

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ДЕРЖАВНИЙ ЗАКЛАД
«ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА»**

Навчально-науковий інститут математики та інформаційних технологій

Кафедра математики та інформатики

Цуй Цзінъвей

**СИСТЕМА ВЕБ-СЕРВІСУ ДЛЯ ЦЕНТРАЛІЗОВАНОГО
АДМІНІСТРУВАННЯ ОПИТУВАНЬ**

Магістерська робота
за спеціальністю 122 «Комп'ютерні науки»

Особистий підпис – _____

Науковий керівник – _____

д.т.н., професор Ю.Г. Козуб

В.о.зав. кафедри – _____

д.т.н., професор Ю.Г. Козуб

Полтава – 2025

АНОТАЦІЯ

Цуй Цзіньвей

Тема: Система веб-сервісу для централізованого адміністрування опитувань

Установа: ДЗ ЛНУ імені Тарача Шеіченка, 2025р.

Кваліфікаційна робота містить: 71 стор., 19 рис., 2 табл., 3 додатки, 21 джерело.

Об'єкт дослідження — веб-сервіс для централізованого адміністрування опитувань.

Мета роботи — розробити веб-сервіс для централізованого адміністрування опитувань, а також компонент для CMS Joomla, що забезпечує взаємодію між веб-сервісом і доменом.

Результати — спроектовано і програмно реалізовано веб-сервіс для централізованого адміністрування опитувань, що складається з клієнтської і серверної частин. Розроблено компонент для CMS Joomla для зв'язку веб-системи і сторонніх доменів.

Ключові слова: веб-сервіс, опитування, соціальні мережі, централізована система управління, CMS Joomla, компонент, управління опитуваннями, веб-маркетинг, SEO, адміністрування опитувань

ABSTRACT

Cui Jinwei

Theme: Web service system for centralized administration of surveys

Speciality: 122 "Computer science"

Institution: Tarach Sheichenko Lviv National University, 2025

Qualification work contains: 71 pages, 19 figures, 2 tables, 3 appendices, 21 sources.

Object of research: A web service for centralized administration of surveys.

Purpose of the work: Develop a web service for centralized administration of surveys, as well as a component for CMS Joomla, which provides interaction between the web service and the domain.

Results: A web service for centralized administration of surveys, consisting of client and server parts, has been designed and programmatically implemented. A component for CMS Joomla has been developed to connect the web system and third-party domains.

Keywords: web service, survey, social networks, centralized management system, CMS Joomla, component, survey management, web marketing, SEO, survey administration

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	9
1.1. Роль опитувань у WEB-маркетингу	9
1.2. Соціальні мережі та роль голосувань в них	11
1.3. Поняття централізованої системи управління і її переваги	14
1.4. Огляд систем опитувань	15
1.5. Висновки до розділу 1	20
РОЗДІЛ 2. ВИБІР ЗАСОБІВ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ	21
2.1. Вибір методу розробки	21
2.2. Вибір програмних засобів	22
2.3. Вибір структури БД.....	29
2.4. Архітектурний патерн	31
2.5. Висновки до розділу 2.....	32
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ	33
3.1. Схема взаємодій користувачів та адміністраторів вебсистеми	33
3.2. Розробка бази даних	34
3.3. Розробка веб-сервісу голосування.....	36
3.4. Розробка компонента для CMS Joomla	44
3.5. Результати роботи	45
3.6. Висновки до розділу 3	47
ВИСНОВКИ	49
СПИСОК ЛІТЕРАТУРИ	50
ДОДАТКИ	52
ДОДАТОК А. Код файлу golosAddController.php	52
ДОДАТОК Б. Код файлу golosAddModel.php	58
ДОДАТОК В. Код файла TransDomainScriptView.php	69

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CMS	-	Content Management System;
MVC	-	Model-View-Controller;
MVP	-	Presenter-Model-View;
SEO	-	Search Engines Optimization;
SMM	-	Social Media Marketing;
AIC	-	автоматизована інформаційна система;
АСУ	-	автоматизована система управління;
ВБД	-	внутрішня база даних;
ЕОМ	-	електронна обчислювальна машина;
ЕС	-	експертна система;
ІКТ	-	інформаційно-комунікаційні технології;
ІТ	-	інформаційні технології;
СКБД	-	система керування базами даних.

ВСТУП

Мережа Інтернет стала загальновизнаним фактором ділового та суспільного життя. Широка поширеність і збільшена пропускна здатність створюють умови, при яких вигідно вирішувати багато задач при допомоги інтернет-технологій.

Однак Інтернет об'єднує в собі багато різних платформ, а інформація міститься в різноманітних джерелах даних. Тому актуальна проблема зв'язку таких різнорідних даних, а також створення способу, який дозволяє отримувати їх у вигляді зручному для подальшої обробки.

Концепція веб-сервісів (Web Services) покликана вирішити цю задачу об'єднання, інтеграції різнорідних систем на основі відкритих стандартів.

На сьогоднішній день величезна кількість людей користується послугами web-сервісів.

Метою створення веб-сервісу для централізованого адміністрування опитуваннями є SEO просування сторонніх доменів за рахунок підвищення кількості відвідувань, так як система орієнтована на соціальні мережі, де побуває велика кількість користувачів, і організація зворотного зв'язку з користувачами ресурсів.

Дана система є актуальною на увазі актуальності питання про SEO просування сайту, коли від кількості відвідувань залежить порядковий номер сайту, при видачі відповіді пошуковим сервісом користувачеві на пошуковий запит. Маючи просту взаємодію з соціальними мережами, опитування викликає до себе увагу користувачів. На всіх етапах роботи веб-сервіс забезпечує безпеку і незмінюваність даних, що забезпечує максимально чесні опитування. Простий і інтуїтивно зрозумілий інтерфейс дозволить створювати і управляти величезною кількістю опитувань відразу на декількох ресурсах, а всі результати будуть зібрані на одній сторінці.

Мета магістерської роботи — розробити веб-сервіс для централізованого адміністрування опитувань, а також компонент для CMS Joomla, що забезпечує взаємодію між веб-сервісом і доменом.

Об'єкт дослідження — веб-сервіс для централізованого адміністрування опитувань.

Веб-сервіс виконує завдання створення опитування з широкими настройками, редагування, видалення або їх блокування, перегляд інформації і результатів опитувань, відображення і редагування профілів користувачів системи, реєстрації нових користувачів, взаємодії з компонентом для CMS Joomla, відправка і отримання даних про опитування.

Компонент для CMS Joomla забезпечує виконання завдань: отримання інформації, сформованої системою, формування з отриманої інформації опитувань і їх результатів, авторизації користувачів через соціальні мережі, надсилання даних системі для фіксації голосів, налаштування прив'язки опитування до пунктів меню, відображення опитувань (всіх, з фільтром по мові або одне певне), авторизації та процесу опитування користувачів, формування публікацій і пропозиції користувачам поділитися нею в соціальних мережах.

У якості *предметної області* опитувань розглядається веб-сервіс централізованого управління опитуваннями. Даний сервіс зручний для проведення опитувань на сайтах під керуванням CMS Joomla. І може використовуватися власниками сайтів навіть з мінімальним рівнем компетентності у використанні комп'ютера.

Для реалізації поставленої мети необхідно вирішити наступні завдання:

1. Провести аналіз відомих рішень;
2. Вибрати методи вирішення задачі;
3. Розглянути найпопулярніші і затребувані CMS платформи і мови для розробки web-сервісів;
4. Вибрати і вивчити систему для розробки web-сервісу;
5. Спроектувати і розробити web-сервіс для централізованого адміністрування опитувань.;

6. Виконати проєктування компонент для CMS Joomla, що забезпечить взаємодію між веб-сервісом і ресурсом користувача.

Результатом роботи є програмна реалізація веб-сервісу для централізованого адміністрування опитувань, яка складається з клієнтської і серверної частин. Розробка компоненту CMS Joomla для зв'язку веб-системи і сторонніх доменів.

Структура роботи складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаних джерел, додатків.

У першому розділі проводиться інформаційний огляд сучасних тенденцій соціальних мереж, веб-маркетингу та розглядаються поняття централізованої система, її переваги та ключові особливості.

У другому розділі описано вже реалізовані проєкти створення опитувань, особливості розміщення їх на сайтах та їх інтеграції з соціальними мережами, переваги та недоліки вже реалізованих проєктів. На основі аналізу результатів описано постановку задачі.

Третій розділі присвячено вибору методу вирішення поставленої задачі. Обрано метод розробки, проведено аналіз та вибрано технології за допомогою яких реалізовано систему, розглянуто підходи до проєктування бази даних, описано структури використаного підходу, проведено аналіз та вибір архітектурних патернів для основи проєкту, розглянуто деталі програмної реалізації сервісу та компоненту. Наведено результати роботи з сервісом.

У додатках надано лістинги коду фрагментів сайту і файлів.

РОЗДІЛ 1

ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Роль опитувань у WEB-маркетингу

У сучасному світі більшість жителів планети мають доступ до мережі Інтернет. За даними статистичних агентств, саме за допомогою глобальної павутини більшість сучасних людей займається пошуком потрібної інформації. Це можуть бути новини, вакансії, послуги, розваги або певні товари.

Через постійно зростаючої кількості користувачів глобальної павутини, все більше фірм і організацій вважають за краще вести свій бізнес в Інтернеті. В даному випадку сайт виконує, як ознайомчу, так і рекламну функцію. Однак саме по собі наявність в тій чи іншій компанії веб-сторінки не є гарантом успішної діяльності. Сайт повинен працювати цілодобово, забезпечуючи безперервний потік відвідувачів. З цієї причини першорядне значення набуває оптимізація сайту і його просування.

Сучасне просування сайтів є складним і багатограним процесом, кінцевим результатом якого повинно стати збільшення цільової аудиторії і загального числа відвідувачів того чи іншого порталу.

WEB-маркетинг – це набір інструментів, методів і підходів різного напрямку реклами в мережі Інтернет для отримання бажаного відгуку від цільової аудиторії в залежності від поставлених перед сайтом завдань.

Зазвичай, використання терміну «WEB-маркетинг» являє собою використання в інтернет-просторі маркетингових стратегій прямого відгуку, які використовуються в поштових розсилках або різного роду телевізійних рекламних роликах.

WEB-маркетинг став особливо популярним з розширенням доступу до мережі Інтернет. Він являє собою невід'ємну частину будь-якої нормальної

маркетингової кампанії. Сегмент інтернет-маркетингу і реклами дуже швидкими темпами зростає в споживчому секторі.

Одним з популярних видів просування веб-ресурсів є маркетинг в соціальних мережах (SMM – Social Media Marketing) – набір методів і способів оптимізації сайту в різних соціальних інтернет-ресурсах, таких як соціальні мережі, тематичні форуми, веб-портали. На відміну від більшості традиційних підходів до просування веб-ресурсів в мережі Інтернет, SMM використовує метод залучення трафіку користувачів на просувний сайт, не за допомогою пошукових систем, а безпосередньо з соціальних веб-ресурсів.

Маркетинг за допомогою соціальних мереж є однією з гілок розвитку SEO. Основним завданням цього виду маркетингу є зацікавити живу людину без участі пошукових систем.

Свою популярність даний метод просування веб-ресурсів почав набирати з появою великих соціальних мереж, таких як Twitter або Facebook, кількість користувачів яких налічує мільярди. Саме соціальні мережі з високою кількістю активних користувачів є основними майданчиками для проведення соціального медіа-маркетингу. Найпопулярнішим видом просування ресурсів є розміщення публікацій в соціальних мережах, які будуть доступні іншим користувачам мережі і виявляться цікавими якомога більшій кількості користувачів. Якщо така публікація зацікавить людини, то він швидше за все проявить активність на веб-ресурсі, просуваючи його.

Самі соціальні мережі сприяють web-маркетингу на їх базі. Більшість популярних соціальних мереж надають розробникам API-інтерфейси для взаємодії веб-додатків з соціальною мережею. Це, наприклад, кнопки «Мені подобається», голосування та коментарі, які розміщуються на сайтах, просуваються і дозволяють автоматично публікувати в соціальній мережі посилання на статті або медіа-контент сайту. Існують цілі веб-сервіси, які надають клієнтам модулі з уже готовими і налаштованими кнопками багатьох соціальних мереж.

1.2 Соціальні мережі та роль голосувань в них

Соціальна мережа - платформа, онлайн-сервіс або веб-сайт, призначені для побудови, відображення і організації соціальних взаємовідносин, візуалізацією яких є соціальні графи.

Цільовими соціальними мережами, для розроблюваної системи є: Фейсбук (Facebook); Інстаграм (Instagram); Твіттер (X / Twitter) та інші (рис. 1.1).

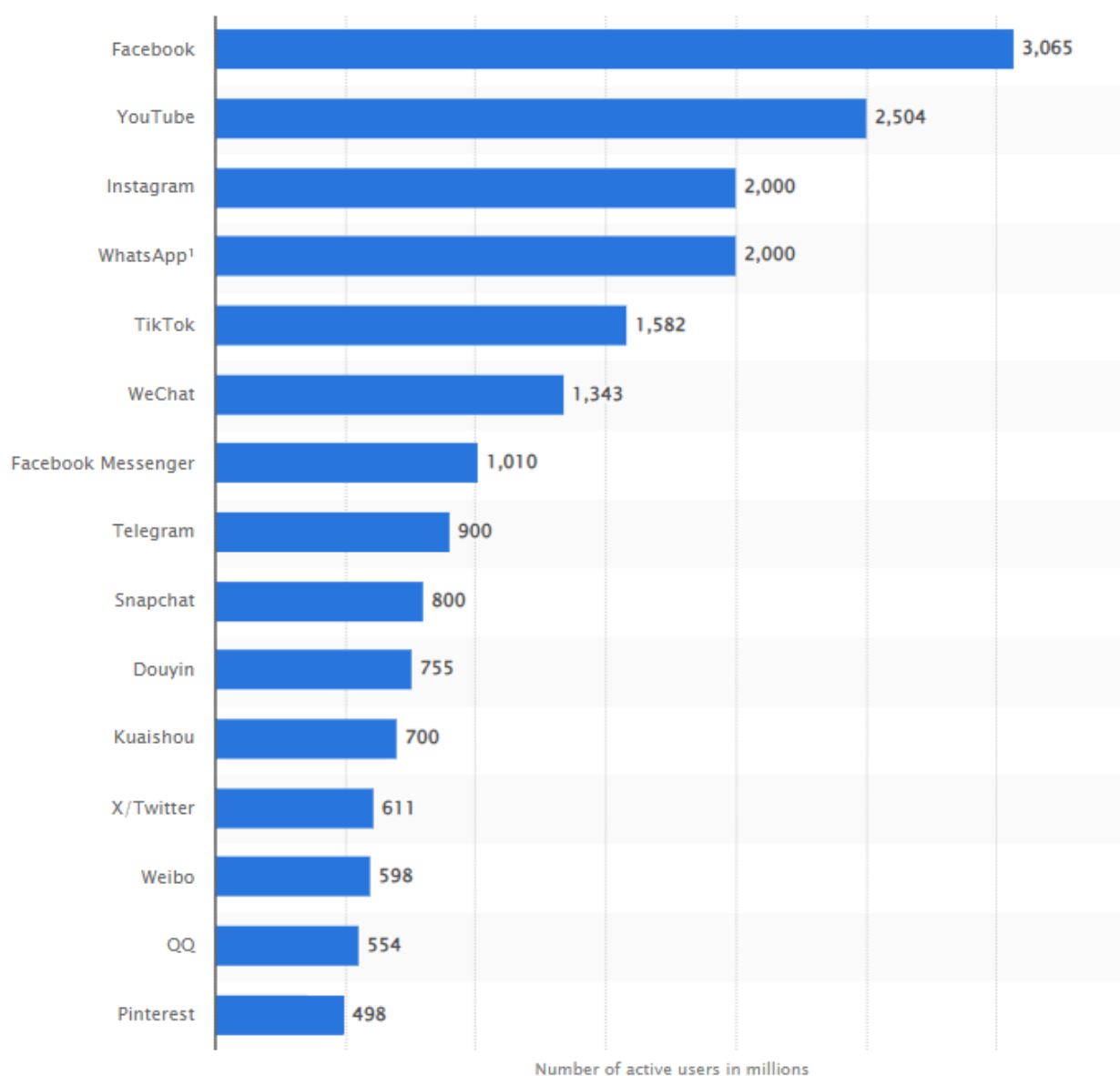


Рисунок 1.1 – Найпопулярніші соціальні мережні станом на квітень 2024 р. за кількістю активних користувачів (млн)

Твіттер (Twitter) (після ребрендинга – X) – соціальна мережа для публічного обміну короткими (до 140 символів) повідомленнями за допомогою веб-інтерфейсу, SMS, засобів миттєвого обміну повідомленнями або сторонніх програм-клієнтів для користувачів інтернету будь-якого віку. Публікація коротких заміток в форматі блогу отримала назву «мікроблогінг». Користування сервісом безкоштовно. Користування за допомогою SMS тарифікується оператором згідно з тарифним планом користувача. Власником системи «Твіттер» є компанія Twitter Inc. Станом на 1 січня 2021 року сервіс налічує більше 400 млн користувачів. 100 мільйонів користувачів проявляють активність хоча б раз на місяць, з них 50 мільйонів користуються «Твіттером» щодня. 55% користуються «Твіттером» на мобільних гаджетах, близько 550 мільйонів унікальних відвідувань отримує за місяць безпосередньо сайт twitter.com.

Фейсбук (Facebook) – одна з найбільших соціальних мереж в світі. Була заснована 4 лютого 2004 року Марком Цукербергом і його сусідами по кімнаті під час навчання в Гарвардському університеті – Едуардо Саверіном, Дастіном Московіцем і Крісом Хьюзом. Спочатку веб-сайт був названий Thefacebook і був доступний тільки для студентів Гарвардського університету, потім реєстрацію відкрили для інших університетів Бостона, а потім і для студентів будь-яких навчальних закладів США, що мають електронну адресу в домені .edu.

Починаючи з вересня 2006 року сайт доступний для всіх користувачів Інтернету у віці від 16 років, що мають адресу електронної пошти. На липень 2014 року аудиторія Facebook склала 1,32 мільярда користувачів – це ті, хто заходив на сайт хоча б раз на місяць або за вказаний проміжок часу був зафіксований за допомогою кнопки Like і стежать cookie. Добова активна аудиторія в березні склала 720 млн чоловік – стільки фіксується стежить мережею Facebook щодня. Близько 810 млн осіб на місяць використовують мобільний додаток Facebook. Кожен день в соціальній мережі користувачі залишають 3,2 мільярда «лайків» і коментарів і публікують 300 мільйонів фотографій. На сайті зафіксовано 125 мільярдів «дружніх зв'язків» (на 31 грудня

2011 р Було 100 млрд). Кількість переглядів сторінок сайту в жовтні-грудні 2024 року склало більше 3 трильйонів.

На рисунку 1.2 можна побачити статистику популярності соціальної мережі Facebook на період за 2008 – 2024 роки [1].

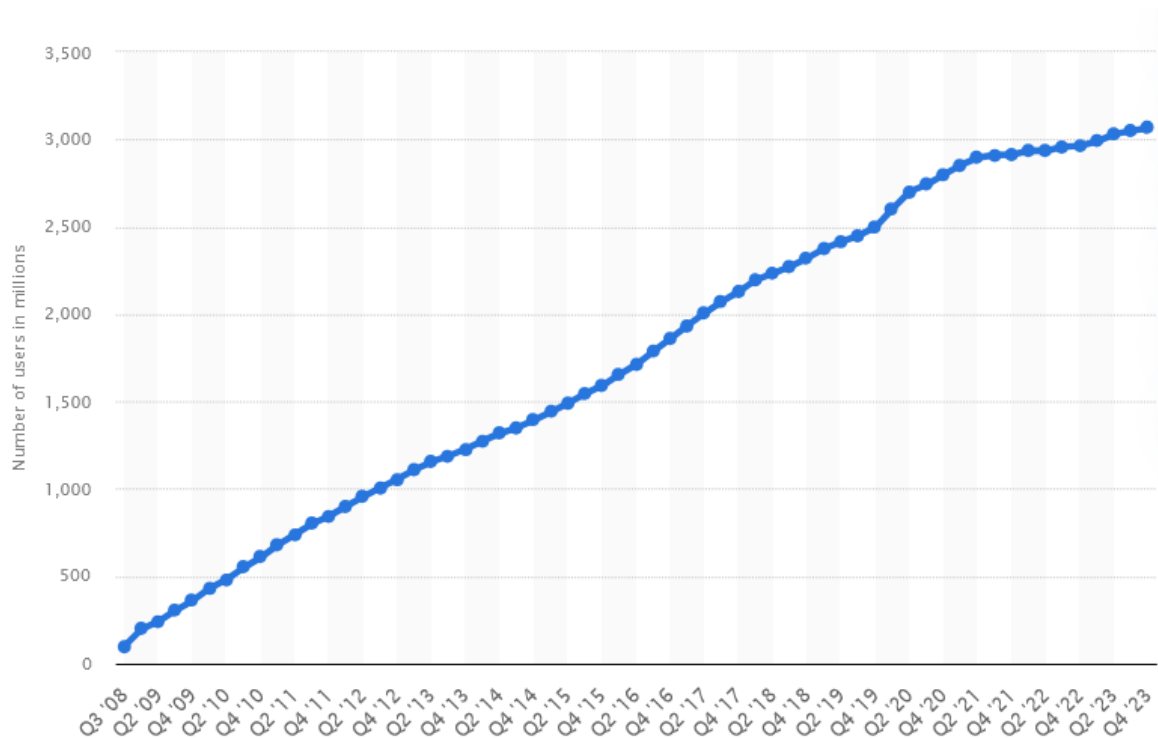


Рисунок 1.2 – Статистика Facebook на період 2008 – 2024 рр.

Соціальні мережі є потужним інструментом маркетингових досліджень, оскільки користувачі добровільно публікують інформацію про себе, своїх поглядах, інтересах, перевагах і так далі. Зважаючи на це рекламодавці можуть досить чітко визначати, які саме користувачів зацікавить їх оголошення, і направити свої рекламні оголошення конкретним користувачам, в залежності від інформації в їх профілях (вік, стать, місце проживання та інше). Такий тип реклами отримав назву таргетованої.

Основний наголос в соціальному маркетингу робиться на створенні контенту, який люди будуть поширювати через соціальні мережі самостійно, вже без участі організатора. Вважається, що повідомлення, що передаються по соціальним мережам, викликають більше довіри у потенційних споживачів

послуги. Це пов'язується з рекомендаційною схемою поширення в соціальних медіа за рахунок соціальних зв'язків, що лежать в основі взаємодії [4].

Важливо відзначити, що соціальний маркетинг застосовується не тільки на товари і послуги. Активно використовують соціальний маркетинг засоби масової інформації. Вони створюють свої акаунти в соціальних мережах, розміщують свій контент і тим самим збирають передплатників (читачів свого продукту).

Дану технологію можна використовувати і в системі, що розробляється. Для збільшення відвідувань сайтів, на яких розміщено голосування.

Іншим способом залучення клієнтів через соціальні мережі є створення компаніями спільнот в соціальних мережах. Такі спільноти дозволяють доносити нову інформацію до користувачів, яких найімовірніше зацікавить продукція або послуги компанії.

Голосування – спосіб прийняття рішення групою людей (зборами, електоратом), при якому загальна думка формуються шляхом підрахунку голосів членів групи. Голосуванню зазвичай передують обговорення.

Інтернет-голосування застосовується в різних офіційних і неофіційних опитуваннях.

1.3 Поняття централізованої системи управління і її переваги

Централізована система управління передбачає реалізацію всіх процесів управління в єдиному центральному керівному органі, який здійснює обробку інформації, що надходить від усіх об'єктів, про їх стан. При виробленні керуючої інформації для кожного з елементів управління в централізованій структурі враховується інформація стану всіх об'єктів.

Переваги централізації полягають в наступному:

- Дозволяє забезпечити високу узгодженість дій;
- Покращує контроль діяльності користувачів;
- Скорочує кількість помилок при прийнятті рішень;

- У концентрації процесу прийняття рішень в руках тих, хто краще знає загальну ситуацію, має більший кругозір, знання, досвід;
- В усуненні невиправданого дублювання управлінських функцій, що приводить до економії відповідних витрат;
- Дозволяє об'єднати всі дані які надходять з різних вузлів в одному місці, що значно спрощує роботу;
- Дозволяє управляти всіма необхідними даними, без потреби звернення до окремих вузлів.

Кожна система управління повинна бути унікальна для кожного окремого випадку. Тому певного підходу централізації управління немає. Але основний підхід зводиться до того, що всі керуючі модулі та системи централізуються в одному місці, що значно спрощує управління.

Як приклад організації централізованої системи управління можна розглянути систему управління контентом Drupal.

Для простих користувачів Drupal надає простий, інтуїтивно зрозумілий інтерфейс, що дозволяє з легкістю додавати, видаляти або модифікувати матеріал на сайті, не вимагаючи знань мови HTML і позбавляючи від необхідності постійно звертатися до веб програміста для внесення змін на сайт.

1.4 Огляд систем опитувань

У процесі дослідження існуючих рішень для маркетингу в соціальних мережах були виявлені три веб-додатки для проведення голосувань за участю соціальних мереж:

- проєкт «Ізбірнет»;
- проєкт Дмитра Денисова «SocialVote»;
- компонент JsosialVotes.

Кожен з досліджених програмних продуктів має свої функціональні характеристики та особливості роботи з ними. Далі розглянемо основні функціональні характеристики кожного з продуктів.

Проект «Ізбірнет» (рис. 1.3) – має можливість створення голосувань з необмеженою кількістю варіантів відповідей. Зберігає дані голосувань без участі бази даних. Має автоматичний підрахунок голосів на основі даних, отриманих від соціальних мереж. Для голосування використовуються соціальні мережі Instagram, Twitter, Google+. Генерує html-код для розміщення голосувань на інших сайтах. Користувач має можливість вибрати тільки один варіанту відповіді для голосування.

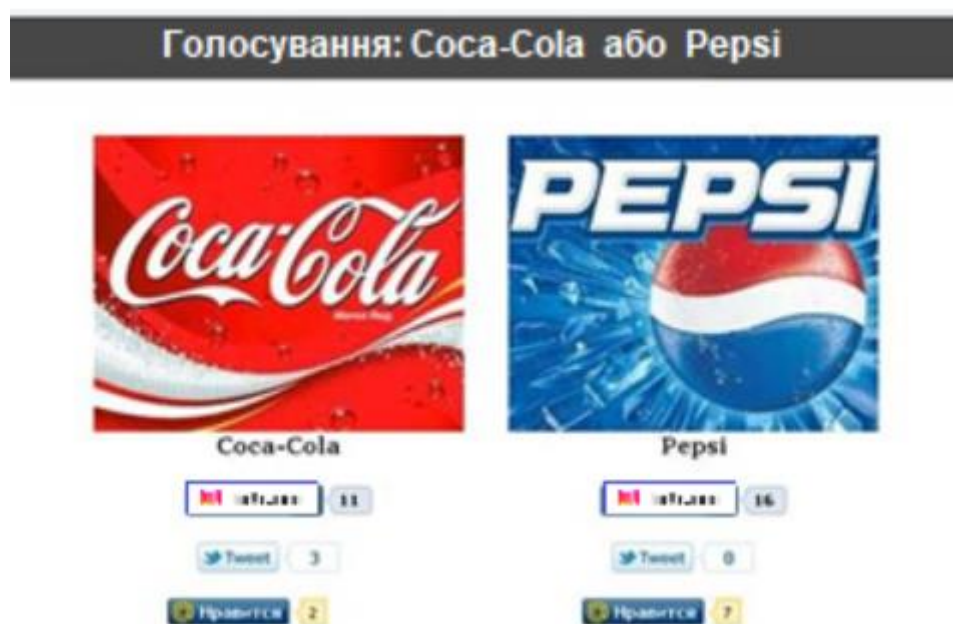


Рисунок 1.3 – Проект «Ізбірнет»

Проект «SocialVote» (рис.1.4) – Має можливість створення голосувань з максимально сьома варіантами відповіді. Збереження даних голосувань проходить за участю бази даних або без її участі. Підрахунок голосів проводиться з бази даних або на основі даних, отриманих від соціальних мереж. Для голосування використовуються соціальні мережі Instagram, Twitter (X), Facebook. Для розміщення голосувань на інших сайтах генерується html-код. Користувач може вибрати тільки один варіант відповіді для голосування. Має захист від повторних голосувань;



Рисунок 1.4 – Проєкт SocialVote

Компонент «JsocialVotes» (рис. 1.5) – Має можливість створення голосувань. Максимальна кількість варіантів відповіді у голосуванні – 5. Підтримується CMS Joomla. Дані голосувань зберігаються в базі даних. Має адміністративну панель для управління голосуваннями. Користувач може обрати тільки один варіант відповіді для голосування. Для голосування використовуються соціальні мережі Instagram, Twitter (X), Facebook, Google+. Сервіс має захист від повторних голосувань.



Рисунок 1.5 – Компонент JsocialVotes

На основі огляду функціональних характеристик, характеристик безпеки, способів адміністрування, підтримки системи і інтегрування голосування в веб-ресурси були виявлені основні недоліки та проблеми існуючих рішень. Розглянемо їх більш детально.

Всі розглянуті рішення мають обмеження на кількість варіантів відповідей, використання різних видів медіа-контенту в вигляді варіантів відповідей, можливість вибору декількох відповідей одночасно. Логічно, що в будь-яких голосуваннях або конкурсах не може бути обмежень на кількість варіантів для голосування, може існувати кілька відповідей на поставлене запитання. Також, до переліку можливих елементів, які будуть привертати увагу користувачів в соціальних мережах слід додати відео-контент. Таким чином, можна проводити онлайн-конкурси і голосування, в яких предметами оцінки будуть фотографії або відео-матеріал, які цікавлять користувачів набагато більше, ніж просто текстова інформація. Всі розглянуті системи дозволяють використовувати тільки зображення у вигляді медіа-контенту, кількість варіантів відповідей варіюється від двох до семи, а голосування налаштовані на можливість вибору тільки однієї відповіді;

Крім основної – соціальної характеристики, програмні продукти повинні гарантувати чесність проведення голосувань, виключати легкі способи отримання перемоги в голосування нечесним шляхом. Будь який захист від повторних багаторазових голосувань, який використаний в реалізації існуючих систем є неефективним. У розглянутих системах використовується три види захисту: блокування по IP-адресою, за логіном зареєстрованого користувача, за допомогою cookies-файлів. Всі вони мають недоліки і певні проблеми. IP-адресу можна легко змінити або скористатися проху-серверами. Також можлива ситуація, коли під одним IP-адресою в мережі Internet знаходиться велика кількість комп'ютерів. У такій ситуації проголосувати зможе тільки одна людина, після чого не можна буде проголосувати з жодного іншого комп'ютера такої мережі. На сайті можна зареєструватися багаторазово з різними логінами, або ж скористатися спеціальними програмами-роботами, які можуть реєструвати

облікові записи на сайті автоматично. Головною проблемою використання для захисту cookies-файлів є те, що файли cookies можна видаляти або користуватися кількома різними браузерами, через що користувачі можуть без проблем багаторазово брати участь в одному і тому ж голосуванні.

Деякі з програмних рішень зберігають дані в звичайних текстових файлах в заздалегідь строго зазначеному форматі, а голоси підраховують безпосередньо за допомогою соціальних мереж. Такий спосіб збереження даних є незручним для людини, яка буде створювати голосування, адже їй потрібно буде знати всі нюанси настройки голосувань, з'являється велика ймовірність припуститися помилки, яку буде дуже складно знайти. Також, файл можна дуже легко випадково видалити, через що втратити дані. Через відсутність бази даних, в двох розглянутих рішеннях використовується алгоритм підрахунку голосів за допомогою соціальних голосувань. Цей алгоритм заснований на тому, що підраховується кількість унікальних посилань, розміщених в соціальних мережах. По-перше, ця інформація часто може оновлюватися з затримками. По-друге, результат голосування, заснований на інформації з соціальних мереж не відповідає дійсності. З одного боку, одна і та ж людина могла розмістити безліч посилань, кожне з яких враховується як відданий голос за той чи інший варіант відповіді в голосуванні. З іншого боку, після голосування користувачі можуть видаляти повідомлення, розміщені в них в соціальних мережах, результатом чого є зменшення кількості голосів. Таким чином, використовуючи дані соціальних мереж для підрахунку голосів, є ризик стати жертвою нечесного голосування або неврахування великої кількості голосів. Тому ідеальним варіантом є реалізація такого алгоритму підрахунку голосів, який не залежить від даних сторонніх ресурсів і від дій користувачів.

Розглянуто рішення, крім компонента для Joomla мають незручний спосіб вбудовування голосування на сторонні сайти. Таким способом є розміщення голосувань в фреймах. По-перше, багато пошукові системи відмовляються реєструвати в своїй базі даних сторінки сайтів, які містять в собі фрейми. По-друге, неможливо однозначно ідентифікувати посилання на конкретне

голосування, так як фрейм має одну адресу, а сторінка, яка містить фрейм - іншу. При цьому у фреймі може міститися кілька голосувань, кожне з яких відображаються після натискання на посилання. Завершальним етапом голосування є розміщення в соціальній мережі повідомлення з посиланням на голосування. У випадку з фреймами це посилання буде на загальну сторінку, яка містить фрейм, але не на конкретне голосування у фреймі. Або ж можна отримати посилання на конкретне голосування у фреймі, але воно буде містити доменну адресу сервісу голосувань, а не сайту, на якому розміщений фрейм.

Жоден з варіантів розглянутих систем не реалізує в повній мірі централізоване управління опитуваннями. Єдиним способом збереження даних централізовано і поширення голосувань, звертаючись до центрального сервісу, є фрейми. Цей варіант – є варіант псевдоцентралізованої системи, так як сформовані сторінки завжди знаходяться на сайті системи управління, а вбудовування їх за допомогою фреймів має ряд проблем і недоліків, розглянутих в попередньому пункті. Також при такому способі централізації зберігання голосувань немає ніякого розподілу голосувань між користувачами, доменами, блокування «чужих» голосувань, зручного управління.

1.5 Висновки до розділу 1

На основі проведеного аналізу можна зробити висновок, що використання соціальних мереж дозволяє отримати найбільш повний охоплення респондентів опитувань як відкритих так і анонімних.

Компонент для CMS Joomla повинен отримувати сформовану системою інформацію, формувати з отриманої інформації голосування і їх результати, виконувати авторизацію користувачів через соціальні мережі, відправляти дані системі для фіксації голосів, мати можливість прив'язки голосування до пунктів меню, відображати голосування (все, з фільтром з мови або одне певне), авторизовані користувачі повинні мати можливість голосувати, формувати публікацію і пропонувати користувачеві поділитися нею в соціальних мережах.

РОЗДІЛ 2

ВИБІР ЗАСОБІВ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Вибір методу розробки

Сервіс централізованої системи управління голосуваннями призначений для автоматизації та централізації управління голосуваннями за допомогою соціальних мереж, надання програмного інтерфейсу для забезпечення проведення голосування на сторонніх ресурсах за допомогою компонента CMS Joomla.

Система є web-орієнтованою, має клієнт-серверну архітектуру. Клієнтом виступає web-браузер Інтернет-користувача або корпоративного користувача системи. Серверна частина складається з бізнес-логіки, інтерфейсної частини і бази даних. Розглянемо кожен з них детальніше.

Бізнес-логіка системи — знаходиться в програмному коді, що виконується на стороні серверів виконавця.

Інтерфейсні частини системи зберігаються в програмному коді, що виконується на стороні клієнтських серверів.

База даних — зберігається в файлах на стороні виконавця. Взаємодія з нею технічно забезпечується системою управління базою даних MySQL, встановленою також на стороні серверів виконавця.

Взаємодія інтерфейсної частини і бізнес-логіки системи, а також бізнес-логіки системи з СКБД реалізується засобами інтеграції через Інтернет-протоколи передачі даних, таких як HTTP та HTTPS. Взаємодія реалізується засобами інтеграції через Інтернет-протоколи передачі даних.

Система передбачає існування наступних категорій користувачів з певним переліком їх ролей (функцій):

- Корпоративний користувач – користувач адміністративного порталу системи, який зареєстрував хоча б одне доменне ім'я в системі;

– Головний адміністратор – корпоративний користувач, який має право реєструвати нових користувачів системи, переглядати історію (логи) дій всіх користувачів системи, налаштовувати систему.

Система повинна забезпечувати високий відсоток гарантії чесних голосувань шляхом перевірок користувача що голосує, на різних стадіях голосування.

Система повинна забезпечувати неможливість зміни даних, що пересилаються від клієнтського ресурсу до централізованого сервера або гарантувати, що можливі навмисні зміни, даних не зможуть вплинути на хід голосування.

2.2 Вибір програмних засобів

Для реалізації централізованої системи управління голосуваннями були обрані такі технології, як HTML, CSS, JavaScript, jQuery, Bootstrap, PHP, MySQL. Розглянемо кожну з них більш детально.

HTML – стандартна мова розмітки документів в інтернеті [5]. Більшість веб-сторінок містять опис розмітки на мові HTML (або XHTML)[6]. Мова HTML інтерпретується браузером; отриманий в результаті інтерпретації форматований текст відображається на екрані монітора комп'ютера або мобільного пристрою [7].

Дана мова розмітки була обрана за її універсальність і простоту використання. Завдяки чому можна реалізувати і в подальшому підтримувати необхідну структуру.

CSS – формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки [8], була обрана на додаток мови розмітки HTML.

Так як ці дві мови між собою тісно пов'язані і доповнюють одна одну, буде можливо організувати інтуїтивно зрозумілий і приємний для ока інтерфейс, зручний для користувача будь-якого рівня.

JavaScript – прототипно-орієнтована сценарна мова програмування [9]. Є реалізацією мови ECMAScript. JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить в браузерях як мова сценаріїв для додання інтерактивності веб-сторінок.

Дана мова була обрана за її простоту використання і доопрацювання. Вона дозволяє управляти каскадними таблицями CSS і розміткою HTML, роблячи сторінки більш «живими», що дозволяє користувачеві комфортніше працювати з системою.

jQuery – бібліотека JavaScript, що фокусується на взаємодії JavaScript і HTML [10]. Бібліотека jQuery допомагає легко отримувати доступ до будь-якого елементу DOM, звертатися до атрибутів і вмісту елементів DOM, маніпулювати ними. Даний продукт доповнює мову сценаріїв Javascript, і спрощує доступ до каскадним таблиць CSS і розмітці HTML.

Bootstrap – вільний набір інструментів для створення сайтів і веб-додатків [11]. Включає в себе HTML і CSS шаблони оформлення для типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсів, включаючи JavaScript розширення.

Це доповнення до каскадних таблиць CSS дозволить підтримати весь проєкт в одному стилі, організовуючи при цьому зручний інтерфейс.

Перш за все під час процесу проєктування потрібно вибрати програму, яка дозволяє визначати, створювати, запитувати, оновлювати та керувати базами даних – систему керування базами даних (СКБД). Існують такі популярні СКБД як: MySQL, PostgreSQL, MongoDB, MariaDB, Microsoft SQL Server, Oracle, Sybase, Microsoft Azure та IBM DB2.

MySQL – це найпопулярніша база даних з відкритим вихідним кодом, відмінна для сайтів CMS та блогів.

Oracle – об'єктно-реляційна СКБД, написана на мові C ++. Якщо проєкт має великий бюджет, це повний спектр послуг, що забезпечує високий рівень обслуговування клієнтів та надійність.

IBM DB2 – це сімейство продуктів IBM з серверних баз даних, створені для обробки передових «великих даних» та аналітики.

Sybase – реляційний модель бази даних серверного сервісу для підприємств, яка в основному використовується в ОС Unix, яка була першою на рівні підприємства на базі СКБД для Linux.

MS SQL Server – розроблена компанією RDBMS база даних на рівні підприємства, яка підтримує як SQL, так і архітектуру NoSQL.

Microsoft Azure – платформа хмарних обчислень, яка підтримує будь-яку операційну систему, і дозволяє зберігати, обчислювати та масштабувати дані в одному місці.

MariaDB – це розширена, опущена версія MySQL.

PostgreSQL – об'єктно-реляційна СКБД на рівні підприємства, що використовує процедурні мови, такі як Perl і Python, на додаток до коду SQL-рівня.

MongoDB – найпопулярніша система NoSQL, особливо серед стартапів. База даних, орієнтована на документи, роботу з JSON-подібними документами в динамічних схемах замість реляційних таблиць, що використовуються на зворотному кінці сайтів, таких як Craigslist, eBay, Foursquare. MongoDB має відкритий код, так що використання – безкоштовне, з хорошим обслуговуванням клієнтів.

Під час проектування розглядались переваги та недоліки двох основних технологій, що могли підійти для даного проєкту MongoDB та MySQL (табл. 2.1).

З точки зору розробника важливо, що MongoDB допомагає розробляти програми швидше, оскільки не використовує таблиці. Це означає, що процедури збереження більше не потрібні. Це дає перевагу розробникам, оскільки раніше таблиці були переведені на об'єктну модель, перш ніж вони могли бути використані в додатку.

Таблиця 2.1

Порівняльна характеристика MongoDB та MySQL

Функція	MongoDB	MySQL
Тип (1)	Безкоштовний та відкритий вихідний текст для крос-платформних документів (0)	Система управління реляційною базою даних з відкритим кодом (1)
Продуктивність (1)	Високий рівень, але менш ніж MySQL (0.8)	Високий рівень, завдяки своїй унікальній архітектурі двигуна зберігання MySQL (1)
Складність у взаємодії (0.5)	Ні, тому що немає поняття таблиці. Робота з JSON. (0.5)	Так, з'єднання баз даних SQL складніше (0.2)
Використання пам'яті (1)	Середній рівень, він використовує більше пам'яті, тому що він повинен зберігати імена ключових слів у кожному документі. (0.7)	Низький рівень (1)
Швидкість (1)	Високий рівень Труднощі можуть мати великі обсяги даних (0.9)	Високий рівень SQL Queries можна використовувати для швидкого та ефективного отримання великих обсягів записів з бази даних. (1)
Масштабування (1)	Горизонтально (1)	Вертикально (0.5)
Технології, які використовуються спільно (1)	Node.js (0)	PHP (1)
Підтримка (0.5)	Вузька (0.3)	Широка (0.5)
Результати:	4.2	6.2

Це може бути дуже корисним для створення таких програм, як відстеження помилок, дискусійні форуми, реклами тощо. Однак він може не в змозі задовольнити всі потреби, які підтримує реляційна база даних.

Розробка додатків повинна забезпечувати високу продуктивність, тому MySQL гарний вибір, оскільки він розроблен для задоволення навіть самих вимогливих додатків, одночасно забезпечує оптимальну швидкість, повнотекстові індекси та унікальні кеш-пам'яті для підвищення продуктивності. Також важливим фактом є те, що PHP у парі зі MySQL широко використовуються для управління базами даних та додатками. Тому було обрано саме MySQL.

Для реалізації серверної сторони розглянуто використання PHP або Node.js. Щоб вибрати одну з них, було проведено аналіз цих технологій на переваги та недоліки (табл. 2.2).

Node.js – відкрите джерело, крос-платформне середовище виконання JavaScript для виконання JavaScript коду на стороні сервера.

Переваги node.js:

- Запускає сценарії на стороні сервера для створення динамічного вмісту веб-сторінок, перш ніж сторінка надсилатиметься веб-браузеру користувача;
- оснащений двигуном Google V8 JS, що є тим самим двигуном, який використовується в веб-браузері Google Chrome;
- Це середовище виконання JS не потребує окремого веб-сервера або інших залежностей, у ньому є вбудовані бібліотеки.
- Модуль потоку полегшує роботу з великими файлами.
- Єдиний синтаксис для клієнтської та серверної частини веб-сайту.

Це покращує багаторазовість коду і полегшує роботу розробника.

- Кешування модулів – модулі завантажуються та ініціалізуються, коли вони викликаються в перший раз, після чого вони постійно доступні.

Недоліки:

- Node.js вимагає більше коду.
- Не підтримує реляційні бази даних.
- Для кодування на Node.js ви повинні розуміти функції зворотного виклику та блокування.

- Не всі веб-хостинг-сервіси підтримують Node.js.
- Процес розгортання важче, ніж з PHP.
- Необхідно кожного разу використовувати зворотний виклик із великою кількістю вбудованих зворотних викликів.

PHP – скриптова мова загального призначення, інтенсивна застосовується для розробки веб-додатків [12]. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів [13]. Обробляється інтерпретатором, який підтримує Zend Engine, що встановлений на веб-серверах, таких як Apache та Nginx [14].

Переваги PHP:

- Висока гнучкість і популярність мови.
- Швидкість: відносно швидко, оскільки він використовує великий системний ресурс.
- Простіше налаштувати середовище розробки для PHP.
- Усі сервіси інтернет-хостингу підтримують PHP.
- Підтримує багато фреймворків та CMS.
- Офіційні рекомендації організації коду.
- PHP має більшу бібліотеку інструкцій і часто заданих питань.
- Спільнота розробників PHP – це величезна та різноманітна група людей, вони постійно переглядають та оновлюють бібліотеки, рамки та інші компоненти PHP.
- Може бути запущений на багатьох платформах, включаючи Windows, Linux і Mac, користувачам легко знайти постачальників послуг хостингу.

Таблиця 2.2

Порівняльна характеристика Node.js та PHP

Функція	node.js	php
Легкість налаштування (0.5)	Так, але з деякими труднощами (0.4)	Так (0.5)
Безпека (1)	Високий рівень (1)	Високий рівень, але з деякими слабкостями (0.8)
Підтримка баз даних (1)	Не підтримує реляційні бази даних (0)	Підтримує реляційні бази даних (1)
Гнучкість (0.7)	Середній рівень (0.5)	Високий рівень (0.7)
Процес розгортання (0.7)	Має деякі складнощі (0.5)	Легкий процес (0.7)
Популярність (0.5)	Середній рівень (0.4)	Дуже високий рівень (0.5)
Підтримка великих проєктів (0.5)	Так (0.5)	Ні (0.1)
Результати:	3.3	4.3

Недоліки:

- Оскільки він відкритий, то всі користувачі можуть бачити вихідний код, якщо в вихідному коді є помилки, то його можуть використовувати люди, щоб вивчити слабкість PHP.
- Не підходить для великих програм: важко підтримувати, оскільки він не дуже модульний.

- Неявне перетворення може здивувати програмістів та призвести до несподіваних помилок.

Навіть з великою кількістю переваг використання Node.js не підходить для цього проєкту, оскільки він не підтримує реляційні бази даних. Також враховувалось те, що процес розгортання проєктів на PHP набагато простіший, ніж Node.JS. А висока гнучкість та підтримка спільноти допоможуть досягнути високоякісного застосування.

Отже PHP обрано в якості мови програмування серверної частини за свою гнучкість, простоту використання і легкість в масштабуванні проєкту за необхідністю.

2.3 Вибір структури БД

Класичною методикою проєктування баз даних є створення окремої таблиці для кожної описуваної моделі даних суті, потім – в процесі нормалізації – виділення окремих таблиць для зберігання атрибутів сутності. Такий підхід добре працює для БД з відносно невеликою кількістю описуваних об'єктів і при нескладних та статичних зв'язках між ними. Однак будь-яка зміна структури збережених даних призводить до внесення змін в структуру таблиць, ці дані відображають. Нескладна на етапі розробки, ця операція стає вкрай проблематичною при великих обсягах. У зв'язку з цим потрібно вирішити проблему масштабованості.

Основні тези для вирішення даної задачі:

- Перш за все необхідно сформулювати основні принципи, на яких будуватиметься проєктована БД.
- Кожна сутність, інформація про яку зберігається в БД, – це об'єкт.
- Кожен об'єкт є унікальним у межах БД і має унікальний ідентифікатор.
- Об'єкт має властивості, які описують атрибути сутності.

- Об'єкти можуть бути пов'язані між собою довільним чином. Зв'язок характеризується пов'язаними об'єктами і типом зв'язку.
- Об'єкт може бути сховищем. У цьому випадку допускається зберігання в ньому інших об'єктів.

Така БД не прив'язана ні до якої бізнес-моделі і дозволяє реалізувати «над собою» практично будь-яку бізнес-логіку. Логіка виділяється в окремий програмний, де за запитом клієнта створюються об'єкти, які завантажили інформацію про себе з БД і реалізують «поведінку» об'єктів реального світу. У той же час, в силу одноманітності моделі зберігання, ці об'єкти досить легко створюються на основі базових класів, інкапсулюючих функціональність по завантаженню та збереженню властивостей і зв'язків в БД.

Структура такої бази даних називається «метамодель» і схематично зображена на рисунку 2.1 [15].

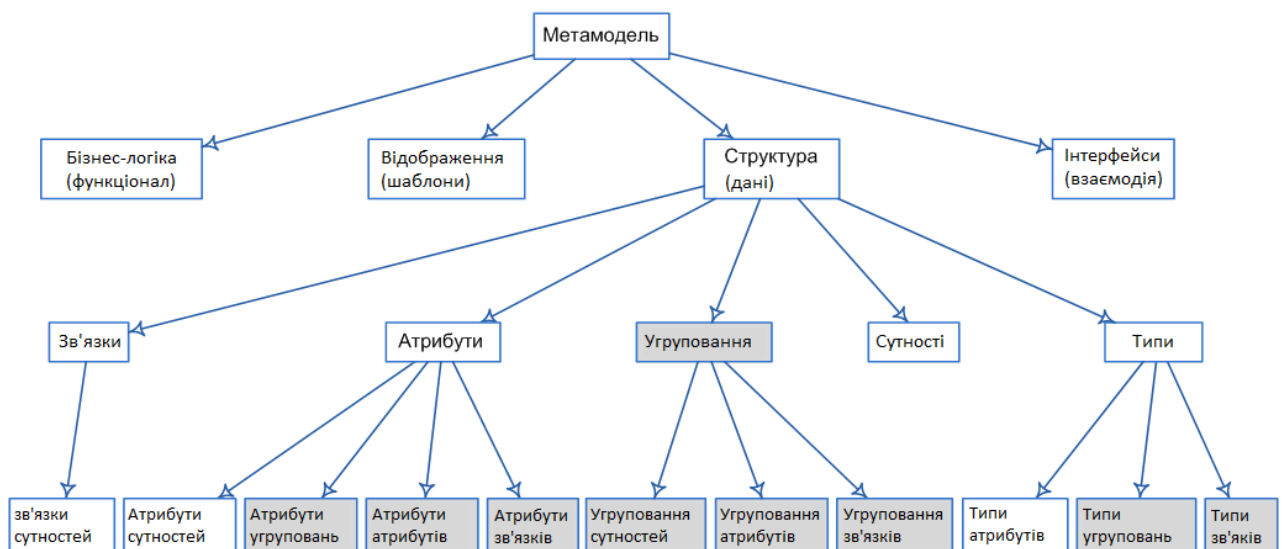


Рисунок 2.1 — Схема метамоделі БД

Тож для спрощеного масштабування системи і однотипності написання запитів до бази даних було обрано схему бази даних за типом мета моделі (рис. 2.1).

2.4 Архітектурний патерн

Архітектурні моделі є визнаними рішеннями, розробленими протягом багатьох років і вважаються галузевими стандартами. Це єдиний спосіб підтримати проєкт чистим, розширюваним і перевірюваним.

Існує два типи архітектурних зразків, які можна застосувати до поточного проєкту: Model-View-Controller (MVC) та Presenter-Model-View (MVP).

MVC підтримує швидкий та паралельний розвиток. Отже, проєкт може швидко розроблятися [16]. MVC також підтримує асинхронну техніку, яка допомагає розробляти додатки, які завантажуються дуже швидко. Шаблон MVC повертає дані, не застосовуючи жодного форматування, тому ті самі компоненти можуть використовуватися і викликатися для використання з будь-яким інтерфейсом [17]. Іншою важливою перевагою MVC є те, що вона має можливість створювати декілька переглядів для моделі. Дублювання коду дуже обмежене, оскільки дані та бізнес-логіка відділені від дисплея.

Програма, розроблена за допомогою MVC, може бути втричі швидшою, ніж додаток, розроблений з використанням інших моделей розробки.

Звичайно, MVC не є ідеальним, і має такі недоліки, і той факт, що для цього потрібні знання розробника щодо кількох технологій. Також для використання MVC нам потрібне гарне розуміння потоку керування між представленням, суб'єктом господарювання та контролером.

Але це залишається найкращим способом для реалізації серверної частини програми.

MVP - це архітектурна модель, яка використовується переважно для побудови користувацьких інтерфейсів. Вона прояснює структуру UI коду користувацького інтерфейсу і може полегшити його підтримку. Програми, створені за допомогою моделі MVP, добре підходять для тестування пристроїв. Дуже важливо, щоб цей шаблон допоміг відокремити логіку користувацького інтерфейсу від коду побудови інтерфейсу користувача.

Отже, для того, щоб програма стала розширюваною, обслуговуваною та перевіряємою, надзвичайно важливо створити глибоке розмежування задач, і це, мабуть, є найбільшою перевагою, яку можна отримати при застосуванні MVP.

При використанні MVP, треба пам'ятати, що ця схема є складною і може бути непотрібною для простих відображень.

В результаті огляду вирішено, завдяки всім перевагам цього патерну, використовувати MVC як головну архітектурну схему для розробки системи.

2.5 Висновки до розділу 2

В результаті огляду середовищ розробки та веб-технологій для досягнення поставлених цілей було вирішено спроектувати веб-сервіс та створити централізовану систему, її масштабовану базу даних та розробити компонент для CMS Joomla. Базу даних було спроектовано використовуючи підхід метамоделі, що дозволить і надалі розвивати проєкт. Сам централізований сервіс опрацювати за MVC патерном для розміщення інтерфейсної частини та бізнес логіки. Для цього сервіс розбито на окремі модулі, такі як «Створення опитувань», або «Мій профіль», кожен такий модуль не впливає на інші і може існувати окремо. Такий підхід дозволить надалі з легкістю підтримувати проєкт у команді. При розробці веб-сервісу в ході реалізації використати такі прогресивні технології як: PHP (5.4.x), MySQL, JavaScript, HTML (5), CSS, jQuery (2.2.4), Ajax, Bootstrap (3).

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Схема взаємодій користувачів та адміністраторів вебсистеми

Питання розробки інтуїтивно зрозумілого веб-сервісу для недосвідченого користувача актуально. Розглянемо створення веб-сервісу голосувань, який застосовується для збільшення числа відвідувачів, зворотного зв'язку з користувачами та SEO-просування сторонніх доменів.

Для цього необхідно розробити компонент для CMS Joomla, який забезпечить обмін даними і виступить сполучною ланкою між центральною системою і стороннім доменом.

Схему взаємодій користувачів та адміністраторів системою можна спостерігати на рисунку 3.1.

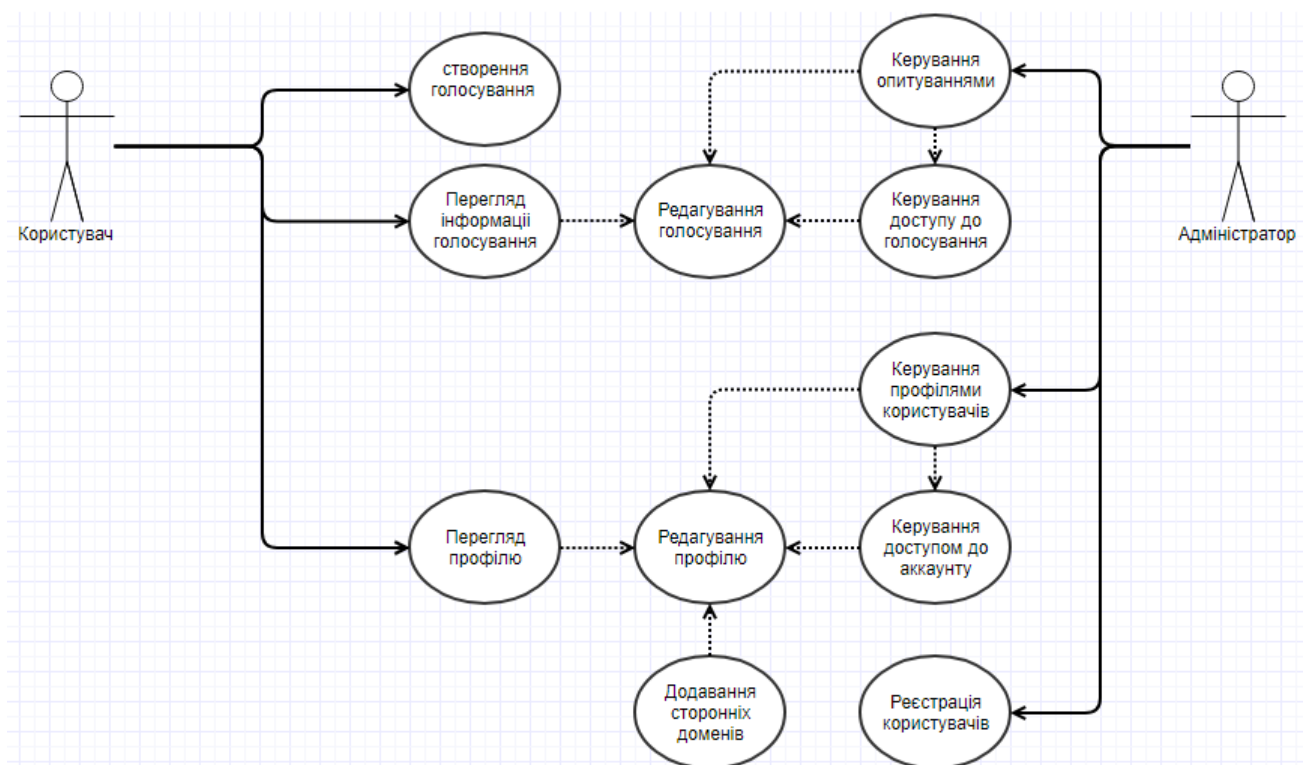


Рисунок 3.1 – UML діаграма варіантів використання

Веб-сервіс повинен надавати користувачу можливість створення голосування з широкими налаштуваннями, а саме мати можливість вказати:

- домени, на яких голосування буде розміщено;
- соціальні мережі, за допомогою яких користувачі зможуть голосувати;
- тип голосування, в якому вказується, який тип відповідей приймати (один варіант відповіді або множинний вибір);
- тип голосування, в якому вказується, якого типу варіанти відповідей будуть формуватися (текст, зображення, відео);
- мови, варіанти одного голосування на різних мовах;
- дати початку і кінця голосування, проміжок часу в якому людям можна буде віддавати свої голоси варіантів до створюваного голосування;
- основну інформацію для голосування (такі поля як текст питання, розширений опис питання, текст для публічних повідомлень, розміщених в соціальних мережах, варіанти відповіді на всіх обраних раніше мовах).

Також користувач повинен мати можливість редагування створених голосувань, перегляду інформації, перегляду результатів створених голосувань, перегляду свого профілю, редагування свого профілю, додавання сторонніх доменів до свого профілю в сервісі.

Для адміністратора передбачити можливості: блокування голосувань, блокування користувачів, створення нових користувачів, редагування профілів користувачів.

Сервіс повинен формувати інформацію про голосуваннях для відправки на сторонній домен

3.2 Розробка бази даних

Для спрощеного масштабування системи і однотипності написання запитів до бази даних було обрано схему бази даних за типом метамоделі, було спроектовано базу даних за структурою що показано на рисунках 3.2 та 3.3. [18]

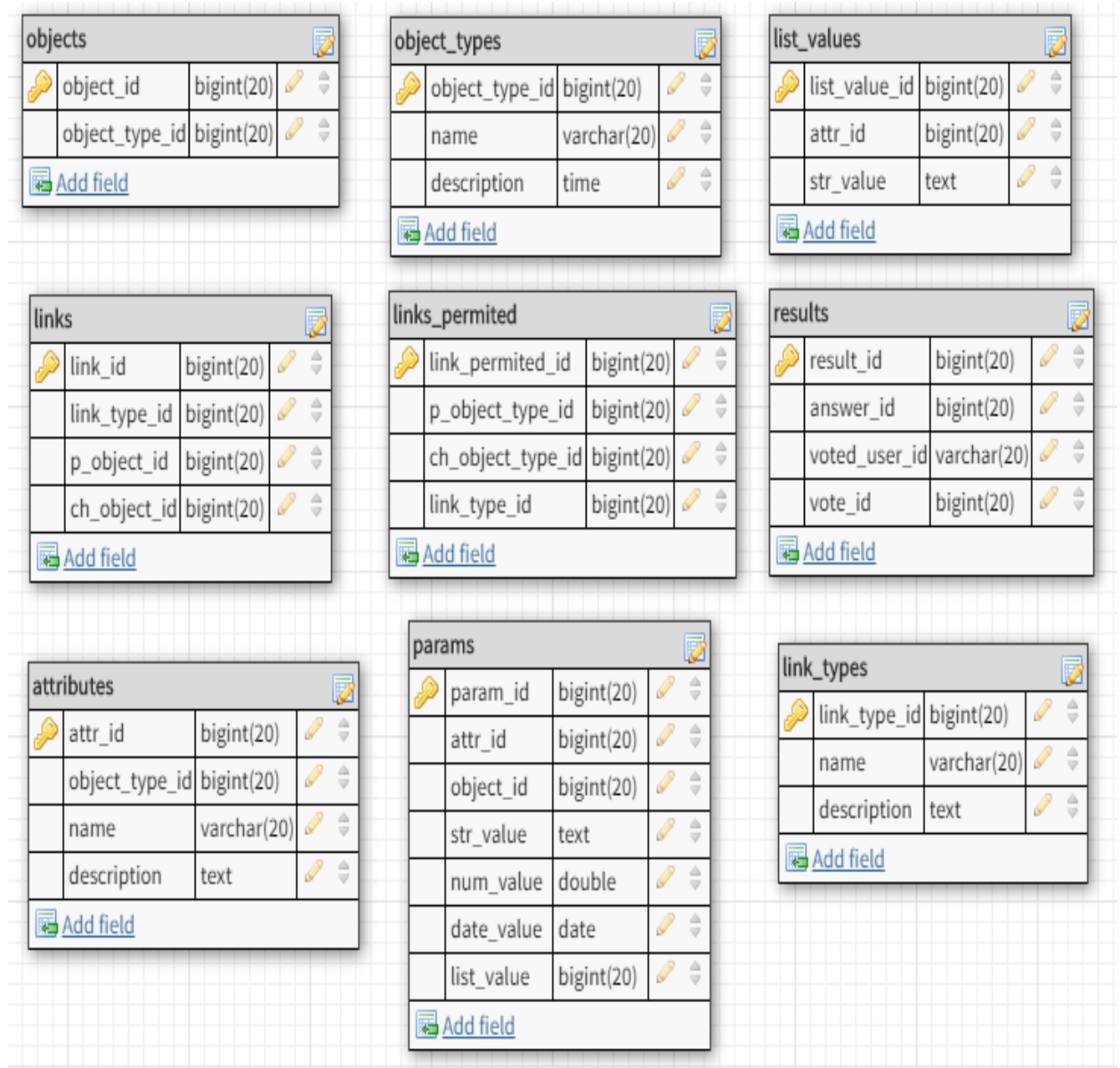


Рисунок 3.2 – Структура бази даних



Рисунок 3.3 – Структура бази даних

3.3 Розробка веб-сервісу голосування

Мета створення централізованої веб-системи управління голосуваннями — автоматизація процесу створення, інтеграції та аналізу опитувань за рахунок об'єднання в централізовану систему наступних модулів:

- модуль управління користувачами системи (рис. 3.5);
- модуль перегляду та редагування інформації про користувача (рис.3.6);

- модуль управління профілем (рис. 3.7);
- модуль реєстрації користувачів;
- модуль управління голосуваннями (рис. 3.8);
- модуль створення голосування (рис. 3.9);
- модуль для відображення результатів голосування (рис. 3.10);

Об'єднання модулів [19] централізованої системи управління голосуваннями показано на рис. 3.4. На одній web-сторінці надано швидкий доступ до основних модулів системи в доступній для користувача формі. Інтуїтивно зрозумілі іконки і підписи під ними дозволяють користувачам будь-якого рівня користуватися даною системою.

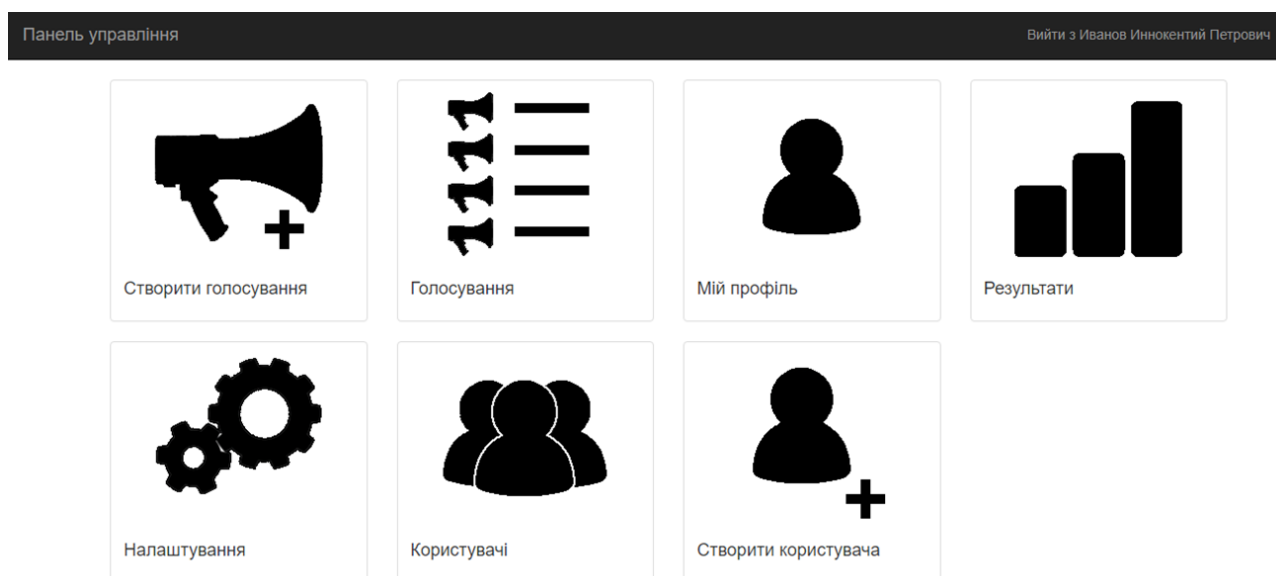


Рисунок 3.4 — Об'єднані модулі централізованої системи управління

Інтерфейс модуля управління акаунтами користувачів системи представлено на рис. 3.4. Всі необхідні функції, які можуть застосовуватися до користувачів системи (Переглянути інформацію про користувача, додавання нового користувача, видалення користувача, блокування/розблокування користувача). Інтуїтивно зрозумілий інтерфейс даної сторінки дозволяє адмініструвати дану систему користувачам будь-якого рівня.

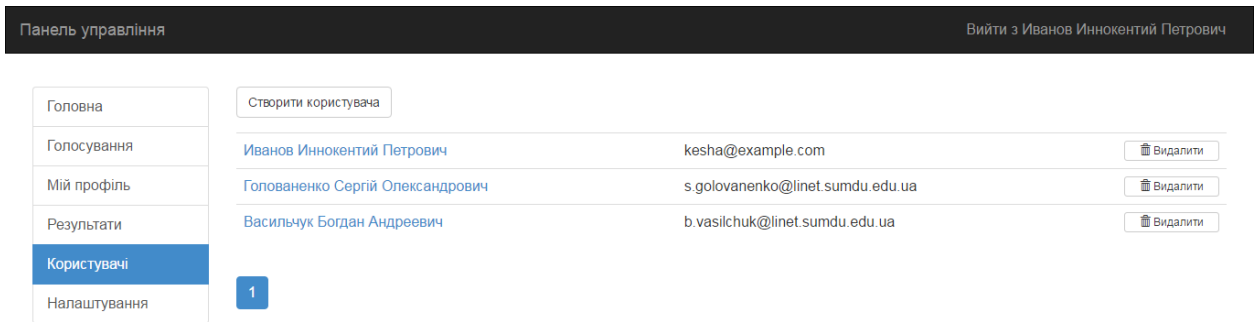


Рисунок 3.5 — Модуль управління користувачами системи

Модуль управління користувачами описано в п'яти файлах, а саме:

- usersView.php – візуальне частина сторінки з усіма користувачами;
- usersController.php – забезпечує взаємодію між базою даних і візуальною частиною;
- userModel.php – реалізовані функції звернення до бази даних;
- usersDelController.php – обробляє запит адміністратора на видалення користувача із системи;
- usersDelModel.php – реалізовані функції звернення до бази даних необхідні для видалення користувача із системи.

Модуль перегляду та редагування інформації про користувача надає всю інформацію про користувача (рис. 3.6).

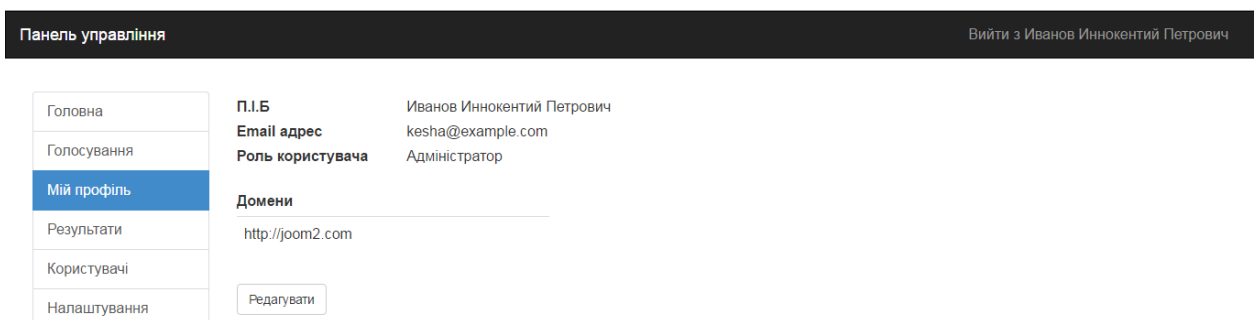


Рисунок 3.6 — Модуль перегляду інформації про користувача

Вищевказаний модуль дозволяє власникові профілю змінювати свої дані (рис. 3.7).

Панель управління

Вийти з Іванов Іннокентій Петрович

Головна

Голосування

Мій профіль

Результати

Користувачі

Налаштування

П.І.Б.

Іванов Іннокентій Петрович

Email адрес

kesha@example.com

Роль користувача

Адміністратор

Пароль

Пароль

Повторіть пароль

Зберегти

Скасувати

Доменні імена

http://joom2.com

Видалити

Додати домен

Рисунок 3.7 — Редагування даних користувача

Модуль управління профілем описаний в десяти файлах:

- profileView.php – візуальна частина сторінки, відображення поточних даних користувача: ім'я, активний адресу електронної пошти, роль і прив'язані до профілю сторонні домени;
- profileController.php – забезпечує взаємодію між базою даних і візуальною частиною,
- profileModel.php – реалізовані функції звернення до бази даних для отримання необхідної інформації;
- profileEditView.php – описана візуальна частина сторінки, відображення поточних даних користувача з можливістю змінити їх і зберегти;
- profileEditController.php – забезпечує взаємодію між базою даних і візуальною частиною;
- profileEditModel.php – реалізовані функції звернення до бази даних для внесення змін в профіль;
- domainAddController.php – реалізована обробка запиту користувача на прив'язку нового стороннього домену до профілю;
- domainAddModel.php – реалізовані функції звернення до бази даних, необхідні для прив'язки нового стороннього домену до профілю користувача;

- domainDelController.php – реалізована обробка запиту користувача на видалення інформації про прив'язку стороннього домену до профілю користувача;

- domainDelModel.php – реалізовані функції звернення до бази даних, необхідні для видалення інформації про прив'язку стороннього домену до профілю користувача.

Модуль реєстрації користувачів описаний в трьох файлах:

- usersAddView.php – описана візуальна частина сторінки створення нового користувача, з полями для введення інформації

- usersAddController.php – здійснюється взаємодія між базою даних і візуальною частиною;

- usersAddModel.php – реалізовані функції звернення до бази даних необхідні для додавання нового користувача в базу.

Модуль управління голосуваннями (рис. 3.87) – даний модуль надає мінімальну інформацію про голосування. Дана сторінка надає можливість користувачу додати нове голосування, ознайомитись з результатами існуючих голосувань, відредагувати або ж видалити їх.

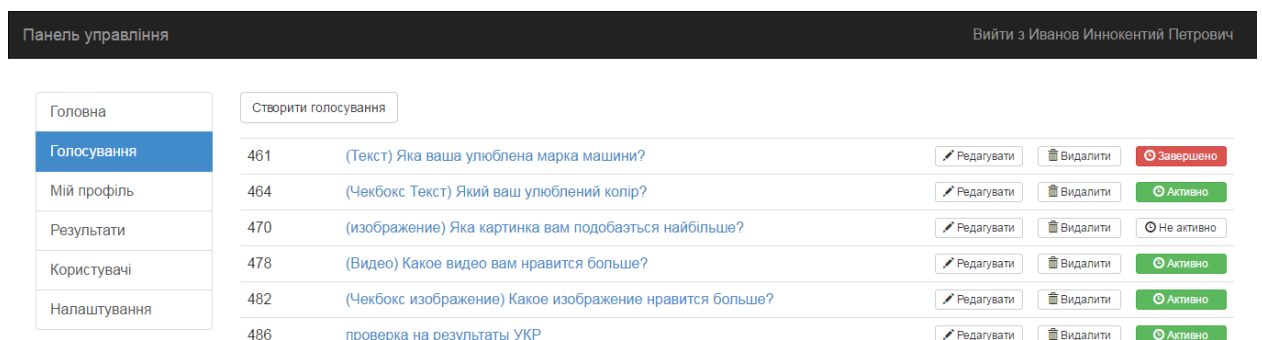


Рисунок 3.8 — Модуль управління голосуваннями

Модуль управління голосуваннями описаний в восьми файлах:

- golosView.php – описана візуальна частина сторінки, відображення всіх голосувань і можливих з ними операцій;

- `golosController.php` – здійснюється взаємодія між базою даних і візуальною частиною;
- `golosModel.php` – реалізовані функції звернення до бази даних;
- `golosEditView.php` – візуальна частина цієї сторінки дозволяє користувачеві редагувати створені раніше їм голосування;
- `golosEditController.php` – здійснюється зв'язка бази даних візуальної частини редагування голосування і звернень до бази даних;
- `golosEditModel.php` – описані функції звернення до бази даних, необхідні для редагування голосувань;
- `golosDelController.php` – обробляється запит користувача на видалення створеного раніше їм голосування;
- `golosDelModel.php` – реалізовані необхідні функції звернення до бази даних для видалення голосувань.

Модуль створення голосувань (рис. 3.9) складається з 7-ми обов'язкових для заповнення користувачем кроків, в яких збирається інформація про голосування і за результатами яких буде сформовано саме голосування. У кроках вказується така інформація як:

- 1) Домени, на яких буде надалі розміщено голосування;
- 2) Соціальні мережі, в яких буде дозволено розповсюджувати інформацію про даний голосуванні;
- 3) Тип голосування, в якому вказується, скільки варіантів відповіді може бути вибрано одноразово (один або кілька);
- 4) Тип голосування, в якому вказується, якого типу варіанти відповідей будуть формуватися (текст, зображення, відео);
- 5) Мови, варіанти одного голосування на різних мовах;
- 6) Дати початку і кінця голосування, проміжок часу в якому людям можна буде віддавати свої голоси;
- 7) Основна інформація голосування. Сюди входять такі поля як текст питання, розширений опис питання, текст для публічних повідомлень,

розміщених в соціальних мережах, варіанти відповіді на всіх обраних ранніх мовами.

Панель управління

Вийти з Іванова Євгенія Петровича

1. Крок 2. Крок 3. Крок 4. Крок 5. Крок 6. Крок 7. Крок

Назад Створити

Українська Русский English

Створіть голосування

Питання

Введіть текст питання

Короткий опис питання

Текст публікації (частина 1)

частина 1

Текст публікації (частина 2)

частина 2

1. Варіант

Текст (Український)

Рисунок 3.9 — Модуль створення голосувань

Модуль створення голосувань описаний в трьох файлах:

- `golosAddView.php`, де описана візуальна частина сторінки і взаємодія елементів на ній за допомогою JavaScript і jQuery;
 - `golosAddController.php`, де здійснюється взаємодія між базою даних і візуальною частиною (Додаток А);
 - `golosAddModel.php`, де реалізовані функції звернення до бази даних.
- Він дозволяє створювати нові голосування (Додаток А).

Модуль відображення інформації про голосування (рис. 3.10) — даний модуль надає всю інформацію про обраний голосування і дозволяє швидко перейти до управління даними голосуванням.

Панель управління

Вийти з Іванов Іннокентій Петрович

Головна

Голосування

Мій профіль

Результати

Користувачі

Налаштування

Українська

Русский

Питання	(Текст) Яка ваша улюблена марка машини?
Опис	Марка
Приклад публікацій	Моя улюблена марка машини А твоя?
Тип голосування	text
Тип вибору відповіді	radio
Дозволені соціальні мережі	facebook,vkontakte,twitter
Дата початку	01-01-1970 02:00
Дата закінчення	01-01-1970 02:00

Мерседес

2 / 67%

Лада

0 / 0%

Ауді

0 / 0%

Очень длинное название марки машины которое может куда-то не влезть

1 / 33%

Рисунок 3.10 — Модуль відображення інформації про голосування

Модуль відображення інформації про голосування описаний в шести файлах:

- resultsView.php – реалізує візуальну частину сторінки з голосуваннями користувача з короткою інформацією про результати кожного голосування;
- resultsController.php – здійснюється взаємодія між базою даних і візуальної частиною;
- resultsModel.php – реалізовані функції звернення до бази даних;
- resultPageView.php – описана візуальна частина сторінки на якій знаходиться повна інформація про голосування і його результати;
- resultPageController.php – здійснюється взаємодія між базою даних і візуальної частиною;
- resultPageModel.php – реалізовані функції звернення до бази даних необхідні для виведення результатів.

3.4 Розробка компонента для CMS Joomla

Для розміщення створених голосувань на сторонніх доменах використано компонент для системи управління контентом Joomla. Цей компонент дозволяє вибрати голосування, необхідні для розміщення, і мову відображення голосування на сторінці (рис. 3.11).

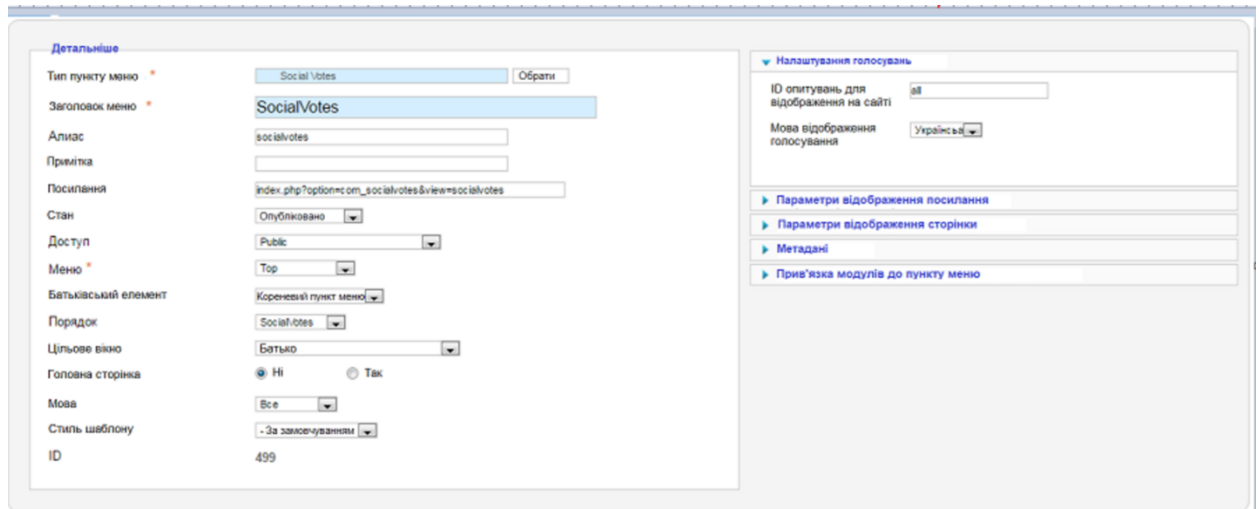


Рисунок 3.11 — Налаштування компонента в Joomla

Відображення компонента на сторінці стороннього ресурсу показано на рис. 3.12.

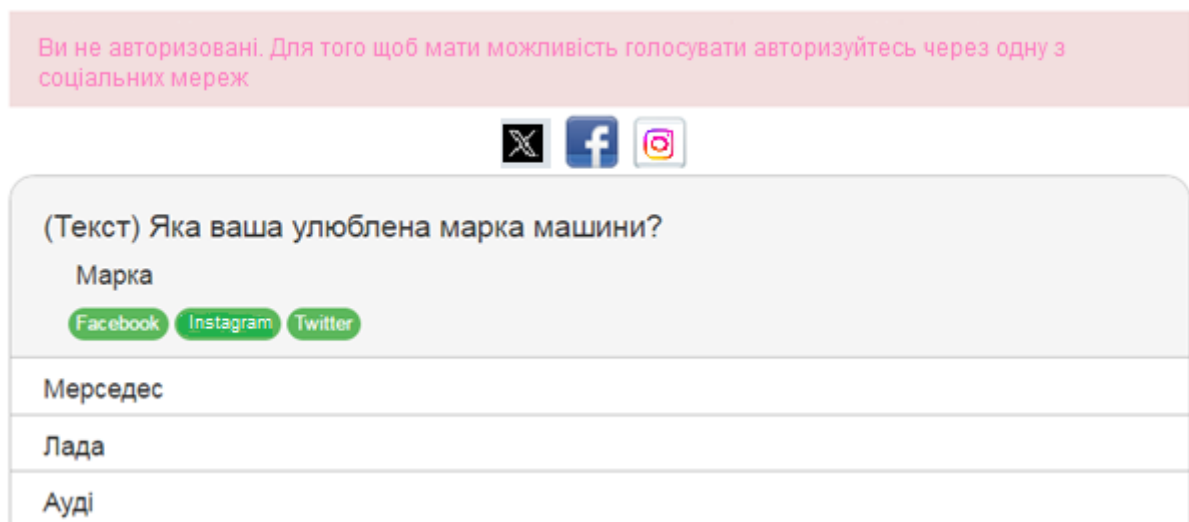


Рисунок 3.12 — Відображення компонента на сторінці

Компонент для Joomla описаний в чотирьох файлах (виключаючи необхідні для інтеграції компонента в CMS Joomla) [20]:

- transDomainScriptView.php, через який компонент отримує відповідь на свій запит у вигляді голосування та інформації про нього (Додаток Б);
- transDomainScriptController.php, де обробляється запит який прийшов від компонента [21] (Додаток В);
- transDomainScriptModel.php, де реалізовані функції звернення до бази даних для відправки їх на компонент (Додаток В);
- default.php, де реалізований висновок голосувань, можливість проголосувати, обробка та відправка даних на центральну систему у вигляді відповідей від користувачів.

3.5 Результати роботи

Система передбачає проведення реєстрації користувачів за допомогою адміністраторів сервісу. Якщо користувач вже має аккаунт у системі він проходить авторизацію за логіном та паролем. Редагувати режими доступу та інформацію користувача можна у модулі «Мій профіль». Перед початком повномірної роботи сервісом необхідно до свого профілю прив'язати один або декілька сторонніх доменів, де планується розміщення опитувань. Для цього користувач повинен мати доступ до FTP сайту, який він хоче прив'язати до свого аккаунту. Система запропонує розмістити унікальний файл у кореневому каталозі сайту, та за запитом користувача перевірить чи доступний він з даного сервісу. Якщо сервіс знайшов файл на сторонньому домені, то він прив'язується до аккаунту користувача і той в змозі створювати опитування для вказаного домену. Якщо система не знайшла файл, то домен прив'язано не буде і користувач буде не в змозі створювати опитування для цього домену. Наступним кроком можна створити голосування, у цьому допоможуть сім етапів формування опитування. Після створення опитування, перейшовши на сайт, де планується його розміщення, за допомогою компоненту, опитування буде відображено. Роботу сервісу можна побачити на рисунку 3.13.

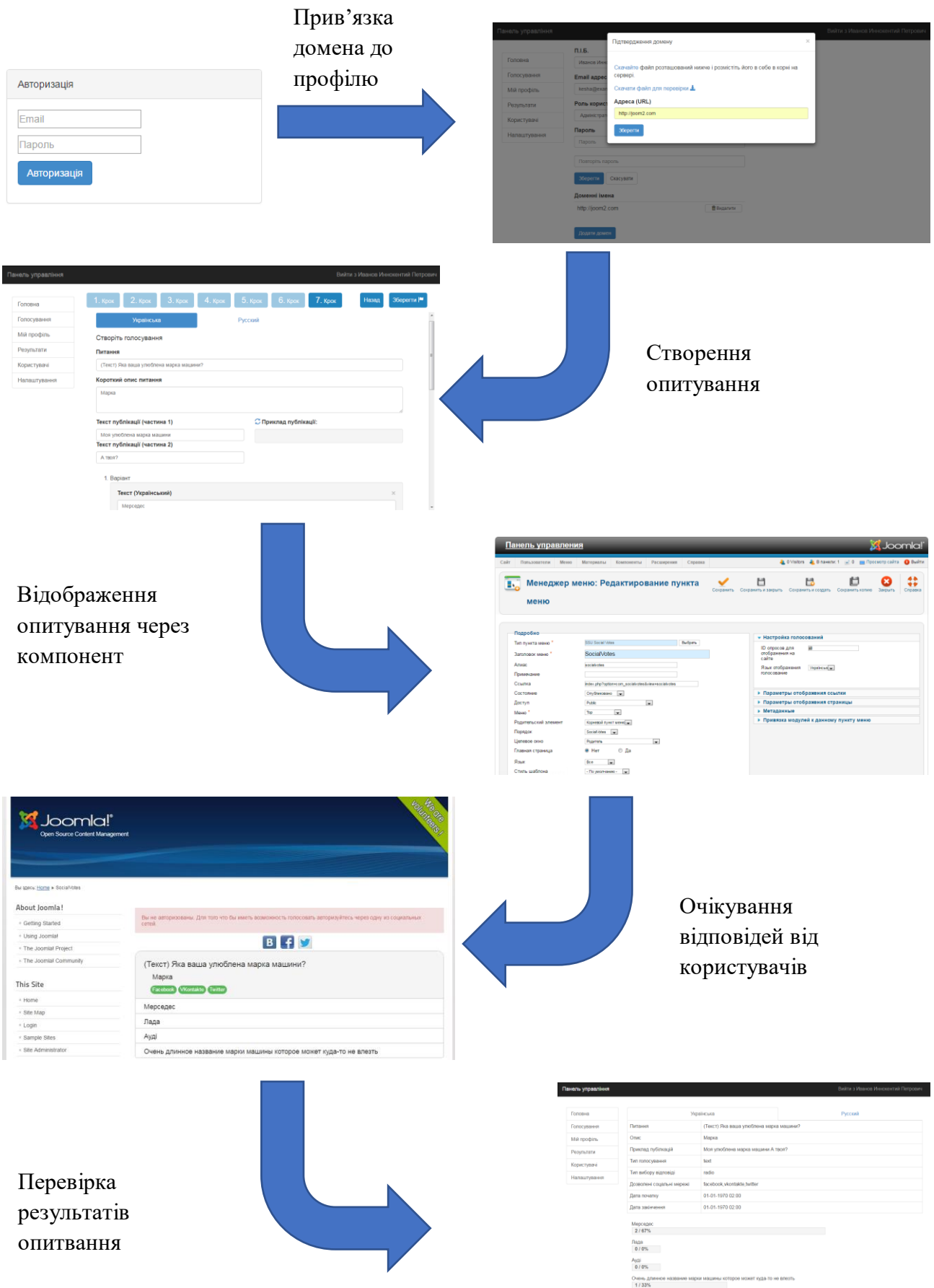


Рисунок 3.13 — Работа сервісу

При наявності одного або декількох сайтів, на яких необхідно розмістити голосування, всі процеси, пов'язані з його створенням, зміною, аналізом результатів, набагато зручніше виконувати за допомогою центральної системи. Створення окремо для кожного домена однакових голосувань викличе необхідність збирати дані з різних доменів і підводити підсумки вручну. Таким чином, рішенням проблеми є централізація цього процесу.

Створивши одне голосування і підв'язавши його до будь-якої кількості сторонніх доменів, можна відображати його на всіх сайтах. При цьому при необхідності внесення змін до опитування їх слід виконувати в централізованій системі. Зміни будуть застосовуватися до всіх відображеннях цього голосування, а результати будуть зібрані на одній сторінці для всіх зазначених мов.

3.6 Висновки до розділу 3

Розроблено інтуїтивно зрозумілий користувачу веб-сервіс для створення голосувань, який застосовуються для збільшення числа відвідувачів, зворотного зв'язку з користувачами та SEO-просування сторонніх доменів. Створено систему, яка передбачає проведення реєстрації користувачів за допомогою адміністраторів сервісу. Якщо користувач вже має аккаунт у системі він проходить авторизацію за логіном та паролем.

Розроблено компоненти для CMS Joomla, які забезпечують обмін даними між центральною системою і сторонніми доменами.

Веб-сервіс надає користувачу можливість створення голосування з широкими налаштуваннями, а саме має можливість вказати домени, на яких голосування буде розміщено; соціальні мережі, за допомогою яких користувачі зможуть голосувати; тип голосування, в якому вказується, який тип відповідей приймати (один варіант відповіді або множинний вибір); тип голосування, в якому вказується, якого типу варіанти відповідей будуть формуватися (текст, зображення, відео); мови, варіанти одного голосування на різних мовах; дати проміжок часу в якому людям можна буде віддавати свої голоси варіантів до створюваного голосування; основну інформацію для голосування.

Також користувач має можливість редагувати створені голосування, перегляд інформації, перегляд результатів створених голосувань, перегляд свого профілю, редагування свого профілю, додавання сторонніх доменів до свого профілю в сервісі.

Для адміністратора передбачено можливості: блокування голосувань, блокування користувачів, створення нових користувачів, редагування профілів користувачів. Сервіс формує інформацію про голосуваннях для відправки на сторонній домен.

ВИСНОВКИ

В результаті роботи розроблено інтуїтивно зрозуміла недосвідченому користувачу система для централізованого адміністрування опитувань і компонент для CMS Joomla за допомогою таких технологій як PHP, HTML, JavaScript і CSS. Розроблений веб-сервіс має можливість створення голосування з широкими настройками, редагування, видалення або їх блокування, перегляду інформації і результатів голосувань, відображення і редагування профілів користувачів системи, реєстрації нових користувачів, взаємодії з компонентом для CMS Joomla, відправка і отримання даних про голосування.

Сервіс працює зі сторонніми ресурсами, на яких встановлена CMS Joomla завдяки компоненту, який забезпечує обмін даними між центральною системою і сторонніми ресурсами. Отримує інформацію, сформовану системою, формує з отриманої інформації голосування. Авторизує користувачів через соціальні мережі, відправляє дані системі для фіксації голосів, формує публікації і пропонує користувачам поділитися ними в соціальних мережах.

Даний сервіс призначено для підвищення кількості відвідувань, так як систему орієнтовано на соціальні мережі, де побуває велика кількість користувачів, для SEO просування сайтів завдяки підвищенню кількості відвідувань, та для отримання зворотного зв'язку з користувачами різних ресурсів.

СПИСОК ЛІТЕРАТУРИ

1. Countries and territories with the highest Instagram audience reach as of April 2024. [Statista]. 2024. URL: <https://www.statista.com/statistics/325567/instagram-penetration-regions/>.
2. Stacy Jo Dixon. Number of Facebook users worldwide from 2017 to 2027 [Statista]. 2024. URL: <https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/>.
3. Most popular social networks worldwide as of April 2024, by number of monthly active users. [Statista]. 2024. URL: <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>.
4. Joomla for Developers. URL: <https://developer.joomla.org>.
5. Freeman El. & Freeman Er. Head First HTML and CSS: OREILLY AND ASSOCIATE. FREE Delivery, ISBN-10: 0596159900. 2012. 768p.
6. Freeman Er. & Freeman El. Head First HTML5 Programming: Building Web Apps with JavaScript. Head First O'Reilly. 2011. 608p.
7. Freeman T., Robson E. Head First JavaScript Programming. Head First O'Reilly. 2014. 661p.
8. Robbins Jennifer. HTML5 Pocket Reference: Quick, Comprehensive, Indispensable, Head First O'Reilly. 5th ed. 2013. 182p.
9. Nixon Robin. Learning Php, MySQL & JavaScript: A Step-By-Step Guide to Creating Dynamic Websites. Head First O'Reilly, 2021. 823p.
10. Benedetti Ryan, & Cranley Ronan. Head First jQuery: A Brain-Friendly Guide. Head First O'Reilly., 2011. 536 p.
11. Bootstrap. відкритий набір інструментів для розробки HTML, CSS та JS [Bootstrap]. 2024. URL: <https://getbootstrap.com>.
12. Flávio Gomes Da Silva Lisboa. PHP Web Development with Laminas: Build a fully secured and functional e-commerce application with PHP using the Laminas framework, Packt. 2022. 394p.

13. Materials from the official website of the programming language PHP [The PHP group] . 2024. URL: <http://www.php.net/>
14. Duckett Jon. PHP & MySQL: Server-side Web Development, Wiley. 2022. 672 p.
15. Beighley Lynn. Head First SQL. Head First O'Reilly. 2007. 607 p.
16. Townsend RJ. Foundation Drupal 7: Learn How to Use the Drupal Framework to Quickly Build Feature-rich Websites. Friends of. 2010. 309 p.
17. Stef Van Looveren & Reva Gomes. Aprendiendo Drupal como framework: Guía práctica con código completo incluido. 2023. 235 p.
18. Service for building diagrams based on .sql. URL: <http://dbdesigner.net/>
19. Murach Joel, Harris Ray. Murach's PHP and MySQL. (4th Edition), #MurachBooks. 2022. 848p.
20. Eric A. Mann, PHP Cookbook: Modern Code Solutions for Professional Developers. Head First O'Reilly. 2023. 431p.
21. Ullma Larry. PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide. Pearson, 2017. 704p.

ДОДАТКИ

ДОДАТОК А. Код файлу golosAddController.php

```

<?php
class GolosAddController extends Controller {
    private $model;
    private $view;

    function __construct($linkDb) {
        $this->model = new GolosAddModel($linkDb);
        $this->view = new View();
        $this->request();
    }
    function request() {
        if (isset($_POST['domain']) &&
        isset($_POST['lang']) && isset($_POST['golosType'][0]) &&
        isset($_POST['social']) &&
        isset($_POST['golosTypeMulti'][0]) &&
        isset($_POST['dateStart']) && isset($_POST['dateFinish']))
        {
            $vote_id = $this->model->createVote();
            $languages = $_POST['lang'];
            $owner = $_SESSION['user_id'];
            $domains = $_POST['domain'];
            $socials = $_POST['social'];
            $golosTypeMulti =
            $_POST['golosTypeMulti'][0];
            $golosType = $_POST['golosType'][0];
            $dateStart = strtotime($_POST['dateStart']);
            $dateFinish =
            strtotime($_POST['dateFinish']);
            $isInserted = $this->model->
            >insertCommonGolosData($vote_id,$domains,$socials,$golosTypeM
            ulti,$golosType,$dateStart,$dateFinish,$owner);

            if ($golosType == 3) // 3 = текст
            {
                $answerTextUkr = $_POST['answerTextUkr'];
                $answerTextRus = $_POST['answerTextRus'];
                $answerTextEng = $_POST['answerTextEng'];
                $this->model->
                >insertDataVoteTextAnswers($vote_id,$answerTextUkr,$answerTex
                tRus,$answerTextEng, $languages);
                foreach($languages as $lang)
                {
                    if ($lang == 'ukr')
                    {
                        if
                        (isset($_POST['questionUkr'],$_POST['questionContinueUkr'],$_
                        POST['socialPartUkr_1'],$_POST['socialPartUkr_2'],$_POST['ans
                        werTextUkr']))

```

```

        {
            $questionUkr =
$_POST['questionUkr'];
            $questionContinueUkr =
$_POST['questionContinueUkr'];
            $socialPartUkr_1 =
$_POST['socialPartUkr_1'];
            $socialPartUkr_2 =
$_POST['socialPartUkr_2'];
            $this->model->
>insertDataVoteUkr($vote_id, $questionUkr,
$questionContinueUkr,$answerTextUkr,$socialPartUkr_1,$socialP
artUkr_2);
        }
    }
    if ($lang == 'rus')
    {
        if
(isset($_POST['questionRus'],$_POST['questionContinueRus'],$_
POST['socialPartRus_1'],$_POST['socialPartRus_2'],$_POST['ans
werTextRus']))
        {
            $questionRus =
$_POST['questionRus'];
            $questionContinueRus =
$_POST['questionContinueRus'];
            $socialPartRus_1 =
$_POST['socialPartRus_1'];
            $socialPartRus_2 =
$_POST['socialPartRus_2'];
            $this->model->
>insertDataVoteRus($vote_id, $questionRus,
$questionContinueRus,$answerTextRus,$socialPartRus_1,$socialP
artRus_2);
        }
    }
    if ($lang == 'eng')
    {
        if
(isset($_POST['questionEng'],$_POST['questionContinueEng'],$_
POST['socialPartEng_1'],$_POST['socialPartEng_2'],$_POST['ans
werTextEng']))
        {
            $questionEng =
$_POST['questionEng'];
            $questionContinueEng =
$_POST['questionContinueEng'];
            $socialPartEng_1 =
$_POST['socialPartEng_1'];
            $socialPartEng_2 =
$_POST['socialPartEng_2'];
            $this->model->
>insertDataVoteEng($vote_id, $questionEng,

```

```

$questionContinueEng,$answerTextEng,$socialPartEng_1,$socialP
artEng_2);
    }
    }
    }
    if ($golosType == 4) //4 = картинка
    {
        $answerImageNameUkr =
$_POST['answerImgTitleUkr'];
        $answerImageNameRus =
$_POST['answerImgTitleRus'];
        $answerImageNameEng =
$_POST['answerImgTitleEng'];
        $image_array = array();

        $colImages =
count($_FILES['answerImagePic']['name']);
        if ($colImages > 0)
        {
            $path_to_dir =
$_SERVER['DOCUMENT_ROOT'] . ADMIN_PATH .
'/images/golos_images/'. $vote_id;
            $dir = mkdir($path_to_dir);
            if ($dir)
            {
                for ($i = 0; $i < $colImages;
$i++)
                {
                    if
($FILES["answerImagePic"]["error"][$i] == 0)
                    {
                        if($FILES["answerImagePic"]["size"][$i] < 1024*2*1024)
                        {
                            if(is_uploaded_file($FILES["answerImagePic"]["tmp_name"][$i]
))
                            {
                                $file_type =
end(explode(".", $FILES["answerImagePic"]["name"][$i]));
                                $new_file_name =
uniqid() . '.' . $file_type;

                                move_uploaded_file($FILES["answerImagePic"]["tmp_name"][$i],
$_SERVER['DOCUMENT_ROOT'] . ADMIN_PATH .
'/images/golos_images/' . $vote_id . '/' . $new_file_name);
                                array_push($image_array,$new_file_name);
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
    $this->model-
>insertDataVoteImageAnswers($vote_id,$answerImageNameUkr,$ans
werImageNameRus,$answerImageNameEng,$image_array);
    foreach($languages as $lang)
    {
        if ($lang == 'ukr')
        {
            if
(isset($_POST['questionUkr'],$_POST['questionContinueUkr'],$_
POST['socialPartUkr_1'],$_POST['socialPartUkr_2']))
            {
                $questionUkr                =
$_POST['questionUkr'];
                $questionContinueUkr        =
$_POST['questionContinueUkr'];
                $socialPartUkr_1            =
$_POST['socialPartUkr_1'];
                $socialPartUkr_2            =
$_POST['socialPartUkr_2'];
                $this->model-
>insertDataVoteImageUkr($vote_id,                $questionUkr,
$questionContinueUkr,$socialPartUkr_1,$socialPartUkr_2);
            }
        }
        if ($lang == 'rus')
        {
            if
(isset($_POST['questionRus'],$_POST['questionContinueRus'],$_
POST['socialPartRus_1'],$_POST['socialPartRus_2']))
            {
                $questionRus                =
$_POST['questionRus'];
                $questionContinueRus        =
$_POST['questionContinueRus'];
                $socialPartRus_1            =
$_POST['socialPartRus_1'];
                $socialPartRus_2            =
$_POST['socialPartRus_2'];
                // $answerTextRus          =
$_POST['answerTextRus'];
                $this->model-
>insertDataVoteImageRus($vote_id,                $questionRus,
$questionContinueRus,$socialPartRus_1,$socialPartRus_2);
            }
        }
        if ($lang == 'eng')
        {
            if
(isset($_POST['questionEng'],$_POST['questionContinueEng'],$_
POST['socialPartEng_1'],$_POST['socialPartEng_2']))
            {

```

```

$questionEng = $_POST['questionEng'];
$questionContinueEng = $_POST['questionContinueEng'];
$socialPartEng_1 = $_POST['socialPartEng_1'];
$socialPartEng_2 = $_POST['socialPartEng_2'];
// $answerTextEng = $_POST['answerTextEng'];
$this->model->insertDataVoteImageEng($vote_id, $questionEng, $questionContinueEng, $socialPartEng_1, $socialPartEng_2);
    }
    }
    if ($golosType == 5) // 5 = видео
    {
        if (isset($_POST['answerVideoLink']))
        {
            $answerVideoLink = $_POST['answerVideoLink'];
            $answerVideoNameUkr = $_POST['answerVideoTitleUkr'];
            $answerVideoNameRus = $_POST['answerVideoTitleRus'];
            $answerVideoNameEng = $_POST['answerVideoTitleEng'];
            $this->model->insertDataVoteVideoAnswers($vote_id, $answerVideoNameUkr, $answerVideoNameRus, $answerVideoNameEng, $answerVideoLink);
            foreach($languages as $lang)
            {
                if ($lang == 'ukr')
                {
                    if (isset($_POST['questionUkr'], $_POST['questionContinueUkr'], $_POST['socialPartUkr_1'], $_POST['socialPartUkr_2']))
                    {
                        $questionUkr = $_POST['questionUkr'];
                        $questionContinueUkr = $_POST['questionContinueUkr'];
                        $socialPartUkr_1 = $_POST['socialPartUkr_1'];
                        $socialPartUkr_2 = $_POST['socialPartUkr_2'];
                        $this->model->insertDataVoteVideoUkr($vote_id, $questionUkr, $questionContinueUkr, $socialPartUkr_1, $socialPartUkr_2);
                    }
                }
            }
        }
    }
}

```



```

        if ($lang == 'rus')
        {
            if
(isset($_POST['questionRus'],$_POST['questionContinueRus'],$_
POST['socialPartRus_1'],$_POST['socialPartRus_2']))
            {
                $questionRus =
$_POST['questionRus'];
                $questionContinueRus =
$_POST['questionContinueRus'];
                $socialPartRus_1 =
$_POST['socialPartRus_1'];
                $socialPartRus_2 =
$_POST['socialPartRus_2'];
                // $answerTextRus =
$_POST['answerTextRus'];
                $this->model->
>insertDataVoteVideoRus($vote_id,$questionRus,
$questionContinueRus,$socialPartRus_1,$socialPartRus_2);
            }
        }
        if ($lang == 'eng')
        {
            if
(isset($_POST['questionEng'],$_POST['questionContinueEng'],$_
POST['socialPartEng_1'],$_POST['socialPartEng_2']))
            {
                $questionEng =
$_POST['questionEng'];
                $questionContinueEng =
$_POST['questionContinueEng'];
                $socialPartEng_1 =
$_POST['socialPartEng_1'];
                $socialPartEng_2 =
$_POST['socialPartEng_2'];
                // $answerTextEng =
$_POST['answerTextEng'];
                $this->model->
>insertDataVoteVideoEng($vote_id,$questionEng,
$questionContinueEng,$socialPartEng_1,$socialPartEng_2);
            }
        }
    }
}

function indexAction($arg = NULL) {
    $data = $this->model->
>getData($_SESSION['user_id']);
    $this->view->genView('golosAddView.php',
'templateAllView.php','Створення голосування','', $data);
}

```

ДОДАТОК Б. Код файлу golosAddModel.php

```

<?php
class GolosAddModel extends Model{
    private $linkDb;

    public function __construct($linkDb) {
        $this->linkDb = $linkDb;
    }

    public function getData($user_id) {
        $data = Array();
        $data['domains'] = $this->getDomains($user_id);
        return $data;
    }

    public function createVote()
    {
        $sql = "INSERT INTO `objects`(`object_type_id`)
VALUES (5)";
        if(!$this->linkDb->query($sql))
        {
            return false;
        }
        return mysqli_insert_id($this->linkDb);
    }

    public function insertCommonGolosData($vote_id,$domains,$socials,$golosTypeMulti,$golosType,$dateStart,$dateFinish,$owner)
    {
        $sql = "INSERT INTO `params`(`object_id`,
`attr_id`,`str_value`,`num_value`,`date_value`,`list_value`) "
        "VALUES
($vote_id,26,null,$owner,null,null), " //владелец
        "($vote_id,3,$dateStart,null,null,null), " //дата
        начала
        "($vote_id,4,$dateFinish,null,null,null), " //дата
        конца
        "($vote_id,7,null,null,null,$golosTypeMulti), " //мульти
        тип голосования (list)
        "($vote_id,8,null,null,null,$golosType), " //тип
        ответа
        "($vote_id,9,null,null,null,15), "
        //опубликовано ли

```

```

"($vote_id,11,null,null,null,14) "; //забанено
ЛИ
        if(!$this->linkDb->query($sql))
        {
            return false;
        }

        if (time() > $dateStart && $dateFinish > time())
        {
            $sql = "INSERT INTO `params`(`object_id`,
`attr_id`,      `str_value`,      `num_value`,      `date_value`,
`list_value`) "

                                                    "VALUES
($vote_id,44,null,null,null,11) "; //активно
ЛИ
            if(!$this->linkDb->query($sql))
            {
                return false;
            }
        }
        else if (time() < $dateStart)
        {
            $sql = "INSERT INTO `params`(`object_id`,
`attr_id`,      `str_value`,      `num_value`,      `date_value`,
`list_value`) "

                                                    "VALUES
($vote_id,44,null,null,null,17) "; //активно
ЛИ
            if(!$this->linkDb->query($sql))
            {
                return false;
            }
        }
        else if (time() > $dateFinish)
        {
            $sql = "INSERT INTO `params`(`object_id`,
`attr_id`,      `str_value`,      `num_value`,      `date_value`,
`list_value`) "

                                                    "VALUES
($vote_id,44,null,null,null,12) "; //активно
ЛИ
            if(!$this->linkDb->query($sql))
            {
                return false;
            }
        }

        foreach($domains as $domain)
        {
            $sql = "INSERT INTO `links`(`link_type_id`,
`p_object_id`, `ch_object_id`) "

```

```

        "VALUES (2,$vote_id,$domain)";
//домены на которых размещено
        if(!$this->linkDb->query($sql))
        {
            return false;
        }
    }

    foreach($socials as $social)
    {
        $sql = "INSERT INTO `params`(`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
        "VALUES
($vote_id,6,null,null,null,$social)"; //соц сети
        if(!$this->linkDb->query($sql))
        {
            return false;
        }
    }
    return true;
}

//INSERT DATA VOTE TEXT
public function insertDataVoteUkr($vote_id,
$questionUkr,
$questionContinueUkr,$answerTextUkr,$socialPartUkr_1,$socialP
artUkr_2)
{
    $sql = "INSERT INTO `params`(`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
    "VALUES
($vote_id,1,'$questionUkr',null,null,null), " //вопрос
    "($vote_id,2,'$questionContinueUkr',null,null,null),
    "//продолжение вопроса
    "($vote_id,10,'$socialPartUkr_1',null,null,null), " //соц
    текст 1
    "($vote_id,41,'$socialPartUkr_2',null,null,null) "; //соц
    текст 2

    if(!$this->linkDb->query($sql))
    {
        return false;
    }
}

public function insertDataVoteRus($vote_id,
$questionRus,
$questionContinueRus,$answerTextRus,$socialPartRus_1,$socialP
artRus_2)

```

```

        {
            $sql = "INSERT INTO `params`(`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
                                "VALUES
($vote_id,29,'$questionRus',null,null,null)," //вопрос
                                .
"($vote_id,30,'$questionContinueRus',null,null,null),"//продо
ложение вопроса
                                .
"($vote_id,31,'$socialPartRus_1',null,null,null), " //соц
текст 1
                                .
"($vote_id,42,'$socialPartRus_2',null,null,null) "; //соц
текст 2
            $this->linkDb->query($sql);
        }
        public function insertDataVoteEng($vote_id,
$questionEng,
$questionContinueEng,$answerTextEng,$socialPartEng_1,$socialP
artEng_2)
        {
            $sql = "INSERT INTO `params`(`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
                                "VALUES
($vote_id,35,'$questionEng',null,null,null), " //вопрос
                                .
"($vote_id,36,'$questionContinueEng',null,null,null),"//продо
ложение вопроса
                                .
"($vote_id,37,'$socialPartEng_1',null,null,null), " //соц
текст 1
                                .
"($vote_id,43,'$socialPartEng_2',null,null,null) "; //соц
текст 2
            $this->linkDb->query($sql);
        }

        public function insertDataVoteTextAnswers($vote_id,$answerTextUkr,$answerText
Rus,$answerTextEng, $languages)
        {
            $colAnswers = 0;
            $ukr = false;
            $rus = false;
            $eng = false;
            foreach ($languages as $lang)
            {
                if ($lang == 'ukr')
                {
                    $ukr = true;
                    if (count($answerTextUkr) > $colAnswers)

```

```

        {
            $colAnswers = count($answerTextUkr);
        }
    }
    if ($lang == 'rus')
    {
        $rus = true;
        if (count($answerTextRus) > $colAnswers)
        {
            $colAnswers = count($answerTextRus);
        }
    }
    if ($lang == 'eng')
    {
        $eng = true;
        if (count($answerTextEng) > $colAnswers)
        {
            $colAnswers = count($answerTextEng);
        }
    }
}
for ($i = 0; $i < $colAnswers; $i++)
{
    $sql = "INSERT INTO
`objects`(`object_type_id`) VALUES (4)";
    if (!$this->linkDb->query($sql))
    {
        return false;
    }
    $insertedId = mysqli_insert_id($this->linkDb);

    if ($ukr)
    {
        $sql = "INSERT INTO `params`(`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
            . "VALUES
($insertedId,12,'$answerTextUkr[$i]',null,null,null)";
        //варианты ответов
        $this->linkDb->query($sql);
    }
    if ($rus)
    {
        $sql = "INSERT INTO `params`(`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
            . "VALUES
($insertedId,32,'$answerTextRus[$i]',null,null,null)";
        //варианты ответов
        $this->linkDb->query($sql);
    }
    if ($eng)

```

```

        {
            $sql = "INSERT INTO `params` (`object_id`,
`attr_id`,      `str_value`,      `num_value`,      `date_value`,
`list_value`) "
                                "VALUES
($insertedId,38,'$answerTextEng[$i]',null,null,null)";
//варианты ответов
            $this->linkDb->query($sql);
        }

        $sql = "INSERT INTO `links` (`link_type_id`,
`p_object_id`, `ch_object_id`) "
                "VALUES (3,$vote_id,$insertedId)";
//привязка к голосованию
        $this->linkDb->query($sql);
    }
}

//INSERT DATA VOTE VIDEO
public function insertDataVoteVideoUkr($vote_id,
$questionUkr,
$questionContinueUkr,$socialPartUkr_1,$socialPartUkr_2)
{
    $sql = "INSERT INTO `params` (`object_id`,
`attr_id`,      `str_value`,      `num_value`,      `date_value`,
`list_value`) "
                                "VALUES
($vote_id,1,'$questionUkr',null,null,null), " //вопрос
                                "($vote_id,2,'$questionContinueUkr',null,null,null),
                                "//продолжение вопроса
                                "($vote_id,10,'$socialPartUkr_1',null,null,null), " //соц
                                текст 1
                                "($vote_id,41,'$socialPartUkr_2',null,null,null) "; //соц
                                текст 2

    if(!$this->linkDb->query($sql))
    {
        return false;
    }
}

public function insertDataVoteVideoRus($vote_id,
$questionRus,
$questionContinueRus,$socialPartRus_1,$socialPartRus_2)
{
    $sql = "INSERT INTO `params` (`object_id`,
`attr_id`,      `str_value`,      `num_value`,      `date_value`,
`list_value`) "
                                "VALUES
($vote_id,29,'$questionRus',null,null,null), " //вопрос

```

```

"($vote_id,30,'$questionContinueRus',null,null,null),
"//продолжение вопроса

"($vote_id,31,'$socialPartRus_1',null,null,null), " //соц
текст 1

"($vote_id,42,'$socialPartRus_2',null,null,null) "; //соц
текст 2

        if(!$this->linkDb->query($sql))
        {
            return false;
        }
    }
    public function insertDataVoteVideoEng($vote_id,
$questionEng,
$questionContinueEng,$socialPartEng_1,$socialPartEng_2)
    {
        $sql = "INSERT INTO `params`(`object_id`,
`attr_id`,`str_value`,`num_value`,`date_value`,
`list_value`) "
"VALUES
($vote_id,35,'$questionEng',null,null,null), " //вопрос
"($vote_id,36,'$questionContinueEng',null,null,null),
"//продолжение вопроса

"($vote_id,37,'$socialPartEng_1',null,null,null), " //соц
текст 1

"($vote_id,43,'$socialPartEng_2',null,null,null) "; //соц
текст 2

        if(!$this->linkDb->query($sql))
        {
            return false;
        }
    }

    public function insertDataVoteVideoAnswers($vote_id,$answerVideoNameUkr,$answ
erVideoNameRus,$answerVideoNameEng,$answerVideoLink)
    {
        $colAnswers = count($answerVideoLink);
        for ($i = 0; $i < $colAnswers; $i++)
        // $answerVideoLink
        {
            $sql = "INSERT INTO
`objects`(`object_type_id`) VALUES (4)";
            if(!$this->linkDb->query($sql))
            {
                return false;
            }
        }
    }

```



```

        }
        $insertedId = mysqli_insert_id($this->linkDb);

        $sql = "INSERT INTO `params` (`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
                                "VALUES
($insertedId,14,'$answerVideoLink[$i]',null,null,null)";
//варианты ответов ссылка
        $this->linkDb->query($sql);

        if ($answerVideoNameUkr != '')
        {
            $sql = "INSERT INTO `params` (`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
                                "VALUES
($insertedId,27,'$answerVideoNameUkr[$i]',null,null,null)";
//варианты ответов название видео Укр
            $this->linkDb->query($sql);
        }

        if ($answerVideoNameRus != '')
        {
            $sql = "INSERT INTO `params` (`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
                                "VALUES
($insertedId,33,'$answerVideoNameRus[$i]',null,null,null)";
//варианты ответов название видео Рус
            $this->linkDb->query($sql);
        }

        if ($answerVideoNameEng != '')
        {
            $sql = "INSERT INTO `params` (`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
                                "VALUES
($insertedId,39,'$answerVideoNameEng[$i]',null,null,null)";
//варианты ответов название видео Анг
            $this->linkDb->query($sql);
        }

        $sql = "INSERT INTO `links` (`link_type_id`,
`p_object_id`, `ch_object_id`) "
                                "VALUES (3,$vote_id,$insertedId)";
//привязка к голосованию
        $this->linkDb->query($sql);
    }
}

//INSERT DATA VOTE IMAGE

```

```

public function
insertDataVoteImageAnswers($vote_id,$answerImageNameUkr,$answerImageNameRus,$answerImageNameEng,$image_array)
{
    $colAnswers = count($image_array);
    for ($i = 0; $i < $colAnswers; $i++)
//$answerVideoLink
    {
        $sql = "INSERT INTO
`objects`(`object_type_id`) VALUES (4)";
        if(!$this->linkDb->query($sql))
        {
            return false;
        }
        $insertedId = mysqli_insert_id($this->linkDb);
        $sql = "INSERT INTO `params`(`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
            . "VALUES
($insertedId,13,'$image_array[$i]',null,null,null)";
//варианты ответов картинка
        $this->linkDb->query($sql);

        if ($answerImageNameUkr != '')
        {
            $sql = "INSERT INTO `params`(`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
                . "VALUES
($insertedId,28,'$answerImageNameUkr[$i]',null,null,null)";
//варианты ответов название видео Укр
            $this->linkDb->query($sql);
        }

        if ($answerImageNameRus != '')
        {
            $sql = "INSERT INTO `params`(`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
                . "VALUES
($insertedId,34,'$answerImageNameRus[$i]',null,null,null)";
//варианты ответов название видео Рус
            $this->linkDb->query($sql);
        }

        if ($answerImageNameEng != '')
        {
            $sql = "INSERT INTO `params`(`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
                . "VALUES
($insertedId,40,'$answerImageNameEng[$i]',null,null,null)";
//варианты ответов название видео Анг

```

```

        $this->linkDb->query($sql);
    }

    $sql = "INSERT INTO `links`(`link_type_id`,
`p_object_id`, `ch_object_id`) "
        . "VALUES (3,$vote_id,$insertedId)";
    //привязка к голосованию
    $this->linkDb->query($sql);
    }
    public function insertDataVoteImageUkr($vote_id,
$questionUkr,
$questionContinueUkr,$socialPartUkr_1,$socialPartUkr_2)
    {
        $sql = "INSERT INTO `params`(`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
            . "VALUES
($vote_id,1,'$questionUkr',null,null,null), " //вопрос
            . "($vote_id,2,'$questionContinueUkr',null,null,null),
            . ";//продолжение вопроса
            . "($vote_id,10,'$socialPartUkr_1',null,null,null), " //соц
            . "текст 1
            . "($vote_id,41,'$socialPartUkr_2',null,null,null) "; //соц
            . "текст 2

        if(!$this->linkDb->query($sql))
        {
            return false;
        }
    }
    public function insertDataVoteImageRus($vote_id,
$questionRus,
$questionContinueRus,$socialPartRus_1,$socialPartRus_2)
    {
        $sql = "INSERT INTO `params`(`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
            . "VALUES
($vote_id,29,'$questionRus',null,null,null), " //вопрос
            . "($vote_id,30,'$questionContinueRus',null,null,null),
            . ";//продолжение вопроса
            . "($vote_id,31,'$socialPartRus_1',null,null,null), " //соц
            . "текст 1
            . "($vote_id,42,'$socialPartRus_2',null,null,null) "; //соц
            . "текст 2

```

```

        if(!$this->linkDb->query($sql))
        {
            return false;
        }
    }

    public function insertDataVoteImageEng($vote_id,
$questionEng,
$questionContinueEng,$socialPartEng_1,$socialPartEng_2)
    {
        $sql = "INSERT INTO `params`(`object_id`,
`attr_id`, `str_value`, `num_value`, `date_value`,
`list_value`) "
        "VALUES
($vote_id,35,'$questionEng',null,null,null), " //вопрос
"($vote_id,36,'$questionContinueEng',null,null,null),
"//продолжение вопроса
"($vote_id,37,'$socialPartEng_1',null,null,null), " //соц
текст 1
"($vote_id,43,'$socialPartEng_2',null,null,null) "; //соц
текст 2

        if(!$this->linkDb->query($sql))
        {
            return false;
        }
    }

    public function getDomains($user_id)
    {
        $sql = "SELECT p1.str_value, p1.object_id FROM
params AS p1 "
        . "JOIN params AS p2 USING(object_id) "
        . "JOIN links ON (p1.object_id =
ch_object_id) "
        . "WHERE link_type_id=1 AND
p2.num_value=0 AND p1.attr_id=18 AND p_object_id = " . $user_id
;
        $result = $this->linkDb->query($sql);
        return $this->resultToArray($result);
    }

    private function resultToArray($result){
        $arrayedResult = array();
        while ($data = mysqli_fetch_assoc($result)){
            $arrayedResult[] = $data;
        }
        return $arrayedResult;
    }
}

```

ДОДАТОК В. Код файла TransDomainScriptView.php

```
<?php header('Access-Control-Allow-Origin: *'); ?>
<?php
    $arr = array ();
    echo $_GET['callback'].'(json_encode ($arr).)';
    //echo $_GET['callback']. $data['request'];
?>
```

Код файлу TransDomainScriptController.php

```
<?php

/*
 * To change this license header, choose License Headers
in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 * Description of delUserController
 *
 * @author s.holovanenko
 */
class TransDomainScriptController extends Controller{
    private $model;
    //private $view;

    function __construct($linkDb) {
        $this->model = new TransDomainScriptModel($linkDb);
        $this->view = new View();
    }

    function indexAction($arg = NULL) {

        $data = $this->model->getData();
        //$this->view-
        >genView('transDomainScriptView.php',
        'templateView.php','request','',,$data);
        header("location:
        ".ADMIN_PATH."/app/crossScript/TransDomainScript.php");
        //$this->view-
        >genView('transDomainScriptView.php',
        'templateEmptyView.php','request','',,$data);
    }
}
```

Код файла TransDomainScriptModel.php

```
<?php
class TransDomainScriptModel extends Model{
    private $linkDb;

    public function __construct($linkDb) {
```

```

        $this->linkDb = $linkDb;
    }

    public function getData() {
        $data = Array();
        $data['request'] = $this->createJSON();
        return $data;
    }

    public function createJSON ()
    {
        $arr = array ('qwer'=>'Bogdan molodec');
        return '(' . json_encode ($arr) . ');';
    }
    /*
    public function isGolosId($id){
        $sql = "SELECT `object_id` AS `object_id` FROM
`objects` WHERE object_id = $id AND object_type_id=5";
        $result = $this->linkDb->query($sql);
        if ($result->num_rows > 0) {
            return true;
        } else {
            return false;
        }
    }

    public function getGolosType($id){
        $sql = "SELECT `list_value` FROM `params` WHERE
object_id = $id AND attr_id = 8";
        $result = $this->linkDb->query($sql);
        $result = $result->fetch_assoc();
        return $result;
    }

    public function getOwner($golosId){
        $sql = "SELECT num_value FROM `params` WHERE
`object_id`=" . $golosId . " AND `attr_id`=26";
        $result = $this->linkDb->query($sql);
        if ($result->num_rows > 0) {
            $result = $result->fetch_assoc();
            return $result['num_value'];
        } else {
            return false;
        }
    }

    public function golosDel($id) {
        $sql = "SELECT p1.object_id FROM params AS p1 "
            . "JOIN params AS p2 USING(object_id) "
            . "JOIN links ON (p1.object_id =
ch_object_id) "
            . "WHERE link_type_id=3 AND p_object_id =
" . $id;
        $result = $this->linkDb->query($sql);
    }

```

```

        $answers = $this->resultToArray($result);
        $colAnswers = count($answers);
        for ($i = 0; $i < $colAnswers; $i++)
        {
            $sql = "DELETE FROM `objects` WHERE
`object_id`=" . $answers[$i]['object_id'];
            $this->linkDb->query($sql);
        }
        $sql = "DELETE FROM `objects` WHERE `object_id`="
$id";
        $this->linkDb->query($sql);
    }
    private function resultToArray($result){
        $arrayedResult = array();
        while ($data = mysqli_fetch_assoc($result)){
            $arrayedResult[] = $data;
        }
        return $arrayedResult;
    }
    */
}

```