ДИПЛОМНА РОБОТА
БАКАЛАВРА

ДРБ. ІП - 19.00.00.000 ПЗ

Група ІП-20-2

Савич Ольга

2024

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

УДК 004.4

ДИПЛОМНА РОБОТА БАКАЛАВРА

Тема: “ Розробка веб-орієнтованого сервісу з реалізацією багаторівневого ступеню доступу та системою моніторингу даних ”

(назва згідно з наказом ректора)

спеціальність – 121 “Інженерія програмного забезпечення”

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДРБ.ІП - 19.00.00.000 ПЗ

(позначення)

Консультанти:

з питань нормоконтролю

асист. /М.М. Піх/
(посада) (підпис) (дата)(розшифровка підпису)

Студент

ІП-20-2 /О.О.Савич/
(шифр групи) (підпис) (дата) (розшифровка підпису)

Керівник роботи

к.т.н., доц. /Р.Б.Вовк/
(посада) (підпис) (дата) (розшифровка підпису)

Допускається до захисту
Завідувач кафедри

к.т.н., доц. /В.В.Бандура/
(посада) (підпис) (дата) (розшифровка підпису)

2024
(рік)

Івано-Франківський національний технічний університет нафти і газу

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітньо-кваліфікаційний рівень бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**
доц. В.В. Бандура
“ ____ ” _____2024 р.

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ БАКАЛАВРА СТУДЕНТОВІ

Савич Ользі Олегівні

(прізвище, ім’я, по-батькові)

1. Тема проекту (роботи) **“ Розробка веб-орієнтованого сервісу з реалізацією багаторівневого ступеню доступу та системою моніторингу даних”**

керівник проекту (роботи) Вовк Роман Богданович, к.т.н., доцент

затверджені наказом закладу вищої освіти від **“** ____ ” травня 2024 р. № _____

2. Строк подання студентом проекту (роботи) 17 червня 2024 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

4. Зміст розрахунково - пояснювальної записки (перелік питань, які потрібно розробити)

1. Огляд існуючих систем багаторівневого ступеню доступу на прикладі веб-форуму
2. Розробка алгоритмічного забезпечення та опис програмного середовища
3. Розробка концептуальної моделі предметної області
4. Створення сервісу багаторівневого ступеню доступу на прикладі веб-форуму
5. Реалізація основних функцій веб-сервісу

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Головна сторінка українського форуму програмістів (рис. 1.1)
2. Класифікація веб-форумів за різними ознаками (рис. 1.2)
3. Представлення роботи шаблону MVC (рис. 1.3)
4. Діаграма класів простору імен Rsdn.Observer (рис. 1.4)
5. Узагальнена структура даних “ядра” веб-спільноти (рис. 2.1)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання **22 травня 2024 р.**

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Огляд існуючих систем багаторівневого ступеню доступу на прикладі веб-форуму	25.05.2024	виконано
2	Розробка алгоритмічного забезпечення та опис програмного середовища	29.05.2024	виконано
3	Розробка концептуальної моделі предметної області	30.05.2024	виконано
4	Створення сервісу багаторівневого ступеню доступу на прикладі веб-форуму	04.06.2024	виконано
5	Реалізація основних функцій веб-сервісу	10.06.2024	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	17.06.2024	виконано

Студент – дипломник _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Ця робота присвячена розробці веб-орієнтованого сервісу з реалізацією багаторівневого ступеню доступу на прикладі веб-форуму. У сучасному інтернет-середовищі веб-форуми є популярним засобом спілкування та обміну інформацією. Метою цієї роботи є створення веб-форуму з використанням високорівневих та низькорівневих заходів безпеки для забезпечення ефективного та безпечного обміну інформацією між користувачами. В результаті здійснена розробка веб-орієнтованого сервісу з реалізацією багаторівневого ступеню доступу для дискусій, обміну інформацією та комунікації.

КЛЮЧОВІ СЛОВА: РІВЕНЬ ДОСТУПУ, КОРИСТУВАЧ, ВЕБ-ФОРУМ, ВІРТКАЛЬНА СПІЛЬНОТА, UML-МОДЕЛЮВАННЯ, СХОВИЩЕ ДАНИХ

ANNOTATION

This work is devoted to the development of a web-oriented service with the implementation of a multi-level degree of access on the example of a web forum. In today's Internet environment, web forums are a popular means of communication and information exchange. The method of this work is to create a web forum using high-level and low-level security measures to ensure efficient and secure exchange of information between users. As a result, the development of a web-oriented service with the implementation of a multi-level degree of access for discussions, information exchange and communication was carried out.

KEY WORDS: ACCESS LEVEL, USER, WEB FORUM, DRIVING COMMUNITY, UML MODELING, DATA WAREHOUSE

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 8
ВСТУП 9

1. ОГЛЯД ІСНУЮЧИХ СИСТЕМ БАГАТОРІВНЕВОГО СТУПЕНЮ ДОСТУПУ НА ПРИКЛАДІ ВЕБ-ФОРУМУ 10
1.1. Проблеми діяльності та порівняння веб-форумів з іншими видами інтернет спілкування 10
1.2. Аналіз вимог до рушія форуму 15
1.3. Системні вимоги до розроблюваного сервісу 17
1.4. Дослідження та опис видів та технологій розробки сервісів на основі веб-форуму 18
1.5. Висновки до розділу 23

2 РОЗРОБКА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ ТА ОПИС ПРОГРАМНОГО СЕРЕДОВИЩА ДЛЯ РОЗРОБКИ 24
2.1. Представлення алгоритмічного забезпечення системи 24
2.2. Програмне середовище для розробки проєкту 28
2.3. Програмні додатки для реалізації серверної основи 36
2.4. Побудова UML-діаграм веб-орієнтованого сервісу 48
2.5. Розробка концептуальної моделі предметної області 52
2.6. Алгоритм перевірки даних користувача під час реєстрації 54
2.7. Висновки по розділу 57

3. СТВОРЕННЯ СЕРВІСУ БАГАТОРІВНЕВОГО СТУПЕНЮ ДОСТУПУ НА ПРИКЛАДІ ВЕБ-ФОРУМУ ТА ЙОГО ІНТЕГРАЦІЯ З СЕРВЕРОМ 58
3.1. Створення функціональної частини 58

3.2. Встановлення веб-ресурсу на хостинг 63
3.3. Реалізація основних функцій веб-сервісу 65
3.4. Висновки до розділу 77

ВИСНОВКИ 78
СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА 79
ДОДАТКИ
БІБЛІОГРАФІЧНА ДОВІДКА

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

MVC - Model View Control

SRS - Software Requirement Specification

Сайт – особлива програма, яку читає браузер за допомогою мов тегів і стилів

Топік – розділ, спеціальна тема форуму

Дебагінг - процес вирішення помилок програми, при якому кожна сторінка сайту розглядається окремо.

ID – Identification – певне значення, яке однозначно ідентифікує певний запис в таблиці.

FilleZilla – безплатне програмне забечпечення для обміну файлів між локальним носієм та сервером.

InfinityFree – безплатний веб-хостінг, який дозволяє розміжати власні сайти з необмеженим розміром.

Адміністратор – людина яка керує форумом.

Платформа –онлайн ресурс, певне джерело інформації чи комунікації.

Комюніті – постійні користувачі, які складають левову частку форуму.

ВСТУП

Веб-форум – один із найвідоміших типів спілкування в Інтернеті, спеціальний сайт, який ставить перед собою одну мету: допомогти багатьом користувачам проводити дискусії, поділитися досвідом та ідеями в певній темі. Форум може бути цілком посвячений певній темі, наприклад, футбольній команді, відомій книзі чи франшизі, де збираються фанати і зацікавлені люди. Також не рідкість побачити професійні форуми, які збирають людей з різних частин світу, з різним досвідом для обговорення і передачі досвіду із певних вузькоспеціалізованих тем. Найпоширеніше ділення веб-форуму: розділи→теми→повідомлення. Звично повідомлення несуть інформацію «автор-тема-зміст-дата/час».

В основу форуму вкладена ідея про анонімність і права всіх користувачів на форумі, розподіл ролей серед модераторів, які повинні слідкувати за дотриманням правил форуму чи спеціальному топіку серед користувачів. Звісно ж, у кожному форумі є людина, яка відповідає за організацію всіх процесів, але дана робота є складною для однієї особи. Велика кількість інформації, яку необхідно переглянути, багато різних тем і людей, які постійно пишуть в різні теми і обмінюються повідомленнями.

А враховуючи велику кількість тем, користувачів, та повідомлень це завдання стає складнішим з розширенням комюніті форуму. Це завдання є складним і відповідальним для однієї людини, чи малій групі адміністраторів, саме тому існують модератори, користувачі які відповідають за певну частину тем і постійно переглядають всі повідомлення в темі на присутність тут повідомлень які не відповідають правилам форуму. Форум має строгу ієрархію класів і кожен клас має всій рівень допуску до функцій форуму, наприклад, користувач, який має один із найнищих рівнів допуску може створювати дописи і редагувати всій профіль, а адміністратор, який має найвищий рівень володіє всіма можливостями, які надає система.

1. ОГЛЯД ІСНУЮЧИХ СИСТЕМ БАГАТОРІВНЕВОГО СТУПЕНЮ ДОСТУПУ НА ПРИКЛАДІ ВЕБ-ФОРУМУ

1.1. Проблеми діяльності та порівняння веб-форумів з іншими видами інтернет спілкування

При розгляді даної приблеми потрібно звернутись до самої суті поняття форуму. В Давному Римі форумом називалась міська ринкова площа, де багато людей одночасно спілкувалтсь на різні теми, таким чином форум був центром суспільного життя Римської держави часу Республіки. Переглядаючи сучасні веб-форуми можемо побачити що в наш час вони пережили еволюцію, оскільки тепер форуми збирають людей об'єднаних однією темою, будь це професією чи певним захопленням. Тому не мало важливою проблемою під час планування створення форуму є його основна тематика, яка може бути найрізноманітнішою і охоплювати всі сфери життя, при чому може визначатися адміністратором при створенні форуму, чи залежати від контингенту учасників.

Форум - це класичний варіант соціального сайту. Перші інтернет-ресурси цього типу були вкрай простими. Але з появою web 2.0 і інших технологій, вони обзавелися принципово новими можливостями.

Багато форумів створюють на існуючих веб-порталах, без прив'язки до якогось певного сайту. Для прикладу можна взяти сайт Microsoft Visual Studio, на головній сторнці в меню навігації є розділ «Форум», де можна знайти різні теми з певними технічними питаннями, де звичайні користувачі даного програмного забезпечення шукають чи створюють теми і топіки з описом питань до спеціалістів, та інших учасників форуму. Таким чином форум при сайті MVS створений для допомоги користувачам їхніх програм у вирішенні певних питань пов'язаними з продуктами компанії, отже модератори чи адміністратор повинні видаляти всі ті повідомлення, які не відповідають темі

форуму, оскільки такі повідомлення просто засмічують даний форум, який має спеціальне призначення.

Розглядаючи загальні проблеми створення системи веб-форуму, потрібно, перш за все, визначити цілі і задачі цієї системи, сформулювати основні проблеми, що потребують розв'язання, виявити основних учасників цього процесу, визначити їх ролі і методи взаємодії. Це - перший етап аналізу даної системи. На другому етапі потрібно знайти шляхи розв'язання ключових проблем та запропонувати відповідні удосконалення системи. Звичайно, постановка проблеми у такому широкому контексті не є предметом даної частини. Насамперед важливо уважно проаналізувати засоби досягнення вказаних цілей і задач, які полягають, насамперед, в організації взаємодії учасників процесу комунікації. Саме організація даного процесу, який розглядається як процес передачі та перегляду інформації, є найбільш валідною загальною проблемою побудови ефективної системи веб-форуму. Саме тому, акцент у діяльності системи буде переміщено саме на побудову зручної комунікативної мережі, організацію моніторингу всіх тем і повідомлень в топіках, безперервне вдосконалення системи відображення та методик їх використання.

Для успішної побудови системи веб-форуму потрібно проаналізувати існуючі форуми і визначити основні пункти необхідні для реалізації даної системи. Для прикладу розглянемо сайт під назвою "Український форум програмістів" (рис. 1.1).

Ми бачимо, що даний сайт умовно розділений на 2 частини: верхня частина, та головний блок форуму. В верхній чпстині знаходяться кнопки навігації, допомоги, а також, блок із реєстрацією нових користувачів, входом на форум і панель налаштувань користувача. Дана частина відводиться для налаштувань профілю користувача, а також для швидкого пошуку інформації про форум чи його правила, вона не займає багато місця на сайті і служить тільки як інструмент для учасників форуму.

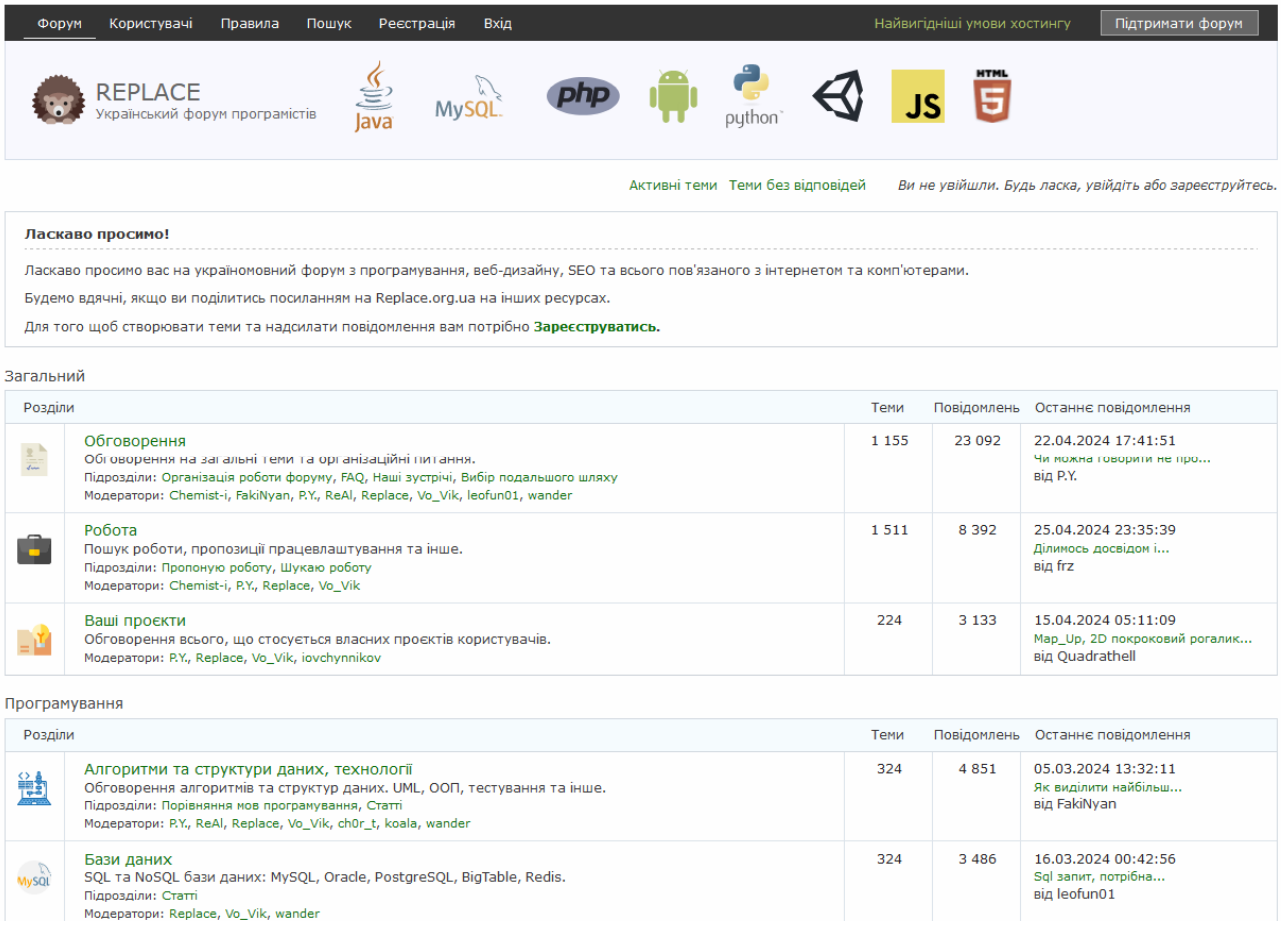


Рисунок 1.1 - Головна сторінка українського форуму програмістів

Для функціонування власне веб-форуму існує головний блок, в якому і відбувається процес обміну інформацією. На головній сторінці теми форуму розбиті на категорії, в кожньої категорії є свої топ теми і відслідковування активності з останнім дописом

Для правильної роботи запропонованої системи теж можна підібрати відлагоджену програму із впровадженням на сервер за прийнятну ціну. Можлива проблема тут - обмін даними. Наприклад, завантаженість роботи форуму із напливом учасників, чи проблеми при роботі з різними браузерами, чи пристроями, таким чином користувачі мобільних пристроїв не повинні відчувати ніяких незручностей в порівнянні з користувачами декстопних браузерів на операційних системах Windows, Linux чи MacOS.

Таким чином, до розроблюваної системи веб-форуму пред'являють спеціальні вимоги:

- Надання повного і якісного комунікативного зв’язку з усіма учасниками форуму.

- Ухвалення ефективних управлінських рішень має ґрунтуватися на повних, точних і актуальних даних по кількості нових постів, їхніх темах, кількості учасників та іншими показниками.

- Використання інформаційно-комунікаційних технологій дозволяє організувати даний процес на сучасному рівні, проводити моніторинг і оцінювати статистику використання даного ресурсу.

- Отримувати і обробляти такі дані може дозволити інтегрована система управління веб-форуму.

Виділимо основні функції сайту форуму:

- Зрозуміла ієрархія категорій, гілок обговорення і тим. Система сортування за кількістю відповідей, останній репліці, кількості переглядів і т. п.

- Оптимізація під смартфони.

- Налаштування і кастомізація профілів. У користувачів повинна бути можливість налаштовувати сторінки своїх профілів під себе, в тому числі змінювати аватарки і інші елементи веб сторінки.

- Особисті повідомлення. Форум - це перш за все співтовариство людей, а людям важливо підтримувати комунікацію. Тому в учасників повинна бути можливість переписувати в особистому форматі.

- Підписи. Одна з поширених функцій сайтів-форумів - індивідуалізована підпис, яка залишається під повідомленнями користувача в загальних обговореннях.

- Система досягнень. На деяких форумах впроваджена система рівнів і нагород, які відкриваються після певного кількостей дій (наприклад, відповідей на пости, відкриття тем). На інших майданчиках є система голосування, яка

визначає репутацію користувача. Обидва варіанти відмінно стимулюють розвиток форуму без зайвих вкладень.

- Високий рівень текстовий редактор з функціями форматування, додавання картинок і смайлів.

- Підтримка BBcode для надання користувачам додаткових можливостей форматування.

- Завантаження та прикріплення картинок, архівів та інших файлів. Реалізація цієї опції вимагає додаткової інвестиції в сховище.

- Підтримка реєстрації і авторизації користувачів. Захист акаунтів від злому через двуофакторну ідентифікацію

Виділяються три основні підходи, в рамках яких і здійснюється автоматизація та інформатизація основної діяльності вищих навчальних закладів.

У наш час не існує книги яка б описувала дослідження стосовно роботи веб-форуму, хоча це дуже важливо для створення оптимальних інформаційних даних, що забезпечують потреби розробки даної системи, тому дане дослідження потрібно провести на основі існуючих порталів. Веб-форуми можна класифікувати за різними ознаками (рис. 1.2).

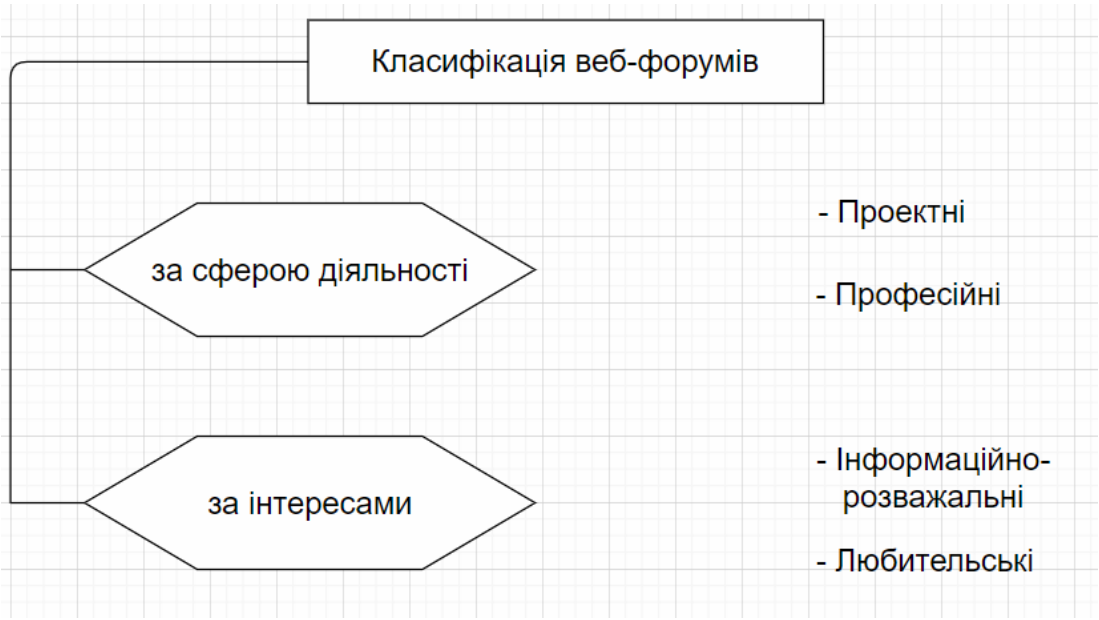


Рисунок 1.2 - Класифікація веб-форумів за різними ознаками

Проектний форум - це спілкування фахівців, які мають спільні цілі і завдання. Це можуть бути працівники однієї організації або співробітники, які працюють над спільним проектом. Як правило, доступ до участі в такому форумі обмежений або зовсім закритий.

Професійні форуми об'єднують людей однієї професії або сфери діяльності. За допомогою форумів здійснюється обмін досвідом між фахівцями.

Любительські форуми створюють люди зі схожими інтересами, захопленнями і хобі. Спілкування в співтоваристві стає способом провести дозвілля, носить виключно добровільний, і тому, як показує статистика, довгостроковий характер.

Веб-форуми інформаційно-розважального характеру носять виключно функцію обміну інформацією, новинами, чутками, анекдотами або смішними історіями чи картинками.

Головною перевагою веб-форуму потрібно назвати об'єднання людей за схожими інтересами: професії, хобі, захоплення. Крім того, спілкування в спільнотах стає приємним і корисним способом провести вільний час. Завдяки форумам кожен користувач може отримати конкретну відповідь на конкретне запитання, поділитися цікавою історією, спитати думку у людей і навіть анонімно поспілкуватися на відверті теми.

1.2. Аналіз вимог до рушія форуму

Розробляючи систему веб-форуму, необхідно врахувати такі умови:

- забезпечення найбільш рівномірного навантаження серверу на учасників процесу;

- забезпечення гнучкості роботи сервера на різних платформах, будь то мобільні пристрої чи декстопні браузери та правильне розміження всіх частин загрузки і відображення розміщення постів в форумі,

- виключення різного роду «закладок» в форумі,

- максимальна відповідність отримуваним і вихідним даним з урахуванням всіх критеріїв,

- коректне відображення всієї інформації про учасників форуму,

- надання можливості в зміні, видаленню та обмеженню в наданні особистої інформації для всіх крім адміністраторів форуму,

- створення системи особистих повідомлень,

- приєднання до групи учасників форуму,

- створення списків друзів/ігнорованих,

- керування вкладеннями,

- список підписаних тем/форумів,

- попередній перегляд посту,

- додання календара,

- пошук по темах/форумах,

- пошук по користувачах,

- перегляд всіх користувачів,

- створення системи зворотного зв’язку.

Обмеженнями в даному випадку можуть бути наступні:

1) *жорсткі обмеження*:

- видалення постів учасників, які порушують правила форуму;

- виключення учасників з форуму.

2) *відносно гнучкі обмеження:*

- рівень доступу до внутрішніх функцій форуму;

- перехід між повідомленнями, їхнє відображення в темі;

- періоди публікацій серед груп користувачів.

Для розробки даної системи веб-форуму важливою вхідною інформацією є організаційна структура підвидів, на які поділяються веб-форуми, отже за структурою форуми поділяють на наступні різновиди:

1. Табличні,
2. Деревовидні,

3. Універсальні.

Табличні форуми мають строгий порядок в постах відвідувачів форуму. Повідомлення в темах створюють стрічки, тому табличні форуми ще називають стрічковими. Для визначення адресата і адресанта відповіді, для діалогу з ними доступна панель інструментів, а саме звернення до автора, цитування та багато іншого. Завдяки суворої ієрархії в форумі з такою структурою користувачам спілкування не завдає труднощів. Натиснувши на одну з тем форуму, можна побачити всі повідомлення даного топіка. Найпопулярніші теми знаходяться в топі, на перших місцях. Рейтинг створюється за рахунок підрахунку повідомлень і перегляду.

Веб-форуми з деревовидної структурою також мають сувору ієрархію повідомлень, але іншого плану, ніж з табличній. У таких форумах все чітко і конкретно, в пості завжди вказується, кому відправляється повідомлення. Легко можна простежити, хто з ким спілкується і до якого відповіді відноситься тій чи іншій коментар. Повідомлення мають вигляд дерева, кожен відповідь і подальші коментарі можуть створити нову гілку обговорення. Якщо тема не цікава користувачеві, він може просто її проігнорувати. Але в цьому і виявляється мінус такого форуму. Для прочитання повідомлень потрібно кожен раз заходити в заголовок теми, тим самим витрачаючи багато часу. І крім того, можна припустити важливе повідомлення в гілці, яку користувач порахував нецікавою.

Універсальні ж форуми пропонує користувачам вибрати, в якому режимі працювати: в табличному або деревовидному. Це демократичний підхід, який складним виявляється тільки для початківців.

1.3. Системні вимоги до розроблюваного сервісу

Питання розробки даного програмного продукту включає в себе певні системні вимоги. Область їх визначення включає в себе пункти перелічені

вище, а також повинні включає в себе різні новітні програмні рішення, такі як: розсилка постів, відповідь на пости, моніторинг постів, розробка ієрархії постів в топіку, рівні доступу, можливість перегляду інформації, наявність керуючих команд, створення закладок і т. д.

Для вирішення даних проблем потрібно побудувати гнучку і легко адаптивну систему на основі перелічених системних вимог, з використанням сучасних веб-технологій. Необхідна система, яка повністю відповідає до всіх критеріїв та задовольняє всі запропоновані вимоги. Для забезпечення надійного функціонування системи на всіх пристроях необхідне створення декількох типових алгоритмів, що реалізують коректне відображення всіх даних з форуму. Дана система повинна мати можливість доповнення і зміни існуючої бази даних і інтерфейсу системи. Все це давало б можливість задавати для кожної окремої теми і топіку спеціалізовані вимоги, що відповідають їх умовам, і за допомогою підбору і настройки відповідної реалізації на сервері отримувати потрібну систему веб-форуму. При аналізі всіх типів форумів дійшли висновку, що табличний форум найкраще відповідає поставленим завданням, а також буде простий в розробці та використанні. Для вибраної системи потрібно розробити належну програму для роботи з сервером, задля стабільної і швидкої взаємодії з учасниками, щоб вони могли отримувати всю інформацію вчасно. Для цього, програмними методами потрібно реалізувати алгоритм взаємодії сервера з різними типами браузерів та мобільних пристроїв і таким чином забезпечити мінімальний час відгуку на запит і максимізувати ефективність збереження даних на сервері.

1.4. Дослідження та опис видів та технологій розробки сервісів на основі веб-форуму

Задачі розробки веб-форуму відносяться до класу динамічних сайтів, для яких надзвичайно важливим є змога оновлювати інформацію на сторінці на

фоні, тобто, для того щоб учасники форуму, переглядаючи пости в термін не повинні були кожні 5-10 хв обновляти сторінку, потрібно розробити алгоритм динамічного оновлення сторінки, але він не повинен загружати багато процесів на сервері і таким чином розв’язати дану проблему простим оновленням всіх існуючих процесів неможливо. Часто такі задачі зводяться до задач змінних сторінок, які формуються програмно на основі обробки запиту учасника. Традиційними методами реалізації даного типу задач розробки є різні шляхи формування сайту, а саме, поєднання різних частин в повноцінну сторінку, заповлення шаблону сайту інформацією та поєднання цих двох типів.

Поєднання різних частин в повноцінну сторінку.

В даному випадку завдання розробки даного сайту зводиться до виділення змінних частин, значення яких необхідно динамічно оновлювати та статичних частин, які мають сталий та не змінний стан протягом всього часу користування. В такому випадку складають план логічної моделі шаблону сайту у вигляді блоків, що виконують певну задачу і накладають певні обмеження на використання інших частин та складання цільової функції.

Застосування алгоритму:

1) Виділення динамічних та статичних частин.

2) Складання логічної моделі (виділення обмежень для різних частин сайту).

3) Складання алгоритму оновлення динамічних блоків.

4) Знаходження максимальної (мінімальної) похибки при роботі з алгоритмом.

Основні недоліки:

1) Збільшення витрат часу та ресурсів сервера на створення нових частин для вже існуючого сайту із зростанням розмірності та нарощуванням функціоналу та додаткових частин.

2) Відсутність гарантії правильної роботи на всіх платформах та пристроях та навантеженості сервера.

3) В силу великої розмірності бази даних та частоти її оновлення складно оцінити вплив різних чинників на процес роботи алгоритму і його результат.

Метод заповнення шаблонів.

В цьому завданні на даних момент найкраще себе редомендує метод MVC.

MVC – це шаблон в архітектурі сайтів, який передбачає поділ всієї системи на три взаємопов'язані частини: M-model (модель даних), V-view (вигляд) та C-controller(керування). Таким чином шаблон MVC ділить всі дані сайту на три різні компоненти: модель, вигляд та керування, таким чином, що кожна модифікація окремого компонента може відбуватися незалежно один від одного. Крім того кожна із частин не працює без інших, а саме Модель – відповідає за витягування і маніпуляцією даних сайту, Вигляд служить для виведення інформації на екран, тобто формуєте на сервері і відображене в браузері сторінки HTML/CSS, а контролер керує всім цим процесом (рис. 1.3).

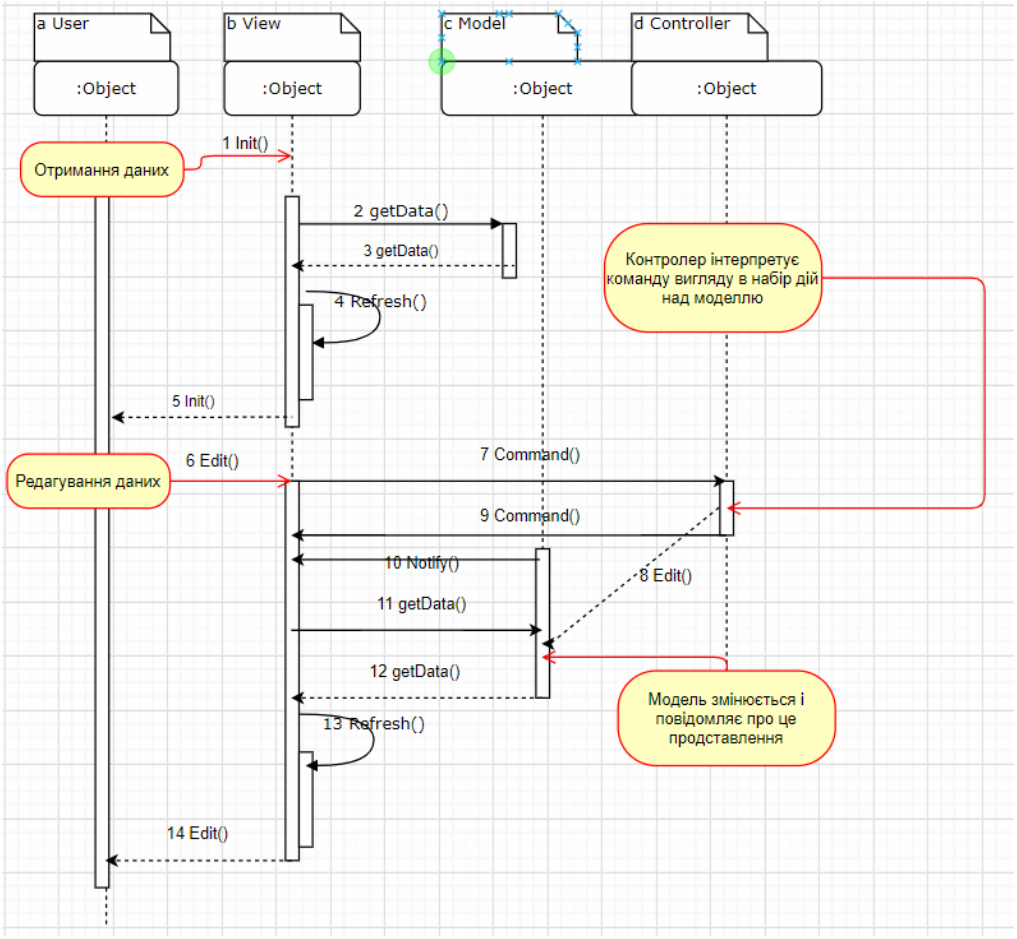


Рисунок 1.3 – Представлення роботи шаблону MVC

Повний цикл роботи даної MVC-тріади: моделі, вигляду і контролера перемикача - можна описати таким чином. При ініціалізації уявлення користувачем воно звертається до моделі і встановлює текст мітки відповідно до поточного стану перемикача. Користувач ініціює зміну перемикача, натискаючи на певну кнопку. При цьому вигляд відправляє відповідну команду контролеру: включити або вимкнути. Контролер інтерпретує команду і змінює модель. Подання реєструє зміни моделі: по цій події воно змінює текст мітки для відповідності новим станом моделі перемикача.

Розглянемо детальніше кожен компонент.

Вигляд і контролер. View безпосередньо управляє відображенням інформації користувачеві. Сама ця інформація в термінах шаблону проектування MVC називається моделлю. Контролер співпрацює з представленням, гарантуючи інтерпретацію дій користувача над даними, знову ж таки - відображенням моделі. Тому вигляд і контролер можуть бути обмежені типом даних (моделлю), з яким вони працюють.

Модель. У порівнянні з виглядом модель, навпаки, не повинна мати обмежень. Обмеження на тип об'єкту, який може функціонувати в якості моделі, в свою чергу обмежить рамки використання компонентів MVC. Необхідно, щоб будь-який об'єкт міг бути моделлю. Наприклад, число з плаваючою крапкою може виступити моделлю для відображення температури, а рядок - моделлю текстового редактора. З іншого боку, модель може бути обмежена типом даних, який вона представляє, і який успадковується від суперкласу для всіх можливих типів - класу Object. Тип даних, яким оперує модель, називають властивістю моделі.

Також в моделі MVC існує активна і пасивна модель Model. Пасивна модель не «знає» про існування уявлення, контролера, і навіть про свою участь в MVC-тріаді. Контролер відстежує зміни моделі і оповіщає уявлення. При цьому або контролер передає поданням інформацію про зміни, або уявлення самостійно вибирає дані з моделі. Більш витонченим рішенням є активна

модель. Активність моделі проявляється в її праві самостійно оповістити вигляд про зміну свого стану. Щоб не порушити основну вимогу MVC про незалежність моделі від вигляду і контролера, механізм оповіщення реалізується на основі шаблону проектування Observer.

Шаблон проектування Observer (Спостерігач), або Publisher / Subscriber (Видавець / Підписчик), використовується в разі, коли кілька об'єктів (підписники) повинні знати про зміни стану або деякі події одного об'єкта (видавця), але потрібно підтримати низький рівень зв'язності з підписниками.

Рішення полягає в тому, що підписники реєструються як обробники деякої події, що генерується видавцем. Так реалізується незалежність спостережуваного об'єкта (видавця) від оглядачів (підписників). Реалізація даного методу представлена на рис 1.4.

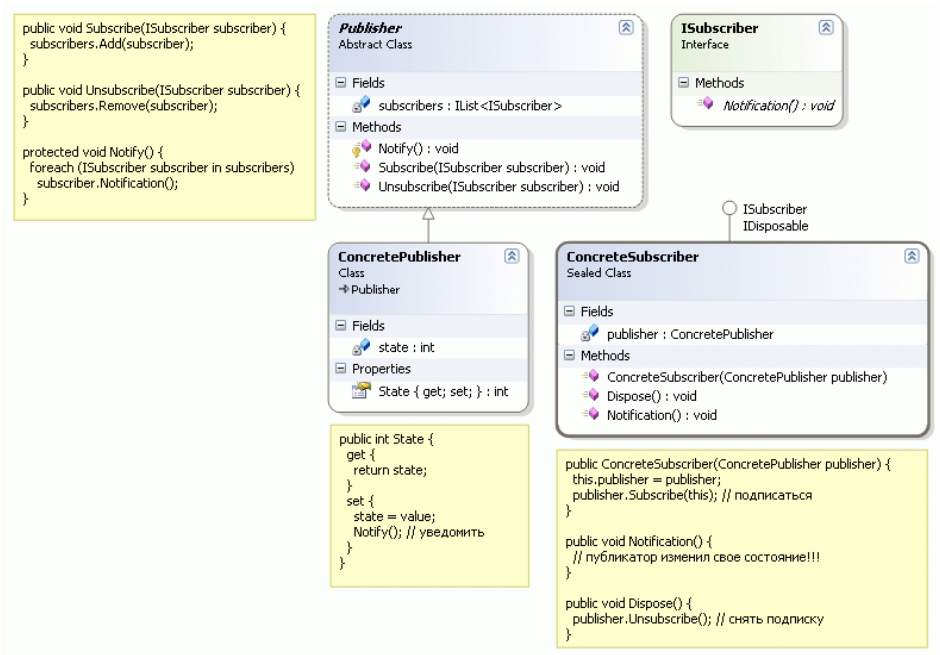


Рисунок 1.4 – Діаграма класів простору імен Rsdn.Observer

Застосування шаблону:

1) Створення одного шаблону, за яким працює весь сайт.

2) Представлення кожної частини сайту у вигляді окремої частини шаблону.

3) З'єднання частинок шаблону програмними засобами у єдиний сайт.

4) Вирішення задачі керування різними частинами шаблону.

Основним недоліком MVC методу є дублювання частин коду з різних частин шаблону.

Комбінований підхід.

Суть застосування комбінованого підходу для вирішення будь-якої задачі є наступною. Розділення завдання на дрібні блоки, для реалізації кожного з яких застосовується один із перелічених вище шаблонів. Мета розробника, в даному випадку – знайти рішення певного завдання таке, щоб воно узгоджувалося з рішеннями інших частин сайту. Блоки домагаються узгодження один з одним шляхом реалізації в них різних методів, будь то поєднання статичних і динамічних частин, або ж реалізація шаблонів.

Застосування підходу:

1) Розділення глобально сайту на більш дрібні блоки.

2) Для кожного виду блоку реалізується найбільш імпонуючий тип.

3) Планування сайту таким чином, щоб на сам перед реалізувати саме сальні сторони кожного із методів, та прикрити недоліки один одного.

Недоліки в даному випадку будуть в суб’єктивному баченні певних частин сайту та в його програмній реалізації. Також слід зазначити що, оскільки, даний метод поєднує собою два попередні, тому і їхні недоліки теж будуть актуальні для цього підходу.

1.5. Висновки до розділу

В даному розділі було розглянуто проблеми реалізації проекту з реалізацією багаторівневого ступеню доступу на прикладі веб-форуму. Зроблено аналіз вимог до сучасних шаблонів розробки веб-форуму і зроблено огляд існуючих програмних рішень.

2 РОЗРОБКА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ ТА ОПИС ПРОГРАМНОГО СЕРЕДОВИЩА ДЛЯ РОЗРОБКИ

2.1. Представлення алгоритмічного забезпечення системи

Загалом усі веб-форуми мають складну схему даних, яка складається з великої кількості таблиць. Однак значна частина цих таблиць є довідниками, які виконують допоміжну функцію.

Розглянувши та проаналізувавши існуючі системи веб-форумів, відкинувши таблиці-довідники та інші другорядні сутності, виділимо основні об’єкти Веб-спільноти як інформаційної системи.

Узагальнена схема даних, яка забезпечує необхідний набір сутностей для аналізу поведінки учасників веб-спільнот, має такий вигляд (рис. 2.1).

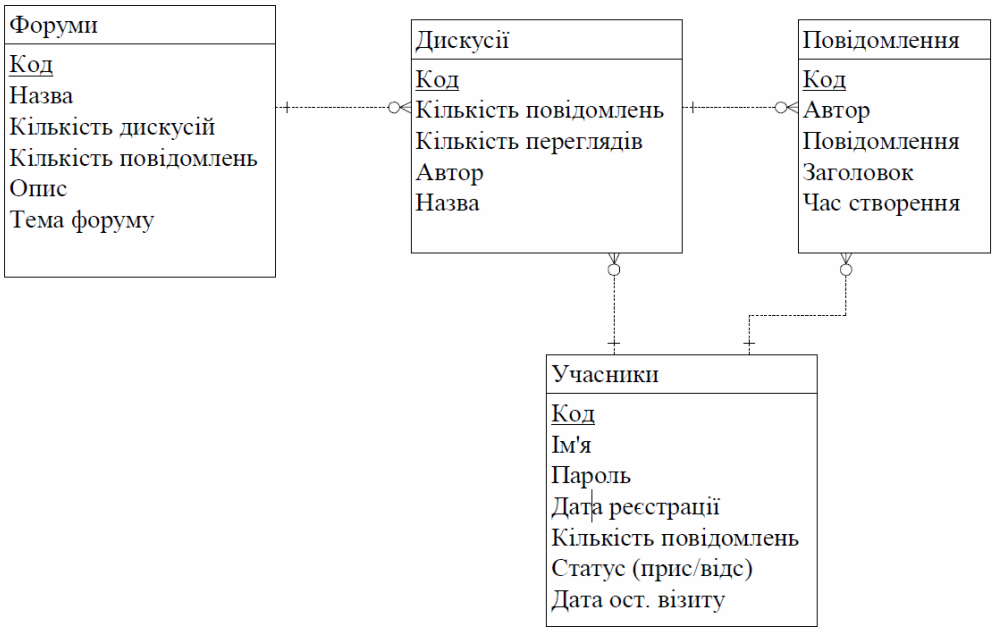


Рисунок 2.1 - Узагальнена структура даних “ядра” веб-спільноти

Форум характеризується такими параметрами:

- Назва, Кількість дискусій (які належать до цього форуму), Кількість повідомлень (які належать до цього форуму), Опис.

- Дискусія характеризується такими параметрами:

Назва, Автор, Форум (до якого належить), Кількість повідомлень, Кількість переглядів.

Повідомлення характеризується такими параметрами:

- Заголовок, Тіло повідомлення, Автор, Дата, Дискусія (до якого належить), Форум (до якого належить).

Перед тим як безпосередньо перейти до опису інформаційної моделі системи, коротко опишемо поведінку учасників спільноти, їх риси та способи їх класифікації. Учасникам Веб-спільнот притаманні такі риси поведінки: Активність, Креативність, Атрактивність, Реактивність, Лояльність.

Для моделювання рис учасників спільноти застосовується апарат нечітких множин. Усі перелічені риси учасників спільноти, відносно усього форуму можна визначити, аналізуючи їх поведінку в межах спільноти: активність визначається кількістю контенту, які вони створюють; креативність – якістю контенту і тим, як реагують на нього інші учасники спільноти; атрактивність – кількістю учасників, які реагують на створений контент; реактивність – способом участи в дискусіях, лояльність – реакцією на контент інших учасників.

Наведемо логічну інформаційну модель бази даних інформаційної системи, що містить інформацію, яка аналізується.

Узагальнена інформаційна модель веб-форуму, зображена на рис. 2.2, складається з чотирьох сутностей:

- Форуми,
- Дискусії,
- Повідомлення,
- Учасники.

Учасників класифікуємо на основі інформації, що міститься у цій базі даних.

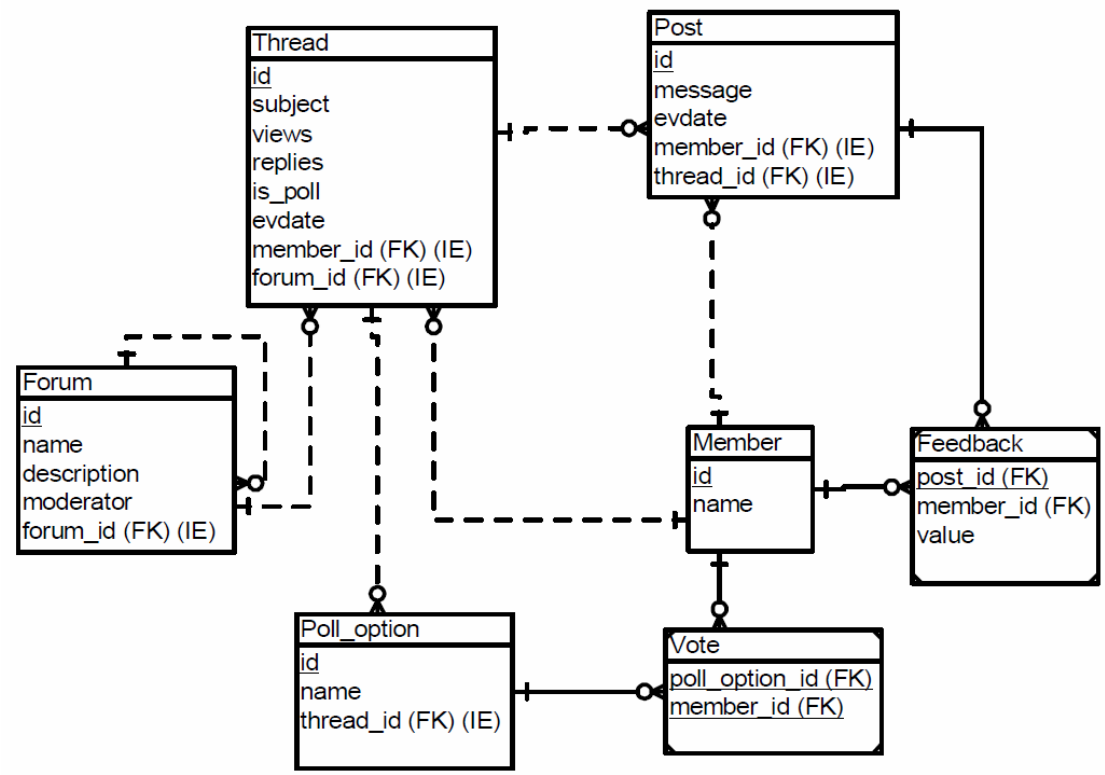


Рисунок 2.2 - Інформаційна модель веб-форуму

Важливим етапом створення інформаційної системи аналізу поведінки учасників веб-спільноти на основі форуму є розроблення структури даних, необхідних для проведення аналізу.

Для створення інформаційної системи необхідно розробити структуру метаданих, котра описуватиме класи учасників, правила класифікації та риси поведінки учасників.

Розроблена інформаційна модель зображена на рис. 2.3.

Розроблена схема даних інформаційної системи аналізу поведінки учасників веб-спільноти містить дані, необхідні для класифікації учасників Веб-форуму. Вона інтегрується з базою даних

Система управління веб-форуму на основі сутності Автор і складається з таких таблиць:

− таблиця rf_class – описує усі класи учасників Веб-спільноти;

− таблиця rf_rule – описує правила класифікації учасників Веб-спільноти;

– таблиця rf_member_feature – призначена для зберігання даних про усі риси учасників;

– таблиця rf_member_feature_variable – описує множину значень кожної лінгвістичної змінної – риси учасника Веб-спільноти.

– таблиця rf_member_feature_parameter – містить інформацію про числові межі значень кожної лінгвістичної змінної;

– таблиця rf_member_feature_parameter_type – таблиця для розрахунку рис учасників Веб-спільноти.

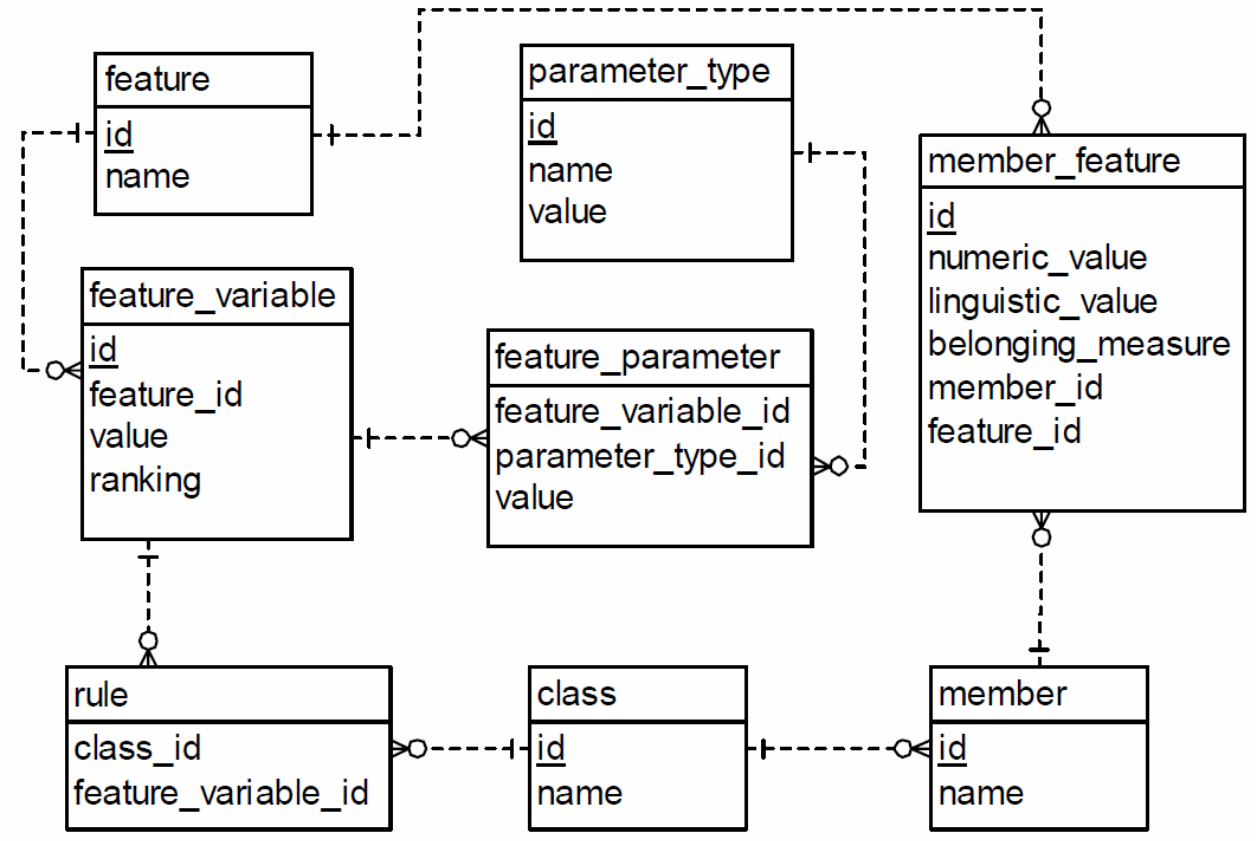


Рисунок 2.3 - Схема даних інформаційної системи аналізу поведінки учасників веб-спільноти

Створення інформаційної системи аналізу поведінки учасників спільноти є важливим і актуальним завданням, оскільки веб-спільноти є дуже популярним явищем, а управління ними – дуже складним завданням.

2.2. Програмне середовище для розробки проєкту

Для виконання поставленої задачі було використано наступне програмне середовище розробки – Microsoft Visual Studio Code. Microsoft Visual Studio Code (VS Code) - це безкоштовний текстовий редактор з відкритим вихідним кодом, розроблений компанією Microsoft. Він призначений для роботи з різноманітними мовами програмування та технологіями, такими як HTML, CSS, JavaScript, Python, Java, C++, і багато інших.

Найбільш популярні розширення представлені на офіційному сайті Microsoft (рис. 2.4)

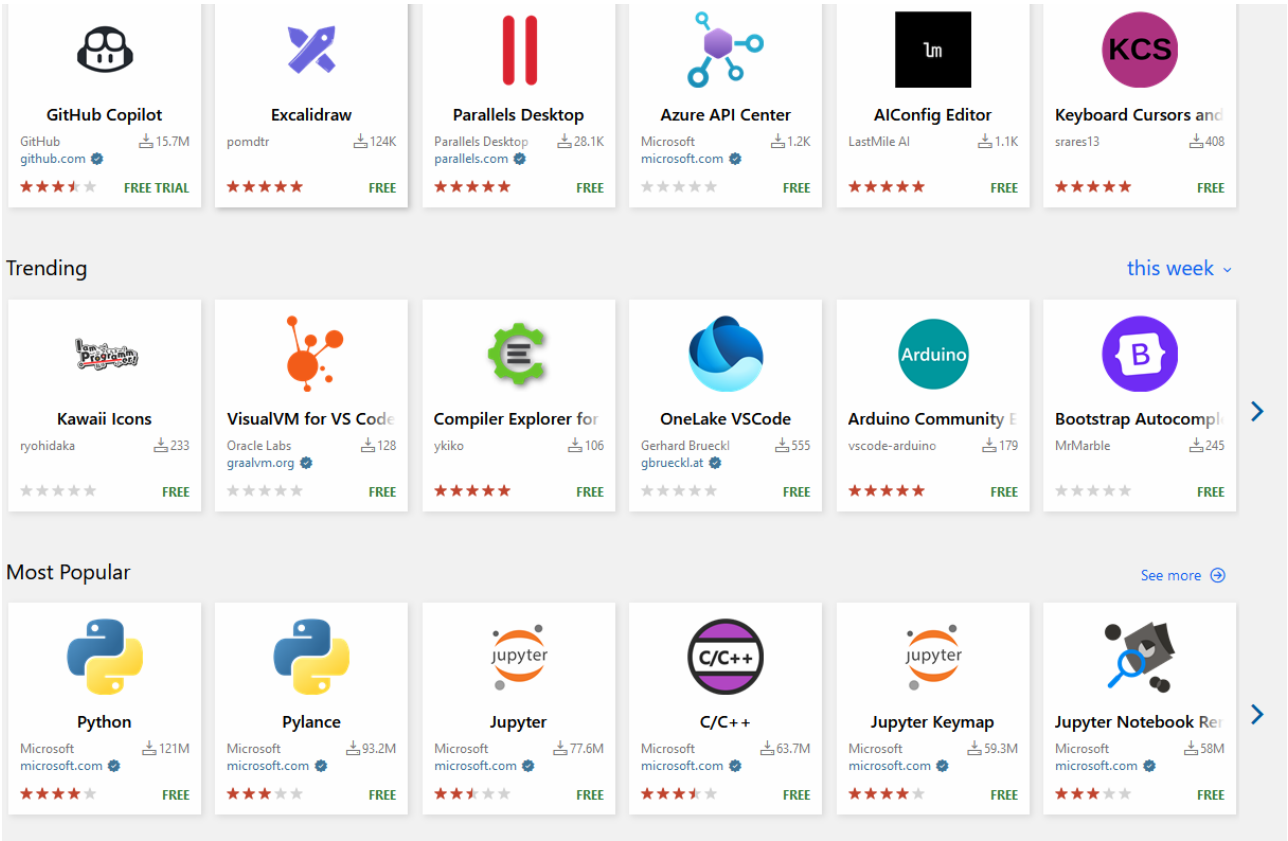


Рисунок 2.4 – Найпопулярніші розширення Visual Studio Code

Visual Studio Code поширюється на безплатній основі і працює на платформах Windows, Linux/Unix та macOS. При чому його вимоги як для сучасного редактора коду, який підтримує багато мов і парадигм

програмування доволі низькі. Ниже представлені вимоги, які рекомендують для установки.

Вимоги обладнання:

- 1,6 ГГц, або більше
- 1 Гб оперативної пам’яті
- Вільний простір на диску 200 Мб

Протестовані платформи:

- OS X Yosemite
- Windows 7 (з .NET Framework 4.5.2), 8.0, 8.1 та 10 (32-розрядні та 64-бітні)
- Linux (Debian): Ubuntu Desktop 14.04, Debian 7
- Linux (Red Hat): Red Hat Enterprise Linux 7, CentOS 7, Fedora 23

На платформі Windows 7 Visual Studio Code буде працювати тільки з версією .NET Framework 4.5.2 і вище. В даному випадку додаток був розроблений на платформі Windows 10 Pro з наступними параметрами:

- Intel(R) Core(TM) i7-7500 CPU/ 2.7 ГГц
- 16 Гб оперативної пам’яті Kingston
- SSD 500 Гб жорсткий диск

Як видно із представленого списку, переметри повністю задовільняють вимогам для редактора коду. Наступним кроком розглянемо структуру інтерфейсу редактора Visual Studio Code (рис. 2.5).

Як і багато інших редакторів коду, VS Code приймає загальний користувальницький інтерфейс та макет провідника зліва, де відображаються всі файли та папки, до яких ви маєте доступ, та редактор праворуч, що показує вміст відкритих файлів.

Користувацький інтерфейс розділений на п'ять областей:

1) Редактор - Основна область редагування ваших файлів, де можливо відкрити стільки редакторів, скільки потрібно та при тому, і вертикально, і горизонтально.

2) Бічна панель - містить різні вкладки, наприклад, Провідник, щоб допомогти вам працювати над проектом.

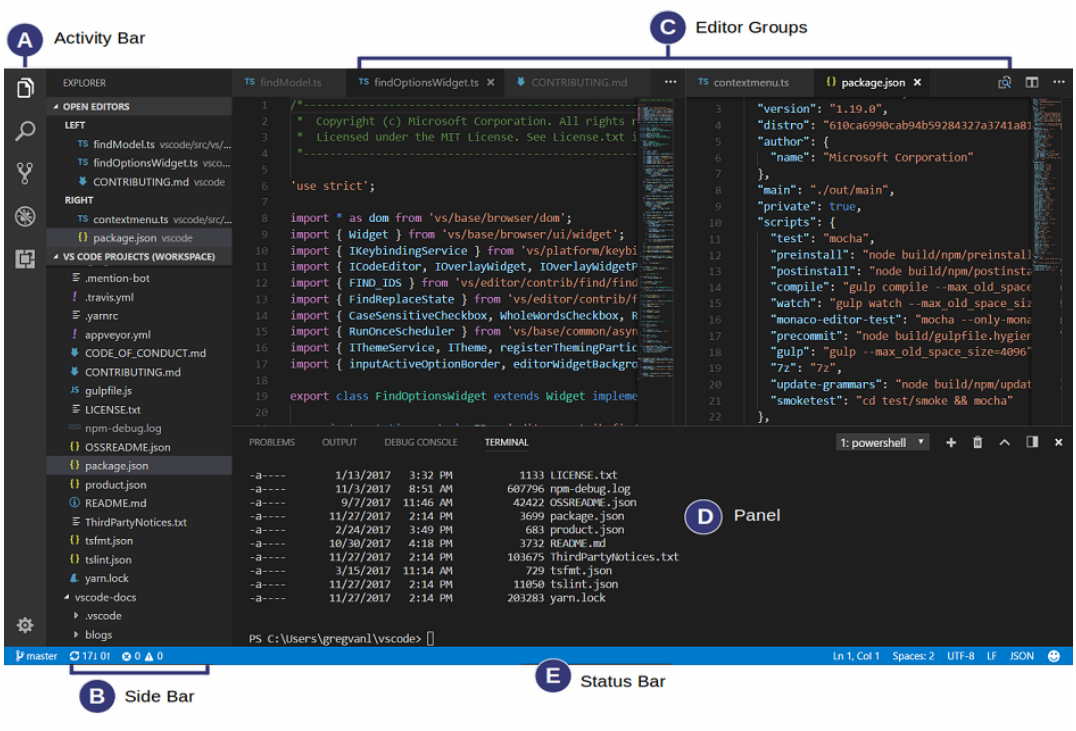


Рисунок 2.5 – Розташування різних частин інтерфейсу користувача

3) Рядок стану - відображає інформацію про відкритий проект та файли, які редагуються.
4) Рядок активності - розташований в лівій крайній частині, це дозволяє перемикатися між різними додатковими функціями та дає додаткові індикатори, що залежать від контексту, як, наприклад, кількість вихідних змін, коли розширення для контролю версій проекту Git увімкнено.
5) Панелі - Ви можете відображати різні панелі нижче області редактора для виведення інформації про налагодження додатку, вивід помилок та попереджень або інтегрованого терміналу. Панель також можна переміщувати праворуч, щоб отримати більше вертикального простору.

Кожен раз, коли запускається код VS Code, він відкривається в тому ж стані, в якому був, коли його востаннє закривали. Папка, макет та відкриті файли зберігаються. Відкриті файли в кожному редакторі відображаються із заголовками з вкладками (Tabs) у верхній частині області редактора.

Мінімапа (контур контуру) дає високий рівень огляду вихідного коду, який корисний для швидкої навігації та розуміння коду. Мінімальна карта файлу відображається в правій частині редактора. Тому можемо натиснути або перетягнути затінену область, щоб швидко перейти до різних розділів потібного файлу (рис. 2.6).

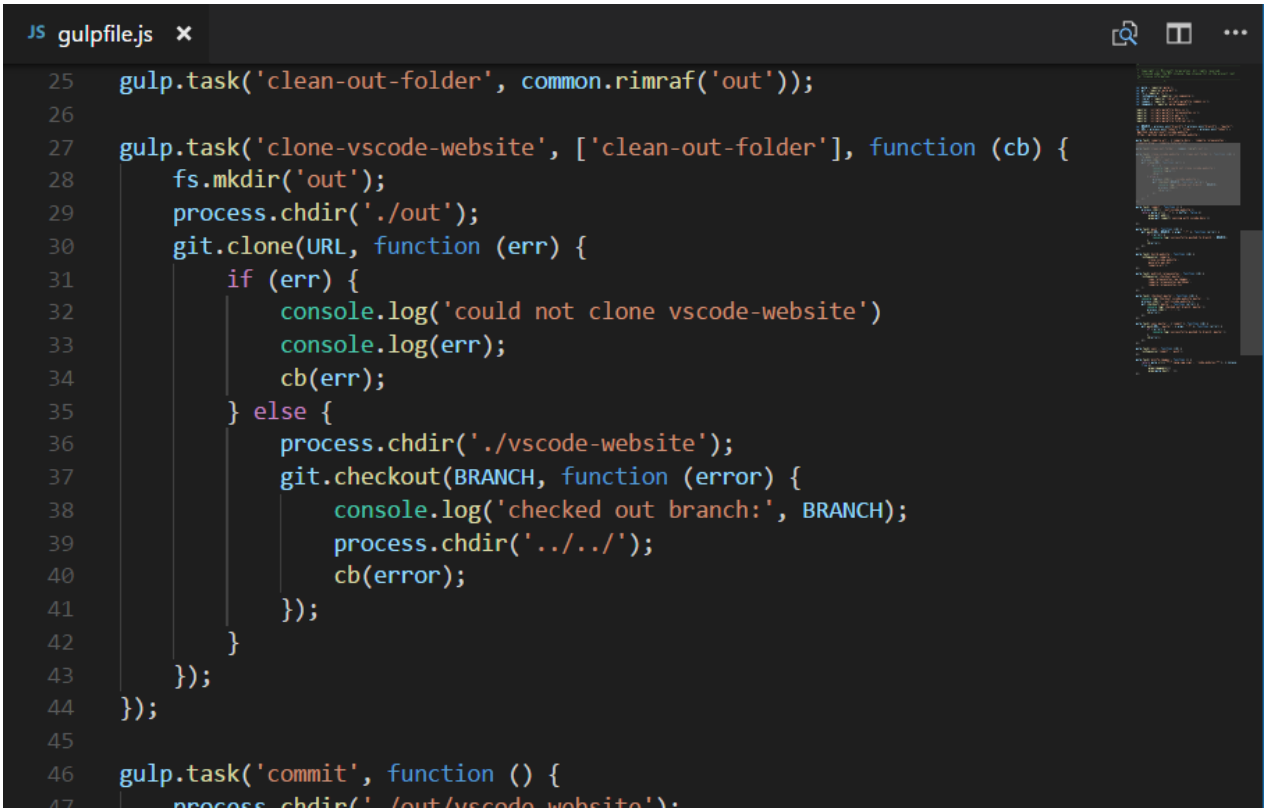


Рисунок 2.6 – Мінімапа редактора коду VS Code.

Редактор має навігаційну панель над своїм вмістом під назвою Breadcrumbs. Він показує поточне місцеположення та дозволяє швидко переходити між папками, файлами та символами (рис. 2.7).

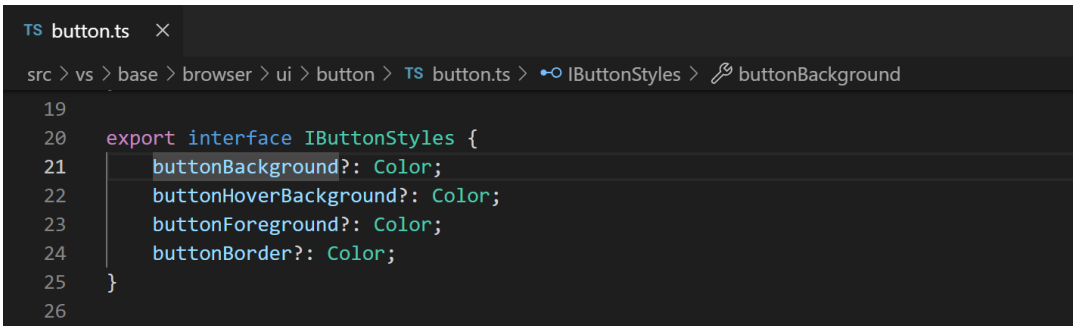


Рисунок 2.7 – «Хлібні крихти» редактора коду VS Code.

Хлібні крихти завжди показують шлях до файлу і якщо поточний тип файлу підтримує мову для символів, шлях символу до позиції курсору.

Провідник використовується для перегляду, відкриття та управління всіма файлами та папками у проєкті (рис. 2.8).

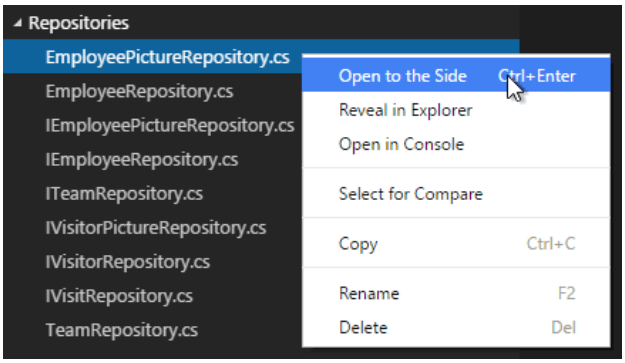


Рисунок 2.8 – Провідник редактора коду VS Code

VS Code орієнтується на файлах і папках - розпочати роботу з ними можна негайно, відкривши файл або папку у VS Code. Після відкриття папки у VS Code вміст папки відображається у Провіднику. З використанням Провідника можна зробити наступні речі:

1) Створення, видалення та перейменування файлів і папок.
2) Переміщення файлів і папок з перетягуванням.
3) Викликається контекстне меню правої клавіші миші.

VS Code надає багато варіантів налаштування редактора. У меню «view» можна приховати або переключити різні частини інтерфейсу користувача, такі

як бічна панель, рядок стану та панель активності. Більшість конфігурацій редактора зберігаються в налаштуваннях, які можна змінити безпосередньо. Можливо встановити параметри в глобальному масштабі за допомогою налаштувань користувача або для кожного проекту/папки через налаштування робочої області.

Visual Studio Code показує відкриті елементи з вкладками (заголовки вкладок) у заголовковій області над редактором. Коли відкривається новий файл, для нього додається нова вкладка (рис. 2.9).

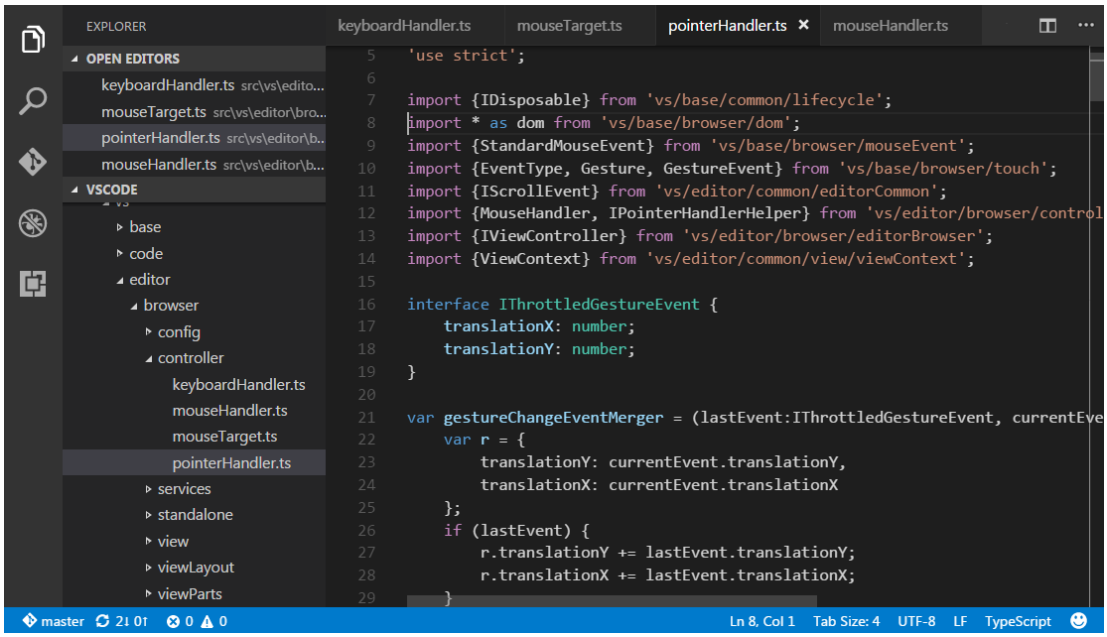


Рисунок 2.9 – Відкриття файлу в новій вкладці.

Вкладки дозволяють швидко переходити між елементами. За допомогою миші можна перетягувати вкладки, щоб змінити їх порядок. Якщо є більше відкритих елементів, ніж вміщається в області заголовка, можна скористатися командою Показати відкриті редактори (доступно за допомогою ...кнопки Додатково) для відображення випадаючого списку елементів з вкладками. За замовчуванням нові вкладки додаються праворуч від існуючих вкладок, але в налаштуваннях можливо керувати тим, як вони будуть відображатися, наприклад, зліва.

Особливості, які Visual Studio Code включає в комплекті - це лише початок. Розширення VS Code дозволяють додавати мови, утиліти та інструменти до інсталяції для підтримки робочого процесу розробки. Багата модель розширюваності VS Code дозволяє авторам розширень підключатися безпосередньо до інтерфейсу VS Code та надавати функціональні можливості через ті самі API, що використовуються VS Code.

Для виконання даної задачі було використано наступні розширення:

1) PHP IntelliSense – розширення для VS Code для написання додатків на мові PHP.
2) PHP Debug - розширення для VS Code для тестування коду написаного на PHP, та має такі особливості, порівняно з іншими схожими розширеннями:
    - Лінійні точки розриву компіляції.
    - Умовні точки розриву компіляції.
    - Точки перериву функції.
    - Step over, step in, step out.
    - Розрив входу.
    - Повідомлення про винятки та помилки/попередження.
    - Кілька паралельних запитів.
    - Сліди стека, змінні області, суперглобали, визначені користувачем константи.
    - Масиви та об'єкти (включаючи назви класу, приватні та статичні властивості).
    - Debug console.
    - Запуск як CLI.
    - Запуск без налагодження.
3) SQL Server (mssql) - розширення для VS Code для створення та роботи з базою даних для додатку. Розширення включає в себе такі функції:

- Підключення до Microsoft SQL Server, бази даних Azure SQL та сховищ даних SQL.
- Створення та керування профілями з'єднань та останніми використаними з’єднаннями.
- Написання сценаріїв T-SQL за допомогою IntelliSense, перехід до визначення, фрагменти T-SQL, кольорові синтаксиси, перевірки помилок T-SQL та роздільник пакетів GO.
- Виконання сценаріїв користувача та перегляд результатів за допомогою простої у користуванні сітки.
- Збереження результатів у форматі файлів json або csv і перегляд їх в редакторі.
- Налаштування параметрів розширення, включаючи ярлики команд і т.д.

4) HTML CSS Support - розширення для VS Code для роботи з документами мови HTML та CSS. Розширення має наступні особливості:
    - Поповнення атрибута класу.
    - Поповнення атрибута індетифікатора.
    - Підтримує завершення кодування Zen для атрибутів класу та індетифікатора.
    - Сканує папку робочої області для файлів css та scss.
    - Підтримує віддалені файли css.
5) All Autocomplete - розширення для VS Code для автозаповнення змінних в проєкті
6) Auto Rename Tag - розширення для VS Code для автоматичного перейменування тегів HTML
7) GitHub - розширення для VS Code для роботи з сервісом GitHub

Отже можна зрозуміти що Visual Studio Code – це не просто редактор коду, а й повністю адаптована під користувача система розробки програмних додатків, саме тому вона і була вибрана як середовище розробки даного застосунку.

2.3. Програмні додатки для реалізації серверної основи

Під час планування апаратної частини додатку було переглянуто багато сайтів – постачальників безкоштовних веб хостингів, іншими словами – серверів, для реалізації розробляємого застосунку на серверній основі, саме тому було обрано безкоштовний веб хостинг InfynityFree (рис. 2.10).

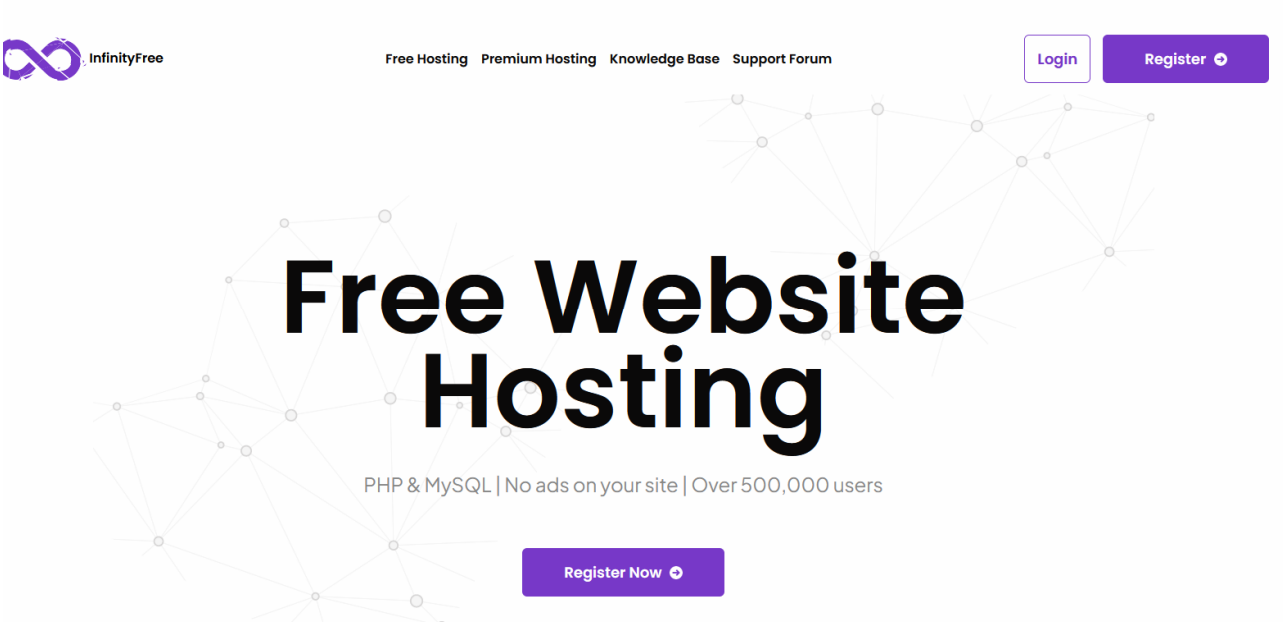


Рисунок 2.10 – Головна сторінка сайту InfynityFree

Даний хостинг має переваги над іншими безкоштовними веб хостингами, а саме:

1. Безлімітний дисковий простір.
2. Безлімітна пропускна здатність.
3. 400 MySQL баз даних.
4. Повна підтримка .htaccess.

5. Безкоштовні доменні імена.
6. Безкоштовний SSL в домені.
7. Безкоштовний сервіс DNS.

Крім того InfynityFree не має якогось певного періоду безкоштовного розміщення сайту на їхньому хості, тому сайт може розміщуватися на такому хості допоки діє акаунт, на який зареєстровано домен. Є кілька способів завантажити на хост свій сайт, можна для цього використати існуючий скрипт, або ж CMS(наприклад, WordPress), за допомогою програми Softaculous, перенести вже існуючий сайт чи завантажити власний код за допомогою FTP.

Крім того важливим плюсом є використання власних доменних імен сайту. InfynityFree дозволяє використовувати доменне ім’я, яке було зареєстровано в зовнішньої компанії, наприклад, NameSilo, та використовувати його в InfynityFree. Перш ніж додати доменне ім’я до свого облікового запису, потрібно перевірити його, встановивши сервери імен домену. Можливо змінити сервери імен зовнішнього домену через панель постачальника доменних імен.

Щоб додати своє доменне ім’я до нового облікового запису хостингу, потрібно просто увійти у свою клієнтську область, натиснути кнопку Створити акаунт та вибрати опцію "Спеціальний домен". Потім ввести своє власне доменне ім'я у поле (без http: // або www) та продовжувати налаштування, як показано там. Також користувач може додати новий домен у створений раніше обліковий запис (можливо, за допомогою іншого користувальницького домену або безкоштовного субдомену). Якщо потрібно створити окремий веб-сайт або просто не має інших доменів в обліковому записі, можна скористатися розділом Addon Domains. Просто введіть туди новий домен, і створиться новий каталог для завантаження файлів веб-сайту (як правило <your domain>/htdocs).

Для того щоб надіслати файли з персонального комп’ютера на сервер потрібно скористатись сервісом FTP. Щоб завантажити файли за допомогою FTP, потрібно отримати наступні дані:

- Ім'я користувача FTP

- FTP-пароль
- Ім'я хоста FTP
- FTP-клієнт

InfynityFree надає FTP-ім'я користувача, пароль та ім’я хоста в області клієнта в розділі Інформація про FTP. Пароль FTP такий же, як і пароль облікового запису хостингу та панелі управління, який відрізняється від пароля для клієнтської області. Для надсилання коду на сервер було використано FTP-клієнт FileZilla (рис. 2.11).

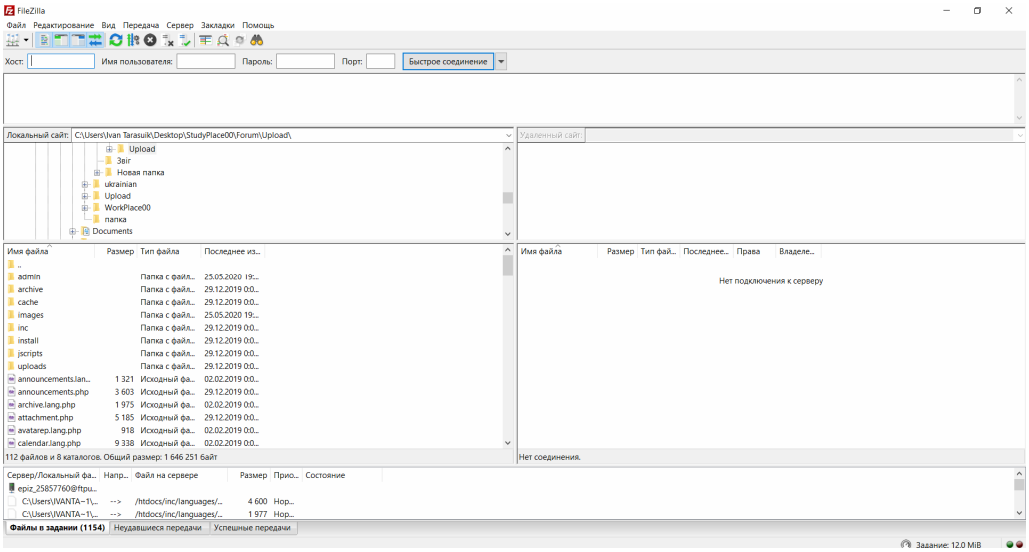


Рисунок 2.11 – Головна сторінка FileZilla

FileZilla - популярний безкоштовний FTP-клієнт, доступний для більшості поширених операційних систем. FileZilla Client мультиплатформний клієнт FTP з відкритим кодом. Доступний на Windows, Linux та Max OS X. Підтримує системи FTP, SFTP та FTPS. Налаштування з'єднання з сервером:

1. Відкрийте FileZilla
2. Відкрийте файл> Менеджер сайтів
3. Клацніть Новий сайт
4. В хості введіть ім'я FTP-сервера: ftpupload.net
5. У спадному меню Logontype виберіть Normal
6. Введіть своє ім’я користувача (наприклад, epiz_12345678)

7. У розділі Пароль введіть свій пароль
8. Натисніть кнопку ОК

При успішному підключенні з сервером FileZilla повинен зберегти інформацію про підключення у FileZilla. Якщо користувач бажає, можоиве знову відкрити Менеджер сайтів і натиснути Новий сайт у розділі Вибрати запис, щоб змінити ім’я з Нового сайту на щось значиме, наприклад, you.example.com. Цей параметр може допомогти вам легше визначити обліковий запис.

Передача файлів:

1. Відкрийте FileZilla
2. Відкрийте файл> Менеджер сайтів
3. Клацніть веб-сайт, до якого ви хочете підключитися, у розділі Вибір запису
4. Клацніть Підключити

Тепер повинна було створитись підключення до свого облікового запису FTP і відображуватися файли праворуч - це файли на комп’ютері, ліворуч повинні відображуватися файли на сервері.

Використовуючи ці два представлення, можливе завантажування файлів перетягуючи їх, вибираючи файли, які можливо скачати зліва, та перетягуючи їх праворуч і навпаки для файлів, які ви хочете завантажити.

Майте на увазі, якщо у вашому обліковому записі є кілька сайтів, таких як субдомени або домени, завжди слід завантажувати їх у відповідне місце для цих піддоменів або доменів (наприклад, /example.com/htdocs або /example.epizy. com / htdocs)

Видалення файлів:

1. Клацніть правою кнопкою миші файли / каталоги, які потрібно видалити
2. Виберіть Видалити

Зміна дозволу на файл / каталог:

1. Клацніть правою кнопкою миші файл / каталог

2. Виберіть дозволи на файл ...
3. За допомогою прапорців встановіть нові дозволи
4. Натисніть кнопку ОК

Після створення підключення переходимо до головної панелі управління акантами (рис 2.12).

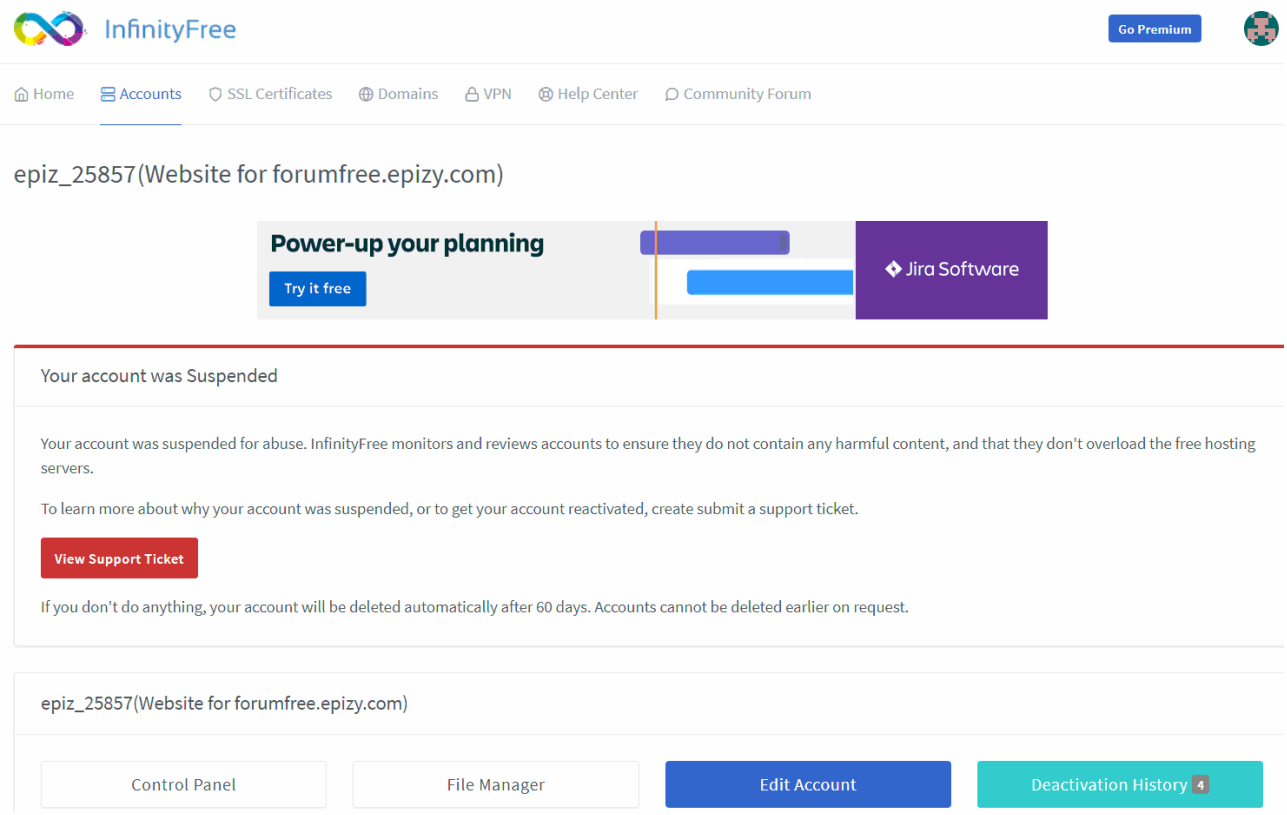


Рисунок 2.12 – Головна сторінка управління акаунтами

В даному випадку даний акаунт призупинений, оскільки час від часу адміни хосту проводять перезапис активних доменів і таким чином проводять оптимізацію та відсіюють сайти якими давно ніхто не користувався. В разі призупинення акаунту на головну сторінку приходить повідомлення, яке можна бачити вище. Для того, щоб активувати акаунт потрібно написати в підтримку хосту. В цілому час відгуку адміністратора складає не більше 1 години, крім того, при необхідності адміністрація хосту вкаже на помилки, які допустилися при активації акаунта і допоможе їх вирішити. В даному випадку, цей акаунт

використовував невідповідну адресу електронної пошти, яка суперечила політиці використання хосту і позиціонувалася як спам-пошта.

Крім того на хості існує змога редагувати та змінювати інформацію про акаунт, що включає в себе зміну даних про акаунт, зміну паролів для FTP та для Панелі контролю, також передбачено видалення акаунту.

На головній сторінці відображені основні дані про акаунт, а саме такі як деталі про акаунт, про FTP та про MySQL (рис. 2.13).

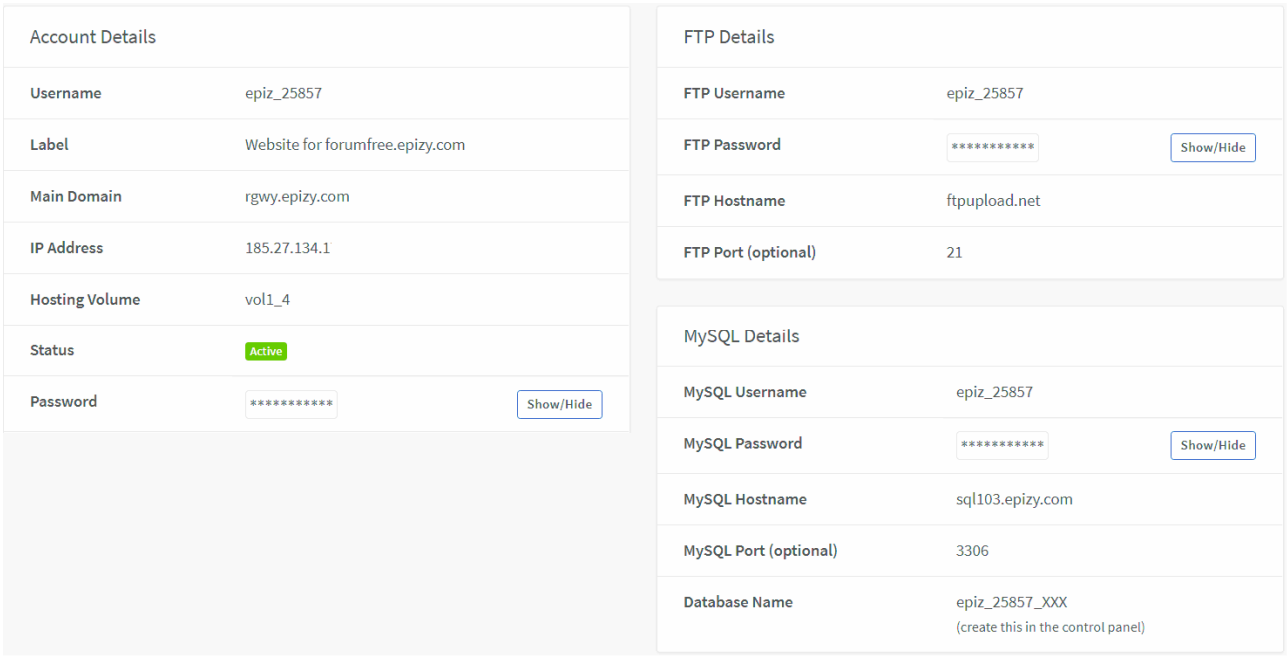


Рисунок 2.13 – Деталі про акаунт

Також в самому хості можливо переглянути файли, які знаходяться в на сервері використовуючи для цього вкладку File Manager (рис 2.14)



Рисунок 2.14 – Перелік файлів і папок, які знаходяться на сервері

Як ми можемо бачити в сам сервер внесено 3 файли по-замовчуванню, а саме .htaccess, .override та DO NOT UPLOAD FILES HERE. Ці файли необхідні для роботи сервера, тоді як сайт користувача повинен зберігатися в папці htdocs (рис. 2.15).

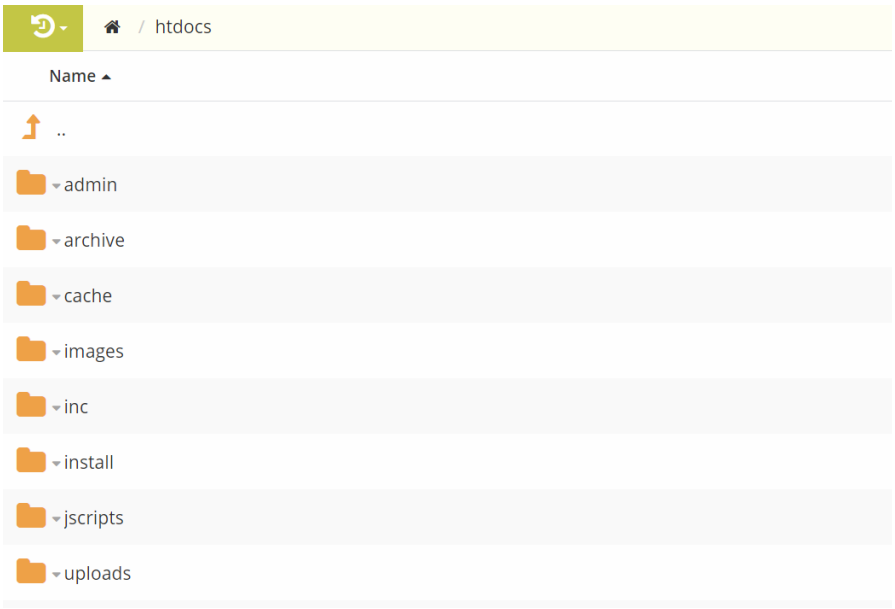


Рисунок 2.15 – Вміст папки htdocs на сервері

Також з головної сторінки керуванням акаунтами можна перейти на панель управління сайтом (рис. 2.16).

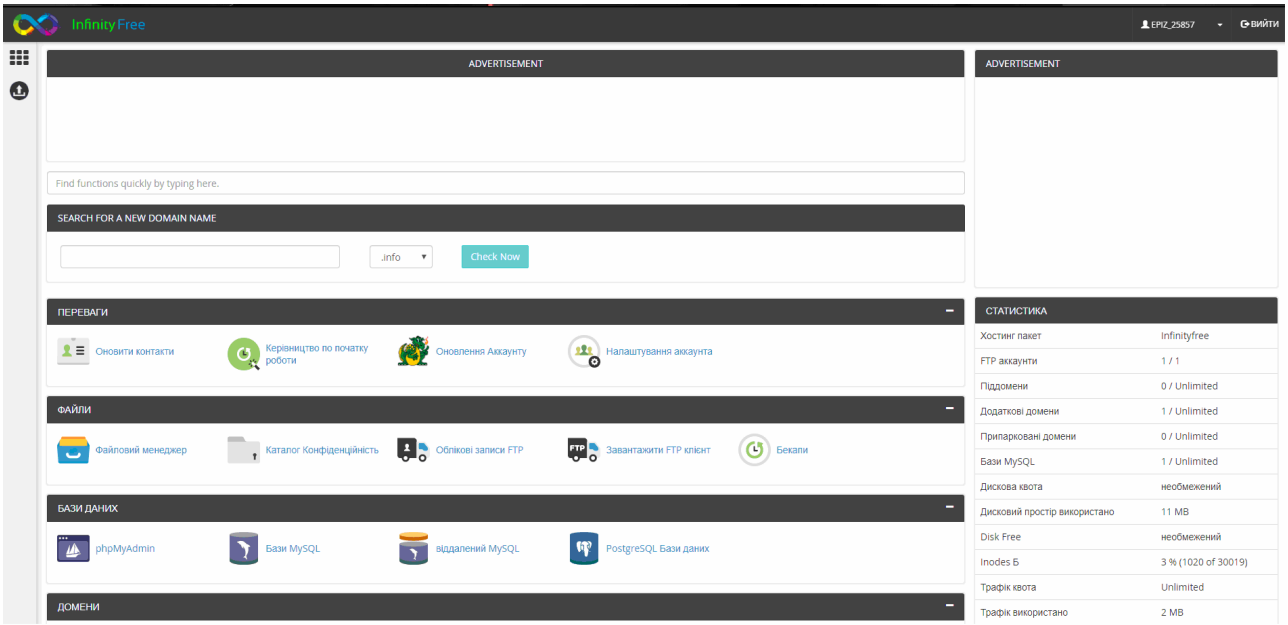


Рисунок 2.16 – Панель управління сайтом

Панель управління сайтом відіграє основну роль в процесі керування сервером, оскільки вона дозволяє не тільки переглядати всію інформацію про підключення, і змінювати настройки приватності, керувати файлами, базами даних та FTP підключеннями. Наприклад, вкладка «налаштування акаунту» дозволяє переглядати і редагувати інформацію про акаунт і сервер (рис. 2.17-2.18).

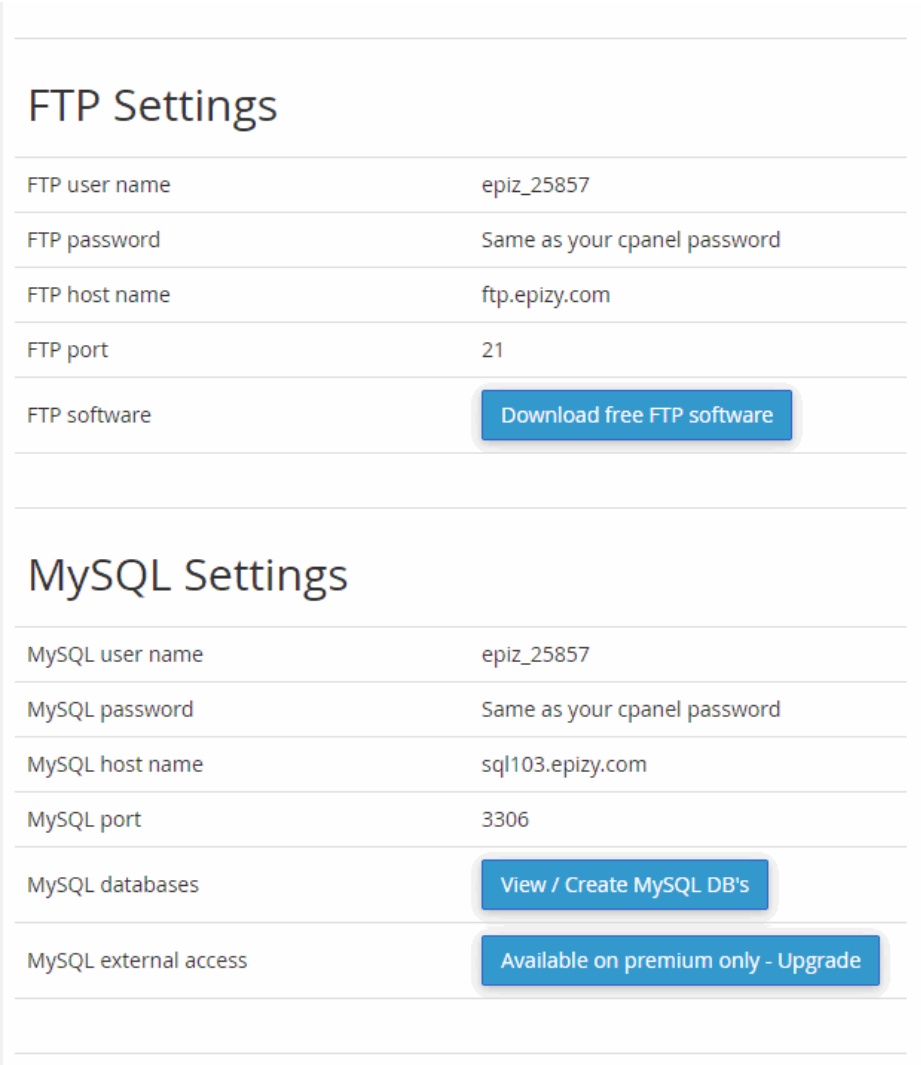


Рисунок 2.17 – Налаштування FTP і MySQL

Дана частина відповідає за налаштування FTP і MySQL акаунту, та дозволяє швидко перейти до завантаження FTP програм та відкрити, або ж створити нову MySQL базу даних.

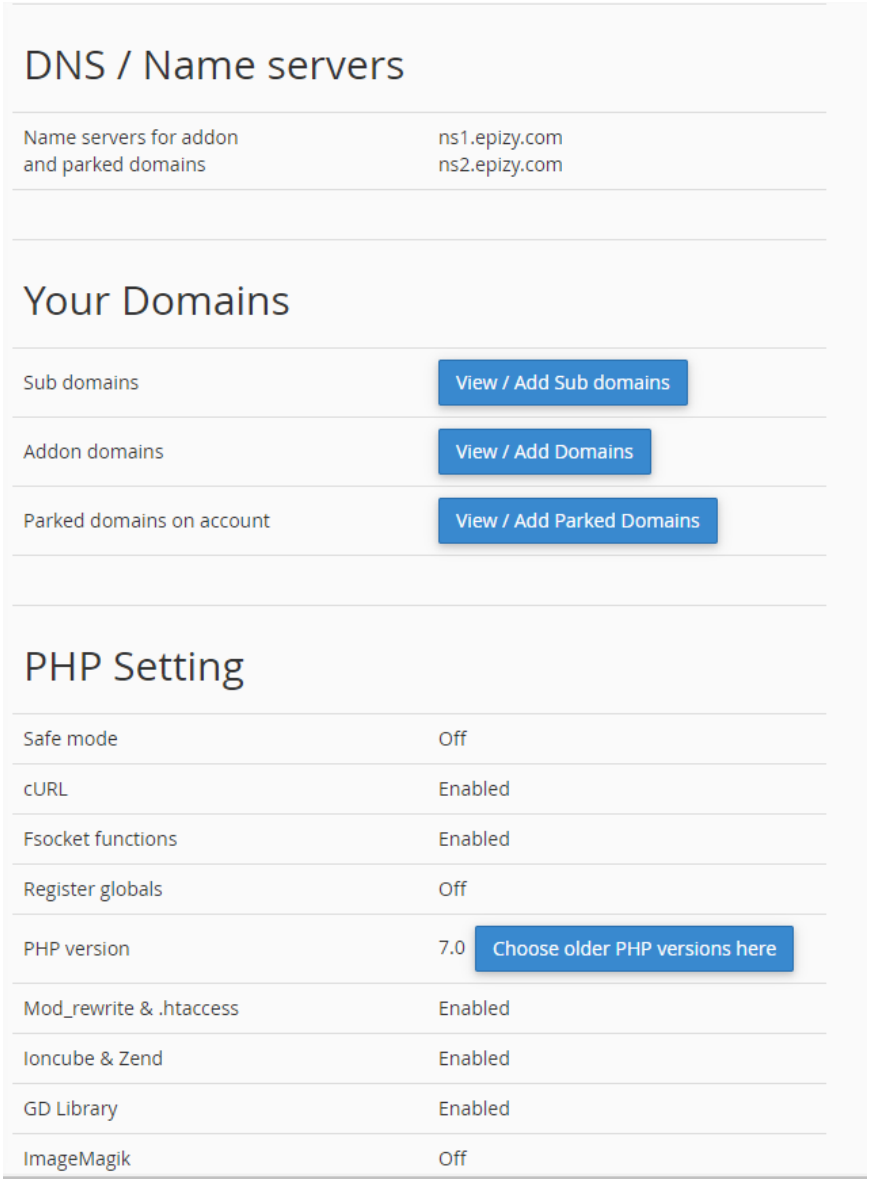


Рисунок 2.18 – Налаштування акаунту з переглядом імен домену і налаштуванням PHP

При створенні власного сайту надзвичайно важливо старатись не допускати помилок після встановлення його на сервер, бо ж краще не чіпати те, що працює. Але ніхто не застрахований від помилок і оскільки даний додаток в майбутньому буде також доповнюватися і розширюватися, та і в процесі його експлуатації можливі виникнення певних збоїв, для цього передбачено наявність бекапу сайту (рис 2.19).

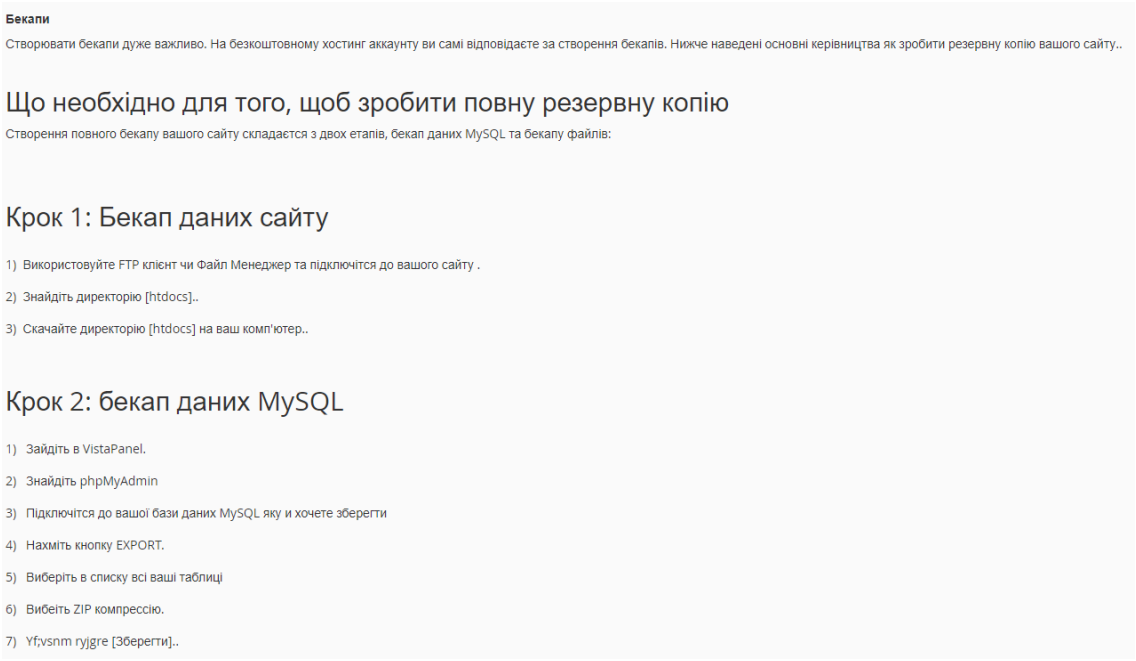


Рисунок 2.19 – Створення бекапу серверу

Також не менш важливим є надійне функціонування бази даних. В даному випадку була використана MySQL база даний надана самим сервером (рис. 2.20).

MySQL – це безкоштовна реляційна база даних з відкритим кодом, на даний момент MySQL найпопулярніша база даний, насамперед через її високу підтримку серед мов програмування і швидкий час відгуку на запит. Саме тому її часто використовують в динамічних сайтах.

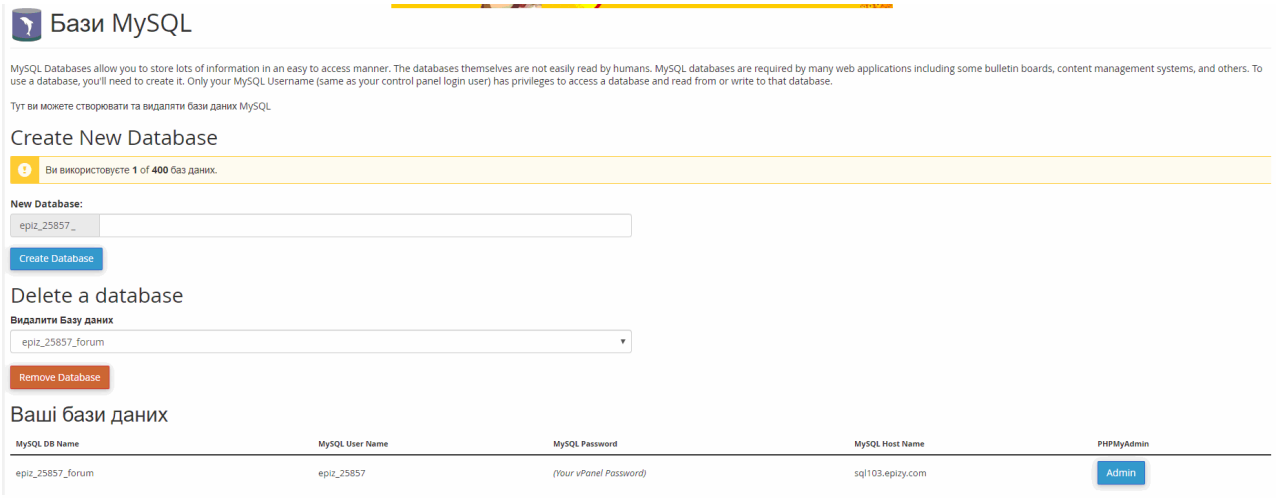


Рисунок 2.20 – Панель керування базами даних сервера

Крім того, InfynityFree надає всім користувачам переглядати статистику сайту (рис 2.21-2.22). Таким чином можливе чітке і точне менеджерування свого сайту, без використання сторонніх засобів моніторингу, які б збирали ваші дані.

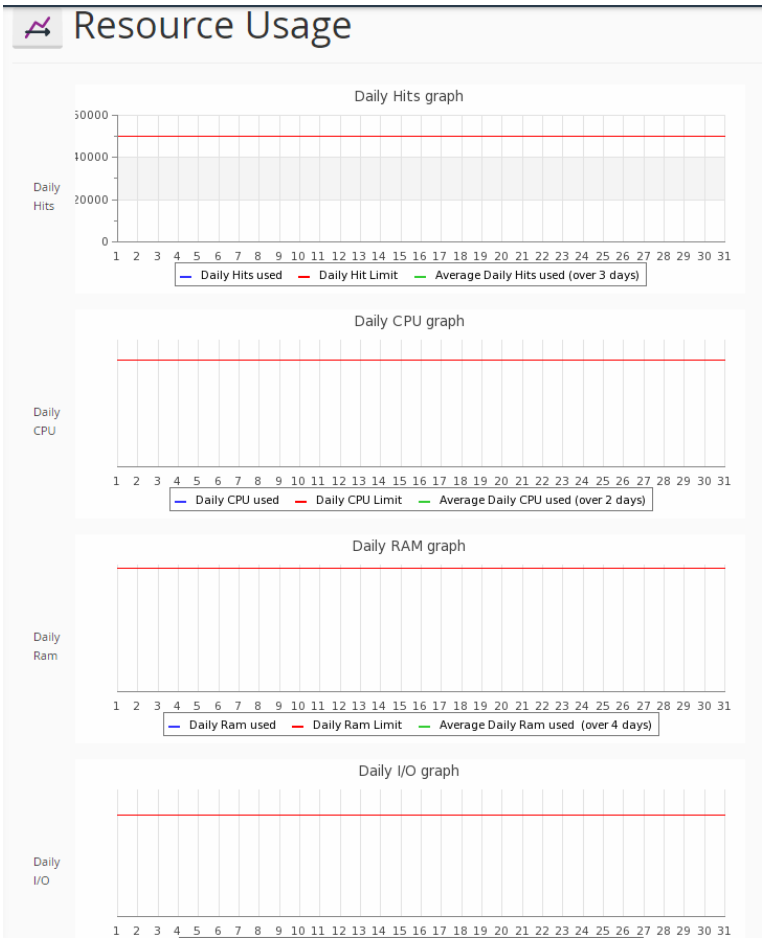


Рисунок 2.21 – Моніторинг даних на сервері

В даному випадку відображається навантаження на сервер в графах. Оскільки даний акаунт був недавно реактивовано, тому всі дані були обнулені, в іншому випадку тут відображається навантаження на сервер, по періодам та датам. В наступному рисунку зображені графи використання трафіку і використання місця на диску.

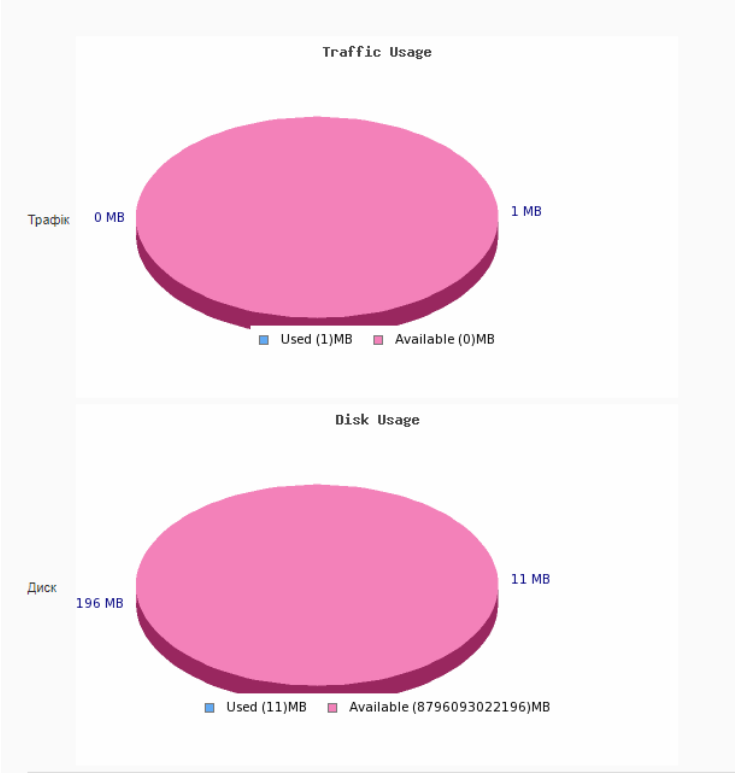


Рисунок 2.22 - Моніторинг даних на сервері

Також, при потребі можливо змінити версію PHP, по замовчуванні на сервері стоїть версія 7.3 (рис. 2.23).

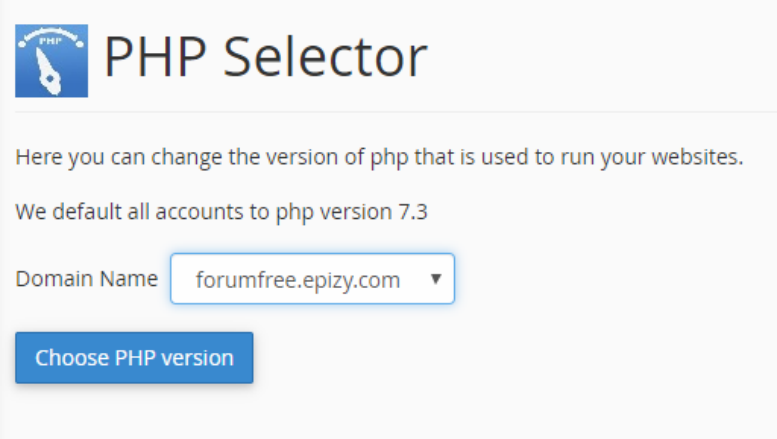


Рисунок 2.23 – Вибір версії PHP

Отже, однією із переваг даного хосту є чудове середовище керування процесами на сервері та менеджерування і моніторинг їх.

2.4. Побудова UML-діаграм веб-орієнтованого сервісу

Ключовими функціональними вимогами до сервісу є:

- кроссбраузерность і кроссплатформенность;
- зручний призначений для користувача інтерфейс;
- захищеність від злому;
- проста і зрозуміла авторизація;
- можливість розширення функціоналу.

Робота з форумом повинна передбачати: створення і перегляд тим, написання і читання коментарів, оцінку коментарів, фільтрацію контента. При вході на форум користувач вибирає режим гість (рис. 2.24) або ж входить під своїм обліковим записом і стає користувачем (рис. 2.25). В разі якщо облікового запису немає, користувач за бажанням може зареєструватися. Всі функції адміністратора форуму наведені на рис. 2.26. Загальна діаграма використання веб-форуму на рисунку 2.27.

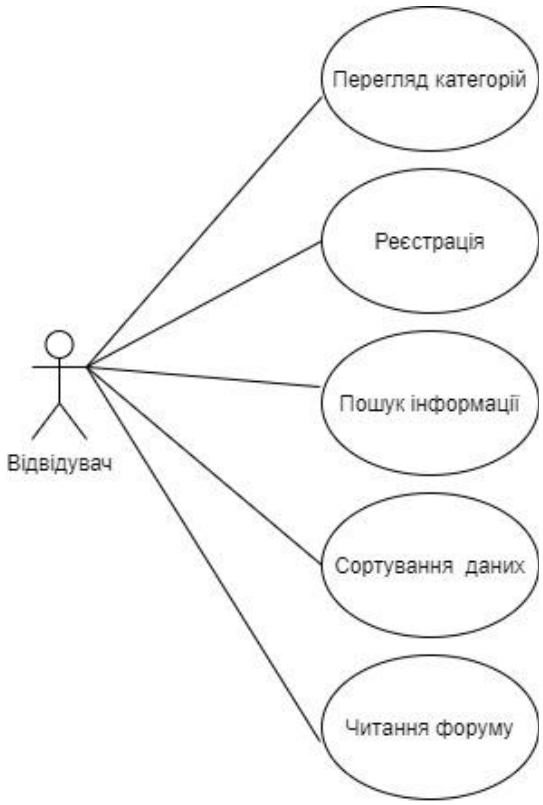


Рисунок 2.24 – Діаграма варіантів використання для актора «Відвідувач»

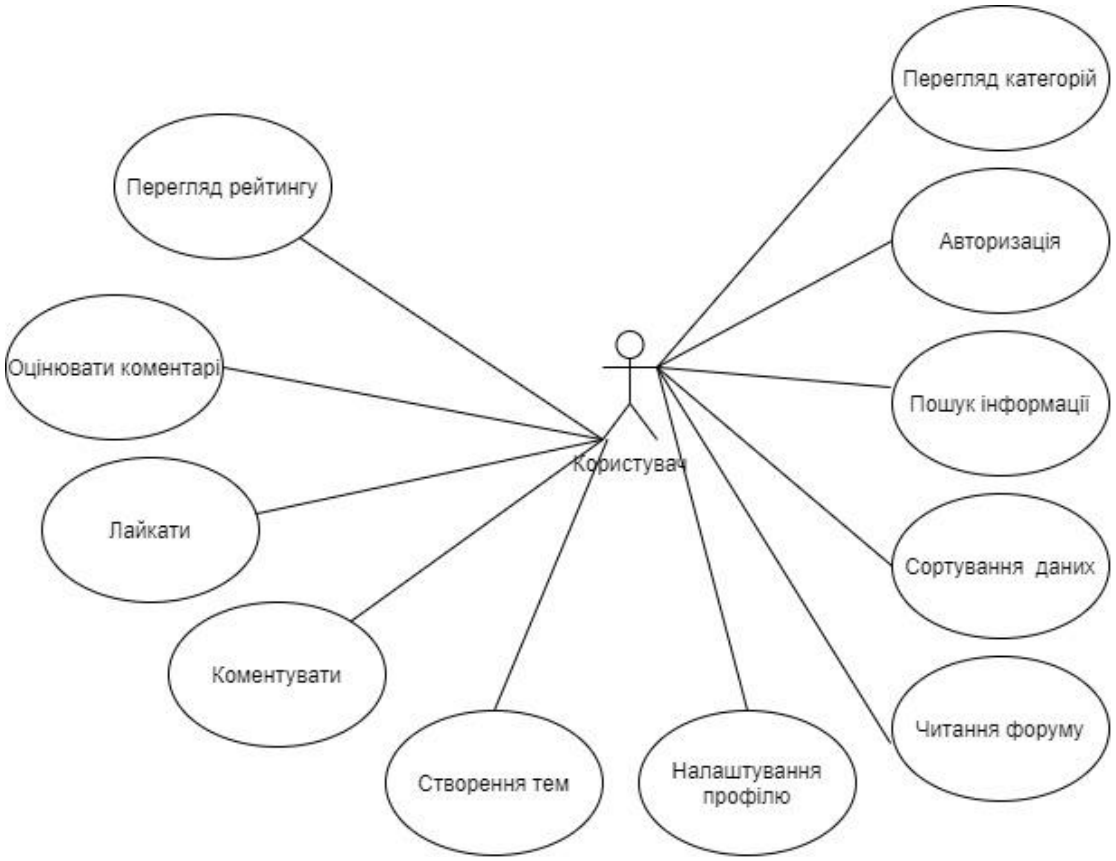


Рисунок 2.25 – Діаграма варіантів використання для актора «Користувач»

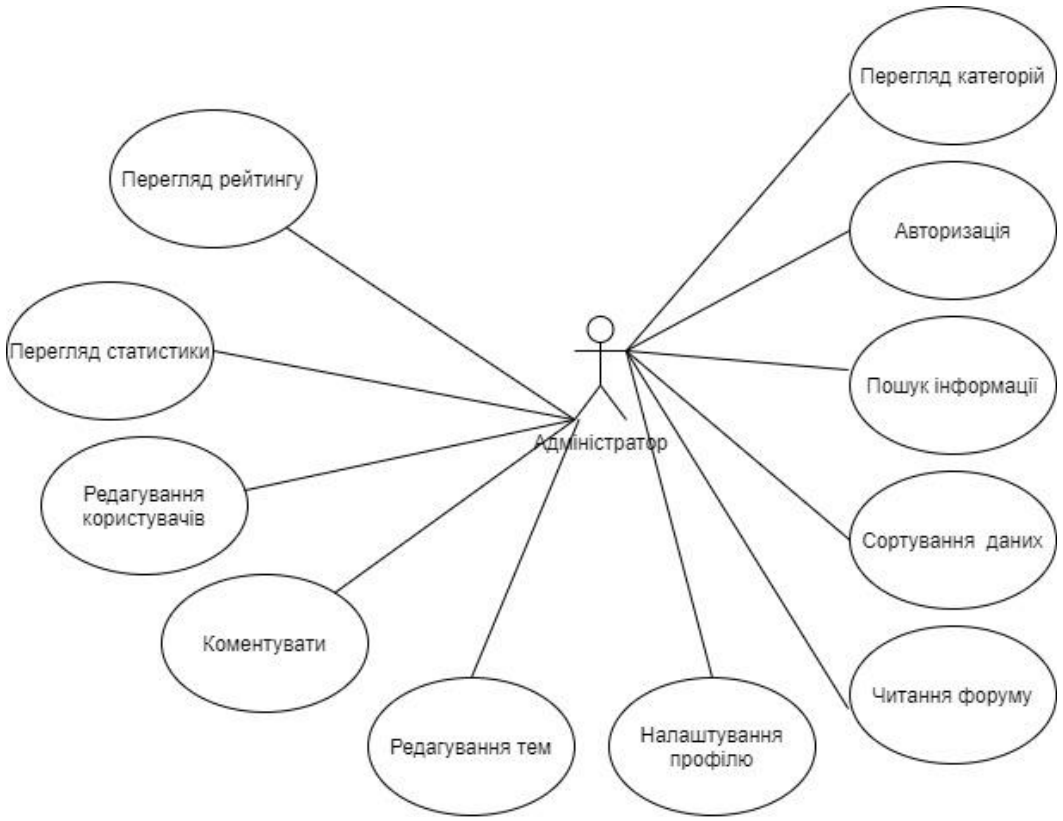


Рисунок 2.26 – Діаграма варіантів використання для актора «Адміністратор»

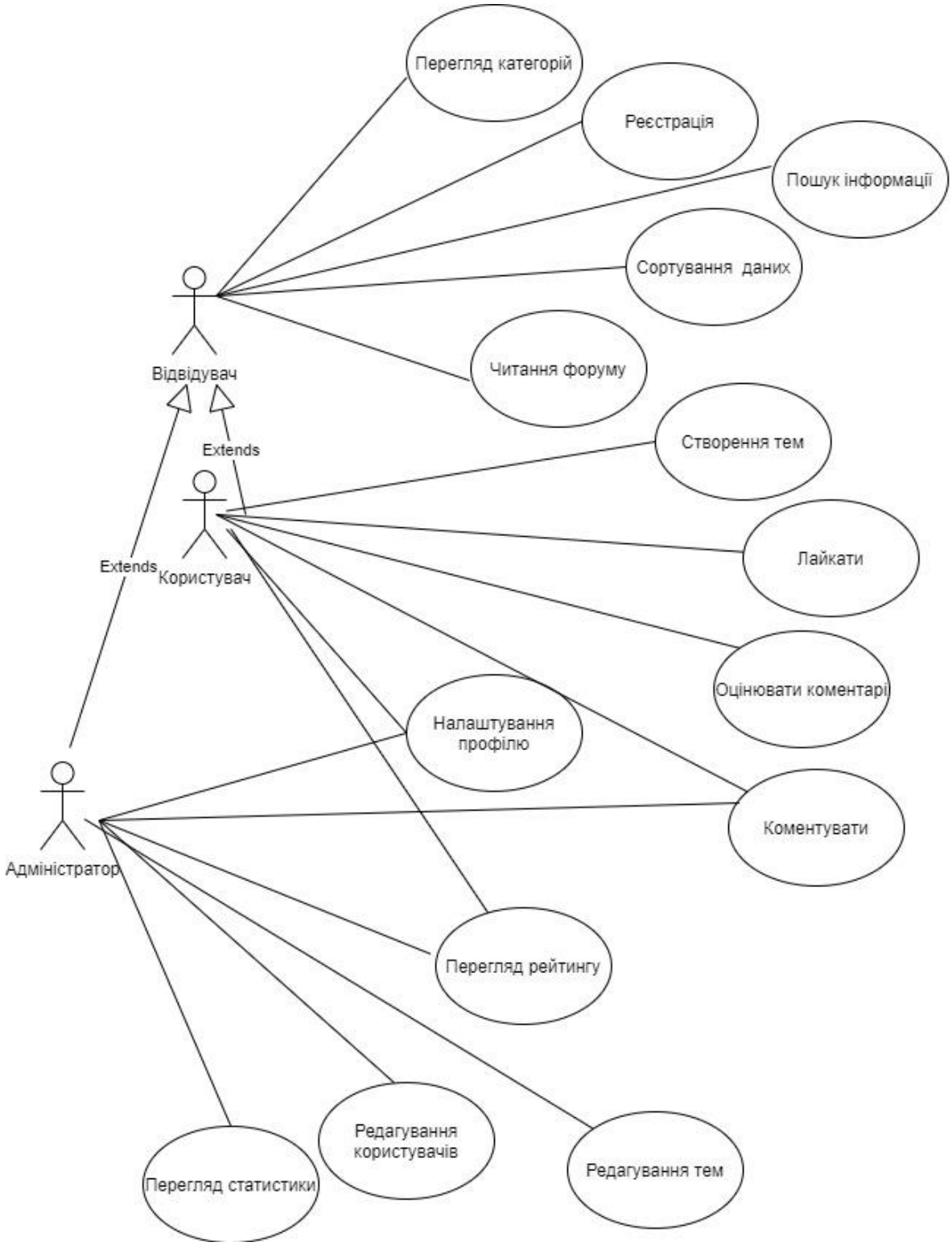


Рисунок 2.27 – Діаграма варіантів використання веб-форуму

Динамічна модель необхідна, щоб відобразити функціональність (поведінку) системи. З цією метою використовуються діаграми, які дозволяють

деталізувати особливості алгоритмічної і процедурної реалізації виконуваних системою операцій. Однією з таких діаграм є діаграма діяльності, яка застосовується для моделювання динамічних аспектів поведінки системи. Діаграма діяльності показує, як потік управління переходить від однієї діяльності до іншої (рис. 2.28 - 2.29).

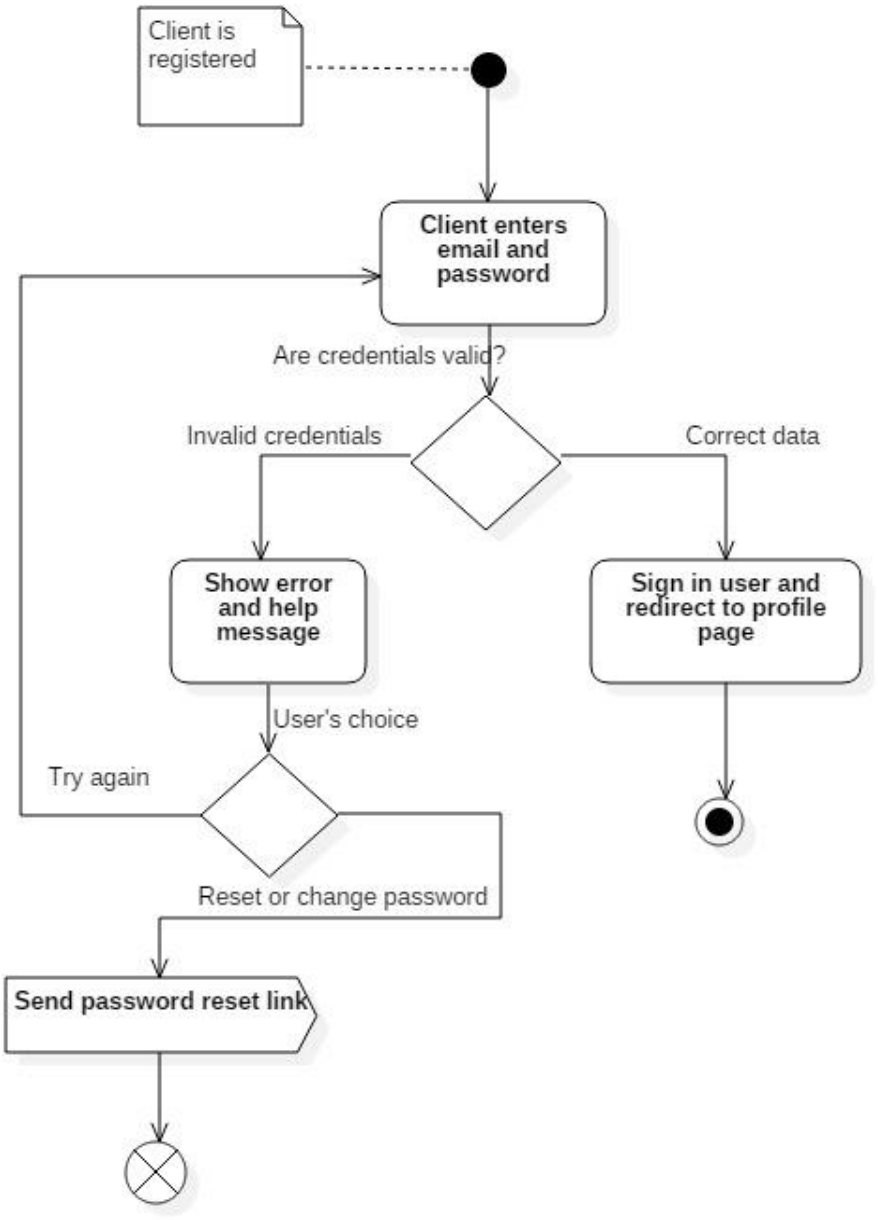


Рисунок 2.28 – Діаграма діяльності «Реєстрація користувача»

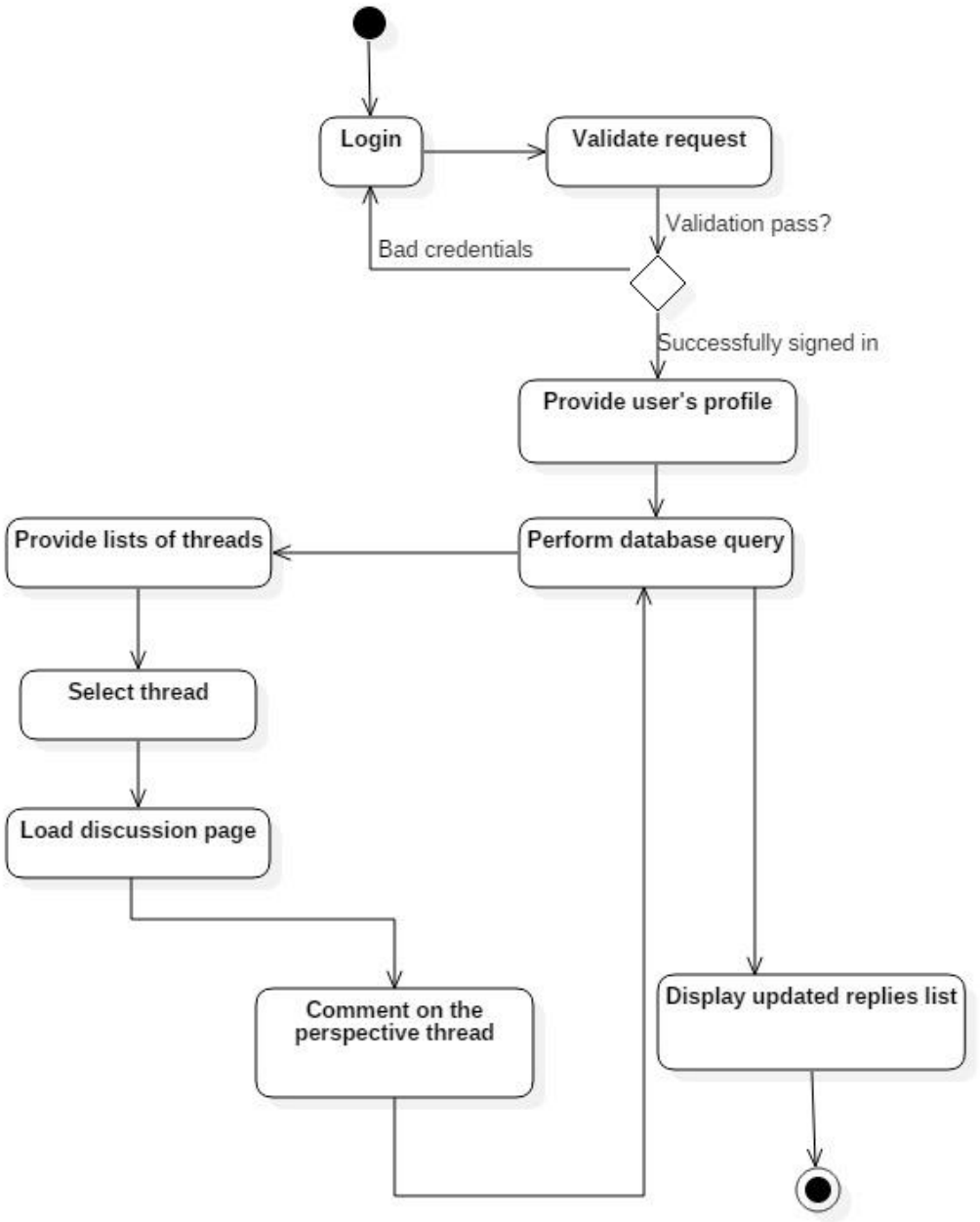


Рисунок 2.29 – Діаграма діяльності «Участь в обговоренні теми»

2.5. Розробка концептуальної моделі предметної області

На підставі вхідних і вихідних документів, а також вимог, що пред'являються до системи, можна побудувати концептуальну модель даних. Концептуальна модель даних описує сутності бази даних, зв'язку між ними, атрибути і відносини. Сутність - це унікальний неподільний ідентифікований елемент предметної області, представлений безліччю властивостей (атрибутів).

Ставлення (зв'язок) сутностей - це асоціація, встановлена між кількома сутностями, і визначає характер їх взаємозалежності.

Концептуальна (змістовна) модель - це абстрактна модель, яка визначає структуру системи, що моделюється з її зв’язками між елементами.

Концептуальна модель предметної області - це абстрактна модель, яка описує основні концепції, поняття, взаємозв'язки та правила, що характеризують предметну область. Вона використовується в інформаційних технологіях, системному аналізі та проектуванні програмного забезпечення для з'ясування потреб користувачів та визначення вимог до системи.

Концептуальна модель складається з сутностей, призначення яких наступне:

- user - містить інформацію про користувача;
- channel - містить інформацію про розподіл форуму;
- like - містить інформацію про оцінку контенту;
- reply - містить інформацію про коментарі;
- thread - містить інформацію про тему форуму;
- activity - містить інформацію про реєстровану подію.

У таблиці 2.1 наведені інформаційні зв'язки між об'єктами предметної області.

Таблиця 2.1 - Зв'язки інформаційних об'єктів

Головний об’єкт	Підлеглий об'єкт	Тип зв’язку
user	activity	1:M
channel	thread	1:M
like	user	M:1
reply	user	M:1
thread	user	M:1
activity	channel	M:1

На рис. 2.30 представлена інформаційно-логічна модель розглянутої БД, побудована відповідно до виявлених інформаційними об'єктами і зв'язками між ними.

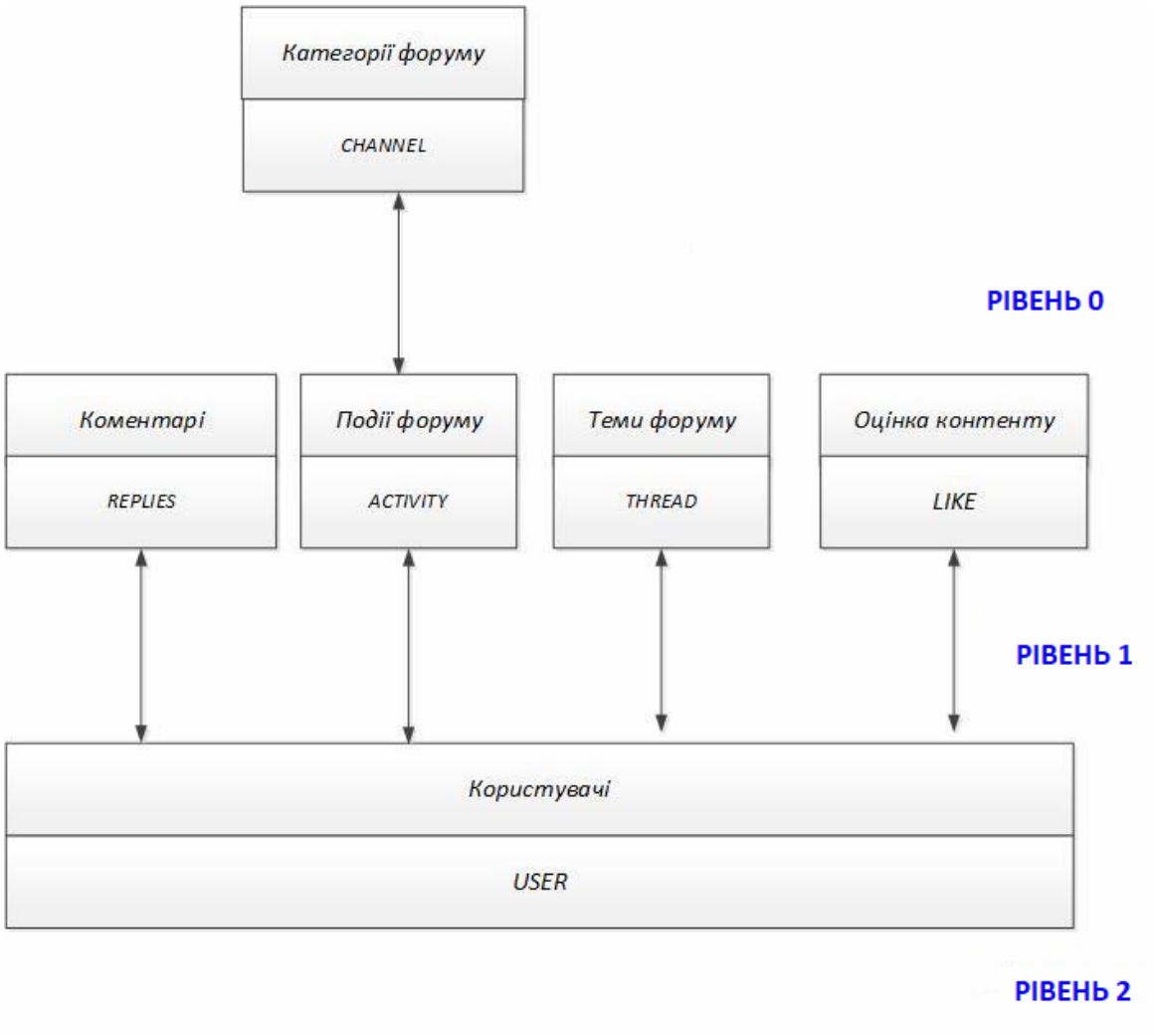


Рисунок 2.30 – Концептуальна модель предметної області

2.6. Алгоритм перевірки даних користувача під час реєстрації

Дані користувача, такі як інтернет-ім'я (нікнейм) для учасника віртуальної спільноти є своєрідною візитною карткою. Від вибору інтернет-імені залежить рівень комунікативної поведінки користувача у спільноті, авторитетність його серед інших учасників спільноти.

На рейтинг спільноти впливає також список учасників спільноти (користувачів, які пройшли реєстрацію у спільноті).

Адміністрація будь-якої віртуальної спільноти в системі правил інтернет-спільноти зазначає вимоги до формату інтернет-імені, яке надає користувач у формі реєстрації веб-спільноти.

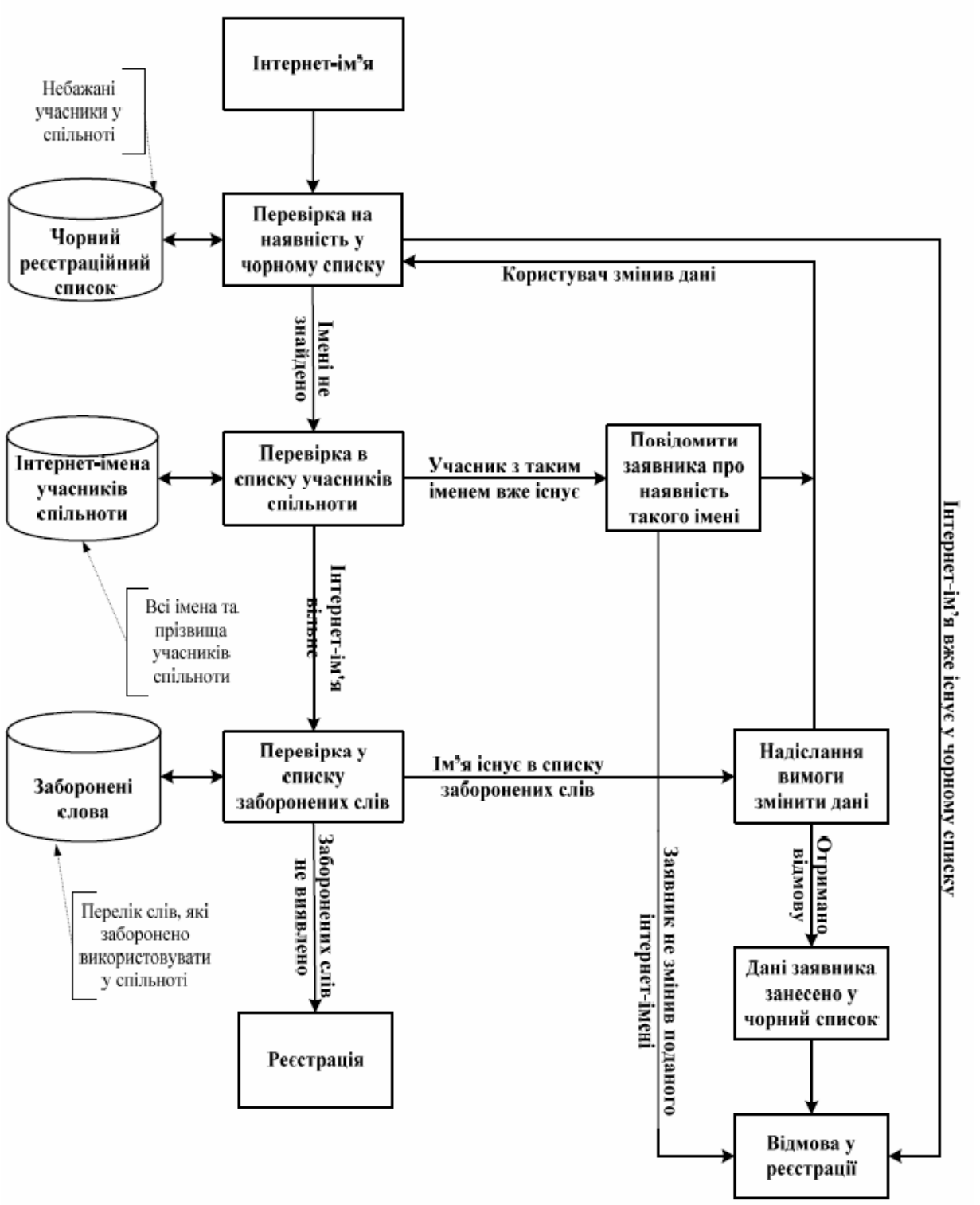


Рисунок 2.31 - Схема перевірки (валідації) нікнейму користувача при реєстрації

Зосередимо увагу на перевірці поля «username» на відповідність формату «Ім'я Прізвище», оскільки вибір такого формату інтернет-імені є найкращим для розвитку, функціонування та модерування веб-спільноти.

В основу перевірки покладено такі бази даних:

– база заборонених слів;

– чорний список заявників на реєстрацію;

– база даних імен учасників веб-спільноти.

Список заборонених слів формується адміністраторами на початку створення веб-спільноти. Прикладом заборонених слів у веб-спільнотах є ненормативна лексика, суржик, лексика, яка не має змістовного навантаження, використання фамільярної лексики, слів, які використовуються в агресивному та зневажливому стилі висловлення думок до інших учасників веб-спільноти тощо.

Чорний список заявників на реєстрацію у веб-спільноті – перелік усіх користувачів спільноти, які вибрали інтернет-ім’я, невідповідне системі правил користувачів веб-спільноти, і при валідації інтернет-імені не забажали змінити дані на коректні. Цей список створений для економії часу адміністраторів, модераторів та зменшення затрат на модерування спільнотою.

Є три етапи валідації інтернет-імені заявника на реєстрацію у веб-спільноті (рис. 2.31):

Перший етап валідації – перевірка інтернет-імені на наявність у чорному реєстраційному списку заявників веб-спільноти. Якщо дані форми реєстрації учасника веб-форуму та інтернет-ім’я не знайдено в цьому списку, то відбувається наступний етап перевірки. Коли ж дані поля «username» вже існують у чорному списку заявників на реєстрацію, то користувачу відмовлено в реєстрації в цьому веб-ресурсі.

Другий етап валідації – аналіз імені користувача форуму форматом «Ім’я Прізвище» в базі інтернет-імен форуму на наявність імені, яке перевіряємо, для уникнення дублювання імен.

Якщо інтернет-ім’я виявлено в базі імен форуму, то надсилається повідомлення заявнику з проханням змінити інтернет-ім’я або ж з уточненням факту реєстрації раніше.

Після зміни даних відбувається перевірка обновленого інтернет-імені заново. Якщо користувач не змінив поданого інтернет-імені, то йому буде відмовлено в реєстрації.

Якщо перевірка відбулась успішно та імені немає в базі імен спільноти, відбувається перевірка імені в базі заборонених слів спільноти.

На третьому етапі валідації у випадку успішної перевірки, тобто заборонених слів не виявлено у формулюванні інтернет-імені заявника, інтернет-ім’я приймається адміністратором для реєстрації і заявнику присвоюється id-учасника, оскільки інтернет-ім’я відповідає системі правил користувачів веб-спільноти.

Якщо заявник на реєстрацію не змінив некоректного інтернет-імені, адміністрація надсилає повідомлення заявнику з вимогою змінити дані. Якщо він ніяк не відреагував на вимогу або ж відмовився змінити дані, то заявника повідомляємо про відмову в реєстрації, попередньо занісши його дані в чорний список заявників на реєстрацію. Якщо користувач змінив дані на вимогу адміністрації, то відбується перевірка зміненого інтернет-імені заново.

2.7. Висновки по розділу

У цьому розділі було проведено розгляд, планування та підбір засобів та середовищ розробки веб-форуму, проведено аналіз існуючих методів, засобів і для розв’язання задачі розробки ефективного веб-форуму та його адміністрування. Також виконано моделювання засобами UML і розроблена концептуальна модель предметної області.

3. СТВОРЕННЯ СЕРВІСУ БАГАТОРІВНЕВОГО СТУПЕНЮ ДОСТУПУ НА ПРИКЛАДІ ВЕБ-ФОРУМУ ТА ЙОГО ІНТЕГРАЦІЯ З СЕРВЕРОМ

3.1. Створення функціональної частини

Для виконання даної задачі нам потрібно перш за все створити шаблонну структуру додатку (рис 3.1).

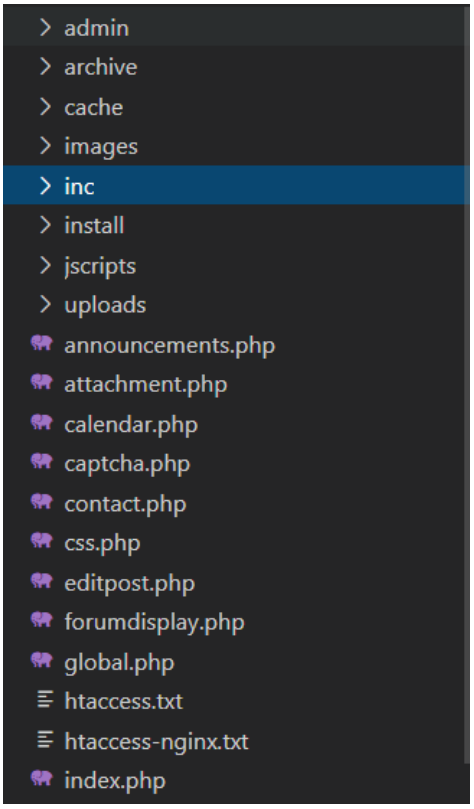


Рисунок 3.1 – Структура додатку

Створюємо ієрархічну частину для веб-форуму з 4 рівнями доступу: адміністратора, модератора, користувача і гостя.

Інтерфейс для гостя має наступний функціонал:

- Навігація.
- Допоміжні елементи інтерфейсу.
- Перегляд форумів/тем.
- Перегляд інформації про користувачів.

- Пошук в форумі.
- Перегляд учасників.
- Перегляд нових дописів.
- Віджети.
- Перегляд довідки для користувача.
- Footer.

Для того щоб мати змогу щось написати в темі форуму потрібно зареєструватися в форумі, після цього гість автоматично переходить до списку користувачів, для яких діє повністю той же функціонал, з додатковими функціями:

- Створення дописів/тем/форумів.
- Додавання інших користувачів в список друзів.
- Надсилання приватних повідомлень.
- Можливість написати і отримати відповідь від підтримки адміністрації форуму.

Наступним рівнем доступу для учасника форуму є модератор форуму. Модератори виконують роль спостерігача за форумом/темою/окремим користувачем і мають в своєму розпорядженні окрім перелічених вище можливостей ще і такі:

- Видалення/блокування користувачів.
- Видалення/редагування публікацій користувачів.
- Перегляд особистої інформації про учасника форуму.

Модератора може бути призначений тільки адміністратором.

Адміністратор в свою чергу володіє всіма передбаченими системою функціями, а саме:

- Присвоєння ролей для користувачів.
- Зміна та перегляд інформації про користувача.
- Видалення форумів.
- Редагування інформації.

- Підтвердження варифікації.

Далі розглянемо основні папки проєкту(рис. 3.2).

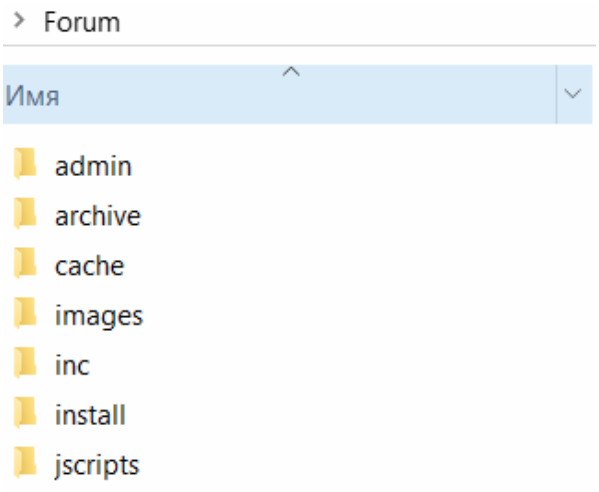


Рисунок 3.2 – Основні кореневі папки проєкту

В папці admin розташовано скрипти та модулі функціонування адмін панелі форуму. В середині розташовані наступні папки (рис. 3.3).

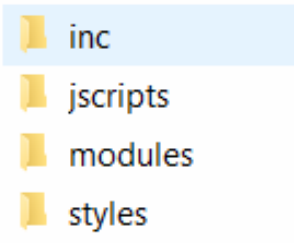


Рисунок 3.3 – Файли розміщені в папці admin

- inc – в даній папці розташовані основні класи і функції для взаємодії з основною частиною веб-форуму.

- jscipts - в даній папці розташовані всі скріпти JS, які використано в панелі адміністратора.

- modules - в даній папці реалізовано всі модулі підключення до сервера і бази даних, а також присутні файли взаємодії з цими даними.

- stules – в даній папці розміщені всі CSS файли стилів, прописаних в панелі адміністратора.

В папці archive розташовані модулі для архівування даних з форуму в декількох варіаціях, а саме в виді WinRar-архіву і варіанту друку цих даних (рис. 3.4).

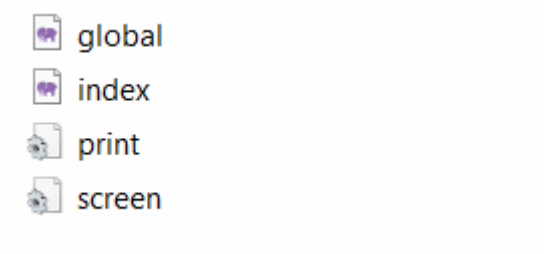


Рисунок 3.4 – Файли розміщені в папці archive

Папка cache розміщує методи реалізації кешу в форумі.

В папці images розташовані всі наявні в форумі зображення, будь то різноманітні лого і ікони, та смайли, які використовуються при спілкуванні в форумі (рис. 3.5).

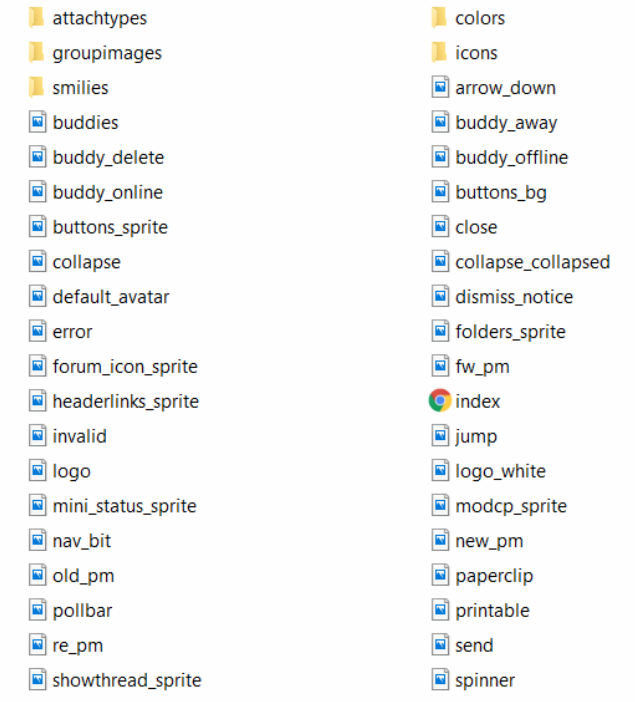


Рисунок 3.5 - Файли розміщені в папці images

Папка inc – одна із головних папок, в якій розміщуються всі основні методи реалізації форуму (рис. 3.6).

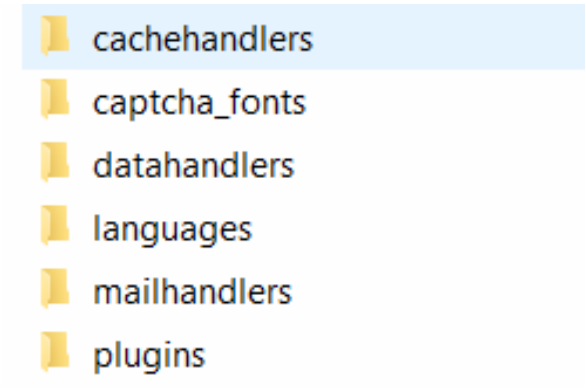


Рисунок 3.6 - Файли розміщені в папці inc.

Всередині розміщені наступні папки:

- cachehandlers – папка в якій реалізовані модулі-обробники кешу.
- captcha_fonts – папка в якій знаходяться засоби перевірки користувача captcha, для знаходження ботів.
- datahandlers – в даній папці знаходяться файли, які реалізують відправку і отримання даних із бази даних.
- Languages – це папка в якій міститься файли локалізації, таким чином можливе створення нових мовних пакетів і при цьому не потрібна повна зміна коду програми.
- Plugins - папка, в якій знаходяться архівовані файли які при встановлені форуму використовуються в процесі встановлення його на сервер.
- Mailhandlers – в даній папці знаходяться файли-опрацьовувачі надсилання повідомлень в форумі і в особистих повідомленнях.

В папці install розміщені файли потрібні для свтановлення на сервер InfynityFree (рис.3.7).

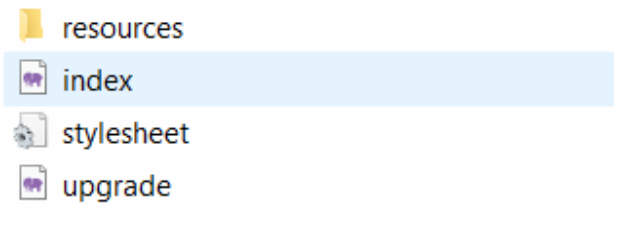


Рисунок 3.7 - Файли розміщені в папці install.

В папці jscripts розміщені всі скріпти JS та Jquery(рис.3.8).

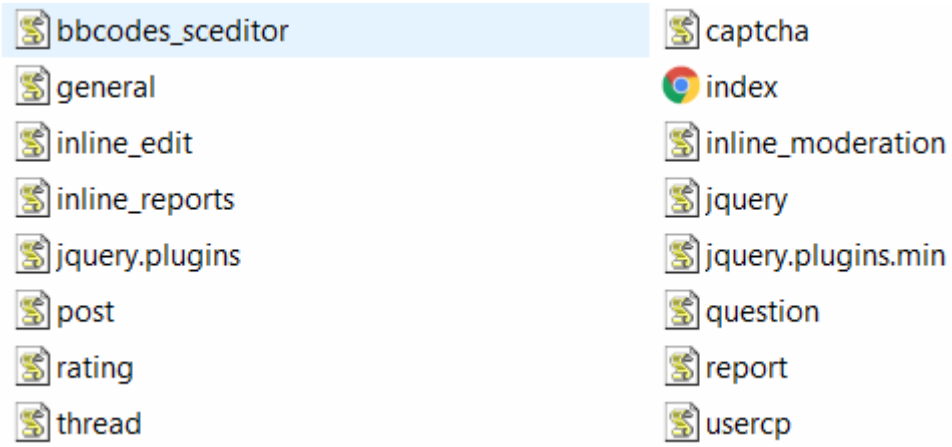


Рисунок 3.8 - Файли розміщені в папці jscripts

Таким чином проєкт повністю завершений і готовий до встановлення на сервер.

3.2. Встановлення веб-ресурсу на хостинг

Після закінчення розробки даного веб-форуму на локальному комп'ютері необхідно надіслати його на сервер для наступного відображення форуму. Для того щоб надіслати файли з робочого комп'ютера на віддалений сервер потбірно підключитись до нього за допомогою програми FileZilla FTP. Для цього отримуємо дані для заповнення з серверу (рис 3.9).

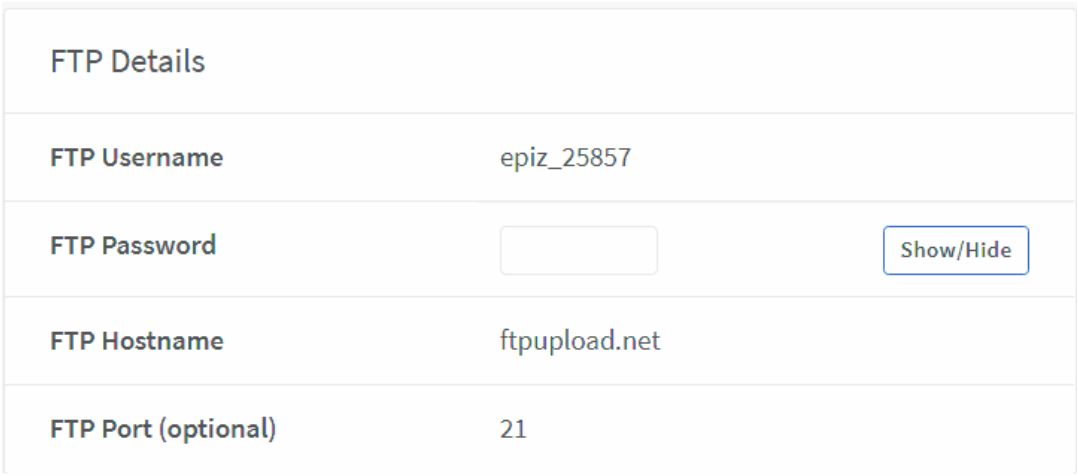


Рисунок 3.9 – Дані для підключення в FTP

Вводимо ці дані в програмі FileZilla в розділі підключення до серверу (рис 3.10).



Рисунок 3.10 – Форма для підключення до віддаленого серверу

При правильному введені даних отримаємо підключення до серверу InfynityFree. Потім переходимо в каталог серверу .htdocs і надсилаємо всі файли на сервер (рис. 3.11).

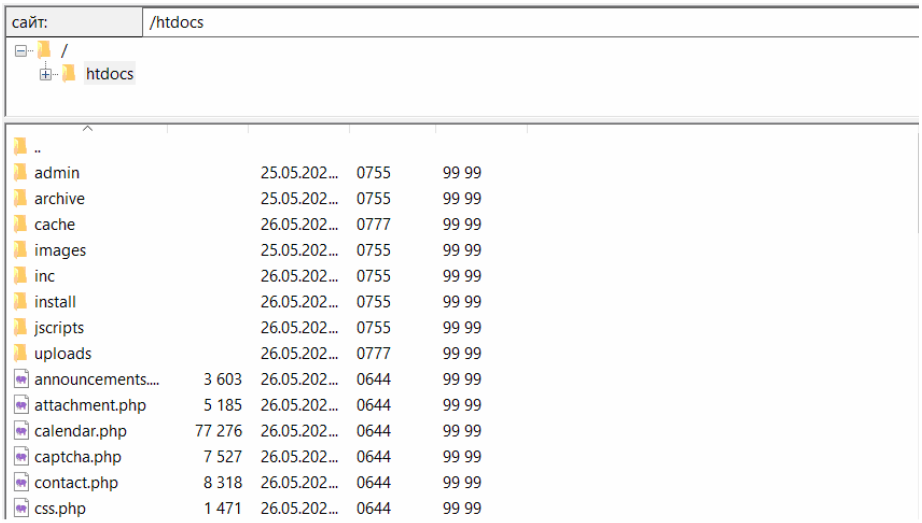


Рисунок 3.11 – Вміст серверу InfynityFree

Перевіримо чи хост прийняв дані використовуючи доступний File manager (рис. 3.12).

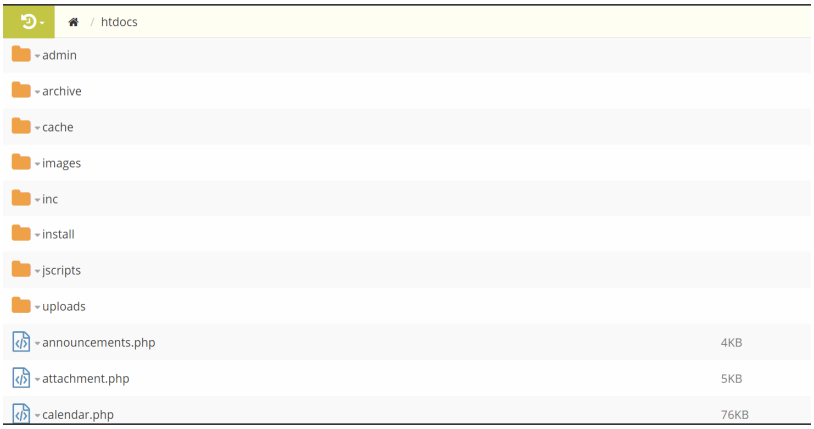


Рисунок 3.12 – Вміст на сервері переглянутий за допомогою File manager

Наступним кроком потрібно створити базу даних на сервері. Хост InfynityFree підтримує імпорт власних баз даних, а також створення нових власне на самому хості (рис. 3.13).



Рисунок 3.13 – Існуючі бази даних приєднані до домену

3.3. Реалізація основних функцій веб-сервісу

Для того щоб перейти з панелі керування хосту до власне даного форуму потрібно зробити декілька кроків:

1) Перейти на сторінку https://app.infinityfree.net/accounts.
2) В блоці Account Details знайти частину Label

3) В даній частині написано доменне ім’я форуму, в нашому випадку Website for forumfree.epizy.com.

4) Копіюємо і вставляємо forumfree.epizy.com в рядок пошуку браузера.

Головна сторінка форуму показана на рис. 3.14.

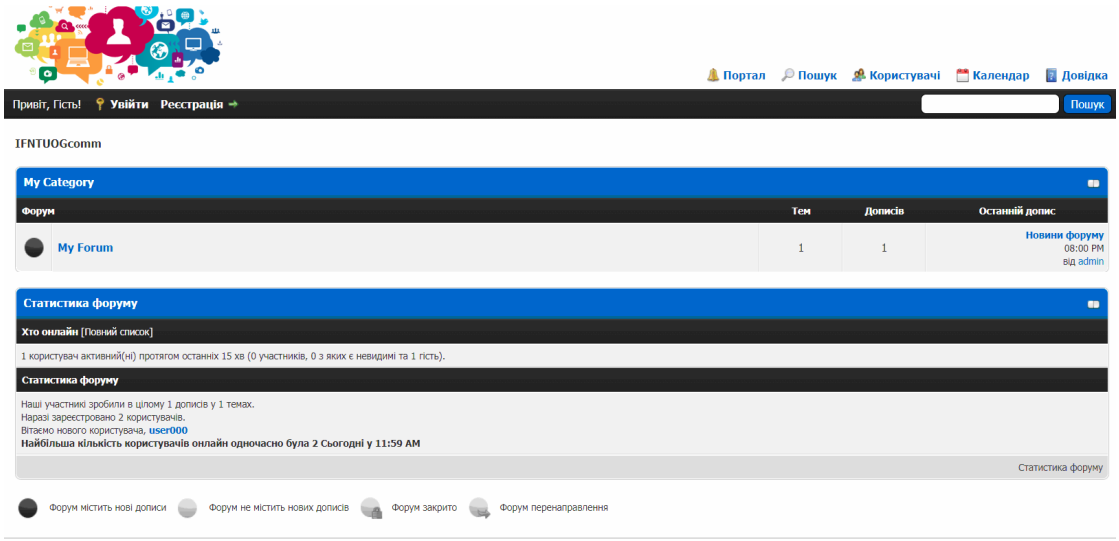


Рисунок 3.14 – Головна сторінка веб-форуму

В даному випадку ми увійшли на форум як гість. Оскільки в форумі чітка і строга ієрархія доступу в даному випадку ми можемо тільки переглядати інформацію на форумах і список користувачів (рис. 3.15 - 3.16).

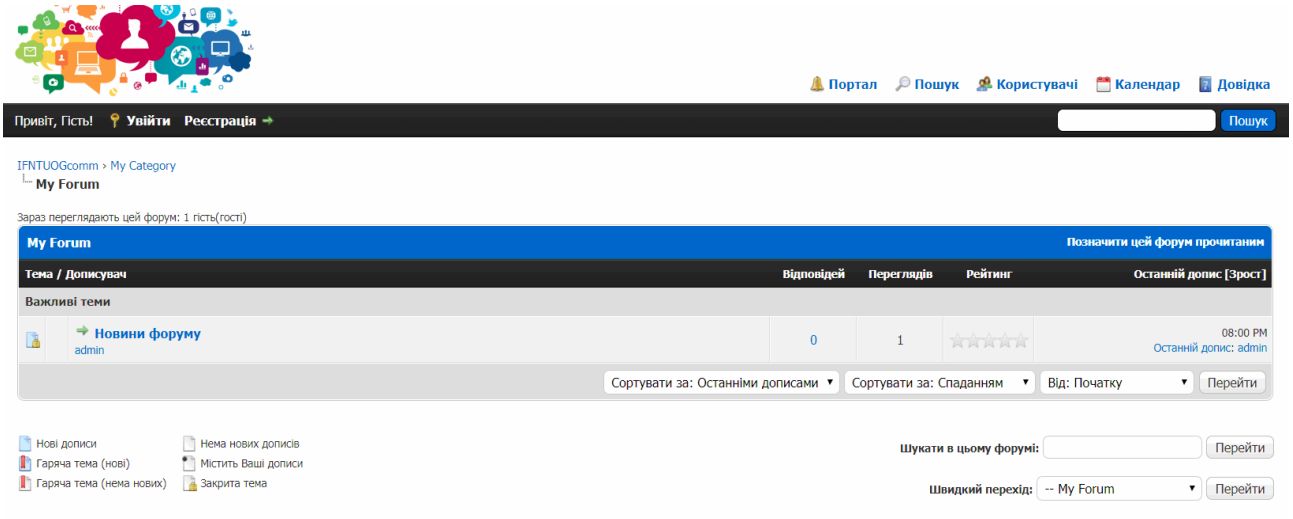


Рисунок 3.15 – Перегляд теми на ресурсі

Хоч і в гостя найвищий рівень доступу, але все одно він може переглядати всі теми на форумах, відслідковувати їх, та шукати тему на форумі.

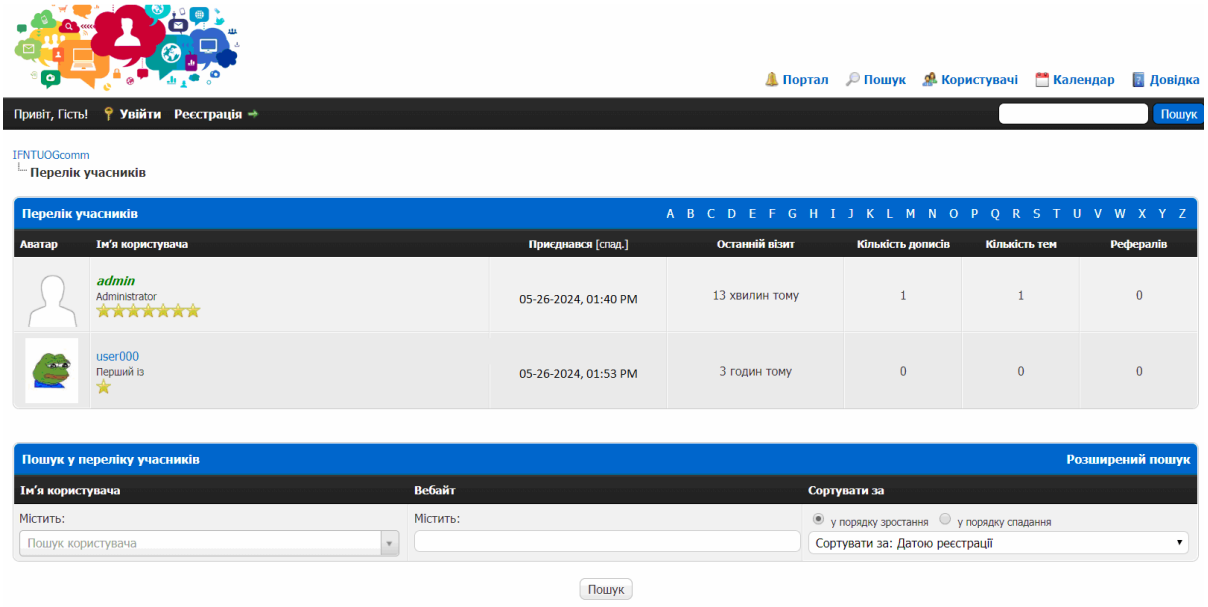


Рисунок 3.16 – Перегляд всіх користувачів на форумі

В даному випадку ми можемо не тільки переглянути список всіх користувачів, і відсортувати їх по критеріях, таких як приєднання до форуму, останньому візиту на сайт, кількості дописів, кількості тем та провести пошук по імені серед бази користувачів. Наступним кроком буде реєстрація нового користувача. Після натискання клавіші реєстрації користувач повинен підтвердити згоду з політикою користування форуму, про яку було вказано вище (рис. 3.17).

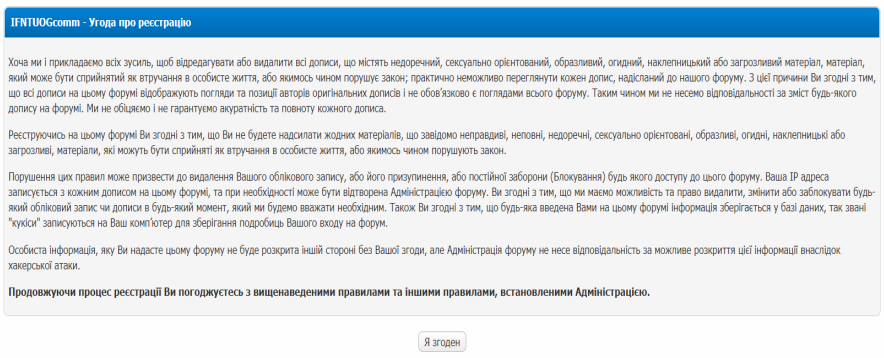


Рисунок 3.17 – Угода про реєстрацію користувача

Після погодження з політикою форуму користувачу надають змогу заповнити персональні дані для реєстрації, яка розбита на декілька блоків. Перший блок відповідає за подробиці облікового запису (рис 3.18).

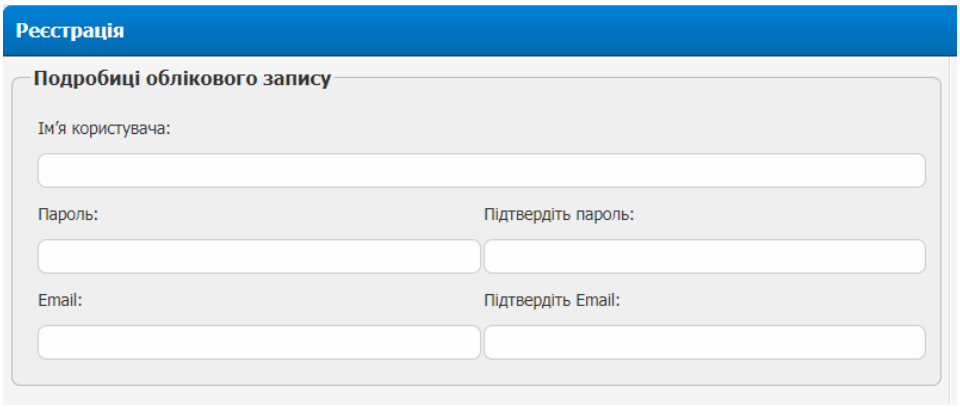


Рисунок 3.18 – Блок реєстрації облікового запису

Наступним кроком потрібно налаштувати приватність облікового запису, в якому попередньо налаштовують дозволи про отримання сповіщень від адміністрації і листів на електронну пошту (рис. 3.19).

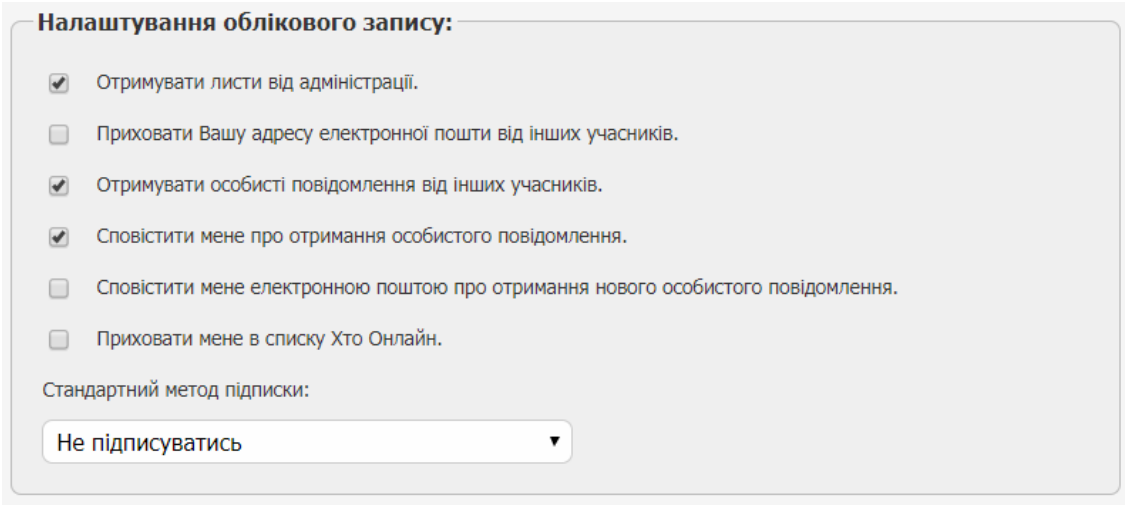


Рисунок 3.19 – Налаштування приватності користувача

Наступна група блоків існує для перевірки істинності користувача та декілька простих запитань для встановлення певних налаштувань з форумом, наприклад, встановлення мови (рис. 3.20).

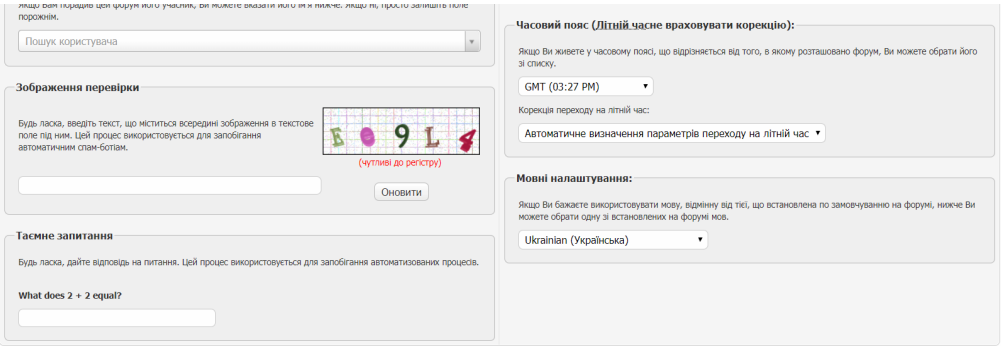


Рисунок 3.20 – Перевірка істинності користувача

При правильній реєстрації форуму відбувається перехід на головну сторінку, але користувач ще не може нічого писати і створювати, оскільки спершу потрібно активувати акаунт, за допомогою листа, який прийшов на вказану при реєстрації адресу електронної пошти, або ж активувати даного користувача через панель адміністратора (рис. 3.21).

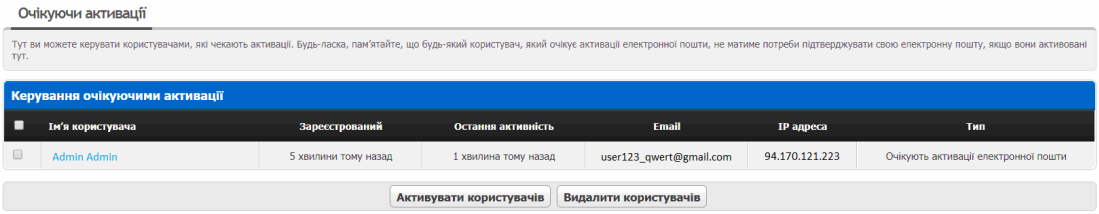


Рисунок 3.21 – Панель активації користувачів

Після активації користувача можливе повне використання можливостей форуму. Наприклад, можливе створення нових тем в форумі (рис. 3.22).

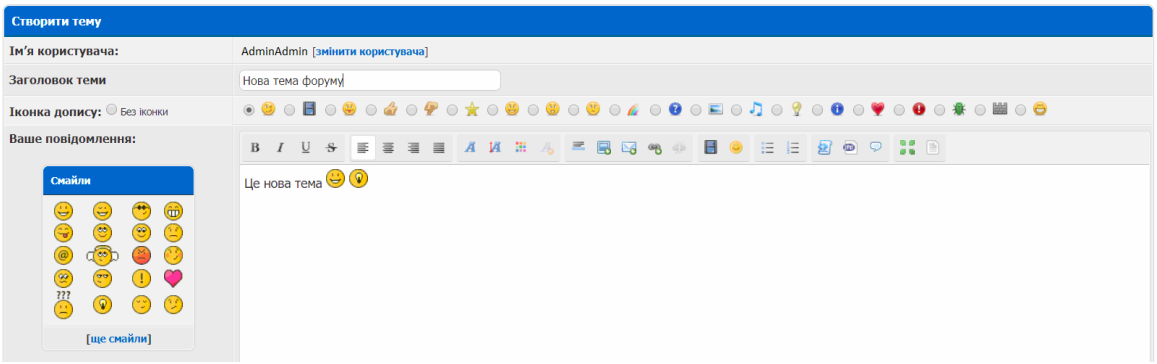


Рисунок 3.22 – Створення нової теми

Після створення нової теми, можна відповісти на неї (рис. 3.23).



Рисунок 3.23 – Форма швидкої відповіді в темі

Крім того користувач отримує змогу приєднатися до іншої теми та додати друга всередині форуму, додати його до переліку ігнорування або ж поскаржитись на користувача, цей блок знаходиться під його профілем (рис. 3.24).

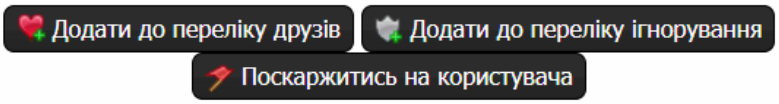


Рисунок 3.24 – перелік дій з списком до інших користувачів.

В разі отримання запиту на дружбу від іншого користувача, приходить системне сповіщення і в списку на прийняття запитів відображається наступне повідомлення (рис. 3.25).

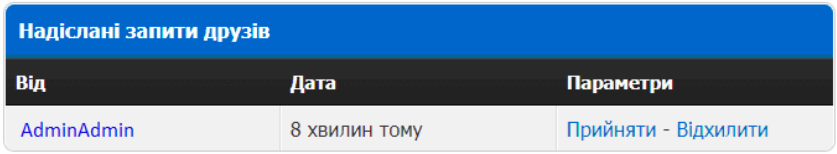


Рисунок 3.25 – Прийняття запиту на дружбу

Коли запит прийнятий, першому користувачу приходить системне повідомлення і в налаштуваннях відображається список друзів (рис. 3.26).

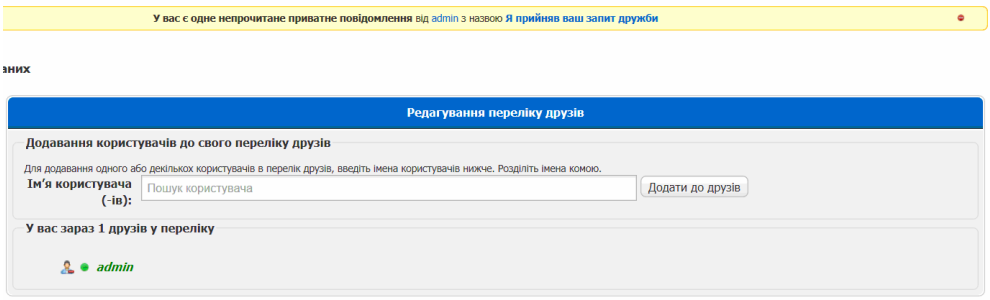


Рисунок 3.26 – Відображення списку друзів

Крім того в налаштуваннях користувач може змінювати і редагувати інформацію про нього, а також переглядати списки своїх приватних повідомлень, списки груп, на які підписаний користувач, керувати вкладеннями, переглядати чернетки, підписані теми та форуми (рис. 3.27).

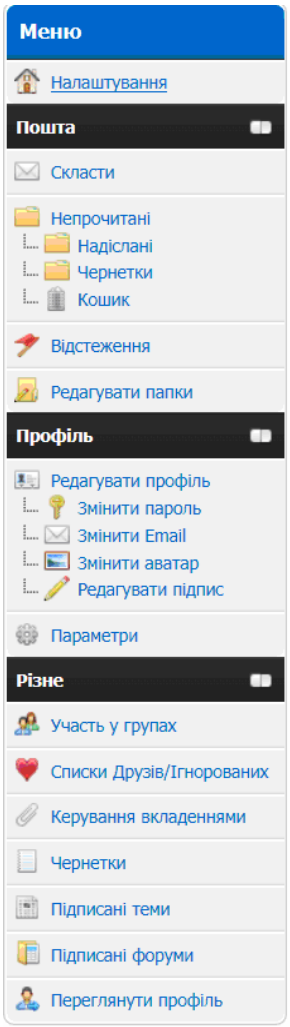


Рисунок 3.27 – Меню налаштувань користувача

При створенні особистого повідомлення воно буде відображатися тільки тому хто надіслав його і отримувачу. Переглянути список всіх приватних повідомлень можна в спеціальній вкладці в налаштуваннях. При написанні такого повідомлення можна вибрати до 5 отримувачів, також можна обирати як між своїми друзями, так і з списку всіх користувачів форуму (рис. 3.28).

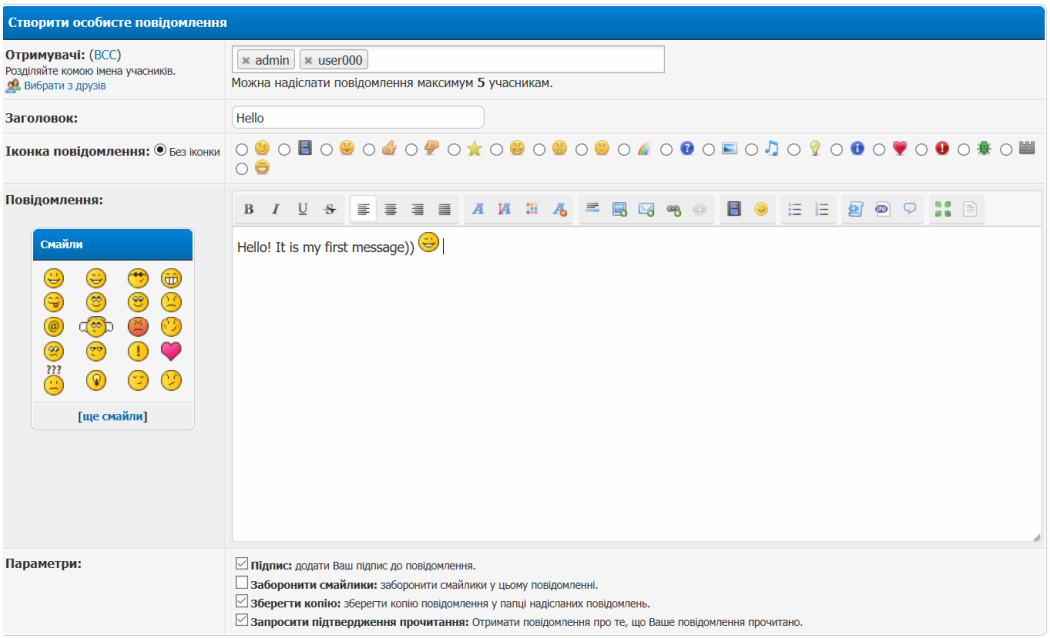


Рисунок 3.28 – Надсилання приватного повідомлення

Але окрім користувача в форумі необхідний і модератор і при тому модератор може бути призначений тільки адміністратором форуму і повинен відповідати за дотримання внутрішніх правил форуму в темах. Такі люди відповідають за певну частину форуму і моніторять його протягом певного часу. Модератор може перейти на його сторінку модерації, клацнувши на відповідну кнопку в заголовку сайту (рис. 3.29).

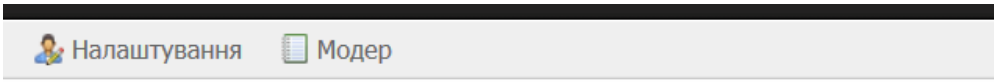


Рисунок 3.29 – Кнопка переходу до панелі модератора

В даній панелі можна переглянути список всіх забанених користувачів, а також написати нотатки для всіх модераторів, перейти в редактор профілю, переглянути вкладку блокування, журнали попереджень, та інше (рис. 3.30).

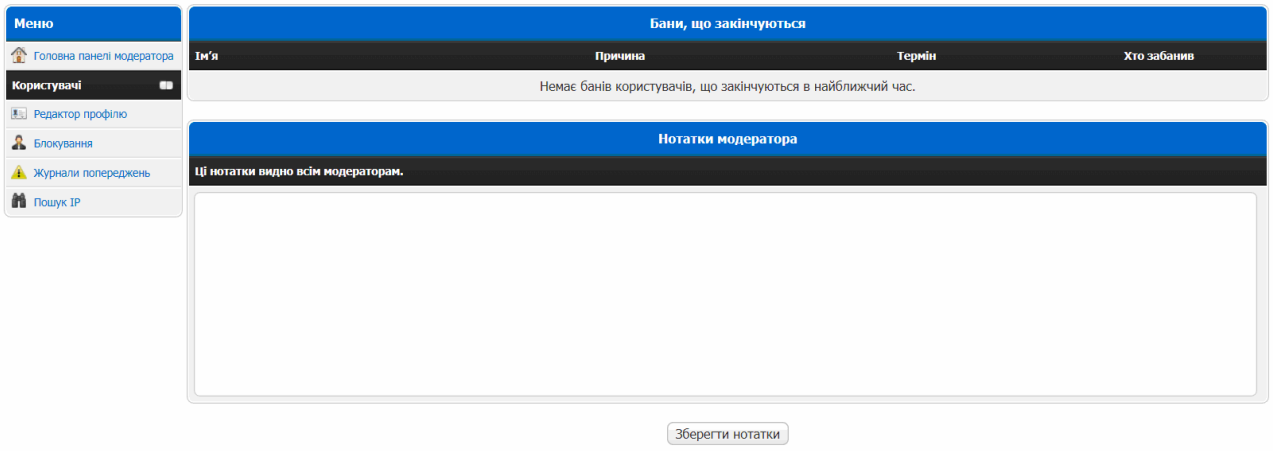


Рисунок 3.30 – Головна сторінка модератора

Також модератор може переглядати і змінювати інформацію про користувача, без його відома, але модератор може змінити тільки необов'язкові поля, такі як дату народження, титул учасника на форумі, змінювати певні контакти, чи редагувати інформацію про відсутність (рис. 3.31).

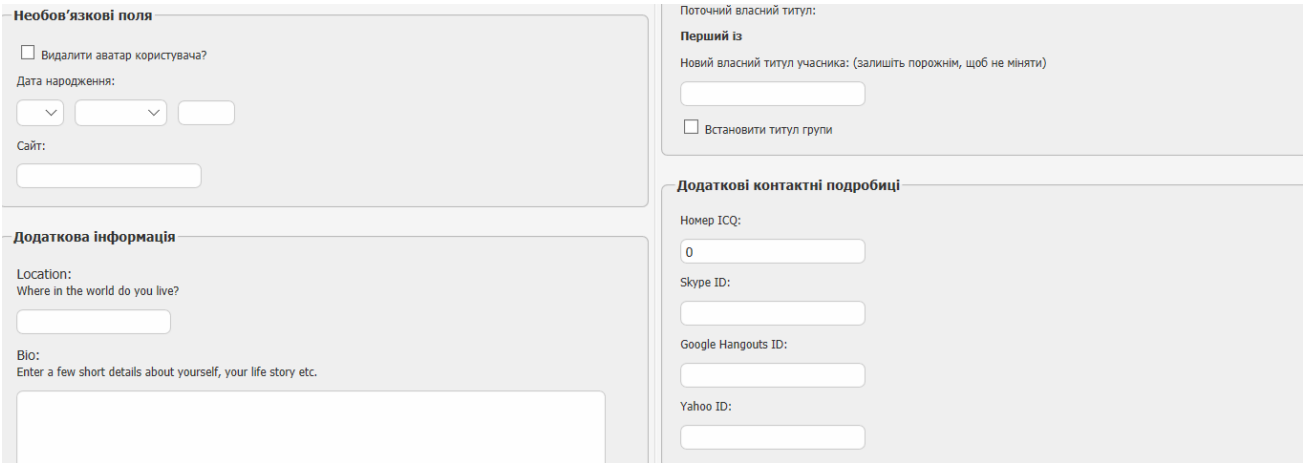


Рисунок 3.31 – форма редагування інформації про користувача

Крім перегляду і редагування інформації модератор також може заблокувати (забанити) користувача, який порушив правила форуму (рис. 3.32).

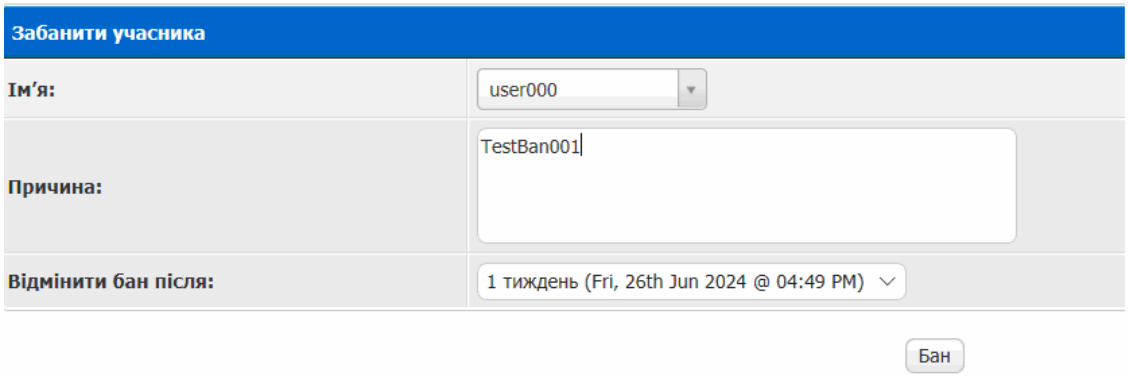


Рисунок 3.32 – Форма бану користувача

В даному випадку потрібно ввести ім'я користувача, причину його бану і термін дії блокування. Під час дії бану, користувач може переглядати всі теми в форумі, приймати і писати особисті повідомлення, але не може створити тему, або ж писати в існуючій.

В разі потреби бану іншого модератора, потрібно повідомити адміністратору, через функцію оформлення скарги на користувача.

Адміністратор – особа яка володіє абсолютними правами в форумі, вона може здійснювати видалення всіх існуючих форумів, тем, користувачів та модераторів, також вона має права модератора і може здійснювати контроль за їх діями. На головній сторінці форуму адміністратор володіє тим же функціоналом, що і інші користувачі (рис. 3.33).

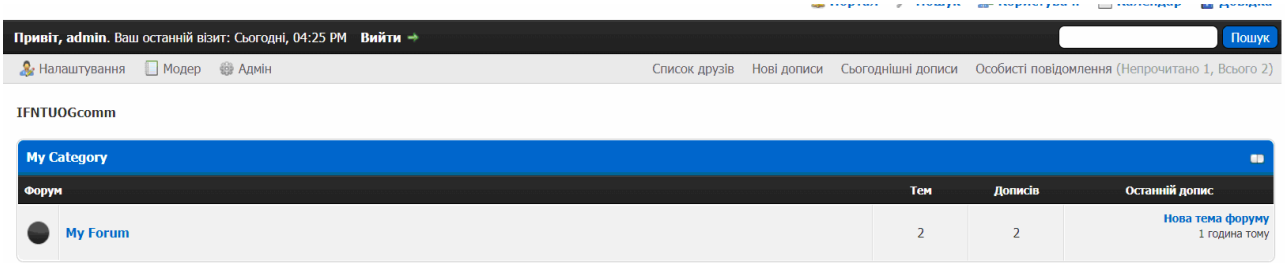


Рисунок 3.33 – головна сторінка форуму адміністратора

Проте реальні можливості адміністратора можна побачити на його панелі управління (рис. 3.34).

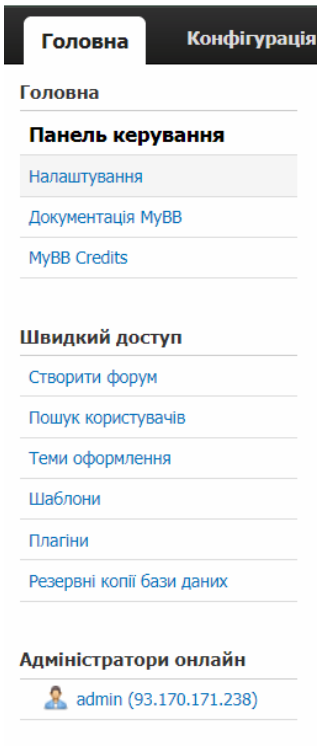


Рисунок 3.34 – Меню панелі управління адміністратора

В даному меню адмін може налаштовувати форум, вести документацію, а також перейти до швидкого доступу до корисних властивостей форуму, наприклад, хоч тему в форумі користувач може створити будь яку, тільки адміністратор може створити новий форум, розробляти нові теми для форуму, шаблони, встановлювати існуючі плагіни, наприклад, з WordPress та створювати резервні копії бази даних.

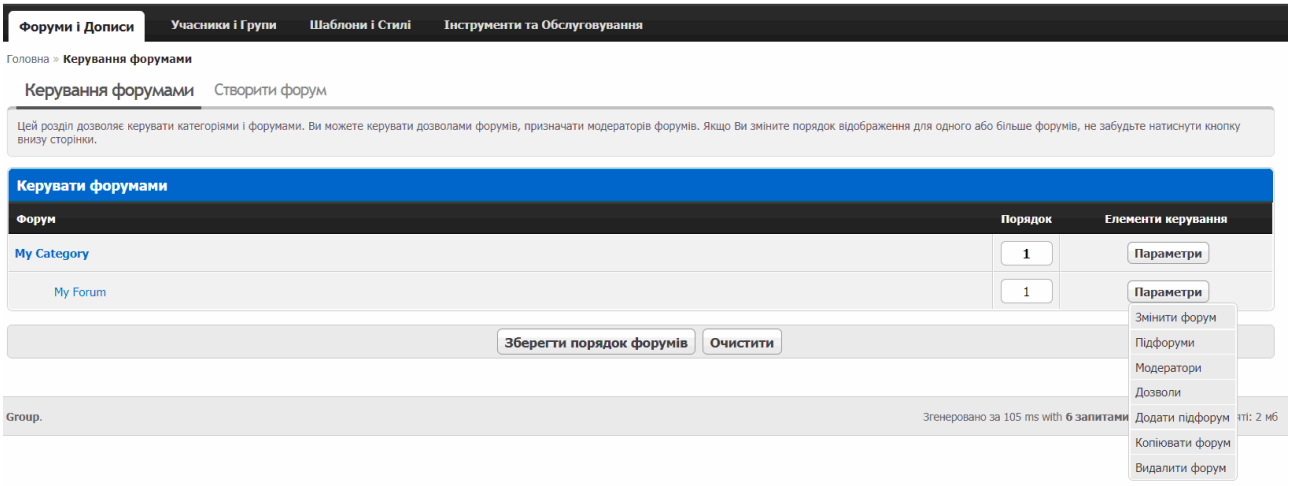


Рисунок 3.35 – Керування форумами

Також адміністратор може швидко керувати всіма форумами, переглядати інформацію, наявних модераторів форуму, та змінювати інформацію про форум, його дозволи, копіювати, додати підфорум або ж видалити його (рис. 3.35).

Крім керування форумами, адміністратор також керує всіма користувачами і модераторами (рис. 3.36).

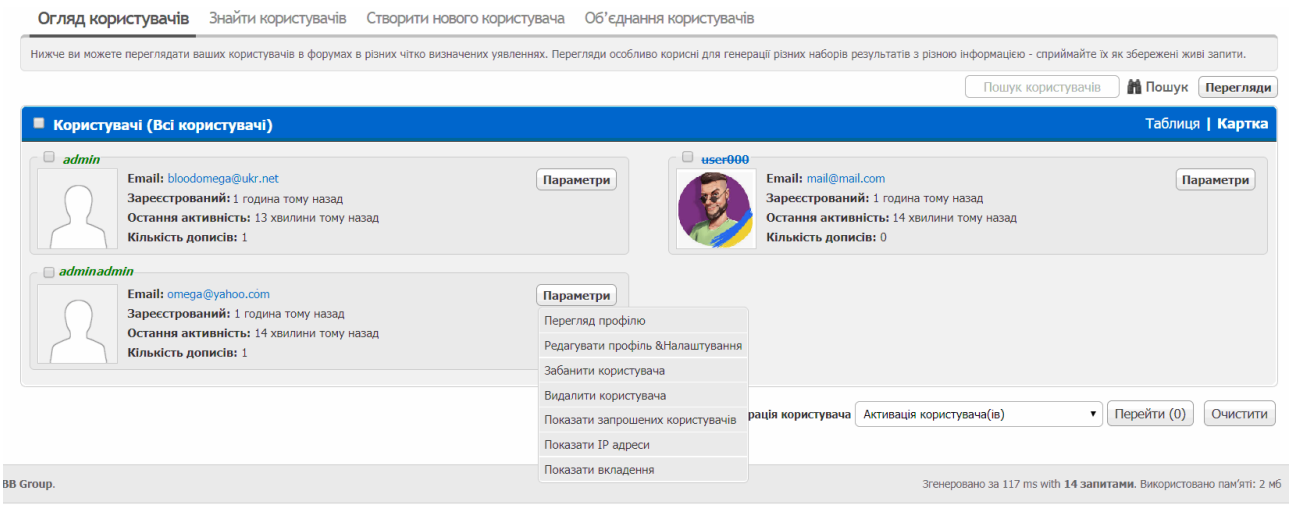


Рисунок 3.36 – Керування користувачами

Можливості надані при керування користувачами дещо більші, наприклад, адміністратор може об'єднати 2 аккаунти в один або ж створити нового користувача. Також адміністратор також через параметри керувати всіма існуючими учасниками форуму, а саме: редагувати профіль, переглядати його, забанити, видаляти та показувати його IP-адресу.

Також адміністратор може переглядати стан системи(рис. 3.37).

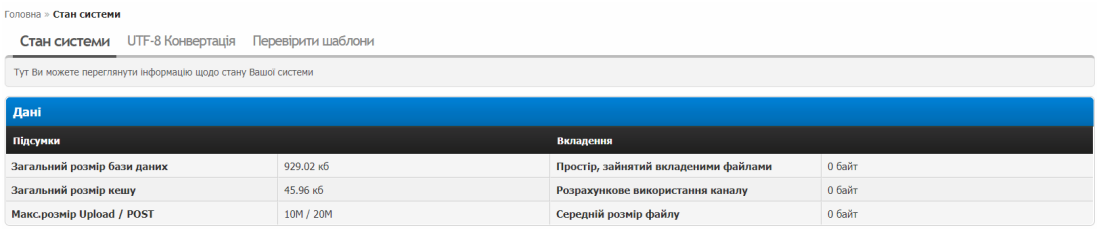


Рисунок 3.37 – Стан роботи форуму

Отож в панелі адміністратора реалізовано функції контролю, моніторингу та зміни форуму, його учасників та роботи системи в загальному.

3.4. Висновки до розділу

В даному розділі було підібрано і реалізовано систему веб-форуму на основі віддаленого серверу з реалізацією багаторівневого ступеню доступу та системою моніторигну даних. За допомогою сторонніх програм відбулося завантаження даних на віддалений сервер. Даний веб-ресурс повністю робочий, та відповідає поставленим вимогам.

ВИСНОВКИ

Під час виконання дипломної роботу було виконано розробку веб-орієнтованого сервісу з реалізацією багаторівневого ступеню доступу та системою моніторингу даних на прикладі побудови веб-форуму.

В ході роботи було проведено аналіз вихідних даних і вимог до системи, на основі якого в подальшому будувався вибір програмного забезпечення системи. Описані підходи та алгоритми проектування сховищ даних, наведена концепція узагальненого гібридного сховища даних, яка дасть змогу проектувати сховище даних з урахуванням запитів, що надходять до сховища даних, тим самим забезпечуючи загальний інтерфейс доступу до інформації, незалежно від фізичної бази даних.

Були реалізовані наступні завдання:

- проаналізовано предметна область і визначені основні поняття, об'єкти і процеси;

- виконано аналіз існуючих веб-форумів: визначено їх функціональні можливості, умови використання і основні характеристики;

- сформульовані основні функціональні можливості і вимоги у вигляді постановки завдання;

- виконано моделювання об'єкта проектування засобами уніфікованої мови моделювання UML;

- побудована концептуальна модель предметної області;

- розроблена програма частина сервісу багаторівневого ступеню доступу та системою моніторингу у вигляді веб-форуму.

Також в роботі представлено метод валідації інтернет-імені користувачів віртуальної спільноти та проаналізовано вимоги до інтернет-імен власників веб-спільнот.

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Kravets. R. Typical ways of web-communities development / R. Kravets, A.M. Peleschyshyn, Yu. Syerov // Proceedings of the International Conference on Computer Science and Information Technologies, CSIT`2006, September 28th-30th, Lviv, Ukraine, p. 56–58.

2. Сєров Ю.О. Моделювання поведінки та класифікація учасників Веб-спільнот на основі нечітких множин // Вісник Нац. ун-ту «Львівська політехніка». – 2008. – № 610 – С. 218–228.

3. Peleschyshyn A. Typical ways of webcommunities development / A. Peleschyshyn, Yu. Syerov // Proceedings of the International Conference on Computer Science and Information Technologies, CSIT`2006, September 28th-30th, Lviv, Ukraine, p.56–58.

4. Пасічник В.В. Аналіз критеріїв ефективності Веб-спільнот на основі форумів / В.В. Пасічник, Р.Б. Кравець, Ю.О. Сєров // Матеріали 12-го форуму, 1-3 квітня 2022 р. Ч. 2, с. 493, Харків.

5. R. Kravets Information System of Web Community Members Behavior Analyzing and Classifying / R. Kravets, Yu. Syerov// Proceedings of the international conference on computer science and information technologies (CSIT’ 2008), September 25th-27th, Ukraine, Lviv, 2008. – p. 41–44.

6. Claudia Imhoff.Corporate information factory / Claudia Imhoff, Ryan Sousa – John Wiley & Sons, 2001 – 382 p.

7. Douglas Hackney. Architectures and Approaches for Successful Data Warehouses. / Douglas Hackney – www.egltd.com/presents/Architectures Approaches.pdf.

8. Goran Velinov. Framework for Generalization and Improvement of Relational Data, IJCSNS International Journal of Computer Science and Network Security / Goran Velinov , Danilo Gligoroski, Margita Kon Popovska, VOL.8 No.3, March 2008.

9. Hugh J. Watson. Data Warehouse Architectures: Factors in the Selection Decision and the Success of the Architectures [Електронний ресурс] / Hugh J. Watson, Thilini Ariyachandra 2003 – Режим доступу: http://www.terry.uga.edu/~hwatson/DW_Architecture_Report.pdf. 5. Ralph Kimball.

10. The data warehouse toolkit: the complete guide to dimensional modeling / Ralph Kimball – Wiley, 2002 – 436 p.

11. W. H. Inmon. Corporate Information Factory Components [Електронний ресурс] / W. H. Inmon – Inmon Data Systems – Режим доступу : http://www.inmoncif.com/view/26.

12. Wayne Eckerson. Four Ways to Build a Data Warehouse [Режим доступу] // Wayne Eckerson – TDWI -2003 – Режим доступу http://www.tdwi.org/research/display.aspx?ID=6699

13. Wen-Yang Lin. A Genetic Selection Algorithm for OLAP Data Cubes// Wen-Yang Lin,I-Chung Kuo – Knowledge and information systems, 2004, vol. 6.

14. Tapio Niemi. Constructing OLAP Cubes Based on Queries. Proceedings of the 4th ACM international workshop on Data warehousing and OLAP / Tapio Niemi, Jyrki Nummenmaa, Peter Thanisch ,2001.

15. Шаховська Н.Б. Сховища та простори даних: Монографія / Н.Б. Шаховська, В.В. Пасічник. – Львів: Вид-во Нац. ун-ту «Львівська політехніка» , 2009. – 244 с.

16. Яцишин А.Ю. Проектування гібридного сховища даних для роботи з державним бюджетом України. II Всеукраїнська науково-практична конференція «Інформаційні технології та автоматизація – 2009» / А.Ю. Яцишин. – Одеса, 2009.

ДОДАТКИ

Додаток А

forumdisplay.php

```
$templatelist = "forumdisplay,forumdisplay_thread,forumbit_depth1_cat,fo
rumbit_depth2_cat,forumbit_depth2_forum,forumdisplay_subforums,forumdisp
lay_threadlist,forumdisplay_moderatedby,forumdisplay_searchforum,forumdi
splay_thread_rating,forumdisplay_threadlist_rating";
$templatelist .= ",forumbit_depth1_forum_lastpost,forumdisplay_thread_mu
ltipage_page,forumdisplay_thread_multipage,forumdisplay_thread_multipage
_more,forumdisplay_thread_gotounread,forumbit_depth2_forum_lastpost,foru
mdisplay_rules_link,forumdisplay_orderarrow,forumdisplay_newthread";
$templatelist .= ",multipage,multipage_breadcrumb,multipage_end,multipag
e_jump_page,multipage_nextpage,multipage_page,multipage_page_current,mul
tipage_page_link_current,multipage_prevpage,multipage_start,forumdisplay
_thread_unapproved_posts,forumdisplay_nothreads";
$templatelist .= ",forumjump_advanced,forumjump_special,forumjump_bit,fo
rumdisplay_password_wrongpass,forumdisplay_password,forumdisplay_inlinem
oderation_custom_tool,forumbit_subforums,forumbit_moderators,forumbit_de
pth2_forum_lastpost_never,forumbit_depth2_forum_lastpost_hidden";
$templatelist .= ",forumdisplay_usersbrowsing_user,forumdisplay_usersbro
wsing,forumdisplay_inlinemoderation,forumdisplay_thread_modbit,forumdisp
lay_inlinemoderation_col,forumdisplay_inlinemoderation_selectall,forumdi
splay_threadlist_clearpass,forumdisplay_thread_rating_moved";
$templatelist .= ",forumdisplay_announcements_announcement,forumdisplay_
announcements,forumdisplay_threads_sep,forumbit_depth3_statusicon,forumb
it_depth3,forumdisplay_sticky_sep,forumdisplay_thread_attachment_count,f
orumdisplay_rssdiscovery,forumbit_moderators_group";
$templatelist .= ",forumdisplay_inlinemoderation_openclose,forumdisplay_
inlinemoderation_stickunstick,forumdisplay_inlinemoderation_softdelete,f
orumdisplay_inlinemoderation_restore,forumdisplay_inlinemoderation_delet
e,forumdisplay_inlinemoderation_manage,forumdisplay_nopermission";
$templatelist .= ",forumbit_depth2_forum_unapproved_posts,forumbit_depth
2_forum_unapproved_threads,forumbit_moderators_user,forumdisplay_inlinem
oderation_standard,forumdisplay_threadlist_prefixes_prefix,forumdisplay_
threadlist_prefixes,forumdisplay_thread_icon,forumdisplay_rules";
$templatelist .= ",forumdisplay_thread_deleted,forumdisplay_announcement
s_announcement_modbit,forumbit_depth2_forum_viewers,forumdisplay_threadl
ist_sortrating,forumdisplay_inlinemoderation_custom,forumdisplay_announc
ement_rating,forumdisplay_inlinemoderation_approveunapprove,forumdisplay
_threadlist_subscription";
require_once "./global.php";
require_once MYBB_ROOT."inc/functions_post.php";
require_once MYBB_ROOT."inc/functions_forumlist.php";
require_once MYBB_ROOT."inc/class_parser.php";
$parser = new postParser;
$orderarrow = $sortsel = array('rating' => '', 'subject' => '', 'starter
' => '', 'started' => '', 'replies' => '', 'views' => '', 'lastpost' =>
'');
$ordersel = array('asc' => '', 'desc' => '');
$datecutsel = array(1 => '', 5 => '', 10 => '', 20 => '', 50 => '', 75 =
> '', 100 => '', 365 => '', 9999 => '');
$rules = '';
// Load global language phrases
$lang->load("forumdisplay");
```

```
$plugins->run_hooks("forumdisplay_start");
$fid = $mybb->get_input('fid', MyBB::INPUT_INT);
if($fid < 0)
{
    switch($fid)
    {
        case "-1":
            $location = "index.php";
            break;

        case "-2":
            $location = "search.php";
            break;
        case "-3":
            $location = "usercp.php";
            break;
        case "-4":
            $location = "private.php";
            break;
        case "-5":
            $location = "online.php";
            break;
    }
    if($location)
    {
        header("Location: ".$location);
        exit;
    }
}
// Get forum info
$foruminfo = get_forum($fid);
if(!$foruminfo)
{
    error($lang->error_invalidforum);
}
$archive_url = build_archive_link("forum", $fid);
$currentitem = $fid;
build_forum_breadcrumb($fid);
$parentlist = $foruminfo['parentlist'];
// To validate, turn & to &amp;amp; but support unicode
$foruminfo['name'] = preg_replace("#&(?!\#[0-
9]+;)#si", "&amp;amp;", $foruminfo['name']);
$forumpermissions = forum_permissions();
$fpermissions = $forumpermissions[$fid];

if($fpermissions['canview'] != 1)
{
    error_no_permission();
}

if($mybb->user['uid'] == 0)
{
    // Cookie'd forum read time
    $forumsread = array();
    if(isset($mybb->cookies['mybb']['forumread']))
    {
```

```
        $forumsread = my_unserialize($mybb->cookies['mybb']['forumread']);
    }
    if(is_array($forumsread) && empty($forumsread))
    {
        if(isset($mybb->cookies['mybb']['readallforums']))
        {
            $forumsread[$fid] = $mybb->cookies['mybb']['lastvisit'];
 }
        else
        {
            $forumsread = array();
        }
    }

    $query = $db->simple_select("forums", "*", "active != 0", array("order_by" => "pid, disporder"));
}
else
{
    // Build a forum cache from the database
    $query = $db->query("
        SELECT f.*, fr.dateline AS lastread
        FROM ".TABLE_PREFIX."forums f
        LEFT JOIN ".TABLE_PREFIX."forumsread fr ON (fr.fid=f.fid AND fr.uid='{$mybb->user['uid']}')
        WHERE f.active != 0
        ORDER BY pid, disporder
    ");
}
while($forum = $db->fetch_array($query))
{
    if($mybb->user['uid'] == 0 && isset($forumsread[$forum['fid']]))
    {
        $forum['lastread'] = $forumsread[$forum['fid']];
    }
$fcache[$forum['pid']][$forum['disporder']][$forum['fid']] = $forum;
}
// Get the forum moderators if the setting is enabled.
if($mybb->settings['modlist'] != 0)
{
    $moderatorcache = $cache->read("moderators");
}
$bgcolor = "trow1";
if($mybb->settings['subforumsindex'] != 0)
{
    $showdepth = 3;
}
else
{
    $showdepth = 2;
}
$subforums = '';
$child_forums = build_forumbits($fid, 2);
$forums = $child_forums['forum_list'];
if($forums)
```

```
{
    $lang->sub_forums_in = $lang->sprintf($lang-
>sub_forums_in, $foruminfo['name']);
    eval("\$subforums = \"".$templates-
>get("forumdisplay_subforums")."\";");
}
$excols = "forumdisplay";
// Password protected forums
check_forum_password($foruminfo['fid']);

if($foruminfo['linkto'])
{
    header("Location: {$foruminfo['linkto']}");
    exit;
}

// Make forum jump...
if($mybb->settings['enableforumjump'] != 0)
{
    $forumjump = build_forum_jump("", $fid, 1);
}
if($foruminfo['type'] == "f" && $foruminfo['open'] != 0 && $fpermissions
['canpostthreads'] != 0 && $mybb->user['suspendposting'] == 0)
{
    eval("\$newthread = \"".$templates-
>get("forumdisplay_newthread")."\";");
}
if($fpermissions['cansearch'] != 0 && $foruminfo['type'] == "f")
{
    eval("\$searchforum = \"".$templates-
>get("forumdisplay_searchforum")."\";");
}
// Gather forum stats
$has_announcements = $has_modtools = false;
$forum_stats = $cache->read("forumsdisplay");
if(is_array($forum_stats))
{
    if(!empty($forum_stats[-
1]['modtools']) || !empty($forum_stats[$fid]['modtools']))
    {
        // Mod tools are specific to forums, not parents
        $has_modtools = true;
    }

    if(!empty($forum_stats[-
1]['announcements']) || !empty($forum_stats[$fid]['announcements']))
    {
        // Global or forum-specific announcements
        $has_announcements = true;
    }
}
$done_moderators = array(
    "users" => array(),
    "groups" => array()
);
$moderators = '';
$parentlistexploded = explode(",", $parentlist);
```

```
foreach($parentlistexploded as $mfid)
{
 // This forum has moderators
    if(is_array($moderatorcache[$mfid]))
    {
        // Fetch each moderator from the cache and format it, appending
it to the list
        foreach($moderatorcache[$mfid] as $modtype)
        {
            foreach($modtype as $moderator)
            {
                if($moderator['isgroup'])
                {
                    if(in_array($moderator['id'], $done_moderators['grou
ps']))
                    {
                        continue;
                    }
                    $moderator['title'] = htmlspecialchars_uni($moderato
r['title']);

                    eval("\$moderators .= \"".$templates-
>get("forumbit_moderators_group", 1, 0)."\";");
                    $done_moderators['groups'][] = $moderator['id'];
                }
                else
                {
                    if(in_array($moderator['id'], $done_moderators['user
s']))
                    {
                        continue;
                    }

                    $moderator['profilelink'] = get_profile_link($modera
tor['id']);
                    $moderator['username'] = format_name(htmlspecialchar
s_uni($moderator['username']), $moderator['usergroup'], $moderator['disp
laygroup']);
                    eval("\$moderators .= \"".$templates-
>get("forumbit_moderators_user", 1, 0)."\";");
                    $done_moderators['users'][] = $moderator['id'];
                }
                $comma = $lang->comma;
            }
        }
    }

    if(!empty($forum_stats[$mfid]['announcements']))
    {
        $has_announcements = true;
    }
}
$comma = '';
// If we have a moderators list, load the template
if($moderators)
{
```

```
    eval("\$moderatedby = \"".$templates-
>get("forumdisplay_moderatedby")."\";");
}
else
{
    $moderatedby = '';
}
// Get the users browsing this forum.
if($mybb->settings['browsingthisforum'] != 0)
{
    $timecut = TIME_NOW - $mybb->settings['wolcutoff'];
    $comma = '';
    $guestcount = 0;
    $membercount = 0;
    $inviscount = 0;
    $onlinemembers = '';
    $doneusers = array();
    $query = $db->query("
        SELECT s.ip, s.uid, u.username, s.time, u.invisible, u.usergroup
, u.usergroup, u.displaygroup
        FROM ".TABLE_PREFIX."sessions s
        LEFT JOIN ".TABLE_PREFIX."users u ON (s.uid=u.uid)
        WHERE s.time > '$timecut' AND location1='$fid' AND nopermission
!= 1
        ORDER BY u.username ASC, s.time DESC
    ");
    while($user = $db->fetch_array($query))
    {
        if($user['uid'] == 0)
        {
            ++$guestcount;
        }
        else
 {
            if(empty($doneusers[$user['uid']]) || $doneusers[$user['uid'
]] < $user['time'])
            {
                $doneusers[$user['uid']] = $user['time'];
                ++$membercount;
                if($user['invisible'] == 1)
                {
                    $invisiblemark = "*";
                    ++$inviscount;
                }
                else
                {
                    $invisiblemark = '';
                }
                if($user['invisible'] != 1 || $mybb-
>usergroup['canviewwolinvis'] == 1 || $user['uid'] == $mybb-
>user['uid'])
                {
                    $user['username'] = format_name(htmlspecialchars_uni
($user['username']), $user['usergroup'], $user['displaygroup']);
                    $user['profilelink'] = build_profile_link($user['use
rname'], $user['uid']);
```

```
                    eval("\$onlinemembers .= \"".$templates-
>get("forumdisplay_usersbrowsing_user", 1, 0)."\";");
                    $comma = $lang->comma;
                }
            }
        }
    }
……
$perpage = $mybb->settings['threadsperpage'];
if($mybb->input['page'] > 0)
{
    $page = $mybb->input['page'];
    $start = ($page-1) * $perpage;
    $pages = $threadcount / $perpage;
    $pages = ceil($pages);
    if($page > $pages || $page <= 0)
    {
        $start = 0;
        $page = 1;
    }
}
else
{
    $start = 0;
  $page = 1;
}

$end = $start + $perpage;
$lower = $start + 1;
$upper = $end;

if($upper > $threadcount)
{
    $upper = $threadcount;
}

// Assemble page URL
if($mybb->input['sortby'] || $mybb->input['order'] || $mybb-
>input['datecut'] || $mybb->input['prefix']) // Ugly URL
{
    $page_url = str_replace("{fid}", $fid, FORUM_URL_PAGED);

    if($mybb->seo_support == true)
    {
        $q = "?";
        $and = '';
    }
    else
    {
        $q = '';
        $and = "&";
    }
……

 $plugins->run_hooks("forumdisplay_get_threads");
    // Allow viewing unapproved threads for logged in users
    if($mybb->user['uid'] && $mybb->settings['showownunapproved'])
```

```
    {
        $tvisibleonly .= " OR (t.fid='$fid' AND t.uid=".$mybb-
>user['uid'].")";
    }
    // Start Getting Threads
    $query = $db->query("
        SELECT t.*, {$ratingadd}t.username AS threadusername, u.username
        FROM ".TABLE_PREFIX."threads t
        LEFT JOIN ".TABLE_PREFIX."users u ON (u.uid = t.uid)
        WHERE t.fid='$fid' $tuseronly $tvisibleonly $datecutsql2 $prefix
sql2
       ORDER BY t.sticky DESC, {$t}{$sortfield} $sortordernow $sortfield2
        LIMIT $start, $perpage
    ");

    $ratings = false;
    $moved_threads = array();
    while($thread = $db->fetch_array($query))
    {
        $threadcache[$thread['tid']] = $thread;
        if($thread['numratings'] > 0 && $ratings == false)
        {
            $ratings = true; // Looks for ratings in the forum
        }

        // If this is a moved thread -
 set the tid for participation marking and thread read marking to that o
f the moved thread
        if(substr($thread['closed'], 0, 5) == "moved")
        {
            $tid = substr($thread['closed'], 6);
            if(!isset($tids[$tid]))
            {
                $moved_threads[$tid] = $thread['tid'];
                $tids[$thread['tid']] = $tid;
            }
        }
        // Otherwise - set it to the plain thread ID
        else
        {
            $tids[$thread['tid']] = $thread['tid'];
                 if(isset($moved_threads[$thread['tid']]))
            {
                unset($moved_threads[$thread['tid']]);
            }
        }
    }
    if($mybb-
>settings['allowthreadratings'] != 0 && $foruminfo['allowtratings'] != 0
 && $mybb->user['uid'] && !empty($threadcache) && $ratings == true)
    {
        // Check if we've rated threads on this page
        // Guests get the pleasure of not being ID'd, but will be checke
d when they try and rate
        $imp = implode(",", array_keys($threadcache));
```

```
        $query = $db-
>simple_select("threadratings", "tid, uid", "tid IN ({$imp}) AND uid = '
{$mybb->user['uid']}'");
        while($rating = $db->fetch_array($query))
        {
            $threadcache[$rating['tid']]['rated'] = 1;
        }
    }
}

// If user has moderation tools available, prepare the Select All featur
e
$selectall = '';
if(is_moderator($fid) && $threadcount > $perpage)
{
     $lang->page_selected = $lang->sprintf($lang-
>page_selected, count($threadcache));
    $lang->select_all = $lang->sprintf($lang-
>select_all, (int)$threadcount);
    $lang->all_selected = $lang->sprintf($lang-
>all_selected, (int)$threadcount);
    eval("\$selectall = \"".$templates-
>get("forumdisplay_inlinemoderation_selectall")."\";");
}
if(!empty($tids))
{
    $tids = implode(",", $tids);
}
// Check participation by the current user in any of these threads -
 for 'dot' folder icons
if($mybb->settings['dotfolders'] != 0 && $mybb-
>user['uid'] && !empty($threadcache))
{
    $query = $db-
>simple_select("posts", "DISTINCT tid,uid", "uid='{$mybb-
>user['uid']}' AND tid IN ({$tids}) {$visibleonly}");
    while($post = $db->fetch_array($query))
    {
        if(!empty($moved_threads[$post['tid']]))
        {
            $post['tid'] = $moved_threads[$post['tid']];
        }
        if($threadcache[$post['tid']])
        {
            $threadcache[$post['tid']]['doticon'] = 1;
        }

    }
}
// Read threads
if($mybb->user['uid'] && $mybb-
>settings['threadreadcut'] > 0 && !empty($threadcache))
{
    $query = $db->simple_select("threadsread", "*", "uid='{$mybb-
>user['uid']}' AND tid IN ({$tids})");
    while($readthread = $db->fetch_array($query))
    {
```

```
        if(!empty($moved_threads[$readthread['tid']]))
        {
            $readthread['tid'] = $moved_threads[$readthread['tid']];
        }
        if($threadcache[$readthread['tid']])
        {
            $threadcache[$readthread['tid']]['lastread'] = $readthread['
dateline'];
        }
    }
}
if($mybb->settings['threadreadcut'] > 0 && $mybb->user['uid'])
{
    $query = $db-
>simple_select("forumsread", "dateline", "fid='{$fid}' AND uid='{$mybb-
>user['uid']}'");
    $forum_read = $db->fetch_field($query, "dateline");
    $read_cutoff = TIME_NOW-$mybb->settings['threadreadcut']*60*60*24;
      if($forum_read == 0 || $forum_read < $read_cutoff)
    {
        $forum_read = $read_cutoff;
    }
}
else
{
    $forum_read = my_get_array_cookie("forumread", $fid);

    if(isset($mybb->cookies['mybb']['readallforums']) && !$forum_read)
    {
        $forum_read = $mybb->cookies['mybb']['lastvisit'];
    }
}
```

БІБЛІОГРАФІЧНА ДОВІДКА

Індивідуальна тема дипломної роботи бакалавра: **«**Розробка веб-орієнтованого сервісу з реалізацією багаторівневого ступеню доступу та системою моніторингу даних»

Обсяг пояснювальної записки: 80 аркушів

Кількість рисунків 53
Кількість додатків 1

Дата завершення роботи 17 червня 2024 року