МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук, фізики та математики
Кафедра комп'ютерних наук та програмної інженерії

РОЗРОБКА СТАРТАП ПЛАТФОРМИ KSU-STARTUPS З ВИКОРИСТАННЯМ ANGULAR ТА FIREBASE

Кваліфікаційна робота (проєкт)
на здобуття ступеня вищої освіти «бакалавр»

Виконав: здобувач спеціальності: 121
Інженерія програмного забезпечення
Освітньо-професійної програми:
Інженерія програмного забезпечення
Щербина Володимир Володимирович
Керівник: доктор педагогічних наук,
професор Співаковський Олександр
Володимирович
Рецензент: доцент кафедри алгебри,
геометрії та математичного аналізу, ХДУ
Григор'єва Валентина Борисівна

Івано-Франківськ – 2023

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API .....Application Programming Interface

CSS .....Cascading Style Sheets

DOM ..... Document Object Mod

HTML..... HyperText Markup Language

JS ..... JavaScript

JSON .....JavaScript Object Notation

SSL .....Secure Socket Layer

CDN ..... Content Delivery Network

ВСТУП

Головна проблема великої кількості стартапів, котрі гинуть ще на початку свого шляху, полягає в нестачі коштів, адже знайти спонсора не так вже й легко. Це одна з причин, які лягли в основу ідеї створення платформи KSU-Startups, яка повинна допомогти стартапам знайти фінансування.

Ця кваліфікаційна робота (проєкт) присвячена розробці веб-сайту для стартапів, який служить платформою для зв'язку стартапів з інвесторами. Веб-сайт розроблено з використанням Angular, популярного фреймворку веб-розробки, NgRx і Firebase.

Актуальність теми. Актуальність цього проєкту полягає у важливості фінансування для економіки стартапів та критичній ролі інвесторів у підтримці стартапів. Зважаючи на зростаючу потребу стартапів у фінансуванні, цей проєкт може допомогти стартапам отримати необхідні кошти, а інвесторам – визначити перспективні стартапи.

Об'єктом дослідження кваліфікаційної роботи є технології розроблення web-додатків.

Предметом дослідження є стартап платформа «KSU-STARTUPS».

Метою кваліфікаційної роботи є розроблення стартап платформи KSU-STARTUPS засобами фреймворку Angular та сервісу Firebase

Виходячи з поставленої мети, були визначенні наступні з**авдання дослідження:**

- Обґрунтування обраних інструментів розробки.
- Огляд додаткових бібліотек.
- Огляд можливостей Firebase.
- Розроблення стартап платформи «KSU-STARTUPS».
- Аналіз процесу розробки

Структура роботи складається з вступу, трьох розділів, висновку і списку літератури.

РОЗДІЛ 1
ОБҐРУНТУВАННЯ ОБРАНИХ ІНСТРУМЕНТІВ РОЗРОБКИ

На початку розробки було обрано Angular як фреймворк для front-end частини платформи, та Firebase замість бекенду. Angular та Firebase самі по собоі й так є неймовірними інструментами, але коли вони використовуються разом, то вони здатні розкрити весь потенціал, який був закладений в ці інструменти такою компанією-гігантом, як Google.

1.1 Чому Angular

Angular - популярний front-end фреймворк з відкритим вихідним кодом, розроблений Google. Він завоював популярність серед розробників завдяки своїй універсальності, надійності та відносній простоті використання. Можна виділити ще декілька причин, чому ангуляр є одним з кращих фреймворків на ринку front-end розробки:

1. Комплексна документація: Angular має одну з найповніших документів серед усіх front-end фреймворків. Документація містить посібники та приклади, які легко дотримуватися, що полегшує розробникам вивчення та впровадження Angular. Крім того, документація постійно оновлюється, гарантуючи, що розробники завжди мають доступ до найсвіжішої інформації. [1]
2. Dependency injection: dependency injection в Angular є потужним інструментом, який дозволяє розробникам вводити залежності в компоненти, служби та інші об'єкти. Ця функція полегшує керування залежностями програми та дозволяє краще розділяти усе на компоненти. [1]
3. Компонентна архітектура: компонентна архітектура Angular полегшує створення та управління великими додатками. Додаток розділено на більш дрібні компоненти, які можна легко

використовувати повторно, що полегшує обслуговування та оновлення програми. [1]

4. Кросплатформна розробка: Angular може використовуватися для розробки веб-додатків, які можуть працювати на різних платформах (наприклад комп’ютери та мобільні пристрої). Це полегшує розробникам створення додатків, які можуть охопити ширшу аудиторію, і підвищує зручність використання програми.
5. Потужні інструменти: Angular має потужний набір інструментів, що полегшує розробникам створення та тестування програм. Angular CLI, наприклад, є потужним інтерфейсом командного рядка, який полегшує створення, створення та тестування додатків Angular. [1]
6. Активна спільнота: Angular має велику та активну спільноту розробників, які завжди готові допомогти та поділитися своїми знаннями. Це означає, що розробники можуть легко знаходити рішення своїх проблем і отримувати допомогу, коли вона їм потрібна.

Однак, не зважаючи на усі ці плюси, Angular не дуже привітний до новачків, через кількість матеріалу, який треба вивчити та зрозуміти, щоб почати використовувати цей фреймворк на повну.

1.2 Чому Firebase

Firebase – це хмарна платформа, яка надає набір послуг для створення мобільних і веб-додатків, зокрема хостинг, бази даних у реальному часі, аутентифікацію тощо. Якщо коротко, то це Backend as a Service (BaaS), тобто сервіс, який надає усі ті самі можливості, що й звичайний бекенд, але, звичайно, сервіс буде менш гнучким. Саме тому Firebase можна модифікувати, створивши backend поверх сервісу. З плюсів використання Firebase можна виділити:

1. Проста інтеграція з Angular: Firebase має офіційну бібліотеку інтеграції Angular під назвою AngularFire, яка надає набір простих у використанні API для роботи зі службами Firebase у програмах Angular. Це дозволяє легко інтегрувати Firebase з додатком Angular. [3]
2. Хостинг: Хостинг Firebase забезпечує швидкий і безпечний спосіб розміщення будь якого додатку. Можна легко розгорнути свій додаток за допомогою однієї команди та скористатися автоматичними сертифікатами SSL, CDN тощо.
3. Аутентифікація: Firebase надає прості у використанні служби аутентифікації, які дають змогу користувачам авторизуватися за допомогою електронної пошти та пароля, Google, Facebook, Twitter та інших соціальних платформ. Це може заощадити багато часу і сил при побудові системи аутентифікації користувача.
4. Великий вибір баз даних: Firebase надає можливість користуватися широким вибором баз даних, щоб зберігати там різну інформацію та файли.

В результаті, комбінація з Angular та Firebase є неймовірно потужною та зручною в використанні для проєктів різних розмірів і саме тому була обрана саме вона.

РОЗДІЛ 2
ПІДГОТОВКА ДО РОЗРОБКИ KSU-STARTUPS

2.1 Огляд додаткових бібліотек

Окрім бібліотек, які йдуть разом з Angular, було вирішено встановити ще @angular/fire, ngrx/store, ngrx/effects та @ngrx/store-devtools.

– *@angular/fire*. Ця бібліотека необхідна для зручного користування Firebase. Вона обробляє запити всередині, щоб повернути відповідь в формі observable, або promise. Також, містить багато цікавих функцій, що спрощують розробку та допомагають покращити читабельність коду, щоб в майбутньому було легше його оновлювати та підтримувати. [4]

– *ngrx/store*. Ця бібліотека додає store, для зберігання даних додатку. Завдяки ній дуже зручно працювати з даними, які мають бути доступні в декількох місцях в додатку, адже для цього треба лише надіслати запит в стор, після чого опрацьовувати отриману відповідь. [5]

– *ngrx/effects*. Бібліотека, що додає можливість виконувати ще якісь дії, під час запиту до стору. Ці дії називають «Ефект» і, зазвичай, під час них проводять маніпуляції з даними, які надходять зі стору або до нього. [5]

– *@ngrx/store-devtools*. Це бібліотека, яка дає можливість маніпулювати стором з використанням інструментів розробника в браузері. Завдяки ній можна переглядати дані, які знаходяться в сторі, а також переглядати через які зміни вони проходять

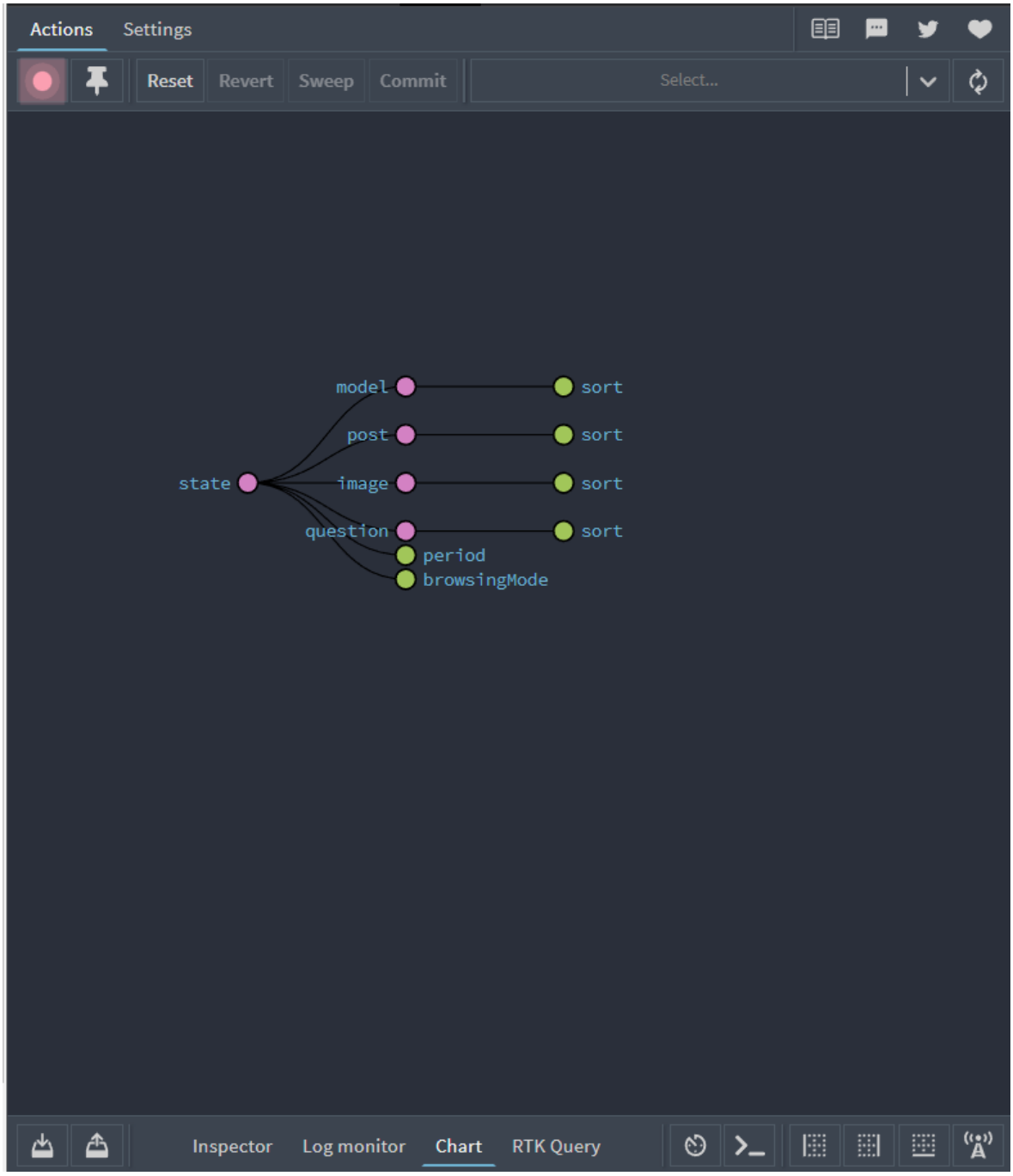


Рисунок 2.1 Приклад роботи @ngrx/dev-tools

2.2 Огляд можливостей Firebase

Firebase надає велику кількість можливостей, але в цьому проєкті використані лише деякі з них:

– *Авторизація*. Firebase дозволяє використовувати велику кількість сервісів для авторизації (Google, Facebook, Twitter тощо).

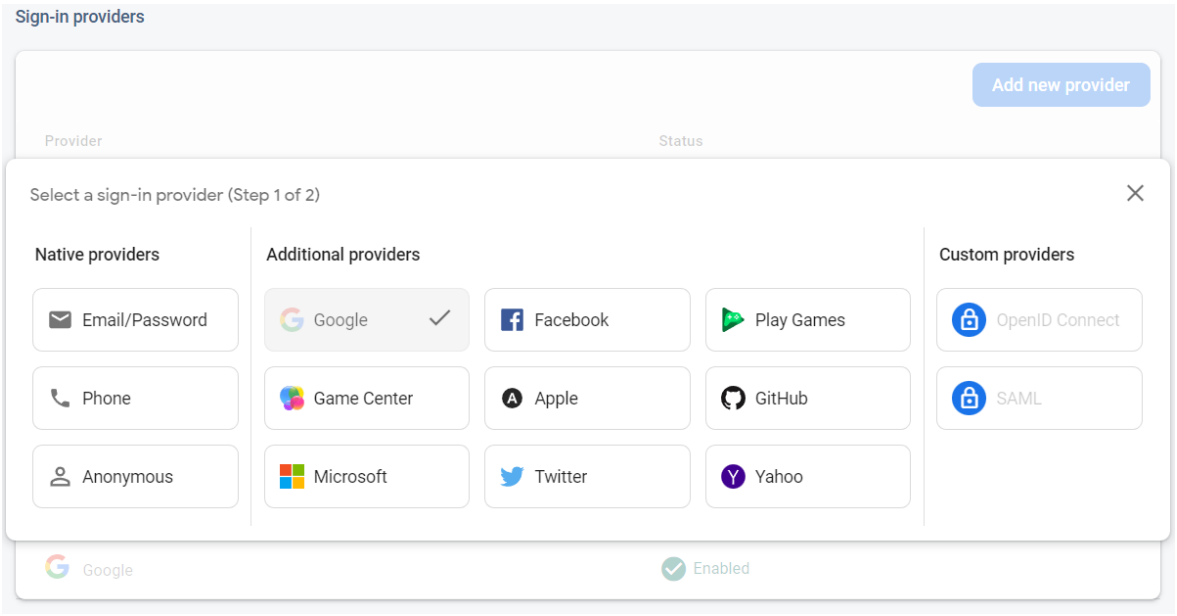


Рисунок 2.2 Демонстрація меню вибора провайдера авторизації

– *Storage*. Firebase дозволяє зберігати файли, для цього використовується Storage.

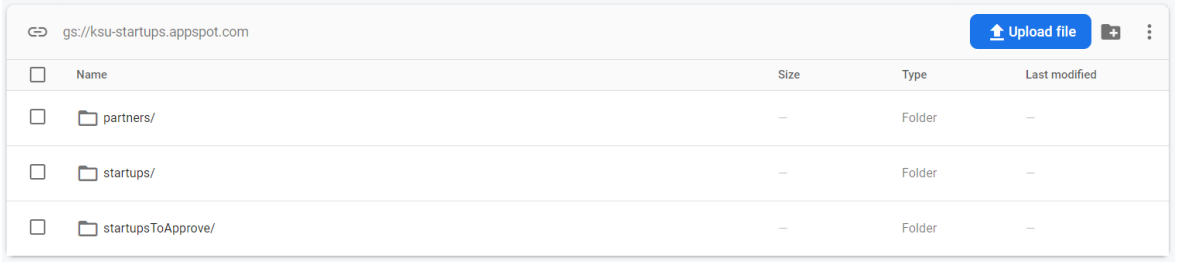


Рисунок 2.3 Демонстрація прикладу storage від Firebase

– *Firestore Database*. Дуже зручна та швидка база даних для зберігання різних типів інформації.

– *Хостинг*. Firebase надає можливість хостити додаток як на створених та зарезервованих сервісом доменах, так на власних. Для цього лише треба розробити додаток та прописати декілька команд в терміналі, після чого користувач вже може користуватися додатком.

Підсумовуючи можна сказати що Firebase є потужним та корисним інструментом для розробки програмного забезпечення

РОЗДІЛ 3
РОЗРОБЛЕННЯ ПЛАТФОРМИ ДЛЯ СТАРТАПІВ «KSU-STARTUPS»

3.1 Розроблення front-end складової додатку

Angular - популярний інтерфейсний фреймворк для створення динамічних і адаптивних веб-додатків. Він використовує компонентну архітектуру та декларативний синтаксис шаблону, що дозволяє легко створювати складні програми. Деякі ключові аспекти, про які слід пам'ятати при розробці з використанням Angular:

1. Компоненти: У Angular компонент є багаторазовим будівельним блоком, який інкапсулює функціональність та представлення частини інтерфейсу користувача. Компоненти є основою додатків Angular і можуть бути складені разом для створення складних користувальницьких інтерфейсів. Кожен компонент має шаблон, який визначає його структуру і клас, який визначає його поведінку. [1]
2. Модулі: В Angular модуль - це логічний контейнер для компонентів, сервісів, директив та іншого пов'язаного з цим коду. Модулі допомагають організувати додаток в дискретні функціональні області і сприяють можливості повторного використання. Додаток Angular може мати кілька модулів, і кожен модуль може імпортувати та експортувати код з інших модулів. [1]
3. Директиви: В Angular директива - це поведінка, яка може бути застосована до елемента HTML. Директиви можуть використовуватися для додавання або видалення елементів DOM, зміни поведінки існуючих елементів або прив'язки даних до представлення даних. Angular надає кілька вбудованих

директив, таких як ngIf і ngFor, а користувацькі директиви можуть бути створені за допомогою декоратора @Directive. [1]

4. Routing: Angular додаток може мати кілька сторінок, і роутинг використовується для навігації між ними. Angular забезпечує потужну систему маршрутизації, яка може обробляти складні навігаційні сценарії. Маршрути можуть бути визначені за допомогою модулів RouterModule і ActivatedRoute, а параметри можуть бути передані за допомогою сервісу ActivatedRoute. [1]
5. Forms: Angular надає надійний API для форм, який полегшує обробку введення та перевірки користувача. В Angular форми можуть бути шаблонними або реактивними, і обидва підходи пропонують переваги залежно від вимог програми. Модулі FormsModule та ReactiveFormsModule використовуються для обробки форм у форматі Angular. [1]
6. State management: У міру ускладнення програми управління станом програми стає все більш складним. Angular надає кілька інструментів для стейт менеджменту, наприклад бібліотеку NgRx, яка є моделлю state management, натхненною Redux. NgRx можна використовувати для управління станом програми, обробки побічних ефектів та забезпечення передбачуваного потоку даних. [1]

Angular - це потужний front-end фреймворк, який надає багатий набір інструментів для створення складних веб-додатків. Використовуючи компоненти, модулі, служби, директиви, маршрутизацію, форми та управління станом, розробники можуть створювати підтримувані, масштабовані та продуктивні програми.

3.1.1 Підготовка до початку розробки

Платформа «KSU-STARTUPS» розробляється для спрощення пошуку інвестицій стартапами і тому має містити наступний функціонал:

- Авторизація користувача;
- Можливість створення стартапу;
- Можливість переглядати інші стартапи ;
- Можливість переглянути список інвесторів;
- Можливість переглянути контакти адміністрації;

Дизайн додатку обирався та розроблявся з урахуванням новітнього бачення web-додатків, зручного інтерфейсу та функціоналу додатку.

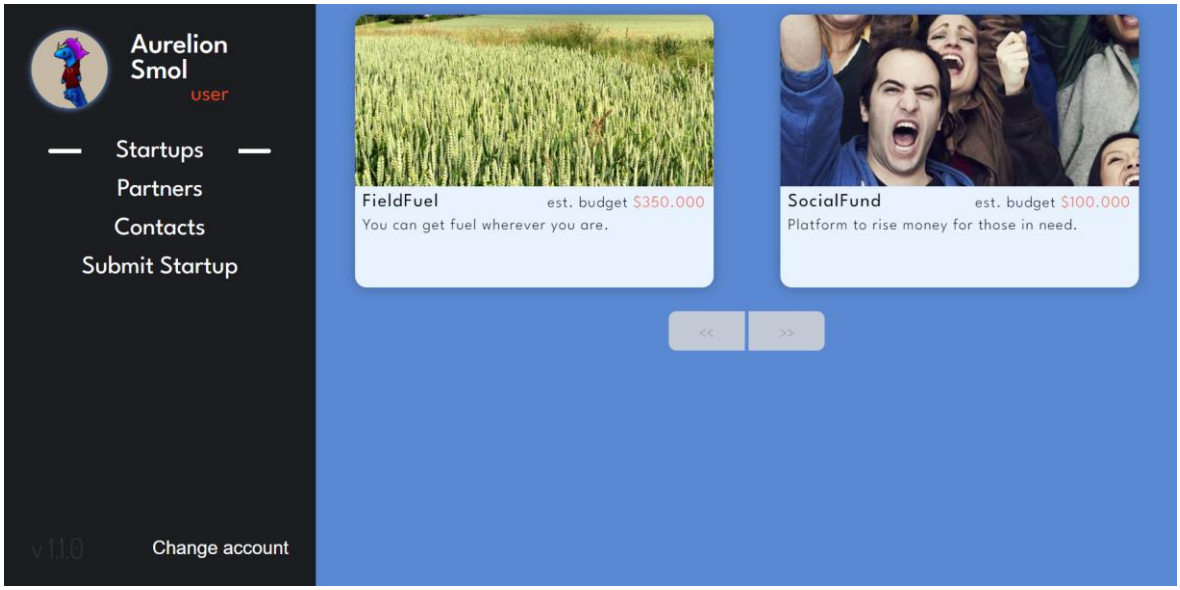


Рисунок 3.1 Головне вікно платформи

3.1.2 Файлова структура

Створення добре організованої структури папок має вирішальне значення для розробки підтримуваних і масштабованих додатків Angular. Добре структурована система папок допомагає розробникам організувати свій код і полегшити його розуміння, тестування та обслуговування.

Однією з часто використовуваних структур папок для додатків Angular є папка "app", яка містить всі компоненти, служби, директиви, труби та інші пов'язані файли програми. У папці "app" розробники можуть

створити кілька вкладених папок, щоб організувати свій код у пов'язані категорії. Нижче наведено кілька поширених вкладених папок, які можуть використовувати розробники. [7, 8]

1. Папка "components": Ця папка містить усі компоненти програми, які є будівельними блоками інтерфейсу користувача.
2. Папка "services": Ця папка містить всі служби програми, які надають функціональність компонентам і дозволяють їм обмінюватися даними із зовнішніми API та службами.
3. Папка "directives": Ця папка містить усі директиви програми, які можна використовувати повторно, і які можна застосувати до елементів DOM.
4. Папка "models": Ця папка містить усі моделі даних програми, які представляють дані, які програма використовуватиме та відображатиме.
5. Папка "guards": Ця папка містить всі функції для захисту програми, які використовуються для захисту певних маршрутів або компонентів від несанкціонованого доступу.
6. Папка "utils": Ця папка містить всі службові файли програми, такі як файли конфігурації, константи та допоміжні функції.

На додаток до папки "app", розробники також можуть створити папку "assets", яка містить статичні файли, такі як зображення, таблиці стилів та шрифти. Папку "environments" також можна використовувати для зберігання специфічних для середовища конфігураційних файлів, таких як середовища розробки та виробництва.

Створення добре організованої структури папок має вирішальне значення для розробки підтримуваних і масштабованих додатків Angular. Організовуючи свій код у пов'язані категорії, розробники можуть полегшити розуміння, тестування та підтримку своєї кодової бази. Описана вище структура папок "app" є широко використовуваним

підходом, але розробники повинні налаштувати її відповідно до своїх конкретних потреб.

Зважаючи на необхідний функціонал, а також на загальноприйняті умови написання коду, було вирішено зробити наступну файлову структуру для платформи:

– *pages*. Ця папка містить усі сторінки, які присутні на платформі.

– *shared*. Містить компоненти, моделі та сервіси, до яких необхідно мати доступ будь-де в коді.

– *state*. Папка, яка містить «стан» платформи. Тобто, данні, які зберігаються в сторі. В середині ще є папки, котрі розділяються, в залежності від того, які дані вони містять.

– *assets*. Папка, з картинками та іконками для веб додатку.

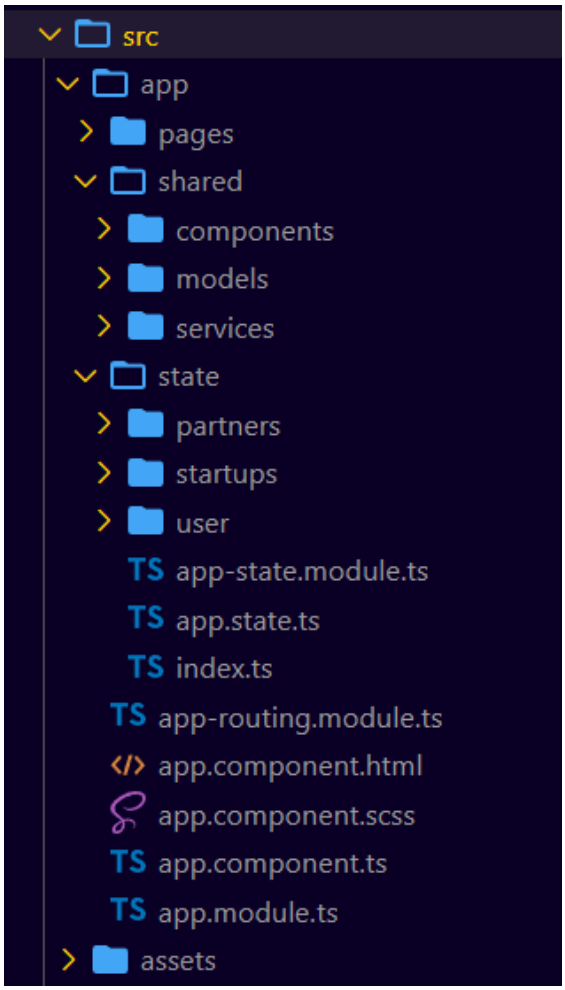


Рисунок 3.2 Файлова структура

3.1.3 Авторизація

Аутентифікація є найважливішим аспектом створення сучасних веб-додатків. Вона дозволяє користувачам безпечно отримувати доступ до своїх даних і запобігає несанкціонованому доступу до конфіденційної інформації. Авторизація за допомогою Google є популярним вибором серед розробників, оскільки вона забезпечує безперебійну роботу користувача та є безпечною.

Для використання сервісу авторизації від Firebase, треба ввімкнути Google, як провайдер авторизації у вже існуючому проєкті в консолі Firebase. Після чого, можна починати її використовувати, відправляючи запрос з використанням AngularFire, офіційної бібліотеки від Google, для проєктів з використанням цього фреймворка.

Рисунок 3.3 Сторінка авторизації

Якщо авторизація успішна, то ми зберігаємо користувача в store, після чого користувач може користуватися платформою.

3.1.4 Сторінка стартапів

Так як платформа «KSU-STARTUPS» повинна допомагати шукати інвестиції стартапам, то інвестори повинні мати можливість переглядати

стартапи. Для цього, нам необхіда сторінка зі стартапами. Щоб отримати дані з бази даних, треба відправити запрос на сервер і для цього знов в нагоді стає AngularFire (як і для будь якого запиту до Firebase).

Після того, як отримано відповідь з даними з серверу, треба їх зберегти до стору, щоб потім користувач міг просто переглянути вже завантажені стартапи і не робити новий запит на сервер. Кожного разу, коли користувач заходить на сторінку стартапів, ми дістаємо інформацію зі стору і просто відображаємо її.

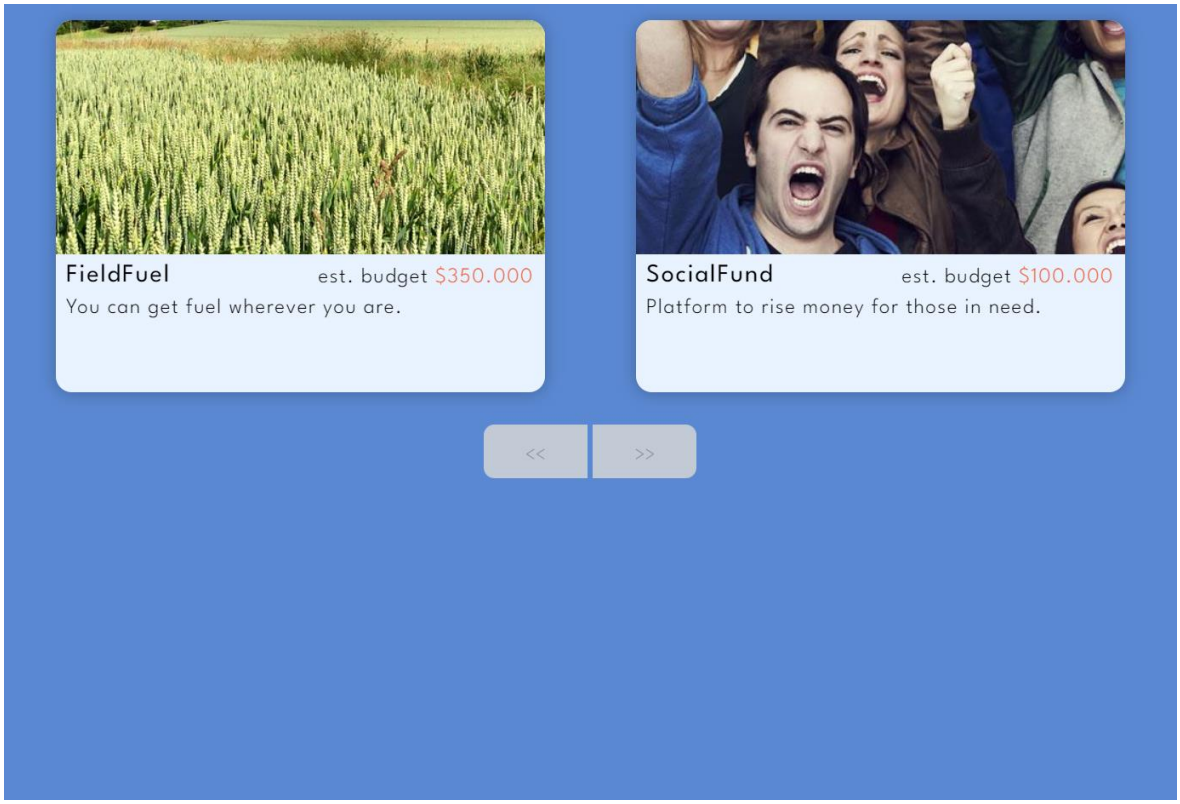


Рисунок 3.4 Сторінка стартапів

3.1.5 Сторінка створення стартапу

Якщо не можна буде створити стартап, то і відображати не буде чого, і сенс платформи зникне.

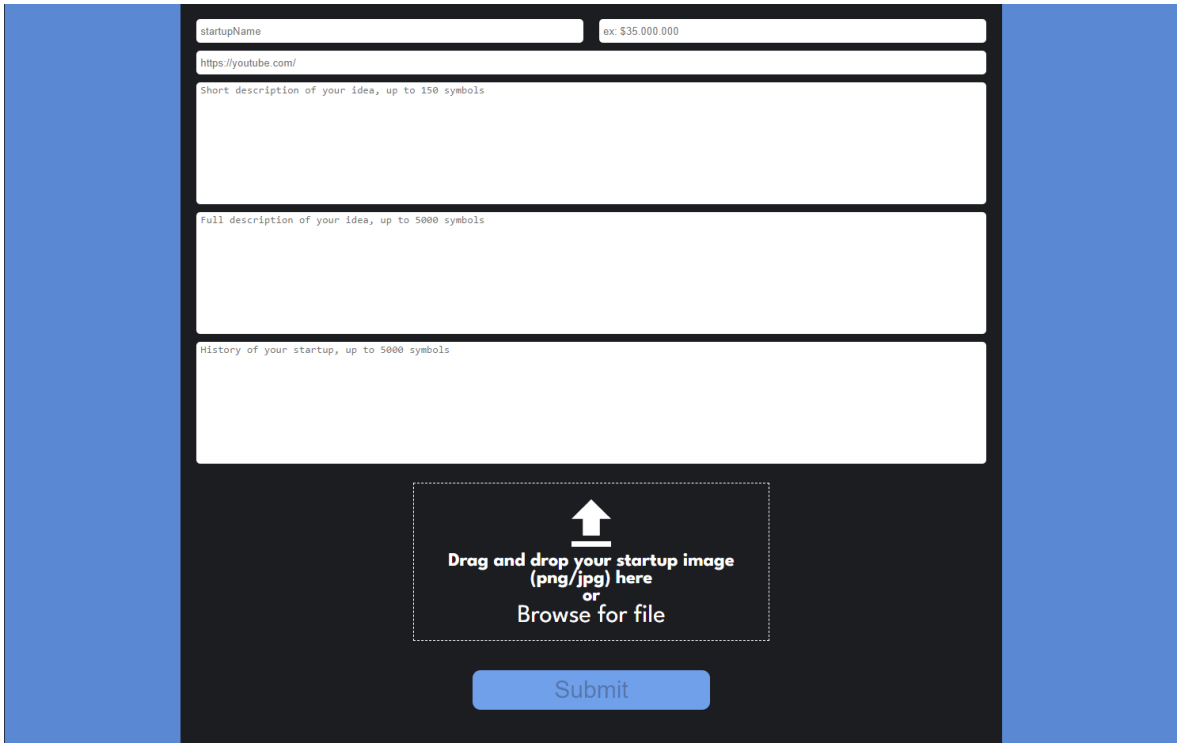


Рисунок 3.4 Сторінка додання стартапів

Автор стартапу заповнює усі поля та додає картинку, після чого спочатку на сервер завантажується картинка і лінк до неї додається до тексту, який заповнив автор. Після цього, дані відправляються до бази даних «на розгляд». Стартап не буде показуватися, доки людина, яка відповідає за це, не розгляне стартап і не перенесе його до бази даних з усіма підтвердженими стартапами.

3.2 Постановка на хостинг

Хостинг Firebase — це потужна та надійна платформа для швидкого та легкого розгортання веб-програм. За допомогою хостингу Firebase розробники можуть розгортати свої веб-програми лише за допомогою кількох простих команд і користуватися перевагами таких функцій, як автоматичний SSL, кешування CDN та безперервне розгортання.

Хостинг Firebase пропонує кілька функцій, які роблять його потужною та надійною платформою для розгортання веб-програм. Однією з таких функцій є автоматичний SSL, який шифрує дані під час

передавання та забезпечує безпеку даних користувача. Хостинг Firebase також забезпечує кешування CDN, що покращує продуктивність веб-додатків, обслуговуючи кешований вміст із найближчого до користувача сервера.

Ще однією потужною функцією хостингу Firebase є безперервне розгортання. Завдяки безперервному розгортанню розробники можуть автоматизувати процес розгортання та гарантувати, що їх веб-додаток завжди оновлюється. Firebase Hosting легко інтегрується з популярними інструментами безперервного розгортання, такими як GitHub Actions і Bitbucket Pipelines.

Firebase Hosting також надає розробникам можливість налаштувати доменне ім'я та сертифікат SSL веб-сайту. Розробники можуть використовувати власне доменне ім'я та сертифікат SSL, щоб забезпечити більш професійний та безпечний досвід для своїх користувачів.

3.2.1 Особливості комунікації з Firebase

Firebase – це популярна платформа backend-as-a-service (BaaS), яка надає розробникам набір інструментів і сервісів для створення веб- і мобільних додатків. Однією з головних переваг використання Firebase є те, що вона дозволяє легко реалізувати синхронізацію даних у реальному часі та зв'язок між клієнтом і сервером.

У Firebase зв'язок між клієнтом і сервером зазвичай досягається за допомогою бази даних реального часу Firebase або хмарного сховища Firestore. Це бази даних NoSQL, які зберігають дані в форматі JSON і забезпечують синхронізацію даних в режимі реального часу між клієнтами і серверами. Це означає, що будь-які зміни, внесені одним клієнтом, автоматично поширюються на всіх інших клієнтів у режимі реального часу, без необхідності будь-якого ручного втручання.

Щоб продемонструвати взаємодію між клієнтом і сервером з використанням Firebase, розглянемо приклад програми чату. У цьому додатку користувачі можуть відправляти і отримувати повідомлення в режимі реального часу. Ось як може виглядати потік зв'язку клієнт-сервер:

1. Клієнт надсилає повідомлення: коли користувач надсилає повідомлення, дані повідомлення надсилаються з клієнта до бази даних реального часу Firebase або Cloud Firestore, залежно від того, яку базу даних використовує програма.
2. Firebase зберігає повідомлення: Firebase зберігає дані повідомлень у базі даних і автоматично поширює зміни на всіх інших підключених клієнтів у режимі реального часу.
3. Інші клієнти отримують повідомлення: коли інші клієнти отримують повідомлення, вони оновлюють свій інтерфейс користувача, щоб відобразити нове повідомлення.
4. Клієнти чекають на зміни: клієнти чекають на зміни в базі даних за допомогою бази даних Firebase у реальному часі або API Cloud Firestore. Коли відбувається зміна, API сповіщає клієнта, а клієнт оновлює свій інтерфейс користувача, щоб відобразити зміни.

Окрім синхронізації даних у реальному часі, Firebase також надає низку інших функцій, які спрощують реалізацію зв'язку між клієнтом і сервером, таких як аутентифікація користувачів, хмарні функції та обмін повідомленнями в хмарі. Автентифікація користувачів дозволяє розробникам безпечно ідентифікувати та автентифікувати користувачів, тоді як хмарні функції дозволяють розробникам виконувати код на стороні сервера у відповідь на події бази даних. Хмарний обмін повідомленнями дозволяє розробникам відправляти push-повідомлення користувачам в режимі реального часу, навіть коли додаток не відкрито.

3.3 Аналіз процесу розробки

Angular – це фреймворк розробки, який дозволяє розробникам створювати складні, масштабовані та підтримувані веб-програми. Процес розробки з Angular можна розбити на кілька етапів, включаючи планування, проектування, впровадження, тестування та розгортання. Давайте розберемо кожен етап процесу розробки з Angular більш детально.

1. Планування: Першим етапом процесу розробки з Angular є планування. Це передбачає визначення вимог додатку, визначення обсягу проекту та створення дорожньої карти розвитку. Планування має вирішальне значення для успіху будь-якого проекту і може допомогти гарантувати, що проєкт буде закінчено вчасно, в рамках бюджету та відповідає потребам кінцевих користувачів.
2. Дизайн: Етап проектування процесу розробки з Angular передбачає створення інтерфейсу користувача програми. Це включає створення каркасів, прототипів та користувацьких потоків, які окреслюють структуру та функціональність програми. Angular надає багатий набір інструментів для проектування інтерфейсу користувача, включаючи компоненти, директиви та шаблони, які можна використовувати для створення модульного та масштабованого дизайну.
3. Реалізація: Етап реалізації процесу розробки з Angular передбачає написання коду, який оживляє дизайн. Це передбачає створення компонентів, сервісів і модулів, які визначають функціональність програми. Angular забезпечує потужне та інтуїтивно зрозуміле середовище розробки, яке включає мову на основі TypeScript, багатий набір вбудованих директив та потужний інструмент командного рядка, який можна використовувати для риштування програми.

4. Тестування: Етап тестування передбачає перевірку відповідності програми вимогам та функціям, як очікувалися. Це включає модульне тестування, інтеграційне тестування та наскрізне тестування, щоб переконатися, що програма не містить помилок і відповідає потребам кінцевих користувачів. Angular надає комплексну структуру тестування, яка включає такі інструменти, як Karma, які можна використовувати для автоматизації процесу тестування.
5. Розгортання: Завершальним етапом процесу розробки з Angular є розгортання. Це передбачає розгортання програми у виробничому середовищі та надання її доступу кінцевим користувачам. Angular надає потужний CLI, який можна використовувати для створення та розгортання програми на різних платформах, включаючи настільні, мобільні та веб-платформи.

ВИСНОВКИ

Поставлену мету – розробка платформи «KSU-STARTUPS» – досягнуто.

Оглянуто додаткові бібліотеки, необхідні під час розробки з обраним набором інструментів..

Результатом виконаної роботи є розроблена платформа «KSU-STARTUPS», яка відповідає наступним вимогам:

- Доступність.
- Швидкодія.
- Безпека даних.
- Легка масштабованість.

Розроблений додаток відповідає всім визначеним вимогам. При цьому був досягнутий компроміс між складністю та швидкістю додатку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Angular документація URL: https://angular.io/docs
2. Angular CLI документація URL: https://angular.io/cli
3. Firebase документація URL: https://firebase.google.com/docs/
4. AngularFire документація URL: https://github.com/angular/angularfire/tree/master/docs
5. NgRx документація URL: https://ngrx.io/docs
6. Git документація URL: https://git-scm.com/doc
7. Угода про стиль коду Angular URL: https://angular.io/guide/styleguide
8. Найкращі практики з Angular URL: https://medium.com/@d.gerbede/angular-best-practices-2022-8aba5910466
9. Visual Studio Code документація URL: https://ru.wikipedia.org/wiki/Visual_Studio_Code
10. GitHub документація URL: https://docs.github.com/en
11. TypeScript URL: https://www.typescriptlang.org/docs/home
12. Що таке API URL: https://en.wikipedia.org/wiki/API
13. RxJs документація URL: https://rxjs.dev/api
14. Простіша для розуміння на початку документація RxJs URL: https://www.learnrxjs.io
15. Найкращі практики з HTML URL: https://www.freecodecamp.org/news/html-best-practices/
16. Найкращі практики з CSS URL: https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Organizing
17. Найкращі практики з TypeScript URL: https://medium.com/@sobitdaniel/typescript-best-practices-610e8facb8df

18. Найкращі практики з TypeScript URL: https://medium.com/@sobitdaniel/typescript-best-practices-610e8facb8df
19. Найкращі практики з RxJs URL: https://betterprogramming.pub/rxjs-best-practices-7f559d811514
20. Швидкий старт з Firebase URL: https://fireship.io/lessons/firebase-quickstart/

Додатки

Додаток А

Кодекс академічної доброчесності здобувача вищої освіти Херсонського державного університету

КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ХЕРСОНСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ

Я, Щербина Володимир Володимирович,
учасник(ця) освітнього процесу Херсонського державного університету, **УСВІДОМЛЮЮ**, що академічна доброчесність – це фундаментальна етична цінність усієї академічної спільноти світу.

ЗАЯВЛЯЮ, що у своїй освітній і науковій діяльності **ЗОБОВ'ЯЗУЮСЯ**:
– дотримуватися:
- вимог законодавства України та внутрішніх нормативних документів університету, зокрема Статуту Університету;
- принципів та правил академічної доброчесності;
- нульової толерантності до академічного плагіату;
- моральних норм та правил етичної поведінки;
- толерантного ставлення до інших;
- дотримуватися високого рівня культури спілкування;

– надавати згоду на:
- безпосередню перевірку курсових, кваліфікаційних робіт тощо на ознаки наявності академічного плагіату за допомогою спеціалізованих програмних продуктів;
- оброблення, збереження й розміщення кваліфікаційних робіт у відкритому доступі в інституційному репозитарії;
- використання робіт для перевірки на ознаки наявності академічного плагіату в інших роботах виключно з метою виявлення можливих ознак академічного плагіату;

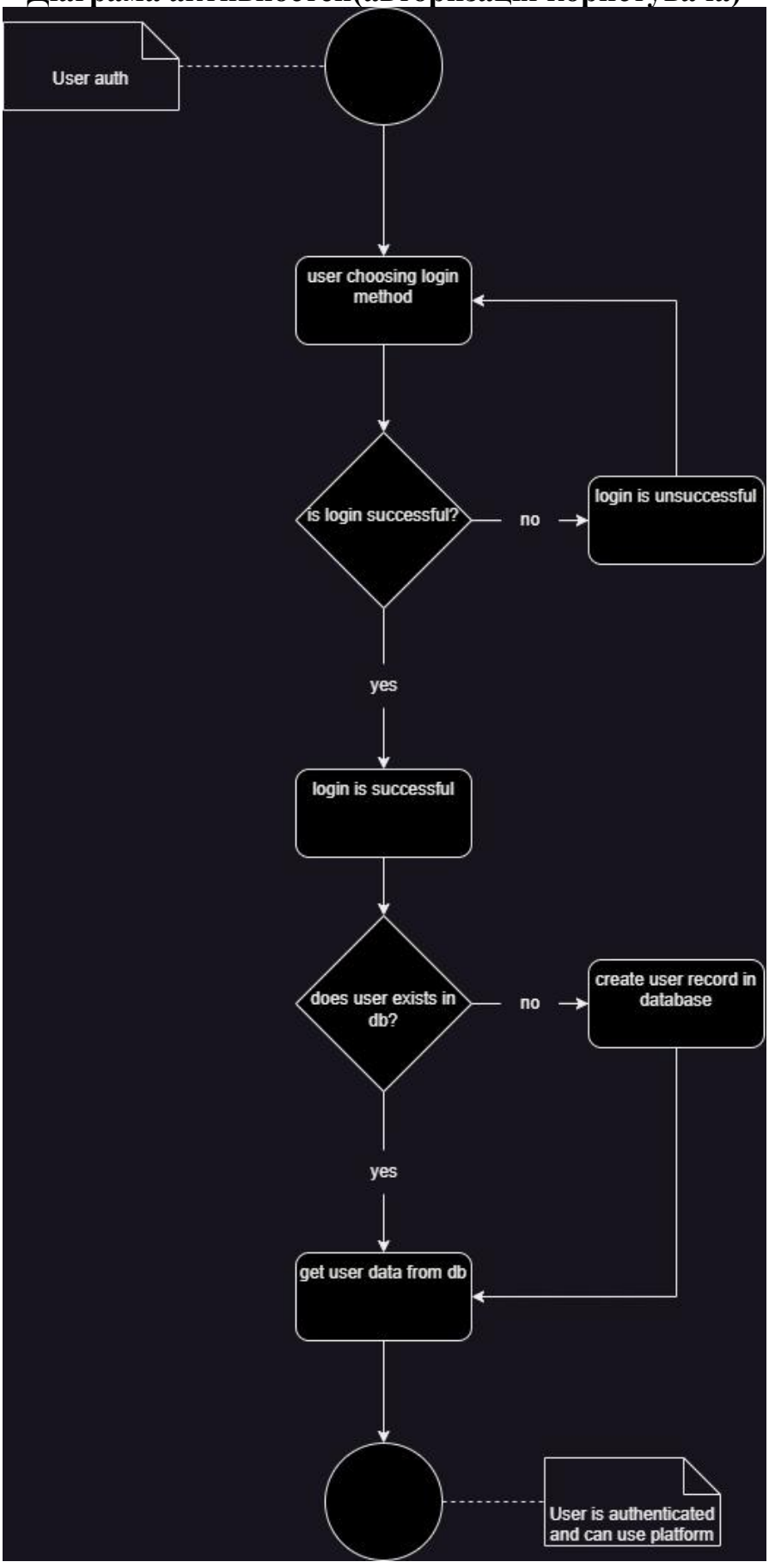
– самостійно виконувати навчальні завдання, завдання поточного й підсумкового контролю результатів навчання;
– надавати достовірну інформацію щодо результатів власної навчальної (наукової, творчої) діяльності, використаних методик досліджень та джерел інформації;
– не використовувати результати досліджень інших авторів без використання покликань на їхню роботу;
– своєю діяльністю сприяти збереженню та примноженню традицій університету, формуванню його позитивного іміджу;
– не чинити правопорушень і не сприяти їхньому скоєнню іншими особами;
– підтримувати атмосферу довіри, взаємної відповідальності та співпраці в освітньому середовищі;
– поважати честь, гідність та особисту недоторканність особи, незважаючи на її стать, вік, матеріальний стан, соціальне становище, расову належність, релігійні й політичні переконання;
– не дискримінувати людей на підставі академічного статусу, а також за національною, расовою, статевою чи іншою належністю;
– відповідально ставитися до своїх обов'язків, вчасно та сумлінно виконувати необхідні навчальні та науководослідницькі завдання;
– запобігати виникненню у своїй діяльності конфлікту інтересів, зокрема не використовувати службових і родинних зв'язків з метою отримання нечесної переваги в навчальній, науковій і трудовій діяльності;
– не брати участі в будь-якій діяльності, пов'язаній із обманом, нечесністю, списуванням, фабрикацією;
– не підроблювати документи;
– не поширювати неправдиву та компрометуючу інформацію про інших здобувачів вищої освіти, викладачів і співробітників;
– не отримувати і не пропонувати винагород за несправедливе отримання будь-яких переваг або здійснення впливу на зміну отриманої академічної оцінки;
– не залякувати й не проявляти агресії та насильства проти інших, сексуальні домагання;
– не завдавати шкоди матеріальним цінностям, матеріально-технічній базі університету та особистій власності інших студентів та/або працівників;
– не використовувати без дозволу ректорату (деканату) символіки університету в заходах, не пов'язаних з діяльністю університету;
– не здійснювати і не заохочувати будь-яких спроб, спрямованих на те, щоб за допомогою нечесних і негідних методів досягати власних корисних цілей;
– не завдавати загрози власному здоров'ю або безпеці іншим студентам та/або працівникам.

УСВІДОМЛЮЮ, що відповідно до чинного законодавства у разі недотримання Кодексу академічної доброчесності буду нести академічну та/або інші види відповідальності й до мене можуть бути застосовані заходи дисциплінарного характеру за порушення принципів академічної доброчесності.

12.09.2019		Володимир Щербина
(дата)	(підпис)	(ім'я, прізвище)

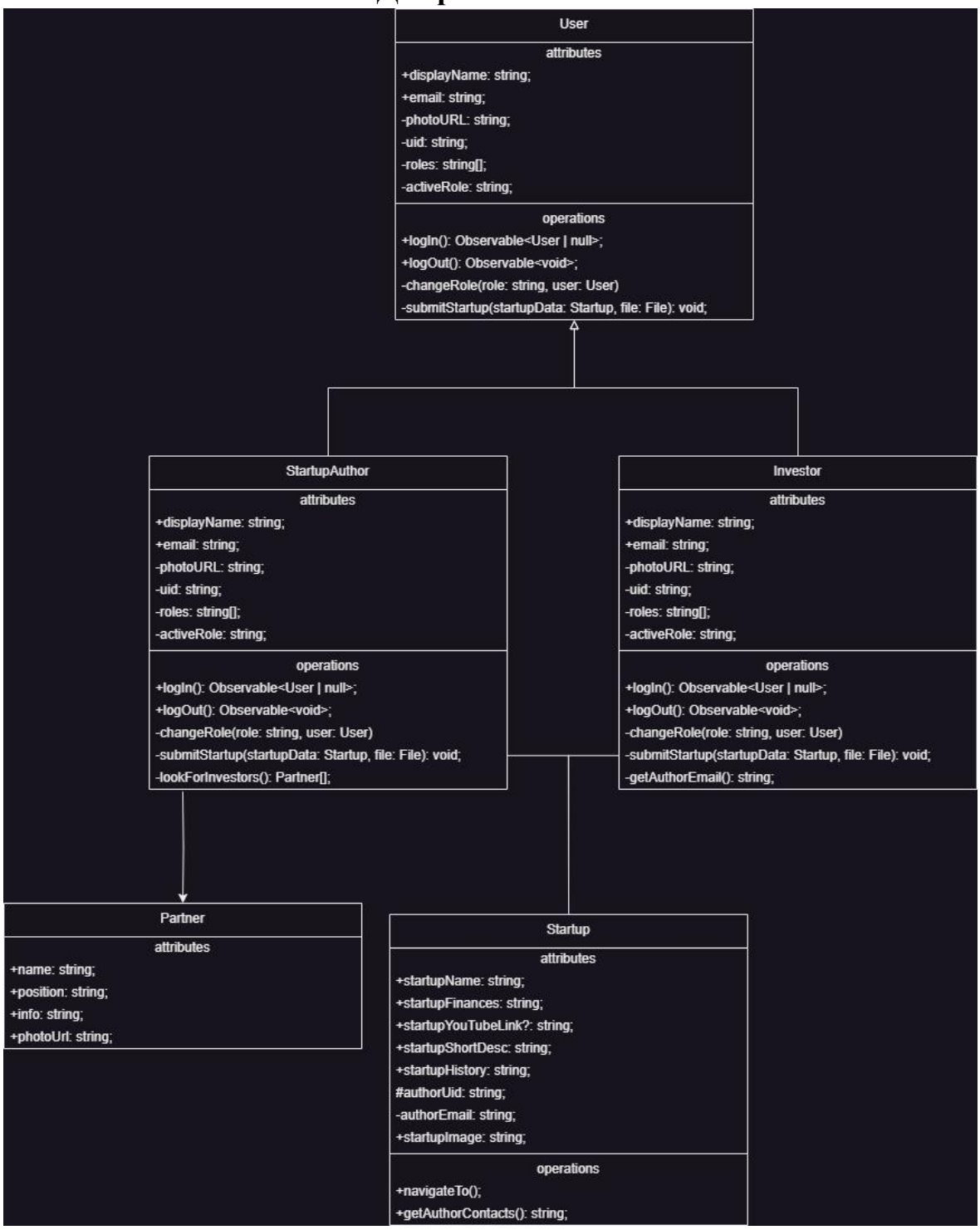
Додаток Б

Діаграма активностей(авторизація користувача)



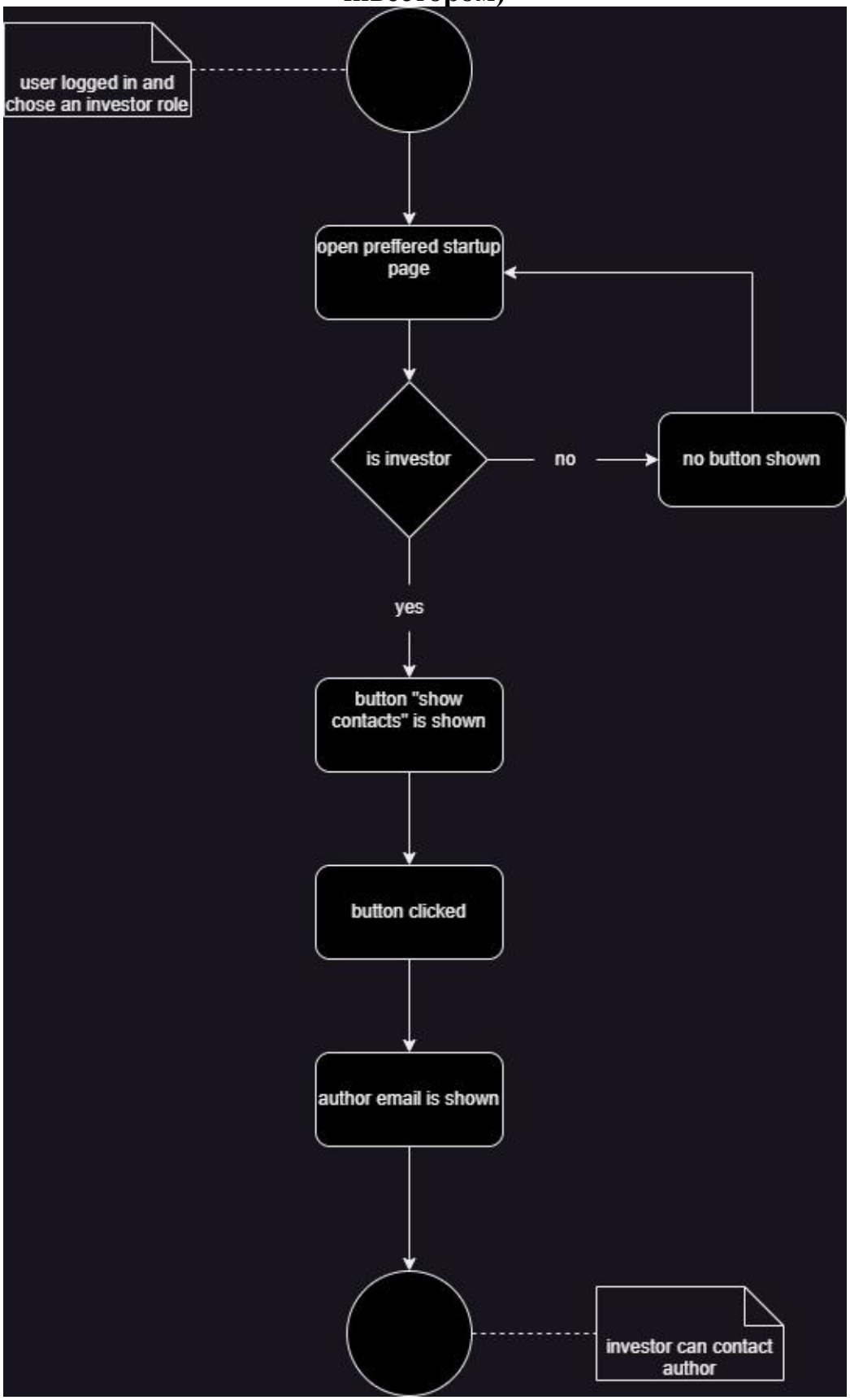
Додаток В

Діаграма класів



Додаток Г

Діаграма активностей(отримання контактів автору стартапу інвестором)



Додаток Д

Діаграма активностей(створення стартапу)