

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

_____ Наталія АУШЕВА

«___» _____ 2022 р.

Дипломна робота

на здобуття ступеня бакалавра

спеціальності 122 «Комп'ютерні науки»

освітня програма «Комп'ютерний моніторинг та геометричне

моделювання процесів і систем»

на тему: «Прототип соціальної мережі»

Виконала:

студентка IV курсу, групи ТР-82

Бойченко Дарина Олександрівна

Керівник:

рецензент

Швайко Валерій Григорович

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студентка _____

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший

спеціальність 122 «Комп’ютерні науки»

освітня програма «Комп’ютерний моніторинг та геометричне моделювання

процесів і систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Наталія АУШЕВА

(підпис)

” ” _____ 2022р.

ЗАВДАННЯ

на дипломну роботу студенту

_____ Бойченко Дарині Олександрівні

1.Тема роботи: «Прототип соціальної мережі»

керівник роботи: _____ асист. Швайко Валерій Григорович

(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ”25” травня 2020р. № 1168-с

2.Строк подання студентом роботи: 10.06.22

3.Вихідні дані до роботи: мова програмування Python, додаток dbgram

4.Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): проаналізувати існуючі програмні засоби, розробити реляційну та просторову базу даних, розробити код програми, написати інструкцію керування для користувача

5.Перелік ілюстративного матеріалу:

6.Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7.Дата видачі завдання "14" _____ грудня _____ 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	14.12.2020	
2.	Вивчення та аналіз задачі	15.12.2020-15.01.2021	
3.	Розробка архітектури та загальної структури системи	16.01.2021-31.01.2021	
4.	Розробка структур окремих підсистем	01.02.2021-28.02.2021	
5.	Програмна реалізація системи	01.03.2021-30.04.2021	
6.	Оформлення пояснювальної записки	01.05.2021-10.05.2021	
7.	Захист програмного продукту	11.05.2021	
8.	Передзахист	25.05.2021	
9.	Захист		

Студент: _____ Бойченко Д.О.
(підпис) (прізвище та ініціали)

Керівник роботи: _____ Швайко В.Г.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка даного Web-додатку дозволить створити медіа-ресурс, спрямований на: спрощення соціальних взаємовідносин; забезпечення інформаційної безпеки; надання актуальної інформації.

Записка містить 48 сторінок, 11 рисунків.

ABSTRACT

The development of this Web-application will create a media resource aimed at: simplifying social relationships; information security; providing up-to-date information.

The note contains pages, pictures.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ.....	6
ВСТУП.....	7
1 ПОСТАНОВКА.....	8
1.1 Задача.....	8
1.2 Необхідні умови для задачі.....	8
2. Опис предметної області.....	9
2.1 Соціальна мережа.....	9
2.2 Аналіз існуючих рішень.....	10
3. Засоби розробки.....	13
3.1 Хешування.....	13
3.2 Мови для веб-розробки.....	16
4. Опис програмної реалізації.....	18
4.1 Опис реалізації	18
4.2 Аналіз програмних засобів та компонентів необхідних для розробки програмного продукту.....	19
4.3 Розробка продукту.....	21
4.4 Засоби розробки програмного продукту.....	23
5. Робота користувача з програмною системою.....	27
5.1 Системні вимоги.....	27
5.2 Робота з програмою.....	27
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	36
ДОДАТОК А.....	38
ДОДАТОК Б.....	43

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API	Application programming interface
HP	Hewlett-Packard
HTML	Hypertext Markup Language
IT	Information Technology
STL	Standard Template Library
UML	Unified Modeling Language
КС	Комп'ютерна система
Мб	Мегабайт
ОС	Операційна система
MVC	Model-View-Controllerpattern

ВСТУП

Соціальна мережа (від англ. Social Network Service) - платформа, онлайн-сервіс або веб-сайт, призначені для побудови, відображення та організації соціальних взаємин.

Соціальні мережі – основна причина, через яку сьогодні зростає кількість часу, що проводиться в Інтернеті. Головні їх переваги – можливість користувачів заявляти про свої інтереси, та розділяти їх із оточуючими. І це дає підстави стверджувати, що соціальні мережі є не лише засобом для спілкування, а й потужним маркетинговим інструментом, більше того, дослідники вважають, що незабаром вони стануть необхідним інструментом ведення діяльності. Соціальні мережі є майданчиком для неформального спілкування, допомагають створювати нову музику, служать серйозним інструментом для пошуку співробітників і партнерів.

У соціальної мережі, що розробляється, повинна бути висока функціональність. Редагування списку друзів та особистої сторінки на розсуд користувача, надсилання особистих повідомлень, ведення особистого блогу, створення груп, завантаження фотографій, музики та відео - необхідні та обов'язкові функції сучасної соціальної мережі. Важливу роль у функціональності сайту відіграє зручний пошук та ідентифікація користувачів. Важливим елементом для соціальної мережі є дизайн, він не повинен відволікати користувачів від спілкування та використання сайту. Яскравий стиль дратуватиме, і втомлюватиме користувача, тому при створенні дизайну рекомендується використовувати переважно постільні та спокійні тони.

1 ПОСТАНОВКА

1.1 Задача

Соціальна мережа - соціальна структура що складається з групи вузлів, якими є соціальні об'єкти (люди), та зв'язків між ними (соціальних взаємин). В даний час в мережі Інтернет існує безліч різних реалізацій соціальних мереж, подібних як до функціональності, так і за принципом надання користувачів.

Система спілкування користувачів, що розробляється в рамках дипломного проектування, з урахуванням їх географічного розташування є потужним інструментом пошуку та спілкування.

Цей проект як спрощує процес спілкування користувачів, а й виводить його більш високий рівень соціальних взаємин. Користуючись цією системою, користувач розширює і поглиблює свої географічні знання, адже крім спілкування, система здатна відобразити і підписати географічні об'єкти, які є пам'яткою того чи іншого міста.

1.2 Необхідні умови для задачі

Для досягнення поставленого завдання необхідна потужна серверна частина Web-програми, здатна безвідмовно працювати з великою кількістю користувачів, а також виконувати покладені на неї функції.

Серверна частина Web-додатка повинна забезпечувати можливість:

- забезпечення зв'язку з клієнтською частиною додатка;
- надання клієнтської частини необхідних їй даних;
- обробка та зберігання даних у базі даних.

Частина програми, що розробляється, повинна бути інтегрована в загальну систему, після чого її можна буде використовувати для виконання покладених на неї функцій.

2. Опис предметної області

2.1 Соціальна мережа

Соціальна мережа (від англ. Social Network Service) - платформа, онлайн-сервіс або веб-сайт, призначені для побудови, відображення та організації соціальних взаємин.

Кінець XX – початок XXI століття відзначено бурхливим розвитком Всесвітньої павутини. У міру її зростання набувають широкого поширення та різноманітні новітні інформаційні технології - використання Інтернету стає звичним у багатьох сферах життя та діяльності людини, раніше далеких від активного використання ЕОМ.

Передумовами виникнення соціальних мереж стали гостьові книги (web-додатки, що складаються зі списку повідомлень, показаних від останніх до перших, які може залишити кожен відвідувач), форуми (повідомлення групуються тематично, кожен відвідувач може залишити повідомлення на задану тему у відповідь на попереднє) та блоги (кожен учасник веде журнал, аналогічний особистому щоденнику, його повідомлення сортуються у хронологічному порядку, інші відвідувачі можуть залишати коментарі до повідомлень, у своїй користувач може створювати списки «друзів» чи обмежити доступом до свого журналу).

Популярність в Інтернеті соціальні мережі почали завойовувати в 1995 році з появою американського порталу Classmates.com («Однокласники» є його російським аналогом). Проект виявився дуже успішним, що у наступні кілька років спровокувала поява не одного десятка аналогічних сервісів.

Соціальні мережі – основна причина, через яку сьогодні зростає кількість часу, що проводиться в Інтернеті. Головні їх переваги – можливість користувачів заявляти про свої інтереси, та розділяти їх із оточуючими. І це дає підстави стверджувати, що соціальні мережі є не лише засобом для спілкування, а й потужним маркетинговим інструментом, більше того,

дослідники вважають, що незабаром вони стануть необхідним інструментом ведення діяльності. Соціальні мережі є майданчиком для неформального спілкування, допомагають створювати нову музику, служать серйозним інструментом для пошуку співробітників і партнерів.

Всі соціальні мережі мають низку спільних рис:

- наявність реєстрації (тобто облікового запису) користувача, при реєстрації користувач вказує деяку інформацію про себе, за якою його можна ідентифікувати;
- вхід у систему за допомогою відкриття сеансу (користувач вказує ім'я та підтверджує свою особу введенням пароля);
- настроювання оточення (наприклад, вказівка додаткових даних про себе, своїх інтересів).

Соціальні мережі є потужним інструментом маркетингових досліджень, оскільки користувачі добровільно публікують інформацію про себе, свої погляди, інтереси, уподобання і так далі. З огляду на це рекламодавці можуть дуже чітко визначати, яких саме користувачів зацікавить їх оголошення, і направити свої рекламні оголошення конкретним користувачам, залежно від інформації в їх профілях (вік, стать, місце проживання та інше).

Головне призначення соціальних мереж – здійснення соціальних взаємовідносин між користувачами Інтернету.

2.2 Аналіз існуючих рішень

Соціальна мережа - це медіа-ресурс який сприяє поліпшенню соціальних взаємовідносин користувачів Інтернет, а й засіб реклами. Для того щоб скласти уявлення про рівень існуючих соціальних мереж було проведено роботу з перегляду та аналізу переваг та недоліків. Нижче наводиться короткий опис кількох соц. мереж та порівняльний аналіз їх можливостей.

Соціальна мережа www.vk.com розробника П. Дурова. Відкриття сайту відбувається швидко, сторінка реєстрації та входу, зображена малюнку 1.1, має приємну колірну гамму, не завантажена громіздкими візуальними компонентами.

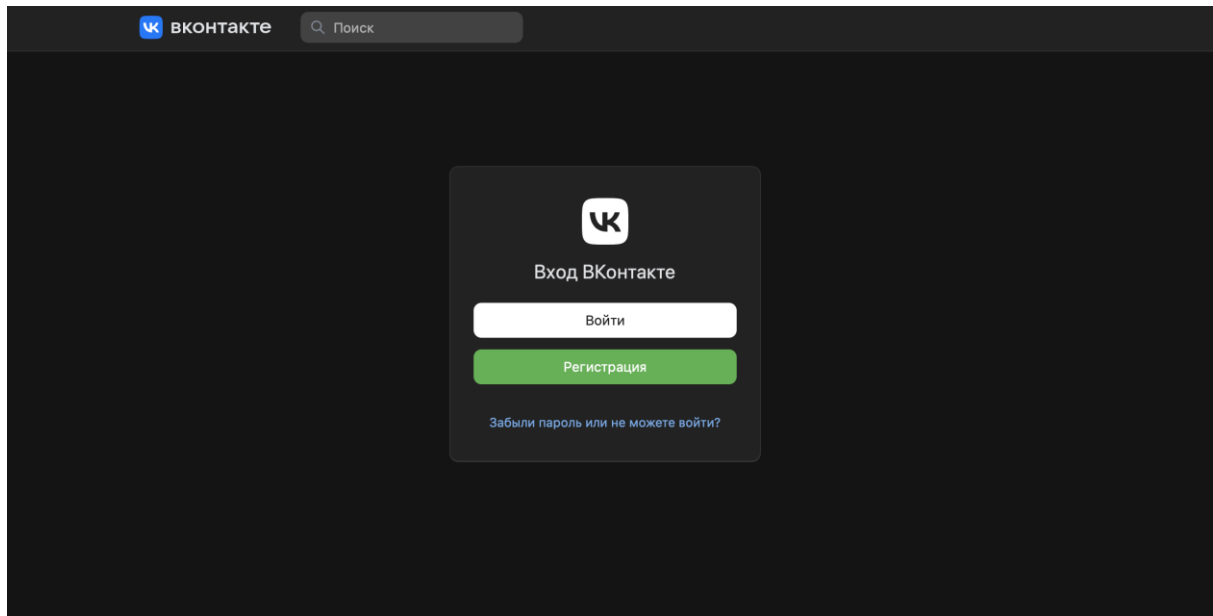


Рисунок 1.1 - Сторінка аутентифікації та реєстрації соціальної мережі

При натисканні на кнопку «Реєстрація» з'являється контекстне меню «Моментальна реєстрація».

Меню має простий дизайн, приємну колірну гамму, а також інтуїтивно зрозумілий інтерфейс. Сторінка реєстрації, за меню, затемняється для концентрації уваги користувача на процесі реєстрації. Варто відзначити зручність такої технології: відсутня переадресація на сторінку наступного етапу реєстрації, меню «Моментальна реєстрація» з'являється миттєво, не вимагає оновлення або переходу на іншу сторінку.

Завершивши процес реєстрації, автоматично створюється особиста сторінка користувача з урахуванням його даних. Сторінка користувача соціальної мережі «В Контакті» виконана в синьо-сіро-білих тонах, не завантажена візуальними компонентами, дуже ергономічна та приємна на

вигляд. Вона фактично розділена на 3 зони: посилання користувача (ліворуч), навігація (зверху) та особиста інформація користувача (посередині).

Плюси соціальної мережі "В Контакті":

- Приємний дизайн та ергономічність;
- Великі функціональні можливості: завантаження аудіо, відео, фото файлів тощо;
- Наявність можливості створення груп та подачі оголошень;
- Висока швидкість завантаження сторінок;

Мінуси:

- Посередня реалізація сторінки реєстрації;
- Велика кількість платного контенту (подарунки, голоси, рейтинг сторінки);
- Відсутність обмеження перегляду контенту, що має віковий рейтинг +18;
- Відсутність контролю інформації;

Соціальна мережа www.facebook.com розроблена М. Цукербергом.

Сторінка аутентифікації та реєстрації, показана малюнку 1.2, завантажується швидко та виконана в синьо-сірих тонах (з градієнтом білого), блок реєстрації виділено великим шрифтом, що сприяє концентрації уваги користувача на важливій для нього інформації.

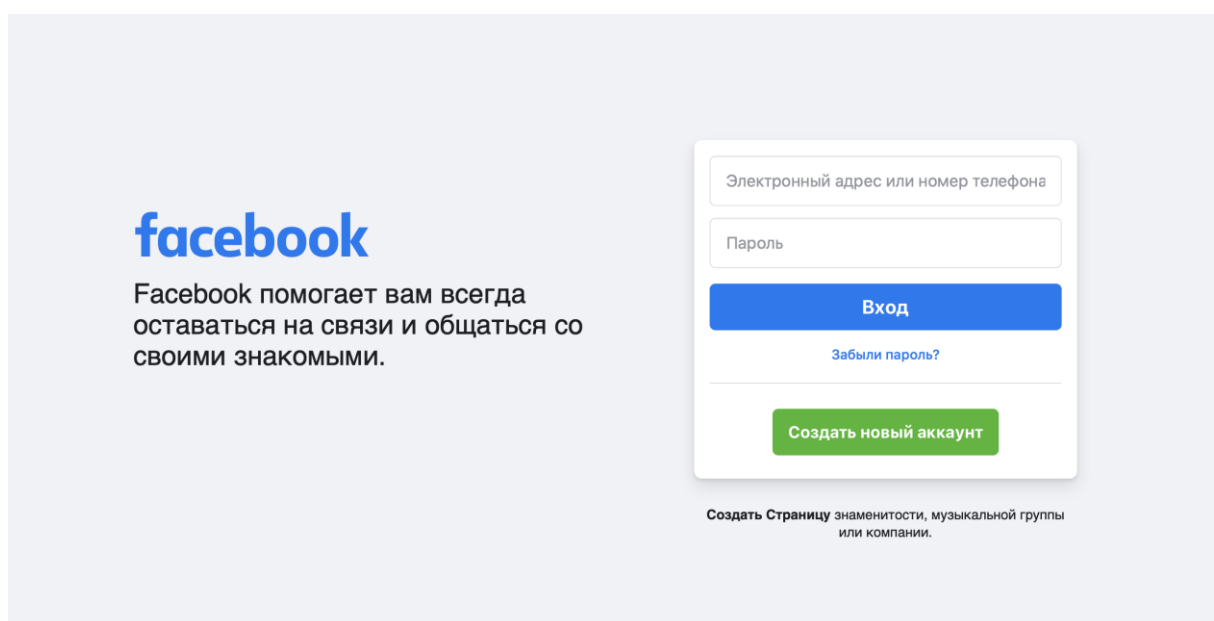


Рисунок 1.2 - Сторінка аутентифікації та реєстрації соціальної мережі

Блок реєстрації розроблений дуже зручно і дозволяє одразу ввести всю необхідну для користувача інформацію. Вказано посилання на «Послуги використання» та «Політику використання даних». Також є можливість реєстрації публічної особи або організації. Ця функція відкриває широкий спектр можливостей надання послуг з реклами. У процесі реєстрації надається допомога користувачеві у вигляді спливаючих підказок та попереджень.

Особиста сторінка користувача захащена зайвими візуальними компонентами. Загальну ергономічність сторінки можна оцінювати як середню. Незручно розташовані пошукові посилання. Варто відзначити невеликий розмір шрифту, що сприяє напрузі очей користувача, який проводить багато часу використовуючи цю соціальну мережу.

Плюси соціальної мережі «Фейсбук»:

- Зручна та приємна реєстрація;
- Можливість реєстрації організації або публічної особи;
- Велика кількість контенту та висока функціональність;
- Наявність окремого блогу;

Мінуси:

- Незручна конструкція особистої сторінки користувача;
- Невеликий розмір шрифту;
- Посередня ергономічність сторінки;
- Відсутність інтуїтивно зрозумілого інтерфейсу;

3. Засоби розробки

3.1 Хешування

Серед багатьох функцій PHP є функція MD5. В її основі алгоритм MD5 RSA Data Security, Inc, вона повертає так звані хеш-рядки, який є 32-значним

шістнадцятковим числом. Для кожного рядка він унікальний, ймовірність збігу його у різних рядків дуже мала.

Хешування – це зовсім не шифрування. Шифрування передбачає якийсь ключ, за допомогою якого шифр можна перевести в текст, що читається. У хешування такого ключа немає. Можна сказати, що це шифрування "в один бік", без можливості дешифрування.

Функцію MD5 дуже зручно використовуватиме зберігання паролів у базі сайту. Якщо паролі лежать у базі у явному вигляді, то у разі зламування такої бази хакер отримає доступ до цілої колекції паролів. У разі існує дві великі проблеми: по-перше, серед паролів можуть бути і паролі користувачів з адміністративними повноваженнями; по-друге, зазвичай користувачі застосують одні й той самий пароль скрізь (на форумах, на своїй скриньці електронної пошти тощо), тому не варто авторам хорошого сайту підставляти своїх користувачів.

На хороших сайтах паролі зберігають у хешованому вигляді. Коли користувач вводить свій пароль, його теж пропускають через md5 і порівнюють з хешем, що лежить в базі - якщо рядки збіглися, значить, користувач ввів саме той пароль, хеш якого лежить в базі.

Насправді, хешування дуже добре захищає пароль, але абсолютного захисту таки не дає. Розшифрувати його не можна, але можна підібрати методом банального перебору. Хеш 8-10-значного пароля, що зберігається в базі, дає не абсолютну, але дуже високий ступінь захисту.

MD5 алгоритм використовується в додатках криптографії та електронно-цифрових підписів для генерації ключа шифрування. Алгоритм розроблений, щоб бути досить швидким на 32-бітних системах і не вимагати великих обсягів пам'яті. MD5 є трохи повільнішим, ніж MD4, але є більш стійким до криптографічних атак. Далі під «словом» буде на увазі кількість інформації в 32 біти, а під «байтом» - 8 біт. Послідовність біт інтерпретується в природній формі як послідовність байт, де кожна група з 8 біт є окремим байтом, причому старший біт байта йде першим. Аналогічно представляється

послідовність байт, як послідовність слів, лише молодший байт йде першим. Передбачається, що як вхідний потік є потік даних N біт. N – невід'ємне ціле (можливо 0), не обов'язково кратне 8.

Для обчислення MD5 хеш-функції необхідно виконати наступні 5 кроків.

Крок 1: вирівнювання потоку.

Вхідний потік вирівнюється так, щоб його довжина стала конгруентною (порівняною) з 448 по модулю 512. Вирівнювання відбувається наступним чином: до потоку додається один біт «1», а потім біти «0» доти, поки довжина потоку не буде порівнянна з 448 по модулю 512. Вирівнювання відбувається завжди, навіть якщо довжина потоку була порівнянна з 448 по модулю 512.

Таким чином, до потоку додається мінімум 1 біт, максимум -512.

Крок 2: додавання довжини.

Якщо довжина потоку перевищує 2A64, додаються молодші 64 біт. Ці біти додаються як 2 32-бітові слова, молодше слово додається першим. Таким чином, на цьому кроці довжина потоку стає кратною 512 біт або 16 32-бітним словам. Крок 3: ініціалізація MD-буфера.

Буфер із 4 слів використовується для обчислення хеш-функції, який ініціалізується у відповідні значення. Крок 4: обробка потоку блоками 16 слів.

Визначаються чотири допоміжні функції, кожна з яких приймає три параметри розмірів у слово та робить результат – слово. Далі вхідний потік обробляється блоками 16 слів.

Крок 5: Висновок MD5.

Результат обчислення (хеш) представлений чотирма 32 бітними словами. 128-бітний MD5 хеш готовий.

Розробка соціальної мережі це тривалий та трудомісткий процес. У рамках написання дипломної роботи проект буде реалізований у демонстраційному вигляді. Створення веб-сайтів охоплює широкий асортимент засобів розробки, мов програмування та ін. У реалізації цього проекту будуть задіяні такі мови програмування: HTML, PHP, CSS, Python.

Нижче наведено список, короткий опис, призначення компонентів, що використовуються у процесі розробки.

3.2 Мови для веб-розробки

HTML - гіпертекстова мова розмітки, яка використовується для створення документів в Інтернеті. Будь-який HTML документ складається із звичайного тексту та розмічальних тегів. Текст, як і будь-яких інших файлах, використовується передачі інформації, а розмічальні теги використовуються її групування і оформлення. Розмічальні теги HTML є спеціальні слова, які оточені з обох боків кутовими дужками, наприклад: `<html>`. Якщо HTML документ відкрити в браузері, то він буде відображено відповідно до розмічальних тегів і буде називатися веб-сторінкою. Головна мета будь-якого веб-браузера (такого як IE, Firefox, Chrome тощо) прочитати HTML документ та відобразити його як веб-сторінку.

Основне призначення гіпертекстової мови розмітки у реалізації даного проекту є створення основної структури сайту.

RНР - мова програмування загального призначення, що використовується для розробки веб-застосунків. У 1994 році датський програміст Расмус Лердорф створив набір скриптів на Perl/CGI для виведення та обліку відвідувачів його онлайн-резюме, який обробляє шаблони HTML-документів. Лердорф назвав набір Personal Home Page (Особиста Домашня Сторінка). Незабаром функціональності та швидкості Perl – інтерпретатора скриптів – перестало хапати, і Лердорф розробив з використанням мови «Сі» новий інтерпретатор шаблонів RНР/FI. В області програмування для мережі Інтернет RНР - одна з популярних сценарних мов завдяки своїй простоті, швидкості виконання, багатій функціональності, кросплатформенності та поширенню вихідних кодів на основі ліцензії RНР.

В рамках створюваної соціальної мережі, php використовуватиметься для розробки додатків, сценаріїв тощо.

CSS – формальна мова опису зовнішнього вигляду документа, написаного за допомогою мови розмітки. CSS дозволяє зберігати інформацію про оформлення документа HTML в окремому зовнішньому файлі з розширенням .css. Редагуючи лише один цей файл, стала можливою зміна оформлення цілого веб-сайту. На даний момент CSS є стандартом оформлення HTML документів та підтримується всіма сучасними браузерами.

При створенні проекту CSS буде застосовуватися для оформлення веб-сторінок. WYSIWYG HTML-редактор компанії Adobe.

У реалізації проекту даний редактор використовується для перевірки та тестування написаних веб-сторінок, а також їх верстки. ++ - вільний текстовий редактор з відкритим вихідним кодом для Windows з підсвічуванням синтаксису великої кількості мов програмування та розмітки.

Цей редактор використовуватиметься переважно для верстки вихідного коду.

Denwer - локальний сервер та програмна оболонка, які використовуються веб-розробниками для розробки сайтів на локальній Windows-машині без необхідності виходу до Інтернету.

Для тестування та демонстрації соціальної мережі використовується локальний сервер. HTTP-сервер – вільний веб-сервер. Apache є кросплатформним програмним забезпеченням, підтримує операційні системи Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS. Основними перевагами Apache вважаються надійність та гнучкість конфігурації. Він дозволяє підключати зовнішні модулі надання даних, використовувати СУБД для автентифікації користувачів, модифікувати повідомлення про помилки тощо.

4. Опис програмної реалізації

4.1 Опис реалізації

Для розробки програми була використана Visual Studio 2019.

Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Visual Studio включає один або декілька з наступних компонентів:

- visual Basic .NET, а до його появи — Visual Basic;
- visual C++;
- visual C#;
- visual F# (входить до складу Visual Studio 2010);
- visual Studio Debugger.

Багато варіантів постачання також включають:

- microsoft SQL Server;
- mSDE Visual Source Safe — файл-серверна система управління версіями.

У минулому, до складу Visual Studio також входили продукти:

- visual InterDev;
- visual J++;
- visual J#;
- visual FoxPro;
- visual Source Safe — файл-серверна система управління версіями.

Visual Studio 2019

Перша версія нової Visual Studio (Preview) була випущена у грудні 2018 року. Вона мала 5 етапів. У березні був вже визначений RC (Release

Candidate). Тому через місяць, а саме 2-го квітня була випущена стабільна версія Visual Studio 2019 на Windows та Visual Studio on MAC. Як і у минулих версіях є три типи (Community, Professional, Enterprise).

4.2 Аналіз програмних засобів та компонентів, необхідних для розробки програмного продукту

HTML - гіпертекстова мова розмітки, яка використовується для створення документів в Інтернеті. Будь-який HTML документ складається із звичайного тексту та розмічальних тегів. Текст, як і будь-яких інших файлах, використовується передачі інформації, а розмічальні теги використовуються її групування і оформлення. Розмічальні теги HTML є спеціальні слова, які оточені з обох боків кутовими дужками, наприклад: <html>. Якщо HTML документ відкрити в браузері, то він буде відображено відповідно до розмічальних тегів і буде називатися веб-сторінкою. Головна мета будь-якого веб-браузера (такого як IE, Firefox, Chrome тощо) прочитати HTML документ та відобразити його як веб-сторінку.

Основне призначення гіпертекстової мови розмітки у реалізації даного проекту є створення основної структури сайту.

PHP - мова програмування загального призначення, що використовується для розробки веб-застосунків. У 1994 році датський програміст Расмус Лердорф створив набір скриптів на Perl/CGI для виведення та обліку відвідувачів його онлайн-резюме, який обробляє шаблони HTML-документів. Лердорф назвав набір Personal Home Page (Особиста Домашня Сторінка). Незабаром функціональності та швидкості Perl – інтерпретатора скриптів – перестало хапати, і Лердорф розробив з використанням мови «Сі» новий інтерпретатор шаблонів PHP/FI. В області програмування для мережі Інтернет PHP - одна з популярних сценарних мов завдяки своїй простоті,

швидкості виконання, багатій функціональності, кросплатформенності та поширенню вихідних кодів на основі ліцензії PHP.

В рамках створюваної соціальної мережі, php використовуватиметься для розробки додатків, сценаріїв тощо.

CSS – формальна мова опису зовнішнього вигляду документа, написаного за допомогою мови розмітки. CSS дозволяє зберігати інформацію про оформлення документа HTML в окремому зовнішньому файлі з розширенням .css. Редагуючи лише один цей файл, стала можливою зміна оформлення цілого веб-сайту. На даний момент CSS є стандартом оформлення HTML документів та підтримується всіма сучасними браузерами.

При створенні проекту CSS буде застосовуватися для оформлення веб-сторінок. WYSIWYG HTML-редактор компанії Adobe.

У реалізації проекту даний редактор використовується для перевірки та тестування написаних веб-сторінок, а також їх верстки.++ - вільний текстовий редактор з відкритим вихідним кодом для Windows із підсвічуванням синтаксису великої кількості мов програмування та розмітки.

Цей редактор буде використовуватись переважно для верстки вихідного коду.

Denwer - локальний сервер та програмна оболонка, які використовуються веб-розробниками для розробки сайтів на локальній Windows-машині без необхідності виходу до Інтернету.

Для тестування та демонстрації соціальної мережі використовується локальний сервер. HTTP-сервер – вільний веб-сервер. Apache є кросплатформним програмним забезпеченням, підтримує операційні системи Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS. Основними перевагами Apache вважаються надійність та гнучкість конфігурації. Він дозволяє підключати зовнішні модулі надання даних, використовувати СУБД для автентифікації користувачів, модифікувати повідомлення про помилки тощо.

4.3 Розробка продукту

Для оптимальної розробки середовища програмного засобу необхідно комбінувати різні мови програмування, тому що кожен з них спрямований на виконання певних цілей та завдань. Як, наприклад, кілька команд PHP дозволяють створити цілу Web-сторінку, але практично майже завжди скрипт використовується спільно з HTML, і зазвичай вихідний текст скрипта містить велику кількість рядків. Але, не дивлячись на це, слід зазначити, що код на PHP може знаходитись у будь-якому місці HTML-документа, однак він не обов'язково повинен використовувати HTML. Необхідно лише забезпечити, щоб PHP-код створював коректний HTML-код, який буде правильно відображений Web-браузером.

HTML - гіпертекстова мова розмітки, яка використовується для створення документів в Інтернеті. За допомогою нього створюється необхідна структура та сітка сторінки, зовнішній вигляд якої надалі удосконалюється CSS та JavaScript. На даний момент останньою версією є HTML5, якій передувала HTML4.01. Більшість Web-ресурсів побудовані на основі цієї мови.

На відміну від HTML 4, який має 3 валідатора, у HTML 5 валідатор один: `<!DOCTYPE html>`. HTML 5 підтримує MathML та SVG.

Нові теги: section, article, aside, hgroup, header, footer, nav, dialog, figure, video, audio, source, embed для вставки контенту з плагіном(тільки), mark, progress, meter, time, ruby, rt, rp , canvas, command, details, datalist, keygen, output.

Нові типи input: tel, search, url, email, datetime, date, month, week, time, datetime-local, number, range, color.

Нові атрибути для тегів: атрибути ping media для a та area тощо.

Зникнення деяких тегів, тому що їх можна замінити CSS: basefont, big, center, font, s, strike, tt, u.

Зникнення фреймів через негативний вплив на всю сторінку.

Зникнення деяких тегів, замінених у оновленій специфікації більш актуальні: `acronym`(використовується `abbr`), `applet`(використовується `object`), `isindex`, `dir`.

Не підтримуються деякі атрибути у тегів через відсутність необхідності: `rev` і `charset` у `link` і `a`, `shape` і `coords` у `a` тощо.

Не підтримуються деякі атрибути у тегів через те, що при використанні CSS досягається найкращий ефект: `align` у всіх тегів, `alink`, `link`, `text`, `vlink` у `body` і так далі.

Нові API: малювання 2D-картинок у реальному часі; контроль за програванням медіафайлів; зберігання даних у браузері; редагування; Drag-and-drop; робота із мережею; MIME; нові елементи в DOM.

CSS – формальна мова опису зовнішнього вигляду документа, написаного за допомогою мови розмітки. CSS це акронім для Cascading Style Sheets/Каскадних таблиць стилів. CSS - це мова стилів, що визначає відображення HTML-документів. Наприклад, CSS працює зі шрифтами, кольором, полями, рядками, висотою, шириною, фоновими зображеннями, позиціонуванням елементів та багатьма іншими речами. HTML може використовуватися для оформлення Web-сайтів, але CSS надає більші можливості і точніший і опрацьований. CSS, на сьогоднішній день, підтримується всіма браузерами. Використовується для структурування вмісту сторінки. CSS використовується для форматування цього структурованого вмісту. З розвитком Web дизайнери почали шукати можливості форматування онлайн-ових документів. Щоб задовольнити вимогам споживачів, виробники браузерів (тоді - Netscape і Microsoft) винайшли нові HTML-теги, такі, наприклад, як ``, які відрізнялися від оригінальних HTML-тегів тим, що вони визначали зовнішній вигляд, а не структуру. Це також призвело до того, що оригінальні структурні теги, такі як `<table>`, стали все більше застосовуватися для дизайну сторінок замість структурування тексту. Багато нових тегів дизайну, таких як `<blink>`, підтримувалися лише одним браузером. "Вам необхідний браузер X для перегляду цієї сторінки" - така відмова стала

звичайним явищем на Web-сайтах. Був створений для виправлення цієї ситуації шляхом надання Web-дизайнерам можливостей точного дизайну, що підтримуються всіма браузерами. Одночасно відбувся поділ подання та вмісту документа, що значно спростило роботу.

Поява CSS стала революцією у світі Web-дизайну.

Конкретні переваги CSS:

- управління відображенням безлічі документів за допомогою однієї таблиці стилів;
- точніший контроль над зовнішнім виглядом сторінок;
- різні уявлення для різних носіїв інформації (екран, друк, і т. д.);
- складна та опрацьована техніка дизайну.

4.4 Засоби розробки програмного засобу

В основу всесвітньої павутини покладено ідеологію та технологію гіпертексту, технічно реалізовану за допомогою мови HTML. HTML (від англ. HyperText Markup Language, мова гіпертекстової розмітки) є мовою, розробленою спеціально для створення web-документів. HTML-документи є звичайними текстовими ASCII-файлами. Крім текстового змісту, вони включають до свого складу спеціальні управляючі конструкції мови - звані теги. Під час відображення документа в браузері самі теги на екран не відображаються. Вони визначають структурні одиниці всередині документа, задають параметри форматування, включають до складу документа нові елементи (вбудовані зображення, Java-аплети тощо) і дозволяють створювати гіпертекстові посилання інші документи та ресурси мережі Internet. Для створення та редагування HTML-документів достатньо простого текстового редактора, наприклад Блокнот Windows. На практиці зазвичай використовують засоби редагування, розроблені спеціально для написання HTML - редактори WYSIWYG (від англ. what you see is what you get, що

бачиш, те й отримуєш) або, як їх називають, засоби візуального редагування. Такі програмні продукти мають графічні інтерфейси, що роблять написання HTML-коду подібним до роботи з програмою редагування текстів та розмітки сторінок. Подібні програмні продукти дозволяють підвищити ефективність і значно скоротити час розробки, особливо якщо йдеться про середні та великі сайти. Спеціалізовані HTML-редактори також працюють з вихідним кодом web-сторінки, але мають у своєму розпорядженні різні допоміжні інструменти, що прискорюють і спрощують процес редагування документів, і додатковими засобами для виконання повторюваних операцій. Крім того, найбільш потужні з них включають до свого складу засоби автоматизації процесу розробки (робота із шаблонами, стилями та бібліотеками елементів) та засоби розміщення файлів сайту на web-сервері. До цієї групи редакторів належить велика кількість програмних продуктів, у тому числі і поширюваних безкоштовно.

Створення сайту може бути виконане з використанням технологій AJAX, jQuery, PHP, CGI, MySQL, Flash, Silver light та ін. Сайт може бути розроблений з використанням одного зі стандартних засобів: Adobe Dreamweaver, MS FrontPage, Amaya та ін. Коректність роботи ресурсу має бути перевірена у більшості стандартних браузерів (у тому числі і їх версіях): IE, Mozilla Firefox, Opera, Safari, Chrome.

Dreamweaver HTML <#"justify"> Створювачі програми Dreamweaver постаралися зробити зручний і простий у використанні інструмент, що полегшує процес створення HTML-сторінок, їх редагування та керування великою кількістю пов'язаних один з одним документів.

Кожен новий HTML-документ відкривається у новій закладці. Для зручнішої роботи з програмою передбачено систему меню та набір палітр інструментів.

Ви можете налаштовувати робочий простір програми таким чином, щоб ваша робота була максимально зручною та ефективною. Програма Dreamweaver, як візуальний HTML-редактор, також дозволяє перемикатися в

режим ручного редагування коду, виконуючи взаємозв'язок між візуальним представленням сторінки та її вихідним кодом. Для перемикання між режимами відображення та редагування HTML-документа призначено три перші кнопки () на панелі інструментів.

Усі основні дії під час створення та редагування HTML-сторінки виконуються у полі документа. Як відомо, крім текстової інформації, HTML-документ може містити різні графічні та мультимедійні елементи. Для зручності вставки таких елементів призначена панель об'єктів вставки. За допомогою кнопок, розташованих на цій панелі, можна швидко вставити різні об'єкти (зображення, таблиці, Flash-анімація) в документ, що редагується.

Будь-який HTML-документ можна як сукупність таких елементів, як слово, абзац, зображення, осередок таблиці та інших. Всі вони мають деякі загальні властивості (наприклад, вирівнювання), але в той же час кожен об'єкт має властиві тільки йому властивості, параметри та налаштування. Для швидкого доступу до редагування параметрів об'єкта призначено панель. Залежно від поточного виділеного об'єкта вона змінює свій зовнішній вигляд - пропонується переглянути та за необхідності змінити набір властивостей саме цього об'єкта.

Після того, як структура майбутнього сайту продумана, можна приступати до розробки самих Web-сторінок (окремих елементів сайту). І почати слід із головної (домашньої) сторінки сайту.

Для створення сторінки необхідно відкрити новий порожній документ. Це можна зробити різними способами, наприклад, обравши команду File=>New у головному меню, а потім обравши HTML зі списку. Перше, що необхідно зробити після створення документа, зберегти файл з необхідним ім'ям (інакше при створенні гіпертекстових посилань, вставці зображень тощо можуть бути використані невірні адреси). Головна сторінка сайту повинна мати спеціальне ім'я – index.html (або index.htm). Після того як сторінці присвоєно ім'я та вона збережена, можна приступати до заповнення її інформаційним змістом.

Крім заголовка, у категорії Appearance цього діалогового вікна можна встановити такі параметри сторінки:

Background image - зображення, яке буде використане як тло сторінки (графічний файл із зображенням можна вибрати, натиснувши кнопку Browse...);

Background/Text - відповідно кольору фону сторінки та тексту, розташованого на ній (можна вибрати колір із палітри, натиснувши на кнопці поруч);

Links / Visited links / Active links - відповідно кольору посилань / відвідуваних посилань / активних посилань;

Left margin / Top margin / Margin width / Margin height - задають відступи (поля) з боків і на початку, і в кінці сторінки.

У HTML-документі необхідно розділяти дані документа та їх подання. Зміст документа задається як теги HTML. Для визначення візуального представлення документа слід використовувати таблиці стилів (CSS - Cascading Style Sheets) (через CSS задаються інтервали між рядками тексту, відступи, кольори, що використовуються для тексту і фону, розмір і стиль шрифтів та інше оформлення).

Існує кілька основних тегів HTML, які мають бути присутніми в тексті будь-якої web-сторінки. Кожна web-сторінка зобов'язана містити тег <HTML>, що розташовується на самому початку (кожен із тегів має відкриваючу (наприклад, <HTML>) та закриваючу частини (наприклад, </HTML>)). Цей тег описує документ як веб-сторінку, виконану у форматі HTML.

Безпосередньо за тегом <HTML> зазвичай слідує тег <HEAD>, який вказує на наявність тексту, що містить найменування сторінки та додаткові відомості про неї. У розділ HEAD зазвичай вкладено тег <TITLE>, який служить для позначення найменування сторінки (найменування сторінок зазвичай відображається у рядку заголовка вікна браузера). Потім слідує тег <BODY>, який вказує на початок «тіла» web-сторінки. У цьому розділі міститься весь текст, який відображається браузером, графіка, таблиці та інші елементи вмісту сторінки, які побачить відвідувач, який звернувся до сайту.

5. Робота користувача з програмною системою.

5.1 Системні вимоги

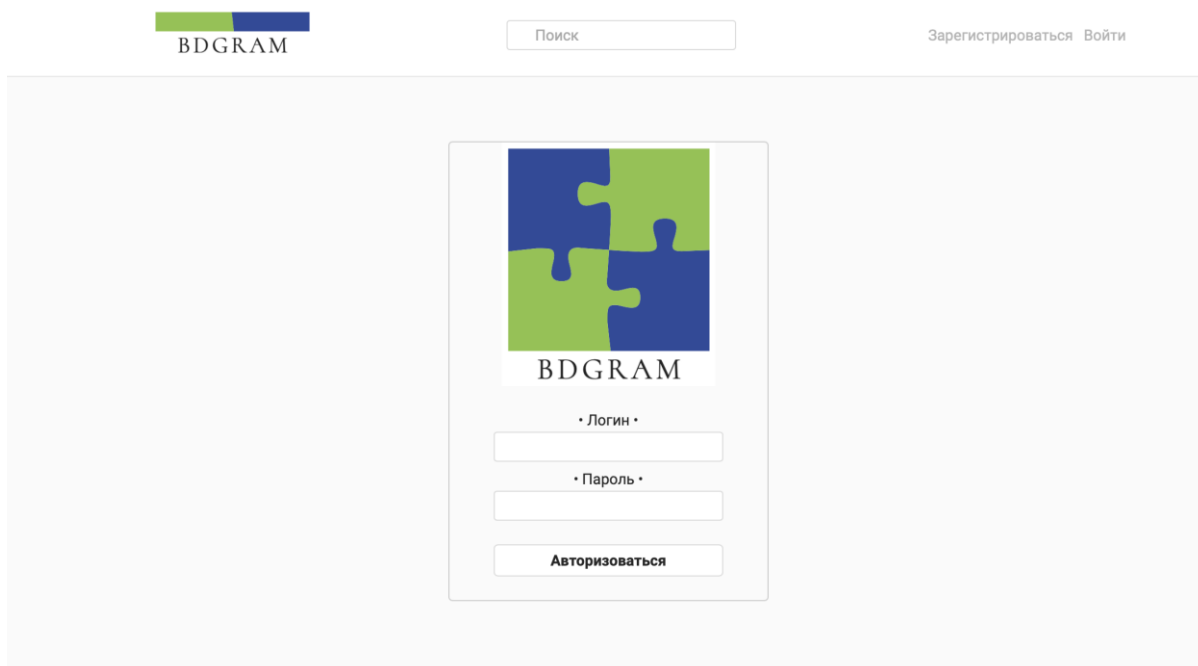
Робота системи розпочинається з вікна авторизація або реєстрація. Далі користувач може переглянути нові пости, додати вподобання, коментарій, виставити новий пост, додати підпис, геопозицію, відредагувати свій акаунт, підписатись/відписатись, подивитись сторінку іншого користувача.

5.2 Робота з програмою

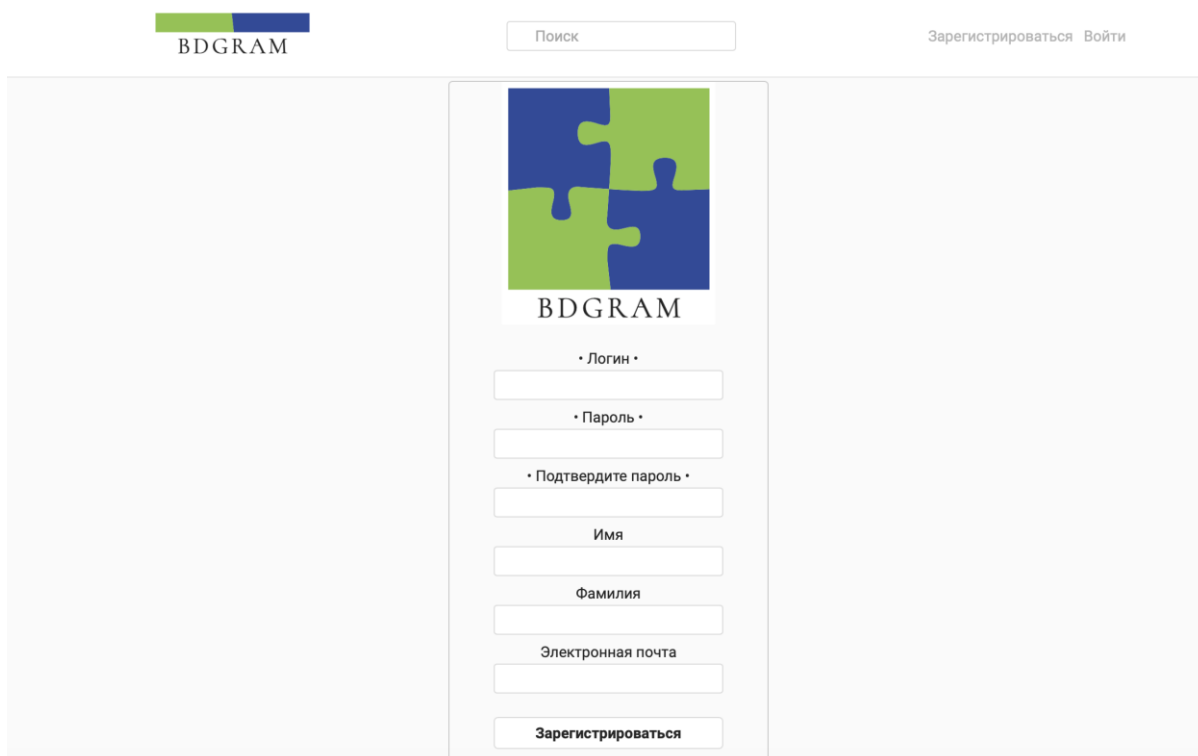
Виходячи з основних обмежень та функціональних можливостей системи можна зробити більш детальний опис алгоритму роботи системи, його компонентів з серверної та клієнтської сторони, та способу взаємодії з продуктом.

Опис користувацького інтерфейсу.

Сторінка «Авторизація» відображає сторінку з полями які потрібно заповнити для входу, які відображені на рисунку 3.1 та реєстрації на рисунку 3.2.

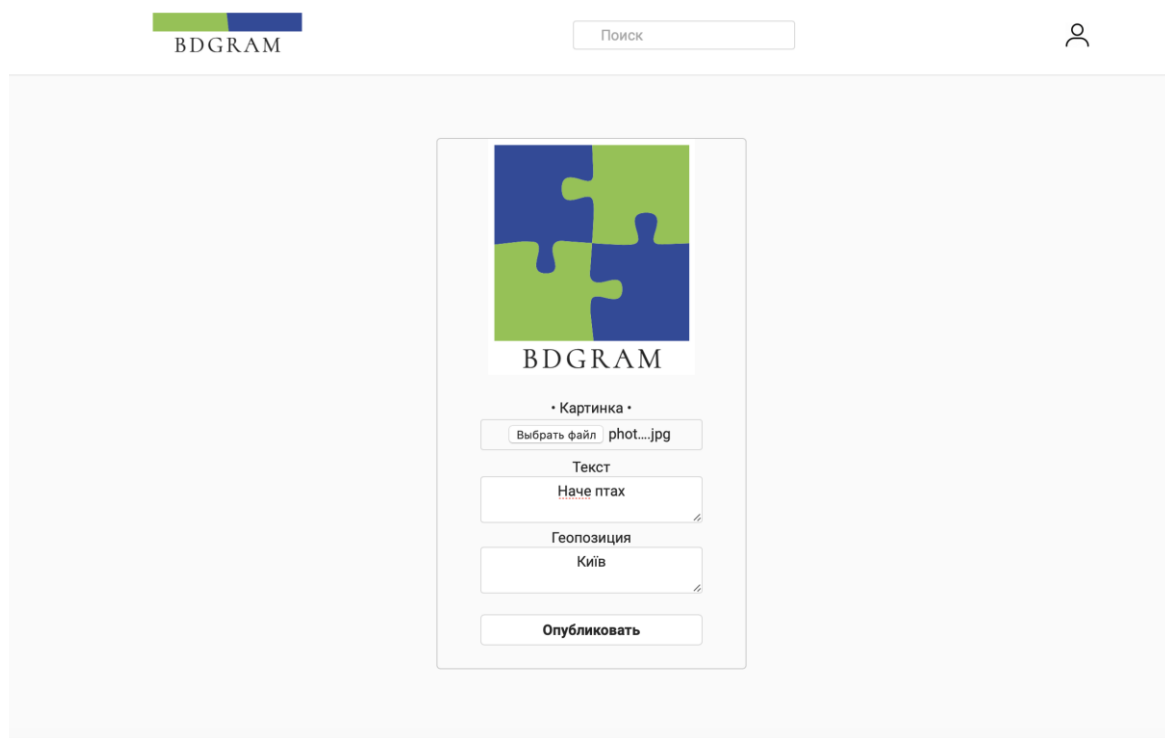


«Рисунок 3.1 – Сторінка авторизація»



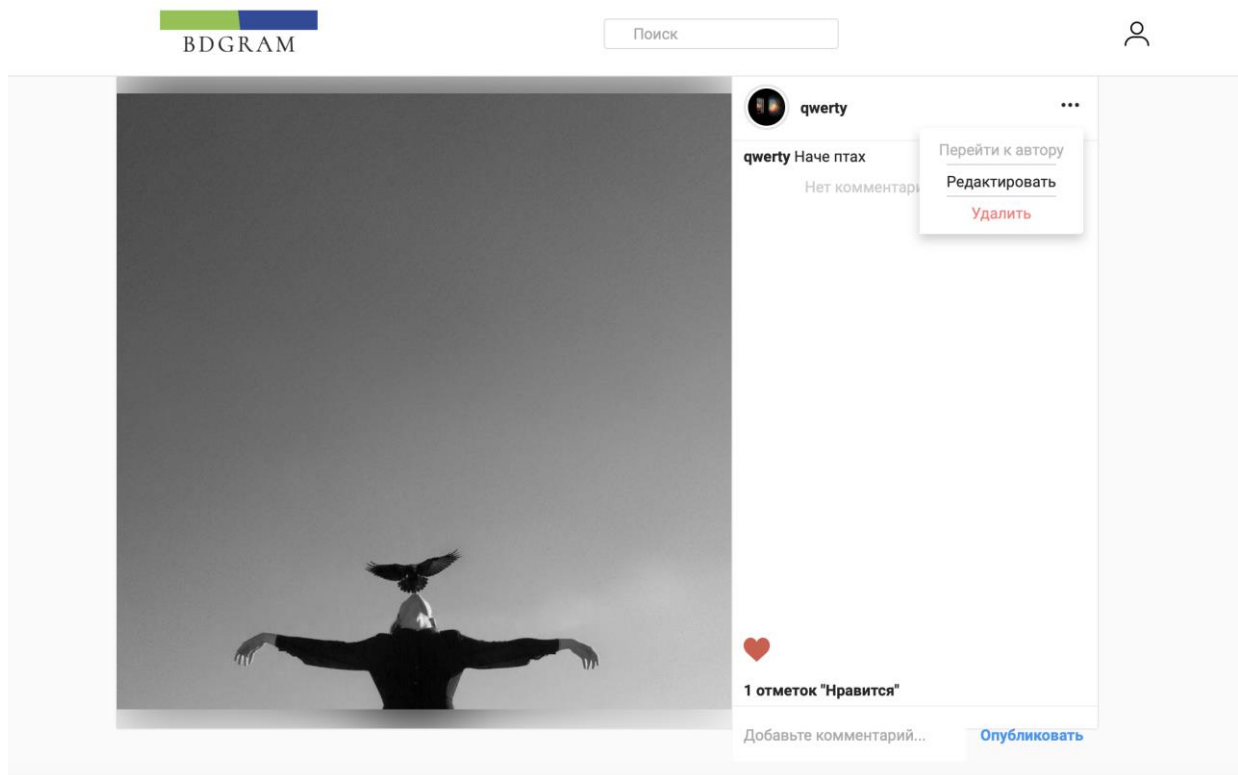
«Рисунок 3.2 – Сторінка авторизація»

Для додавання посту виконується наступний інтерфейс програми як на рисунку 3.3 та рисунку 3.4.



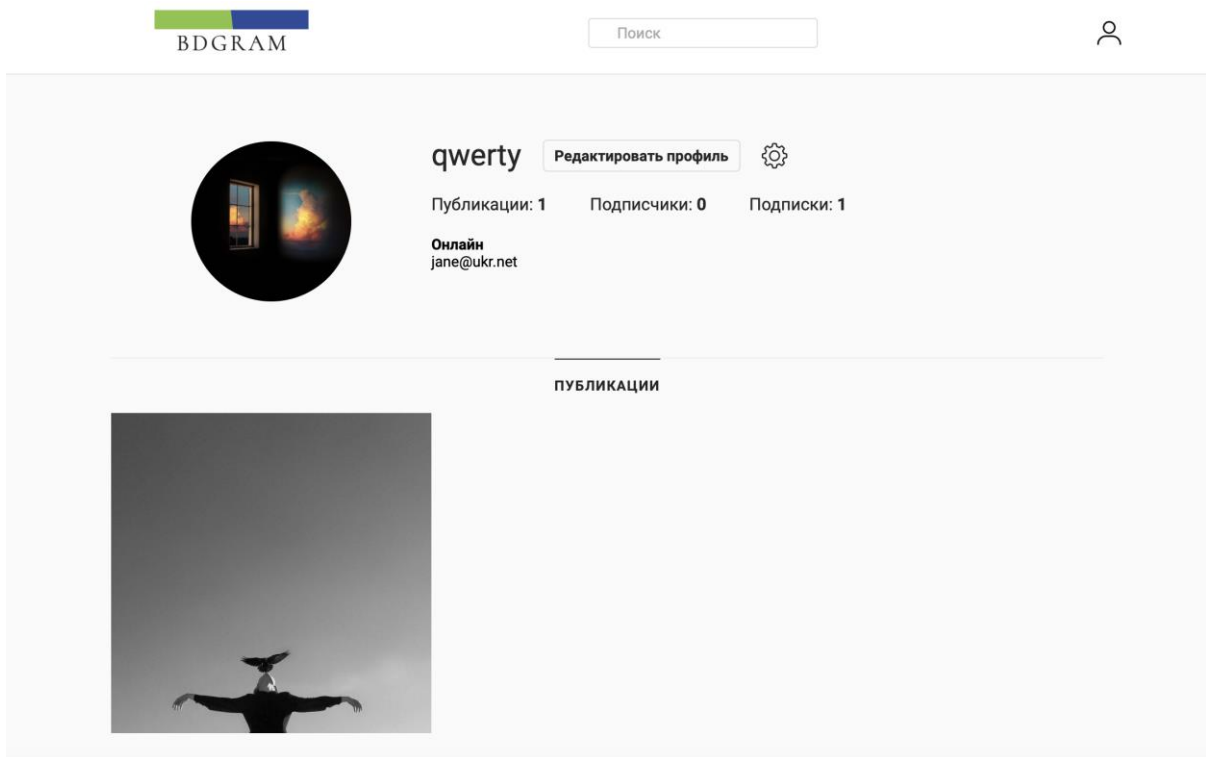
The screenshot displays the user interface for creating a post in the BDGRAM application. At the top, there is a header bar containing the BDGRAM logo on the left, a search bar with the text "Поиск" in the center, and a user profile icon on the right. The main content area is a light gray rectangle. In the center of this area is a white card with a rounded border. The card features a square profile picture at the top, which is a 2x2 grid of blue and green puzzle pieces. Below the picture is the text "BDGRAM". Underneath the profile picture section is a toggle switch labeled "• Картинка •". Below this is a file selection button that says "Выбрать файл" followed by the filename "phot...jpg". Below the file selection is a text input field labeled "Текст" containing the text "Наче птах". Below the text field is a location selection field labeled "Геопозиция" containing the text "Київ". At the bottom of the card is a button labeled "Опубликовать".

«Рисунок 3.3 – Сторінка додавання посту»

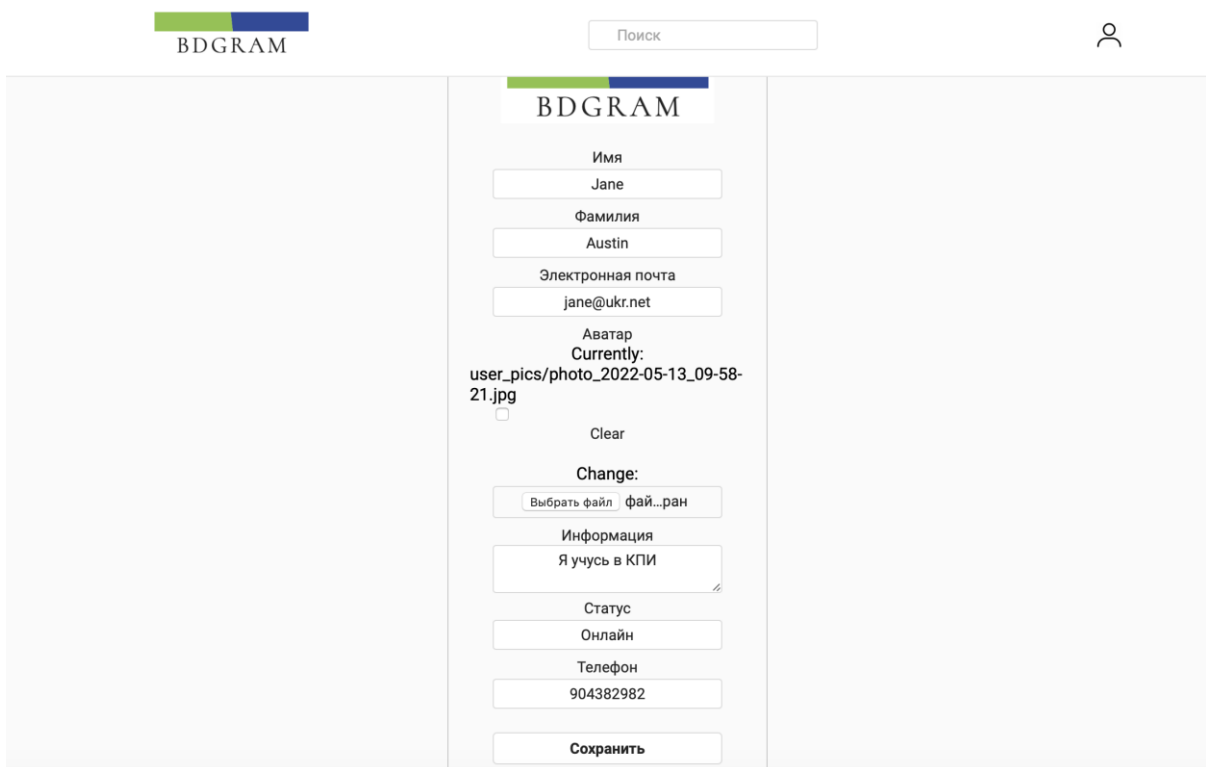


«Рисунок 3.4 – Сторінка пост та його функції»

Для власника акаунту виконується наступний інтерфейс програми, як на рисунку 3.5 та рисунку 3.6.

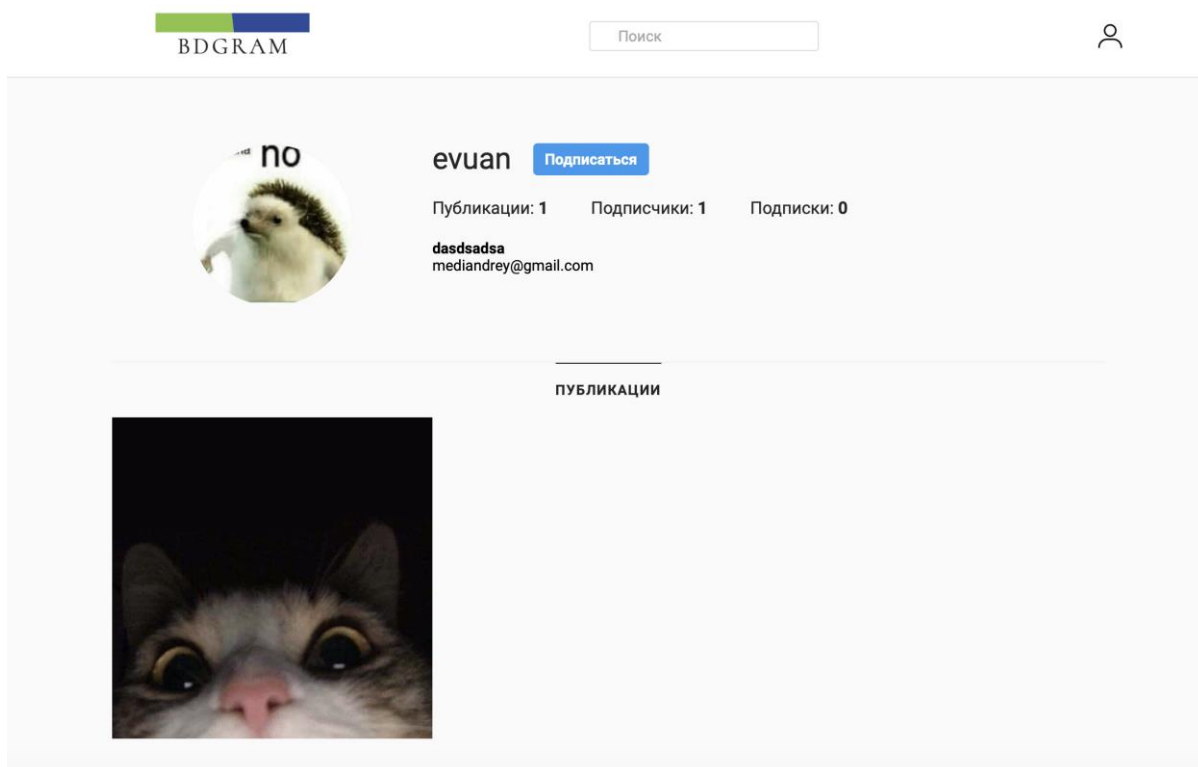


«Рисунок 3.5 – Сторінка власний акаунт»

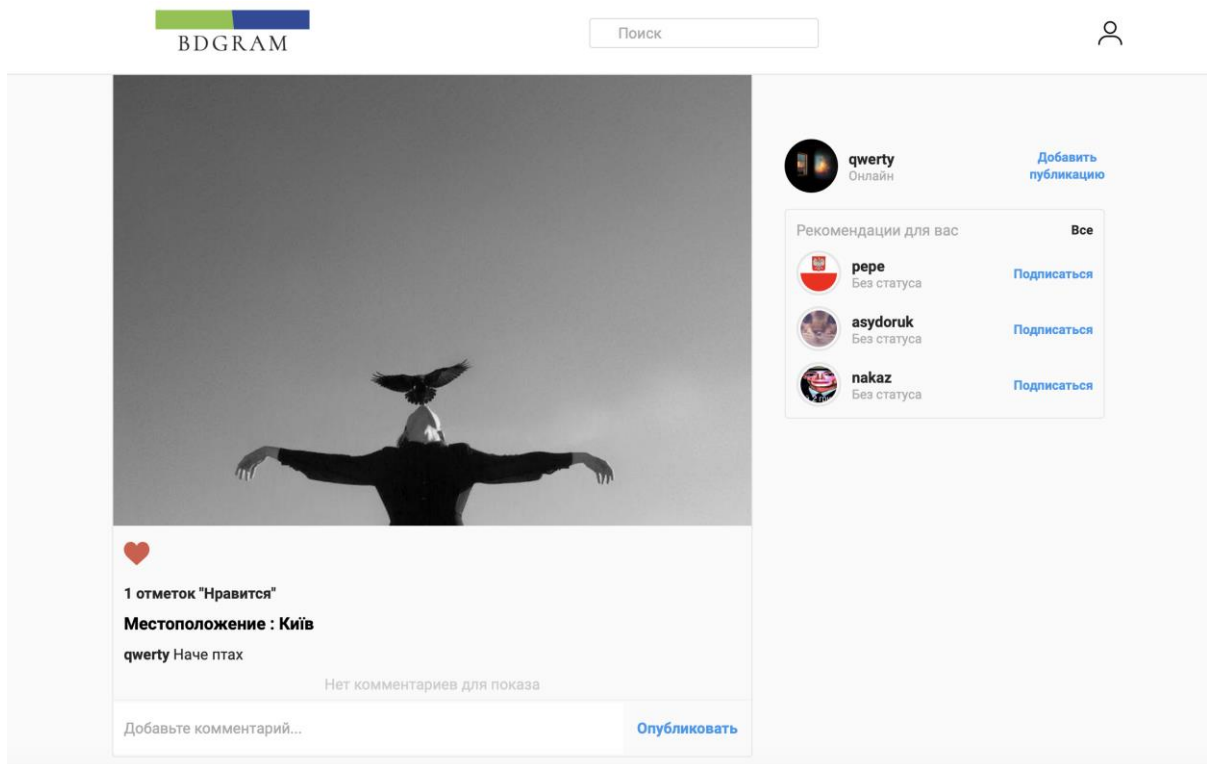


«Рисунок 3.6 – Сторінка редагування профілю»

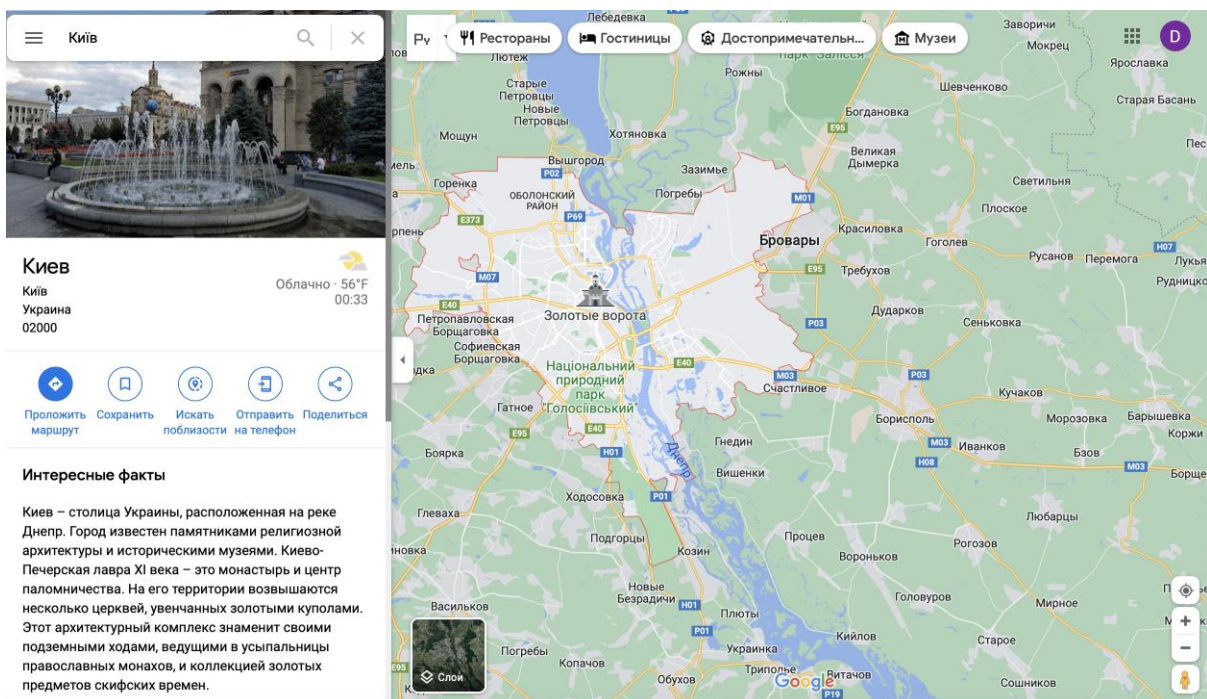
Для інших користувачів виконується наступний інтерфейс програми, як на рисунку 3.7, рисунку 3.8 та рисунку 3.9.



«Рисунок 3.7 – Сторінка інший користувач»



«Рисунок 3.8 – Сторінка стрічка новин»



«Рисунок 3.9 – Сторінка геопозиція»

Вибір мови програмування Python є досить вдалий через її орієнтованості на клієнт-серверного програмного забезпечення та більш безпечною для розробки багато потокових , безпечних проєктів. Основними компонентами моєї системи є Користувачі. Інтерфейс додатку зрозумілий та легкий для використання.

ВИСНОВКИ

Розробка даного проекту провела по кожному кроку створення програмного продукту і дала можливість реалізувати знання здобуті за навчання. Проект має непогані перспективи на ринку за досить лояльної концепції спілкуванням та можливої створення фан-бази користувачів які будуть використовувати її. Соціальна сфера з кожним роком розвивається великими темпами тому проект в подальшому не втратить актуальності.

Самі інструменти та технології розробки проекту показали себе з різних сторін, які потрібно буде враховувати далі, вибір стеку вважаємо вдалим через досить стабільність на ринку на сьогоднішній день. Весь інструментарій є безкоштовний . Стек технологій перевірений часом, тому при розробці швидко знаходяться відповіді на незрозумілі реалізації концепцій.

Розрахувавши необхідний початковий бюджет проекту і план монетизації та розвитку проекту можливо звернутись до інвесторів для вливання коштів. Зацікавленість проекту можлива завдяки економічній ефективності та капіталовкладень. Собівартість проекту для інвесторів досить приваблива.

Також, для повної легалізації та законності існування проекту було розписано основні права по охороні праці, які запобігають травми на виробництві.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Microsoft Visual Studio [Електронний ресурс] // Вікіпедія – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio.
2. Середовище розробки Microsoft Visual Studio [Електронний ресурс] – Режим доступу до ресурсу: https://informatics.in.ua/programming_csharp/part_01.php.
3. Клієнт-серверна архітектура [Електронний ресурс] // Вікіпедія – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Клієнт-серверна_архітектура.
4. Клієнт-серверна архітектура та ролі серверів. [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@IvanZmerzlyi/клієнт-серверна-архітектура-та-ролі-серверів-9893d8048229>.
5. Romano F., Gaston C. Hillar, Ravindar A. : Book Collection: Learning Path, Learning Web, Development with Python. First published: December 2018. ISBN 978-1-78995-329-9. URL: <https://edu.anarcho-copy.org/Programming%20Languages/Python/learn-web-development-python-hands.pdf>
6. Html & css. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3.org/standards/webdesign/htmlcss>
7. Full Stack Python. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.fullstackpython.com/web-development.html>
8. Introduction to hashing. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.interviewbit.com/courses/programming/hashing>
9. What can PHP do? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/manual/en/intro-whatcando.php>

10. HTTP servers. [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.python.org/3/library/http.server.html>
11. Denwer. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.softprime.net/razrabotka/web-razrabotchiku/546-denwer-lokalnyy-web-server.html>
12. Perl/CGI programming. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.geeksforgeeks.org/perl-cgi-programming/>
13. A brief history of PHP. [Електронний ресурс] – Режим доступу до ресурсу:
https://docstore.mik.ua/orelly/weblinux2/php/ch01_02.htm
14. Тестування клієнта/сервера. [Електронний ресурс] – Режим доступу до ресурсу:
<https://uk.myservername.com/types-software-testing>
15. MD4 // Вікіпедія – Режим доступу до ресурсу:
<https://ru.wikipedia.org/wiki/MD4>
16. MD5 // Вікіпедія – Режим доступу до ресурсу:
<https://ru.wikipedia.org/wiki/MD5>
17. Html element reference. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.w3schools.com/TAGs/default.asp>
18. Html frame tag. [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.geeksforgeeks.org/html-frame-tag/>
19. What is WYSIWYG? [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.tiny.cloud/blog/wysiwyg/>
20. Методи розробки Web-сайтів. [Електронний ресурс] – Режим доступу до ресурсу:
<https://sites.google.com/site/tz5103voinovakateryna/metodi-rozrobki-web-sajtiv>

ДОДАТОК А

```
from django.db import models

class Post(models.Model):

    author = models.ForeignKey(

        'users.User',

        related_name='posts',

        on_delete=models.CASCADE,

        verbose_name='Автор',

        null=False,

        blank=False,

        default=None

    ) //додавання посту

    image = models.ImageField(

        null=False,

        blank=False,

        upload_to='post_pics',

        verbose_name='Картинка'

    ) // завантаження фоторграфії

    text = models.TextField(

        max_length=500,

        verbose_name='Текст',
```

```

        null=True,

        blank=True

    ) // підпис

location = models.TextField(

    max_length=500,

    verbose_name='Геопозиція',

    null=True,

    blank=True

) // локація

created_at = models.DateTimeField(

    verbose_name='Дата публікації',

    auto_now_add=True,

    null=True,

    blank=True

) // дата

def __str__(self):

    text = ' '.join(self.text.split(' ')[:4])

    return f'{self.author.username} - {text}...'

```

```

class Comment(models.Model):

    author = models.ForeignKey(

        'users.User',

        related_name='comments',

        on_delete=models.CASCADE,

        verbose_name='Автор',

        null=False,

        blank=False,

        default=None

    ) // коментарій

    post = models.ForeignKey(

        'publications.Post',

        related_name='comments',

        on_delete=models.CASCADE,

        verbose_name='Пост',

        null=False,

        blank=False,

        default=None

    ) // пост

```



```

text = models.TextField(

    verbose_name='Текст',

    null=False,

    blank=False,

    default=None,

    max_length=100

)

created_at = models.DateTimeField(

    verbose_name='Дата публикации',

    auto_now_add=True,

    null=True,

    blank=True

)

```

```

def __str__(self):

    text = ' '.join(self.text.split(' ')[:4])

    return f'{self.post.pk} - {self.author.username} - {text}'

```

```

class Like(models.Model):

    user = models.ForeignKey(

```

```

        'users.User',

        related_name='likes',

        on_delete=models.CASCADE,

        verbose_name='Пользователь',

        null=False,

        blank=False,

        default=None

    ) //вподобання користувачів

    post = models.ForeignKey(

        'publications.Post',

        related_name='likes',

        on_delete=models.CASCADE,

        verbose_name='Пост',

        null=False,

        blank=False,

        default=None

    ) // вподобання постів


    def __str__(self):

        return f'{self.user.username} понравился пост {self.post}'

```

ДОДАТОК Б

```
from django import forms
```

```
from .models import User
```

```
class UserEditForm(forms.ModelForm):
```

```
    class Meta:
```

```
        model = User
```

```
        fields = [
```

```
            'first_name',
```

```
            'last_name',
```

```
            'email',
```

```
            'avatar',
```

```
            'description',
```

```
            'status',
```

```
            # 'gender',
```

```
            'phone'
```

```
        ]
```

```
        labels = {
```

```
            'first_name': 'Имя',
```

```
            'last_name': 'Фамилия',
```

```
            'email': 'Электронная почта',
```

```
            'avatar': 'Аватар',
```

```
            'description': 'Информация',
```

```
            'status': 'Статус',
```

```
            # 'gender': 'Пол',
```

```
            'phone': 'Телефон',
```

```
        } //інформація про користувача
```

```
class UserPasswordChangeForm(forms.ModelForm):
```

```
    old_password = forms.CharField(label="Старый пароль", strip=False,  
widget=forms.PasswordInput)
```

```
    password = forms.CharField(label="Новый пароль", strip=False,  
widget=forms.PasswordInput)
```

```
    password_confirm = forms.CharField(label="Подтвердите пароль", strip=False,  
widget=forms.PasswordInput)
```

```
    def clean_old_password(self):
```

```
        old_password = self.cleaned_data.get('old_password')
```

```

    if not self.instance.check_password(old_password):
        raise forms.ValidationError('Старый пароль неверен')
    return old_password

def clean_password(self):
    password = self.cleaned_data.get("password")
    if self.instance.check_password(password):
        raise forms.ValidationError('Новый пароль совпадает со старым')
    return password
// зміна паролю
def clean_password_confirm(self):
    password = self.cleaned_data.get("password")
    password_confirm = self.cleaned_data.get("password_confirm")
    if password and password_confirm and password != password_confirm:
        raise forms.ValidationError('Пароли не совпадают')
    return password_confirm

def save(self, commit=True):
    user = self.instance
    user.set_password(self.cleaned_data["password"])
    if commit:
        user.save()
    return user

class Meta:
    model = User
    fields = ['old_password', 'password', 'password_confirm']
    labels = {
        'old_password': 'Старый пароль',
        'password': 'Пароль',
        'password_confirm': 'Подтверждение пароля',
    }

class UserRegisterForm(forms.ModelForm):
    password = forms.CharField(label='Пароль', strip=False, widget=forms.PasswordInput)
    password_confirm = forms.CharField(label='Подтвердите пароль', strip=False,
    widget=forms.PasswordInput)

    def clean_password_confirm(self):
        password = self.cleaned_data.get('password')
        password_confirm = self.cleaned_data.get('password_confirm')
        if password and password_confirm and password != password_confirm:
            raise forms.ValidationError('Пароли не совпадают')
        return password_confirm

```

```

// перевірка паролю
def save(self, commit=True):
    user = super().save(commit=False)
    user.set_password(self.cleaned_data['password'])
    if commit:
        user.save()
    return user

class Meta:
    model = User
    fields = ['username', 'password', 'password_confirm', 'first_name', 'last_name', 'email']
    labels = {
        'username': 'Логин',
        'password': 'Пароль',
        'password_confirm': 'Подтверждение пароля',
        'first_name': 'Имя',
        'last_name': 'Фамилия',
        'email': 'Электронная почта',
    }

class UserLoginForm(forms.ModelForm):
    password = forms.CharField(label='Пароль', strip=False, widget=forms.PasswordInput)

    def clean(self):
        return self.cleaned_data

    def clean_username(self):
        username = self.cleaned_data.get('username')
        try:
            User.objects.get(username=username)
        except User.DoesNotExist:
            raise forms.ValidationError('Неверное имя пользователя')
        return username

    def clean_password(self):
        username = self.cleaned_data.get('username')
        password = self.cleaned_data.get('password')
        user = User.objects.get(username=username)

        if not user.check_password(password):
            raise forms.ValidationError('Неверный пароль')
        return password

class Meta:

```

```
model = User
fields = ['username', 'password']
labels = {
    'username': 'Логин',
    'password': 'Пароль',
}
// перевірка правильності паролю
```