Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет ННЦЗФН

(повна назва)

Кафедра Штучного інтелекту

(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Вебзастосунок «Віртуальна конференція ММФ»

(тема)

Виконав:

студент 4 курсу, групи ІТШІз-20-1

Мудруд В.Р.

(прізвище, ініціали)

Спеціальність 122 Комп’ютерні науки

(код і повна назва спеціальності)

Освітня програма Штучний інтелект

(повна назва спеціалізації)

Керівник ст. викл. Гриньова О.Є.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри   В.О. Філатов

(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет ННЦЗФН

(повна назва)

Кафедра Штучного інтелекту

(повна назва)

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп’ютерні науки

(код і повна назва)

Освітня програма Штучний інтелект

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри

(підпис)

« » 20 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Мудруду Владиславу Романовичу

(прізвище, ім’я, по батькові)

1. Тема роботи Вебзастосунок «Віртуальна конференція ММФ»

затверджена наказом університету від   6      травня    20 24 р. № 70Стз

2. Термін подання студентом роботи до екзаменаційної комісії   ХХ      червня    20 24 р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих

наукових проектів

4. Перелік питань, що потрібно опрацювати в роботі

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва eтапів роботи | Терміни  виконання етапів роботи | Примітка |
| 1 | Отримання завдання на кваліфікаційну роботу | 06.05.2024 | виконано |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Дата видачі завдання   6      травня    20 24 р.

Студент

(підпис)

Kepiвник роботи ст. викл. Гриньова О.Є.

(підпис) (посада, прізвище, ініціали)

**РЕФЕРАТ**

Пояснювальна записка: ХХ с., ХХ рис., ХХ табл., ХХ дод., ХХ джерел.

КЛЮЧОВІ СЛОВА.

Об’єкт дослідження – ХХХ.

Предмет дослідження – ХХХ.

Мета роботи – ХХХ.

Методи дослідження – ХХХ.

Анотація

**ABSTRACT**

Bachelor’s thesis contains: 61 pp., 25 fig., 3 tabl., 1 ann., 17 references.

**ЗМІСТ**

[ВСТУП 8](#_Toc167829993)

[1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ 11](#_Toc167829994)

[1.1 Аналіз проведення ММФ 11](#_Toc167829995)

[1.2 Аналіз існуючих платформ для проведення ММФ 13](#_Toc167829996)

[1.3 Мета розробки вебзастосунку 14](#_Toc167829997)

[1.4 Аналіз технологій для розробки вебзастосунку 15](#_Toc167829998)

[2 ПОСТАНОВКА ЗАДАЧІ 19](#_Toc167829999)

[2.1 Визначення функціональних вимог 19](#_Toc167830000)

[2.2 Визначення нефункціональних вимог 21](#_Toc167830001)

[2.3 Вибір технологій для розробки вебзастосунку 22](#_Toc167830002)

[2.4 Проектування архітектури вебзастосунку 24](#_Toc167830003)

[3 ПРОГРАМНА РЕАЛІЗАЦІЯ 30](#_Toc167830004)

[3.1 Реалізація графіки 30](#_Toc167830005)

[3.2 Реалізація інтерфейсу клієнта 30](#_Toc167830006)

[3.3 Реалізація API 30](#_Toc167830007)

[ВИСНОВКИ 31](#_Toc167830008)

[ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 32](#_Toc167830009)

[ДОДАТОК А Відомість кваліфікаційної роботи 33](#_Toc167830010)

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – штучний інтелект;

AI – Artificial Intelligence – штучний інтелект;

ММФ – Міжнародний молодіжний форум;

IoT – internet of things – интернет вещей;

3D – тривимірний.

# ВСТУП

Сучасний світ прогресує з неймовірною швидкістю, особливо в галузі цифрових технологій. Ці технології проникли в кожну сферу нашого життя, викликаючи кардинальні зміни в економіці, освіті, комунікаціях, управлінні та багатьох інших сферах. В цьому контексті веб-технології відіграють ключову роль, особливо у міжнародних комунікаціях, які стають все більш розповсюдженими та важливими.

У центрі цієї динаміки знаходяться міжнародні молодіжні форуми (ММФ). Міжнародні молодіжні форуми (ММФ) є критично важливими платформами для інтелектуального, культурного та соціального розвитку молоді в глобальному масштабі. Ці форуми стали важливою платформою для обговорення актуальних проблем, обміну досвідом, навчання та розвитку молодих людей з усього світу. Вони пропонують універсальний простір для діалогу, де молодь може зібратися разом, щоб обговорити та вирішити ключові питання сучасного світу. Їхнім головним завданням є стимулювання активного діалогу між молоддю різних країн, формування міжкультурної взаємодії, підтримка інновацій та творчості. ММФ слугують майданчиками для обговорення найбільш відповідальних та актуальних питань сучасності, від глобальних проблем, які впливають на всю людство, до конкретних наукових, технологічних та соціальних викликів.

Наприклад, в контексті радіоелектроніки та обчислювальної техніки, ММФ можуть слугувати важливими платформами для обговорення таких стратегічних розробок, як штучний інтелект, машинне навчання, IoT, обчислювальна техніка високої продуктивності, кібербезпека та багато інших. Наприклад, на такому форумі молоді вчені, інженери та розробники можуть представити свої останні дослідження, обговорити виклики та можливості, а також випрацювати спільні рішення для прогресу в цих областях.

Водночас, ММФ – це не просто місце для обговорення вже існуючих ідей, але й платформа для генерації нових концепцій, винаходів і інновацій. За допомогою спільної роботи, обміну досвідом і знаннями, учасники ММФ можуть розробляти нові технології, стратегії та підходи, що можуть мати глобальний вплив на розвиток суспільства.

Сучасні глобальні виклики, зокрема пандемія COVID-19 та війна в Україні, спонукали ММФ до переходу в віртуальний формат. Однак цей швидкий перехід виявився непростим і викликав ряд проблем, проте, одночасно, відкрив нові можливості для вдосконалення проведення таких форумів.

Також варто зазначити, що важливою складовою успішного ММФ є зручний і ефективний доступ до інформації про проведення форуму. Це включає не тільки програму заходу, але і презентації, доповіді, матеріали дискусій тощо. Стандартні інструменти часто не забезпечують об'ємного та структурованого представлення цієї інформації, що ускладнює навігацію та використання даного контенту для учасників. Наприклад, відсутність можливості створювати віртуальні «кімнати» для презентацій, де учасники можуть обговорювати ідеї та дослідження як у справжніх умовах, впливає на ефективність форуму. Це не тільки ускладнює процес представлення і сприйняття інформації, але і знижує рівень зацікавленості та взаємодії учасників.

Існує ряд альтернативних підходів, якими можна вирішити актуальну проблему віртуалізації Міжнародних молодіжних форумів.

Перший підхід – це використання стандартних інструментів відеоконференцій, таких як Zoom або Microsoft Teams. Ці платформи дозволяють організувати відеозв'язок, поділитися презентаціями і провести групові дискусії. Однак, вони зазвичай не надають достатньої інтерактивності та взаємодії, необхідних для повноцінного ММФ. Також вони зазвичай не мають зручного інтерфейсу для навігації та доступу до презентацій та інших матеріалів форуму.

Другий підхід – це створення спеціалізованих мобільних додатків. Це може бути корисним для учасників, які використовують мобільні пристрої, але водночас це вимагає значних витрат на розробку, адаптацію та підтримку різних версій додатку для різних операційних систем. Додатково, мобільні додатки зазвичай мають обмеження щодо великих обсягів контенту та можливостей для взаємодії.

Третій підхід – це використання веб-сайтів з динамічним контентом та інтерактивними елементами. Це може бути корисним для простої навігації та доступу до контенту, але не забезпечує достатнього рівня взаємодії та не вирішує проблему з організацією презентацій та дискусій.

В результаті, незважаючи на наявність різних альтернатив, вебзастосунок, спеціально розроблений для ММФ, видається найбільш вдалим варіантом. Він дозволяє об'єднати переваги інших підходів, забезпечуючи високий рівень інтерактивності, гнучкість, доступність, зручність використання та широкі можливості для подання та організації контенту.

Отже, проблема, яку планується вирішити, полягає в розробці вебзастосунку для віртуальних ММФ, що забезпечує високий рівень інтерактивності та зручність доступу до інформації. Цей застосунок має бути використаним у парі з іншими інструментами для комунікації, такими як відеоконференції та месенджери, забезпечуючи цілісний та продуктивний досвід участі в ММФ.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

## 1.1 Аналіз проведення ММФ

ММФ характеризуються різноманітністю форматів і видів активностей. Вони можуть включати панельні дискусії, робочі групи, інтерактивні сесії, воркшопи та тренінги, презентації та постерні сесії, зустрічі з експертами та лідерами громадської думки, культурні заходи, мережеві зустрічі та ін.

Основними учасниками ММФ є молоді люди з різних країн, які представляють різні соціокультурні контексти. Вони приходять на форуми з різними цілями – отримати нові знання, поділитися досвідом, зробити презентацію, знайти партнерів для проектів, встановити професійні та особисті контакти, отримати натхнення та ідеї для своєї діяльності.

Віртуальний формат ММФ, який став особливо актуальним в умовах пандемії COVID-19, має свої специфічні особливості. Він забезпечує більш широкий доступ до форумів, незалежно від географічного розташування учасників, але водночас створює додаткові виклики з організації ефективного онлайн-спілкування та інтерактивності.

Проведення віртуальних ММФ вимагає високих технологічних компетенцій від організаторів, забезпечення стабільних та зручних інструментів для відеозв'язку та онлайн-презентацій, створення сприятливого онлайн-середовища для дискусій та взаємодії, розробки стратегій привернення уваги та залучення учасників, що є невід'ємними складниками успіху ММФ.

Харківський національний університет радіоелектроніки (ХНУРЕ) є одним з провідних навчальних закладів в Україні, що регулярно організовує та приймає участь в численних наукових та освітніх заходах. Одним із ключових заходів, що проводиться у ХНУРЕ, є Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті» [1].

Цей форум є важливою платформою для обміну знаннями, досвідом та ідеями між молодими науковцями, інженерами, викладачами та студентами в області радіоелектроніки та суміжних дисциплін. Він включає ряд сесій, на яких обговорюються актуальні проблеми та перспективи розвитку радіоелектроніки, новітні технології, інноваційні проекти та ін.

В контексті цієї роботи, перехід цього форуму на віртуальний формат і питання організації ефективної віртуальної взаємодії представляє особливий інтерес. Він слугує прекрасним прикладом викликів, з якими можуть стикатися організатори і учасники подібних заходів, а також можливостей, які вони можуть отримати від використання нових технологій та стратегій комунікації. Тому досвід проведення цього форуму в ХНУРЕ буде використано в даній роботі для аналізу проблеми та формулювання рекомендацій щодо її розв'язання.

Один з викликів, з якими зіштовхуються організатори Міжнародних молодіжних форумів, – це потреба втримати увагу та зацікавленість учасників. Деякі учасники можуть відчувати незацікавленість або стомленість від довгих сесій, формальних презентацій та обговорень, особливо в онлайн-форматі. Це може призвести до пасивної участі, високого рівня відхилень та низького рівня задоволеності.

В цьому контексті гейміфікація може стати потужним інструментом для стимулювання активної участі, взаємодії та навчання. Впровадження елементів гри та ігрових принципів у неігрове середовище допомагає підсилити мотивацію, підвищити концентрацію та ефективність сприйняття інформації, викликати позитивні емоції [2].

Однією з ідей, які можна впровадити в цьому напрямку, є створення віртуальних кімнат. Вони можуть створити більш розслабленим і стимулююче середовище, яке сприяє креативності, співпраці та взаємному навчанню.

Крім створення віртуальних кімнат, можна розглянути такі ідеї, як використання системи віртуальних відзнак та нагород за активність, проведення віртуальних квестів та викликів, створення віртуальних сценаріїв для навчання та взаємодії. Все це може допомогти зробити ММФ більш живим, динамічним та захоплюючим для учасників.

Отже, проведення ММФ – це складний багатоаспектний процес, який вимагає глибокого розуміння контексту, цілей та потреб учасників, а також використання ефективних технологічних рішень та стратегій комунікації.

## 1.2 Аналіз існуючих платформ для проведення ММФ

Для проведення ММФ в сучасних умовах активно використовуються такі цифрові платформи як Zoom, Microsoft Teams, Google Meet та Skype. Кожна з них надає широкий спектр можливостей для віртуальної комунікації, але водночас має певні обмеження та недоліки, що ускладнюють їх ефективне використання в контексті ММФ.

Zoom є однією з найбільш популярних платформ для відеоконференцій. Її переваги полягають у високій стабільності, гнучкості та здатності вміщувати велику кількість учасників. Проте, Zoom має обмежені можливості для неструктурованого спілкування та мережевої взаємодії між учасниками. Більше того, недостатня інтеграція з іншими платформами та додатками може знизити ефективність групової взаємодії.

Microsoft Teams вирізняється високою надійністю та стабільністю, а також гнучкістю в налаштуванні параметрів конференцій. Однак, ця платформа також зосереджена на структурованому спілкуванні, і може не надавати достатньо можливостей для неформальних взаємодій між учасниками. Крім того, Teams може виявитися складною для користувачів, які не знайомі з екосистемою Microsoft.

Google Meet, хоча і є надійним та простим у використанні, має обмеження щодо взаємодії між учасниками. Наприклад, вона не дозволяє учасникам вільно перемикатися між різними кімнатами або сесіями, що може бути критичним для динамічного форуму, як ММФ.

Skype, хоч і був одним з перших сервісів відеоконференцій, зараз відстає від конкурентів з точки зору функціональності та стабільності. Крім того, Skype не надає можливостей для групової роботи над документами та проектами, що є важливим для соціального характеру ММФ.

Недостатня спроможність залучити активну участь учасників до ММФ є ключовим недоліком існуючих платформ, незважаючи на їхню стабільність та надійність. Вони не надають достатнього простору для вільного спілкування та взаємодії учасників та взагалі не піднімають ідею залучення гейміфікації у процес конференцій.

## 1.3 Мета розробки вебзастосунку

Інновація у процес ММФ яка вирішить багато поточних проблем може стати гейміфікація, так як її основна мета – стимулювання активності користувачів. Відповідні стратегії, такі як введення системи балів, організація викликів, створення віртуальних кімнат або навіть віртуального простору форуму, можуть підсилити залученість учасників.

Вирішальним фактором у підвищенні недостатнього рівня активної взаємодії на ММФ може стати саме створення віртуальних кімнат. У такому середовищі учасники можуть вільно перемикатися між різними сесіями, обговорювати питання в невеликих групах і взаємодіяти з іншими учасниками на більш особистому рівні.

Таким чином, не лише забезпечуються більш активна взаємодія учасників, але це також допомагає створити більш захоплююче середовище, що в свою чергу підвищує успішність ММФ.

Симбіоз існуючих платформ для комунікації з віртуальними кімнатами може бути перспективним рішенням для підвищення ефективності проведення ММФ.

Розглянемо це на прикладі симбіозу вебзастосунку для віртуальних конференцій ММФ та одного з існуючих додатків для комунікацій, скажімо, Zoom. Ці дві платформи, що доповнюють одна одну, мають потенціал створити сильну, цілісну систему для проведення ММФ.

Вебзастосунок має потенціал створити віртуальний простір, де аватари учасників можуть взаємодіяти з контентом, представленим на форумі. У такому середовищі кожен учасник, представлений своїм віртуальним аватаром, має можливість відвідувати віртуальні кімнати, де відбуваються різні сесії, обмінюватися думками з іншими учасниками, брати участь у віртуальних викликах та інтерактивно взаємодіяти з представленим контентом.

З іншого боку, додаток Zoom може бути використаний як доповнення до вебзастосунку, забезпечуючи надійну платформу для прямого спілкування між учасниками. Він гарантує високу якість звуку та відео, що є важливим для пленарних сесій, обговорень груп або презентацій.

Сполучення цих двох інструментів надає можливість створити гнучку, динамічну та інтерактивну платформу для ММФ. Віртуальні кімнати дозволяють створити атмосферу неформального робочого простору, а надійність та стабільність Zoom гарантують успішність комунікацій між учасниками.

Таким чином, симбіоз вебзастосунку для віртуальних конференцій ММФ та додатку для комунікацій, як Zoom, може представляти собою ефективний інструмент для підвищення ефективності проведення ММФ, забезпечуючи високий рівень залученості та зацікавленості учасників.

## 1.4 Аналіз технологій для розробки вебзастосунку

Розглядаючи технологічні аспекти реалізації вебзастосунку, мається на увазі значний спектр технологічних варіантів. Відповідно до специфіки нашого проекту, необхідно обрати високопродуктивні, гнучкі, інтерактивні та надійні технологічні стеки. Ми маємо зосередитися на аналізу інструментів для створення графіки, клієнта, сервера та бази даних.

Ключовим аспектом є створення ефективної та візуально привабливої 3D графіки. Вибір технологій, що дозволяють реалізувати ці цілі, є вирішальним для успіху проекту.

Three.js є однією з найпопулярніших JavaScript бібліотек для створення та візуалізації 3D графіки в браузері [3]. Переваги Three.js полягають у його гнучкості, багатосторонності та широких можливостях кастомізації. Завдяки використанню WebGL, Three.js надає можливість створювати деталізовані 3D моделі з плавним рендерингом. Проте, як і будь-яка технологія, Three.js має деякі обмеження, зокрема складність в освоєнні та велику залежність від ресурсів системи.

Babylon.js – альтернатива Three.js, також розроблена для створення 3D графіки. Більш дружній до розробника, Babylon.js пропонує більш обширний набір функцій і може бути використаний для створення високоякісних 3D сцен. Недоліком Babylon.js може бути його продуктивність. Не дивлячись на те, що він має широкий спектр функцій, додатки, розроблені з використанням Babylon.js, можуть працювати повільніше, особливо на старіших комп'ютерах або пристроях з обмеженими ресурсами, що може бути критичним для вебзастосунків, які повинні бути доступними для якомога більш широкої аудиторії.

A-Frame – це вебфреймворк для створення сцен віртуальної реальності, що використовує HTML. A-Frame пропонує простий синтаксис та широкі можливості для реалізації VR-досвіду. Однак, варто враховувати, що для складних 3D сцен він може бути менш потужним, ніж Three.js чи Babylon.js. Також у нашому не має сенсу використовувати повільнішу та більш складну технології за інші, так як ми не плануємо інтегруватися з VR технологією.

Розробка на основі ігрових рушіїв, таких як Unity або Unreal Engine, також можлива, але не завжди доцільна для вебзастосунків. Ці рушії мають великі можливості для створення 3D графіки, але вони є важкими для браузерів і можуть вимагати додаткового програмного забезпечення для користувача.

При обранні технологій для реалізації клієнт-серверної взаємодії веб-застосунку, варто враховувати декілька ключових факторів: продуктивність, гнучкість, швидкість виконання та спільнота розробників.

Node.js з Express.js визнаний відмінним вибором для розробки API, завдяки асинхронному виконанню, що забезпечує високу продуктивність та швидкість відповіді сервера. Express.js пропонує просту та гнучку структуру для розробки вебдодатків та API. Основним недоліком може бути відносно висока складність для новачків.

Deno є сучасною альтернативою Node.js. Deno пропонує більш безпечне середовище для виконання JavaScript, вбудований модуль TypeScript та набір стандартних утиліт, що полегшують розробку. Однак, Deno ще є новим інструментом з менш розвиненою екосистемою, що може ускладнювати отримання допомоги або використання певних пакетів.

Python з Flask/Django може бути гарним варіантом для розробки API. Python відомий своєю легкістю вивчення та чистим синтаксисом. Flask дозволяє швидко створювати прості API, тоді як Django пропонує більш багатий функціонал, але й більш складний в освоєнні. Однак, Python може бути повільнішим за Node.js для виконання інтенсивних обчислень.

Розробка API та бази даних для вебзастосунку вимагає використання технологій, що забезпечують надійність, масштабованість та безпеку. Одним з наборів технологій, що вже встиг себе зарекомендувати в цьому контексті, є комбінація Java, Spring, JPA та MySQL.

Java є об'єктно-орієнтованою мовою програмування, що забезпечує безпеку типів, підтримку багатопоточності та широкий спектр API, що охоплює велику кількість задач. Завдяки своїй надійності та відмінній продуктивності, Java є чудовим вибором для розробки високонавантажених систем.

Spring - це високорівневий фреймворк для Java, який спрощує процес розробки великих корпоративних додатків [4]. Spring надає гнучкі абстракції для роботи з різними технологіями, включаючи JPA та MySQL, і допомагає розробникам створювати код, що легко тестується та підтримується.

JPA (Java Persistence API) - це стандарт для збереження, відновлення та запиту даних між об'єктами Java та реляційними базами даних. JPA дозволяє розробникам працювати з базами даних на високому рівні абстракції, мінімізуючи кількість повторюваного коду та спрощуючи розробку.

MySQL - це одна з найпопулярніших реляційних баз даних, відома своєю надійністю, високою продуктивністю та гнучкістю. MySQL підтримує транзакції, має гнучкі механізми безпеки та може легко масштабуватися для обробки великих обсягів даних.

При виборі технологій для розробки вебзастосунку важливо провести детальний аналіз та оцінити кожен з потенційних варіантів. Вибір технологій безпосередньо впливає на час розробки, складність проекту та кінцеву якість вебзастосунку.

Варто звернути увагу, що навіть найкращі технології не зможуть забезпечити успіху проекту, якщо вони не відповідають специфічним вимогам або не використовуються правильно. Тому критично важливо вести неперервний процес оцінки та адаптації, щоб забезпечити, що вибрані технології продовжують відповідати потребам проекту на всіх етапах його розробки.

# 2 ПОСТАНОВКА ЗАДАЧІ

## 2.1 Визначення функціональних вимог

Визначення вимог до системи – це один з базових етапів в процесі проектування програмних систем, що впливає на всі подальші стадії їх розробки. Віртуальна конференція ММФ має забезпечити цілісний та інтерактивний досвід для учасників, забезпечуючи при цьому високий рівень функціональності та зручності використання. Основні функціональні вимоги включають наступні аспекти:

Основною вимогою є створення віртуальних кімнат, які імітують реальне середовище конференції, У цих кімнатах аватари учасників будуть здатні перебувати та вільно переміщатися. Віртуальні кімнати повинні бути обладнані всіма необхідними елементами, що відображають атмосферу реального місця проведення заходу. Це важливо для підтримки активної взаємодії між учасниками та створення атмосфери реальності.

Відповідно до цієї вимоги, необхідно розробити механізми переміщення у віртуальному просторі. Це забезпечить користувачам можливість вільно рухатися, відвідувати різні стенди та взаємодіяти з віртуальним середовищем. Вільне переміщення є ключовим аспектом імітації реального досвіду участі в конференції. Користувачі будуть мати можливість не просто ходити у віртуальному просторі, а також бігати та стрибати.

У віртуальних кімнатах мають розміщуватися стенди з презентаціями, доповідями та іншими матеріалами. Ця вимога забезпечує інформативність вебзастосунку, дозволяючи учасникам легко знайти необхідну інформацію та отримати доступ до навчальних матеріалів.

Окрім цього має бути можливість завантажити будь-який навчальний матеріал, що присутній у кімнаті. Це дозволить учасникам легко обмінюватися цікавою інформацією.

Також на'ява інтерфейсу користувача є ключовою вимогою до вебзастосунку. Він повинен відображати велику кількість інформації в структурований та зрозумілий спосіб, забезпечуючи при цьому швидку та легку взаємодію користувача з системою.

Основний функціонал користувача також повинен включати можливість підключення до віртуальних кімнат за допомогою унікального коду. Це не тільки наддасть додаткового рівня захисту, але й дозволить учасникам легко приєднатися до потрібної кімнати, отримавши код із запрошенням.

Користувач має мати можливість переглядати список усіх доступних віртуальних кімнат. Це допоможе учасникам отримати загальне уявлення про структуру конференції та визначити, які кімнати їх зацікавлять. Крім того, користувач має мати можливість фільтрувати та сортувати цей список, що значно спростить процес пошуку потрібної кімнати. Після цього юзер зможе підключитися до потрібної кімнати у один клік.

Також на сайті має бути присутня зручна навігація, що дозволить користувачам швидко переходити до будь-якої сторінки.

У вебзастосунку має бути присутнім функціонал адміністратора системи, який має мати можливість створювати нові віртуальні кімнати та наповнювати їх матеріалами. Окрім цього він зможе видаляти вже існуючі кімнати. Для доступу у функціонал адміністратора користувачу треба буде авторизуватися за допомогою логіна та пароля.

Виважені і детально пророблені функціональні вимоги до вебзастосунку створюють надійну основу для успішного планування та виконання проекту. Вони відображають ключові аспекти системи, що будуть реалізовані. Після успішної реалізації вимог отримаємо потужний інструмент для проведення віртуальних конференцій, здатний задовольнити потреби будь-яких користувачів і забезпечити їм високоякісний досвід участі в таких заходах.

## 2.2 Визначення нефункціональних вимог

Нефункціональні вимоги визначають якісні атрибути системи, що впливають на зручність використання, надійність та продуктивність вебзастосунку.

Приваблива графіка віртуальних кімнат є ключовою вимогою для створення привабливого та занурюючого досвіду для користувачів. Відтворення деталей, якість текстур та висока роздільна здатність є важливими чинниками для забезпечення цієї вимоги.

Зручне керування у віртуальних кімнатах є необхідністю для забезпечення інтуїтивно зрозумілої та простої взаємодії користувачів із системою. Відповідно, вебзастосунок повинен мати прості та зрозумілі елементи керування.

Висока продуктивність віртуальних кімнат вимагає оптимізації та ефективної роботи з ресурсами. Кімнати повинні гладко працювати навіть на не дуже потужних пристроях, забезпечуючи при цьому високу якість графіки, а також низьку затримку завантаження.

Швидкість завантаження віртуальних кімнат є важливою вимогою, оскільки вона впливає на загальний досвід користувача. Наявність візуального індикатора процесу завантаження, такого як лоадер, також допомагає покращити сприйняття продуктивності системи.

Завантаження навчальних матеріалів має відбуватися швидко і безперебійно, щоб забезпечити зручність використання та високу продуктивність навчального процесу.

Захищеність кімнат є важливою вимогою, що забезпечує конфіденційність інформації, яка обмінюється у віртуальній кімнаті, та гарантує, що лише учасники з правильним кодом доступу можуть увійти до кімнати.

Адаптивний та привабливий інтерфейс користувача, що працює гладко на всіх типах пристроїв і розмірах екрану, є важливим для зручності користувачів.

Для захисту даних адміністратора необхідне шифрування паролів. Це забезпечує додатковий рівень безпеки та захисту від несанкціонованого доступу.

Ефективна обробка помилок на клієнті допомагає підвищити стабільність вебзастосунку та поліпшує досвід користувача, забезпечуючи своєчасне виявлення та виправлення помилок.

Вебзастосунок повинен мати високу доступність та надійність, забезпечуючи безперервний доступ до своїх служб та ресурсів. Він також повинен бути здатний керувати великими обсягами даних та підтримувати велику кількість одночасних користувачів.

Безпека даних на сервері є критично важливою вимогою, що передбачає використання надійних алгоритмів шифрування, регулярне резервне копіювання та використання захищених протоколів передачі даних.

Всі ці нефункціональні вимоги допомагають створити високоякісний вебзастосунок, який надає користувачам надійний та ефективний інструмент для віртуальних конференцій.

## 2.3 Вибір технологій для розробки вебзастосунку

Основуючись на аналізі функціональних та нефункціональних вимог до вебзастосунку, а також на аналізі технологій розробки зробленим раніше, було вирішено використовувати наступний стек технологій:

API частина додатку буде розроблена на основі Java, Spring, JPA та MySQL.

Java – це надійна, безпечна та широко використовувана мова програмування, яка дозволяє створювати високопродуктивні та масштабовані вебзастосунки.

Фреймворк Spring, що базується на Java, надає велику кількість інструментів і сервісів для створення безпечних, масштабованих і високопродуктивних вебдодатків. Він надає механізми ін'єкції залежностей, аспектно-орієнтованого програмування, транзакційних операцій, управління життєвим циклом компонентів, обробки винятків та інше.

Java Persistence API (JPA) – це стандартний інтерфейс в Java, який дозволяє розробникам управляти реляційними даними в їхніх додатках. JPA автоматизує процеси зберігання, доступу, оновлення та видалення даних, забезпечуючи простий спосіб взаємодії з базами даних.

MySQL використовується як система управління базами даних. MySQL є однією з найбільш популярних відкритих систем керування базами даних, яка забезпечує високу швидкість обробки запитів, стабільність та гнучкість [5].

На серверній стороні використовується Node.js, який є легким та ефективним середовищем для виконання JavaScript на сервері. З використанням Express.js, який є ведучим фреймворком для Node.js, можна додати ряд маршрутів, мідлварів та інших можливостей для побудови вебзастосунків.

Мова програмування JavaScript, як основна мова програмування для сторони клієнта, буде застосована через свою гнучкість, масштабованість та широкі можливості для розробки інтерактивних вебінтерфейсів. JavaScript дозволяє інтегрувати різноманітні бібліотеки та фреймворки, що розширюють його функціональні можливості.

Також на стороні клієнта використовується Three.js для реалізації графіки віртуальних кімнат. Three.js – це бібліотека JavaScript, яка надає високорівневий інтерфейс для створення і анімації 3D графіки в браузері. HTML і CSS використовуються для створення вебінтерфейсу, забезпечуючи структуру та стиль відповідно.

Для обробки контенту кімнат використовується бібліотека pdfjs-dist. Ця бібліотека дозволяє розглядати PDF документи прямо в браузері, без необхідності встановлювати додаткові плагіни або програми.

З урахуванням комплексності та об'єму проекту, буде дотримано основні принципи об'єктно-орієнтованого програмування, що дозволять створити масштабований, гнучкий і легко підтримуваний код. Крім того, буде використано такі принципи як SOLID та DRY, що є фундаментальними в сучасній програмній інженерії.

Вибраний стек технологій забезпечує багатофункціональність та високу продуктивність майбутнього застосунку. Враховуючи високу надійність та здатність до масштабування цих технологій, можна очікувати стабільне функціонування додатку незалежно від навантаження.

Це дозволить забезпечити високу швидкість обробки запитів та відгуку системи, що є критично важливим для надання користувачам найкращого досвіду.

Окрім того, використання таких технологій гарантує високий рівень безпеки даних користувачів та інформації, що перебуває у системі. Це включає захист від несанкціонованого доступу, забезпечення цілісності даних та конфіденційності.

Тому вибір цього стеку технологій є оптимальним для реалізації вебзастосунку.

## 2.4 Проектування архітектури вебзастосунку

В процесі проектування архітектури вебзастосунку можна розглянути різні варіанти щодо структуривання додатку. Зокрема, такі типи архітектури: монолітна, мікросервісна та серверна.

Монолітна архітектура є однією з найпростіших структур для вебзастосунків, оскільки всі елементи системи – від інтерфейсу користувача до обробки бізнес-логіки та роботи з базою даних – знаходяться в одному застосунку і виконуються в одному процесі.

Ця архітектура має свої переваги. По-перше, її простота в розробці. Необхідність координувати лише один застосунок може значно спростити процес розробки. По-друге, впровадження монолітної архітектури також є більш простим, так як не потрібно розгортати та налаштовувати кілька окремих служб. Нарешті, монолітна архітектура може надати кращу продуктивність у деяких випадках через безпосередню взаємодію компонентів, що уникнути додаткового мережевого затримання.

Проте, монолітна архітектура не є без недоліків. Одним з головних є проблеми з масштабуванням. В монолітних системах, коли потрібно масштабувати окремі частини додатку, це може бути важким завданням, оскільки все повинно бути масштабовано разом. Іншим недоліком є ризик впливу помилок – якщо стається збій в одній частині системи, це може вплинути на весь додаток. Крім того, внесення змін в монолітний додаток може виявитися складним і часозатратним, оскільки потрібно перевіряти вплив змін на всю систему, а не тільки на ізольований компонент. Хоча монолітна архітектура може бути корисною в певних контекстах, вона може не бути оптимальною для великих, складних систем, які потребують більшої гнучкості та можливостей масштабування. Такий підхід вже вважається застарілим. Приклад реалізації монолітної архітектури на рисунку 2.1.

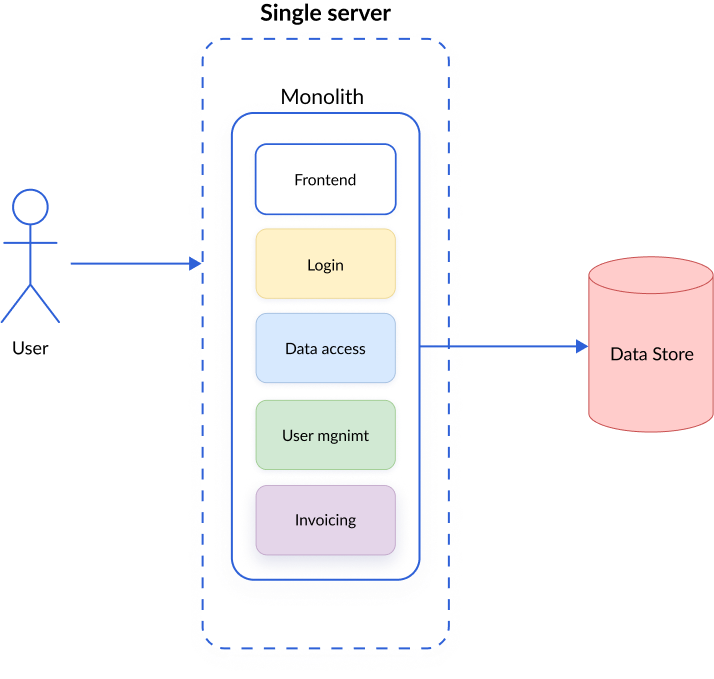


Рисунок 2.1 – Приклад монолітної архітектури

У контексті мікросервісної архітектури, вебзастосунок розбивається на набір незалежних сервісів, кожен з яких виконує свою специфічну функцію та спілкується з іншими через визначений API. Ці сервіси можуть бути розроблені, розгорнуті та масштабовані незалежно один від одного, що надає значну гнучкість та можливості для масштабування.

Мікросервісна архітектура володіє рядом переваг. Незалежність сервісів допомагає забезпечити локалізацію помилок – якщо стається збій в одному сервісі, це не впливає на роботу інших. Окрім цього, кожен сервіс може бути розроблений за допомогою технології, яка найкраще підходить для його задач. Це може підвищити ефективність розробки та продуктивність застосунку. Також, мікросервіси можуть бути масштабовані окремо, що дозволяє краще використовувати ресурси.

Але разом з перевагами, мікросервісна архітектура включає і свої виклики. Однією з проблем може бути управління взаємодією між сервісами. Зв'язок між мікросервісами може бути складним через потребу в обміні даними та сумісності інтерфейсів. Також, потрібно враховувати ускладнений процес розгортання, оскільки кожен сервіс має бути розгорнутим окремо.

Відповідно, мікросервісна архітектура вимагає від додаткового управління та координації, але при правильному підході вона може виявитися надзвичайно потужною та гнучкою. Така архітектура зараз популярна і вважається гарною практикою. Приклад реалізації мікросервісної архітектури на рисунку 2.2.

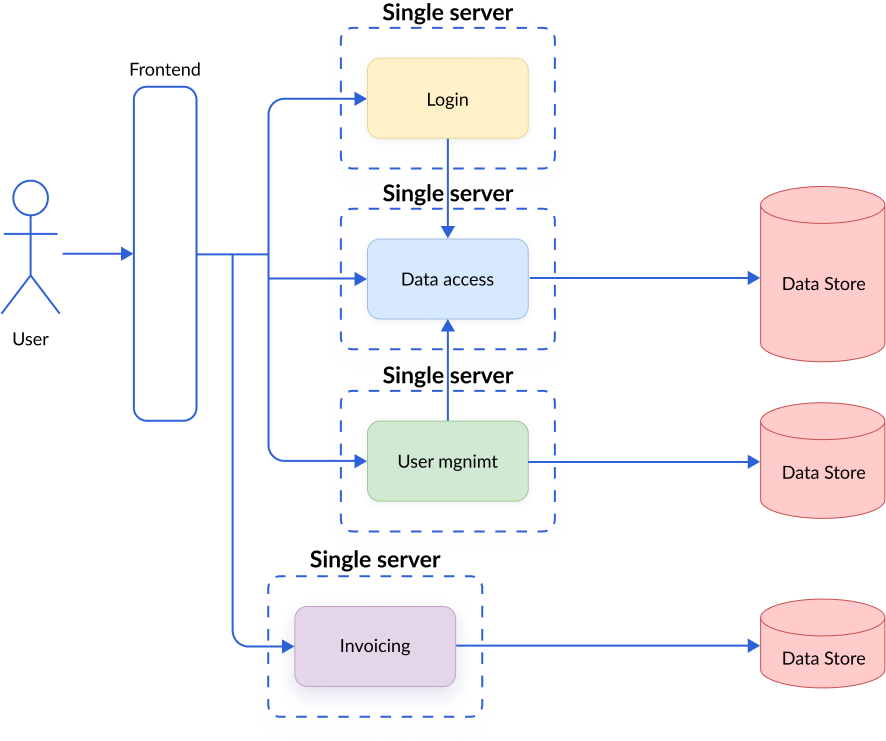


Рисунок 2.2 – Приклад мікросервісної архітектури

Так, як монолітна архітектура буде обмежувати нас у виборі технологій, а мікросервісна архітектура є занадто складною для обсягу поставлених задач то для реалізації вебзастосунку кращим рішенням буде обрати комбінований підхід. Архітектура додатку включатиме чотири основні компоненти: клієнт, вебсервер, аплікейшенсервер та базу даних.

Клієнт – це користувацький інтерфейс, який відповідає за надання відповідей користувачеві та відправку запитів до сервера.

Вебсервер відповідає за адресацію запитів та логіку, що відповідає за презентацію даних. Він є центральним місцем для всіх запитів від клієнта, де він перенаправляє їх до аплікейшенсервера.

Аплікейшенсервер, що представляє собою API, виконує бізнес-логіку та взаємодіє з базою даних. Він обробляє запити від вебсервера, виконує необхідні дії та відсилає відповіді назад до вебсервера.

База даних зберігає всю інформацію, необхідну для нашого додатку, та взаємодіє з аплікейшенсервером.

Такий підхід дозволяє нам використовувати різні технології для вебсервера та аплікейшенсервера, що дає можливість вибрати найкращі рішення для кожного з них. У нашому випадку технології які використовуються на вебсервері спрямовані на роботу з графікою для віртуальних кімнат, а аплікейшенсервер сконцентрований на безпеці та технологіях які краще і легше реалізують роботу з базами даних.

Цей підхід забезпечує максимальну гнучкість при розробці і впровадженні, дозволяючи нам розробляти та масштабувати різні частини системи незалежно одна від одної. Ми отримуємо можливість інтегрувати додаткові сервери за потребою, наприклад для реалізацію функціоналу кешування. Такий підхід також надає нам простір до можливого розширення функціоналу у майбутньому. Поява нових аплікейшенсерверів для реалізації іншого теоретичного функціоналу перетворить нашу архітектуру у мікросервісну, якщо складність поставлених задач буде цього вимагати. Структура архітектури додатку показана на рисунку 2.3.

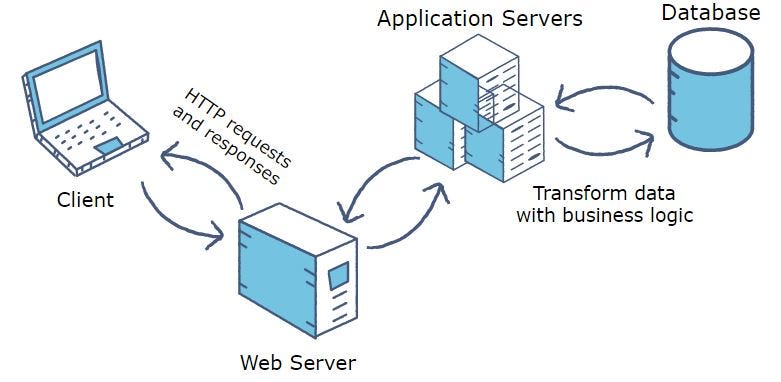


Рисунок 2.3 – Структура архітектури вебзастосунку

# 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

## 3.1 Реалізація графіки

## 3.2 Реалізація інтерфейсу клієнта

## 3.3 Реалізація API

# ВИСНОВКИ

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

# ДОДАТОК А Відомість кваліфікаційної роботи

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Позначення | | | | | Найменування | | Дод.  відомості | |
|  | | | | | Текстові документи | |  | |
| **1.** | | | | | Пояснювальна записка | | ХХ с. | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | | Інші документи | |  | |
|  | | | | |  | |  | |
| **2.** | | | | | Презентаційні матеріали | | \_\_ плакатів | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  | | | | |  | |  | |
|  |  |  |  |  |  | | | |
|  |  |  |  |  |
| . | . | Прізвище та ініціали. | Підп. | Дата |
| Розробив | |  |  |  | Тема | Шифр групи | | Код напр./спец. |
| Перевірив | |  |  |  | ІТШІ-20-1 | | 122 |
| Н.контр. | | Малєєва І.А. |  |  | ХНУРЕ  кафедра ШІ | | |
| Затв. | | Філатов В.О. |  |  |