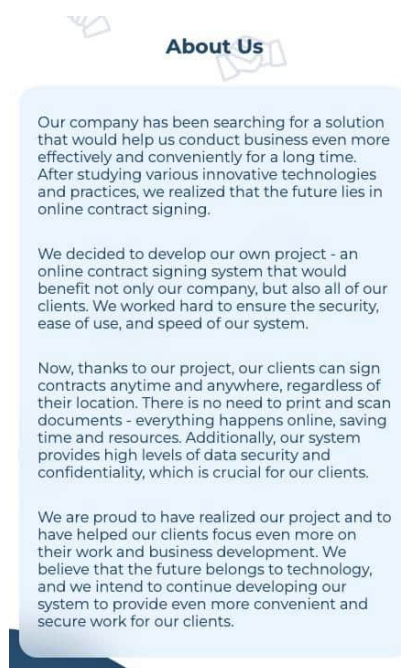


Рисунок 2.13 – Сторінка «Про нас» на мобільних пристроях.

2.3 Сторінка «Контакти»

Використавши навігацію з головної сторінки або з меню сайту ми маємо змогу перейти на сторінку «Контакти» (рис. 2.14).

Тут можна ознайомитись з усією контактною інформацією, які є у підтримки додатку:

- соціальні мережі
- електронні адреси
- номери телефонів

Також працює інтерактивна мапа, яка використовує сервіси Google. За допомогою неї можна ознайомитись з розташуванням офісу компанії.



Рисунок 2.14 – Сторінка «Контакти»

Мобільний вигляд сторінки «Контакти» зображено на рисунку 2.15.

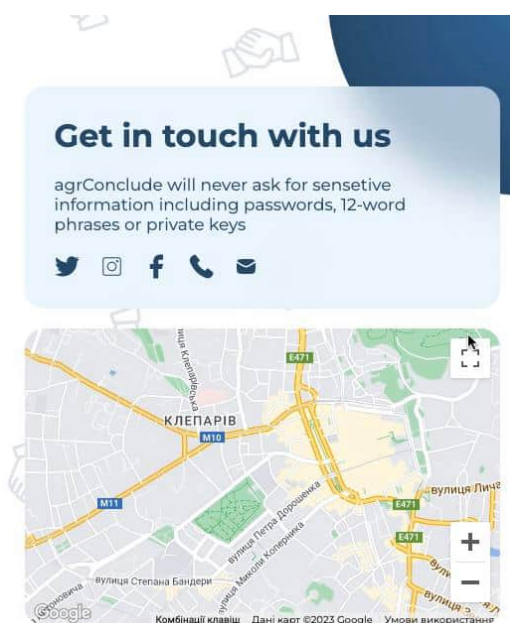


Рисунок 2.15 – Сторінка «Контакти» на мобільних пристроях.

2.4 Сторінка «Часті запитання»

На цій сторінці є відповіді на всі часті запитання (рис. 2.16). Користувачі можуть надсилати свої питання на електронну пошту, вказану на сторінці «Контакти» і усі питання, що часто будуть повторятись, попадатимуть на цю сторінку з детальною відповіддю.

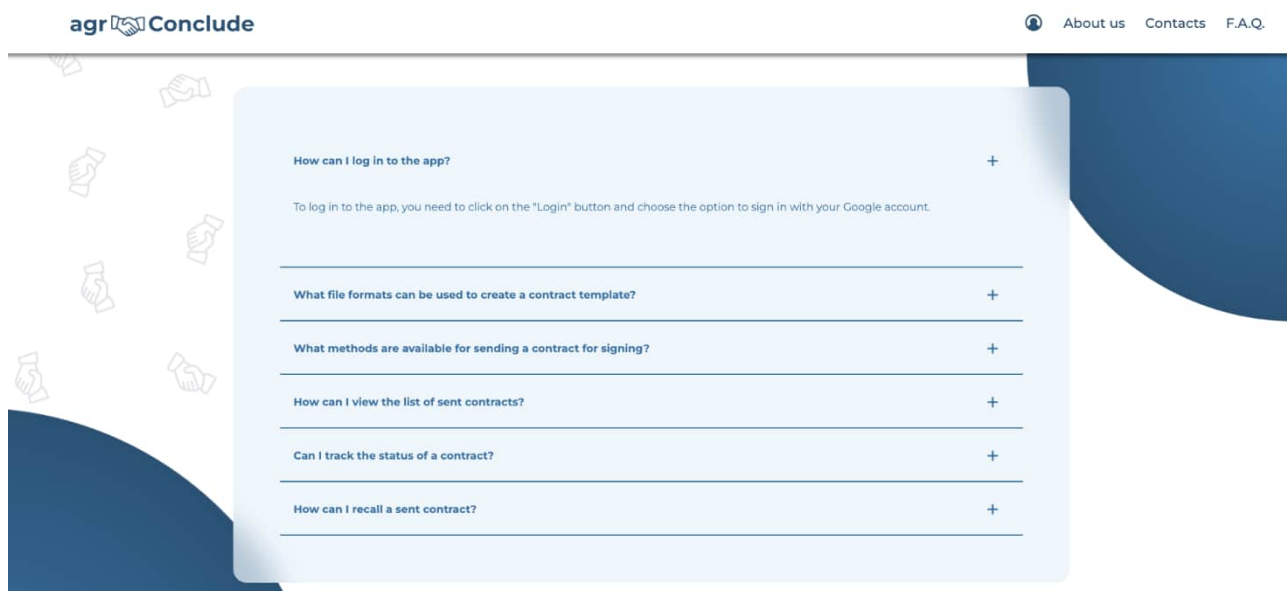


Рисунок 2.16 – Сторінка «Часті запитання».

Мобільний вигляд сторінки «Часті запитання» зображено на рисунку 2.17.

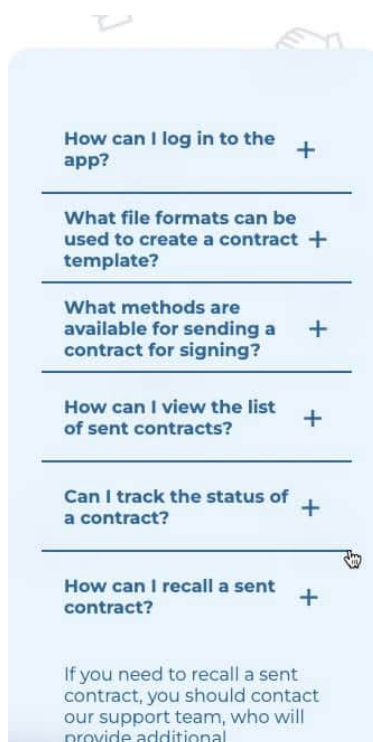


Рисунок 2.17 – Сторінка «Часті запитання» на мобільних пристроях.

2.5 Сторінка «Мій кабінет»

Після успішної авторизації на сайті користувач має змогу перейти до свого кабінету.

У своєму кабінеті є можливість:

- перегляд інформації користувача
- створення угод.
- перегляду створених угод
- перегляду угод, надісланих користувачу на підписання
- перегляду шаблонів для угод, які існують на Google Drive користувача

Щоб здійснювати навігацію по кабінету користувача існує панель з усіма наявними сторінками (рис. 2.18). Вона теж розроблена у стилі «гамбургер» меню, тому на мобільних пристроях її вигляд не змінюється.

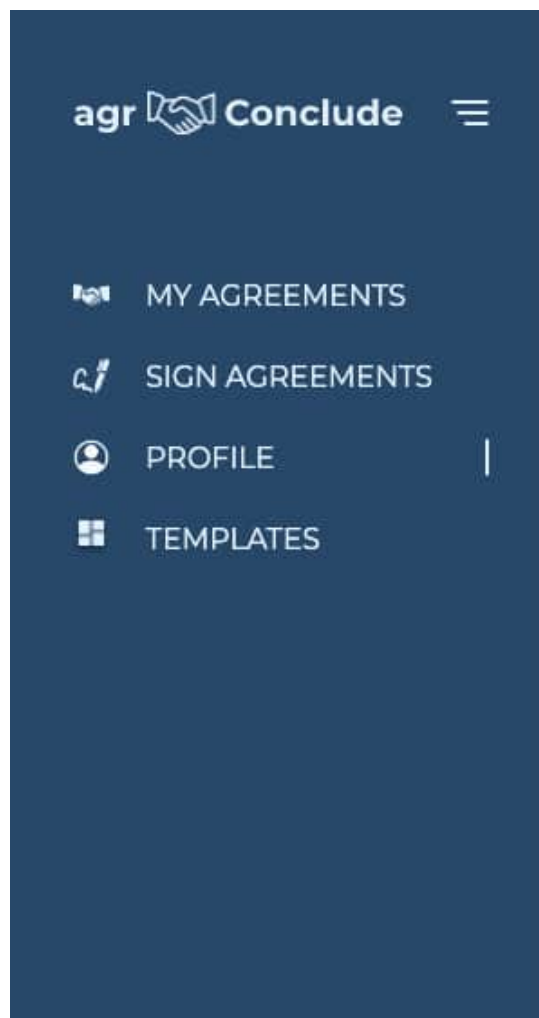


Рисунок 2.18 – Бокова панель навігації по кабінету.

На профілі користувача (рис. 2.19) є можливість додавання підпису та редагування такої інформації як день народження, стать, номер телефону та посилання на соціальні мережі.

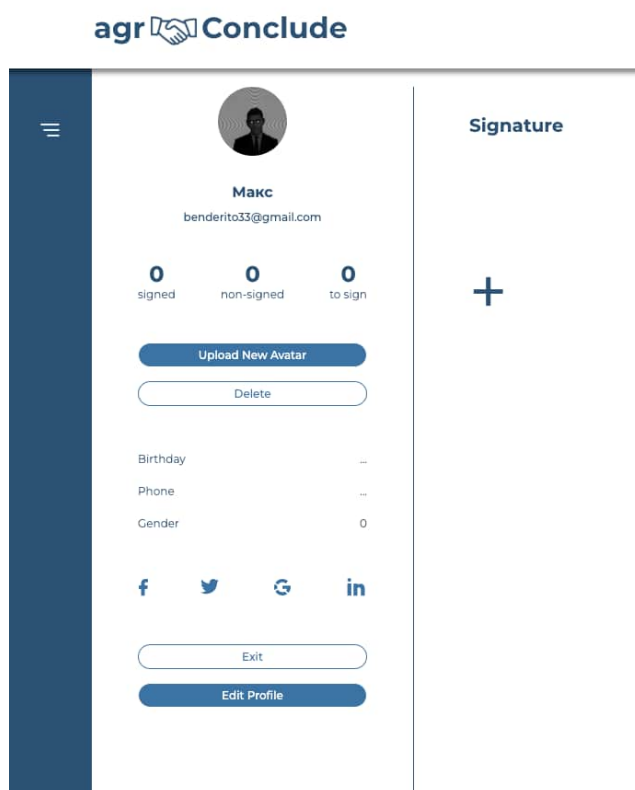


Рисунок 2.19 – Профіль користувача.

Діалогове вікно для створення / заміни підпису



Рисунок 2.20 – Створення підпису.

Для того, щоб створити з кимось угоду, користувачу спершу потрібно мати шаблон тієї угоди, яку би він хотів підписати з кимось. Для створення такого роду шаблонів у AgrConclude потрібно лише Google Docs.

Шаблон угоди – звичайний Google Docs документ (рис. 2.21), який модифіковано за алгоритмом. Він полягає у додаванні полів, які потрібно заповнити отримувачу.

Акт виконання робіт					
					{{date}}
№	ПІБ застрахованої особи	Дата надання послуг	Назва наданих послуг	кількість	сума
1					
2					
3					
4					
5					

Названі послуги на суму: {{sum}}

Виконавець: {{fullPIB}}

Підпис: {{sign}}

Рисунок 2.21 – Приклад шаблону у Google Docs.

Алгоритм створення шаблону:

- Створити новий Google Document на аккаунті, що прив'язаний до AgrConclude
- Дати йому назву, яка починається з AGC_TEMPLATE
- Створити угоду, яку вам потрібно
- Місця, де вимагається заповнення інформації іншою людиною, оформляти наступним чином:
 - {{test}} - місце, де потрібно звичайний текст. (Замість test може бути будь-що. Важливо щоб це було англійською мовою та без пробілів!!!)
 - {{date}} - місце, де потрібно вказувати дату. (Можна писати date/date2/date3/dateBirth і так далі)
 - {{sign}} - місце, де потрібно ставити підпис

У назві документу обов'язково потрібно вказувати «AGC_TEMPLATE» (рис.2.22), оскільки лише документи такого типу буде показано в додатку (див.рис.2.23).



Рисунок 2.22 – Приклад назви шаблону.

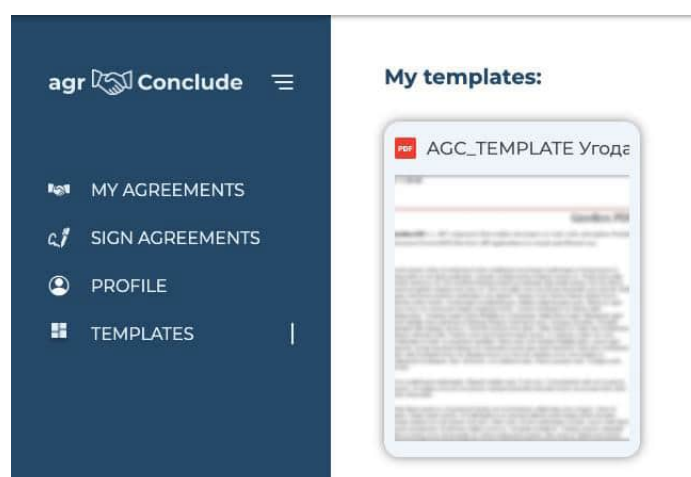


Рисунок 2.23 – Шаблони користувача.

Для того, щоб переглянути створені власноруч угоди, є спеціальна панель (рис. 2.24)

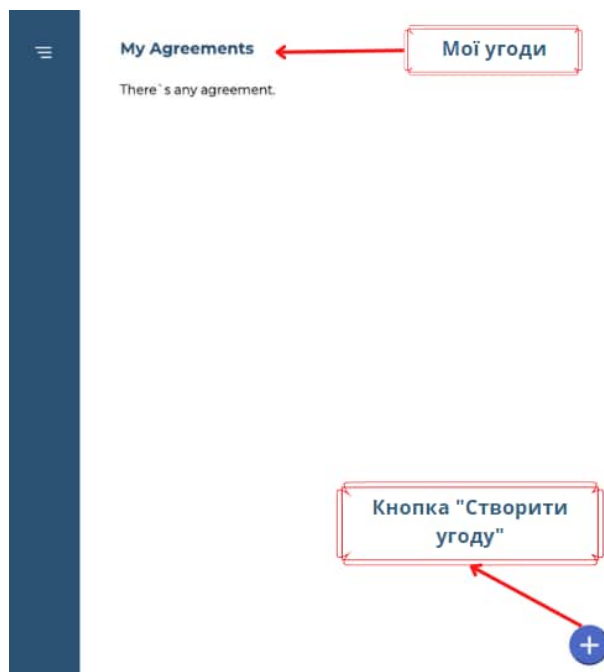


Рисунок 2.24 – Панель для перегляду списку угод.

Для того, щоб створити угоду, є спеціальна кнопка у вигляді «+», після натискання на яку, у нас відкривається модальне вікно з формою.

У цьому вікні потрібно вказати один з створених шаблонів, отримувача та додаткову інформацію (рис. 2.25).

Рисунок 2.25 – Модальне вікно для створення угоди.

Після створення, список угод оновлюється і містить нову угоду. В залежності від сторони угоди міняється й вигляд карточки (рис. 2.26 та 2.27).

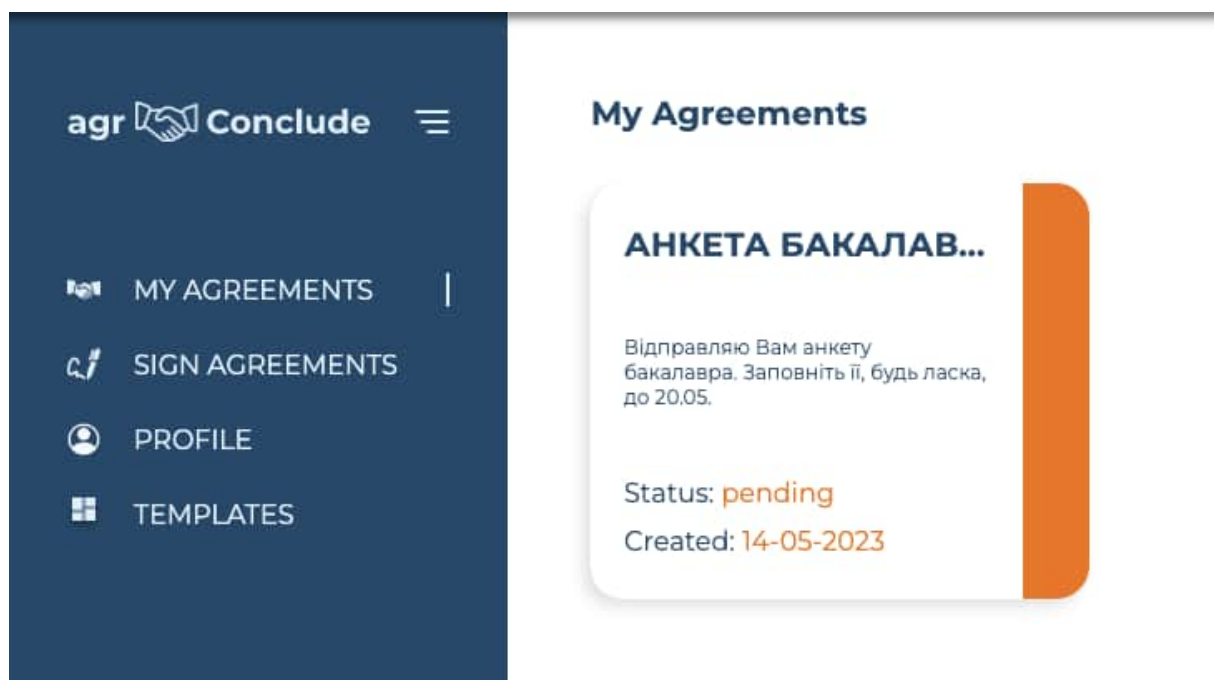


Рисунок 2.26 – Список угод з сторони відправника.

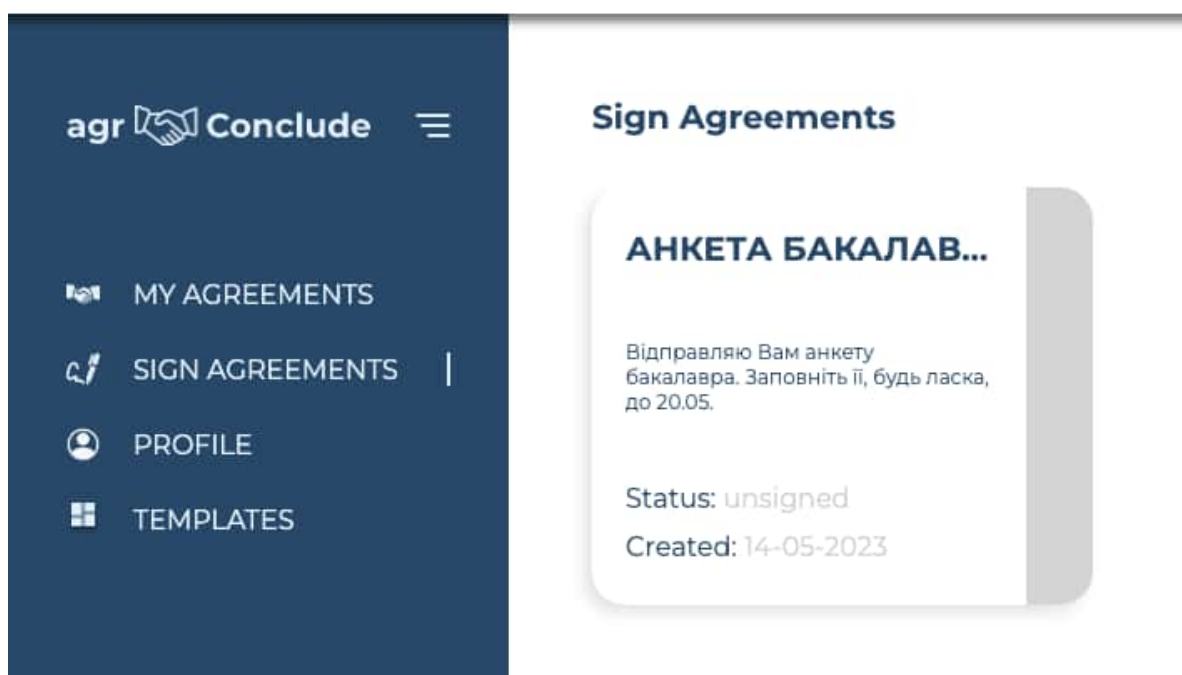


Рисунок 2.27 – Список угод з сторони отримувача.

Колір угоди залежить від її статусу. Їх є три – «**підписано**», «**відхилено**» та «**в обробці**».

Також останні угоди, з якими взаємодіяв користувач, появлятимуться у його профілі в секції «Останні надіслані / отримані угоди» (рис.2.28 та рис. 2.29).

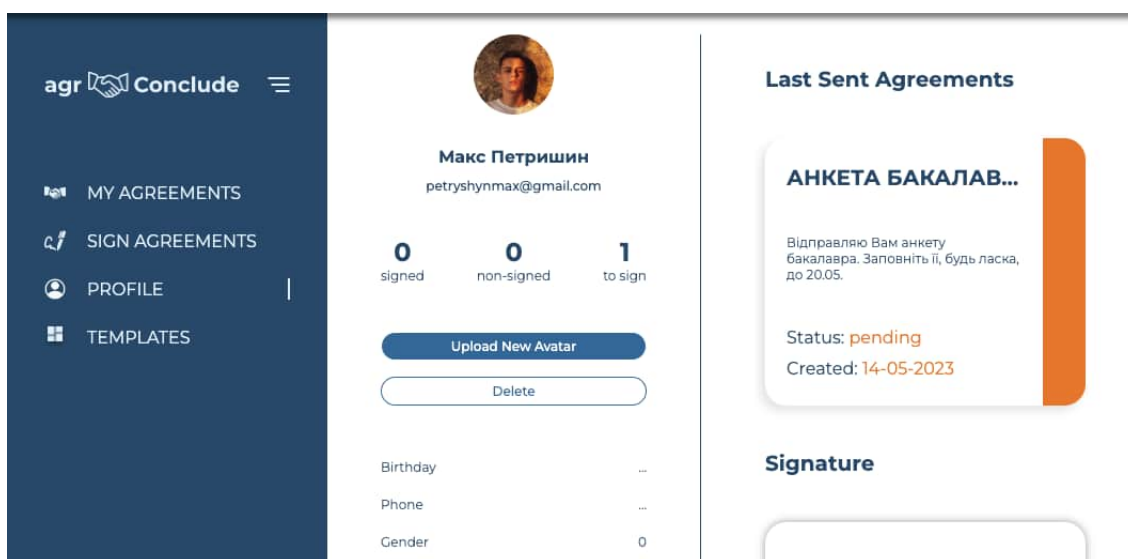


Рисунок 2.28 – Заповнений профіль з сторони відправника.

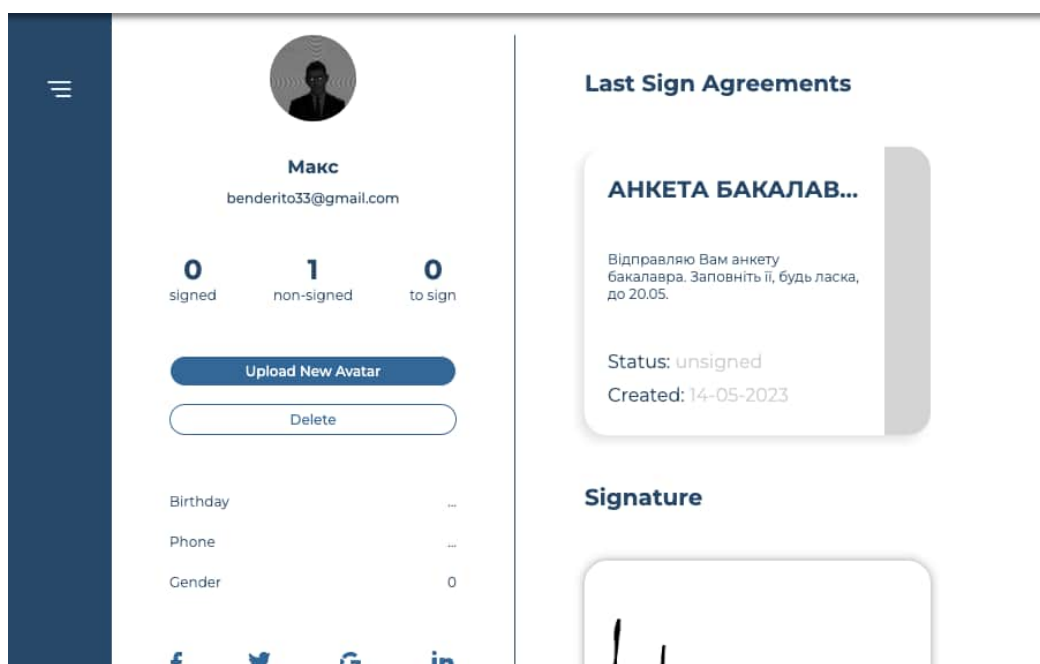


Рисунок 2.29 – Заповнений профіль з сторони отримувача.

Колір статусу карточки відрізняється. Це зроблено спеціально для того, щоб правильно розпізнавати угоди. Як уже згадувалось, всього є три можливі статуси. Однак статус «concluded» на стороні отримувача виглядатиме як «signed», а статус «pending» - «unsigned».

Натиснувши на карточку будь-якої угоди, у користувача є можливість перейти на сторінку угоди, де є більш детальна інформація та навігаційна панель для взаємодії з угодою.

Тут можна ознайомитись детальніше з описом угоди, датою її створення, термін її дійсності, відправника та отримувача.

Якщо користувач є **відправником** угоди, він може її:

- Скасувати
- Відкрити шаблон для ознайомлення

Якщо користувач є **отримувачем** угоди, він може її:

- Скасувати
- Відкрити шаблон для ознайомлення
- Підписати

Якщо користувач **не є фігурантом** угоди, то він може лише ознайомитись із інформацією про неї. Навігаційна панель для нього не доступна.

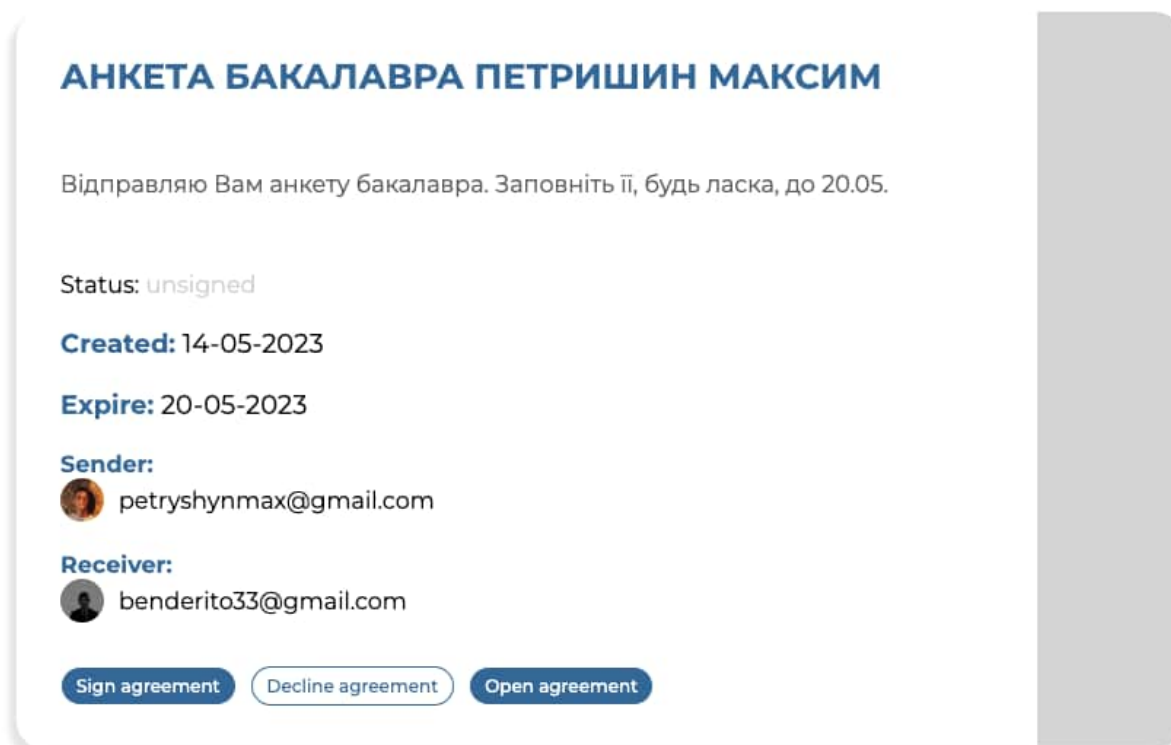


Рисунок 2.30 – Сторінка угоди.

При скасуванні угоди будь-які дії над нею стають недоступними (див. рис. 2.32 та 2.33). Для скасування існує спеціальне модальне вікно (рис. 2.33).

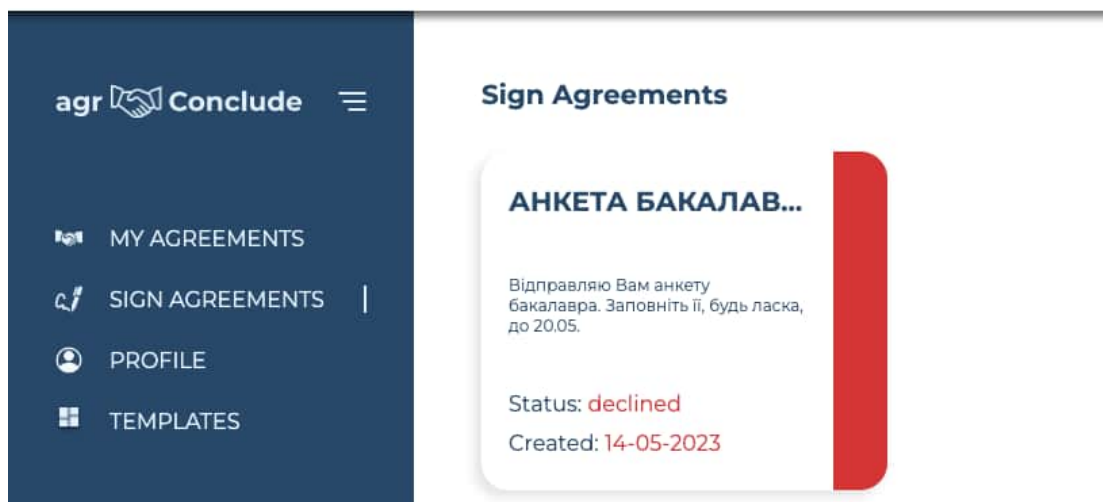


Рисунок 2.31 – Вигляд у списку угод.

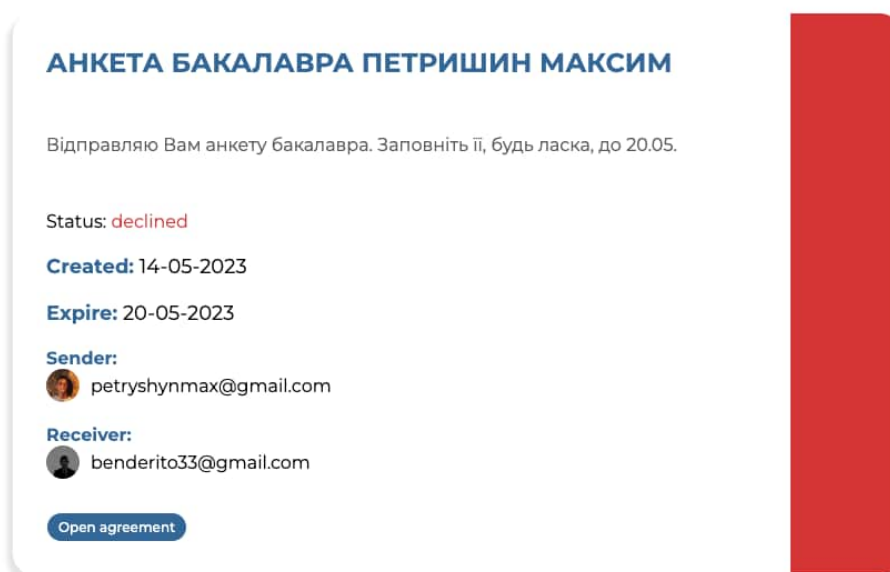


Рисунок 2.32 – Вигляд на сторінці угоди.

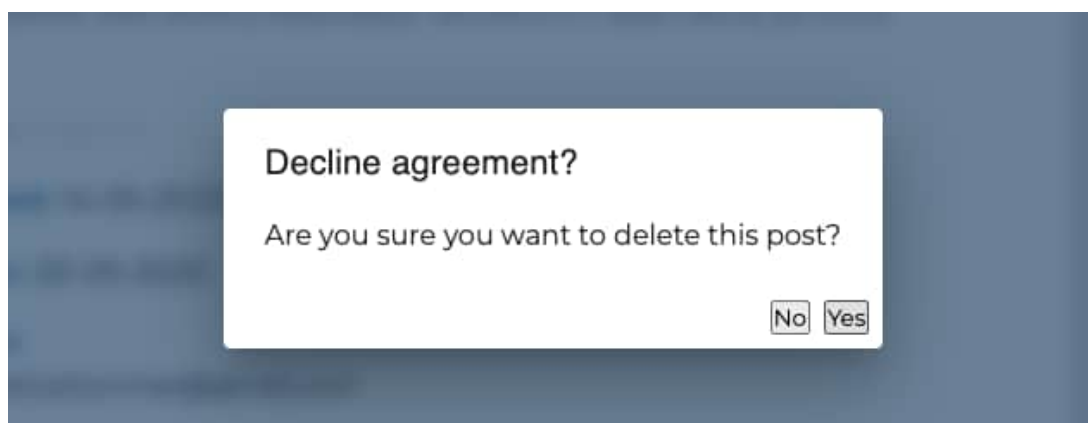


Рисунок 2.33 – Модальне вікно для скасування угоди.

При підписанні угоди додаток:

- сканує шаблон, по якому вона була створена
- зберігає усі шаблонні поля, які потрібні для підписання
- генерує форму

Форма здебільшого складається з текстових полів (рис.2.34). Однак для вводу дати є можливість заповнення з календаря (рис. 2.35), а для підпису – вибір підпису (рис. 2.36).

The screenshot shows the 'Conclude' form interface. At the top left is the 'Conclude' logo, and at the top right are links for 'About us' and 'Contact'. The form itself is a rounded rectangle containing several input fields: a 'Date:' field with a placeholder 'dd.mm.yyyy' and a calendar icon; a 'Sum:' field; a 'Full P I B:' field with a placeholder 'Full P I B'; and a 'Sign:' dropdown menu with the text 'Select an option'. A blue 'Sign' button is located at the bottom left of the form.

Рисунок 2.34 – Форма для заповнення угоди.

This image shows a close-up of the 'Date:' input field. The field contains the text '15.05.2023'. Below the field, a calendar for 'May 2023' is displayed. The calendar grid shows days from 1 to 31. The 15th is highlighted in blue, indicating it is the selected date. At the bottom of the calendar, there are 'Clear' and 'Today' buttons.

Рисунок 2.35 – Поле для введення дати.

Sign:

This image shows a close-up of the 'Sign:' dropdown menu. The dropdown is open, showing a list of options. The top option is 'Select an option'. Below it, there are several handwritten signatures. The signature 'Your sign' is clearly visible and appears to be the selected option.

Рисунок 2.36 – Поле для вибору підпису.

Після успішного підписання угоди, її статус міняється. На стороні відправника він стає «concluded» (рис 2.3), а на стороні отримувача – «signed» (рис. 2.38)

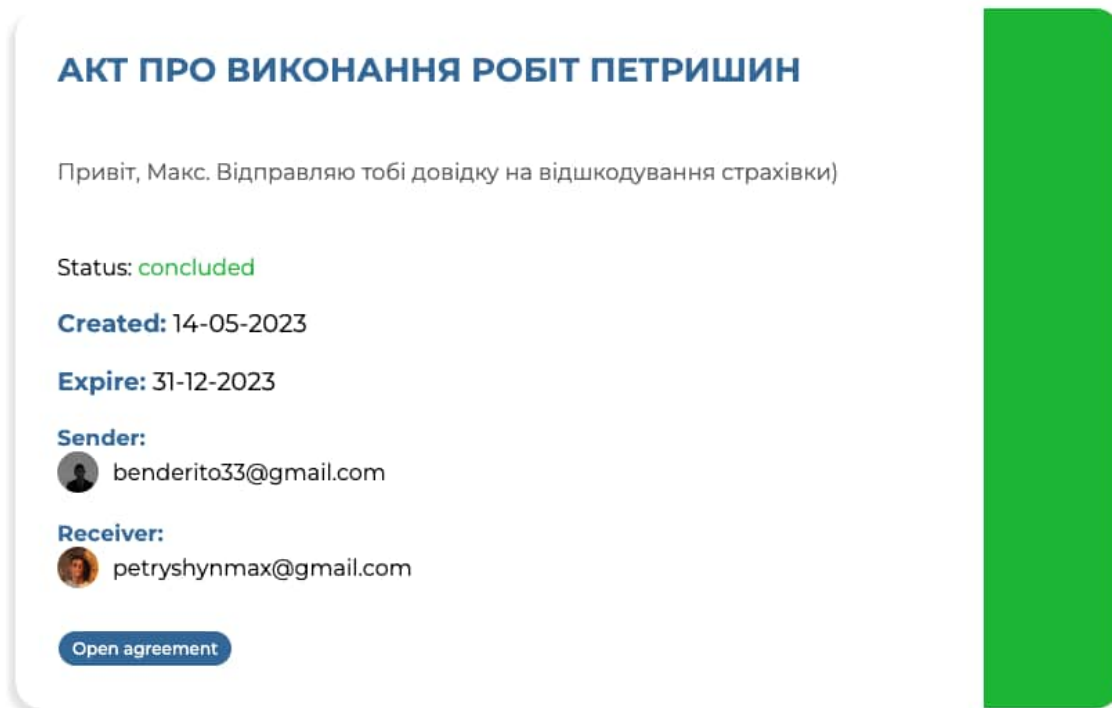


Рисунок 2.37 – Вигляд з сторони відправника.

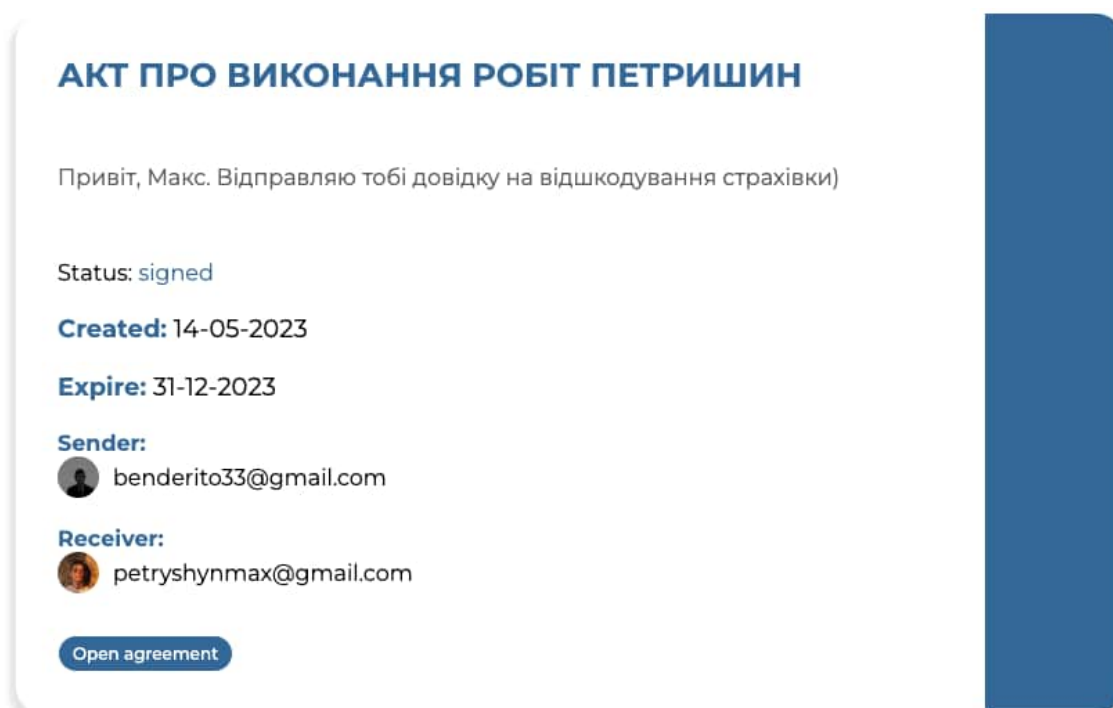


Рисунок 2.38 – Вигляд з сторони отримувача.

Також додаток підписує наш документ (рис. 2.39) по заданому шаблону, створює копію (рис. 2.40) та надає доступ **лише учасникам** угоди.

Акт виконання робіт					
					{{date}}
№	ПІБ застрахованої особи	Дата надання послуг	Назва наданих послуг	кількість	сума
1					
2					
3					
4					
5					

Названі послуги на суму: {{sum}}

Виконавець: {{fullPIB}}

Підпис: {{sign}}

Рисунок 2.39 – Угода до підписання.

Акт виконання робіт					
					2023-05-15
№	ПІБ застрахованої особи	Дата надання послуг	Назва наданих послуг	кількість	сума
1					
2					
3					
4					
5					

Названі послуги на суму: 1440 грн.

Виконавець: Петришин М.Ю

Підпис: 

Рисунок 2.40 – Угода після підписання.

Мобільний вигляд сторінки «Мій кабінет» зображено на рисунках 2.41, 2.42, 2.43, 2.44.

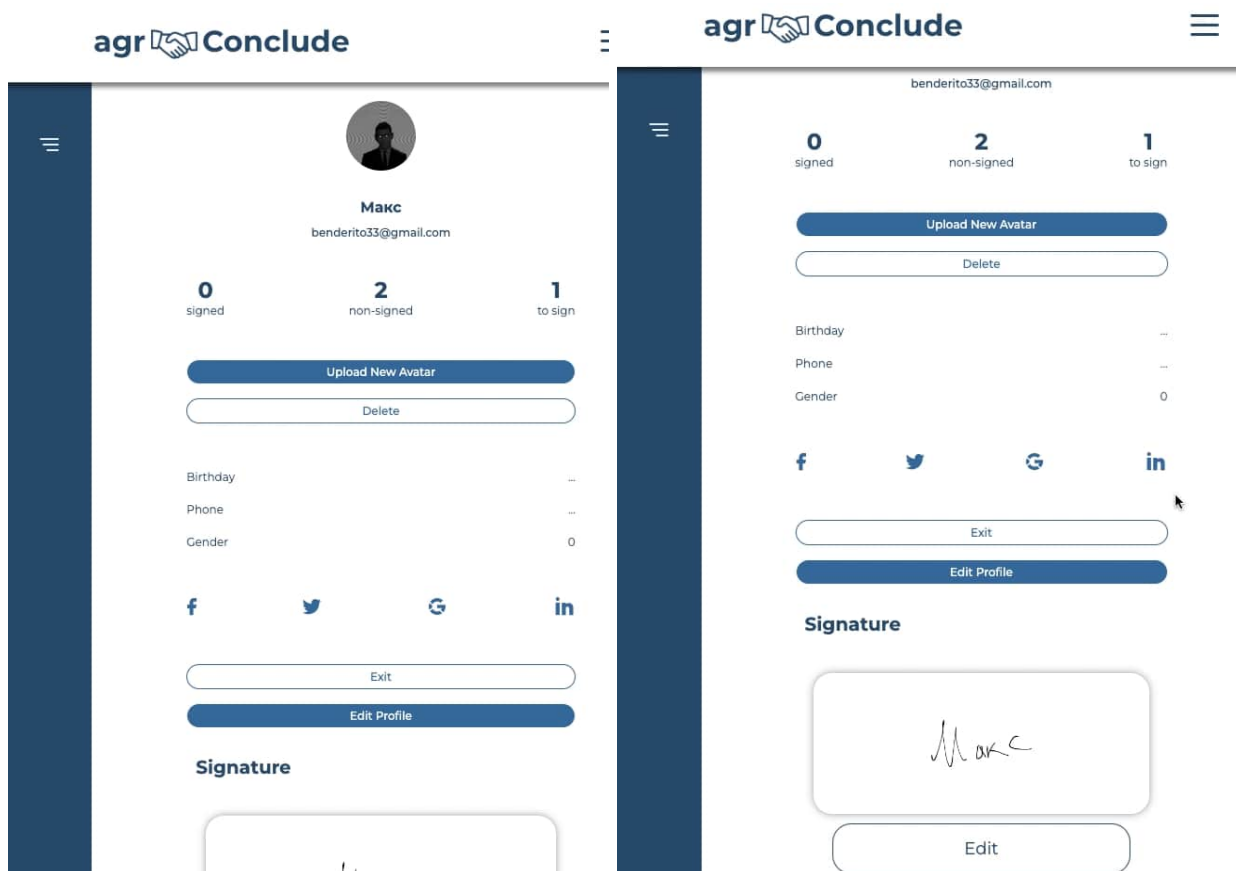


Рисунок 2.41 – Панель профілю на мобільних пристроях.

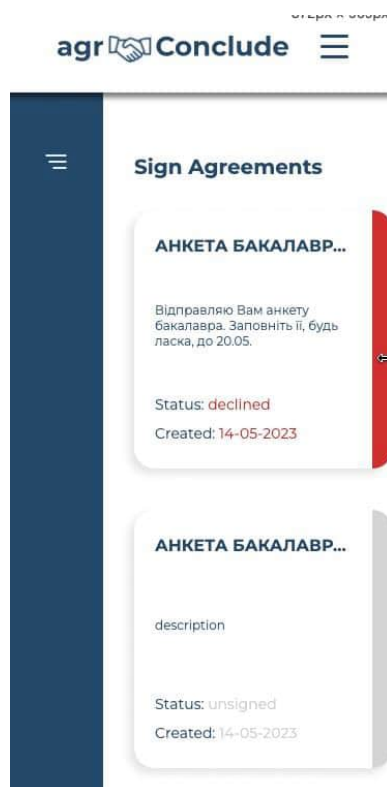


Рисунок 2.42 – Панель вхідних угод на мобільних пристроях.

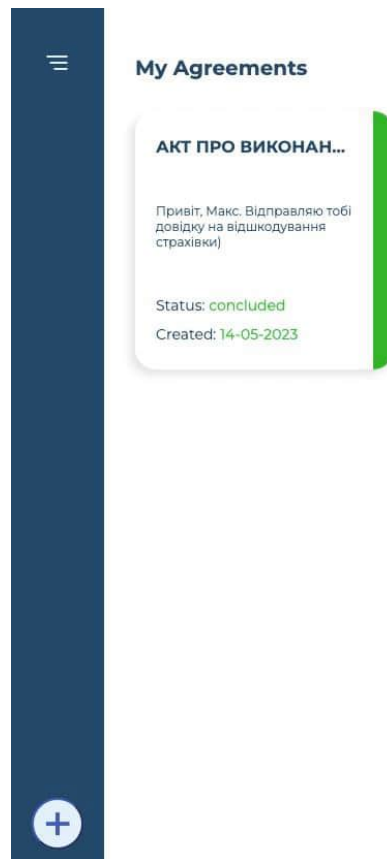


Рисунок 2.43 – Панель надісланих угод на мобільних пристроях.

Agreement Template:
Select an option

Receiver:
Select an option

Title:
Title

Description:
Description

Expire Date:
dd.mm.yyyy

Create

Рисунок 2.44 – Форма для створення угод на мобільних пристроях.

ВИСНОВКИ

Результатом моєї дипломної роботи стала платформа, яка дозволить користувачу зручно та швидко підписувати угоди. Завдяки зручному, простому та універсальному інтерфейсу користувач без проблем зможе створювати, видаляти угоди та слідкувати за їх кількістю, а також безпосередньо контролювати усі процеси незалежно від того яким пристроєм він користується – телефоном чи комп'ютером.

Додаток задовільняє усі поставлені до неї вимоги, є оптимізованим та легким до масштабування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Андрій Бояр, Катерина Махновець, Вплив глобальної пандемії на світову та вітчизняну економіку: Міжнародні відносини, суспільні комунікації та регіональні студії. URL: <https://relint.vnu.edu.ua/index.php/relint/article/view/219/197>
2. React documentation. URL: <https://uk.legacy.reactjs.org/>
3. ASP .NET CORE documentation. URL: <https://docs.microsoft.com/en-us/aspnet/core/>
4. Redux documentation. URL: <https://redux.js.org/introduction/getting-started>
5. Saga documentation. URL: <https://redux-saga.js.org/docs/introduction/GettingStarted>
6. React Signature documentation. URL: <https://www.npmjs.com/package/react-signature-canvas>
7. PostgreSQL documentation. URL: <https://www.postgresql.org/about/>
8. Formik documentation. URL: <https://formik.org/docs/overview>
9. Yup documentation. URL: <https://www.npmjs.com/package/yup>
10. Axios documentation. URL: <https://axios-http.com/docs/intro>
11. Google Drive API. URL: <https://developers.google.com/apis-explorer/#p/drive/v3/>
12. Google Docs API. URL: <https://developers.google.com/docs/api/>
13. React Router documentation. URL: <https://reactrouter.com/en/main/start/tutorial>
14. React Google Login documentation. URL: <https://www.npmjs.com/package/react-google-login>
15. Material UI documentation. URL: <https://mui.com/material-ui/getting-started/overview/>