

Список використаних літературних джерел

- [1] Роберт Мартін. Чиста Архітектура: Мистецтво створення програмного забезпечення. Видання друге / пер. з англ. І. Бондар-Терещенко. – Вид-во «Ранок»: Фабула, 2023. – 368 с.
- [2] Роберт Мартін. Чистий код: створення і рефакторинг за допомогою Agile. Видання друге / пер. з англ. І. Бондар-Терещенко. – Вид-во «Ранок»: Фабула, 2022. – 448 с.
- [3] Роберт Мартін. Чистий Agile: назад до основ/ пер. з англ. І. Бондар-Терещенко. – Вид-во «Ранок»: Фабула, 2021. – 416 с.

**Андрій Микитин, Любомир Флуд*, Андрій Нечепуренко*, Мар'ян
Опришко***

магістрант кафедри КН, НЛТУ України, Львів, Mykytyn.A@nltu.lviv.ua.

* к.т.н, доцент кафедри ІСКМ, НЛТУ України, Львів. <https://orcid.org/0000-0002-8347-4265>. flud@nltu.edu.ua.

* старший викладач кафедри ІПЗ, НЛТУ України, Львів.

<https://orcid.org/0000-0003-3955-1849> andriy.nechepurenko@gmail.com.

* асистент кафедри ІПЗ, НЛТУ України, Львів. <https://orcid.org/0009-0009-7981-9249> m.opryshko@nltu.edu.ua.

Розроблення застосунку "E-mail HUB"

Анотація – "E-mail Hub" - це застосунок, який був розроблений на базі технологій React та Node.js для полегшення та оптимізації роботи з електронною скринькою. За допомогою цього застосунку, який використовує протокол ІМАР, користувачі можуть підключити та працювати з багатьма електронними скриньками без зайвих переходів та авторизації, що значно заощаджує час та зручно організовує роботу з поштою. Застосунок також надає можливість спілкування в режимі он-лайн та включає в себе календар для зручного планування зустрічей та подій. "E-mail Hub" створений з метою полегшення та покращення ефективності роботи з електронною скринькою та ставить на перший план зручність користувача.

Ключові слова – E-mail Hub, React, Node.js, електронна скринька, протокол ІМАР, підключення, реєстрація, авторизація.

"E-mail Hub" - розроблений на базі технологій React та Node.js, розділений на три ключові компоненти: Backend, Frontend і базу даних. Ця архітектура представлена на схемі 1., де Backend використовує Node.js для обробки запитів, взаємодію з поштовими серверами через протокол IMAP, та зберігає дані в базі даних MongoDB. Frontend розроблений на React і забезпечує користувачам зручний інтерфейс для роботи з електронною поштою та іншими функціональностями.

Фронтенд застосунку "E-mail Hub" розроблений з використанням технології React. Ця частина додатку відповідає за інтерфейс, з яким користувач взаємодіє. На фронтенді користувач може реєструватися та авторизуватися в системі, вказуючи необхідні особисті дані. Після входу в систему, користувач має можливість підключити свої електронні поштові скриньки за допомогою IMAP (Internet Message Access Protocol). Після успішного підключення до поштових скриньок, він отримує доступ до свого поштового вмісту безпосередньо з фронтенду.

Тут для користувача надані засоби для перегляду, фільтрації, сортування та пошуку електронних листів.

У застосунку "E-mail Hub" реалізовано календар, де використовується бібліотека React Big Calendar, яка надає зручні інструменти для планування подій та відображення календаря. Ось невеликий фрагмент коду для прикладу:

```
// Код для використання React Big Calendar
const MyCalendar = (props) => (
  <Calendar
    localizer={localizer}
    events={props.events}
    startAccessor="start"
    endAccessor="end"
    style={{ height: 500 }}
  />
);
```

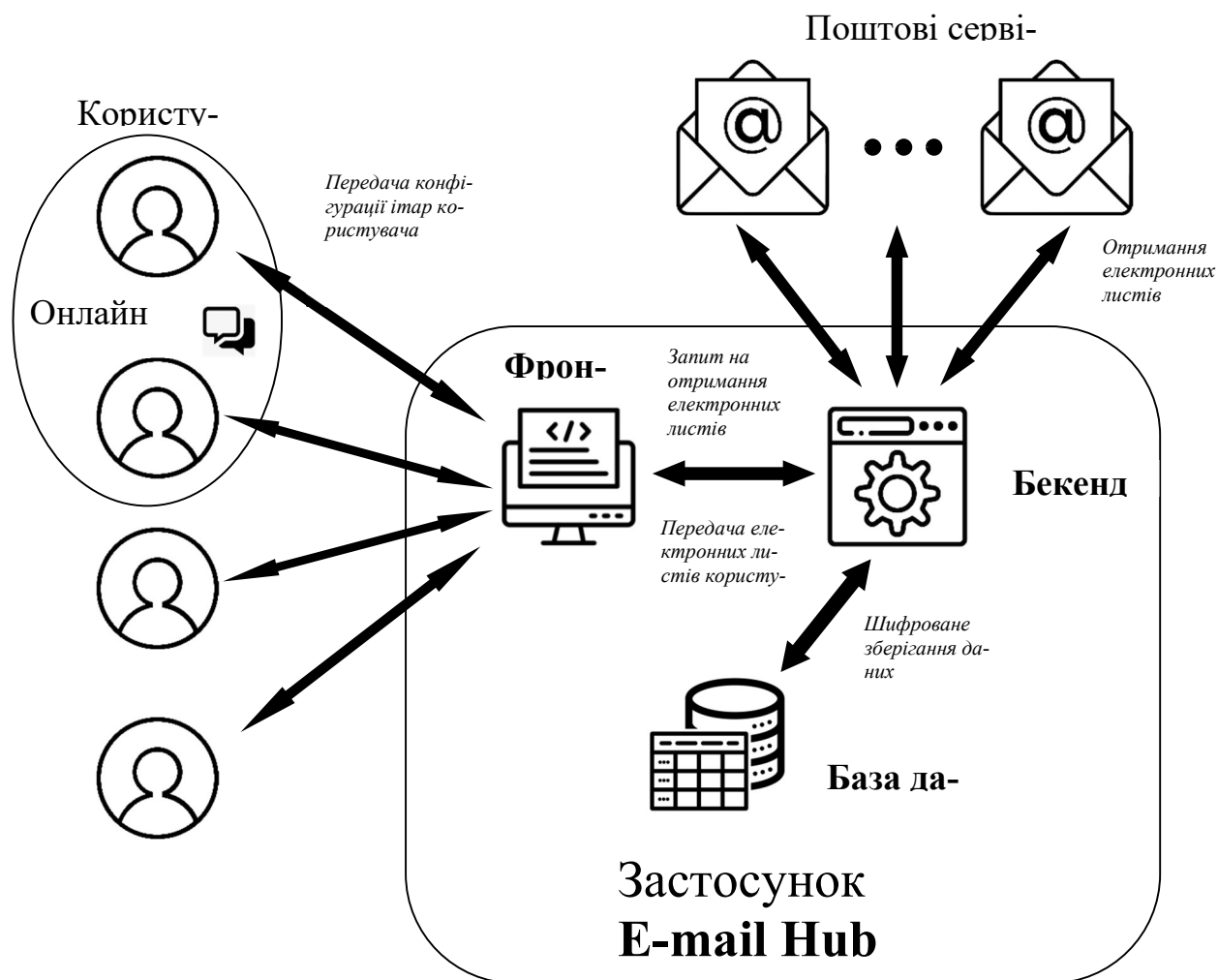


Рис 1. Робота застосунку «E-mail HUB»

За Backend нашого застосунку "E-mail Hub" відповідає серверна частина, яка використовує технології Node.js та MongoDB для забезпечення надійності та ефективності роботи. Основні функції та технічні особливості Backend можна розподілити на кілька ключових пунктів:

Автентифікація та авторизація: Backend відповідає за процес реєстрації та авторизації користувачів. Він перевіряє дані, які користувачі надають при реєстрації, та створює унікальний обліковий запис для кожного користувача. Також відбувається авторизація, яка гарантує доступ до конкретних функцій лише після введення вірних облікових даних.

Підключення до поштових сервісів: Backend з допомогою бібліотека node-imap надає можливість користувачам підключати свої поштові скриньки

(наприклад, Gmail, Yahoo, інші) за допомогою протоколу IMAP (Internet Message Access Protocol). Фрагмент коду:

```
// Налаштування для підключення до поштового серверу
const Imap = require('imap');
const imap = new Imap({
  user: 'your_email@gmail.com', // Ваша електронна адреса
  password: 'your_password', // Ваш пароль
  host: 'imap.gmail.com', // IMAP сервер для Gmail
  port: 993, // Порт IMAP сервера
  tls: true // Використовувати TLS});
imap.connect();// Підключення до поштового серверу
```

Чат: Backend забезпечує можливість спілкування в режимі онлайн через WebSocket, що дозволяє користувачам обмінюватися повідомленнями в реальному часі. Фрагмент коду:

```
// Код для налаштування WebSocket чату
const socket = new WebSocket('ws://chat-server-url');
socket.onopen = () => { console.log('З'єднання з WebSocket встановлено');};
socket.onmessage = (event) => {
  const message = JSON.parse(event.data);
  console.log('Отримано повідомлення:', message); // Обробка та відображення повідомлення в чаті
};
socket.onclose = () => {
  console.log('З'єднання з WebSocket закрито');
};
```

Безпека та шифрування: Backend з допомогою бібліотеку crypto забезпечує безпеку даних, особливо при роботі з поштовими скриньками. Він використовує шифрування для зберігання та передачі облікових даних користувачів, а також для забезпечення конфіденційності інформації. Фрагмент коду:

```
// Функція для шифрування даних
function encryptData(data, encryptionKey) {
  const cipher = crypto.createCipher('aes-256-cbc', encryptionKey);
```

```

    let encryptedData = cipher.update(data, 'utf8', 'hex');
    encryptedData += cipher.final('hex');
    return encryptedData;
}
// Функція для розшифрування даних
function decryptData(encryptedData, encryptionKey) {
    const decipher = crypto.createDecipher('aes-256-cbc', encryptionKey);
    let decryptedData = decipher.update(encryptedData, 'hex', 'utf8');
    decryptedData += decipher.final('utf8');
    return decryptedData;
}

```

Висновок. розроблено застосунок "E-mail Hub," з використанням технологій React та Node.js і надає користувачам можливість ефективно керувати та організовувати свою електронну пошту. Однією з ключових особливостей "E-mail Hub" є можливість підключення та керування багатьма електронними скриньками без зайвих переходів та авторизації, що значно спрощує роботу з поштою користувачів.

Застосунок надає можливість спілкування в режимі онлайн та включає в себе календар для зручного планування зустрічей та подій, роблячи його універсальним інструментом для роботи з електронною поштою та організації робочого часу. "E-mail Hub" є актуальним та має потенціал для застосування в різних галузях, де важлива ефективна робота з електронною поштою та плануванням робочого часу.

Список використаних літературних джерел

- [1] Документація React (<https://react.dev/>);
- [2] Документація Node.js (<https://nodejs.org/en/docs/>);
- [3] Документація MongoDB (<https://docs.mongodb.com/>);
- [4] Документація з бібліотеки node-imap (<https://github.com/mscdex/node-imap>);
- [5] Документація з розробки онлайн-чату на основі веб-сокетів (<https://socket.io/docs/v4/index.html>);
- [6] Документація з розробки календаря (<https://fullcalendar.io/docs>);

[7] Документація з розробки криптографічної безпеки (<https://nodejs.org/api/crypto.html>).

Р. І. Кордюк, В. К. Овсяк*

аспірант, Українська Академія Друкарства, Львів, Україна.
kordukroman@gmail.com.

*д.т.н., професор Української Академії Друкарства, Львів, Україна.
vovsyak@ukr.net.

Інформаційна технологія опрацювання графічних зображень. Модель колажів інформаційних технологій опрацювання графічних зображень

Анотація – Робота презентує розроблену інноваційну технологію для опрацювання графічних зображень та формування колажів, засновану на унікальній концепції дерева моделі колажів зображень. Проведено детальний аналіз та порівняння веб-редакторів [1, 2, 3], що висвітлює переваги використання Fabric JS [4], Angular [5] і TypeScript завдяки їхнім відмінним характеристикам та можливостям. Концепція моделі дерева демонструє, що кожен її елемент є незалежним, але при цьому може взаємодіяти в складній структурі, відкриваючи нові перспективи для розвитку та інтеграції системи. Представлена технологія може стати значущим інструментом для різноманітних інтернет-платформ у галузі графічного дизайну та обробки зображень.

Ключові слова – Інформаційні технології, Графічні Зображення, Колажі, Модель Дерева, Fabric JS, Angular, TypeScript, Опрацювання Зображень, Он-лайн Платформи.

Розробка інноваційної технології обробки графічних зображень та створення колажів є складним завданням, що вимагає глибокого аналізу та моделювання процесів взаємодії та інтеграції різноманітних графічних елементів. У цьому контексті ключовими аспектами стають визначення оптимальних параметрів для обробки графічних зображень та розробка ефективних алгоритмів для створення колажів.

Метою даної роботи є розробка та впровадження ефективних методів та технологій для обробки графічних зображень та створення колажів. Це плану-