НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:
Завідувач кафедри
_____ Сергій СТІРЕНКО
(підпис)

“__” _____2023 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою “Комп'ютерні системи та мережі”

спеціальності 123 “Комп’ютера інженерія”

на тему: Веб застосунок для інтерактивного спілкування між авторами та читачами

Виконав : студент __4__ курсу, групи _ІВ-92
(шифр групи)

_____Кузьмич Андрій Анатолійович_____
(прізвище, ім’я, по батькові) (підпис)

Керівник _____асистент Пустовіт Л. М._____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант (нормоконтроль) ст. викл. Виноградов Ю.М._____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент __ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2023 р.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп'ютерні системи та мережі”

спеціальності 123 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій СТІРЕНКО
(підпис)
“__” _____2023 р.

ЗАВДАННЯ
на бакалаврський дипломний проєкт студента

Кузьмича Андрія Анатолійовича

1. Тема проєкту *Веб застосунок для інтерактивного спілкування між авторами та читачами.*
керівник проєкту Пустовіт Леонід Михайлович, асистент,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від

2. Термін здачі студентом закінченого проєкту _____ *2023 р.*

3. Вихідні дані до проєкту *технічна документація, теоретичні дані.*

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

Розділ 1. Теоретичні основи проектування додатків інтерактивного спілкування.

Розділ 2. Особливості проектування додатків інтерактивного спілкування.

Розділ 3. Етапи розробки додатку інтерактивного спілкування за допомогою фреймворку Django.

Розділ 4. Візуалізація роботи створеного застосунку.

5. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень) структурна схема системи, функціональна схема (діаграма класів), алгоритм дій програмного забезпечення.

6. Консультанта проєкту, з вказівкою розділів проєкту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Виноградов Ю.М.		

7. Дата видачі завдання « » 2023 р.

Календарний план

№ п/п	Найменування етапів дипломного проєкту	Терміни виконання етапів проєкту	Примітки
1.	Затвердження теми проєкту		
2.	Вивчення та аналіз завдання		
3.	Розробка архітектури та загальної структури системи		
4.	Розробка структур окремих підсистем		
5.	Програмна реалізація системи		
6.	Оформлення пояснювальної		
7.	Захист програмного продукту		
8.	Передзахист		
9.	Захист		

Студент-дипломник _____ Андрій КУЗЬМИЧ
(підпис)

Керівник проєкту _____ Леонід ПУСТОВІТ
(підпис)

АНОТАЦІЯ

У цій бакалаврській дипломній роботі було розроблено веб-застосунок для інтерактивного спілкування між авторами та читачами, використовуючи мову програмування Python. Цей застосунок надає можливість проводити бесіди та обмінюватися ідеями через таку глобальну мережу як Інтернет.

ANNOTATION

In this bachelor's thesis, a web application was developed for interactive communication between authors and readers using the Python programming language. This application provides an opportunity to hold conversations and exchange ideas over a global network such as the Internet.

справки	Формат	*Значення*	*Найменування*	Кіл. листів	№ екземпля	*Додаток*
	А4		Завдання на дипломний проект			
	А4	***ІАЛЦ.467200.001 ВП***	Відомість проекту	2		
	А4	***ІАЛЦ.467200.002 ТЗ***	Технічне завдання	3		
	А4	***ІАЛЦ.467200.003 ПЗ***	Пояснювальна записка	100		
	А4	***ІАЛЦ.467200.004 Д1***	Структурна схема	1		
	А4	***ІАЛЦ.467200.005 Д2***	Структурна схема	1		
	А4	***ІАЛЦ.467200.006 Д3***	Функціональна схема	1		
	А4	***ІАЛЦ.467200.007 Д4***	Функціональна схема	1		
	А4	***ІАЛЦ.467200.007 Д5***	Лістинг програми	1		

Зм	*Лист*	*№ докум.*	*Підп*	*Дата*	***ІАЛЦ.467200.001 ВП***		
Розроб		Кузьмич А.А.			*Веб застосунок для інтерактивного спілкування між авторами та читачами.* Опис альбому	Літ.	Аркуш 1 / Аркушів 1
Перев		Пустовіт Л.М.					
						КПІ ім. Ігоря Сікорського ФІОТ ІВ-92	

ТЕХНІЧНЕ ЗАВДАННЯ
ДО ДИПЛОМНОГО ПРОЄКТУ

на тему: «*Веб застосунок для інтерактивного спілкування між авторами та читачами.*»

Київ – 2023

ЗМІСТ

НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ 2

ПІДСТАВИ ДЛЯ РОЗРОБКИ 2

МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ 2

ДЖЕРЕЛА РОЗРОБКИ 2

ТЕХНІЧНІ ВИМОГИ 2

Вимоги до розробленого продукту 2

ЕТАПИ РОЗРОБКИ 3

1 НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Найменування: Веб застосунок для інтерактивного спілкування між авторами та читачами. Область застосування: використання веь додатку для публікації новин та обговорення їх в коментарях.

2 ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки даної системи є завдання для виконання роботи кваліфікаційно-освітнього рівня «бакалавр інженерії програмного забезпечення», який був затверджений факультетом “Інформатики та обчислювальної техніки” кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний інститут ім. Ігоря Сікорського».

3 МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Мета проекту полягає в тому, щоб створити веб додаток, який дозволить користувачам обмінюватись новинами та обговорювати їх в коментарях.

4 ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки даного дипломного проекту є офіційні документації, публікації та статті в мережі Інтернет на дану тему, науково-технічна література.

5 ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробленого продукту

Розроблена система має виконувати такі вимоги:

- Одночасно може працювати з необмеженою кількість осіб
- Можливість обмінюватись новинами
- Можливість обговорення новин інших користувачів
- Повинен бути зручним у використанні

- Мати цікавий дизайн, щоб викликати інтерес у користувача.
- Мати сторінку для адміністрування

6 ЕТАПИ РОЗРОБКИ

Назва етапів виконання	Термін виконання
Затвердження теми роботи	
Вивчення та аналіз завдання	
Розробка архітектури та загальної структури додатку	
Розробка структур окремих частин додатку	
Програмна реалізація системи	
Виправлення помилок	
Оформлення пояснювальної записки	
Попередній захист та проходження нормативного контролю	
Захист дипломного проєкту	

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОГО ПРОЄКТУ

на тему: «*Веб застосунок для інтерактивного спілкування між авторами та читачами*»

Київ – 2023

ЗМІСТ

ВСТУП 5

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРОЕКТУВАННЯ ДОДАТКІВ ІНТЕРАКТИВНОГО СПІЛКУВАННЯ 8

1.1. Види та підвиди додатків мультимедіа 8

1.2. Найпоширеніші області застосування мультимедійних додатків 14

1.3. Технології створення додатків мультимедіа 15

1.4. Аналіз платформ для створення додатків мультимедіа 18

РОЗДІЛ 2. ОСОБЛИВОСТІ ПРОЕКТУВАННЯ ДОДАТКІВ ІНТЕРАКТИВНОГО СПІЛКУВАННЯ 22

2.1. Особливості мов програмування для створення додатків (Python, PHP, Java, Ruby) 22

2.2. Сутність, типологія та функціонал фреймворків 31

2.2.1. Особливості використання веб-фреймворку Yii та Django Framework 38

2.2.2. Особливості використання веб-фреймворку Laravel 49

2.2.3. Особливості використання веб-фреймворку Spring 54

2.2.4. Особливості використання веб-фреймворку Ruby on Rails 56

РОЗДІЛ 3. ЕТАПИ РОЗРОБКИ ДОДАТКУ ІНТЕРАКТИВНОГО СПІЛКУВАННЯ ЗА ДОПОМОГОЮ ФРЕЙМВОРКУ DJANGO 61

3.1. Опис функціональності розробки 61

РОЗДІЛ 4. ВІЗУАЛІЗАЦІЯ РОБОТИ СТВОРЕНОГО ЗАСТОСУНКУ 90

ВИСНОВКИ 99

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 101

Реферат

Дипломна робота за темою**: Веб застосунок для інтерактивного спілкування між авторами та читачами.**

Виконав студент: Кузьмич Андрій Анатолійович

Дипломна робота містить: 101 сторінка друкованого тексту, 20 рисунків, 13 таблиць та нараховує 25 джерел використаної інформації.

Мета дослідження полягає в розробці веб застосунку для інтерактивного спілкування між авторами та читачами.

Об’єкт дослідження: розробка веб застосунку.

Предмет дослідження: сукупність необхідних умов, що забезпечують найкращий підхід до створення веб застосунку для інтерактивного спілкування між авторами та читачами.

Завдання дослідження можна сформулювати так:

- розглянути основні види та підвиди додатків мультимедіа;
- перелічити найпоширеніші області застосування мультимедійних додатків;
- проаналізувати технології створення додатків мультимедіа та провести аналіз платформ для їх створення;
- дослідити особливості таких мов програмування для створення додатків, як Python, PHP, Java, Ruby;
- розкрити сутність, типологія та функціонал таких фреймворків, як Yii, Django Framework, Laravel, Spring та Ruby on Rails;
- описати головні етапи розробки додатку інтерактивного спілкування за допомогою фреймворку django;
- дати опис функціональності розробки та провести тестування працездатності застосунку.

Відзначено, що веб-застосунки представляють собою поєднання різних способів візуалізації інформації. Зазвичай, це означає поєднання таких

елементів, як текст, звук, графіка, анімація, відео та тривимірне моделювання в межах комп'ютерної системи. При дослідженні технологій створення веб-застосунків розробляється сценарій, який описує їх процес створення. З цього випливає, що кожен веб-застосунок для інтерактивного спілкування складається з різних компонентів та тематик. Варто зазначити, що мобільні додатки надають своїм користувачам різноманітні функціональні можливості, такі як швидке зв'язку з іншими, потужні мобільні процесори та портативні камери.

Досліджені найпопулярніші платформи для розробки мобільних додатків, такі як Android, iPhone, Python, Java Me та BlackBerry. Більшість з них є безкоштовними та легкими для освоєння, що сприяє постійному оновленню та розширенню мобільних додатків. Були описані особливості поширення різних мов програмування, їх унікальні риси та відмінності. Було зазначено, що серед найпопулярніших фреймворків для розробки мультимедійних додатків для платформи Android можна виділити такі, як Spring (Java), Yii (PHP), Laravel (PHP), Ruby on Rails (Ruby). Були описані етапи розробки інтерактивного веб-застосунку за допомогою фреймворку Django. Були описані функціональні та нефункціональні вимоги до розробки. Була надана діаграма класів та цілей веб-застосунку для інтерактивного спілкування між авторами та читачами. Була представлена програмна реалізація веб-застосунку та перевірена його працездатність.

YII, DJANGO FRAMEWORK, LARAVEL, SPRING TA RUBY ON RAILS, ФРЕЙМВОРК, PYTHON, PHP, JAVA, ВЕБ ЗАСТОСУНОК, СПІЛКУВАННЯ.

Рік виконання дипломної роботи – 2023

ВСТУП

У сучасному світі індустрія розробки веб-застосунків стрімко розвивається і стала однією з провідних та надзвичайно популярних галузей на початку XXI століття. Кількість веб-розробників постійно зростає, що призводить до збільшення числа самих застосунків. Багато компаній та організацій проявляють інтерес до створення програм, які допоможуть їм досягти успіху в своїй галузі та випередити конкурентів.

Основною перевагою веб-застосунків є їх доступність для користувачів з будь-якого місця, якщо у них є доступ до Інтернету, хоча деякі застосунки можуть працювати навіть в автономному режимі. Це великою мірою спрощує процес їх використання та дозволяє впроваджувати нові можливості.

Актуальність дослідження. Застосунки веб-технологій є збірними інформаційними засобами, що поєднують різноманітні способи подання даних. Це включає в себе текст, звук, графіку, мультиплікацію, відео та просторове моделювання. Таке поєднання різних видів представлення інформації значно підвищує рівень сприйняття, оскільки користувач не просто читає інформацію, а активно взаємодіє з нею. Веб-застосунки надають можливість впливати на різні органи чуття користувача, що сприяє більш ефективній взаємодії та кращому сприйняттю.

Для створення веб-застосунків доступний широкий спектр платформ, кожна з яких має власні характеристики та зазвичай не є сумісною з іншими (тобто програма, створена для однієї платформи, не зможе функціонувати на іншій). Більш того, кожен мобільний пристрій підтримує лише деякі платформи. Таким чином, для того, щоб максимізувати ринок і прибуток від додатків, розробникам важливо визначати, на яких платформах вони будуть працювати. Після появи перших портативних комп'ютерів у 1980-х, популярність таких платформ значно посилилася.

При вивченні технології розробки веб-застосунків розробляється сценарій, який описує процес їх створення. Таким чином, логічно вважати, що кожен веб-

застосунок для інтерактивного спілкування складається з різних компонентів, що відповідають різним аспектам.

Створення веб-застосунку для інтерактивного спілкування може бути розділене на кілька етапів, включаючи вибір тематики, розмітку робочого простору, використання шарів, створення символів, написання скриптів, роботу зі звуком, додавання тексту, створення ефектів, імпортування зображень, використання готових компонентів, налаштування навігації та використання мов розмітки та скриптових мов. Дослідження, що представлене, присвячене розробці веб-застосунку для інтерактивного спілкування між авторами та читачами.

Мета дослідження полягає в розробці веб застосунку для інтерактивного спілкування між авторами та читачами.

Об'єкт дослідження: розробка веб застосунку.

Предмет дослідження: сукупність необхідних умов, що забезпечують найкращий підхід до створення веб застосунку для інтерактивного спілкування між авторами та читачами.

Завдання дослідження можна сформулювати так:

- розглянути основні види та підвиди додатків мультимедіа;
- перелічити найпоширеніші області застосування мультимедійних додатків;
- проаналізувати технології створення додатків мультимедіа та провести аналіз платформ для їх створення;
- дослідити особливості таких мов програмування для створення додатків, як Python, PHP, Java, Ruby;
- розкрити сутність, типологія та функціонал таких фреймворків, як Yii, Django Framework, Laravel, Spring та Ruby on Rails;
- описати головні етапи розробки додатку інтерактивного спілкування за допомогою фреймворку django;

– дати опис функціональності розробки та провести тестування працездатності застосунку.

Методи дослідження: перегляд наявної методичної та наукової літератури з подальшим аналізом зібраного матеріалу, проведення порівнянь, класифікація, проектування, теоретичне моделювання, встановлення причинно-наслідкових зв'язків, огляд документації та результатів діяльності дослідників, пов'язаних з темою дослідження, а також експертна оцінка та проведення результуючого експерименту.

В роботі також використано такі методи: аналітичний – використання математичного апарату при обробці фактичного матеріалу; логічний метод використовується в процесі всього дослідження, в т.ч. дедукція, класифікація матеріалу, постановка та шляхи вирішення проблеми.

Наукова новизна роботи полягає у наявності оригінального дослідницького матеріалу по напрямку проведеного дослідження.

Теоретична та практична цінність роботи полягає в наявності теоретичного матеріалу по дослідженню, відсіяного з-поміж іншого в процесі пошуку інформації по темі, та в систематизації матеріалу напрямку дослідження. Проведене дослідження має більш глибокий ступінь розробки напрямку дослідження, відносно попередніх досліджень вчених, дисертантів/

Для вирішення вищеназваних завдань, ми використовуємо такі джерела інформації, як збірники наукових праць, монографії, періодичні видання, підручники, довідники, а також спеціалізовані фахові журнали.

Структура роботи. Дипломна робота організована з урахуванням цілей і завдань дослідження та містить вступ, чотири розділи, висновки, список використаних ресурсів та додатки. Обсяг дослідження становить 101 сторінку машинописного тексту. Під час роботи було аналізовано 25 літературних джерел.

РОЗДІЛ 1.
ТЕОРЕТИЧНІ ОСНОВИ ПРОЕКТУВАННЯ ДОДАТКІВ ІНТЕРАКТИВНОГО СПІЛКУВАННЯ

1.1. Види та підвиди додатків мультимедіа

Мобільні додатки, або аплікації, є типом програмного забезпечення, яке спеціально розроблене для використання на смартфонах та планшетах. Ці додатки можна встановити безпосередньо на пристрої або ж завантажити їх з онлайн-магазинів застосунків, таких як App Store, Google Play, Windows Phone Store та інші [25]. Сьогодні, у нашому сучасному світі, важко уявити повсякденне життя без мобільних телефонів, адже вони надають нам широкий спектр функціональності, включаючи швидкі комунікації, потужні обчислювальні засоби, камеру, адресну книгу та, звісно, доступ до таких ресурсів, як Вікіпедія.

Розповсюдження мобільних пристроїв привносить новітні можливості їх застосування у різноманітних секторах, як-от бізнес, освіта та розваги. Важливістю набуває їх використання у навчальному процесі, що підкреслює необхідність розроблення свіжих підходів до освіти, а також створення відповідних навчальних матеріалів та технологій, які беруть до уваги потенціал мобільних пристроїв.

Зі зростанням функціональних можливостей сучасних мобільних пристроїв з різними операційними системами, такими як Android, iPhone та Windows CE, з'являється все більше сервісів, які пропонують ефективні мобільні додатки, багато з яких є безкоштовними. При вивченні ефективності використання мобільних додатків у навчальній сфері, варто зазначити, що правильне їх використання має багато переваг. Однак, для вибору простого, але інформативного додатку слід дотримуватись декількох порад, таких як чітке визначення його змісту та спрямованості, вивчення ринку аналогічних пропозицій та вибір найкращого варіанту.

У класі додатків потокового аудіо/відео в реальному часі, клієнти отримують доступ до вмісту, що транслюється в реальному часі. Це можуть бути

прямі трансляції спортивних подій, концертів, новинних програм, вебінарів або онлайн-уроків. У таких додатках інформація передається безпосередньо з сервера до клієнта в режимі реального часу. Користувач може спостерігати за подіями, які відбуваються в даний момент, і отримувати актуальну інформацію миттєво.

Інтерактивні аудіо/відео додатки в реальному часі дозволяють користувачам активно взаємодіяти з контентом, що транслюється. Це можуть бути онлайн-конференції, відеодзвінки, чати або вебінари, де користувачі можуть спілкуватися між собою, обмінюватися повідомленнями, задавати запитання та отримувати відповіді в режимі реального часу. Такі додатки забезпечують активну взаємодію і співпрацю між користувачами, створюючи можливості для динамічних обмінів інформацією.

Кожен з цих класів додатків має свої особливості і відмінності, проте вони спільно допомагають створити інтерактивне спілкування та забезпечують різноманітні можливості для користувачів отримувати, передавати та взаємодіяти з медіа-контентом.

Програми, які працюють у режимі безперервного відтворення, надають можливість користувачам насолоджуватись мультимедійним контентом без перебоїв. Наприклад, додатки, як Fex.net і Google Drive, дозволяють відтворювати медіафайли безперервно, зберігаючи вихідний час запису. Таким чином, користувачі можуть насолоджуватись неперервним потоком аудіо та відео.

Додатки для потокового аудіо і відео в реальному часі подібні до традиційного радіо і телебачення, але передача відбувається через Інтернет. Ці додатки надають можливість користувачам приймати прямі радіо- або телепередачі з будь-якого місця у світі. Оскільки цей тип медіа не зберігається локально, користувачі не можуть перемотувати його вперед. Проте деякі додатки дозволяють паузувати та перемотувати пряму трансляцію.

Інтерактивні додатки для аудіо та відео в реальному часі дозволяють людям спілкуватися один з одним за допомогою аудіо та відео в режимі

реального часу. Наприклад, вони можуть використовуватись для ведення Інтернет-телефонії або проведення відеоконференцій. Ці додатки надають можливість активно взаємодіяти з медіаконтентом у реальному часі та спілкуватися безпосередньо один з одним.

Інтернет-телефонія може потенційно надати приватну телефонну станцію та локальний або міжміський телефонний сервіс за дуже доступною вартістю. Вона також може полегшити впровадження нових сервісів, які є складними для підтримки в традиційних мережах з комутацією каналів. За допомогою інтерактивного відео в реальному часі, інакше званого відеоконференцією, люди мають можливість спілкуватися візуально і вербально. У програмах інтерактивного аудіо/відео в реальному часі користувач може в будь-який момент висловлюватися або рухатися. Такі програми, як ZOOM, Viber, Telegram, Google Meets, є хорошими прикладами.

Інтернет-телефонія відкриває багато можливостей, таких як приватна телефонна станція та доступний місцевий та міжміський зв'язок за низькими витратами. Вона також сприяє швидкому розгортанню нових послуг, які важко забезпечити у традиційних канальних мережах. Інтерактивні відеоконференції, або відеозв'язок в реальному часі, дозволяють людям спілкуватися візуально та голосовими комунікаціями. У додатках для інтерактивного аудіо/відео зв'язку в реальному часі користувачі можуть говорити та рухатися в будь-який момент. Прикладами таких додатків є ZOOM, Viber, Telegram, Google Meet, які надають можливість вести відеоконференції та спілкуватися в реальному часі.

Впровадження мобільного навчання вимагає спеціального підходу до розробки інформаційних ресурсів. Особливості цих ресурсів полягають у врахуванні обмежень мобільних пристроїв, таких як маленький розмір екрану, обмежена клавіатура та пропускна здатність для мультимедійних файлів. Дизайн мобільної інформації повинен бути пристосований до цих особливостей, а не просто повторювати стандартну структуру звичайного контенту. Наприклад, розробка мобільних версій наукових або навчальних сайтів, які забезпечують

швидкий доступ та оптимізовані параметри для роботи з мобільними пристроями, є актуальною.

При розробці мультимедійних матеріалів слід враховувати формати, які підтримуються мобільними пристроями, такі як PNG, GIF, 3GP, MP3, MP4. Ефективним є створення коротких навчальних програм, розрахованих на 10-15 хвилин навчання, оскільки мобільні пристрої використовуються в перервах між основними видами діяльності. Тому мобільне навчання повинно займати короткі проміжки часу.

Розробка навчальних матеріалів повинна враховувати анатомічні особливості користувачів, такі як розмір пальців і якість зору. Мобільні навчальні матеріали повинні мати великий чіткий шрифт і зрозумілі зображення. Важливо уникати перевантаження матеріалів великими мультимедіа, оскільки це може вплинути на швидкість завантаження. Також потрібно постійно оновлювати навчальні матеріали, слідкуючи за останніми змінами та розвитком області навчання.

Є чимало різних платформ для створення мобільних застосунків, проте вони не є взаємно сумісними. Для розробників важливо точно вибрати, які платформи вони планують обслуговувати, це допоможе їм розширити цільовий ринок та підвищити доходи від своїх програм. Інтерес до мобільних платформ значно збільшився з моменту випуску перших переносних комп'ютерів у 1980-х роках.

Кожна платформа для мобільних додатків має своє середовище розробки, яке надає інструменти для проектування, тестування та розгортання додатків відповідно до обраної платформи.

У веб-додатках для інтерактивного спілкування використовуються різні компоненти та технології. Мультимедійні додатки можна поділити на різні підвиди залежно від їх характеристик і функціональності.

У свою чергу, можна класифікувати мультимедійні додатки на різні підвиди. Всі основні категорії мультимедійних додатків можна знайти у таблиці

1.1. Ці підвиди відображають різноманітність застосування мультимедійних додатків у різних сферах життя та діяльності.

Таблиця 1.1 - Основні поняття підвидів мультимедіа-додатків

Анімація	По-кадрова анімація використовує кадри зображень, які поступово змінюються, щоб створити візуальний ефект руху. Програмна анімація використовує програмні скрипти або алгоритми для зміни зображень або елементів візуального вмісту з певною послідовністю дій. Вона може використовувати змінні, які контролюють швидкість, напрямок, розмір або інші аспекти змін візуального вмісту. Обидва підходи використовуються для створення руху і анімації, але вони відрізняються в способах, якими досягається цей ефект. По-кадрова анімація базується на послідовності зображень, тоді як програмна анімація використовує програмні скрипти або алгоритми для управління змінами візуального вмісту.
Додатки для web:	Банери - це графічні зображення або текстові блоки рекламного характеру, що використовуються в Інтернеті для привертання уваги відвідувачів і перенаправлення їх на веб-сторінку з розширеним описом продукту або послуги. Це ефективний спосіб залучення потенційних клієнтів і створення позитивного іміджу. Додатки для передачі даних, такі як гостьова книга, використовуються для обміну і передачі інформації між користувачами. Наприклад, гостьова книга дозволяє відвідувачам залишати коментарі, повідомлення або відгуки на веб-сторінці. Це зручний спосіб взаємодії зі спільнотою або збору даних від користувачів.
Ігри	Розважальні ігри - це програми, які призначені для розваги і дозволюють користувачеві провести час весело і захопливо. Ці ігри можуть мати різноманітні жанри, включаючи головоломки, стратегії, пригоди, гонки тощо. Вони сприяють відпочинку, розвагам і відволікають від повсякденних справ. Навчальні ігри - це програми, які мають навчальну спрямованість і допомагають користувачеві підвищити рівень своїх знань в певній області. Ці ігри комбінують навчання з ігровим процесом, що робить їх більш захопливими та ефективними. Вони можуть охоплювати різні теми, включаючи мови, математику, науку, історію тощо, і надають можливість користувачеві вчитися в легкій ігровій формі.

Продовження таблиці 1.1

Мультимедіа-галереї	Кадрова зміна зображень - це процес зміни зображень або картинок через певний проміжок часу. Це може бути використано для створення анімаційних ефектів, створення слайд-шоу або зміни зображень на веб-сторінках з певною частотою. Панорама - це широке та багатопланове зображення або перспектива, яка дає можливість користувачеві оглядати великий відкритий простір. Це може бути створено за допомогою зшивання кількох фотографій чи використання спеціальної технології для створення панорамного зображення. Інтерактивна галерея - це галерея або колекція зображень, яка надає користувачеві можливість управляти та навігувати по зображеннях. Користувач може взаємодіяти з галереєю, переглядати різні зображення, збільшувати їх, переходити між зображеннями або виконувати інші дії, що дають більше контролю над переглядом зображень.
Відеоплеєри	Формування по-кадрового фільму - це процес підготовки та розташування зображень, послідовності фотографій або кадрів, що створюють враження руху. Цей процес включає в себе вибір, сортування та впорядкування зображень таким чином, щоб вони логічно поєднувалися та створювали плинний рух, надаючи візуальний ефект анімації. Мультимедійний плеєр для потокового відео - це програмне забезпечення, що дозволяє програвати потокове відео у форматах, таких як avi, mpeg та інші. Він забезпечує можливість керування відтворенням відео, такими операціями, як запуск, пауза та перемотування на початок відеофрагменту. Потокове відео передається через мережу і відтворюється в режимі реального часу без необхідності повного завантаження всього відеофайлу.
Презентація	Лінійна презентація - це динамічний ролик, що містить складну графіку, відеовставки, звуковий супровід, але не має системи навігації. Вона презентує вміст у послідовному порядку, з обмеженими можливостями користувача для вибору та перегляду конкретних елементів. Лінійна презентація показується визначеним способом, без можливості вибору порядку чи навігації у вмісті. Інтерактивна презентація - це сукупність мультимедійних компонентів, що організовані за ієрархічним принципом та керовані через спеціальний інтерфейс для користувача. Вона надає більшу взаємодію та контроль користувача над презентацією. Користувач може вибирати порядок перегляду елементів, переходити між різними розділами, виконувати дії та взаємодіяти з вмістом презентації. Інтерактивна презентація надає більшу гнучкість і можливість персоналізації для користувача.

Кінець таблиці 1.1

Для створення веб-застосунків для інтерактивного спілкування, вони можуть складатися з таких компонентів:

1. Вибір теми створюваного мультимедійного додатку.

2. Розмітка робочої області, включаючи масштаби і фон.

3. Кадри, які використовуються для структурування і відображення інформації.

4. Використання шарів, що дозволяє розділити елементи і керувати їх відображенням.

5. Створення символів різних типів, які використовуються для представлення об'єктів і дій.

6. Включення змінних і написання скриптів на мові програмування, для створення взаємодії і динамічного змінювання веб-застосунку.

7. Робота зі звуковими файлами, включаючи їх відтворення і контроль.

8. Додавання тексту, які використовуються для представлення інформації і комунікації з користувачем.

9. Створення ефектів, які поліпшують візуальний вигляд і враження веб-застосунку.

10. Використання і імпортування зображень, що додають візуальний контент до веб-застосунку.

11. Використання готових компонент бібліотек, що надають функціональність і стилізацію.

12. Створення навігації, яка дозволяє користувачам переходити між сторінками і взаємодіяти з контентом.

13. Використання мов розмітки тексту і скриптових мов, які визначають структуру і поведінку веб-застосунку.

Ці компоненти допомагають забезпечити ефективну взаємодію з користувачем і створення багатофункціонального та привабливого мультимедійного додатку.

1.2. Найпоширеніші області застосування мультимедійних додатків

Мультимедійні додатки знайшли широке застосування в різних сферах діяльності, забезпечуючи значну підтримку і прискорення роботи. Наприклад, в бізнесі вони дозволяють командам працювати разом незалежно від місця розташування, сприяють комунікації, навчанню персоналу і груповим презентаціям. У маркетингу і рекламі вони використовуються для покращення маркетингу нових продуктів шляхом використання привабливих презентацій,

анімації та звукових ефектів. В освіті мультимедійні додатки використовуються для навчання через комп'ютерні ігри та інтерактивні розвиваючі програми, які надають дітям багато можливостей для ігрового вивчення і отримання детальної інформації.

В останні роки мультимедійні додатки знаходять все більше застосування в банківській сфері. Банки використовують мультимедіа для передачі інформації клієнтам, надання фінансових послуг, показу різних схем і презентацій на моніторах ПК. Онлайн-банкінг і інтернет-банкінг є популярними серед клієнтів і широко використовують мультимедійні можливості. У медицині мультимедіа використовується для моніторингу стану пацієнтів у реальному часі, консультацій з експертами та хірургами, діагностики захворювань і навчання медичного персоналу. Мультимедійні дисплеї використовуються під час критичних операцій для відображення важливої інформації.

Мультимедійні додатки в сфері освіти можуть виступати як цікаві та мотивуючі навчальні посібники. Аудіовізуальна підтримка вчителя може сприяти зацікавленості учнів у навчальній темі. Мультимедійний репетитор може запропонувати учням різні завдання, щоб стимулювати їхній інтерес. Крім простого управління кнопками, мультимедійні додатки можуть надати інструкції з інтелектуальним моделюванням, створенням посилань, спільною роботою та системним тестуванням взаємодій з користувачем.

Комунікаційні технології та мультимедійні послуги включають в себе різні можливості, що стали доступними завдяки розвитку обчислювальної потужності і засобів зв'язку. Ці сервіси можуть охоплювати основні телебачення, інтерактивні розважальні програми, цифрове аудіо, відеоконтент на вимогу, фінансові транзакції, інтерактивні ігри для одного чи кількох гравців, цифрові мультимедійні архіви, електронні версії газет та журналів.

1.3. Технології створення додатків мультимедіа

Існує деяка класифікація програм, які дозволяють створювати мультимедійні додатки. Їх можна поділити на такі групи:

1. Спеціалізовані програми: це програми, призначені для швидкого створення певних типів мультимедійних додатків, зокрема презентацій і публікацій в Інтернеті.

2. Авторські інструментальні засоби мультимедіа: ці програми знаходяться між програмами мультимедійних презентацій і мовами програмування. Вони надають більше можливостей для взаємодії з користувачем і дозволяють створювати програмні продукти з високим ступенем взаємодії.

3. Мови програмування: цей тип програм використовується для розробки мультимедійних додатків, що мають більш широкий функціонал і можливості. Використання програмування дозволяє реалізувати більш складні ідеї, але вимагає більшого зусилля і часу.

Використання спеціалізованих програм дозволяє зекономити час і зусилля, але може обмежувати можливості програми. Застосування мов програмування надає більше гнучкості та можливостей, але вимагає більшої витрати часу та ресурсів. Авторські інструментальні засоби поєднують в собі деякі можливості програм мультимедійних презентацій і можливості розробки програмного забезпечення з взаємодією з користувачем.

Кожен з цих типів програм має свої переваги і обмеження, і вибір залежить від потреб і вимог проекту.

Останнім часом з'явилося багато систем, які дозволяють створювати мультимедійні додатки без необхідності в програмуванні. Навіть спеціалізовані мови програмування служать доповненням до можливостей таких систем. Сучасні програмувальні мови надають інструменти для прискореного створення додатків, що полегшує задачу генерації вихідного коду програми на вибраній мові після визначення користувачем зовнішнього вигляду програми.

Таким чином, вибір засобу розробки мультимедійних додатків є непростим завданням, і універсального рішення, яке б підійшло для всіх сценаріїв, не існує. Для створення мультимедійних виробів необхідний набір технічного обладнання, який включає в себе мультимедійний ПК, кольоровий сканер (можливо, планшетний) з відповідним ПЗ для обробки зображень, пристрій для

запису на CD, а також, за потреби, відеокарту для перетворення відео в цифровий формат.

Ці інструменти дозволяють організувати малу мультимедійну студію, надаючи необхідну функціональність для створення мультимедійних програм.

Розглянемо детальніше способи розробки мультимедійних додатків.

1. Створення мультимедійних презентацій:

Сучасні програми для створення презентацій, зокрема Microsoft PowerPoint, все більше акцентуються на мультимедіа. PowerPoint володіє багатим набором образотворчих і анімаційних ефектів, нерідко перевершуючи авторські інструментальні засоби мультимедіа. У відмінності від авторських систем, PowerPoint дозволяє розробляти презентації, які не мають строго дотримуватися певного сценарію, а здатні адаптуватися до реакції користувача. Додатково, програма підтримує розширення за допомогою Visual Basic, що відкриває можливості для створення складних програмних додатків. PowerPoint активно застосовується в освітньому середовищі та бізнесі для підготовки лекцій, презентацій, курсових робіт та іншого.

2. Публікація в Інтернеті:

Найпоширеніші програми для мережевої публікації включають Microsoft Front Page, Macromedia Dreamweaver і Arachnophilia. Ці програми дозволяють створювати та редагувати веб-сторінки з мультимедійним вмістом. Вони надають зручні інтерфейси для роботи з різними елементами веб-сторінок, такими як текст, графіка, аудіо та відео. Вони також підтримують розширення HTML, CSS і JavaScript, що дозволяє створювати більш інтерактивні та динамічні веб-сторінки.

Кожна з цих програм має свої переваги та особливості, і вибір конкретного інструменту залежить від потреб розробника та специфіки проекту.

У програмах Microsoft Office, включаючи PowerPoint, Word і Excel, користувачі мають змогу створювати та редагувати HTML-сторінки, а також конвертувати наявні документи до формату HTML. Наприклад, PowerPoint

дозволяє зберігати презентації в HTML форматі, що дозволяє легко публікувати їх в Інтернеті.

FrontPage, також розроблений компанією Microsoft, є програмою для створення та редагування веб-сторінок. Вона дозволяє змінювати структуру розміщення сторінок на сайті, використовувати стилі для єдиної оформлення веб-сторінок та автоматично генерувати навігаційні елементи. FrontPage поєднує в собі можливості редакторів HTML-тегів і візуальних редакторів, що спрощує процес розробки веб-сторінок.

Adobe Dreamweaver, розроблений фірмою Macromedia (згодом придбаною компанією Adobe), є професійним інструментом для побудови веб-сайтів. Ця програма надає широкі можливості для редагування HTML-коду і візуального редагування, інтегруючи засоби верстки і графічного дизайну. Dreamweaver спрощує процес розробки веб-сторінок, надаючи зручний інтерфейс та функції, які дозволяють уникнути необхідності вручну кодувати HTML теги.

Обидва ці продукти, FrontPage і Dreamweaver, мають свої особливості та набір інструментів, що дозволяє розробникам побудувати веб-сайти з різноманітними мультимедійними елементами в зручний спосіб. Вибір між ними залежить від потреб користувача та рівня професіоналізму.

1.4. Аналіз платформ для створення додатків мультимедіа

Є багато платформ для створення програм, але вони мають взаємну несумісність, це означає, що програми, створені для однієї платформи, не будуть працювати на іншій. До того ж, кожний мобільний пристрій підтримує лише певну специфічну платформу.

Тому, для збільшення ринку збуту та доходів від програм, розробники повинні чітко визначити, які платформи вони будуть підтримувати. З появою перших портативних комп'ютерів у 1980-х роках популярність цих платформ значно зросла. Багато моделей смартфонів на початку 2000-х років дозволяють користувачам встановлювати довільне програмне забезпечення.

Кожен фреймворк для створення мобільних застосунків має власне розробницьке середовище, яке забезпечує інструментарій для дизайну, перевірки та запуску програм на вибраній платформі. Аналізуючи інформацію з Інтернету та документацію мобільних платформ, можна узагальнити характеристики фреймворків для розробки мобільних застосунків в таблиці 1.2.

Таблиця 1.2 - Порівняльна характеристика платформ для розробки мобільних додатків

Платформа	Наявність емулятора	Міжплатформенне розгортання	Ціна засобів розробки	Мова програмування	Наявність вбудованого середовища для роботи	Актуальність
Android	Безкоштовний емулятор	Android	безкоштовно	Java	Eclipse, Android (Плагін для Netbeans)	Найпопулярніша наразі платформа для мобільних додатків
BlackBerry	Безкоштовний емулятор	BlackBerry	безкоштовно	Java	JDE-BlackBerry, Java середовище розробки	Використовується, найчастіше, для електронних пошт, актуально
Bedrock	наявний	Java ME, BREW, BlackBerry, iPhone, PSP	Наявність комерційної ліцензії	Java	Eclipse	Використовується для комп'ютерних ігор, досі актуально.
Java ME	Безкоштовний емулятор, Sun Java ME Java Wireless Toolkit, mpowerplayer	Так, хоча багато віртуальних машин мають конфлікти з деякими телефонами	безкоштовно	Java	Eclipse, LMA, Netbeans Mobility Pack	Актуально, використовується як для ігор, так і для простих додатків
iPhone OS	Об'єднаний з iPhone SDK, вбудований в Xcode IDE	Лише iPhone і iPod Touch	Засоби безкоштовні для Intel-based Mac,	Об'єктний С	Xcode	Одна з найпопулярніших мов, однак є дороговартісною

Продовження таблиці 1.2

Платформа	Наявність емулятора	Міжплатформе не розгортання	Ціна засобів розробки	Мова програмування	Наявність вбудованого середовища для роботи	Актуальність
Microbrowse r Based	багато	багато	безкоштовно	XHTML, WML (WAP 1.2)	Основна сторінка з настройками для різних браузерів	Веб-ресур, є безкоштовним та легким у засвоєнні
Symbian	Безкоштовний емулятор	багато	Доступні комерційні та безкоштовні засоби	C++	так	Досить популярна та використовується для пристроїв з обмеженими ресурсами
Python	Python	Декілька, включаючи плагіни до Eclipse	безкоштовно	наявний	Інтерпретована мова доступна тільки для Nokia S60, хоча є порти до інших платформ включаючи Palm OS	Потужна мова програмування, яка є легкою у засвоєнні та досить популярною
Lazarus	Так може відладжуватися на IDE через ActiveSync	Використовує емулятори платформ	Компільована мова придатна для Windows CE, linux пристроях і Symbian	Об’єктний Pascal	Lazarus IDE, включає будований дизайнер і відладчик	безкоштовно
Windows Mobile	Безкоштовний емулятор, також об’єднаний з IDE	Visual Studio, eMbedded VC++ (безкоштовно)	Безкоштовні commandline засоби чи eMbedded VC++, чи Visual Studio (Стандартний випуск чи краще)	C, C++	Windows Mobile, Windows CE	Аналог MS Windows, яка адаптована для мобільних пристроїв, однак втрачає вже свою популярність

Кінець таблиці 1.2

В таблиці представлено ТОП-10 найпопулярніших платформ для розробки мобільних додатків на сучасний момент. Більшість з цих платформ є

безкоштовними і легкими у вивченні, що дозволяє розробникам постійно оновлювати та розширювати свої мобільні застосунки.

Рис. 1.1. Найпопулярніші платформи для розробки мобільних додатків

Додатково до таблиці, створено інфографіку (рис. 1.1), яка сприяє кращому усвідомленню дослідженої інформації та полегшує її використання у майбутньому.

РОЗДІЛ 2.
ОСОБЛИВОСТІ ПРОЕКТУВАННЯ ДОДАТКІВ ІНТЕРАКТИВНОГО СПІЛКУВАННЯ

2.1. Особливості мов програмування для створення додатків (Python, PHP, Java, Ruby)

Для електронних пристроїв, таких як комп'ютери, мобільні телефони, планшети та інші, потрібна мова програмування, яка визначає команди і дії, які ці пристрої повинні виконувати. Існують різні варіації мов програмування, кожна з яких має свої особливі цілі та принципи. В основі будь-якої мови програмування лежить словник, що вміщує в себе граматичні правила, які слугують для передавання інструкцій до комп'ютера чи іншого обчислювального апарату.

Кожна мова кодування володіє унікальним комплектом ключових слів та синтаксисом для організації програмного коду. Вони поділяються на низькорівневі та високорівневі мови програмування. Мови низького рівня, як-от машинні мови або асемблери, характеризуються більш простою структурою, проте вони важчі для сприйняття. На відміну від них, високорівневі мови розроблені для комфорту програмістів і пропонують легко читаємий синтаксис із логічними словами та символами.

Протягом всієї історії комп'ютерів було створено багато різних мов програмування для різних типів розробки. Кожна з них має свої переваги та обмеження, і вибір мови залежить від конкретних потреб і вимог проекту.

Мови програмування мають широке застосування і використовуються для різних цілей. Наприклад, їх використовують для розробки програм та додатків, розвитку штучного інтелекту, розробки баз даних, відеоігор, драйверів та апаратного інтерфейсу, Інтернету та веб-сторінок, а також для написання сценаріїв.

Мови програмування спеціально розроблені для того, щоб дати точні інструкції комп'ютеру, оскільки вони не можуть тлумачити розмитий або

двозначний код. Це пов'язано з тим, що комп'ютери працюють на основі точних інструкцій, і вони не можуть розуміти невизначеності, як люди.

Однією з популярних мов програмування є Python, яка широко використовується в різних галузях IT. Python знаходить застосування в машинному навчанні, розробці програмних додатків, написанні автоматизаційних скриптів і багатьох інших областях. Це мова високого рівня, яка володіє значною універсальністю і лідирує в багатьох сферах. З часом Python завоював лідерські позиції в галузі машинного навчання, став популярним для розробки десктопних додатків та зайняв значну частку ринку веб-розробки.

Завдяки своїй універсальності, Python нерідко називають мовою програмування №1. А актуальні рейтинги і сервіси статистики тільки підтверджують це. Наприклад, у PYPL, TIOBE та statista. com «пітон» стабільно знаходиться в трійці лідерів ось уже кілька років поспіль.

Програмування стало більш динамічним завдяки наступним особливостям Python:

- Динамічна типізація. Типізація буває статичної і динамічної. У випадку з останньою у розробника немає необхідності вказувати тип змінних – ЯП призначить його самостійно.
- Зручний повернення функцією декількох значень. Розробник може перерахувати значення через кому – в цьому випадку вони перетворюються в список автоматично.
- Автоматичне виділення пам’яті. Програмісту не потрібно самостійно виділяти пам’ять – Python це зробить самостійно. Так, це трохи знижує контроль над програмою, але тут є один істотний плюс – розробка прискорюється в рази.
- Автоматичне збирання сміття. Дана опція оптимізує пам’ять і істотно спрощує процес розробки.
- Прив’язка типу даних. У Python тип даних прив’язується до значення, а не до змінних.

Мова програмування Python використовується для різних основних завдань:

1. Аналіз даних: Python є популярним вибором для роботи з даними. Він надає розширені бібліотеки, такі як pandas і NumPy, для ефективного оброблення і аналізу даних.

2. Візуалізація даних: Python має потужні інструменти для візуалізації даних, такі як matplotlib і seaborn. Це дозволяє створювати різноманітні типи графіків і візуалізацій для зрозумілого представлення даних.

3. Машинне навчання: Python є популярною мовою для розробки моделей машинного навчання. З використанням бібліотек, таких як scikit-learn і TensorFlow, можна створювати та навчати моделі для класифікації, регресії, кластеризації та інших завдань машинного навчання.

4. Розробка програмного забезпечення: Python використовується на різних етапах розробки програмного забезпечення. Він підходить для прототипування, тестування, відлагодження та підтримки програмного забезпечення.

5. Веб-розробка: Python використовується веб-розробниками для створення серверної частини веб-додатків. Фреймворки, такі як Django і Flask, допомагають в розробці веб-додатків з використанням Python.

6. Автоматизація задач: Python можна використовувати для написання скриптів, які автоматизують рутинні задачі. Це можуть бути скрипти для обробки файлів, автоматизації роботи з базами даних, взаємодії зі сторонніми сервісами та інші.

Python є потужним і гнучким інструментом, який знаходить застосування в багатьох сферах програмування і розвитку програмного забезпечення.

PHP - це мова кодування, широко використовувана і спосібна на інтеграцію з HTML, що робить її привабливою для тих, хто займається веб-розробкою. HTML може бути не зручним через його величезний розмір та множинність команд, в той час як PHP сторінки включають HTML з вбудованим кодом, що спрощує управління ними. PHP розшифровується як Hypertext Preprocessor.

Основне відмінність між PHP та JavaScript полягає в тому, що JavaScript виконується на клієнтській стороні (наприклад, в браузері), в той час як PHP працює на стороні сервера, створюючи HTML, який потім передається до клієнта.

Основні сфери, де використовується PHP:

1. Серверні сценарії – основне поле для PHP.
2. Сценарії командного рядка – відмінно підійде для сценаріїв, які були створені за допомогою cron чи планувальника завдань. Вони також відмінно працюють для простої текстової обробки.
3. Написання настільних додатків – PHP, мабуть, не найкраща мова для створення настільних додатків, але для досвідченого веб-розробника він надає набагато більше можливостей, ніж його конкуренти. PHP можна використовувати у всіх основних операційних системах (Linux, Windows та Mac OS X та інших). PHP також підтримує багато найбільш широко використовуваних серверів, це означає, що у вас буде розкіш, якщо ви вирішите використовувати його [12].

PHP має низку переваг, серед яких особливо привабливим для новачків є його легкість вивчення. Незалежно від цього, він також пропонує широкий спектр розширених функцій, які задовольнять вимоги навіть досвідчених професіоналів. Варто зауважити, що PHP може бути використаний для створення різноманітних елементів, починаючи від зображень і закінчуючи фільмами, а його можливості не обмежуються лише HTML.

Унікальні особливості PHP: Проста, звична та зручна у використанні: У народі відома своєю простотою, знайомістю та простотою вивчення мови, оскільки синтаксис подібний до синтаксису мови C або мови Паскаль.

PHP є мовою програмування, яка має деякі переваги:

1. Надійність та організованість: PHP є добре структурованою та логічною мовою, що дозволяє легко організувати програми та розробляти надійне програмне забезпечення.

2. Розширені функції: PHP має велику кількість заздалегідь визначених функцій, що дозволяє розробникам швидко впроваджувати динамічні веб-сайти та працювати з захищеними даними.

3. Простота використання: PHP дозволяє використовувати змінні без декларування типів даних, що робить програмування більш зручним та легким для початківців.

4. Гнучкість та інтегрованість: PHP може бути легко інтегрований з різними технологіями, такими як HTML, XML, JavaScript. Він працює на різних операційних системах та може бути використаний на різних пристроях. Також він добре інтегрується з базами даних та дозволяє створювати настільні програми.

5. Розширені можливості: PHP надає широкі можливості для розробки веб-додатків, автоматизації завдань та роботи з базами даних.

Загалом, PHP є потужним та гнучким інструментом для розробки програмного забезпечення, зокрема веб-додатків. Його простота використання та широкий функціонал роблять його популярним вибором серед розробників.

PHP є потужним інструментом для роботи з реляційними базами даних. Він підтримує різноманітні бази даних, такі як MySQL, SQLite, Oracle, Sybase, Informix та PostgreSQL, що дозволяє розробникам використовувати PHP для взаємодії з цими базами даних.

PHP надає вбудовані бібліотеки та функції для роботи з базами даних, що спрощує створення з'єднання, виконання запитів і обробку результатів. Розробники можуть використовувати PHP для створення динамічних веб-сторінок, які взаємодіють з базою даних, виконуючи додавання, оновлення, видалення та вибірку даних з таблиць.

Крім того, PHP є відкритим джерелом, що означає, що він доступний для використання безкоштовно, і має велику спільноту розробників, які активно публікують помилки, вносять внески до коду та розробляють фреймворки і інструменти, які полегшують роботу з PHP. Це знижує вартість розробки

програмного забезпечення і дозволяє розробникам ефективно використовувати PHP для різних проектів.

Багато фреймворків, таких як Codeigniter, Zend Framework, CakePHP, побудовані на основі PHP і надають розширені можливості та структуру для розробки веб-додатків. Ці фреймворки спрощують роботу з базами даних, реалізуючи шаблони проектування та надаючи готові рішення для типових задач.

Таким чином, PHP є потужним інструментом для роботи з базами даних і веб-розробки, завдяки своїм функціям, безкоштовності та великій спільноті розробників.

Java має кілька переваг і особливостей, які варто відзначити:

1. Портативність: Завдяки використанню віртуальної машини Java (JVM) та байт-коду, програми Java можуть бути виконані на будь-якій платформі, на якій є відповідна реалізація JVM. Це забезпечує високу переносимість програм та дозволяє їх виконання на різних операційних системах та апаратних платформах.

2. Безпека: Використання віртуальної машини дозволяє ізолювати виконання програми Java від хост-системи, що забезпечує безпеку від потенційно шкідливого коду. Крім того, Java має вбудовану систему керування пам'яттю, яка допомагає уникнути помилок пам’яті, таких як переповнення буфера або виток пам'яті.

3. Об'єктно-орієнтованість: Java підтримує об'єктно-орієнтовану парадигму програмування, що спрощує організацію та керування кодом. Вона дозволяє розробникам створювати класи, об'єкти та спадкування, що сприяє покращенню модульності та повторному використанню коду.

4. Динамічність: Java має можливості динамічного завантаження та модифікації коду під час виконання. Це дозволяє розробникам розширювати та оновлювати програми без їх повної перекомпіляції та перезавантаження.

5. Велика спільнота та розширюваність: Java має широку спільноту розробників, яка активно підтримує та розвиває цю мову. Існує велика кількість

сторонніх бібліотек та фреймворків, які дозволяють розширити функціональність та прискорити розробку програм на Java.

Хоча Java може бути трохи повільнішою у порівнянні з деякими іншими мовами програмування, вона забезпечує переваги у вигляді безпеки, портативності та розширюваності, що робить її популярним вибором для великих розподілених систем та мобільних додатків.

Основні риси мови програмування Java включають:

1. Портативність: Java працює на будь-якій платформі, оскільки код компілюється в байт-код, який виконується на віртуальній машині Java (JVM). Це означає, що програми Java можуть працювати на різних операційних системах, таких як Windows, macOS або Linux, без необхідності перекомпіляції.

2. Об'єктно-орієнтований підхід: Java побудована на основі об'єктно-орієнтованої парадигми програмування. Об'єкти в Java складаються з даних (полів або атрибутів) та методів (коду для роботи з даними). Об'єкти можуть бути розширені або успадковані, що сприяє повторному використанню коду та модульності програм.

3. Автоматичне управління пам'яттю: Java використовує механізм автоматичного збирання сміття, який відслідковує використання об'єктів та автоматично вивільняє пам'ять, коли об'єкт стає недоступним. Це спрощує роботу з пам'яттю та уникнення витоків пам'яті, що часто зустрічаються у мовах програмування, які вимагають ручного керування пам'яттю.

4. Безпека: Виконання програм Java відбувається всередині віртуальної машини, що надає додатковий рівень безпеки. Java використовує пісочницю (sandboxing) для обмеження доступу програм до системних ресурсів, що запобігає виконанню шкідливого коду та забезпечує безпеку виконання.

5. Простота вивчення: Синтаксис Java схожий на мови, такі як C++, що дозволяє програмістам з досвідом у C++ швидко освоїти Java. Це сприяє швидкому старту в розробці програм на Java та полегшує перехід між мовами.

Загалом, Java є потужною мовою програмування з великим спектром функцій, що робить її популярним вибором для розробки різноманітних додатків та систем.

Ruby - це повністю об'єктно-орієнтована мова програмування, яка була розроблена Юкіхіро Мацумото в середині 1990-х років в Японії. В Ruby все є об'єктом, за винятком блоків, але для них також є заміни. Розробка мови Ruby була спрямована на створення інтуїтивно зрозумілої мови програмування, яка полегшує взаємодію між програмістами та комп'ютерами. Синтаксис Ruby подібний до синтаксису інших популярних мов програмування, таких як C і Java, що дозволяє програмістам цих мов швидко освоїти Ruby. Мова підтримується на різних платформах, включаючи Windows, Mac і Linux.

Ruby базується на впливі багатьох інших мов програмування, включаючи Perl, Lisp, Smalltalk, Eiffel і Ada. Це інтерпретована мова, що означає, що програми Ruby виконуються безпосередньо інтерпретатором, без необхідності попереднього компілювання в машинний код. Розробники Ruby також мають доступ до потужного репозиторію RubyGems, який забезпечує стандартний формат для програм і бібліотек Ruby. RubyGems дозволяє легко встановлювати, керувати та використовувати сторонні програми і бібліотеки в Ruby [15].

Основні переваги Ruby полягають у наступному:

– Код, написаний на Ruby, відрізняється невеликим обсягом, елегантністю та потужністю, що дозволяє досягати високої продуктивності за допомогою меншої кількості коду.
– Розробка веб-додатків на Ruby відбувається швидко та просто, що сприяє зниженню робочого навантаження.
– Ruby є динамічною мовою програмування, що надає гнучкість та мінімальні обмеження стосовно вбудованих функцій, що робить її схожою до природної мови.

Недоліки Ruby:

– Ruby є новою мовою з унікальним синтаксисом, що може ускладнити початкове програмування на ній. Проте, з досвідом роботи з Ruby вона стає легшою в освоєнні.
– Редагування коду на Ruby може бути складним, оскільки багато коду виконується під час його виконання, що робить його менш зрозумілим для читачів.
– Порівняно з іншими мовами програмування, Ruby має менше інформаційних ресурсів доступних для вивчення.
– Ruby є інтерпретованою мовою сценаріїв, які зазвичай працюють повільніше, ніж компільовані мови, тому виконання Ruby-коду може бути повільнішим порівняно з іншими мовами.
– Ruby широко використовується для розробки різних типів веб-додатків, особливо за допомогою Ruby on Rails (RoR) - популярного веб-фреймворку, який допомагає прискорити процес розробки та зекономити час [16]

Рис. 2.1. Порівняльна інфографіка Ruby та PHP

Також була створена інфографіка, яка порівнює особливості двох мов програмування - Ruby та PHP (рис. 2.1).

2.2. Сутність, типологія та функціонал фреймворків

Фреймворк - це структура підтримки, яка визначається конкретними програмними артефактами або модулями і використовується для організації та розробки програмного забезпечення [16].

Іншими словами, фреймворк може бути розглянутий як шаблон, схема або концептуальна структура, заснована на технологіях, які полегшують розробку. Фреймворки допомагають уникнути можливих помилок програмування, спрощуючи процес роботи. Важливо розрізняти фреймворки від бібліотек. Бібліотека - це компонент архітектури програмного забезпечення, що надає певну функціональність. Вона може бути використана як набір підсистем з близькою функціональністю, не впливаючи на архітектуру основного програмного продукту і не накладаючи на неї обмежень.

Фреймворк не лише надає розробнику необхідний функціонал, але також встановлює правила для архітектури додатку, визначаючи поведінку за умовчанням на початковому етапі розробки. Він формує каркас, який можна розширювати і змінювати відповідно до вимог проекту. Крім того, фреймворк може включати допоміжні програми, бібліотеки коду, мову сценаріїв та інше програмне забезпечення, яке спрощує розробку та об'єднання різних компонентів у великому програмному проекті.

Фреймворк - це структура підтримки, яка надає основу для організації та розробки програмного забезпечення. Він включає концептуальну та технологічну структуру, а також конкретні програмні артефакти або модулі. Фреймворк допомагає спростити роботу програмістам та уникнути помилок програмування шляхом надання шаблонів, схем або концепційної структури.

У порівнянні з бібліотеками, які є простішими компонентами програмного забезпечення, фреймворки включають не лише функціональні можливості, але й встановлюють правила для побудови архітектури додатку. Вони визначають

поведінку за замовчуванням і надають каркас, який можна розширювати і змінювати залежно від вимог. Фреймворк також може включати додаткові інструменти, бібліотеки коду, мови сценаріїв та інше програмне забезпечення, що сприяє полегшенню розробки і об'єднанню різних компонентів великого програмного проекту [16]

Фреймворки можуть виконувати дві основні функції в залежності від їх типу: серверну частину, призначену для роботи на стороні сервера, або клієнтську частину, призначену для роботи з інтерфейсом.

Фреймворк - це набір інструментів і модулів, призначених для повторного використання в різних проектах, зокрема у розробці веб-додатків. Використання фреймворку сприяє організації роботи та зменшенню ризику помилок. Він пропонує такі важливі переваги:

1. Економія часу: фреймворк вже має побудовану структуру, що дозволяє значно зекономити час, який інакше був би витрачений на створення власного робочого простору.

2. Співпраця: стандартизація коду спрощує співпрацю між розробниками, дозволяючи їм працювати разом.

3. Спеціальні інструменти: фреймворк надає спеціально розроблені інструменти та утиліти, які можна використовувати в будь-який момент.

4. Уникнення повторення коду: використовуючи фреймворк, не потрібно програмувати загальні частини, які вже є необхідними для роботи, що дозволяє ефективніше програмувати програму.

Рис. 2.2. Приклади найпопулярніших фреймворків

Таким чином, використання фреймворків допомагає зберегти час, полегшує співпрацю, надає спеціальні інструменти та уникнення повторювання коду. [16]

Фреймворки можна класифікувати за їхнім призначенням:

- Веб-програмування: ці фреймворки призначені для розробки веб-додатків, включаючи веб-дизайн сторінок та веб-сервісів.
- Програмні фреймворки: ці фреймворки дозволяють розширити структуру програм для операційних систем.
- AJAX: ці фреймворки спрощують розробку додатків з використанням технології AJAX, яка дозволяє здійснювати запити на сервер без перезавантаження сторінки.
- Фреймворки управління вмістом (CMF): ці фреймворки полегшують програмування систем управління вмістом (CMS), таких як WordPress, Joomla тощо.
- Мультимедійні фреймворки: ці фреймворки допомагають розробникам, що працюють з відео, аудіо та зображеннями, а також у створенні

мультимедійних додатків загалом, наприклад, відеоконференцій або медіаконвертерів.

Кожен тип фреймворку має свої особливості та інструменти, які сприяють розробці додатків у відповідній області.

Фронтенд-фреймворки, також відомі як фреймворки для розробників інтерфейсів, спрощують створення веб-інтерфейсів, таких як веб-сторінки або додатки. Вони надають засоби для розробки клієнтської частини програми і використовують різноманітні технології, включаючи JavaScript, HTML та CSS. Деякі популярні фронтенд-фреймворки включають AngularJS, ReactJS, Vue.js, Bootstrap, Material Design та багато інших.

Фреймворки для бекенд-розробників, або серверні фреймворки, спрощують роботу з серверними мовами програмування, такими як Python, Ruby, PHP, Node.js, Java і т. д. Вони надають інструменти для керування базами даних, обробки даних, роботи з файлами cookie та сеансами, а також інші функціональні можливості, необхідні для створення серверної частини програми. Деякі популярні серверні фреймворки включають Django (Python), Ruby on Rails (Ruby), Laravel (PHP) і т. д.

Ці два типи фреймворків взаємодіють між собою, допомагаючи розробникам створювати повноцінні веб-додатки з відповідним інтерфейсом та серверною логікою.

Клієнтські фреймворки, або фреймворки інтерфейсу, використовуються для розробки зручного та функціонального користувацького інтерфейсу веб-додатків. Вони працюють всередині браузера і не мають прямого зв'язку з бізнес-логікою.

Фреймворки Python, у свою чергу, забезпечують розробникам зручне середовище для роботи з цією мовою програмування. Вони включають набір інструментів, модулів та бібліотек, які допомагають виконувати конкретні завдання. Це дозволяє економити час і уникати повторення рутинних дій, замість

цього використовуючи готові інструменти для вирішення відповідних задач. Фреймворки спрощують роботу і дозволяють сфокусуватися на логіці проєкту.

У фреймворків для Python є свої особливості та призначення, і найпопулярніші з них пропонують різні можливості. Деякі фреймворки призначені для складної багаторівневої розробки, забезпечуючи потужні інструменти для цього процесу. Інші фреймворки, натомість, пропонують спрощений інтерфейс, який дозволяє швидко створювати окремі модулі та блоки. Вибір конкретного фреймворку залежить від поставлених завдань та цілей проєкту, що допомагає забезпечити оптимальну реалізацію проекту.

Використання фреймворків дозволяє автоматизувати процеси, що прискорює роботу і зменшує кількість рутинних завдань. Вони надають розробникам розширені можливості для створення ефективних програм або веб-сайтів.

CherryPy є мікро-фреймворком Python, спрямованим на розв'язання конкретних задач і може запускати програми на будь-якій операційній системі. Він має дружнє та зрозуміле середовище розробки, а також є функціональним та повноцінним фреймворком, на якому можна створювати непогані програми. CherryPy є відкритим джерелом, що робить його безкоштовним для розробників, і має активну спільноту, яка завжди готова допомогти з вирішенням завдань.

Недоліками CherryPy є його обмежені можливості для виконання складних завдань, він більше призначений для конкретних рішень, таких як розробка плагінів або модулів. Також можливості "з коробки" у CherryPy не настільки широкі.

CherryPy є відмінним вибором, коли необхідно створити програму з невеликою кількістю коду, простим функціоналом та здатністю виконувати лише одну або кілька задач. Також він надає можливість керувати декількома

серверами одночасно, відстежувати продуктивність додатків і навіть розробляти під Android.

CherryPy є частиною Python фреймворків, які призначені для розв'язання конкретних завдань, і він ідеально підходить для розробки під Android.

Pyramid є популярним фреймворком Python, призначеним для розробки складних об'єктів і вирішення багатофункціональних завдань. Цей фреймворк широко використовується професійними програмістами і добре підходить для ідентифікації та маршрутизації. Pyramid орієнтований на широку аудиторію і ідеально підходить для розробки прототипів API.

Pyramid є відмінним інструментом для швидкої розробки базових застосунків. Цей фреймворк володіє гнучкістю та легкістю у вивченні. Його успіх ґрунтується на фундаментальних принципах та використанні простих та базових технік програмування. Pyramid має мінімалістичний підхід, проте надає користувачам велику свободу дій. Він здатний ефективно працювати як з невеликими програмами, так і з потужними функціональними програмами. Відхилитися від основних принципів важко, оскільки цей фреймворк Python вирішує багато речей за вас. Легкі програми можна реалізувати просто, але для складних і масштабних проектів потрібно повністю освоїти середовище та дотримуватися його принципів.

Фреймворк Pyramid використовується в таких випадках:

1. Якщо потрібні інструменти для виявлення проблем та внесення змін і коректив у програму своєчасно.

2. Якщо вам необхідно використовувати кілька мов програмування одночасно.

3. Для звітності та фінансових розрахунків, прогнозування.

4. Для отримання стабільних автоматичних звітів про роботу.

5. Якщо потрібно швидко створити простий застосунок.

За допомогою Python Web Framework Pyramid також можна створювати складні програми з великим функціоналом.

Фреймворк Grok є розширенням фреймворка Zope3 і працює з шаблонами. Основна його мета - уникнути повторення коду шляхом використання шаблонів, які можуть бути використані повторно. Це спрощує та прискорює розробку.

Grok призначений для створення невеликих програм з кількома простими завданнями. Він має спрощену структуру та простий процес установки модулів, що розширює його можливості та гнучкість. Однак, він не підходить для складних завдань, оскільки має обмежені можливості.

Спільнота Grok не є дуже великою через його обмежену популярність. Проте, він все ще використовується програмістами Python, які хочуть комфортно розробляти. Grok надає простоту розробки та можливість швидко створити простий застосунок.

Grok підходить для розробників у таких випадках:

- Для програмістів з малою кількістю досвіду, яким зручніше використовувати готові компоненти замість розробки власних.
- Коли потрібно швидко створити простий застосунок.
- Коли функціонал програми простий і інтерфейс не є ключовим аспектом.

Незважаючи на свою простоту, Grok входить до списку кращих Python Web Frameworks. Його зрозумілість та широкий набір можливостей роблять його ідеальним для комфортного розробки.

TurboGears - це відкритий фреймворк для розробки веб-застосунків з широкими можливостями. Він надає зручні шаблони, гнучку ORM та підтримку

різних баз даних. TurboGears є універсальним середовищем розробки, яке підходить як для новачків, так і для досвідчених програмістів.

Особливості TurboGears включають елементи командного рядка, підтримку різних баз даних та можливість роботи з різними сервісами. Він допомагає прискорити розробку, скорочує кількість коду та зменшує ймовірність помилок. Також TurboGears має нову систему віджетів, що спрощує створення потужних застосунків.

Хоча на початку свого існування TurboGears отримав кілька негативних відгуків через швидкий старт розробки, багато з цих проблем було виправлено, і фреймворк зараз працює добре.

TurboGears застосовується часто, особливо в таких випадках:

- Потрібно створити потужний "важкий" застосунок протягом короткого часу.
- Необхідна гнучкість та підтримка різних баз даних.
- Потрібна валідація з Formencode.
- Потрібні інструменти командного рядка.

2.2.1. Особливості використання веб-фреймворку Yii та Django Framework

Веб-фреймворк - це інструмент, який допомагає створювати та запускати веб-додатки. Використання веб-фреймворку полегшує роботу розробників, оскільки він надає основні шаблони і структуру для створення надійних та масштабованих веб-додатків.

Веб-фреймворки спрощують розробку веб-програм шляхом надання готових компонентів, які можна повторно використовувати. Вони забезпечують базовий функціонал, такий як маршрутизація, керування HTTP-запитами, обробка форм та доступ до бази даних.

Відмінності між веб-фреймворком і веб-сервером полягають у їхніх функціональних можливостях. Веб-сервер запускає веб-додаток та обробляє

HTTP-запити, тоді як веб-фреймворк надає інструменти та структуру для створення самого веб-додатка.

Застосування веб-фреймворку значно прискорює розробку веб-додатків, спрощує управління кодом та дозволяє розробникам фокусуватися на бізнес-логіці та функціональності додатку. Веб-фреймворки є незамінним інструментом для веб-розробки і дозволяють створювати високоякісні та ефективні веб-додатки.

Загальні функції веб-фреймворків включають:

1. Маршрутизацію URL-адрес: Фреймворки надають механізми для визначення маршрутів URL-адрес, які вказують, які дії або обробники викликати для кожного запиту.

2. Управління та перевірка форм вводу: Фреймворки допомагають валідувати та обробляти дані, введені користувачами через веб-форми.

3. Конфігурація підключення до бази даних та ORM: Фреймворки дозволяють налаштовувати з'єднання з базою даних та використовувати об'єктно-реляційні картографування (ORM) для спрощення роботи з даними.

4. Сховище сесій та пошук: Фреймворки забезпечують механізми для зберігання та управління сесіями користувачів та підтримки пошуку в додатках.

5. Механізм шаблонування: Фреймворки надають інструменти для створення та керування різними типами шаблонів, такими як HTML, XML, JSON, що спрощує генерацію відповідей.

6. Веб-безпека: Фреймворки надають захист від різних видів атак, таких як міжсайтова підробка (CSRF), внедрення SQL-запитів (SQL Injection), міжсайтові сценарії (XSS) та інші шкідливі дії.

Деякі приклади популярних веб-фреймворків включають:

1. Django: Фреймворк Python для швидкої розробки веб-додатків з потужним функціоналом та широкою спільнотою розробників.

2. Ruby on Rails: Фреймворк Ruby, що спрощує розробку веб-додатків шляхом використання конвенцій над конфігурацією.

3. Laravel: Фреймворк PHP з елегантним синтаксисом та багатим набором інструментів для розробки сучасних веб-додатків.

4. Yii: Високопродуктивний PHP-фреймворк з широкими можливостями розширення та швидкими шаблонами для розробки веб-додатків.

5. Spring: Фреймворк Java, який надає широкий спектр інструментів для розробки корпоративних веб-додатків.

Ці фреймворки пропонують потужність та функціональність для розробки різноманітних веб-додатків та мають велику спільноту розробників, що надає підтримку та ресурси для розробників.

Рис. 2.3. Найпопулярніші приклади PHP веб-фреймворків

Дані SimilarTech про впровадження фреймворків в 2019 році показують, що Ruby on Rails був використаний для найбільшої кількості сайтів порівняно з іншими фреймворками. Django також займає важливе місце, наближаючись до популярності Ruby on Rails. Laravel, хоча ще не досяг рівня популярності Ruby on Rails і Django, виявляє ознаки зростання популярності.

Важливо враховувати, що дані від SimilarTech відображають статистику за певний період часу і можуть бути змінені або оновлені з плином часу. Крім того, популярність фреймворків може варіюватись в залежності від конкретного регіону або галузі.

Рис. 2.4. Статистика від SimilarTech щодо поширеності фреймворків

Використання веб-фреймворків може значно полегшити та прискорити процес розробки веб-додатків та веб-сервісів. Фреймворки надають розробникам готовий функціонал і шаблони для створення надійних та масштабованих додатків, а також забезпечують доступ до різних бібліотек та інструментів, які спрощують роботу з кодом.

Рекомендується вибирати фреймворк в залежності від ваших потреб та особливостей проекту. Популярні та успішні фреймворки, такі як Ruby on Rails, Django та інші, були розроблені з метою забезпечити потужний функціонал та спростити роботу розробників. Проте, перед використанням веб-фреймворка варто врахувати кілька речей, як-то сумісність з мовою програмування, гнучкість та розширюваність фреймворка, наявність необхідного функціоналу та підтримка спільноти розробників.

В цілому, веб-фреймворки є корисними інструментами для розробки веб-додатків, які спрощують роботу розробників і допомагають створювати ефективний та надійний код.

При початку роботи з веб-фреймворком необхідно враховувати кілька аспектів:

– Взаємодія з базами даних: Багато веб-фреймворків мають вбудовану підтримку для взаємодії з базами даних, що спрощує роботу з ними в контексті веб-додатків.
– Генерація HTML: Фреймворки часто надають можливості для створення HTML-коду з використанням шаблонів, що спрощує створення стандартизованих та простих у використанні веб-сторінок.
– Маршрутизація: Більшість фреймворків мають вбудовані механізми для обробки маршрутизації URL-адрес, які забезпечують відповідність запитів користувачів до відповідних функцій або контролерів.

Використання веб-фреймворків сприяє вирішенню різноманітних завдань, починаючи від створення шаблонів і роботи з базами даних до управління сеансами і повторного використання коду. Вони представляють собою потужний інструмент для розробки динамічних веб-сайтів, а також можуть бути використані для створення статичних веб-сайтів.

Yii - це один з веб-фреймворків, який відзначається своєю високою ефективністю та має компонентну структуру. Цей PHP-фреймворк призначений спеціально для швидкої розробки об'ємних веб-додатків.

Рис. 2.5. Логотип фреймворку Yii

Yii - це фреймворк PHP, який дозволяє ефективно розробляти веб-додатки і максимально використовувати концепцію повторного використання коду. Для запуску додатків на Yii потрібен веб-сервер з підтримкою PHP версії 5.1.0 або

вище. Розуміння об'єктно-орієнтованого програмування є корисним для розробників, оскільки Yii є строго об'єктно-орієнтованим фреймворком.

Yii має численні переваги, зокрема високу ефективність, широкий функціонал і добре структуровану документацію. Цей фреймворк був ретельно розроблений з урахуванням вимог, які виникають при створенні складних веб-додатків. Yii не є побічним продуктом або надбудовою над іншим проектом, а є результатом досвіду розробників у галузі веб-додатків та їх досліджень найпопулярніших фреймворків та додатків.

Основні переваги і можливості фреймворка Yii включають наступне:

1. Висока продуктивність порівняно з іншими PHP-фреймворками.
2. Базується на парадигмі MVC (Модель-Представлення-Контролер).
3. Надає інтерфейси DAO і ActiveRecord для роботи з базами даних з використанням PDO, а також підтримку інтернаціоналізації.
4. Можливість кешування сторінок і окремих фрагментів.
5. Забезпечує перехоплення та обробку помилок.
6. Має функціонал для роботи з формами, включаючи побудову та валідацію.
7. Реалізована система аутентифікації і авторизації.
8. Зручний для реалізації AJAX-інтерфейсів і має інтеграцію з jQuery.
9. Включає вбудовані генератори базового PHP-коду для CRUD-операцій (скаффолдинг).
10. Підтримує роботу з темами оформлення.
11. Дозволяє підключати сторонні бібліотеки.
12. Надає зручні засоби для роботи з міграціями баз даних, включаючи генерацію, застосування та відкат.
13. Підтримує автоматичне тестування і розробку в стилі TDD.
14. Має вбудовану підтримку стилю REST.

Ці можливості дозволяють розробникам зручно та ефективно створювати веб-додатки з використанням фреймворка Yii.

Yii використовує шаблон проектування Модель-Представлення-Контролер (MVC) для організації веб-додатків. Цей шаблон дозволяє розподілити бізнес-логіку та користувацький інтерфейс на окремі компоненти, що дозволяє розробникам змінювати частини програми незалежно одна від одної.

У шаблоні MVC модель відповідає за управління даними і бізнес-логікою, уявлення забезпечує відображення інформації, а контролер взаємодіє з моделлю та уявленням, обробляючи вхідні дані від користувача і забезпечуючи передачу відповідних команд та даних.

Yii також використовує фронт-контролер, який відомий як додаток (application) і він відповідає за обробку запитів та створення контексту обробки. Додаток збирає інформацію про запит і передає її відповідному контролеру для подальшої обробки.

Застосування шаблону MVC разом із фронт-контролером у фреймворку Yii дозволяє організувати розробку веб-додатків, розділивши логіку, відображення та обробку запитів на окремі компоненти. Це полегшує розробку, підтримку та розширення веб-додатків з використанням Yii.

Додатки Yii складаються з різних сутностей, які співпрацюють між собою для обробки запитів і забезпечення функціональності. Основні сутності включають:

1. Вхідні скрипти: Це PHP скрипти, які приймають вхідний запит від користувача і розпочинають обробку. Вони відповідають за запуск і направлення запиту до відповідного додатку Yii.

2. Додатки: Це глобально доступні об'єкти, які керують життєвим циклом додатків Yii. Вони здійснюють координацію роботи різних компонентів і забезпечують правильну обробку запитів.

3. Компоненти програми: Це об'єкти, які зареєстровані в додатку і надають різні функціональні можливості для обробки запиту. Наприклад, компоненти можуть включати об'єкти для роботи з базами даних, кешування, обробки форм і т. д.

4. Модулі: Це самодостатні пакети, які містять повний набір компонентів для реалізації архітектури MVC. Додаток може складатись з декількох модулів, які надають окремі функціональні блоки.

5. Фільтри: Це код, який виконується до і після обробки запиту контролерами. Вони дозволяють здійснювати додаткові операції, такі як авторизація, перевірка даних або обробка помилок.

6. Віджети: Це об'єкти, які можуть включати в себе інші об'єкти і представляють собою невеликі візуальні компоненти. Вони використовуються для повторного використання та спрощення створення складних елементів інтерфейсу.

Ці сутності співпрацюють, виконуючи свої функції, і дозволяють розробникам ефективно створювати веб-додатки з використанням Yii.

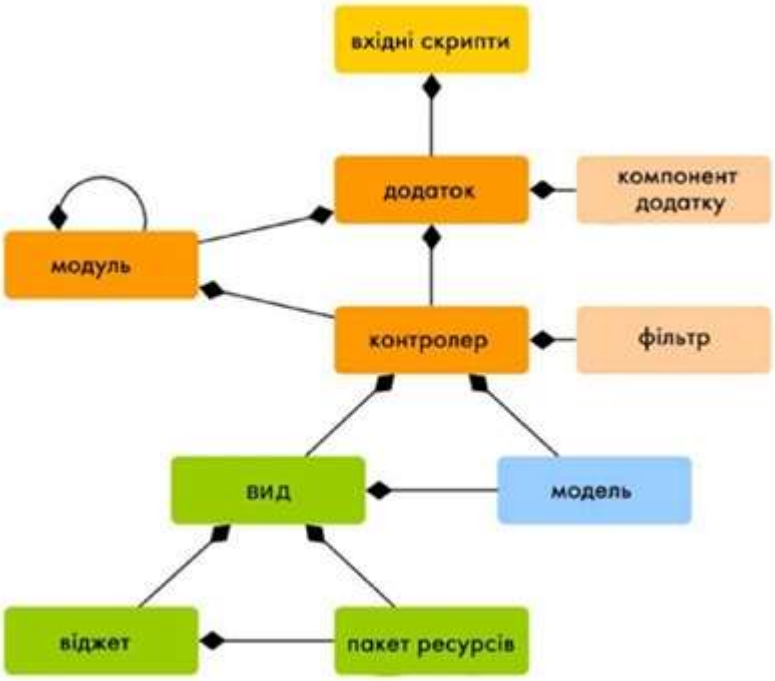


Рис. 2.6. Структура фреймворку Yii

Yii рекомендує встановлювати окремі параметри конфігурації для кожного члена команди і не фіксувати їх у сховищі версій, оскільки ці дані можуть бути конфіденційними, такими як паролі бази даних та ключі API. Крім того, фреймворк розуміє, що існують різні серверні середовища (розробка, тестування,

виробництво) і рекомендує мати окремі конфігураційні файли для кожного середовища.

Yii пропонує просту концепцію середовища, де кожне середовище представлене набором файлів у каталозі "environments". Команда Yii може використовуватись для перемикання між цими середовищами. При перемиканні Yii копіює файли конфігурації з каталогу середовища в кореневий каталог, де розташовані всі програми, що дозволяє використовувати відповідні параметри конфігурації для кожного середовища без необхідності повторного визначення їх кожного разу.

Django - це full-stack фреймворк з відкритим вихідним кодом для розробки веб-додатків на Python. Він дозволяє розробникам швидко переходити від прототипування до готового рішення, автоматизуючи процеси та прискорюючи роботу за допомогою вбудованих функцій та бібліотек. Django має багато готових рішень, таких як панель адміністратора, міграції баз даних, різноманітні форми і інструменти для аутентифікації користувачів, що полегшує розробку. Він має зрозумілу та просту структуру, високий рівень безпеки і зручну роботу з базами даних за допомогою ORM. Django реалізує патерн Model-View-Controller (MVC), який в поточній версії фреймворку відомий як Model-View-Template (MVT). Ця архітектура дозволяє розробникам працювати з візуальним представленням та бізнес-логікою програми окремо. Компоненти MVT можна використовувати незалежно один від одного.

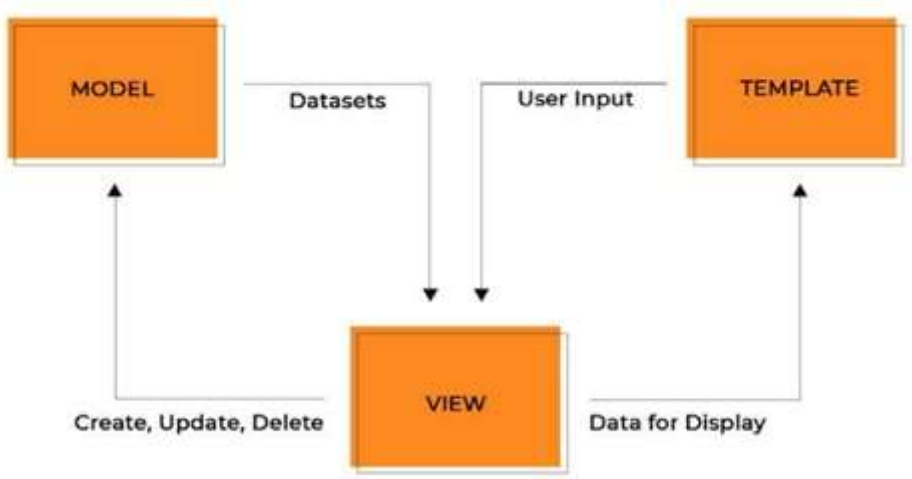


Рис. 2.7. Схема архітектури MVC в Django

Django, як фреймворк повного стеку, використовує модель-представлення-шаблон (MVT) для розділення бізнес-логіки, представлення і роботи з даними. Модель (model) відповідає за дані і містить поля та поведінку, які відображаються у таблицях бази даних. Django підтримує різні системи управління базами даних, такі як PostgreSQL, MySQL, SQLite та Oracle. Модель також дозволяє розробникам створювати, зчитувати, оновлювати та видаляти об'єкти в базі даних.

HTTP-запити обробляються модулем "Подання" (view), який виконує бізнес-логіку, методи та властивості, і надсилає HTTP-відповіді на запити. Подання отримує дані від моделі і передає їх до шаблонів для відображення або передбачуваної обробки перед передачею до шаблонів.

Шаблони (templates) використовуються для відображення даних в HTML-коді. Вони можуть бути статичними або динамічними. Шаблони не містять бізнес-логіки і служать лише для відображення даних.

Незважаючи на широкі можливості, Django має деякі недоліки. Він є потужним та монолітним, що призводить до повільного розвитку фреймворку. Хоча він надає безліч універсальних модулів, швидкість розробки самого Django може бути уповільненою.

Django є ідеальним вибором у випадках, коли потрібно:

– Розробляти серверну частину API.

– Створювати веб-застосунки.
– Часто вносити зміни та розгортати застосунок.
– Масштабувати програму як у бік збільшення, так і зменшення.
– Розв'язувати складні завдання, в яких потрібна підтримка спільноти.
– Використовувати ORM для роботи з базою даних.
– Інтегрувати нові технології, такі як машинне навчання.

Django, як один з найпопулярніших веб-фреймворків Python, активно використовується мільйонами розробників і готовий впоратися з будь-якими завданнями.

Переваги фреймворка Django:

1. Розвинена екосистема: Django має широкий набір модулів і компонентів, що забезпечують структуру проекту та різноманітні можливості.
2. Зрілість: Django існує протягом багатьох років, постійно вдосконалюється та розвивається, а також має велику активну спільноту розробників.
3. Адміністративна панель: Django автоматично генерує адміністративну панель, що полегшує роботу з управлінням даними в програмі. Крім того, можна використовувати сторонні додатки для розширення функціональності панелі.
4. SEO-дружність: Django генерує SEO-дружні URL та надає зручні інструменти для реалізації пошукової оптимізації.
5. Розширюваність: Django дозволяє розширювати свою функціональність за допомогою плагінів, що дозволяє швидко додавати нові функції до проекту.
6. Бібліотека: Django має велику бібліотеку, яка допомагає вирішувати різноманітні завдання та спрощує розробку додатків.

Django - це потужний фреймворк, який має багато переваг і є популярним серед розробників. Однією з його переваг є підтримка використання багатьох бібліотек при розробці веб-додатків. Деякі з популярних бібліотек включають:

– Django Rest Framework, що полегшує роботу з API.

– Django CMS-зручний інструмент для управління контентом.
– Django-allauth - з його допомогою реалізуються функції реєстрації, авторизації, управління обліковими записами.

ORM (Object-Relational Mapping) є важливою складовою Django. Він забезпечує зручну і автоматичну взаємодію між програмою і базами даних, такими як PostgreSQL, MySQL і SQLite.

ORM у Django перетворює дані з таблиць бази даних в об'єкти, з якими розробник може працювати у своєму коді. Це полегшує розробку, оскільки розробникам не потрібно писати складний SQL-код або безпосередньо працювати з базою даних.

ORM також дозволяє легко змінювати базу даних, не змінюючи багато коду. Розробники можуть легко перемикатися між різними базами даних, просто змінюючи конфігурацію.

Загалом, ORM у Django спрощує взаємодію з базами даних, прискорює розробку та робить код більш читабельним та зрозумілим.

2.2.2. Особливості використання веб-фреймворку Laravel

Laravel - це один із популярних фреймворків для розробки веб-додатків на PHP. Він використовує архітектурний шаблон MVC (Model-View-Controller), що спрощує розподіл бізнес-логіки, управління даними та користувацьким інтерфейсом у програмі.

Один з головних переваг Laravel - це чистий та елегантний синтаксис, який дозволяє швидко розробляти веб-додатки. Він надає розробникам потужні інструменти та функції, які полегшують процес розробки і забезпечують продуктивність.

Деякі функціональні можливості Laravel включають:

– Маршрутизацію та обробку URL-адрес.
– Можливість використовувати шаблони для генерації HTML-коду.

– Багатофункціональний ORM для роботи з базами даних.
– Підтримку сеансів і аутентифікації користувачів.
– Інструменти для валідації даних та обробки форм.
– Вбудований інструмент для роботи з запитами AJAX.

Широку підтримку створення API.

Також Laravel має велику активну спільноту розробників, яка надає підтримку, документацію та сторонні пакети для розширення функціональності фреймворку.

Загалом, Laravel є потужним та популярним фреймворком, який допомагає розробникам ефективно створювати веб-додатки з використанням архітектурного шаблону MVC та елегантного синтаксису.

Рис. 2.8. Логотип останньої версії Laravel

Laravel - це PHP-фреймворк, який був випущений 9 червня 2011 року під ліцензією MIT. Вихідний код Laravel розміщений на GitHub, що сприяє його активному розвитку та співпраці з розробниками.

Основні переваги та можливості Laravel включають:

1. Підтримка архітектурного шаблону MVC: Laravel дозволяє розробникам розділити бізнес-логіку, презентацію та управління даними. Це полегшує організацію та підтримку проектів, забезпечуючи чистоту коду та покращену продуктивність.

2. Краща структура проекту: Laravel надає полегшені шаблони для створення красивих макетів, що дозволяє динамічно виводити контент. Це полегшує розробку веб-сторінок та додає більшу гнучкість при роботі з дизайном.

3. Багатий функціонал: Laravel має широкий набір вбудованих функцій, які полегшують розробку. Він надає готові рішення для аутентифікації, маршрутизації, кешування, сесій, роботи з базами даних та багато іншого. Це

дозволяє розробникам ефективно використовувати функціонал, що вже вбудований у фреймворк.

4. Велика активна спільнота: Laravel має велику спільноту розробників, яка надає підтримку, документацію та розширення. Це дозволяє швидко отримати допомогу, знайти відповіді на питання та використовувати сторонні пакети, які розширюють функціональність Laravel.

5. Популярність: Laravel є одним з найпопулярніших PHP-фреймворків разом з Symfony, Nette, CodeIgniter та Yii2. Це свідчить про його широке використання та високу оцінку серед розробників та бізнесу.

В цілому, Laravel є потужним та популярним PHP-фреймворком з відкритим кодом, який допомагає розробникам швидко та ефективно створювати веб-додатки з використанням архітектурного шаблону MVC та зручного синтаксису.

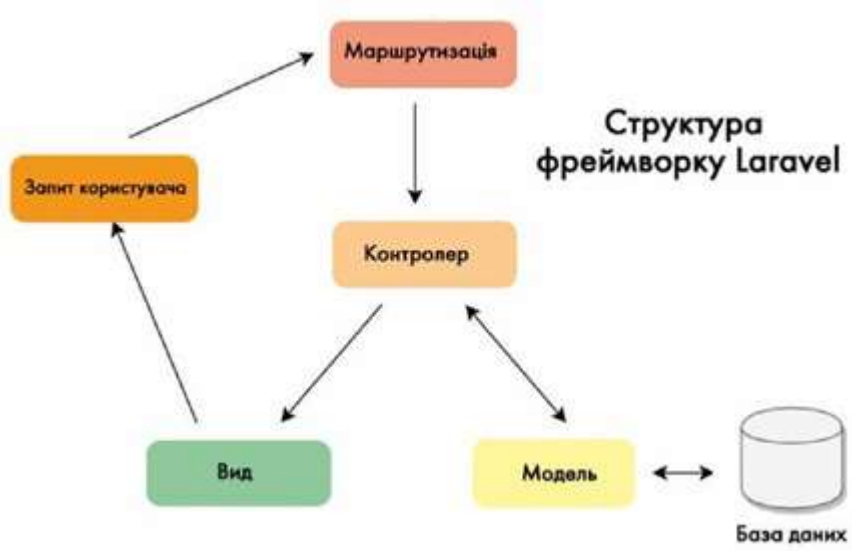


Рис. 2.9. Структура фреймворку Laravel

Laravel забезпечує розширення, які допомагають розробнику змінювати та настроювати макет веб-сторінки без зайвих зусиль. Це включає віджети з CSS та JS-кодом, які мають стабільну структуру. Шаблони Laravel інноваційно розроблені для створення простого та виразного макета з розділами.

Laravel також надає потужне об'єктно-реляційне відображення (ORM), яке спрощує роботу з базою даних. ORM у Laravel має просту реалізацію активного запису PHP та порівняно швидкий швидше, ніж в інших PHP-фреймворках.

Одна з головних переваг Laravel полягає в його простоті в використанні, що робить його підходящим для будь-яких веб-проектів. Це призводить до того, що близько 20% користувачів Laravel використовують його для розробки бізнес- та галузевих веб-сайтів, згідно з даними від SimilarTech. Laravel дозволяє розробникам виконувати запити до бази даних за допомогою синтаксису PHP, замість того, щоб писати SQL-код. Кожна таблиця бази даних має відповідну модель, що спрощує взаємодію з даними.

Щодо безпеки, Laravel має надійний механізм для захисту веб-додатків. Він використовує хешування та солі для захисту паролів у базі даних. Laravel також має механізми, які запобігають атакам ін'єкції SQL та забезпечують високий рівень безпеки вашого додатку.

Фреймворк Laravel має інструмент командного рядка під назвою Artisan, який допомагає автоматизувати рутинні завдання розробки. Artisan дозволяє створювати власні команди та виконувати різні корисні операції.

Laravel також постачається з об'єктно-орієнтованими бібліотеками, включаючи попередньо встановлені бібліотеки, які немають аналогів в інших фреймворках. Ці бібліотеки включають бібліотеку автентифікації, яка має багато популярних функцій, таких як хешування паролів Bcrypt, захист CSRF, шифрування та скидання пароля. Laravel розділений на модулі згідно з сучасними принципами PHP, що дозволяє розробникам створювати модульні та гнучкі веб-додатки.

Laravel є одним з найкращих фреймворків для обробки помилок. Він інтегрується з бібліотекою реєстрації Monolog та підтримує декілька потужних обробників журналів. Фреймворк надає зручні механізми для виявлення та обробки помилок, зокрема при обробці форм введення даних, щоб інформувати користувачів про проблеми.

Laravel надає можливості інтеграції інструментів для підвищення продуктивності веб-додатків. Один з таких інструментів - кешування бекенда, що допомагає прискорити завантаження веб-сторінок. Інтеграція інструментів кешування сприяє поліпшенню продуктивності сайту, що має важливе значення для користувацького досвіду, конверсії та SEO.

Система міграцій Laravel дозволяє розширювати структуру бази даних веб-додатка без необхідності повторного створення бази даних та ризику втрати даних. Використання міграцій спрощує зміну структури бази даних та надає зручний і безпечний спосіб модифікації бази даних за допомогою PHP-коду замість SQL. Laravel Schema Builder допомагає створювати таблиці бази даних та вставляти індекси.

Laravel також відомий своїм потужним модульним тестуванням, що допомагає розробникам перевіряти функціональність веб-додатків і уникати появи помилок. За допомогою PHPUnit і файла phpunit.xml, Laravel забезпечує підтримку модульного тестування, яке дозволяє виявляти та усувати помилки перед запуском веб-додатку.

Laravel прагне до стабільних випусків та надійності, і його розробники можуть легко виконувати модульні тести для свого коду, щоб перевірити правильність роботи додатка.

Автоматизоване тестування допомагає розробникам забезпечити більш точну та швидку перевірку програмного забезпечення, зменшити трудомісткість та забезпечити виявлення помилок, збоїв та неполадок. Це дозволяє виявляти проблеми на ранніх етапах розробки і виправляти їх перед релізом програми.

Перевірка та тестування є невід'ємною частиною процесу веб-розробки і дозволяють переконатися, що програмне забезпечення працює належним чином, забезпечує очікувану функціональність та надійність.

Сторінка помилки в веб-додатку може вплинути на користувацький досвід і спричинити втрату клієнтів. Тому важливо виявляти та вирішувати помилки, щоб уникнути порожніх сторінок помилок та забезпечити надійну та зручну роботу веб-додатка для користувачів.

2.2.3. Особливості використання веб-фреймворку Spring

Додатки Java можуть мати різноманітний характер, від аплетів до корпоративних серверних додатків зі складною структурою. Вони зазвичай складаються з набору об'єктів, які взаємодіють між собою для створення програми. Це означає, що об'єкти в додатку мають взаємні залежності.

Рис. 2.10. Логотип фреймворку Spring

Хоча Java надає багато можливостей для розробки додатків, вона не має вбудованих засобів для організації цих додатків у цілісне ціле. Архітекторам та розробникам доводиться самостійно організовувати будівельні блоки і використовувати шаблони дизайну, такі як Factory, Abstract Factory, Builder, Decorator та Service Locator, для створення різних класів та об'єктів.

Таблиця 2.1 - Порівняльна характеристика Spring та Yii

Основа порівняння	Spring	Yii
Тестування	Тут тестування повинно бути нестандартним з такими компонентами Synifony. як HttepKemel. DomCrawler тошо.	Тут тестування вимагає лише PHPUnit та Codeception.
Вимоги	ШЕ NetBeans 7.2,7.3,7.4,80, JavaEE Комплект для розробника на мові Java (JDK) версія 7 або 8	PHP 5.4 або новппа версія
Кешування	Він мае таи компоненти для кешування: HasMap, ConcurentMap, База даних, Файл	Він мае таи компоненти лля кешування: APC, База даних, Файл, Memcached, WinCache, XCache
Об’єктне реляпійне відображення ORM	Екземпляри JDBC DataSource	Об’єкти доступ}' до даних (DAO) Шаблон активного запису (AcnveRecord) та доктрина 2 через плагінн
Міграція	Дія міграпії потрібен клас мпрашї та інструмент заповнення даних.	Для цього потрібен лише клас міграпії.

Кінець таблиці 2.1

Однак, Spring Framework вирішує цю проблему шляхом надання формалізованих засобів для складання різноманітних компонентів у готовий до

використання додаток. Він кодифікує шаблони дизайну як першокласні об'єкти, які можна інтегрувати у власні програми. Багато організацій і компаній використовують Spring Framework для створення надійних та легко налагоджуваних програм.

Нижче перераховані деякі ключові переваги використання Spring Framework:

- Spring використовує існуючі технології, такі як ORM-фреймворки, реєстраційні фреймворки та інші, замість винаходження власних рішень.
- Тестування програм на Spring є простим, оскільки код, що залежить від середовища, може бути легко переміщений в цей фреймворк. Використання POJO JavaBean-style спрощує ін'єкцію залежностей для тестових даних.
- Веб-фреймворк Spring є добре розробленим MVC фреймворком, що пропонує альтернативу іншим веб-фреймворкам, таким як Struts.
- Spring надає зручний API для перетворення винятків, специфічних для технологій, на послідовні, неперевірені винятки.
- Легкі контейнери IoC, які надає Spring, ефективні та мають низькі вимоги до пам'яті та ресурсів процесора.
- Spring забезпечує послідовний інтерфейс для керування транзакціями, який може масштабуватися від локальних до глобальних транзакцій.
- Аспектно-орієнтоване програмування (AOP) є важливою складовою Spring, що дозволяє відокремлювати функціональні аспекти, такі як журналування, декларативні транзакції, безпека, кешування, від бізнес-логіки програми.

Рис. 2.11. Структура фреймворку Spring

У ООП ключовою одиницею модульності є клас, а в АОП одиницею модульності є аспект. Dependency Injection (DI) допомагає відокремити об'єкти додатків один від одного, тоді як Aspect-Oriented Programming (AOP) допомагає відокремити наскрізні проблеми від об'єктів, на які вони впливають.

Spring є модульним фреймворком, що дозволяє вибрати необхідні модулі та компоненти, адаптувати їх під вимоги проекту і використовувати без необхідності включення зайвих модулів. Це робить Spring потенційно універсальною платформою для корпоративних програм, дозволяючи розробникам вибирати найбільш підходящі модулі та компоненти для своїх потреб.

2.2.4. Особливості використання веб-фреймворку Ruby on Rails

Rails, або Ruby on Rails, є відкритим фреймворком, написаним мовою програмування Ruby. Він був створений Девідом Хайнмайером Ханссоном у 2003 році і часто відомий як DHH.

Творець Ruby on Rails прагнув створити фреймворк "для щастя програміста і прекрасного коду", зосереджуючись на зручності розробки і якості написаного коду (рис. 2.12).

Рис. 2.12. Логотип фреймворку Ruby on Rails

Ruby on Rails надає структуру, що значно спрощує процес створення веб-сайтів та додатків. Фреймворк автоматизує багато повторюваних завдань для розробників, таких як створення форм, таблиць та меню. Розробники можуть використовувати готові рішення для цих завдань, уникнувши необхідності розробляти все з нуля.

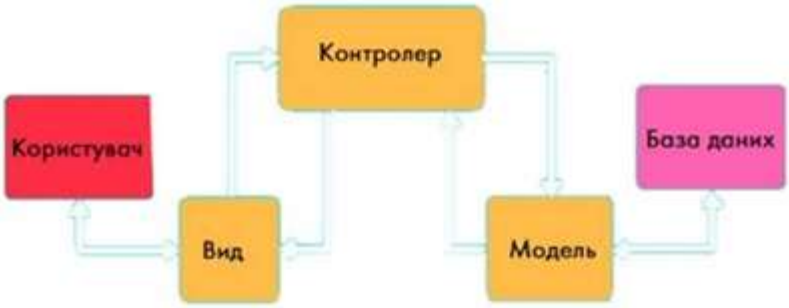


Рис. 2.13. Структура фреймворку Ruby on Rails

Ruby on Rails є популярним фреймворком з великою кількістю веб-сайтів, побудованих на ньому. Він має відкритий вихідний код та широку спільноту розробників, що розробляють безкоштовні доповнення для фреймворку. Ruby on Rails дозволяє створювати багатофункціональні програми швидко і ефективно. Він має кілька переваг, включаючи широку екосистему, чистий код, високу масштабованість, безпеку, ефективність часу та витрат, тестове середовище та багато інших.

Ruby on Rails використовує підхід Model-View-Controller (MVC), що дозволяє відокремити робочу програму на три підсистеми: моделі, контролери та представлення.

Таблиця 2.2 - Таблиця порівняння Laravel проти Ruby on Rails

Основа порівняння	Laravel	Ruby on Rails
Мова	На основі PHP	На основі Ruby
Використання	Додаток на рівні підприємства для невеликих сайтів	Веб-додаток із підтримкою бази даних, метапрограмування
Синтаксис	Використовує крапку з комою та фігурні дужки	Не підтримує крапку з комою та фігурні дужки
Громада	Ларакасти, стрімко зростаючі	Rails Casts, вже відома спільнота Rails
Існування	Початковий випуск – червень 2011 року	Прийшов у 2004 році
Особливості	Швидкий ORM, власні можливості автентифікації	Ruby з чітким синтаксисом, сторонніми плагінами, хорошою підтримкою MVC
ОРМ	Красномовним є використовуваний ОРМ	ActiveRecord – це використовуваний ORM
Шаблон двигуна	Лезо – це механізм шаблонів	ERB є двигуном за замовчуванням

Кінець таблиці 2.2

Моделі відповідають за обробку даних та бізнес-логіку додатка. Контролери відповідають за обробку користувальницького інтерфейсу та взаємодію з додатком. Представлення відповідають за відображення даних та взаємодію з користувачем. Ця архітектурна модель дозволяє проводити паралельну розробку та прискорює інженерний процес. Вона надає готові рішення для розділення бізнес-логіки додатку, що дозволяє зберегти час розробки для компаній, які використовують Ruby on Rails.

Ruby on Rails пропонує послідовність і чистий код, завдяки готовим компонентам, що спрощують реалізацію функцій і полегшують читання коду. Це забезпечує швидкі та безперешкодні оновлення програми.

DRY (Don't Repeat Yourself) – ще один принцип, який використовується в Ruby on Rails. Якщо є повторюване завдання, не потрібно писати однаковий код знову і знову. Фреймворк сприймає повторювані завдання як шаблони, які можна використовувати безліч разів.

Ruby on Rails має високу масштабованість. Він здатний обробляти великі навантаження, навіть коли кількість запитів на секунду значно збільшується. Платформа електронної комерції Shopify є прикладом високої продуктивності Ruby on Rails, оскільки вона обробляє велику кількість запитів на секунду.

Ці переваги роблять Ruby on Rails привабливим вибором для проектів, які потребують швидкого розвитку, масштабованості та оптимізованого коду.

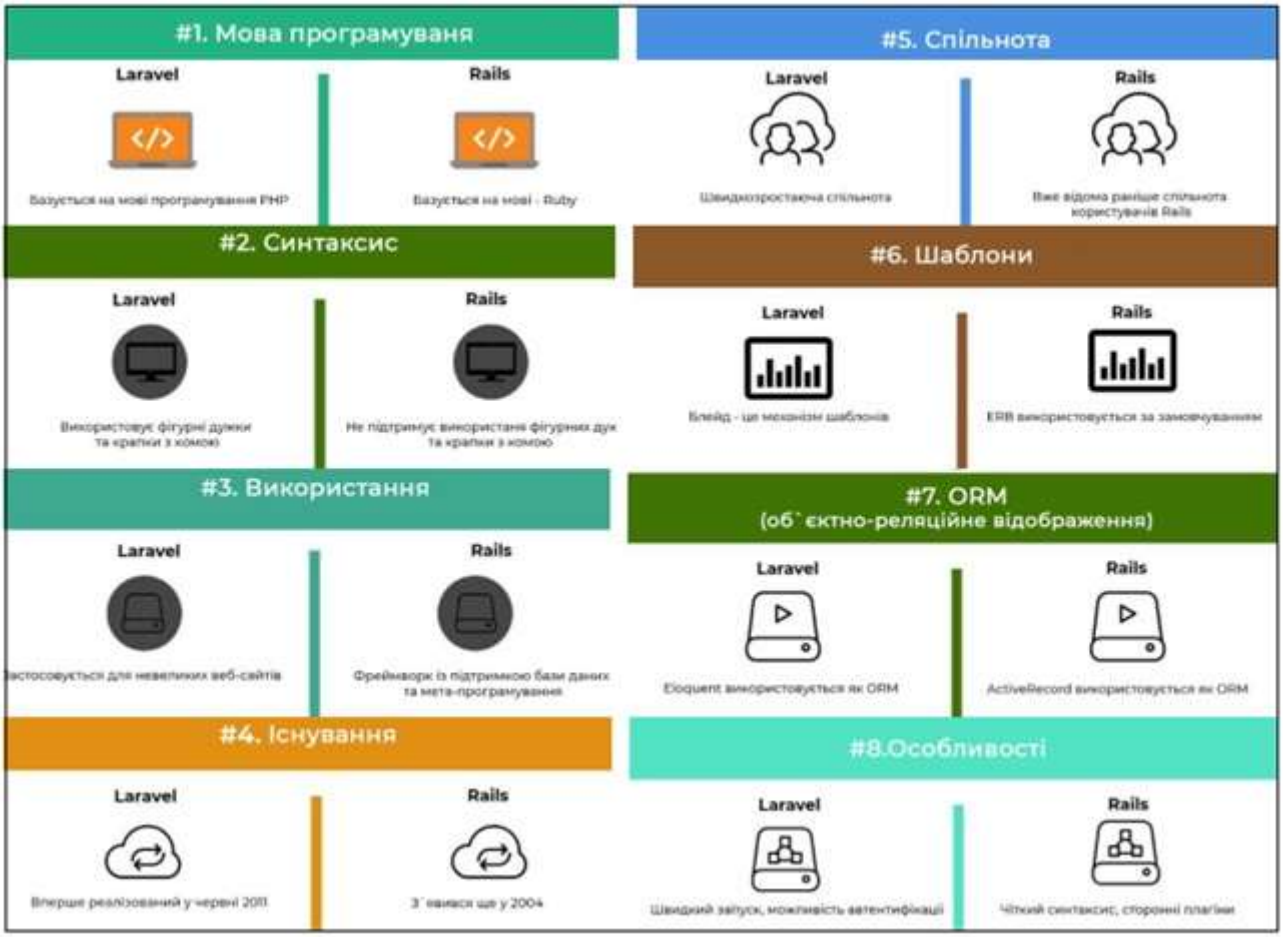


Рис. 2.14. Порівняльна інфорграфіка Laravel проти Ruby on Rails

Ruby on Rails має вбудовані функції безпеки, що захищають програми від SQL-ін'єкцій та атак XSS. Це додає додатковий рівень безпеки до проектів.

Фреймворк Ruby on Rails дозволяє зекономити час та витрати завдяки своїм функціям і продуктивності. Швидка розробка додатків (RAD) є однією з головних сфер використання Ruby on Rails, що допомагає прискорити процес розробки.

Ruby on Rails надає тестове середовище з трьома режимами за замовчуванням (виробництво, розробка та тестування), що дозволяє проводити тестування на різних етапах розробки. Це сприяє виявленню та виправленню помилок раніше, зменшуючи кількість проблем.

Конвенції щодо конфігурації в Ruby on Rails полегшують розробникам налаштування файлів та дозволяють швидше почати використовувати фреймворк. Це забезпечує читабельний та компактний код і спрощує навігацію всередині веб-програми Ruby on Rails.

Рис. 2.15. Завантаження Rails

Ruby on Rails має кілька недоліків, які варто врахувати:

1. Документація: Знайти потрібну документацію для Ruby on Rails може бути викликом, особливо в питаннях, пов'язаних з рідкіснішими функціями або гемами.

2. Швидкість виконання: Швидкість виконання Ruby on Rails не є найшвидшою серед інших об'єктно-орієнтованих мов програмування, що може вплинути на продуктивність додатків у деяких випадках.

3. Швидкість завантаження: Запуск Ruby on Rails може зайняти більше часу через велику кількість файлів та залежностей, що використовуються в проекті.

4. Хостинг веб-сайтів: Не всі хостинг-провайдери підтримують Ruby on Rails, і вимоги фреймворка можуть бути важким завданням для провайдерів з обмеженими ресурсами.

Ці недоліки варто враховувати і оцінювати, чи відповідає Ruby on Rails потребам проекту, перед вибором фреймворку.

РОЗДІЛ 3.
ЕТАПИ РОЗРОБКИ ДОДАТКУ ІНТЕРАКТИВНОГО СПІЛКУВАННЯ ЗА ДОПОМОГОЮ ФРЕЙМВОРКУ DJANGO

3.1. Опис функціональності розробки

Згідно мети нашого проекту, яка полягає в створенні веб застосунку для інтерактивного спілкування між авторами та читачами проаналізуємо функціональні та нефункціональні вимоги до створюваної системи.

Функціональні вимоги до створюваного веб застосунку для інтерактивного спілкування відображають весь набір функцій розроблюваної системи, якими вона повинна бути наділена. Функціональні вимоги до веб застосунку дозволяють виявити модель прецедентів. Прецедент описує послідовність виконуваних системою дій з метою надання конкретних результатів користувачеві. Складовою частиною моделі прецедентів є діаграма варіантів використання (use case diagram), яка включає представлення всіх можливих дійових осіб, асоційованих з усілякими варіантами використання. Діаграма випадків використання повинна повністю описувати функціональність системи на деякому рівні абстракції.

Всі параметри якості веб застосунку – атрибути якості, описуються нефункціональними вимогами. Атрибутами якості є характеристики програмного продукту, що відображають призначені для користувача характеристики: доступність, зручність установки, продуктивність, надійність, умови експлуатації і т.д. до найбільш важливих атрибутів якості відносяться: надійність, продуктивність, зручність використання програмного продукту.

Створюваний веб застосунок повинен задовольняти наступним функціональним вимогам. Система повинна:

- надавати можливість створити акаунт користувачеві;
- надавати можливість проводити налаштування свого кабінету;

– надавати можливість налаштовувати шляхи комунікації з іншими користувачами (читачами) згідно запитів користувача;
– надавати можливість заблокувати певного користувача;
– могти проводити особисті спілкування з іншим користувачем (щоб інші люди не бачили сутність переписки);
– створити повідомлення для інформування користувача тоді коли інший користувач хоче поспілкуватись (звукове повідомлення або звуковий сигнал);
– могти завантажувати в текст переписки певні аудіо- або відеофайли для поширення серед читачів;
– надавати можливість створювати нотатки та працювати з ними;
– надавати можливість дивитися статистику по проведеному спілкуванню;
– дозволяти вносити зміни про дані користувача.

Під час аналізу предметної області було виділено основні функції та підфункції системи, що відповідають потребам користувача:

– Нефункціональні вимоги до створюваного веб застосунку не менш важливі при розробці. Вони дозволяють зробити роботу з програмним продуктом більш комфортним для користувача.
– Вимоги до надійності. Одним з основних атрибутів якості програмного забезпечення є його надійність – здатність безвідмовно працювати і зберігати характеристики при дотриманні певних умов використання.

Серед вимог до надійності веб застосунку виділяють:

– Відновлення після збоїв. При виникненні апаратних збоїв (при відключенні електроживлення мережі), система повинна автоматично відновлюватися після усунення неполадок і / або причин збоїв. Виняток становлять ситуації, пов’язані з технічним пошкодженням носіїв інформації з виконуваним програмним кодом. А також, ситуації, пов’язані з доступом до інформації на сторонньому ресурсі.

– Обробка помилкових ситуацій. В системі повинні бути передбачені засоби інформування користувача про помилковість його дій. При виникненні винятків система повинна проаналізувати і спробувати ідентифікувати помилку. При успішній ідентифікації помилки система повинна вивести користувачеві відповідне повідомлення.
– Вимоги до продуктивності. Найважливішими аспектами розробки програмного продукту є продуктивність і продуктивність роботи, тобто здатність програми виконувати передбачені алгоритмом функції швидко і якісно.

Виділяють наступні вимоги до продуктивності веб застосунку:

– час відгуку системи має становити не більше 3 секунд, а час відповіді стороннього ресурсу на запит відповідних даних – не більше 5 секунд.
– система повинна бути здатна виводити імпортовані дані за прийнятний час. Виняток становить випадок помилки імпорту даних.
– Вимоги до ергономіки і зручності експлуатації. Користувальницький інтерфейс забезпечує взаємодію користувача і програмного забезпечення, отже, від його характеристик залежить ефективність використання і задоволення користувача від роботи з системою.

Виділяються наступні вимоги до зручності експлуатації та ергономіки до створюваного веб застосунку:

– інтерфейс системи повинен бути інтуїтивно зрозумілий користувачеві;
– повідомлення про помилки повинні бути короткими, але інформативними.
– зовнішня поведінка (відгук на дії користувача) однотипних елементів інтерфейсу повинні бути однаково реалізовані.

Також веб застосунок для спілкування повинен задовольняти наступним нефункціональним вимогам: сучасна версія програмного забезпечення; українська та англійська версія користувальницького інтерфейсу; додаток повинен мати зручний і зрозумілий інтерфейс, бути безкоштовним і без реклами.

Архітектура веб застосунку для інтерактивного спілкування між авторами та читачами визначає найбільш загальне уявлення про структуру і логіку функціонування, вказує основні складові і зв'язки між ними, визначає подальшу реалізацію програмного продукту. Для опису архітектури веб застосунку для інтерактивного спілкування використовується уніфікована мова моделювання (UML) відповідно до рекомендацій методології Rational Unified Process. Пропонована нами архітектура поєднує та автоматизує рівні прийому, зберігання, обробки та спеціального аналізу. Прототип складається з двох основних робочих процесів для ефективної обробки різних потоків даних, а саме: 1) Конвеєр великих даних; 2) Конвеєр обробки даних.

Конвеєр великих даних. Основна роль конвеєра великих даних полягає в автоматизації переміщення величезних даних з різними типами даних із зовнішніх джерел потоків даних до бази даних для подальшого аналізу.

Крім переміщення даних, також використовуємо дані з традиційних реляційних баз даних та статичних сховищ даних, таких як файли: jpg, mp3, mp4, json, щоб їх можна було ефективніше аналізувати наступними процесами в базі даних.

Виділяються наступні компоненти UML-специфікації веб застосунку для інтерактивного спілкування між авторами та читачами:

- Концептуальна модель, що включає опис основних сценаріїв роботи системи, варіантів використання, учасників системи.
- Логічна модель, що включає опис класів і функціональних модулів веб застосунку.
- Модель розміщення, що включає вимоги до апаратного забезпечення користувача, опис розміщення компонентів системи.
- Модель реалізації, що включає опис фізичних артефактів веб застосунку для інтерактивного спілкування між авторами та читачами: файли, виконувані файли, використовувані бібліотеки, а також зв'язки між ними.

Концептуальна модель.

В рамках концептуальної моделі веб застосунку виявляються учасники системи-актори, основні сценарії роботи системи і варіанти використання – прецеденти. Зв’язки акторів і прецедентів представляються на діаграмі варіантів використання. Концептуальна модель програмного забезпечення була побудована на етапі виявлення функціональних вимог.

Логічна модель.

В рамках логічної моделі веб застосунку для інтерактивного спілкування між авторами та читачами описуються значущі функціональні модулі і класи системи. Модулі проектованої системи представлені в діаграмі класів (див. рис. 3.1).

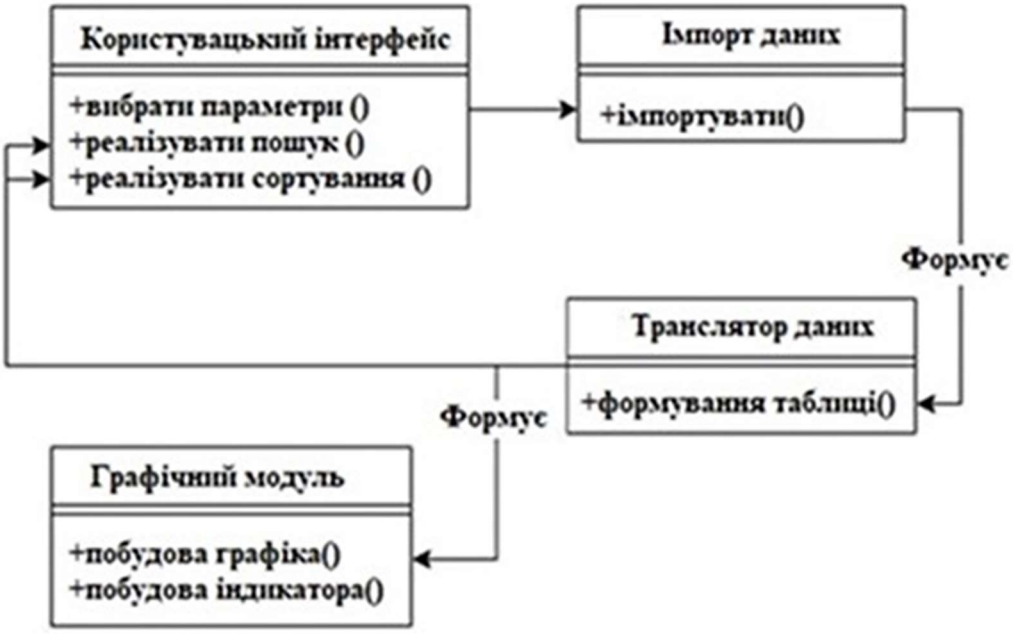


Рис. 3.1. Діаграма класів веб застосунку для інтерактивного спілкування між авторами та читачами

Опис модулів системи наведено в таб. 3.1. Програмне забезпечення створює веб-сторінку і ці дані знаходяться в вільному доступі, інформація про ціни відкриття, закриття, максимуму і мінімуму, про тему спілкування надається користувачеві з затримкою на 15 сек. Імпортовані дані формуються в таблицю, потім проводиться побудова графіка і індикатора на графіку.

Таблиця 3.1 - Опис модулів веб застосунку для інтерактивного спілкування між авторами та читачами

Найменування	Опис
Користувацький інтерфейс	Містить атрибути взаємодії користувача з програмою. Дозволяє вибрати параметри даних, реалізувати пошук і сортування.
Імпорт даних	Містить атрибут, що виконує імпорт даних з сайту спілкування (добавляння цікавих співрозмовників та видалення агресивних користувачів)
Транслятор даних	Містить атрибут, що формує імпортовані дані в таблицю.
Графічний модуль	Містить атрибут, що виробляє побудову графіка і індикатора на основі імпортованих даних.

Кінець таблиці 3.1

Модель розміщення. В рамках моделі розміщення веб застосунку формується діаграма розгортання, що містить опис фізичного простору для розміщення компонентів. Діаграма розгортання представлена на рис. 3.2.

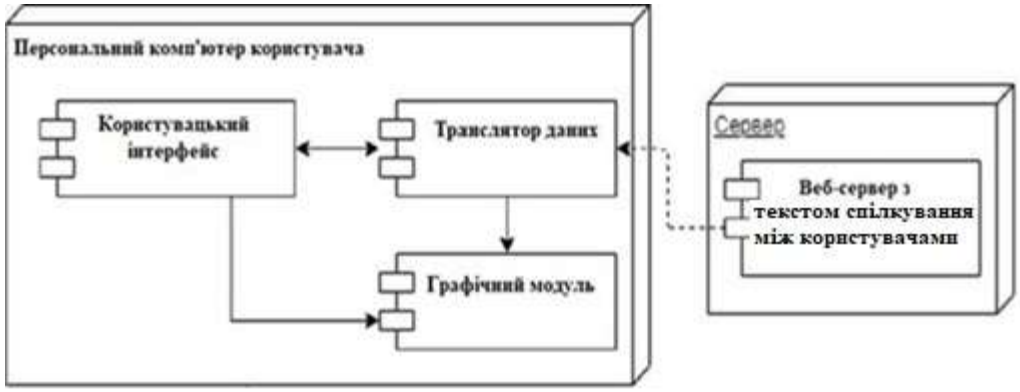


Рис. 3.2. Діаграма розгортання веб застосунку для інтерактивного спілкування

Модель реалізації. Для безпосередньої реалізації веб застосунку для інтерактивного спілкування необхідно виділити основні логічні і функціональні структури. Для опису елементів системи використовується діаграма компонентів.

Дерево цілей для веб застосунку для інтерактивного спілкування будується шляхом переформулювання проблем у цілі, створення взаємовідносин «засоби – результати» та обрання однієї чи кількох конкретних цілей. Алгоритм дерева рішень веб застосунку для інтерактивного спілкування належить до сімейства алгоритмів навчання з наглядом. На відміну від інших алгоритмів навчання з

наглядом, алгоритм дерева цілей також можна використовувати для вирішення проблем регресії та класифікації.

Згідно основної мети даного дослідження є робота з використанням веб застосунку для інтерактивного спілкування. Досягти поставлену мету можна за допомогою реалізації таких цілей: аналіз предметної області, проектування системи, реалізація програмної частини, удосконалення системи, підтримка продукту. Описане дерево цілей для даної роботи можна переглянути на рис. 3.3 у такому вигляді.

Рис. 3.3. Зовнішній вигляд дерева цілей згідно досліджуваної проблематики в роботі

Наступними кроком при системному аналізі об’єкту дослідження є створення матриці порівняння критеріїв з типами систем (таб. 3.2). Нормалізацією елементів стовпця та знаходженням середнього значення рядка визначено вектор пріоритетів.

Таблиця 3.2 - Ранжування критеріїв в дереві цілей для веб застосунку для інтерактивного спілкування між авторами та читачами

№	Назва критерію	Порівняння критеріїв						Вектор пріоритету
		1	2	3	4	5	6	
1	Актуальність	1	1 / 4	1 / 3	5	1 / 4	2	0,15
2	Точність	1	3	7	7	1	1	0,26
3	Зручність	1 / 7	4	1	5	1 / 2	1 / 7	0,114
4	Коректність	2	1	4	1 / 6	3	1	0,17
5	Доступність	1 / 3	1 / 7	1 / 5	1 / 5	1 / 8	1	0,04
6	Якість	1 / 2	7	1	6	1	8	0,33

Кінець таблиці 3.2

Після чого здійснюється попарне порівняння критерію з типами систем. Таким чином було здійснене попарне порівняння систем для кожного визначеного критерію (таб. 3.3–3.9).

Таблиця 3.3 - Матриця попарного порівняння систем за критерієм «Актуальність» в дереві цілей для створенні веб застосунку для інтерактивного спілкування між авторами та читачами

№	Назва системи	Порівняння систем					Вектор пріоритету
		1	2	3	4	5	
1	Інформаційно-консультаційні	1 / 2	1 / 7	6	1 / 6	1 / 7	0,4
2	Інформаційно-керівні	6	1 / 3	5	1 / 4	1 / 4	0,2
3	Інформаційно-пошукові	1 / 5	1 / 6	1 / 2	1 / 4	1 / 6	0,3
4	Інтелектуальні інформаційні	4	1 / 3	5	1	1 / 2	0,5
5	СППР	5	3	5	1 / 2	1	0,45

Кінець таблиці 3.3

Таблиця 3.4 - Матриця попарного порівняння систем за критерієм «Точність» в дереві цілей для веб застосунку для інтерактивного спілкування між авторами та читачами

№	Назва системи	Порівняння систем					Вектор пріоритету
		1	2	3	4	5	
1	Інформаційно-консультаційні	1 / 2	6	1 / 4	1 / 5	7	0,74
2	Інформаційно-керівні	1 / 2	3	1 / 4	1 / 4	6	0,28
3	Інформаційно-пошукові	1 / 5	1 / 7	1 / 5	1 / 3	5	0,22
4	Інтелектуальні інформаційні	1 / 4	3	3	1 / 8	1 / 3	0,27
5	СППР	1 / 6	1 / 2	1 / 4	1 / 5	1 / 4	0,21

Кінець таблиці 3.4

Таблиця 3.5 - Матриця порівняння систем для критерію «Зручність» в дереві цілей веб застосунку для інтерактивного спілкування між авторами та читачами

№	Назва системи	Порівняння систем					Вектор пріоритету
		1	2	3	4	5	
1	Інформаційно-консультаційні	1 / 2	1 / 4	2	1 / 5	5	0,2
2	Інформаційно-керівні	4	1 / 2	5	1 / 4	3	0,5
3	Інформаційно-пошукові	1 / 5	1 / 6	1 / 5	1 / 4	7	0,4
4	Інтелектуальні інформаційні	1 / 7	3	5	2	4	0,38
5	СППР	3	1 / 3	1 / 3	1 / 4	4	0,17

Кінець таблиці 3.5

Таблиця 3.6 - Матриця порівняння систем для критерію «Коректність» дереві цілей для веб застосунку для інтерактивного спілкування між авторами та читачами

№	Назва системи	Порівняння систем					Вектор пріоритету
		1	2	3	4	5	
1	Інформаційно-консультаційні	1 / 3	2	3	3	7	0,6
2	Інформаційно-керівні	1 / 3	4	1 / 5	4	5	0,24
3	Інформаційно-пошукові	1 / 2	3	1 / 3	3	1 / 2	0,36
4	Інтелектуальні інформаційні	1 / 2	1	1 / 2	1 / 2	3	0,22
5	СППР	1 / 2	3	4	1 / 5	1 / 4	0,28

Кінець таблиці 3.6

Таблиця 3.7 - Матриця порівняння систем для критерію «Доступність» в дереві цілей для веб застосунку для інтерактивного спілкування між авторами та читачами

№	Назва системи	Порівняння систем					Вектор пріоритету
		1 / 2	3	4	3	5	
1	Інформаційно-консультаційні	3	1 / 4	½	5	1 / 3	0,24
2	Інформаційно-керівні	1 / 7	1 / 2	1 / 3	2	3	0,22
3	Інформаційно-пошукові	3	6	4	3	4	0,6
4	Інтелектуальні інформаційні	4	3	¼	3	1 / 3	0,6
5	СППР	3	2	1 / 3	1 / 4	1	0,49

Кінець таблиці 3.7

Таблиця 3.8 - Матриця порівняння систем для критерію «Якість» в дереві цілей для веб застосунку для інтерактивного спілкування між авторами та читачами

№	Назва системи	Порівняння систем					Вектор пріоритету
		2	3	3	6	3	
1	Інформаційно-консультаційні	3	7	9	1 / 5	2	0,47
2	Інформаційно-керівні	5	3	1 / 8	3	4	0,33
3	Інформаційно-пошукові	1 / 2	1 / 7	1 / 3	1 / 2	3	0,24
4	Інтелектуальні інформаційні	1 / 9	4	1 / 3	1 / 7	½	0,26
5	СППР	1 / 7	6	4	1 / 4	6	0,44

Кінець таблиці 3.8

Наступним кроком при аналізі дерева цілей було створено таблицю, де зібрано усі коефіцієнти критеріїв кожного типу систем, та було визначено кінцеву пріоритетність веб застосунку для інтерактивного спілкування між авторами та читачами у таб. 3.9.

Таблиця 3.9 - Результати методу аналітичної ієрархії в дереві цілей для веб застосунку для інтерактивного спілкування між авторами та читачами

Критерій Тип систем	Актуальність	Точність	Зручність	Коректність	Доступність	Якість	Пріоритет
Інформаційно-консультаційні	0,015	0,034	0,165	0,018	0,086	0,2566	0,49
Інформаційно-керівні	0,15	0,064	0,044	0,029	0,018	0,085	0,28
Інформаційно-пошукові	0,17	0,029	0,023	0,024	0,023	0,036	0,12
Інтелектуальні інформаційні	0,032	0,035	0,042	0,026	0,022	0,033	0,19
СППР	0,033	0,019	0,011	0,028	0,019	0,045	0,29

Кінець таблиці 3.9

Результатом проведеного порівняння є найбільше значення у інформаційно-консультаційній системі. За критичним фактором успішності відібрано цей варіант для використання веб застосунку для інтерактивного спілкування між авторами та читачами.

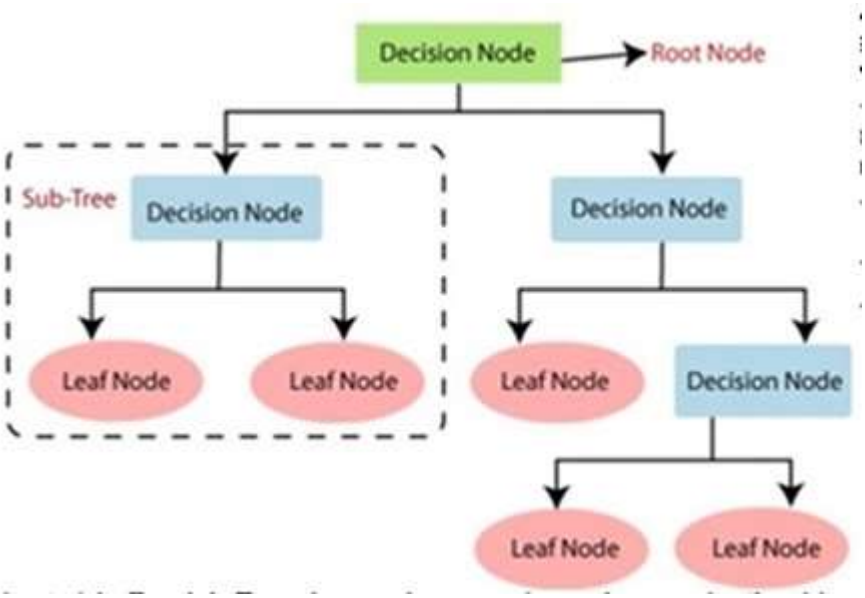


Рис. 3.4. Приклад роботи алгоритму дерева цілей для веб застосунку для інтерактивного спілкування між авторами та читачами

Контекстна діаграма або діаграма нульового рівня, зображена на рис. 3.6 є основою та базовою діаграмою для потоків даних.

Рис. 3.5. Діаграма потоків даних нульового рівня

В результаті розробки контекстної діаграми визначено два елементи DFD для нульового рівня, тобто контекстної діаграми:

– *Користувач.* Зображено зовнішною сутністю, яка використовується для представлення користувач, який взаємодіє з системою.
– *Надати можливість доступу.* Зображено основним процесом, який виконує система, завдяки якому буде відбуватись обробка та потік даних.

Сформовані процеси, їхні потоки та бази даних зображено на рис. 3.6, а саме: створити обліковий запис; сформувати список; відправити сформований результат опрацювань; бази даних: користувачі та база даних та їх потреб.

Рис. 3.6. Декомпозиція системи «Прогнозування зміни» першого рівня для веб застосунку для інтерактивного спілкування

Рис. 3.7. Декомпозиція процесу «Створити обліковий запис» для веб застосунку для інтерактивного спілкування

Наступним кроком потрібно зображено декомпозицію другого рівня процесу «Створити обліковий запис», який розбито на такі процеси: зареєструвати клієнта; авторизувати клієнта.

Зв’язані між собою обліковими даними, які будуть передаватись при авторизації користувача для опрацювання даних та надання доступу до застосунку користувачу.

Рис. 3.8. Декомпозиція процесу «Сформувати список» актуальних тем для автора у веб застосунку для інтерактивного спілкування

Наступний процесом описано дії користувача в межі пошуку результатів тематики спілкування, зображено на рис. 3.9. Сформований запит на перегляд результатів спілкування автора з читачами відправляється користувачем системи та опрацьовується системою. Система реагує на запит та формує список доступних тем.

Наступним складеним процесом є «Сформувати список» обраної тематики. Він складається із процесів, а саме: створити запит на завантаження даних; затвердити вибрані дані. Об’єднує між собою опрацьований матеріал, тобто програми які зараз мають можливість транслюватись. На виході отримуємо опрацьовані дані та запит на перегляд даних.

Рис. 3.9. Декомпозиція процесу «Відправити сформований результат опрацювань» проведених тем спілкування автора

Дана декомпозиція зображена на рис. 3.9 складається з двох процесів:

– вибрати тематику спілкування, що цікава для автора;
– закріпити проведене спілкування в архіві автора.

Дані процеси пов’язані між собою за допомогою програм, які будуть обрані користувачем та опрацьованих запитів. На вході отримано запит пошуку архівних тем спілкування та відновлення переписки при запиті користувача.

В ході проектування бази даних були створені наступні таблиці:

1. TrainingApparatus-Конструктор тематик спілкування;

– Name-Ім’я користувача;
– Description-формулювання запиту до групи спілкування;
– Exercise_type – тема спілкування;
– Exercises_amount – кількість читачів у групі спілкування;
– Allotted_time-виділений час на спілкування.

2. ExerciseType – тип спілкування; Name-Найменування типу; Description-Опис.
3. TrainigTest – проведене спілкування;

Apparatus-Конструктор тематик спілкування;

User-користувач, який бере участь у спілкуванні;

Grade-Оцінка якості і цікавості тематики спілкування;

Solved_exercises – кількість вирішених проблемних тем;

Correct_answers – кількість користувачів на кінець спілкування (актуальність теми);

Time_start – час початку;

Time_spent-витрачений час.

Exercise-предмет спілкування

User-Користувач: Login-Логін; Password-Пароль; Email-Електронна пошта; First_name-Ім’я; Last_name-Прізвище; Second_name-По Батькові; Phone_number – номер телефону; School-навчальний заклад; City – місто.

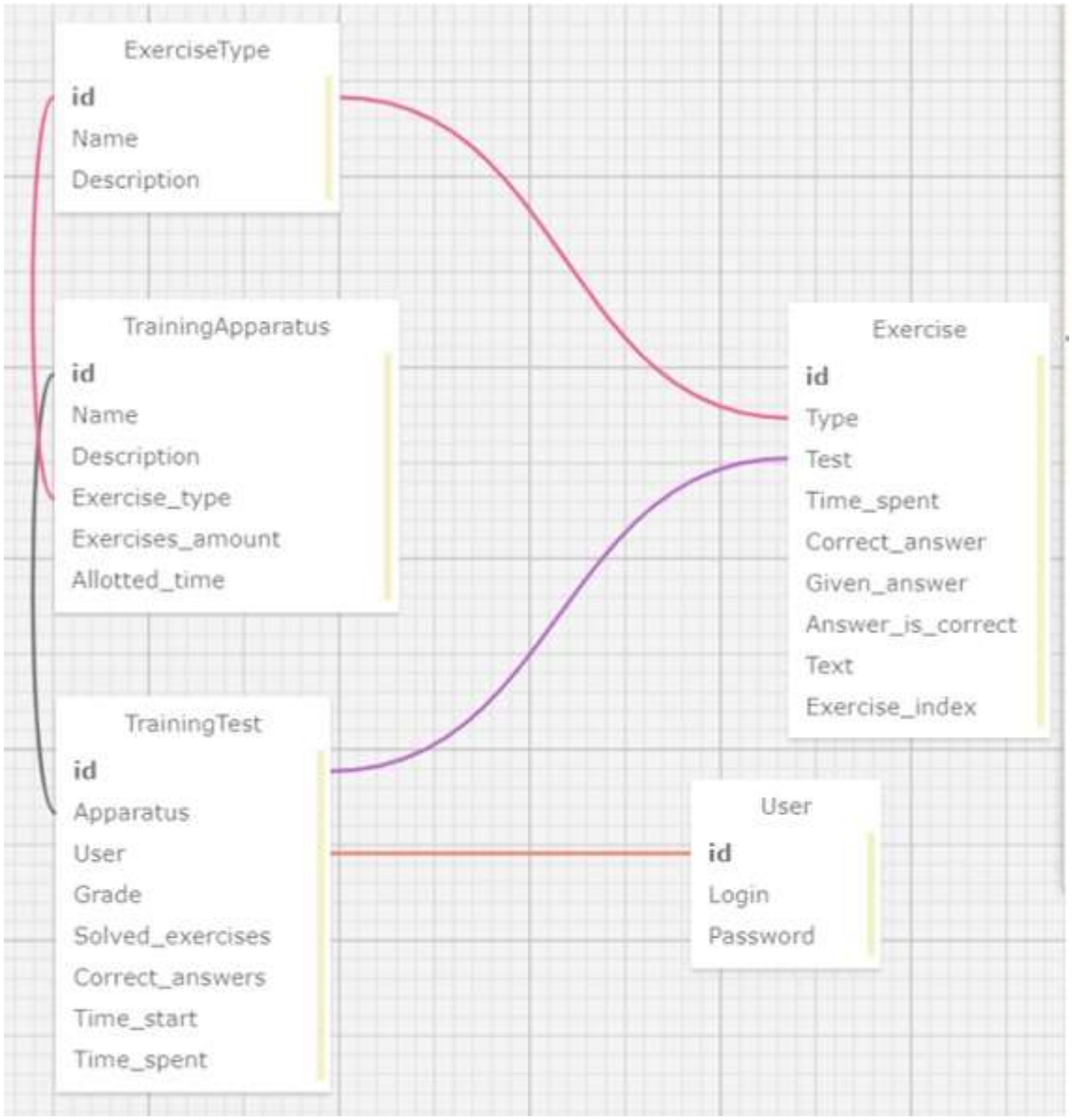


Рис. 3.10. Фрагмент бази даних веб застосунку

Веб застосунок для спілкування був написаний на Django, і він складається з модулів, кожен з яких відповідає за свій розділ і складається з наступних елементів:

1. Модель. Моделі відображають інформацію про дані, з якими необхідно працювати. Вони містять поля та поведінку даних. Звичайна одна модель представляє одну таблицю в базі даних. Код, який описує моделі, знаходиться у файлі models. ру всередині розроблюваного модуля. На основі цих файлів будуть створюватися міграція. Приклад моделі можна побачити на рис. 3.11.

```
from django.db import models

class MyModelName(models.Model):
    """
    A typical class defining a model, derived from the Model class.
    """

    # Fields
    my_field_name = models.CharField(max_length=20, help_text="Enter field documentation")
    ...

    # Metadata
    class Meta:
        ordering = ["-my_field_name"]

    # Methods
    def get_absolute_url(self):
        """
        Returns the url to access a particular instance of MyModelName.
        """
        return reverse('model-detail-view', args=[str(self.id)])

    def __str__(self):
        """
        String for representing the MyModelName object (in Admin site etc.)
        """
        return self.field_name
```

Рис. 3.11. Приклад моделі Django

2. Міграції-це спосіб Django поширювати зміни, які Розробник вносить в свої моделі (додавання поля, видалення моделі і т.д.), в схему бази даних. Файли міграції для кожної програми знаходяться в каталозі «міграції» всередині цієї програми і призначені для передачі та розповсюдження як частина її кодової бази. Міграції будуть виконуватися однаково з однаковим набором даних і давати послідовні результати, а це означає, що те, що бачить Розробник при розробці на проміжному етапі, за тих самих обставин буде саме тим, що станеться у виробництві. Django виконає міграції для будь – яких змін моделей або полів – навіть для параметрів, які не впливають на базу даних, оскільки єдиний спосіб правильно відновити поле-це зберегти всі зміни в історії. Приклад міграції можна побачити на рис. 3.12.

```
from django.db import migrations, models

class Migration(migrations.Migration):

    dependencies = [("migrations", "0001_initial")]

    operations = [
        migrations.DeleteModel("Tribble"),
        migrations.AddField("Author", "rating", models.IntegerField(default=0)),
    ]
```

Рис. 3.12. Приклад міграції Django

3. Уявлення. Функція подання, або коротко подання-це функція Python, яка приймає web-запит і повертає Web-відповідь. Відповіддю може бути HTML-вміст сторінки, або перенаправлення, або 404 помилка, або XML-документ, або зображення. Подання містить всю необхідну логіку для створення відповіді. Цей код може бути де завгодно, головне, щоб він знаходився в PYTHON_PATH. Незважаючи на можливість розташувати код подань де завгодно, прийнято тримати його у файлі views. py, який знаходиться в каталозі проекту або програми. Приклад подання в Django можна побачити на рис. 3.13.

```
class BookListView(ListView):
    model = Book

    def head(self, *args, **kwargs):
        last_book = self.get_queryset().latest('publication_date')
        response = HttpResponse(
            # RFC 1123 date format.
            headers={'Last-Modified': last_book.publication_date.strftime('%a, %d %b %Y %H:%M:%S
GMT')},
        )
        return response
```

Рис. 3.13. Приклад подання Django

4. Серіалізатори. Стерилізатори дозволяють перетворювати складні дані, такі як Набори запитів та екземпляри моделей, у власні типи даних Python, які потім можуть бути легко перетворені в JSON, XML або інші типи вмісту. Серіалізатори також забезпечують десеріалізацію, дозволяючи перетворювати проаналізовані дані назад у складні типи після першої перевірки вхідних даних. Серіалізатори в середовищі REST працюють

подібно до КЛАСІВ Django Form та ModelForm. Клас Serializer дає потужний універсальний спосіб управління виведенням ваших відповідей, А клас ModelSerializer надає швидкий спосіб для створення серіалізаторів, які працюють з екземплярами моделі і наборами запитів. Приклад серіалізатора можна побачити на рис. 3.14.

```
from rest_framework import serializers

class CommentSerializer(serializers.Serializer):
    email = serializers.EmailField()
    content = serializers.CharField(max_length=200)
    created = serializers.DateTimeField()
```

Рис. 3.14. Приклад серіалізатора Django

5. Посилання. Будь – яка сторінка в Інтернеті потребує власної URL-адреси. Таким чином додаток точно знає, що показати користувачеві, який відкриває конкретну URL-адресу. У Django використовується так званий URLconf (англ. URL configuration, конфігурація URL). URLconf-це набір шаблонів, які Django спробує порівняти з отриманою URL-адресою, щоб вибрати правильний метод відображення (view). Приклад URLconf у Django можна побачити на рис. 3.15.

```
from django.urls import path

from . import views

urlpatterns = [
    path('articles/2003/', views.special_case_2003),
    path('articles/<int:year>/', views.year_archive),
    path('articles/<int:year>/<int:month>/', views.month_archive),
    path('articles/<int:year>/<int:month>/<slug:slug>/', views.article_detail),
]
```

Рис. 3.15. Приклад URLconf в Django

Тепер, знаючи структуру модулів Django, можна приступати до їх розробки. За технічним завданням необхідно розробити три модулі: Тренажер, Контроль і Статистика.

У модулі Тренажер згідно зі схемою бази даних створені наступні моделі: exercise Type; grades; training Apparatus; training Test; exercise.

На рис. 3.16 представлена модель конструктора тренажера Training Apparatus, яка успадковується від моделі оцінок Grades. Це зроблено для того, щоб у кожного тренажера за замовчуванням були поля для конфігурації процентажу успішно виконаних завдань для досягнення тієї чи іншої оцінки. Також, за допомогою параметра verbose_name було налаштовано відображення полів в панелі адміністратора, а за допомогою класу Meta було налаштовано відображення всієї моделі.

```
class TrainingApparatus(Grades):
    name = models.CharField(verbose_name=u"Название тренажера", max_length=200)
    description = models.TextField(verbose_name="Описание", max_length=1000, blank=True, null=True)
    exercises_type = models.ForeignKey(ExerciseTypes, verbose_name=u"Тип(ы) примеров", on_delete=models.SET_DEFAULT,
                                       default=None, blank=True, null=True, help_text=u"В РАЗРАБОТКЕ")
    exercises_amount = models.PositiveIntegerField(verbose_name=u"Количество примеров", default=0)
    allotted_time = models.CharField(verbose_name=u"Время на выполнение", max_length=9, default="00:00:00", help_text=u"Формат - "00:00:00"')

    def __str__(self):
        return self.name

    class Meta:
        verbose_name = u'тренажер'
        verbose_name_plural = u'тренажеры'
```

Рис. 3.16. Модель Training Apparatus

Щоб правильно реалізувати роботу тренажера, був розроблений наступний алгоритм:

1. На сторінці «Тематика спілкування» відображається список, що випадає, містить в собі всі теми спілкування автора. Щоб це зробити, необхідно написати подання, яке буде робити запит в базу даних з допомогою ORM, отримувати всі об’єкти конструкторів тренажерів і малювати їх за допомогою параметра context в шаблоні сторінки. Також необхідно передбачити, щоб доступ до цієї сторінки мали тільки зареєстровані користувачі. Це можна зробити за допомогою спеціального декоратора @login_required. Декоратор-це функція, яка дозволяє обернути іншу функцію для розширення її функціональності без безпосередньої зміни її коду. Реалізована функція зображена на рис. 3.17.

```
@login_required
def index(request):
    context = {
        'tests': TrainingApparatus.objects.all()
    }
    return render(request, 'mysite/lms.html', context)
```

Рис. 3.17. Функція подання, що передає в шаблон всі тренажери

Після отримання запиту про початок тренування викликається подання create_test, яке, в залежності від конструктора створює екземпляр класу TrainingTest, прив'язаний до користувача, який почав тренування і генерує список завдань, прив'язаний до цього тесту. Фрагмент подання створення тесту представлений на рис. 3.18.

```
def create_test(request):
    print(request.body)
    req = json.loads(request.body)
    apparatus = TrainingApparatus.objects.get(name=req['trainingApparatus'])
    user = Account.objects.get(pk=request.user.id)
    if req['testType'] == 'training':
        test = TrainingTest(apparatus=apparatus, user=user, grade=0)
    elif req['testType'] == 'control':
        test = ControlTest(apparatus=apparatus, user=user, grade=0)
    test.time_start = datetime.datetime.now()
    test.save()
```

Рис. 3.18. Фрагмент подання створення тесту

Після генерації завдань відбувається переадресація користувача на сторінку вирішення завдань. На цьому етапі перевіряється, чи є створений тест контрольним, для того щоб система визначила, чи враховувати його результати на сторінці «Контроль». Код подання exercise_view, що описує логіку даного етапу, можна побачити на рис. 3.19.

```
def exercise_view(request):
    test = TrainingTest.objects.filter(user_id=request.user.id).last()
    control_tests = ControlTest.objects.all().values_list('id', flat=True)
    is_control = False
    if test.id in control_tests:
        is_control = True
    context = {
        'is_control': is_control,
        'test_description': test.apparatus.description,
    }
    return render(request, 'polls/primer.html', context)
```

Рис. 3.19. Код подання exercise_view

2. За запитами від клієнта сервер видає згенеровані раніше завдання, перевіряючи, чи не скінчився час і чи не останнє це завдання. При виконанні однієї з цих умов відбувається переадресація на сторінку результатів тестування. Фрагмент коду, що описує цю логіку можна побачити на рис. 3.20.

```
if exercise_index <= test.apparatus.exercises_amount and time_spent <= get_seconds_from_string(test.apparatus.allotted_time):
    unsolved_exercise = test.exercises.filter(exercise_index=exercise_index)[0]
else:
    return JsonResponse({
        "url": "/end_test/",
        "test_id": test.id
    })
```

Рис. 3.20. Перевірка умов переходу на сторінку результатів

На цьому ж етапі сервер за запитом видає список з уже вирішених завдань в ході цього тестування, в якому вказано, чи вірно вирішено завдання, скільки на нього витрачено часу, яка відповідь була дана, яка відповідь вірна і т. п. цей функціонал обгорнутий в конструкцію try-except, щоб запобігти помилку при спробі отримати історію в момент, коли ще жодного прикладу не було вирішено. Фрагмент коду, що відповідає за складання списку показаний на рис. 3.21.

```
solved_exercises = Exercise.objects.filter(test=req['test_id'], time_spent__isnull=False)
for exercise in solved_exercises:
    solved_exercise = {
        'text': exercise.text,
        'pk': exercise.pk,
        'is_correct': exercise.answer_is_correct,
        'correct_answer': exercise.correct_answer,
        'given_answer': exercise.given_answer,
        'time_spent': exercise.time_spent
    }
    history.append(solved_exercise)
```

Рис. 3.21. Складання списку вирішених прикладів

Під час надсилання відповіді на сервер викликається подання check_answer. Воно порівнює дану відповідь з вірним і оновлює відповідну сутність Exercise, заповнюючи в ній поля given_answer, time_spent і answer_is_correct. Також необхідно запобігти відправку порожнього поля, це необхідно врахувати при програмуванні. Фрагмент коду з подання check_answer можна побачити на рис. 3.22.

```
def check_answer(request):
    req = json.loads(request.body)
    exercise = Exercise.objects.get(pk=req['pk'])
    exercise.given_answer = req['value']
    exercise.exercise_index = req['exercise_index']
    test = TrainingTest.objects.filter(user_id=request.user.id).last()
    exercise.time_spent = str(datetime.datetime.fromtimestamp(req['time_spent']).time()).split('.')[0]
    if str(exercise.given_answer) != '' and exercise.given_answer == exercise.correct_answer:
        is_correct = True
    else:
        is_correct = False
```

Рис. 3.22. Фрагмент коду подання check_answer

Після закінчення тестування відбувається переадресація користувача на сторінку результатів і їх збереження в базу даних. В цей же момент, виходячи з відсотка вірних відповідей і налаштувань конструктора тестів, розраховується отримана оцінка. Фрагмент, що реалізує цю логіку представлений на рис. 3.23.

```
def end_test_id(request, test_id):
    test = TrainingTest.objects.get(id=int(test_id))
    correct_answers = test.exercises.filter(answer_is_correct=True).count()
    solved_exercises = test.exercises.filter(time_spent__isnull=False)
    exercises_amount = test.apparatus.exercises_amount
    correct_answers_percentage = correct_answers/exercises_amount*100
    if correct_answers_percentage >= test.apparatus.perfect:
        grade = 5
    elif correct_answers_percentage >= test.apparatus.good:
        grade = 4
    elif correct_answers_percentage >= test.apparatus.satisfactory:
        grade = 3
    else:
        grade = 2
    test.grade = grade
    test.solved_exercises = solved_exercises.count()
```

Рис. 3.23. Фрагмент коду, що визначає оцінку за спілкування по зацікавленості читачів

Під кожне подання модуля тематики спілкувань автором був прописаний відповідний url, щоб система змогла визначити, що саме від неї хоче користувач. Фрагмент списку посилань продемонстрований на рис. 3.24.

```
urlpatterns = [
    # ex: /polls/
    path('', views.index, name='index'),


    # ex: /polls/primer
    url(r'^primer$', views.exercise_view, name='primer'),
    path('api_create_test/', views.create_test, name='create_test'),
    path('api_get_exercise/', views.get_exercise, name='get_exercise'),
    path('api_get_allotted_time/', views.get_allotted_time, name='get_allotted_time'),
    path('api_check_answer/', views.check_answer, name='check_answer'),
    path('api_get_history/', views.get_history, name='get_history'),
```

Рис. 3.24. Фрагмент коду urls. ру модуля тематики спілкувань

Після того, як була реалізована сторінка «Тематики спілкувань», була розпочата розробка сторінки «Статистика», на якій передбачається спостерігати прогрес освоєння навичок учнем за допомогою візуалізації даних про його результати.

На сторінці статистики, також, як і на сторінці тематик спілкування, за допомогою меню, вибирається тема архівного або запланованого спілкування автора, за яким необхідно подивитися результати. У цей момент на сервер надсилається запит до подання get_status, який повертає тести, відфільтровані за запитом користувача та обрану ним тему. Код, що описує логіку надання даних про результати зображений на рис. 3.25.

```
def get_stats(request):
    req = json.loads(request.body)
    tests = TrainingTest.objects.filter(user_id=request.user.id, apparatus__name=req['trainingApparatus'],
                                        time_spent__isnull=False)
    serializer = TrainingTestSerializer(tests, many=True)
    return JsonResponse({
        'tests': serializer.data,
    })
```

Рис. 3.25. Надання статистики за результатами тематик спілкування автора

На рис. 3.26 видно, що в поданні йде звернення до серіалізатора TramingTestSerializer, при цьому, важливо, що йому передається параметр many=True, що говорить про те, що в серіалізатор буде відправлений не один об’єкт, а масив об’єктів, і на виході вийде масив з об’єктів, серіалізованих в формат JSON.

Крім цього, були створені моделі для категорій тестів, а саме моделі TestGroupConstructor і User Test Group. Модель TestGroupConstructor, представлена на рис. 3.26, необхідна для загального конфігурування набору тестів, що входять в категорію, її назви, а також порядкового номера в списку категорій, в той час як модель UserTestGroup, фрагмент якої представлений на рис. 3.27, необхідна для відстеження індивідуального прогресу кожного учня. Об’єкт цієї моделі створюється під кожного учня і зберігає в собі поточний статус проходження.

```
class TestGroupConstructor(models.Model):
    name = models.CharField(u'название', max_length=255)
    tests = models.ManyToManyField(TrainingApparatus, related_name='groups', verbose_name=u'тесты')
    order = models.PositiveIntegerField()
    active = models.BooleanField(default=False)
```

Рис. 3.26. Модель TestGroupConstructor

```
class UserTestGroup(models.Model):
    user = models.ForeignKey(Account, verbose_name=u"Пользователь",
                             on_delete=models.CASCADE, default=None, blank=True, null=True)
    constructor = models.ForeignKey(TestGroupConstructor, verbose_name=u'Группа',
                                    on_delete=models.CASCADE, default=None, blank=True, null=True)

    @property
    def is_complete(self):
        is_complete = False
        tests = ControlTest.objects.filter(user=self.user, apparatus__groups__id__exact=self.constructor.id).filter(grade__gte=3)
        apparatus_set = set(tests.values_list('apparatus', flat=True))
```

Рис. 3.27. Модель UserTestGroup

Модуль зберігання і обробки документів містить в собі дві моделі: модель категорії документа і модель екземпляра документа. Примірник документа має зв'язок «один до одного», що дозволить здійснювати пошук і фільтрацію документів за категоріями.

Модель категорії документів повинна містити в собі наступні поля:

– Заголовок категорії-рядковий тип даних довжиною 150 символів, обов'язковий атрибут.
– Текст опису категорії-текстовий тип даних, необов'язковий атрибут.

Модель екземпляра документа повинна включати в себе наступні поля:

– Назва документа-строковий тип даних довжиною 150 символів, обов'язковий атрибут.
– Дата додавання документа-формат зберігання дати і часу, обов'язковий атрибут. Автоматично приймає поточне значення часу в момент додавання документа.
– Файл документа-рядковий тип довжиною 255 символів, що зберігає шлях до файлу на сервері, обов'язковий атрибут.

– Категорія-зв’язок» один до одного», що зберігає прив’язку файлу до екземпляра моделі категорії документів. Є обов’язковим атрибутом.

Модуль зворотного зв’язку служить для зручного спілкування користувачів з автором. Існує велика кількість методів реалізації зворотного зв’язку з користувачем. Це сервіси миттєвих повідомлень, сервіси зворотного дзвінка, форми відправки e-mail. На сайті був реалізований сервіс форми відправки e-mail зі створенням заявки в панелі управління сайтом. Заявки, прийняті з форми зворотного зв’язку, зберігаються у вигляді екземплярів моделі Feedback модуля зворотного зв’язку. Ця модель повинна містити наступні поля:

– Дата подачі заявки-формат зберігання дати і часу, обов’язковий атрибут. Автоматично приймає поточне значення часу в момент додавання заявки.
– Автор-текстовий тип довжиною 100 символів, обов’язковий атрибут.
– Зворотна адреса електронної пошти-текстовий тип довжиною 150 символів, обов’язковий атрибут.
– Текст повідомлення-текстовий тип даних, обов’язковий атрибут.

Протокол HTTP (HyperText Transfer Protocol) має кілька способів взаємодії користувача з сервером. Це методи GET, POST, PUT, DE-lete. Для передачі даних зазвичай використовуються GET і POST. Метод запиту POST призначений для запиту, при якому веб-сервер приймає дані, укладені в тіло повідомлення, для зберігання. Він часто використовується для завантаження файлу або подання заповненої веб-форми. На відміну від нього, метод HTTP GET призначений для отримання інформації від сервера. В рамках GET запиту деякі дані можуть бути передані в рядку запиту URL, що вказують, наприклад, умови пошуку, діапазони дат, або іншу інформацію, що визначає запит. В рамках POST запиту довільну кількість даних будь-якого типу може бути відправлено на сервер в тілі повідомлення запиту.

Обробка заявки користувача на зворотний зв’язок покладена на функцію get_feedback. Як вхідні дані функція приймає дані із заповненої Користувачем форми. Ці дані передаються методом POST. Отримані дані перевіряються на

наявність в повідомленні шкідливого коду, який міг би вплинути на роботу сайту. Після перевірки дані зберігаються у вигляді нового екземпляра моделі заявки в модулі зворотного зв’язку

РОЗДІЛ 4.
ВІЗУАЛІЗАЦІЯ РОБОТИ СТВОРЕНОГО ЗАСТОСУНКУ

Особливості веб-додатків також включають:

1. Можливість створювати великі розподілені системи: Веб-додатки можуть бути розгорнуті на розподілених серверах та спілкуватися між собою за допомогою запитів, що дозволяє розподіляти навантаження та зберігати великі обсяги даних.

2. Недоліки стосовно кібербезпеки: Високонавантажені проекти можуть бути більш вразливими до кібератак через відкритий доступ до запитів та більший час реакції користувача. Забезпечення безпеки є важливою частиною розробки обсяжних веб-додатків.

3. Слабкий захист інтелектуальної власності: Інформація вантажиться на веб-клієнт та передається для перегляду, що може створювати ризики щодо захисту інтелектуальної власності. Проте популярність веб-додатку може допомогти залучити багато користувачів до корисних сервісів.

4. Зміна бізнес-моделі: Веб-додатки змінюють традиційну бізнес-модель, оскільки програмне забезпечення стає постійно підтримуваним продуктом, а не самостійним продуктом. Це вимагає постійної підтримки та оновлень.

Враховуючи ці особливості, розробники веб-додатків повинні оптимізувати додаток для забезпечення стійкості, безпеки та ефективності, а також розглянути важливість захисту інтелектуальної власності та постійної підтримки продукту.

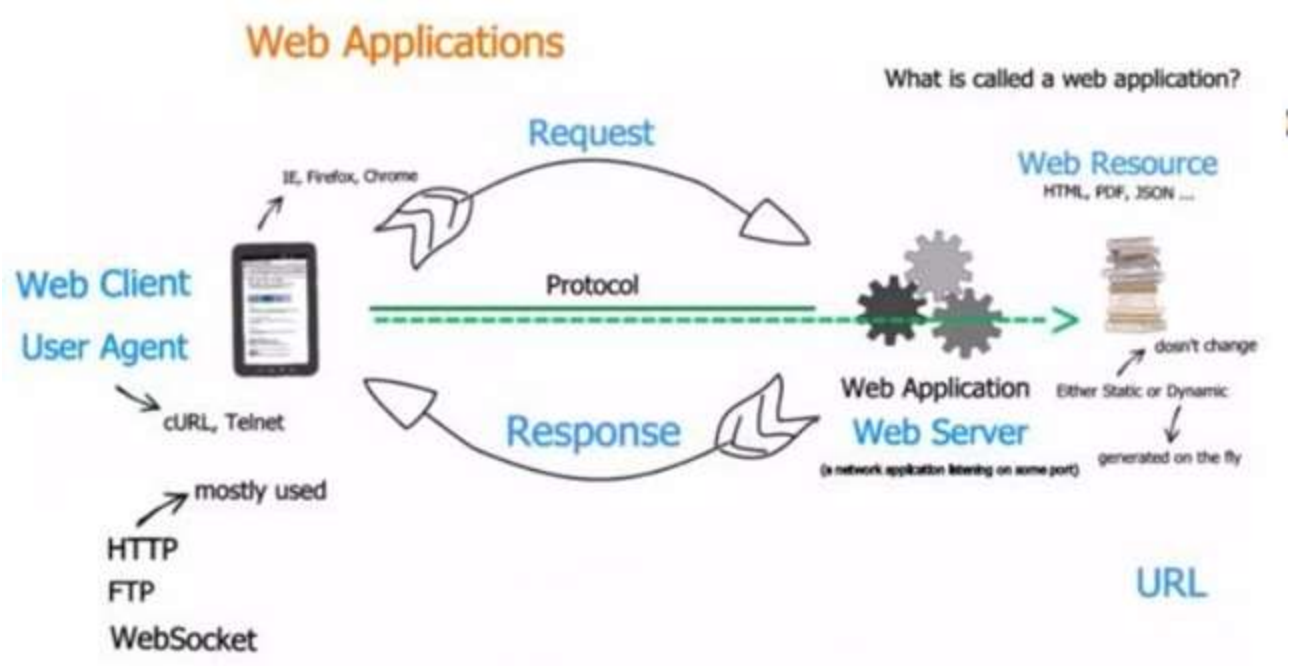


Рис. 4.1. Схема функціонування веб застосунку

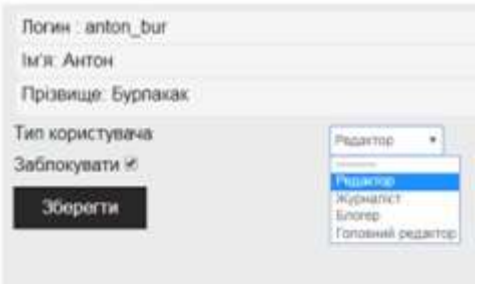


Рис. 4.2. Ілюстрація списку вибору ролі користувача

Для входу в систему, головний редактор має призначити ролі користувачам. У кабінеті головного редактора є пункт меню "Ролі користувачів", де він може вибрати користувача та назначити йому відповідну роль. Також у цьому пункті меню головний редактор може переглянути всіх активних та неактивних користувачів, включаючи тих, які ще не підтвердили свою електронну пошту або були заблоковані (рис. 4.3).

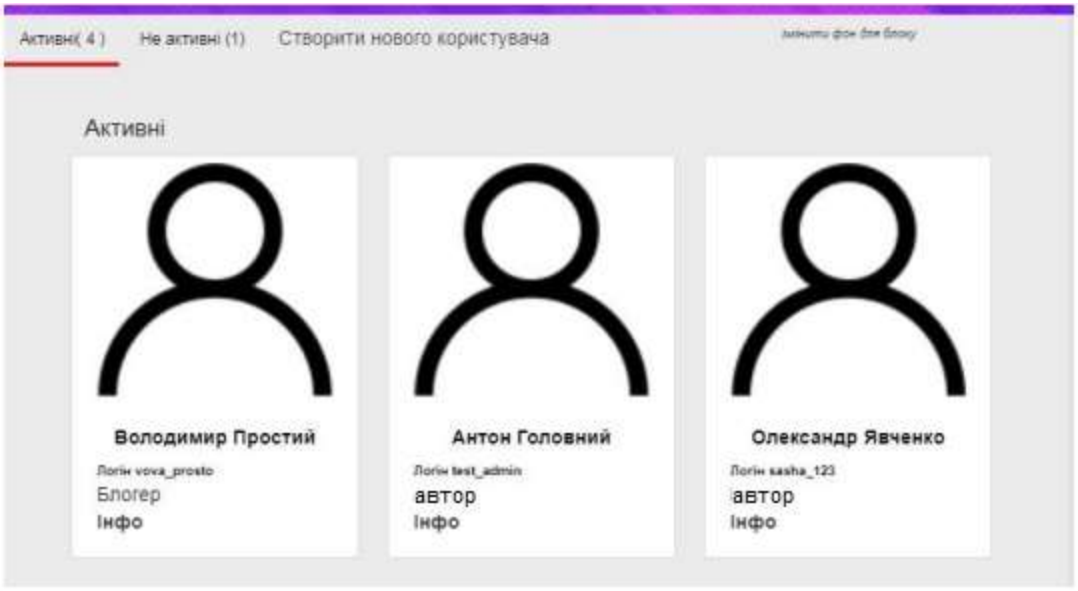


Рис. 4.3. Список всіх активних користувачів

Після того, як користувач має статус в системі, він може виконати вхід і отримати доступ до свого робочого простору. Для цього він повинен заповнити просту форму входу, вказавши свій логін і пароль.

Рис. 4.4. Форма авторизації користувача

Алгоритм роботи модуля авторизації (рис. 4.4) включає наступні кроки:

1. Користувач вводить свої дані для входу.
2. Виконується HTTP-запит методом POST на сервер.
3. Функція-контролер перевіряє коректність введених даних.
4. Якщо дані введено правильно, виконується запит в базу даних для отримання типу користувача в системі.
5. Якщо вхід до системи супроводжується помилками, користувач отримує відповідне повідомлення про помилку.

6. Далі генерується динамічне URL та перенаправляється до відповідного кабінету користувача.

Рис. 4.5. UML-діаграма алгоритму авторизації користувача в системі

Після успішної авторизації в системі, користувач отримує доступ до свого особистого кабінету, в якому доступні наступні пункти навігації (рис. 4.6):

1. Користувацький профіль: Користувач може переглядати та редагувати свої особисті дані, такі як ім'я, фотографія, контактна інформація тощо.

2. Налаштування: Користувач може налаштовувати різні параметри свого облікового запису, такі як налаштування конфіденційності, повідомлення, електронна пошта, пароль тощо.

3. Написати публікацію: Користувач може створювати нові публікації або писати нові матеріали, які будуть опубліковані на платформі.

4. Список публікацій: Користувач може переглядати список своїх публікацій, включаючи опубліковані матеріали, чернетки або видалені публікації.

Ці пункти навігації дозволяють користувачу легко переміщатися по особистому кабінету та виконувати необхідні дії пов'язані з керуванням своїм обліковим записом та створенням контенту.

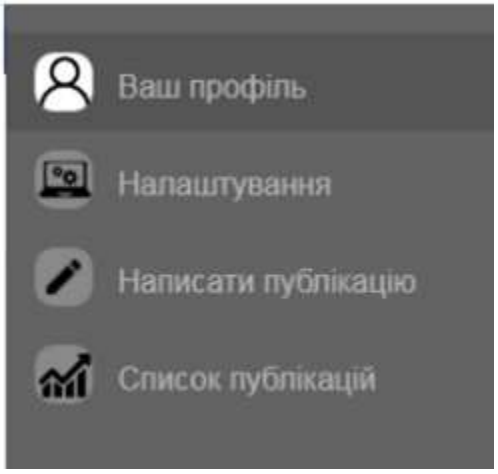


Рис. 4.6. Пункти меню для налаштування автора

На сторінці "Ваш профіль" користувач може переглядати свою особисту інформацію, а також загальну статистику для всіх опублікованих публікацій. Для зміни особистої інформації користувач може натиснути на іконку "редагування особистої інформації". У цьому розділі користувач має можливість:

1. Заповнити посилання на свої профілі в соціальних мережах, щоб інші користувачі могли зв'язатися з ним через ці канали.

2. Змінити своє фото, щоб оновити зображення, яке відображатиметься в профілі.

3. Вибрати тип представлення на сайті: публікуватися під повним ім'ям або використовувати псевдонім.

Ці опції дозволяють користувачеві налаштувати свій профіль та особисту інформацію на свій смак і надати додаткові дані для зв'язку та ідентифікації.

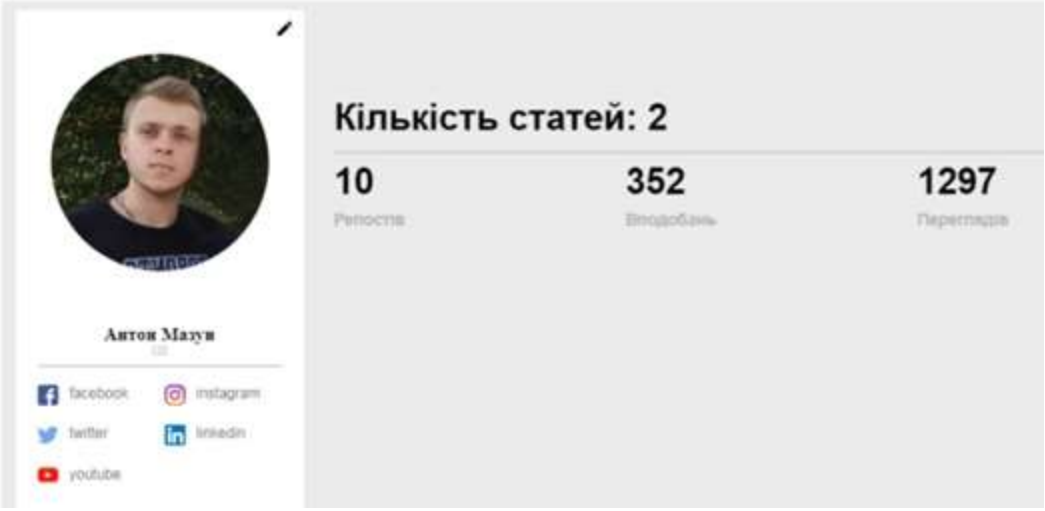


Рис. 4.7. Загальна статистика та особиста інформація користувача

На кожній сторінці веб застосунку читач має можливість перейти до публікацій певної категорії, всі категорії знаходяться у верхньому блоці сайту.

Рис. 4.8. Форма для початку спілкування автора з певної тематики

Рис. 4.9. Діалогове вікно з архівного спілкування автора

На головній сторінці веб-застосунку розміщено чотири найсвіжіші новини у верхньому блоку, що виділені окремо. Далі, на сторінці, розміщені інші новини у вигляді різнотипових блоків залежно від їхнього типу та займаної ширини основного блоку. Розміщення новин відбувається за таким принципом:

- Блок шириною 3/3 всієї ширини
- Три блоки шириною 1/3
- Один блок шириною 1/3 та другий блок шириною 2/3
- Три блоки шириною 1/3
- Один блок шириною 4/4
- Шість блоків шириною 1/3 у два рядки

Новини підгружаються циклічно з використанням функції "lazy load". Коли користувач прогортає до кінця сторінки, відбувається асинхронний ajax-запит для підвантаження додаткових новин, які динамічно додаються на сторінку. Ця функція має обмеження на кількість запитів - максимум 60 новин.

На головній сторінці також присутній блок, що містить останні події доби та тижня, з можливістю фільтрування за певними критеріями.

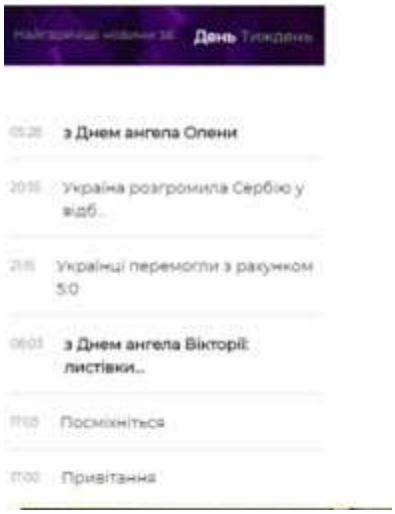


Рис. 4.10. Блок останніх новин за добу / тиждень від автора

При натисканні на заголовок або зображення статті, користувач буде перенаправлений на окрему сторінку, де він зможе переглянути повний текст статті, включаючи вкладені медіа-файли. На цій сторінці також будуть відображені дані про автора статті, такі як ім'я автора та його інформація, а також кількість переглядів та репостів статті.

Додатково, на сторінці статті буде присутній блок з новинами за добу або тиждень, які можуть бути пов'язані з темою статті.

Авторизовані користувачі матимуть можливість залишати коментарі під статтею або відповідати на існуючі коментарі. Це дозволить створити обговорення та залучити користувачів до взаємодії та обміну думками на тему статті.

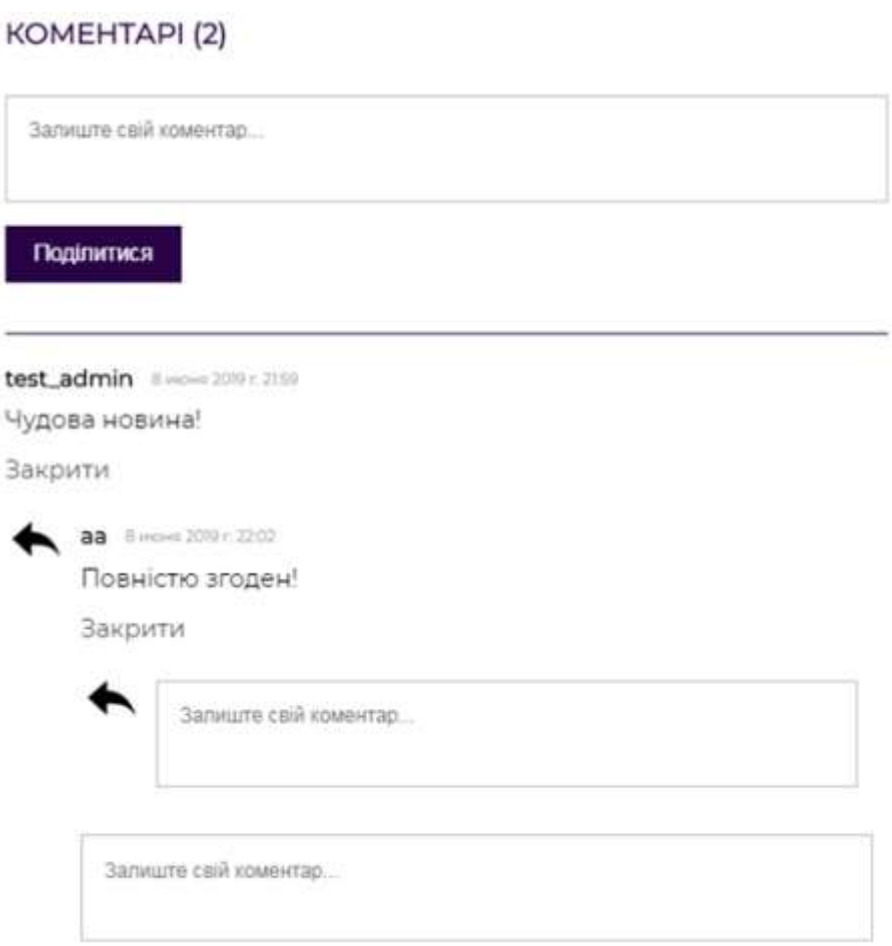


Рис. 4.11. Блок коментарів від читача

Так, при розробці компоненту було враховано різні розміри екранів та гаджетів, які користувачі можуть використовувати для доступу до веб-додатку. Адаптивне розташування блоків дозволяє оптимально відображати контент на різних пристроях, забезпечуючи зручне користування незалежно від розміру екрану.

На мобільних пристроях блоки будуть розташовуватись в спеціально адаптованому форматі, щоб забезпечити оптимальне використання екранного простору та зручну навігацію. Таким чином, користувачі зможуть зручно переглядати і взаємодіяти з контентом навіть на невеликих екранах своїх мобільних пристроїв.

Такий підхід дозволяє забезпечити зручну та доступну взаємодію з веб-додатком незалежно від пристрою, який використовується користувачем.

ВИСНОВКИ

1. На початку роботи сказано, що веб застосунки являють собою об’єднання різних способів представлення інформації. Зазвичай під ним мається на увазі об’єднання в комп’ютерній системі таких засобів представлення інформації, як текст, звук, графіка, мультиплікація, відеозображення і просторове моделювання. При вивченні технології створення веб застосунків будується сценарій, в якому описується, як вони будуть створюватися. У зв’язку з цим логічно припустити, що кожен веб застосунок для інтерактивного спілкування складається з різних компонентів (різних тематик).

2. Наголошено, що мобільні додатки надають своїм власникам різні функціональні можливості, такі як: швидкий зв’язок з будь-ким; потужні мобільні процесори; переносна камера, телефонна книга.

3. На сьогоднішній день найпопулярнішими платформами для розробки мобільних додатків є Android та iOS (iPhone). Ці платформи надають безкоштовні та легкі у засвоєнні інструменти для розробки, що дозволяє розробникам постійно оновлювати та розширювати свої мобільні додатки. Крім того, існують інші платформи, такі як Python, Java ME та BlackBerry, які також можуть використовуватись для розробки мобільних додатків, але вони не мають такої широкої популярності, як Android та iOS.

4. Мови програмування використовуються для взаємодії з комп'ютерами та іншими електронними пристроями. Кожна мова програмування має свої особливості, синтаксис та можливості. Наприклад, для розробки додатків для платформи Android однією з популярних мов програмування є Java, а для розробки веб-додатків існують різні фреймворки, такі як Spring (Java), Yii (PHP), Laravel (PHP) та Ruby on Rails (Ruby). Враховуючи різноманітність мов програмування, розробники можуть обрати ту, яка найкраще підходить для їх потреб та має популярні фреймворки для розробки додатків на конкретній платформі.

5. В практичній частині дослідження описано етапи розробки додатку інтерактивного спілкування за допомогою фреймворку django. Описано функціональні та нефункціональні вимоги до розробки. Наведено діаграму класів та цілей веб застосунку для інтерактивного спілкування між авторами та читачами. Приведено програмну реалізацію веб застосунку та перевірено його працездатність.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Азарян А. А., Карабут Н. О. Основи алгоритмізації та програмування: Навчальний посібник. Кривий Ріг: Вид-во ОктанПринт, 2014. 308 с.
2. Алгоритми та структури даних / уклад. О. В. Щербаков, Ю. Е. Парфьонов, В. М. Федорченко. Харків: ХНЕУ ім. С. Кузнеця, 2017. 58 с.
3. Беллиньясо М. Розробка Web-додатків в средовищі Visual Studio Code: Website Programming: М.: «Диалектика», 2007. С. 640.
4. Боднарчук І. О. Експертна система проектування архітектури програмного забезпечення: О. Г. Харчеко, І. О. Боднарчук, В. В. Яцишин, 2013. 210 с.
5. Брайн Харді, Білл Філліпс Android. Програмування для професіоналів. СПб.: Пітер, 2016. 640 с.
6. Глібовець Н. М. Створення рекомендаційної системи навчального типу з використанням фреймворку. *Проблеми інтелектуалізації комп'ютера: зб. ст. Інститут кібернетики ім. В. М. Глушкова НАН України*. К., 2012. С. 176.
7. Дронов В. А. Django практика створення Web-сайту на Python. К.: БХВ-Столиця, 2016. 528 с.
8. Метод багатокритеріальної оптимізації програмної архітектури на основі аналізу компромісів [Текст] / Харченко О. Г., Боднарчук І. О., Галай І. О. // Інженерія програмного забезпечення. 2012. № 3–4 (11–12). С. 5–11.
9. Назаров Д. Е., Драничников П. С. Клієнт-серверне взаємодія інтелектуальних мобільних пристроїв. *Комп'ютерні та інформаційні науки*. К.: Прогрес, 2001. С. 120–220.
10. Рахімов Б. К. Оптимальний алгоритм взаємодії інформаційного ресурсу з мобільними додатками. *Молодий вчений.* 2016. № 4 (108). С. 158–162.
11. Форсьє Д. Django. Розробка веб-додатків на Python. Символ, 2010. 456 с.
12. Харченко О. Г. Експерта система проектування архітектури програмного забезпечення. *Комп'ютерні технології друкарства.* № 29. 2013. С. 10–26.
13. Офіційна документація фреймворку UIKit. – Режим доступу: https: // developer. apple. com / library / ios / documentation / UIKit / Reference / UIK it_Framework. (дата звернення: 05.06.2023).

14. Java URL: https: // www. theserve rside. com / definition / Java. (дата звернення: 05.06.2023).
15. Ruby URL: https: // www. geeksf orgeeks. org / ruby-introduction-to-multi-threading / ?ref=rp. (дата звернення: 05.06.2023).
16. Веб-фреймворк URL: https: // www. goodfirms. co / glossary / web-framework / (дата звернення: 05.06.2023).
17. Фреймворк Laravel URL: https: // medium. com/@ronak8036 / top-features-of-laravel-php- framework – 9082ac9eb4f2. (дата звернення: 05.06.2023).
18. Опис стандарту IDEF0 – URL: http: // syque. com / quality_tools / tools / Tools19. htm. (дата звернення: 05.06.2023).
19. Порівняння фреймворків Laravel та Ruby on Rails URL: https: // www. educba. com / laravel-vs-ruby-on- rails / ?source =leftnav. (дата звернення: 05.06.2023).
20. Фреймворк Laravel URL: https: // laravel. com / docs / 8. x / facades (дата звернення: 05.06.2023).
21. Фреймворк [Електронний ресурс]. Режим доступу: https: // cocosolution. com / en / what-is-a-framework / (дата звернення: 05.06.2023).
22. Фреймворк Yii [Електронний ресурс]. Режим доступу: https: // www. yiiframework. com / doc / guide / 2.0 / ru / structure-overview. (дата звернення: 05.06.2023).
23. Фреймворк Yii URL: https: // www. yiiframework. com / doc / guide / 1.1 / ru / quickstart. first-app (дата звернення: 05.06.2023).
24. Фреймворк Laravel URL: https: // solidbrain. com. ua / ua / news / rozrobka-php-s-laravel-porivniannia-pliusiv-i- minysiv (дата звернення: 05.06.2023).
25. Фреймворк. URL: http s: // techterms. com / definition / framework (дата звернення: 05.06.2023).

ДОДАТОК 1

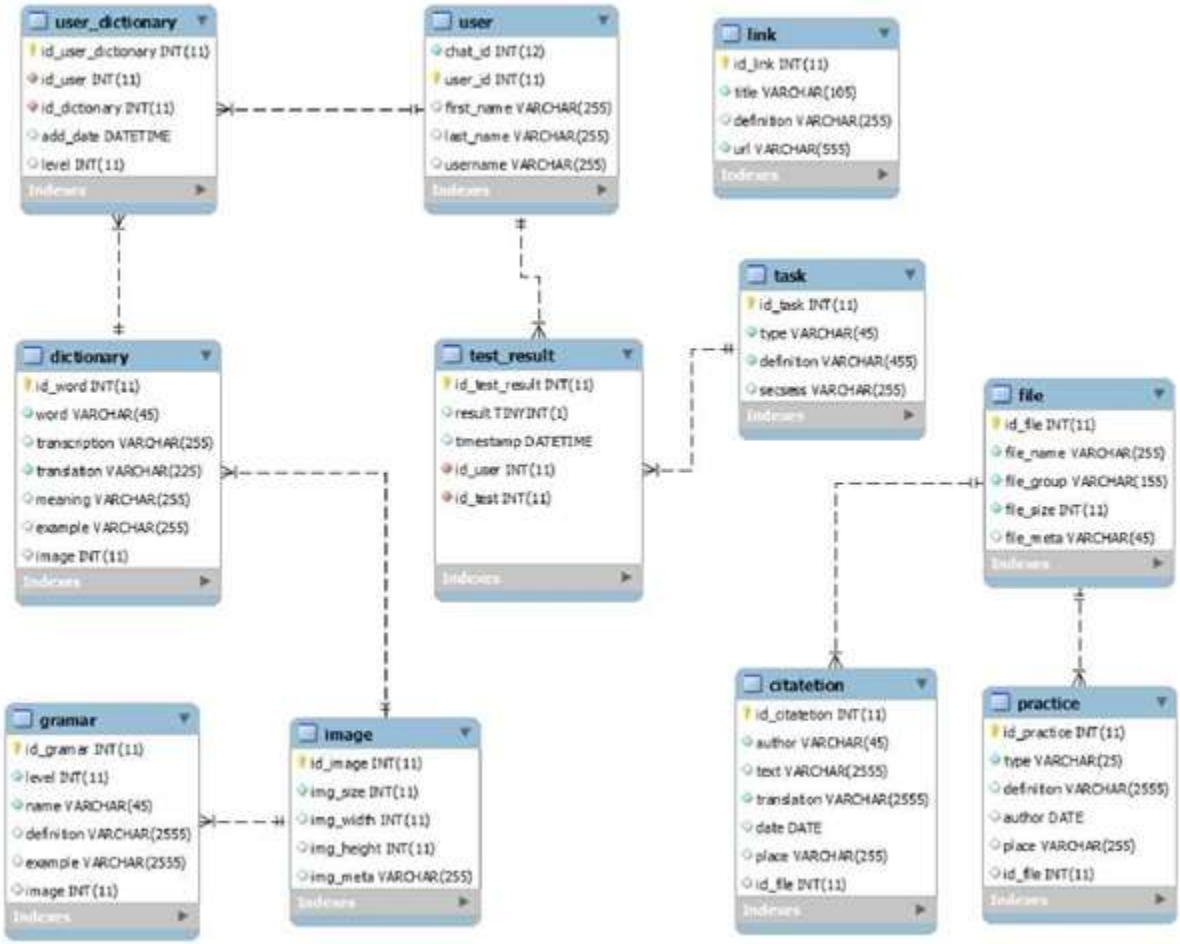
Веб застосунок для інтерактивного спілкування між авторами та читачами

Фізична модель БД в вигляді EER діаграми

ІАЛЦ.467200.004 Д1

Аркушів 1

Київ 2023 р



					ІАЛЦ.467200.004 Д1		
Вим	Лист	№ докум.	Підп.	Дата			
Виконав		Кузьмич А.А.			**Веб застосунок для інтерактивного спілкування між авторами та читачами** Фізична модель БД в вигляді EER діаграми	Літера	Аркуш 1 / Аркушів
Керівник		Пустовіт Л.М.					
Консульт.						КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-92	
Н-контр.		Виноградов Ю.М.					
Зав. каф.							

ДОДАТОК 2

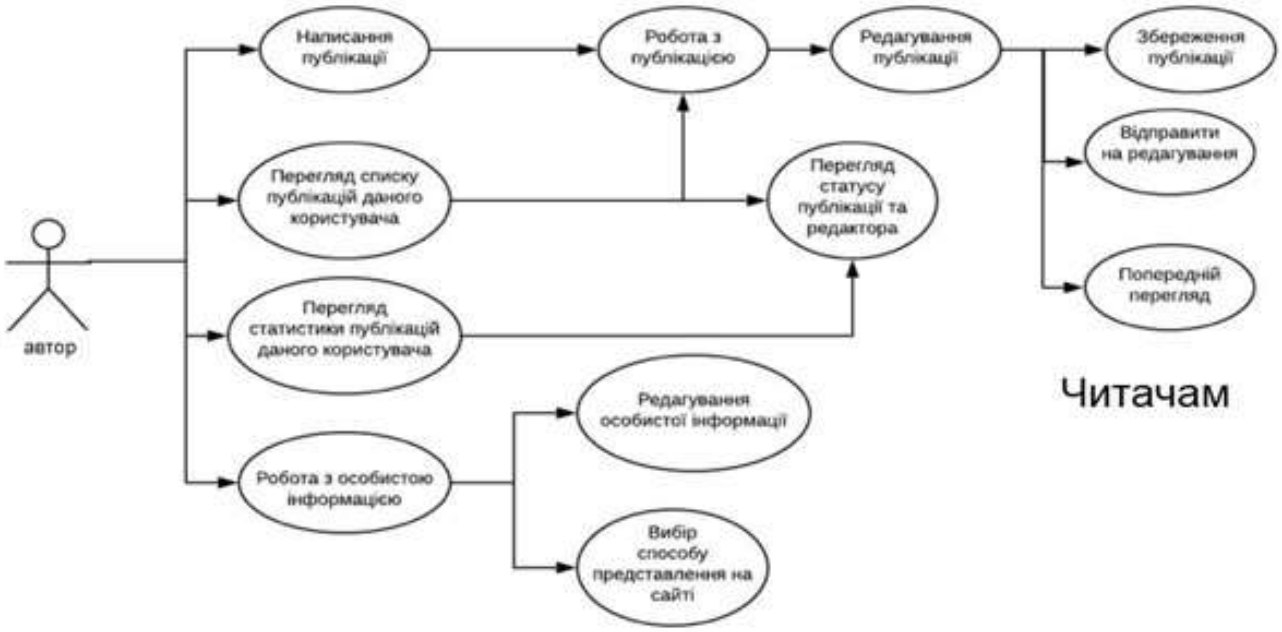
Веб застосунок для інтерактивного спілкування між авторами та читачами

UML-діаграма можливостей автора в системі

ІАЛЦ.467200.004 Д2

Аркушів 1

Київ 2023 р



					ІАЛЦ.467200.004 Д2		
Вим	Лист	№ докум.	Підп.	Дата			
Виконав		Кузьмич А.А.			**Веб застосунок для інтерактивного спілкування між авторами та читачами** UML-діаграма можливостей автора в системі	Літера	Аркуш 1 / Аркушів
Керівник		Пустовіт Л.М.					
Консульт.						КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-92	
Н-контр.		Виноградов Ю.М.					
Зав. каф.							

ДОДАТОК 3

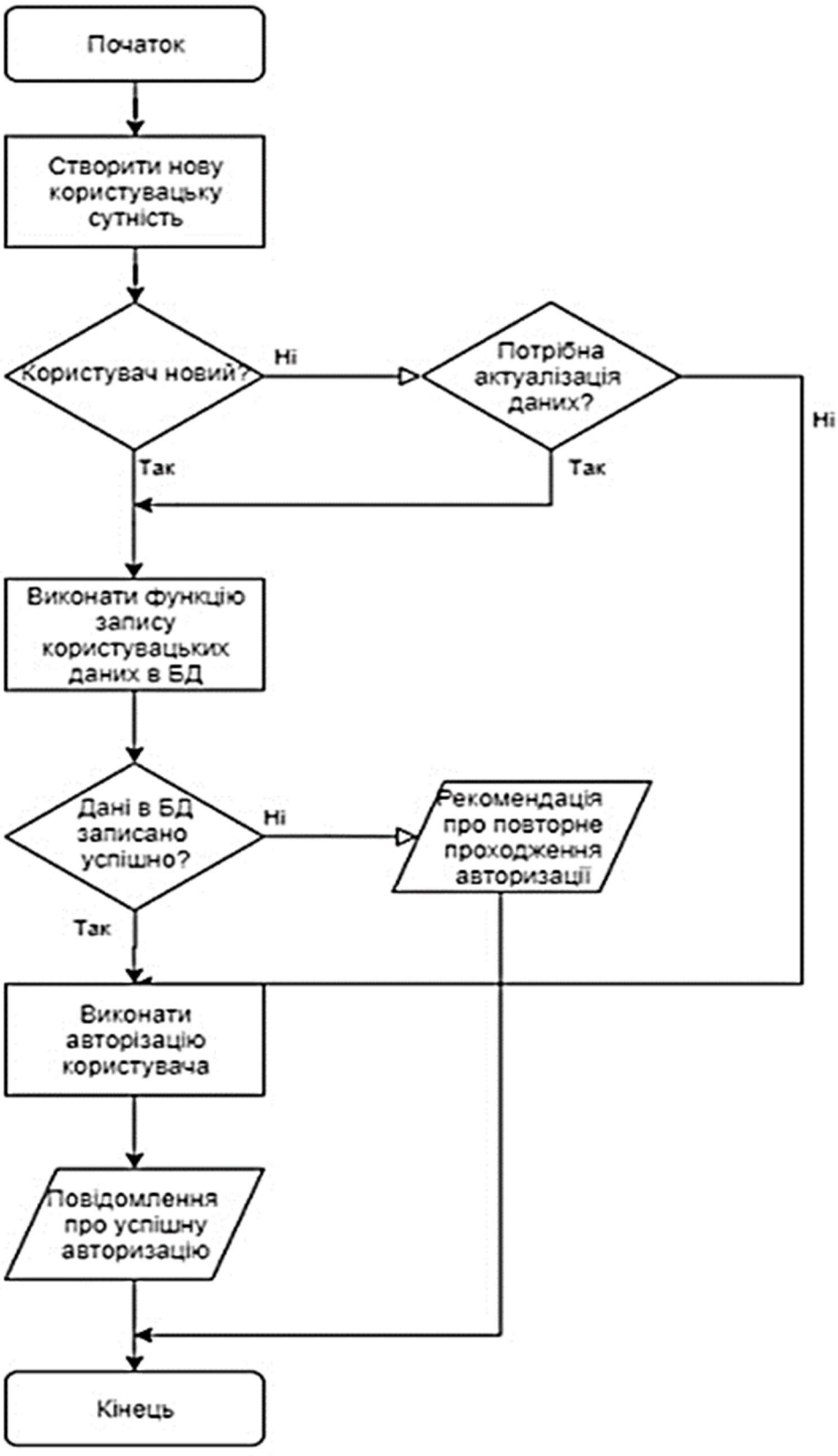
Веб застосунок для інтерактивного спілкування між авторами та читачами

Алгоритм роботи модуля обробки нового користувача

ІАЛЦ.467200.004 Д3

Аркушів 1

Київ 2023 р



					ІАЛЦ.467200.004 Д3		
Вим	Лист	№ докум.	Підп.	Дата			
Виконав		Кузьмич А.А.			**Веб застосунок для інтерактивного спілкування між авторами та читачами** Алгоритм роботи модуля обробки нового користувача	Літера	Аркуш 1 / Аркушів
Керівник		Пустовіт Л.М.					
Консульт.						КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-92	
Н-контр.		Виноградов Ю.М.					
Зав. каф.							

ДОДАТОК 4

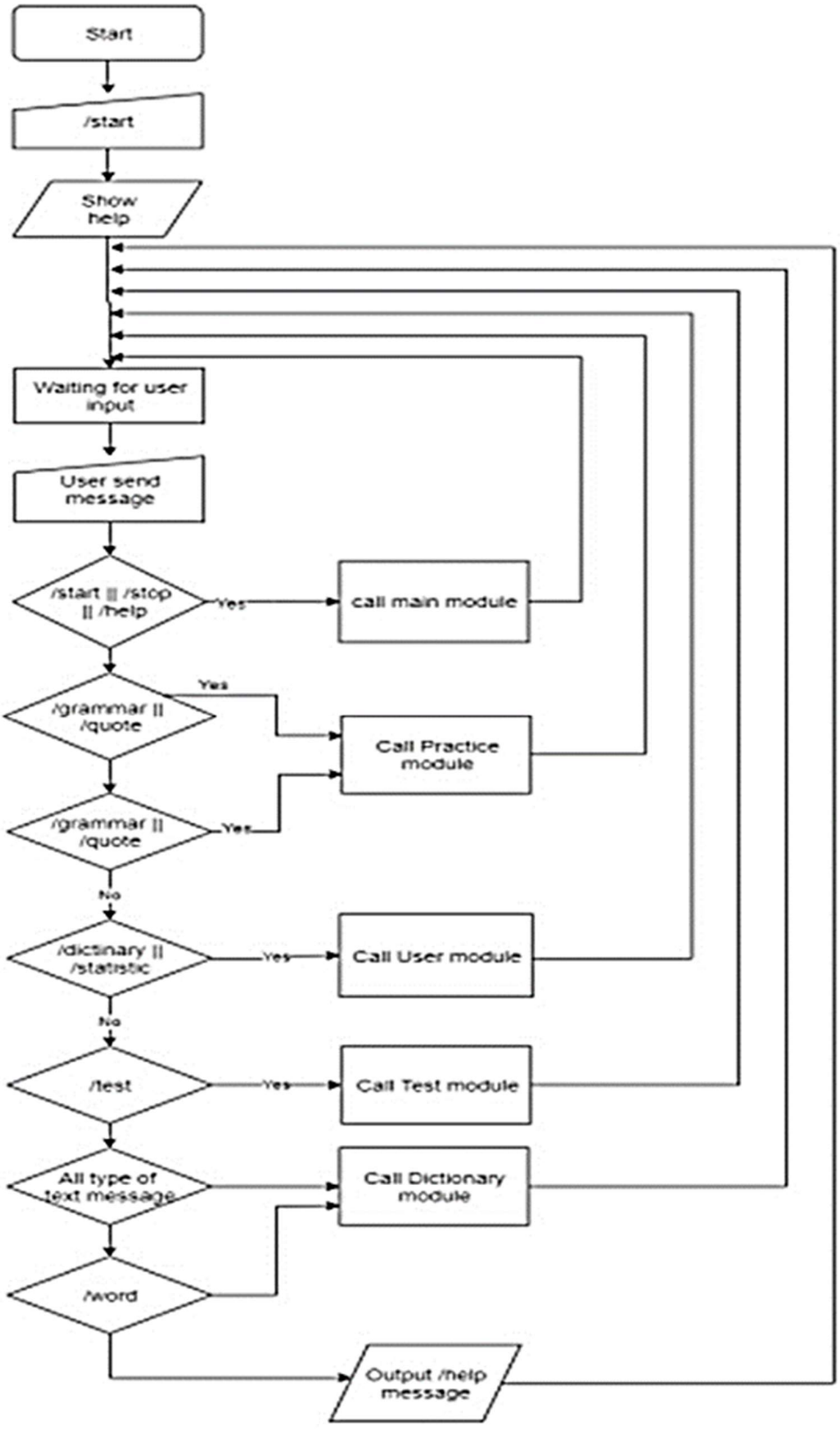
Веб застосунок для інтерактивного спілкування між авторами та читачами

Командна мапа користувача

ІАЛЦ.467200.004 Д4

Аркушів 1

Київ 2023 р



					ІАЛЦ.467200.004 Д4		
Вим	Лист	№ докум.	Підп.	Дата			
Виконав		Кузьмич А.А.			**Веб застосунок для інтерактивного спілкування між авторами та читачами** Командна мапа користувача	Літера	Аркуш 1 / Аркушів
Керівник		Пустовіт Л.М.					
Консульт.						КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-92	
Н-контр.		Виноградов Ю.М.					
Зав. каф.							

ДОДАТОК 5

Веб застосунок для інтерактивного спілкування між авторами та читачами

Фрагмент коду для модулю models.py

ІАЛЦ.467200.004 Д5

Аркушів 1

Київ 2023 р

```python
from django.db import models
# from categories.models import Categories
from categories.models import Categories
from django.contrib.auth.models import User
from django.contrib.auth import get_user_model
from ckeditor.fields import RichTextField
from ckeditor_uploader.fields import RichTextUploadingField
from manage_user.models import EditorUser, AdminMain
from django.conf import settings
import django
import datetime
from django.utils import timezone

# from elasticsearch_dsl import (
#     DocType,
#     Date,
#     Keyword,
#     Text,
#     Boolean,
#     Integer
# )

# Create your models here.

STATUS_SITE = (
    ('safe', 'надійне джерело'),
    ('unsafe', 'ненадійне джерело')
)
STATUS_ARTICLE = (
    ('inactive', 'Не опрацьована'),
    ('edited', 'На опрацюванні'),
    ('to_main_admin', 'У головного редактора'),
    ('back_to_user', 'Повернено автору'),
    ('done', 'В публікацію')
)


# def get_sentinel_user():
#     return get_user_model().objects.get_or_create(username='deleted')[0]


class Photo(models.Model):
    title = models.CharField(max_length=255, blank=True)
    file = models.ImageField(upload_to='photos_uploads/')
    uploaded_at = models.DateTimeField(auto_now_add=True)

    def delete(self, *args, **kwargs):
        # You have to prepare what you need before delete the model
        storage, path = self.file.storage, self.file.path
        # Delete the model before the file
        super(Photo, self).delete(*args, **kwargs)
        # Delete the file after the model
        storage.delete(path)
```

					ІАЛЦ.467200.004 Д5			
Вим	Лист	№ докум.	Підп.	Дата				
Виконав		Кузьмич А.А.			Веб застосунок для інтерактивного спілкування між авторами та читачами	Літера	Аркуш	Аркушів
Керівник		Пустовіт Л.М.					1	
Консульт.					Фрагмент коду для модулю models.py	КПІ ім. Ігоря Сікорського, ФІОТ, ІВ-92		
Н-контр.		Виноградов Ю.М.						
Зав. каф.								