Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

\_Навчально науковий інститут інформаційних технологій та робототехніки\_

(повна назва факультету)

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

(повна назва кафедри)

**Пояснювальна записка**

**до дипломного проєкту (роботи)**

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

(освітньо-кваліфікаційний рівень)

на тему

Розроблення інформаційно-розважальної системи

(база даних та наповнення контенту)

Виконав: студент 4 курсу, групи 402-ТН

спеціальності

122 Комп’ютерні науки \_

(шифр і назва спеціальності)

Ціруліс Р.В. \_

(прізвище та ініціали)

Керівник Дмитренко Т.А. \_

(прізвище та ініціали)

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

(прізвище та ініціали)

Полтава – 2021 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

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

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

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

**на тему**

**«Розроблення інформаційно-розважальної системи  
(база даних та наповнення контенту)»**

**Студента групи 402-ТН Ціруліса Романа Вячеславовича**

Керівник роботи

кандидат технічних наук,

доцент Дмитренко Т.А.

Завідувач кафедри

кандидат технічних наук,

доцент Головко Г.В.

Полтава – 2021

**РЕФЕРАТ**

Кваліфікаційна робота бакалавра: 73 с., 29 рисунків, 3 додатки, 28 джерел.

**Об’єкт дослідження:** вебсайт, котрий ідентифікується як інформаційно-розважальна система.

**Мета роботи:** розроблення системи, функціонал якої надає змогу користувачам переглядати, створювати та розповсюджувати цікавий та розважальний контент, а також створення для неї бази даних.

**Методи:** проєктування роботи функціоналу вебсайту, розроблення бази даних з урахуванням потреб інформаційно-розважальної системи та її реалізація обраним засобами розробки.

**Ключові слова:** інформаційно-розважальна система, база даних, PHP, MySQL, Open Server, phpMyAdmin, публікація, коментування, голосування, сортування, контент, розповсюдження, посилання, підписка, теми.

**ANNOTATION**

Qualification work of bachelor`s degree: 73 pp., 29 figures, 3 appendices, 28 sources.

**Object of research:** a website that identifies as an infotainment system.

**The goal of the work:** to develop a system whose functionality allows users to view, create and share interesting and entertaining content, as well as create a database for it.

**Methods:** designing the work of the website functionality, developing a database tailored to the needs of the infotainment system and its implementation by the selected development tools.

**Keywords:** infotainment system, database, PHP, MySQL, Open Server, phpMyAdmin, publishing, commenting, voting, sorting, content, share, links, subscription, topics.

**ЗМІСТ**

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 6](#_Toc74257235)

[ВСТУП 7](#_Toc74257236)

[РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ   
ДОСЛІДЖЕННЯ 9](#_Toc74257237)

[1.1 Опис предметної області 9](#_Toc74257238)

[1.2 Огляд чинних інформаційно-розважальних систем 10](#_Toc74257239)

[1.3 Формулювання завдання 19](#_Toc74257240)

[РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНО-РОЗВАЖАЛЬНОЇ   
СИСТЕМИ 22](#_Toc74257241)

[2.1 Внутрішня структура сайту 22](#_Toc74257242)

[2.2 Моделювання роботи системи 24](#_Toc74257243)

[2.2.1 Діаграма варіантів використання. 24](#_Toc74257244)

[2.2.2 Діаграма діяльності. 26](#_Toc74257245)

[РОЗДІЛ 3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНО-РОЗВАЖАЛЬНОЇ СИСТЕМИ 29](#_Toc74257246)

[3.1 Вибір та обґрунтування використання інструментарію для бекенду 29](#_Toc74257247)

[3.1.1 Обрання мови PHP. 29](#_Toc74257248)

[3.1.2 Обрання фреймворку CodeIgniter. 30](#_Toc74257249)

[3.1.3 Обрання бази даних MySQL. 32](#_Toc74257250)

[3.2 Вибір та обґрунтування використання інструментарію для фронтенду 33](#_Toc74257251)

[3.2.1 Обрання мов HTML та CSS. 33](#_Toc74257252)

[3.2.2 Обрання мови JavaScript. 34](#_Toc74257253)

[3.3 Вибір та обґрунтування використання портативного локального сервера Open Server 36](#_Toc74257254)

[3.4 База даних 38](#_Toc74257255)

[3.4.1 Наповнення контенту. 41](#_Toc74257256)

[3.5 Опис функціоналу створеної системи 42](#_Toc74257257)

[РОЗДІЛ 4 ТЕСТУВАННЯ ІНФОРМАЦІЙНО-РОЗВАЖАЛЬНОЇ СИСТЕМИ 51](#_Toc74257258)

[4.1 Тест-план 51](#_Toc74257259)

[4.2 Тестування взаємодії з базою 52](#_Toc74257260)

[4.3 Тестування функціоналу 54](#_Toc74257261)

[4.4 Тестування роботи посилань 55](#_Toc74257262)

[4.5 Залиття на хостинг 57](#_Toc74257263)

[ВИСНОВКИ 58](#_Toc74257264)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 59](#_Toc74257265)

[ДОДАТОК А ДАМП БАЗИ ДАНИХ 62](#_Toc74257266)

[ДОДАТОК Б ЛІСТИНГ КОДУ З ПРИКЛАДОМ КОМАНД ДЛЯ ВЗАЄМОДІЇ  
З БАЗОЮ 69](#_Toc74257267)

[ДОДАТОК В ЛІСТИНГ КОДУ ДЛЯ ФОРМИ НАДСИЛАННЯ ПУБЛІКАЦІЇ 72](#_Toc74257268)

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

**UGC –** User-Generated Content, користувальницький контент.

**ІС** – інформаційна система.

**Пост** – синонім до слова «публікація».

**UML** – Unified Modeling Language, уніфікована мова моделювання.

**PHP** – Hypertext Preprocessor, мова програмування.

**HTML** – HyperText Markup Language, мова гіпертекстової розмітки.

**MVC** – Model-View-Controller, модель-представлення-контролер.

**СУБД** – система управління базами даних.

**БД** – база даних.

**Юзабіліті** – зручність і простота використання.

**SQL** – Structured Query Language, мова структурованих запитів.

**ODBC** – Open DataBase Connectivity, програмний інтерфейс доступу до БД.

**ПК** – персональний комп'ютер.

**CSS** – Cascading Style Sheets, каскадні таблиці стилів.

**JS** – JavaScript, мова програмування.

**AJAX** – Asynchronous Javascript and XML, асинхронний JavaScript і XML.

**FTP** – File Transfer Protocol, протокол передачі файлів по мережі.

**SCP** – Secure Copy, утиліта для копіювання файлів між комп'ютерами.

**CSV** – Comma-Separated Values, текстовий формат, призначений для представлення табличних даних.

**XML** – Extensible Markup Language, розширювана мова розмітки.

**ODS** – OpenDocument Spreadsheet, формат для електронних таблиць.

**MySQLi** – MySQL improved, розширення драйвера реляційних баз даних.

**ПЗ** – програмне забезпечення.

**URL** – Uniform Resource Locator, уніфікований покажчик ресурсу.

**SSL** – Secure Sockets Layer, криптографічний протокол захисту.

**ACID** – Atomicity, Consistency, Isolation, Durability, вимоги до транзакційної системи.

**ВСТУП**

В наш час, у зв'язку з розвитком технологій, Інтернет є найзручнішим способом обміну інформацією. Він має багато переваг. Наприклад, засоби масової інформації, які реалізують себе в Інтернеті, публікують свої матеріали набагато раніше, ніж виходять до друку газети або новинні телепередачі. Покоління, яке виросло в оточенні новітніх технологій, надає перевагу інтернет-простору як майданчику для спілкування, обміну інформацією та отримання новин.

Кожній людині для відволікання від повсякденної рутини потрібне місце, де можливо швидко та безплатно знайти відпочинок у вигляді розваг. І тоді на допомогу приходять безліч інформаційно-розважальних порталів. У них поєднуються два найбільш потрібних для сучасної молоді напрямки: інформація та разом з тим і забави. Згідно зі статистичними даними, сайти саме такого характеру мають більшу кількість постійних читачів, коментарів, відгуків і щоденних переглядів. До того ж опубліковані на їхньому майданчику тематичні матеріали викликають більший відгук в аудиторії.

Метою дипломного проєкту є розроблення інтерактивної й зручної інформаційно-розважальної системи де у кожного є можливість висловлювати свої думки та викладати в Інтернет креативну творчість без надмірних обмежень у будь-якому прояві – від звичайної дискусії до створення так званих «мемів», що дозволить людям весело та цікаво проводити свій вільний час. Досягти цього можливо шляхом дослідження та порівняння чинних схожих аналогів для виявлення їхніх найкращих особливостей й по можливості уникнення їхніх недоліків та додання своїх фішок.

У даних порталів обов’язково повинна бути система заохочування у вигляді рейтингу, котра прикрашає процес публікування креативної творчості, адже усім хочеться отримати похвалу від багатьох інших користувачів всесвітньої павутини, або ж поспілкуватися з ними та дізнатися їхні думки.

Також система рейтингу дозволяє сортувати різноманітний контент, а саме відсіювати порожнє та нецікаве, залишаючи у топі лише гідні пости для залучення й утримання уваги великої кількості аудиторії, тому і відпадає потреба у непомірному втручанні в інтереси користувачів та управлінню вручну всієї системи.

Завдяки відсутності обмежень формату контенту котрий публікується та його гнучкому розділенню на відповідні тематики, можливі напрями застосування системи різнобічні, наприклад:

* насамперед усі користувачі мають можливість поділитися чим завгодно, спитати про щось, або знайти що-небудь їм притаманне і цікаве тим самим розширюючи свій кругозір;
* засоби масової інформації спроможні повістувати свої новини й в такому випадку система виступає в ролі агрегатора новин;
* початківці з різних сфер, будь то музиканти, ютубери, художники та ін., можуть публікувати свою творчість для набрання популярності, відгуків та можливих зауважень і відповідних порад в подальшій роботі;
* різноманітні видавництва і журнали здатні оприлюднювати власні дослідження й статті та використовувати систему для зв’язку зі своєю аудиторією;
* різні компанії, бренди, виробники продукції й т.д. можуть використовувати систему для пошуку нової аудиторії шляхом реклами – причому мова йде як і про банальні банери, так і про більш спритно приховану, що маскується під меми й «персональні» пости.

**РОЗДІЛ 1  
АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ**

**1.1 Опис предметної області**

Людина влаштована таким чином, що їй вкрай необхідний відпочинок і розслаблення. Не завжди є можливість виділити час і гроші, щоб організувати собі якісне дозвілля. Однак не варто впадати у відчай, адже є гідна альтернатива. Йдеться про численні розважальні портали, де можна дізнатися масу захопливої, потрібної й корисної інформації. Тут не йдеться про сухі факти. Такі сайти пропонують своїм читачам лише такий контент, який не відрізняється нав'язливою зайвою серйозністю.

Інформаційно-розважальні портали реалізовані, щоб прикрашати людям вільні хвилини, веселити їх, дивувати та надихати на новітні звершення. Тут можливо знайти потішне відео, почитати захопливі статті, глянути фрагменти програм, взяти участь в цікавій дискусії, іншими словами все те, що власно здатне в короткий термін підняти людині настрій [1].

Розважальні портали вкрай виділяються на тлі інших ресурсів. Вони відрізняються яскравою графікою, великою кількістю анімацій і всіляких інтерактивних елементів.

Дані портали також можуть мати мобільні додатки з іграми в режимі онлайн, форумами або чатами. В їх рамках є можливість ознайомитися з галереями фотографій або відеозаписами певного змісту.

Розважальні портали можуть бути доповненими стрічками новин, каталогами й так далі. Тобто, в одному місці зосереджено все те, що може виявити інтерес для сучасної людини зі здоровими амбіціями. Контент таких сайтів спрямований на те, щоб захопити читача. Тут описуються різні яскраві події шоу-бізнесу, світу моди, мистецтва і пропонуються до вивчення останні досягнення. В рамках розважальних сайтів завжди можна прочитати гороскоп, знайти нове блюдо або дозвілля, що здатне викликати інтерес. На сторінках таких порталів зібрано все, що спроможне зарядити людину позитивом і змінити її настрій в кращу сторону.

З доступом в Інтернет сьогодні проблем немає, тому ніщо не заважає людині зануритися у світ цікавих, пізнавальних і веселих новин. Пізнавальні портали сміливо можна назвати ефективними «таблетками» від нудьги.

Зазвичай такі сайти наповнюються інформацією за допомогою **UGC** – це який завгодно тип контенту (відео, фото, аудіо, GIF-файли, меми та повідомлення в блогах), створений та розповсюджений користувачами в Інтернеті [2].

**1.2 Огляд чинних інформаційно-розважальних систем**

Для подальшої розробки своєї інформаційно-розважальної системи потрібно зробити огляд наявних популярних аналогів, визначити всі ключові характеристики ресурсів, їх особливості й недоліки.

Далі наведено деякі з розглянутих сайтів:

1. **Reddit** – це англомовна платформа з корисним контентом, яка поєднує в собі соцмережу, форум і портал одночасно. Саме цим пояснюється популярність Reddit серед людей. Прості користувачі з великою довірою ставляться до тих джерел, де такі ж люди з такими ж проблемами діляться своїми порадами, досвідом та рекомендаціями.

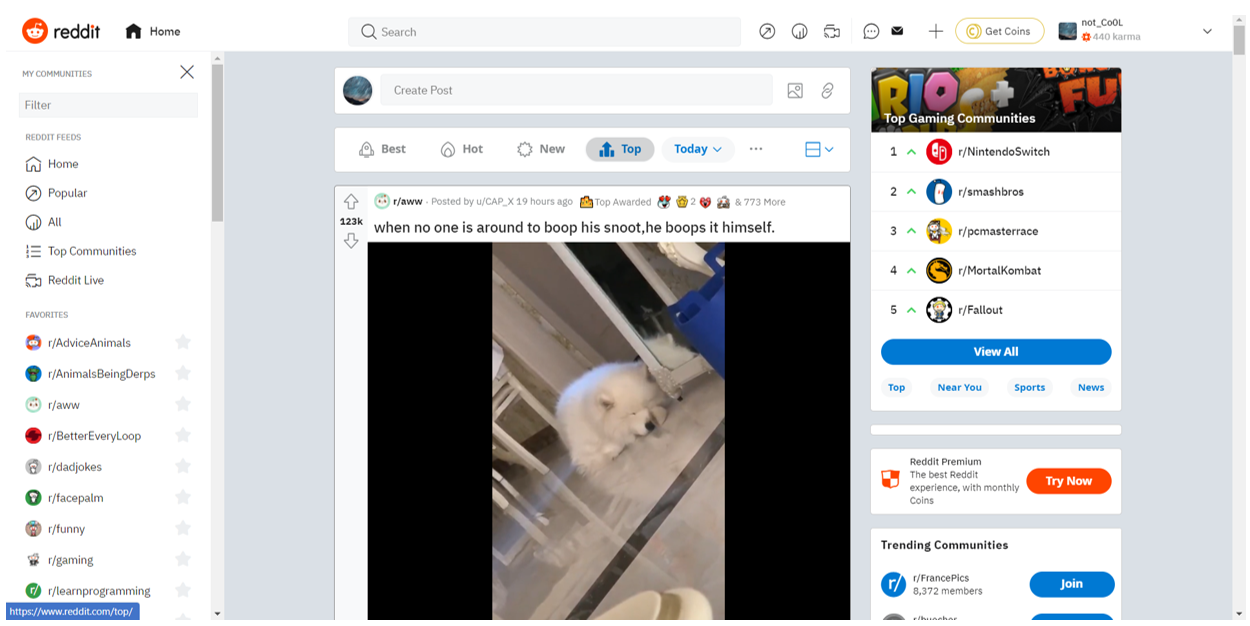
Reddit для зручності використання розподілений на тематичні розділи – subreddits (сабреддіти). Користувачі самотужки викладають на сайті різноманітний контент, котрий інші реддітори мають змогу прокоментувати й похвалити чи навпаки «замінусувати».

Алгоритм системи зображає на головній сторінці Reddit публікації, які набули найбільше голосів. Голоси теж враховуються як плюси у карму автора публікації, а внаслідок рівня карми залежить, чи може користувач публікувати або коментувати в окремих сабреддітах [3].

За даними Similarweb, Reddit відвідує трохи менш ніж 1,5 мільярда користувачів щомісяця. Завдяки такій популярності та відносній анонімності, Reddit неодноразово ставав джерелом різного роду витоків зі «внутрішньої кухні» корпорацій, урядів та закритих спільнот [4].

Спочатку Reddit був написаний мовою програмування Common Lisp, але ще у 2005 році переписаний на Python – для більшої гнучкості й доступу до численних бібліотек [5].

Про дизайн сайту Reddit можна сказати що він легкий та зрозумілий, не має перевантаження текстом. Панель навігації знаходиться у лівому кутку, в яке входять фіди, цікаві користувачеві сабреддіти та різні налаштування, а у правому кутку – актуальні топи сабреддітів за різними категоріями (рис. 1.1). Переглядати Reddit можливо в трьох режимах: «Класичному», схожому на стару стрічку сайту, «Картковому», який нагадує дизайн Facebook, і «Компактному», де блоки розташовані більш щільно один до одного.

  
Рисунок 1.1 – Інформаційно-розважальна система «Reddit»

Особливості даного ресурсу:

* зручне поділення контенту на тематичні розділи – subreddits (сабреддіти);
* можливість швидко відстежувати будь-які новини;
* легка реєстрація – за допомогою електронної пошти (потрібно задати ім’я, пароль та підтвердити пошту), або за допомогою облікового запису Google та Apple;
* на головній сторінці сайту зображаються тільки ті пости, на теми яких ви підписалися;
* наявні зручні фільтри, що дозволяють переглядати новини за популярністю, датою виходу і так далі;
* можливість поспілкуватися з іншими людьми на зацікавленні теми коментуючи пости, або у персональному чаті;
* великий спектр налаштувань (є можливість налаштувати економію трафіку);
* проста система рейтингу, так звана «карма»;
* наявна особиста платна система винагород, котрими можливо нагороджувати авторів гідних публікацій або влучних коментарів, та підняти собі й автору таким чином карму;
* простий процес створення свого сабреддіту, але є дві вимоги – обліковому запису має бути не менше ніж 30 днів та наявність 500+ карми;
* зручний та широкий простір для рекламодавців та брендів;
* відомий розділ Ask me Anything (запитайте у мене що завгодно) у якому на питання користувачів відповідають всесвітньовідомі люди;
* двофакторна аутентифікація для захисту облікових записів користувачів.

Недоліки:

* занадто велика кількість правил (навіть у кожного сабреддіта є свої правила які розміщені в боковому меню, за порушення яких видається бан);
* для відключення реклами необхідно придбати платну підписку Reddit Premium (також відкриває ряд додаткових можливостей, як розширення кастомізації свого аватару).

1. **9GAG** – один з найпопулярніших джерел короткометражних розваг в Інтернеті, команда сервісу якого базується в Гонконзі, з безліччю мемів, GIF-файлів та іншого вірусного контенту, що публікується щодня, використовуючи користувальницький контент, який є загальний для всієї спільноти.

За останні п’ять років популярність 9GAG стрімко зросла: між своїм вебсайтом та мобільним додатком 9GAG зараз залучає понад 150 мільйонів унікальних відвідувачів щомісяця [6].

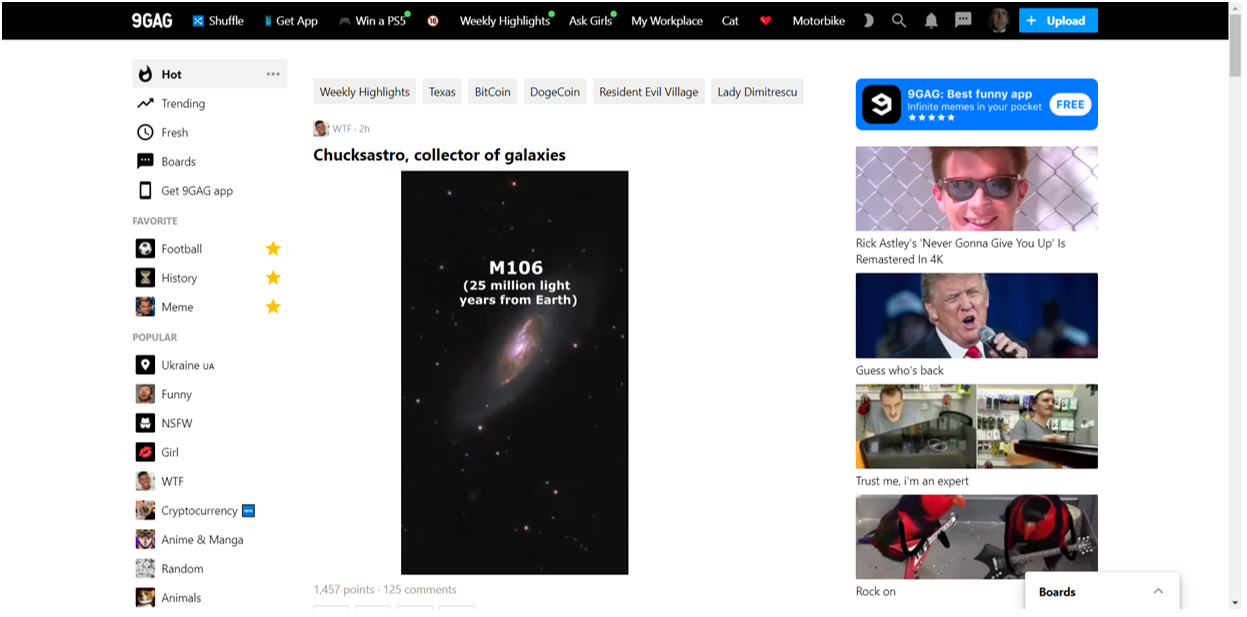
Більшу частину трафіку сайт отримує з США та Німеччини. За даними аналітичного сервісу SimilarWeb, на ці дві країни припадає понад 17% переходів на вебсайт. У «п'ятірку лідерів» також входять Франція, Великобританія та Бразилія [7].

Для організації постів на сайті існують три різні категорії сторінок:

* перша називається «Гарячі» і показує найпопулярніші пости, а це означає, що вони зібрали найбільше голосів користувачів 9gag;
* друга називається «Трендові» і збирає пости, які випереджають процес голосування, тобто ці пости все ще потребують великої кількості голосів, щоб бути опублікованими на першій сторінці, але вже набули певної популярності;
* останній тип сторінки називається «Свіжі» і представляє пости, які безпосередньо публікуються користувачами 9gag: ця сторінка є свого роду розплідником для постів, оскільки вони представляються на їх перший процес голосування.

Голоси, а значить і популярність, ділять пости на категорії й тому на гарячу сторінку часто пропускаються проблемні публікації (наприклад, з використанням явного контенту). Іншими словами, саме більшість користувачів дозволяє або не дозволяє культурним цінностям, специфічним для інтернет-спільнот, з'являтися або поширюватися. З цієї точки зору система, підтримувана 9gag, є цікавою лабораторією для обміну і поширення специфічних культурних тенденцій, зі своїми власними перевагами та обмеженнями.

Дизайн даного ресурсу трохи нагадує Reddit, зроблений у чорно-білій стилістиці (рис. 1.2), у лівому кутку знаходяться фільтри по популярності та новизні наявного контенту, потім список обраних улюблених розділів користувача та найпопулярніші розділи, а у правому кутку – випадкові трендові пости за місяць, які завжди оновлюються. Цей сайт легко сприймається новачками, адже тут не має нічого зайвого і тому йде швидке звикання до даного дизайну.

  
Рисунок 1.2 – Інформаційно-розважальна система «9GAG»

Серед переваг можна виділити:

* простий та відповідний дизайн для даної категорії сайтів, з відсутністю перевантаженням тексту;
* можливість сортувати пости за популярністю, рейтингом, новизною або категоріями;
* не складна реєстрація, за допомогою облікового запису Facebook та Google, або поштовою скринькою з наступним її підтвердженням;
* наявні основні необхідні налаштування акаунту, з можливістю завантажити свій аватар або обрати з випадково пропонованих;
* можливість спілкуватися коментуючи пости, або у наявному колективному чаті на присутні теми;
* зручне створення постів, де завантаження фото або відео відбувається через посилання, або на пряму з пристрою, також можливо робити меми відразу на сайті за допомогою представлених інструментів;
* лояльна система рейтингу для постів та коментарів (один користувач може додати або відняти один бал рейтингу);
* можливість швидкого перемикання на наступні та попередні пости за допомогою клавіш стрілок на клавіатурі.

Недоліки:

* не зручний вбудований відеоплеєр (не має можливості перемотувати або ставити на паузу відео та регулювати рівень гучності);
* не має можливості створювати як персональний, так і колективний чат.

1. **Fishki.net** – один з найпопулярніших розважальних порталів на російському медіаринку, який наповнюють новинами адміністраторами та самими користувачами. Крім росіян, даний ресурс читають українці й латиші, казахи та ізраїльтяни [8].

На сьогодні, за даними Similarweb, сайт Fishki.net набирає 45 мільйонів відвідувань на місяць. В окремі дні на фішки заходить по два мільйони унікальних відвідувачів. У день модерацію проходять до тисячі постів-публікацій.

Fishki.net цікаві не тільки любителям чистого гумору, а і цінителям більш серйозних матеріалів, в тому числі історичних, автомобільних, політичних, обговорень відносин між статями, кулінарних рецептів, проблем медицини та освіти, всього, що роблять умільці своїми руками, а також питань культури та мистецтва [9].

Новини на головній сторінці формуються за особливим алгоритмом, який зображає в топі найбільш актуальні пости на цей момент. Сайт має 3 основні розділи: «Гаряче» – це головна сторінка, «Краще» і «Нове».

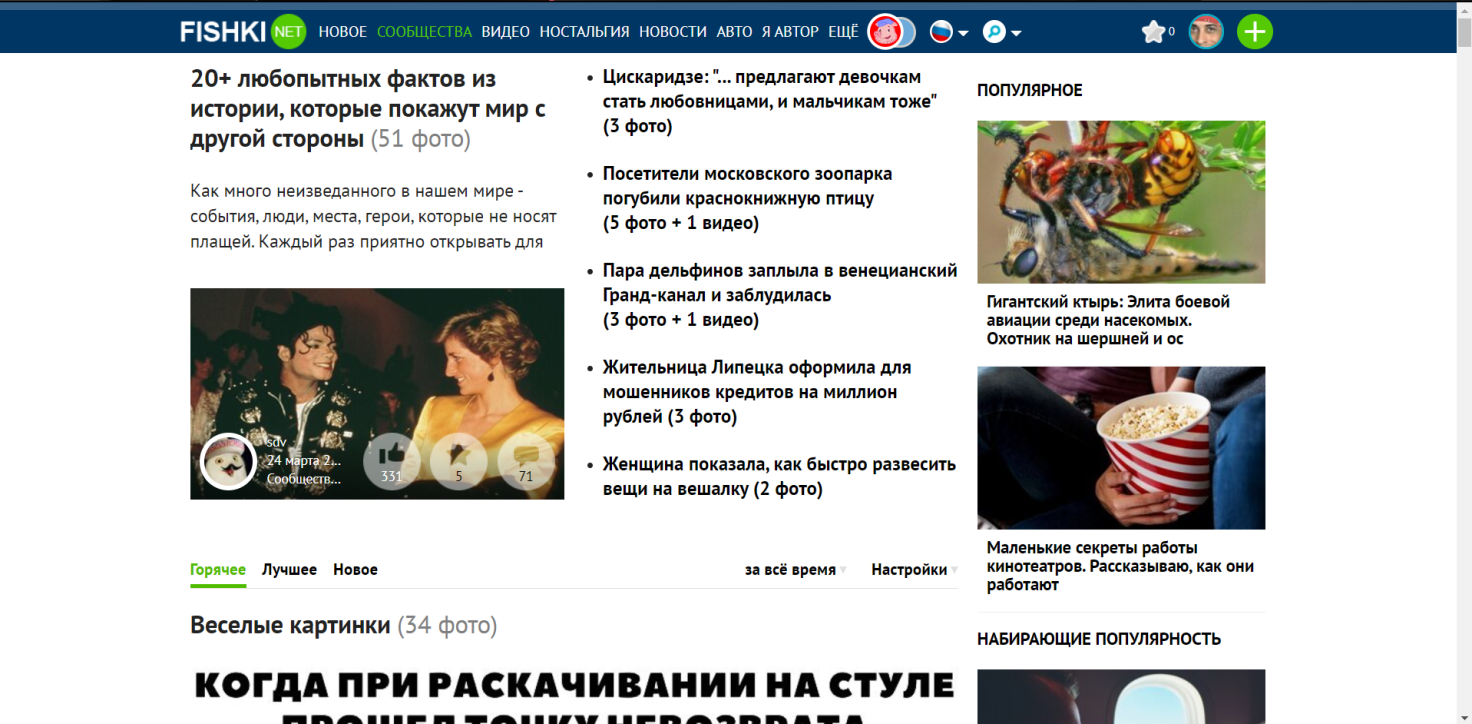
Даний сайт наявний двома мовами – російською та англійською, при чому дизайн англійською мовою злегка спрощений, а саме не має у правому кутку стрічки популярних постів, топів товариств за різний період (за день, неділю, місяць та за весь період) та популярних тегів.

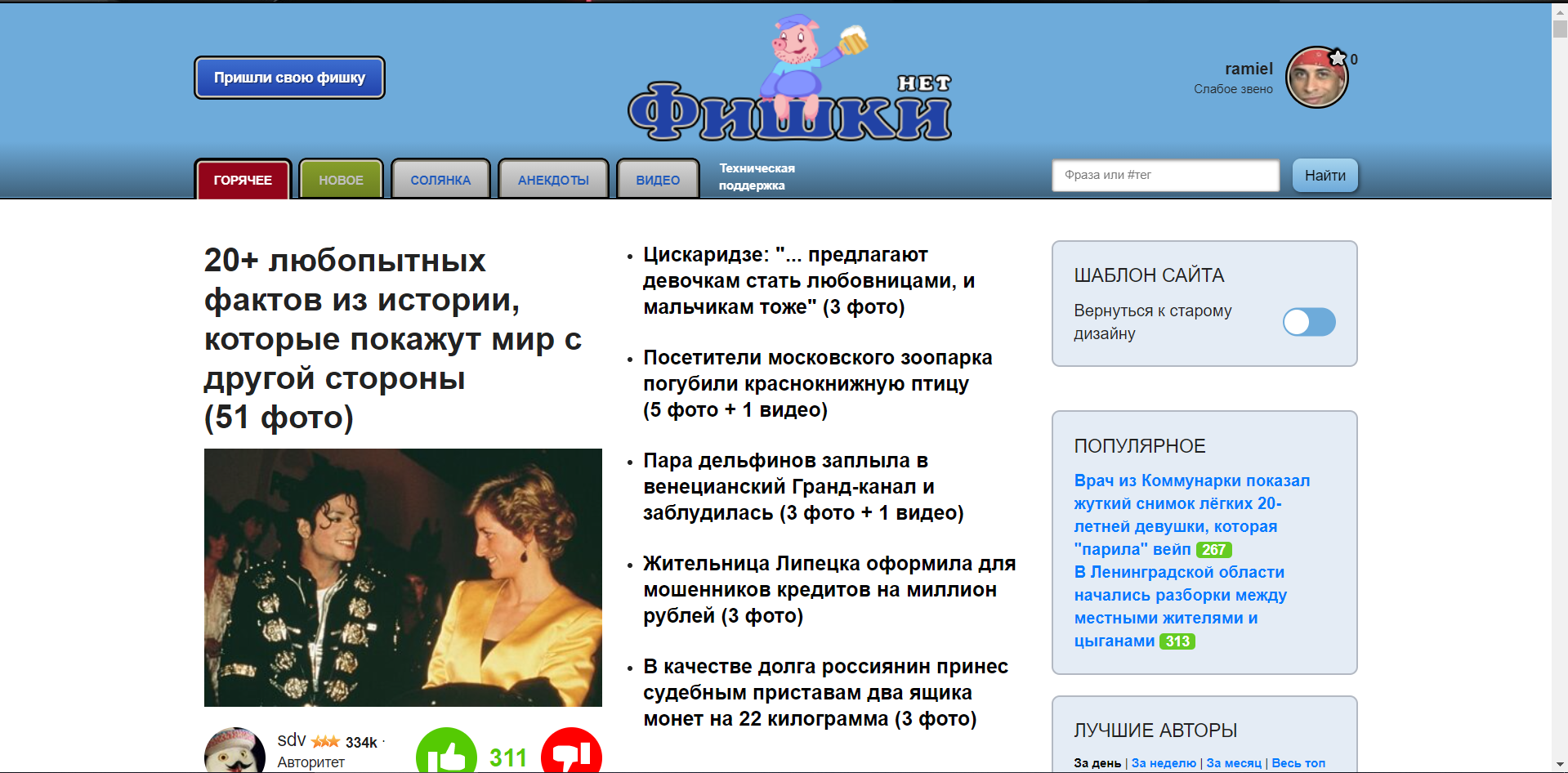
Фішки мають два шаблони свого сайту, а саме старий (рис. 1.3) та новий (рис. 1.4) дизайн, які мають ряд відмінностей:

* нові фішки отримали назву RealFishki.net – це закрита спільнота, весь контент якого користувацький і не перетинається з головними фішками. Цензура мінімальна, це досягнуто внаслідок відключення реклами на Реалфішках і тому скарг на контент від рекламодавців не буде;
* хедер сайту у новому дизайні набагато більший від старої версії через безглуздий та великий логотип, наслідками якого стало виникнення помітного порожнього простору;
* при гортанні стрічки постів у нових фішках зліва з’являється бургер меню сайту та стрілки вгору і вниз, а зверху по центру – лого сайту яке перекидає на головну сторінку. В старих фішках з’являється лише стрілка вгору, при використанні якої вона заміняється на стрілку вниз до того ж місця де знаходився користувач;
* у новому дизайні набагато менше категорій поділу постів ніж у старому, а саме наявні лише п’ять, котрі знаходять в хедері сайту. Пошук бажаних постів доводиться роботи по тегах.

При використанні обох версій сайту більш привабливим та зручним користувачі можуть знайти старий дизайн через відсутність зайвих елементів та більшого функціонала.

Сам дизайн хоч і простий, у звичайних фішок, але, на перший погляд, не досить зрозумілий, наприклад, на головній сторінці, де знаходяться актуальні новини є відчуття перевантаження текстом, чого не можна сказати про інші сторінки, де всі пости йдуть один за одним в низ.

  
Рисунок 1.3 – Старий дизайн інформаційно-розважальної системи «Fishki.net»

  
Рисунок 1.4 – Новий дизайн інформаційно-розважальної системи «Fishki.net»

На сайті Fishki.net система рейтингу (карма) дає деякі обмеження, отримати її можливо якщо інші зареєстровані користувачі оцінять створені пости (+/- 1 карма) або коментарі (+/- 0.5 карма). Для чого вона потрібна:

* карма більша за 10 – з’являється можливість прикріплювати картинку до коментаря;
* карма більша за 500 – створювані пости будуть розміщуватися в «Новому» без премодерації;
* публікація потрапить у «Гаряче», якщо пройшла модерацію і набрала більше ніж 10 лайків;
* якщо пост набере -15 карми, то він буде прихований з «Нового», але буде доступний в профілі автора;
* щоб розмістити пост тільки з відео, карма повинна бути не менше 1000 (через багато спроб рекламувати свої YouTube канали).

Всі пости проходять постмодерацію. Якщо пост був прихований з «Нового», модератор вкаже причину в коментарях.

Переваги:

* проста реєстрація, де необхідно придумати логін для входу, нікнейм, пароль, ввести електронну скриньку та підтвердити її;
* зручне поділення контенту на відповідні розділи та з можливістю його сортувати;
* наявні необхідні налаштування для профілю, конфіденційності, стрічки новин та коментарів;
* у профілю присутня функція збереження історії перегляду постів, а саме усі пости які було оцінено, прокоментовано або збережено в обрані;
* можливість швидкого пошуку контенту по тегах;
* просте створення власного поста (перед публікацією є попередній перегляд);
* можливість створювати або підписуватися на товариства;
* оцінити публікацію зареєстрований користувач зможе лише після її перегляду, в іншому випадку з’явиться повідомлення з помилкою;
* наявні цікаві розважальні тести на різні тематики (створені лише адміністраторами).

Недоліки:

* з першого погляду незручний дизайн сайту для новачків;
* наявно багато реклами, що відвертає увагу;
* відсутність персональних та колективних чатів.

**1.3 Формулювання завдання**

На основі детального порівняльного аналізу чинних аналогів інформаційно-розважальних систем, таких як Reddit, 9GAG та Fishki.net було розкрито їхні основні особливості.

На підставі вищевикладеного визначаємо необхідність розробки нової інформаційно-розважальної системи, яка стане брати до уваги введенні рішення і найкращі напрацювання, які властиві даному виду ресурсів. Ця система безумовно повинна, по-перше, виконувати такі функції:

* проста реєстрація користувачів;
* публікація нових постів (з посиланням на стороні джерела або текстом);
* повноцінна інтерактивна інтеграція з іншими популярними системами;
* створення користувачами сайту тем при публікуванні постів;
* розділення постів на теми;
* залишення коментарів в пості;
* відповідання на коментарі;
* перегляд постів опублікованих на сайті;
* голосування за рейтинг поста, а саме «за» чи «проти»;
* пошук постів за допомогою ключових слів в назві, описі та темі публікації;
* різноманітне сортування постів з урахуванням критеріїв: кількість голосів, коментарів та дати створення;
* різноманітне сортування коментарів з урахуванням критеріїв: кількість голосів, відповідей та дати створення;
* підписування на теми, з можливістю відстежувати їх;
* зручне розповсюдження посилань на публікації та коментарі з сайту за допомогою кнопки «Поділитися»;
* перегляд залишених користувачами коментарів;
* голосування за рейтинг коментаря – «за» чи «проти»;
* спостереження за інтересами користувачів, а саме перегляд тем з найбільшою кількістю постів та підписників;
* збереження всієї активності користувачів в їхньому профілі;
* спостереження активності користувача в його профілі;
* збереження постів в секцію «улюблене» котра знаходиться в власному профілі;
* збереження коментарів в секцію «улюблене»;
* посилання скарг на пости, коментарі й теми;
* надходження повідомлень користувачам, якщо на їх пости або коментарі відповідають чи голосують за них;
* комфортне оформлення тексту постів, коментарів й додання різного мультимедіа або смайлів до них;
* можливість змінити пароль в налаштуваннях власного профілю;
* можливість відновлення паролю при його втраті;
* швидке здобуття допоміжної інформації для зручної роботи системи, котра може бути потрібна користувачу;
* інтуїтивно зрозумілий інтерфейс, який відповідає сучасним стандартам розробки вебсайту.

По-друге, можна відзначити ще не першочергові та двоїсті функції, котрі можливо буде додати в майбутньому в систему:

* заохочування у більшому постінгу публікацій шляхом зачислення голосів у власний профіль користувачів у вигляді карми, проте це призведе до нерівності аудиторії;
* додання чатів, як і особистих (між двома особами) та й колективних (між багатьма особами), котрі користувачі можуть створювати самі, проте це більш притаманно соціальним мережам та може відбити бажання спілкуватися в коментарях;
* широка кастомізація особистих профілів та створених тем, наприклад, створення або завантаження власного аватара, також більш притаманно соціальним мережам;
* можливість швидкої реєстрації за допомогою різних сторонніх сервісів, наприклад Google Account та Facebook, не є невідкладною функцією, оскільки звичайної реєстрації більш ніж достатньо.

Як правило, в інформаційно-розважальних системах виділяються 3 види користувачів, а саме:

* Гість – має обмежені можливості у використанні системи, в основному перегляд усієї інформації на сайті, розповсюдження та може пройти реєстрацію для перетворення на рядового користувача;
* Користувач – рядовий користувач, який володіє усіма основними функціями системи, котрі були описані вище, відсутній доступ лише до адміністрування;
* Адміністратор – має всі можливості рядового користувача та доступ до адмінської панелі через власний профіль, де може виконувати належні йому функції, як перегляд скарг, видалення контенту й користувачів та редагування контенту.

Таким чином, було зроблене уявлення майбутньої розроблюваної інформаційно-розважальної системи, шляхом визначення основних вимог до її функціоналу та виявлення видів користувачів.

**РОЗДІЛ 2  
ПРОЄКТУВАННЯ ІНФОРМАЦІЙНО-РОЗВАЖАЛЬНОЇ СИСТЕМИ**

**2.1 Внутрішня структура сайту**

Внутрішня структура сайту – це схема розміщення його основних елементів (сторінок, записів, розділів) відносно один одного. Вона зображає не тільки будову сайту, але і логічну зв'язку всіх його складових [22]. Неправильне компонування розділів призводить до того, що частина інформації залишається «за кадром» як для пошукових систем, так і для користувачів.

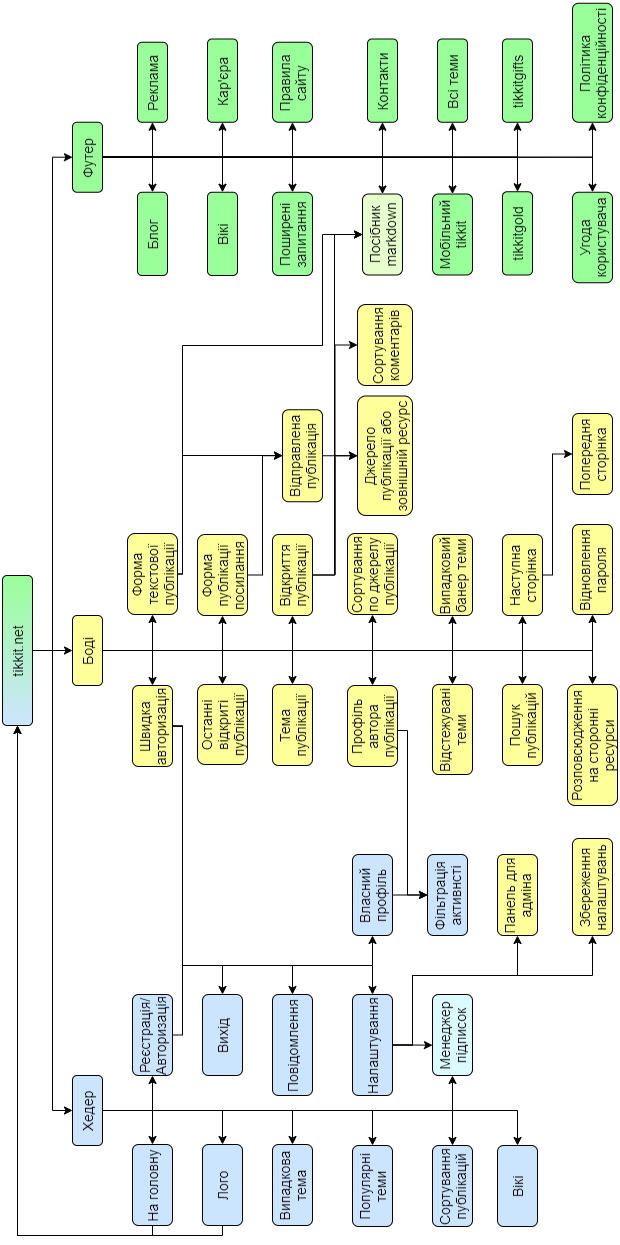
Грамотна побудова ресурсу необхідна для досягнення наступних цілей:

1. Простота розробки. Якщо чітко визначити контури проєкту і заздалегідь встановити зв'язки між об'єктами, часу на написання коду витрачається менше, а готовий вебпродукт виходить якіснішим.
2. Можливість модернізації. Обдумана структура сайту – міцна основа для подальшого розвитку проєкту. При наявності оптимально побудованої схеми ресурсу нарощування кількості сторінок не позначиться на якості просування.
3. Юзабіліті. Зовнішня організація сторінок повинна бути такою, щоб користувачеві було зручно дивитися матеріали, а на пошук потрібної інформації витрачалося мінімум часу.
4. Високе ранжування. Найбільше клієнтів приверне портал, що займає високі позиції в пошукових системах. Увійти ж в топ без логічної організації контенту можна лише завдяки щасливому випадку і то ненадовго.

Оскільки розроблюваний ресурс динамічний, то на рис. 2.1 проілюстровані основні його переходи, а для зручності перегляду структури сайту було зроблене розділення на «хедер», «боді» та «футер».

Динамічні сайти використовують серверні мови для обробки запитів під час створення сторінки в режимі реального часу. Це означає, що динамічні вебсайти взаємодіють з користувачем на сайті, і зображають контент в залежності від його дій або налаштувань [23].

Рисунок 2.1 – Внутрішня структура системи



**2.2 Моделювання роботи системи**

При моделюванні роботи системи виникає необхідність не тільки представити процес зміни її станів, а й деталізувати особливості алгоритмічної та логічної реалізації виконуваних системою операцій [25].

Створення інформаційної системи – складний процес, який можливо представити як поетапний спуск від загальної концепції майбутньої ІС, через розуміння її логічної структури до найбільш детальних моделей, що описують фізичну реалізацію.

Для опису роботи системи використовувалась уніфікована мова моделювання (UML), котра була розроблена з метою забезпечити єдину візуальну мову з багатою семантикою і розгорнутим синтаксисом для проєктування й впровадження програмних систем зі складною структурою та комплексною поведінкою.

В цілому, діаграми UML описують межі, структуру та поведінку як всієї системи, так і окремих об'єктів в її складі.

У якості графічного представлення для виділення основних функцій системи застосовується діаграма варіантів використання.

Діаграма варіантів використання дає уявлення що повинна робити система. А для відповіді на питання як саме вона може працювати використовують діаграму діяльності.

**2.2.1 Діаграма варіантів використання.** Це тип поведінкової діаграми, котрі часто використовується для аналізу різних систем. Вони дозволяють візуалізувати різні типи ролей в системі й те, як ці ролі саме взаємодіють з системою [24].

В залежності від вимог можливо використовувати ці діаграми різними способами. Нижче наведено декілька способів їх використання:

* ідентифікація функцій і як з ними взаємодіють ролі – основне призначення діаграм сценаріїв використання;
* для представлення системи на високому рівні – особливо корисно при поданні керівникам або зацікавленим сторонам. Можна виділити ролі, які взаємодіють з системою, і функціональні можливості, що надаються системою, не заглиблюючись у внутрішню роботу системи;
* ідентифікація внутрішніх і зовнішніх факторів – це може здатися простим, але у великих складних проєктах система може бути ідентифікована як зовнішня роль в іншому випадку використання.

Актори, прецеденти й відносини – це основні елементи нотації діаграм варіантів використання.

Актором називається будь-який об'єкт, суб'єкт або система, що взаємодіє з модельованою системою ззовні.

Прецедент описує, що робить система, але ніяк не уточнює, як вона це робить. Також діаграма прецедентів не зображає послідовність, в якій будуть виконуватися варіанти використання.

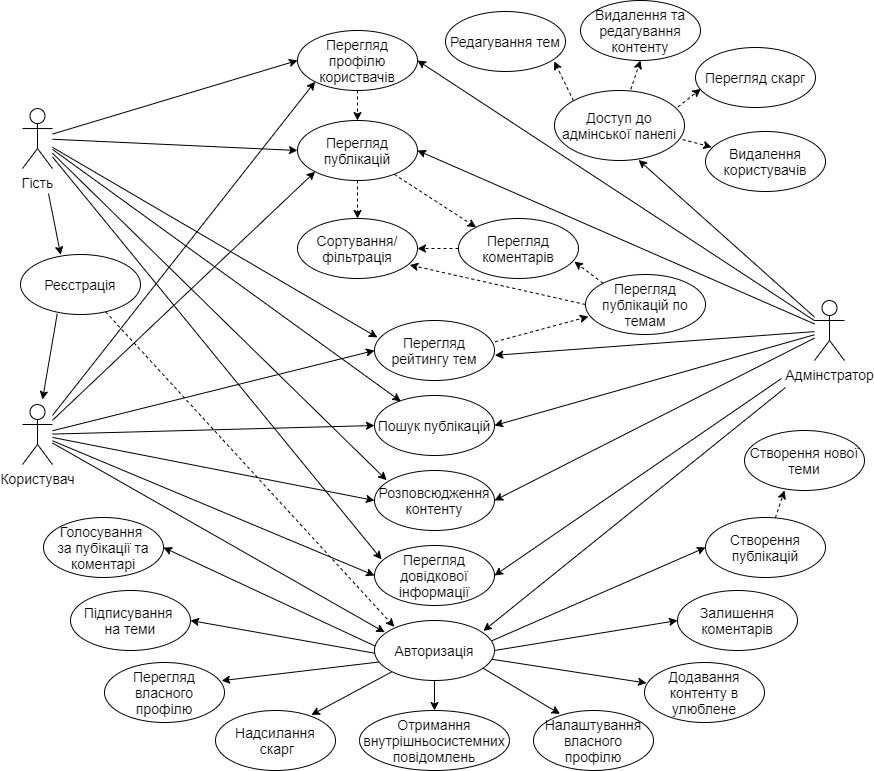
Відношення – це семантичний зв'язок між окремими елементами моделі. Поміж акторами та прецедентами діаграми варіантів використання можуть існувати різного роду відносини, які показують, що екземпляри дійових осіб і варіантів використання взаємодіють один з одним.

В основному на діаграмах прецедентів використовуються наступні типи відносин:

* асоціації – структурне відношення, яке показує, що об'єкт певним чином пов'язаний з іншим об'єктом;
* включення – говорить про те, що вихідний прецедент явним чином охоплює поведінку цільового;
* розширення – показує, що цільовий прецедент розширює поведінку вихідного;
* узагальнення – відношення між загальною сутністю і її конкретним втіленням.

На рис. 2.2 проілюстрована головна діаграма варіантів використання розроблюваної інформаційно-розважальної системи.

Рисунок 2.2 – Діаграма варіантів використання



Отже, на представленій діаграмі було зображено всі межі системи (актори), її основна функціональність (прецеденти), котрі були виведені в формулюванні завдання та відношення між ними.

**2.2.2 Діаграма діяльності.** Це діаграма поведінки UML, яка показує потік управління або потік об'єктів з акцентом на послідовність і умови потоку [26].

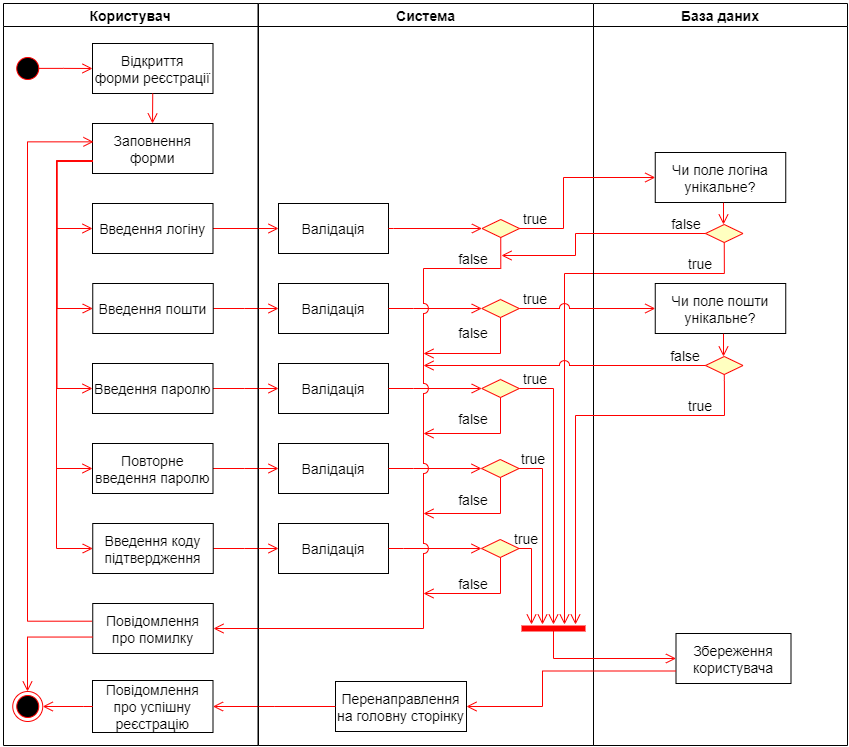
Діяльність є специфікацією параметризованої поведінки в формі координованої послідовності підпорядкованих одиниць, індивідуальними елементами яких є дії.

Дія являє собою елементарну одиницю специфікації поведінки, яка не може бути далі декомпонована у формі діяльності.

Дуга діяльності є абстрактним класом для спрямованих з'єднань між двома вузлами діяльності.

Для повноцінного використання користувачем всіх можливостей розроблюваної системи обійтися без реєстрації неможливо, тому на рис. 2.3 зображено саме цей процес.

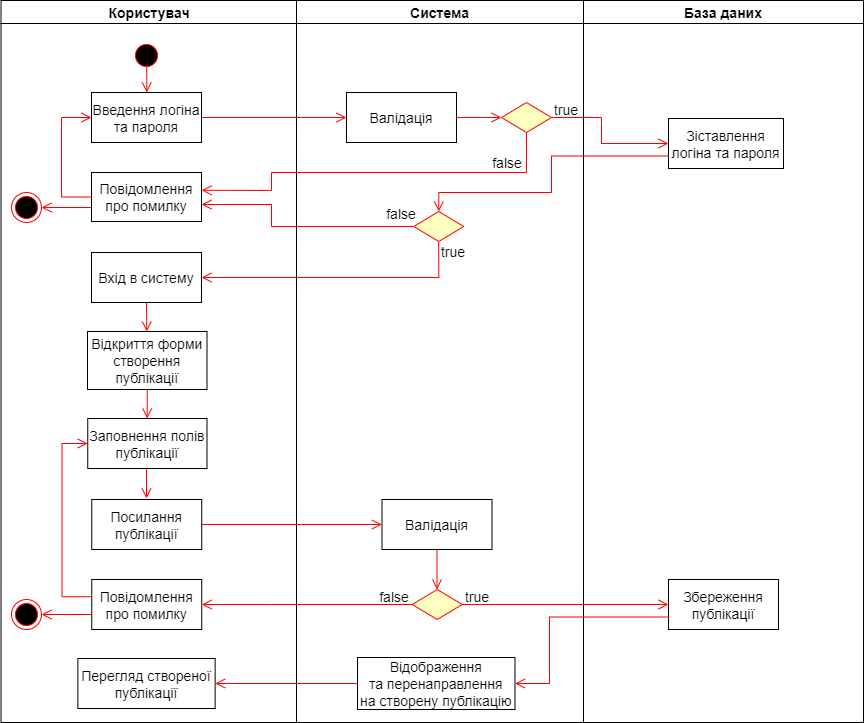
Рисунок 2.3 – Діаграма діяльності для зображення процесу реєстрації користувача



Як видно з діаграми, для проходження реєстрації користувачу необхідно пройти валідацію всіх рядків форми, а саме правильно ввести дані відповідно до стандарту або вимог. А для логіну та пошти йде додаткова валідація – перевірка унікальності в базі даних, для уникнення однакових імен та повторень електронних адрес.

Безумовно найголовніша функція будь-якої інформаційно-розважальної системи – це публікування користувачами постів, без неї ця система не існувала б і не набирала великої популярності (рис. 2.4).

Рисунок 2.4 – Діаграма діяльності для зображення процесу створення публікації користувачем



Діаграма демонструє, що для публікації користувачу потрібно спочатку авторизуватися в системі, тобто пройти валідацію системи на зіставлення вхідних даних з базою, далі відкрити саму форму публікації, заповнити її та відправити на валідацію, а головна перевірка при публікації – це проходження коду підтвердження, щоб уникнути нападів ботів.

**РОЗДІЛ 3  
РОЗРОБЛЕННЯ ІНФОРМАЦІЙНО-РОЗВАЖАЛЬНОЇ СИСТЕМИ**

**3.1 Вибір та обґрунтування використання інструментарію для бекенду**

**3.1.1 Обрання мови PHP.** PHP – це мова сценаріїв, створена для комунікації на бекенді, тобто вона може обробляти різні функції на стороні сервера, такі як збір даних форми, управління файлами на сервері, зміна баз даних і багато іншого [10].

Основні переваги мови PHP:

* широко використовується для розробки абсолютно різних платформ, будь то, інтернет-магазин, блог, соціальна мережа і т.д. Згідно зі статистикою 79% всіх сайтів використовують PHP;
* невисокий поріг входження. Мову легко вчити, оскільки вона має відмінну документацію з прикладами, що описує функції. Синтаксис мови входить в групу C-подібних, а це дає можливість швидше зрозуміти логіку в коді додатків на Java, C#, C++ і т.д.;
* PHP – частково об'єктноорієнтована мова. Це дає можливість повторно використовувати код, що економить час і сили в процесі розробки, також існує безліч фреймворків PHP: Symfony, CodeIgniter, Laravel, Joomla, WordPress та ін.;
* PHP часто вибирають для створення кросплатформних додатків, через її гнучкість і універсальність. Вона сумісна з усіма основними платформами: від Windows, Unix і Linux до MacOS; підтримує більшість серверів (включаючи Apache, Microsoft IIS, Netscape, iPlanet та ін.) і понад 20 баз даних (серед яких – MySQL, MongoDB, PostgreSQL та ін.);
* сервіси на PHP добре масштабуються і документуються. Це дозволяє запускати різні модулі один за іншим, вносячи по ходу необхідні поправки;
* безпроблемна підтримка й апдейт. Проєкти на PHP легко і швидко підлаштовуються під нові додатки, які з'являються на ринку, а також під нові бізнес-вимоги;
* універсальність PHP робить його популярним серед хостинг-провайдерів. Будь-який надійний провайдер підтримує PHP і надає хостингові послуги для сайтів на ньому;
* PHP – це проєкт з відкритим кодом, тому його можна використовувати безплатно.

Хоча PHP і вважається скриптовою мовою загального призначення, найбільш широко вона використовується у веброзробці. Це обумовлено однією з її видатних особливостей – можливістю вбудовування в файл HTML.

Припустимо, якщо не хочеться, щоб люди бачили вихідний код, можна приховати його за допомогою цієї мови сценаріїв. Достатньо написати код у файлі PHP, вставити його в HTML, і люди ніколи не дізнаються про оригінальний зміст.

**3.1.2 Обрання фреймворку CodeIgniter.** CodeIgniter – це багатофункціональний фреймворк написаний на PHP, за допомогою якого можна реалізувати повноцінні вебсистеми, сайти й додатки [11].

У Codeigniter компоненти завантажуються і процедури виконуються тільки за запитом, а не глобально. Система не робить ніяких припущень щодо того, що може знадобитися крім мінімальних основних ресурсів, тому система за замовчуванням дуже легка.

Компоненти фреймворку слабо пов'язані між собою і не залежать один від одного, а чим менше компонентів залежить один від одного, тим більш гнучкою і багаторазової стає система.

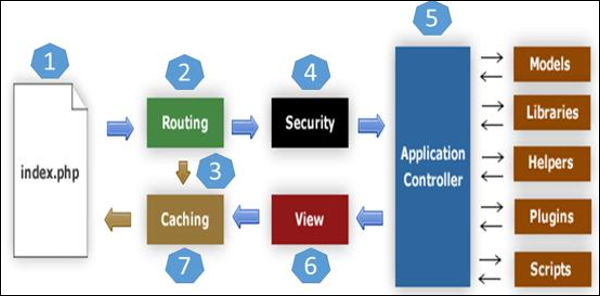
Хоча CodeIgniter працює досить швидко, обсяг динамічної інформації, що зображається на сторінках, буде прямо залежати від використовуваних ресурсів сервера, пам'яті й циклів обробки, які впливають на швидкість завантаження сторінок.

Тому CodeIgniter дозволяє кешувати сторінки для досягнення максимальної продуктивності, за допомогою вбудованого компонента кешування.

Переваги CodeIgniter:

* відмінна документація та англомовна спільнота;
* висока продуктивність фреймворку;
* невеликий розмір фреймворку;
* надає легкі та прості рішення для розробки;
* підходить для швидкої розробки невеликих сайтів і вебдодатків;
* структура фреймворку не вимагає строгих правил кодування;
* не вимагає складної настройки, майже нульова конфігурація;
* MVC-архітектура вебдодатків;
* слабка зв'язаність компонентів;
* безліч бібліотек і помічників які можливо приєднати.

Розглянемо архітектуру фреймворку CodeIgniter (рис. 3.1).

  
Рисунок 3.1 – Архітектура програми Codeigniter

1. Всякий раз, коли запит приходить в CodeIgniter, він спочатку переходить на сторінку index.php.
2. На другому етапі маршрутизація вирішує, чи передавати цей запит кроку 3 для кешування або передавати цей запит кроку 4 для перевірки безпеки.
3. Якщо запитувана сторінка вже знаходиться в режимі кешування, служба маршрутизації передасть запит етапу 3, а відповідь повернеться до користувача.
4. Якщо запитувана сторінка не існує в кешуванні, то служба маршрутизації передає запитувану сторінку на крок 4 для перевірки безпеки.
5. Перед передачею запиту в «Application Controller» перевіряється безпека представлених даних. Після перевірки безпеки контролер додатків завантажує необхідні моделі, бібліотеки, помічники, плагіни й скрипти та передає їх у «View».
6. «View» зображує сторінку з доступними даними й передає її для кешування. Оскільки запитувана сторінка раніше не кешувалася, тому на цей раз вона буде кешована в «Caching», щоб швидко обробити цю сторінку для майбутніх запитів.

**3.1.3 Обрання бази даних MySQL.** MySQL – це одна з найзнаменитіших й розповсюджених СУБД в мережі Інтернет. Вона не створена для роботи з величезним обсягом інформації, проте її використання чудово підходить для вебсайтів, як і малих, так й вельми великих.

MySQL виділяться чудовою швидкодією, надійністю та еластичністю. При її використанні, зазвичай, проблеми не з'являються. В постачанні PHP автоматично включається підтримка сервера MySQL [12].

MySQL володіє значною кількістю переваг перед іншими системами:

* одна з найшвидших баз даних серед наявних на сучасному ринку;
* високопродуктивна і відносно проста у використанні, яку значно простіше інсталювати й адмініструвати, ніж багато інших великих систем;
* поширюється безплатно для домашнього використання;
* розуміє команди SQL, підтримує інтерфейс ODBC, протокол інтерфейсу з базами даних, розроблений компанією Microsoft;
* сервер дозволяє підключатися одночасно необмеженій кількості користувачів. Доступ до сервера можна здійснити в режимі онлайн за допомогою різних інтерфейсів, що дозволяють вводити запити та переглядати отримані результати: це програми-клієнти, що працюють з командним рядком, веббраузери, програмні інтерфейси для мов C, Perl, Java, PHP і Python. Так що можна використовувати як готове клієнтське програмне забезпечення, так і створювати своє власне;
* призначена для роботи в мережі, може бути доступна через Інтернет, при цьому забезпечена розвиненою системою захисту від несанкціонованого доступу;
* відмінно працює як під управлінням різних версій UNIX, так і під управлінням інших систем: Windows і OS/2. При цьому система працює як на потужних серверах, так і на домашніх ПК;
* дистрибутив СУБД MySQL можна отримати, скориставшись веббраузером. Що найважливіше – можна отримати й вихідний код та внести в нього корективи.

Таким чином, дана СУБД приваблює своєю надійністю і вільним поширенням. Якщо справа стосується веброзробки, то MySQL практично завжди буде найкращим варіантом (з урахуванням потреб, специфіки проєкту та економічної доцільності).

**3.2 Вибір та обґрунтування використання інструментарію для фронтенду**

**3.2.1 Обрання мов HTML та CSS.** HTML і CSS – це фундаментальні технології, без знання яких не обійтися будь-якому веброзробнику. За допомогою мови гіпертекстової розмітки HTML створюється розмітка (каркас) кожної інтернет-сторінки. Потім мова стилів CSS перетворює сайт і надає йому привабливий і ефектний зовнішній вигляд [13].

HTML на відміну від інших технологій – незамінний компонент для всіх вебсторінок. Це необхідний базис, але одним базисом обійтися не вийде. Такі сторінки не впишуться в сучасні стандарти сайтобудування і навряд чи вразять користувачів зовнішнім виглядом.

Мова розмітки HTML відповідає не тільки за структуру сайту, але і за її зовнішній вигляд. Це зручно для сайтів з однією сторінкою, але якщо сторінок буде багато, і стануть з'являтися розділи з вкладками, знадобиться інше рішення. Для таких проєктів підійде мова CSS.

CSS – своєрідне доповнення до мови HTML. CSS являє собою формальну мову опису зовнішнього вигляду документа і відповідає за візуальну складову, тобто за дизайн. Цією мовою задаються всі важливі параметри – колірна гамма елементів, розмір шрифтів, відступи, відстані й т.д.

У каскадної таблиці є одна незаперечна перевага – вона спрощує роботу дизайнерів і програмістів шляхом принципово іншого підходу до верстки. Сайти створюються простіше: досить один раз задати стиль і застосовувати до всіх сторінок відразу. Якщо буде необхідність змінити одну каскадну таблицю, всі сторінки, котрі її використовували зміняться автоматично. Код стає простіше, кількість повторюваних елементів в ньому зменшується: при використанні одного лише HTML такого ефекту отримати не можливо.

**3.2.2 Обрання мови JavaScript.** JavaScript – це кроссплатформенна об'єктноорієнтована мова сценаріїв, що використовується для створення інтерактивних вебсторінок (наприклад, використання складної анімації, створення натискуваних кнопок, меню що спливає і т.п.). Програми цією мовою називаються скриптами. Вони можуть бути написані прямо в вебсторінці HTML та запускатися автоматично при її завантаженні [14].

Можливості JavaScript у веброзробці практично необмежені. Все, що можливо собі уявити на сайті, можна реалізувати за допомогою JS. А у зв'язці з технологією AJAX він взагалі може творити чудеса.

AJAX, якщо спрощено, це технологія, яка дозволяє оновлювати частину інформації на сторінці, не перезавантажуючи сторінку цілком. Завдяки цьому значно економиться трафік і, отже, збільшується швидкість роботи сайтів.

Якщо проводити аналогії, то HTML створює на сайті розмітку, CSS відповідає за її оформлення, а JavaScript – за «пожвавлення» всіх елементів.

Ось лише деякі із можливостей JS:

* додавання всілякої анімації – від падучих крапель дощу до автоматично перегортання каталогу;
* створювання подій у відповідь на будь-які дії користувача – клік миші, наведення покажчика на потрібну область, натискання клавіші, прокручування і т.д.;
* перевіряння на відповідність вимогам введених користувачем даних у формі, наприклад, формат номера телефону або електронної пошти;
* витягування та аналізування інформації про пристрій користувача – назву браузера, тип пристрою і т.п.;
* створювання та змінювання HTML-тегів;
* внесення змін в стилі CSS;
* оброблення даних cookie.

Це лише базові можливості, якими аж ніяк не вичерпується JavaScript. В цілому, на JS можна створити який завгодно сайт, з самим хитромудрим і складним функціоналом.

До плюсів можна віднести:

* створення програми можна в будь-якому текстовому редакторі, включаючи Microsoft Word і навіть Блокнот;
* не дуже складний синтаксис, який під силу вивчити навіть новачкам (за умови, якщо докласти достатньо зусиль);
* широкі можливості при створенні найрізноманітніших сайтів;
* JavaScript «читається» будь-яким сучасним браузером – як десктопним, так і мобільним;
* він постійно вдосконалюється;
* скрипти легко інтегруються в HTML-код;
* на ринку праці багато фахівців, які знають JS, також багато вакансій для таких розробників.

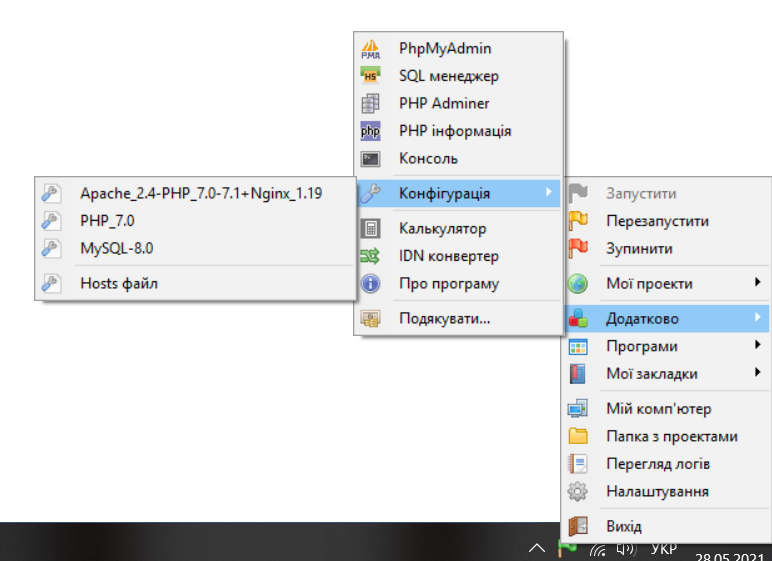
JavaScript зараз одна з найпоширеніших мов у світі. В індексі TIOBE вона займає 7 місце, а за кількістю репозиторіїв на GitHub і популярності за результатами опитування Stack Overflow вона зовсім знаходиться на першому місці.

**3.3 Вибір та обґрунтування використання портативного локального сервера Open Server**

У разі необхідності запуску сайту на домашньому комп'ютері (наприклад під час тестування), користувачі звертаються за допомогою до програм, що розгортаються на локальному сервері. Одним з кращих локальних серверів є Open Server, настройка якого навіть у недосвідченого користувача не займе багато часу.

Перевага цього рішення перед найближчими аналогами полягає в грамотній реалізації системи управління доданими доменами, а також у присутності графічного інтерфейсу (рис. 3.2). Це дозволить користувачу зовсім забути про використання «командного рядка» для управління власним сервером.

Open Server – це безплатна і вільно розповсюджувана програма для веброзробників, яка містить пакет серверних програмних компонентів, за її допомогою можна зробити аналог Linux-серверів під Windows, і без проблем запускати сайти, наприклад, написані на мові PHP.

  
Рисунок 3.2 – Графічний інтерфейс програми Open Server

Відкритий сервер дуже простий у використанні, не потрібно налаштовувати сервери Apache і Nginx або БД MySQL для запуску програми.

Основні можливості Open Server:

* невимогливий до системних вимог пристрою;
* сервер швидко запускається та зупиняється;
* включення й відключення запису логів;
* можливість керувати сервером через консоль;
* підтримка профілів налаштувань;
* велика кількість інших додаткових утиліт, таких як HeidiSQL, Adminer, RockMongo, phpMyAdmin та інші;
* зручний перегляд логів усіх компонентів;
* вибір HTTP, MySQL і PHP модулів в абиякому поєднанні;
* швидкий доступ до доменів та шаблонів конфігурації модулів;
* файлова система FTP-сервера;
* великомовний інтерфейс (наявна українська мова);
* відсутня установка (портативний), може працювати на флешці, що дає змогу швидко та легко переносити свої проєкти в повному обсязі з одного комп'ютера на інший;
* працює як на локальному, мережевому так і на зовнішньому IP адресі;
* підтримка SSL з відсутністю необхідності налаштування;
* просте створення домену через створювання папки;
* можливість створювати кириличні домени та аліаси (доменні синоніми).

Програма представлена в 3 версіях: Basic, Premium та Ultimate. Відмінність у них полягає в чисельності допоміжних утиліт для роботи. У першій версії наявний лише базовий набір для запуску сервера. Преміум версія має основні серверні програми та допоміжні утиліти, такі як: Git, ImageMagick, MongoDB, Rockmongo, PostgreSQL і phpPgAdmin. Повна версія має багату кількість допоміжних легко поширюваних програм для роботи з графікою, різноманітні текстові й HTML-редактори, програми для роботи з Інтернетом, менеджери завантажень, клієнти SCP і т.д., при чому є можливість самотуж обирати що з цього докачати.

Відкритий сервер має можливість працювати в обмеженому режимі в ситуації, якщо відсутня змога внести зміни в хост файл, коли права доступу обмежені чи заблоковані брандмауером. В такому разі для сайтів буде наявна тільки одна адреса локального хоста – 127.0.0.0.1.

**3.4 База даних**

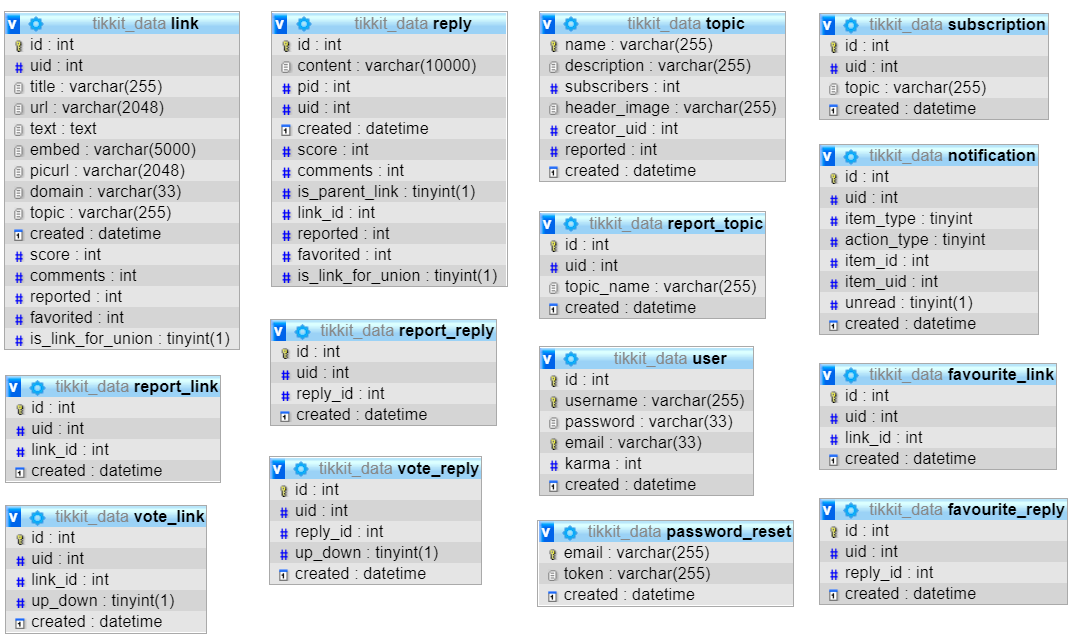
База даних побудована за допомогою програми phpMyAdmin (рис. 3.3), котра допомагає в зручній графічній формі, без потреби введення SQL-запитів, управляти БД на сервері MySQL.

Вебдодаток phpMyAdmin – універсальний і досить потужний інструмент написаний на мові PHP для управління базами даних, використовується розробниками, адміністраторами, а також звичайними користувачами – власниками сайтів. Наданий phpMyAdmin вебінтерфейс дозволяє отримувати доступ до управління базою даних з будь-якого браузера в будь-якій операційній системі [18].

Найчастіше необхідно швидко відредагувати поля таблиці БД. В цьому випадку написання SQL-запитів навіває тугу. Або необхідно швидко створити, видалити або очистити базу – phpMyAdmin дозволить це зробити в лічені секунди, буквально за кілька кліків.

В phpMyAdmin наявні всі необхідні функції MySQL, також є можливість задавати кодування, трансформувати дані в будь-який формат, імпорту та експорту БД (Додаток А) в різних форматах: SQL, CSV, XML, ODS та інші.

PhpMyAdmin має необхідні інструменти для адміністрування, а саме створення користувачів і управління їх правами, також можна змінити адресу до самого phpMyAdmin для захисту.

  
Рисунок 3.3 – Структура бази даних системи

Спроєктована база даних складається з 14 таблиць:

1. link – таблиця котра містить всю інформацію про публікацію: поля-ідентифікатори для самих публікацій та їх авторів, поля заголовків публікацій, посилань на сторонні джерела, текстів публікацій, ембедів джерел публікацій, посилань на рисунки сторонніх джерел, доменів джерел публікацій, тем публікацій, дат створення публікацій, поля-лічильники для рейтингів публікацій, коментарів, скарг і уподобань та поле для відмінності публікації від коментаря в профілі;
2. report\_link – таблиця для збереження скарг користувачів, вона містить в собі: поля-ідентифікатори скарг, їх відправників, публікацій та поле дати посилання;
3. vote\_link – таблиця, яка містить записи про голосування користувачів за публікації (поля-ідентифікатори голосів, їх відправників, публікацій й поля значень голосів та дат створення);
4. reply – таблиця, яка містить всю інформацію про коментар: поля-ідентифікатори для самих коментарів, авторів, публікацій та їх батьківських елементів, тобто id публікації або коментаря, поля для змістів коментарів, дат створення, визначень чи коментарі є батьківськими елементами, відмінності коментарів від публікацій в профілі, поля-лічильники для рейтингів коментарів, відповідей на коментарі, скарг й уподобань;
5. report\_reply – таблиця для збереження скарг на коментарі, вона містить в собі: поля-ідентифікатори скарг, їх відправників, коментарів та поле дати посилання;
6. vote\_reply – таблиця, яка містить записи про голосування користувачів за коментарі (поля-ідентифікатори голосів, їх відправників, коментарів й поля значень голосів та дат створення);
7. topic – таблиця переліку тем публікацій, котра містить: поля-ідентифікатори для назв тем і їх створювачів, поля-лічильники для кількості підписників тем й скарг, поля описів, шляхів до рисунка теми та дат створення;
8. report\_topic – таблиця для збереження скарг на теми, вона містить в собі: поля-ідентифікатори скарг, їх відправників, назв тем та поле дати посилання;
9. user – таблиця з усіма даними про користувачів системи (поля-ідентифікатори для користувачів, їх імен й електронних адрес, поля для збережень паролів, значень карми та дат створення);
10. password\_reset – таблиця списку запитів про відновлення паролю користувачів, яка містить: поле-ідентифікатор запиту, поля для унікальних кодів відновлення паролів та дат посилання;
11. subscription – таблиця переліку підписок користувачів на теми, містить: поля-ідентифікатори підписки, користувачів, назв тем та поле дати посилання;
12. notification – таблиця, яка містить повідомлення користувачів: поля-ідентифікатори для повідомлень, їх отримувачів, об’єктів повідомлень та відправників, поля для визначень об’єктів повідомлень (публікація чи коментар), дій над об’єктами повідомлень (голос «вгору» або «вниз», додання в улюблене та залишення коментарю), чи прочитані повідомлення і дат створення;
13. favourite\_link – таблиця списку улюблених публікацій користувача (поля-ідентифікатори для улюблених публікацій, користувачів, публікацій та поле дати створення);
14. favourite\_reply – таблиця зі списком улюблених коментарів користувача (поля-ідентифікатори для улюблених коментарів, користувачів, коментарів та поле дати створення).

У кожної таблиці, окрім двох – topic і password\_reset є первинний ключ, котрий являє собою поле яке визначає унікальність записів, не може містити порожнє значення і з властивістю «AUTO\_INCREMENT», суть якого полягає в тому, що при додаванні нових записів значення цього стовпця встановлюється автоматично, причому значення дорівнює на одиницю більше попереднього [19].

У таблицях topic, user і password\_reset присутні унікальні ключі для полів назв тем, імен користувачів та електронних адрес користувачів, які створюють обмеження, що всі значення повинні бути різними.

**3.4.1 Наповнення контенту.** Для запису контенту в базу даних залучено MySQLi – розширення драйверу MySQL, яке використовується в мові програмування PHP та додає нові функції до інтерфейсу БД. Функції MySQLi є як процедурними, так і об'єктноорієнтованими, причому першу парадигму розширення успадкувало від більш ранньої версії MySQL.

Сама MySQL розбиває завдання на лінійні, покрокові процедури, що ускладнює внесення змін, оскільки доводиться редагувати код зверху. Тим часом MySQLi розглядає дані як набір взаємозамінних об'єктів з функціями, дозволяючи користувачам легко додавати або видаляти дані [21].

Заради зручності запису контенту в базу даних та його виклику на сайті були створені різні запити в PHP за допомогою SQL команд для вставки, оновлення, видалення, вибірки, фільтрації та сортування даних у вигляді функцій, котрі викликаються при відповідних діях користувача.

Для коректного наповнення контенту в базу даних, а саме записування 4-байтних символів – різноманітних смайлів, котрі широко використовуються в популярних ресурсах, таких як Instagram, YouTube, Twitter та ін., та його сортування, було задане кодування «utf8mb4\_unicode\_ci» для всіх таблиць і їх відповідних полів.

Кодування «utf8mb4\_unicode\_ci», засноване на стандарті Unicode по сортуванню і порівнянні рядків, використовує складні алгоритми для точного сортування широкого числа мов і при використанні спецсимволів. Ці правила беруть до уваги специфічні угоди для мови, адже не завжди сортування йде відповідно до «алфавітним» порядкам [20].

Оскільки розроблювана система розрахована на користувачів зі всього світу, котрі будуть публікувати та коментувати на різних мовах, а також використовувати всякі емодзі навіть для свого імені, використання кодування «utf8mb4\_unicode\_ci» є доцільним.

З метою збереження місця в базі було використано тип полів «VARCHAR» замість «CHAR», тому що в іншому випадку MySQL буде резервувати по чотири байти для кожного символу в стовпці, оскільки це максимально можлива довжина.

**3.5 Опис функціоналу створеної системи**

Розроблена інформаційно-розважальна система вийшла цілком відповідною до її задуму, з легким та зрозумілим дизайном та притаманним функціоналом.

На головній сторінці (рис. 3.4) в хедері, котрий є незмінним для всіх сторінок, додаються лише додаткові елементи при вході в систему, можна побачити основні необхідні переходи – лого, головна, кнопки для випадкової теми, список найпопулярніших тем, усі теми, різне сортування (гаряче, нове, що зростає, обговорюване та топ), вікі сайту (при перебуванні в темі стає вікі теми (рис. 3.7), де знаходиться її опис, виводиться кількість підписників та створювач) та реєстрація / вхід. Якщо знаходитися в темі або профілі користувача (рис. 3.10), то в хедері після логотипа буде видно саме назву цієї теми або ім’я користувача.

В боді на стартовій сторінці зображується список «гарячих» публікацій, те що і є сенсом системи, де можна відразу проголосувати за публікацію, відкрити її, відправити скаргу, поділитися посиланням на неї в різні ресурси, також перейти в профіль автора, тему в яку опублікований пост, та відфільтрувати всі публікації по джерелу, в кінці списку присутня кнопка наступної сторінки, після використання якої з’являється належна кнопка попередньої сторінки (рис. 3.5).

Справа знаходиться швидкий пошук по ключовому слову публікації в назві, описі, темі або джерелу. Далі швидка авторизація, відновлення паролю, випадкова тема з банером на яку можна перейти, кнопки для публікацій постів, лічильник всіх постів, тем та коментарів.

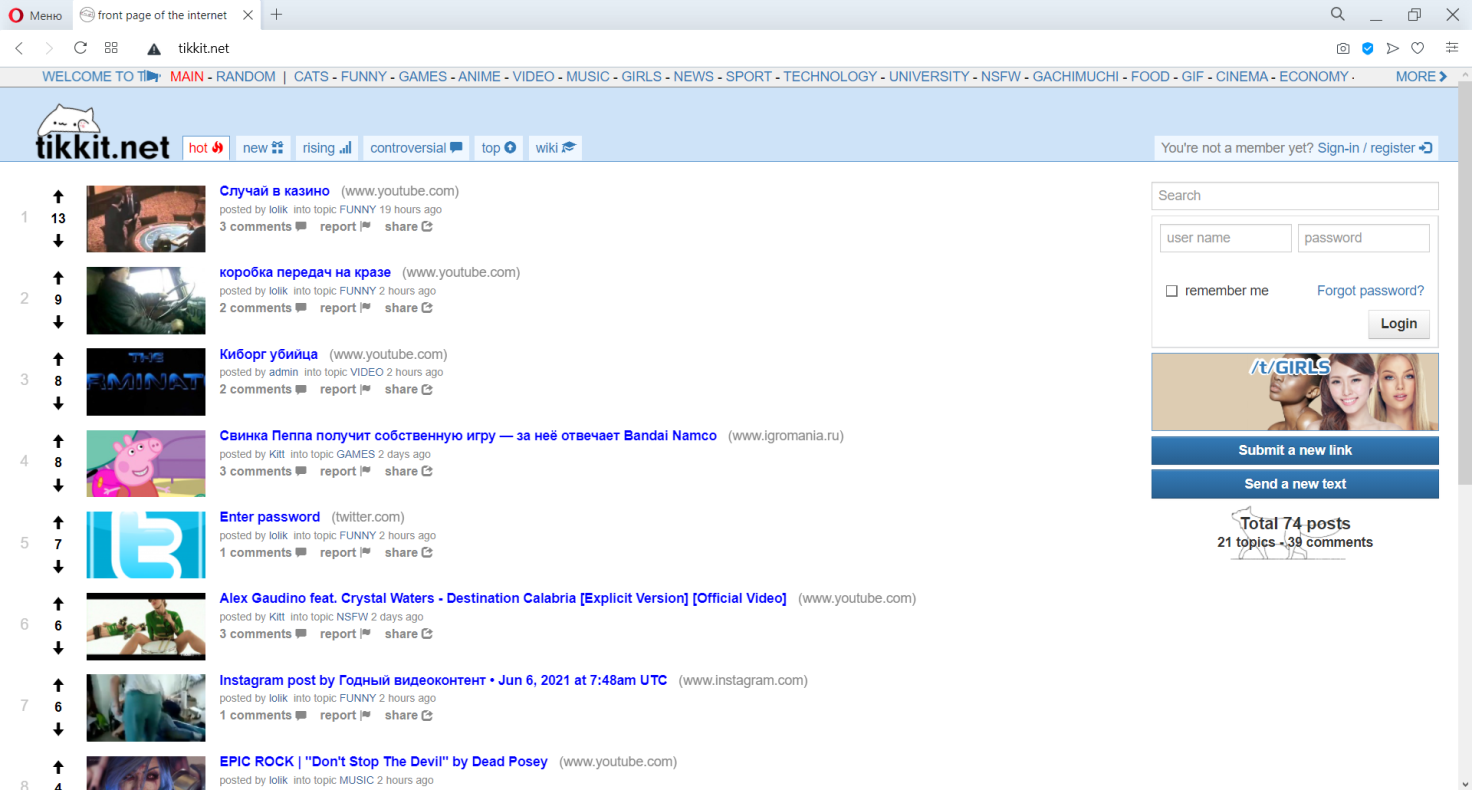
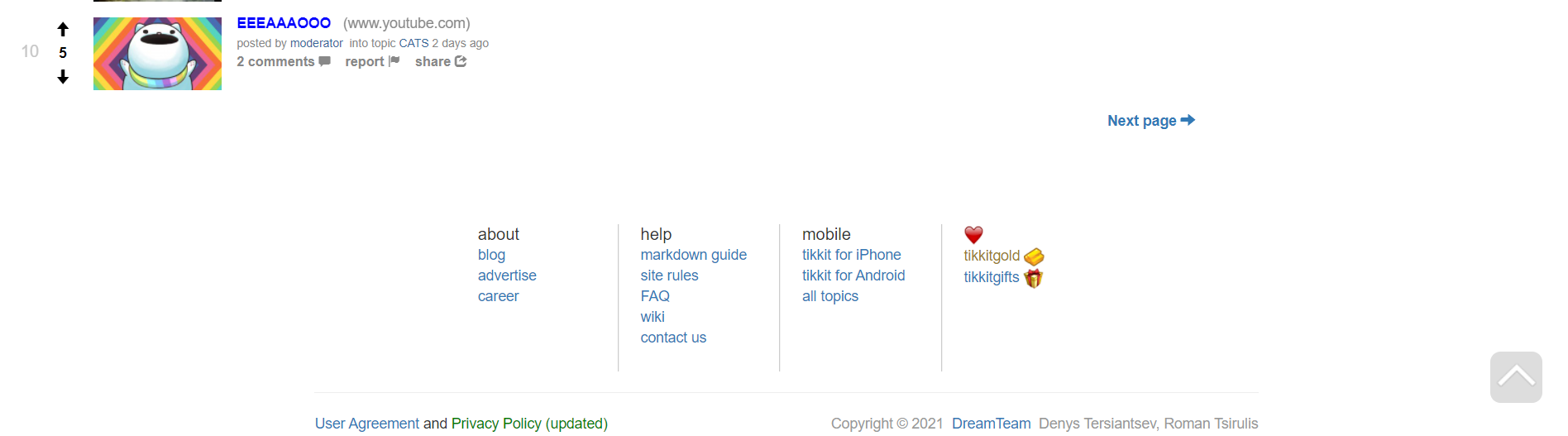
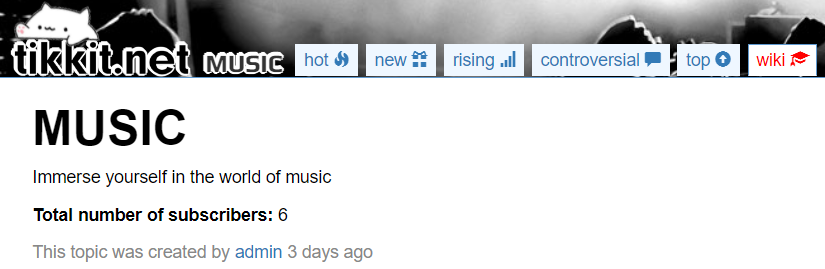
  
Рисунок 3.4 – Головна сторінка системи

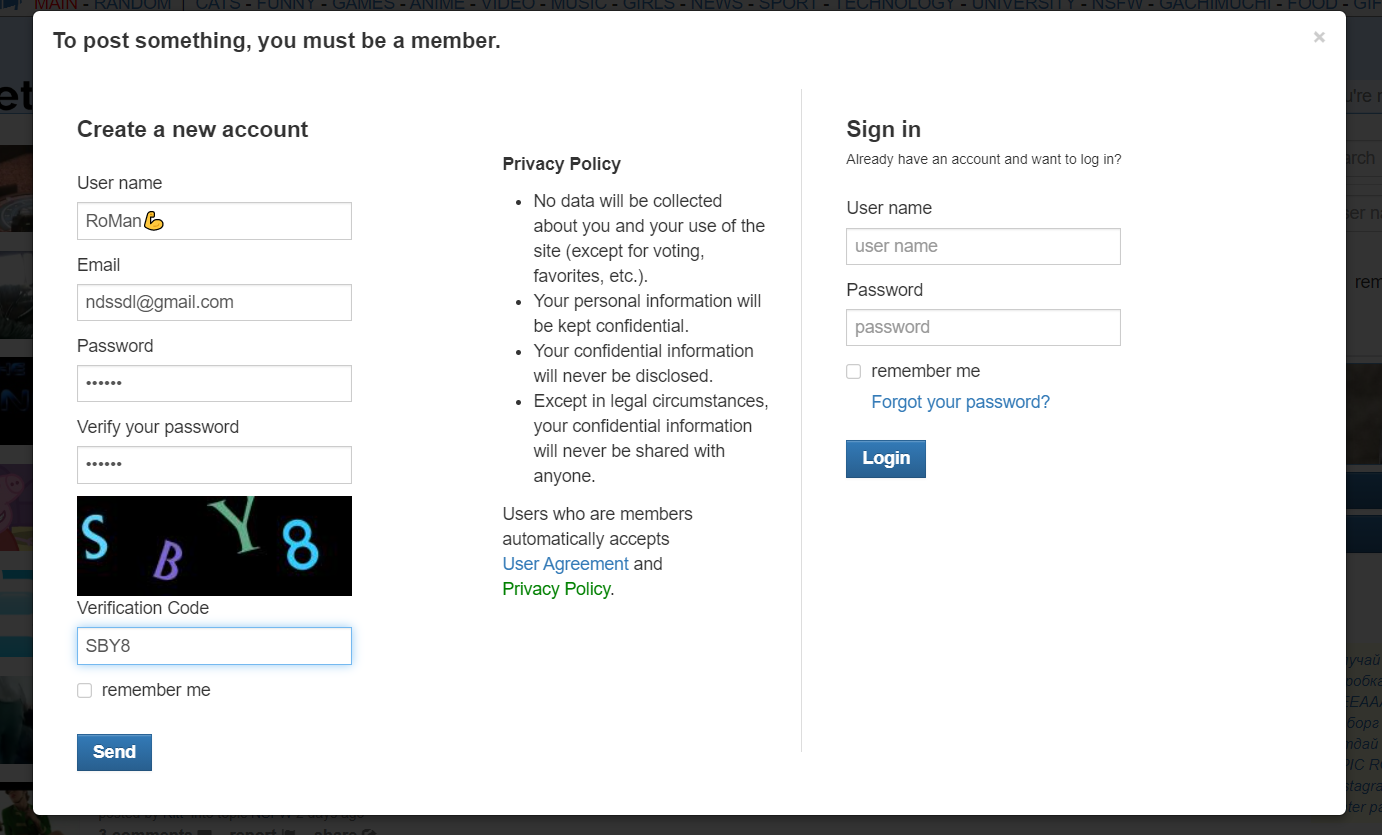
  
Рисунок 3.5 – Кнопки наступної та попередньої сторінки

На нижній частині сайту (рис. 3.6) присутні переходи до довідникової інформації, які зображені в внутрішній структурі системи. Також, для зручності повернення на «гору» сайту, зроблена доцільна кнопка, котра з’являється лише при гортанні вниз.

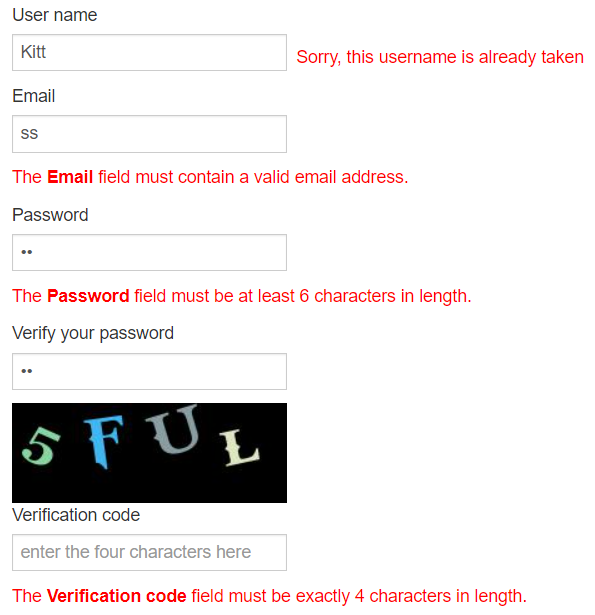
  
Рисунок 3.6 – «Футер» системи

  
Рисунок 3.7 – Вікі теми

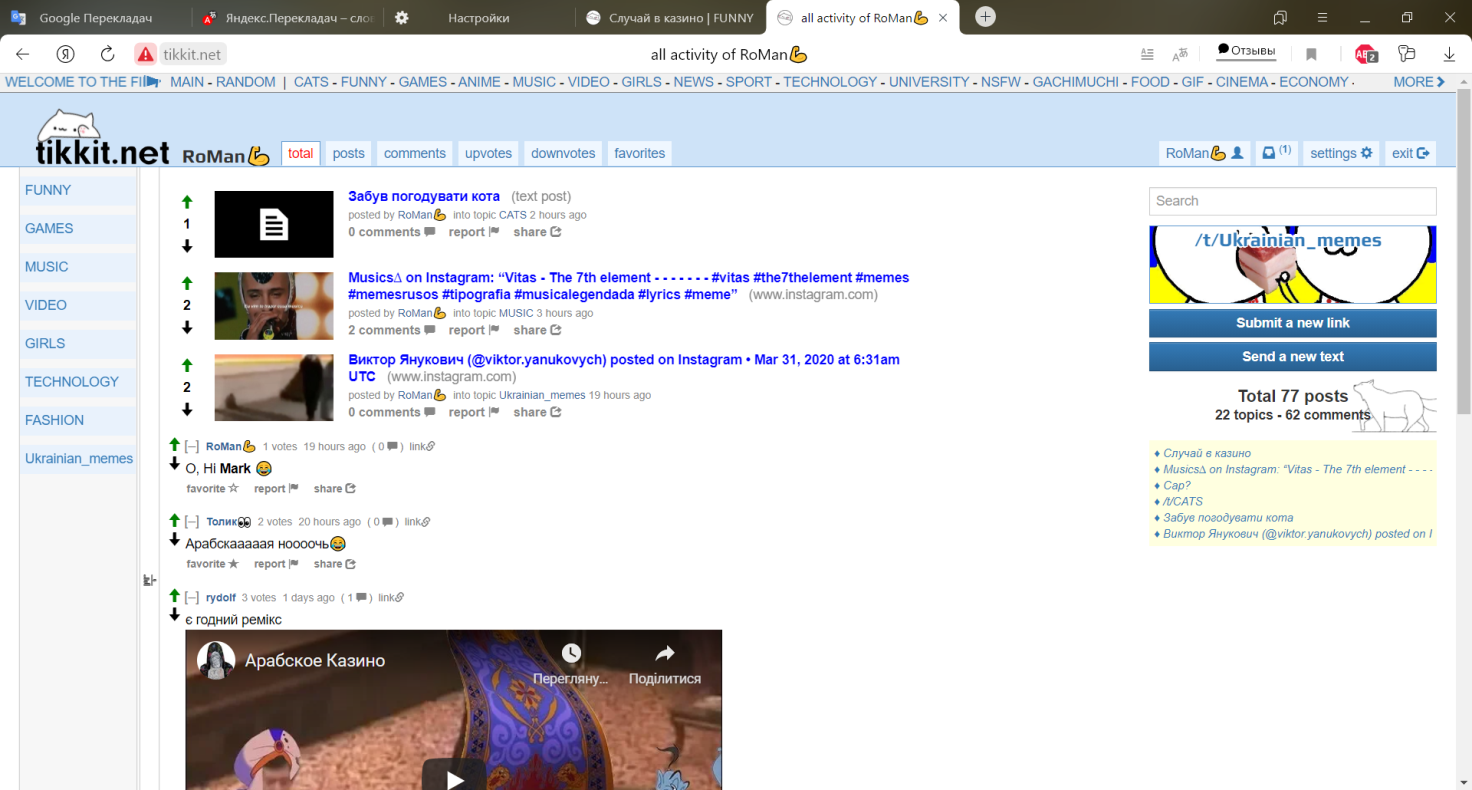
Форма реєстрації / авторизації наведена на рис. 3.8. Перейти на неї можливо в хедері, або при спробі використання функціоналу дозволеному тільки зареєстрованому користувачу, котрий наведений в діаграмі використання.

  
Рисунок 3.8 – Вікно реєстрації / авторизації

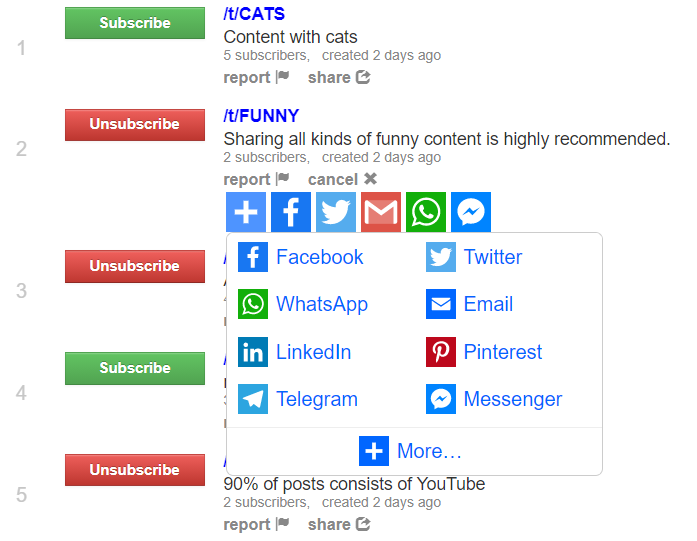
При невірному введенню даних в усіх формах, будь-то логін котрий вже присутній в базі, перевищення ліміту дозволених символів, не проходження мінімального ліміту символів, не вірний формат або просто порожнє поле, система виведе належне повідомлення (рис. 3.9).

  
Рисунок 3.9 – Приклад невірно заповненої форми

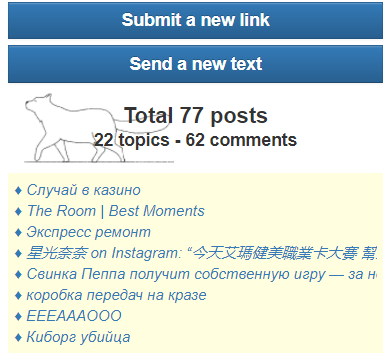
Після реєстрації користувачу доступний власний профіль (рис. 3.10), котрий містить історію всієї його активності, її можна фільтрувати по списку постів або коментарів, що користувач сам опублікував, по списку публікацій та коментарів за які проголосував «верх» або «вниз» чи додав в улюблене. Відкрити її можна натиснувши на свій нікнейм.

  
Рисунок 3.10 – Власний профіль користувача

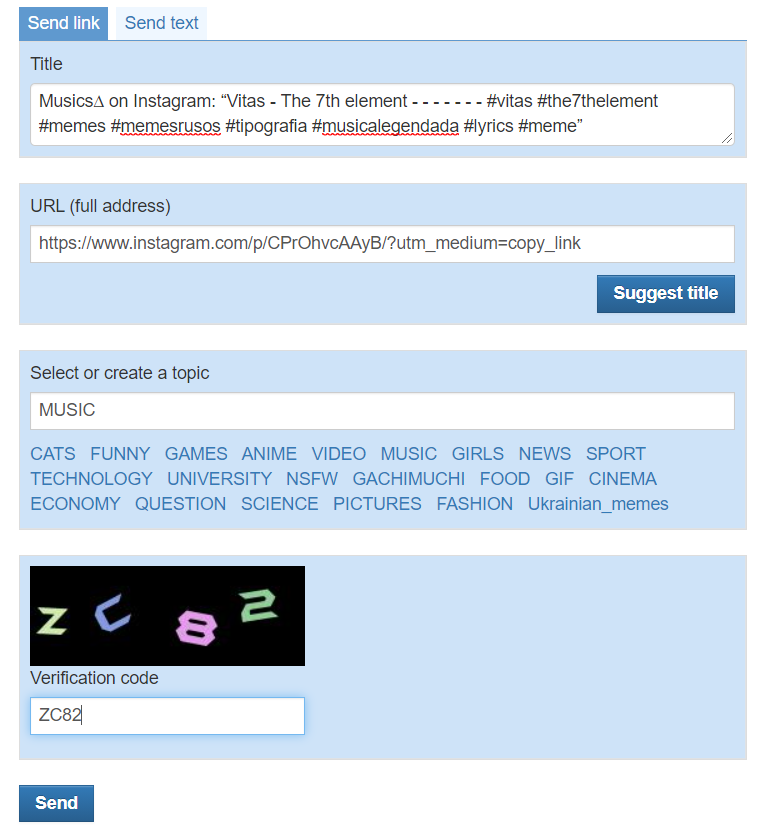
При підписуванні на теми через їх список (рис. 3.11) вони додаються у панель підписок, котра знаходиться зліва в авторизованого користувача, а відкривається список після натискання на неї (рис. 3.10). На тему також можна відправити скаргу або поділитися посиланням.

  
Рисунок 3.11 – Приклад списку тем та кнопки розповсюдження

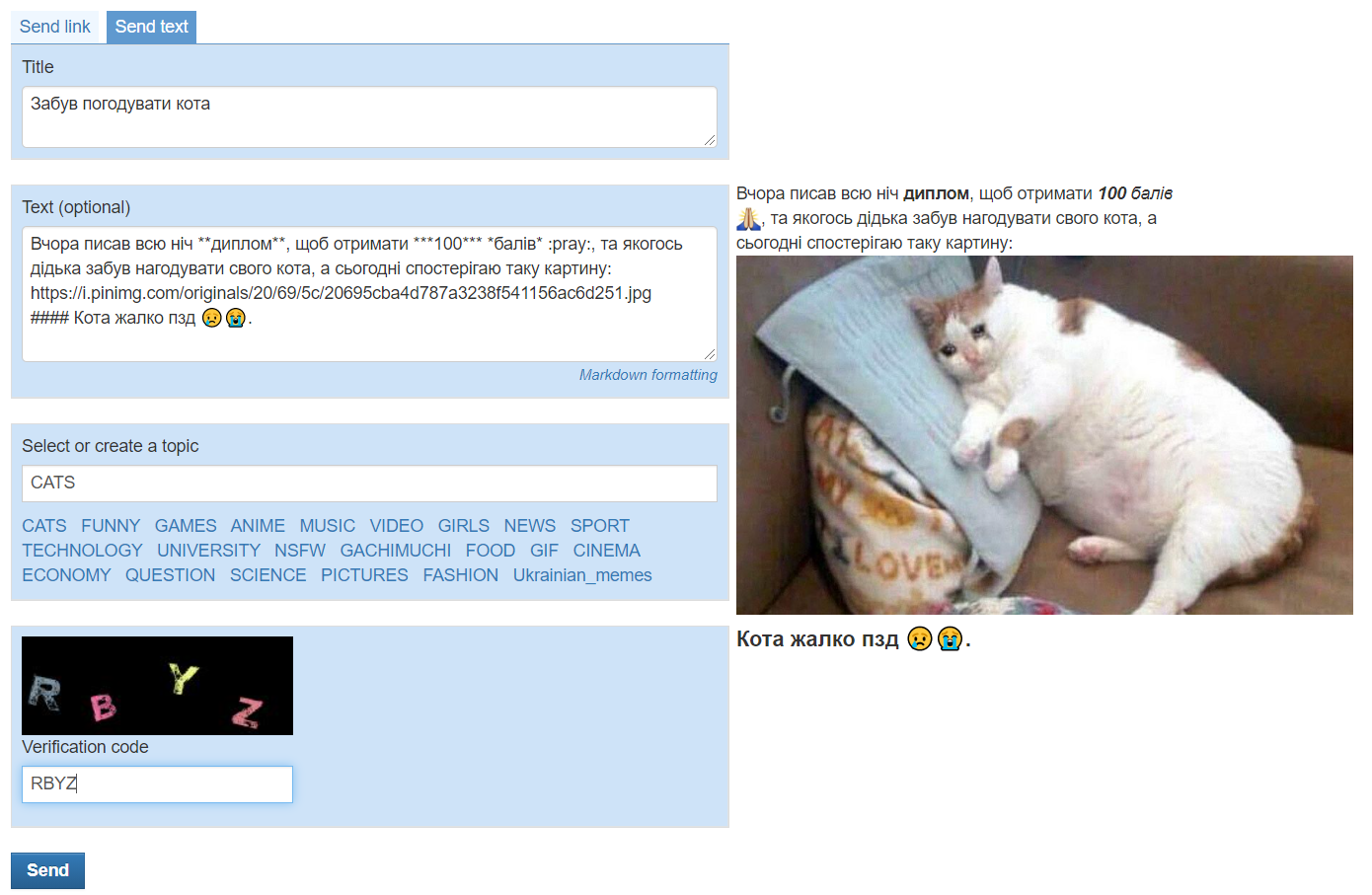
Історія останніх відкритих публікацій показується за лічильником системи, через неї можна знову відкрити ці пости, максимум зображається у списку 8 публікацій (рис. 3.12).

  
Рисунок 3.12 – Список останніх відкритих публікацій

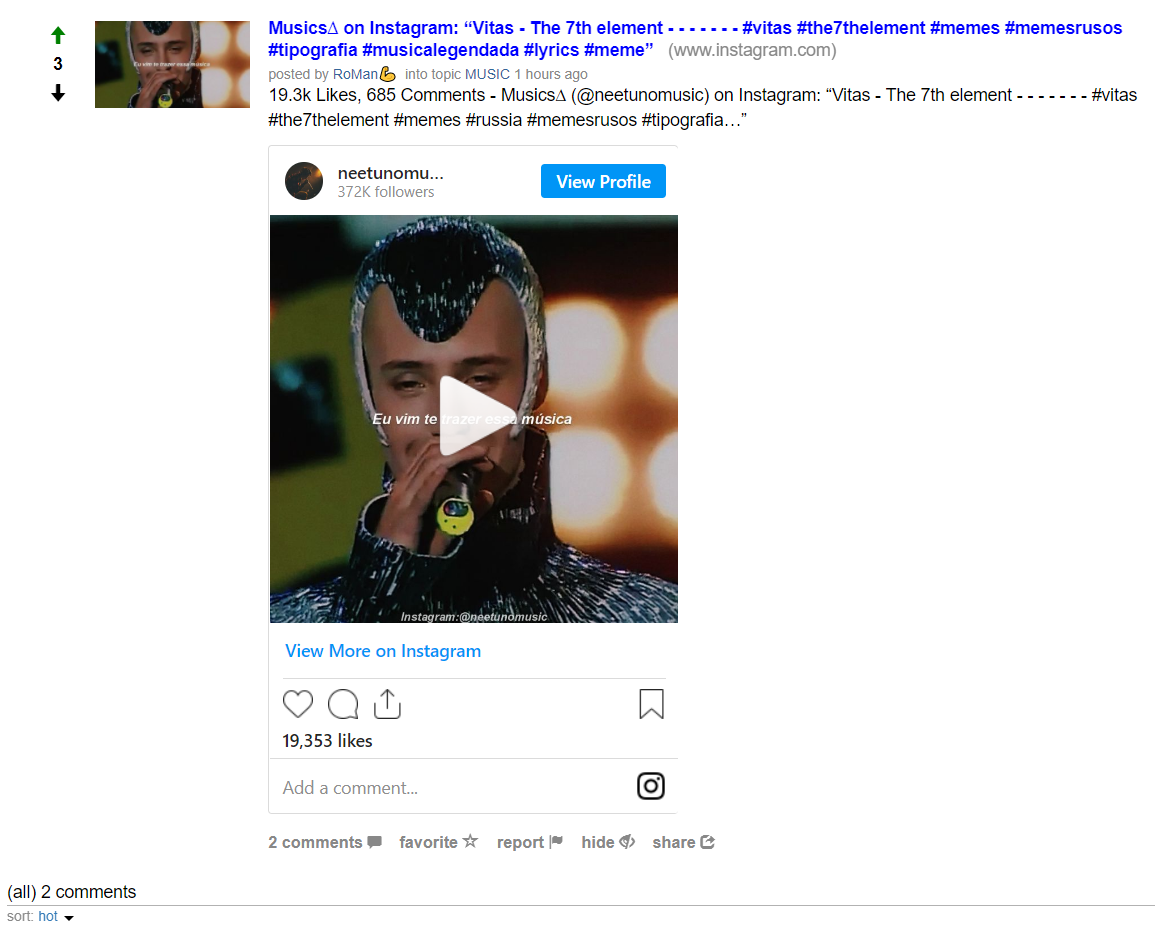
Наявні два види публікації постів (рис. 3.12) – з посиланням або текстом. У формі публікації з посиланням (рис. 3.13) є опція запропонувати заголовок, система дістає заголовок з посилання якщо він присутній, якщо ні пише «Немає заголовка». В обох формах можна вибрати тему зі списку найпопулярніших або створити свою.

  
Рисунок 3.13 – Форма публікації посилання

У формі текстового публікування (рис. 3.14) та в коментарях для форматування тексту використовується Markdown – полегшена мова розмітки. Для того, щоб користувач не уявляв як саме буде виглядати його текст, справа буде зображатися попередній перегляд.

  
Рисунок 3.14 – Форма текстової публікації

На рис. 3.15 показано приклад вигляду поста та ембед стороннього ресурсу, а саме елементу стрічки публікації Instagram. Тут можна проголосувати за нього, переглянути, відсортувати або залишити коментарі (рис. 3.16), додати в улюблене, відправити скаргу, приховати (для перегляду коментарів в повний екран) та поділитися посиланням як на саму публікацію або на її коментар, в останньому випадку при переході за посиланням сторінка відкриється в тому місці де залишено коментар та підсвітить його (рис. 3.17), аналогічно відбувається і при переході з повідомлень (рис. 3.18).

  
Рисунок 3.15 – Вигляд публікації в системі

Приклад вигляду «вітки» коментарів, де видно їх структуру, який за котрим йде, наведено на рисунку 3.16. В коментарях також можна залишати посилання на сторонні ресурси.

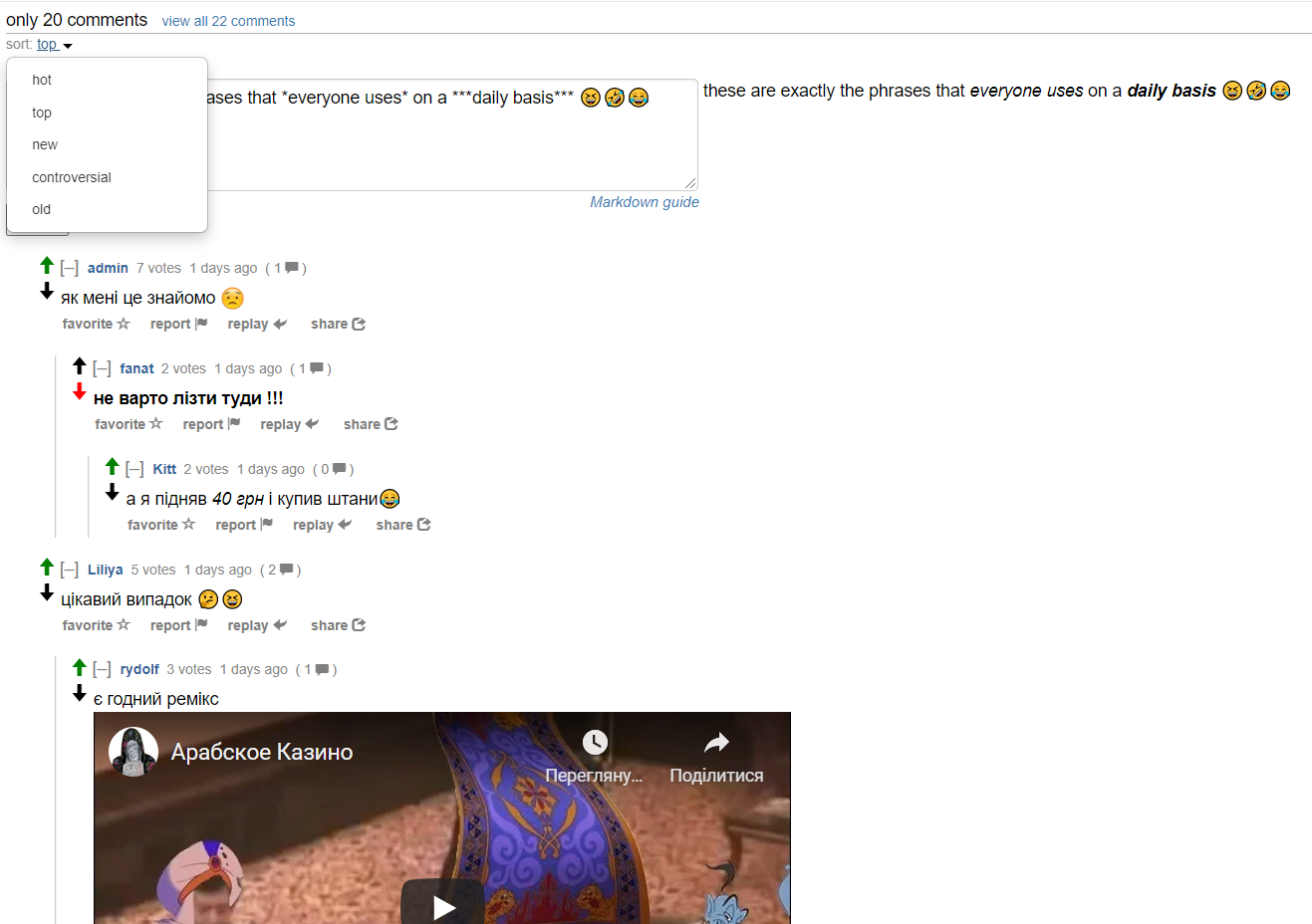
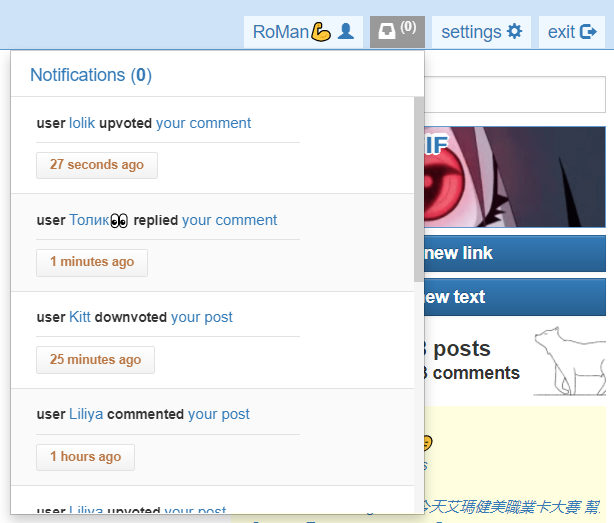
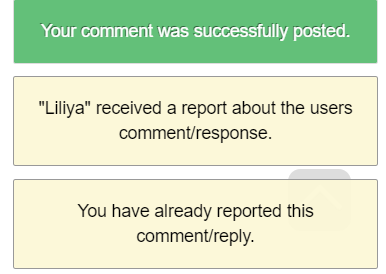
Рисунок 3.16 – Вигляд коментарів та їх сортування

  
Рисунок 3.17 – Приклад підсвічування коментаря при його розповсюджені

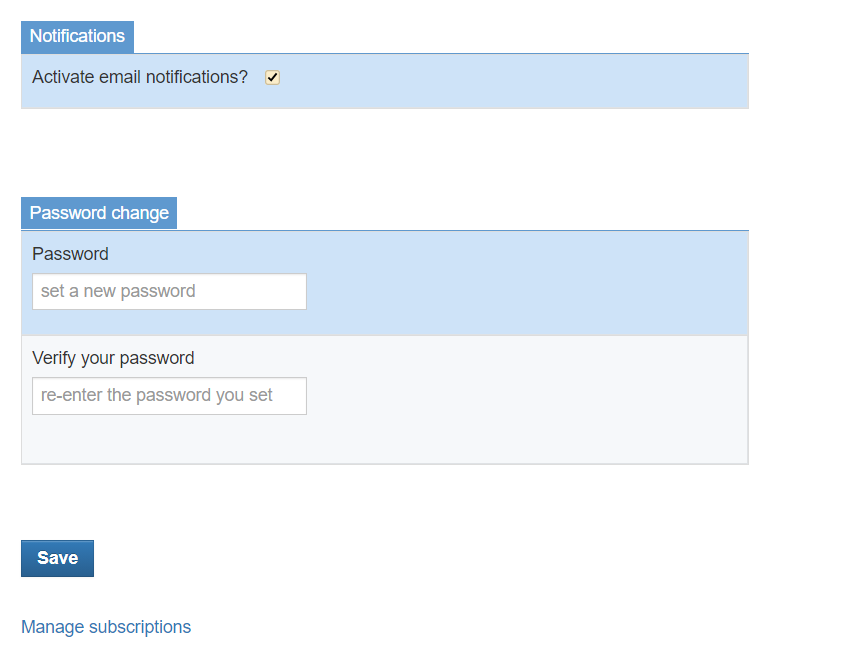
Користувачу надходять повідомлення, якщо на його пости або коментарі відповідають чи голосують за них (рис. 3.18).

  
Рисунок 3.18 – Список повідомлень

При деяких діях користувача, система виводить допоміжні повідомлення (рис. 3.19) як, наприклад при завершенні реєстрації, голосуванні за пост або коментар, чи при спробі повторного голосування та інші.

  
Рисунок 3.19 – Приклад допоміжних повідомлень, які спливають при конкретних діях користувача

Налаштування користувача, які наведені на рис. 3.20, надають можливості відключення або включення повідомлень на пошту (при наявності поштового сервісу), зміни паролю та відкриття менеджера підписувань на теми. У адміністратора ще наявна додаткова секція для адміністрування.

  
Рисунок 3.20 – Налаштування користувача

Отже, увесь функціонал, який був описаний в формулюванні завдання, повністю реалізований в вище проілюстрованій інформаційно-розважальній системі.

**РОЗДІЛ 4  
ТЕСТУВАННЯ ІНФОРМАЦІЙНО-РОЗВАЖАЛЬНОЇ СИСТЕМИ**

Вебтестування – це ретельна перевірка сайту на наявність потенційних помилок. В ході тестування визначається зручність й привабливість сайту для користувача, швидкість отримання і доступність потрібної інформації та, найголовніше, чи відповідає він тим бізнес-цілей, які були поставлені на початку.

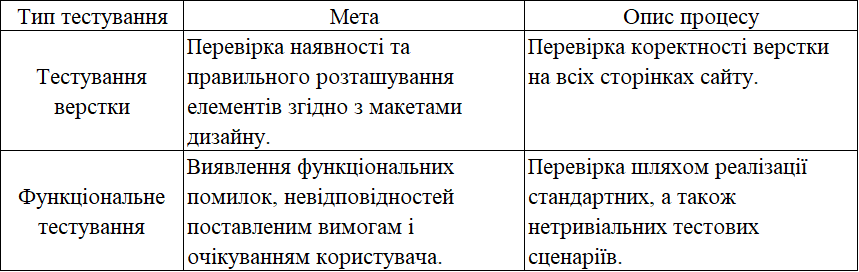
Загальна мета вебтестування – знайти проблеми, також відомі як помилки або дефекти, які можуть мати негативний вплив на вебсайт або додаток. Вебтестування також можна використовувати для виявлення певних областей або аспектів вебсайту, які можна поліпшити та отримати кращі результати, будь-то більше запитів, більше продажів, більше постійних відвідувачів або щось ще.

**4.1 Тест-план**

Тест-план – це спеціалізований технічний документ, всередині якого описується весь майбутній фронт робіт з тестування розроблюваного продукту, починаючи з документування самого об'єкта, використаного на початку стратегії розробки, і критеріїв знаходження помилок [27].

Щоб протестувати розроблену систему складемо тест-план (табл. 4.1), для визначення етапів тестування, а саме розділення цього процесу на п’ять типів, формулюванні мети цих етапів та опису процесу проведення.

Таблиця 4.1 – Тест-план до тестування системи



Продовження таблиці 4.1



Далі, після проведення регресійного тестування, створимо чек-листи до тестування відповідної частини розробленої інформаційно-розважальної системи – функціонал, взаємодія з базою та роботи посилань.

**4.2 Тестування взаємодії з базою**

Тестування бази даних може охоплювати перевірку правдивості даних, перевірку цілісності даних, перевірку продуктивності, пов'язану із базою даних, й тестування процедур, тригерів і функцій в БД [28].

Основні характерні особливості, на які необхідно звертати увагу при тестуванні баз даних:

* зображення інформації: зіставлення полів у формі користувальницького інтерфейсу із подібними полями всередині бази;
* обновлення запису в таблиці: всякий раз, коли певна дія буде виконуватися в зовнішньому інтерфейсі ПЗ, відповідна дія повинна викликатися й у внутрішньому інтерфейсі;
* тестування властивостей ACID операцій: всі операції, котрі виконуються в базі, мусять повною мірою відповідати таким 4 властивостям: атомарність (гарантує, що кожен запит в транзакції буде виконаний успішно, або в разі помилки одного, взагалі не відбудеться), узгодженість (вимога, яка передбачає, що в результаті роботи транзакції дані будуть допустимими, наприклад, перевірка вірності формату введеної електронної адреси), ізольованість (під час виконання транзакції паралельні транзакції не повинні впливати на її результат) і довговічність (зміни, отримані в результаті транзакції, повинні залишатися збереженими незалежно від будь-яких збоїв, інакше кажучи, якщо користувач отримав сигнал про завершення транзакції, він може бути впевнений, що дані збережені) [16];
* збереженість і цілісність даних: вся інформація, а також її модифікація, повинні зображатися однаково і в зрозумілій для користувача формі, тобто необхідно переконатися, що система демонструє крайні зміни в таблицях ідентично у всіх місцях, де вони зображаються;
* відповідність типів для полів: у кожного поля повинен бути належний тип, наприклад, якщо поле дати – формат «datetime»;
* захист важливих даних: наприклад хешування паролю.

Результати тестування взаємодії з базою даних наведені в табл. 4.2.

Таблиця 4.2 – Чек-лист до тестування взаємодії з базою



**4.3 Тестування функціоналу**

Функціональне тестування – це перевірка функціональних вимог до системи і якості отриманих результатів.

Основний акцент при тестуванні робиться на кінцевий результат, на ті результати, які виходять при виконанні призначених для користувача сценаріїв і на ті вихідні дані, які є підсумком введення призначених для користувача даних.

Імітація практичного використання системи її користувачем дозволяє безпомилково встановити ступінь функціональної придатності програми до виходу на ринок, її точність і відповідність заданим вимогам, а також визначити здатність до взаємодії з іншими сумісними додатками, захищеність і стабільність рішення [15].

На підставі проведених тестів був складений чек-лист, котрий представлений в таблиці 4.3.

Таблиця 4.3 – Чек-лист до тестування функціоналу системи



Продовження таблиці 4.3



**4.4 Тестування роботи посилань**

Інтернет постійно розвивається й змінюється, кожен день з'являються тисячі нових блогів та сайтів, і кожен день майже стільки ж зникає. Тому деякі посилання з часом можуть перетворитися в биті, тобто посилаються на документи або об'єкти, що вже не існують [17].

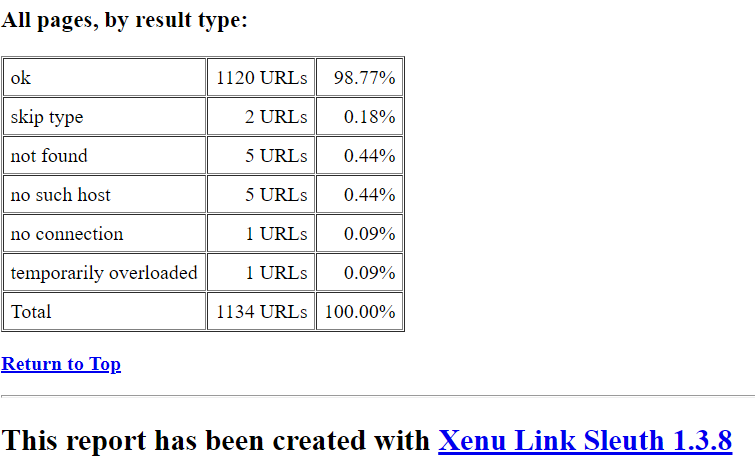
В створеній інформаційно-розважальній системі уникнути битих посилань не можливо, через зовнішні посилання з різноманітних ресурсів в публікаціях, в їх описі та в коментарях.

Перевірка вебсайту на наявність битих посилань відбувалася за допомогою програми Xenu Link Sleuth. Перевірка посилань виконується для звичайних посилань, зображень, фреймів, плагінів, фонів, локальних карт зображень, таблиць стилів, скриптів і Java-аплетів. Вона зображає постійно оновлюваний список URL-адрес, які можна сортувати за різними критеріями, а сам звіт може бути підготовлений в будь-який час.

Основні функції програми:

* + простий користувальницький інтерфейс без зайвих надмірностей;
  + може повторно перевіряти посилання, котрі не працюють (корисно при тимчасових мережевих помилках);
  + простий формат звіту, також його можна відправити на електронну пошту;
  + виконуваний файл розміром менше ніж 1 МБ;
  + підтримує вебсайти з криптографічним протоколом SSL;
  + часткове тестування сайтів з протоколом передачі файлів FTP і Gopher;
  + виявляє і повідомляє про перенаправлені URL-адреси;
  + будує карту переходів всього сайту.

Результати тестування створеної системи за допомогою Xenu Link Sleuth наведені на рисунку 4.1.

  
Рисунок 4.1 – Загальний звіт тестування програмою Xenu

Зі звіту тестування видно, що майже всі посилання справні, окрім п’яти переходів на стороні ресурси котрі були в описі YouTube постів, тобто система цілком адекватно працює.

**4.5 Залиття на хостинг**

Після ретельного тестування та усунення усіх дефектів системи, сайт готовий до викладання в інтернет. Для цього можна скористатися послугами різних хостинг-провайдерів – організацій, що надають професійні послуги хостингу або, іншими словами, надають дисковий простір і потужності сервера для розміщення сайту в мережі. Зазвичай, у кожного хостинг-провайдера є своя детальна інструкція по залиттю сайтів, опис всіх привілеїв їхніх послуг та технічна підтримка.

Традиційно, сценарій залиття такий – потрібно обрати та зареєструвати домен (унікальна «адреса» сайту в мережі), замовити хостинг (майданчик для розміщення файлів сайту, тобто сервер), прив’язати домен до хостингу й останнім кроком розмістити сайт на хостингу, через панель керування хостингом, завантаживши архів проєкту в кореневий каталог, розпакувати його та імпортувати базу даних. Або через FTP-клієнт, відкриваючи його (краще використовувати спеціалізовані рішення в дусі Filezilla), треба під'єднатися по протоколу FTP до свого віртуального хостингу, авторизуватися через пароль, виданий хостингом, та так само копіювати файли чи архів на сервер, в останньому випадку розпакувати вбудованим архіватором і при необхідності видати права на читання і запис завантажених директорій й документів.

**ВИСНОВКИ**

Під час виконання кваліфікаційної роботи бакалавра було проведено аналіз проблеми дослідження, шляхом визначення навіщо взагалі потрібні інформаційно-розважальні системи з подальшим оглядом найпопулярніших аналогів, та на їх основі сформульовано завдання до розроблення власної.

Коли всі вимоги до проєкту були визначені, розпочалося проєктування структури майбутньої системи для кращого уявлення її роботи. Здійснювалося це за допомогою створення внутрішньої структури сайту, діаграми варіантів використання для усіх користувачів і діаграм діяльності, котрі зображають найголовніші процеси системи із взаємодією сайту з базою даних, а саме авторизацію та викладення публікацій.

Після вибору та обґрунтування використання технологій для створення сайту та розроблення бази даних з урахуванням потреб системи, було практично реалізовано інформаційно-розважальну систему і проілюстровано результати з повноцінним описом її функціонала.

Для впевненості в адекватності роботи створеної системи та зручності її використання було проведено детальне тестування до відповідної розробленої частини системи за прописаним тест-планом: тестування взаємодії сайту з базою, функціоналу та роботи посилань за допомогою програми Xenu Link Sleuth. Коли всі дефекти та помилки були усунені, сайт готовий до залиття на хостинг.

Таким чином, розроблено інформаційно-розважальну систему, котра цілком відповідає поставленій меті цієї дипломної роботи. Вона надає можливість користувачам в зручній формі насолоджуватися переглядом та публікуванням різноманітного контенту, як для відпочинку та і для саморозвитку, з можливістю спілкуватися в притаманній для цього секції коментувань. А наявне сортування інформації за допомогою системи голосувань спрощують та покращують даний процес. Для майбутніх удосконалень системи можна додати: можливість кастомізувати власний профіль і створені теми, підключення поштового сервісу та збереження карми в профілі.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Кому нужны развлекательные порталы и для чего их создают. *business-gazeta.ru* :вебсайт. URL: <https://www.business-gazeta.ru/article/359830> (дата звернення: 18.02.2021).
2. Що таке UGC?. *bazilik.media* :вебсайт. URL: <https://bazilik.media/shcho-take-ugc/> (дата звернення: 18.02.2021).
3. На передовій інтернету. Що таке Reddit і на чому він заробляє. *adsider.com* :вебсайт. URL: <https://adsider.com/ua/na-peredoviy-internetu-shcho-take-reddit-i-na-chomu-vin-zarobliaie/> (дата звернення: 19.02.2021).
4. У Києві з’явився офіс Reddit. Компанія шукає розробників. *nachasi.com* :вебсайт. URL: <https://nachasi.com/2020/02/21/reddit-kyiv/> (дата звернення: 19.02.2021).
5. Языки программирования, которые используют Google, Reddit и другие гиганты. *dev.by* :вебсайт. URL: <https://dev.by/news/yazyki-programmirovaniya-kotorye-ispolzuyut-google-reddit-i-drugie-giganty> (дата звернення: 19.02.2021).
6. How 9GAG Scaled Fun to 150 Million Users with Cloudflare. *cloudflare.com* :вебсайт. URL: <https://www.cloudflare.com/case-studies/9gag-cdn-ddos-protection> (дата звернення: 20.02.2021).
7. Как устроен бизнес 9GAG: мобильные игры, юмористический сайт и собственный мессенджер. *vc.ru* :вебсайт. URL: <https://vc.ru/flood/9757-9gag> (дата звернення: 20.02.2021).
8. Fishki.net: «Мы предугадываем желания аудитории». *sostav.ru* :вебсайт. URL: <https://www.sostav.ru/publication/fishki-20936.html> (дата звернення: 21.02.2021).
9. О проекте. *fishki.net* :вебсайт. URL: <https://fishki.net/about/> (дата звернення: 21.02.2021).
10. Что Такое PHP? Руководство для Начинающих. *hostinger.com.ua* :вебсайт. URL: <https://www.hostinger.com.ua/rukovodstva/chto-takoe-php> (дата звернення: 26.05.2021).
11. Разработка веб сайтов на фреймворке CodeIgniter. *redline.by* :вебсайт. URL: <https://redline.by/novosti/codeigniter.html> (дата звернення: 26.05.2021).
12. Что такое MySQL. Взаимодействие с PHP. *softtime.ru* :вебсайт. URL: <http://www.softtime.ru/bookphp/gl12_1.php> (дата звернення: 27.05.2021).
13. Что должен знать FrontEnd разработчик в 2019 году. *itvdn.com* :вебсайт. URL: <https://itvdn.com/ru/blog/article/frontend-2019> (дата звернення: 27.05.2021).
14. Язык JavaScript: что это и для чего?. *itlogia.ru* :вебсайт. URL: <https://itlogia.ru/article/yazyk_javascript_chto_eto_i_dlya_chego> (дата звернення: 27.05.2021).
15. Функциональное тестирование программного обеспечения. *qawebmart.ru* :вебсайт. URL: <http://qawebmart.ru/funkcionalnoe-testirovanie.html> (дата звернення: 31.05.2021).
16. Транзакции, ACID, CAP. *gb.ru* :вебсайт. URL: <https://gb.ru/posts/acid_cap_transactions> (дата звернення: 31.05.2021).
17. Битые ссылки. Как проверить, найти и удалить. *asbseo.ru* :вебсайт. URL: <https://asbseo.ru/optimizaciya-i-prodvizhenie-bloga/bitye-ssylki-kak-proverit-najti-i-udalit.html> (дата звернення: 31.05.2021).
18. Работа в phpMyAdmin. *itproffi.ru* :вебсайт. URL: <https://itproffi.ru/rabota-v-phpmyadmin/> (дата звернення: 01.06.2021).
19. Как сбросить AUTO\_INCREMENT в MySQL/MariaDB. *zalinux.ru* :вебсайт. URL: <https://zalinux.ru/?p=2582#:~:text=AUTO_INCREMENT> (дата звернення: 01.06.2021).
20. MySQL Unicode. *engineertips.wordpress.com* :вебсайт. URL: <https://engineertips.wordpress.com/2020/05/26/mysql-unicode/> (дата звернення: 02.06.2021).
21. PHP-подключение к БД MySQL (Два Способа с Примерами). *hostinger.ru* :вебсайт. URL: <https://www.hostinger.ru/rukovodstva/php-podkliuchenije-k-bd-mysql> (дата звернення: 02.06.2021).
22. Структура сайта: схемы, советы по разработке, примеры. *sales-generator.ru* :вебсайт. URL: <https://sales-generator.ru/blog/struktura-sayta/> (дата звернення: 03.06.2021).
23. Статические и динамические сайты – в чем разница и что выбрать. *wp-system.ru* :вебсайт. URL: <https://wp-system.ru/sozdanie-sayta/staticheskie-i-dinamicheskie-sajty/> (дата звернення: 03.06.2021).
24. Учебное пособие по диаграмма прецедентов. *creately.com* :вебсайт. URL: <https://creately.com/blog/ru/диаграмма/учебное-пособие-по-диаграмма-прецеде/> (дата звернення: 04.06.2021).
25. Варіант використання. *mybiblioteka.su* :вебсайт. URL: <https://mybiblioteka.su/3-1878.html> (дата звернення: 04.06.2021).
26. UML activity diagrams. *uml-diagrams.org* :вебсайт. URL: <https://www.uml-diagrams.org/activity-diagrams.html> (дата звернення: 05.06.2021).
27. Как писать план тестирования. *testmatick.com* :вебсайт. URL: <https://testmatick.com/ru/kak-pisat-plan-testirovaniya/> (дата звернення: 06.06.2021).
28. Тестирование базы данных. *coderlessons.com* :вебсайт. URL: <https://coderlessons.com/tutorials/kachestvo-programmnogo-obespecheniia/izuchite-testirovanie-bazy-dannykh/testirovanie-bazy-dannykh-kratkoe-rukovodstvo> (дата звернення: 07.06.2021).

**ДОДАТОК А  
ДАМП БАЗИ ДАНИХ**

-- phpMyAdmin SQL Dump

-- version 4.8.5

-- https://www.phpmyadmin.net/

--

-- Хост: 127.0.0.1:3306

-- Час створення: Чрв 04 2021 р., 16:47

-- Версія сервера: 8.0.19

-- Версія PHP: 7.0.33

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- База даних: `tikkit\_data`

--

-- --------------------------------------------------------

--

-- Структура таблиці `favourite\_link`

--

CREATE TABLE `favourite\_link` (

`id` int NOT NULL,

`uid` int NOT NULL,

`link\_id` int NOT NULL,

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=COMPACT;

--

-- Дамп даних таблиці `favourite\_link`

--

INSERT INTO `favourite\_link` (`id`, `uid`, `link\_id`) VALUES

(1, 1, 157),

(2, 8, 175),

(3, 9, 185),

(4, 1, 185);

-- --------------------------------------------------------

--

-- Структура таблиці `favourite\_reply`

--

CREATE TABLE `favourite\_reply` (

`id` int NOT NULL,

`uid` int NOT NULL,

`reply\_id` int NOT NULL,

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=COMPACT;

--

-- Дамп даних таблиці `favourite\_reply`

--

INSERT INTO `favourite\_reply` (`id`, `uid`, `reply\_id`) VALUES

(4, 1, 35);

-- --------------------------------------------------------

--

-- Структура таблиці `link`

--

CREATE TABLE `link` (

`id` int NOT NULL,

`uid` int NOT NULL,

`title` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

`url` varchar(2048) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`text` text CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci,

`embed` varchar(5000) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`picurl` varchar(2048) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`domain` varchar(33) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`topic` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`score` int NOT NULL DEFAULT '0',

`comments` int NOT NULL DEFAULT '0',

`reported` int NOT NULL DEFAULT '0',

`favorited` int NOT NULL DEFAULT '0',

`is\_link\_for\_union` tinyint(1) NOT NULL DEFAULT '1'

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=COMPACT;

--

-- Дамп даних таблиці `link`

--

INSERT INTO `link` (`id`, `uid`, `title`, `url`, `text`, `embed`, `picurl`, `domain`, `topic`, `score`, `comments`, `reported`, `favorited`, `is\_link\_for\_union`) VALUES

(16, 2, 'U Got That | FULL VERSION [MMV]', 'https://www.youtube.com/watch?v=uuBETyA\_yxc', 'U Got that MMV========Gesugao ship girl illustration.=========Artist : ebiblue ()Twitter : https://twitter.com/ebiblueinstagram : https://www.in...', '<iframe width=\"560\" height=\"315\" src=\"https://www.youtube.com/embed/uuBETyA\_yxc\" frameborder=\"0\" allowfullscreen></iframe>', 'https://i.ytimg.com/vi/uuBETyA\_yxc/hqdefault.jpg', 'www.youtube.com', 'ANIME', 2, 0, 0, 0, 1);

-- --------------------------------------------------------

--

-- Структура таблиці `notification`

--

CREATE TABLE `notification` (

`id` int NOT NULL,

`uid` int NOT NULL,

`item\_type` tinyint NOT NULL,

`action\_type` tinyint NOT NULL,

`item\_id` int NOT NULL,

`item\_uid` int NOT NULL,

`unread` tinyint(1) NOT NULL DEFAULT '1',

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=COMPACT;

--

-- Дамп даних таблиці `notification`

--

INSERT INTO `notification` (`id`, `uid`, `item\_type`, `action\_type`, `item\_id`, `item\_uid`, `unread`) VALUES

(5, 8, 0, 0, 180, 6, 0);

-- --------------------------------------------------------

--

-- Структура таблиці `password\_reset`

--

CREATE TABLE `password\_reset` (

`email` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

`token` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп даних таблиці `password\_reset`

--

INSERT INTO `password\_reset` (`email`, `token`) VALUES

('Dormin@i.ua', '0e1158e3a3ab165ee039267f79b4d923');

-- --------------------------------------------------------

--

-- Структура таблиці `reply`

--

CREATE TABLE `reply` (

`id` int NOT NULL,

`content` varchar(10000) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

`pid` int NOT NULL,

`uid` int NOT NULL,

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`score` int NOT NULL DEFAULT '0',

`comments` int NOT NULL DEFAULT '0',

`is\_parent\_link` tinyint(1) NOT NULL DEFAULT '1',

`link\_id` int NOT NULL,

`reported` int NOT NULL DEFAULT '0',

`favorited` int NOT NULL DEFAULT '0',

`is\_link\_for\_union` tinyint(1) NOT NULL DEFAULT '0'

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=DYNAMIC;

--

-- Дамп даних таблиці `reply`

--

INSERT INTO `reply` (`id`, `content`, `pid`, `uid`, `score`, `comments`, `is\_parent\_link`, `link\_id`, `reported`, `favorited`, `is\_link\_for\_union`) VALUES

(6, 'ММмм...Як вкусно пахне :yummy:', 137, 8, 0, 0, 1, 137, 0, 0, 0);

-- --------------------------------------------------------

--

-- Структура таблиці `report\_link`

--

CREATE TABLE `report\_link` (

`id` int NOT NULL,

`uid` int NOT NULL,

`link\_id` int NOT NULL,

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп даних таблиці `report\_link`

--

INSERT INTO `report\_link` (`id`, `uid`, `link\_id`) VALUES

(2, 1, 152);

-- --------------------------------------------------------

--

-- Структура таблиці `report\_reply`

--

CREATE TABLE `report\_reply` (

`id` int NOT NULL,

`uid` int NOT NULL,

`reply\_id` int NOT NULL,

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Структура таблиці `report\_topic`

--

CREATE TABLE `report\_topic` (

`id` int NOT NULL,

`uid` int NOT NULL,

`topic\_name` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп даних таблиці `report\_topic`

--

INSERT INTO `report\_topic` (`id`, `uid`, `topic\_name`) VALUES

(2, 6, 'CATS');

-- --------------------------------------------------------

--

-- Структура таблиці `subscription`

--

CREATE TABLE `subscription` (

`id` int NOT NULL,

`uid` int NOT NULL,

`topic` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп даних таблиці `subscription`

--

INSERT INTO `subscription` (`id`, `uid`, `topic`) VALUES

(6, 1, 'CATS');

-- --------------------------------------------------------

--

-- Структура таблиці `topic`

--

CREATE TABLE `topic` (

`name` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

`description` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`subscribers` int NOT NULL DEFAULT '0',

`header\_image` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`creator\_uid` int NOT NULL DEFAULT '1',

`reported` int NOT NULL DEFAULT '0',

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Дамп даних таблиці `topic`

--

INSERT INTO `topic` (`name`, `description`, `subscribers`, `header\_image`, `creator\_uid`, `reported`) VALUES

('CATS', 'Content with cats', 4, 'https://cdnimg.rg.ru/img/content/178/22/40/kotik\_d\_850.jpg', 1, 1);

-- --------------------------------------------------------

--

-- Структура таблиці `user`

--

CREATE TABLE `user` (

`id` int NOT NULL,

`username` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

`password` varchar(33) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

`email` varchar(33) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

`karma` int NOT NULL DEFAULT '0',

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=DYNAMIC;

--

-- Дамп даних таблиці `user`

--

INSERT INTO `user` (`id`, `username`, `password`, `email`, `karma`) VALUES

(1, 'admin', '00a1f187721c63501356bf791e69382c', 'dormin@i.ua', 0);

-- --------------------------------------------------------

--

-- Структура таблиці `vote\_link`

--

CREATE TABLE `vote\_link` (

`id` int NOT NULL,

`uid` int NOT NULL,

`link\_id` int NOT NULL,

`up\_down` tinyint(1) NOT NULL DEFAULT '1',

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=DYNAMIC;

--

-- Дамп даних таблиці `vote\_link`

--

INSERT INTO `vote\_link` (`id`, `uid`, `link\_id`, `up\_down`) VALUES

(7, 1, 92, 1);

-- --------------------------------------------------------

--

-- Структура таблиці `vote\_reply`

--

CREATE TABLE `vote\_reply` (

`id` int NOT NULL,

`uid` int NOT NULL,

`reply\_id` int NOT NULL,

`up\_down` tinyint(1) NOT NULL DEFAULT '1',

`created` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=DYNAMIC;

--

-- Дамп даних таблиці `vote\_reply`

--

INSERT INTO `vote\_reply` (`id`, `uid`, `reply\_id`, `up\_down`) VALUES

(3, 9, 15, 0);

--

-- Індекси збережених таблиць

--

--

-- Індекси таблиці `favourite\_link`

--

ALTER TABLE `favourite\_link`

ADD PRIMARY KEY (`id`);

--

-- Індекси таблиці `favourite\_reply`

--

ALTER TABLE `favourite\_reply`

ADD PRIMARY KEY (`id`);

--

-- Індекси таблиці `link`

--

ALTER TABLE `link`

ADD PRIMARY KEY (`id`);

ALTER TABLE `link` ADD FULLTEXT KEY `title` (`title`);

ALTER TABLE `link` ADD FULLTEXT KEY `text` (`text`);

ALTER TABLE `link` ADD FULLTEXT KEY `domain` (`domain`);

ALTER TABLE `link` ADD FULLTEXT KEY `url` (`url`);

ALTER TABLE `link` ADD FULLTEXT KEY `topic` (`topic`);

--

-- Індекси таблиці `notification`

--

ALTER TABLE `notification`

ADD PRIMARY KEY (`id`);

--

-- Індекси таблиці `password\_reset`

--

ALTER TABLE `password\_reset`

ADD UNIQUE KEY `email` (`email`);

--

-- Індекси таблиці `reply`

--

ALTER TABLE `reply`

ADD PRIMARY KEY (`id`);

--

-- Індекси таблиці `report\_link`

--

ALTER TABLE `report\_link`

ADD PRIMARY KEY (`id`);

--

-- Індекси таблиці `report\_reply`

--

ALTER TABLE `report\_reply`

ADD PRIMARY KEY (`id`);

--

-- Індекси таблиці `report\_topic`

--

ALTER TABLE `report\_topic`

ADD PRIMARY KEY (`id`);

--

-- Індекси таблиці `subscription`

--

ALTER TABLE `subscription`

ADD PRIMARY KEY (`id`);

--

-- Індекси таблиці `topic`

--

ALTER TABLE `topic`

ADD PRIMARY KEY (`name`);

--

-- Індекси таблиці `user`

--

ALTER TABLE `user`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `username` (`username`),

ADD UNIQUE KEY `email` (`email`);

--

-- Індекси таблиці `vote\_link`

--

ALTER TABLE `vote\_link`

ADD PRIMARY KEY (`id`);

--

-- Індекси таблиці `vote\_reply`

--

ALTER TABLE `vote\_reply`

ADD PRIMARY KEY (`id`);

--

-- AUTO\_INCREMENT для збережених таблиць

--

--

-- AUTO\_INCREMENT для таблиці `favourite\_link`

--

ALTER TABLE `favourite\_link`

MODIFY `id` int NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

--

-- AUTO\_INCREMENT для таблиці `favourite\_reply`

--

ALTER TABLE `favourite\_reply`

MODIFY `id` int NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

--

-- AUTO\_INCREMENT для таблиці `link`

--

ALTER TABLE `link`

MODIFY `id` int NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=206;

--

-- AUTO\_INCREMENT для таблиці `notification`

--

ALTER TABLE `notification`

MODIFY `id` int NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=70;

--

-- AUTO\_INCREMENT для таблиці `reply`

--

ALTER TABLE `reply`

MODIFY `id` int NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=38;

--

-- AUTO\_INCREMENT для таблиці `report\_link`

-- ALTER TABLE `report\_link`

MODIFY `id` int NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=35;

--

-- AUTO\_INCREMENT для таблиці `report\_reply`

--

ALTER TABLE `report\_reply`

MODIFY `id` int NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT для таблиці `report\_topic`

--

ALTER TABLE `report\_topic`

MODIFY `id` int NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT для таблиці `subscription`

--

ALTER TABLE `subscription`

MODIFY `id` int NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=40;

--

-- AUTO\_INCREMENT для таблиці `user`

--

ALTER TABLE `user`

MODIFY `id` int NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=13;

--

-- AUTO\_INCREMENT для таблиці `vote\_link`

--

ALTER TABLE `vote\_link`

MODIFY `id` int NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=61;

--

-- AUTO\_INCREMENT для таблиці `vote\_reply`

--

ALTER TABLE `vote\_reply`

MODIFY `id` int NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=11;

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

**ДОДАТОК Б  
ЛІСТИНГ КОДУ З ПРИКЛАДОМ КОМАНД ДЛЯ ВЗАЄМОДІЇ З БАЗОЮ**

<?php

class Link\_model extends CI\_Model

{

public function \_\_construct()

{

$this->load->database();

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

$this->load->helper('human\_timing');

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

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

$this->load->library('hashids');

$this->hashids = new Hashids($this->config->item('hashids\_salt'), 6);

$this->load->helper('link\_submission');

$this->load->library('image\_lib');

}

public function insert\_link()

{

$this->db->where('username', $this->session->userdata('username'));

$this->db->select('id');

$this->db->limit(1);

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

$row = $query->row\_array();

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

$parsed = parse\_url($url);

$topic = str\_replace(' ', '', $this->input->post('topic'));

$topic = preg\_replace('/[^\w\_]+/u', '\_', $topic);

list($picurl, $text, $embed) = analyze\_url($url);

if (!empty($this->input->post('text'))) {

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

}

if (isset($url)) {

$data = array(

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

'url' => $url,

'text' => $text,

'embed' => $embed,

'picurl' => $picurl,

'domain' => $parsed['host'],

'topic' => $topic,

'uid' => $row['id'], //User's ID

'score' => 0,

'comments' => 0

);

}

else{

$data = array(

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

'url' => $url,

'text' => $text,

'embed' => $embed,

'picurl' => $picurl,

'topic' => $topic,

'uid' => $row['id'], //User's ID

'score' => 0,

'comments' => 0

);

}

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

$id = $this->hashids->encode($this->db->insert\_id());

$ext = pathinfo(parse\_url($picurl, PHP\_URL\_PATH), PATHINFO\_EXTENSION);

$target\_path = 'assets/img/link\_thumbnails/'.$id.'.'.$ext;

if (isset($url)) {

if (empty($picurl)){

goto a;}

else{

copy($picurl, $target\_path);

}

}

a:

$config['image\_library'] = 'gd2';

$config['source\_image'] = $target\_path;

$config['create\_thumb'] = true;

$config['maintain\_ratio'] = true;

$config['width'] = 248;

$this->image\_lib->initialize($config);

$this->image\_lib->resize();

$this->image\_lib->clear();

return $id;

}

public function insert\_reply()

{

$this->db->where('username', $this->session->userdata('username'));

$this->db->select('id');

$this->db->limit(1);

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

$row = $query->row\_array();

$content = xss\_clean($this->input->post('content'));

$content = trim($content);

$data = array(

'content' => $content,

'pid' => $this->hashids->decode($this->input->post('pid'))[0],

'uid' => $row['id'],

'score' => 0,

'comments' => 0,

'is\_parent\_link' => $this->input->post('is\_parent\_link'),

'link\_id' => $this->hashids->decode($this->input->post('link\_id'))[0]

);

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

return $this->hashids->encode($this->db->insert\_id());

}

public function up\_score()

{

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

$id = $this->hashids->decode($id)[0];

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

$this->db->set('score', 'score+1', false);

$this->db->update('link');

}

public function down\_score()

{

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

$id = $this->hashids->decode($id)[0];

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

$this->db->set('score', 'score-1', false);

$this->db->update('link');

}

public function rply\_up\_score()

{

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

$id = $this->hashids->decode($id)[0];

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

$this->db->set('score', 'score+1', false);

$this->db->update('reply');

}

public function rply\_down\_score()

{

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

$id = $this->hashids->decode($id)[0];

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

$this->db->set('score', 'score-1', false);

$this->db->update('reply');

}

public function increase\_comments()

{

$id = $this->input->post('pid');

$id = $this->hashids->decode($id)[0];

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

$this->db->set('comments', 'comments+1', false);

$this->db->update('link');

}

public function increase\_rply\_comments()

{

$id = $this->input->post('pid');

$id = $this->hashids->decode($id)[0];

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

$this->db->set('comments', 'comments+1', false);

$this->db->update('reply');

}

}

**ДОДАТОК В  
ЛІСТИНГ КОДУ ДЛЯ ФОРМИ НАДСИЛАННЯ ПУБЛІКАЦІЇ**

<div class="container-fluid">

<div class="row-fluid">

<div class="span5">

<ul class="nav nav-tabs form-tabs" id="myTab">

<?php if ($is\_text\_post) { ?>

<li><a class="submit" href="#link">Send link</a></li>

<li class="active"><a class="submit" href="#text">Send text</a></li>

<?php } else { ?>

<li class="active"><a class="submit" href="#link">Send link</a></li>

<li><a class="submit" href="#text">Send text</a></li>

<?php };?>

</ul>

<div class="tab-content">

<?php if ($is\_text\_post) { ?>

<div class="tab-pane" id="link">

<?php } else { ?>

<div class="tab-pane active" id="link">

<?php };?>

<?php echo form\_open('send');?>

<fieldset>

<table class="table table-bordered submit-form"><tr><td>

<label for="title">Title</label>

<textarea rows="2" class="span12" id="title" name="title" value="<?php echo set\_value('title');?>" placeholder="link title..."/></textarea><br />

<div style="color:red"><?php echo form\_error('title');?></div>

</td></tr></table>

<table class="table table-bordered submit-form"><tr><td>

<label for="url">URL (full address)</label>

<input autocomplete="off" required type="text" class="span12" name="url" id="url" value="<?php echo set\_value('url');?>" placeholder="the URL of the link is http://... starting with"/><br />

<div style="color:red"><?php echo form\_error('url');?></div>

<a href="javascript:void(0)" class="btn btn-primary pull-right btn-blue" id="get\_title">Suggest title</a>

</td></tr></table>

<table class="table table-bordered submit-form"><tr><td>

<label for="topic">Select or create a topic</label>

<input type="text" class="span12 topic" id="topic" value="<?php echo set\_value('topic');?>" name="topic" placeholder="the topic of the link..."/><br />

<div style="color:red"><?php echo form\_error('topic');?></div>

<!--Show popular topics-->

<div class="popular-topics">

<?php foreach($topics\_for\_header as $topic): ?>

<a href="javascript:void(0)" onclick="set\_topic(this)"><?php echo $topic['topic']; ?></a>&nbsp;&nbsp;

<?php endforeach ?>

</div>

</td></tr></table>

<table class="table table-bordered submit-form"><tr><td>

<img src="<?php echo base\_url('user/captcha');?>" />

<br/>

<label for="captcha">Verification code</label>

<input autocomplete="off" type="text" name="captcha" placeholder="enter the four characters here"/>

<div style="color:red;"><?php if(!empty($error)){echo $error;}?><?php echo form\_error('captcha');?></div>

</td></tr></table>

<button class="btn btn-primary btn-blue pull-left" type="submit" name="submit" >Send</button>

</fieldset>

</form>

</div>

<?php if ($is\_text\_post) { ?>

<div class="tab-pane active" id="text">

<?php } else { ?>

<div class="tab-pane" id="text">

<?php };?>

<?php echo form\_open('send');?>

<fieldset>

<table class="table table-bordered submit-form"><tr><td>

<label for="title">Title</label>

<textarea rows="2" class="span12" id="title" name="title" value="<?php echo set\_value('title');?>" placeholder="the title of the post..."/></textarea><br />

<div style="color:red"><?php echo form\_error('title');?></div>

</td></tr></table>

<table class="table table-bordered submit-form"><tr><td>

<label for="text">Text (optional)</label>

<textarea rows="5" class="span12" id="text" name="text" value="<?php echo set\_value('text');?>" placeholder="enter the contents... markdown formatting is supported"/></textarea><br />

<a class="link-to-guide pull-right" href="<?php echo base\_url('pages/markdown\_guide');?>">Markdown formatting</a>

</td></tr></table>

<table class="table table-bordered submit-form"><tr><td>

<label for="topic">Select or create a topic</label>

<input type="text" class="span12 topic" id="topic" value="<?php echo set\_value('topic');?>" name="topic" placeholder="the topic of the post..."/><br />

<div style="color:red"><?php echo form\_error('topic');?></div>

<!--Show popular topics-->

<div class="popular-topics">

<?php foreach($topics\_for\_header as $topic): ?>

<a href="javascript:void(0)" onclick="set\_topic(this)"><?php echo $topic['topic']; ?></a>&nbsp;&nbsp;

<?php endforeach ?>

</div>

</td></tr></table>

<table class="table table-bordered submit-form"><tr><td>

<img src="<?php echo base\_url('user/captcha');?>" />

<br/>

<label for="captcha">Verification code</label>

<input autocomplete="off" type="text" name="captcha" placeholder="enter the four characters here"/>

<div style="color:red;"><?php if(!empty($error)){echo $error;}?><?php echo form\_error('captcha');?></div>

</td></tr></table>

<button class="btn btn-primary btn-blue pull-left" type="submit" name="submit" >Send</button>

</fieldset>

</form>

</div>

</div>

</div>

<div class="live-preview"></div>