



**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ПОЛТАВСЬКА ПОЛІТЕХНІКА  
ІМЕНІ ЮРІЯ КОНДРАТЮКА**

**ЗБІРНИК МАТЕРІАЛІВ**

**77-ї НАУКОВОЇ КОНФЕРЕНЦІЇ ПРОФЕСОРІВ,  
ВИКЛАДАЧІВ, НАУКОВИХ ПРАЦІВНИКІВ,  
АСПІРАНТІВ ТА СТУДЕНТІВ УНІВЕРСИТЕТУ**

**16 травня – 22 травня 2025 р.**

*А.В. Курнаков, студент групи 403-ТН,  
І.Ю. Закаблук, студент групи 403-ТН,  
Ю.М. Здоренко, к.т.н.*

*Національний університет  
«Полтавська політехніка імені Юрія Кондратюка»*

## **РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ УПРАВЛІННЯ ЗАДАЧАМИ КОРИСТУВАЧІВ НА ОСНОВІ ПЛАТФОРМИ NODE.JS**

У сучасному цифровому середовищі ефективне управління завданнями є ключовим елементом успішної діяльності не лише в корпоративному секторі, а й у повсякденному житті. Зі зростанням популярності дистанційної роботи, гнучких графіків та динамічних командних проєктів виникає потреба у зручних, масштабованих та безпечних інструментах, які дозволяють організувати, контролювати та аналізувати задачі. Веб-додатки на основі платформи Node.js [1], [2] є одним із найперспективніших рішень для реалізації подібних систем.

Перший етап дослідження був присвячений аналізу існуючих інструментів управління задачами та виявленню їх сильних і слабких сторін. Було вивчено такі платформи, як Trello, Jira, Notion, які поєднують простоту візуального управління із складною логікою налаштування робочих процесів. Однією з основних тенденцій розвитку таких систем є автоматизація рутинних процесів, інтеграція з іншими сервісами та забезпечення спільної роботи в реальному часі.

Node.js, як асинхронне середовище виконання програмного коду на мові JavaScript, виявилось оптимальним вибором для серверної частини системи, завдяки своїй продуктивності, масштабованості та розвинутій екосистемі наявних програмних модулів. Його подієво-орієнтована модель дозволяє ефективно обслуговувати численні одночасні запити, що особливо важливо для динамічних систем керування завданнями з активною взаємодією користувачів.

На другому етапі було визначено архітектуру веб-додатку. Пропонується реалізацію системи здійснювати за клієнт-серверною моделлю з використанням Node.js та модулю Express для створення RESTful API. У якості системи керування баз даних було обрано MongoDB [3], яка забезпечує гнучке зберігання неструктурованих даних і добре підходить для реалізації складних об'єктів, таких як задачі, теги, підзадачі, коментарі та історія змін. Передбачена також можливість розширення функціоналу за рахунок використання WebSocket для реалізації миттєвого обміну даними між клієнтами.

Для забезпечення безпеки реалізовано механізм аутентифікації на основі JWT, а також систему прав доступу до окремих ресурсів залежно від ролі користувача (адміністратор, виконавець, гість тощо). Було розроблено зручний інтерфейс адміністратора для моніторингу активності, аналітики завдань і керування проєктами.

На етапі реалізації особлива увага приділялася розробці зручного та інтуїтивного інтерфейсу для користувача. Заплановано використання React або аналогічного фреймворку для побудови адаптивної клієнтської частини з підтримкою функцій drag-and-drop, фільтрами, push-сповіщеннями та календарною прив'язкою дедлайнів. Завдяки цьому користувачі можуть легко створювати нові завдання, призначати відповідальних осіб, відстежувати виконання та вести спільне планування.

Важливою особливістю системи є можливість інтеграції з зовнішніми сервісами, такими як Google Calendar, Telegram або Slack. Це дозволяє розширити функціональність додатку без необхідності дублювати вже наявні інструменти в екосистемі користувача.

Завершальним етапом стало дослідження перспектив впровадження системи у різних галузях: освіті, бізнесі, розробці ПЗ, а також у приватному використанні. За умови адаптації функціоналу система може використовуватись як навчальний щоденник, проєктний менеджер, CRM-інструмент чи трекер персональних завдань. Розробка подібного веб-додатку є актуальною відповіддю на виклики цифрової трансформації, що вимагає простих і ефективних інструментів для організації праці.

Таким чином, розроблений веб-додаток для управління задачами на базі Node.js демонструє високу гнучкість, масштабованість і відповідність сучасним вимогам до інформаційних систем. Проєкт поєднує технічну ефективність з практичною користю, відкриваючи нові можливості для управління задачами у різних процесах. Подальший розвиток системи може включати впровадження елементів штучного інтелекту, аналітики продуктивності, а також створення мобільних клієнтів, що лише посилить її конкурентоспроможність у цифрову епоху.

#### *Література:*

1. *Node.js Documentation*. URL: <https://nodejs.org/en/docs> (дата звернення – 20.04.2024 р.)
2. *Express – Fast, unopinionated, minimalist web framework for Node.js*. URL: <https://expressjs.com> (дата звернення – 20.04.2024 р.)
3. *MongoDB Manual*. URL: <https://www.mongodb.com/docs/manual> (дата звернення – 20.04.2024 р.)