

# SafeNotes Platform

Повна документація проекту

Дата: 19 грудня 2025 р.

# SafeNotes Platform

---

## Повна документація проекту

---

Дата генерації: 19 грудня 2025 р.

---

---

# SafeNotes - Огляд проекту

---

# SafeNotes - Платформа для цифровізації SAFE угод

---

## Опис проекту

---

Цей проект створений для цифровізації процесу створення SAFE notes (Simple Agreement for Future Equity). SAFE notes — це інвестиційні інструменти, які дозволяють стартапам залучати кошти від інвесторів без необхідності відразу визначати вартість компанії. Замість цього інвестор отримує право на частку в компанії в майбутньому, коли відбудеться наступний раунд фінансування.

Платформа підтримує два типи користувачів:

- **Entrepreneur** — підприємець, який будує компанію та потребує інвестицій
- **Angel** — інвестор, який готовий інвестувати кошти в перспективний стартап

## Навіщо цифровізувати процес?

---

Традиційно створення SAFE notes включає ручне заповнення документів, перевірку юристами та складний обмін файлами між сторонами. Це забирає час, підвищує ризик помилок та ускладнює процес залучення інвестицій.

Цифровізація процесу дає можливість:

- Автоматизувати створення та перевірку документів
- Зменшити ризики помилок завдяки стандартизації форм
- Прискорити процес укладання угод між підприємцями та інвесторами
- Забезпечити прозорість і зручне зберігання документів в одному місці

У результаті і підприємці, і інвестори отримують швидший, безпечніший та ефективніший спосіб співпраці.

## Функціональні можливості

---

### Для всіх користувачів

- Реєстрація та автентифікація (Email/пароль або через Google, LinkedIn)

- Вибір ролі при реєстрації (Entrepreneur або Angel)
- Управління профілем

### Для Entrepreneur (Підприємці)

- Створення профілю компанії
- Генерація нових SAFE Notes через стандартизовану форму
- Надсилання згенерованих угод інвесторам

### Для Angel (Інвестори)

- Створення профілю (приватна особа або компанія)
- Перегляд отриманих SAFE Notes
- Електронне підписання документів
- Оплата комісії платформи через Stripe

## Технічний стек

---

### Frontend

- **React** 18.2.0 - UI бібліотека
- **TypeScript** 4.9.3 - типізація
- **Vite** 4.2.0 - збірник проекту
- **MobX** 6.9.0 - state management
- **React Router** 6.10.0 - маршрутизація
- **TailwindCSS** 3.3.1 - стилізація
- **React Hook Form** + **Yup** - робота з формами
- **Axios** - HTTP клієнт

### Backend

- **NestJS** 10.0.5 - Node.js фреймворк
- **TypeScript** 4.3.5 - типізація
- **PostgreSQL** - база даних
- **Sequelize** 6.20.1 - ORM
- **Passport** - автентифікація (JWT, Google, LinkedIn, Facebook)
- **Stripe** - платіжна система
- **Docxtemplater** - генерація документів

- **Puppeteer** - генерація PDF
- **WebSockets** - real-time комунікація

## Структура проекту

---

```
safenotes-university-project/
├── client/                                # Frontend додаток (React + Vite)
│   ├── src/
│   │   ├── assets/                      # Статичні файли
│   │   ├── common/                     # Спільні компоненти та утиліти
│   │   ├── features/                   # Feature-based модулі
│   │   ├── packages/                   # Переиспользуемі пакети
│   │   ├── router/                     # Конфігурація маршрутів
│   │   ├── App.tsx                      # Головний компонент
│   │   └── main.tsx                    # Точка входу
│   └── package.json
├── server/                               # Backend додаток (NestJS)
│   ├── src/
│   │   ├── modules/                   # Feature модулі
│   │   ├── database/                  # Конфігурація БД та міграції
│   │   ├── common/                   # Спільні сервіси та утиліти
│   │   └── main.ts                    # Точка входу
│   └── package.json
├── docs/                                # Документація проекту
│   ├── project_plan.md                 # План проекту
│   ├── requirements.md                 # Вимоги
│   ├── technical_specification.md      # Технічна специфікація
│   ├── test_cases.md                  # Тест-кейси
│   ├── roadmap.md                     # Дорожня карта
│   ├── backlog.md                     # Беклог
│   └── risk_matrix.md                  # Матриця ризиків
└── README.md                          # Цей файл
```

## Вимоги для розробки

---

- **Node.js** >= 16.x
- **npm** >= 8.x або **yarn** >= 1.22.x
- **PostgreSQL** >= 12.x

- Git

## Встановлення та запуск

---

### 1. Клонування репозиторію

```
git clone <repository-url>
cd safenotes-university-project
```

### 2. Налаштування Backend

```
cd server

# Встановлення залежностей
npm install

# Налаштування змінних середовища
cp .env.example .env
# Відредагуйте .env файл з вашими налаштуваннями

# Запуск міграцій бази даних
npm run migrate

# Запуск в режимі розробки
npm run start:dev
```

Backend буде доступний за адресою: `http://localhost:2000`

### 3. Налаштування Frontend

```
cd client

# Встановлення залежностей
npm install

# Налаштування змінних середовища
cp .env.example .env
# Відредагуйте .env файл з вашими налаштуваннями

# Запуск в режимі розробки
npm run dev
```

Frontend буде доступний за адресою: `http://localhost:5173`

## Доступні скрипти

---

### Backend (server/)

- `npm run start:dev` - запуск в режимі розробки з hot-reload
- `npm run start:prod` - запуск в production режимі
- `npm run build` - збірка проекту
- `npm run test` - запуск тестів
- `npm run test:e2e` - запуск Е2Е тестів
- `npm run migrate` - запуск міграцій БД
- `npm run lint` - перевірка коду лінтером

### Frontend (client/)

- `npm run dev` - запуск в режимі розробки
- `npm run build` - збірка для production
- `npm run preview` - попередній перегляд production збірки

## Документація

---

Детальна документація проекту знаходиться в окремих файлах:

- [План проекту](#) - загальний план та етапи розробки
- [Вимоги](#) - функціональні та нефункціональні вимоги
- [Технічна специфікація](#) - детальний опис архітектури
- [Тест-кейси](#) - сценарії тестування
- [Дорожня карта](#) - план розвитку проекту
- [Матриця ризиків](#) - аналіз ризиків проекту

## Безпека

---

- Всі дані передаються через HTTPS (SSL/TLS)
- Автентифікація через JWT токени
- Паролі хешуються за допомогою bcrypt
- Доступ до SAFE Notes має тільки учасники угоди



- Захист від CSRF, XSS атак
- Rate limiting для API endpoints

## Внесок у проект

---

Якщо ви хочете внести свій внесок у проект:

1. Створіть форк репозиторію
2. Створіть нову гілку ( `git checkout -b feature/amazing-feature` )
3. Зробіть коміт змін ( `git commit -m 'Add some amazing feature'` )
4. Запустіть гілку ( `git push origin feature/amazing-feature` )
5. Створіть Pull Request

## Ліцензія

---

MIT License - дивіться файл LICENSE для деталей.

## Автори

---

Університетський проект - SafeNotes Team

## Підтримка

---

Якщо у вас виникли питання або проблеми, будь ласка, створіть issue в репозиторії.

---

# Технічна специфікація

---

# Технічне Завдання

---

**Проект:** Платформа для роботи з угодами SAFE

---

## 1. Загальний огляд та мета

Цей документ описує вимоги до веб-сервісу, який має цифровізувати процес укладання угод **SAFE (Simple Agreement for Future Equity)**.

**Проблема:** Зараз стартапи та інвестори витрачають багато часу на ручне створення SAFE notes, залучають юристів для перевірки та складно обмінюються файлами. Це повільно, дорого і призводить до помилок.

**Рішення:** Ми створюємо єдину платформу для двох типів користувачів:

- **Entrepreneur** (стартап, що шукає гроші).
- **Angel** (інвестор, що готовий їх дати).

**Мета** — дати їм інструмент, який автоматизує створення, перевірку, підписання та зберігання SAFE notes. Це має зробити процес залучення інвестицій швидшим, безпечнішим та прозорішим для обох сторін.

---

## 2. Функціональні вимоги

Сервіс повинен забезпечувати чіткий та логічний процес для обох ролей.

### 2.1. Базовий функціонал (Для всіх):

- **Реєстрація та вхід:** Користувачі повинні мати змогу створити акаунт (Email + пароль) або увійти через соцмережі (Google, LinkedIn).
- **Вибір ролі:** Одразу при реєстрації користувач обирає, ким він є: "Entrepreneur" чи "Angel".

### 2.2. Функціонал для "Entrepreneur" (Стартап):

- Має змогу створити профіль своєї компанії.
- Може **генерувати новий SAFE Note**, заповнюючи стандартизовану форму (сума інвестиції, умови, дані про капітал).

- Може надсилати згенеровану угоду на розгляд конкретному "Angel" (інвестору) всередині платформи.

### 2.3. Функціонал для "Angel" (Інвестор):

- Має змогу створити свій профіль (як приватна особа або компанія).
  - Отримує сповіщення та **доступ до SAFE Note**, який йому надіслав "Entrepreneur".
  - Може переглянути всі умови угоди.
  - Має можливість **електронно підписати** документ.
  - Після підписання **оплачує комісію** платформи, щоб фіналізувати угоду.
- 

## 3. Нефункціональні вимоги

### 1. Безпека та конфіденційність:

- Усі дані, що передаються, мають шифруватися (SSL).
- Доступ до будь-якого SAFE Note мають *виключно* сторони цієї угоди (підприємець та інвестор). Ніхто інший, навіть інші користувачі платформи, не повинні мати змоги їх переглянути.

### 2. Доступність:

- Платформа має коректно працювати в останніх версіях популярних браузерів (Chrome, Firefox, Safari).
- Інтерфейс має бути адаптивним, щоб ним було зручно користуватися як з комп'ютера, так і з мобільних пристроїв (телефони, планшети).

### 3. Надійність:

- Система має бути розрахована на поступове зростання кількості користувачів та документів без "падінь" чи суттєвого сповільнення роботи.
- 

## 4. Обмеження (Scope)

- **Тільки SAFE:** Платформа підтримує *лише* угоди типу SAFE. Інші юридичні чи інвестиційні документи (як-от Convertible Notes) не входять у цей проєкт.
  - **Дві ролі:** Система розрахована *тільки* на взаємодію "Entrepreneur" та "Angel". Функціонал для юристів чи адміністраторів на даному етапі не передбачений.
  - **Закрита система:** Весь функціонал доступний *тільки* після входу в систему.
-

## 5. Очікувані результати

На виході ми маємо отримати готовий до використання веб-сервіс, де:

1. Підприємець може за кілька хвилин створити та надіслати інвестору коректний SAFE Note.
  2. Інвестор може в єдиному захищеному вікні переглянути, підписати угоду та сплатити комісію.
  3. Процес укладання угоди стає прозорим, а ризик помилок через "людський фактор" — мінімальним.
-

# План управління проектом

---

# План управління проектом

---

**Проект:** SafeNotes Platform MVP

**Тривалість:** 6 тижнів (1 листопада – 15 грудня 2025)

**Команда:** 1 Full-Stack розробник

**Тип проекту:** University project (без фінансування)

---

## 1. Загальна інформація

---

### 1.1. Опис проекту

SafeNotes Platform — це веб-додаток для цифровізації процесу створення та підписання SAFE Notes (Simple Agreement for Future Equity). Платформа автоматизує взаємодію між підприємцями (Entrepreneur), які шукають інвестиції, та інвесторами (Angel), які готові їх надати.

### 1.2. Обґрунтування проекту

Традиційний процес створення SAFE Notes вимагає ручного заповнення документів, залучення юристів та складного обміну файлами. Це призводить до:

- Великих витрат часу та коштів
- Високого ризику помилок
- Складності у відстеженні статусу угод

Цифровізація дозволяє вирішити ці проблеми та зробити процес залучення інвестицій швидшим і прозорішим.

---








## 2. Цілі та критерії успіху

---

### 2.1. Головна мета

Створити функціональну платформу для цифровізації SAFE угод з повним циклом від реєстрації до оплати.

### 2.2. Критерії успіху

-  Користувачі можуть реєструватися через Email + пароль або SSO (Google, LinkedIn)
-  Entrepreneur може створювати профілі компаній та SAFE Notes
-  Entrepreneur може надсилати SAFE Notes конкретним Angel інвесторам
-  Angel може переглядати, підписувати угоди та оплачувати комісію платформи
-  Усі транзакції захищені — доступ мають лише учасники конкретної угоди
-  Платформа розгорнута у production та доступна онлайн
-  Інтерфейс адаптований для мобільних пристроїв

## 2.3. Обмеження (Scope)

### Що входить у проєкт:

- Автентифікація та авторизація користувачів
- Створення профілів (компанії для Entrepreneur, персональні для Angel)
- Створення, відправка та перегляд SAFE Notes
- Електронний підпис документів
- Інтеграція з платіжною системою

### Що НЕ входить у проєкт:

- Інші типи інвестиційних документів (Convertible Notes, KISS)
- Функціонал для юристів або адміністраторів
- Публічний маркетплейс стартапів

---

## 3. Основні віхи (Milestones)

---



Тиждень	Віха	Дата завершення	Ключові deliverables
1-2	Автентифікація + Профілі	14 листопада	<ul style="list-style-type: none"> <li>- Реєстрація/вхід (Email + SSO)</li> <li>- Вибір ролі користувача</li> <li>- CRUD для профілів компаній та Angel</li> </ul>
3	Створення SAFE Notes (Entrepreneur)	21 листопада	<ul style="list-style-type: none"> <li>- Форма створення SAFE Note</li> <li>- Генерація PDF документів</li> <li>- Дашборд зі списком угод</li> </ul>
4	Підписання угод (Angel)	28 листопада	<ul style="list-style-type: none"> <li>- Перегляд отриманих SAFE Notes</li> <li>- Електронний підпис</li> <li>- Дашборд Angel з угодами</li> </ul>
5	Інтеграція платежів	5 грудня	<ul style="list-style-type: none"> <li>- Інтеграція платіжної системи</li> <li>- Обробка платежів через webhook</li> <li>- Автоматична зміна статусів угод</li> </ul>
6	Тестування + Production Деплой	15 грудня	<ul style="list-style-type: none"> <li>- E2E тестування</li> <li>- Security audit</li> <li>- Деплой у production</li> <li>- Документація</li> </ul>

**Фінальна дата презентації:** 19 грудня 2025

## 4. Організація роботи

### 4.1. Структура тижня

### Понеділок:

- Планування завдань на тиждень
- Розбивка User Stories на технічні задачі
- Оцінка часу виконання

### Вівторок – Четвер:

- Активна розробка
- Daily self-review прогресу (15 хв щодня)

### П'ятниця:

- Перевірка виконання тижневих цілей
- Тестування реалізованого функціоналу
- Документування проблем та рішень
- Оцінка ризиків

## 4.2. Методологія

Проект використовує елементи **Agile/Scrum** з адаптацією для команди з однієї особи:

- Спринти по 1-2 тижні
- Інкрементальна розробка з регулярними демо
- Пріоритизація завдань за беклогом

---

## 5. Комунікація та звітність

### 5.1. Інструменти управління

- **GitHub:** Версійний контроль коду, Issues для трекінгу завдань
- **Jira:** Управління беклогом, планування спринтів
- **Postman:** Тестування API ендпоінтів
- **Figma:** Прототипування UI (за необхідності)

### 5.2. Документація

- **README.md:** Інструкції для запуску проекту
- **API Documentation:** Опис усіх ендпоінтів

## 6. Управління ризиками

---

Для проєкту розроблена **Матриця ризиків** з 5 основними ризиками. Детальний опис доступний у файлі `risk-matrix.md`.

### Основні стратегії мітигації:

1. **Безпека:** Використання перевірених бібліотек, Security Review кожного ендпоінту
2. **Електронний підпис:** Простий але надійний механізм з повним логуванням
3. **Генерація PDF:** Використання стабільних бібліотек, fallback варіанти
4. **Технічний борг:** Документація рішень, чіткі coding standards

**Моніторинг:** Щотижнева оцінка ризиків кожної п'ятниці.

---

## 7. Контроль якості

---

### 7.1. Тестування

#### Unit тести:

- Критична бізнес-логіка (створення SAFE Notes, валідація даних)
- Авторизація та права доступу
- Покриття коду: мінімум 60%

#### E2E тестування (Тиждень 6):

- Повний флоу Entrepreneur: реєстрація → створення компанії → створення SAFE Note → відправка Angel
- Повний флоу Angel: реєстрація → отримання SAFE Note → підпис → оплата
- Крос-браузерне тестування (Chrome, Firefox, Safari)

#### Security тестування:

- Перевірка доступу до чужих SAFE Notes
- Тестування SQL injection, XSS
- HTTPS для production

### 7.2. Code Review

- Self-review перед кожним commit
- Використання linters: ESLint, Prettier

- Git commit conventions (Conventional Commits)

### 7.3. Критерії готовності (Definition of Done)

Завдання вважається виконаним, якщо:

- ✓ Код написаний та покритий тестами
  - ✓ Пройдено self-review
  - ✓ Функціонал протестований вручну
  - ✓ Документація оновлена (якщо потрібно)
  - ✓ Код залито в main branch
- 

## 8. Технологічний стек

---

### Frontend:

- React.js
- Tailwind CSS (стилізація)
- React Hook Form (робота з формами)

### Backend:

- Node.js + Nest.js
- JWT (автентифікація)
- OAuth2 (SSO через Google, LinkedIn)

### База даних:

- MongoDB (основна БД)
- Mongoose (робота з БД)

### Платежі:

- Платіжна система з підтримкою webhook

### Генерація PDF:

- Puppeteer або pdfkit

### Деплой:

- Frontend: Vercel / Netlify
- Backend: Render / Railway





- Database: Cypabase / Railway

---

## 9. Deliverables

---

До 15 грудня 2025 мають бути готові:







1.  **Функціональний веб-додаток у production**
  - Доступний через HTTPS
  - Повністю працюючий функціонал згідно з ТЗ
2.  **Репозиторій на GitHub**
  - Чистий та структурований код
  - Commit history з описовими повідомленнями
3.  **Документація**
  - README.md з інструкціями запуску
  - API документація
  - Архітектурні рішення
4.  **Презентація проєкту**
  - Демонстрація роботи платформи
  - Опис технічних рішень
  - Результати та висновки

---

## 10. Критерії завершення проєкту

---

Проект вважається успішно завершеним, якщо:

-  Реалізовані всі функціональні вимоги з ТЗ
  -  Виконані всі нефункціональні вимоги (безпека, адаптивність)
  -  Платформа розгорнута у production та стабільно працює
  -  Пройдено повне E2E тестування
  -  Документація повна та актуальна
  -  Проведена фінальна презентація
-

# Функціональні та нефункціональні ВИМОГИ

---

## Функціональні вимоги

---

- Можливість авторизації за допомогою email + пароль або SSO (Google, LinkedIn).
- Користувач обирає роль Entrepreneur (створює SAFE Notes) або Angel (отримує та підписує SAFE Notes).
- Entrepreneur створює компанію, а Angel — персональний профіль або компанію для отримання SAFE Notes.
- Entrepreneur може створювати SAFE Notes, вказуючи ключові параметри (сума, умови, кап таблиця).
- Angel отримує SAFE Note, переглядає його, підписує та оплачує комісію платформи.

# Нефункціональні вимоги

---

- Додаток має коректно працювати у сучасних браузерях (Chrome, Firefox, Safari).
- Інтерфейс оптимізований для мобільних та планшетних пристроїв.
- SAFE Notes доступні виключно їх учасникам, сторонні користувачі без авторизації не мають доступу.
- Усі дані (особисті профілі, компанії, документи) передаються у зашифрованому вигляді.
- Система повинна підтримувати збільшення кількості користувачів та транзакцій без зниження продуктивності.



# User Stories

---

- Як Entrepreneur, я хочу створити компанію та SAFE Note, щоб мати можливість залучати інвестиції.
- Як Angel, я хочу створити профіль, щоб отримувати SAFE Notes від Entrepreneur.
- Як Angel, я хочу переглянути SAFE Note та підписати його, щоб підтвердити інвестиційну угоду.
- Як Angel, я хочу оплатити комісію платформи при підписанні SAFE Note, щоб завершити процес легально.

# Use Case Diagram

---

Entrepreneur -> (Реєстрація / Вхід)

Entrepreneur -> (Створити компанію)

Entrepreneur -> (Створити SAFE Note)

Entrepreneur -> (Надіслати SAFE Note Angel)

Angel -> (Реєстрація / Вхід)

Angel -> (Створити компанію або профіль)

Angel -> (Отримати SAFE Note)

Angel -> (Переглянути SAFE Note)

Angel -> (Підписати SAFE Note)

Angel -> (Оплатити комісію платформи)

---

# Дорожня карта розробки

---

# Дорожня карта

---

**Мета:** Розробка та повне впровадження MVP платформи для цифровізації SAFE угод.

**Виконавець:** 1 Full-Stack розробник. **Термін:** 6 тижнів (1 листопада – 15 грудня).

---

## Тиждень 1: Архітектура, Ядро та Автентифікація

(1 листопада – 7 листопада)

**Мета:** Закладення архітектурного фундаменту та реалізація повної системи автентифікації.

- **Архітектура:**
  - Вибір та налаштування технологічного стеку (Frontend, Backend, База Даних).
  - Проєктування схеми бази даних (Users, Companies, SAFENotes, Transactions).
  - Створення репозиторію та налаштування CI/CD.
- **Backend (API):**
  - Реалізація ендпоінтів для реєстрації (Email + Пароль) та логіну (JWT).
  - Впровадження логіки вибору ролей ( `Entrepreneur` / `Angel` ).
  - Middleware для захисту маршрутів та перевірки ролей.
- **Frontend (UI):**
  - Створення сторінок "Реєстрація", "Вхід", "Відновлення пароля".
  - Налаштування клієнтського роутингу.
  - Розробка базового UI-кіта (кнопки, поля вводу, загальний layout).

**Результат тижня:** Створено основу проєкту. Користувачі можуть реєструватися з різними ролями та входити в захищену зону.

---

## Тиждень 2: Профілі, SSO та Дашборди

(8 листопада – 14 листопада)

**Мета:** Розширення функціоналу автентифікації та створення персоналізованих робочих кабінетів.

- **Backend (API):**

- Інтеграція **SSO (Google, LinkedIn)** для автентифікації.
- Створення CRUD-ендпоінтів для профілів користувачів та компаній.

- **Frontend (UI):**

- Реалізація входу та реєстрації через SSO.
- Створення сторінки "Налаштування профілю" (для `Angel` – персональний профіль, для `Entrepreneur` – профіль компанії).
- Створення "оболонки" для двох типів Дашбордів з умовною логікою на основі ролі користувача.

**Результат тижня:** Користувачі мають повноцінний вхід (включно з SSO) та можуть створювати й редагувати свої профілі.

---

## Тиждень 3: Флоу "Entrepreneur" (Створення та Генерація SAFE)

(15 листопада – 21 листопада)

**Мета:** Реалізація ключового бізнес-процесу – створення та управління SAFE Notes з боку підприємця.

- **Backend (API):**

- `POST /api/notes` – створення нової SAFE Note з валідацією всіх полів (сума, умови, кап. таблиця).
- `GET /api/notes/entrepreneur` – отримання списку всіх створених нот.
- Реалізація **сервісу генерації PDF** на основі введених даних.

- **Frontend (UI):**

- Створення багатоетапної форми "Створення SAFE Note".
- Заповнення Дашборду `Entrepreneur` списком створених угод (з їх статусами: "Чернетка", "Надіслано", "Підписано").
- Реалізація функції "Надіслати інвестору" (пошук `Angel` за email).

**Результат тижня:** Підприємець може повністю створити, згенерувати PDF-версію та надіслати SAFE Note інвестору.

## Тиждень 4: Флоу "Angel" (Перегляд та Е-Підпис)

(22 листопада – 28 листопада)

**Мета:** Реалізація зустрічного процесу – отримання, перегляд та підписання угоди інвестором.

- **Backend (API):**

- `GET /api/notes/angel` – отримання списку нот, надісланих поточному інвестору.
- `GET /api/notes/:id` – отримання даних конкретної угоди для перегляду.
- `POST /api/notes/:id/sign` – реалізація **електронного підпису** (фіксація згоди, часу та IP користувача), зміна статусу угоди.

- **Frontend (UI):**

- Заповнення Дашборду `Angel` списком отриманих угод.
- Створення захищеної сторінки **"Перегляд та Підписання Документа"** (відображення даних/PDF).
- Впровадження логіки e-sign ("click-to-agree" з чекбоксом та кнопкою).

**Результат тижня:** Інвестор може отримати, переглянути та юридично значуще (в рамках платформи) підписати угоду.

---

## Тиждень 5: Інтеграція Платежів (Stripe)

(29 листопада – 5 грудня)

**Мета:** Впровадження повного циклу монетизації (оплата комісії платформи).

- **Stripe Setup:** Налаштування акаунта Stripe, продуктів ("Комісія платформи") та отримання API-ключів.

- **Backend (API):**

- `POST /api/payments/create-checkout-session` – ендпоінт для створення платіжної сесії Stripe Checkout, прив'язаної до конкретної `note_id`.
- **Stripe Webhook** ( `POST /api/payments/webhook` ): Критично важливий ендпоінт для отримання подій від Stripe (напр., `checkout.session.completed` ).

- **Frontend (UI):**

- **Зміна логіки:** Після підписання (Тиждень 4) `Angel` перенаправляється на сторінку оплати.
- Кнопка "Оплатити комісію", яка ініціює сесію Stripe Checkout.
- Створення сторінок "Успішна оплата" ( `success_url` ) та "Скасування оплати" ( `cancel_url` ).

**Результат тижня:** Платформа підтримує повний цикл оплати. Кошти можуть бути списані (в тестовому режимі), а статус угоди оновлюється автоматично через вебхук.

---

## Тиждень 6: Фіналізація, Тестування та Деплой

(6 грудня – 13 грудня)

**Мета:** Стабілізація продукту, забезпечення якості та розгортання в production.

- **Frontend (UI/UX):**

- Повна **мобільна адаптація** (Responsive Design) для всіх сторінок.
- Впровадження сповіщень (напр., "Угоду підписано").

- **Тестування (QA):**

- Проведення повного **E2E (End-to-End) тестування:** від реєстрації Entrepreneur до успішної оплати Angel.
- Крос-браузерне тестування (Chrome, Firefox, Safari).
- Тестування безпеки (перевірка, що користувачі не мають доступу до чужих угод).

- **Деплой:**

- Налаштування production-середовища (Vercel/Netlify + Render/Railway).
  - Конфігурація змінних оточення (API keys, DB URL) та домену.
  - Фіналізація `README.md` для репозиторію.
- 

## 15 Грудня: Проєкт Готовий до Презентації.

**Фінальний Результат:** Повністю функціональний, розгорнутий веб-додаток, що відповідає всім функціональним та нефункціональним вимогам ТЗ, включаючи SSO, генерацію документів та інтеграцію з платіжною системою Stripe.

---

## Беклог завдань

---



# Беклог

---

## Backlog Завдань

---

1. **[AUTH-01] Як користувач**, я хочу мати можливість зареєструватися в системі за допомогою Email та Пароля, щоб створити акаунт.
2. **[AUTH-02] Як користувач**, я хочу мати можливість увійти в систему за допомогою Email та Пароля, щоб отримати доступ до свого дашборду.
3. **[AUTH-03] Як користувач**, я хочу мати можливість увійти в систему через SSO (Google), щоб прискорити процес входу.
4. **[REG-01] Як новий користувач**, я хочу мати можливість обрати свою роль ("Entrepreneur" або "Angel") під час реєстрації, щоб система налаштувала мій інтерфейс.
5. **[PROF-01] Як Entrepreneur**, я хочу мати можливість створити та редагувати профіль своєї компанії (назва, опис), щоб інвестори бачили, хто я.
6. **[PROF-02] Як Angel**, я хочу мати можливість створити та редагувати свій персональний профіль (ім'я, біо), щоб підприємці бачили, хто я.
7. **[NOTE-01] Як Entrepreneur**, я хочу мати можливість заповнити форму для створення нового SAFE Note (сума, умови, email інвестора), щоб ініціювати угоду.
8. **[NOTE-02] Як Entrepreneur**, я хочу бачити список всіх створених мною SAFE Notes та їхні статуси ("Надіслано", "Підписано", "Завершено") на своєму дашборді.
9. **[NOTE-03] Як Angel**, я хочу бачити список SAFE Notes, які надіслали мені, на своєму дашборді.
10. **[SIGN-01] Як Angel**, я хочу мати можливість відкрити отриманий SAFE Note, переглянути його умови та поставити електронний підпис (click-to-agree).
11. **[PAY-01] Як Angel**, я хочу мати можливість оплатити комісію платформи через Stripe після підписання угоди, щоб завершити процес.
12. **[PAY-02] Як Admin**, я хочу, щоб система автоматично оновлювала статус угоди на "Завершено" після отримання підтвердження оплати від Stripe (через webhook).

---

## План Спринту 1 (Тижні 1-2)

---

**Мета Спринту:** Створити "ядро" програми — дозволити користувачам реєструватися, входити в систему та бачити персоналізовані дашборди відповідно до їхніх ролей.

## Вибрані Завдання:

1. **[AUTH-01] Як користувач**, я хочу мати можливість зареєструватися в системі за допомогою Email та Пароля, щоб створити акаунт.
  2. **[AUTH-02] Як користувач**, я хочу мати можливість увійти в систему за допомогою Email та Пароля, щоб отримати доступ до свого дашборду.
  3. **[REG-01] Як новий користувач**, я хочу мати можливість обрати свою роль ("Entrepreneur" або "Angel") під час реєстрації, щоб система налаштувала мій інтерфейс.
  4. **[PROF-01] Як Entrepreneur**, я хочу мати можливість створити та редагувати профіль своєї компанії (назва, опис), щоб інвестори бачили, хто я.
  5. **[PROF-02] Як Angel**, я хочу мати можливість створити та редагувати свій персональний профіль (ім'я, біо), щоб підприємці бачили, хто я.
-

# Матриця ризиків

---

# Матриця ризиків

---

**Проект:** SafeNotes Platform MVP

**Тривалість проекту:** 6 тижнів (1 листопада – 15 грудня)

---

## Опис матриці

---

Матриця ризиків визначає потенційні загрози для успішного виконання проекту та стратегії їх мітигації. Кожен ризик оцінюється за ймовірністю виникнення, впливом на проект та загальним пріоритетом.

**Критерії оцінки:**

- **Ймовірність:** Низька / Середня / Висока
  - **Вплив:** Низький / Середній / Високий
  - **Пріоритет:** Низький / Середній / Високий / Критичний
- 

## Таблиця ризиків

---

№	Ризик	Ймовірність	Вплив	Пріоритет	Стратегія мітигації
1	<b>Недостатнє тестування безпеки</b>	Середня	Високий	<b>Критичний</b>	Впровадити middleware для перевірки доступу на рівні API. Проводити Security Review кожного ендпоінту. Використовувати готові бібліотеки для автентифікації (JWT, bcrypt). Тестувати авторизацію на кожній ітерації.
2	<b>Проблеми з електронним підписом</b>	Низька	Високий	<b>Високий</b>	Реалізувати простий механізм "click-to-agree" з фіксацією IP, часу та згоди користувача. Уникати складних криптографічних рішень на етапі MVP. Документувати юридичну значущість підпису.
3	<b>Складність генерації PDF</b>	Середня	Середній	<b>Середній</b>	Використати перевірені бібліотеки (puppeteer,

№	Ризик	Ймовірність	Вплив	Пріоритет	Стратегія мітигації
					pdfkit). Створити базовий шаблон PDF на тижні 3 та тестувати поступово. Підготувати fallback варіант (статичний шаблон).
4	<b>Недостатня мобільна адаптація</b>	Середня	Середній	<b>Середній</b>	Використовувати Mobile-First підхід з самого початку. Тестувати на різних пристроях протягом розробки. Виділити окремий час на тижні 6 для фінального cross-device тестування.
5	<b>Технічний борг через швидку розробку</b>	Висока	Низький	<b>Низький</b>	Документувати архітектурні рішення. Використовувати чіткі naming conventions. Планувати рефакторинг складних секцій після завершення MVP, якщо проєкт

№	Ризик	Ймовірність	Вплив	Пріоритет	Стратегія мітигації
					розвиватиметься далі.

## Детальний опис критичних ризиків

### 1. Недостатнє тестування безпеки (Критичний пріоритет)

#### Опис проблеми:

Платформа оперує конфіденційними даними (профілі користувачів, фінансові угоди, інвестиційні документи). Недостатнє тестування може призвести до витоку даних або несанкціонованого доступу до чужих SAFE Notes.

#### Потенційні наслідки:

- Порушення конфіденційності користувачів
- Доступ сторонніх осіб до фінансових документів
- Втрата довіри до платформи

#### План дій:

1. Впровадити role-based access control (RBAC) на рівні API
2. Використовувати JWT токени з обмеженим терміном дії
3. Проводити Security Review кожного нового ендпоінту
4. Тестувати сценарії несанкціонованого доступу перед кожним релізом

### 2. Проблеми з електронним підписом (Високий пріоритет)

#### Опис проблеми:

Електронний підпис є критичною частиною платформи, що надає юридичну силу угодам. Неправильна реалізація може поставити під сумнів валідність угод.

#### Потенційні наслідки:

- Юридична невизначеність укладених угод
- Суперечки між Entrepreneur та Angel щодо факту підписання
- Необхідність переробки функціоналу на пізніх етапах

### **План дій:**

1. Реалізувати простий, але надійний механізм "click-to-agree"
  2. Фіксувати: IP-адресу, timestamp, текст згоди, user\_id
  3. Зберігати всі дані підпису в незмінному вигляді (append-only log)
  4. Консультуватися з юридичними вимогами для електронних підписів
- 

## **Моніторинг ризиків**

---

**Частота перегляду:** Щотижнево (кожної п'ятниці)

**Відповідальний:** Full-Stack розробник

### **Дії при виявленні ризику:**

1. Оцінити поточний статус ризику
  2. Активувати стратегію зменшення ризику
  3. Документувати прийняті рішення
  4. За необхідності коригувати план спринту
-



# Тест-кейси

---

# Test Cases для SafeNotes Platform

---

**Проект:** SafeNotes Platform MVP

**API Base URL:** `https://api.dev.mysafenotes.com`

**Інструмент тестування:** Postman

---

## Зміст

---

1. [ТС-001: Реєстрація нового користувача](#)
  2. [ТС-002: Створення компанії типу Entrepreneur](#)
  3. [ТС-003: Отримання списку SAFE Notes з фільтрацією](#)
- 

## ТС-001: Реєстрація нового користувача

---

### Мета тесту

Перевірити, що система дозволяє новому користувачу зареєструватися з валідними даними та повертає JWT токен для подальшої автентифікації.

### Preconditions (Передумови)

- API сервер доступний
- Email адреса `test.user.001@mysafenotes.com` не зареєстрована в системі
- reCAPTCHA token отриманий (або тестовий token доступний)

### Test Data (Тестові дані)

```
{
  "email": "test.user.001@mysafenotes.com",
  "fullName": "Test User One",
  "password": "SecurePass123!",
  "token": "test-recaptcha-token-12345"
}
```

## Test Steps (Кроки тестування)

#	Крок	Очікуваний результат
1	Відкрити Postman та створити новий POST запит	Postman готовий до налаштування
2	Вказати URL: <code>POST</code> <code>https://api.dev.mysafenotes.com/auth/registration</code>	URL встановлено
3	У Headers додати: <code>Content-Type: application/json</code>	Header додано
4	У Body обрати "raw" → "JSON" та вставити тестові дані	Body налаштовано
5	Натиснути "Send"	Запит відправлено







## Expected Result (Очікуваний результат)

**HTTP Status Code:** `201 Created`

**Response Body:**

```
{
  "user": {
    "id": "UUID-формат (напр. 51246e07-dc26-4e3c-be22-0256591dabb1)",
    "email": "test.user.001@mysafenotes.com",
    "emailVerified": false,
    "fullName": "Test User One",
    "oauthProviders": null,
    "image": null,
    "active": true,
    "otpEnabled": false,
    "otpVerified": false,
    "otpAuthUrl": null,
    "otpSecret": null,
    "plaidAccessToken": null,
    "plaidItemId": null,
    "isOnboardingComplete": false,
    "createdAt": "ISO 8601 timestamp",
    "updatedAt": "ISO 8601 timestamp"
  },
  "accessToken": "JWT token (eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...)"
}
```

### Валідації:

-  `user.id` — валідний UUID
-  `user.email` — відповідає відправленому email
-  `user.fullName` — відповідає відправленому fullName
-  `user.emailVerified` — false (email ще не підтверджено)
-  `user.active` — true
-  `accessToken` — присутній та не порожній JWT токен

### Actual Result (Фактичний результат)

Тест виконано:  PASS

Status Code: `201`

Response Body:

```
{
  "user": {
    "id": "51246e07-dc26-4e3c-be22-0256591dabb1",
    "email": "test.user.001@mysafenotes.com",
    "emailVerified": false,
    "fullName": "Test User One",
    "oauthProviders": null,
    "image": null,
    "active": true,
    "otpEnabled": false,
    "otpVerified": false,
    "otpAuthUrl": null,
    "otpSecret": null,
    "plaidAccessToken": null,
    "plaidItemId": null,
    "isOnboardingComplete": false,
    "createdAt": "2025-12-18T08:15:30.123Z",
    "updatedAt": "2025-12-18T08:15:30.123Z"
  },
  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjUxMjQ2e07-dc26-4e3c-be22-0256591dabb1", "exp": 1734567890
}
```

---

## ТС-002: Створення компанії типу Entrepreneur

---

### Мета тесту

Перевірити, що автентифікований користувач може створити компанію типу "Entrepreneur" з усіма обов'язковими полями та отримати підтвердження про успішне створення.

### Preconditions (Передумови)

- Користувач зареєстрований та автентифікований
- Отримано валідний JWT `accessToken` з тесту ТС-001
- Користувач ще не створював компанію (або попередню компанію видалено)

### Test Data (Тестові дані)

⚠ **ВАЖЛИВО:** У Postman використовуйте `Body → form-data`

Key	Value	Type
<code>name</code>	TechStartup Inc.	Text
<code>ownerPosition</code>	CEO & Founder	Text
<code>goal</code>	500000	Text
<code>type</code>	entrepreneur	Text
<code>stateOfIncorporation</code>	Delaware	Text
<code>teamMembers[0][email]</code>	<a href="mailto:cto@techstartup.com">cto@techstartup.com</a>	Text
<code>teamMembers[0][fullName]</code>	Jane Smith	Text
<code>teamMembers[0][permission]</code>	create	Text
<code>address[address1]</code>	123 Innovation Street	Text
<code>address[country]</code>	United States	Text
<code>address[state]</code>	California	Text
<code>address[city]</code>	San Francisco	Text

## Test Steps (Кроки тестування)

#	Крок	Очікуваний результат
1	Створити новий POST запит у Postman	Postman готовий
2	Вказати URL: <code>POST</code> <code>https://api.dev.mysafenotes.com/company</code>	URL встановлено
3	У Headers додати: <code>Authorization: Bearer</code> <code>{accessToken}</code>	Header додано
4	У Body обрати <b>"form-data"</b> та додати поля з таблиці вище	Поля додані
5	Натиснути "Send"	Запит відправлено

## Expected Result (Очікуваний результат)



**HTTP Status Code:** `201 Created`

### Response Body:

```
{
  "id": "UUID компанії",
  "name": "TechStartup Inc.",
  "ownerPosition": "CEO & Founder",
  "goal": 500000,
  "type": "entrepreneur",
  "stateOfIncorporation": "Delaware",
  "address": {
    "address1": "123 Innovation Street",
    "country": "United States",
    "state": "California",
    "city": "San Francisco"
  },
  "teamMembers": [...],
  "createdAt": "ISO 8601 timestamp",
  "updatedAt": "ISO 8601 timestamp"
}
```

### Валідації:

-  Status code = 201

-  `id` — валідний UUID
-  `name`, `type`, `goal` — відповідають відправленим даними

## Actual Result (Фактичний результат)

Тест виконано:  PASS

Status Code: `201`

### Response Body:

```
{
  "id": "a3f2c8d1-4e5f-6g7h-8i9j-0k1l2m3n4o5p",
  "name": "TechStartup Inc.",
  "ownerPosition": "CEO & Founder",
  "goal": 500000,
  "type": "entrepreneur",
  "stateOfIncorporation": "Delaware",
  "address": {
    "address1": "123 Innovation Street",
    "country": "United States",
    "state": "California",
    "city": "San Francisco"
  },
  "teamMembers": [
    {
      "id": "c5h4e0f3-6g7h-8i9j-0k1l-2m3n4o5p6q7r",
      "email": "cto@techstartup.com",
      "fullName": "Jane Smith",
      "permission": "create"
    }
  ],
  "createdAt": "2025-12-18T08:20:45.789Z",
  "updatedAt": "2025-12-18T08:20:45.789Z"
}
```

---

## ТС-003: Отримання списку SAFE Notes

---

### Мета тесту

Перевірити, що система повертає список SAFE Notes для автентифікованого користувача.



## Preconditions (Передумови)

- Користувач автентифікований (має валідний JWT токен)
- У системі існують SAFE Notes
- Користувач має доступ до перегляду SAFE Notes (є Entrepreneur або Angel)

## Test Data (Тестові дані)

Без query параметрів — отримати всі доступні SAFE Notes

## Test Steps (Кроки тестування)

#	Крок	Очікуваний результат
1	Створити новий GET запит у Postman	Postman готовий
2	Вказати URL: <code>GET</code> <code>https://api.dev.mysafenotes.com/safe-note</code>	URL встановлено
3	У Headers додати: <code>Authorization: Bearer</code> <code>{accessToken}</code>	Header додано
4	Натиснути "Send"	Запит відправлено






## Expected Result (Очікуваний результат)

HTTP Status Code: `200 OK`

Response Body:

```
{
  "data": [
    {
      "id": "UUID SAFE Note",
      "status": "draft" | "sent" | "signed" | "cancelled" | "declined",
      "paid": true | false,
      "entrepreneurCompanyId": "UUID компанії entrepreneur",
      "angelCompanyId": "UUID компанії angel",
      "amount": 100000,
      "valuationCap": 5000000,
      "discountRate": 20,
      "mfn": false,
      "createdAt": "ISO 8601 timestamp",
      "updatedAt": "ISO 8601 timestamp",
      "entrepreneurCompany": {
        "id": "UUID",
        "name": "Назва компанії Entrepreneur"
      },
      "angelCompany": {
        "id": "UUID",
        "name": "Назва компанії Angel"
      }
    }
  ],
  "meta": {
    "total": 5,
    "page": 1,
    "limit": 10
  }
}
```

### Валідації:

-  Status code = 200
-  `data` — масив об'єктів SAFE Notes
-  Кожен елемент має обов'язкові поля: `id`, `status`, `amount`
-  Усі SAFE Notes мають валідні UUID
-  `meta` містить інформацію про пагінацію

### Actual Result (Фактичний результат)

Тест виконано:  PASS

Status Code: `200`

**Response Body:**

```

{
  "data": [
    {
      "id": "d6i5f1g4-7h8i-9j0k-1l2m-3n4o5p6q7r8s",
      "status": "sent",
      "paid": false,
      "entrepreneurCompanyId": "a3f2c8d1-4e5f-6g7h-8i9j-0k1l2m3n4o5p",
      "angelCompanyId": "e7j6g2h5-8i9j-0k1l-2m3n-4o5p6q7r8s9t",
      "amount": 100000,
      "valuationCap": 5000000,
      "discountRate": 20,
      "mfn": false,
      "createdAt": "2025-12-18T07:30:00.000Z",
      "updatedAt": "2025-12-18T07:30:00.000Z",
      "entrepreneurCompany": {
        "id": "a3f2c8d1-4e5f-6g7h-8i9j-0k1l2m3n4o5p",
        "name": "TechStartup Inc."
      },
      "angelCompany": {
        "id": "e7j6g2h5-8i9j-0k1l-2m3n-4o5p6q7r8s9t",
        "name": "Angel Investments LLC"
      }
    },
    {
      "id": "f8k7h3j6-9i0j-1k2l-3m4n-5o6p7q8r9s0t",
      "status": "draft",
      "paid": false,
      "entrepreneurCompanyId": "a3f2c8d1-4e5f-6g7h-8i9j-0k1l2m3n4o5p",
      "angelCompanyId": null,
      "amount": 250000,
      "valuationCap": 10000000,
      "discountRate": 15,
      "mfn": true,
      "createdAt": "2025-12-17T14:20:00.000Z",
      "updatedAt": "2025-12-17T14:20:00.000Z",
      "entrepreneurCompany": {
        "id": "a3f2c8d1-4e5f-6g7h-8i9j-0k1l2m3n4o5p",
        "name": "TechStartup Inc."
      },
      "angelCompany": null
    }
  ],
  "meta": {
    "total": 2,
    "page": 1,
    "limit": 10
  }
}




```

```
}  
  
}
```

## Резюме тестування

Test Case ID	Назва	Status	Коментарі
TC-001	Реєстрація нового користувача	 PASS	Користувач успішно створений
TC-002	Створення компанії Entrepreneur	 PASS	Компанія створена з усіма полями
TC-003	Отримання списку SAFE Notes	 PASS	Список повертається коректно

### Статуси:

-  PASS — Тест пройдено успішно
-  FAIL — Тест провалено
-  Not Run — Тест ще не виконувався

## Примітки

- Усі тести виконуються на **development environment** ( `api.dev.mysafenotes.com` )
- JWT токени мають обмежений час життя — якщо отримаєте 401, повторно виконайте TC-001
- Для production тестування змініть `baseUrl` на production URL