**Форма № Н-9.02**

Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

\_Навчально-науковий інститут інформаційних технологій та робототехніки\_

(повна назва факультету)

\_Кафедра комп’ютерних та інформаційних технологій і систем\_

(повна назва кафедри)

**Пояснювальна записка**

**до дипломного проекту (роботи)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_бакалавра\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(освітньо-кваліфікаційний рівень)

на тему

\_\_\_\_\_\_Розроблення інформаційної системи для проведення онлайн-\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_конференцій\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Виконав: студент \_4\_\_ курсу, групи 401 ТН

спеціальності

\_\_122 Комп’ютерні науки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(шифр і назва напряму)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Пєвнєв Д.П.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник \_\_\_\_\_Гайтан О.М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Рецензент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Полтава – 2021 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

**КАФЕДРА КОМП’ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**Спеціальність 122 «Комп’ютерні науки»**

**на тему**

**«Розроблення інформаційної системи для проведення онлайн-конференцій»**

**Студента групи 401-ТН Пєвнєва Дмитра Павловича**

Керівник роботи

старший викладач

Гайтан О.М.

Завідувач кафедри

кандидат технічних наук,

доцент Головко Г.В.

Полтава – 2021

**РЕФЕРАТ**

Кваліфікаційна робота бакалавра: 69 с., 29 рисунків, 1 додаток, 17 джерел.

**Об’єкт дослідження:** інформаційна система для проведення онлайн-конференцій

**Мета роботи:** проектування та реалізація інформаційної системи для проведення онлайн-конференцій.

**Методи:** Об’єктно-орієнтоване проєктування, проєктування баз даних, методологія розробки MVC.

**Ключові слова:** вебсайт, інформаційна система, MVC, онлайн-конференція, доповіді.

**ABSTRACT**

Bachelor's qualification work: 69 pages, 29 drawings, 1 appendix, 17 sources.

**Object of research**: information system for online conferences

**Purpose:** design and implementation of an information system for online conferences.

**Methods**: Object-oriented design, database design, MVC development methodology.

**Keywords:** website, information system, MVC, online conference, reports.

ЗМІСТ

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 6](#_Toc74666836)

[ВСТУП 7](#_Toc74666837)

[РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ 9](#_Toc74666838)

[1.1 Онлайн-конференція як форма проведення науково-практичних конференцій 9](#_Toc74666839)

[1.2 Переваги та недоліки інтернет-конференцій 9](#_Toc74666840)

[1.3 Приклади рішень 11](#_Toc74666841)

[1.4 Постановка задачі дипломної роботи 13](#_Toc74666842)

[РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОВЕДЕННЯ ІНТЕРНЕТ-КОНФЕРЕНЦІЙ 14](#_Toc74666843)

[2.1 Життєвий цикл розроблення програмного забезпечення 14](#_Toc74666844)

[2.2 Принцип роботи веб-сайту 19](#_Toc74666845)

[РОЗДІЛ 3 ПРОЦЕС РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН-КОНФЕРЕНЦІЙ 22](#_Toc74666846)

[3.1 Вибір та обґрунтування веб-технологій для розроблення програмного забезпечення 22](#_Toc74666847)

[3.1.1 Вибір та обґрунтування використання мови PHP.. 22](#_Toc74666848)

[3.1.2 Вибір та обґрунтування використання MySQL. 24](#_Toc74666849)

[3.1.3 Вибір та обґрунтування використання XAMPP.. 27](#_Toc74666850)

[3.1.4 Вибір та обґрунтування використання фреймворку Codeigniter. 28](#_Toc74666851)

[3.2 Проєктування інформаційної системи 29](#_Toc74666852)

[3.3 Розробка структури сайту та макетів 35](#_Toc74666853)

[3.4 Розробка бази даних 40](#_Toc74666854)

[3.5 Реалізація програмного продукту 44](#_Toc74666855)

[РОЗДІЛ 4 ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОВЕДЕННЯ ІНТЕРНЕТ-КОНФЕРЕНЦІЙ 51](#_Toc74666856)

[4.1 Тестування верстки сайту 51](#_Toc74666857)

[4.2 Тест на крос-браузерність 53](#_Toc74666858)

[ВИСНОВКИ 55](#_Toc74666859)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 56](#_Toc74666860)

[ДОДАТОК А ВИХІДНИЙ КОД ОСНОВНИХ КОМПОНЕНТІВ 58](#_Toc74666861)

[ДОДАТОК B УЧАСТЬ У КОНФЕРЕНЦІЇ](#_Toc74666861) 67

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

**SDLC–**Software development lifecycle

**UML –** Unified Modeling Language

**FTP –** File Transfer Protocol

**HTTP –** HyperText Transfer Protocol

**ПК –**Персональний комп’ютер

**URL –** Uniform Resource Locator

**DNS –** Domain Name System

**PHP –**Мова програмування

**HTML –**HyperText Markup Language

**CSS –**Cascading Style Sheets

**MySQL–**Вільна система керування реляційними базами даних

**XAMPP –** Багатоплатформова збірка веб-сервера

**СУБД –**Система управління баз даних

**БД –**База даних

**MVC –** Model-view-controller

**ІС –**Інформаційна система

**W3С –**World Wide Web Consortium

**ВСТУП**

В наш час розвиток наукової діяльності є одним із найважливіших процесів сьогодення. Передача інформації та її освоєння, набуття навичок, як теоретичних, так і практичних, є необхідним для впровадження технологій, які, зі свого боку, зможуть облегшити життя кожної людини.

Одним із основних напрямів поширення такого роду інформації є конференція.

Конференція – це зустріч груп осіб, окремих осіб, в певному місці для обговорення конкретних питань або проблем, які були визначенні заздалегідь.

Наразі, проведення очних конференцій обмежено, через ситуацію із COVID-19 у світі, тому є необхідність використати такий ефективний інструмент розповсюдження інформації як інтернет.

Інтернет-конференція існує в мережі інтернет. Це інтерактивний веб-сайт, де учасники входять у систему та читають статті, беруть участь в дискусіях з іншими делегатами та ведучими, проводять чат у прямому ефірі та можуть поставити свої питання, взаємодіяти з людьми з усього світу. Вона має схожу структуру з фізичною конференцією, за винятком того, що семінари тривають довше, обговорення набагато ширші, а учасники конференції - більш зосереджені.

Крім того, традиційні конференції вимагають від учасників переміщення та перебування в певному місці. Це, зі свого боку, займає час та гроші. Але, оскільки онлайн-конференція є в інтернеті, учасники можуть отримати доступ до неї з будь-якої точки світу і можуть зробити це в будь-який час, використовуючи стандартне програмне забезпечення браузера. Учасники можуть увійти в систему коли завгодно - до, після або під час робочого часу. Вони отримують пароль для доступу до різних семінарів конференцій. Брати участь може кожен, хто має доступ до мережі інтернет.

Нині, Полтавська політехніка імені Юрія Кондратюка активно використовує такий формат конференцій, проводячи Міжнародну науково-технічну конференцію “Комп’ютерна математика в науці, інженерії та світі”.

Метою кваліфікаційної роботи бакалавра є розроблення інформаційної системи для проведення онлайн-конференцій.

Після завершення всіх етапів роботи передбачається розміщення інтернет-конференції на самостійний хостинг, та подальша робота із ним. Також передбачається підтримка та вдосконалення майбутнього проекту.

**РОЗДІЛ 1**

**АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ**

1. **Онлайн-конференція як форма проведення науково-практичних конференцій**

Онлайн-конференція– це інтерактивний веб-сайт, де учасники входять у систему та читають статті, беруть участь в дискусіях з іншими делегатами та ведучими, проводять чат у прямому ефірі та можуть поставити свої питання, взаємодіяти з людьми з усього світу.

Проведення інтернет-конференції відбувається в декілька етапів:

1. Створення оголошення про конференцію, оприлюднення теми, дати та часу проведення та дедлайни подачі тез.
2. Реєстрація учасників конференції.
3. Приймання тез та текстів доповідей комітетом.
4. Завантаження на сайт файлів із тезами та доповідями учасників.
5. Обговорення питань, яке здійснюється дистанційно за допомогою програм відеоконференцій [1].
6. Розміщення на сайтів збірника тез та/або матеріалів конференції.
7. **Переваги та недоліки інтернет-конференцій**

До переваг онлайн-конференцій можна віднести:

* онлайн-конференції доступні майже в кожному місті. За допомогою спеціального програмного забезпечення людина, яка бере участь в конференції, може взаємодіяти з іншими учасниками, не зважаючи на дистанцію між ними. Програмне забезпечення пропонує міцні та продуктивні стосунки між більш віддаленими особами. Це покращує зв’язок між людьми. Онлайн-зустрічі забезпечують платформу для безперебійного спілкування. Сучасний світ вважає онлайнові зустрічі одним з найбільш ефективних методів спілкування;
* збереження часу є другою важливою перевагою інтернет-конференцій. Учаснику немає потреби планувати своє бізнес відрядження та витрачати час на подорож. Все що йому необхідно, – лише інтернет;
* оскільки немає потреби у бізнес відрядженні, онлайн конференції не є витратними. Вони вигідні як для звичайних учасників, так і для компаній, які відправляють своїх робітників для участі в конференції;
* через відсутність необхідності в подорожі інтернет-конференція зменшує вплив людства на навколишнє середовище через забруднення повітря;
* віртуальна конференція також є корисною для здоров’я самих учасників. Люди, які часто подорожують, страждають від стресів, поганого сну, та поганого приймання їжі. Оскільки під час конференції учасники можуть залишиться вдома, їх самопочуття буде кращим [2];
* оскільки значно спрощується можливість потрапляння на конференцію, то підвищується її загальна відвідуваність. Як наслідок, у обговоренні питання, винесеного на конференцію, бере участь більше людей;
* під час онлайн конференцій набагато легше обмінюватись інформацією. Немає необхідності друкувати копії робіт та надсилати поштою, адже все робиться через інтернет;
* онлайн-конференції надають можливість легшого нотування інформації. Учасник може записати виступ колеги або зробити скріншоти його роботи, для більш чіткого ознайомлення.

Серед недоліків інтернет-конференцій можна виділити:

* для пояснення та обговорення певних питань доцільніше мати прямий контакт із людиною. Тому у реальних зборах інформація засвоюється краще;
* нерідко в онлайн конференціях у доповідачів виникають проблеми із інтернет з’єднанням. Ще гірші ситуації, коли вони не мають змоги вплинути на несправність. Це може привести до втрати обговорення важливих питань, вплинути на загальний стан конференції та відволікати інших учасників від роботи;
* неможливо повністю випробувати деякі продукти, просто переглянувши їх на екрані. На очних конференціях учасники мають можливість доторкнутися до зразків або спробувати прототип для себе;
* члени конференції можуть перебувати в різних часових поясах. У випадку міжнародних конференцій час проведення конференції може бути зручним не для всіх учасників;
* також, на жаль, наразі не можливо із впевненістю сказати, що уся інформація надійно захищена. Існує можливість кібер-атак, які можуть привести до втрати важливих документів, персональних даних користувачів, та втрати інтернет-майданчику в цілому.

1. **Приклади рішень**

Приклад оформлення сайту-конференції Міжнародна науково-практична онлайн-конференція «Актуальні проблеми, пріоритетні напрямки та стратегії розвитку України» за 2021 рік. Як можемо бачити, першим чином користувачу одразу дають можливість зареєструватись за допомогою форми. Нижче приведені теми питань та обговорень (рис.1.1 – 1.2) [4].

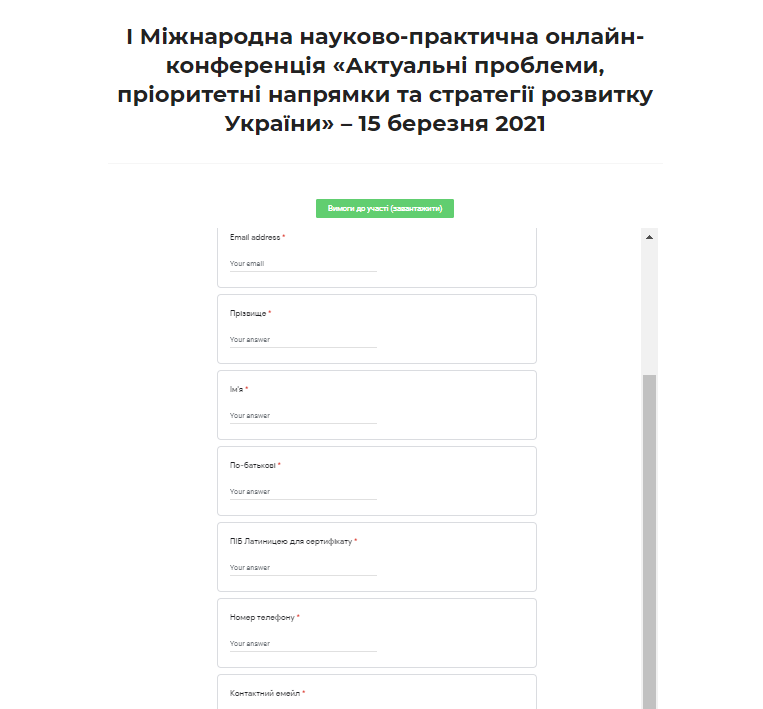


Рисунок 1.1 – Форма реєстрації на конференцію

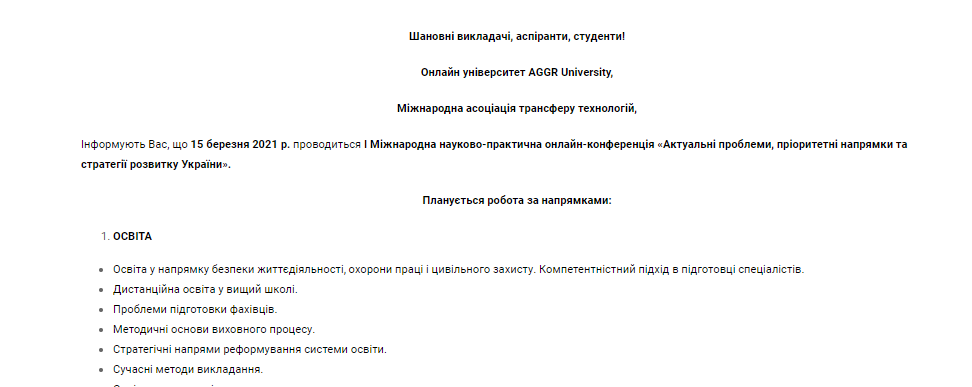


Рисунок 1.2 – Відображення загальної інформації конференції

Розглянемо приклад оформлення онлайн конференції, яка вже відбулась. Як видно з рис. 1.3, на сторінці архіву конференції відображається таблиця, в якій позначено секції, авторів та посилання на їх доповіді. Зверху вказана назва та дата конференції [5].

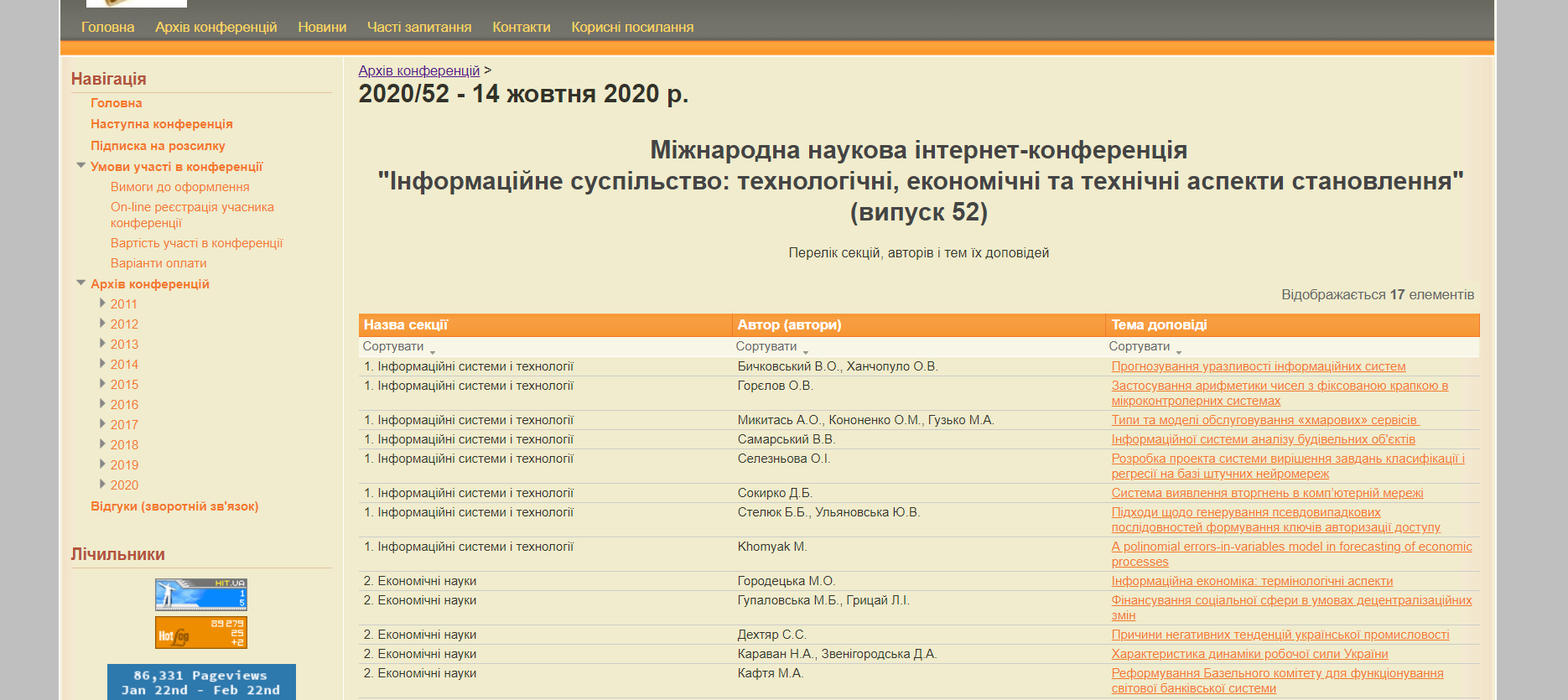


Рисунок 1.3 – Приклад відображення конференції, яка відбулася

1. **Постановка задачі дипломної роботи**

У кваліфікаційній роботі бакалавра необхідно спроектувати та програмно реалізувати інформаційну систему для проведення онлайн-конференцій, яка повинна забезпечувати наступні функції:

1. Наявність інформаційного блоку про поточну конференцію (загальна інформація, терміни, секції, новини тощо) та інформаційних блоків про минулі конференції.
2. Забезпечення можливостей адміністратора: керування ролей користувачів, створення та видалення новин, редагування контенту сайту.
3. Забезпечення можливостей користувача: завантаження на сайт доповідей, перегляд та можливість залишити коментарі.
4. Реалізація реєстрації та авторизації.
5. Можливість завантаження та перегляду файлів.
6. Створення панелі адміністратора з можливістю редагування бази даних.

**РОЗДІЛ 2**

**ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОВЕДЕННЯ ІНТЕРНЕТ-КОНФЕРЕНЦІЙ**

1. **Життєвий цикл розроблення програмного забезпечення**

Життєвий цикл розробки програмного забезпечення (SDLC) – це низка етапів, яких повинна виконувати команда розробників для розробки та обслуговування програмного забезпечення. Життєвий цикл SDLC починається з рішення про створення програмного забезпечення і закінчується виведенням його з експлуатації. Він складається з набору завдань, необхідних для виконання на кожному етапі процесу розробки.

SDLC допомагає знизити складність розробки системи з нуля. Важливо використовувати життєвий цикл розробки програмного забезпечення, оскільки це допомагає перетворити ідею проєкту в функціональну структуру. На додаток до висвітлення технічних аспектів розробки системи, SDLC допомагає у розробці процесів, управлінні змінами, користувацькому досвіді та політиках. Ще однією перевагою SDLC є те, що він дозволяє планувати заздалегідь, визначати витрати, визначати цілі,для підвищення якості кінцевого продукту.

Модель водоспаду – це лінійний підхід до управління проектами, де вимоги зацікавлених сторін та замовників збираються на початку проєкту, а потім створюється послідовний план проєкту, який відповідає цим вимогам. Модель водоспаду названа так, тому що кожна фаза проєкту переходить у наступну, вниз, як водоспад [7].

Етапи життєвого циклу розробки програмного забезпечення моделі водоспаду:

* + 1. Планування.
    2. Аналіз.
    3. Дизайн.
    4. Розробка.
    5. Тестування та інтеграція.
    6. Впровадження та випуск.
    7. Технічне обслуговування.

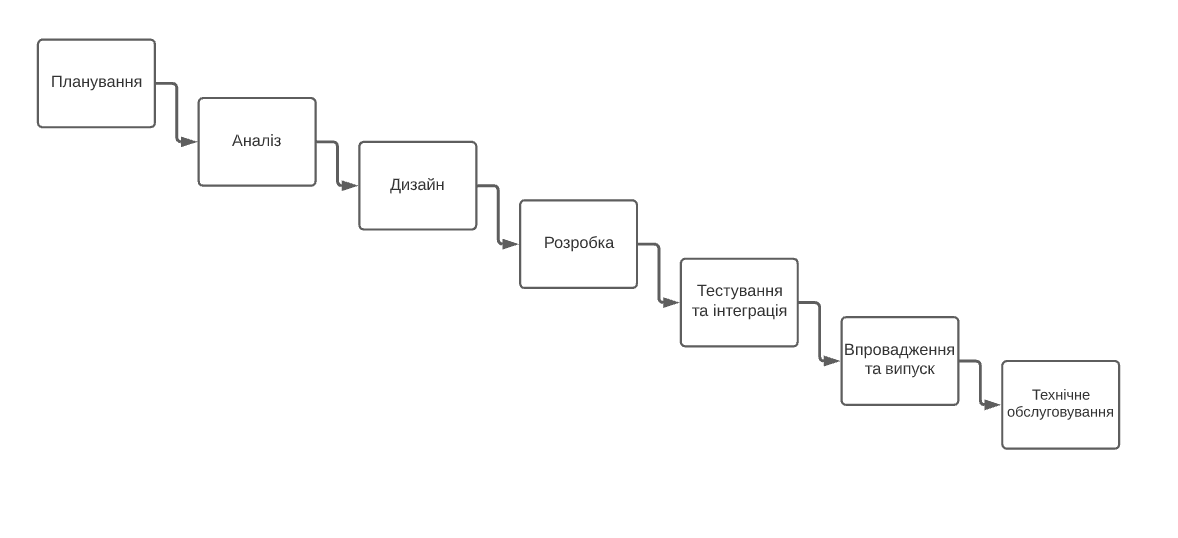


Рисунок 2.1 – Етапи життєвого циклу програмного забезпечення

* планування. Є найважливішим етапом процесу SDLC. Включає визначення обсягу проєкту для розуміння всебічного плану дій щодо проєкту та визначення проблем, які необхідно вирішити. Цей етап визначає, що відбуватиметься протягом усього життєвого циклу, і визначає успіх всього проєкту. На цьому етапі слід враховувати часові рамки, бюджет, безпеку та інші фундаментальні фактори [8];
* аналіз. Далі, необхідно провести аналіз функціональних вимог системи, щоб переконатися, що вона відповідає очікуванням цільової аудиторії. Потім ці очікування документуються. Наступним кроком є ​​проведення техніко-економічного обґрунтування, щоб переконатися, що система має сенс з фінансових, технологічних та організаційних позицій [8];
* дизайн. Після того, як аналіз закінчиться і вимоги сформулюються, починається фаза проєктування. Цей етап описує, як підходити до проєктування архітектури системи, користувацького інтерфейсу, мережі, баз даних та безпеки, щоб вони відповідали вимогам та дозволяли майбутні оновлення [8];
* розробка. Етап у якому проходить процес реалізації проєкту, де розробники пишуть код та допрацьовують технології, які будуть брати участь у кінцевому продукті [8];
* тестування та інтеграція. Цей етап включає тестування та інтеграцію системи та усіх пов'язаних з нею процедур для оцінки того, чи працює вона належним чином і повністю відповідає вимогам [8];
* впровадження та випуск. На цьому етапі нова система розгортається до виробничого середовища, замінюючи стару. Цей етап виконується шляхом переміщення нових даних та компонентів туди, де була стара система. Після впровадження система стане доступною для кінцевих користувачів [8];
* технічне обслуговування. Після випуску будь-яка програмна система переходить на стадію безперервного обслуговування. Сьогодні будь-який програмний продукт потрібно часто перевіряти на наявність помилок та оновлювати з урахуванням великих і малих функцій [8].

Спіральна модель – модель, що використовується для управління ризиками. Вона використовується при великих та затратних проєктах. У своєму схематичному зображенні модель виглядає як спіраль з безліччю петель. Точна кількість петель спіралі невідома і може варіюватися залежно від проєкту. Кожен цикл спіралі називається Фазою процесу розробки програмного забезпечення. Точна кількість фаз, необхідних для розробки продукту, може змінюватися керівником проєкту залежно від ризиків [9].

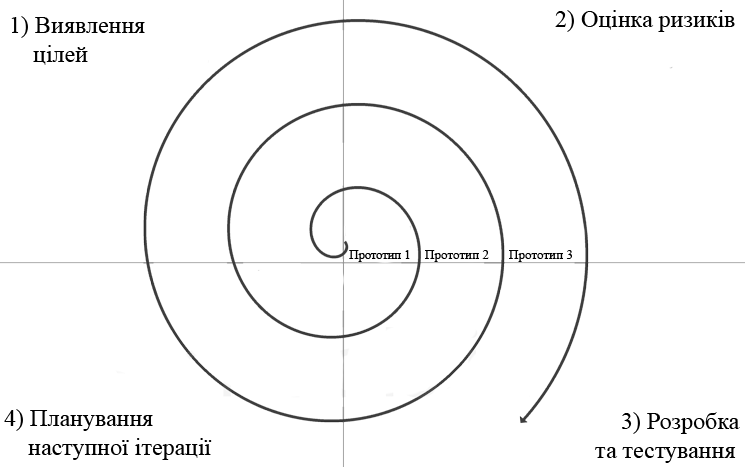


Рисунок 2.2 –Схема спіральної моделі життєвого циклу ПЗ

Кожна фаза спіральної моделі розділена на чотири квадранти, як показано на малюнку вище. Нижче представлений опис кожної функції моделі:

Виявлення цілей: в першому квадранті створюються певні вимоги на базі потреб користувачів та замовника. На базі вже цих вимог формуються цілі, які ставить перед собою проєкт [10].

Оцінка ризиків: під час другого квадранту оцінюються всі можливі рішення, та вибір кращого. Потім ідентифікуються ризики, пов’язані з цим рішенням, на базі цих ризиків формується стратегія [10].

Розробка та тестування: під час третього квадранту визначені функції продукту розробляються та перевіряються за допомогою тестування. Наприкінці третього квадранту доступна наступна версія програмного забезпечення [10].

Планування наступної ітерації: у четвертому квадранті Замовники оцінюють розроблену дотепер версію програмного забезпечення. У випадку, потреби доробки продукту, визначається нова фаза. Якщо продукт задовольняє потреби, процес зупиняється [10].

До переваг спіральної моделі можна віднести:

* управління ризиками. Якщо проєкт має велику кількість невідомих ризиків, що виникають у процесі розробки, в цьому випадку спіральна модель є найкращою моделлю розвитку, якій слід керуватися завдяки аналізу ризиків та обробці ризиків на кожному етапі [9];
* корисна для великих проєктів: рекомендується використовувати спіральну модель у великих та складних проектах [9];
* гнучкість у вимогах. За допомогою цієї моделі є можливість зміни вимог до проєкту у наступному етапі [9];
* задоволення клієнта. Клієнт може бачити розвиток продукту на ранній стадії розробки програмного забезпечення, і, отже, вони звикли до системи, використовуючи її до завершення всього продукту [9].

До недоліків спіральної моделі слід віднести:

* складність. Спіральна модель набагато складніша за інші моделі SDLC [9].
* велика ціна. Спіральна модель не підходить для невеликих проектів, оскільки вона вимагає багато ресурсів [9];
* залежність від планування ризиків. Успішне завершення проєкту дуже залежить від аналізу ризиків. Без досвідчених експертів не вдасться розробити проект за цією моделлю[9];
* складність в управлінні часом: Оскільки кількість фаз невідома на початку проєкту, тому оцінка часу дуже складна [9].

Методологія Agile - це спосіб управління проектом, розбивши його на кілька фаз. Він передбачає постійну співпрацю із зацікавленими сторонами та постійне вдосконалення на кожному етапі. Після того, як робота розпочнеться, групи циклічно переглядають процес планування, виконання та оцінки. Постійна співпраця життєво необхідна як з членами команди, так і з зацікавленими сторонами проєкту [10].

Agile методи або Agile процеси, як правило, сприяють дисциплінованому процесу управління проєктами, який заохочує часті перевірки та адаптацію, філософію керівництва, яка заохочує командну роботу, самоорганізацію та підзвітність, набір найкращих інженерних практик, призначених для швидкої доставки високоякісного програмного забезпечення, та діловий підхід, який узгоджує розвиток із потребами клієнтів та цілями компанії [10].

Оскільки продукт виконує одна людина, та проєкт не є масштабним, для розробки інформаційної системи онлайн-конференції була обрана модель водоспаду.

1. **Принцип роботи веб-сайту**

Вебсторінка – це простий текстовий файл, який також містить теги розмітки, що описують спосіб форматування тексту на екрані. Веб-сторінка зберігається на комп’ютері, який називається веб-сервером. Для того, щоб веб-сторінка відображалася на цьому комп’ютері або на іншому комп’ютері, вона повинна отримувати доступ та інтерпретувати спеціально розроблену програму, яка називається клієнтським програмним забезпеченням (коротше клієнт). Якщо клієнтське програмне забезпечення знаходиться на комп’ютері, відмінному від сервера, цей комп’ютер часто також називають клієнтом. Веб-браузер - це тип клієнтського програмного забезпечення, яке може запитувати код веб-сторінки з сервера через Інтернет, інтерпретувати розмітку та відображати його на екрані.

Для того, щоб подати запит на веб-сторінку, браузер повинен слідувати мережевому протоколу, набору правил щодо передачі даних. Одним із найпростіших для розуміння мережевих протоколів є протокол передачі файлів (FTP). На практиці клієнт FTP запитує у сервера точну копію файлу та зберігає його на клієнті (або навпаки). Хоча програми FTP не використовуються для безпосереднього доступу до розмітки веб-сторінок, програми FTP є важливими частинами веб-публікації, оскільки веб-сторінки, які можуть бути розроблені на ПК, повинні бути розміщені на веб-сервері, щоб бути доступними для Інтернету.

Веб-браузери використовують інший мережевий протокол, який називається протоколом передачі гіпертексту (HTTP). HTTP - це протокол прикладного рівня, призначений для передачі інформації між мережевими пристроями і працює поверх інших рівнів стеку мережевого протоколу. Типовий потік через HTTP включає клієнтську машину, яка робить запит на сервер, який потім надсилає відповідне повідомлення.

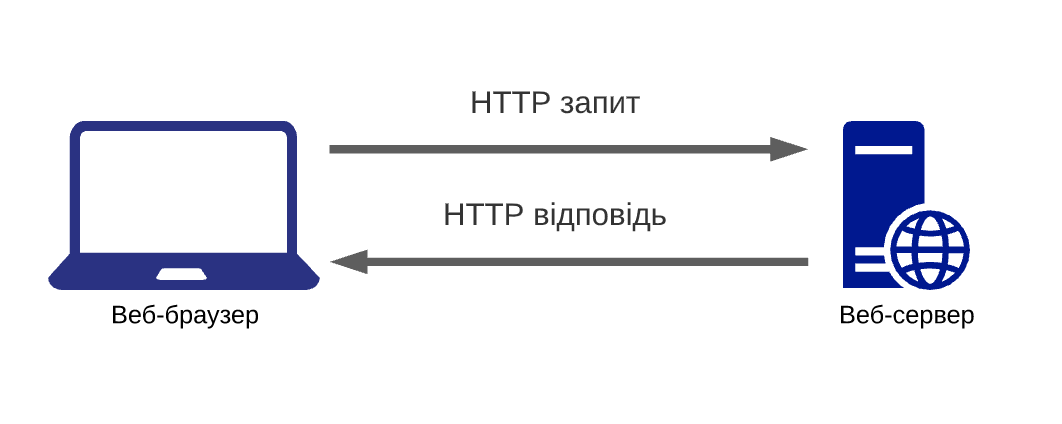


Рисунок 2.3 – Принцип роботи запитів HTTP

Кожен HTTP-запит, зроблений в Інтернеті, містить у собі ряд закодованих даних, що несуть різні типи інформації. Типовий HTTP-запит містить:

* тип версії HTTP;
* URL;
* метод HTTP;
* заголовки HTTP-запитів;
* необов’язковий корпус HTTP.

Метод HTTP, вказує на дію, яку HTTP-запит очікує від запитуваного сервера. Наприклад, два найпоширеніші методи HTTP - «GET» та «POST»; запит "GET" очікує інформацію у відповідь (зазвичай у формі веб-сайту), тоді як запит "POST" зазвичай вказує, що клієнт подає інформацію на веб-сервер (наприклад, інформацію про форму, надіслане ім'я користувача та пароль).

Заголовки HTTP містять текстову інформацію, що зберігається в парі ключ-значення, і вони включаються до кожного запиту HTTP. Ці заголовки передають основну інформацію, наприклад, який браузер використовує клієнт, які дані запитуються.

Приклад заголовків запитів HTTP із мережевої вкладки Google Chrome:

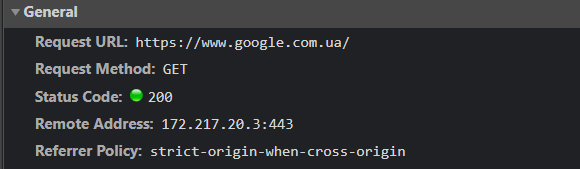


Рисунок 2.4 – Заголовки запитів HTTP

Нижче представлена схема, яка загально демонструє роботу веб-сайту.

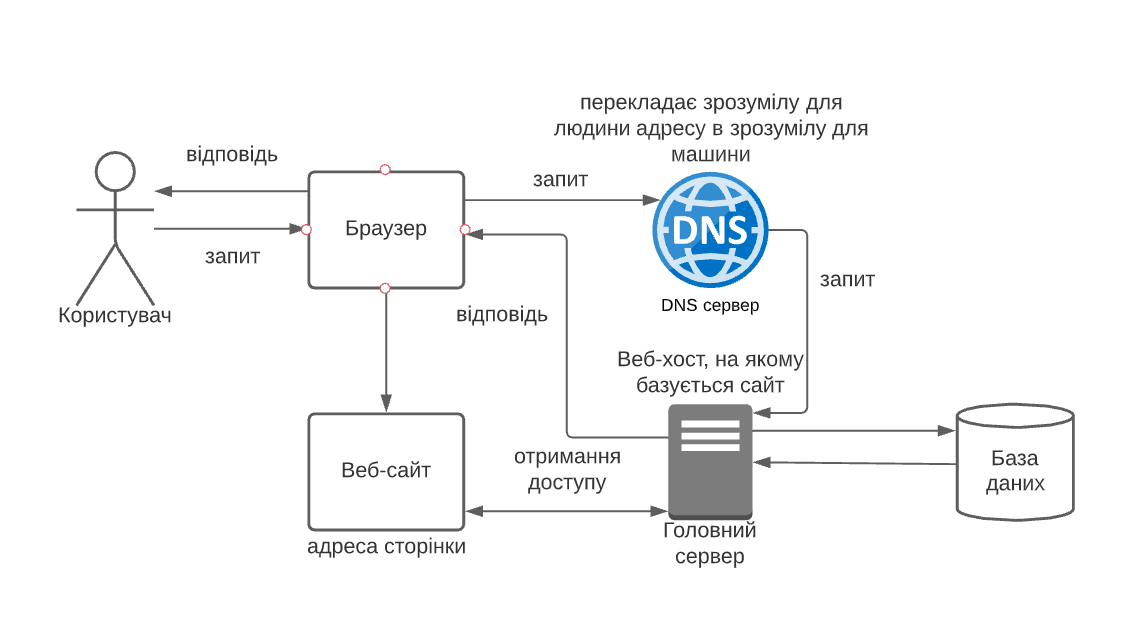


Рисунок 2.5 – Принцип роботи веб-сайт

# РОЗДІЛ 3 ПРОЦЕС РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН-КОНФЕРЕНЦІЙ

1. **Вибір та обґрунтування веб-технологій для розроблення  
    програмного забезпечення**
   * 1. **Вибір та обґрунтування використання мови PHP.** PHP (HypertextPreprocessor) - це мова сценаріїв, яка зазвичай використовується у веброзробці на стороні сервера. Мови сценаріїв - це підмножина мов кодування, що використовуються для автоматизації процесів, які в іншому випадку повинні були б виконуватися поетапно в коді сайту кожного разу, коли вони виникають.

PHP зазвичай використовується як мова на стороні сервера (на відміну від мови, такої як JavaScript, яка зазвичай виконується на стороні клієнта). На стороні клієнта йде звернення до діяльності вебсайту, яка відбувається локально на комп'ютері користувача через веббраузер користувача. Мови на стороні клієнта, такі як HTML, CSS та JavaScript, дають вказівки, що веб-браузери можуть аналізувати та перетворювати вміст на екран комп’ютера користувача [14].

До причин вибору мови програмування PHPналежать:

* популярність. Станом на 2021 рік, PHPвикористовується в 80% сайтів, які надають інформацію про мову написання. Серед них є такі актуальні сайти як: Facebook, Wikipedia, Tumblr [14];
* достатня документація. Доступно багато навчальних посібників та інших довідкових матеріалів, що полегшує розробку та може надати допомогу у важких ситуаціях. Ці матеріали також корисні для програмістів-початківців, що полегшує процес навчання. Вивчення PHP є порівняно простим, хоча і не таким простим, як вивчення Python, наприклад, що ще більше додає його популярності;
* покращена швидкість завантаження. Використання PHP призводить до швидшого завантаження сторінок вебсайтів порівняно з багатьма іншими технологіями веброзробки. Наприклад, нині, для більшості сценаріїв використання PHP приблизно втричі швидший, ніж Python. Менший час завантаження є важливим фактором рейтингу SEO, який допомагає надалі просувати вебсайт, забезпечуючи конкурентні переваги. Вища швидкість застосування забезпечує задоволення користувачів [14];
* широкий вибір баз даних. PHP дозволяє підключатися майже до будь-якого типу бази даних. Найпоширенішим вибором є MySQL, головним чином тому, що він безплатний, ефективний і популярний серед розробників. Іншими надійними варіантами систем управління базами даних, сумісними з PHP, є mSQL, MS-SQL, SQLite, PostgreSQL та ін. Крім того, PHP можна однаково добре використовувати з ElasticSearch, Redis, MongoDB та іншими нереляційними базами даних. Таким чином, розробники не обмежуються використанням певної бази даних і можуть вибрати оптимальну для майбутньої програми, беручи до уваги всі відповідні фактори [14];
* недороге ПЗ з відкритим кодом. PHP - це безплатна технологія, яка забезпечує значну економію бюджету розвитку. Крім того, більшість засобів розробки, які зазвичай використовуються в поєднанні з PHP, є програмним забезпеченням з відкритим кодом і можуть використовуватися безплатно; таким чином, вони додатково знижують вартість проєкту [14];
* дешевші послуги хостингу. Найбільш поширеним сценарієм роботи веб-сайту PHP є стек XAMP. Це означає, що вебсайт працює на вебсервері Apache HTTP, розгорнутому в системі Windows, і використовує MySQL як базу даних. Всі ці компоненти безплатні, а стек добре перевірений, що передбачає скорочення необхідного часу та коштів на розробку [14];
* сумісність з HTML. PHP пропонує вбудоване програмування HTML, що є причиною неймовірної синергії між цими двома технологіями. У більшості випадків PHP-скрипт не втручається в HTML-код вебсторінки, а замість цього завершує його, залишаючись всередині меж, визначених тегами. І навпаки, при створенні цілої сторінки коду PHP, є можливість інтегрування HTML-коду за допомогою сценарію. У цьому випадку відмінністю від традиційної HTML-сторінки буде перевернутий порядок відкриття тегів: PHP-теги передують HTML-тегам;
* гнучкість. Гнучкість дозволяє PHP ефективно поєднувати з багатьма іншими мовами програмування, щоб програмний продукт міг використовувати найбільш ефективну технологію для кожної конкретної функції. Щобільше, PHP є міжплатформовою мовою, що означає, що розробники можуть використовувати будь-яку основну операційну систему – Windows, Linux, macOS– для виконання кодування. Така гнучкість значно полегшує процес розробки, роблячи його швидшим і дешевшим [14];
* сумісність із хмарними сервісами. Нині, багато сучасних продуктів, як правило, використовують хмарні обчислювальні рішення, такі як AmazonWebServices, для різних цілей. Додатки, написані на PHP, підтримуються різними хмарними сервісами. Таким чином, PHP-програму можна розгорнути на хмарному сервері та досягти чудової масштабованості та інших корисних ефектів. Щобільше, область хмарних обчислень не монополізована іншими мовами кодування, тому PHP зайняв своє місце в таких реалізаціях [14].
  + 1. **Вибір та обґрунтування використання MySQL.** MySQL – це система управління базами даних, яка дозволяє управляти реляційними базами даних. Це програмне забезпечення з відкритим кодом, яке підтримується компанією Oracle. Відкритий код означає, що кожен може використовувати та модифікувати програмне забезпечення. Будь-хто може завантажити програмне забезпечення MySQL з Інтернету та користуватися ним. При необхідності є можливість редагування вихідного коду. Програмне забезпечення MySQL використовує GPL для ознайомлення із можливостями програмного забезпечення у різних ситуаціях [15].

Реляційна база даних зберігає дані в окремих таблицях, а не зберігає всі дані в одному великому сховищі. Структури баз даних організовані у фізичні файли, оптимізовані для швидкості. Логічна модель з такими об'єктами, як бази даних, таблиці, подання, рядки та стовпці, пропонує гнучке середовище програмування. Ви встановлюєте правила, що регулюють взаємозв'язки між різними полями даних, такими як "один до одного", "один до багатьох", унікальний, обов'язковий або необов'язковий, і "вказівники" між різними таблицями. База даних застосовує ці правила, так що завдяки добре розробленій базі даних ваша програма ніколи не бачить невідповідних, дублікатів, застарілих, застарілих або відсутніх даних [15].

Програмне забезпечення баз даних MySQL – це система клієнт / сервер, яка складається з багатопотокового SQL-сервера, що підтримує різні фони, кілька різних клієнтських програм та бібліотек, адміністративні засоби та широкий спектр інтерфейсів прикладного програмування (API).

Частина SQL в "MySQL" означає "Мова структурованих запитів". SQL є найпоширенішою стандартизованою мовою, що використовується для доступу до баз даних. Залежно від середовища програмування, ви можете вводити SQL безпосередньо (наприклад, для створення звітів), вбудовувати оператори SQL у код, написаний іншою мовою, або використовувати специфічний для мови API, який приховує синтаксис SQL [15].

Сервер MySQL може працювати на настільному комп'ютері або ноутбуці, поряд з іншими програмами, вебсерверами тощо, не вимагаючи майже ніякої уваги. При використанні всієї машини як сервер, є можливість налаштування параметрів, щоб скористатися всією доступною пам’яттю, потужністю центрального процесора та місткістю вводу-виводу. MySQL також може масштабуватися до кластерів машин, об'єднаних у мережі [17].

SQL визначається стандартом ANSI / ISO SQL.

До причин вибору СУБД MySQL можна віднести:

* безпека даних. MySQL відомий у всьому світі як найбезпечніша та найнадійніша система управління базами даних, що використовується у таких популярних веб-додатках, як WordPress, Drupal, Joomla, Facebook та Twitter. Захист даних та підтримка обробки транзакцій, які супроводжують останню версію MySQL, можуть мати велику користь для будь-якого бізнесу, особливо якщо це електронна комерція, яка передбачає часті грошові перекази;
* масштабованість за запитом. MySQL пропонує неперевершену масштабованість, щоб полегшити управління глибоко вбудованими програмами, використовуючи менший розмір, навіть у масивних сховищах, які складають терабайт даних. Гнучкість на вимогу - головна особливість MySQL. Це рішення з відкритим кодом дозволяє повністю налаштувати бізнес електронної комерції з унікальними вимогами до сервера баз даних;
* висока продуктивність. MySQL має чітку структуру механізму зберігання, яка полегшує системним адміністраторам налаштовувати сервер баз даних MySQL для бездоганної продуктивності. Будь то вебсайт електронної комерції, який щодня отримує мільйон запитів, або високошвидкісна система обробки транзакцій, MySQL розроблена для задоволення навіть найвимогливіших програм, забезпечуючи при цьому оптимальну швидкість, повнотекстові індекси та унікальний кеш-пам'яті для підвищення продуктивності;
* цілодобовий час роботи. MySQL постачається з гарантією безперебійної роботи і пропонує широкий спектр доступних рішень, таких як спеціалізовані кластерні сервери;
* комплексна транзакційна підтримка. MySQL очолює список надійних механізмів транзакційних баз даних, доступних на ринку. Завдяки таким функціям, як повна атомна, послідовна, ізольована, довговічна підтримка транзакцій, підтримка транзакцій з декількома версіями та необмежене блокування на рівні рядків, це рішення для повної цілісності даних. Це гарантує миттєву ідентифікацію тупикової ситуації завдяки примусовому посиланню цілісності;
* повне управління робочим процесом. Оскільки середній час завантаження та встановлення становить менш як 30 хвилин, MySQL означає зручність використання з першого дня. Незалежно від того, яка платформа користувача: Linux, Microsoft, Macintosh або UNIX, MySQL - це комплексне рішення з функціями самокерування, які автоматизують все, від розширення та налаштування простору до проєктування даних та адміністрування баз даних;
* гнучкість відкритого коду. Безпечна обробка та надійне програмне забезпечення MySQL поєднують в собі ефективні транзакції для великих обсягів проєктів. Це робить обслуговування, налагодження та оновлення швидким та простим, одночасно покращуючи взаємодію з кінцевим користувачем.
  + 1. **Вибір та обґрунтування використання XAMPP.** XAMPP – багатоплатформова збірка вебсервера, яка міститьHTTP-сервер Apache, інтерпретатори скриптів PHP та Perl та базу даних MySQL.

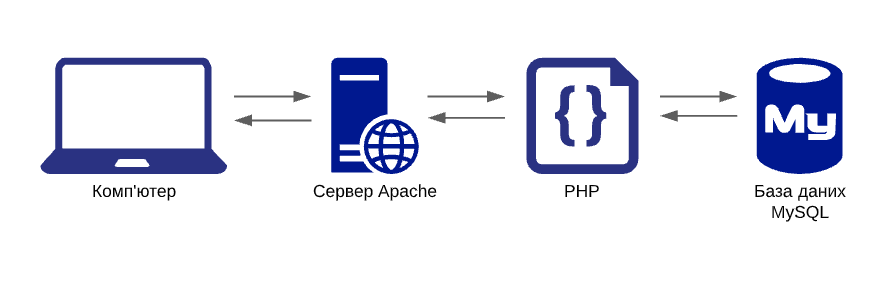


Рисунок 3.1 – Схема роботи сайту на локальному комп’ютері

Основними причинами вибору збірки XAMPP можна вважати:

* його легко встановити, у порівнянні з іншими вебсерверами, такими як WAMP;
* крос-платформеність. XAMPP доступний для Windows та Linux;
* постачається з низкою інших модулів, включаючи OpenSSL, phpMyAdmin, MediaWiki, Joomla, WordPress та багато іншого;
* XAMPP пропонується як у повній, так і в стандартній версії (менша версія);
* можливість запустити і зупинити весь стек вебсервера та бази даних однією командою.
  + 1. **Вибір та обґрунтування використання фреймворку Codeigniter**. Codeigniter - це фреймворк PHP MVC, який використовується для швидкої розробки вебдодатків. Codeigniter пропонує готові бібліотеки для підключення до бази даних та виконання різних операцій, таких як надсилання електронних листів, завантаження файлів, управління сеансами тощо.

До переваг цього фреймворку можна віднести наступні пункти:

* розмір. Весь вихідний код фреймворку Codeigniter близький до 2 МБ. Це полегшує освоєння Codeigniter та спосіб його роботи. Це також спрощує його розгортання та оновлення;
* швидкість. Codeigniter, завантажується в середньому близько 50 мс. Додатковий час, витрачений на оптимізацію, як у випадку з іншим фреймворком, звільняється під час роботи з фреймворком Codeigniter;
* автономність. Вбудовані функції призначені для незалежної роботи, не надто покладаючись на інші компоненти. Це полегшує обслуговування та модернізацію;
* архітектура MVC. Фреймворк PHP Codeigniter використовує архітектурний дизайн Model-View-Controller. Це стандартна галузева практика роботи з вебдодатками. MVC розділяє дані, ділову логіку та презентацію [16];

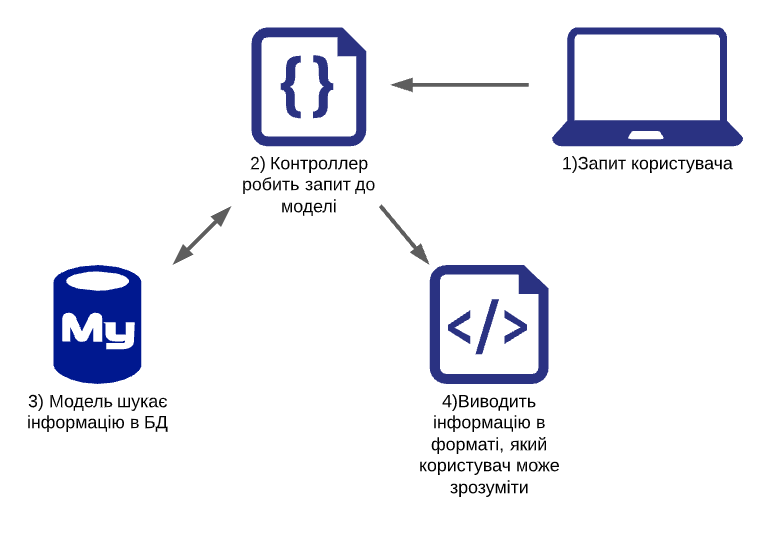


Рисунок 3.2 – Принцип роботи MVC

* зрозуміла документація. Структура добре задокументована, а на Codeigniter є хороші книги, навчальні посібники та відповіді на запитання форуму. Це означає, що з будь-якою проблемою, яка у вас є, є ймовірність, що хтось уже стикався з проблемою, вирішував її, і рішення є для вас;
* вбудовані компоненти. Codeigniter має компоненти для надсилання електронної пошти, управління базами даних, управління сесіями та багато іншого, про що ви дізнаєтесь, продовжуючи навчальні посібники;
* доступні розширення. Codeigniter постачається з деякими бібліотеками та помічниками. Якщо того, чого ви хочете, немає, або ви хотіли б застосувати функцію, що існує,чё за вашим способом. Тоді ви можете зробити це легко, створивши свої бібліотеки, помічники, пакети тощо;
* складність. Codeigniter легко освоїти для тих, хто вже знайомий з PHP. За дуже короткий час студент може вивчити Codeigniter і розпочати розробку професійних додатків;

1. **Проєктування інформаційної системи**

Результатом етапу проєктування ІС є розробка документа, що описує всі аспекти створюваного сайту: схему розташування елементів дизайну, структуру і особливості системи навігації, функціональність і особливості  роботи кожного модуля.

Будуємо контекстну діаграму типу IDEF0 в якій зображено взаємозв’язок процесу розробки ІС з зовнішнім середовищем.

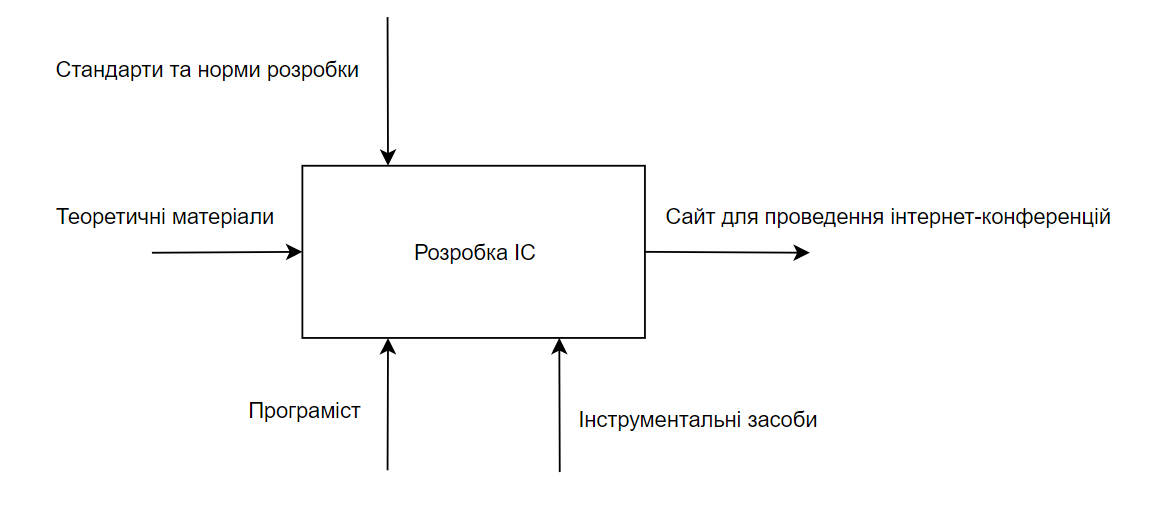


Рисунок 3.3 – Контекстна діаграма процесу розробки сайту

В якості входу виступають теоретичні матеріали по принципу роботи сайту та життєвого циклу розробки ПЗ, з боку керування виступають стандарти розробки сайту, затвердженні W3С. Механізмом виступає програміст, який буде виконувати роботу, та інструментальні засоби (мови програмування, фреймворки, веб-сервер, база даних тощо), з допомогою яких буде виконуватися робота. На виході отримуємо готовий сайт для проведення інтернет-конференцій.

На основі контекстної діаграми будуємо діаграму декомпозиції.

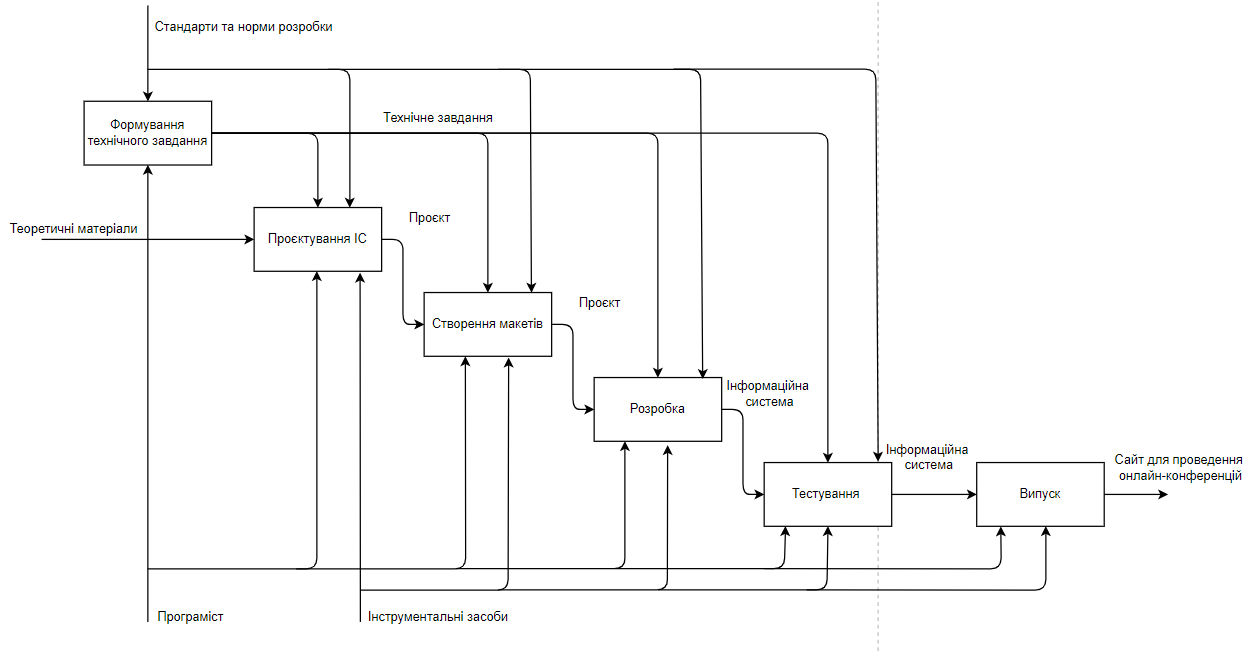


Рисунок 3.4 – Діаграма декомпозиції процесу розробки сайту

Тепер, за допомогою діаграми декомпозиції визначаємо основні етапи створення інформаційної системи:

1. Формування чіткого технічного завдання, який описує необхідні функції сайту. В цьому випадку, до основних можливостей можна віднести: авторизацію/реєстрацію користувачів, завантаження на сайт файлів доповідей та реалізація блоку коментарів.
2. Проєктування ІС. Виділення усіх етапів розробки, планування логіки роботи модулів сайту.
3. Створення макетів. Попередня реалізація виду майбутнього сайту, створення структури сайту.
4. Розробка. Повна реалізація усіх функцій сайту, під’єднання бази даних, верстка макету та поєднання верстки із логікою.
5. Тестування. Перевірка роботи усіх функцій сайту, перевірка верстки.
6. Випуск. Завершення роботи.

Для формування логіки роботи модулів сайту, створимо діаграму варіантів використання, яка буде зображувати ролі акторів та їх можливості роботи в ІС.

Діаграма варіантів використання - це спосіб узагальнити деталі системи та користувачів у цій системі. Як правило, це показано як графічне зображення взаємодій між різними елементами в системі. Діаграми випадків використання вказуватимуть події в системі та те, як ці події протікають, однак діаграма випадків використання не описує, як ці події реалізуються.

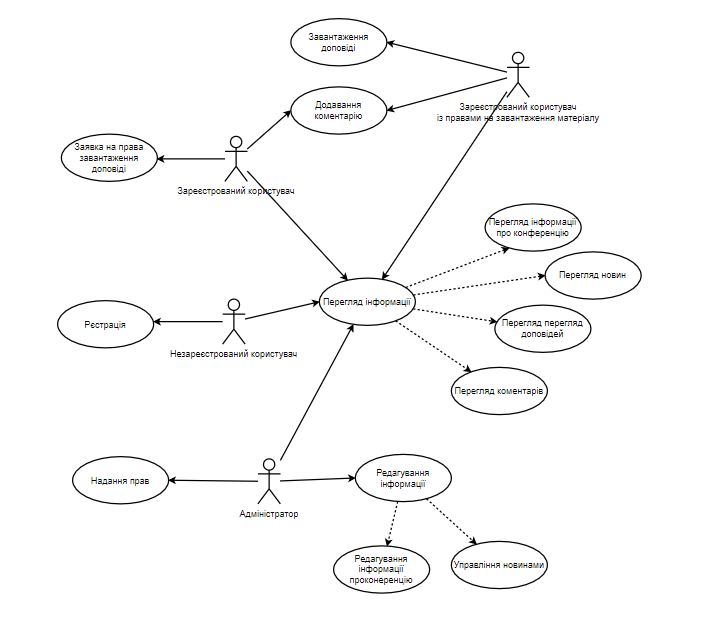


Рисунок 3.5 – Діаграма варіантів використання для ІС інтернет-конференції

Таблиця 3.1 – Виявлення акторів

|  |  |
| --- | --- |
| Актор | Короткий опис |
| Незареєстрований користувач | Має можливість перегляду інформації на ресурсі. Може пройти реєстрацію для збільшення можливостей |
| Зареєстрований користувач | Має можливість перегляду інформації на ресурсі та створення коментарів до робіт інших користувачів. Може заповнити форму на права завантаження доповіді для збільшення можливостей |
| Зареєстрований користувач із правами на завантаження матеріалу | Має можливість перегляду інформації на ресурсі та створення коментарів до робіт інших користувачів, завантаження своїх робіт. |
| Адміністратор ресурсу | Має доступ для перегляду інформації ІС та її редакції, управління новинами, та надання прав іншим користувачам. |

Логіка модулей ІС зображена діаграмами діяльності. Діаграма діяльності виражає послідовність кроків, необхідних для формування складного процесу. Діаграма діяльності зображає потік управління, як звичайна діаграма, але розглядається як його вдосконалена версія.

Розглянемо діаграму діяльності для надання користувачу прав на завантаження файлу.

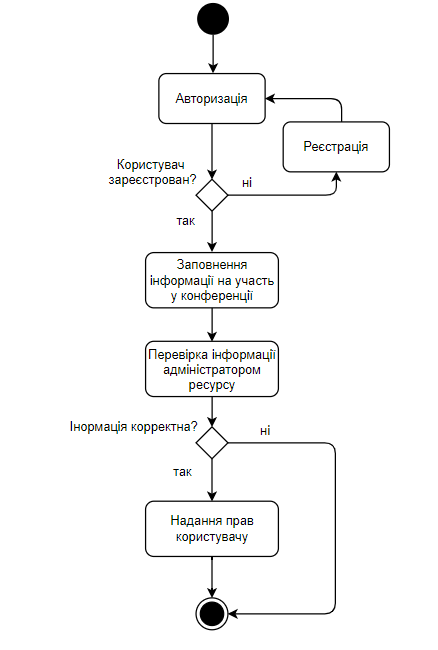


Рисунок 3.6 – Діаграма діяльності для надання користувачу прав на завантаження файлу.

Спочатку виконується авторизація користувача; якщо користувач не зареєстрований, він повинен пройти реєстрацію. Після цього користувач заповнює форму на участь у конференції. Далі адміністратор перевіряє коректність заповненої інформації та, у разі успіху надає необхідні права користувачу.

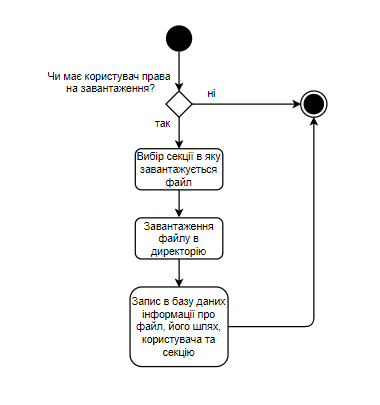


Рисунок 3.7 –Діаграма діяльності для завантаження документу на сайт

Спочатку йде перевірка прав користувача. У випадку наявності їх, користувач обирає секцію, до якої належить його доповідь. Після того, як користувач натискає на кнопку «Додати», файл зберігається в директорію проекту. Після цього, контролер записує в таблицю бази даних інформацію про файл, його шлях в директорії, користувача, який завантажив файл та секцію, яку він обрав.

1. **Розробка структури сайту та макетів**

Розглянемо структуру сайту для ІС, який містить усі необхідні модулі (рис. 3.8).

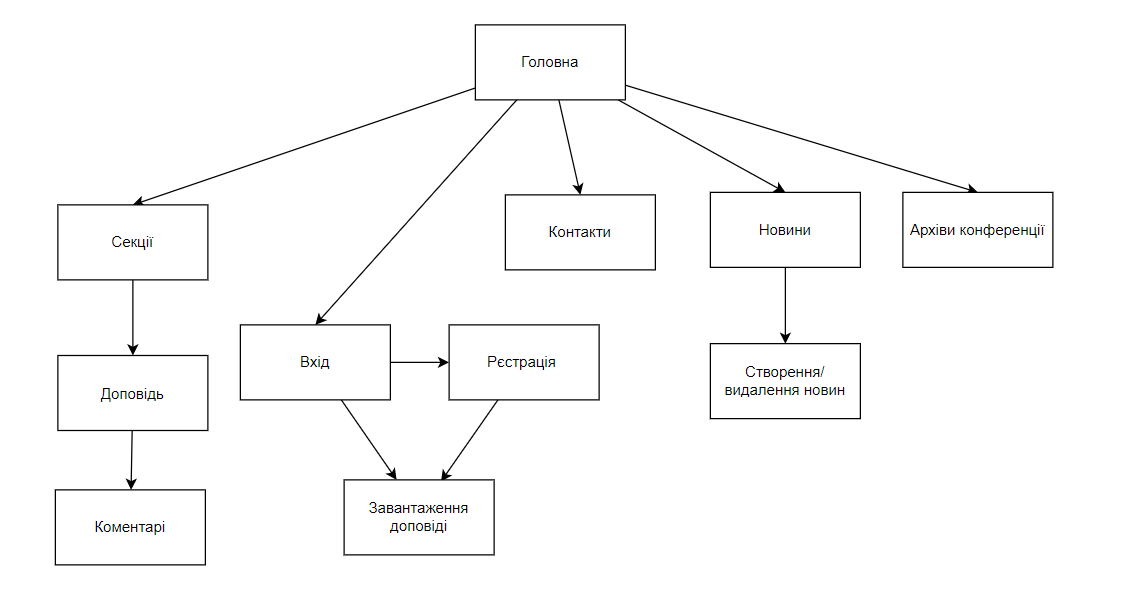


Рисунок 3.8 – Структура сайту

Від головної сторінки є можливість потрапити на будь-який модуль. Модуль секцій містить різні направлення наукових робіт. При виборі однієї із секцій, виходимо на перегляд окремих доповідей по цій темі із можливістю коментарів.

Модуль входу в систему приводить до реєстрації в системі, у разі, якщо користувач не зареєстрований. Після проходження необхідних умов, користувач отримує доступ до завантаження власної роботи.

Модуль контактів – інформаційна сторінка, яка містить контактну інформацію людей, пов’язаних із організацією конференції.

Модуль новостей відображає новини. Адміністратор сайту має доступ до редакції.

Архіви конференції містять в собі минулі конференції.

На основі структури ІС створюємо макет основних сторінок сайту.

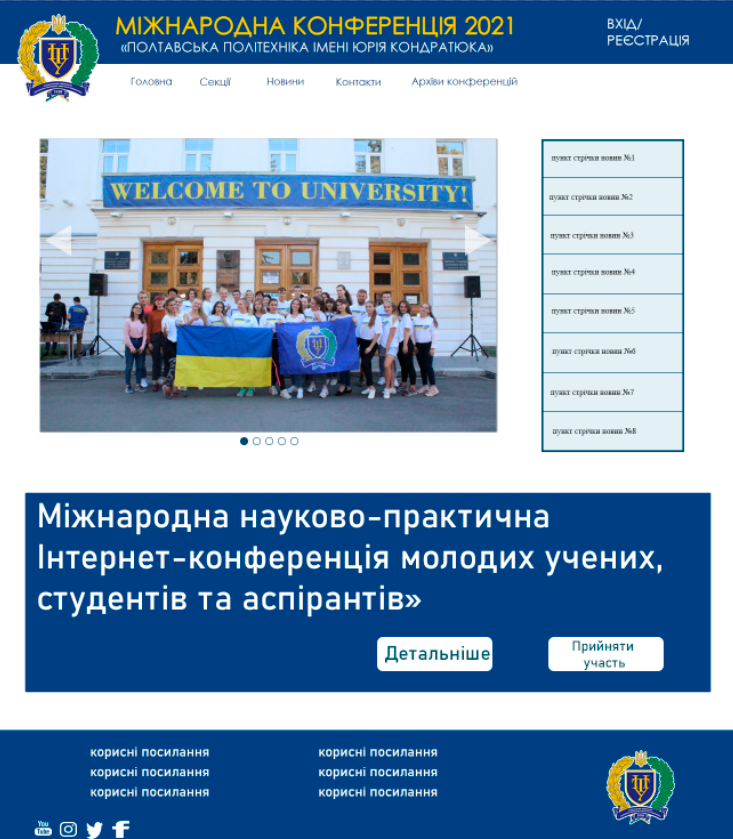


Рисунок 3.9 – Сторінка макету головної сторінки сайту

Бачимо, на головній сторінці розташована інформація про конференцію, вкладки, по яким користувач може переходити, короткий блок новин та кнопка авторизації.

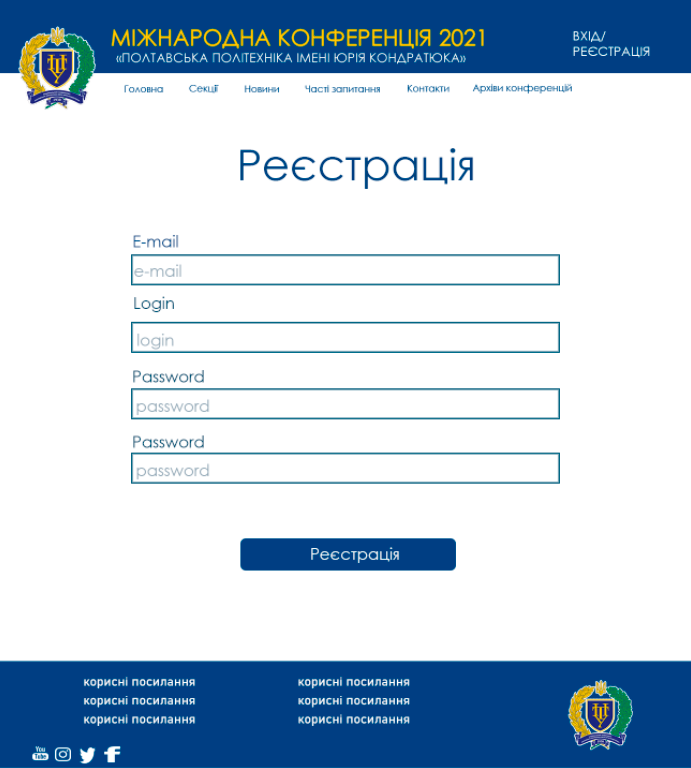


Рисунок 3.10 – Сторінка макету реєстрації

В макеті реєстрації користувач повинен занести свої дані для роботи із сайтом. Після проходження реєстрації, користувач може заповнити форму для отримання можливості завантаження своїх робіт.

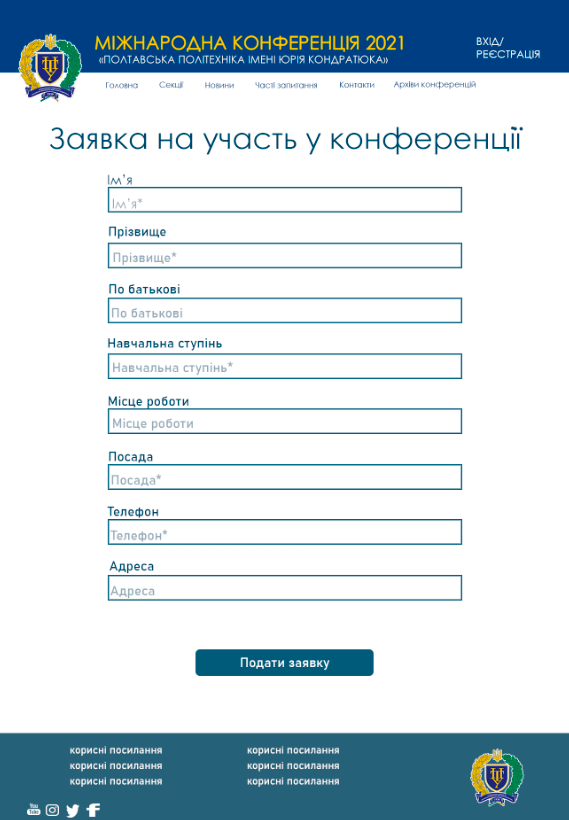


Рисунок 3.11 – Сторінка макету заявки на участь

Тут користувач заповнює форму участі на конференцію.

1. **Розробка бази даних**

Під час розробки інформаційної системи сайту онлайн-конференції була створена база даних, в яку входять наступні таблиці (рис. 3.10):

* конференція;
* новини;
* секції доповідей;
* файли;
* коментарі;
* користувачі;
* ролі;
* інформація про користувачів.

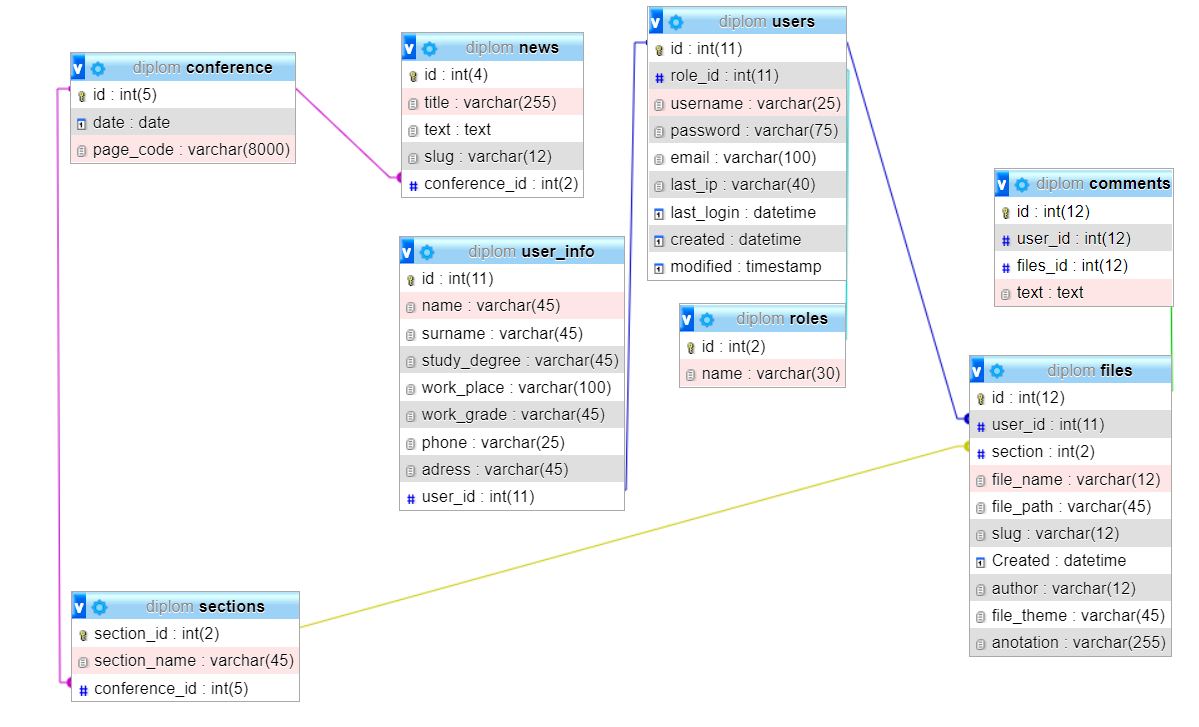


Рисунок 3.12 – Схема бази даних

Конференція – таблиця, в якій зберігається інформація про поточну конференцію.

Таблиця 3.2 – Таблиця «conference»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Назва | Тип даних | Розмір | Опис |
| 1 | id | integer | 5 | Первинний ключ таблиці |
| 2 | date | date | date | Дата поточної конференції |
| 3 | page\_code | varchar | 8000 | Код конференції |

Новини – таблиця, в якій зберігається інформація про новини на сайті.

Таблиця 3.3 – Таблиця «news»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Назва | Тип даних | Розмір | Опис |
| 1 | id | integer | 4 | Первинний ключ таблиці |
| 2 | title | varchar | 255 | Заголовок новини |
| 3 | text | text |  | Текст новини |
| 4 | slug | varchar | 12 | URL ідентифікатор |
| 5 | conference\_id | integer | 5 | Зовнішній ключ таблиці conference |

Секції доповідей – таблиця яка містить інформацію про секції.

Таблиця 3.4– Таблиця «section»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Назва | Тип даних | Розмір | Опис |
| 1 | section\_id | integer | 2 | Первинний ключ таблиці |
| 2 | section\_name | varchar | 45 | Назва секції |
| 3 | conference\_id | integer | 5 | Зовнішній ключ таблиці conference |

Файли – таблиця, яка містить інформацію про файли, які завантажують користувачі.

Таблиця 3.5 – Таблиця «files»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Назва | Тип даних | Розмір | Опис |
| 1 | id | integer | 12 | Первинний ключ таблиці |
| 2 | user\_id | integer | 11 | Зовнішній ключ таблиці users |
| 3 | section | integer | 11 | Зовнішній ключ таблиці section |
| 4 | file\_name | varchar | 12 | Назва файлу |
| 5 | file\_path | varchar | 45 | Повний шлях до файлу |
| 6 | slug | varchar | 12 | URL ідентифікатор |
| 7 | created | datetime |  | Дата завантаження файлу на сайт |
| 8 | author | varchar | 12 | Автори доповіді |
| 9 | file\_theme | varchar | 45 | Тема доповіді |
| 10 | anotation | varchar | 255 | Анотація до доповіді, поле може бути порожнім |

Коментарі – таблиця, яка містить інформацію про коментарі до доповідей.

Таблиця 3.6 – Таблиця «comments»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Назва | Тип даних | Розмір | Опис |
| 1 | id | integer | 12 | Первинний ключ таблиці |
| 2 | user\_id | integer | 12 | Зовнішній ключ таблиці users |
| 3 | files\_id | integer | 12 | Зовнішній ключ таблиці files |
| 4 | text | text |  | Текст коментаря |

Користувачі – таблиця яка містить інформацію про користувачів, які зареєстровані на сайті.

Таблиця 3.7 – Таблиця «users»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | назва | Тип даних | Розмір | Опис |
| 1 | id | integer | 11 | Первинний ключ таблиці |
| 2 | role\_id | integer | 11 | Зовнішній ключ таблиці roles |

Продовження таблиці 3.7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | назва | Тип даних | Розмір | Опис |
| 3 | username | varchar | 11 | Ім’я користувача |
| 4 | password | varchar | 12 | Пароль користувача |
| 5 | email | varchar | 45 | e-mail користувача |
| 6 | last\_ip | varchar | 12 | Ip адреса користувача |
| 7 | last\_login | datetime |  | Дата та час останнього відвідування сайту |
| 8 | created | datetime |  | Дата та час створення аккаунту |

Ролі ­– таблиця із ролями користувачів.

Таблиця 3.8 – Таблиця «user\_info»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Назва | Тип даних | Розмір | Опис |
| 1 | id | integer | 11 | Первинний ключ таблиці |
| 2 | name | varchar | 45 | Ім’я користувача |
| 3 | surname | varchar | 45 | Прізвище користувача |
| 4 | study\_degree | varchar | 11 | Научний ступінь користувача, поле може бути порожнім |
| 5 | work\_place | varchar | 100 | Місце роботи, поле може бути порожнім |
| 6 | work\_grade | varchar | 45 | Посада користувача, поле може бути порожнім |
| 7 | phone | varchar | 25 | Номер телефону користувача |
| 8 | adress | varchar | 45 | Адреса користувача |
| 9 | user\_id | integer | 11 | ID користувача |

Інформація про користувачів –таблиця в якій зберігається інформація про користувачів, які відправили заявку на участь у конференції.

Таблиця 3.9 – Таблиця «roles»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Назва | Тип даних | Розмір | Опис |
| 1 | id | integer | 2 | Первинний ключ таблиці |
| 2 | name | varchar | 30 | Назва ролі |

1. **Реалізація програмного продукту**

На основі макету було створено головну сторінку сайту (рис.3.11).

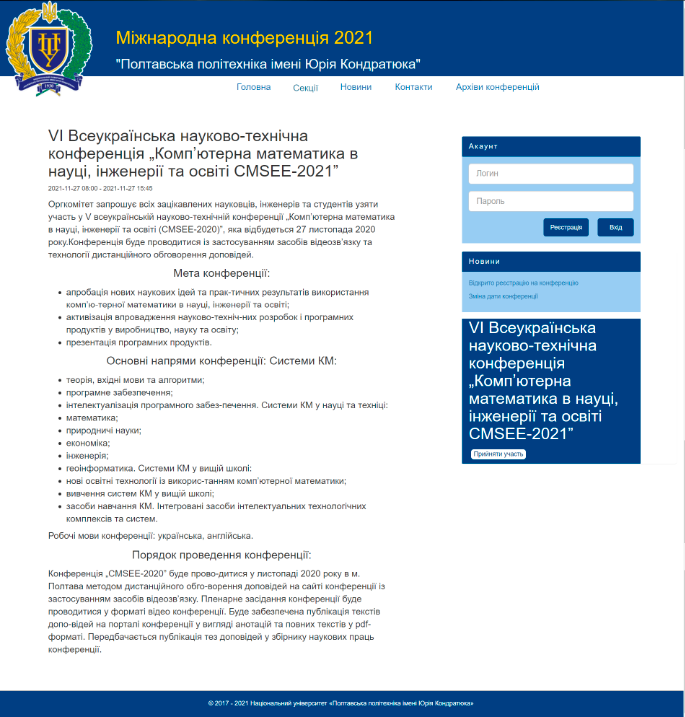


Рисунок 3.13 – Головна сторінка сайту

Головна сторінка сайту містить інформацію про поточну конференцію. З лівого боку розташовано блок новин, посилання на реєстрацію конференції та панель авторизації. Через панель авторизації користувач може зайти у свій акаунт. Якщо користувач не має акаунту він може натиснути на кнопку «Реєстрація». Він потрапить на сторінку реєстрації де має заповнити реєстраційну форму(рис. 3.12).

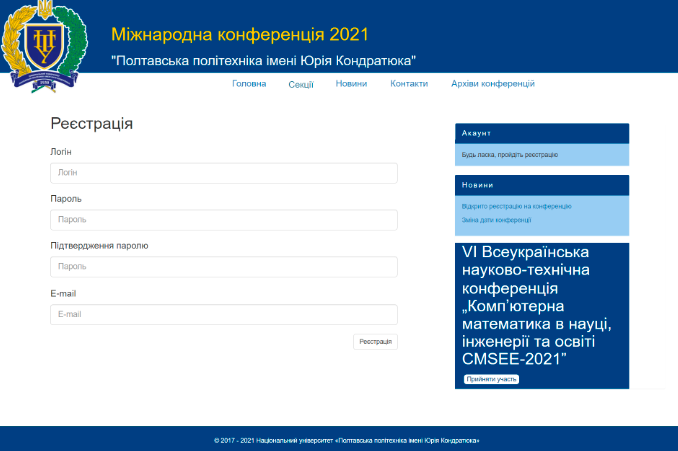


Рисунок 3.14 –Сторінка реєстрації

Під час заповнення форми користувач повинен вказати свій логін, пароль та e-mail. Після цього користувач отримує можливість коментування робіт інших користувачів та подачі заявки на участь у конференції(рис.3.13).

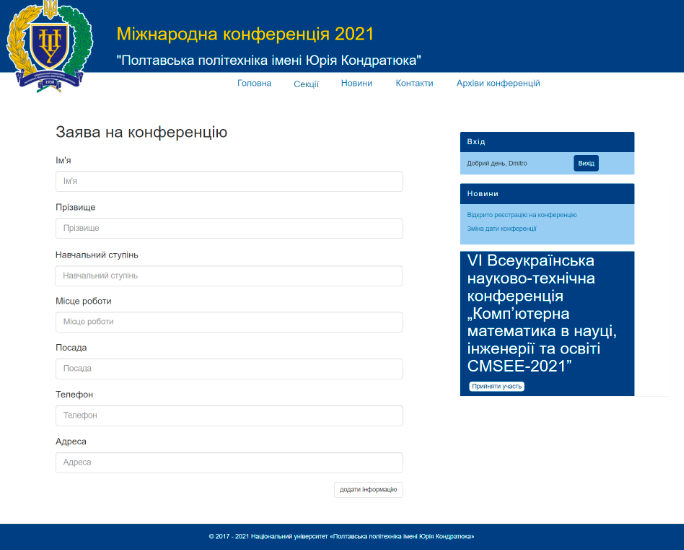


Рисунок 3.15 – Заява на конференцію

В процесі заповнення форми, користувач має вказати свої персональні дані. Потім ці дані перевіряє адміністратор сайту. Якщо дані проходять перевірку, адміністратор надає право на завантаження документів на сайт. Для цього було створене спеціальне вікно адміністратора, у якому зображено усю поточну інформацію користувачів, які надіслали свої заявки на участь у конференції(рис. 3.14).

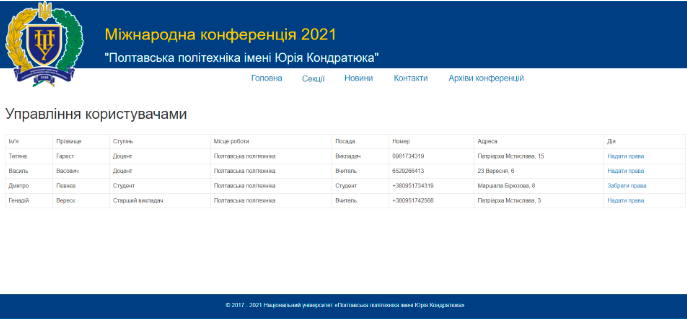


Рисунок 3.16 – Сторінка адміністратора

У випадку, якщо за таким же посиланням спробує пройти користувач без прав адміністратора, він отримає помилку(рис. 3.15).

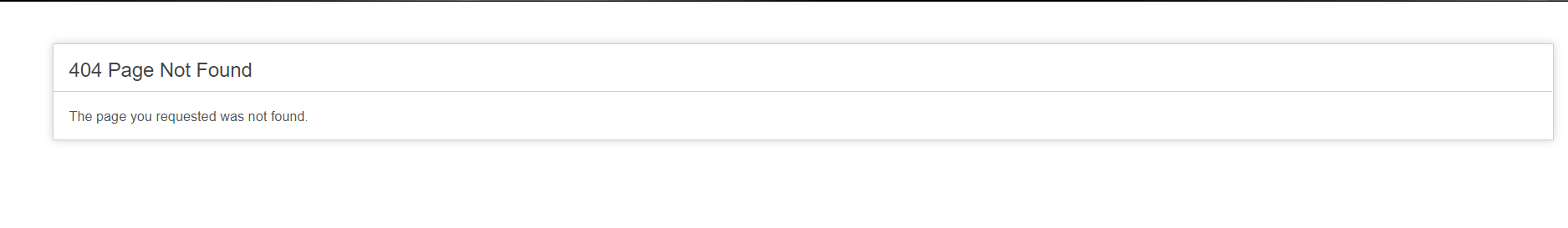


Рисунок 3.17 – Помилка пошуку сторінки

За це відповідає рядок коду перевірки на адміністратора «is\_admin()»(рис. 3.16).

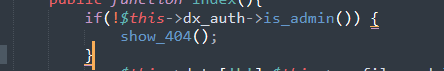


Рисунок 3.18 – Частина коду

Проходить запит по поточній сесії користувача в базу даних, звідти дістається інформація про поточну роль користувача.

На сторінці із завантаженням доповіді, користувач зустрічається із новою формою, де має вказати тему своєї доповіді, обрати секцію, та вписати авторів. За бажанням він може додати анотацію до доповіді(рис. 3.17).

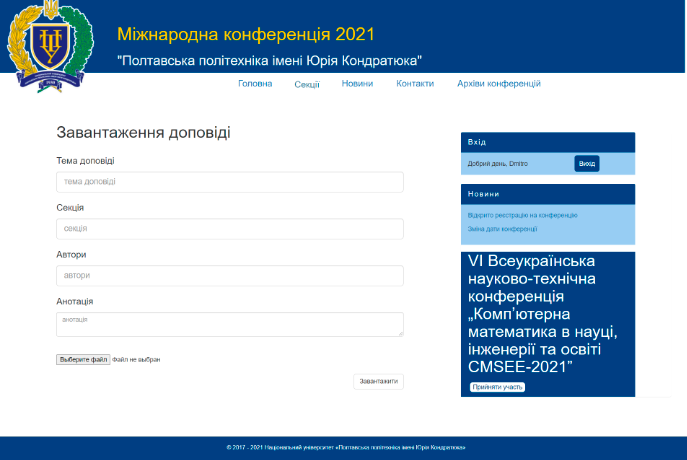


Рисунок 3.19 – Завантаження доповіді

На завантажені доповіді діють обмеження по розміру файлу та його типу. Програмно дозволено завантажувати файли типу «.doc»,«.docx»та«.pdf», обмеження по розміру сягає 10 мегабайтів.

Після того, як файл завантажено на сайт, інші користувачі можуть обрати необхідну секцію зі списку та переглянути усі завантажені доповіді у цій секції(рис. 3.18).

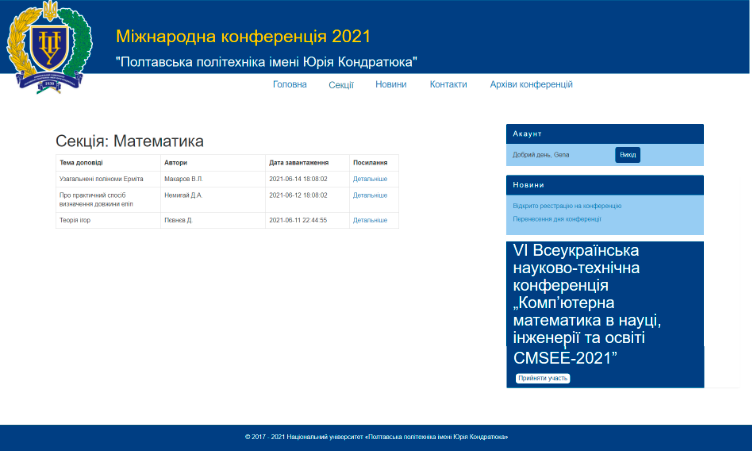


Рисунок 3.20 – Список доповідей за секцією

Для того, щоб переглянути більш детальну інформацію про конкретну доповідь та коментарі до неї, користувач має натиснути на посилання «Детальніше» (рис. 3.19).

На сторінці наводиться повна інформація по доповіді та посилання на її завантаження. Також присутні коментарі та їх автори. Згідно з діаграмою використання у випадку, якщо користувач зареєстрований в системі, він має можливість залишити свій коментар.

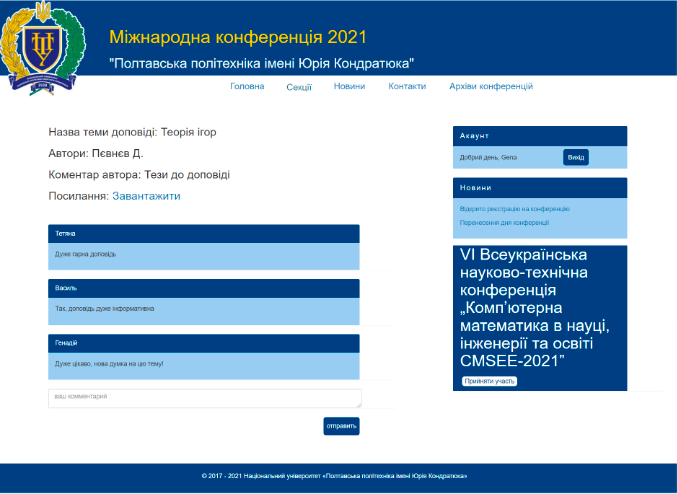


Рисунок 3.21 –Сторінка доповіді

Коментарі можуть залишати лише зареєстровані в системі користувачі. В іншому випадку, при спробі залишити коментар, незареєстрованому користувачу буде запропоновано пройти реєстрацію.

# РОЗДІЛ 4 ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯПРОВЕДЕННЯ ІНТЕРНЕТ-КОНФЕРЕНЦІЙ

1. **Тестування верстки сайту**

Під час тестування верстки сайту було перевірено:

* коректність полів вводу;
* адаптивність верстки;
* роботу усіх посилань та кнопок.

Усі дефекти були занесені до таблиці 4.1, в якій вказано відповідний дефекту багрепорт.

Таблиця 4.1 – Опис дефектів

|  |  |  |
| --- | --- | --- |
| № | Короткий опис | Багрепорт |
| 1 | При масштабі 125% праве меню перекриває основний текст | 001 |
| 2 | При реєстрації поле «пароль» не приховує текст, що вводиться | 002 |

Таблиця 4.2 – Багрепорт №001

|  |  |
| --- | --- |
| Багрепорт №001 | |
| Короткий Опис (Summary) | При масштабі 125% праве меню перекриває основний текст. |
| Проєкт (Project) | Інформаційна система для проведення онлайн-конференцій |
| Компонент додатку (Component) | Головна сторінка |
| Серйозність (Severity) | S4 Незначний (Minor) |
| Пріоритет (Priority) | P3 Низький (Low) |
| Статус (Status) | Відкритий |
| Автор (Author) | Пєвнєв Д.П. |

Продовження таблиці 4.2

|  |  |
| --- | --- |
| Оточення | |
| ОС / Сервіс Пак і т.і. / Браузер | Windows 10 / Google Chrome v91.0.4472.77 (64-bit) |
| Опис | |
| Кроки відтворення (StepstoReproduce) | 1. Відкрити головну сторінку сайту;  2. Збільшити масштаб до 125%. |
| Фактичний результат (Result) | Праве меню перекриває основний текст. |
| Очікуваний результат (ExpectedResult) | Коректне відображення блоків верстки |

Таблиця 4.3 – Баг-репорт №002

|  |  |
| --- | --- |
| Багрепорт №002 | |
| Короткий Опис (Summary) | При реєстрації поле «пароль» не приховує текст, що вводиться |
| Проєкт (Project) | Інформаційна система для проведення онлайн-конференцій |
| Компонент додатку (Component) | Сторінка реєстрації |
| Серйозність (Severity) | S4 Незначний (Minor) |
| Пріоритет (Priority) | P3 Низький (Low) |
| Статус (Status) | Відкритий |
| Автор (Author) | Пєвнєв Д.П. |
| Оточення | |
| ОС / Сервіс Пак іт.і. / Браузер | Windows 10 / Google Chrome v91.0.4472.77 (64-bit) |

Продовження таблиці 4.3

|  |  |
| --- | --- |
| Опис | |
| Кроки відтворення (StepstoReproduce) | 1. Відкрити головну сторінку сайту(користувач повинен бути не авторизованим);  2. В меню авторизації натиснути на кнопку «Реєстрація»;  3. На сторінці реєстрації обрати поле «пароль»;  4. Почати вводити дані. |
| Опис | |
| Фактичний результат (Result) | Поле «пароль» не приховує текст. |
| Очікуваний результат (ExpectedResult) | Приховування тексту в полі «пароль». |

Усі дефекти були виправлені.

1. **Тест на крос-браузерність**

Для тестування на крос-браузерність, були використані наступні браузери та їх версії:

* Google Chrome, версія 91.0.4472.77;
* Opera версія 77.0.4054.64;
* Microsoft Edge версія 91.0.864.37;

Нижче представлено контрольний список (чек-лист) перевірок для реалізації функціонального тестування.

Таблиця 4.4 – Результат крос-браузерного тестування

|  |  |  |  |
| --- | --- | --- | --- |
| Опис | GChrome | Opera | Edge |
| Коректне зображення відповідно до запланованого дизайну | Passed | Passed | Passed |
| Коректне зображення тексту | Passed | Passed | Passed |
| Коректне зображення кнопок, блоків меню, тощо | Passed | Passed | Passed |

Продовження таблиці 4.4

|  |  |  |  |
| --- | --- | --- | --- |
| Опис | GChrome | Opera | Edge |
| Коректна робота посилань | Passed | Passed | Passed |
| Перевірка масштабування | Passed | Passed | Passed |
| Реєстрація | Passed | Passed | Passed |
| Авторизація | Passed | Passed | Passed |
| Завантаження файлу | Passed | Passed | Passed |
| Заявка на конференцію | Passed | Passed | Passed |
| Робота сторінки адміністратора | Passed | Passed | Passed |

Із чек-листу видно, що сайт коректно працює різних браузерах.

**ВИСНОВКИ**

У ході виконання кваліфікаційної роботи бакалавра, було спроєктовано та розроблено інформаційну систему онлайн-конференції. Дана система надає можливість збирати та систематизувати доповіді під час конференцій, отримати інформацію про поточну конференцію, переглянути коментарі до доповіді інших учасників конференції та залишити свій. Сайт має систему авторизації та розподіл ролей, таким чином, тільки перевірені особи зможуть долучитися до конференції та завантажувати свої доповіді. Інформаційна система дозволяє обмінюватися знаннями між учасниками через інтернет.

Під час розробки сайту було використано та обґрунтовано новітні технології.

Система призначена для користувачів, які мають намір взяти участь у конференції як у ролі учасника, так і в ролі слухача.

Переваги використання інформаційної системи:

* чітко відсортовані за секціями доповіді;
* можливість завантаження доповідей;
* можливість коментування доповідей;
* повна інформація про поточну та попередні конференції;
* реалізований блок новин.

Розроблена інформаційна система задовольняє вимоги, які були поставлені на етапі визначення завдання. При подальшому вдосконаленні системи є можливим сортування доповідей за авторами, реалізація персональних повідомлень між користувачами, а також реалізація відеоконференцій на базі сайту.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Правила створення наукових онлайн-конференцій [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://www.nature.com/articles/s41578-020-0194-0.
2. Онлайн-конференції під час пандемії коронавірусу[Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://www.nature.com/ articles/d41586-020-01489-0.
3. Ресурс для відстеження трафіку сайтів[Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://www.similarweb.com
4. Приклад програмного рішення подачі заявки Ресурс для відстеження трафіку сайтів [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://itta.org.ua/onlajn-konferenciya-15-03-2021.
5. Приклад програмного рішення реалізації продукту [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: http://www.konferenciaonline. org.ua.
6. Вплив коронавірусу на відео-конференції [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://blog.dataart.com/the-webconfer encing-boom-covid-19-s-effect-on-the-video-call-market.
7. Модель життєвого циклу водоспаду головні поняття [Електронний ресурс]: [Веб-сайт]. – Електронні дані. Режим доступу: https://www.tutorialspoin t.com/sdlc/sdlc\_waterfall\_model.htm.
8. Фази водоспадної моделі [Електронний ресурс]: [Веб-сайт]. – Електронні дані. Режим доступу: https://www.toolsqa.com/softwaretesting/waterfall-model.
9. Спіральна модель життєвого циклу [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://www.tutorialspoint.com/sdlc/sdlc\_spiral \_model.htm.
10. Етапи спіральної моделі життєвого циклу [Електронний ресурс]: [Веб-сайт]. – Електронні дані. Режим доступу: https://www.geeksforgeeks.org/software engineering-spiral-model.
11. Agileметодологія [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://www.javatpoint.com/agile-sdlc.
12. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 4-е изд. — СПб.: Питер, 2016. — 768 с.: ил. — (Серия «Бестселлеры O’Reilly»).
13. Скляр, Д. Изучаем РНР 7: руководство по созданию интерактивных веб-сайтов. : Пер. с англ. — СпБ.: ООО “Альфа-книга”, 2017. — 464 с. : ил. — Парал. тит. англ.
14. Вступ до php [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу:https://www.tutorialspoint.com/php/php\_introduction.htm.
15. Інформація по СУБД MySQL[Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html.
16. MVC [Електронний ресурс]: [Веб-сайт]. – Електронні дані. Режим доступу:https://web-creator.ru/articles/mvc.

**ДОДАТОК А   
ВИХІДНИЙ КОД ОСНОВНИХ КОМПОНЕНТІВ**

1. Контроллер роботи із файлами (Files.php)

if ( ! defined('BASEPATH')) exit('Nodirectscriptaccessallowed');

classfilesextendsCI\_Controller {

function \_\_construct() {

parent::\_\_construct();

$this->load->model('file');

$this->load->model('news\_model');

}

publicfunctionindex($section=null){

$this->data['files'] = $this->file->getRows($section);

$this->data['news'] = $this->news\_model->getNews();

//loadtheview

$this->data['title'] = "доповіді";

$this->load->view('templates/header', $this->data);

$this->load->view('files/index');

$this->load->view('templates/footer')

}

publicfunctionview($slug=NULL){

$this->data['files\_item'] = $this->file->getFiles($slug);

$files\_slug = $this->file->getFiles($slug);

$files = $this->file->getFiles();

$data['files'] = $this->file->getRows($slug);

$this->data['news'] = $this->news\_model->getNews();

if(empty($this->data['files\_item'])){

show\_404();

}

$this->data['id'] = $files\_slug['id'];

$this->load->model('comments\_model');

$this->data['comments'] = $this->comments\_model->getComments($files\_slug['id'],100);

$files['files'] = $this->file->getFiles($slug);

$this->data['content'] = $this->data['files\_item']['file\_name'];

$this->load->view('templates/header', $this->data);

$this->load->view('files/view', $files);

$this->load->view('templates/footer');

}

publicfunctiondownload($slug){

if(!empty($slug)){

//loaddownloadhelper

$this->load->helper('download');

//getfileinfofromdatabase

$fileinfo = $this->file->download($slug);

//filepath

$file = 'uploads/'.$fileinfo['file\_name'];

//downloadfilefromdirectory

force\_download($file, NULL);

}

}

publicfunctioncomment(){

if($this->input->post('user\_id') && $this->input->post('files\_id') && $this->input->post('text')) {

$this->load->model('comments\_model');

$user\_id = $this->input->post('user\_id');

$files\_id = $this->input->post('files\_id');

$text = $this->input->post('text');

if($this->comments\_model->setComments($user\_id, $files\_id, $text)) {

$this->data['title'] = 'Комментарий добавлен!';

}

}

else{

show\_404();

}

$this->data['title'] = 'Комментарий добавлен!';

$this->load->view('news/commentCreated');

}

1. Контроллер для завантаження файлів (upload.php)

class Upload extends CI\_Controller {

public function \_\_construct()

{

parent::\_\_construct();

$this->load->helper(array('form', 'url'));

$this->load->database();

$this->load->library('form\_validation');

$this->load->helper('form');

$this->load->model('news\_model');

}

public function index()

{

$this->data['news'] = $this->news\_model->getNews();

$this->data['title'] = "додатидоповідь";

$this->load->view('templates/header', $this->data);

$this->load->view('files/upload\_file', $this->data);

$this->load->view('templates/footer');

}

public function do\_upload()

{

$config['upload\_path'] = './uploads/';

$config['allowed\_types'] = 'doc|docx|pdf';

$config['max\_size'] = 10000;

$this->load->library('upload', $config);

if ( ! $this->upload->do\_upload('userfile'))

{

$error = array('error' => $this->upload->display\_errors());

$this->load->view('files/upload\_file', $error);

}

else

{

$uniqid = uniqid();

$data = array(

'file\_name' => $this->upload->data('file\_name'),

'file\_path' => $this->upload->data('full\_path'),

'created' =>date("Y-m-d H:i:s"),

'slug' => $uniqid,

'author' => $this->input->post('author'),

'section' => $this->input->post('section'),

'file\_theme' => $this->input->post('title'),

'anotation' => $this->input->post('anotation'),

'user\_id' => $this->input->post('user\_id'));

$this->load->view('files/upload\_success');

return $this->db->insert('files', $data);

}

$this->load->view('templates/header', $this->data);

$this->load->view('files/upload\_success', $data);

$this->load->view('templates/footer');

}

}

1. Контроллер роботи із користувачами (Profile.php)

defined('BASEPATH') OR exit('nodirectscriptaccessallowed');

classProfileextendsMY\_Controller{

publicfunction \_\_construct(){

parent::\_\_construct();

$this->load->model('profile\_model');

$this->load->helper('other\_helper');

$this->load->helper('url');

$this->load->library('DX\_Auth');

}

publicfunctionindex()

{

$this->data['title'] = "профіль";

$this->load->view('templates/header', $this->data);

$this->load->view('users/profile\_create', $this->data);

$this->load->view('templates/footer')

}

publicfunctionregister(){

$this->data['title'] = "Реєстрація";

$this->load->view('templates/header', $this->data);

$this->load->view('users/registration');

$this->load->view('templates/footer');

}

publicfunctionpage(){

$id = $this->dx\_auth->get\_user\_id();

$this->data['h']=$this->profile\_model->getUserProfile($id);

$this->load->view('templates/header', $this->data);

$this->load->view('users/user\_page', $this->data);

$this->load->view('templates/footer');

}

publicfunctioncreate(){

$this->data['title'] = "записатися на конференцію";

if( $this->input->post('name') &&

$this->input->post('surname') &&

$this->input->post('study\_degree')&&

$this->input->post('work\_place')&&

$this->input->post('work\_grade')&&

$this->input->post('phone') &&

$this->input->post('adress')&&

$this->input->post('user\_id')

){

$name = $this->input->post('name');

$surname = $this->input->post('surname');

$study\_degree = $this->input->post('study\_degree');

$work\_place = $this->input->post('work\_place');

$work\_grade = $this->input->post('work\_grade');

$phone = $this->input->post('phone');

$adress = $this->input->post('adress');

$user\_id = $this->input->post('user\_id');

if($this->profile\_model->setProfile($name, $surname, $study\_degree, $work\_place, $work\_grade, $phone, $adress, $user\_id)){

$this->load->view('templates/header', $this->data);

$this->load->view('users/success', $this->data);

$this->load->view('templates/footer');

}

} else {

show\_404();

}

}

}

1. Контроллер роботи із новинами(News.php)

defined('BASEPATH') OR exit('nodirectscriptaccessallowed');

classNewsextendsMY\_Controller{

publicfunction \_\_construct(){

parent::\_\_construct();

$this->load->model('news\_model');

$this->load->helper('other\_helper');

$this->load->helper('url');

}

publicfunctionindex(){

$this->data['title'] = "Усіновини";

$this->data['news'] = $this->news\_model->getNews();

$this->load->view('templates/header', $this->data);

$this->load->view('news/index', $this->data);

$this->load->view('templates/footer');

}

publicfunctionview($slug=NULL){

$this->data['news\_item'] = $this->news\_model->getNews($slug);

$news\_slug = $this->news\_model->getNews($slug);

if(empty($this->data['news\_item'])){

show\_404();

}

$this->data['id'] = $news\_slug['id'];

$this->load->model('comments\_model');

$this->data['comments'] = $this->comments\_model->getComments($news\_slug['id'],100);

$this->data['title'] = $this->data['news\_item']['title'];

$this->data['content'] = $this->data['news\_item']['text'];

$this->load->view('templates/header', $this->data);

$this->load->view('news/view', $this->data);

$this->load->view('templates/footer');

}

publicfunctioncreate(){

$this->data['title'] = "додати новину";

if($this->input->post('slug') && $this->input->post('title') && $this->input->post('text')){

$slug = $this->input->post('slug');

$title = $this->input->post('title');

$text = $this->input->post('text');

if($this->news\_model->setNews($slug, $title, $text)){

$this->load->view('templates/header', $this->data);

$this->load->view('news/success', $this->data);

$this->load->view('templates/footer');

}

} else {

$this->load->view('templates/header', $this->data);

$this->load->view('news/create', $this->data);

$this->load->view('templates/footer');

}

}

publicfunctiondelete($slug=NULL){

$this->data['news\_delete'] = $this->news\_model->getNews($slug);

if(empty($this->data['news\_delete'])){

show\_404();

}

$this->data['title'] = "видаленняновинии";

$this->data['result'] = "помилкавидалення".$this->data['news\_delete']['title'];

if ($this->news\_model->deleteNews($slug)){

$this->data['result'] = $this->data['news\_delete']['title']."видалена";

}

$this->load->view('templates/header', $this->data);

$this->load->view('news/delete', $this->data);

$this->load->view('templates/footer');

}

publicfunctioncomment() {

if($this->input->post('user\_id') && $this->input->post('news\_id') && $this->input->post('comment\_text')) {

$this->load->model('comments\_model');

$user\_id = $this->input->post('user\_id');

$movie\_id = $this->input->post('news\_id');

$comment\_text = $this->input->post('comment\_text');

if($this->comments\_model->setComments($user\_id, $movie\_id, $comment\_text)) {

$this->data['title'] = 'Комментарий добавлен!';

}

}

else{

show\_404();

}

$this->data['title'] = 'Комментарий добавлен!';

$this->load->view('news/commentCreated');

}

}

1. Модель новостей(News\_model.php)

if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Profile\_model extends CI\_Model{

public function \_\_construct(){

$this->load->database();

}

public function setProfile($name, $surname, $study\_degree, $work\_place, $work\_grade, $phone, $adress, $user\_id){

$data = array(

'name' => $name,

'surname' => $surname,

'study\_degree' => $study\_degree,

'work\_place' => $work\_place,

'work\_grade' => $work\_grade,

'phone' => $phone,

'adress' => $adress,

'user\_id' => $user\_id

);

return $this->db->insert('user\_info', $data);

}

public function getProfiles(){

$query = $this->db->get('user\_info');

return $query;

}

public function getUserProfile($id=FALSE){

$query = $this->db->get\_where('user\_info', array('user\_id'=> $id));

return $query->row\_array();

}

}

1. Модель новостей(File.php)

if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class File extends CI\_Model{

public function \_\_construct(){

$this->load->database();

}

function getRows($section=null,$params = array()){

$this->db->select('\*');

$this->db->from('files');

$this->db->where('section', $section);

$this->db->order\_by('created','desc');

if(array\_key\_exists('id',$params) && !empty($params['id'])){

$this->db->where('id',$params['id']);

//get records

$query = $this->db->get();

$result = ($query->num\_rows() > 0)?$query->row\_array():FALSE;

}else{

//set start and limit

if(array\_key\_exists("start",$params) &&array\_key\_exists("limit",$params)){

$this->db->limit($params['limit'],$params['start']);

}elseif(!array\_key\_exists("start",$params) &&array\_key\_exists("limit",$params)){

$this->db->limit($params['limit']);

}

//get records

$query = $this->db->get();

$result = ($query->num\_rows() > 0)?$query->result\_array():FALSE;

}

//return fetched data

return $result;

}

public function getFiles($slug=FALSE){

if($slug===FALSE){

$query = $this->db->get('files');

return $query->result\_array();

}

$query = $this->db->get\_where('files', array('slug'=> $slug));

return $query->row\_array();

}

public function download($slug){

$query = $this->db->get\_where('files',array('slug'=>$slug));

return $query->row\_array();

}

}

1. Модель профілю (Profile\_model.php)

if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Profile\_model extends CI\_Model{

public function \_\_construct(){

$this->load->database();

}

public function setProfile($name, $surname, $study\_degree, $work\_place, $work\_grade, $phone, $adress, $user\_id){

$data = array(

'name' => $name,

'surname' => $surname,

'study\_degree' => $study\_degree,

'work\_place' => $work\_place,

'work\_grade' => $work\_grade,

'phone' => $phone,

'adress' => $adress,

'user\_id' => $user\_id

);

return $this->db->insert('user\_info', $data);

}

public function getProfiles(){

$query = $this->db->get('user\_info');

return $query;

}

public function getUserProfile($id=FALSE){

$query = $this->db->get\_where('user\_info', array('user\_id'=> $id));

return $query->row\_array();

}

}

1. Вид створення профілю (profile\_create.php)

<div class="wrapper">

<div class="container-fluid">

<div class="row">

<div class="col-lg-8 сol-md-8">

<div class = "container main-info">

<h1>Заява на конференцію</h1></br>

<form action="/profile/create/" method = "post">

<p class="main-text">Ім'я</p>

<input class="form-control input-lg" type="input" name="name" placeholder="Ім'я"></br>

<p class="main-text">Прізвище</p>

<input class="form-control input-lg" type="input" name="surname" placeholder="Прізвище"></br>

<p class="main-text">По-батькові</p>

<input class="form-control input-lg" type="input" name="lastname" placeholder="По-батькові"></br>

<p class="main-text">Навчальнийступінь</p>

<input class="form-control input-lg" type="input" name="study\_degree" placeholder="Навчальнийступінь"></br>

<p class="main-text">Місцероботи</p>

<input class="form-control input-lg" type="input" name="work\_place" placeholder="Місцероботи"></br>

<p class="main-text">Посада</p>

<input class="form-control input-lg" type="input" name="work\_grade" placeholder="Посада"></br>

<p class="main-text">Телефон</p>

<input class="form-control input-lg" type="input" name="phone" placeholder="Телефон"></br>

<p class="main-text">Адреса</p>

<input class="form-control input-lg" type="input" name="adress" placeholder="Адреса"></br>

<input class="hidden" type="input" name="user\_id" value="<?php echo $this->dx\_auth->get\_user\_id(); ?>">

<input class="btnbtn-default pull-right" type="submit" name="submit" value="додатиінформацію">

</form>

</div>

</div>

<?php $this->load->view('templates/menu'); ?>

</div>

</div>

</div>

</div>

1. Вид реєстрації (registration.php)

<div class="col-lg-8 сol-md-8">

<div class = "container main-info">

<h1>Реєстрація</h1></br>

<form action="/auth/register/" method = "post">

<p class="main-text">Логін</p>

<input class="form-control input-lg" type="input" name="username" placeholder="Логін"></br>

<p class="main-text">Пароль</p>

<input class="form-control input-lg" type="password" name="password" placeholder="Пароль"></br>

<p class="main-text">Підтвердженняпаролю</p>

<input class="form-control input-lg" type="password" name="confirm\_password" placeholder="Пароль"></br>

<p class="main-text">E-mail</p>

<input class="form-control input-lg" type="input" name="email" placeholder="E-mail"></br>

<input class="btnbtn-default pull-right" type="submit" name="submit" value="Реєстрація">

</form>

</div>

</div>

<div class="wrapper">

<div class="col-lg-4 сol-md-4">

<div class="panel panel-info">

<div class="panel-heading">

<div class="sidebar-header">

Вхід

</div>

</div>

<div class="panel-body">

Будь ласка, пройдітьреєстрацію

</form>

</div>

<div class="panel panel-info">

<div class="panel-heading">

<div class="sidebar-header">

Новини

</div>

</div>

<div class="panel-body">

<?php foreach ($news as $key => $value): ?>

<p><a href="/news/view/<?php echo $value ['slug']; ?>"><?php echo $value['title']; ?></a></p>

<?php endforeach ?>

</div>

<div class="left-main-block">

<h1>VI Всеукраїнськанауково-технічнаконференція „Комп’ютерна математика в науці, інженерії та освіті CMSEE-2021”</h1>

<form action="#" target="\_blank" class="conf-reg"></form>

<button>Прийняти участь</button>

</div>

</div>

</div>

</div>

**ДОДАТОК B   
УЧАСТЬ У КОНФЕРЕНЦІЇ**

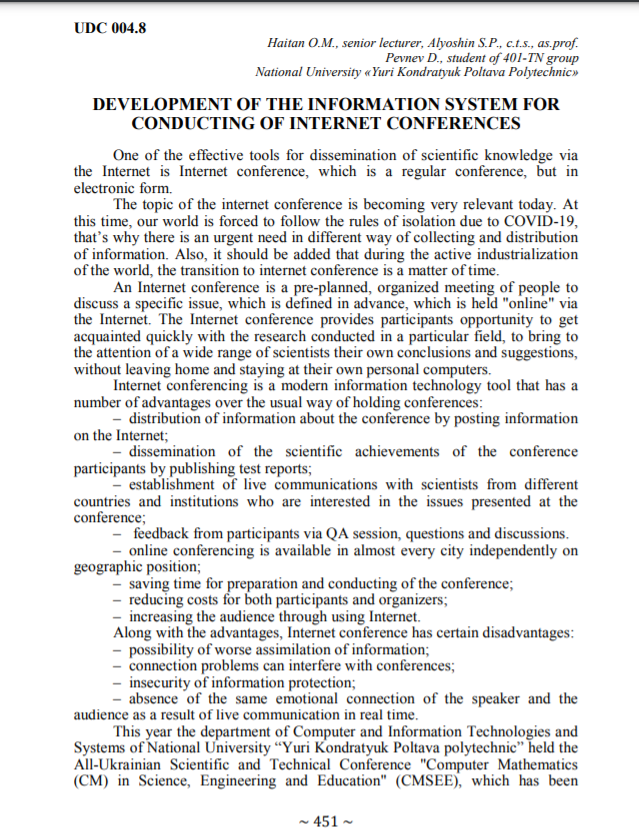


Рисунок B.1 – Перша частина тез

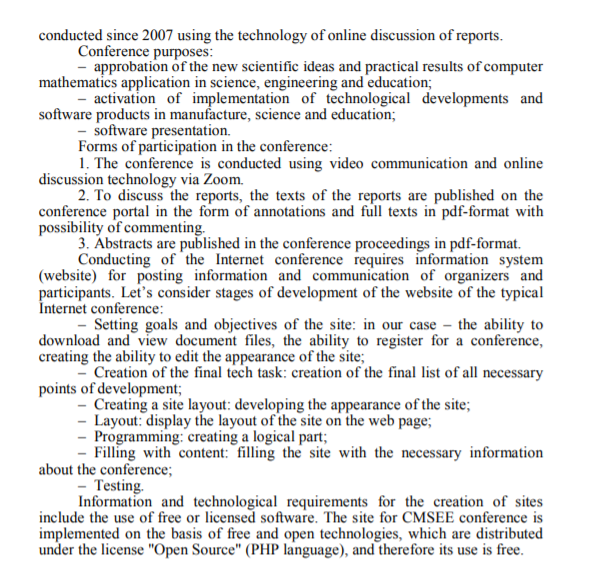


Рисунок B.2 – Друга частина тез