МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

**КУРСОВА РОБОТА**

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Сайт-магазин будiвельних матерiалiв:

**«Соцыальна мережа»**

студента II курсу групи ІПЗ-20-2

спеціальності 121 «Інженерія програмного забезпечення»

Вакулович Нікіта Вікторович

(прізвище, ім’я та по-батькові)

Керівник: ктн, доц. А.В. Морозов

Дата захисту: " 14 " січня 2022 р.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.С.Граф

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.Г. Чижмотря .

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В.Кузьменко

(підпис) (прізвище та ініціали)

Житомир – 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп’ютерних технологій

Кафедра комп’ютерних наук

Освітній рівень: бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

В.о. зав. кафедри

\_\_\_\_\_\_\_\_\_\_\_ А.В. Морозов

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_2021р.

ЗАВДАННЯ

НА КУРСОВИЙ ПРОЕКТ СТУДЕНТУ

Вакуловичу Нікіті Вікторовічу

1. Тема роботи: Сайт-магазин будiвельних матерiалiв,

керівник курсового проекту: ст. викладач кафедри ІПЗ Чижмотря О.Г.

1. Строк подання студентом: “ 10 ” січня 2021 р.
2. Вхідні дані до роботи: Розробити Сайт-магазин будiвельних матерiалiв.
3. Зміст розрахунково-пояснювальної записки(перелік питань. Які підлягають розробці)
   * + 1. Постановка завдання
       2. Аналіз аналогічних розробок
       3. Алгоритми роботи програми
       4. Опис роботи програми
       5. Програмне дослідження
4. Перелік графічного матеріалу(з точним зазначенням обов’язкових креслень)
5. Презентація;

Посилання на репозиторій: https://github.com/Chikintos/VakulovychNikitaCourseWork-web-

1. Консультанти розділів проекту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посади консультанта | Підпис, дата | |
| завдання  видав | завдання прийняв |
| 1 | ст. викладач кафедри ІПЗ Чижмотря О.Г. |  |  |
| 2 | ст. викладач кафедри ІПЗ Чижмотря О.Г. |  |  |
| 3 | ст. викладач кафедри ІПЗ Чижмотря О.Г. |  |  |
|  |  |  |  |
|  |  |  |  |

1. Дата видачі завдання “ 15 ” вересня 2021 р.

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів курсового проекту | Строк виконання етапів проекту | Примітки |
| 1 | Постановка задачі | 01.11-03.11 | виконано |
| 2 | Пошук, огляд та аналіз аналогічних розробок | 04.11-12.11 | виконано |
| 3 | Формулювання технічного завдання | 13.11-20.11 | виконано |
| 4 | Опрацювання літературних джерел | 21.11-31.11 | виконано |
| 5 | Проектування структури | 01.12-05.12 | виконано |
| 6 | Написання програмного коду | 06.12-02.12 | виконано |
| 7 | Відлагодження | 03.01-06.01 | виконано |
| 8 | Написання пояснювальної записки | 07.01-18.01 | виконано |
| 9 | Захист | 19.01 |  |

**Студент** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Вакулович Н.В.

(підпис) (прізвище та ініціали)

**Керівник проекту** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Чижмотря О.Г.

(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до курсового проекта на тему «Розробка Сайту-магазину будiвельних матерiалiв»

складається з переліку умовних

скорочень, вступу, трьох розділів, висновків, списку використаної літератури та

додатку.

Текстова частина викладена на 15 сторінках друкованого тексту.

Пояснювальна записка має 3 сторінки додатків. Список використаних джерел містить 14 найменувань і займає 2 сторінку. В роботі наведено 13 рисунків. Загальний обсяг роботи – 28 сторінок.

У першому розділі було обґрунтовано створення сайту на тему Сайт-магазин будiвельних матерiалiв

У другому розділі проведено проектування і розробка сайту.

У третьому розділі проведено тестування програмного продукту.

Висновок містить в собі результати виконаної роботи при створенні сайту на тему «Розробка Сайту-магазину будiвельних матерiалiв».

ЗМІСТ

[ВСТУП 8](#_Toc136038962)

[РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ 9](#_Toc136038963)

[1.1 Аналіз задачі, засобів та методів її вирішення 9](#_Toc136038964)

[1.2 Аналіз існуючого програмного забезпечення за тематикою курсового проекту 10](#_Toc136038965)

[Рисунок 1.2 – prom.ua 11](#_Toc136038966)

[Рисунок 1.2 – foxtrot.com.ua 12](#_Toc136038967)

[Висновки до першого розділу 12](#_Toc136038968)

[РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 13](#_Toc136038969)

[2.1 Проектування загального алгоритму роботи програми 13](#_Toc136038970)

[2.3 Розробка програмного забезпечення 15](#_Toc136038971)

[Висновки до другого розділу 17](#_Toc136038972)

[РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ 18](#_Toc136038973)

[3.1 Опис роботи з програмним додатком 18](#_Toc136038974)

[Висновки до третього розділу 22](#_Toc136038975)

[ВИСНОВКИ 23](#_Toc136038976)

[ДОДАТКИ 26](#_Toc136038977)

[Додаток А 27](#_Toc136038978)

[Додаток Б 28](#_Toc136038979)

# ВСТУП

Розробка веб-сайтів є дуже актуальною в наш час, адже веб-технології є провідними на даний момент у сфері інформаційних технологій. У цій курсовій роботі буте продемонстровано процес розробки сайту-магазину будiвельних матерiалiв.

Веб-сайт для магазину будівельних матеріалів є необхідним інструментом для привертання клієнтів та покращення ефективності бізнесу. Цей сайт дозволяє представити широкий асортимент продуктів, надати інформацію про їх характеристики та ціни, а також забезпечити зручність онлайн-замовлення. Крім того, веб-присутність дозволяє залучати нових клієнтів через пошукові системи та соціальні мережі, підвищуючи конкурентоспроможність компанії. Все це сприяє зростанню продажів і задоволенню потреб покупців у будівельних матеріалах.

Метою розробки даної курсової роботи є дослідження проектування веб-додатку з асинхронною клієнт-серверною архітектурою, а також створення системи адміністрування веб-сайтом.

Об’єктом дослідження курсової роботи є товари зазначеної предметної області, зокрема дані про них, які потрібні для грамотного розподілення контенту на веб-сайті та побудови CMS системи.

Предметом дослідження є використання різного роду веб-технологій задля розміщення товарів на сайті інтернет-магазину, їх редагування та видалення, а також для можливості оформити замовлення користувачем.

# РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

## Аналіз задачі, засобів та методів її вирішення

Створення веб-сайту — це складна та комплексна задача, інтернет-магазину зокрема. Грамотний процес поставлення завдань, порядок їх виконання є вирішальними під час створення багатофункціонального, складного за програмною архітектурою веб-додатку. Отже, сформульовано основні задачі, які стоять перед нами.

На курсовий проект було отримано завдання розробити веб-сайт з використанням html, js або python на обрану тематику, який включає виконання таких етапів:

* Розробка сайту
* Розміщення в мережі Інтернет

Вимогами до сайту:

* Сайт з використанням MVC, PHP7, JavaScript, HTML та CSS
* Мінімальна кількість модулів 5
* Адаптивність сайту
* Реалізація реєстрації та входу
* Використання бази даних sqlite3

Кожен студент обирає індивідуальну тему для розробки сайту.

На етапі розробки дизайну веб-сайту студенти повинні розробити прототип майбутнього веб-сайту. Передбачити наповнення контентом сторінок сайту. Розробити цілісний дизайн відповідно до сучасних трендів у дизайні веб-сторінок.

Мною була вибрана тема сайту – «Сайт-магазин будiвельних матерiалiв».

сервера.

## Аналіз існуючого програмного забезпечення за тематикою курсового проекту

На данний момент соціальні мережі грають дуже важливу роль у нинішньому світі, за допомогою соціальних мереж ви можете ділитися своїми успіхами, розвивати свій бізнес, знаходити нових друзів та спілкуватися з ними на відстані не використовуючи папірець та ручку.

Сайт-магазину мережа має наступні переваги:

1. Компактність –ваш улюблений магазин завжди у вас в кишені.

2. Зручність – економія часу, грошей.

3. Інформативність – інформація про товар не обмежується коротким цінником чи розповіддю консульнта.

Для аналізу я обрав наступні сайти: rozetka.com.ua, prom.ua, foxtrot.com.ua.

1. rozetka.com.ua – один з найкращих та найпрофесійніших інтернет магазинів сучасної України, .



Рисунок 1.1 – rozetka.com.ua

Даний сайт має простий та інтуїтивно зрозумілий дизайн та не містить в собі складних анімацій та елементів.

Плюси:

* Зручний інтерфейс
* Розвинута система пошуку
* Цікаві доповнення у вигляді посилань на відеогляди товарів

Мінуси:

* Можлива повільна робота на малопотужних системах
* Поганий захист від кібер атак

1. Prom.ua – один з найкращих та найпрофесійніших інтернет магазинів сучасної України, в сфері підтриманих речей та досяжності для мас .

## Рисунок 1.2 – prom.ua

Плюси:

* Зручний інтерфейс
* Розвинута система пошуку серед магазинів
* Можливість швидко та легко створити магазин

Мінуси:

* Неочевидність роботи з інтерфейсом
* Поганий захист від кібер атак

3. foxtrot.com.ua — стара мережа фізичних та віртуальних магазині, розповсюджена в Україні та за її кордонами

## Рисунок 1.2 – foxtrot.com.ua

Плюси:

* Розвинута система пошуку серед товарів
* можна дізнатись про наявність товару
* Можливість швидко та легко створити магазин

Мінуси:

* Неочевидність роботи з інтерфейсом
* Поганий захист від кібер атак

## Висновки до першого розділу

У ході виконання першого розділу було обрано засоби та інструменти, за допомогою яких буде реалізовано веб-додаток. Також було отримано теоретичний фундамент для розробки додатку. Під час реалізації першого розділу також було розглянуто існуючі веб-сайти аналоги та визначено їх переваги і недоліки, завдяки чому було отримано уявлення про власний програмний продукт.

# РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 2.1 Проектування загального алгоритму роботи програми

Веб-розробка – процес створення веб-сайту або веб-додатку. Основними етапами процесу є веб-дизайн, верстка сторінок, програмування для веб на стороні клієнта і сервера, а також конфігурація веб-сервера.

Гарантією створення якісного сайту може стати правильна і послідовна робота по розробці сайту плюс відточена до дрібниць робота на кожній стадії його створення. Хтось думає, що створювати веб-сайти зовсім не складно. Подумаєш, вигадав відповідне оформлення, зібрав воєдино потрібні коди, написав код і відправив все це красуватися на сайт. Однак, на ділі виявляється все зовсім не так легко.

Головна сторінка нашого сайту це /.

Форма авторизації міститься на сторінці /user/login, реєстрації на /user/register.

Всі основі сторінки є зверху сайту

category/ - сторінка з категоріями товарів

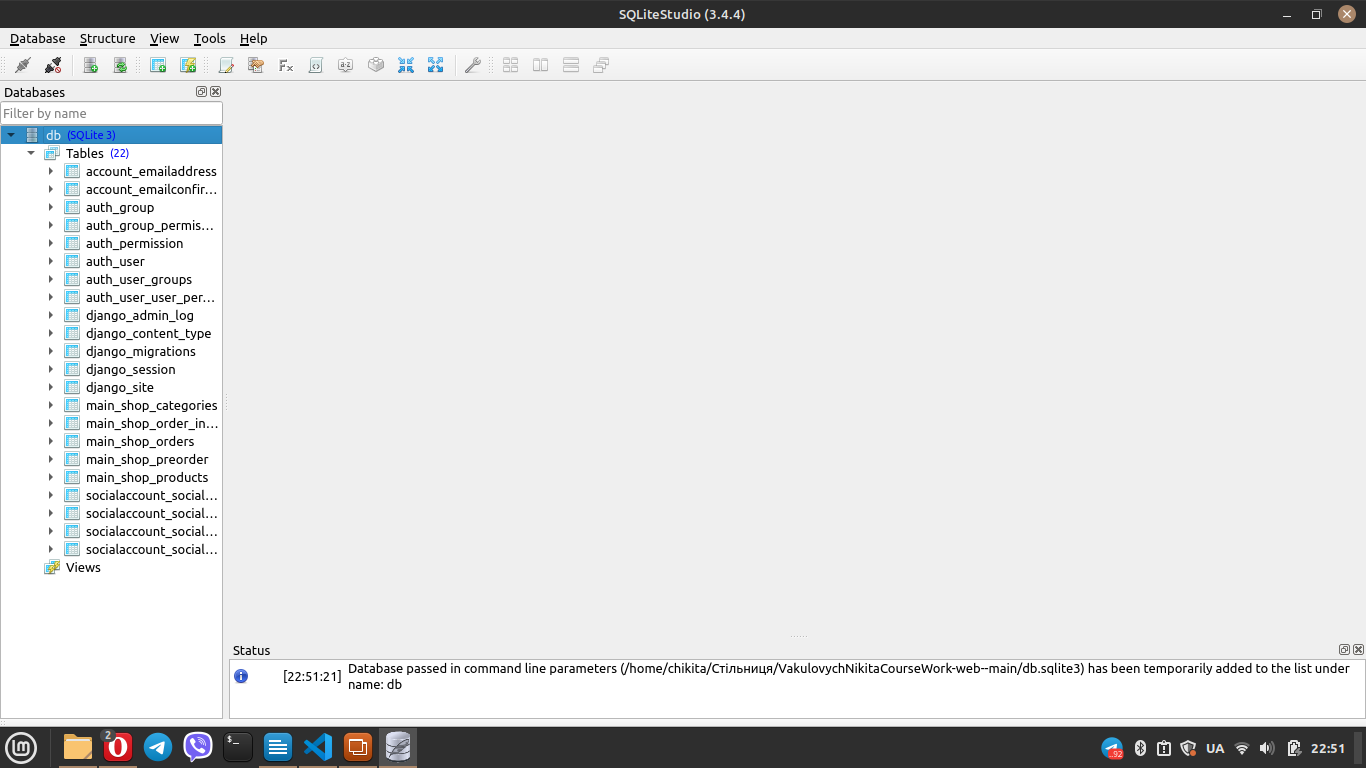
about-us/ - сторінка розповідач про вигадану компанію

Рисунок 2.1 –База даних

База даних веб-сайту має назву db та містить в собі 22 таблиці.

Таблиця accounts\_user містить в собі всю інформацію про користувачів, а саме їх логін(login), зашифрований пароль(password) та інші особисті данні.

Таблиця accounts\_user\_user\_permission має інформацію про доступ аккаунту.

Таблиця main\_shop\_products зберігає в собі інформацію про товари.

Пята таблиця main\_shop\_orders має дані про виконані замовлення

Таблиця main\_shop\_categories містить данні категорії.

Це були основні таблиці бази данних підвязаної до цього проекту.

2.2 Розробка функціональних алгоритмів роботи програми

Поняття алгоритму інтуїтивно зрозуміло та часто використовується в математиці та комп’ютерних науках. Говорячи неформально, алгоритм – це довільна коректно визначена обчислювальна процедура, на вхід якої подається деяка величина або набір величин, а результатом виконання якої є вихідна величина або набір значень. Таким чином, алгоритм є послідовністю обчислювальних кроків, які перетворюють вхідні величини у вихідні.

Алгоритм можна також розглядати як інструмент, який призначений для вирішення коректно поставленої обчислювальної задачі. У постановці задачі в загальних рисах визначаються відношення між входом та виходом. В алгоритмі описується конкретна обчислювальна процедура, за допомого якої можна досягнути виконання вказаних дій.

Отже алгоритми є невід’ємною частиною будь-якої програми чи веб-сайту. На нашому сайті також реалізовано декілька алгоритмів, наприклад алгоритм реалізації реєстрації(рис. 2.3).

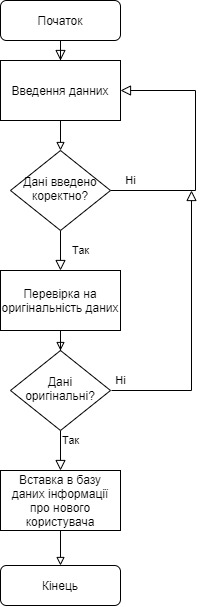


Рисунок 2.2 – Алгоритм реєстрації

Як можна побачити з рисунка вище, алгоритм реєстрації складається з трьох кроків: введення даних, перевірка на оригінальність та вставка даних в базу даних.

Якщо дані введено невірно, або користувач з такими даними вже існує ми повертаємося до першого кроку, а саме – введення даних, та виводимо користувачу де саме сталася помилка, щоб він міг виправити помилку та спробувати ще раз.

## 2.3 Розробка програмного забезпечення

Розробка веб-сайту ведеться з використанням фреймворку Django, тому замість MVC тут реалізований патерн MVT, який за замовчуванням вже присутній у фреймворку Django.

Основна мета застосування MVT полягає в розділенні даних і бізнес-логіки від візуалізації. За рахунок такого поділу підвищується можливість повторного використання програмного коду: наприклад, додати представлення даних будь-якого існуючого маршруту не тільки у вигляді HTML, але і в форматах JSON, XML, PDF, XLSX стає дуже просто і не вимагає зміни шару бізнес-логіки вихідного маршруту. Також спрощується і супровід програмного коду: внесення змін у зовнішній вигляд, наприклад, не відображаються на бізнес-логіці, а зміни бізнес-логіки не зачіпають візуалізацію.

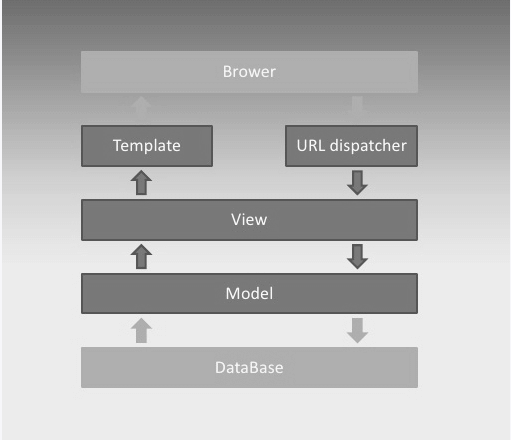


Рисунок 2.3 – структура MVT

Основні елементи патерну:

URL dispatcher: при отриманні запиту на підставі запитаної URL-адреси визначає, який ресурс повинен обробляти даний запит.

View: отримує запит, обробляє його та відправляє у відповідь користувачеві певну відповідь. Якщо обробки запиту необхідно звернення до моделі та базі даних, то View взаємодіє із нею. Для створення відповіді може використовуватися Template або шаблони. В архітектурі MVC цьому компоненту відповідають контролери (але не уявлення).

Model: описує дані, що використовуються у програмі. Окремі класи, як правило, відповідають таблицям у базі даних.

Template: представляє логіку уявлення у вигляді згенерованої розмітки HTML. У MVC цьому компоненту відповідає View, тобто уявлення.

**Висновки до другого розділу**

Отже в даному розділі було повністю розроблено наш проект, при цьому було описано роботу паттерну MVT, та його реалізацію в нашому проекті. Також були описані основні алгоритми роботи окремих частин проекту, та загальної роботи програми.

## Висновки до другого розділу

Результатом роботи на даному етапі є спроектовані функціональні алгоритми програми, які для наочності були підкріплені блок-схемами та діаграмами. Була розроблена та описана структура проекту. А також описані найважливіші методи та приведено їхній лістинг.

# 

# РОЗДІЛ 3. ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

## 3.1 Опис роботи з програмним додатком

Давайте розглянемо роботу з веб-сайтом.

При переході на головну сторінку сайту буде відображено сторінку з товарами, навігаційною панеллю та деякою додатковою інформацією

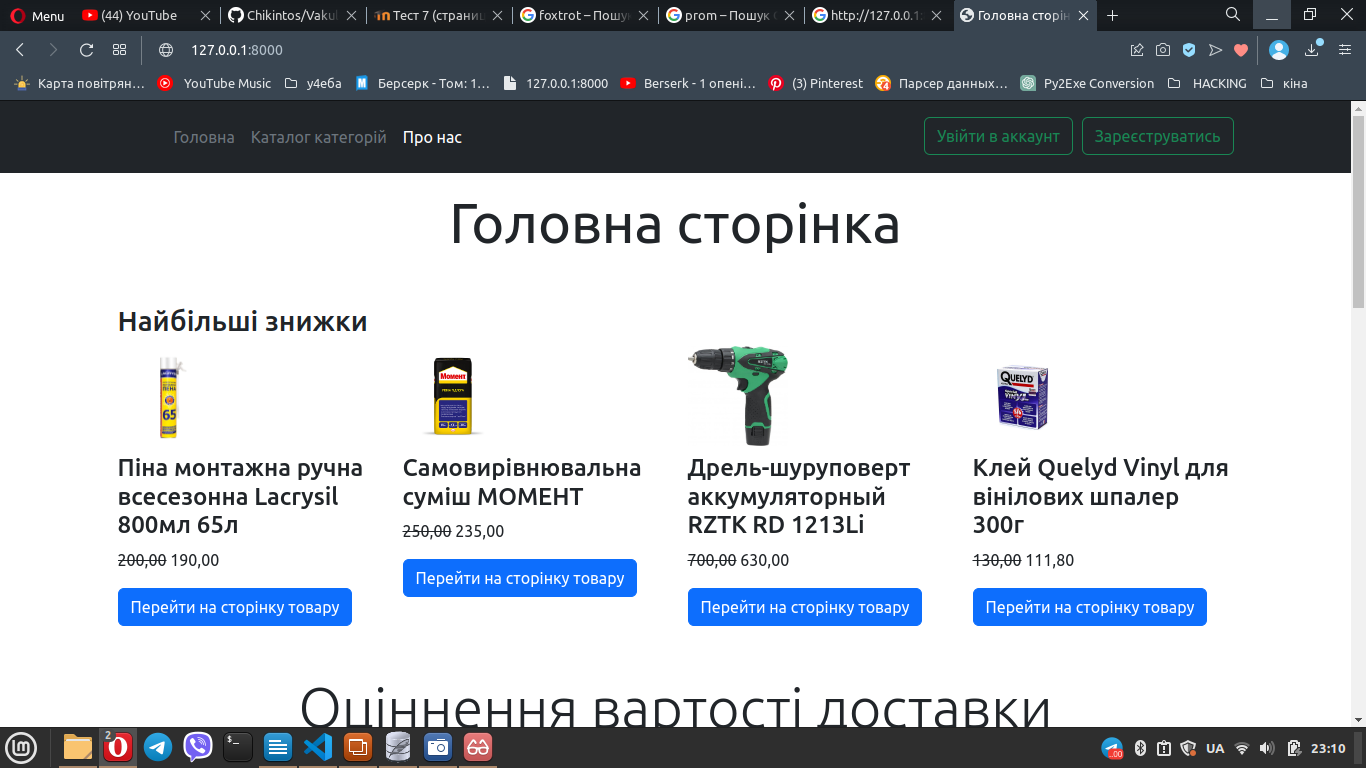


Рисунок 3.1 – Головна сторінка сайту

На даній сторінці можна перейти на допоміжні вкладки, такі як «Каталог категорій», «Про нас» та на цій же сторінці можна залогінитись,зареєструватись та розлогінитись .

На сторінці «Каталог категорій» можна обрати категорію та отримати список товарів, які належать до неї

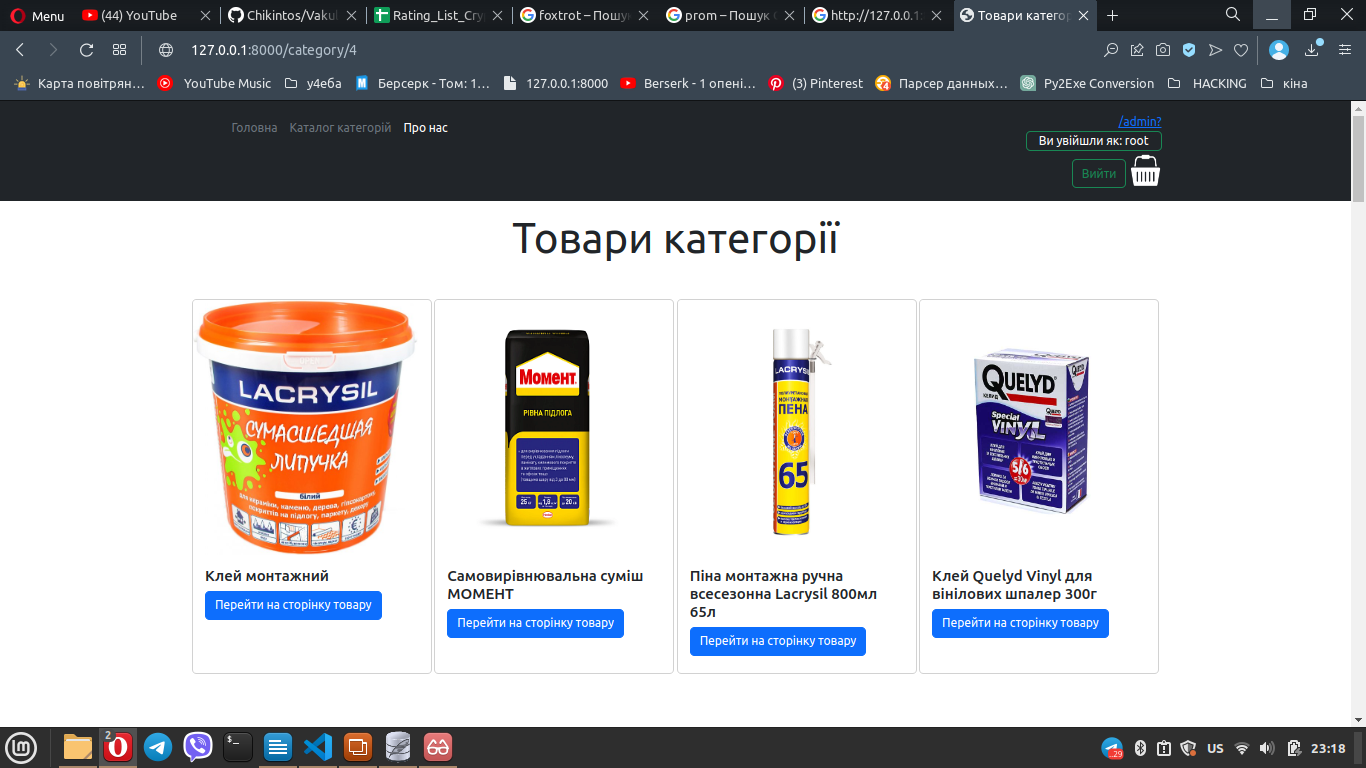
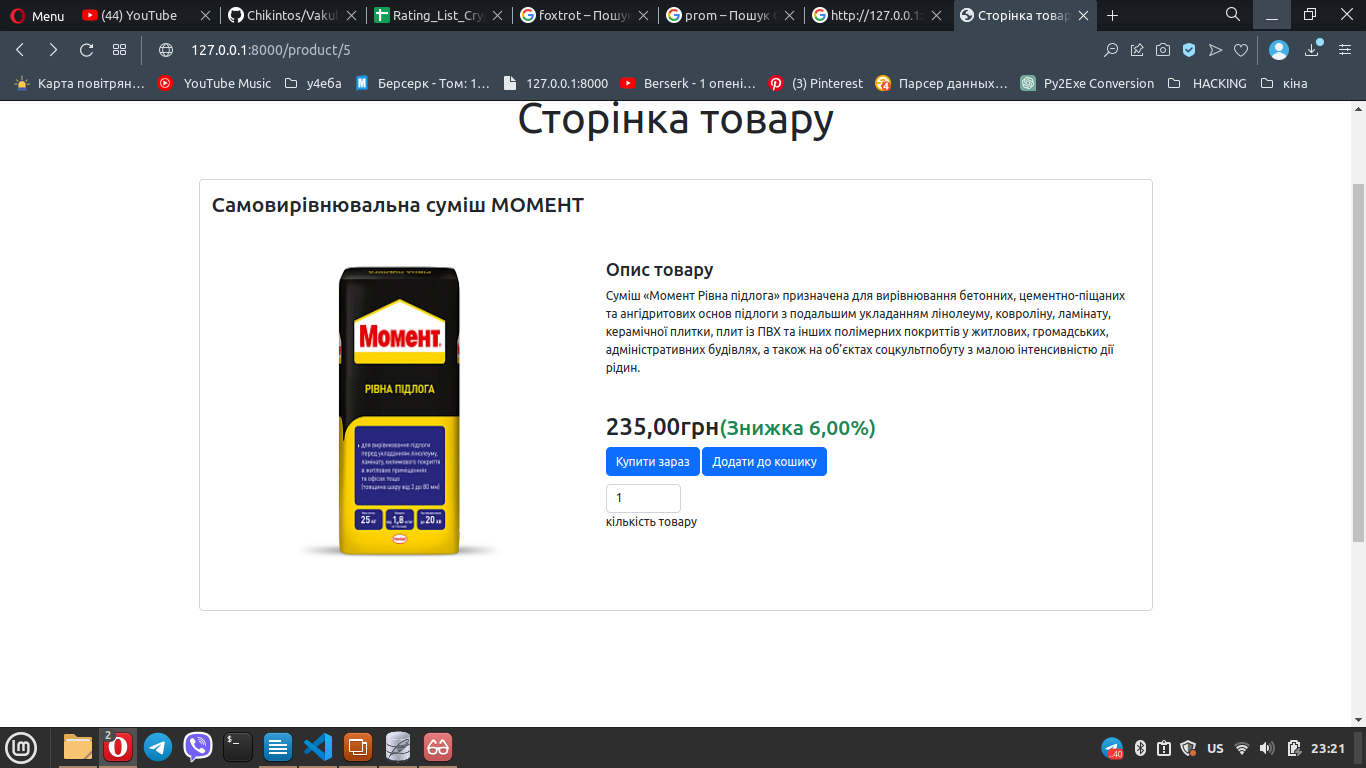
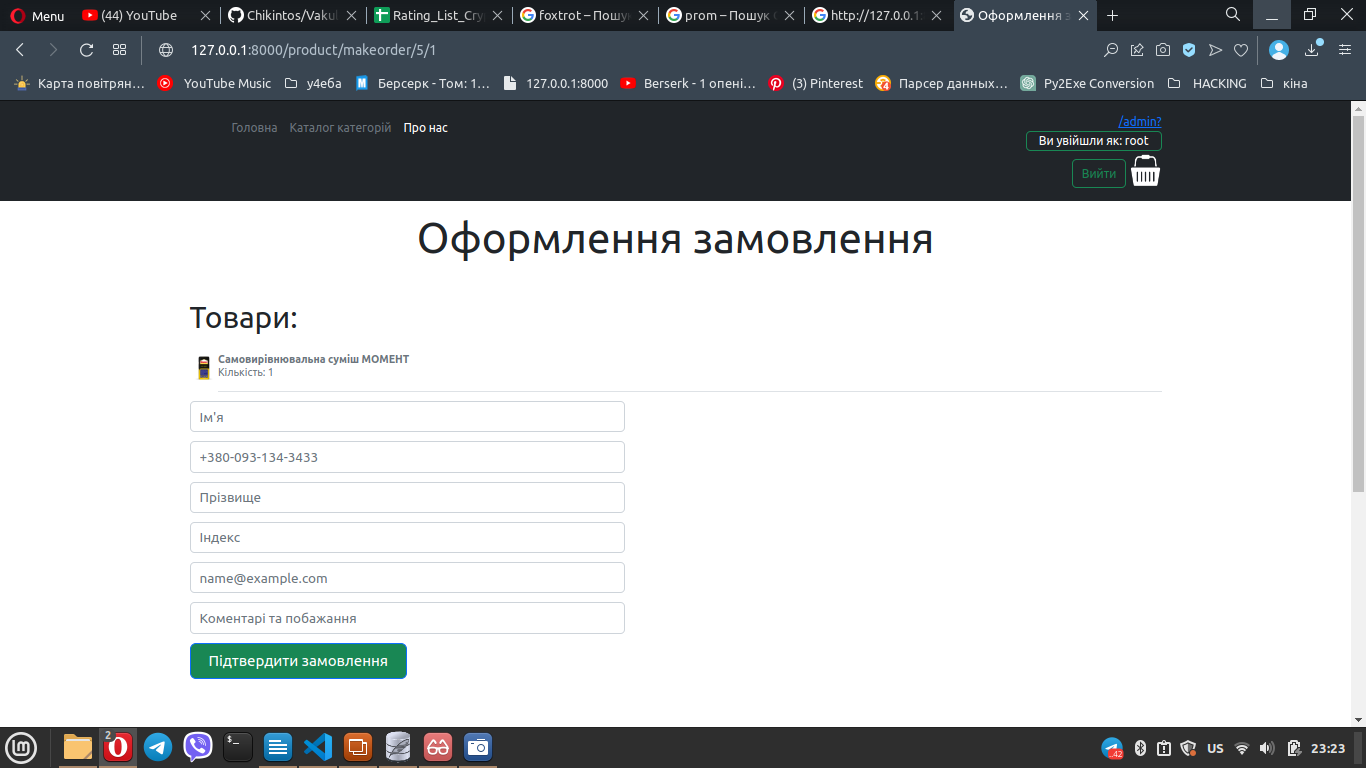


Рисунок 3.2 – «Каталог товарів»

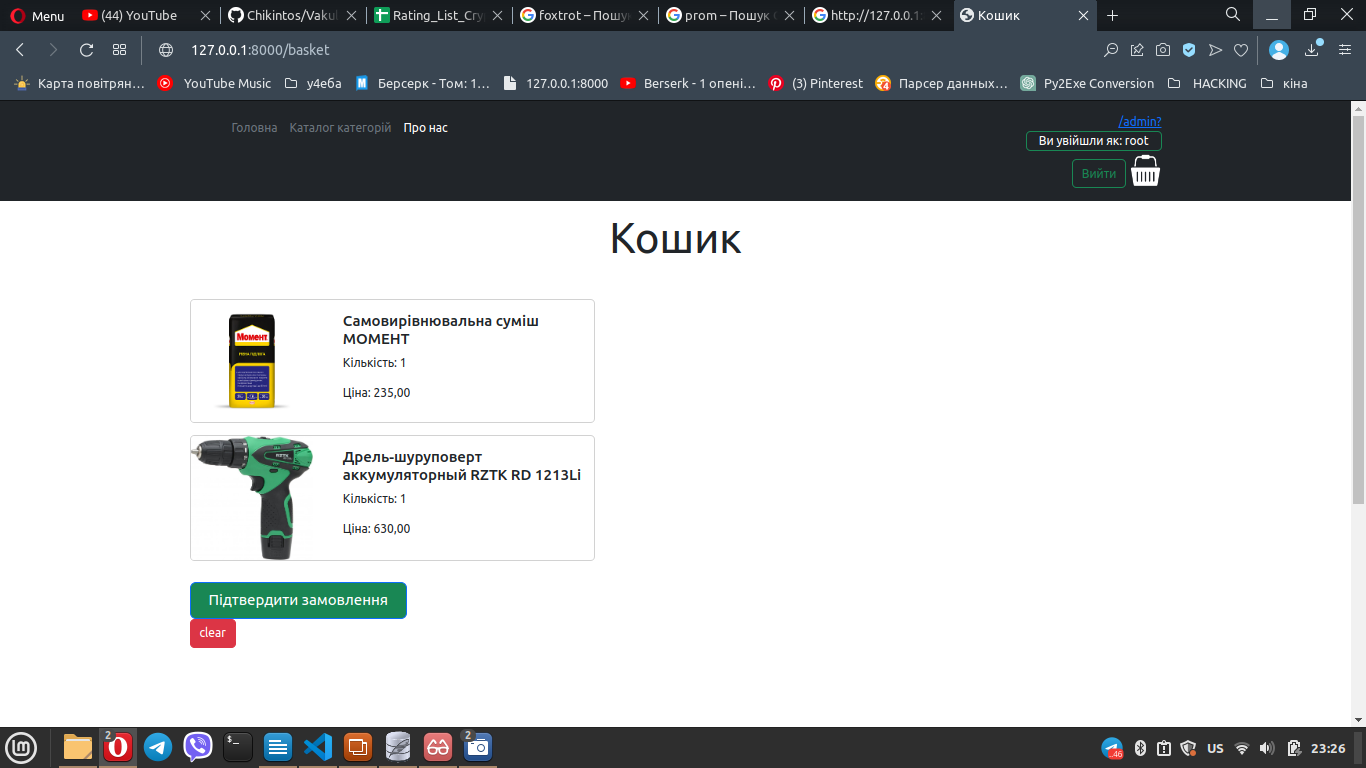
Натиснувши на певний товар можна отримати його сторінку,з більш детальною інформацією

Рисунок 3.3 – «Сторінка товару»

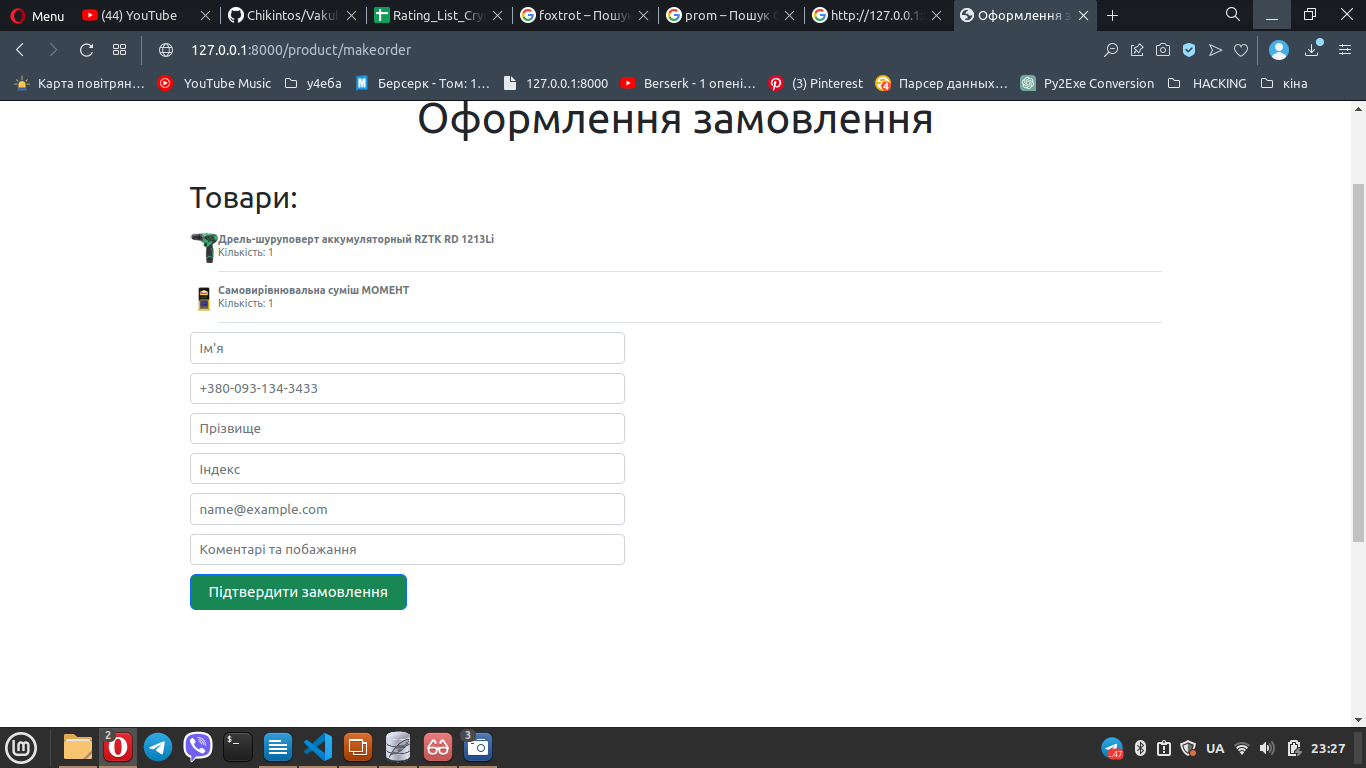
З неї можливо Додати товар до кошику та Замовити його

 Рисунок 3.4 – «Оформлення замовлення»

При додавання декількох товарів до кошику можна зробити групове замолення.

 Рисунок 3.5 – «Заповнений кошик»

Якщо підтвердити замовлення то отримаємо наступну сторінку

 Рисунок 3.6 – «Оформлення замовлення»

Далі розглянемо форми реєстрації(рис. 3.3) та авторизації(рис. 3.5).

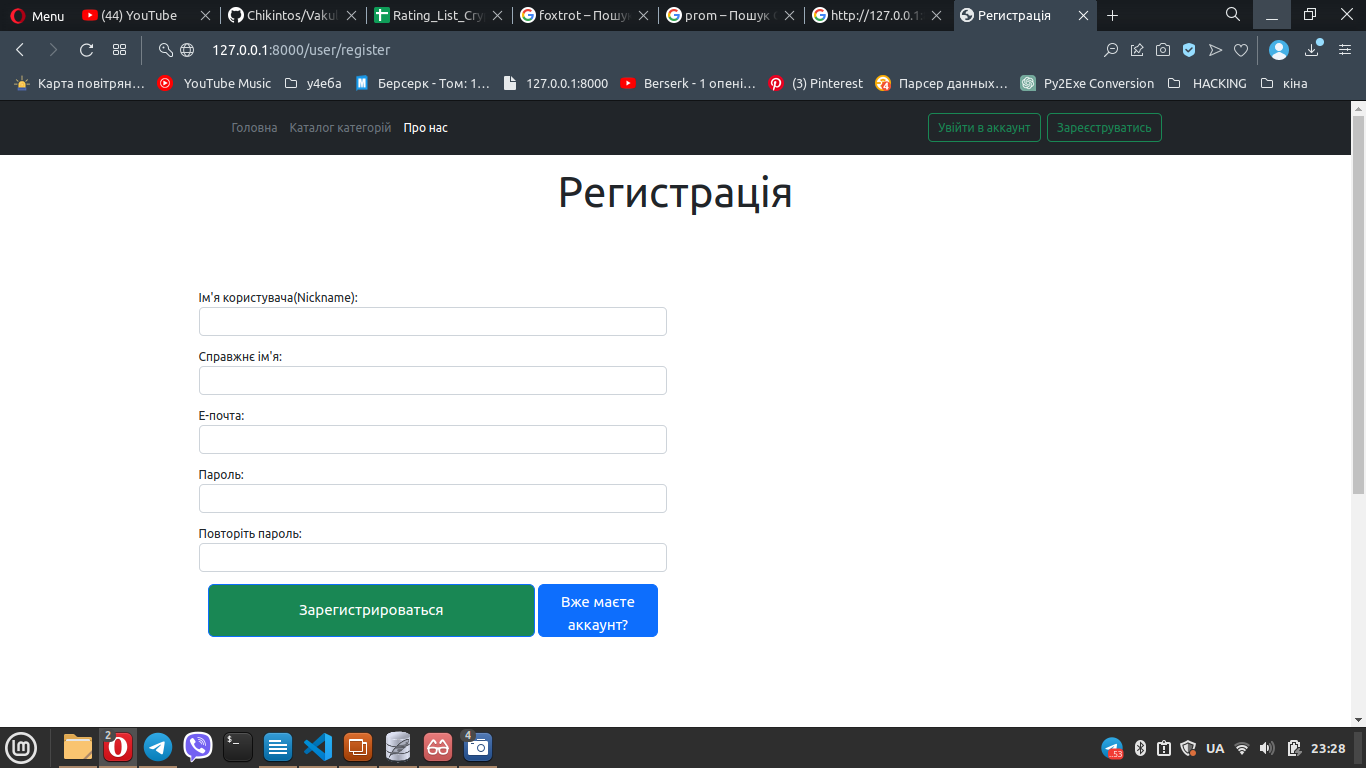


Рисунок 3.7 – Сторінка реєстрації

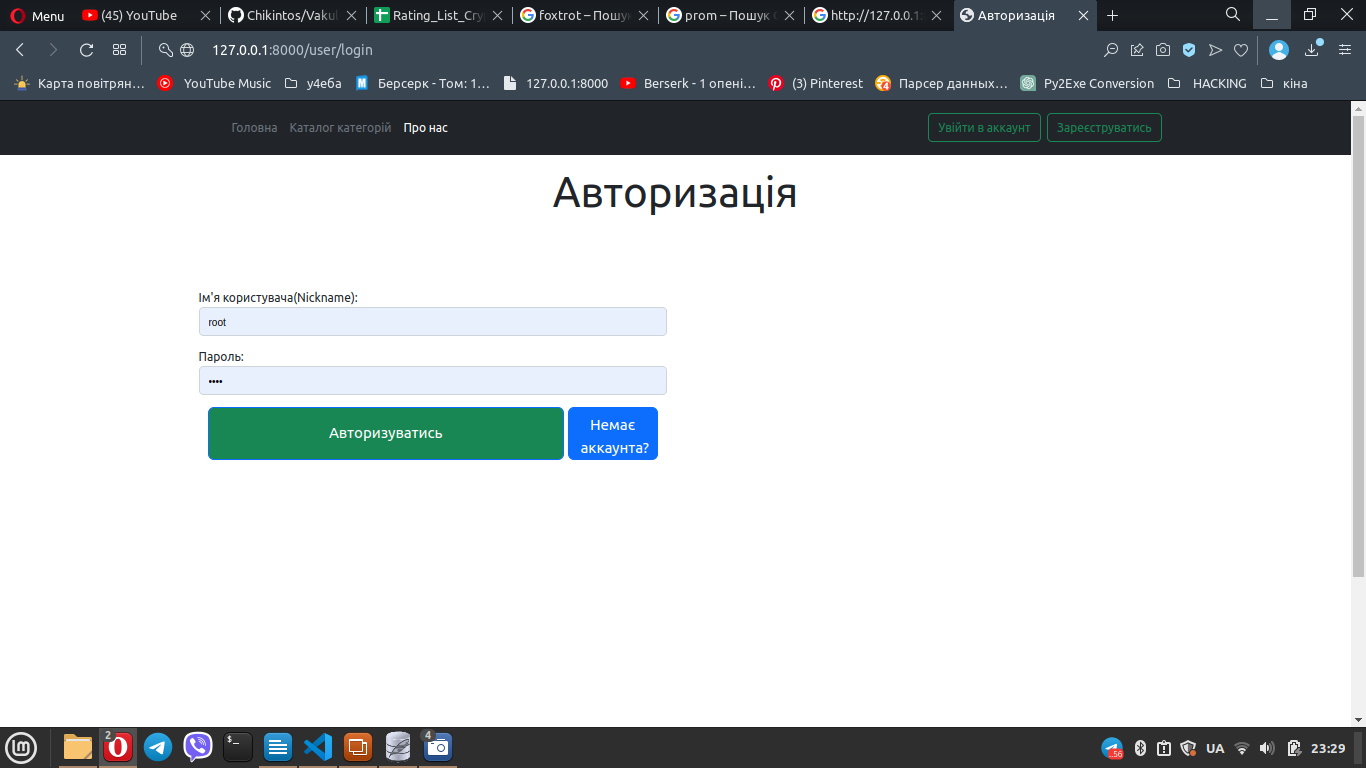


Рисунок 3.4 – Вікно входу

## Висновки до третього розділу

У ході розробки цього розділу було продемонстровано користування веб-додатком. Після його розробки, відбулося тестування. Було змодельовано можливі некоректні ситуації та розглянути всі можливі шляхи розвитку дій користувача. Таким чином вдалось гарантувати стабільну роботу веб-додатку.

# 

# ВИСНОВКИ

Під час виконання курсового проекту нами було закріплено теоретичні знання з HTML, CSS, JavaScript та Python. Та було здобуто багато нових практичних навичок.

Мною був розроблений веб-сайт під назвою «Соціальна мережа» з звикористанням патерну MVT. Було пройдено наступны етапи розробки сайту: моделювання, розробка та тестування.

Під час моделювання було створену схему сторінок веб-сайту, розробка велася за допомогою професійного інструмента VSCode. Тестування сайту проводилось на двох різних браузерах, при цьому було виявлено декілька багів та оперативно виправлено.

У результаті виконання курсової роботи я одержав такі знання:

* базову архітектуру взаємодії сценаріїв на Web-серверах;
* об’єктно - орієнтовані технології розробки прикладних програм для Інтернету;
* можливості сучасних систем обробки Web - сценаріїв;

та зумів:

* оцінив доцільність та можливості розробки додатків на основі Web – технологій;
* самостійно спроектував архітектуру взаємодії сценаріїв заданого призначення;
* створив технологічні програмні продукти на основі Web-сценаріїв;
* врахував вимоги Web - дизайну;
* мінімізував інформаційні потоки з метою найшвидшої роботи Web-додатків.

.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Фрэйн, Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Б. Фрэйн ; [перевод с английского В. Черник]. - Санкт-Петербург [и др.] : Питер, 2014. - 298 с.

2. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. 2-е изд. – СПб.: Питер, 2012. – 304 с.

3. Флэнаган Д. JavaScript. Подробное руководство, 6-е издание – Пер. с англ. – СПб: Символ-Плюс, 2012. – 1080 с.

4. А.Зандстра, Ф.Матт. PHP Objects, Patterns, and Practice, 2013. Режим доступу: <https://www.apress.com/la/book/9781430260325>

5. admin. MS Access и C# [Електронний ресурс] / admin. – 2018. – Режим доступу до ресурсу: https://vscode.ru/prog-lessons/ms-access-i-c-sharp-rabotaem-s- bd.html

6. admin. Все операции с БД в графическом приложении [Електронний ресурс] / admin. – 2015. – Режим доступу до ресурсу: <https://metanit.com/sharp/adonet/3.5.php>

7. Дмитрий Кирсанов. “Веб-дизайн” [електронний ресурс] - 2006. Режим доступу: <http://web-diz.com.ua/skachat/veb-dizayn--dmitriy-kirsanov-skachat-besplatno/>

8. Тим Кедлек. “Адаптивный дизайн. Делаем сайты для любых устройств” [електронний ресурс] - 2013. Режим доступу: <https://www.ozon.ru/context/detail/id/21098804/>

9. Документація бібліотеки JQuery [Електронний ресурс] // JQuery incorp. – 2016. – Режим доступ: https://jquery-docs.ru

10. Выразительный JavaScript [електронний ресурс] - 2017. Режим доступу: <https://legacy.gitbook.com/book/karmazzin/eloquentjavascript_ru/details>

11. Основы наследования [електронний ресурс] - 2015. Режим доступу: https://professorweb.ru/my/ level7/7\_1.php

12. Плюси і мінуси обєктно-орієнтованого програмування [електронний ресурс] - 2015. Режим доступу: <http://damp.biz/plyusi-i-minusi-obyektno-oriyentovanogoprogramuvannya/>

13. SQL [електронний ресурс] - 2016. Режим доступу: <https://uk.wikibooks.org/wiki/SQL>

14. Создание SQL запросов [електронний ресурс] - 2016. Режим доступу: https://www.politerm.com/zuludoc/spatial\_query\_sql.htm#\_sql\_overview

# ДОДАТКИ

## Додаток А

Requirements.txt

aioredis==1.3.1

appdirs==1.4.3

asgiref==3.4.1

asn1crypto==1.4.0

async-timeout==3.0.1

attrs==20.3.0

autobahn==21.3.1

Automat==20.2.0

backports.zoneinfo==0.2.1

CacheControl==0.12.6

certifi==2019.11.28

cffi==1.14.5

channels==3.0.4

channels-redis==3.3.1

chardet==3.0.4

colorama==0.4.3

constantly==15.1.0

contextlib2==0.6.0

cryptography==3.4.7

daphne==3.0.2

distlib==0.3.0

distro==1.4.0

Django==4.0

djangorestframework==3.12.4

gunicorn==20.1.0

hiredis==2.0.0

html5lib==1.0.1

hyperlink==21.0.0

idna==2.8

incremental==21.3.0

ipaddr==2.2.0

lockfile==0.12.2

msgpack==1.0.3

packaging==20.3

pep517==0.8.2

Pillow==8.2.0

progress==1.5

pyasn1==0.4.8

pyasn1-modules==0.2.8

pycparser==2.20

PyHamcrest==2.0.2

pyOpenSSL==21.0.0

pyparsing==2.4.6

pytoml==0.1.21

pytz==2021.1

requests==2.22.0

retrying==1.3.3

service-identity==21.1.0

six==1.14.0

sqlparse==0.4.1

twisted-iocpsupport==1.0.2

Twisted==21.2.0

txaio==21.2.1

urllib3==1.25.8

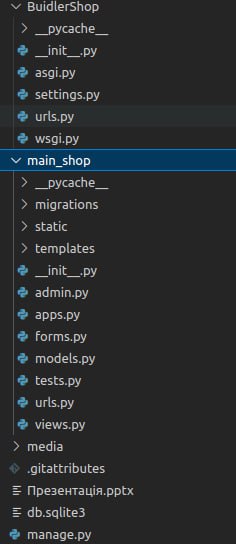
webencodings==0.5.1

whitenoise==5.3.0

zope.interface==5.4.0

## Додаток Б

Повна структура проекту



**Лістинг коду Views.py**

from cProfile import Profile

from re import T

from django.shortcuts import render

import json

from .models import \*

from django.contrib.auth.mixins import LoginRequiredMixin

from django.contrib.auth.forms import UserCreationForm

from django.views.generic.detail import SingleObjectMixin

from django import forms

from django.views.generic.edit import FormView

from .forms import UserRegisterForm

from .forms import UserLoginForm

from .forms import UserDataChangeForm, UserPasswordChangeForm

from django.contrib import messages

from django.shortcuts import redirect, render

from django.contrib.auth import login, logout

from .models import Order\_In\_Wrork, Orders

all\_cats = Categories.objects.all()

all\_products = Products.objects.all()

def index(request):

    discount=Products.objects.order\_by("product\_discount")[:4]

    return render(request, "main\_shop\index.html", {'title': "Головна сторінка", "Categories": all\_cats,"dicountPrd":discount, "Products": all\_products})

def UserLogout(request):

    logout(request)

    return redirect('main')

def Settings(request):

    if request.method == "POST":

        form = UserDataChangeForm(request.POST, instance=request.user)

        if form.is\_valid():

            form.save()

            messages.success(request, "Реєстрація відбулась успішно")

            return redirect('login')

        else:

            form = UserDataChangeForm()

            messages.error(request, "Помилка реєстрації")

    form = UserDataChangeForm(instance=request.user)

    return render(request, "main\_shop\settings.html", {'title': "Редагування профілю", "Categories": all\_cats, "Products": all\_products, "form": form})

def multyOrderToJSON(prdid,prdnum,prdname,prdphotopath):

    OrderDict={}

    sendItem={}

    indx=0

    for i in range(1,len(prdid)+1):

        ordItem={}

        ord1Item={}

        ord1Item["productId"]=prdid[indx]

        ord1Item["productNumber"]=prdnum[indx]

        ordItem["productId"]=prdid[indx]

        ordItem["productNumber"]=prdnum[indx]

        ordItem["productName"]=prdname[indx]

        ordItem["productPhotoPath"]=prdphotopath[indx]

        OrderDict[i]=ordItem

        sendItem[i]=ord1Item

        indx+=1

    jsonDict=json.dumps(OrderDict, indent = 4, ensure\_ascii=False)

    return [OrderDict,sendItem]

def aboutUs(request):

    return render(request, "main\_shop/aboutus.html", {'title': "Про нас", "Categories": all\_cats, "Products": all\_products})

    pass

def clearBasket(request):

    username=request.user

    Basketdel = Orders.objects.filter(user\_id = username)

    Basketdel.delete()

    return redirect('basket')

def OrderPage(request, id=0, num=0):

    if id == 0 and num==0:

        prdid = request.POST['prdId'].split(",")

        prdnum= request.POST['prdnum'].split(",")

        prdname=request.POST['prdName'].split(",")

        prdphotopath=request.POST['product\_img\_path'].split(",")

        prdphotopath=prdphotopath[::2]

        Multyreturn=multyOrderToJSON(prdid,prdnum,prdname,prdphotopath)

        JsonDict=Multyreturn[0]

        sendDict=Multyreturn[1]

        x=2

        return render(request, "main\_shop\order\_page.html", {'title': "Оформлення замовлення", "Categories": all\_cats,"multyOrder":JsonDict ,"Products": all\_products,"SendDict":sendDict,"number": num,"x":x})

    else:

        OrderProduct = Products.objects.get(pk=id)

        x=1

        return render(request, "main\_shop\order\_page.html", {'title': "Оформлення замовлення", "Categories": all\_cats, "Products": all\_products, "ProdInOrder": OrderProduct, "number": num,"x":x})

def orderInfoJSON(id, num):

    order = {}

    order[1] = {"prod\_id": id, "num\_of\_prod": num}

    return order

def saveorder(request):

    order = Order\_In\_Wrork()

    if request.POST['NumOfProduct']=="0":

        order.order\_info = request.POST['orderInfo']

    else:

        order.order\_info = orderInfoJSON(

        request.POST['itemId'], request.POST['NumOfProduct'])

    order.user\_id = request.POST['username']

    order.ord\_surname = request.POST['surname']

    order.phone\_number = request.POST['phone\_number']

    order.adress = request.POST['index']

    order.additional\_info = request.POST['comments']

    order.ord\_name = request.POST['ownname']

    order.save()

    return render(request, "main\_shop/index.html", {'title': 'Сторінка товару', "Categories": all\_cats, "Products": all\_products, "product": product})

def addtobasket(request, id, num, user):

    order = Orders()

    productInitial=str(Products.objects.filter(pk=id).values("product\_price\_initial")).replace("<QuerySet [{'product\_price\_initial': Decimal('", '').replace("')}]>", '')

    productDiscount=str(Products.objects.filter(pk=id).values("product\_discount")).replace("<QuerySet [{'product\_discount': Decimal('", '').replace("')}]>", '')

    print(productInitial)

    print(productDiscount)

    a=float(productInitial)-float(productInitial)/100\*float(productDiscount)

    order.user\_id = user

    order.product\_id = id

    order.num\_of\_products=float(num)

    order.product\_name = str(Products.objects.filter(pk=id).values("product\_name")).replace("<QuerySet [{'product\_name': '", '').replace("'}]>", '')

    order.product\_png = str(Products.objects.filter(pk=id).values("product\_png")).replace("<QuerySet [{'product\_png': '", '').replace("'}]>", '').replace(' ',"")

    order.product\_final\_price=(float(productInitial)-(float(productInitial)/100\*float(productDiscount)))\*num

    order.save()

    print(id)

    return redirect('/product/'+str(id))

def changepassword(request):

    if request.method == "POST":

        form = UserPasswordChangeForm(data=request.POST, user=request.user)

        if form.is\_valid():

            form.save()

            messages.success(request, "Успішно")

            return redirect('main')

        else:

            messages.error(request, "Помилка")

            return redirect('passwordchange')

    else:

        form = UserPasswordChangeForm(user=request.user)

        return render(request, "main\_shop\passchange.html", {'title': "Редагування паролю", "Categories": all\_cats, "Products": all\_products, "form": form})

def registrPanel(request):

    if request.method == "POST":

        form = UserRegisterForm(request.POST)

        if form.is\_valid():

            form.save()

            messages.success(request, "Реєстрація відбулась успішно")

            return redirect('login')

        else:

            form = UserRegisterForm()

            messages.error(request, "Помилка реєстрації")

    else:

        form = UserRegisterForm()

    return render(request, "main\_shop/register.html", {'title': 'Регистрація', "Categories": all\_cats, "Products": all\_products, 'form': form})

def LoginPanel(request):

    if request.method == "POST":

        form = UserLoginForm(data=request.POST)

        if form.is\_valid():

            user = form.get\_user()

            login(request, user)

            return redirect('main')

        else:

            form = UserLoginForm()

            messages.error(request, "Помилка авторизації")

    else:

        form = UserLoginForm()

    return render(request, "main\_shop/login.html", {'title': 'Авторизація', "Categories": all\_cats, "Products": all\_products, 'form': form})

# Create your views here.

def ProductPage(request, id):

    product = Products.objects.get(pk=id)

    return render(request, "main\_shop/product\_page.html", {'title': 'Сторінка товару', "Categories": all\_cats, "Products": all\_products, "product": product})

def basket(request):

    username = request.user

    ordDb = Orders.objects.filter(user\_id=username).values()

    ord = ordDb

    return render(request, "main\_shop/basket.html", {'title': 'Кошик', "Categories": all\_cats, "Products": all\_products, "basketProd": ord})

    print('=================')

def category(request,id):

    query\_cat=Products.objects.filter(Category\_name=id)

    return render(request, "main\_shop/category.html", {'title': 'Товари категорії',"Categories": all\_cats, "Category": query\_cat, "Products": all\_products, "basketProd": ord})

**Лістинг коду models.py**

from distutils.command.upload import upload

from email.policy import default

from tabnanny import verbose

from django.db import models

from numpy import double, maximum, product

from django.core.validators import MinValueValidator, MaxValueValidator

from soupsieve import select

from tomlkit import datetime, value

from django.contrib.auth.models import User

from django.db import models

from django.conf import settings

from django.core.validators import RegexValidator

# Create your models here.

class Categories(models.Model):

    Category\_name = models.CharField("Назва категорії",max\_length=50)

    Category\_desc= models.TextField("Короткий опис товарів у категорії",max\_length=50)

    def \_\_str\_\_(self):

        return self.Category\_name

    class Meta:

        verbose\_name="Категорія"

        verbose\_name\_plural="Категорії"

class Orders(models.Model):

    user\_id=models.CharField("Username",max\_length=200,default=0)

    product\_id=models.CharField("ID продукту",max\_length=200,default=0)

    product\_name=models.CharField("Назва продукту",max\_length=200,default=0)

    product\_png=models.ImageField(upload\_to="media\img\%Y\%m\%d",default=0,null=True,blank=True)

    num\_of\_products=models.DecimalField("кількість товару",max\_digits=15,decimal\_places=0,default=0)

    product\_final\_price=models.DecimalField("Ціна замовлення",max\_digits=8,decimal\_places=2,default=0,validators=[MinValueValidator(0),MaxValueValidator(100)])

    def get\_final\_price(self):

        return round(self.num\_of\_products)

    def \_\_str\_\_(self):

        return str(self.id)

class Products(models.Model):

    product\_name = models.CharField("Назва товару",max\_length=150)

    Category\_name = models.ForeignKey(Categories,on\_delete=models.CASCADE)

    product\_description=models.CharField("опис товару",max\_length=350)

    number\_of\_prouct=models.DecimalField("кількість товару",max\_digits=15,decimal\_places=2,default=0)

    product\_png=models.ImageField(upload\_to="media\img\%Y\%m\%d",default=0,null=True,blank=True)

    product\_price\_initial=models.DecimalField("Початкова ціна",max\_digits=15,decimal\_places=2)

    product\_discount=models.DecimalField("Відсоток знижки",max\_digits=4,decimal\_places=2,default=0,validators=[MinValueValidator(0),MaxValueValidator(100)])

    Date\_of\_get=models.DateField("Дата отримання",auto\_now=True)

    unit\_of\_measure = models.CharField("Одиниці вимірювання",max\_length=150,default='кг')

    def \_\_str\_\_(self):

         return self.product\_name

    def get\_way(self):

        return self.product\_png

    def get\_num(self):

        return round(self.number\_of\_prouct,0)

    def get\_final\_price(self):

        return round(self.product\_price\_initial - self.product\_price\_initial/100 \* self.product\_discount,2)

class Order\_In\_Wrork(models.Model):

        phone\_regex = RegexValidator(regex=r'^\+?1?\d{9,15}$', message="Phone number must be entered in the format: '+999999999'. Up to 15 digits allowed.")

        order\_info=models.CharField("Дані про замовлення",max\_length=2000000,default=0)

        user\_id=models.CharField("Username",max\_length=2000,default=0)

        ord\_name=models.CharField("Ім'я отримувача",max\_length=2000,default=0)

        ord\_surname=models.CharField("Прізвище отримувача",max\_length=2000,default=0)

        phone\_number = models.CharField(validators=[phone\_regex], max\_length=17, blank=True,default=0) # Validators should be a list

        adress =models.CharField("Адреса користувача",max\_length=2000,default=0)

        additional\_info=models.CharField("Додаткова інформація",max\_length=2000,default=0)

        def \_\_str\_\_(self):

            return self.adress

**Додаток В**