

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Програмний додаток-довідник інструментального
виробництва»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ-62 Рибалка Данііл Дмитрович

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ «___» _____ 2020 р.

Науковий керівник

(підпис)

к.т.н., доц., Ващенко С.М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

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

Студент _____
(підпис)

Суми-2020

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ
Зав. секцією ІТП

_____ В. В. Шендрик
«_____» _____ 2020 р.

З А В Д А Н Н Я НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Рибалка Данііл Дмитрович

1 Тема роботи Програмний додаток-довідник інструментального виробництва

керівник роботи Ващенко Світлана Михайлівна, к.т.н., доцент,

затверджені наказом по університету від «14» травня 2020 р. № 0576-III

2 Строк подання студентом роботи «1» червня 2020 р.

3 Вхідні дані до роботи Технічне завдання на розробку програмного додатка-довідника

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз предметної області, Проектування web-додатку, Розробка web-додатку

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
актуальність роботи, аналіз програмних продуктів-аналогів, мета і задачі дипломного проекту, засоби реалізації, демонстрація веб-додатку

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.10.2019

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	до 19.03.20	
2	Ініціювання проекту	до 23.03.20	
3	Планування	до 04.04.20	
4	Аналіз роботи типових додатків	до 10.04.20	
5	Визначення вимог до додатку	до 20.04.20	
6	Визначення інструментарію для реалізації	до 25.04.20	
7	Планування змісту та структури робіт	до 28.04.20	
8	Розробка календарного плану	до 29.04.20	
9	Визначення ризиків	до 30.04.20	
10	Розробка дизайну системи	до 05.05.20	
11	Реалізація	до 29.05.20	
12	Оформлення документації	до 01.06.20	
13	Запуск і підтримка	до 08.06.20	

Студент

(підпис)

Рибалка Д.Д.

Керівник роботи

(підпис)

к.т.н., доц. Ващенко С.М.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Програмний додаток-довідник інструментального виробництва».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, додатків списку використаних джерел із 19 найменувань. Загальний обсяг роботи – 74 сторінок, робота включає в себе 1 таблицю, 19 ілюстрацій.

Кваліфікаційну роботу бакалавра присвячено розробці програмного додатку-довідника інструментального виробництва.

Під час виконання кваліфікаційної роботи бакалавра проведено огляд новітніх досліджень і публікацій щодо сучасної веб-розробки, виконано проектування і моделювання системи, проведено аналіз аналогічних програмних продуктів.

У роботі виконано розробку контекстної діаграми, що демонструє взаємодію процесів в загальному вигляді, без уточнення деталей і подробиць, та розробку діаграми декомпозиції, яка детально описує логіку та послідовність робіт.

Результатом проведеної роботи є розроблений програмний додаток, який надає можливість який дозволить студентам, викладачам і аспірантам кафедри ПМіТКМ швидко знайти виробничий прилад, який їх цікавить та інформацію про нього.

Практичне значення програмного продукту полягає в тому, що ним будуть активно користуватись студенти, викладачі, аспіранти кафедри ПМіТКМ та інші зацікавлені особи.

Ключові слова: веб-додаток, Django, веб-сайт, дипломне проектування, інструментальне виробництво, онлайн довідник.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Огляд останніх досліджень і публікацій	7
1.2 Аналіз програмних продуктів - аналогів.....	8
1.3 Постановка задачі	11
2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ	12
2.1 Моделювання web-додатку в IDEF0.....	12
2.2 Моделювання варіантів використання	15
2.3 Проектування бази даних.....	16
3 РОЗРОБКА WEB-ДОДАТКУ	18
3.1 Архітектура програмного додатку.....	18
3.2 Програмна реалізація.....	19
3.3 Використання програмного додатку.....	27
ВИСНОВКИ	32
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	34
ДОДАТОК А	36
ДОДАТОК Б	43
ДОДАТОК В	54

ВСТУП

Однією з найпоширеніших причин, чому люди щодня проводять багато часу в Інтернеті - це велика кількість інформації, якою вона завантажена. Інформація генерується з декількох джерел і ретельно організовується у вигляді файлів та веб-сторінок, які при згрупуванні разом для формування єдиного об'єкта перетворюються у веб-сайт.

Особливо важливу роль у сучасному веб-просторі грають онлайн-довідники. Використовуючи їх можна отримати необхідну інформацію щодо події, явища, виробу, послуги тощо. Неможливо переоцінити значення онлайн-довідників в умовах дистанційного навчання, викликаного карантинним режимом. Отримати доступ до інформації можна не виходячи з дому і не ризикуючи власним здоров'ям.

Саме на цей тип проекту було отримано замовлення від кафедри прикладного матеріалознавства і технології конструкційних матеріалів факультету технічних систем і енергоефективних технологій Сумського державного університету.

Сформульовано мету проекту – розробити програмний додаток-довідник інструментального виробництва з функціоналом пошуку. Для досягнення мети необхідно вирішити перелік задач:

- проаналізувати і дослідити сучасні web-технології;
- сформулювати перелік вимог до продукту;
- обрати інструменти і методи реалізації задачі;
- спроектувати базу даних;
- створити програмний додаток-довідник;
- провести тестування.

Практичне значення програмного продукту полягає в тому, що ним будуть активно користуватись студенти, викладачі, аспіранти кафедри ПМіТКМ та інші зацікавлені особи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

У всьому світі понад усе цінується якість, тому інформація на сайтах обов'язково повинна бути якісною, цікавою, розгорнутою і корисною. На яку б тему не був створений сайт, він повинен розкривати її максимально широко, від самого початку і до найменших дрібниць. Якщо тематика сайту – web-технології, не достатньо просто описати що таке сайт, персональний комп'ютер і для чого створюються сайти, необхідна наявність інформації сайтобудування в самих різних його проявах, тоді користувачі обов'язково зацікавляться проектом і відвідають його знову і знову, знаючи, що з кожним днем, інформаційний обсяг сайту буде поповнюватися чимось цікавим, і в той же час корисним для відвідувачів.

Звичайно, інформацію можна черпати з різних джерел [7], наприклад з книг, засобів масової інформації, зі слів фахівців, але ж саме інтернет і сайти, дозволяють зберігати цю інформацію для тривалого користування, робити закладки і повертатися до необхідного джерела в будь-який момент. Саме це і робить сайтобудування таким затребуваним, а якісні сайти - успішними, які приносять дохід своїм власникам.

Одним з найважливіших критеріїв є якість основного змісту сторінки. Тобто, якщо метою є продаж товару, сторінка повинна пропонувати детальну інформацію про його характеристики та ціну; якщо вона розважає, вміст повинен бути представлений приємним та привабливим для користувачів Інтернету.

Для досягнення якості потрібні час, зусилля, спеціалізовані знання або дуже специфічні навички.

Щоб визначити цей рівень якості центрального контенту, ми завжди маємо пам'ятати, що тема веб-сайту полягає у встановленні того, що ми очікуємо знайти на порталі, про який йдеться.

Поряд з основним змістом веб-сторінки зазвичай містять низку допоміжних елементів, що збагачують користувацький досвід. Це стосується меню та посилань, які дозволяють розширити інформацію чи пропозиції для доступу до відповідного вмісту.

Завдяки тому, що інтернет є невичерпним джерелом надання різноманітної інформації, створення сайтів дуже актуально, незалежно від обраної тематики. Але варто приділяти увагу не стільки створення свого сайту, скільки його наповнення, тому що актуальним буде тільки якісний сайт, який проб'ється в топ видачі пошукової системи і зацікавить користувача.

1.2 Аналіз програмних продуктів - аналогів

У мережі Інтернет можна знайти велику кількість сайтів-довідників, мета яких – відобразити інформацію щодо шуканого об'єкта.

На рисунках 1.1 – 1.3 зображені сайти з функцією пошуку потрібного об'єкта: медичного препарату, вищого навчального закладу, шкодочинного об'єкта для аграрних культур.

Сайт «Аптеки онлайн» має мінімалістичний дизайн у світлих тонах, наявне поле для введення назви медичного препарату, пошук якого відбувається в базі даних ресурсу. Також є функція відображення медичних препаратів по першій літері назви. Недоліком сайту є те, що права бокова панель занадто довга (1863 px) відносно основної частини контенту, отже сайт здається дещо пустим.



Рисунок 1.1 – Головна сторінка довідника «Аптека онлайн» [11]

Онлайн-ресурс «Довідник ВНЗ» [12] має зручний і зрозумілий інтерфейс користувача. Дизайн даного довідника не викликає питань, що підкреслює його практичність. Обрати регіон, напрям та спеціальність можна завдяки випадуючого списку. Також у наявності навігаційний ланцюг (breadcrumbs).

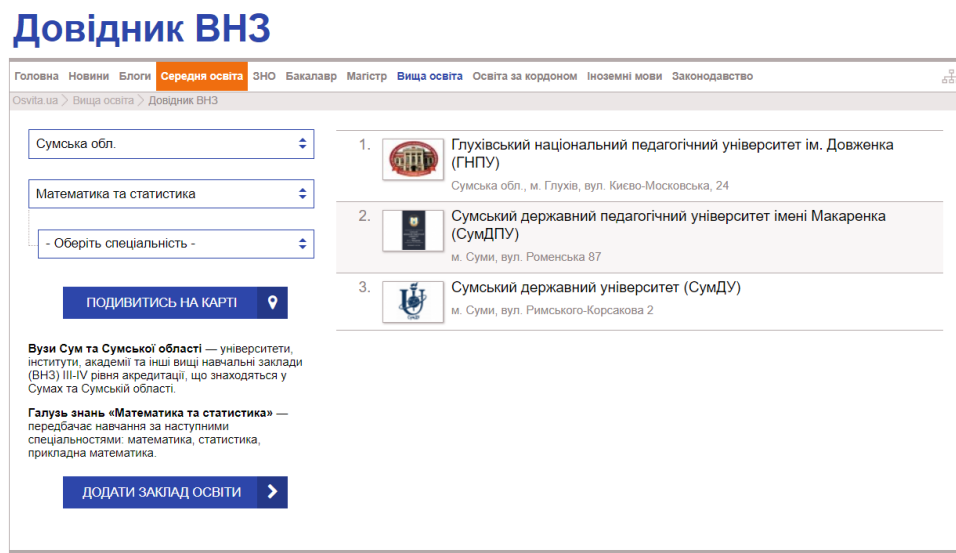


Рисунок 1.2 – Сторінка довідника українських ВНЗ

Онлайн-довідник шкодочинних об'єктів [10] використовує в якості механізму пошуку як випадуючий список, так і поле для введення тексту. Присутній навігаційний ланцюг. Є функція відображення рослинної культури по першій літері назви. Інтерфейс даного сайту не є інтуїтивно зрозумілим будь-якому користувачу.

Відображення деяких символів (апостроф, крапка з комою тощо) відбувається некоректно.

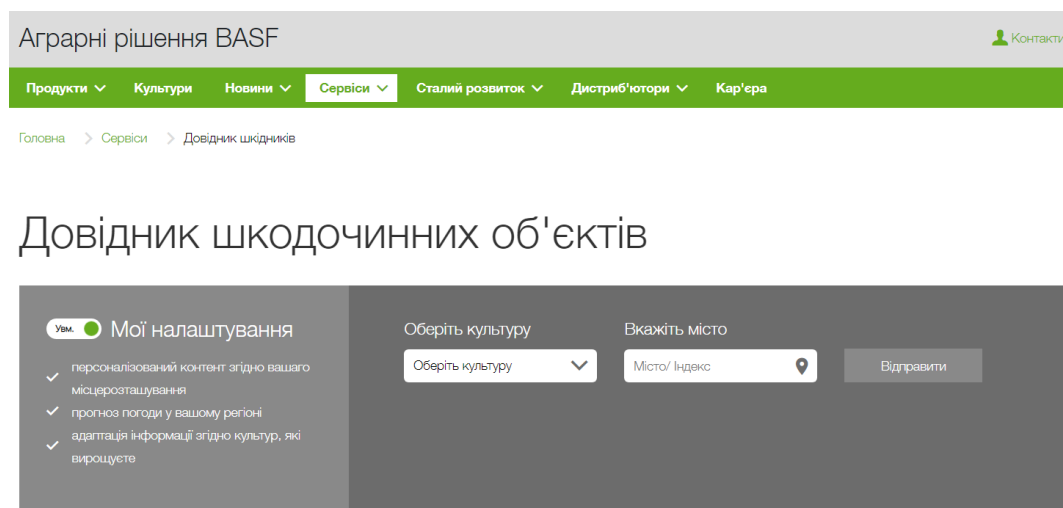


Рисунок 1.3 – Сторінка довідника шкочинних об'єктів

Спільного у продемонстрованих сторінок онлайн-довідників є те, що вони мають невимушений легкий дизайн, поле або інший механізм для пошуку, відображення інформації щодо цікавлячого об'єкта.

Порівняння сайтів-аналогів наведено в таблиці 1.

Таблиця 1 – Порівняльна характеристика аналогів

Критерій	Назва сайту		
	«Аптека онлайн»	Довідник українських ВНЗ	Довідник шкочинних об'єктів
Поле введення	Так	Ні	Так
Навігаційний ланцюг	Ні	Так	Так
Випадаючий список	Ні	Так	Так
Інтуїтивний інтерфейс	Так	Так	Ні
Відображення об'єкту по першій літері назви	Так	Ні	Так
Кольорова гама	Світла	Світла	Світла
Наявність недоліків	Так	Ні	Так

1.3 Постановка задачі

Основною метою даної роботи є розробка веб-додатку для сайту кафедри прикладного матеріалознавства факультету технічних систем і енергоефективних технологій, який дозволить студентам і аспірантам швидко знайти цікавлячий їх виробничий прилад та інформацію про нього. Для створення даного веб-ресурсу необхідна база даних, котра буде зберігати інформацію щодо адміністрації сайту, категорій виробів, зображень виробів, найменувань виробів тощо.

У якості основної технології створення проекту було обрано фреймворк Django [1] мови програмування Python [6] через налаштованість «з коробки» адміністративної панелі та філософію фреймворка:

- Django [2] намагається зробити кожен елемент проекту незалежним від інших;
- менше коду, а отже швидкий темп розвитку;
- код не повинен повторюватись;
- сувора підтримка чистого дизайну у власному коді.

У якості бази даних під час розробки було обрано MySQL через велику кількість готових методів роботи з нею. Проте на етапі завантаження програмного додатку на хмарну платформу Heroku в якості бази даних було обрано PostgreSQL через глибоку інтегрованість цих систем та безпосередні рекомендації розробників Heroku.

Повна та детальна інформація про вимоги представлена в технічному завданні, яке було розміщене у додатку А. Планування робіт представлено у додатку Б.

2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ

2.1 Моделювання web-додатку в IDEF0

IDEF0 [4] – це методологія моделювання функцій для опису виробничих функцій, яка пропонує функціональну мову моделювання для аналізу, розробки, реінжинірингу та інтеграції інформаційних систем; бізнес-процеси; або аналіз інженерного програмного забезпечення.

IDEF0 [5] є частиною сімейства мов моделювання IDEF у галузі інженерії програмного забезпечення та будується на технічній структурі методів структурованого аналізу та дизайну (SADT).

Елементи контекстної діаграми.

- Вхід – це споживана або змінювана роботою інформація або матеріал.
- Вихід – інформація або матеріал, які виробляються роботою.
- Управління – процедури, правила, стратегії або стандарти, якими керується робота.
- Механізми – ресурси, які виконують роботу.

Провівши аналіз відносно головних елементів для контекстної діаграми «Пошук довідкової інформації інструментального виробництва», було створено наступний перелік даних.

- Вхід: «Запит користувача на пошук інформації про інструментальний виріб».
- Вихід: «Відображення інформації щодо виробу».
- Управління: «Технічне завдання», «Документація Django».
- Механізми: «Електронний довідник», «Програмне забезпечення», «Апаратне забезпечення», «Веб-сервер».

На основі цих даних, була розроблена контекстна діаграма у програмному продукті Erwin Process Modeler, що представлена на рис. 2.1.

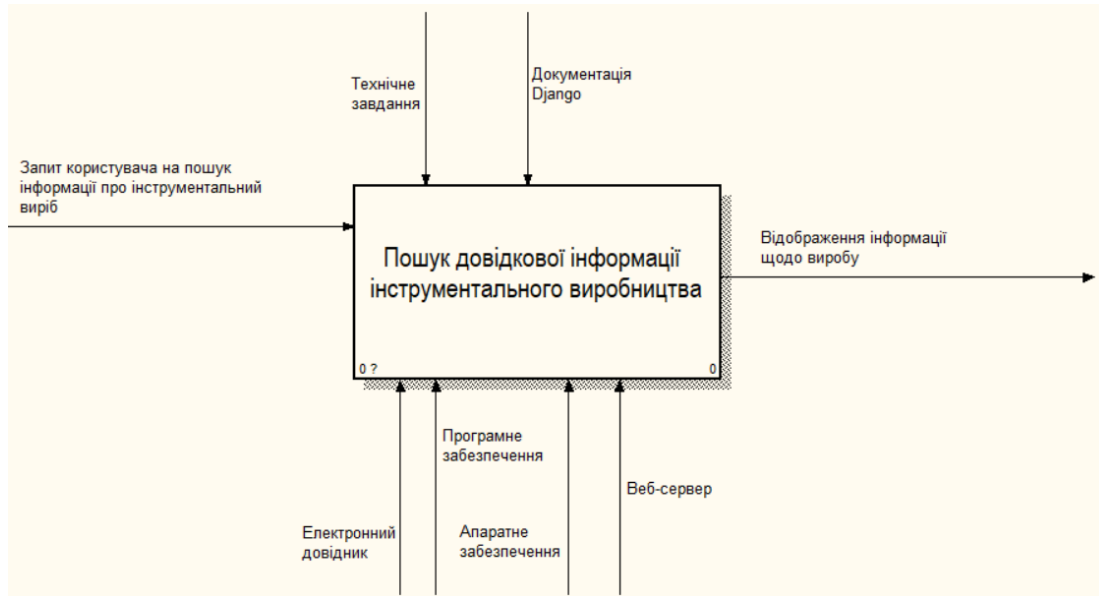


Рисунок 2.1 – Контекстна діаграма

Отримана діаграма демонструє лише взаємодію процесів в загальному вигляді, без уточнення деталей і подробиць. Для більш детального опису логіки та послідовності робіт необхідно виконати декомпозицію (рис. 2.2).

«Пошук довідкової інформації інструментального виробництва» декомпозується на три процеси – «Вибір категорії виробів», «Пошук виробу», «Вибір необхідного виробу».

Процес «Вибір категорії виробів» на вході має запит користувача на пошук інформації про інструментальний вибір. На виході ми отримуємо натискання кнопки «Детальніше». Управління процесом здійснюється згідно технічного завдання, а також документації Django. Механізмами процесу виступають електронний довідник, програмне забезпечення, апаратне забезпечення і веб-сервер.

Вхідною інформацією для «Пошук виробу» є запит користувача на пошук інформації про інструментальний вибір. На виході ми отримуємо вивід списку

виробів. Механізмом процесу виступає електронний довідник, програмне забезпечення, апаратне забезпечення і веб-сервер.

На вхід процесу «Вибір необхідного виробу» подається натискання кнопки «Детальніше». Механізмом є електронний довідник, програмне забезпечення, апаратне забезпечення і веб-сервер. На виході отримується відображення інформації щодо виробу.

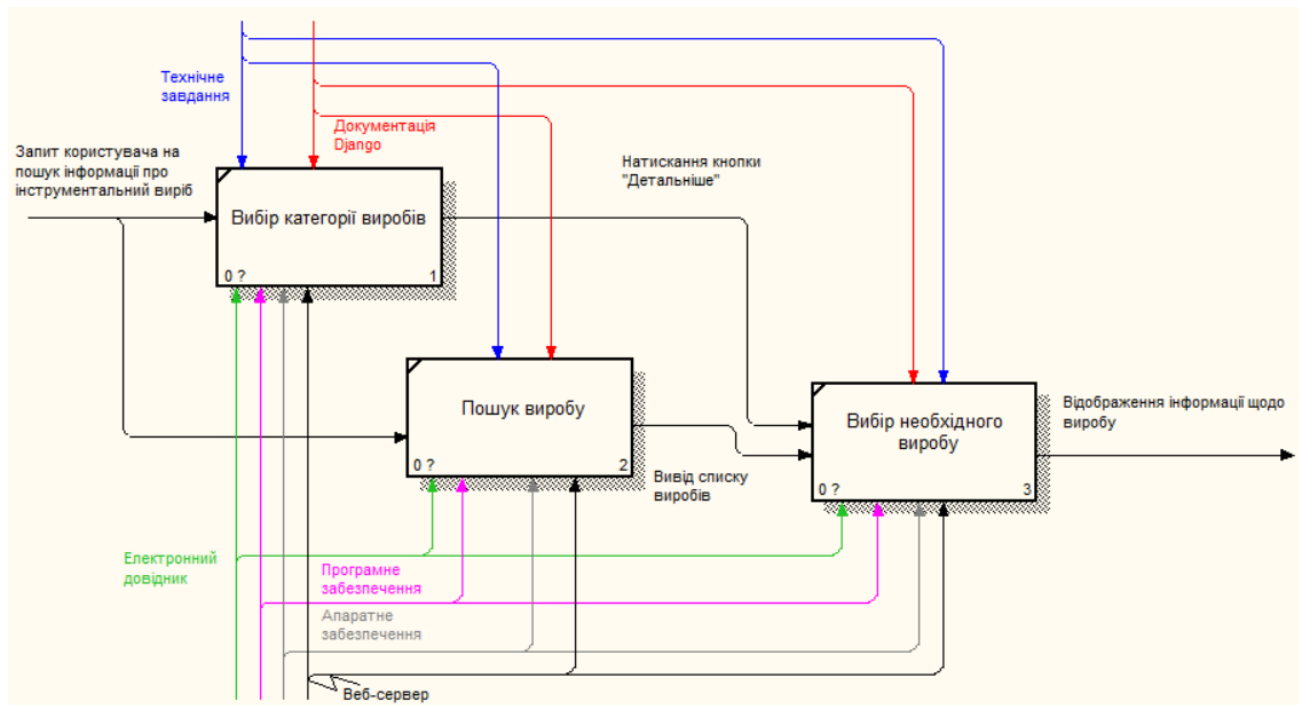


Рисунок 2.2 – Діаграма декомпозиції

2.2 Моделювання варіантів використання

Для кращого розуміння веб-додатку виконаємо моделювання варіантів використання [14]. Схема варіантів використання зображена на рисунку 2.3.

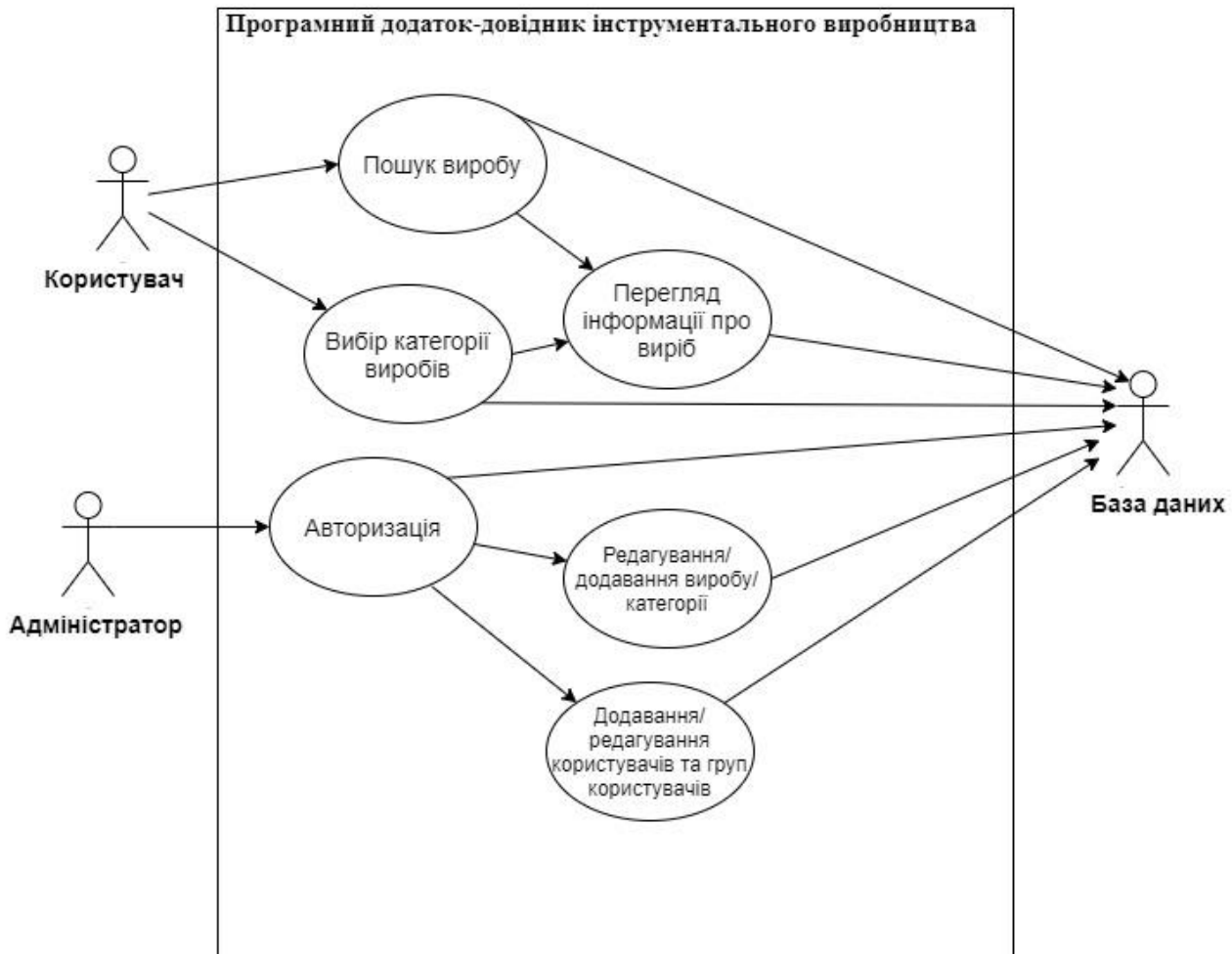


Рисунок 2.3 – Діаграма варіантів використання

Акторами є користувачі та адміністратор, які взаємодіють з веб-додатком. Пошук вибору – це процес вводу користувачем назви необхідного виробу. Вибір категорії виробів – це процес вибору користувачем запропонованих категорії виробів. Перегляд інформації про виріб – це процес ознайомлення користувача з інформацією про виріб. Авторизація – це процес автентифікації адміністратора в адмін-панель веб-додатку. Реагування/додавання виробу/категорії – це процес

оновлення та додавання інформації про виріб та категорію адміністратором. Додавання / реагування користувачів та груп користувачів - це процес додавання нових адміністраторів та оновлення про них інформації, та створення для них груп. Для виконання процесів необхідно звертатися до бази даних.

2.3 Проектування бази даних

Для реалізації веб-додатку було визначено два класи в моделі Django. Вони собою являють не що інше, як сутності бази даних:

- Category – зберігає інформацію про категорії інструментальних виробів;
- Item – зберігає інформацію про вироби.



#	Имя	Тип
1	id 	int(11)
2	title	varchar(255)
3	text	longtext
4	img	varchar(100)
5	category_id 	int(11)
6	slug	varchar(100)

Рисунок 2.4 – Сутність Item


#	Имя	Тип
1	id 	int(11)
2	title	varchar(255)
3	img	varchar(100)

Рисунок 2.5 – Сутність Category

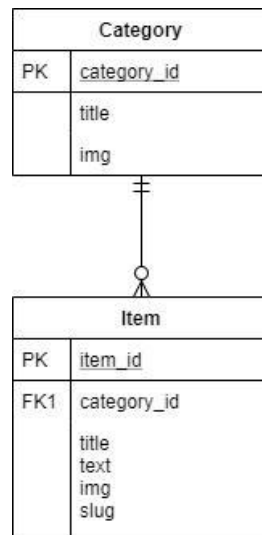


Рисунок 2.6 – Діаграма бази даних

3 РОЗРОБКА WEB-ДОДАТКУ

3.1 Архітектура програмного додатку

Архітектура розробленого сайту показано на рис. 3.1.

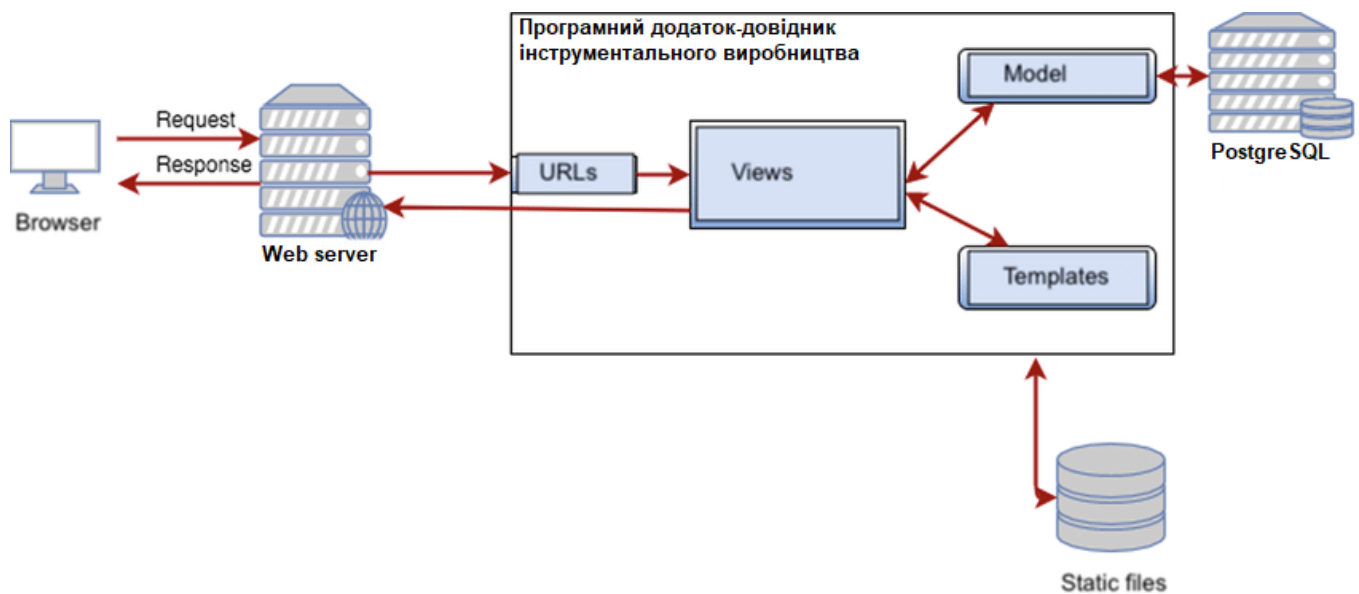


Рисунок 3.1 - Архітектура системи

Браузер взаємодіє з веб-сервером, надсилаючи йому запити. Сервер оброблює запит і передає інформацію Django додатку. Архітектура MVT (Model-View-Template) дозволяє розробнику працювати з візуальним представленням і логікою додатка окремо. Компоненти MVT можна використовувати незалежно один від одного.

Документація Django визначає модель (model) як «джерело інформації про дані, в яких містяться ключові поля і поведінку даних». Зазвичай одна модель вказує на одну таблицю в базі даних. Django підтримує бази даних PostgreSQL, MySQL, SQLite і Oracle.

Моделі містять інформацію про дані. Ці дані представлені атрибутами або полями. Оскільки модель являє собою простий клас, вона нічого не знає про інші рівні Django. Взаємодія між рівнями відбувається через API. Модель відповідає за логіку, методи, властивості і інші елементи, пов'язані з маніпуляцією даними. Також моделі дозволяють розробникам створювати, читати, оновлювати та видаляти об'єкти в базі даних.

Представлення (view) вирішує три завдання: приймає HTTP-запити, реалізує логіку, визначену методами і властивостями, відправляє HTTP-відповідь на запити. Тобто представлення отримує дані від моделі і надає шаблонам (templates) доступ до цих даних або попередньо обробляє дані і потім надає до них доступ шаблонам.

У Django реалізований потужний движок шаблонів і власну мову розмітки. Шаблони являють собою файли з HTML-кодом, за допомогою якого відображаються дані. Вміст файлів може бути статичним або динамічним. Шаблони не містять логіки. Тому вони тільки відображають дані.

3.2 Програмна реалізація

Перед початком реалізації знаходячись у каталозі «work» було використано команду «django-admin startproject ToolProduction». Це створило каталог проекту «ToolProduction», у якому міститься скрипт «manage.py» і підпапка з ідентичною проекту назвою, в якій знаходяться файли «settings.py», «urls.py», «wsgi.py». Скрипт «manage.py» використовується для створення додатків, роботи з базами даних і для запуску сервера налагодження.

Наступним кроком було створено додаток «directory», використавши команду «python manage.py startapp directory». Ця команда створила нову папку і наповнила її файлами різних частин програми. Більшість файлів названі, виходячи з їх призначення (наприклад контролери (views) повинні знаходитися під views.py,

моделі в `models.py`, тести в `tests.py`, настройки адміністративної частини в `admin.py`, реєстрація додатків в `apps.py`) і вже містять деякий шаблонний код для роботи з вищеназваними об'єктами.

Після створення додатку потрібно зареєструвати його в проєкті, щоб різні утиліти зачіпали його своєю дією (наприклад при додаванні моделей в базу даних). Додатки реєструються додаванням їх назв в список `INSTALLED_APPS` у настройках проєкту (скрипт `settings.py`).

Під час розробки веб-додатку було використано базу даних MySQL:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'mydjango',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
```

Для роботи з цією базою даних у Python необхідно завантажити інтерфейс «`mysqlclient`». Для цього було виконано команду «`pip install mysqlclient`».

При створенні сайту, був створений файл зіставлення URL (`urls.py`) в корені проєкту. Хоча можна використовувати його для обробки всіх URL адрес, більш доцільно підключати окремі файли зіставлень для кожної програми.

Наведені нижче рядки (файл «`urls.py`») додають новий елемент до списку «`urlpatterns`». Цей елемент містить `url ()` який направляє запити з пустого URL до модулю «`directory.urls`»:

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path(' ', include('directory.urls')),
]
```

Запити з URL 'admin/' ініціюють виконання функції «admin.site.urls», яка згенерує сторінку адміністративної панелі.

Django не розташовує статичні файли (CSS, JavaScript, і зображення) за замовчуванням, але це вкрай корисно на етапі розробки сайту. В самому кінці URL діспетчера можна включити розміщення статичних файлів:

```
if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL,
document_root=settings.MEDIA_ROOT)
    urlpatterns += static(settings.STATIC_URL,
document_root=settings.STATIC_ROOT)
```

Всередині каталогу «directory» необхідно створити файл «urls.py» і додати код, щоб визначити відокремлений «urlpatterns»:

```
from django.urls import path
from .views import *

app_name = 'directory'
urlpatterns = [
    path("", directory, name='home'),
    path('s/', search, name='пошук'),
    path('metalproduction/', metalproduction, name='Вироби металургійного
виробництва'),
    path('machinebuild/', machinebuild, name='Вироби машинобудівного
виробництва'),
    path('instrproduction/', instrproduction, name='Вироби інструментального
виробництва'),
    path('carbuild/', carbuild, name='Вироби автомобілебудівного виробництва'),
    path('traktorbuild/', traktorbuild, name='Вироби тракторного та
сільськогосподарського машинобудування'),
    path('chembuild/', chembuild, name='Вироби хімічного машинобудування'),
    path('metalproduction/<str:slug>', metalproductionPost,
name='metalproductionPost'),
    path('machinebuild/<str:slug>', machinebuildPost, name='machinebuildPost'),
    path('instrproduction/<str:slug>', instrproductionPost,
name='instrproductionPost'),
```

```

    path('carbuild/<str:slug>', carbuildPost, name='carbuildPost'),
    path('traktorbuild/<str:slug>', traktorbuildPost, name='traktorbuildPost'),
    path('chembuild/<str:slug>', chembuildPost, name='chembuildPost'),
]

```

Моделі зазвичай визначаються в файлі додатку «models.py». Вони реалізуються як підкласи «django.db.models.Model», і можуть включати поля, методи і метадані. Модель категорій інструментальних виробів:

```

from django.db import models

class Category(models.Model):
    title = models.CharField(max_length=255)
    img = models.ImageField(null=True, blank=True, upload_to='images/category/',
        verbose_name='Зображення категорії')

    class Meta:
        verbose_name_plural = "категорії"

    def __str__(self):
        return self.title

```

Всі необхідні налаштування, які необхідно включити в «admin» додаток веб-сайту, були зроблені автоматично, коли створювався каркас проекту. В результаті все, що необхідно зробити для того, щоб додати моделі в додаток «admin», це зареєструвати їх. Файл «admin.py»:

```

from django.utils.html import format_html
from .models import *
from django.contrib import admin
from django.conf import settings

class ProductAdmin(admin.ModelAdmin):
    def picture(self, obj):
        return format_html(f'')

```

```
picture.allow_tags = True
list_display = ('title', 'picture')
```

```
admin.site.register(Category, ProductAdmin)
admin.site.register(Item, ProductAdmin)
```

У цьому файлі реєструються моделі додатку в адміністративній панелі та форматується відображення зображення моделі у адмін-панелі.

Відображення є функцією, яка обробляє HTTP-запит, отримує дані з бази даних (при необхідності), які застосовуються для генерації сторінки HTML. Файл «views.py»:

```
from .models import *
from django.shortcuts import render

def directory(request):

    obj1 = Category.objects.get(id=1)
    obj2 = Category.objects.get(id=2)
    obj3 = Category.objects.get(id=3)
    obj4 = Category.objects.get(id=4)
    obj5 = Category.objects.get(id=5)
    obj6 = Category.objects.get(id=6)
    context = {
        "data1": obj1,
        "data2": obj2,
        "data3": obj3,
        "data4": obj4,
        "data5": obj5,
        "data6": obj6,
    }
    return render(request, 'home.html', context)
```

Функція «directory» генерує зображення шаблону «home.html», передаючи в нього словник «context», що включає в себе дані про категорії виробів.

Шаблон це текстовий файл, який визначає структуру і розташування даних у файлі, крім того, в ньому розміщують спеціальні мітки (placeholders), які використовуються для відображення реального вмісту, тобто даних. За замовчуванням Django шукає файли шаблонів у директорії з ім'ям 'templates' всередині додатку. Шаблон головної сторінки сайту повинен відповідати стандарту розмітки HTML для розділів «head» і «body», крім того мати розділи для навігації і показу деякого вступного тексту. Велика частина цієї структури буде однаковою для всіх сторінок нашого сайту. Таким чином, щоб уникнути копіювання однієї і тієї ж інформації, мова створення шаблонів Django дозволяє оголосити базовий шаблон, а потім розширити його, заміщаючи тільки ті частини, які є специфічними для кожної сторінки.

Наприклад, базовий шаблон «base.html» може виглядати як показано нижче:

```
<!DOCTYPE html>
<html lang="en">
<head>
  {% block title %}<title>Local Library</title>{% endblock %}
</head>

<body>
  {% block sidebar %}<!-- insert default navigation text for every page -->{%
endblock %}
  {% block content %}<!-- default content text (typically empty) -->{% endblock
%}
</body>
</html>
```

Як видно, цей файл містить деяку "загальну" структуру HTML, розділи для заголовка, панель навігації та вміст, зазначене тегамі шаблону «block» і «endblock». Дані блоки можуть бути порожніми, або мати вміст, який буде використовуватися "за замовчуванням" усіма сторінками-спадкоємцями. Файл «chembuild_post.html»:

```
{% extends 'chembuild.html' %}
{% block title %} {{ item.title }} {% endblock title %}
```



```

{% block content % }
  <style>
    .breadcrumb{display:-ms-flexbox;display:flex;-ms-flex-wrap:wrap;flex-
wrap:wrap;padding:.15rem 1rem;margin-bottom:1rem;list-style:none;
      background-color:#e9ecef;border-radius:.25rem}
    .breadcrumb p::before{
      content: "►";
    }
  </style>
{% block breadcrumbs % }
  <div class="breadcrumb">
    <a href="/">Категорії</a>
    <p></p>
    <a href="/chembuild">Вироби хімічного машинобудування</a>
    <p>{{ item.title }}</p>
  </div>
{% endblock breadcrumbs % }
  <h2 style="margin-left:10px;">{{ item.title }}</h2>
  
  <p style="border: 1.5px solid gray;padding: 5px; ">{{ item.text }}</p>
{% endblock content % }

```

Вищезазначений шаблон наслідує шаблон «chembuild.html», заповнюючи блоки «title», «content», «breadcrumbs» даними з бази даних.

У даному додатку реалізовано механізм пошуку даних із бази даних. Файл «home.html»:

```

...
<form class="example" action="/s/">
  <input type="text" placeholder="Шукати.." name="q" role="search">
  <button type="submit">Пошук</button>
</form>
...

```

При натисканні на кнопку «Пошук» виконується співставлення URL-адрес, що призводить до виконання функції «search». Файл «views.py»:

```

...
def search(request):
    try:
        q = request.GET.get('q')
    except:
        q = None
    if q:
        items = Item.objects.filter(Q(title__icontains=q) | Q(text__icontains=q))
        paginator = Paginator(items, 4)
        page_number = request.GET.get('page', 1)
        page_obj = paginator.get_page(page_number)
        context = {'query': q, 'items': page_obj.object_list, 'paginator': paginator,
'page_number': int(page_number), }
    else:
        return redirect('directory:home')
    return render(request, 'results.html', context)
...

```

Якщо форма пошуку була порожня, то при виконанні пошуку відбувається перенаправлення на поточну сторінку. Якщо дані були введенні, то для них будуть виконані умови фільтрації по назві і по тексту виробу. Якщо назва виробу або текст щодо нього включає в себе введенні дані, буде зображено «results.html» з результатом пошуку. Якщо пошук виявився нерезультативним, буде відображено відповідний надпис.

Реалізовано механізм навігаційних ланцюгів (breadcrumbs). Це горизонтальне відтворення гіперпосилань на сторінки з верхніх рівнів ієрархії. Файл «carbuild_post.html»:

```

...
{% block breadcrumbs %}
    <div class="breadcrumb">
        <a href="/">Категорії</a>
        <p></p>
        <a href="/carbuild">Вироби автомобілебудівного виробництва</a>
        <p>{{ item.title }}</p>

```

```

</div>
{% endblock breadcrumbs % }
...

```

Реалізовано механізм пагінації. Файл «results.html»:

```

...
{% if items % }
  <ul class="pagination">
    {% for i in paginator.page_range % }
      <li class="page-item {% if i == page_number % } active {% endif % }">
        <a class="page-link" href="?q={{ query }}&page={{ i }}">{{ i }}</a>
      </li>
    {% endfor % }
  </ul>
{% endif % }
...

```

Згідно вищезазначеному фрагменту коду, якщо пошук не виявився безрезультатним, будуть відображатися елементи списку, що являють собою кнопки сторінок.

Для коректної роботи з базою даних MySQL під час розробки додатку було використано безкоштовну збірку веб-серверу XAMPP [13]. Розроблений веб-додаток було завантажено на хмарну PaaS-платформу Heroku [18]. У процесі завантаження базу даних було змінено на PostgreSQL.

3.3 Використання програмного додатку

Розроблений додаток не розмежовує права використання на типи користувачів, проте має функціональну адміністративну панель. Щоб увійти в неї необхідно ввести ім'я користувача і пароль (рис. 3.2).

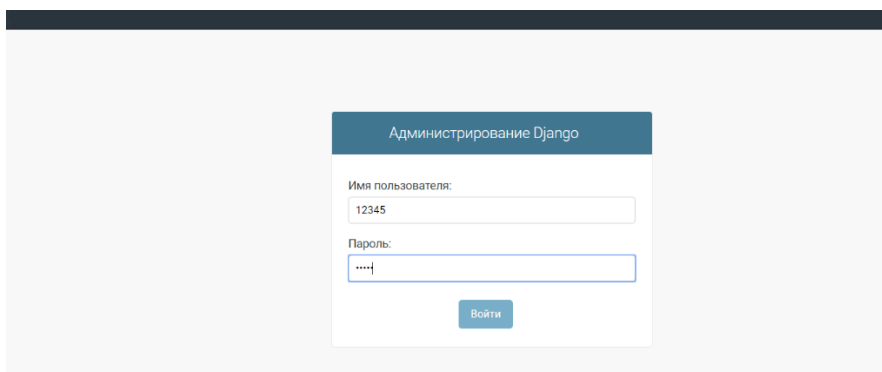


Рисунок 3.2 – Вхід до Адміністративної панелі Django

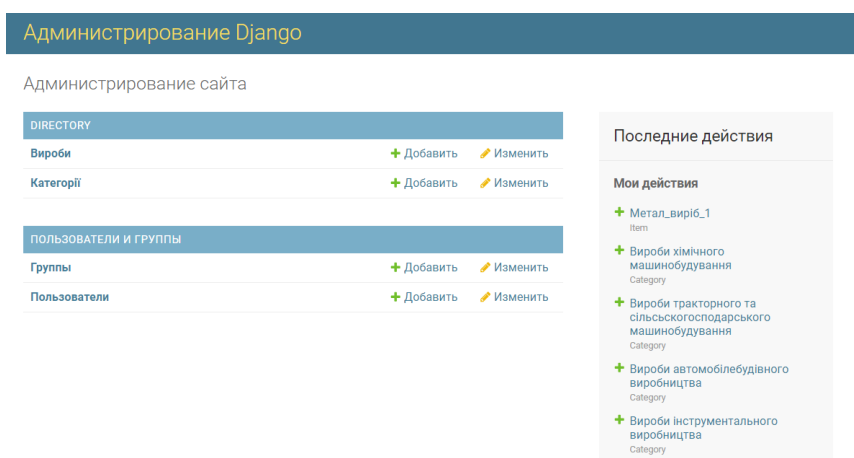


Рисунок 3.3 – Головна сторінка Адміністрування сайту

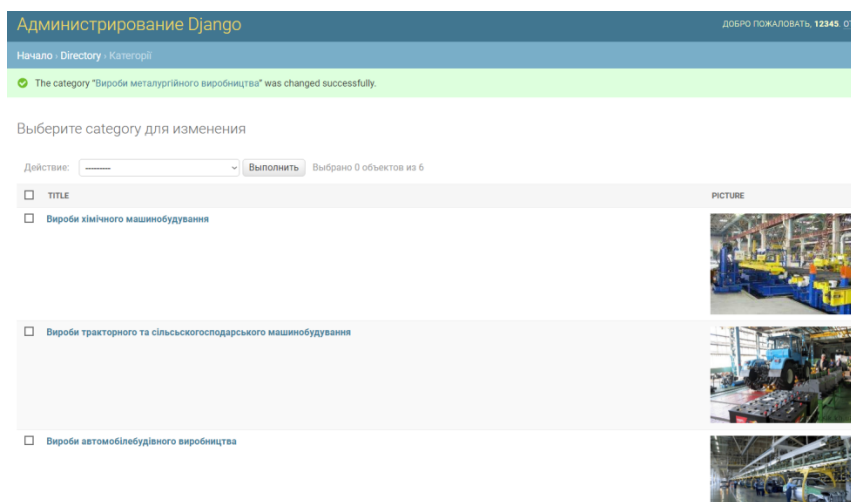


Рисунок 3.4 – Редагування категорій виробів усередині адмін-панелі

Администрирование Django

Начало · Directory · Вироби · Добавить item

Добавить item

Category:

Title:

Text:

Призначення та умови роботи:
 Пошарове зрізання (обпилювання) деталей із м'яких матеріалів з метою доведення виробу до заданої форми, розмірів, та шорсткості поверхні. Умови роботи: невисокі швидкості різання (до 15 м/хв.), невисока температура розігрівання та навантаження.
 Можливі причини виходу з ладу або причини виникнення дефектів при експлуатації деталі:
 Знос, сколювання і викривлення ріжучої кромки, налипання матеріалу оброблювального виробу, зневуглцювання, знеміцнення, жолоблення.
 Технічні вимоги до властивостей деталі:
 Висока твердість, що перевищує твердість оброблювального матеріалу; висока зносостійкість, висока механічна міцність.

Slug:

Зображення виробу: 150_yfv_0.jpg

[Сохранить и добавить другой объект](#)

Рисунок 3.5 – Додавання нового виробу

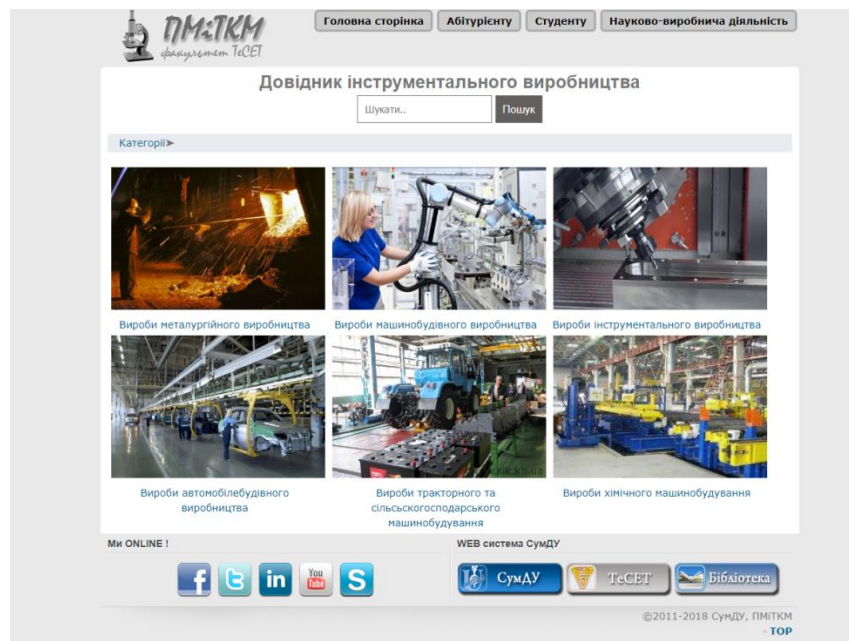


Рисунок 3.6 – Головна сторінка веб-додатку

При натисканні на одне із зображень категорій відбувається відображення відповідної сторінки.

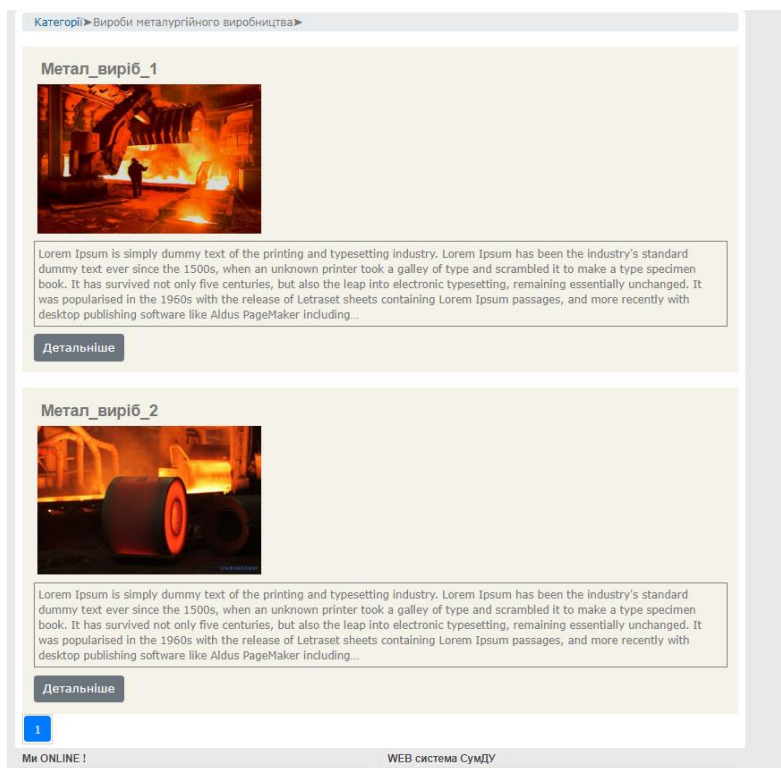


Рисунок 3.7 – Відображення виробів однієї категорії

Натиснувши на кнопку «Детальніше» у блоці виробу буде виконано переадресацію на сторінку конкретного виробу.



Рисунок 3.8 – Сторінка конкретного виробу

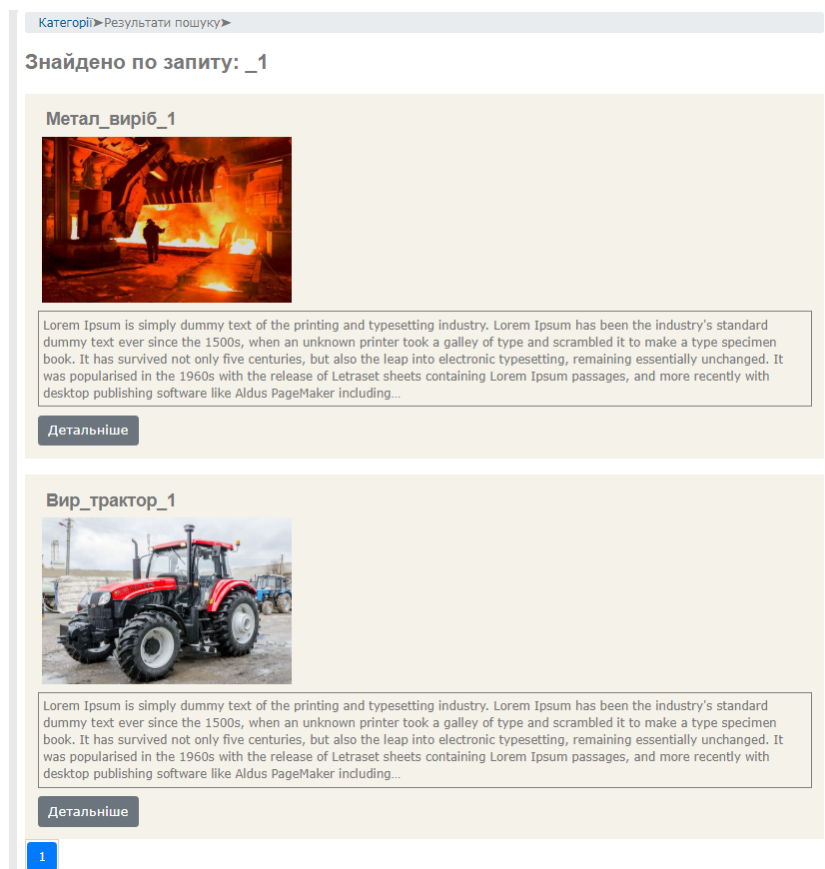


Рисунок 3.9 – Відображення результатів пошуку виробів

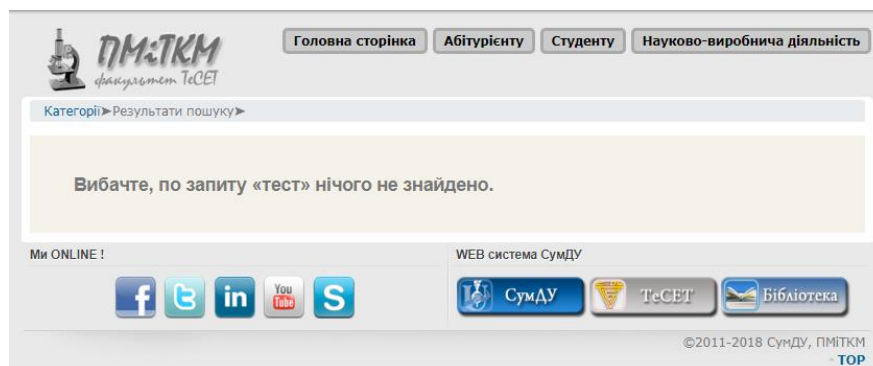


Рисунок 3.10 – Безрезультатний пошук

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розроблено програмний додаток-довідник інструментального виробництва.

У ході аналізу предметної області були розглянуті сучасні підходи створення веб-сайтів, організація контенту.

Після прийняття рішення по розробці програмного додатку було визначено мету та задачі проекту, сформульовано технічне завдання на виконання робіт.

Було виконано аналіз програмних продуктів-аналогів, мета яких – надати інформацію щодо шуканого користувачем об'єкта.

За допомогою моделі варіантів використання, контекстної діаграми та діаграми декомпозиції було створено опис процесів роботи програмного додатку. Розроблені в рамках виконання бакалаврської роботи календарний план і аналіз можливих ризиків для забезпечення досягнення цілей проекту.

Було продемонстровано архітектуру системи, опираючись на яку було розроблено програмний додаток. В якості бази даних під час розробки було використано MySQL, проте під час завантаження проекту в мережу сховище даних було замінено на більш інтегрований з хмарною PaaS-платформою PostgreSQL. Клієнтська частина була розроблена за допомогою веб-технологій, таких як HTML, CSS, JavaScript та спеціальної мови наслідування шаблонів, вбудованої у використаний Python-фреймворк. Логіка веб-додатку і серверна частина були створені з використанням популярного веб-фреймворку Django 3. Результатом виконання кваліфікаційної роботи став веб-сайт, який відповідає усім функціональним вимогам, що описані в технічному завданні. Зображено використання веб-додатку, що демонструє його працездатність.

Розроблений програмний продукт створений з метою забезпечення Кафедри прикладного матеріалознавства і технології конструкційних матеріалів Факультету технічних систем і енергоефективних технологій працездатним додатком-

довідником інструментального виробництва зі зручною функціональною адміністративною панеллю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Плюсы и минусы Django [Электронный ресурс] – Режим доступа до ресурсу: <https://habr.com/ru/post/473042/> (дата звернення: 23.04.2020).
2. Django – фреймворк для разработки на Python [Электронный ресурс] – Режим доступа до ресурсу: <https://web-creator.ru/articles/django/> (дата звернення: 23.04.2020).
3. Тренды веб разработки 2020 [Электронный ресурс] – Режим доступа до ресурсу: <https://merehead.com/ru/blog/web-development-trends-in-2020/> (дата звернення: 23.04.2020).
4. Методология IDEF0 [Электронный ресурс] – Режим доступа до ресурсу: <https://itteach.ru/bpwin/metodologiya-idef0/> (дата звернення: 18.03.2020).
5. IDEF0 как инструмент моделирования процессов [Электронный ресурс] – Режим доступа до ресурсу: <http://www.betec.ru/index.php?id=06&sid=73/>(дата звернення: 23.04.2020).
6. Лучшие фреймворки Python для веб-разработки, которые подойдут начинающим [Электронный ресурс] – Режим доступа до ресурсу: <https://tproger.ru/articles/best-python-frameworks-for-beginners/> (дата звернення: 23.04.2020).
7. Інформація: види, її місце й роль в інформаційному суспільстві [Электронный ресурс] – Режим доступа до ресурсу: <http://ru.osvita.ua/vnz/reports/journalism/25033/> (дата звернення: 24.04.2020).
8. Отображение заинтересованных сторон (SAM) [Электронный ресурс] – Режим доступа до ресурсу: <http://cleenet.org/index.php/ru/onlajn-kurs/modul-1/62-otobrazhenie-zainteresovannykh-storon/> (дата звернення: 23.04.2020).
9. Функции матрицы ответственности в проектном управлении [Электронный ресурс] – Режим доступа до ресурсу: <http://projectimo.ru/komanda-i-motivaciya/matrica-otvestvennosti.html/> (дата звернення: 24.04.2020).

10. Аграрні рішення [Електронний ресурс] – Режим доступу до ресурсу: <https://www.agro.basf.ua/uk/Services/Pest-Guide/> (дата звернення: 24.04.2020).
11. Аптека онлайн [Електронний ресурс] – Режим доступу до ресурсу: <http://www.apteka.pl.ua/> (дата звернення: 24.04.2020).
12. Освіта в Україні [Електронний ресурс] – Режим доступу до ресурсу: <https://osvita.ua/vnz/guide/search-17-0-66-0-0.html/> (дата звернення: 24.04.2020).
13. XAMPP — готовый комплект Web-сервера [Електронний ресурс] – Режим доступу до ресурсу: <http://www.uamedwed.com/blog/soft/xampp-gotovyj-komplekt-web-servera.html/> (дата звернення: 24.04.2020).
14. Диаграмма вариантов использования (Use Case diagram) [Електронний ресурс] – Режим доступу до ресурсу: https://flexberry.github.io/ru/fd_use-case-diagram.html/ (дата звернення: 23.04.2020).
15. Новиков Б. А. Основы технологий баз данных: навч. посіб. – Москва: ДМК Пресс, 2020.
16. Моргунов Е. П. PostgreSQL. Основы языка SQL: навч. посіб. – Санкт-Петербург: «БХВ-Петербург», 2018.
17. Гольцман В. MySQL 5.0 – Санкт-Петербург: Питер, 2010.
18. Хмарна платформа для створення Python-додатків [Електронний ресурс] – Режим доступу до ресурсу: <https://www.heroku.com/python/> (дата звернення: 24.04.2020).
19. С. Hillar Gaston, Romano Fabrizio Learn Web Development with Python // Livery place, Birmingham, UK – 2018.

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

1 Призначення й мета створення інформаційної системи

1.1 Призначення інформаційної системи

Інформаційна система повинна представляти онлайн-довідник Кафедри прикладного матеріалознавства і технології конструкційних матеріалів Факультету технічних систем і енергоефективних технологій Сумського державного університету.

1.2 Мета створення інформаційної системи

Метою створення ІС є забезпечення кафедри-замовника програмним додатком-довідником і доступом до його адміністрування.

1.3 Цільова аудиторія

У цільовій аудиторії інформаційної системи можна виділити наступні групи:

1. Абітурієнти;
2. Студенти;
3. Викладачі;
4. Співробітники;
5. Аспіранти;
6. Інші зацікавлені відвідувачі.

2 Вимоги до інформаційної системи

2.1 Вимоги до інформаційної системи в цілому

2.1.1 Вимоги до структури й функціонування інформаційної системи

Інформаційна система повинна бути інтегрована у сайт кафедри ПМіТКМ, доступного в мережі Інтернет під доменним іменем <https://pmitkm.teset.sumdu.edu.ua>.

2.1.2 Вимоги до персоналу

Для підтримки сайту й експлуатації веб-інтерфейсу системи керування сайтом (CMS) від персоналу не повинно вимагатися спеціальних технічних навичок, знання технологій або програмних продуктів, за винятком загальних навичок роботи з персональним комп'ютером і стандартним веб-браузером (наприклад, Opera, Google Chrome тощо).

2.1.3 Вимоги до збереженні інформації

Процедура резервного копіювання повинна проводитися співробітником, відповідальним за підтримку сайту, не рідше 1 разу на тиждень. Бекап виконується вручну, архівацією проекту.

2.1.4 Вимоги до розмежування доступу

Інформація, розташована на сайті, є загальнодоступною.

Користувачів сайту можна розділити на 2 групи відповідно до прав доступу:

1. Відвідувачі
2. Адміністратор

Відвідувачі мають доступ тільки до загальнодоступної частини сайту. Доступ до адміністративної частини мають користувачі із правами адміністратора.

Адміністратор може виконувати всі ті ж дії, що й Відвідувач, і крім того:

- додавати користувачів із правами Адміністратора;

- редагувати контент сайту.

Доступ до адміністративної частини повинен здійснюватися з використанням унікального логіна й пароля.

2.2 Вимоги до функцій, виконуваних сайтом

2.2.1 Основні вимоги

2.2.1.1 Структура сайту

Сайт повинен складатися з наступних розділів:

- Головна – виводиться меню категорій виробів.
- Інформація щодо конкретного виробничого виробу

2.2.1.2 Навігація

Користувацький інтерфейс сайту повинен забезпечувати наочне, інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації, швидкий і логічний перехід до розділів і сторінок. Навігаційні елементи повинні забезпечувати однозначне розуміння користувачем їх змісту: посилання на сторінки повинні бути мати заголовок, умовні позначки відповідати загальноприйнятим. Графічні елементи навігації повинні бути мати альтернативний підпис.

Повинна бути реалізована система навігаційних ланцюгів.

2.2.1.3 Наповнення сайту (контент)

Сторінки всіх розділів сайту повинні формуватися програмним шляхом на підставі інформації з бази даних на сервері.

Модифікація вмісту розділів повинна здійснюватися за допомогою адміністраторського веб-інтерфейсу (системи керування сайтом), який без застосування спеціальних навичок програмування (без використання програмування й спеціального кодування або форматування) повинен передбачати можливість редагування інформаційного вмісту сторінок сайту. Наповнення інформацією повинне проводитися з використанням шаблонів сторінок сайту.

2.2.2 Вимоги до функціональних можливостей

Система керування контентом (адміністративна частина сайту) повинна надавати можливість додавання, редагування й видалення вмісту динамічних сторінок.

2.2.2.1 Функціональні можливості розділів

На головній сторінці будуть представлені наступні елементи

- навігаційна стежка;
- можливість переключення мов;
- можливість виконання пошуку по базі інструментальних виробів;
- меню категорій виробничих виробів;
- вибір з випадаючого списку інформації про кафедру, науково-виробничу діяльність кафедри, корисної інформації для студентів та абітурієнтів;
- посилання на ресурси СумДУ;
- посилання на соціальні мережі кафедри;
- можливість виконання пошуку інформації на головному сайті кафедри.

2.2.2.2 Загальні вимоги

Стиль сайту можна описати як сучасний, діловий. У якості фонових кольорів рекомендується використовувати кольори головного сайту кафедри ПМіТКМ.

Оформлення не повинне заземляти інформативність: хоча на сайті не повинно бути досить багато графіки, він повинен бути зручний користувачам у плані навігації й цікавий для багаторазового відвідування.

2.2.2.3 Типові навігаційні й інформаційні елементи

- Шапка сайту
- Основне меню
- Основне поле контенту

– Підвал сайту

2.2.2.4 Шапка сайту

Шапка сайту повинна містити кнопки перемикання мови і можливість пошуку по головному сайту кафедри.

2.2.2.5 Основне меню

Основне меню повинне розташовуватися у верхній частині вікна (під шапкою) і містити посилання на всі розділи першого рівня (які представляють собою випадаючі списки з вкладеними в них підрозділами).

2.2.2.6 Основне поле контенту

Основне поле контенту повинне розташовуватися в центрі сторінки. У цьому полі відображається основний зміст обраного розділу. Стильове оформлення матеріалів і їх елементів (посилань, заголовків, основного тексту, зображень, форм, таблиць і т.п.) повинне бути єдиним для всього веб-сайту.

2.2.2.7 Підвал сайту

Підвал сайту повинен розташовуватися унизу сторінки. У підвалі знаходяться посилання на соц.мережі кафедри і посилання на ресурси університету.

2.3 Вимоги до видів забезпечення

2.3.1 Вимоги до інформаційного забезпечення

Реалізація сайту відбувається з використанням:

- Apache
- Django 3
- MySQL

2.3.2 Вимоги до лінгвістичного забезпечення

Сайт повинен бути виконаний українською мовою.

2.3.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище;
- Включена підтримка javascript, Flash і cookies.

2.3.4 Вимоги до апаратного забезпечення

Апаратне забезпечення серверної частини повинне задовольняти наступним вимогам:

- Веб-сервер Apache;
- Не менш 500 МБ вільного місця на диску.

3 Склад і зміст робіт зі створення сайту

Докладний опис етапів роботи зі створення сайту наведено в таблиці А.1.

Таблиця А.1 – Етапи створення сайту

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Розробка каркасу: Проектування розмітки та наповнення сайту	1 день
2	Авторизація: Розроблення та реалізація Авторизації на адміністративній панелі веб-сайту для адміністратора	1 день
3	Реалізація: Розробка та реалізація основного функціоналу онлайн-довідника: навігаційна стежка, механізм пошуку інструментальних виробів, інтерфейс відображення інформації про виріб і т.д.	12

Продовження табл. А.1.

4	<p>Завершення роботи: Проведення стилістичних виправлень веб-сайту, перевірка (тестування) реалізованого функціоналу</p>	2 дні
	<p>Загальна тривалість робіт (з урахуванням резервного строку на налагодження й виправлення помилок) і строк закінчення проекту</p>	16

4 Вимоги до складу й змісту робіт із введення сайту в експлуатацію

Для створення умов функціонування, при яких гарантується відповідність створюваного сайту вимогам сьогодення ТЗ і можливість його ефективної роботи, в організації Замовника повинен бути проведений певний комплекс заходів.

Для переносу сайту на хостинг необхідно, щоб параметри хостинга відповідали вимогам, зазначеним у технічному завданні. На хостинг переноситься програма (сайт), зверстаний шаблон дизайну й структура й наповнення бази даних з подальшою їх доробкою.

ДОДАТОК Б

ПЛАНУВАННЯ РОБІТ

1 Ідентифікація ідеї проекту

Кафедра прикладного матеріалознавства факультету технічних систем і енергоефективних технологій потребує працездатний веб-сайт із інтуїтивно простим та зрозумілим дизайном і ефективною у використанні адмін-панеллю.

2 Деталізація мети методом SMART

SMART - це аббревіатура, яка дає критерії, що керуються у встановленні цілей, наприклад, в управлінні проектами, управлінській діяльності та особистому розвитку. Букви S і M означають конкретність (Specific) та вимірюваність (Measurable). A – досяжність (Achievable), R – релевантність (Relevant), T – обмеженість у часі (Time-limited).

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити додаток-довідник інструментальних виробів.
Measurable (вимірювана)	Використовуючи якнайменше ресурсів розробити працездатний програмний продукт.
Achievable (досяжна)	Поставлена мета впливає у результаті актуальних проблем.
Relevant (реалістична)	У наявності є всі необхідні програмні засоби. Розробники мають допустиму кваліфікацію для виконання проекту.

Продовження таблиці Б.1

Time-framed (обмежена у часі)	Виконання завдання має часове обмеження. Строки досягнення мети проекту визначаються за домовленістю замовником та виконавцем.
-------------------------------------	--

3 Описання фази розробки ІТ– проекту

3.1 Планування змісту структури робіт ІТ– проекту (WBS)

WBS – це графічне подання згрупованих елементів проекту у вигляді пакета робіт, які ієрархічно пов’язані з продуктом проекту. На верхньому першому рівні WBS фіксується продукт проекту. Він повинен відповідати продукту проекту. Наступний II рівень відповідає діям або основним заходам для досягнення продукту проекту. Потім триває розбивка цих дій доти, поки не відбувається виконання дій елементарних робіт.

Діаграма WBS зображена на рис. Б.1.

3.2 Матриця аналізу зацікавлених сторін (SAM)

Аналіз зацікавлених сторін (його також називають картою зацікавлених сторін) є важливим кроком при розробці нової програми. Зацікавлені сторони включають осіб, лідерів громад, груп та інших організацій, на яких програма впливатиме або може вплинути на результат. Вони можуть бути внутрішніми або зовнішніми.

Цей шаблон може бути використаний для виявлення зацікавлених сторін проекту, включаючи рівень їх впливу, які важливі для них питання та як вони будуть їх вирішувати. На рисунку Б.2 представлена Stakeholder analysis matrix [8] проекту.

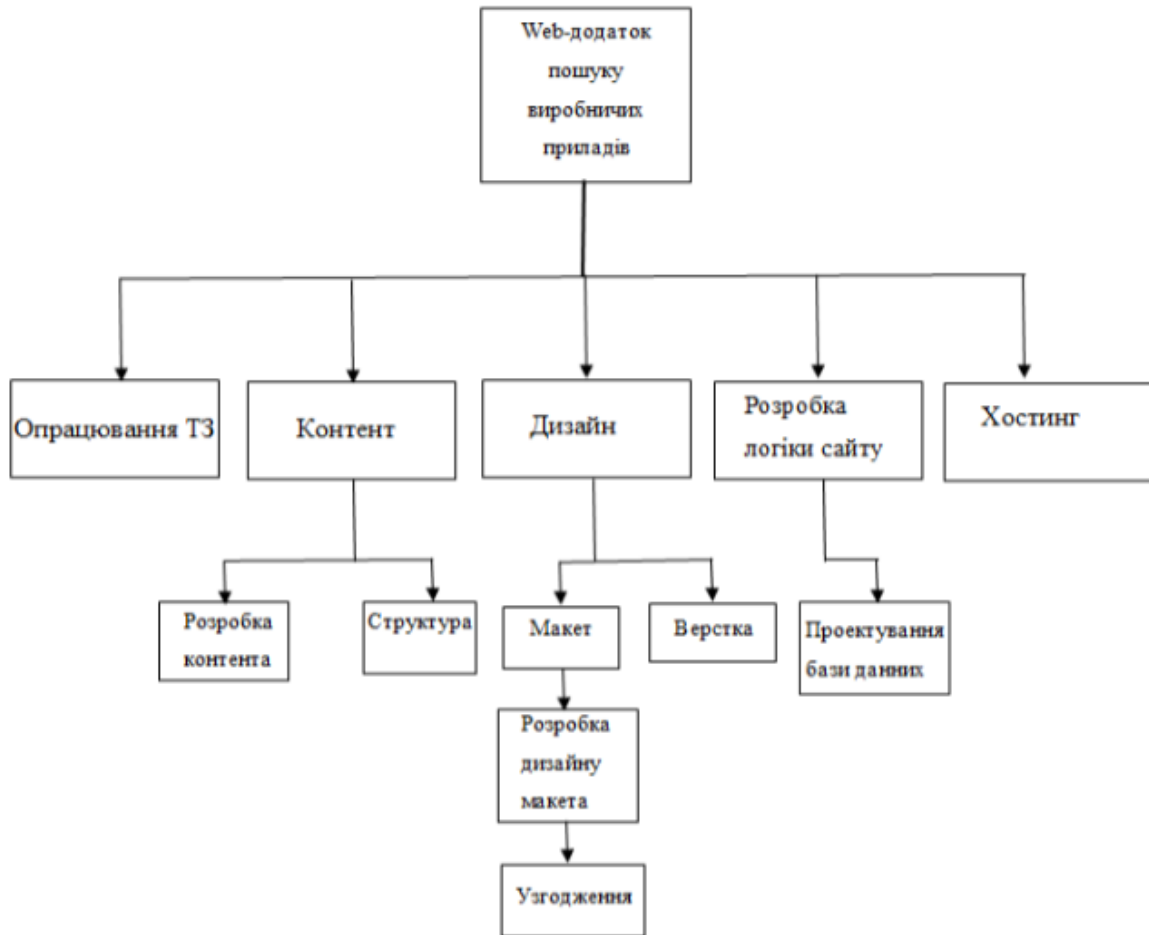


Рисунок Б.1 – WBS. Структура робіт Web-додатку пошуку приладів інструментального виробництва

Name	Contact info	Impact	Influence importance	What's important to this?	How could the project contribute?	How could the project block?	Role in the project
Вашченко Світлана Михайлівна	099 xxx xx xx vashchenko@gmail.com	Високий	Високий	Якість виконання, укладання в термін та документація.	Перевірка та внесення правок на будь-якому етапі проекту.	Може вносити пропозиції для змін елементів диплому.	Дипломний керівник
Рибалка Данііл Дмитрович	050 980 95 93 daniilrybalk@gmail.com	Високий	Високий	Отримання диплому бакалавра, визнання та похвала від дипломного керівника.	Внесення деяких правок щодо дизайну веб-додатку та безпосередньо розробка сайту.	Може не прислухатися до порад дипломного керівника та технічного завдання замовника.	Розробник веб-додатку.
Дядюра Костянтин Олександрович	099 xxx xx xx dvadyura@gmail.com	Середній	Високий	Отримання від розробника веб-додатку працездатну систему.	Внесення будь-яких правок щодо зовнішнього вигляду (дизайну) веб-додатку.	Відмова від замовлення веб-додатку.	Завідуючий кафедри-замовника.
Шендрік Віра Вікторівна	099 xxx xx xx shendrvk@gmail.com	Низький	Низький	Наявність диплома та своєчасність його написання.	Перевірка на якому етапі знаходиться написання диплому.	Може заборонити написання диплому через погану успішність.	Завідуюча кафедри ІТТІ

Рисунок Б.2 – SAM-структура проекту

3.3 Побудова матриці відповідальності (виконавців пакетів робіт)

На підставі SAM та WBS структур будують матрицю відповідальності проекту. Матриця відповідальності закріплює за кожною елементарною роботою виконавця. Матриця відповідальності [9] представлена на рисунку Б.3.

	Рибалка Д.Д.	Вашенко С.М.	Шендрик В.В.
Опрацювання ТЗ	R	A	I
Контент	R	I	I
Розробка контента	R	I	
Структура	R	A	I
Дизайн	R	I	
Макет	R	I	
Розробка дизайну макета	R	I	
Узгодження	R	A	
Верстка	R	I	
Розробка логіки сайту	R	I	
Проектування бази даних	R	I	
Хостинг	R	I	

Рисунок Б.3 – Матриця відповідальності

3.4 Управління ризиками

Ризик – це ймовірна подія, яка у випадку своєї появи негативно або позитивно впливає на проект.

Управління ризиком – це процес реагування на події та зміни ризиків у процесі виконання проекту. При цьому важливим є проведення моніторингу ризиків.

Моніторинг ризиків включає контроль ризиків протягом усього життєвого циклу проекту. Якісний моніторинг ризиків забезпечує управління інформацією, яка допомагає приймати ефективні рішення до настання ризикових подій.

Найбільш розповсюдженою характеристикою ризику є загроза або небезпека виникнення невдач у тій чи іншій діяльності, небезпека виникнення несприятливих

наслідків, змін зовнішнього середовища, які можуть викликати втрати ресурсів, збитки, а також небезпеку, від якої слід застрахуватися.

Процес управління ризиками включає в себе такі пункти.

- 1) Ідентифікація ризиків (виявлення ризиків).
- 2) Оцінювання ризиків (оцінка ймовірності та впливу).
- 3) Заходи реагування на ризики.
- 4) Моніторинг ризиків.

Ідентифікація ризиків – виявлення ризиків здатних вплинути на проект, документальне оформлення їх характеристик.

В даному випадку на першому етапі, в процесі виявлення ризиків можна виділити ряд ризиків.

Зовнішні непередбачувані ризики:

- природні катастрофи: повені; землетруси; шторми; кліматичні катаклізми тощо;
- злочини: вандалізм; саботаж; тероризм;
- неочікувані зовнішні ефекти: екологічні; соціальні;
- зриви: у створенні необхідної інфраструктури; через банкрутство підрядників; у фінансуванні: через помилки у визначенні цілей проекту; через неочікувані політичні зміни.

Зовнішні передбачувані (проте не визначені) ризики:

- ринковий ризик у зв'язку із: зміною вимог споживачів; економічними змінами; посиленням конкуренції; втратою позицій на ринку;
- операційні ризики: порушення безпеки; відступ від цілей проекту; неможливість підтримання робочого стану елементів проекту; неприпустимий екологічний вплив; негативні соціальні наслідки; зміни валютних курсів, нерозрахована інфляція.

Внутрішні ризики. Внутрішні організаційні ризики:

- зриви планів робіт через: недостачу робочої сили; нестачу часу; помилки проектування; помилки планування; недоліки координації робіт; зміни керівництва; конфлікти та саботаж; зміну можливостей замовника проекту; недостатнє управління.
- перевитрати коштів через: зриви планів робіт; невірну стратегію; некваліфікований персонал; переоплати за роботу/матеріали; неузгодження частин проекту; невірний кошторис.

Внутрішні технічні ризики:

- зміни технології;
- специфічні ризики технології, що закладаються до проекту;
- помилки в проектно-кошторисній документації.

Інші ризики:

- прямі втрати майна: по транспортних спорах; обладнання і т.д.;
- непрямі збитки: перестановка обладнання; неотримання орендного прибутку; порушення запланованого ритму діяльності; збільшення необхідного фінансування;
- ризики, що підлягають обов'язковому страхуванню: нещасні випадки на виробництві; пошкодження майна; викрадення транспортних засобів.

За джерелами виникнення:

- неправильна оцінка ринкової ситуації (ємність ринку, рівень конкуренції);
- невизначеність мети, інтересів і поведінки учасників проекту, проблеми управління командою;
- виробничо-технологічне устаткування, виробничі ризики (аварії, брак);
- неточність проектної документації (витрати, строки реалізації проекту, технічні і технологічні параметри);
- ризик зміни пріоритетів розвитку підприємства і підтримки з боку керівництва;

- неповнота або неточність інформації про фінансовий стан та ділову репутацію учасників.

Систематичний ризик – визначається зовнішніми обставинами, не залежить від суб'єкта та не регулюється ним:

- ризики, пов'язані з нестабільністю нормативного поля режиму інвестування;
- можливість змін природно-кліматичних умов, стихійного лиха;
- можливість погіршення політичної ситуації, несприятливі соціально-політичні зміни;
- коливання ринкової кон'юнктури, валютного курсу.

За сферами прояву:

- економічні, пов'язані із зміною економічних факторів;
- політичні, пов'язані із зміною політичного курсу країни;
- соціальні, пов'язані з соціальними складнощами;
- екологічні, пов'язані з екологічними катастрофами;

За видами втрат (наслідками):

- трудові втрати – втрати фондів робочого часу;
- перевитрати часу – уповільнення процесу реалізації проекту у порівнянні з планом;
- фінансові втрати – пов'язані з непередбаченими виплатами (штрафи, уплата додаткових податків), неотриманням коштів з передбачених джерел;
- перевищення витрат – виникає внаслідок зміни початкового плану реалізації проекту або заниження розрахункових витрат на будівництво, затримання строків будівництва;
- соціальні втрати – шкода здоров'ю та життю людей, довіллю, престижу організації, іміджу учасників;
- ризик нежиттєздатності проекту – передбачені доходи від проекту можуть бути недостатніми для покриття усіх видів витрат;

- ризик недоплати заборгованостей – тимчасове зниження доходу через короткострокове падіння попиту на продукт, або зниження ціни на нього;
- податковий ризик – неможливість використання податкових пільг, встановлених законодавством; зміна податкового законодавства;
- виробничі ризики – виникають після завершення впровадження проекту; пов’язані з проблемами технічної підготовки і розробки проекту, поганого інжинірингу, незадовільного навчання персоналу, нестачі сировини і зростання витрат виробництва.

В процесі аналізу для визначення числових значень ймовірності виникнення ступеня впливу, зазвичай застосовується метод експертних оцінок. На їх основі визначається ранг ризику, як потенційний вплив ризику на проект, який оцінюється як добуток ймовірності виникнення та ступеню впливу.

Шкала оцінки ризику може відповідати емпіричній шкалі оцінки ризику:

5 балів - критичний ризик (0,81 - 1);

4 бали - максимальний ризик (0,61 - 0,8);

3 бали - високий ризик (0,41 - 0,6);

2 бали - нормальний ризик (0,31 - 0,4);

1 бал - малий ризик (0 - 0,3).

Обчислення рангу ризику виконується за формулою (Б.1):

$$R = P * L, \quad (\text{Б.1})$$

де R – ранг ризику;

P – ймовірність виникнення ризику;

L – ступінь впливу.

В таблиці Б.2 представлені ризики проекту.

Таблиця Б.2 – Ризики проекту

№	Об'єкт ризику	Ризик	P	L	R
1.	Час	Недотримання виконавцем та замовником календарних строків проекту	0.9	0.5	0.45
2.	Час	Зміна пріоритету проекту замовником	0.5	0.3	0.15
3.	Технологія	Недостовірні інформація про характеристики базового програмно-апаратного комплексу замовника або його значуща зміна в ході реалізації проекту	0.4	0.4	0.16
4.	Якість	Невідповідність системи задачам, грубі помилки в алгоритмах процесів, критичні збої системи	0.1	0.4	0.04
5.	Бюджет	Виникнення незапланованих робіт по проекту	0.7	0.2	0.14
6.	Трудові ресурси та їх кваліфікація	Неможливість участі в запланованих роботах по проекту необхідних співробітників зі сторони замовника та виконавця у зв'язку з відпусткою, відрядженням та ін.	0.1	0.1	0.01

Продовження таблиці Б.2

7.	Інтеграція	Недостовірна інформація про зовнішні системи, з яких передбачена взаємодія в рамках проекту	0.5	0.4	0.2
8.	Ринок	Розширення функціональних характеристик програмних продуктів, що вже використовуються замовником в рамках цілей проекту	0.1	0.05	0.005
9.	Ринок	Зміна вимог замовника, економічні зміни, посилення конкуренції, втрата позицій на ринку;	0.5	0.1	0.05
10.	Бюджет	Зриви планів робіт; невірна стратегію; некваліфікований персонал; переплати за роботу/матеріали; неузгодження частин проекту; невірний кошторис.	0.5	0.2	0.1

ДОДАТОК В

ЛІСТИНГ ПРОГРАМНОГО КОДУ

base.html

```
{% load static %}
<!DOCTYPE html>
<html lang="ua">
<head>
  <link href="https://pmitkm.teset.sumdu.edu.ua/favicon.ico" rel="shortcut icon" type="image/x-icon">
  <script src="https://pmitkm.teset.sumdu.edu.ua/media/system/js/mootools.js"></script>
  <script src="https://pmitkm.teset.sumdu.edu.ua/media/system/js/caption.js"></script>
  <script src="https://pmitkm.teset.sumdu.edu.ua/plugins/content/phocagalleryslideshow.js"></script>
  <script>
    var gaJsHost = (("https:" == document.location.protocol) ? "https://ssl." : "http://www.");
document.write(unescape("%3Cscript src='" + gaJsHost + "google-analytics.com/ga.js'
type='text/javascript'%3E%3C/script%3E"));
  </script>
  <script src="https://ssl.google-analytics.com/ga.js"></script>
  <script>
    DD_roundies.addRule('div.ry-search .inputbox', '3px', true);
    DD_roundies.addRule('.ry-mainnav ul li.active a', '6px', true);
    DD_roundies.addRule('.ry-mainnav ul li a:hover, .ry-mainnav ul li a:focus, .ry-mainnav ul li.sfhover a', '6px', true);
    DD_roundies.addRule('.ry-mainnav ul li ul, .ry-mainnav ul li ul li, .ry-mainnav ul li ul li a', '3px', true);
  </script>
  <script type="text/javascript">
    try { var pageTracker = _gat._getTracker("UA-33624505-1"); pageTracker._setCustomVar( 1, "user_ip",
"192.168.18.254", 1 ); pageTracker._trackPageview();
    } catch(err) {}
  </script>
  <link rel="stylesheet" href="https://pmitkm.teset.sumdu.edu.ua/templates/ry_river/css/default.css" type="text/css">
  <link rel="stylesheet" href="https://pmitkm.teset.sumdu.edu.ua/templates/ry_river/css/template.css" type="text/css">
  <link rel="stylesheet" href="https://pmitkm.teset.sumdu.edu.ua/templates/ry_river/css/menu.css" type="text/css">
  <link rel="stylesheet" href="https://pmitkm.teset.sumdu.edu.ua/templates/ry_river/css/layout.css" type="text/css">
  <script src="https://pmitkm.teset.sumdu.edu.ua/templates/ry_river/libs/js/moomenu.js"></script>
  <script src="https://pmitkm.teset.sumdu.edu.ua/templates/ry_river/libs/js/ry.script.js"></script>
  <script src="https://pmitkm.teset.sumdu.edu.ua/templates/ry_river/libs/js/ry.scroll.js"></script>
  <script src="https://pmitkm.teset.sumdu.edu.ua/templates/ry_river/libs/js/DD_roundies_0.0.2a-min.js"></script>
  <style>.breadcrumb{ display:-ms-flexbox;display:flex;-ms-flex-wrap:wrap;flex-wrap:wrap;padding:.15rem 1rem;margin-
```

```
bottom: 1rem;list-style:none;background-color:#e9ecef;border-radius:.4rem}</style>
```

```
<style>.breadcrumb-item+.breadcrumb-item{padding-left:.5rem}.breadcrumb-item+.breadcrumb-
item::before{display:inline-block;padding-right:.5rem;color:#6c757d;content:"/"}.breadcrumb-item+.breadcrumb-
item:hover::before{text-decoration:underline}.breadcrumb-item+.breadcrumb-item:hover::before{text-
decoration:none}.breadcrumb-item.active{color:#6c757d}.pagination{display:-ms-flexbox;display:flex;padding-left:0;list-
style:none;border-radius:.25rem}.page-link{position:relative;display:block;padding:.5rem .75rem;margin-left:-1px;line-
height:1.25;color:#007bff;background-color:#fff;border:1px solid #dee2e6}.page-link:hover{z-index:2;color:#0056b3;text-
decoration:none;background-color:#e9ecef;border-color:#dee2e6}.page-link:focus{z-index:3;outline:0;box-shadow:0 0 0
.2rem rgba(0,123,255,.25)}.page-item:first-child .page-link{margin-left:0;border-top-left-radius:.25rem;border-bottom-left-
radius:.25rem}.page-item:last-child .page-link{border-top-right-radius:.25rem;border-bottom-right-radius:.25rem}</style>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport"
```

```
content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
```

```
<meta http-equiv="X-UA-Compatible" content="ie=edge">
```

```
<title>{% block title %} Довідник інструментального виробництва {% endblock title %}</title>
```

```
<style>
```

```
.ry-content li {background-image: none;}
```

```
</style>
```

```
</head>
```

```
<body id="bd" class="fs4">
```

```
<div id="ry-septop" class="clearfix">
```

```
<div class="main">
```

```
<div class="date">
```

```
<p>
```

```
</p>
```

```
</div>
```

```
<div class="ry-search">
```

```
<form action="index.php" method="post">
```

```
<div class="search">
```

```
<input name="searchword" id="mod_search_searchword" maxlength="20" alt="Пошук" class="inputbox"
```

```
type="text" size="20" value="Пошук..." onblur="if(this.value=="") this.value='Пошук...';"
```

```
onfocus="if(this.value=='Пошук...') this.value=";">
```

```
</div>
```

```
<input type="hidden" name="task" value="search">
```

```
<input type="hidden" name="option" value="com_search">
```

```
<input type="hidden" name="Itemid" value="50">
```

```
</form>
```

```
</div>
```

```
</div>
```

```

</div>
<div id="ry-head" class="clearfix">
  <div class="main">
    <div id="header-logo">
      <div class="logo">
        <a href="https://pmitkm.teset.sumdu.edu.ua/" title="Кафедра прикладного матеріалознавства і технології
конструкційних матеріалів"><span>Кафедра прикладного матеріалознавства і технології конструкційних
матеріалів</span></a>
      </div>
    </div>
  <div class="ry-mainnav">
    <ul class="menu"><li class="parent item1">
      <a href="https://pmitkm.teset.sumdu.edu.ua/">
        <span>Головна сторінка</span>
      </a>
      <ul style="height: 0px; left: -999em;">
        <li class="item6"><a href="https://pmitkm.teset.sumdu.edu.ua/home/news.html">
          <span>Новини кафедри</span></a></li>
        <li class="item8"><a href="https://pmitkm.teset.sumdu.edu.ua/home/presentaciya.html">
          <span>Презентація спеціальності</span></a></li>
        <li class="item48"><a href="https://pmitkm.teset.sumdu.edu.ua/home/2015-12-23-09-36-35.html">
          <span>Викладачі кафедри ПМіТКМ</span></a></li>
        <li class="item7"><a href="https://pmitkm.teset.sumdu.edu.ua/home/istoriya.html">
          <span>Історія кафедри</span></a></li>
        <li class="item45"><a href="https://pmitkm.teset.sumdu.edu.ua/home/dosyagnennyakafedri.html">
          <span>Наші досягнення</span></a></li>
        <li class="item20"><a href="https://pmitkm.teset.sumdu.edu.ua/home/mapsgoogle.html">
          <span>Розташування кафедри</span></a></li>
        <li class="item22"><a href="https://pmitkm.teset.sumdu.edu.ua/home/kontakt.html">
          <span>Контактна інформація</span></a></li></ul></li>
      <li class="parent "><a href="https://pmitkm.teset.sumdu.edu.ua/abiturientu.html">
        <span>Абітурієнту</span></a>
        <ul style="height: 0px; left: -999em;">
          <li class="item31"><a href="https://pmitkm.teset.sumdu.edu.ua/abiturientu/present.html">
            <span>Презентація спеціальності</span></a></li>
          <li class="item30"><a href="https://pmitkm.teset.sumdu.edu.ua/abiturientu/rabota.html">
            <span>Перспективи працевлаштування</span></a></li>
          <li class="item49"><a href="https://pmitkm.teset.sumdu.edu.ua/abiturientu/2015-12-23-09-36-
35.html"><span>Викладачі кафедри ПМіТКМ</span></a></li>
          <li class="item41"><a

```


href="https://pmitkm.teset.sumdu.edu.ua/abiturientu/kontakt.html">Контактна інформація
 <li id="current" class="active item50">Випускники кафедри
 <li class="item51">Активне дозвілля
 <li class="item52">Практика
 <li class="item53">
 Правила прийому
 <li class="item54">
 Освітня програма (бак.)
 <li class="item55">Освітня програма
 (маг.)
 <li class="item56">Вартість навчання
 <li class="item57">Приймальна
 комісія
 <li class="parent item3">Студенту<ul style="height: 0px; left: -
 999em;">
 <li class="item26">Дошка оголошень
 <li class="item34">Навчальний процес
 <li class="item29">Опитування
 <li class="item32">Фотоісторія
 <li class="parent item4">Науково-
 виробнича діяльність<ul style="height: 0px; left: -999em;">
 <li class="item47"><a href="https://pmitkm.teset.sumdu.edu.ua/nauka/2014-02-24-11-21-
 17.html">Підвищення кваліфікації
 <li class="item24">Наукові
 напрямки
 <li class="item23">Лабораторії кафедри
 <li class="item46">Ми
 пропонуємо<li class="item33">
 Зворотній

```
зв'язок</span></a></li></ul></li></ul>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div id="ry-container" class="clearfix">
```

```
<div class="main">
```

```
<div class="content-wrap clearfix">
```

```
<div class="content-bd" style="width:100%;">
```

```
<div class="ry-content" style="padding-right:0px;">
```

```
<nav class="article_content">
```

```
{ % block content % }
```

```
{ % endblock % }
```

```
</nav>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div id="ry-bots" class="clearfix">
```

```
<div class="main">
```

```
<div class="bot-coloum bc-left" style="width:50%;">
```

```
<div class="ry-moduletable moduletable clearfix">
```

```
<h3><span>Ми ONLINE !</span></h3>
```

```
<div class="ry-box-ct clearfix">
```

```
<p style="text-align: center;">
```

```
<a href="http://www.facebook.com/groups/190055787721249" target="_blank"></a> <a
```

```
href="https://twitter.com/PMiTKM" target="_blank"></a> <a
```

```
href="http://www.linkedin.com/pub/pmitkm-teset/45/a38/391" target="_blank"></a> <a
href="http://www.youtube.com/user/pmitkm" target="_blank"></a> <a
href="skype:pmitkm.teset.sumdu?add"></a></p> </div>
</div>
</div>
<div class="bot-coloum bc-right" style="width:50%;">
<div class="ry-moduletable moduletable clearfix">
<h3><span>WEB система СумДУ</span></h3>
<div class="ry-box-ct clearfix">
<p><a href="http://sumdu.edu.ua" target="_blank"></a><a href="http://teset.sumdu.edu.ua" target="_blank"></a><a href="http://library.sumdu.edu.ua" target="_blank"></a></p> </div>
</div>
</div>
</div>
</div>
<div id="ry-footer" class="clearfix">
<div class="main clearfix">
<div class="footer">

```

```

</div>
<div class="fot-menu ">
    <!-- please do not remove the credit link from RayaWebDesign.
    if you want to remove the credit link, please confirm to the RayaWebDesign and do a minimum donation $2-->
    <div style="visibility:hidden; font-size:1px">Design&nbsp;by&nbsp;<a href="http://rayawebdesign.com" title="Jasa
Pembuatan Website Murah" target="_blank">RayaWebDesign</a>.
    Valid&nbsp;<a href="http://validator.w3.org/check?uri=referer" title="Valid XHTML"
target="_blank">XHTML</a>&nbsp;&nbsp;&nbsp;<a href="http://jigsaw.w3.org/css-validator/check/referer?profile=css3"
title="Valid CSS" target="_blank">CSS</a>.</div>
    <div>©2011-2018 СумДУ, ПМіТКМ
    </div> &nbsp;<a href="#" id="gotop" title="Press and scroll to top">TOP</a>
</div>
</div>
</div>
</body>
</html>

```

results.html

```

{% extends 'home.html' %}
{% block title %} Результати пошуку {% endblock title %}

{% block content %}
    {% block breadcrumbs %}
    <div class="breadcrumb">
        <a href="/">Категорії</a>
        <p>Результати пошуку</p>
    </div>
    {% endblock breadcrumbs %}
    {% if items %}
        <h1>Знайдено по запиту: {{query}}</h1>
    {% endif %}
    {% if items %}
        {% for item in items %}
            <div style="">
                <div style="background-color:#f5f2e9;margin-top:20px;">
                    <div style="padding:15px;">
                        <h2 class="mb-0" style="margin-left:10px;">{{ item.title }}</h2>
                        
                    </div>
                </div>
            </div>
        {% endfor %}
    {% endif %}

```

```

        <p style="border: 1.5px solid gray;padding: 5px; ">{{ item.text|truncatechars:550 }}</p>
        <div><a href="/{{ item.href }}/{{ item.slug }}" role="button" class="btn btn-secondary
otstup">Детальніше</a></div>
    </div>
</div>
</div>
</div>
{% endfor %}
{% else %}
<div style="">
    <div style="background-color:#f5f2e9;margin-top:10px;">
        <div style="padding:40px;">
            <h2 class="mb-0" style="margin-left:10px;">Вибачте, по запиту &laquo;{{ query }}&raquo; нічого не
знайдено.</h2>
        </div>
    </div>
</div>
{% endif %}
{% if items %}
<ul class="pagination">
    {% for i in paginator.page_range %}
        <li class="page-item {% if i == page_number %} active {% endif %}">
            <a class="page-link" href="?q={{ query }}&page={{ i }}">{{ i }}</a>
        </li>
    {% endfor %}
</ul>
{% endif %}
{% endblock content %}

```

carbuild_post.html

```

{% extends 'carbuild.html' %}
{% block title %} {{ item.title }} {% endblock title %}
{% block content %}
<style>
    .breadcrumb{display:-ms-flexbox;display:flex;-ms-flex-wrap:wrap;flex-wrap:wrap;padding:.15rem 1rem;margin-
bottom:1rem;list-style:none;background-color:#e9ecef;border-radius:.25rem}
    .breadcrumb p::before{
        content: "►";
    }

```

```

</style>
{% block breadcrumbs %}
  <div class="breadcrumb">
    <a href="/">Категорії</a>
    <p></p>
    <a href="/carbuild">Вироби автомобілебудівного виробництва</a>
    <p>{{ item.title }}</p>
  </div>
{% endblock breadcrumbs %}
<h2 style="margin-left:10px;">{{ item.title }}</h2>

<p style="border: 1.5px solid gray;padding: 5px; ">{{ item.text }}</p>
{% endblock content %}

```

carbuild.html

```

{% extends 'home.html' %}
{% block title %} Вироби автомобілебудівного виробництва {% endblock title %}

{% block content %}
  {% block breadcrumbs %}
  <div class="breadcrumb">
    <a href="/">Категорії</a>
    <p>Вироби автомобілебудівного виробництва</p>
  </div>
  {% endblock breadcrumbs %}
  {% for item in items %}
    <div style="">
      <div style="background-color:#f5f2e9;margin-top:20px;">
        <div style="padding:15px;">
          <h2 class="mb-0" style="margin-left:10px;">{{ item.title }}</h2>
          
          <p style="border: 1.5px solid gray;padding: 5px; ">{{ item.text|truncatechars:550 }}</p>
          <div><a href="/carbuild/{{ item.slug }}" role="button" class="btn btn-secondary otstup">Детальніше</a></div>
        </div>
      </div>
    </div>
  {% endfor %}

```

```

{% if items %}
  <ul class="pagination">
    {% for i in paginator.page_range %}
      <li class="page-item {% if i == page_number %} active {% endif %}">
        <a class="page-link" href="?page={{ i }}">{{ i }}</a>
      </li>
    {% endfor %}
  </ul>
{% endif %}
{% endblock content %}

```

urls.py

```

from django.urls import path
from .views import *

app_name = 'directory'
urlpatterns = [
    path("", directory, name='home'),
    path('s/', search, name='пошук'),
    path('metalproduction/', metalproduction, name='Вироби металургійного виробництва'),
    path('machinebuild/', machinebuild, name='Вироби машинобудівного виробництва'),
    path('instrproduction/', instrproduction, name='Вироби інструментального виробництва'),
    path('carbuild/', carbuild, name='Вироби автомобілебудівного виробництва'),
    path('traktorbuild/', traktorbuild, name='Вироби тракторного та сільськогосподарського машинобудування'),
    path('chembuild/', chembuild, name='Вироби хімічного машинобудування'),
    path('metalproduction/<str:slug>', metalproductionPost, name='metalproductionPost'),
    path('machinebuild/<str:slug>', machinebuildPost, name='machinebuildPost'),
    path('instrproduction/<str:slug>', instrproductionPost, name='instrproductionPost'),
    path('carbuild/<str:slug>', carbuildPost, name='carbuildPost'),
    path('traktorbuild/<str:slug>', traktorbuildPost, name='traktorbuildPost'),
    path('chembuild/<str:slug>', chembuildPost, name='chembuildPost'),
]

```

home.html

```

{% extends 'base.html' % }
{% block content % }
<style>
  .pic p{text-align: center;}
  .pic{ max-width: 300px;float:left;margin-right: 10px;}
  .breadcrumb{display:-ms-flexbox;
    display:flex;-ms-flex-wrap:wrap;flex-wrap:wrap;
    padding:.15rem 1rem;margin-bottom: 1rem;list-style:none;
    background-color:#e9ecef;border-radius:.25rem}
  .breadcrumb::after{
    content: "►";
  }
  form.example input[type=text] {
    padding: 10px;
    font-size: 14px;
    border: 1px solid grey;
  }
  form.example button {
    padding: 10px;
    background: #61605f;
    color: white;
    font-size: 15px;
    border: 1px solid grey;
    border-left: none;
    cursor: pointer;
    margin-bottom: 15px;
  }
</style>
<p><center><h1>Довідник інструментального виробництва</h1></center></p>
<center>
  <form class="example" action="/s/">
    <input type="text" placeholder="Шукати.." name="q" role="search">
    <button type="submit">Пошук</button>
  </form>
</center>
{% block breadcrumbs % }
  <div class="breadcrumb"><a href="#" ">Категорії</a></div>
{% endblock breadcrumbs % }
<a href="metalproduction/"><div class="pic"><p>{{ data1.title

```



```

}}</p></div></a>
<a href="machinebuild/"><div class="pic"><p>{{ data2.title
}}</p></div></a>
<a href="instrproduction/"><div class="pic"><p>{{ data3.title
}}</p></div></a>
<a href="carbuild/"><div class="pic"><p>{{ data4.title
}}</p></div></a>
<a href="traktorbuild/"><div class="pic"><p>{{ data5.title
}}</p></div></a>
<a href="chembuild/"><div class="pic"><p>{{ data6.title
}}</p></div></a>

{% endblock content %}

```

models.py

```

from django.db import models
class Category(models.Model):
    title = models.CharField(max_length=255)
    img = models.ImageField(null=True, blank=True, upload_to='images/category/', verbose_name='Зображення категорії')
    class Meta:
        verbose_name_plural = "категорії"

    def __str__(self):
        return self.title

class Item(models.Model):
    category = models.ForeignKey(Category, on_delete=models.CASCADE)
    title = models.CharField(max_length=255)
    text = models.TextField()
    slug = models.CharField(max_length=100)
    img = models.ImageField(null=True, blank=True, upload_to='images/items', verbose_name='Зображення виробу')

    @property
    def href(self):
        """Returns the item's href name."""
        if self.category.title == "Вироби металургійного виробництва":
            return "metalproduction"
        elif self.category.title == "Вироби машинобудівного виробництва":

```

```

        return "machinebuild"
elif self.category.title == 'Вироби інструментального виробництва':
    return 'instrproduction'
elif self.category.title == 'Вироби автомобілебудівного виробництва':
    return 'carbuild'
elif self.category.title == 'Вироби тракторного та сільськогосподарського машинобудування':
    return 'traktorbuild'
elif self.category.title == 'Вироби хімічного машинобудування':
    return 'chembuild'

class Meta:
    verbose_name_plural = "Вироби"

def __str__(self):
    return self.title

```

views.py

```

from django.shortcuts import render
from .models import *
from django.core.paginator import *
from django.db.models import Q
from django.shortcuts import redirect

def directory(request):

    obj1 = Category.objects.get(id=1)
    obj2 = Category.objects.get(id=2)
    obj3 = Category.objects.get(id=3)
    obj4 = Category.objects.get(id=4)
    obj5 = Category.objects.get(id=5)
    obj6 = Category.objects.get(id=6)
    context = {
        "data1": obj1,
        "data2": obj2,
        "data3": obj3,
        "data4": obj4,
        "data5": obj5,
        "data6": obj6,

```

```

}
return render(request, 'home.html', context)

```

```
def search(request):
```

```

    try:
        q = request.GET.get('q')
    except:
        q = None
    if q:
        items = Item.objects.filter(Q(title__icontains=q) | Q(text__icontains=q))
        paginator = Paginator(items, 4)
        page_number = request.GET.get('page', 1)
        page_obj = paginator.get_page(page_number)
        context = {'query': q, 'items': page_obj.object_list, 'paginator': paginator, 'page_number': int(page_number), }
    else:
        return redirect('directory:home')
    return render(request, 'results.html', context)

```

```
def metalproduction(request):
```

```

    items = Item.objects.filter(category_id=1)
    paginator = Paginator(items, 4)
    page_number = request.GET.get('page', 1)
    page_obj = paginator.get_page(page_number)
    context = {
        "items": page_obj.object_list,
        "paginator": paginator,
        "page_number": int(page_number),
    }
    return render(request, 'metalproduction.html', context)

```

```
def machinebuild(request):
```

```

    items = Item.objects.filter(category_id=2)
    paginator = Paginator(items, 4)
    page_number = request.GET.get('page', 1)
    page_obj = paginator.get_page(page_number)
    context = {
        "items": page_obj.object_list,
        "paginator": paginator,
    }

```

```
    "page_number": int(page_number),
}
return render(request, 'machinebuild.html', context)
```

```
def instrproduction(request):
    items = Item.objects.filter(category_id=3)
    paginator = Paginator(items, 4)
    page_number = request.GET.get('page', 1)
    page_obj = paginator.get_page(page_number)
    context = {
        "items": page_obj.object_list,
        "paginator": paginator,
        "page_number": int(page_number),
    }
    return render(request, 'instrproduction.html', context)
```

```
def carbuild(request):
    items = Item.objects.filter(category_id=4)
    paginator = Paginator(items, 4)
    page_number = request.GET.get('page', 1)
    page_obj = paginator.get_page(page_number)
    context = {
        "items": page_obj.object_list,
        "paginator": paginator,
        "page_number": int(page_number),
    }
    return render(request, 'carbuild.html', context)
```

```
def traktorbuild(request):
    items = Item.objects.filter(category_id=5)
    paginator = Paginator(items, 4)
    page_number = request.GET.get('page', 1)
    page_obj = paginator.get_page(page_number)
    context = {
        "items": page_obj.object_list,
        "paginator": paginator,
        "page_number": int(page_number),
```

```
}  
return render(request, 'traktorbuild.html', context)
```

```
def chembuild(request):  
    items = Item.objects.filter(category_id=6)  
    paginator = Paginator(items, 4)  
    page_number = request.GET.get('page', 1)  
    page_obj = paginator.get_page(page_number)  
    context = {  
        "items": page_obj.object_list,  
        "paginator": paginator,  
        "page_number": int(page_number),  
    }  
    return render(request, 'chembuild.html', context)
```

```
def metalproductionPost(request, slug):  
    item = Item.objects.filter(slug=slug).first()  
    context = {'item': item}  
    return render(request, 'metalproduction_post.html', context)
```

```
def machinebuildPost(request, slug):  
    item = Item.objects.filter(slug=slug).first()  
    context = {'item': item}  
    return render(request, 'machinebuild_post.html', context)
```

```
def instrproductionPost(request, slug):  
    item = Item.objects.filter(slug=slug).first()  
    context = {'item': item}  
    return render(request, 'instrproduction_post.html', context)
```

```
def carbuildPost(request, slug):  
    item = Item.objects.filter(slug=slug).first()  
    context = {'item': item}  
    return render(request, 'carbuild_post.html', context)
```

```
def traktorbuildPost(request, slug):  
    item = Item.objects.filter(slug=slug).first()  
    context = {'item': item}
```

```
return render(request, 'traktorbuild_post.html', context)
```

```
def chembuildPost(request, slug):
    item = Item.objects.filter(slug=slug).first()
    context = {'item': item}
    return render(request, 'chembuild_post.html', context)
```

settings.py

```
"""
```

Django settings for ToolProduction project.

Generated by 'django-admin startproject' using Django 3.0.3.

For more information on this file, see

<https://docs.djangoproject.com/en/3.0/topics/settings/>

For the full list of settings and their values, see

<https://docs.djangoproject.com/en/3.0/ref/settings/>

```
"""
```

```
import os
```

```
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
```

```
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

```
# Quick-start development settings - unsuitable for production
```

```
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/
```

```
# SECURITY WARNING: keep the secret key used in production secret!
```

```
SECRET_KEY = '$kitrg0r8^95slsmi^0fon#x+0ws3*%p%yuy5=!%in9zn^pb&g'
```

```
# SECURITY WARNING: don't run with debug turned on in production!
```

```
DEBUG = True
```

```
ALLOWED_HOSTS = []
```

```
# Application definition
```

```
INSTALLED_APPS = [  
    'directory',  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django_extensions',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'media',  
    'django_filters',  
  
]
```

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
    'django.middleware.locale.LocaleMiddleware',  
  
]
```

```
CSRF_COOKIE_SECURE = False  
SESSION_COOKIE_SECURE = False  
SECURE_SSL_REDIRECT = False
```

```
ROOT_URLCONF = 'ToolProduction.urls'
```

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': ['templates'],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  

```

```

        'django.template.context_processors.debug',
        'django.template.context_processors.request',
        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',
    ],
},
],
]

```

```
WSGI_APPLICATION = 'ToolProduction.wsgi.application'
```

```
# Database
```

```
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases
```

```
DATABASES = {
```

```
    'default': {
```

```
        'ENGINE': 'django.db.backends.mysql',
```

```
        'NAME': 'mydjango',
```

```
        'USER': 'root',
```

```
        'PASSWORD': '',
```

```
        'HOST': 'localhost',
```

```
        'PORT': '3306',
```

```
    }
```

```
}
```

```
# Password validation
```

```
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators
```

```
AUTH_PASSWORD_VALIDATORS = [
```

```
    {
```

```
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
```

```
    },
```

```
    {
```

```
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
```

```
    },
```

```
    {
```

```
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
```

```
    },
```



```
{
    'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
},
]
```

Internationalization

<https://docs.djangoproject.com/en/3.0/topics/i18n/>

LANGUAGE_CODE = 'uk-UA'

LOCALE_PATHS = [os.path.join(BASE_DIR, 'locale')]

TIME_ZONE = 'Europe/Kiev'

USE_I18N = True

USE_L10N = True

USE_TZ = True

Static files (CSS, JavaScript, Images)

<https://docs.djangoproject.com/en/3.0/howto/static-files/>

STATIC_ROOT = os.path.join(os.path.dirname(BASE_DIR), 'static')

STATIC_ROOT = os.path.join(BASE_DIR, 'static')

STATIC_URL = '/static/'

MEDIA_ROOT = os.path.join(os.path.dirname(BASE_DIR), 'media')

MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

MEDIA_URL = '/media/'

STATICFILES_DIRS = (

os.path.join(BASE_DIR, 'media'),

)

admin.py

```
from django.utils.html import format_html
from .models import *
from django.contrib import admin
from django.conf import settings

class ProductAdmin(admin.ModelAdmin):
    def picture(self, obj):
        return format_html(f'')

    picture.allow_tags = True
    list_display = ('title', 'picture')
admin.site.register(Category, ProductAdmin)
admin.site.register(Item, ProductAdmin)
```