

## РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ РЕСУРСАМИ АЕРОПОРТУ

Державне некомерційне підприємство "Державний університет "Київський авіаційний інститут"

### **Анотація.**

*У статті розглянуто процес розробки системи управління ресурсами аеропорту (Airport Resource Management System). Описано архітектуру системи, основні функціональні можливості та використані технології. Система реалізована як веб-додаток із використанням Nest.js для серверної частини та Next.js для клієнтської частини. Проведено аналіз ефективності застосування обраних технологій для забезпечення високої продуктивності та масштабованості.*

**Ключові слова:** система управління ресурсами, аеропорт, веб-додаток, Nest.js, Next.js, REST API, база даних, авіація.

### **Abstract.**

*The article discusses the development process of the Airport Resource Management System. The architecture, main functionalities, and technologies used are described. The system is implemented as a web application using Nest.js for the backend and Next.js for the frontend. The efficiency of the selected technologies in ensuring high performance and scalability is analyzed.*

**Keywords:** resource management system, airport, web application, Nest.js, Next.js, REST API, database, aviation.

### **Вступ**

Аеропорти є складними інфраструктурними об'єктами, що включають велику кількість ресурсів, таких як повітряні судна, транспортні засоби, технічне обладнання, склади, персонал тощо. Для ефективного управління цими ресурсами необхідні сучасні інформаційні системи, які забезпечують швидкий доступ до даних, їх актуалізацію та контроль використання. Відсутність автоматизованих рішень може призвести до неефективного розподілу ресурсів, затримок у роботі аеропорту та зростання операційних витрат[1].

На сьогоднішній день існує низка інформаційних систем для управління ресурсами, проте вони можуть бути надто загальними або не відповідати специфічним вимогам аеропортів. У зв'язку з цим актуальним є розробка спеціалізованої системи управління ресурсами аеропорту, яка дозволить автоматизувати процеси адміністрування, моніторингу та аналізу використання ресурсів[2].

Метою даної роботи є розробка та впровадження ефективної веб-орієнтованої системи управління ресурсами аеропорту, що забезпечить зручний доступ до інформації про ресурси, їх оптимальний розподіл та підвищення загальної ефективності роботи аеропорту. Для досягнення цієї мети використовуються сучасні технології веб-розробки, зокрема фреймворки Nest.js для серверної частини та Next.js для клієнтської частини.

Основними завданнями даної роботи є:

1. аналіз існуючих рішень для управління ресурсами аеропорту;
2. визначення основних вимог до системи;
3. розробка архітектури веб-додатка;
4. реалізація функціональних можливостей для управління ресурсами;
5. тестування та оцінка ефективності розробленої системи.

Об'єктом дослідження є процеси управління ресурсами в аеропортах, а предметом дослідження – методи та інструменти автоматизації цих процесів за допомогою веб-технологій.

Практичне значення роботи полягає у можливості впровадження запропонованої системи в аеропортах для покращення координації ресурсів, зменшення операційних витрат і підвищення загальної продуктивності роботи аеропортової інфраструктури.

## Архітектурна система

Розроблена система управління ресурсами аеропорту базується на клієнт-серверній архітектурі, що забезпечує масштабованість, продуктивність і зручність у використанні. Основними компонентами системи є серверна частина (back-end), клієнтська частина (front-end) та база даних для зберігання інформації.

Серверна частина (back-end) реалізована за допомогою фреймворку Nest.js, що дозволяє створювати ефективні та масштабовані REST API. Використання мови програмування TypeScript підвищує надійність коду та полегшує підтримку проєкту. Основні функції сервера:

1. Обробка запитів від клієнтської частини.
2. Аутентифікація та авторизація користувачів за допомогою JSON Web Token та OAuth.
3. Взаємодія з базою даних MongoDB через ODM Mongoose.
4. Валідація даних і забезпечення безпеки запитів.

Клієнтська частина (front-end) побудована за допомогою фреймворку Next.js, що забезпечує високу продуктивність та покращену взаємодію користувача із системою[3]. Інтерфейс розроблений із використанням бібліотеки компонентів MUI та Tailwind CSS для забезпечення адаптивності та зручності у використанні. Основні можливості клієнтської частини:

1. Відображення даних про ресурси аеропорту у зручному форматі.
2. Фільтрація та пошук ресурсів.
3. Механізми для додавання, редагування та видалення даних.
4. Динамічне завантаження сторінок для покращення продуктивності.

Система використовує базу даних MongoDB для зберігання даних про ресурси аеропорту. Основні переваги цієї бази даних:

1. Гнучка схема, що дозволяє легко оновлювати структуру даних.
2. Висока продуктивність при роботі з великими обсягами інформації.
3. Масштабованість для підтримки зростаючих обсягів даних[4].

Для забезпечення ефективної взаємодії між компонентами системи використовується архітектура REST API, що надає стандартизований інтерфейс для обміну даними. Таким чином, система управління ресурсами аеропорту поєднує сучасні технології та підходи для створення надійного та продуктивного програмного забезпечення.

## Основні функціональні можливості

Розроблена система управління ресурсами аеропорту забезпечує низку функціональних можливостей, спрямованих на оптимізацію процесів адміністрування та ефективне використання ресурсів. Основними функціями системи є:

1. Аутентифікація та авторизація користувачів
  - 1.1. Використання JSON Web Token та OAuth (Google та GitHub) для безпечної автентифікації.
  - 1.2. Розмежування прав доступу відповідно до ролей користувачів (адміністратор, менеджер, оператор тощо).
2. Управління ресурсами
  - 2.1. Додавання нових ресурсів (транспортні засоби, обладнання, працівники, тощо).
  - 2.2. Редагування існуючих даних про ресурси.
  - 2.3. Видалення застарілих або невикористовуваних ресурсів.
  - 2.4. Перегляд детальної інформації про кожен ресурс.
  - 2.5. Можливість кооперативного управління ресурсами аеропорту.
3. Фільтрація та пошук ресурсів
  - 3.1. Можливість пошуку ресурсів за різними параметрами (тип, статус, місцезнаходження, тощо).
  - 3.2. Використання динамічних фільтрів для швидкого доступу до необхідних даних.
4. Моніторинг та контроль використання ресурсів
  - 4.1. Відстеження статусу ресурсів у реальному часі.
5. Інтерактивний інтерфейс для адміністрування
  - 5.1. Зручний та адаптивний веб-інтерфейс для керування ресурсами.

## 5.2. Використання графічних елементів для візуалізації даних.

Завдяки цим функціям система забезпечує ефективне адміністрування ресурсів аеропорту, підвищуючи продуктивність роботи та оптимізуючи використання наявних ресурсів.

### Використані технології

У процесі розробки системи управління ресурсами аеропорту було використано сучасні веб-технології, що забезпечують високу продуктивність, масштабованість та безпеку. Цей розділ описує основні технології, застосовані у серверній та клієнтській частинах проєкту.

Серверна частина (Back-end) розроблена з використанням Nest.js – фреймворку для Node.js, який базується на архітектурі модулів і підтримує створення масштабованих серверних застосунків.

1. Nest.js – забезпечує структуровану розробку та підтримку принципів OOP, FP та SOLID.
2. MongoDB – NoSQL база даних для зберігання інформації про ресурси аеропорту.
3. Mongoose – ODM-бібліотека для взаємодії з MongoDB.
4. JSON Web Token (JWT) – для аутентифікації та авторизації користувачів.
5. OAuth 2.0 (Google та GitHub) – для аутентифікації та авторизації користувачів через сторонні сервіси.
6. Swagger – для створення документації до API.
7. Cache-manager – для оптимізації швидкодії завдяки кешуванню даних.
8. NestJS Throttler – для обмеження кількості запитів від одного користувача на певний період часу (Максимальна кількість запитів – 60 в хвилину).
9. Docker – для контейнеризації застосунку та його зручного розгортання.

Клієнтська частина (Front-end) розроблена за допомогою фреймворку Next.js, що дозволяє реалізовувати як серверний, так і клієнтський рендеринг, покращуючи продуктивність веб-додатка.

1. Next.js – фреймворк для React, що підтримує серверний рендеринг (SSR) та статичну генерацію (SSG).
2. TypeScript – для типізації та запобігання помилок на етапі компіляції проєкту.
3. Material-UI (MUI) – бібліотека компонентів для швидкої побудови інтерфейсу.
4. Tailwind CSS – утилітарний CSS-фреймворк для стилізації компонентів.
5. Formik та Yup – для валідації та управління формами користувачів.
6. Framer Motion – для анімацій і покращення взаємодії користувача із системою.
7. Jotai – для використання глобальних станів.

Інструменти розробки:

1. Git та GitHub – система контролю версій для командної роботи.
2. Insomnia – для тестування API-запитів.
3. ESLint та Prettier – для забезпечення якості та чистоти коду.

Завдяки застосуванню цих технологій вдалося створити ефективну та надійну систему управління ресурсами аеропорту, яка відповідає сучасним вимогам щодо продуктивності та масштабованості.

### Висновки

У результаті проведеного дослідження та розробки було створено ефективну систему управління ресурсами аеропорту, яка дозволяє автоматизувати ключові процеси адміністрування та оптимізувати використання наявних ресурсів.

Основні досягнення:

1. Ефективне адміністрування ресурсів: Розроблена система забезпечує централізоване управління всіма типами ресурсів аеропорту, включаючи транспортні засоби, обладнання, персонал та інвентар. Це дозволяє покращити координацію та контроль за використанням ресурсів, зменшити час на їх пошук та розподіл, а також мінімізувати ризик помилок, пов'язаних з ручним управлінням.
2. Оптимізація використання ресурсів: Завдяки функціям моніторингу та аналізу даних, система дозволяє відстежувати стан ресурсів у реальному часі, прогнозувати їх використання та планувати необхідні заходи для забезпечення їх оптимального використання. Це допомагає зменшити простой, підвищити ефективність роботи та знизити операційні витрати.
3. Висока продуктивність, безпека та масштабованість: Використання сучасних веб-технологій, таких як Nest.js та Next.js, забезпечує високу продуктивність та швидкість роботи системи.

Застосування JSON Web Token (JWT) та OAuth для аутентифікації та авторизації користувачів гарантує безпеку доступу до даних. Архітектура системи дозволяє легко масштабувати її для підтримки зростаючих обсягів даних та користувачів.

4. Універсальність застосування: Розроблена система може бути використана не лише в аеропортах, а й у логістичних компаніях, транспортних структурах та інших організаціях, що потребують ефективного управління ресурсами. Її гнучка архітектура та широкий набір функцій дозволяють адаптувати систему до специфічних потреб різних користувачів.

Загалом, розроблена система управління ресурсами аеропорту є важливим кроком у напрямку автоматизації та оптимізації процесів управління ресурсами. Її впровадження дозволить підвищити ефективність роботи аеропортів та інших організацій, зменшити витрати та покращити якість обслуговування.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. European Union Aviation Safety Agency (EASA). (2022). European Aviation Environmental Report 2022.
2. Wells, A. T., & Young, S. B. (2004). Airport planning & management (5th ed.). McGraw-Hill Professional.
3. Next.js документація [Електронний ресурс] – <https://nextjs.org/docs>
4. MongoDB документація [Електронний ресурс] - <https://www.mongodb.com/docs/>

**Буханенко Артем Павлович** – студент групи Б-123-21-3-СП, факультет комп'ютерних наук та технологій, кафедра інтелектуальних кібернетичних систем, Державне некомерційне підприємство "Державний університет "Київський авіаційний інститут", м.Київ, e-mail: [5791054@stud.kai.edu.ua](mailto:5791054@stud.kai.edu.ua)

Науковий керівник: **Наталія Вікторівна Апенко** – доцент, кандидат технічних наук, кафедра інтелектуальних кібернетичних систем, Державне некомерційне підприємство "Державний університет "Київський авіаційний інститут", м.Київ, e-mail: [nataliia.apenko@npp.kai.edu.ua](mailto:nataliia.apenko@npp.kai.edu.ua)

**Bukhanenko Artem P.** – Faculty of Computer Science and Technologies, State Non-Profit Enterprise "State University "Kyiv Aviation Institute", Kyiv, Ukraine, e-mail: [5791054@stud.kai.edu.ua](mailto:5791054@stud.kai.edu.ua)

Supervisor: **Nataliia Viktorivna Apenko** - Associate Professor, Candidate of Technical Sciences, Department of Intelligent Cybernetic Systems, State Non-Profit Enterprise "State University "Kyiv Aviation Institute", Kyiv, Ukraine, e-mail: [nataliia.apenko@npp.kai.edu.ua](mailto:nataliia.apenko@npp.kai.edu.ua)