**Державний вищий навчальний заклад**

**Ужгородський національний університет**

**Факультет інформаційних технологій**

**ЛАБОРАТОРНА РОБОТА № 3**

**Тема:** Каркас (структура) проєкту.

Виконала студент(ка) І І І курсу

спеціальності «Комп’ютерні науки» (Інформатика)

Капустяк Анна Юріївна

**Ужгород-2024**

***Мета:*** представити структуру та стек проєкту.

***Завдання до роботи:***

1. Детальний опис ідеї з відповідними ілюстраціями (необхідно, щоб зрозуміти структуру папок/файлів)
2. Представлення структури проєкту, директорій...
3. Обгрунтувати обране рішення (по архітектурі)
4. Представити стек проєкту.
5. Детальне обгрунтування у випадку відхилення від запропонованої на заняттях архітектури.
6. Опублікувати всі зміни на GitHub
7. Створити PullRequest (нас додати як переглядачів (reviewers))
8. ВИСНОВКИ

**Хід роботи**

1. **Детальний опис ідеї з відповідними ілюстраціями (необхідно, щоб зрозуміти структуру папок/файлів)**

Ідея мого веб застосунку полягає у створенні користувачами власних профілів, де будуть розміщуватися їхні бажання.

Інші користувачі матимуть можливість знайти свого знайомого у застосунку, перейти на його профіль та переглянути його бажання. Сам власник профілю матиме можливість додавати власні бажання.

При створенні бажання потрібно вказати назву, за потреби додати короткий опис, фото та посилання на якийсь ресурс, де можна придбати даний подарунок.

1. **Представлення структури проєкту, директорій.**
2. Коренева папка presently

* Містить як клієнтську, так і серверну частини програми.

1. Клієнтська частина (presently-front)

* Папка node\_modules: Містить всі залежності для фронтенду.
* Папка public: містить статичні файли для веб-сайту, наприклад, файл index.html, який є основою для React-додатка.
* Папка src: Основна папка для вихідного коду.
  + components: містить компоненти інтерфейсу.
  + App.tsx: Основний компонент додатка, в якому збирається інтерфейс.
  + CreatingPage.tsx, HomePage.tsx, ProfilePage.tsx: Основні сторінки додатка.
  + index.tsx: Вхідна точка програми React, де зазвичай відбувається рендеринг кореневого компонента App.
  + index.css: Глобальні стилі.
  + logo.svg: Зображення логотипу.
  + Тестові файли (App.test.tsx, setupTests.ts): Використовуються для тестування.
  + Конфігураційні файли (tsconfig.json, react-app-env.d.ts).

1. Серверна частина (WishApiBack)

* Папка WishApiBack: Основна папка з кодом Django-додатка.
* manage.py: Скрипт для керування Django-проєктом.
* requirements.txt: Файл з переліком залежностей для серверної частини.
* db.sqlite3: SQLite база даних.
* .env: Файл для зберігання конфігурацій середовища.

1. **Обгрунтувати обране рішення (по архітектурі)**

Структура проєкту розділена на фронтенд і бекенд. Використання окремих директорій дозволяє легше керувати залежностями, організовувати вихідний код і розгортати проєкт. Відокремлення клієнтської частини на базі React (TypeScript) та серверної частини на базі Django забезпечує розподіл обов’язків між обробкою даних на бекенді і відображенням інтерфейсу на фронтенді, що спрощує підтримку та тестування.

**4.Представити стек проєкту.**

**Фронтенд**

* React: Фреймворк для створення інтерфейсів користувача, який дозволяє створювати компонентну архітектуру та реалізовувати односторінкові додатки (SPA).
* TypeScript: Надбудова над JavaScript, яка додає статичну типізацію. Це підвищує надійність коду та спрощує його підтримку, оскільки помилки виявляються ще на етапі розробки.
* CSS: Використовується для стилізації компонентів інтерфейсу.

**Бекенд**

* Django: Веб-фреймворк на Python, який надає засоби для швидкої розробки бекенду. Django має вбудовані можливості для керування базами даних, обробки запитів і аутентифікації користувачів.
* Django REST Framework (DRF) (імовірно, для API): Це розширення для Django, яке спрощує розробку RESTful API, що дозволяє легко інтегрувати фронтенд із бекендом.

**База даних**

* SQLite: Легка база даних, що зазвичай використовується для розробки та тестування. Може бути замінена на більш потужне рішення (наприклад, PostgreSQL або MySQL) у продакшн-оточенні.

**Інше**

* Node.js і npm: Використовується для керування залежностями фронтенду (пакет node\_modules) і для запуску інструментів розробки (наприклад, webpack або vite).
* .env файли: Використовуються для зберігання змінних середовища, які зберігають конфіденційні дані (наприклад, ключі API або конфігурації бази даних).
* Git: Система контролю версій для відстеження змін у коді.

**Пояснення вибору стеку**

* Такий стек вибраний для забезпечення швидкої та надійної розробки. React забезпечує гнучкість та модульність інтерфейсу, TypeScript знижує кількість помилок, а Django з його потужним ORM полегшує роботу з базами даних і підвищує безпеку додатка. Цей стек дозволяє ефективно розробляти веб-додатки з чітким розділенням на клієнтську та серверну частини, забезпечуючи гарну інтеграцію через REST API.

**Висновок:**

В ході даної лабораторної роботи було представлено структуру та стек проєкту.