

тестування, оскільки він надає зручний і простий інтерфейс, має багатий функціонал і підтримує високий рівень надійності.

У роботі були розглянуті різні види тестування, включаючи функціональне тестування, регресійне тестування, тестування взаємодії з користувачем та інші. Для організації тестових сценаріїв був використаний патерн PageObject, що дозволяє структурувати тестовий код та полегшує його підтримку та розширення.

Список використаних літературних джерел

- [1] TypeScript [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://www.typescriptlang.org/docs/> (дата звернення 25.09.2023). – Назва з екрана.
- [2] Mocha TypeScript [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://mochajs.org/> (дата звернення 25.09.2023). – Назва з екрана.
- [3] Cypress Dashboard TypeScript [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://docs.cypress.io/guides/dashboard> (дата звернення 25.09.2023). – Назва з екрана.
- [4] Cypress TypeScript [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://www.cypress.io/blog/> (дата звернення 25.09.2023). – Назва з екрана.
- [5] TypeScript TypeScript [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://devblogs.microsoft.com/typescript/> (дата звернення 25.09.2023). – Назва з екрана.
- [6] GitLab TypeScript [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://about.gitlab.com> (дата звернення 25.09.2023). – Назва з екрана.

Ростислав Завацький, Олександр Сторожук*, Наталія Гіссовська*

магістрант каф. КН НЛТУ України, Львів, Україна,
Zavatskyu.R@nltu.lviv.ua

* доцент каф. ІСКМ, НЛТУ України, Львів, Україна, <https://orcid.org/0000-0001-6566-5271>. storozhuk@nltu.lviv.ua

* асистент каф. ІСКМ, НЛТУ України, Львів, Україна.,
gissovaska.n@nltu.edu.ua.

Розроблення хмарного файлового менеджера з функцією синхронізації даних

Анотація - Робота присвячена розробці веборієнтованого застосунку, що дозволяє користувачам зберігати свої дані в хмарному сховищі та здійснювати вхід у свій акаунт та реєстрацію акаунту Cloudinary безпосередньо у розробленому застосунку. Робота написана з використанням бібліотеки React JS та за допомогою платформи NodeJS. Під час розроблення вебзастосунку було використано бібліотеку React JS, за допомогою якої будується користувацький інтерфейс та надсилаються запити на сервер, також було використано фреймворк ExpressJS, для спрощення створення серверу на NodeJS. В якості хмарного сховища, було використано Cloudinary.

Ключові слова - ReactJS, ExpressJS, NodeJS, Cloudinary.

Виклад основного матеріалу. Хмарні сховища мають багато переваг. Одна з переваг хмарних сховищ – це екологічність. Великі вузли зберігання інформації потребують багато електроенергії, але вони значно екологічніші за сотні або тисячі потужних настільних комп'ютерів. Також до переваг можна внести високий рівень надійності. Інформація в хмарі розміщується на різних серверах, завдяки цьому і зберігається високий рівень надійності. Але як і всі технології хмарні сховища мають і свої недоліки. Один з таких є конфіденційність. Інформація, яка розміщується в різних сховищах знаходиться у введенні провайдерів, тому недобросовісним особам простіше спричиняти її збій. Також “ахіллесова п'ята” даної технології шифрування. Недолік полягає в тому, що у випадку, якщо паролі раптом стануть доступними зловмисникам, ресурс виявиться дуже вразливим і не допоможуть навіть сучасні системи безпеки [1].

Незважаючи на недоліки дана технологія є дуже популярною, яка швидко розвивається, тому було прийнято рішення написати хмарний файловий менеджер, який буде зберігати дані в хмарному сховищі і спрощувати процес маніпуляцій з ними.

У застосунку було розроблено Back-end частину, в якій містяться основні запити такі як: запит на завантаження файлів, запити для отримання файлів, за-

пит на видалення файлів та було розроблено підключення локального сервера і підключення до хмарного сховища Cloudinary. Дане хмарне сховище служить для оптимізації та доставки медіафайлів та звичайних файлів, для вебсайтів та застосунків. Це хмарне сховище надає розширені можливості обробки медіа та зручне API [2]. Також була розроблена Front-end частина застосунку, в якій містяться основні компоненти FileList та UploadForm.

FileList – компонент, який містить в собі таблицю в якій користувач зможе переглянути інформацію про файл та буде мати можливість видалити, або завантажити його повторно на власний пристрій;

UploadForm – компонент, який містить в собі кнопку вибору файлу та кнопку завантаження файлу в хмарне сховище.

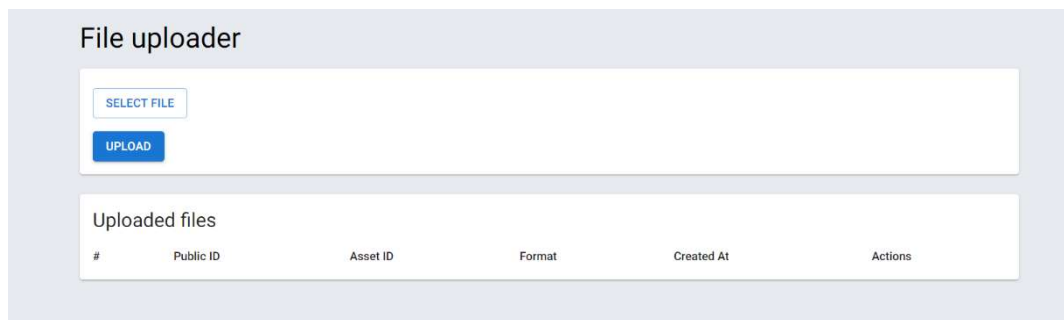


Рис. 1. Інтерфейс застосунку

Висновки. Розроблено хмарний файловий менеджер з функцією синхронізації даних. Розроблено бекенд частину застосунку для його запуску на локальному сервері та швидку взаємодію клієнта з хмарним сховищем за допомогою платформи Node JS та фреймворку Express JS. Організована обробка запитів без перезавантаження сторінки за допомогою “Single Page Application”. Розроблено зручний англomовний та адаптований під більшість екранів, користувачський інтерфейс із зрозумілою навігацією засобами React JS.

Список використаних літературних джерел

- [1] nic.ua [Електронний ресурс] : [Веб-сайт]. – Режим доступу до ресурсу: <https://info.nic.ua/uk/blog-uk/cloudstorage-2/> – доступний станом на 10.10.2023;
- [2] Cloudinary Official Site [Електронний ресурс] : [Веб-сайт]. – Режим доступу до ресурсу: <https://cloudinary.com/> – доступний станом на 10.10.2023;