

3. Десять перспективніших напрямів розвитку фінтеху в Україні. *FintechInsider*. 24.08.2024. URL: <https://fintechinsider.com.ua/desyat-najperspektyvnishyh-napryamiv-rozvytku-fintehu-v-ukrayini/>

УДК 004.42

*Шманов Я. К., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук, доцент,
доцент кафедри інформаційних технологій*

ВЕБДОДАТОК ДЛЯ СПІЛЬНОЇ КОМУНІКАЦІЇ РОЗРОБНИКІВ У МЕЖАХ ІТ-ПРОЄКТУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Сучасні ІТ-компанії працюють у швидкоплинному середовищі, де ефективна комунікація є ключовим складником успіху проєктів. Відсутність централізованої платформи для обміну повідомленнями, управління завданнями та інтеграції з іншими інструментами ускладнює командну роботу. Актуальність такої платформи обумовлена:

- необхідністю ефективних інструментів для комунікації та координації в сучасних ІТ-командах;
- низкою проблем, що виникають без централізованої платформи: втрата інформації, неузгодженість дій, складнощі у відстеженні прогресу;
- потребою в рішеннях для організації спільної роботи розробників.

Основними функціями цього вебдодатка мають стати: управління проєктами, візуалізація робочого процесу, управління завданнями, автентифікація та безпека.

У межах управління проєктами реалізації підлягають задачі: створення, редагування та перегляд проєктів; можливість додавання та видалення учасників проєкту.

У межах візуалізації робочого процесу реалізуються задачі: створення та управління колонками (статусами) завдань; переміщення завдань між колонками (Drag-and-Drop функціонал за допомогою react-beautiful-dnd).

Управління завданнями передбачає функції: створення, редагування, видалення завдань; призначення виконавців (учасників проєкту) на конкретні завдання; додавання описів, міток / маркерів до завдань для кращої деталізації.

Функції автентифікації та безпеки мають враховувати: реєстрацію та вхід користувачів у систему; використання JWT (JSON Web Tokens) та Passport.js для безпечної автентифікації та авторизації запитів; хешування паролів (bcryptjs) для безпечного зберігання; керування сесіями користувачів (зберігання токена та перевірка його актуальності) [1].

Технологічні рішення під час розробки цього програмного продукту повинні враховувати особливості архітектури та технологічний стек. Такий додаток має бути побудованим за принципами архітектури клієнт-серверної моделі, яка перед-

бачає чітке розділення на Frontend (інтерфейс користувача) та Backend (серверна логіка, робота з даними). Реалізація інтерфейсу користувача використовує технології: React, Redux (для управління станом), React Router (для навігації), використання Axios для запитів до Backend [2]. Реалізація серверної частини використовує технології: Node.js, Express.js. Також необхідно враховувати створення бази даних (PostgreSQL з використанням Sequelize ORM для взаємодії) та API (RESTful підхід для обміну даними між клієнтом та сервером) [3].

Перевагами розробки цього вебдодатка є:

- централізація інформації;
- покращена прозорість робочих процесів;
- спрощення комунікації та координації;
- ефективне управління завданнями та ресурсами.

Отже, розроблений вебдодаток сприятиме покращенню організації роботи команд розробників, підвищенню продуктивності та спрощенню процесів управління проектами. Інтеграція основних інструментів у єдину платформу дає змогу значно зменшити інформаційне навантаження та підвищити ефективність комунікації.

Список використаних джерел

1. Шаблони проектування. Повторне використання об'єктно-орієнтованого програмного забезпечення / Е. Гамма, Р. Гельм, Р. Джонсон, Дж. Влісідес. Київ: Діалектика, 2020. 448 с.
2. Head First Design Patterns / Е. Freeman, Е. Robson, В. Bates, К. Sierra. O'Reilly Media, 2020. 694 p.
3. Flanagan D. JavaScript: The Definitive Guide. 7th Edition. O'Reilly Media, 2020. 706 p.
4. Banks A., Porcello E. Learning React: Modern Patterns for Developing React Apps. O'Reilly Media, 2020. 350 p.