

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ННІ ЕНЕРГЕТИКИ, АВТОМАТИКИ І ЕНЕРГОЗБЕРЕЖЕННЯ**

ПОГОДЖЕНО

**Директор ННІ енергетики,
автоматики і енергозбереження**

проф., д.т.н. КАПЛІУН В.В.
(підпис)

« _____ » _____ 2026 р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

**В.о завідувача кафедри
автоматики та робототехнічних систем
ім. акад. І.І. Мартиненка**

доц., к.т.н. ОПРИШКО О.О.
(підпис)

« _____ » _____ 2026 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

**на тему: «АВТОМАТИЗОВАНА СИСТЕМА АНАЛІЗУ ДАНИХ ТА
ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО АБІТУРІЄНТІВ ННІ
ЕНЕРГЕТИКИ, АВТОМАТИКИ І ЕНЕРГОЗБЕРЕЖЕННЯ»**

Спеціальність 174 – Автоматизація, комп'ютерно-інтегровані технології та
робототехніка

Освітня програма Автоматизація, комп'ютерно-інтегровані технології та
робототехніка

Орієнтація освітньої програми освітньо-наукова

Гарант освітньої програми

Д.Т.Н., ДОЦЕНТ
(науковий ступінь та вчене звання)

(підпис)

Івашук В.В.
(ПІБ)

Керівник магістерської кваліфікаційної роботи

К.Т.Н., ДОЦЕНТ
(науковий ступінь та вчене звання)

(підпис)

Дудник А.О.
(ПІБ)

Виконав

(підпис)

Новак Б. В.
(ПІБ)

КИЇВ – 2026

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ННІ ЕНЕРГЕТИКИ, АВТОМАТИКИ І ЕНЕРГОЗБЕРЕЖЕННЯ**

ПОГОДЖЕНО
Директор ННІ енергетики, автоматики і
енергозбереження

проф., д.т.н. _____ **КАПЛУН В.В.**
(підпис)
« ____ » _____ 2026 р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
В.о завідувача кафедри
автоматики та робототехнічних систем ім.
акад. І.І. Мартиненка
доц., к.т.н. _____ **ОПРИШКО О.О.**
(підпис)
« ____ » _____ 2026 р.

**ЗАВДАННЯ
ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ ЗДОБУВАЧУ
Новаку Божену Владиславовичу**

Спеціальність 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Освітня програма Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Орієнтація освітньої програми освітньо-наукова

Тема магістерської кваліфікаційної роботи: «Автоматизована система аналізу даних та підтримки прийняття рішень щодо абітурієнтів ННІ енергетики, автоматики і енергозбереження» затверджена наказом від 22.10.2025 № 2440

Термін подання завершеної роботи на кафедру 22. 05. 2026

Вихідні дані до магістерської кваліфікаційної роботи: відомості про регіони та райони України; відомості про населені пункти та їх типи (місто, село, селище тощо); відомості про навчальні заклади (назва, ЄДРПОУ, тип, тип освіти, тип фінансування, місцезнаходження); відомості про кафедри навчально-наукового інституту енергетики, автоматики і енергозбереження НУБіП України; перелік спеціальностей інституту з кодами; перелік месенджерів для зв'язку з абітурієнтами.

Перелік питань, що підлягають дослідженню:

1. Постановка задачі та аналіз предметної області
2. Проектування та реалізація автоматизованої системи
3. Тестування системи
4. Оцінка ефективності системи

Дата видачі завдання «18» листопада 2024 р.

**Керівник магістерської
кваліфікаційної роботи**

(підпис)

Дудник А.О.
(ПІБ)

**Директор ННІ енергетики,
автоматики і енергозбереження**

(підпис)

Каплун В.В.
(ПІБ)

Завдання прийняв до виконання

(підпис)

Новак Б.В.
(ПІБ)

АНОТАЦІЯ

Кваліфікаційна робота рівня магістр містить аналіз існуючих систем збору даних абітурієнтів, обґрунтування технологічного стеку та архітектурного підходу, проектування і реалізацію автоматизованої системи, а також оцінку її ефективності.

Мета роботи полягає у розробці автоматизованої системи аналізу даних та підтримки прийняття рішень щодо абітурієнтів навчально-наукового інституту енергетики, автоматики і енергозбереження НУБіП України.

Записка складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Записка містить 18 рисунків, 12 таблиць та 56 посилань на використані джерела. Повний обсяг роботи становить 73 сторінок, з них 56 сторінок основного тексту.

У першому розділі проведено порівняльний аналіз чотирьох існуючих систем збору даних абітурієнтів, обрано та обґрунтовано технологічний стек системи, проаналізовано підходи до побудови мікросервісних систем та сформульовано вимоги до розроблюваної системи.

У другому розділі описано архітектуру системи з двома мікросервісами NestJS зі спільною базою даних PostgreSQL, спроектовано логічну модель даних із 17 сутностями, реалізовано алгоритм підрахунку показника підозрілості заявок, дворівневий рольовий контроль доступу, статистичні звіти та аналітичні інструменти адмін-панелі.

У третьому розділі описано тестування системи, проведено оцінку ефективності впровадження та економічне обґрунтування. Впровадження системи забезпечує економію близько 39 годин праці науково-педагогічних працівників на рік та зниження відсотку незавершених заявок абітурієнтів з 38% до 12%.

КЛЮЧОВІ СЛОВА: МІКРОСЕРВІСИ, NESTJS, POSTGRESQL, RBAC, JWT, ЗАЯВКА АБІТУРІЄНТА, КЛАСИФІКАЦІЯ ДАНИХ.

ABSTRACT

The Master's qualification work includes an analysis of existing applicant data collection systems, justification of the technology stack and architectural approach, design and implementation of an automated system, and evaluation of its effectiveness.

The aim is to develop an automated data analysis and decision support system for applicants to the Institute of Energy, Automation and Energy Saving of NUBiP of Ukraine.

The thesis consists of an introduction, four chapters, conclusions, a list of references and appendices. The thesis contains 18 figures, 12 tables and 56 references. The total volume of the work is 73 pages, of which 56 pages are the main text.

The first chapter provides a comparative analysis of four existing applicant data collection systems, justifies the technology stack, analyses microservice architectural approaches and formulates system requirements.

The second chapter describes the system architecture with two NestJS microservices sharing a PostgreSQL database, designs a logical data model with 17 entities, and implements a suspicion scoring algorithm, two-level role-based access control, statistical reports and admin panel analytical tools.

The third chapter describes system testing, evaluates implementation effectiveness and provides economic justification. The system implementation saves approximately 39 hours of academic staff work per year and reduces the applicant dropout rate from 38% to 12%.

KEYWORDS: MICROSERVICES, NESTJS, POSTGRESQL, RBAC, JWT, APPLICANT APPLICATION, DATA CLASSIFICATION.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1	
ПОСТАНОВКА ЗАДАЧІ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	12
1.1. Аналіз існуючих систем збору та обробки даних абітурієнтів	12
1.2. Вибір та обґрунтування технологічного стеку	14
1.3. Аналіз підходів до побудови мікросервісних систем	16
1.4. Вимоги до системи збору та аналізу даних абітурієнтів	18
Висновки до розділу 1	19
РОЗДІЛ 2	
ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ	21
2.1. Архітектура системи та організація мікросервісів	21
2.2. Логічна модель даних автоматизованої системи.....	25
2.3. Реалізація програмних модулів та алгоритмів обробки даних	29
2.4. Статистичні звіти та аналітичні інструменти адмін-панелі	34
Висновки до розділу 2	39
РОЗДІЛ 3	
ТЕСТУВАННЯ СИСТЕМИ	41
3.1. Тестування та налагодження системи.....	41
Висновки до розділу 3	49
РОЗДІЛ 4	
ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ.....	50
4.1. Оцінка ефективності системи	50
Висновки до розділу 4	53
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТКИ.....	64

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- API — Application Programming Interface (інтерфейс програмування застосунків)
- CDN — Content Delivery Network (мережа доставки контенту)
- CRUD — Create, Read, Update, Delete (створення, читання, оновлення, видалення)
- DTO — Data Transfer Object (об'єкт передачі даних)
- ERD — Entity-Relationship Diagram (діаграма сутність-зв'язок)
- JWT — JSON Web Token (токен автентифікації формату JSON)
- NPV — Net Present Value (чиста приведена вартість)
- ORM — Object-Relational Mapping (об'єктно-реляційне відображення)
- RBAC — Role-Based Access Control (рольова модель контролю доступу)
- REST — Representational State Transfer
- ROE — Return on Equity (рентабельність власного капіталу)
- SPA — Single Page Application (односторінковий застосунок)
- SQL — Structured Query Language (мова структурованих запитів)
- VPS — Virtual Private Server (віртуальний виділений сервер)
- МОН — Міністерство освіти і науки України
- ННІ — навчально-науковий інститут
- НПП — науково-педагогічний працівник
- ПЗ — програмне забезпечення
- СКБД — система керування базами даних
- ЧПВ — чиста приведена вартість

ВСТУП

Актуальність теми. В умовах інтенсивного розвитку цифрових технологій та зростання конкуренції між закладами вищої освіти за абітурієнтів дедалі більшої актуальності набуває автоматизація процесів вступної кампанії. Наявна практика збору заявок абітурієнтів навчально-наукового інституту енергетики, автоматики і енергозбереження НУБіП України (ННІ) на основі Google Forms виявила низку системних недоліків: відсутність структурованих довідників навчальних закладів призводить до суттєвих помилок і дублювань при введенні даних; адміністратори не мають розмежування доступу за кафедрами; формування статистичних звітів вимагає до 10 годин ручної обробки на кожні 100 заявок; близько 38% абітурієнтів не завершують заповнення форми через незручний інтерфейс [16, 17]. Жодне з наявних загальних рішень — Google Forms, Jotform, Paperform чи Вступ.ОСВІТА.UA — не задовольняє комплексних потреб ННІ, що зумовлює необхідність розробки спеціалізованої автоматизованої системи.

Мета роботи полягає у розробці автоматизованої системи аналізу даних та підтримки прийняття рішень щодо абітурієнтів навчально-наукового інституту енергетики, автоматики і енергозбереження НУБіП України.

Завдання роботи:

- проаналізувати існуючі рішення для збору та обробки даних абітурієнтів і обґрунтувати необхідність розробки власної системи;
- обрати та обґрунтувати технологічний стек для реалізації системи;
- спроектувати мікросервісну архітектуру системи та логічну модель даних;
- розробити алгоритм автоматичного підрахунку показника підозрілості заявок;
- реалізувати рольову модель доступу з розмежуванням за кафедрами;

- розробити модуль статистичних звітів для підтримки прийняття управлінських рішень;
- провести тестування системи та оцінити ефективність її впровадження.

Об'єкт дослідження — процес збору, обробки та аналізу даних абітурієнтів у ході вступної кампанії навчально-наукового інституту.

Предмет дослідження — методи та засоби автоматизації збору заявок абітурієнтів, алгоритми інтелектуального аналізу даних вступної кампанії та архітектурні підходи до побудови відповідних інформаційних систем.

Методи дослідження. У процесі виконання магістерської кваліфікаційної роботи застосовано такі методи: системний аналіз — для дослідження існуючих систем збору даних абітурієнтів та виявлення їхніх недоліків; порівняльний метод — для вибору та обґрунтування технологічного стеку шляхом зіставлення альтернативних рішень; метод проектування інформаційних систем — для розробки логічної моделі даних та архітектури мікросервісів; метод алгоритмізації — для формалізації алгоритмів підрахунку показника підозрілості заявок та рольового контролю доступу; емпіричний метод — для тестування розробленої системи та оцінки ефективності її впровадження шляхом порівняння з попередньою практикою.

Наукова новизна отриманих результатів полягає у наступному: вперше запропоновано поєднання автоматизованого збору заявок абітурієнтів із використанням структурованих географічних довідників та довідників навчальних закладів як засобу забезпечення якості вхідних даних інформаційної системи; удосконалено алгоритм класифікації заявок абітурієнтів на основі правил підрахунку балів підозрілості за вісьмома критеріями, що на відміну від методів машинного навчання забезпечує повну прозорість та інтерпретованість результатів класифікації.

Практична цінність роботи полягає у розробці та впровадженні системи, яка забезпечує: скорочення часу заповнення заявки абітурієнтом з 10 до 5 хвилин; зниження відсотку незавершених заявок з 38% до 12%; рольову модель доступу з автоматичним розмежуванням за кафедральною приналежністю науково-педагогічних працівників; автоматичне формування статистичних звітів із обчисленням коефіцієнтів конверсії для оцінки ефективності рекрутерів; економію близько 39 годин праці науково-педагогічних працівників на рік завдяки автоматизації статистичної обробки даних.

Апробація результатів. Основні положення та результати магістерської кваліфікаційної роботи доповідались та обговорювались на засіданні секції «Енергозабезпечення, електротехнології, електротехніка та інтелектуальні управляючі системи в АПК» 79-ї науково-практичної конференції студентів НУБіП України (Київ, 2026). Систему впроваджено у реальних умовах вступної кампанії ННІ НУБіП України, де вона накопичила дані щодо 509 заявок абітурієнтів.

Положення, винесені на захист:

1. Результати порівняльного аналізу існуючих систем збору даних абітурієнтів, що обґрунтовують необхідність розробки спеціалізованої системи для ННІ НУБіП України.
2. Архітектурне рішення системи на основі двох мікросервісів NestJS зі спільною базою даних PostgreSQL, матриця рольового контролю доступу для трьох ролей користувачів та схема JWT-автентифікації без реєстрації для абітурієнтів.
3. Логічна модель даних системи, що охоплює 17 сутностей, зокрема обґрунтування застосування JSONB-полів для зберігання показника підозрілості заявок та гнучких метаданих.
4. Алгоритм автоматичного підрахунку показника підозрілості заявки за вісьмома критеріями з трьома рівнями класифікації, що реалізує метод класифікації на основі правил без застосування машинного навчання.

5. Результати оцінки ефективності впровадженої системи, що підтверджують економію близько 39 годин праці науково-педагогічних працівників на рік та зниження відсотку незавершених заявок з 38% до 12%.

Структура роботи. Магістерська кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Основний текст роботи викладено на 73 сторінках і містить 12 таблиць, 18 рисунків. Список використаних джерел налічує 56 найменувань.

РОЗДІЛ 1

ПОСТАНОВКА ЗАДАЧІ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз існуючих систем збору та обробки даних абітурієнтів

В умовах інтенсивного розвитку інформаційних технологій та цифровізації освітнього середовища дедалі більшої актуальності набуває автоматизація адміністративних процесів у закладах вищої освіти. Одним із таких процесів є збір та обробка даних абітурієнтів у ході вступної кампанії [16]. Автоматизовані інформаційні системи дозволяють скоротити витрати часу на рутинні операції, підвищити точність зібраних даних та забезпечити їх структурованість для подальшого аналізу [17].

Системи автоматизованого збору даних абітурієнтів поділяються на дві категорії: загальні системи збору форм та спеціалізовані системи для подачі заявок на вступ. Для визначення відповідності існуючих рішень потребам ННІ НУБіП було проведено аналіз чотирьох найбільш поширених платформ.

Google Forms є безкоштовним сервісом Google для створення онлайн-форм з інтеграцією у Google Sheets [30]. Попри простоту використання, сервіс має суттєві обмеження: назва школи вводиться довільним текстом, що унеможлиблює коректну статистичну агрегацію; відсутнє розмежування доступу за кафедрами; інструменти аналітики є мінімальними.

Jotform є хмарним конструктором форм із широким набором типів полів [31]. Розширена функціональність доступна лише за платною підпискою, а структуровані довідники навчальних закладів та механізми розмежування доступу за кафедрами відсутні.

Paperform вирізняється сучасним дизайном та гнучкістю налаштування форм [32]. Проте сервіс є повністю платним, не має інструментів статистичної звітності та спеціалізованих довідників.

Вступ.ОСВІТА.UA є офіційною платформою МОН України для подання заяв на вступ до закладів вищої освіти [33]. Попри зручний інтерфейс для

абітурієнтів та інтеграцію з державними реєстрами, платформа не передбачає рольового доступу для НПП за кафедральною приналежністю та внутрішньої аналітики вступної кампанії.

Результати порівняльного аналізу наведено у таблиці 1.1.

Таблиця 1.1

Порівняльний аналіз існуючих систем збору та обробки даних абітурієнтів

Критерій	Google Forms	Jotform	Paperform	Вступ.ОСВІТА.UA
Безкоштовність	✓	Частково	✗	✓
Структуровані довідники шкіл	✗	✗	✗	✓
Розмежування доступу за кафедрами	✗	✗	✗	✗
Статистичні звіти	Обмежено	Обмежено	✗	✗
Валідація даних	✗	Частково	Частково	✓
Інтерфейс для НПП	✗	✗	✗	✗

Таким чином, жодна з розглянутих платформ не задовольняє комплексних потреб ННІ НУБіП. Відсутність структурованих довідників у загальних системах призводить до суттєвих проблем із якістю даних. Практичний досвід використання Google Forms засвідчив, що близько 38% абітурієнтів не завершують подачу заявки, а ручна обробка статистики займає до 10 годин на кожні 100 заявок [34, 9]. За умови приблизно 200 заявок на рік сукупні витрати часу НПП на адміністративну обробку даних становлять близько 20 годин лише на статистику, що зумовлює об'єктивну потребу у розробці спеціалізованої

системи. Аналіз вимог до такої системи та обґрунтування технологічного стеку для її реалізації є предметом розгляду наступного підрозділу.

1.2. Вибір та обґрунтування технологічного стеку

Вибір технологічного стеку є одним із ключових етапів проектування інформаційної системи, оскільки безпосередньо визначає її масштабованість, надійність та зручність подальшої підтримки [22].

Мова програмування та серверний фреймворк. Для реалізації серверної частини обрано TypeScript — надмножину JavaScript зі статичною типізацією, що дозволяє виявляти помилки на етапі компіляції та суттєво підвищує якість коду при розробці складних багатосервісних систем [35, 18, 36]. Як серверний фреймворк обрано NestJS, архітектура якого побудована на модульності та впровадженні залежностей (Dependency Injection) [37]. Декораторна модель NestJS природно відображається на мікросервісну структуру системи, а вбудована підтримка бібліотеки class-validator забезпечує декларативну валідацію вхідних даних.

Засіб об'єктно-реляційного відображення. Обрано MikroORM, що реалізує патерн Unit of Work — відстеження всіх змін сутностей та їх групове збереження в єдиній транзакції [39]. Порівняно з TypeORM [40] та Prisma, MikroORM забезпечує вищу передбачуваність поведінки при роботі зі складними графами взаємопов'язаних сутностей.

Система керування базами даних. Порівняльний аналіз трьох найбільш поширених СКБД наведено у таблиці 1.2.

Таблиця 1.2

Порівняльний аналіз систем керування базами даних

Критерій	MySQL	MongoDB	PostgreSQL
Реляційна модель	✓	✗	✓
ACID-транзакції	✓	Частково	✓
Підтримка JSONB	✗	✓	✓
Складні запити	Середня	Обмежено	Висока
Цілісність даних	Висока	Середня	Висока

Обрано PostgreSQL як потужну реляційну СКБД із підтримкою ACID-транзакцій та типу даних JSONB [19, 41]. Критичною перевагою для розроблюваної системи є підтримка JSONB, що дозволяє зберігати напівструктуровані дані полів `otherInfo` та `suspicionInfo` без ускладнення основної схеми таблиць.

Клієнтська частина. Для розробки обох клієнтських застосунків обрано React — бібліотеку побудови інтерфейсів на основі компонентного підходу з розвиненою екосистемою та повноцінною підтримкою TypeScript [42]. Для адмін-панелі додатково обрано фреймворк `refine.dev`, що надає вбудований механізм постачальника даних для REST API та готові CRUD-сторінки з інтеграцією бібліотеки компонентів Ant Design [43, 44].

Інфраструктура. Розгортання серверної частини здійснюється через Docker Compose — інструмент оркестрації контейнерів з єдиним файлом конфігурації для всіх середовищ [45]. Nginx використовується як зворотний проксі для маршрутизації трафіку між мікросервісами [46]. Клієнтські застосунки розгорнуто на Cloudflare Pages — безкоштовному CDN-хостингу зі статичними React SPA та автоматичним HTTPS [48]. Для роботи з датами обрано

бібліотеку date-fns завдяки її модульності та незмінному підходу до об'єктів дат [50].

Зведену таблицю обраного технологічного стеку наведено у додатку А.1.

Таким чином, обраний технологічний стек забезпечує реалізацію всіх функціональних вимог до системи та створює необхідне підґрунтя для побудови мікросервісної архітектури, аналіз підходів до якої є предметом розгляду наступного підрозділу.

1.3. Аналіз підходів до побудови мікросервісних систем

Вибір архітектурного підходу є фундаментальним рішенням при проектуванні серверної частини системи. Архітектура програмного забезпечення являє собою сукупність структурних рішень, що визначають спосіб організації системи та взаємодію між її компонентами [22, 3]. Розглянуто три основні підходи.

Монолітна архітектура передбачає розгортання всіх компонентів системи як єдиного процесу. Перевагами є простота розробки та відсутність мережеских накладних витрат між модулями. Недоліками — складність масштабування окремих компонентів та ризик каскадних відмов при збої одного модуля.

Мікросервісна архітектура розподіляє застосунок на повністю незалежні сервіси, кожен із яких може бути розгорнутий та масштабований окремо [12, 4]. Це забезпечує ізоляцію відмов та технологічну гетерогенність, однак вносить значну складність міжсервісної комунікації та операційного управління.

Модульний моноліт є проміжним підходом: єдина кодова база, чітко розділена на ізольовані модулі зі строгими межами між ними [23]. Поєднує простоту розгортання моноліту із розмежуванням відповідальностей мікросервісів.

Порівняльний аналіз підходів наведено у таблиці 1.3.

Таблиця 1.3

Порівняльний аналіз архітектурних підходів

Критерій	Моноліт	Мікросервіси	Модульний моноліт
Складність початкової розробки	Низька	Висока	Середня
Незалежне масштабування	×	✓	Частково
Ізоляція відмов	×	✓	Частково
Складність розгортання	Низька	Висока	Низька
Придатність для одного розробника	Висока	Низька	Висока

За результатами аналізу для серверної частини системи обрано **модульний мікросервісний підхід** — два окремих NestJS-застосунки зі спільною базою даних PostgreSQL. Три доменні відповідальності системи (абітурієнти, адміністратори, довідники) природно відображаються на два сервіси. Спільна база даних усуває складність забезпечення консистентності між сервісами, відсутність міжсервісних HTTP-викликів знімає операційну складність класичних мікросервісів, а для команди одного розробника такий підхід є оптимальним балансом між ізоляцією та простотою [13].

Щодо організації точки входу для клієнтських запитів обрано Nginx як зворотний проксі замість спеціалізованого API Gateway, що є достатнім та простішим рішенням для масштабу системи з двома сервісами [47, 5].

Автентифікацію реалізовано на основі JWT-токенів відповідно до специфікації RFC 7519 [28]: короткоживучий access token для автентифікації запитів та довгоживучий refresh token із ротацією при кожному використанні. Кожен сервіс самостійно верифікує токен, що усуває залежність від централізованого сервісу автентифікації.

Спроектowana архітектура є основою для формулювання конкретних вимог до компонентів системи, що розглядається у наступному підрозділі.

1.4. Вимоги до системи збору та аналізу даних абітурієнтів

На основі проведеного аналізу існуючих рішень та обраного архітектурного підходу сформульовано вимоги до розроблюваної системи, систематизовані за чотирма групами [24].

Вимоги до інтерфейсу

Портал абітурієнта повинен забезпечувати покрокову форму подачі заявки з індикатором прогресу, каскадні випадаючі списки для вибору навчального закладу (регіон → тип населеного пункту → місто → школа) та рекрутера (кафедра → рекрутер), можливість редагування заявки протягом 15 хвилин після подачі, відображення статусу заявки та адаптивний дизайн для мобільних пристроїв.

Адмін-панель повинна надавати табличний перегляд заявок із сортуванням та фільтрацією за кафедрою, статусом, датою та ознакою підозрілості; inline-редагування статусу вступу та примітки безпосередньо в таблиці; дашбورد статистики з фільтрами; інтерфейс управління довідниками.

Функціональні вимоги

User Service забезпечує: подачу заявки без реєстрації на основі sessionId; валідацію полів форми через DTO; автоматичне обчислення показника підозрілості; 15-хвилинне вікно редагування; перегляд власної заявки.

Admin Service забезпечує: автентифікацію за email та паролем з видачею JWT; рольову модель доступу (НПП / системний адміністратор / начальство); автоматичну фільтрацію заявок за кафедрою для ролі НПП; inline-редагування статусу та примітки; коментування заявок; шість статистичних звітів із

фільтрацією за датою; масове оновлення заявок; експорт у Excel; CRUD-управління довідниками; журнал аудиту змін та подій безпеки.

Вимоги до безпеки

- хешування паролів алгоритмом Argon2 — переможцем Password Hashing Competition 2015, стійким до GPU-атак, на відміну від раніше поширеного алгоритму bcrypt [51], який не передбачає налаштування витрат пам'яті [29];
- JWT з ротацією refresh-токенів при кожному використанні;
- RBAC-захист на рівні кожного ендпоінту з автоматичним поверненням HTTP 403;
- фіксація IP-адреси при подачі заявки для алгоритму підозрілості;
- журналювання подій безпеки в таблиці `securityEvent`.

Операційні вимоги

- час відповіді API не більше 2 секунд для типових запитів;
- доступність системи 24/7;
- щотижневе резервне копіювання бази даних;
- розгортання на VPS за допомогою єдиної команди Docker Compose;
- коректне відображення у сучасних браузерях на десктопних та мобільних пристроях.

Сформульовані вимоги є вихідними даними для проектування архітектури системи та схеми бази даних, що розглядається у розділі 2.

Висновки до розділу 1

У першому розділі проведено комплексний аналіз предметної галузі та сформовано теоретичне підґрунтя для розробки системи.

Аналіз чотирьох існуючих рішень — Google Forms, Jotform, Paperform та Вступ.ОСВІТА.UA — показав, що жодне з них не задовольняє комплексних потреб ННІ НУБіП: відсутні структуровані довідники навчальних закладів, розмежування доступу за кафедрами та автоматизована статистична звітність.

На основі порівняльного аналізу обрано технологічний стек системи: TypeScript та NestJS для серверної частини, MikroORM як ORM, PostgreSQL як СКБД, React разом із refine.dev та Ant Design для клієнтських застосунків, Docker Compose, Nginx та Cloudflare Pages для інфраструктури.

Аналіз архітектурних підходів обґрунтував вибір модульного мікросервісного підходу — два окремих NestJS-застосунки зі спільною базою даних PostgreSQL, що забезпечує ізоляцію доменів за відсутності складності міжсервісної комунікації.

Сформульовано чотири групи вимог до системи, що є основою для проектування, яке розглядається у розділі 2.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

2.1. Архітектура системи та організація мікросервісів

Відповідно до обраного у підрозділі 1.3 модульного мікросервісного підходу серверна частина системи реалізована у вигляді двох незалежних NestJS-застосунків зі спільною базою даних PostgreSQL.

Загальна схема архітектури

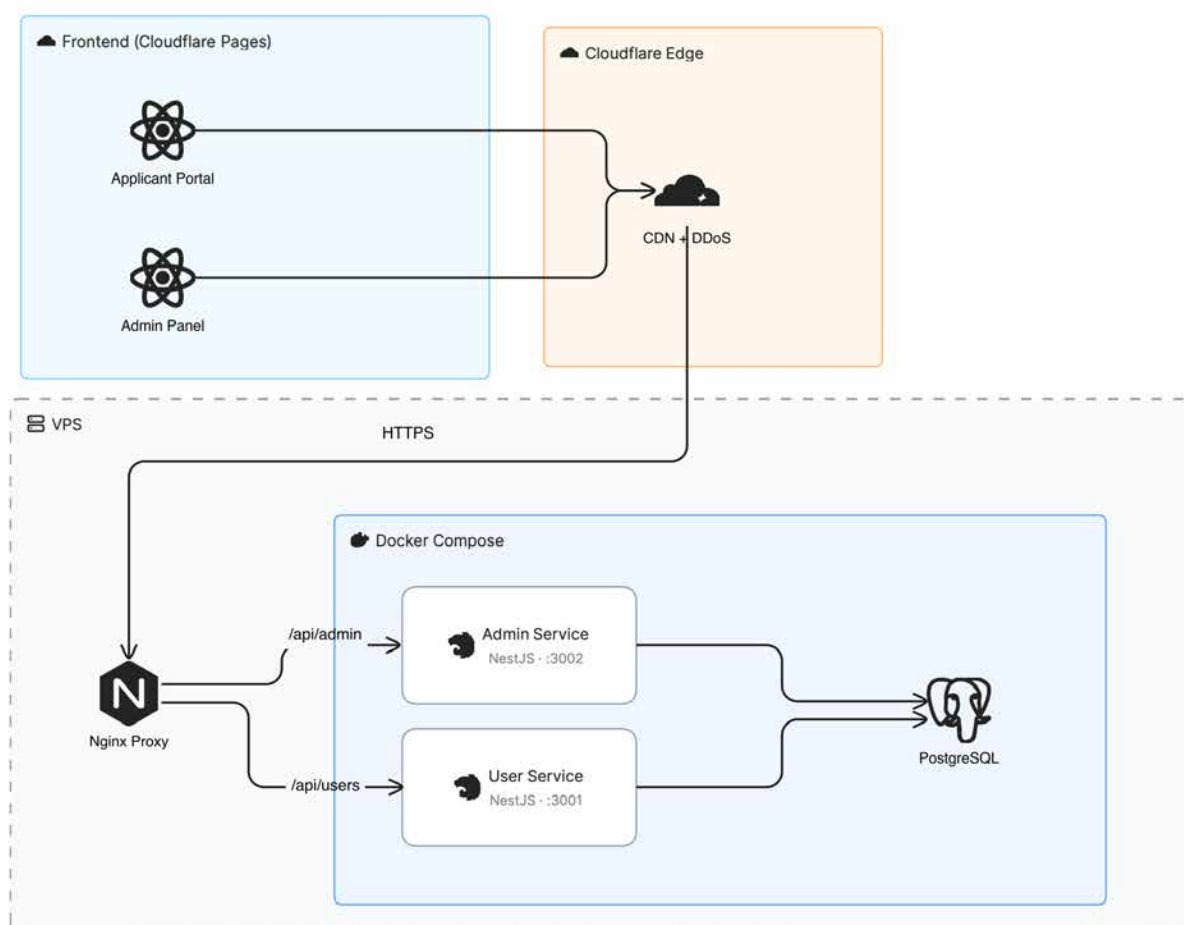
Система складається з чотирьох рівнів. На **клієнтському рівні** функціонують два React SPA на Cloudflare Pages: портал абітурієнта та адмін-панель [48]. На **рівні маршрутизації** Nginx виступає єдиною точкою входу, маршрутизуючи запити за субдоменом відповідно до таблиці 2.1 [47].

Таблиця 2.1

Таблиця маршрутизації Nginx

Субдомен	Мікросервіс	Порт
api.uniply.com.ua	User Service	3001
admin-api.uniply.com.ua	Admin Service	3002

На **серверному рівні** функціонують два мікросервіси: User Service обслуговує запити абітурієнтів та надає довідникові дані, Admin Service обслуговує запити НПП та адміністраторів. На **рівні зберігання даних** — єдиний екземпляр PostgreSQL, спільний для обох сервісів, що забезпечує консистентність даних на рівні транзакцій [19]. Зовнішній периметр захищає Cloudflare від DDoS-атак [49]. Загальну схему архітектури наведено на рис. 2.1.



Uniply

Розгортання здійснюється через Docker Compose із трьома контейнерами: `user-service`, `admin-service` та `postgres` [45]. Усі контейнери функціонують у спільній Docker-мережі, змінні середовища передаються через файли `.env`.

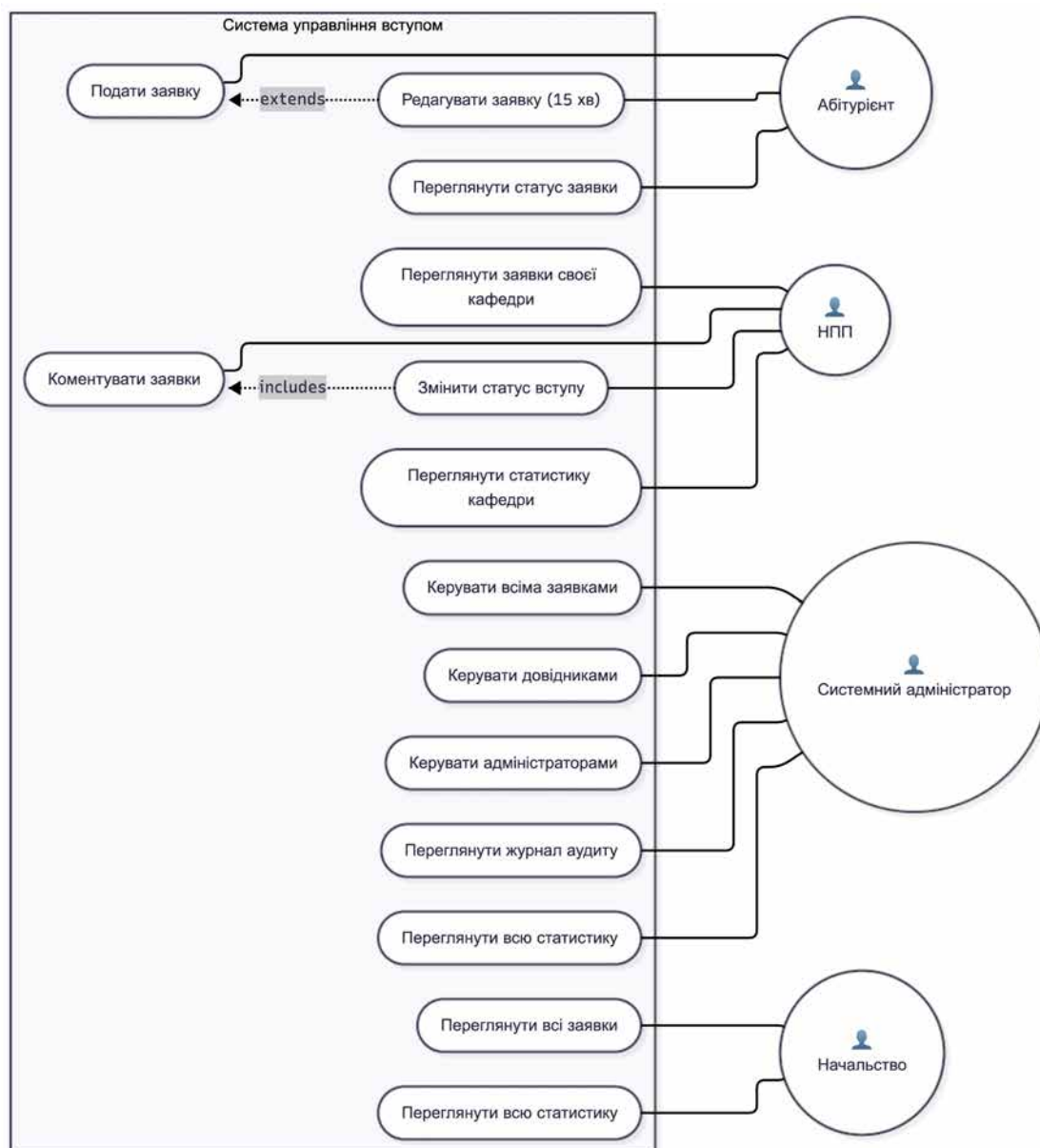
Рольова модель доступу

Система реалізує три ролі з різним рівнем повноважень. Матрицю доступу наведено у таблиці 2.2. Діаграму прецедентів системи наведено на рис. 2.2.

Таблиця 2.2

Матриця рольового доступу системи

Дія	НПП	Системний адміністратор	Начальство
Переглядати заявки своєї кафедри	✓	✓	✓
Переглядати всі заявки	✗	✓	✓
Inline-редагування статусу та примітки	✓	✓	✓
Редагувати поля заявки	✗	✓	✗
CRUD довідників	✗	✓	✗
Переглядати всю статистику	✗	✓	✓
Управляти адміністраторами	✗	✓	✗
Переглядати журнал аудиту	✗	✓	✗



Технічна реалізація RBAC здійснена через кастомний NestJS Guard, що зчитує роль із JWT payload та перевіряє матрицю дозволів [38, 6]. При недостатності прав повертається HTTP 403. Для ролі НПП декоратор `@DepartmentScope()` автоматично інjektує умову `WHERE referrer.department = :departmentId` до всіх запитів, унеможливаючи доступ до даних інших кафедр.

Схема JWT-автентифікації

Система реалізує два потоки автентифікації відповідно до RFC 7519 [28].

Автентифікація абітурієнта здійснюється без реєстрації: при першому зверненні сервер генерує `sessionId` та видає пару токенів — access token (TTL 15 хв) та refresh token (TTL 90 днів). При кожному використанні refresh token замінюється новим. Схему JWT-автентифікації наведено у дод. Б.1.

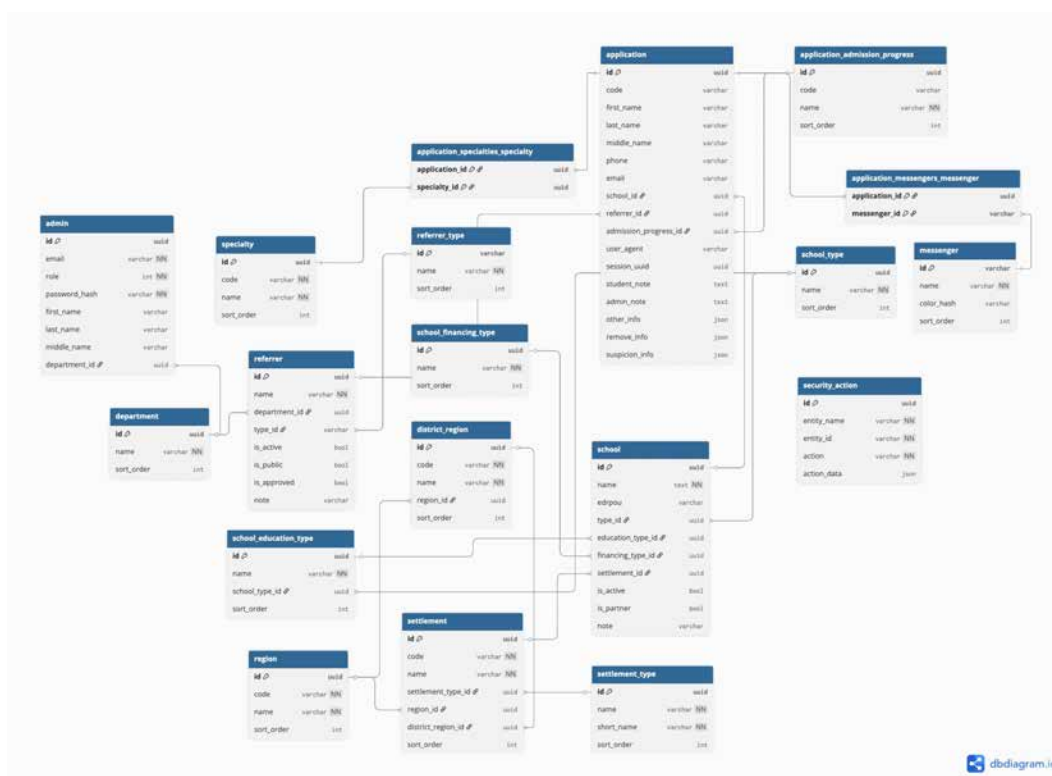
Автентифікація адміністратора реалізована за схемою email + пароль із верифікацією через хеш Argon2 [29]. Access token має TTL 30–60 хв, refresh token — 180–360 днів з урахуванням сценарію, коли НПП може не заходити до системи тижнями між вступними кампаніями.

Журнал аудиту `changeEvent` фіксує кожну мутуючу дію адміністратора, журнал `securityEvent` — події безпеки. Обидва реалізовані як окремі NestJS-сервіси, що викликаються після виконання основної дії.

Спроектowana архітектура визначає структуру взаємодії між компонентами та є основою для проектування схеми бази даних у наступному підрозділі.

2.2. Логічна модель даних автоматизованої системи

Логічна модель даних є проміжним етапом між концептуальною та фізичною моделями і описує структуру зберігання даних у термінах реляційних таблиць та зв'язків між ними [27]. Спільна база даних системи містить 17 сутностей, що охоплюють усі предметні галузі: дані заявок абітурієнтів, географічні довідники, довідники навчальних закладів, рекрутерів та спеціальностей, адміністративні сутності. ER-діаграму системи наведено на рис. 2.3.



ER-діаграма логічної моделі даних системи

Базові класи сутностей

Усі сутності системи побудовані на двох абстрактних базових класах. **BaseEntity** містить поля **createdAt** та **updatedAt** для аудиту часу створення та оновлення запису. **SoftDeletableEntity** розширює **BaseEntity** полем **deletedAt** — датою м'якого видалення [39]. За цим патерном запис не видаляється фізично, а лише позначається міткою часу. МікроORM-фільтр **softDelete** автоматично виключає такі записи з усіх запитів, що забезпечує збереження історичних даних.

Центральна сутність: Заявка

Сутність **Заявка** є центральною сутністю системи. Її ключові поля наведено у таблиці 2.3.

Таблиця 2.3

Опис ключових полів сутності Заявка

Поле	Тип	Опис
id	uuid	Первинний ключ
firstName, lastName, middleName	varchar	ПІБ абітурієнта
phone, email	varchar	Контактні дані
school_id	uuid (FK)	Навчальний заклад
referrer_id	uuid (FK)	Рекрутер
admission_progress_id	uuid (FK)	Прогрес вступу
sessionUuid	varchar	Ідентифікатор сесії пристрою
studentNote, adminNote	text	Нотатки абітурієнта та адміністратора
ipAddress, userAgent	varchar	Технічні дані для алгоритму підозрілості
otherInfo	jsonb	Гнучкі додаткові дані
suspicionInfo	jsonb	Результат алгоритму підозрілості
removeInfo	jsonb	Метадані видалення

Три JSONB-поля зберігають напівструктуровані дані змінної структури. Поле **otherInfo** містить дані про нестандартний навчальний заклад або рекрутера, введених вручну, та ознаки участі в підготовчих курсах і олімпіадах. Поле **suspicionInfo** зберігає результат алгоритму підозрілості: { **points**, **reason[]**, **isRefuted** }. Поле **removeInfo** містить метадані про видалення заявки. Використання JSONB дозволяє зберігати ці дані без

ускладнення основної схеми таблиці, зберігаючи можливості індексування PostgreSQL [41].

Сутність Заявка також має два зв'язки ManyToMany: зі сутністю **Спеціальність** через зведену таблицю `application_specialties_specialty` та зі сутністю **Месенджер** через `application_messengers_messenger`.

Географічні та довідникові сутності

Географічні довідники формують чотирирівневу ієрархію для каскадної фільтрації: **Область** → **Район** → **Населений пункт** → **Навчальний заклад**. Кожен рівень пов'язаний із наступним зовнішнім ключем. Додатково для Населеного пункту визначено **Тип населеного пункту** (місто, село, селище тощо).

Сутність **Навчальний заклад** містить поля `name`, `edrpou`, зв'язки з трьома довідниками типізації (тип закладу, тип освіти, тип фінансування), `isActive` та `isPartner` — ознаку підписаного договору про профорієнтаційну діяльність.

Сутність **Рекрутер** має поля `isPublic` (відображення в списку для абітурієнта) та `isApproved` (підтвердження адміністратором), що дозволяє керувати видимістю рекрутерів. Рекрутери пов'язані з **Кафедрою** та **Типом рекрутера** (викладач або студент).

Зв'язки між ключовими сутностями

Ключові зв'язки логічної моделі наведено у таблиці 2.4. Приклади SQL-запитів для всіх зв'язків наведено у додатку Б.2.

Таблиця 2.4

Ключові зв'язки логічної моделі даних

Зв'язок	Тип	Зовнішній ключ
З а я в к а → Н а в ч а л ь н и з а к л а д	ManyToOne	school_id
З а я в к а → Р е к р у т е р	ManyToOne	referrer_id
З а я в к а ↔ С п е ц і а л ь н і с т ь	ManyToMany	application_specialties_special ty
З а я в к а ↔ М е с е н д ж е р	ManyToMany	application_messengers_mess enger
Н а с е л е н и й п у н к т → О б л а с т ь	ManyToOne	region_id
Н а в ч а л ь н и й з а к л а д → Н а с е л е н и й п у н к т	ManyToOne	settlement_id

Повний перелік сутностей із зазначенням базового класу та ключових зв'язків наведено у додатку Б.3.

Спроектowana логічна модель даних охоплює всі предметні галузі системи та є основою для реалізації програмних модулів, що розглядається у наступному підрозділі.

2.3. Реалізація програмних модулів та алгоритмів обробки даних

На основі спроектованої логічної моделі даних реалізовано програмні модулі обох мікросервісів системи. Проект організовано як NestJS-монорепозиторій із двома застосунками в директорії **apps/** та спільними бібліотеками в **libs/** [38]. Бібліотека **libs/orm** містить усі MikroORM-

сутності, спільні для обох сервісів, `libs / core` — Guard авторизації, інтерфейс Voter, базові Pipe та сервіс журналювання.

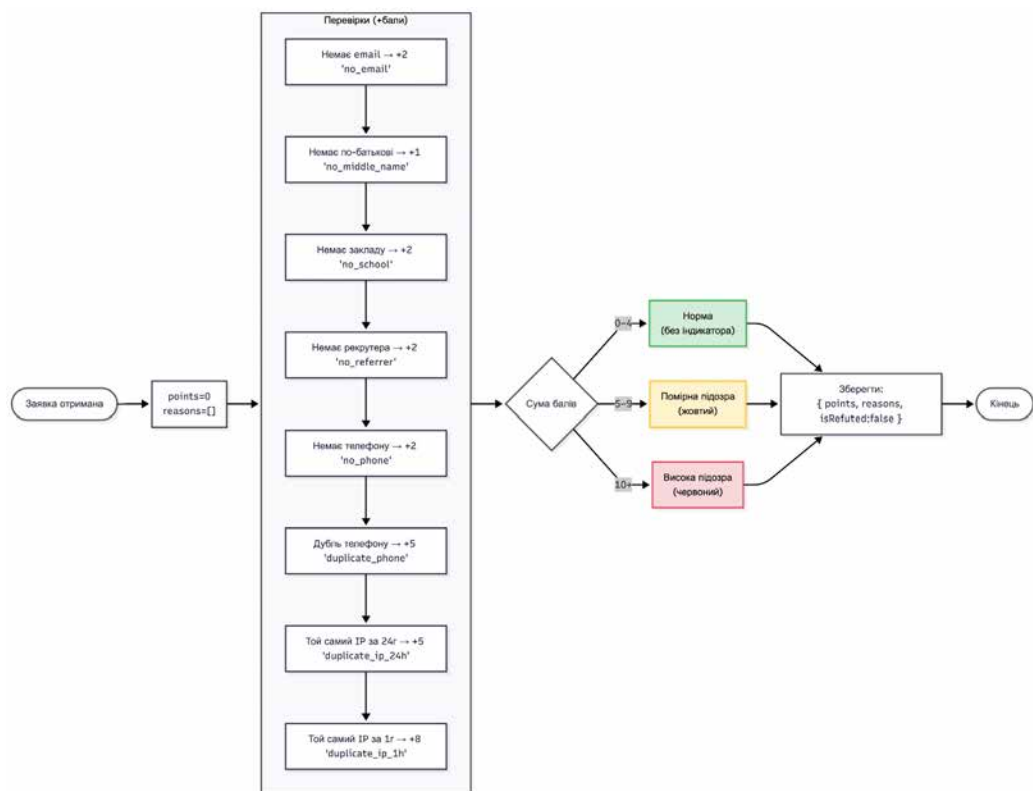
Архітектурний патерн Action та Voter

Ключовою архітектурною відмінністю від стандартного NestJS є застосування патерну **Action** у поєднанні з **Voter**. Action — клас, що реалізує одну конкретну операцію відповідно до принципу єдиної відповідальності [38]. Voter — клас, що здійснює авторизацію конкретної операції над конкретним екземпляром сутності (row-level security). Voter викликається з REQUEST-scoped Pipe: якщо `voter.vote(entity, user)` повертає `false`, Pipe генерує HTTP 403. Таким чином авторизація відбувається на двох рівнях: RolesGuard перевіряє право на тип операції, Voter — право на конкретний запис. Повний перелік модулів Admin Service наведено у додатку Б.4.

Алгоритм подачі заявки та підрахунку підозрілості

Подача заявки складається з трьох етапів. На **першому етапі** вхідні дані проходять валідацію через NestJS ValidationPipe на основі class-validator [53, 8]. При порушеннях повертається HTTP 400 із деталізованим переліком помилок. Діаграму діяльності процесу подачі заявки наведено у дод. Б.5.

На **другому етапі** обчислюється показник підозрілості заявки. Алгоритм послідовно перевіряє вісім умов (табл. 2.5) та накопичує бали. Блок-схему алгоритму підрахунку показника підозрілості наведено на рис. 2.4.



Таблиця 2.5

Умови алгоритму підрахунку підозрілості заявки

Умова	Бали	Код
Відсутній email	+2	no_email
Відсутнє по батькові	+1	no_middl e_name
Відсутній навчальний заклад	+2	no_school
Відсутній рекрутер	+2	no_referrer
Відсутній телефон	+2	no_phone
Дублікат телефону в базі	+5	dupl i cate_phone
Дублікат IP за 24 години	+5	dupl i cate_i p_24h
Дублікат IP за 1 годину	+8	dupl i cate_i p_1h

За результатами підрахунку заявці присвоюється рівень підозрілості: 0–4 бали — нормальна; 5–9 — помірна (жовтий індикатор); 10 і більше — висока (червоний індикатор). Результат зберігається у JSONB-полі `suspicionInfo`. Адміністратор може спростувати підозрілість через поле `isRefuted`.

На **третьому етапі** заявка зберігається з полями `ipAddress`, `userAgent` та `sessionId`. У відповіді повертається поле `canEditUntil` = `createdAt` + 15 хвилин.

Алгоритм 15-хвилинного вікна редагування

При PATCH-запиті сервер обчислює `canEditUntil` = `createdAt` + 15 хвилин та порівнює з поточним часом. Якщо вікно закінчилось — повертає

HTTP 403. Клієнт відображає таймер зворотного відліку на основі поля `canEditUntil`.

Алгоритм RBAC через Guards та Voters

RolesGuard зчитує JWT, верифікує підпис, витягує `{ adminId, role, departmentId }` та перевіряє матрицю дозволів [28, 38, 7]. Voter здійснює перевірку на рівні запису: для ролі НПП доступ до заявки дозволяється лише якщо `application.referrer.department.id = admin.departmentId`. Декоратор `@DepartmentScope()` автоматично інжектує умову фільтрації до запитів списку, виключаючи заявки без рекрутера з видимості НПП.

Статистичні звіти

Статистичні звіти реалізовані через сирі SQL-запити з PostgreSQL-агрегатами `COUNT(...)` `FILTER (WHERE ...)` [41], що дозволяє обчислювати кілька показників в одному запиті. Усі звіти приймають спільні параметри фільтрації: `dateFrom`, `dateTo`, `departmentIds`, `schoolTypeIds`, `admissionProgressCodes`. Для ролі НПП `@DepartmentScope()` автоматично перевизначає `departmentIds` значенням із JWT.

Звіт «Рекрутери» обчислює три показники через умовні агрегати: `total_count` — усі заявки рекрутера; `applied_count` — заявки зі статусом `3_APPLICATION_SUBMITTED`; `enrolled_count` — зі статусом `4_ADMITTED`. На їх основі обчислюються коефіцієнти конверсії `applied_count / total_count` та `enrolled_count / applied_count`.

Масове оновлення та об'єднання рекрутерів

Масове оновлення реалізує принцип fail-closed: перед застосуванням змін система завантажує лише ті заявки зі списку IDs, що доступні поточному

користувачу. Якщо хоч один ID недоступний — вся операція відхиляється з HTTP 403. Усі зміни виконуються в єдиній транзакції через `em flush()`.

Алгоритм об'єднання рекрутерів-дублікатів переприв'язує заявки до цільового запису, м'яко видаляє дублікати та фіксує операцію в журналі `SecurityAction`.

Реалізовані модулі та алгоритми забезпечують виконання всіх функціональних вимог, визначених у підрозділі 1.4. На їх основі у наступному підрозділі описуються статистичні звіти адмін-панелі.

2.4. Статистичні звіти та аналітичні інструменти адмін-панелі

Одним із ключових недоліків попередньої системи на основі Google Forms була повна відсутність автоматизованої статистики: формування звітів вимагало до 20 годин ручної обробки на рік. Адмін-панель розробленої системи усуває цю проблему через набір статистичних звітів із фільтрацією в режимі реального часу та функціональну таблицю заявок. Запровадження автоматизованої аналітики на основі даних відповідає сучасним підходам до проектування систем підтримки прийняття рішень в управлінні освітою [10].

Сторінка заявок

Головна сторінка адмін-панелі поєднує оперативну статистику та інтерактивну таблицю (рис. 2.5). У верхній частині відображаються три агреговані показники: загальна кількість заявок, кількість без обраної спеціальності та без встановленого статусу вступу. Нижче розміщено розподіл заявок за спеціальностями та за статусами вступного процесу. Ці показники формуються автоматично та оновлюються відповідно до застосованих фільтрів. Такий формат відповідає сучасній практиці освітніх дашбордів для управлінських рішень [11].

Таблиця заявок містить стовпці: ідентифікатор, ПІБ абітурієнта, телефон, email, навчальний заклад, спеціальність, дата подання, статус вступу, примітка. Реалізовано сортування та фільтрацію для всіх стовпців. Статус вступу та

примітка підтримують **inline-редагування** безпосередньо в таблиці без переходу на окрему сторінку — це відповідає вимозі технічного завдання щодо максимального наближення до формату електронних таблиць. Кнопка «Експорт» формує Excel-файл із поточними даними з урахуванням фільтрів [52].

ID	Прізвище	Ім'я	По батькові	Телефон	Email	Навчальний заклад	Спеціальність
1	Зозуля					Васильківський академічний ліцей "Успіх"	Електроенергетика, електротехніка та електромеханіка
2	Соловійов					Середня загальноосвітня школа №235 імені Вячеслава Чорновола	—
3	Назаренко					Середня загальноосвітня школа №235 імені Вячеслава Чорновола	—

Статистичні звіти

Система містить чотири статистичні звіти, зведений опис яких наведено у таблиці 2.6.

Таблиця 2.6

Статистичні звіти адмін-панелі

Звіт	Групування	Показники	Призначення
Навчальні заклади	За закладом	Кількість заявок, тип закладу	Виявлення ключових партнерських шкіл
Рекрутери	За рекрутером	Зареєстровані, подали заяву, вступили	Оцінка ефективності НПП
Заклади + Рекрутери	За парою (заклад, рекрутер)	Кількість заявок	Найпродуктивніші пари
Кафедри	За кафедрою	Школи / Коледжі окремо	Структура вступників кафедри

Усі звіти підтримують фільтрацію за часовим діапазоном, кафедрою, типом закладу та статусом вступу. Для ролі НПП фільтр кафедри встановлюється автоматично і не може бути змінений клієнтом.

Звіт «Рекрутери» є ключовим аналітичним

інструментом: три колонки показників дозволяють обчислити конверсію «привів → подав документи» ($\text{applied_count} / \text{total_count}$) та «подав → вступив» ($\text{enrolled_count} / \text{applied_count}$), що є головними метриками ефективності профорієнтаційної роботи НПП. Статистичний звіт «Рекрутери» наведено на рис. 2.6. Статистичний звіт «Заклади освіти та рекрутери» наведено на рис. 2.7.

Рекрутери

Період

Start date

→ End date

Тип закладу

Всі типи

Кафедри

Всі кафедри

Статус вступу

Всі статуси

Пройдено НМТ

Подано заявку до університету

Успішно вступив (-ла)

Не вступив (-ла)

контракт

N		Зареєстровані	Подали заяву на вступ	Вступили
1	Сисак Катерина Ярославівна	20	0	0
2	Сергій Володимирович Шостак	12	0	0
3	Мейш Юлія Анатоліївна	11	0	0
4	Панталієнко Людмила Анатоліївна	8	0	0
5	Савчук Світлана Геннадіївна	6	0	0
6	Щербак Тетяна Миколаївна	4	0	0
7	Пронь Тетяна Володимирівна	1	0	1

Заклади освіти + Рекрутери

Період

Start date

→ End date

Тип закладу

Всі типи

Кафедри

Всі кафедри

Статус вступу

Всі статуси

☐ Лише заклади, з якими підписано контракт

Nº	Навчальний заклад	НПП	Кількість
1	гімназія №178 м. Києва	Сисак Катерина Ярославівна	17
2	Бучанська спеціалізована загальноосвітня школа I-III ст. №5 з поглибленим вивченням іноземних мов Бучанської міської ради Київської області	Сергій Володимирович Шостак	8
3	Середня загальноосвітня школа №235 імені Вячеслава Чорновола	Мейш Юлія Анатоліївна	7
4	Володимир-Волинський фаховий коледж	Панталієнко Людмила Анатоліївна	4

Звіт «Кафедри» відображає розподіл заявок між кафедрами у розрізі типів навчальних закладів — шкіл та коледжів окремо, — що відображає різні освітні

рівні вступників і дозволяє кафедрам планувати навчальні ресурси. Статистичний звіт «Кафедри» наведено на рис. 2.8.

Кафедри

Період

Start date

→ End date

Тип закладу

Всі типи

Кафедри

Всі кафедри

Статус вступу

Всі статуси

☐ Лише заклади, з якими підписано контракт

№	Кафедра	Школи	Коледжі
1	Автоматики та робототехнічних систем ім. акад. І.І. Мартиненка	6	15
2	Вищої та прикладної математики	52	10
3	Електротехніки, електромеханіки та електротехнологій	47	147
4	Кафедра інженерії енергосистем	8	33
5	Не вказувати	6	0
6	Фізики	12	5
Всього		131	210

Звіт «Навчальні заклади» дозволяє визначити навчальні заклади, що є основними джерелами абітурієнтів, та заклади, у яких кількість заявок є низькою і потребує посилення профорієнтаційної роботи. Звіт підтримує фільтрацію за кафедрою, типом закладу та часовим діапазоном, а результати сортуються за кількістю заявок за спаданням. Інтерфейс звіту наведено на рис. 2.9.

Навчальні заклади

Період: Start date → End date  Тип закладу: Всі типи 

Кафедри: 

- Не вказувати
- Автоматики та робототехнічн...
- Вищої та прикладної математ... контракт
- Електротехніки, електромеха...
- Фізики
- Кафедра інженерії енергосист...

		Тип закладу	Кількість заявок
1	гімназія №178 м. Києва	Школа	17
2	Бучанська спеціалізована загальноосвітня школа I-III ст. №5 з поглибленим вивченням іноземних мов Бучанської міської ради Київської області	Школа	8
3	Середня загальноосвітня школа №235 імені Вячеслава Чорновола	Школа	7
4	Володимир-Волинський фаховий коледж	Коледж	4
5	Комунальний навчальний заклад «Навчально-виховний комплекс № 141 «Освітні ресурси та технологічний тренінг» міста Києва (дошкільний навчальний заклад - школа I ступеня - спеціалізована школа II ступеня з поглибленим вивченням іноземних мов та інформаційних технологій - технологічний ліцей)»	Школа	4

Висновки до розділу 2

У другому розділі описано проектування та реалізацію автоматизованої системи.

Спроековано архітектуру системи з двома мікросервісами NestJS зі спільною базою даних PostgreSQL, зворотним проксі Nginx, розгортанням через Docker Compose та хостингом на Cloudflare Pages. Визначено матрицю рольового доступу для трьох ролей та дві схеми JWT-автентифікації.

Спроековано логічну модель даних із 17 сутностями. Обґрунтовано використання механізму м'якого видалення та JSONB-полів для показника підозрілості, додаткових відомостей та метаданих видалення.

Реалізовано архітектурний патерн Action та Voter, алгоритм підрахунку підозрілості за вісьмома критеріями з трьома рівнями класифікації, алгоритм 15-хвилинного вікна редагування, дворівневий RBAC, п'ять статистичних звітів на основі PostgreSQL умовних агрегатів, безпечне масове оновлення за принципом fail-closed та алгоритм об'єднання рекрутерів-дублікатів.

Адмін-панель реалізує inline-редагування, масові дії, автоматичну статистику та Excel-експорт, що замінює до 20 годин ручної обробки на автоматичний розрахунок у режимі реального часу.

РОЗДІЛ 3

ТЕСТУВАННЯ СИСТЕМИ

3.1. Тестування та налагодження системи

Тестування є критичним етапом розробки програмного забезпечення, що гарантує його якість та надійність [25]. У процесі перевірки розробленої системи застосовано два види тестування: ручне та модульне.

Ручне тестування — процес перевірки програмного забезпечення шляхом відтворення реальних сценаріїв використання [26]. **Модульне тестування** — процес ізольованої перевірки окремих модулів вихідного коду, що дозволяє виявляти регресії при змінах коду [15].

Функціональне тестування порталу абітурієнта

Перевірено три ключові сценарії. **Успішна подача заявки:** покрокове заповнення форми з каскадним вибором навчального закладу та рекрутера, підтвердження у модальному вікні — заявка успішно збережена, відображено таймер редагування (рис. 3.1). **Валідація полів:** спроба подачі без обов'язкових полів — сервер повернув HTTP 400 із деталізованим переліком помилок по кожному полю (рис. 3.2). **15-хвилинне вікно:** редагування в межах вікна — HTTP 200, зміни збережено; спроба після закінчення — HTTP 403 (рис. 3.3).

1/1

Заявку отримано

Ми збережемо дані та зв'яжемося з вами найближчим часом.

- Можна редагувати ще 3:27

ОСОБИСТІ

ПІБ	Шевченко Володимир Миколайович
Телефон	+380952026555
Email	vol.sh@gmail.com
Месенджери	Telegram, WhatsApp

ЗАКЛАД

Навчальний заклад	Київський професійний енергетичний ліцей
-------------------	--

Редагувати

1/6

1 поле потребує уваги
Перевірте позначені поля нижче, щоб продовжити.

• КРОК 1

Контакти

Як з вами зв'язатись

Прізвище
Шевченко

Ім'я
Володимир

По батькові не обов'язкове
Миколайович

Телефон
+38095202605051000

⚠ Перевірте формат: +380 XX XXX XX XX

Email не обов'язкове
vol.sh@gmail.com

МЕСЕНДЖЕРИ

Telegram Viber WhatsApp

Продовжити

Рис. 3.2. Портал абітурієнта: відображення помилок валідації форми

1/1

Заявку отримано

Ми збережемо дані та зв'яжемося з вами найближчим часом.

- Час для редагування минув

ОСОБИСТІ

ПІБ: Шевченко Володимир Миколайович

Телефон: +380952026555

Email: vol.sh@gmail.com

Месенджери: Telegram, WhatsApp

ЗАКЛАД

Навчальний заклад: Київський професійний енергетичний ліцей

СПЕЦІАЛЬНОСТІ

G3: Електроенергетика, електротехніка та електромеханіка

G7: Автоматизація комп'ютерно-інтегрованих систем

Рис. 3.3. Портал абітурієнта: повідомлення про закінчення вікна редагування

Функціональне тестування адмін-панелі

Перевірено п'ять сценаріїв. **Рольовий доступ:** вхід із обліковими даними НПП відображає лише заявки своєї кафедри; зміна фільтра кафедри не впливає на результат — сервер підставляє кафедру з JWT (рис. 3.4). **Inline-редагування:** зміна статусу вступу через dropdown та редагування примітки зберігаються без переходу на окрему сторінку (рис. 3.5). **Масове оновлення:** вибір кількох заявок через чекбокси, застосування змін — API повернув `{ updated: N }` (рис. 3.6). **Статистичні звіти:** фільтрація за кафедрою та датою — результати збіглися з прямими SQL-запитами до бази даних (рис. 3.7). **Excel-експорт:** отримано `xlsx`-файл із коректними даними відповідно до застосованих фільтрів (рис. 3.8).

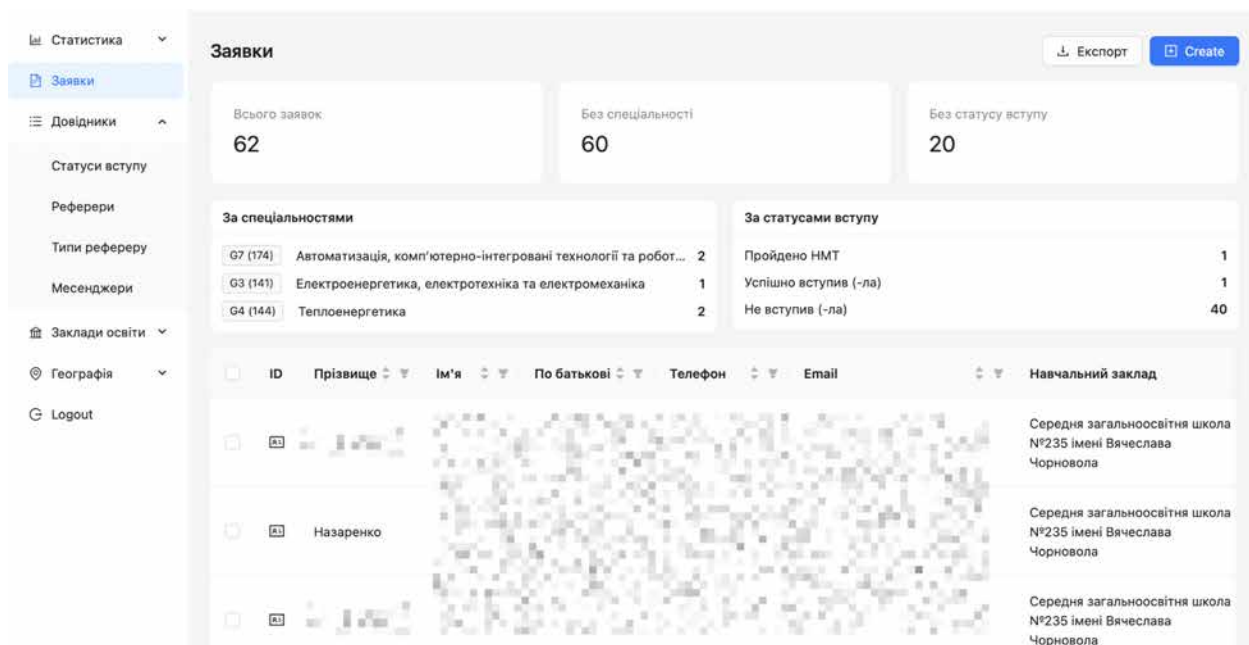


Рис. 3.4. Адмін-панель: таблиця заявок для ролі НПП з автоматичним фільтром кафедри

Навчальний заклад	Спеціальність	Дата подання	Статус вступу
Середня загальноосвітня школа №235 імені Вячеслава Чорновола	—	09.11.2023	Не вступив (-ла)
Середня загальноосвітня школа №235 імені Вячеслава Чорновола	—	09.11.2023	Не вступив (-ла)
Середня загальноосвітня школа №235 імені Вячеслава Чорновола	—	09.11.2023	Не вступив (-ла)

Рис. 3.5. Адмін-панель: inline-редагування статусу вступу в таблиці заявок

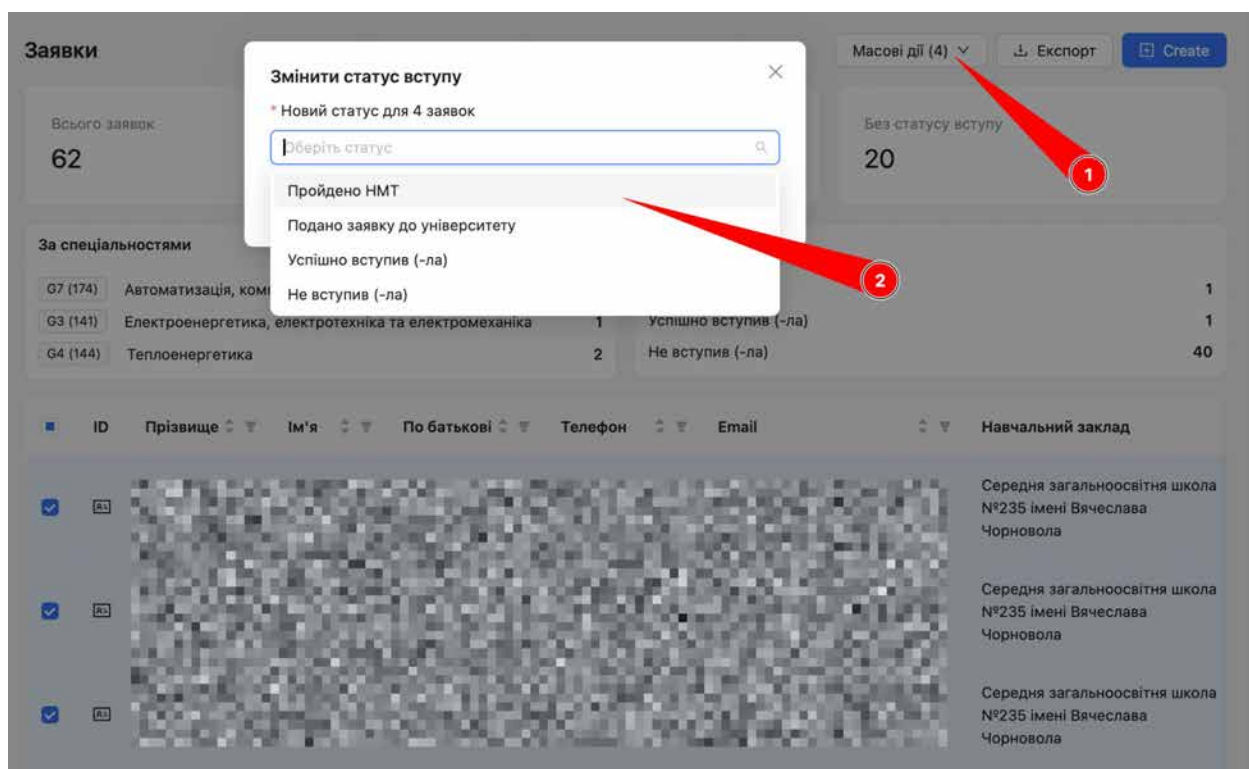


Рис. 3.6. Адмін-панель: масове оновлення вибраних заявок

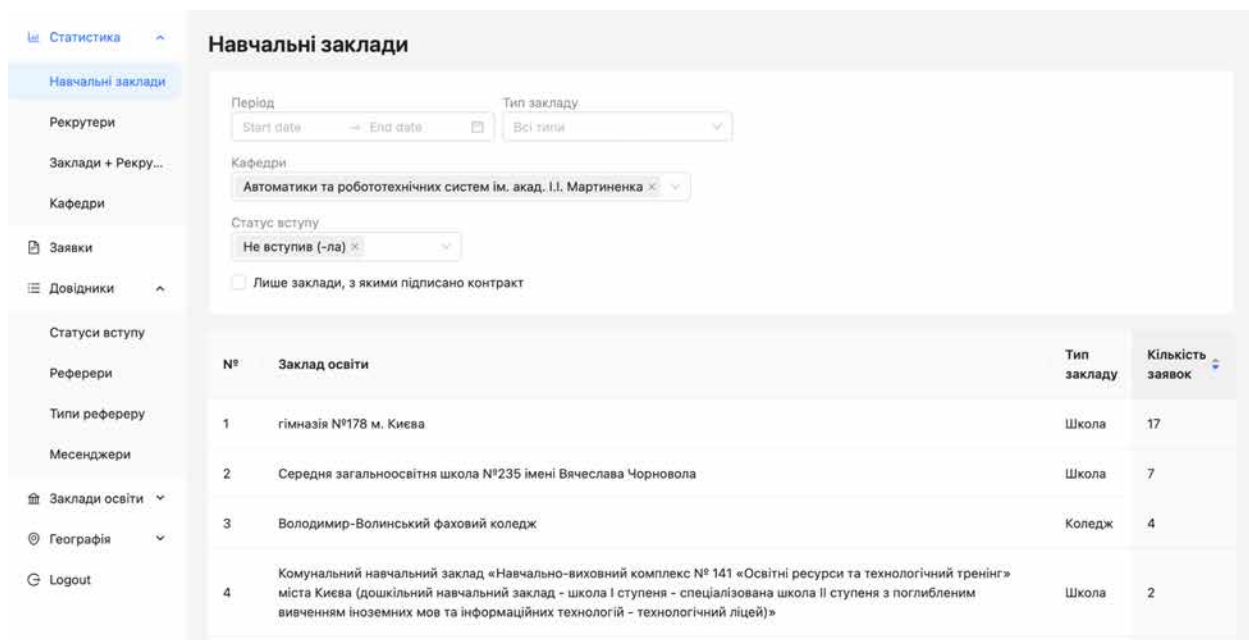


Рис. 3.7. Адмін-панель: статистичний звіт із застосованими фільтрами

ID	Код	Прізвище	Ім'я	По батькові	Телефон	Email	Навчальний заклад
1ff11ea9-2553-4095-81d4-1515e895f05a							Комунальний заклад освіти "навчально-виховний к
31a15ac6-2e78-42f8-ad22-1187c6525cb2							середня загальноосвітня школа №221 м. Києва
c894f912-c0e8-4fa6-a552-a887e047ba24							Бучанська спеціалізована загальноосвітня школа І-ІІ
65737754-6448-4ad3-9fc0-88598dd1197e							Бучанська спеціалізована загальноосвітня школа І-ІІ
5c77ab63-1c1d-45a7-97d2-5675d99b17f8							Бучанська спеціалізована загальноосвітня школа І-ІІ
b993f42c-07b4-4cf2-9e76-245b34c13a43							Бучанська спеціалізована загальноосвітня школа І-ІІ
eba6ee91-03bd-4382-8dd6-0f3dce17a798							Бучанська спеціалізована загальноосвітня школа І-ІІ
53255340-e099-4f98-b1e8-4771daeca360							Бучанська спеціалізована загальноосвітня школа І-ІІ
5ad0b2f7-394f-455b-95e5-4551a126c9a6							Бучанська спеціалізована загальноосвітня школа І-ІІ
d9c9d2c7-75a5-46af-91ee-f0d627fb7ce5							Бучанська спеціалізована загальноосвітня школа І-ІІ
69ea9986-d7fe-43f5-9e98-a3c3610af321							Гайворонський політехнічний фаховий коледж
4ec5fec2-d008-4b45-aafc-1e5cc503114f							Гайворонський політехнічний фаховий коледж
308d9a88-93ce-43fa-84eb-eefc59141e03							Блиставицький заклад загальної середньої освіти І-ІІІ
630915dd-ea36-4311-be25-23e5c9fb30f6a							Блиставицький заклад загальної середньої освіти І-ІІІ
95b15561-f593-4e88-ab3b-3b26c0e07e50							Блиставицький заклад загальної середньої освіти І-ІІІ
8c71b43a-877f-44b6-a195-d0410799fbbd							Бабинська загальноосвітня школа І-ІІІ ступенів
48c44876-fd11-40c4-b710-12371936fb96							Мирогородський аграрний фаховий коледж
50485433-475a-4385-995e-89aa9ab48f8d							Дібрівський аграрний ліцей Тетіївської міської ради
6bb6111b-dbf6-46b4-b5c0-b33fa2a16778							Таращанський навчально-виховний комплекс "Гімназія"

Рис. 3.8. Результат Excel-експорту даних заявок

Тестування безпеки

Результати тестування безпеки наведено у таблиці 3.1.

Таблиця 3.1

Результати тестування безпеки системи

Сценарій	Очікуваний результат	Фактичний результат
Запит без токена	HTTP 401	HTTP 401 ✓
Прострочений access token	HTTP 401	HTTP 401 ✓
НПП запитує заявку іншої кафедри	HTTP 403	HTTP 403 ✓
НПП відв'язує рекрутера через bulk-update	HTTP 403	HTTP 403 ✓
НПП виконує CRUD довідників	HTTP 403	HTTP 403 ✓
НПП переглядає журнал аудиту	HTTP 403	HTTP 403 ✓

Продовження таблиці 3.1

Редагування після 15-хвилинного вікна	HTTP 403	HTTP 403 ✓
Bulk-update із чужим ID заявки	HTTP 403	HTTP 403 ✓

Усі вісім сценаріїв дали результат, що збігається з очікуваним, підтверджуючи коректність дворівневого RBAC через RolesGuard та Voter-патерн.

Модульне тестування

Автоматизовані e2e-тести реалізовано з використанням Jest [54] для обох мікросервісів у директоріях `apps/admin/test/` та `apps/applicant/test/`. Запуск тестів виконується автоматично та займає кілька секунд, тоді як повне ручне тестування аналогічного обсягу сценаріїв займає значно більше часу (рис. 3.9). Поєднання обох видів тестування забезпечує перевірку як коректності бізнес-логіки, так і якості користувацького досвіду [15].

```

> js-jest@1.0.0 test
> jest --coverage

PASS Ecommerce ./string-format.test.js
  all string formats work as expected
    ✓ truncates a string correctly (5 ms)

(node:1088333) ExperimentalWarning: The Fetch API is an experimental feature. This feature could change at any time
(Use "node --trace-warnings ..." to show where the warning was created)
PASS Ecommerce ./functions.test.js
  ✓ functions are called multiple times correctly (10 ms)

PASS Ecommerce ./apis.test.js
  ✓ gets a todo object with the right properties (712 ms)

-----
File      % Stmts   % Branch  % Funcs   % Lines  Uncovered Line #s
-----
All files    60      57.14     100     84.61
  apis.js    100      100      100     100
  functions.js 83.33      50      100     100    2
  string-format.js 71.42      60      100     71.42    7,12
-----

Test Suites: 3 passed, 3 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        1.843 s, estimated 2 s
Ran all test suites.

```

Рис. 3.9. Результати автоматизованого тестування Jest

За результатами тестування підтверджено коректну роботу всіх функціональних модулів системи та відповідність визначеним вимогам безпеки.

Висновки до розділу 3

У третьому розділі проведено комплексну перевірку системи.

Функціональне тестування підтвердило коректну роботу порталу абітурієнта та адмін-панелі за вісьмома сценаріями. Тестування безпеки за матрицею з восьми сценаріїв підтвердило коректність дворівневого RBAC — усі спроби несанкціонованого доступу відхилялися з очікуваними HTTP-кодами.

РОЗДІЛ 4

ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ

4.1. Оцінка ефективності системи

Масштаб проблеми та економія часу

За даними МОН України, у 2022 році загальна кількість випускників закладів вищої освіти становила 63 671 особу [34]. ННІ НУБіП приймає близько 200 заявок на рік. Витрати часу НПП на ручну обробку даних у попередній системі розподілялися так: формування статистичних звітів — близько 20 годин на рік, перевірка та перегляд заявок — близько 17 годин, пошук дублікатів — близько 5 годин. Сукупно це становило приблизно 42 години на рік, що еквівалентно більш ніж п'яти робочим дням [24].

Порівняльний аналіз систем наведено у таблиці 4.1.

Таблиця 4.1

Порівняльний аналіз Google Forms та системи Uniply

Критерій	Google Forms	Uniply
Час заповнення заявки	~10 хвилин	~5 хвилин
Відсоток незавершених заявок	~38%	~12%
Формування статистики (200 заявок)	~20 год вручну	0 хвилин
Структуровані дані про заклад	Довільний текст	Вибір з довідника
Розмежування доступу за кафедрами	×	✓
Виявлення підозрілих заявок	×	Автоматично
Конверсійна аналітика рекрутерів	×	2 коефіцієнти
Масові дії над заявками	×	До 1000 заявок

Впровадження системи забезпечує економію близько 39 годин праці НПП на рік (табл. 4.2).

Таблиця 4.2

Порівняльна оцінка витрат часу НПП

Категорія	До (год/рік)	Після (год/рік)	Економія
Статистична обробка	~20	0	~20 год
Перегляд заявок	~17	~3	~14 год
Виявлення дублікатів	~5	~0	~5 год
Всього	~42	~3	~39 год

Наукова складова

Система реалізує чотири методи інтелектуальної обробки даних. **Нормалізація введення** — вибір навчального закладу зі структурованого довідника усуває проблему довільного тексту та унеможливорює дублювання записів [24]. **Класифікація на основі правил** — алгоритм підрахунку балів підозрілості відносить кожну заявку до одного з трьох класів. На відміну від машинного навчання, такий підхід є повністю прозорим: поле **reason []** містить конкретні причини класифікації, а адміністратор може спростувати результат через **isRefuted** [21, 2]. **Описова статистика** — п'ять звітів реалізують підрахунок, групування та ранжування, а звіт «Рекрутери» обчислює два коефіцієнти конверсії як ключові метрики ефективності [20]. **Кластеризація** — звіт «Кафедри» здійснює організаційну кластеризацію із розбивкою за типом закладу, звіт «Навчальні заклади» — географічну, що виявляє регіональні патерни походження абітурієнтів [21].

Економічна оцінка

Прямий економічний ефект від впровадження для ННІ НУБіП при середній погодинній ставці НПП 150 UAH становить:

$$39 \text{ год} \times 150 \text{ UAH/год} = \mathbf{5\ 850 \text{ UAH/рік}}$$

При масштабуванні на 16 навчально-наукових інститутів та факультетів НУБіП потенційна економія становить 93 600 UAH/рік. Оцінка ефективності пов'язана з показником рентабельності власного капіталу (ROE): оскільки система зменшує витрати при незмінному результаті діяльності, ROE університету як організації покращується [56].

За комерційною моделлю SaaS-ліцензування (\$40/місяць на університет, витрати на хостинг \$8/місяць) при базі з 20 університетів чистий прибуток становить близько \$4 680 на рік. Чиста приведена вартість за формулою 4.1 є позитивною, що підтверджує економічну доцільність комерційного розвитку системи.

$$NPV = \sum_{t=1}^5 \frac{CF_t}{(1+r)^t} - IC$$

де CF = \$4 680 — річний грошовий потік;

r = 0,15 — ставка дисконтування;

t = 1..5 — рік;

IC — початкові інвестиції.

Розгортання та експлуатація

Систему розгорнуто на VPS-сервері із Docker Compose, зворотним проксі Nginx, SSH-захистом із парою ключів та мережевим екраном UFW. Моніторинг доступності здійснюється через Uptime Kuma з повідомленнями у Telegram [55]. Резервне копіювання бази даних виконується щотижня. Система доступна 24/7 та коректно відображається у сучасних браузерях на десктопних та мобільних пристроях. На момент проведення оцінки в системі зареєстровано 509 заявок абітурієнтів.

Висновки до розділу 4

У четвертому розділі проведено оцінку ефективності системи.

Впровадження системи забезпечує економію близько 39 годин праці НПП на рік. Застосовано чотири методи інтелектуальної обробки даних: нормалізацію введення, класифікацію на основі правил, описову статистику з коефіцієнтами конверсії та кластеризацію. Прямий економічний ефект для ННІ становить 5 850 UAH/рік, розрахунок NPV підтвердив доцільність комерційного розвитку.

Систему розгорнуто та введено в експлуатацію. На момент оцінки зареєстровано більше 500 заявок абітурієнтів, що підтверджує практичну цінність розробленого рішення.

ВИСНОВКИ

У кваліфікаційній магістерській роботі відповідно до завдання розроблено автоматизовану систему збору, обробки та аналізу даних абітурієнтів навчально-наукового інституту енергетики, автоматики і енергозбереження НУБіП України. Мету роботи — розробку системи, що автоматизує вступну кампанію та забезпечує підтримку прийняття управлінських рішень на основі аналізу даних — досягнуто в повному обсязі.

При виконанні кваліфікаційної магістерської роботи отримано такі результати:

1. Проведено аналіз існуючих рішень — Google Forms, Jotform, Paperform та Вступ.ОСВІТА.UA. Встановлено, що жодне з них не відповідає комплексним потребам ННІ НУБіП: відсутні структуровані довідники навчальних закладів, рольове розмежування доступу за кафедрами та автоматизована статистична звітність. Виявлено, що попередня система на основі Google Forms спричиняла відсоток незавершених заявок близько 38% та вимагала до 20 годин ручної статистичної обробки на рік.
2. Обрано та обґрунтовано технологічний стек системи на основі порівняльного аналізу альтернатив: TypeScript та NestJS для серверної частини, MikroORM з Unit of Work патерном як засіб об'єктно-реляційного відображення, PostgreSQL із підтримкою JSONB як система керування базами даних, React разом із refine.dev та Ant Design для клієнтських застосунків, Docker Compose, Nginx та Cloudflare Pages для інфраструктури розгортання.
3. Проведено аналіз архітектурних підходів та обрано модульний мікросервісний підхід: два окремих NestJS-застосунки зі спільною базою даних PostgreSQL. Такий підхід забезпечує ізоляцію доменів на рівні кодової бази за відсутності складності міжсервісної комунікації та є оптимальним для команди одного розробника.

4. Сформульовано п'ять груп вимог до системи: вимоги до інтерфейсу порталу абітурієнта та адмін-панелі, функціональні вимоги в розрізі двох мікросервісів, вимоги до безпеки із застосуванням Argon2 та JWT, операційні вимоги щодо доступності та резервного копіювання, а також вимоги до аналізу даних.
5. Спроектовано архітектуру системи з двома мікросервісами — Admin Service та Applicant Service, — зворотним проксі Nginx для маршрутизації трафіку, розгортанням через Docker Compose та хостингом клієнтських застосунків на Cloudflare Pages. Реалізовано матрицю рольового доступу для трьох ролей (НПП, системний адміністратор, начальство) та дві незалежні схеми JWT-автентифікації з ротацією refresh-токенів для абітурієнтів і адміністраторів.
6. Спроектовано логічну модель даних із 17 сутностями, що охоплюють усі предметні галузі системи: заявки абітурієнтів, географічні довідники, довідники навчальних закладів, рекрутерів та спеціальностей, адміністративні сутності. Обґрунтовано використання механізму м'якого видалення для збереження історичних даних та JSONB-полів для зберігання показника підозрілості, додаткових відомостей про заявку та метаданих видалення.
7. Реалізовано архітектурний патерн Action та Voter, що забезпечує принцип єдиної відповідальності та авторизацію на рівні окремих записів бази даних. Розроблено алгоритм обчислення показника підозрілості заявки за вісьмома критеріями з трьома рівнями підозрілості — нормальна, помірно підозріла та високо підозріла — а також алгоритм 15-хвилинного вікна редагування заявки абітурієнтом.
8. Реалізовано дворівневий рольовий контроль доступу: RolesGuard перевіряє право на тип операції на рівні маршруту, Voter - патерн здійснює авторизацію на

рівні конкретного запису через ланцюжок Application

→ `referrer` → `department`. Декоратор

`@DepartmentScope()` автоматично інjektує

кафедральне обмеження до всіх запитів для ролі НПП, унеможливаючи доступ до даних інших кафедр навіть при прямому зверненні до API.

9. Реалізовано п'ять статистичних звітів на основі PostgreSQL умовних агрегатів із фільтрацією за довільним діапазоном дат, кафедрою, типом навчального закладу та статусом вступу. Звіт "Рекрутери" обчислює два коефіцієнти конверсії — "привів → подав документи" та "подав → вступив" — що є ключовими метриками ефективності профорієнтаційної роботи НПП.
10. Реалізовано функціональну таблицю заявок із сортуванням, фільтрацією, текстовим пошуком, вибором через чекбокси, безпечним масовим оновленням до 1000 заявок за принципом fail-closed, inline-редагуванням статусу вступу та примітки без переходу на окрему сторінку, та експортом у Excel. Реалізовано алгоритм об'єднання записів рекрутерів-дублікатів із автоматичним переприв'язуванням пов'язаних заявок.
11. Проведено функціональне тестування системи за вісьмома сценаріями для порталу абітурієнта та адмін-панелі, тестування безпеки за матрицею з восьми сценаріїв та тестування валідації вхідних даних за шістьма граничними випадками. Усі перевірені сценарії підтвердили коректність роботи системи та відповідність визначеним вимогам.
12. Встановлено, що впровадження системи забезпечує економію близько 39 годин праці НПП на рік: 20 годин завдяки автоматизації статистичної

обробки, 14 годин завдяки структурованому перегляду заявок та 5 годин завдяки автоматичному виявленню підозрілих записів. Застосовані методи інтелектуальної обробки даних — класифікація на основі правил, описова статистика з обчисленням коефіцієнтів конверсії та кластеризація — формують наукову складову роботи.

13. Систему розгорнуто та введено в експлуатацію на VPS-сервері із використанням Docker Compose, Nginx, SSH-захистом із парою ключів, мережевим екраном UFW та моніторингом доступності через Uptime Kuma з повідомленнями в Telegram. На момент проведення оцінки в системі зареєстровано 509 заявок абітурієнтів.
14. Прямий економічний ефект від впровадження системи для ННІ НУБіП становить близько 5 850 UAH на рік, потенційна економія при масштабуванні на рівень НУБіП — 93 600 UAH на рік. Розрахунок чистої приведеної вартості за 5-річний горизонт підтвердив економічну доцільність комерційного поширення системи.

Розроблена система відповідає всім сформульованим вимогам та забезпечує якісно новий рівень автоматизації вступної кампанії ННІ НУБіП порівняно з попередньою практикою. Практична цінність системи підтверджена реальним впровадженням: система функціонує в умовах реальної вступної кампанії та накопичила дані про 509 абітурієнтів. Перспективи подальшого розвитку охоплюють інтеграцію з Єдиною державною електронною базою з питань освіти, застосування методів машинного навчання для предиктивної аналітики конверсії абітурієнтів, розробку Telegram-бота для абітурієнтів та розширення системи на інші інститути і університети України.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Система Uniply

1. Uniply — автоматизована система збору та аналізу даних абітурієнтів : веб-сайт. URL: <https://uniply.app> (дата звернення: 20.05.2026).

Наукові статті у журналах (Scopus / Web of Science)

2. Assiri B., Bashraheel M., Alsuri A. Enhanced Student Admission Procedures at Universities Using Data Mining and Machine Learning Techniques. *Applied Sciences*. 2024. Vol. 14. No. 3. P. 1109. DOI: [10.3390/app14031109](https://doi.org/10.3390/app14031109).
3. Di Francesco P., Lago P., Malavolta I. Architecting with Microservices: A Systematic Mapping Study. *Journal of Systems and Software*. 2019. Vol. 150. P. 77–97. DOI: [10.1016/j.jss.2019.01.001](https://doi.org/10.1016/j.jss.2019.01.001).
4. Avritzer A., Ferme V., Janes A., Russo B., van Hoorn A., Schulz H., Menasché D. S., Rufino V. Scalability Assessment of Microservice Architecture Deployment Configurations: A Domain-based Approach Leveraging Operational Profiles and Load Tests. *Journal of Systems and Software*. 2020. Vol. 165. P. 110564. DOI: [10.1016/j.jss.2020.110564](https://doi.org/10.1016/j.jss.2020.110564).
5. Karabey Aksakalli I., Çelik T., Can A. B., Tekinerdogan B. Deployment and Communication Patterns in Microservice Architectures: A Systematic Literature Review. *Journal of Systems and Software*. 2021. Vol. 180. P. 111014. DOI: [10.1016/j.jss.2021.111014](https://doi.org/10.1016/j.jss.2021.111014).
6. Le H.-T., Shar L. K., Bianculli D., Briand L. C., Nguyen C. D. Automated Reverse Engineering of Role-based Access Control Policies of Web Applications. *Journal of Systems and Software*. 2022. Vol. 184. P. 111109. DOI: [10.1016/j.jss.2021.111109](https://doi.org/10.1016/j.jss.2021.111109).
7. Logrippo L. Data Flow Security in Role-based Access Control. *Journal of Information Security and Applications*. 2025. Vol. 90. P. 103997. DOI: [10.1016/j.jisa.2025.103997](https://doi.org/10.1016/j.jisa.2025.103997).

8. Bareedu Y. S., Frühwirth T., Niedermeier C., Sabou M., Steindl G., Thuluva A. S. et al. Deriving Semantic Validation Rules from Industrial Standards: An OPC UA Study. *Semantic Web*. 2024. Vol. 15. No. 2. P. 517–554. DOI: [10.3233/SW-233342](https://doi.org/10.3233/SW-233342).
9. Caballero I., Gualo F., Rodríguez M., Piattini M. BR4DQ: A Methodology for Grouping Business Rules for Data Quality Evaluation. *Information Systems*. 2022. Vol. 109. P. 102058. DOI: [10.1016/j.is.2022.102058](https://doi.org/10.1016/j.is.2022.102058).
10. Wang W. Model Construction and Research on Decision Support System for Education Management Based on Data Mining. *Computational Intelligence and Neuroscience*. 2021. Vol. 2021. P. 9056947. DOI: [10.1155/2021/9056947](https://doi.org/10.1155/2021/9056947).
11. Curran F. C., Carlo S., Harris-Walls K. Making the Data Visible: A Systematic Review of Systems-Level Data Dashboards for Leadership and Policy in Education. *Review of Educational Research*. 2024. Vol. 90. No. 1. P. 6–23. DOI: [10.3102/00346543241288249](https://doi.org/10.3102/00346543241288249).
12. Waseem M., Liang P., Shahin M. A Systematic Mapping Study on Microservices Architecture in DevOps. *Journal of Systems and Software*. 2020. Vol. 170. P. 110798. DOI: [10.1016/j.jss.2020.110798](https://doi.org/10.1016/j.jss.2020.110798).
13. Waseem M., Liang P., Shahin M., Di Salle A., Márquez G. Design, Monitoring, and Testing of Microservices Systems: The Practitioners' Perspective. *Journal of Systems and Software*. 2021. Vol. 182. P. 111061. DOI: [10.1016/j.jss.2021.111061](https://doi.org/10.1016/j.jss.2021.111061).

Тези конференцій

14. Новак Б. В. Автоматизована система аналізу даних та підтримки прийняття рішень щодо абітурієнтів ННІ енергетики, автоматики і енергозбереження / Б. В. Новак ; наук. кер. А. О. Дудник // Енергозабезпечення, електротехнології, електротехніка та інтелектуальні управляючі системи в АПК : тези доп. 79-ї наук.-практ. конф. студентів. Київ : НУБіП України, 2026. С. 182–183.

- 15.Василевський В. О., Романюк О. В. Недоліки використання модульного тестування як основної технології тестування // *Інформаційно-комп'ютерні технології – 2020* : тези доп. XI міжнар. наук.-техн. конф. Житомир : Житомирська політехніка, 2020. 252 с.

Навчальна література та монографії

- 16.Юринець В. Є., Юринець Р. В. Автоматизовані інформаційні системи : електронний посібник. 694 с.
- 17.Інформаційні системи і технології / С. Г. Карпенко та ін. 189 с.
- 18.Elliott E. Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Modern JS Libraries. Sebastopol : O'Reilly Media, 2014. 254 p.
- 19.Silva R. Essential Postgres. 2020.
- 20.Field A. Discovering Statistics Using IBM SPSS Statistics. 6th ed. London : SAGE Publications Ltd, 2024. 1144 p.
- 21.Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. 3rd ed. Burlington : Morgan Kaufmann, 2011. 703 p.
- 22.Завгородній В. В., Ялова К. М. Конспект лекцій з дисципліни «Архітектура та проектування програмного забезпечення». Кам'янське : ДДТУ, 2019. 144 с.
- 23.Цибульник С. О., Барандич К. С. Технології розроблення програмного забезпечення. Ч. 1. Життєвий цикл програмного забезпечення. Київ : КПІ ім. Ігоря Сікорського, 2022. 270 с.
- 24.Білик В. М. Інформаційні технології та системи : навч. посібник. 2006. 233 с.
- 25.Дідковська М. В., Тимошенко Ю. О. Тестування: основні визначення, аксіоми та принципи : текст лекцій. Ч. 1. Київ : КПІ, 2010. 62 с.
- 26.Старух А. І. Конспект лекцій з навчальної дисципліни «Методологія тестування програмного забезпечення». Львів : ЛНУ ім. Івана Франка, 2020. 3 с.

- 27.Редько Н. Н. та ін. Реляційні бази даних, табличні алгебри та SQL-подібні мови. 2001. 192 с.

Стандарти та специфікації

- 28.Jones M., Bradley J., Sakimura N. JSON Web Token (JWT) : RFC 7519 / Internet Engineering Task Force. 2015. URL: <https://www.rfc-editor.org/rfc/rfc7519> (дата звернення: 20.04.2026).
- 29.Biryukov A., Dinu D., Khovratovich D. Argon2: The Memory-Hard Function for Password Hashing and Other Applications. Password Hashing Competition, 2015. URL: <https://github.com/P-H-C/phc-winner-argon2/blob/master/argon2-specs.pdf> (дата звернення: 20.04.2026).

Веб-сайти та документація

- 30.Google Forms : сервіс для створення онлайн-форм : веб-сайт. URL: <https://www.google.com/forms/about/> (дата звернення: 20.04.2026).
- 31.Jotform : платформа для створення форм : веб-сайт. URL: <https://www.jotform.com/> (дата звернення: 20.04.2026).
- 32.Paperform : сервіс для створення форм : веб-сайт. URL: <https://paperform.co/> (дата звернення: 20.04.2026).
- 33.Вступ.ОСВІТА.UA : офіційна платформа вступу до закладів вищої освіти України : веб-сайт. URL: <https://vstup.osvita.ua/> (дата звернення: 20.04.2026).
- 34.Статистика вищої освіти України. Міністерство освіти і науки України : веб-сайт. URL: <https://mon.gov.ua/ua/osvita/visha-osvita/statistika-vishhoyi-osviti> (дата звернення: 20.04.2026).
- 35.TypeScript : документація мови програмування : веб-сайт. URL: <https://www.typescriptlang.org/docs/> (дата звернення: 20.04.2026).
- 36.Node.js : документація платформи виконання JavaScript : веб-сайт. URL: <https://nodejs.org/en/docs/> (дата звернення: 20.04.2026).
- 37.NestJS : документація фреймворку для Node.js : веб-сайт. URL: <https://docs.nestjs.com/> (дата звернення: 20.04.2026).

- 38.NestJS Guards and Authorization : документація механізму авторизації : веб-сайт. URL: <https://docs.nestjs.com/guards> (дата звернення: 20.04.2026).
- 39.MikroORM : документація ORM для TypeScript : веб-сайт. URL: <https://mikro-orm.io/docs/> (дата звернення: 20.04.2026).
- 40.TypeORM : документація ORM для Node.js : веб-сайт. URL: <https://typeorm.io/> (дата звернення: 20.04.2026).
- 41.PostgreSQL Documentation. JSON Types : документація типів даних JSONB : веб-сайт. URL: <https://www.postgresql.org/docs/current/datatype-json.html> (дата звернення: 20.04.2026).
- 42.React : документація бібліотеки для побудови інтерфейсів : веб-сайт. URL: <https://react.dev/> (дата звернення: 20.04.2026).
- 43.Refine : документація фреймворку для адмін-панелей : веб-сайт. URL: <https://refine.dev/docs/> (дата звернення: 20.04.2026).
- 44.Ant Design : документація бібліотеки UI-компонентів : веб-сайт. URL: <https://ant.design/docs/react/introduce> (дата звернення: 20.04.2026).
- 45.Docker : документація платформи контейнеризації : веб-сайт. URL: <https://docs.docker.com/> (дата звернення: 20.04.2026).
- 46.Nginx : документація веб-сервера та зворотного проксі : веб-сайт. URL: <https://nginx.org/en/docs/> (дата звернення: 20.04.2026).
- 47.Nginx Reverse Proxy Guide : керівництво з налаштування зворотного проксі : веб-сайт. URL: <https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/> (дата звернення: 20.04.2026).
- 48.Cloudflare Pages : документація платформи хостингу статичних сайтів : веб-сайт. URL: <https://developers.cloudflare.com/pages/> (дата звернення: 20.04.2026).
- 49.Cloudflare : документація платформи мережевої безпеки та CDN : веб-сайт. URL: <https://developers.cloudflare.com/> (дата звернення: 20.04.2026).
- 50.date-fns : документація бібліотеки для роботи з датами : веб-сайт. URL: <https://date-fns.org/docs/> (дата звернення: 20.04.2026).

- 51.bcrypt : пакет для хешування паролів. npm : веб-сайт. URL:
<https://www.npmjs.com/package/bcrypt> (дата звернення: 20.04.2026).
- 52.ExcelJS : бібліотека для роботи з Excel-файлами : веб-сайт. URL:
<https://github.com/exceljs/exceljs> (дата звернення: 20.04.2026).
- 53.class-validator : бібліотека декораторної валідації для TypeScript : веб-сайт. URL: <https://github.com/typestack/class-validator> (дата звернення: 20.04.2026).
- 54.Jest : документація фреймворку для тестування JavaScript : веб-сайт. URL:
<https://jestjs.io/docs/getting-started> (дата звернення: 20.04.2026).
- 55.Uptime Kuma : документація системи моніторингу доступності : веб-сайт. URL: <https://github.com/louislam/uptime-kuma/wiki> (дата звернення: 20.04.2026).
- 56.Кобилецький В. Р. Коефіцієнт рентабельності власного капіталу. Financial Analysis online : веб-сайт. URL: <https://analizfinancial.com/slovnik-ekonomichnikh-terminiv/338-pokaznik-rentabelnosti-vlasnogo-kapitalu> (дата звернення: 20.04.2026).

ДОДАТКИ

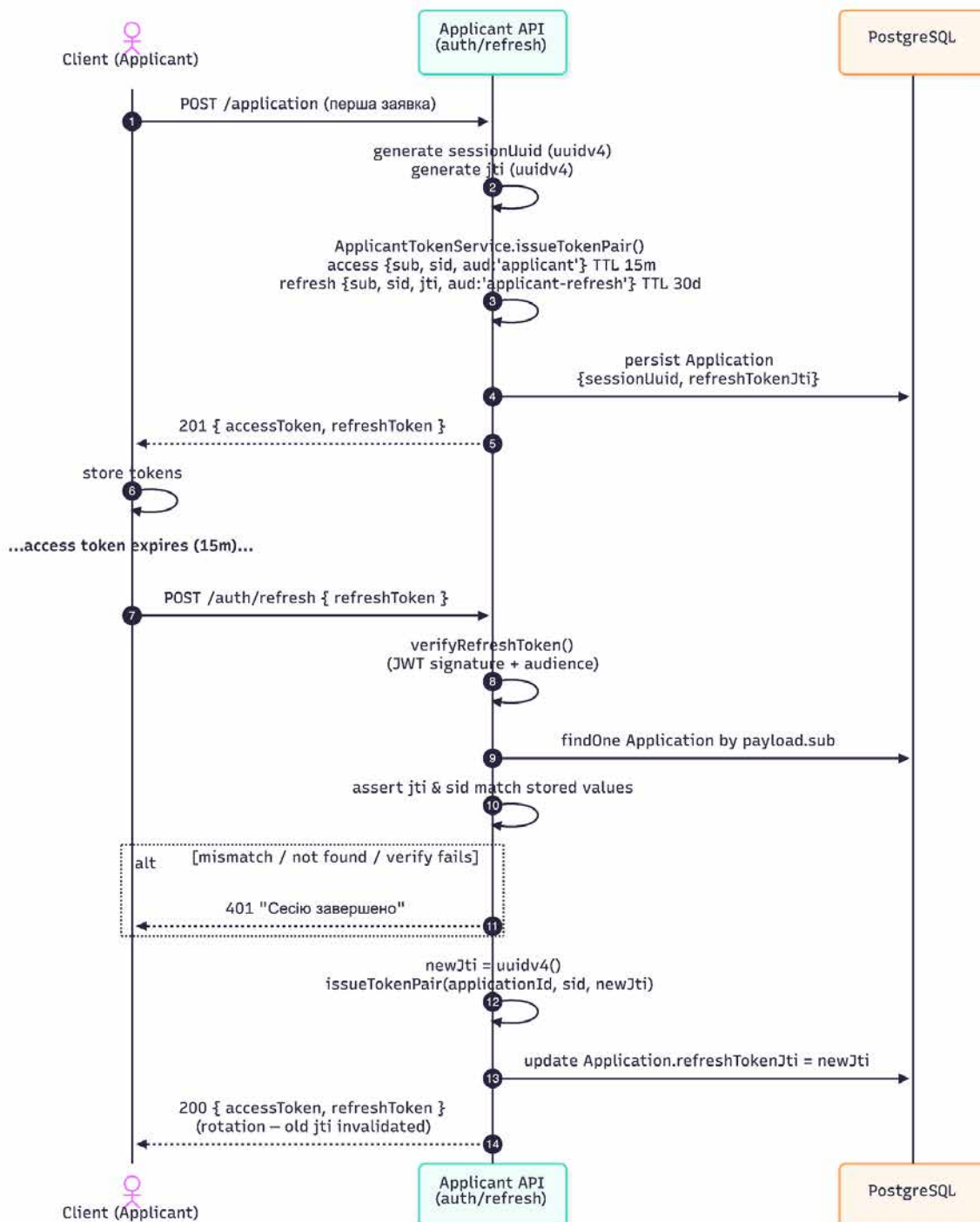
Додаток А.1

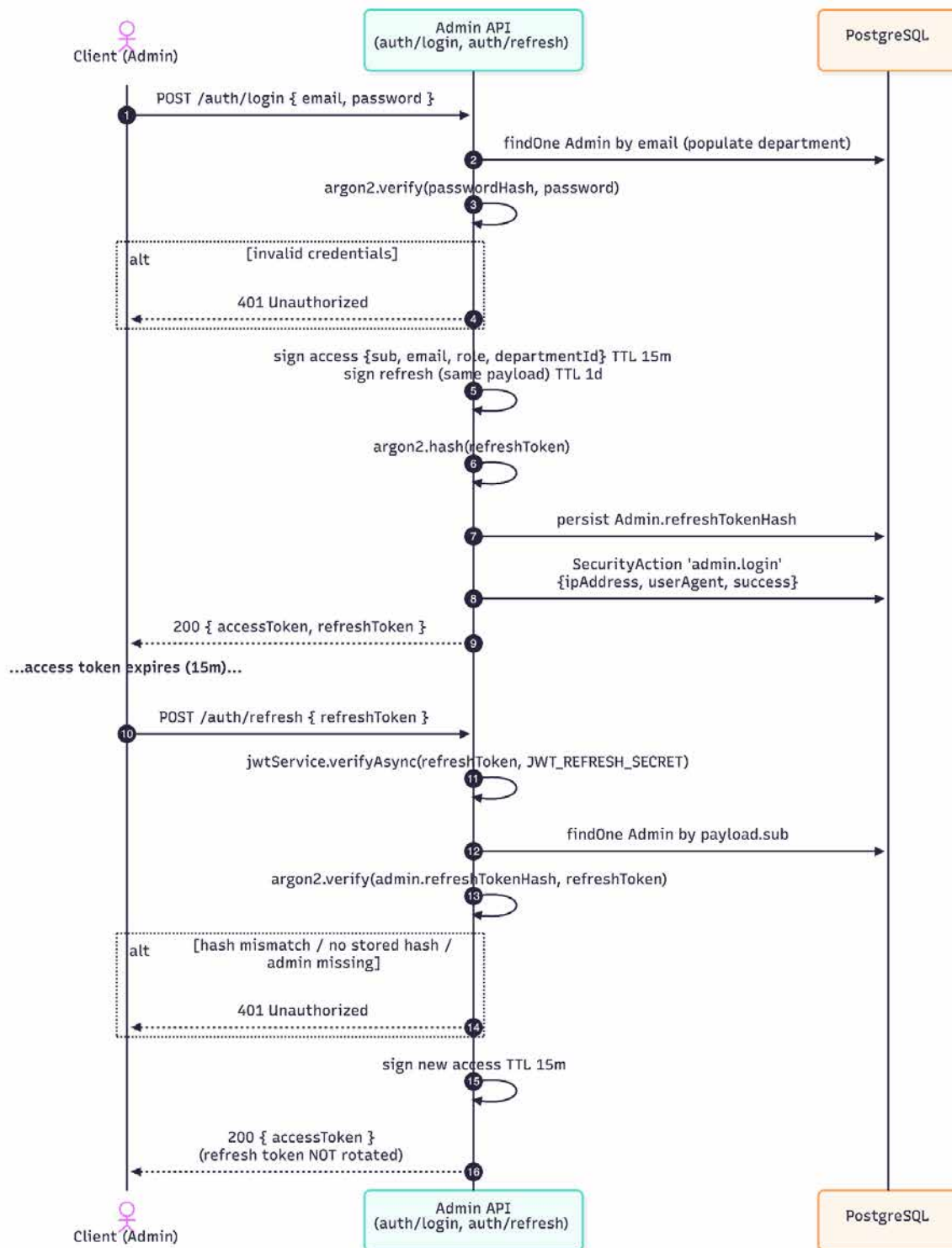
Зведена таблиця обраного технологічного стеку

Рівень системи	Обрана технологія	Ключове обґрунтування
Мова програмування	TypeScript	Статична типізація, виявлення помилок на етапі компіляції
Серверний фреймворк	NestJS	Модульність, Dependency Injection, підтримка декораторів
ORM	MikroORM	Unit of Work, Identity Map, висока підтримка TypeScript
СКБД	PostgreSQL	ACID-транзакції, підтримка JSONB, складні запити
Фронтенд (адмін-панель)	React + refine.dev + Ant Design	Data provider, готові CRUD-сторінки, UI-компоненти
Фронтенд (портал абітурієнта)	React	Компонентна архітектура, підтримка TypeScript
Контейнеризація	Docker Compose	Відтворюваність середовища, єдина конфігурація
Зворотний проксі	Nginx	Маршрутизація між мікросервісами, SSL termination
Хостинг фронтенду	Cloudflare Pages	Безкоштовний CDN, автоматичний HTTPS
Робота з датами	date-fns	Модульність, незмінність об'єктів дат

Додаток Б.1

Схема JWT-автентифікації для абітурієнтів та адміністраторів





Додаток Б.2

Приклади SQL-запитів для зв'язків між сутностями бази даних

Заявка — Навчальний заклад (ManyToOne)

Одна заявка може бути пов'язана з одним навчальним закладом; один навчальний заклад може мати багато заявок. Зовнішній ключ `school_id` у таблиці `application` посилається на первинний ключ таблиці `school`.

```
SELECT a.last_name, s.name AS school_name
FROM application a
JOIN school s ON a.school_id = s.id
WHERE a.id = '...';
```

Заявка — Спеціальність (ManyToMany)

Один абітурієнт може подавати заявку на кілька спеціальностей; одна спеціальність може фігурувати в багатьох заявках. Зв'язок реалізований через зведену таблицю `application_specialties_specialty` з колонками `application_id` та `specialty_id`.

```
SELECT a.last_name, sp.name AS specialty_name
FROM application a
JOIN application_specialties_specialty aps ON a.id = aps.application_id
JOIN specialty sp ON aps.specialty_id = sp.id
WHERE a.id = '...';
```

Заявка — Рекрутер (ManyToOne)

Одна заявка може бути пов'язана з одним рекрутером; один рекрутер може фігурувати в багатьох заявках. Зовнішній ключ `referrer_id` у таблиці `application` посилається на первинний ключ таблиці `referrer`.

```
SELECT a.last_name, r.name AS referrer_name
FROM application a
JOIN referrer r ON a.referrer_id = r.id
WHERE a.id = '...';
```

Населений пункт — Область (ManyToOne)

Один населений пункт належить до однієї області; одна область може мати багато населених пунктів. Зовнішній ключ `region_id` у таблиці `settlement` посиляється на первинний ключ таблиці `region`.

```
SELECT s.name AS settlement_name, r.name AS region_name
FROM settlement s
JOIN region r ON s.region_id = r.id
WHERE s.id = '...';
```

Навчальний заклад — Населений пункт (ManyToOne)

Один навчальний заклад розташований в одному населеному пункті; один населений пункт може мати багато навчальних закладів. Зовнішній ключ `settlement_id` у таблиці `school` посиляється на первинний ключ таблиці `settlement`.

```
SELECT sc.name AS school_name, se.name AS settlement_name
FROM school sc
JOIN settlement se ON sc.settlement_id = se.id
WHERE sc.id = '...';
```

Додаток Б.3

Зведена таблиця сутностей логічної моделі даних

Сутність	Базовий клас	Ключові зв'язки
Заявка	SoftDeletable	Навчальний заклад, Спеціальності (M2M), Месенджери (M2M), Рекрутер, Прогрес вступу
Адміністратор	SoftDeletable	Кафедра
Кафедра	SoftDeletable	Адміністратори, Рекрутери
Рекрутер	SoftDeletable	Кафедра, Тип рекрутера
Тип рекрутера	Base	Рекрутери
Навчальний заклад	SoftDeletable	Тип закладу, Тип освіти закладу, Тип фінансування закладу, Населений пункт
Тип закладу	Base	Навчальні заклади, Типи освіти закладів
Тип освіти закладу	SoftDeletable	Навчальні заклади, Тип закладу
Тип фінансування закладу	SoftDeletable	Навчальні заклади
Населений пункт	SoftDeletable	Область, Район, Тип населеного пункту, Навчальні заклади
Тип населеного пункту	Base	Населені пункти
Область	SoftDeletable	Населені пункти, Райони
Район	SoftDeletable	Область, Населені пункти
Спеціальність	SoftDeletable	Заявки (M2M)

Месенджер	Base	Заявки (M2M)
-----------	------	--------------

Додаток Б.4

Модулі Admin Service

Модуль	Відповідальність
AuthModule	Автентифікація адміністраторів, JWT-стратегії
ApplicationModule	CRUD заявок, масове оновлення, експорт
ApplicationAdmissionProgressModule	CRUD довідника прогресу вступу
StatisticsModule	П'ять статистичних звітів
AdminCrudModule	CRUD адміністраторів
AdminUserModule	Отримання поточного адміністратора
ReferrerModule	CRUD рекрутерів, об'єднання, підрахунок заявок
ReferrerTypeModule	CRUD типів рекрутерів
DepartmentModule	CRUD кафедр
SchoolModule	CRUD навчальних закладів
SchoolTypeModule	CRUD типів закладів
SchoolEducationTypeModule	CRUD типів освіти закладів
SchoolFinancingTypeModule	CRUD типів фінансування закладів
SettlementModule	CRUD населених пунктів
SettlementTypeModule	CRUD типів населених пунктів
RegionModule	CRUD областей

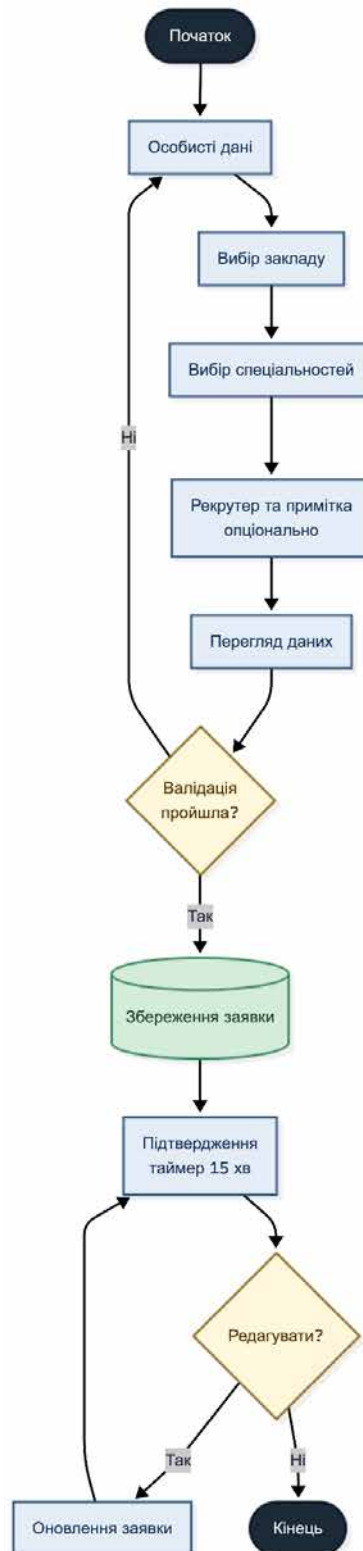
DistrictRegionModule	CRUD районів
SpecialtyModule	CRUD спеціальностей
MessengerModule	CRUD месенджерів
CommonModule	Пагінація, контекст запиту, middleware

Модулі Applicant Service

Модуль	Відповідальність
ApplicationModule	Подача заявки абітурієнтом
DirModule	Довідникові дані для форми (регіони, школи, рекрутери тощо)

Додаток Б.5

Діаграма діяльності процесу подачі заявки абітурієнтом



Додаток В.1

Результати тестування валідації вхідних даних

№	Поле	Некоректне значення	Очікувана помилка	Результат
1	email	"not-an-email"	Невалідний формат електронної пошти	✓
2	phone	"абвгд"	Некоректний формат номера телефону	✓
3	firstName	(порожній рядок)	Поле не може бути порожнім	✓
4	specialtyIds	[] (порожній масив)	Масив має містити щонайменше 1 елемент	✓
5	ids (bulk-update)	масив із 1001 елементом	Масив не може містити більше 1000 елементів	✓
6	id (будь-яке поле)	"not-a-uuid"	Некоректний формат UUID	✓