

Київський національний університет імені Тараса Шевченка

Факультет комп'ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

ЗВІТ
з виробничої практики

Виконав:

студент 3-го курсу

Вербицький Артем Віталійович

Керівник практики:

Жереб К. А.

Київ – 2025

Зміст

Вступ.....	3
Робота над проєктом “Confero”.....	4
Висновки	7

Вступ

У межах виробничої практики я та моя команда розпочали розробку Web-платформи для структуризації інформації про органи студентського самоврядування (ОСС) Університету та автоматизації їхньої роботи. Наша команда складається з п'ятьох студентів, кожен із яких відповідає за певний напрям розробки. Моя роль у проєкті – створення серверної частини (backend).

Метою розробки є впорядкування роботи студентського самоврядування, спрощення документообігу, організації голосувань, комунікації між учасниками та інших ключових процесів. Ми прагнемо створити зручний, ефективний та безпечний інструмент для автоматизації управлінських завдань.

У процесі роботи ми спроектували архітектуру системи, обрали технологічний стек, розробили серверну частину, базу даних та API для взаємодії з клієнтським інтерфейсом. Окрему увагу приділили безпеці, реалізувавши механізми автентифікації та контролю доступу.

Для покращення знань та навичок у межах проєкту я ознайомився з новими технологіями та підходами до розробки. Зокрема, я вивчив додаткові можливості фреймворку NestJS, оптимізацію баз даних та принципи побудови безпечних API.

Робота над проєктом “Confero”

Мета проєкту “Confero” – розробити платформу, яка зробить знайомство студентів з ОСС Університету легшим і більш зрозумілим, а також автоматизує певні внутрішні процеси цих органів, такі як скликання засідань, голосування, документообіг тощо. Мої завдання – це робота над серверною частиною платформи (backend).

1. Технологічний стек

У ході обговорень з командою було вирішено застосувати мікросервісну архітектуру для даної платформи, адже вона найкраще відповідає поставленим завданням. Для реалізації серверної частини було обрано такий стек:

- Nest.js – фреймворк для створення сервісної частини проєкту
- Sequelize – ORM для роботи з базами даних
- PostgreSQL, MongoDB – для баз даних
- RabbitMQ – для обміну повідомленнями
- Docker – для зручної розробки

2. Ознайомлення з предметною областю

Перед початком розробки Web-платформи для органів студентського самоврядування було проведено детальне дослідження предметної області. Це допомогло визначити основні потреби користувачів, виявити ключові проблеми, що потребують автоматизації, та сформулювати вимоги до функціональності системи.

Структура студентського самоврядування

Органи студентського самоврядування в університетах відіграють важливу роль у представництві інтересів студентів, організації заходів, захисті прав та контролі за якістю освіти. Вони включають різні рівні:

- **Рівень Університету та Студмістечка** – органи самоврядування, що координують діяльність на рівні всього університету та студмістечка відповідно.
- **Рівень структурних підрозділів та гуртожитків** – студентські органи структурних підрозділів та гуртожитків що займаються локальними питаннями.

А також типи управління:

- Студентський парламент
- Студентська рада
- Конференція студентів
- Контрольно-ревізійна комісія студентів
- Центральна виборча комісія студентів
 - Окружна виборча комісія
 - Дільнична виборча комісія

Основні функції студентського самоврядування

На основі аналізу внутрішніх процесів самоврядування були визначені ключові завдання, які має підтримувати система:

- **Документообіг** – зберігання та обмін документами (протоколи, рішення, звернення).
- **Голосування та опитування** – організація виборів, прийняття рішень шляхом голосування.
- **Комунікація** – взаємодія між учасниками через оголошення.
- **Календар подій** – планування заходів, зустрічей та їх координація.
- **Аналітика та звітність** – збір статистичних даних щодо діяльності самоврядування.

Проблеми, що потребують вирішення

Аналіз існуючого підходу до управління студентським самоврядуванням виявив такі основні проблеми:

- **Відсутність єдиного цифрового простору** – інформація зберігається в різних джерелах (Google Docs, соціальні мережі, месенджери), що ускладнює доступ і координацію.
- **Складність документообігу** – паперові або неструктуровані електронні документи ускладнюють управління.
- **Низький рівень автоматизації** – багато процесів, таких як голосування або звітність, виконуються вручну.
- **Обмежена прозорість діяльності** – учасники самоврядування не завжди мають швидкий доступ до важливої інформації.

Отримані результати дослідження предметної області лягли в основу вимог до розробки Web-платформи. Запропонована система дозволить оптимізувати діяльність студентського самоврядування, покращити координацію та забезпечити ефективну взаємодію між усіма учасниками.

3. Розробка серверної частини проєкту (backend).

- Перший етап нашої роботи – **проєктування архітектури системи**. Ми прагнули створити гнучку та масштабовану платформу, тому обрали клієнт-серверну архітектуру з розподіленими мікросервісами. Backend вирішили реалізувати на **NestJS**, оскільки цей фреймворк забезпечує модульність, стабільність та гарну підтримку TypeScript.
- Для роботи з базою даних ми використали **Sequelize** – ORM (Object-Relational Mapping) для **PostgreSQL**, що дозволяє ефективно керувати моделями, виконувати запити та взаємодіяти з даними на високому рівні абстракції. Це значно спрощує розробку, забезпечує захист від SQL-ін'єкцій та дає можливість міграції схеми бази даних.
- Однією з найважливіших частин розробки була **серверна логіка**. Ми створили REST API, який обробляє запити від клієнтів, керує користувачами, документами, голосуваннями та повідомленнями. Важливим аспектом стало впровадження **автентифікації та авторизації**, оскільки безпека даних – ключова вимога нашої системи. Ми реалізували автентифікацію за допомогою **JWT (JSON Web Token)** та забезпечили контроль доступу до різних частин платформи залежно від ролей користувачів.
- Окремо варто відзначити роботу над **оптимізацією продуктивності**. Щоб забезпечити швидкість обробки запитів, ми використали **кешування через Redis**, а також продумали систему **індексів у базі даних** для прискорення запитів. Крім того, для підвищення стабільності система розгортається в **Docker-контейнерах**, що дозволяє автоматизувати оновлення та розгортання сервісів.

Висновки

У результаті виробничої практики було почато роботу над створенням масштабованої та функціональної платформи, яка вирішує основні проблеми студентського самоврядування, автоматизує документообіг, голосування, комунікацію та інші процеси.

Завдання на практику дозволили мені поглибити власні знання у галузі програмної інженерії та Web-технологій, справитися з викликами роботи у команді, застосувати власні вміння у створенні надійної архітектури проекту.

Завдання є близькими до тих, що зустрічаються на підприємствах. Нові уміння надають мені можливість впевнено почати роботу над реальними проектами; проходити технічні співбесіди, де потрібно виконувати тестові завдання; більш глибоко розвиватися у напрямі розробки Web-застосунків.

Я ретельно опрацював матеріали та опанував нові технології, покращив свої навички у проектуванні, написанні, тестуванні, деплойменті програмного забезпечення.