Міністерство освіти і науки України
Чернівецький національний університет
імені Юрія Федьковича

Кафедра математичного моделювання
(назва кафедри)

ЗВІТ

з науково-дослідної практики

(вид практики)

Студента (ки) групи ____501_____
(номер групи)

Фльори Тетяни Миколаївни
(прізвище, ім'я, по батькові)

Початок практики 1 вересня 2025 р.

Закінчення практики 29 травня 2026 р.

Керівник практики: _____ кандидат фізико-математичних наук,
доцент Лариса ПІДДУБНА
(підпис) (посада, ім'я, прізвище)

Звіт захищено з оцінкою _____

«_01_» червня 2026 р.

м. Чернівці

2026 р.

Вступ

Я проходила науково-дослідну практику з 01.09.2025 р. по 29.05.2026 р. на кафедрі математичного моделювання Чернівецького національного університету імені Юрія Федьковича. У перші дні практики я обговорила з керівником та остаточно вибрала тему своєї майбутньої кваліфікаційної роботи. Оскільки було прийнято рішення, що я продовжуватиму кваліфікаційну роботу XXXXXX IIII, наступним кроком стало наше знайомство та детальне обговорення її теми й уже проробленої роботи. Основною метою її кваліфікаційного дослідження була розробка інтерактивного вебкомплексу для проведення онлайн-квізів у реальному часі.

На перший погляд, ця задача здалася об'ємною, із складною структурою, але водночас дуже цікавою та досить складною з технічної точки зору. Оцінюючи стек технологій, який використала Марія, я зрозуміла, що фронтенд-частина написана на базі бібліотеки React - бібліотеки, яка вже була мені знайома. Проте викликом стало те, що бекенд-платформа та вся логіка були побудовані на базі JavaScript/TypeScript (фреймворк NestJS), з якими я раніше не працювала і які мені необхідно було детально вивчити. Додаткову складність викликала необхідність реалізації real-time синхронізації між багатьма користувачами, оскільки на той момент я не мала практичного досвіду роботи з WebSocket-комунікацією та архітектурою режиму реального часу. Усвідомивши недостатній рівень практичної підготовки у цьому напрямку, я визначила необхідність поглибленого вивчення відповідних технологій та архітектурних підходів. Це визначило моє першочергове завдання - підбір та опрацювання правильної науково-технічної літератури, документації та профільних джерел, щоб заповнити ці прогалини й підготувати базу для подальшого розвитку проєкту.

Обрана література:

Мотиваційна література (обґрунтування актуальності та гейміфікації):

1. **Ковтун О. В.** Інноваційні технології навчання у вищій школі: акмеологічний підхід. — К.: Видавничий дм «Кондор», 2021. — 312 с.

Вказана монографія присвячена дослідженню інтерактивних методик навчання та впровадженню цифрових інструментів у вищій школі. Матеріали праці використано для теоретичного обґрунтування актуальності розробки: інтерактивні вебкомплекси розглядаються як ефективний засіб підвищення внутрішньої мотивації студентів, подолання бар'єрів пасивності під час дистанційної або змішаної освіти, а також як інструмент пізнавальної діяльності.

2. **Zichermann G., Cunningham C.** Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps. — Sebastopol: O'Reilly Media, 2011. — 208 p.

У книзі детально аналізуються базові механіки залучення користувачів (системи балів, динамічні лідерборди, обмеження часу, рівні складності). Праця послужила основою для проєктування нових ігрових режимів вебкомплексу (зокрема, командних

квізів та режимів на виліт), оскільки в ній описуються математичні та алгоритмічні підходи до інтеграції ігрових елементів у неігрові (освітні) середовища для стимулювання когнітивної активності.

Опис технологій (технічна база):

1. **Flanagan D.** JavaScript: The Definitive Guide. 7th ed. — Sebastopol: O'Reilly Media, 2020. — 704 p.

Фундаментальне керівництво використано для детального вивчення асинхронної моделі виконання коду, роботи механізму Event Loop, промісів (Promises) та асинхронних функцій. Ці концепції є базовими для розробки високонавантаженої серверної частини, здатної обробляти тисячі одночасних запитів.

2. **NestJS Documentation: Techniques & WebSockets** [Електронний ресурс]. — Режим доступу: https://docs.nestjs.com/websockets/gateways.

Офіційна документація фреймворку NestJS опрацьована в частині архітектурної організації серверних застосунків. Особливу увагу приділено розділу побудови WebSocket-шлюзів (Gateways) на базі Socket.IO, реалізації механізмів Dependency Injection (впровадження залежностей) та розмежуванню логіки через модулі, що забезпечує масштабованість системи.

3. **Banks A., Porcello E.** Learning React: Modern Patterns for Developing Real-Time Web Apps. 2nd ed. — Sebastopol: O'Reilly Media, 2020. — 362 p.

У виданні досліджено сучасні шаблони проектування Single Page Applications (SPA). Матеріал книги допоміг структурувати знання з керування глобальним станом інтерфейсу (React Hooks, Context API), оптимізації рендерингу компонентів за допомогою Virtual DOM та організації клієнт-серверної взаємодії в режимі реального часу.

4. **TypeORM Documentation: Active Record vs Data Mapper** [Електронний ресурс]. — Режим доступу: https://typeorm.io.

Джерело опрацьовано для дослідження принципів об'єктно-реляційного відображення (ORM). Огляд документації дозволив проаналізувати ER-структуру сутностей бази даних, автоматизацію CRUD-операцій, управління міграціями та безпечне проєктування зв'язків між таблицями бази даних PostgreSQL.

5. **PostgreSQL 16.0 Documentation** [Електронний ресурс]. — Режим доступу: https://www.postgresql.org/docs/16/index.html.

Офіційні технічні специфікації СУБД PostgreSQL досліджено в розрізі оптимізації складних реляційних запитів, використання індексів для прискорення вибірки даних лідербордів та забезпечення транзакційності й цілісності даних при одночасному доступі багатьох користувачів.

6. **Docker Docs: Containerization and Docker Compose** [Електронний ресурс]. — Режим доступу: https://docs.docker.com.

Документація інструменту контейнеризації опрацьована з метою вивчення принципів

ізоляції середовища розробки. Матеріали використано для написання конфігураційних файлів Dockerfile та docker-compose.yml, що дозволяє розгортати NestJS-сервер, клієнтську частину React та базу даних PostgreSQL в єдиній локальній екосистемі за допомогою кількох команд.

Наступним кроком моєї роботи став безпосередній глибокий аналіз архітектурних рішень, реалізованих у магістерській роботі Марії Венгринюк. Ознайомившись із матеріалами її дослідження, я детально вивчила спроєктовану систему, де клієнтську частину побудовано на базі бібліотеки React, а серверну частину реалізовано як модульний NestJS-застосунок. Проаналізувавши ER-діаграму та структуру реляційної бази даних PostgreSQL, я відзначила високу надійність використання TypeORM для опису сутностей та автоматизації CRUD-операцій.

Отже, в процесі проходження практики мною було опрацьовано документацію NestJS, Socket.IO, TypeORM та PostgreSQL, досліджено принципи побудови real-time систем і механізми синхронізації стану між клієнтами через WebSocket-з'єднання. Окрему увагу було приділено вивченню модульної організації NestJS-застосунків, принципів dependency injection та контейнеризації проєкту за допомогою Docker. Окремо було досліджено принципи побудови клієнт-серверної взаємодії та особливості використання REST API у поєднанні з WebSocket-комунікацією. Також було проведено аналіз можливостей масштабування системи та сформовано власні пропозиції щодо модернізації архітектури комплексу.

Аналіз існуючих рішень та конкурентів платформи

Для чіткого розуміння аналогів та шляху подальшого вдосконалення системи, я дослідила проведений у роботі Марії аналіз існуючих платформ для проведення онлайн-вікторин. Я доповнила аналіз новими платформами та провела їх порівняння. Результати порівняльного аналізу ключових конкурентів дозволили виділити їхні основні переваги та обмеження:

- Kahoot!: одна з найпоширеніших платформ для проведення інтерактивних освітніх вікторин завдяки простоті використання, яскравому візуальному стилю та стабільній real-time динаміці. Проте його головним недоліком є платна версія, яка суттєво обмежує безкоштовний функціонал, а також закрита архітектура, що унеможливлює кастомізацію або локальну інтеграцію з університетськими системами.
- Quizizz: орієнтований на індивідуальний темп виконання завдань та асинхронні режими роботи, пропонуючи гнучке використання медіаматеріалів. Проте він так само є платним продуктом із жорстко обмеженими рамками, встановленими розробниками, без можливості зміни ігрової логіки.
- Blooket: представник нового покоління освітніх систем, що повністю змінює класичну концепцію вікторин. Замість простого вибору варіантів відповідей, правильні дії гравця відкривають доступ до різноманітних real-time мікроігор (стратегій, симуляторів, Tower Defense), які проходять синхронно в межах однієї кімнати. Цей сервіс аналізується як приклад максимального рівня гейміфікації навчального процесу та досліджується з точки зору навантаження на мережеві протоколи, оскільки сервер має безперервно синхронізувати

складні ігрові стани між десятками пристроїв одночасно.

Порівняння з цими платформами підтвердило високу практичну цінність розробленого вебкомплексу. Проєкт Марії вдало запозичив найкращі real-time механіки Kahoot!, але при цьому отримав відкриту та модульну архітектуру, готову до безкоштовного масштабування та модифікації.

Напрямки вдосконалення та нові види розвитку системи

Враховуючи виявлені технічні обмеження, досвід провідних платформ (зокрема Blooket у розрізі гейміфікації) та результати опрацювання нової науково-технічної літератури, я визначила чіткий план модернізації комплексу, який планую реалізувати у власній кваліфікаційній роботі:

1. Інтеграція нових видів та режимів ігор (Game Modes):

- Командні квізи (Team vs Team): розширення логіки реєстрації гравців з можливістю розподілу на команди, де сервер через WebSockets динамічно агрегуватиме результати групи в реальному часі й виводитиме командний лідерборд.
- Режим асинхронних «перегонів» (Progress Mode): створення автономного режиму, у якому студенти проходять квіз у власному темпі без синхронного очікування команди ведучого, а на головному екрані хоста відображається інтерактивний графічний прогрес кожної команди.
- Режим виживання (Knockout/Survival): гейміфікований режим на виліт, де користувачі, які надали неправильну відповідь або не вклалися в тайм-ліміт раунду, переходять у статус глядачів.

2. Розширення типів інтерактивних запитань:

До наявних типів запитань (вибір з варіантів та відкритий текст) заплановано додати питання на встановлення відповідності (Matching Pairs) та на визначення правильної послідовності (Ordering/Sorting) із реалізацією зручних Drag-and-Drop механік на фронтенді (на базі Mantine).

Результати цього комплексного аналізу, оцінка напрацювань Марії та сформульований план модернізації безпосередньо є грунтовною частиною першого розділу моєї власної кваліфікаційної роботи. Мною було узагальнено відомості про предметну область, визначено нову розширену мету дослідження та сформульовано технічні завдання для створення масштабованої та надійної системи інтерактивного навчання.

Висновок

Таким чином, у процесі проходження науково-дослідної практики мною було проведено аналіз сучасних технологій розробки інтерактивних вебсистем, досліджено архітектуру існуючого програмного комплексу та визначено основні напрямки його подальшої модернізації. Отримані результати стали основою для формування концепції власної кваліфікаційної роботи та подальших практичних досліджень у сфері real-time вебзастосунків.

01.06.2026 _____

(дата) (підпис)