

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

**«Інформаційна система обміну досвідом для
програмних інженерів»**

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Берест О.Б.

Студента групи ІН – 72

Задесенець Д.С.

СУМИ 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедри Довбиш А.С.

“ _____ ” _____ 2021 р.

**ЗАВДАННЯ
до випускної роботи**

Студента четвертого курсу, групи ІН-72 спеціальності “Інформаційно-комунікаційні технології” денної форми навчання Задесенця Дениса Сергійовича.

Тема: “Інформаційна система обміну досвідом для програмних інженерів”

Затверджена наказом по СумДУ

№ _____ від _____ 2021 р.

Зміст пояснювальної записки: 1) аналітичний огляд методів побудови інформаційної системи; 2) постановка завдання й формування завдань дослідження; 3) огляд, опис і обрання засобів для розробки; 4) розробка інформаційної системи обміну досвідом для програмних інженерів; 5) тестування та аналіз результатів.

Дата видачі завдання “ _____ ” _____ 2021 р.

Керівник випускної роботи _____ Берест О.Б.

Завдання прийняв до виконання _____ Задесенець Д. С.

РЕФЕРАТ

Записка: 58 стор., 28 рис., 1 табл., 1 додаток, 10 джерел.

Об'єкт дослідження — інформаційна система обміну досвідом для програмних інженерів.

Мета роботи — розробка інформаційної системи обміну досвідом для програмних інженерів. Сайт повинен бути зрозумілим для нових користувачів, простим у використанні, мати сучасний вигляд. Також повинна бути адаптація під мобільні пристрої.

Результати — розроблено інформаційну систему щодо поставлених задач. Розроблений сайт реалізовано мовою PHP та системою управління базами даних MySQL.

ІНФОРМАЦІЙНА СИСТЕМА, ФОРУМ ПРОГРАМІСТІВ, HTML5, CSS3, PHP, JAVASCRIPT, MYSQL.

Зміст

ВСТУП.....	5
1 ІНФОРМАЦІЙНИЙ ОГЛЯД	7
1.1 Дослідження актуальності реалізації веб-форуму.....	7
1.2 Огляд аналогів.....	8
1.3 Успішний проект.....	14
1.4 Постановка задачі	17
2 ПРОЕКТУВАННЯ СТРУКТУРИ ТА АРХІТЕКТУРИ	19
2.1 Вибір мови програмування.....	19
2.2 Огляд баз даних і систем управління БД.....	27
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	35
3.1 Структурна схема продукту	35
3.2 Процес ініціалізації форуму	37
3.3 Опис реєстрації, авторизації, зміни паролю та виходу	38
3.4 Опис створення теми та коментування.....	40
3.5 Оптимізація	40
3.6 Тестування сайту.....	41
3.7 Адаптація.....	45
ВИСНОВОК	47
СПИСОК ЛІТЕРАТУРИ.....	48
Додаток А	49

ВСТУП

Форум - це англomовне слово, яке в буквaльному перекладі означає «вільна дискусія» або «конференція». В інтернет-термінології під цим поняттям мають на увазі клас веб-додатків для організації спілкування відвідувачів веб-сайту. Перший веб-форум з'явився в 1994 році, одним з розробників якого виступив *Ari Luotonen*, відомий тим, що працював на початку 90-х років над створенням веб-сервера CERN (W3C). Даний форум був призначений тільки для розробників і носив виключно комерційний характер. Пізніше, навесні 1998 року, з'явився його безкоштовний аналог на мові PHP, розробником якого був *Brian Moon*. Перша назва його проекту було *WebThreads*, але згодом за ним зміцнилося ім'я *Phorum*.

Форум пропонує користувачам обмінюватися думками та досвідом в інтернеті. Користувачі спілкуються в різний час, а не в реальному часі, наприклад, у чаті.

Як правило, учасники форуму обмінюються думками на певну тему. Підтеми, що стосуються основної теми, також можуть обговорюватися на окремих підфорумах. Особливою популярністю користуються так звані довідкові форуми, учасники яких дають одне одному поради з певної теми. Але також великою популярністю користуються форуми, що служать для обміну інформацією про виробників обладнання та програмного забезпечення, а також форуми на актуальні політичні теми. На додаток до безкоштовних некомерційних форумів, компанії також пропонують форуми підтримки для допомоги та обговорення конкретних проблем із продуктами та послугами.

Для реалізації форуму доступні різні форми спілкування, такі як Usenet, в якому учасники надають доступ для читання та коментування, а також обмін електронною поштою через сервер форуму (списки розсилки) і навіть веб-форуми. Більшість веб-форумів не мають відкритого доступу, як Usenet, але вважають себе закритою спільнотою та потребують реєстрації.

Форум програмістів – це другий дім для розробників програмного забезпечення, місце, де вони можуть знайти експертів в області програмування, зустріти однодумців і того, на кого можна рівнятися. Саме про форум програмування кажуть «дві голови краще, ніж одна» і «місце сили».

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Дослідження актуальності реалізації веб-форуму

Метою даної роботи є програмна реалізація інтернет сценарію «веб-форум». Сценарій представляє собою набір модулів у вигляді файлів з розширенням php, що реалізують можливості обміну інформацією користувачів в мережі. Областю застосування даної курсової роботи можуть бути завдання забезпечення спілкування, користувачів між комп'ютерами по мережі Інтернет.

Актуальність вивчення даного питання пов'язана з затребуваністю подібних ресурсів в мережі інтернет.

Завдяки форумам люди можуть спілкуватися, обговорювати різні питання і проблеми, що незручно робити через коментарі до записів в блогах, соціальних мереж, чатів та гостьових книг. На форумах можна писати більш розгорнуто. Пости на форумах, що пройшли перевірку модератором за певними критеріями, залишаються там назавжди і стають доступними для пошукових систем, які знаходять їх за запитом користувачів мережі інтернет що робить ці пости корисними не тільки для тих людей які обговорювали подібну тему раніше, але і в подальшому для тих, хто стикається зі схожими питаннями. У соціальних проектах і в програмах обміном миттєвими повідомленнями подібні функції не реалізовані або присутні частково.

У той час як форум - це якесь «місце», в якому постійно кипить якесь життя: йдуть обговорення тієї чи іншої новини, проекту, it-пристрої і кожна людина може знайти собі однодумців на улюблені йому теми.

Ще один важливий аспект - це завоювання аудиторії молодим і талановитим письменником, художником, музикантом, і ін. Для цього більш зручний і підходить блог, де ти головний, а інші читачі. На практиці - на активному форумі тебе буде читати хоч хтось і шансів бути поміченим значно більше, а у не розкритого блозі є велика ймовірність так і залишитися з нулем коментарів.

Також важливим аспектом є зовнішній вигляд. Нажаль, зараз досить мало українських форумів в цілому, а якщо взяти и зарубіжні сайти, то вони виглядають старими, дизайн не змінювався зі старту роботи, функціоналу у них досить багато, тому склавши це до купи, новому користувачеві буде досить складно так відразу розібратися на сайті зі старим дизайном и новим функціоналом.

Можна зробити висновок що реалізація такого проекту як веб-форум для програмних інженерів буде актуальна і затребувана користувачами мережі інтернет.

1.2 Огляд аналогів

Перед початком розробки був проведений аналіз подібного роду веб-додатків в мережі інтернет і був зроблений висновок, що в даний час існує досить багато рішень такого завдання зважаючи на актуальність об'єкту досліджень. Але часто такі веб-форуми вимагають оплати послуг за надання доступу до розділів позначаючи їх закритим або працюють із застарілими даними. В цій частині протестуємо 4 форуму.

1) «Компьютерный форум Ru.Board» (рис 1.1)



[Новости](#) • [Файловые архивы](#)
[Поиск](#) • [Активные темы](#) • [Топ лист](#)
[Правила](#) • [Кто в on-line?](#)

[Вход](#) • [Забыли пароль?](#) • [Первый раз на этом сайте?](#) • [Регистрация](#)
 Привет, **Guest**
 Приветствуем новичка по имени **Mehanik83**
 Участников **2282286, 26542796** сообщ. в **755992** темах

11-02-2006 В помощь сисадмину: [Групповые политики Active Directory](#)
Внимание! Кому не приходит пароль, запросите его тут.

Информация о форуме	Ответы	Темы	Модератор	Обновления
Закладки	777	1	Guest	Дата: 02-09-2005 Время: 20:44 Автор: scales01 Форум-Закладки. Новая фича на...
» Компьютеры				
Операционные системы Выбор, установка, настройка, решение проблем с операционными системами. Включает форумы: Microsoft Windows, UNIX, Другие ОС	1160882	68157	lynx, dg, KLASS, IFkO, vu1tur	Дата: 04-06-2021 Время: 14:06 Автор: RandyOrton Обновления для Microsoft Windows XP...
Игры Все о компьютерных играх и вокруг них. (Активные темы) Общие вопросы Strategy Action & Arcade RPG & Adventure Sport & Simulation Небольшие интересные игры Форумные игры Новости Игрового Мира	1569660	23808	Cossack, RapSoldier, VdV	Дата: 04-06-2021 Время: 13:02 Автор: model15 Diablo: выбираем лучшую

Рисунок 1.1 – Головна сторінка сайту Ru-board

Головною перевагою цього сайту є його функціонал, хоча він застарілий. Кожна тема розбивається на декілька підтем. Після обрання підтеми відображаються останні запитання. Реєстрація тут проста автоматична, пароль приходить на пошту. Можна відмітити, що люди коментують записи 15-річної давності, можна зрозуміти, що сайтом користуються, але онлайн малий, гостеві доступний лише перегляд створених форумів, для інших дій потрібно зареєструватися та авторизуватися.

Панель навігації зовсім незручна. Більшість функцій в ній для гостьового користувача буде недоступна. (рис 1.2)

[Новости](#) • [Файловые архивы](#)
[Поиск](#) • [Активные темы](#) • [Топ лист](#)
[Правила](#) • [Кто в on-line?](#)

[Вход](#) • [Забыли пароль?](#) • [Первый раз на этом сайте?](#) • [Регистрация](#)
 Привет, **Guest**
 Приветствуем новичка по имени **hamenozі**
 Участников **2282289, 26542830** сообщ. в **755992** темах

Рисунок 1.2 – Панель навігації сайту Ru-board

Пошук незрозумілий, реалізований в технології «Google», але він відбувається по форумі, тому функціонує правильно. Отже, сайту потрібен редизайн, заохочення нових користувачів. Головний мінус це те, що сайт знаходиться на незахищеному домені, тому ним користуватися потрібно досить обережно і ніякі особисті дані, картки там не залишати.(рис 1.3)

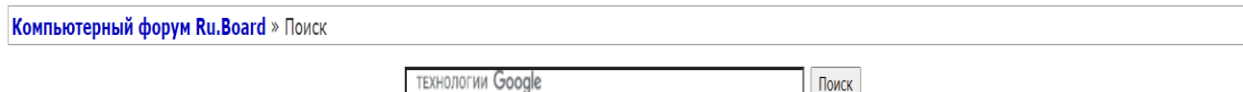


Рисунок 1.3 – Пошук на сайті Ru-board

2) «CyberForum.ru - форум программистов и сисадминов» (рис 1.4)

Форум программистов и сисадминов
Киберфорум
 КиберФорум - форум программистов, системных администраторов, администраторов баз данных, компьютерный форум, форум по электронике и бытовой технике, обсуждение софта. Бесплатная помощь в решении задач по программированию, математике, физике и другим наукам, решение проблем с компьютером, операционными системами.

Добро пожаловать, **Den1us**.
 Последний визит: Сегодня 14:36
 Ваши уведомления: 3
 Мои темы | Темы с моими ответами | Выход

Профиль | Правила | Блоги | Сообщество | Новые сообщения [разделы] | Поиск | Навигация

Форум программистов

C++
 Тем: 384,108 / Ответов: 1,941,339 / Онлайн: 912
 C++ для начинающих | C++: Сети | C++: WinAPI | C++: COM, OLE, ActiveX | Visual C++ | Visual C++: Базы данных | C++ Builder | C++ Builder: Базы данных | C++ Builder: FireMonkey | Boost C++ | C++: OpenCV | Кроссплатформенная разработка | C++_Qt | QML | C++ под Linux | C++: IDE, инструментарий

.NET
 Тем: 269,548 / Ответов: 1,295,251 / Онлайн: 550
 C# .NET | C# для начинающих | C# Windows Forms | C#: Базы данных, ADO.NET | C# Web, ASP.NET | C#: ASP.NET MVC | C#: API, боты | C#: ASP.NET Core | C#: Веб-сервисы и WCF | Visual Basic .NET | F# .NET | C++/CLI | C++/CLI Windows Forms | Xamarin | LINQ | Visual Studio | Mono

Python
 Тем: 47,119 / Ответов: 221,383 / Онлайн: 282

Компьютерный форум

Выбор конфигурации компьютера
 Тем: 18,974 / Ответов: 203,819 / Онлайн: 17
 Собрать офисный компьютер | Собрать игровой компьютер | Собрать сервер, NAS | Ангрейд компьютера | Собрать домашний компьютер | Собрать рабочую станцию

Компьютерное железо
 Тем: 83,634 / Ответов: 716,668 / Онлайн: 277
 Общие проблемы с компьютером | Материнские (системные) платы | Носители информации | SSD | Flash-накопители | Восстановление данных | Звук и акустические системы | Корпуса | Мониторы | BIOS, EFI | Процессоры | Видеокарты | Жесткие диски, HDD | CD, DVD, Blu-ray (BD) | Внешние жесткие диски | Оперативная память | Охлаждение | Блоки питания, UPS (ИБП) | Серверы и серверное железо | Прошивки BIOS

Ноутбуки, нетбуки, ультрабуки
 Тем: 26,642 / Ответов: 206,606 / Онлайн: 88
 Общие проблемы с ноутбуком | Ремонт ноутбуков | Прошивки BIOS ноутбуков | Выбор ноутбука | Схемы плат ноутбуков | Ангрейд ноутбуков

Рисунок 1.4 – Головна сторінка сайту CyberForum

Дизайн сайту виглядає застаріло і завдяки цьому важко розібратися на форумі, але ним користується дуже велика кількість людей, відповідь можна отримати досить швидко, реклама відсутня, навігація зручна, працює з 2006 року. Також там присутні українські запитання, з цього можна зробити висновок, що ним користуються українці.

Однією з переваг є створення свого блогу. Можливість його збереження, перегляд блогів інших користувачів. Також там присутній пошук. Отже, функціонал сайту досить великий, але сайт виглядає застаріло і не оптимізований під мобільні пристрої. (рис 1.5)

Результаты поиска
 Форум программистов и сисадминов Киберфорум > Поиск по форуму

Добро пожаловать, Den1us.
 Последний визит: Сегодня 14:36
 Ваши уведомления: 3
 Мои темы | Темы с моими ответами | Выход

Профиль Правила Блоги Сообщество Новые сообщения [разделы] Поиск Навигация

Поиск: Новые сообщения

Показано с 1 по 30 из 480.
 На поиск затрачено 0.17 сек.

Объявление: Обу
 Java Mentor (PR)

Ментор с оплатой после трудоустройства

Показов: 1,174
 Java 06.05.2021

Тема / Автор	Последнее сообщение	Ответов	Показов	Раздел
Резкий скачок посе verter12	Сегодня 17:42 от kotopes54	2	372	Google

Рисунок 1.5 – Блоги сайта CyberForum

3) «ForumDesign Technical Support» (рис 1.6)

FORUMDESIGN
 Technical Support

Форум О нас Портфолио Ссылки Участники Правила Поиск Регистрация Войти

Активные темы Мои Закладки

Объявление

Привет, Гость! Войдите или зарегистрируйтесь.

Добро пожаловать на ForumDTS.org!
 ForumDesign TechSupport - это некоммерческий проект помощи начинающим администраторам форумов и сайтов. Проект сочетает в себе помощь по дизайну, техническую поддержку и обучение всевозможным нюансам создания, разработки, оформления и раскрутки интернет-проектов. Подробнее о нас вы можете прочесть здесь: FAQ - Что такое ForumDesign TS?

События

Осенний нарафон: конкурс осенних наборов - если ты дизайнер, то участвуй в нашем по особенному осеннем конкурсе на создание лучших графических наборов для форумов.

Осенний нарафон: абсурдно-осенний коллаж - участвуй в осеннем конкурсе и получи конкурсные баллы!

Осенний нарафон: осенний скриншот - простой конкурс для всех пользователей форума! Окунись в осенне

Проект ForumDesign TS

Правила и FAQ	FAQ - Учимся делать скриншот форума!
Зал славы	Nikodima
Конкурсы и акции	Обсуждения конкурсов/предложения о проведении
Организационные вопросы	Занять должность Дизайнера
Разностороннее общение	Ваше логическое животное

Дизайн и техническое обслуживание на заказ

Рисунок 1.6 – Головна сторінка сайту ForumDesign

При аналізі даного продукту було виявлено, що для пошуку по форуму потрібне детальне вивчення інструкції і правильного введення ключових слів використовуючи спеціальні приставки «AND» щоб визначити слова, які повинні бути в результатах, «OR» для слів, які можуть бути в результатах, і «NOT» для слів, яких непотрібно шукати. Дані правила є не для всіх зрозумілими що ускладнить комфортний пошук по темам форуму. (рис 1.7)

Рисунок 1.7 – Пошук по форуму ForumDesign

На даному форумі є велика кількість реклами, навіть дорослого характеру і вона не фільтрується, що згубно впливає на комфортне використання сервісу

З плюсів можна також відзначити зручну навігацію, розташовану в центральній частині екрана, що дозволяє легко орієнтуватися по форуму. (рис 1.8)

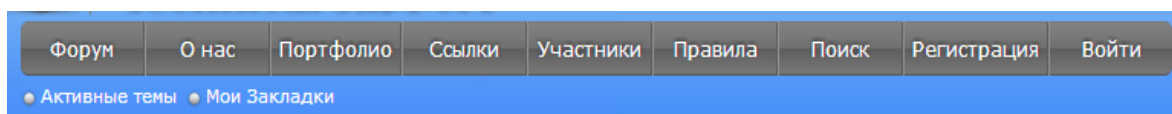


Рисунок 1.8 – Навігаційне меню форуму ForumDesign

Ще однією функцією є кнопка «Информация | Настройки», яка розташовується у верхній частині екрану. Котра має не зовсім читабельний вигляд, але є досить корисною. (рис 1.9)

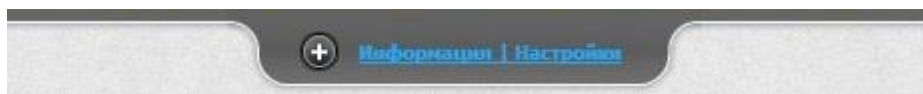


Рисунок 1.9 – Приклад кнопка навігації

Її активація розгортає навігаційну панель в якій користувач може авторизуватися або зареєструватися, а учаснику який побажав залишитися під обліковим записом «Гість» відкрито розділ «Діалог з адміністрацією». Також є кнопки «Налаштування форуму» які включають в себе опис розділів і зміну зовнішнього стилю сторінок форуму так звані «скіни». В іншій частині навігаційної панелі розташовуються кнопки «Контакти» що дозволяє отримати

зв'язок з адміністрацією або користувачами проекту через особисті повідомлення.

Таким чином основними недоліками форуму є: спливаюча реклама та реклама відволікаюча від перегляду, незручне сортування, не завжди інтуїтивно зрозумілий інтерфейс.

4) «REPLACE Український форум програмістів» (рис 1.10)

Активні теми Теми без відповідей Ви не увійшли. Будь ласка, увійдіть або зареєструйтесь.

Ласкаво просимо!

Ласкаво просимо вас на україномовний форум з програмування, веб-дизайну, SEO та всього пов'язаного з інтернетом та комп'ютерами.

Будемо вдячні, якщо ви поділитесь посиланням на Replase.org.ua на інших ресурсах.

Для того щоб створювати теми та надсилати повідомлення вам потрібно [Зареєструватись](#).

Український форум програмістів → Обговорення

Сторінки 1 | 2 | 3 | ... | 22 | Наступна Ви повинні увійти або зареєструватися для створення нової теми

Підфоруми: 4

Підфоруми	Теми	Повідомлень	Останнє повідомлення
Організація роботи форуму Все, що пов'язано з роботою форуму. Модератори: Chemist-1, Replase, leofun01	144	3 577	Вчора 23:05:33 від Droid 77

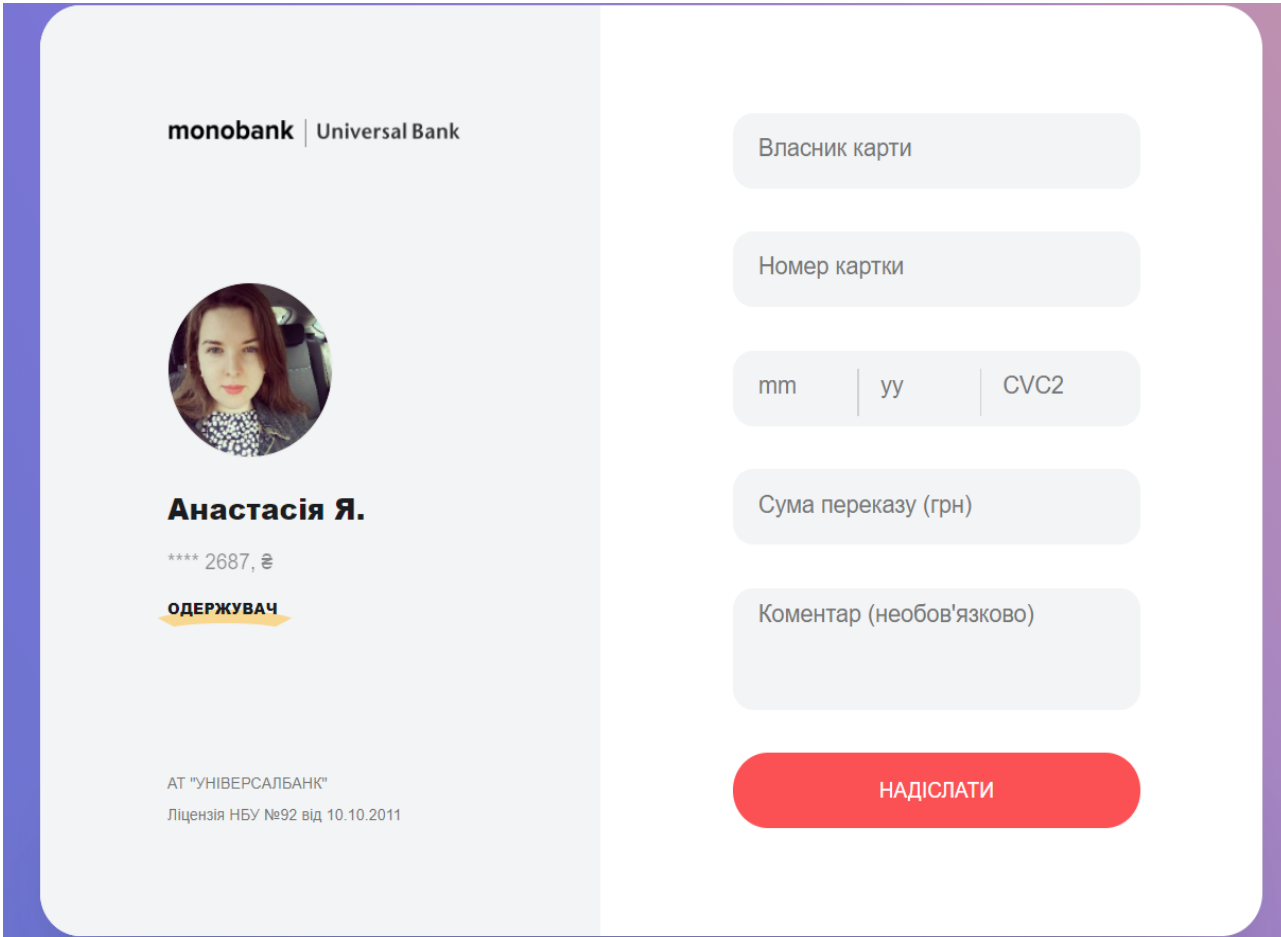
Рисунок 1.10 – Головна сторінка сайту REPLACE

REPLACE український форум програмістів – сайт, котрий легко знайти за однієї умови – він єдиний український. Виглядає сучасніше за інших представників. Реєстрація проста та захищена. Панель навігації зручна, пошук на сайті працює. (рис 1.11)

Рисунок 1.11 – Панель навігації сайту REPLACE

На панелі навігації знаходиться все саме головне, але там є реклама хостингу, на якому знаходиться сайт, хоча вона відкривається лише за власним бажанням, тому не заважає, а також присутня кнопка «Підтримати форум»,

завдяки ній нас перенаправляє на нову вкладку, на котрій знаходиться форма монобанку для переказу коштів, щоб підтримати розробників сайту. (рис 1.12)



monobank | Universal Bank

Анастасія Я.

**** 2687, ₪

ОДЕРЖУВАЧ

АТ "УНІВЕРСАЛБАНК"
Ліцензія НБУ №92 від 10.10.2011

Власник карти

Номер картки

mm | yy | CVC2

Сума переказу (грн)

Коментар (необов'язково)

НАДІСЛАТИ

Рисунок 1.12 – Форма переказу коштів монобанку

Отже, сайт має сучасний дизайн, але йому не вистачає функціоналу, потрібно приваблювати більше користувачів, тому що сайт працює з 2012 року, а піковий онлайн ледве досягає 70. І головне те, що додаток не оптимізований під мобільні пристрої та планшети, котрі з кожним днем все більше набирають популярності для серфінгу в інтернеті.

1.3 Успішний проект

А. Переваги:

Форум об'єднує людей, які цікавляться певною темою. На онлайн-форумах ті, хто шукає відповіді, зустрічаються з експертами, які зазвичай передають свої

знання без комерційного інтересу. Користувачі обмінюються ідеями, думками, досвідом, думками і архівуємо їх.

Перевага: *детальність*.

Успішні форуми надають найдетальнішу інформацію з конкретних питань і проблем. Обслуговується певна ніша, теми обговорюються на онлайн-форумах більш широко, ніж іншими способами. Контент, створений разом з плином часу, є структурованим і стійким. Це те, чим інтернет-форуми відрізняються від пропозицій соціальних мереж, таких як Facebook. У групах Facebook відповіді на питання губляться на часовій шкалі, і на одні й ті ж питання часто відповідають знову і знову.

Перевага: *надійність*.

В цілому на форумі одні і ті ж питання не задаються знову і знову, оскільки запитувач може і повинен спочатку знайти існуючу відповідь. Якщо питання задають двічі, на нього часто відповідають посиланням або посиланням на вже отриману відповідь. Відвідувачі мають можливість задавати питання і в ідеалі, швидко отримують відповіді або поради, які вони шукають.

Перевага: *анонімність*.

Додаткова анонімність – ще один аргумент на користь форумів. На відміну від пропозицій в соціальних мережах, з відповідним чином налаштованим на форумі є можливість задавати питання і відповідати на них анонімно. Це знижує поріг заборони відвідувачів сайту, коли мова йде про неприємні або дуже особистих темах.

З іншого боку, у так званих інтернет-тролів більше свободи, і вони легше поширюють зло. Обов'язки оператора форуму чітко регламентовані.

В. Етикет спілкування успішного форуму:

- Ведіть себе так само, як якщо б розмовляли з людьми в реальному житті;
- При створенні нового повідомлення створіть короткий опис;

- Намагайтеся з усіх сил правильно писати слова і використовувати правильну пунктуацію. Однак також усвідомте, що ніхто не ідеальний;
- Постійне підштовхування на повідомлення або заклик інших учасників до відповіді не допомагає і зазвичай викликає несхвалення;
- *«НЕ ПИШІТЬ УСІМА ВЕЛИКИМИ ГОЛОВНИМИ ЛІТЕРАМИ»*, так як це створює враження, ніби ви кричите;
- Перед створенням нової теми прочитайте правила публікації або інші прикріплені повідомлення;
- Поважайте всіх інших учасників форуму;
- Намагайтеся не вдаватися в подробиці і не робити пост більше трьох абзаців;
- Не відповідайте на старі теми або некропости;
- Дотримуйтесь правил, встановлені учасниками форуму.

С. Зробити форум відомим та успішним:

Крім іншого, можна розміщувати рекламу, хвалити свій форум в уже існуючих і посилатися на додаткову інформацію на своєму форумі в групах Facebook. Особливо важлива хороша видимість в основних пошукових системах. В ідеалі, коли хтось задає питання в Google, відповідь на який є на форумі, ця сторінка повинна з'являтися на першій сторінці результатів пошуку. Щоб це працювало, сайт повинен бути оптимізований для пошукових систем. Більшість представлених програмних рішень для форумів уже включають функції пошукової оптимізації.

Брендинг:

Крім можливості бути знайденим, важлива впізнаваність: потрібен хороший логотип. Це відображається на головній сторінці та на всіх підсторінках. Зверніть увагу на налаштування фавікону, тоді логотип з'явиться у вкладці браузера.

Швидкість:

Якщо веб-сайт не завантажується в протягом трьох секунд, 40% відвідувачів знову йдуть. Щоб цього не сталося, сторінка повинна завантажуватися швидко.

Мобільна адаптація:

Сьогодні більш ніж у 50% випадків для серфінгу в Інтернеті використовується мобільний пристрій. Тому веб-сайти повинні бути оптимізовані для смартфонів і планшетів і, звичайно ж, добре виглядати на великому екрані настільного ПК. Щоб це працювало, форум повинен використовувати адаптивний дизайн.

1.4 Постановка задачі

1. Загальні відомості.

1.1 Назва продукту розробки.

«Інформаційна система обміну досвідом для програмних інженерів».

1.2 Призначення продукту.

Функціональне та експлуатаційне призначення програми. Утримання вже наявних і залучення нових користувачів, а також застосування сайту як інструменту для більш зручної комунікації між клієнтом юзерами.

Необхідно, щоб користувач в результаті відвідування сайту:

Побачив – спеціалістів, котрі користуються форумом.

Відчув – доступність і легкість користування.

Зробив – зареєструвався та опублікував свою тему.

Отримав – вирішення його проблеми.

2. Цільова аудиторія.

Цільова аудиторія – програмісти будь-якого віку, котрі бажають швидко вирішити свою проблему.

3. Вимоги до продукту.

3.1 Вимоги в цілому.

Програмний продукт повинен бути розроблений на мові високого рівня програмування.

Додаток повинен забезпечувати:

- Зручний інтерфейс;
- Легкість в користуванні.

3.2 Апаратні вимоги.

Для створення сайту необхідні наступні мінімальні апаратні вимоги:

- Процесор з частотою більше 2 ГГц;
- Оперативна пам'ять 4 ГБ;
- Жорсткий диск 10 ГБ вільного сховища;
- Відеокарта 512 МБ.

3.3 Вказівки системного програмного забезпечення.

Для створення і використання програми необхідно наступне програмне забезпечення:

- Windows XP/ Vista / Windows 7/ Windows 8/ Windows 10;
- Браузери: Internet Explorer 7.0 і вище, Firefox 3.6 і вище, Google Chrome;
- Редактор коду Sublime Text 3;
- Локальний веб сервер Open Server.

До основних характеристик операційних систем відносять:

- розрядність;
- одно або багатозадачність ОС;
- підтримання багатокористувацького режиму;
- підтримання роботи в мережі та без підтримання роботи в мережі. [3]

4. Вимоги до користувацького інтерфейсу.

4.1 Розміщення інформації на екрані, дизайн.

Оформлення повинно бути читабельне.

4.2 Стиль сайту.

Форум повинен відповідати дизайну сучасних сайтів.

2 ПРОЕКТУВАННЯ СТРУКТУРИ ТА АРХІТЕКТУРИ

2.1 Вибір мови програмування

У наші дні, коли йде широке використання веб-додатків та іншого, зовнішній вигляд продукту і послуги є одним з вирішальних факторів успіху або невдачі продукту чи послуги. Щоб задовольнити широке коло клієнтів і охопити маси, веб-сайти і онлайн-портали використовуються в великих масштабах. В епоху інтернет-магазинів зовнішній вигляд веб-сайту має найбільше значення, оскільки він допомагає залучити клієнтів на сайт і допомагає генерувати трафік на порталі. Щоб працювати в цьому напрямку, використання мов веб-розробки або веб-програмування збільшується з кожним днем. Використовуючи ці мови, можна працювати над естетикою веб-сайтів і веб-порталів.

HTML & CSS

Мова розмітки гіпертексту і каскадні таблиці сценаріїв - одні з найбільш бажаних мов розробниками веб-сайтів. Ця мова – одна з найпопулярніших мов, що використовується веб-розробниками в наші дні. HTML допомагає визначити базову структуру веб-сайтів, тоді як за допомогою CSS можна керувати зовнішнім виглядом веб-сайту. За допомогою CSS, який має вбудовані бібліотеки, естетика веб-сайту може бути доведена до найкращого рівня, щоб допомогти залучити високий трафік на веб-сайт.

Мови веб-програмування діляться на дві групи: клієнтські та серверні. (рис 2.1)

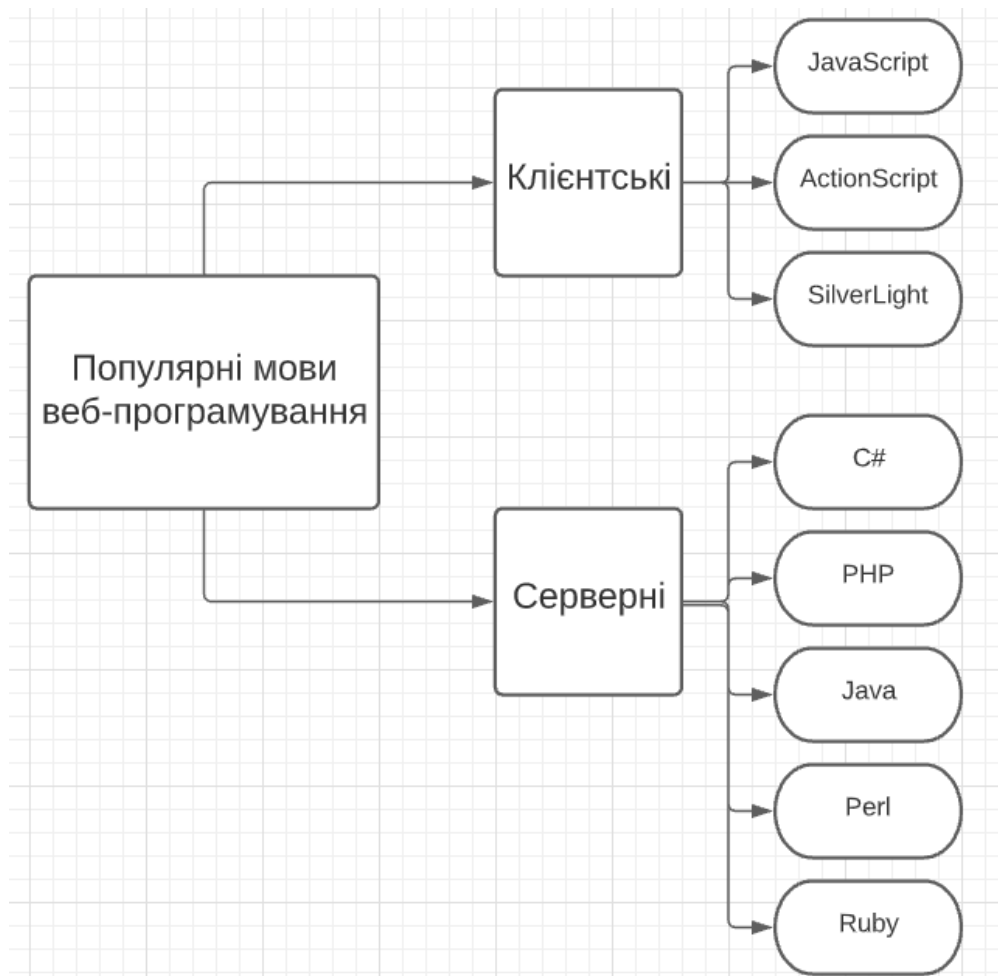


Рисунок 2.1 – Групи мов веб-програмування

Програми, написані на клієнтських мовах програмування обробляються на клієнтському комп'ютері. Говорячи простіше, скрипти обробляє браузер, що стоїть на Вашому комп'ютері. З цієї особливості випливають їх плюси і мінуси.

Найбільшим плюсом клієнтських мов програмування є те, що немає необхідності відправляти веб-документ для обробки на сервер. Ця особливість значно підвищує швидкість читання скрипту. Зменшується трафік, що економить гроші і знижується навантаження на сервер. Так само немає потреби ставити додаткове програмне забезпечення для обробки скриптів, тому що їх обробляють браузери.

Недоліки клієнтських мов програмування також випливають з особливостей. Різні браузери можуть інтерпретувати скрипти, виходячи зі своїх індивідуальних переваг. Тому, результати показу одних і тих же документів

можуть відрізнятись. Хоча, останнім часом робота різних браузерів все більш уніфікується. Так само до недоліків можна віднести і відкритість коду таких мов для користувачів.

JavaScript

Найпопулярніша клієнтська мова програмування – JavaScript. Поряд з HTML і CSS, JavaScript важливий для інтерфейсної веб-розробки. Більшість найпопулярніших веб-сайтів, від Facebook і Twitter до Gmail і YouTube, використовують JavaScript для створення інтерактивних веб-сторінок і динамічного відображення контенту для користувачів.[4] Сила полягає не тільки в тому, що він може працювати в браузері і на сервері з використанням Node.js, але і в прекрасних фреймворках, і бібліотеках для веб-розробки, і розробки додатків.

Приклад роботи:

1. Користувач відкриває свій веб-браузер (клієнт);
2. Користувач починає перегляд;
3. Клієнт отримує джерело сторінки і відображає його;
4. Тепер користувач вводить в рядок пошуку;
5. Потім клієнт відправляє дані на сервер;
6. Сервер обробляє дані і повертає відповідний результат пошуку;
7. Клієнт знову відображає його для перегляду користувачем;
8. Користувач отримує доступ до заданого посилання.

Серверні мови програмування відповідно працюють на стороні сервера. У взаємодії з базами даних вони підтримують зв'язок між користувачем і сервером. Отримуючи запит з адресою веб-документу від браузера, серверні програми зв'язуються з базою даних. БД віддає інформацію про веб-сторінку скриптам сервера, і ті обробивши її, відсилають для інтерпретації браузеру клієнта, який і виводить результат спільної роботи на монітор. (рис 2.2)

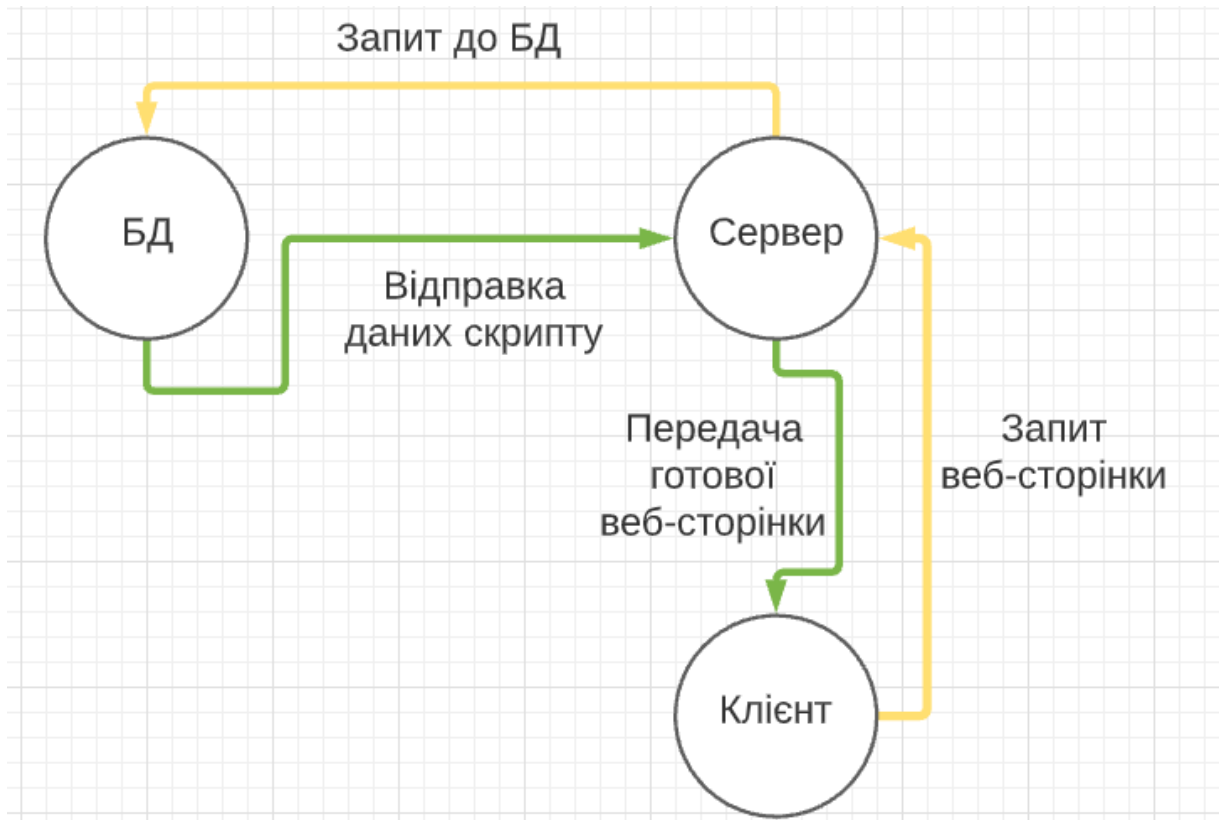


Рисунок 2.2 – Схема роботи сервера

Перевагою серверних мов є їх безмежні можливості і те, що їх робота не піддається впливу користувачів і прихована від їх поглядів. Недолік – залежність від програмного забезпечення. Так само до недоліків можна віднести і складність освоєння новачками серверних мов програмування.

Робота на стороні серверу:

- Обробка вводу користувача;
- Відображення запрошуваних сторінок;
- Структура веб-додатків;
- Взаємодія з серверами / сховищами;
- Взаємодія з базами даних;
- Запит бази даних;
- Кодування даних в HTML;
- Операції з базами даних, такі як видалення, оновлення.

C#

Як і C++, C# – це об'єктно-орієнтована мова загального призначення, побудована на основі C. Спочатку він був розроблений Microsoft як частина її платформи .NET для створення додатків Windows. C# використовує синтаксис, аналогічний синтаксису інших мов, похідних від C, таких як C++, тому його легко зрозуміти, якщо ви переходите з іншої мови сімейства C. C# – це не тільки засіб розробки додатків Microsoft, але і мова, яку мобільні розробники використовують для створення кросплатформних додатків.[5] Крім того, всім, хто цікавиться розробкою VR, слід подумати про вивчення C#. Він рекомендований мову для створення 3D і 2D відеоігор з використанням популярного ігрового движку Unity, який виробляє третину кращих ігор на ринку.

PHP

PHP (personal home page) – одна з найбільш часто використовуваних мов для веб-розробки та програмування з відкритим вихідним кодом. Це серверна мова і виконується тільки на серверах. Він в основному використовується для створення динамічного і привабливого вмісту веб-сторінок. Вона найбільш дружна до операційних систем, і її можна використовувати практично в будь-якій системі.

Це одна з найпростіших у вивченні мов, яка також відкриває широкі можливості для розробників. Динамічна серверна мова сценаріїв, яка спрощує створення повнофункціональних веб-додатків. [6] Якщо цього недостатньо, половина Інтернету працює на PHP, згадайте WordPress, найпопулярніше програмне забезпечення для веб-додатків, створене на PHP, а так веб-гігант Facebook.

Java

Java також є однією з кращих мов програмування з 38% користувачів у всьому світі через її добре підтримувані бібліотеки і веб-фреймворки. Це

об'єктно-орієнтована мова з відкритим вихідним кодом, який користується гарною репутацією завдяки своїй стабільності.

Основну сутність мови Java складають бібліотеки файлів, звані класами, кожен з яких містить невеликі фрагменти перевіреного, готового до виконання коду. Подібно цеглин, будь-які з цих класів можна вбудовувати в нову програму, і таким чином, для остаточного завершення програми зазвичай залишається написати невелику частину коду. [7]

Іншою перевагою є JVM (віртуальна машина Java), яка також дозволяє їй працювати на різних платформах, а розробка може обмежити ризики небезпеки і автоматично управляти сховищем пам'яті. Також, вона широко використовується для розробки мобільних додатків для Android або складних веб-проектів корпоративного масштабу. Через ці властивості Java важче вивчати, і новачки, ймовірно, будуть боротися з багатослівними і складними кодами, якщо вони забажають розпочати свою кар'єру з цієї мови. Крім того, процес розробки веб-програмного забезпечення на Java в невеликих масштабах практично не здійснюється.

Perl

Perl – універсальна мова програмування, створена Ларрі Уоллом в 1987 році. Ця мова отримала широку популярність завдяки передбаченій в ній потужній підтримці регулярних виразів і широкими можливостям обробки текстів.[8] Мову Perl можна вважати нащадком РНР. Ще її називають «скотчем для інтернету», тому що він може з'єднувати будь-які проекти, написані на будь-яких мовах:

- веб-сервер і БД;
- інтернет-магазини і служби поштових розсилок;
- термодатчики і систему управління шлюзами в гідроелектростанції;
- платну підписку на сайті і смс-оповіщення;
- систему відеоспостереження і міську службу швидкого реагування.

Perl добре вміє запускати системні команди в операційній системі. Завдяки цьому він може взяти результат роботи однієї програми, перетворити їх як потрібно і віддати в іншу. Його сила в роботі з регулярними виразами. Повна підтримка яких вже вбудована в мову. Він може взяти будь-які дані, обробити їх за потрібними правилами і дуже швидко видати результат в зручному для користувача вигляді. Його недолік – це складність для новачків, якщо вибрати його своїм першим мовою для вивчення. У ньому все не так просто з ООП, занадто вільне трактування операторів порівняння і величезні комбінації спецсимволів, в яких складно розібратися на старті. Тому Perl краще не вибирати як першу мову, а повернутися до нього, коли виникне практична необхідність.

Ruby

Сама мова є високорівневою і чисто об'єктно-орієнтованою, що означає, що «кожне значення є об'єктом», і в ній відсутні примітивні типи даних. У ньому також є сильна динамічна типізація і автоматичне прибирання сміття – форма для управління пам'яттю. Його найбільше люблять ділові люди і нові компанії, і він пропонує широкий вибір «діамантів» (додаткових предметів), які можуть змусити його робити практично все, що вам потрібно.

Підтримка Ruby включена практично в усі редактори і дуже розвинена. Є корисні і зрілі інтегровані середовища (IDE), які частково включають кошти побудови графічного інтерфейсу користувача (ГІІ). Безумовно, значно зросло і змінилося співтовариство. Сьогодні Ruby вже ніхто не назве ничжою мовою; він використовується в державних установах, наприклад в НАСА і в Національному управлінні океанічних і атмосферних досліджень (NOAA), в корпораціях, наприклад IBM і Motorola, і на таких добре відомих сайтах, як Wikipedia, GitHub і Twitter. Він застосовується для створення графічних додатків, роботи з базами даних, численних розрахунків і багато чого іншого. [9]

Все ідеально, але він має ряд недоліків:

- швидкість роботи дуже низька;
- досить складно знайти гарну документацію;

- вивчення мови високого рівня є досить складним;
- він досить повільно розвивається.

На основі цих даних розглянемо таблицю 2.1 – можливості мов програмування.

Таблиця 2.1 – Можливості мов програмування

Можливість/мова	Java	JavaScript	Perl	PHP	Ruby
Шаблони	+	-	-	+	-
Створення об'єктів	+	+	-	-	-
Збір сміття	+	+	+	+	+
Цілі числа довільної довжини	+	-	+	+	+
Параметричний поліморфізм	+	-	-	+	-
Значення параметрів за змовчуванням	-	-	-	+	+
Локальні функції	-	+	+	+	-
Лексичні замкнення	-	+	+	+	+
Вивід сигнатури	-	+	-	+	+
Інтерфейси	+	+	-	+	-
Мультиметоди	-	-	-	-	-
Множинне наслідування	-	-	+	-	-

Збір сміття – звільнення пам'яті від об'єктів, котрі не використовуються.

Цілі числа довільної довжини – числа, необмеженої розрядності.

Параметричний поліморфізм – здатність вказування `type classes` або `Constraints` для параметрів.

Значення параметрів за замовчуванням – виклик функції з опусканням деяких параметрів, щоб підставлялись значення за замовчуванням, вписані при її визначенні.

Локальні функції – здатність визначити функцію всередині іншої.

Інтерфейси – синтаксична і семантична конструкція в коді програми, яка використовується для специфікації послуг, що надаються класом.

2.2 Огляд баз даних і систем управління БД.

Для правильної роботи сайту потрібні не тільки файли з кодом сторінок, але і бази даних. Для взаємодії з БД використовуються системи управління базами даних (СУБД).

Що таке дані?

Дані – це будь-яка інформація, яка зберігається в пам'яті комп'ютера. Ця інформація пізніше може бути використана для веб-сайту, додатка або будь-якого іншого клієнта для зберігання в майбутньому. Найпоширеніша інформація – це інформація про користувача в вигляді особистої, адресною і банківської інформації. Давайте розглянемо Facebook, він зберігає наші особисті дані, зображення, повідомлення, коментарі та багато іншого. Банківський додаток також зберігає призначені для користувача дані, деталі їх транзакцій, зведення коштів тощо. Вся ця інформація є даними, але коли вона збирається і зберігається структурним чином, вона стає інформаційними даними.

Але як ці додатки або веб-сайти отримують дані? Коли ви публікуєте статус в Facebook, виконуєте банківську транзакцію в Інтернеті або завантажуєте селфі в Instagram, ви фактично відправляєте дані на сайт або, якщо бути точним, на їх сервер. Отже, ми можемо сказати, що будь-яка передана або передана інформація насправді є даними. Сервер відфільтровує необхідні дані і зберігає їх в базі даних. Тут виникає концепція БД.

Що таке БД?

Це випадковий набір речей, стислих разом? Ні, бази даних організовані, у них є структура, і всі дані, які вони зберігають, вписуються в цю структуру. Зокрема, це електронна система, яка дозволяє зберігати дані, легко отримувати до них доступ, маніпулювати і оновлювати.

Отже, ми тільки що бачили визначення баз даних, але де в реальному житті ми можемо побачити такі речі? Що ж, всі ми добре знаємо такі речі, як телефонний довідник. Ну, телефонна книга / телефонний довідник насправді містить імена людей, які знаходяться у порядку їх прізвищ. Ми можемо зберігати дані і ефективно шукати їх, використовуючи прізвище людини. БД складається в основному з двох компонентів: даних і значимого методу доступу до даних і управління ними. Без цих двох баз даних буде просто випадковий набір даних. Більш точним прикладом бази даних може бути словник, в якому велика кількість даних зберігається у вигляді пар «ключ-значення». У той же час він також має значимий метод доступу до даних за допомогою ключа.

БД дуже схожі на електронні таблиці, оскільки вони в основному складаються з таблиць, які містять рядки і стовпці, як електронна таблиця. Вона повинна бути розміщена або створена на якийсь спеціальній платформі. Ось деякі з відомих платформ баз даних:

- PostgreSQL;
- MySQL;
- Microsoft Access;
- SQLite.

Ми знаємо, що коли ми використовуємо будь-яке програмне забезпечення, наприклад додаток, дані або додаються в базу даних, або оновлюються відповідно до дії, або витягуються відповідно до виконуваних нами дією. Але в комп'ютері теж має бути щось, щоб виконувати таку дію. Програмне забезпечення, яке може виконувати різні операції з базами даних, відомо як система управління базами даних (СУБД).

Що таке СУБД?

Система управління базою даних – це програмне забезпечення, що використовується для виконання різних операцій, таких як додавання, доступ, оновлення та видалення даних, наприклад додавання Вашого імені в базу даних для інтернет-магазину роздрібною торгівлі в якості клієнта. Вона виступає в

якості основи БД і спрощує її використання, оскільки значно полегшує доступ до даних і управління ними. Тому, потрібно відчутти різницю між БД та СУБД.

БД чи СУБД?

Що ж, можна подумати, що база даних і система управління базами даних схожі один на одного, але є дійсно велика різниця. Давайте розберемося в цьому на прикладі.

Тут ми бачимо (Рисунок 2.3) додаток або веб-сайт з власною базою даних. Отже, основна функція – підключити базу даних до додатка або веб-сайту. Хоча база даних може надавати дані на веб-сайт і мати осмислений спосіб доступу до даних, вона як і раніше не може зрозуміти мову або метод, які сайт може використовувати для отримання даних, тобто вона не може зрозуміти команди самостійно. Таким чином, щоб вирішити цю проблему, ми вводим концепцію системи управління базами даних. СУБД розуміє команди і запити, які визначають, які дані потрібні додатком або веб-сайтом, і таким чином, використовує осмислений метод доступу до бази даних для отримання інформації. Можна сказати, що СУБД діє як сполучний міст між базою даних і користувачем, а СУБД дійсно розуміє запити і допомагає базі даних зрозуміти вимоги. (рис 2.3)

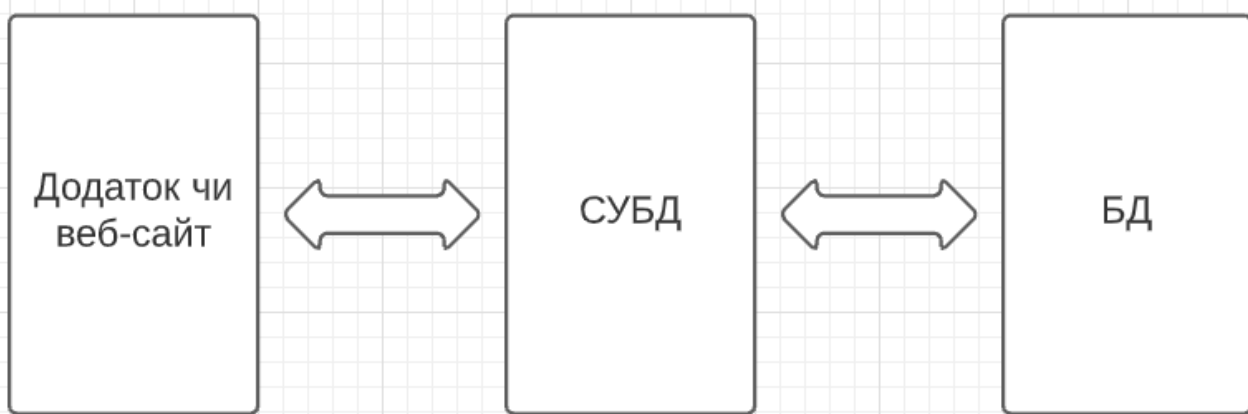


Рисунок 2.3 – Схема зв'язку між додатком і БД

СУБД діє як з'єднувач між додатком і базою даних. З цим головною відмінністю ясно, що база даних зберігає дані і надає метод для доступу до них,

СУБД фактично перетворює запити в значиму команду для виклику методу, використовуваного для доступу до бази даних.

PostgreSQL

Проект PostgreSQL стартував в 1986 році на факультеті комп'ютерних наук Берклі Каліфорнійського університету. Спочатку проект називався POSTGRES в зв'язку з більш старою базою даних Ingres, яка також розроблялася в Берклі. Метою проекту POSTGRES було додавання мінімальних функцій, необхідних для підтримки декількох типів даних. У 1996 році проект POSTGRES був перейменований в PostgreSQL, щоб наочно продемонструвати підтримку SQL. Сьогодні PostgreSQL зазвичай скорочується до Postgres.

PostgreSQL працює у всіх основних ОС, включаючи Linux, варіанти UNIX та Windows, і повністю сумісний з ACID, з повною підтримкою зовнішніх ключів, об'єднань, уявлень, тригерів і збережених процедур (на декількох мовах). Як БД корпоративного класу PostgreSQL може похвалитися такими функціями, як:

- багатoversійний контроль;
- відновлення даних за часом;
- табличність;
- точки збереження;
- запис в журнал для забезпечення відмовостійкості;
- онлайн або резервне копіювання.

Він також підтримує міжнародні набори символів, багатобайтові кодування символів, сортування, чутливість до регістру і форматування. PostgreSQL має високу масштабованість як за кількістю даних, якими він може управляти, так і за кількістю одночасних користувачів, яких він може обслуговувати. У виробничих середовищах існують активні системи PostgreSQL, які обробляють понад 4 ТБ даних.

MySQL

MySQL - це структурована реляційна система управління базами даних з відкритим вихідним кодом, добре відома в силу її продуктивності, простоти у використанні і надійності. Це найбільш поширений варіант для веб-додатків на основі реляційної бази даних. На поточному ринку тисячі веб-додатків спираються на MySQL, включаючи такі гіганти, як Facebook, Twitter і Wikipedia. [10]

Спочатку MySQL був запущений ще в 1995 році. З тих пір він зазнав кілька змін в сфері керування, перш ніж в 2010 році перейшов в корпорацію Oracle Corporation. Хоча зараз керує Oracle, MySQL і раніше був програмним забезпеченням з відкритим вихідним кодом, Це означає, що ви можете вільно використовувати і змінювати його.

Коли справа доходить до зберігання даних в, ви можете використовувати різні підходи. MySQL вибирає підхід, званий реляційної БД. У реляційній базі даних ваші дані розбиваються на кілька окремих областей зберігання, званих таблицями, замість того, щоб збирати все разом в одну велику одиницю зберігання. Наприклад, припустимо, що ви хочете зберегти два типи інформації:

- Клієнти – їх ім'я, адреси, реквізити;
- Замовлення – наприклад, які продукти були придбані, ціну, хто зробив замовлення.

Якщо ви спробуєте об'єднати всі ці дані в один великий горщик, у вас виникне кілька проблем, наприклад:

- Різні дані – дані, які вам потрібно зібрати для замовлення, відрізняються від даних для клієнта;
- Повторювані дані – у кожного покупця є ім'я, і у кожного замовлення також є ім'я покупця. Обробка цих повторюваних даних стає безладною;

- Як надійно зв'язати інформацію про замовлення з інформацією про клієнта?

Щоб вирішити ці проблеми, реляційна база даних буде використовувати одну окрему таблицю для клієнтів і іншу окрему таблицю для замовлень. Однак Ви, ймовірно, також забажаєте сказати: «покажіть мені все замовлення для Джона». Ось тут і з'являється реляційна частина.

Використовуючи так званий «ключ», ви можете пов'язати дані з цих двох таблиць разом, щоб ви могли маніпулювати і комбінувати дані в різних таблицях в міру необхідності. Важливо відзначити, що ключ – це не ім'я клієнта. Замість цього ви використовували б щось 100% унікальне, наприклад числовий ідентифікатор. Ключ – ідентифікаційний номер – це те, що пов'язує все воєдино. І ось так вони «зв'язані» один з одним, незважаючи на те, що зберігають дані в окремих таблицях.

Переваги MySQL:

- багатопоточність, підтримка декількох одночасних запитів;
- простота в використанні, наявність багатьох плагінів;
- великий функціонал;
- гарна безпека;
- масштабованість, можливість працювати і з малими, і з великими об'ємами даних;
- висока продуктивність.

MySQL функціонує на більшості платформ:

- AIX;
- BSDi;
- FreeBSD;
- HP-UX;
- Linux;
- Mac OS X;

- NetBSD;
- 2 Warp;
- Solaris;
- SunOS;
- SCO OpenServer;
- SCO UnixWare;
- Tru64;
- Windows 95;
- Windows 98;
- Windows 2000;
- Windows XP;
- Windows Server 2003;
- Windows Vista;
- Windows 7;
- Windows 8.1;
- Windows 10.

Таким чином у цьому розділі були розглянуті найвідоміші засоби реалізації нашого форуму. За основну мову веб-програмування був взятий PHP, тому що:

- легкий у вивченні та використанні;
- простий у втіленні в html-код;
- велика безкоштовна документація;
- працює з багатьма ОС;
- підходить для невеликих проектів.

Жоден сайт неможливий без бази даних. Проаналізувавши розглянуті БД було обрано MySQL. Обрана БД:

- проста у освоєнні;
- гарно захищена;
- швидка в роботі;

- підтримується багатьма ОС;
- є безкоштовною.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Структурна схема продукту

Структурна схема – схема, що відображає склад і взаємодію з управління частин продукту, що розробляються.

Діаграма використання (use case diagram) - це найбільш загальне уявлення функціонального призначення системи. На діаграмі використання застосовуються два типи основних сутностей: варіанти використання і дійові особи, між якими встановлюються такі основні типи відносин:

- асоціація між дійовою особою і варіантом використання;
- узагальнення між діючими особами;
- узагальнення між варіантами використання;
- залежності між варіантами використання. [1]

Гість має можливість зареєструватися, авторизуватися та переглянути теми. Коли людина авторизована, вона має можливість створити топик та залишити коментарі. Щодо функціоналу додатку, коротко він представлений на рисунку 3.1.

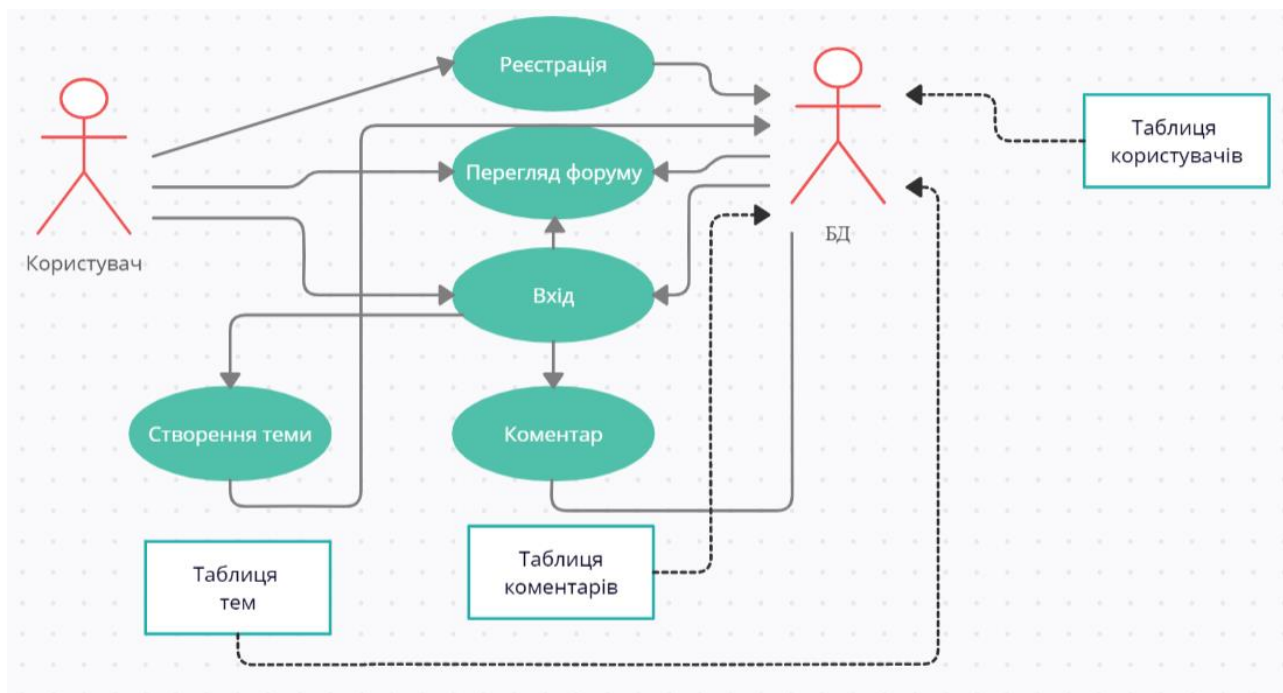


Рисунок 3.1 – діаграма use-case

Одним з найбільш відомих і зрозумілих методів проектування реляційних баз даних є метод, в основу якого покладена модель «сутність-зв'язок». Ключовими елементами моделі являються сутності, їх зв'язки та характеристики. [2]

Модель має 4 сутності:

- «Гість»;
- «Профіль»;
- «Тема»;
- «Коментар».

Кожен гість має можливість створити лише 1 профіль. Користувач може створювати декілька тем та їх коментувати. В свою чергу другі повинні мати своїх власників, а коментарі повинні мати теми. (рис 3.2)

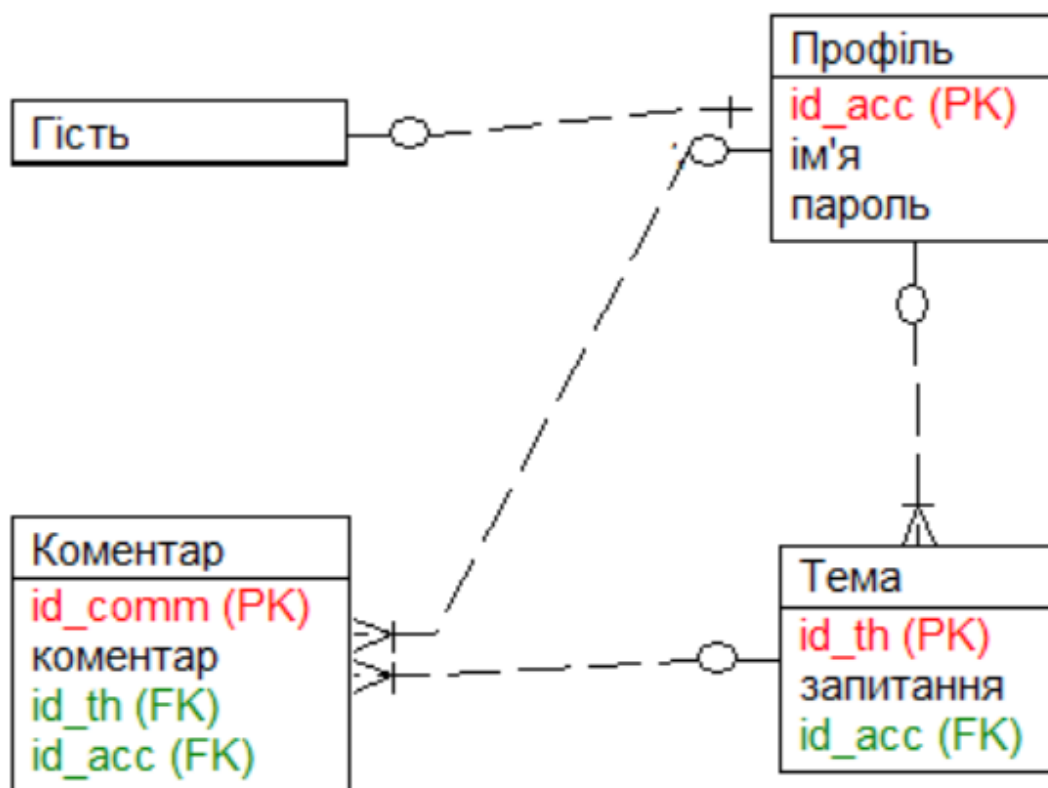


Рисунок 3.2 – ER діаграма

Для створення сайту був використаний редактор коду Sublime Text 3. Можна виділити його переваги:

- безкоштовний;
- висока швидкість;
- низькі системні вимоги;
- підтримка багатьох ОС;
- величезна кількість плагінів та доповнень.

3.2 Процес ініціалізації форуму

За допомогою PhpMyAdmin була створена база даних з ім'ям «progforum». Дані будуть кодуватися стандартом «Windows-1251» - тобто повна сумісність з українською мовою. Далі будуть створені таблиці на віддаленому сервері з розміщеним на ньому MySQL і Open Server. Створена таблиця «users» відповідає за права для роботи з базою даних. У цю таблицю буде занесений користувач для подальшого заповнення таблиць даними. Таблиці «comment», «topic» відповідно містять дані про коментарі та теми. Дана БД буде підключена до програмного модулю для внесення даних і подальшої модифікації. Після чого на сервері запускається консоль управління MySQL і Open Server.

Для з'єднання нашого сайту з БД розглянемо код:

```
define ('DB_HOST', 'localhost');
define ('DB_USER', 'root');
define ('DB_PASSWORD', 'root');
define ('DB_NAME', 'progforum');
$Connect_db = mysqli_connect(DB_HOST, DB_USER,
DB_PASSWORD);
mysqli_select_db($Connect_db, DB_NAME);
```

Визначаються константи та встановлюються параметри, задається логін, пароль та назва БД. Виконується підключення та обрання бази.

3.3 Опис реєстрації, авторизації, зміни паролю та виходу

Для входу користувач повинен авторизуватися. Авторизація виконується при введенні логіна і пароля у відповідні поля форми авторизації. Для цього користувач повинен бути заздалегідь зареєстрований на форумі, якщо гість не зареєстрований він може викликати форму реєстрації та заповнити всі поля. Після чого він зможе увійти на форум використовуючи особисті дані для входу ввівши їх в форму авторизації на сайті. Звичайному користувачу котрий пройшов авторизацію присвоюється обліковий запис Гість, в рамках якої можна переглядати опис розділів форуму, без можливості створення тем, коментування та оцінки.

```
$Login= $_POST['login'];
$Pass= $_POST['password'];
$Pass2= $_POST['password2'];
if (! filter_var($Login, FILTER_VALIDATE_EMAIL))
exit('Невірна пошта');
$result= mysqli_query($Connect_db, "SELECT 1 FROM
`users` WHERE login = '{$Login}' LIMIT 1 ");
if (mysqli_num_rows($Result) != 0)
exit('Ця пошта вже зареєстрована, поверніться назад');
```

Отримуємо дані введені в полях, перевірка на вірність введення пошти та наявність в базі.

```
if ($Pass != $Pass2)
exit('Паролі не співпадають, поверніться назад');
if (mb_strlen($Pass) < 6 or mb_strlen($Pass) > 9)
exit('Пароль повинен бути більше 6 символів і менше 9,
поверніться назад');
if (mb_strlen($Pass2) < 6 or mb_strlen($Pass2) > 9)
exit('Пароль повинен бути більше 6 символів і менше 9,
поверніться назад');
$Pass= md5($Pass);
mysqli_query($Connect_db, "INSERT IGNORE INTO users
(login, pass) VALUES ('{$Login}', '$Pass') ");
$id= mysqli_insert_id($Connect_db);
$_SESSION['id']= $id;
$_SESSION['login']= $Login;
```

```
header('Location: index.php');
```

Потім відбувається перевірка паролю на його співпадіння та правильність введення, для безпеки він хешується та додається в базу користувач. Після цього його відразу направляє на головну сторінку вже авторизованим.

Аналогічним чином відбувається зміна паролю.

```
$Pass= $_POST['password'];
$Pass2= $_POST['password2'];
$Pass3= $_POST['password3'];
if ($Pass2 != $Pass3)
exit('Паролі не співпадають, поверніться назад');
if (mb_strlen($Pass) < 6 or mb_strlen($Pass) > 9)
exit('Пароль повинен бути більше 6 символів і менше 9,
поверніться назад');
if (mb_strlen($Pass2) < 6 or mb_strlen($Pass2) > 9)
exit('Пароль повинен бути більше 6 символів і менше 9,
поверніться назад');
$Pass= md5($Pass);
$result= mysqli_query($Connect_db, "SELECT 1 FROM
`users` WHERE pass = '{$Pass}' LIMIT 1 ");
if (mysqli_num_rows($Result) != 0) // Перевірка
пройдена
{
$Pass2= md5($Pass2);
mysqli_query($Connect_db, "UPDATE users SET pass
= '{$Pass2}' WHERE pass = '{$Pass}' LIMIT 1 ");
header('Location: new_pass.php');
exit();
}
else //Перевірка не пройдена
exit('Старий пароль невірний, поверніться назад');
```

Отримуємо дані з введених полів. Перевірка нових полів на вірність введення. Хешування старого паролю, для того щоб знайти його в базі. Після знаходження старого паролю в базі відбувається хешування нового, оновлення таблиці с паролями.

Вихід відбувається досить просто. Спочатку йде перевірка на користувача, функція `unset` видаляє змінну профілем та перекидає на головну сторінку. Ось так це реалізовано:

```
session_start();
unset($_SESSION['login']);
header('Location: index.php');
```

3.4 Опис створення теми та коментування

Для створення теми та коментуванню потрібно бути авторизованим.

```
$Login= $_SESSION['login'];
$title= $_POST['title'];
$text= $_POST['text'];
$date= date('d.m.Y');
mysqli_query($Connect_db, "INSERT INTO topic (title,
text, login, date, views) VALUES ('$title', '$text',
'$Login', '$date', '0') ");
$id= mysqli_insert_id($Connect_db);
header('Location: topic.php?id=' . $id);
```

Спершу відбувається зчитування полів, отримання назви теми, її вміст та дата. Запис цих даних до бази та присвоєння темі унікально `id`. Користувача після створення відразу направляє на його тему.

```
$id= $_POST['id'];
$Login= $_SESSION['login'];
$text= $_POST['text'];
$date= date('d.m.Y');
mysqli_query($Connect_db, "INSERT INTO comment
(topic_id, login, text, date, lince) VALUES ('$id',
'$Login', '$text', '$date', '0') ");
header('Location: topic.php?id=' . $id);
```

За аналогічним принципом додаються коментарі до БД. Отримуємо дані введені в поля. Додаємо їх до бази та повертаємося на сторінку з відкритою темою.

3.5 Оптимізація

Файли `footer.php` та `header.php` були винесені окремо, для того щоб їх підключати в кожен файл ось так:


```
require_once('header.php');
require_once('footer.php');
```

Ця дія зменшила код в інших файлах на 150 рядків.

3.6 Тестування сайту

Великі сили були вкладені в роботу над інтерфейсом, зокрема в дизайн, адже привабливий дизайн відразу налаштовує користувача на позитивне і довге проведення часу на сайті.

На панелі навігації (рис 3.3) користувач може переключатися між вкладками «Головна», «Швидке запитання» (рис 3.7), «Правила» (рис 3.8). Також є можливість зареєструватися (рис 3.11) або виконати вхід (рис 3.10) в шапці. Натиснувши на 3 стрічки відкривається вікно (рисунок 3.4), яке дозволяє вийти користувачеві або змінити пароль. (рис 3.5)

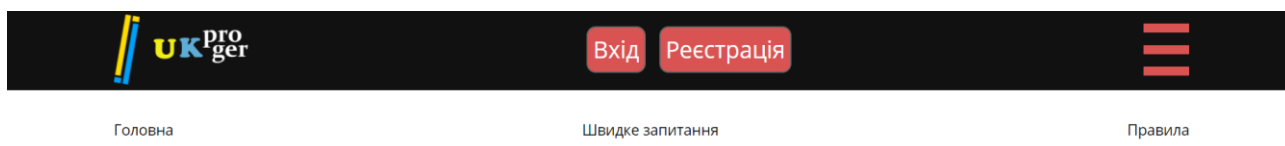


Рисунок 3.3 – Панель навігації

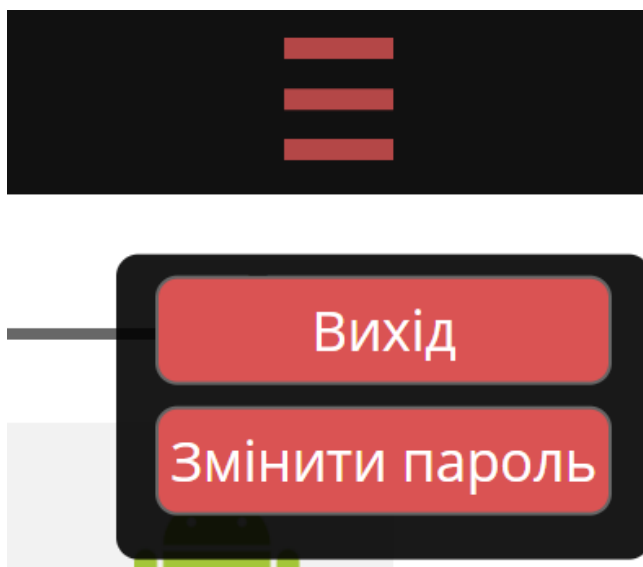


Рисунок 3.4 – Вигляд бургеру

Старий пароль

Новий пароль

Підтвердіть пароль

[Змінити пароль](#)

Рисунок 3.5 – Форма зміни паролю

На головній користувач може виконати пошук по назві тем або відобразити всі останні, дізнатися кількість відповідей в них, переглядів та дату створення. В кінці таблиці присутня пагінація, ще нижче знаходиться підвал на кожній сторінці.

Головна Швидке запитання Правила

HTML JS MySQL php Python Java Android

[Пошук](#) [Всі теми](#)

Нещодавно опубліковані	Відповіді	Переглядів	Активний
Як додати стил?	0	1	09.06.2021
Документація про бутстрап 4.	0	1	09.06.2021
Як змінити button?	0	1	09.06.2021

<< 1 2 3 4 5 >>

© Copyright 2021, Всі права захищені

[f](#) [@](#) [t](#)

Рисунок 3.6 – Головна сторінка сайту

Через вкладку «Швидке запитання» зареєстрований користувач направляє на сторінку, на котрій знаходиться форма, щоб створити тему. (рис 3.7)

Нове запитання

Заголовок

Пишіть тут

Створити

Рисунок 3.7 – Форма створення теми

На вкладці «Правила» всі користувачі мають змогу ознайомитися з правилами форуму та політикою конфіденційності. (рис 3.8)


deniszadesenets


Головна
Швидке запитання
Правила

Правила форуму

1. Загальні положення

1. Форум призначений для обговорення питань ІТ-тематики і суміжних з нею. Обговорення політичних тем заборонено.
2. Кожен користувач погоджується з цими Правилами фактом реєстрації на форумі і участі в ньому. Незнання Правил не звільняє від відповідальності.
3. Якщо користувач відмовляється дотримуватися цих Правил, то він повинен або самостійно покинути форум, або його аккаунт буде заблокований.
4. Офіційними мовами форуму є українська та англійська мови. Розміщення повідомлень на інших мовах заборонено.
5. Справжній форум є приватним ресурсом, тому Адміністрація форуму має виняткове право визначення політики щодо користувачів, їх обов'язків, правил поведінки і публікуються ними контенту, а також має право відмовити в публікації і / або присутності на форумі кому б то не було.

2. Політика щодо користувачів і публікованого ними контенту

1. Ці Правила поширюються на всіх користувачів і весь контент, що публікується ними на форумі. Акаунти користувачів форуму не видаляються.
2. Повідомлення і теми, а також інший контент, що розміщується на форумі, на прохання користувачів не видаляється і не закривається.
3. Теми, повідомлення і будь-який інший контент, розміщений на форумі, може бути відредагований або видалений Адміністрацією форуму без повідомлення їх автора.

3. Правила поведінки на форумі

Рисунок 3.8 – Правила форуму

Головна Швидке запитання Правила

Як додати стилі? 09.06.2021

denizzadesenets
Як додати стилі в хтмл файл?

створений	відповідей
09.06.2021	0

Коментар

Відповісти

Рисунок 3.9 – Вигляд відкритої теми

Швидке запитання

Авторизація X

Пошта

Пароль

Ввійти

Рисунок 3.10 – Вікно авторизації

Швидке запитання

Реєстрація X

Пошта

Пароль

Педтвердіть пароль:

Зареєструватися

Рисунок 3.11 – Вікно реєстрації

3.7 Адаптація

Сайт орієнтувався на мобільних користувачів, тому був розроблений також під них. Нижче наведено рисунки сайту з використання мобільного пристрою. Для повної картини, деякі картинки були зроблені через функцію «прокрутка».

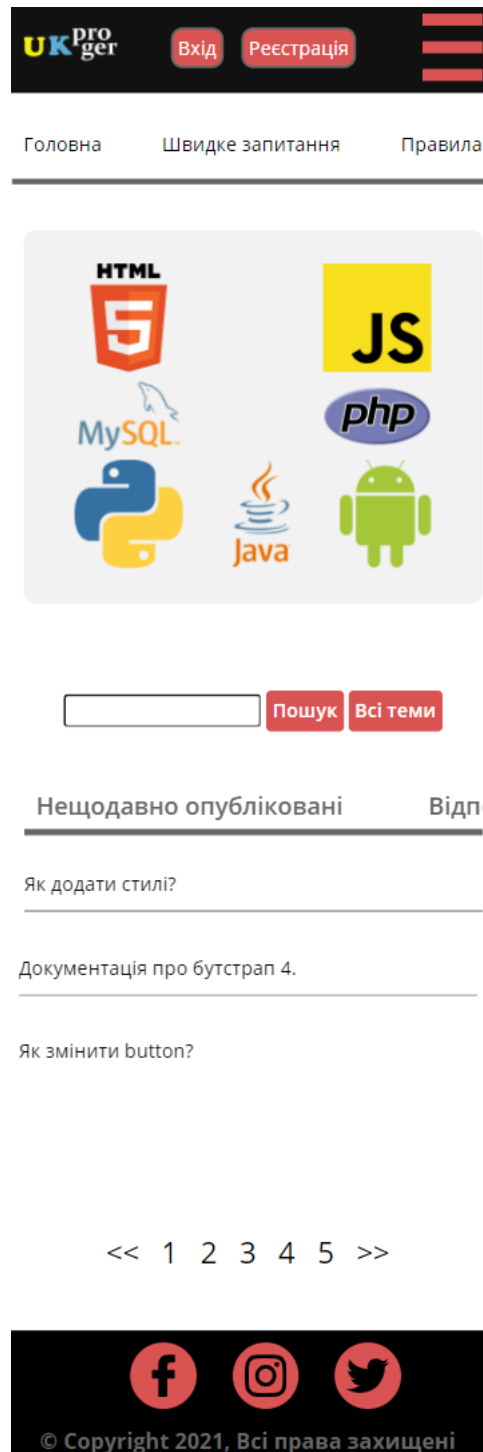


Рисунок 3.12 – Головна сторінка з телефону

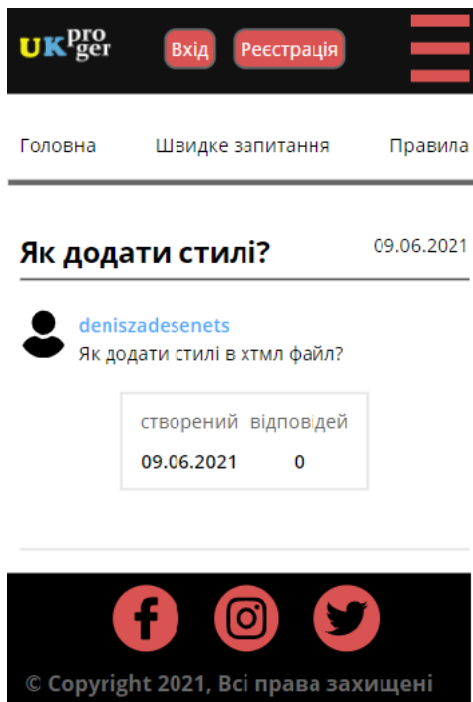


Рисунок 3.13 – Відкрита тема з телефону

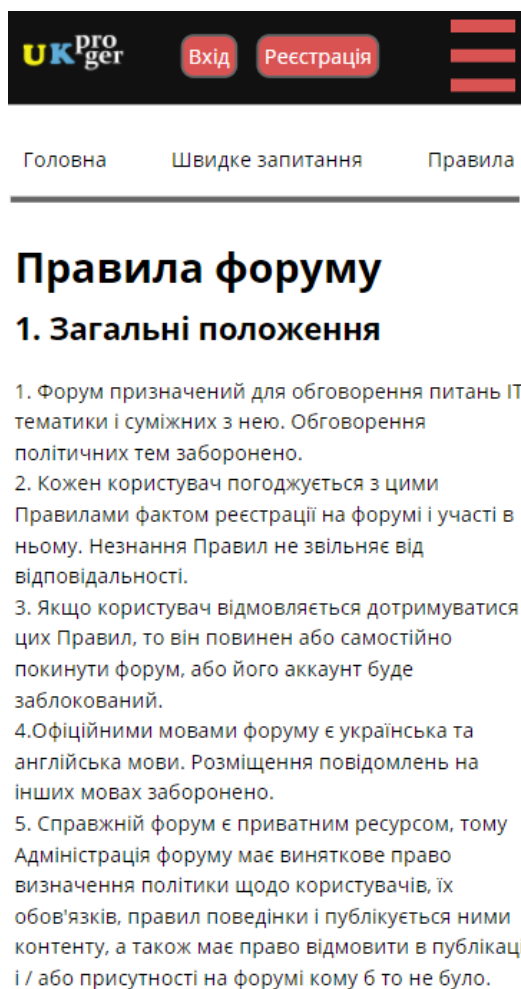


Рисунок 3.14 – Сторінка правил з телефону

ВИСНОВОК

В даній дипломній роботі було розроблено форум для програмних інженерів. Проведена робота з актуальності сайту, постановки задачі, аналізу методів програмування, вибору інструменту для реалізації.

Для розробки сайту була використана мова розмітки HTML5, функції були написані на PHP, стилі сайту були зроблені за допомогою стандартних стилів CSS3. Для реалізації баз даних була обрана реляційна система управління базами даних MySQL. Локальним сервером, на якому тестувався наш проект та велося управління ним був OpenServer.

В результаті було розроблено систему, яка відповідає поставленим задачам. Основний функціонал було перевірено то протестовано. Розроблений продукт легко піддається змінам уже існуючого та розробці нового функціоналу. Сайт є стабільним, безпечним, сучасним, адаптивним до мобільних пристроїв та готовим до використання.

СПИСОК ЛІТЕРАТУРИ

1. Денис Иванов и Федор Новиков Моделирование на UML Учебно-методическое пособие // СПб.: СПбГУ ИТМО, 2010 – С. 24 – 25.
2. Тахагхогхи С., Вильямс Хью Е. Руководство по MySQL // Пер. с англ. — М.: Русская редакция, 2007. – С. – 48.
3. Шпетний, І.О. Інформатика [Електронний ресурс] / І.О. Шпетний, С.І. Проценко, К.В. Тищенко. – Електронне вид. каф. Електроніки, загальної та прикладної фізики. - Суми: СумДУ, 2018. – С. 115.
4. Марейн Хавербеке Элегантный JavaScript, 2018. – С. 18 – 19.
5. Роб Майлз Желтая Книга Программирования на С#, 2016. – С. 10.
6. Дэвид Скляр Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов. : Пер. с англ. — СПб.: ООО “Альфа-книга”, 2017. – С. 26.
7. Майк МакГрат Программирование на Java для начинающих / [пер. с англ. М.А. Райтмана]. – Москва : Издательство «Э», 2016. – С. 10.
8. Майкл Фицджеральд Регулярные выражения: основы.: Пер. с англ. - М.: ООО «И.Д. Вильямс», 2015. – С. 39.
9. Хэл Фултон, Андрэ Арко Путь Ruby третье издание // ДМК Пресс, 2016. – С. 31.
10. Чаллавала Ш., Лакхатария Дж., Мехта Ч., Патель К. MySQL 8 для больших данных / пер. с англ. А. В. Логунова. – М.: ДМК Пресс, 2018. – С. 32.

Додаток А

Лістинг файлу conf.php

```
<?php
define ('DB_HOST', 'localhost');
define ('DB_USER', 'root');
define ('DB_PASSWORD', 'root');
define ('DB_NAME', 'progforum');
$Connect_db= mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD);
mysqli_select_db($Connect_db, DB_NAME);
?>
```

Лістинг файлу registration.php

```
<?php
session_start();
header('Content-type: text/html; charset=utf-8');
require_once('inc/conf.php');
>Login= $_POST['login'];
$Pass= $_POST['password'];
$Pass2= $_POST['password2'];
if (! filter_var($Login, FILTER_VALIDATE_EMAIL))
exit('Невірна пошта');
$result= mysqli_query($Connect_db, "SELECT 1 FROM `users` WHERE login =
'{$Login}' LIMIT 1 ");
if (mysqli_num_rows($Result) != 0)
exit('Ця пошта вже зареєстрована, поверніться назад');
if ($Pass != $Pass2)
exit('Паролі не співпадають, поверніться назад');
if (mb_strlen($Pass) < 6 or mb_strlen($Pass) > 9)
exit('Пароль повинен бути більше 6 символів і менше 9, поверніться назад');
if (mb_strlen($Pass2) < 6 or mb_strlen($Pass2) > 9)
exit('Пароль повинен бути більше 6 символів і менше 9, поверніться назад');
$Pass= md5($Pass);
mysqli_query($Connect_db, "INSERT IGNORE INTO users (login, pass) VALUES
('{$Login}', '{$Pass}') ");
$id= mysqli_insert_id($Connect_db);
$_SESSION['id']= $id;
$_SESSION['login']= $Login;
header('Location: index.php');
exit();
?>
```

Лістинг файлу authorize.php

```
<?php
session_start();
header('Content-type: text/html; charset=utf-8');
require_once('inc/conf.php');
>Login= $_POST['login'];
$Pass= $_POST['password'];
if (! filter_var($Login, FILTER_VALIDATE_EMAIL))
exit('Невірна пошта');
if (mb_strlen($Pass) < 6 or mb_strlen($Pass) > 9)
exit('Пароль повинен бути більше 6 символів і менше 9, поверніться назад');
$Pass= md5($Pass);
```

```

$Result= mysqli_query($Connect_db, "SELECT 1 FROM `users` WHERE login =
'{$Login}' AND pass = '{$Pass}' LIMIT 1 ");
if (mysqli_num_rows($Result) != 0) // аутентифікація пройдена
{
$Result2= mysqli_query($Connect_db, "SELECT rol FROM `users` WHERE login =
'{$Login}' AND pass = '{$Pass}' LIMIT 1 ");
$Arr= mysqli_fetch_assoc($Result2);
$_SESSION['login']= $Login;
$_SESSION['rol']= $Arr['rol'];
if ($_SESSION['rol'] == '')
{
header('Location: index.php');
exit();
if ($_SESSION['rol'] == 'admin')
{
header('Location: admin.php');
exit();
}
}
else // аутентифікація не пройдена
exit('логін або пароль невірні, натисніть назад');
?>

```

Лістинг файлу logout.php

```

<?php
session_start();
unset($_SESSION['login']);
header('Location: index.php');
?>

```

Лістинг файлу create_new_pass.php

```

<?php
require_once('inc/conf.php');
$Pass= $_POST['password'];
$Pass2= $_POST['password2'];
$Pass3= $_POST['password3'];
if ($Pass2 != $Pass3)
exit('Паролі не співпадають, поверніться назад');
if (mb_strlen($Pass) < 6 or mb_strlen($Pass) > 9)
exit('Пароль повинен бути більше 6 символів і менше 9, поверніться назад');
if (mb_strlen($Pass2) < 6 or mb_strlen($Pass2) > 9)
exit('Пароль повинен бути більше 6 символів і менше 9, поверніться назад');
$Pass= md5($Pass);
// Перевірка старого паролю
$Result= mysqli_query($Connect_db, "SELECT 1 FROM `users` WHERE pass =
'{$Pass}' LIMIT 1 ");
if (mysqli_num_rows($Result) != 0) // Перевірка пройдена
{
$Pass2= md5($Pass2);
mysqli_query($Connect_db, "UPDATE users SET pass = '{$Pass2}' WHERE
pass = '{$Pass}' LIMIT 1 "); // змінюємо пароль.
header('Location: new_pass.php');
exit();
}
else //Перевірка не пройдена
exit('Старий пароль невірний, поверніться назад');
?>

```

Лістинг файлу create_topic.php

```

<?php
session_start();
require_once('inc/conf.php');
$Login= $_SESSION['login'];
$title= $_POST['title'];
$text= $_POST['text'];
$date= date('d.m.Y');
mysqli_query($Connect_db, "INSERT INTO topic (title, text, login, date, views)
VALUES ('$title', '$text', '$Login', '$date', '0') ");
$id= mysqli_insert_id($Connect_db);
header('Location: topic.php?id=' . $id);
exit();
?>

```

Лістинг файлу comment.php

```

<?php
session_start();
require_once('inc/conf.php');
$id= $_POST['id'];
$Login= $_SESSION['login'];
$text= $_POST['text'];
$date= date('d.m.Y');
mysqli_query($Connect_db, "INSERT INTO comment (topic_id, login, text, date,
lice) VALUES ('$id', '$Login', '$text', '$date', '0') ");
header('Location: topic.php?id=' . $id);
exit();
?>

```

Лістинг файлу comment_like.php

```

<?php
session_start();
require_once('inc/conf.php');
$id= $_POST['id2'];
mysqli_query($Connect_db, "UPDATE comment SET lice = lice + 1 WHERE `id`
= '{$id}' LIMIT 1 ");
exit("ok");
?>

```

Лістинг файлу topic.php

```

<?php
require_once('header.php');
$id= $_GET['id'];
mysqli_query($Connect_db, "UPDATE topic SET views = views + 1 WHERE `id`
= '{$id}' LIMIT 1 ");
$result= mysqli_query($Connect_db, "SELECT * FROM topic WHERE id= '{$id}'
LIMIT 1 ");
$arr= mysqli_fetch_assoc($result);
$arrLogin= explode("@", $arr["login"]);
$title= $arr['title'];
$text= $arr['text'];
$date= $arr['date'];

```

```

$Result2= mysqli_query($Connect_db, "SELECT * FROM comment WHERE topic_id=
'{$id}' LIMIT 10 ");
$Count_comments= 0;
while (mysqli_fetch_assoc($Result2))
$Count_comments++;
$Result2->data_seek(0); // Перемістим вказівник на
перший рядок в $Result2
?>
<section class="topic">
  <div class="container">
    <div class="topic__inner">
      <div class="topic__theme">
        <div class="topic__theme-top">
          <h1 class="topic__title">
            <?php echo $Title; ?>
          </h1>
        </div>
        <div class="topic__theme-ques">
          
          <div class="topic__theme-info">
            <p class="topic__theme-author">
              <?php echo $ArrLogin[0]; ?>
            </p>
            <p class="topic__theme-time"> <?php echo $Date; ?> </p>
            <p class="topic__theme-text">
              <?php echo $Text; ?>
            </p>
          </div>
        </div>
        <div class="topic__theme-bottom">
          <div class="topic__theme-bottomBlock">
            <p class="topic__theme-bottomBlockTitle">
              створений
            </p>
            <p class="topic__theme-bottomBlockSubtitle">
              <?php echo $Date; ?>
            </p>
          </div>
          <div class="topic__theme-bottomBlock">
            <p class="topic__theme-bottomBlockTitle">
              відповідей
            </p>
            <p class="topic__theme-bottomBlockSubtitle">
              <?php echo $Count_comments; ?>
            </p>
          </div>
        </div>
      </div>
    </div>
    <?php
    while ($Arr2= mysqli_fetch_assoc($Result2))
    {
      $ArrLogin2= explode("@", $Arr2["login"]);
      if (isset($_SESSION['login']) )
      echo '
        <div class="topic__com">
          
          <div class="topic__theme-info">
            <p class="topic__theme-author">
              ' . $ArrLogin2[0]. '
            </p>
            <p class="topic__theme-time"> ' . $Arr2["date"]. ' </p>

```

```

        <p class="topic__theme-text">
            ' .\$Arr2["text"]. '
        </p>
        <div class="topic__theme-likeshare">
            <button class="topic__like" id=" ' .\$Arr2["id"].' " onclick="
like(this.id) " >  </button>
            &nbsp;&nbsp;&nbsp;' .\$Arr2["lice"]. '
        </div>
        </div>
    </div>
';
else
echo '
        <div class="topic__com">
            
            <div class="topic__theme-info">
                <p class="topic__theme-author">
                    ' .\$ArrLogin2[0]. '
                </p>
                <p class="topic__theme-time"> ' .\$Arr2["date"]. ' </p>
                <p class="topic__theme-text">
                    ' .\$Arr2["text"]. '
                </p>
                <div class="topic__theme-likeshare">
                    <button class="topic__like"> 
</button>
                    &nbsp;&nbsp;&nbsp;' .\$Arr2["lice"]. '
                </div>
            </div>
        </div>
';
}
?>
        </div>
    </div>
</section>
</main>
<?php
if (isset($_SESSION['login'])) )
echo '
<form action="comment.php" method="post" align="center">
<textarea name="text" type="text" placeholder="Коментар" class="textareal"
required></textarea> <br><br>
<input type="submit" value="Відповісти" class="question__button" />
<input name="id" value=" ' .\$id.' " type="hidden" />
</form>
';
?>
<br>
<script>
function like(id)
{
xhttp= new XMLHttpRequest();
xhttp.open('POST', 'comment_like.php', true);
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send('id2=' +id);
xhttp.onreadystatechange = function()
{
if (xhttp.readyState == 4 && xhttp.status == 200)
if (xhttp.responseText == "ok")

```

```

alert("Лайк поставлений");
else
alert(xhttp.responseText);
}
} // Кінець like(id)
</script>
<?php
require_once('footer.php');
?>

```

Лістинг файлу index.php

```

<?php
require_once('header.php');
if ( !isset($_GET["p"]) )
$_GET["p"]= 1;
$P= $_GET["p"];
$Col_Record_Na_Page= 3;
$Offset= ($P-1) * $Col_Record_Na_Page;
if (isset($_POST["reset"]) )
unset($_GET["find"]);
// Якщо з рядка пошуку щось прийшло, то шукаємо значення $ Find в поле заголовка
title.
if (isset($_GET["find"]) )
{
$Find= $_GET["find"];
$result= mysqli_query($Connect_db, "SELECT * FROM topic WHERE `title` LIKE
'%"$Find%" ORDER BY id DESC LIMIT $Offset, $Col_Record_Na_Page ");
}
else
$result= mysqli_query($Connect_db, "SELECT * FROM topic ORDER BY id DESC
LIMIT $Offset, $Col_Record_Na_Page ");
?>
<div class="lang">
<div class="container">
<div class="lang_inner">







</div>
</div>
</div>
<br><br><br>
<form action="index.php" method="GET" align="center" style="width: 100%;
height: auto; /*background: #F0F8FF*/ ">
<input type="text" name="find" class="search" />
<input type="submit" value="Пошук" class="input_search" />
<input type="submit" name="reset" value="Всі теми" class="input_search" />
<input type="hidden" name="p" value="<?php echo $P; ?>" >
</form>
<section class="recently">
<div class="container">
<div class="recently_inner">
<div class="recently_top">
<h6 class="recently_title">

```

```

        Нещодавно опубліковані
    </h6>
    <div class="recently_data">
        <p class="recently_data-item">
            Відповіді
        </p>
        <p class="recently_data-item">
            Переглядів
        <p class="recently_data-item">
            АКТИВНИЙ
        </p>
    </div>
</div>
<div class="recently_blocks">
<?php
while ($Arr= mysqli_fetch_assoc($Result))
{
$id= $Arr["id"];
$Result2= mysqli_query($Connect_db, "SELECT * FROM comment WHERE topic_id=
'{$id}' LIMIT 10 ");
$Count_comments= 0;
while (mysqli_fetch_assoc($Result2))
$Count_comments++;
echo '
        <div class="recently_block">

            <div class="recently_block-left">
                <a class="recently_block-title" href="topic.php?id='.$id.'"
">
                    '.$Arr["title"]. '
                </a>
            </div>
            <div class="recently_block-right">
                <p class="recently_block-item"> '.$Count_comments.' </p>
                <p class="recently_block-item"> '.$Arr["views"]. ' </p>
                <p class="recently_block-item"> '.$Arr["date"]. ' </p>
            </div>
        </div>

';
}
?>
        </div>
    </div>
</div>
</section>
</main>
<?php
$ArrNumPages[]= $P;
$Val= $P;
for ($i= 0; $i < 2; $i++)
{
$Val= $Val - 1;
if ($Val < 1)
break;
$ArrNumPages[]= $Val;
}
$Val2= $P;
$Count_ArrNumPages= count($ArrNumPages);

```

```

for ($i= 0; $i < 5 - $Count_ArrNumPages; $i++)
{
$Val2= $Val2 + 1;
$ArrNumPages[]= $Val2;
}
sort($ArrNumPages);
$Result3= mysqli_query($Connect_db, "SELECT FOUND_ROWS() FROM topic ");
$Row= ceil($Result3->num_rows / $Col_Record_Na_Page);
?>
<div class="pagination" align="center">
<a href="?p=1"> << </a>
<?php
if ( isset($Find) )
for ($i= 0; $i < count($ArrNumPages); $i++)
echo '
<a href="?find='.$Find.'&p='.$ArrNumPages[$i].'"> '.$ArrNumPages[$i].' </a>
';
if ( !isset($Find) )
for ($i= 0; $i < count($ArrNumPages); $i++)
echo '
<a href="?p='.$ArrNumPages[$i].'"> '.$ArrNumPages[$i].' </a>
';
?>
<a href="?p=<?php echo $Row; ?>"> >> </a>
</div>
<?php
require_once('footer.php');
?>

```

Лістинг файлу header.php

```

<?php
session_start();
require_once('inc/conf.php');
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>UKproger</title>
<link rel="stylesheet" href="css/style.min.css">
<link rel="preconnect" href="https://fonts.gstatic.com">
<link
href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@300;400;600;700&di
splay=swap" rel="stylesheet">
</head>
<body>
<div class="wrapper">
<header class="header">
<div class="container">
<div class="header__inner">
<div class="header__logo">


</div>
<div class="header__logreg">
<?php
if ( !isset($_SESSION['login']) )
echo '

```



```

<a href="#openModal2" class="header_log">Вхід</a>
<a href="#openModal" class="header_reg">Реєстрація</a>
';
else
{
$ArrLogin= explode("@", $_SESSION["login"]);
echo '
<div style="color: #ef555d; font-size: 25px; ">' . $ArrLogin[0]. ' </div>
';
}
?>

</div>
<div class="header_searchMenu">
<div class="header_menu"><span></span></div>
</div>
<div class="header_menu-list" id="header_menu_list">
<a href="logout.php" class="header_log">Вихід</a>
<a href="new_pass.php" class="header_log">Змінити пароль</a>
</div>
</div>
</header>
<main class="main">
<div class="menu">
<div class="container">
<ul class="menu_list">
<li class="menu_item"><a class="menu_link"
href="index.php">Головна</a></li>
<!-- <li class="menu_item"><a class="menu_link"
href="#">Новини</a></li> -->
<?php
if (isset($_SESSION['login'])) )
echo '
<li class="menu_item"><a class="menu_link" href="question.php"> Швидке
запитання</a></li>
';
else
echo '
<li class="menu_item"><a class="menu_link" onclick="alert(&quot;Спочатку
авторизуйтеся&quot;)" href=""> Швидке запитання</a></li>
';
?>
<li class="menu_item"><a class="menu_link"
href="page.php">Правила</a></li>
</ul>
</div>
</div>
<script>
var buttons= document.querySelectorAll('.header_menu');
Array.from(buttons).forEach(function(button)
{
button.addEventListener('click', function(e)
{
header_menu_list.classList.add("active");
})
});
</script>

```

Лістинг файлу footer.php

```
<?php
require_once('sub_footer_form_auth_regi.php');
?>
<footer class="footer">
  <div class="container">
    <h6 class="footer__copy">
      © Copyright 2021, Всi права захищенi
    </h6>
    <ul class="footer__social">
      <li class="footer__item"><a href="https://facebook.com/"
class="footer__link" target="_blank">
        
      </a></li>
      <li class="footer__item"><a href="https://instagram.com"
class="footer__link" target="_blank">
        
      </a></li>
      <li class="footer__item"><a href="https://twitter.com"
class="footer__link" target="_blank">
        
      </a></li>
    </ul>
  </div>
</footer>
</div>
<link rel="stylesheet" href="inc/css/form_comment.css" />
</body>
</html>
```