

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ  
СІКОРСЬКОГО» ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА  
ОБЧИСЛЮВАНОЇ ТЕХНІКИ КАФЕДРА ОБЧИСЛЮВАНОЇ  
ТЕХНІКИ

КУРСОВА РОБОТА

з дисципліни: "Інженерія програмного забезпечення"

(назва дисципліни)

на тему: Система управління відкритими даними

Студента 2 курсу ІО-91 групи

спеціальності комп'ютерна інженерія

Кармазін Назар Валентинович

(прізвище та ініціали)

Керівник

Болдак А.О

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Київ - 2021 рік

## **Зміст**

### **Вступ**

#### **1 Постановка задачі**

##### **1.1 Вимоги до сайту**

##### **1.2 Аналіз існуючих рішень**

##### **1.3 Платформа для розробки**

#### **2 Проектування програмного забезпечення**

##### **2.1 Можливості акторів**

##### **2.1.1 Use case**

#### **3 Проектування графічного інтерфейсу користувача**

##### **3.1 Макет графічного інтерфейсу**

#### **4 . Короткі нотатки відносно коду**

##### **4.1 Frontend**

##### **4.2 Backend**

#### **5. Скріншоти готового проекту**

### **Висновки**

## ВСТУП

Об'єктом розробки даної курсової роботи є створення системи управління відкритими даними.

Метою курсової роботи є закріплення теоретичних знань і практичних навичок з проектування, моделювання, розробки та тестування програмного забезпечення з графічним інтерфейсом.

## 1. ПОСТАНОВКА ЗАДАЧИ

### 1.2 Вимоги до сайту

Наша мета створити стабільну системи управління відкритими даними з зручним інтерфейсом, з можливістю розміщувати на сайті свої файли та завантажувати файли інших користувачів.

### 1.3 Аналіз існуючих рішень.

критерії/сервіси	Єдиний державний веб-портал відкритих даних	Національна комісія з цінних паперів та фондового ринку	Державне агентство з питань електронного урядування України	Відкриті дані Верховно Ради України	Розшукові обліки Міністерства внутрішніх справ України
Functionality(функціональні вимоги)	набіри даних розділені за категоріями, зручний інтерфейс	інтерфейс користувача на середньому рівні	інтерфейс користувача на середньому рівні	інтерфейс користувача на середньому рівні	інтерфейс користувача на середньому рівні
Можливість розміщення контенту користувачем	відсутня	відсутня	відсутня	відсутня	відсутня
Usability(вимоги до зручності роботи)	онлайн-доступ	онлайн-доступ	онлайн-доступ	онлайн-доступ	онлайн-доступ
Наявність пошукового фільтру	наявний	наявний	наявний	наявний	наявний
Багатомовність	часткова(англ/ укр)	часткова(англ/ укр)	наявна	часткова(англ/ укр/рос)	відсутня
Reliability (вимоги до надійності)	висока надійність	надійна	надійна	надійна	надійна
Актуальність даних	регулярне оновлення	регулярне оновлення	регулярне оновлення	регулярне оновлення	регулярне оновлення
Performance (вимоги до продуктивності)	висока продуктивність	висока продуктивність	висока продуктивність	висока продуктивність	висока продуктивність
Supportability (вимоги до підтримки)	потрібне постійне оновлення даних	потрібне постійне оновлення даних	потрібне постійне оновлення даних	потрібне постійне оновлення даних	потрібне постійне оновлення даних

## 1.4 Платформа для розробки

Для реалізації нашого проекту ми обрали Django(Python), так як, вона була знайома нам, давала можливість одночасно працювати з базами даних, їх підключенням та користувацьким інтерфейсом. Також в проекті були використані html, json, css

Короткі нотатки відносно Django

**Django** (Джанго) — високорівневий відкритий Python-фреймворк (програмний каркас) для розробки веб-систем.

Сайт на Django будується з однієї або декількох частин, які рекомендується робити модульними. Це одна з істотних архітектурних відмінностей цього фреймворку від деяких інших .

Архітектура Django подібна на «Модель-Вигляд-Контролер» (MVC). Однак, те що називається «контролером» в класичній моделі MVC, в Django називається «вид» (англ. *view*), а те, що мало б бути «видом», називається «шаблон» (англ. *template*). Таким чином, MVC розробники Django називають MTV («Модель-Шаблон-Вигляд»).

Початкова розробка Django, як засобу для роботи новинних ресурсів, досить сильно позначилася на його архітектурі: він надає ряд засобів, які допомагають у швидкій розробці вебсайтів інформаційного характеру. Так, наприклад, розробнику не потрібно створювати контролери та сторінки для адміністративної частини сайту, в Django є вбудований модуль для керування вмістом, який можна включити в будь-який сайт, зроблений на Django, і який може керувати відразу декількома сайтами на одному сервері. Адміністративний модуль дозволяє створювати, змінювати і вилучати будь-які об'єкти наповнення сайту, протоколюючи всі дії, а також надає інтерфейс для управління користувачами і групами (з призначенням прав). У дистрибутиві Django також включені програми для системи коментарів, синдикації RSS і Atom, «статичних сторінок»(якими можна управляти без необхідності писати контролери та відображення), перенаправлення URL та інше.

## 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Можливості акторів:

Користувач

- переглядати файли
- завантажувати файли з сайту
- завантажувати файли на сайт

Адміністратор(модератор)

- переглядати файли
- завантажувати файли з сайту
- завантажувати файли на сайт
- редагувати файли
- переглядати інформацію користувачів

## Система

- завантаження файлів видалення файлів

- видалення файлів

**НАЗВА:** Завантаження файлів з набору даних

**УЧАСНИКИ:** Користувач, система

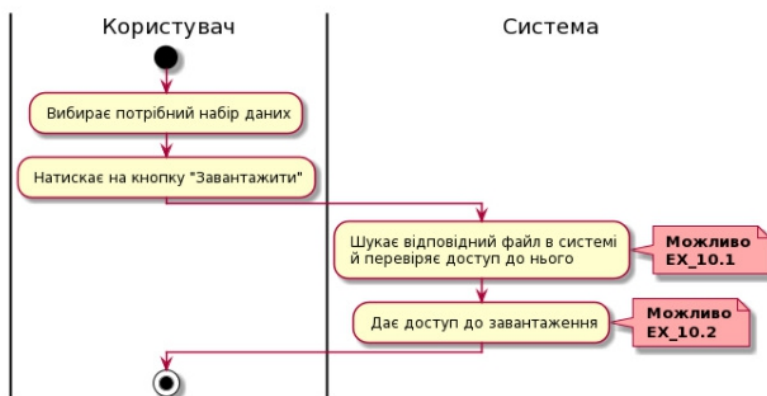
**ПЕРЕДУМОВИ:** В системі присутній набір даних, з якого будуть завантажувати файли

**РЕЗУЛЬТАТ:** Завантажені файли

**ВИКЛЮЧНІ СИТУАЦІЇ:**

- EX\_10.1 Вибраний файл на даний момент недоступний
- EX\_10.2 Користувач не підтверджує завантаження

**ОСНОВНИЙ СЦЕНАРІЙ:\***



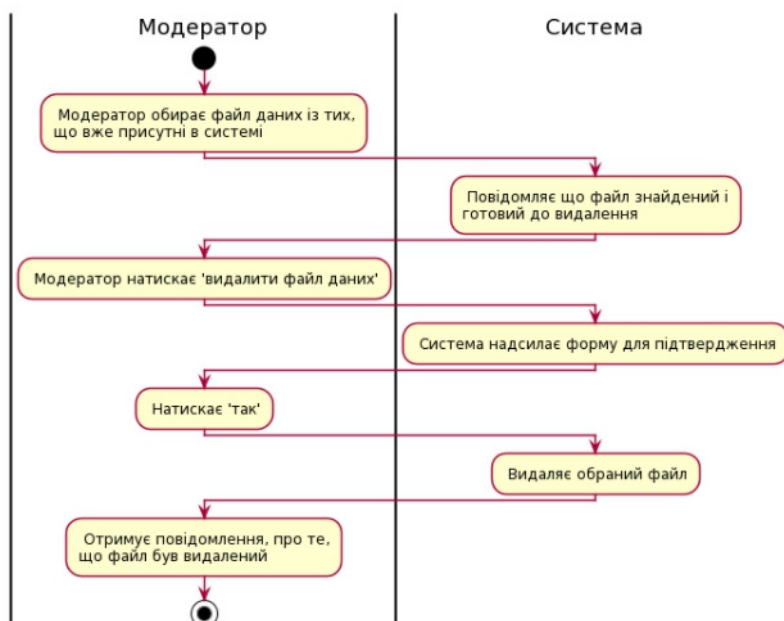
**НАЗВА:** Видалити файл даних

**УЧАСНИКИ:** Модератор, система

**ПЕРЕДУМОВИ:** Модератор увійшов в систему

**РЕЗУЛЬТАТ:** Файл даних видалено

**ОСНОВНИЙ СЦЕНАРІЙ:**



**НАЗВА:** Додати набір даних

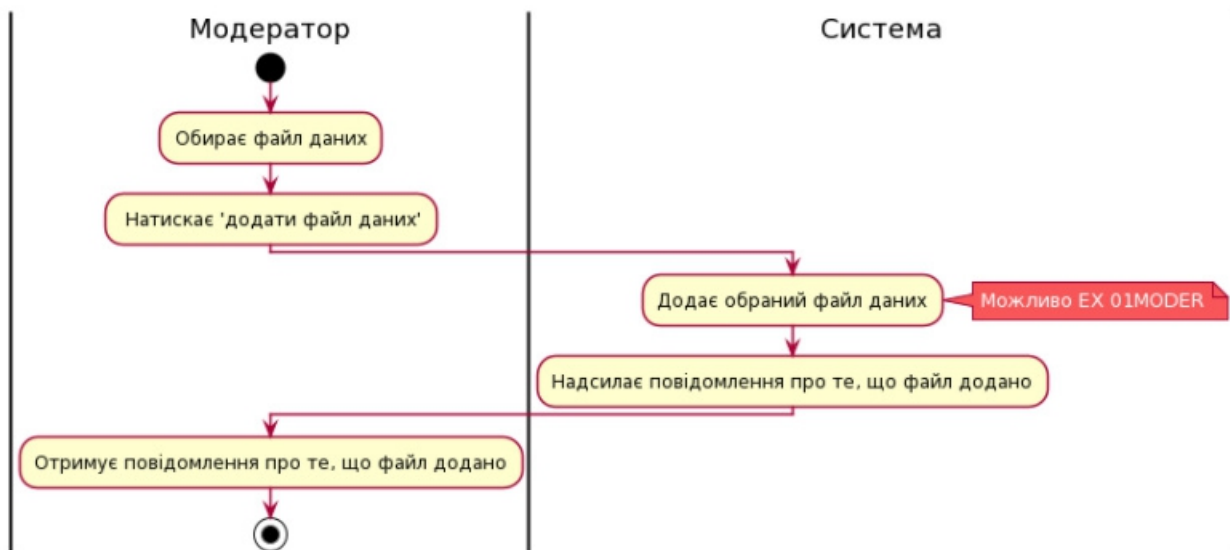
**УЧАСНИКИ:** Модератор, система

**ПЕРЕДУМОВИ:** Модератор ввійшов в систему

**РЕЗУЛЬТАТ:** Файл даних додано

**ВИКЛЮЧНІ СИТУАЦІЇ:** Відсутні

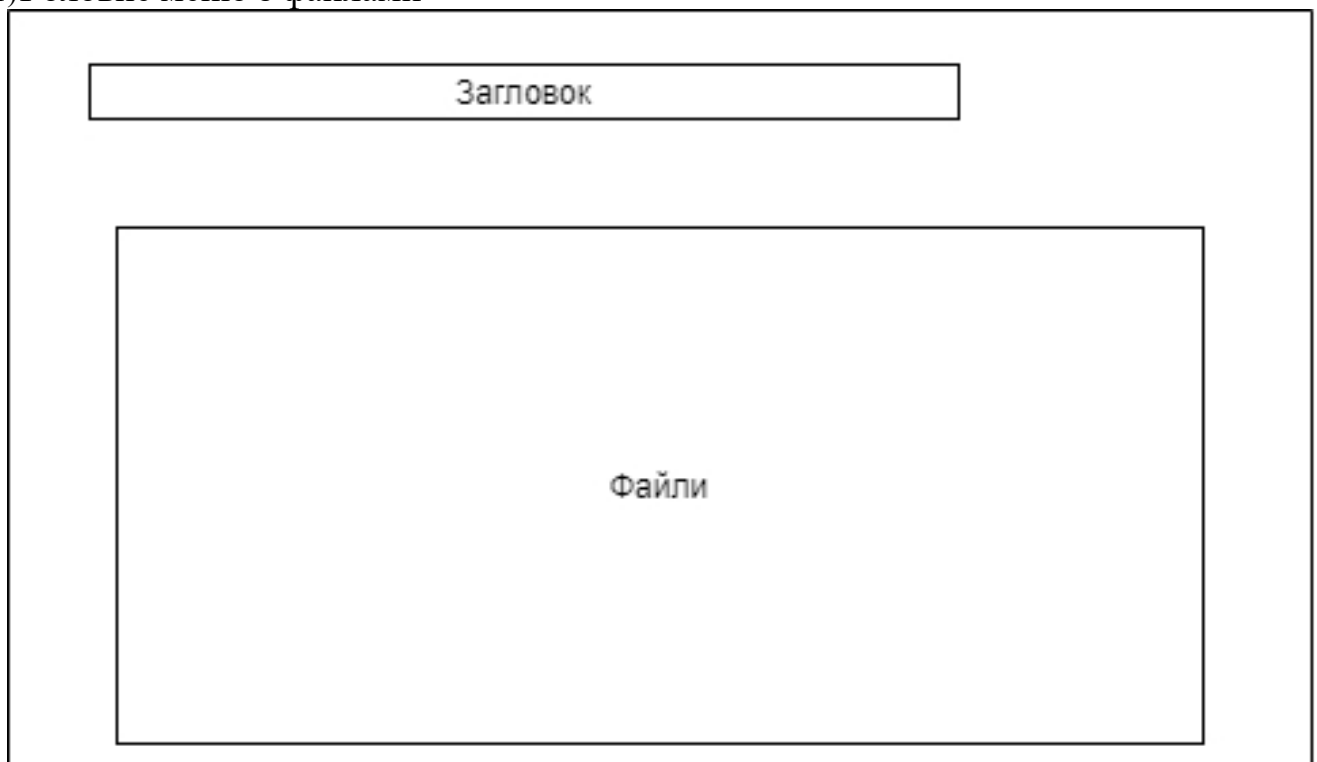
**ОСНОВНИЙ СЦЕНАРІЙ:**



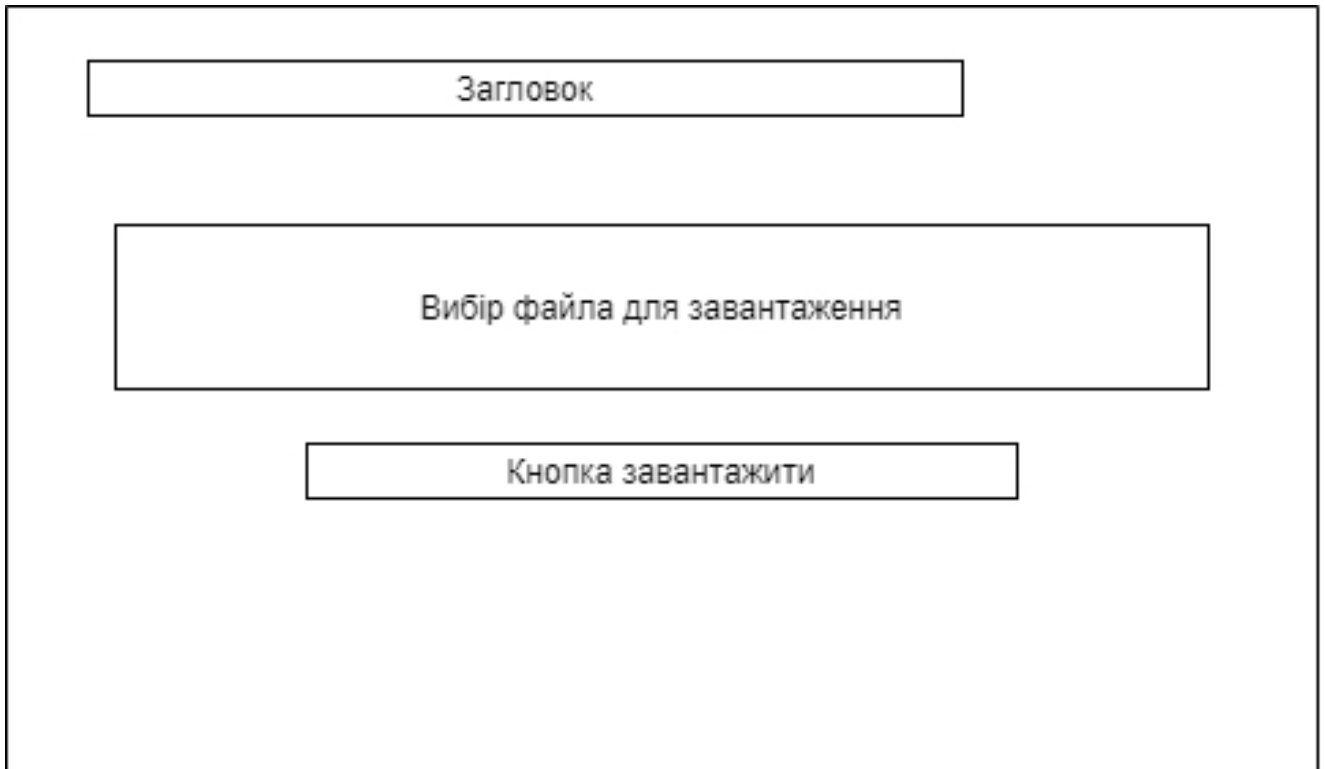
### 3. Проектування графічного інтерфейсу користувача

#### 3.1.1 Макет графічного інтерфейсу

##### 1) Головне меню з файлами



## 2)Інтерфейс для завантаження на сайт



The diagram illustrates a simple web interface for file uploads, enclosed in a large rectangular frame. It consists of three vertically stacked rectangular boxes. The top box is labeled 'Заголовок' (Header). The middle box is labeled 'Вибір файла для завантаження' (File selection for upload). The bottom box is labeled 'Кнопка завантажити' (Upload button).

## 4. Короткі нотатки відносно коду

### 4.1 Frontend

Для Frontend частини були використанні HTML та CSS.Відносно макету графічного дизайну були спроектовані відповідні html каскади та створений дизайн котрий був задуманий на початку розробки. Головна сторінка знаходиться в файлі file.html, кожна сторінка має окремий html файл,та тісно пов'язана с Backend частиною. Файли типу .html мають специфічні теги типу {% ... %}, це теги, які є притаманними для Django і використовуються шаблонізатором Jinja.

### Приклад HTML фрагменту

```
<ol class="breadcrumb">
  <li class="breadcrumb-item">
    <a href="#"></a>
    <a class="btn btn-sm btn-success" href="{% url 'form' %}"
style="padding: 8px; float: right; background-color: green; color:
white;">Upload</a>
  </li>
</ol>
```















## Приклад CSS фрагменту

```
#bg{  
  
    background-repeat: no-repeat;  
    background-size: 1800px 900px;  
    background-position: top;  
    background-attachment: fixed;  
}
```

База даних створена за допомогою MS SQL Server

## Структура бази даних

### Table

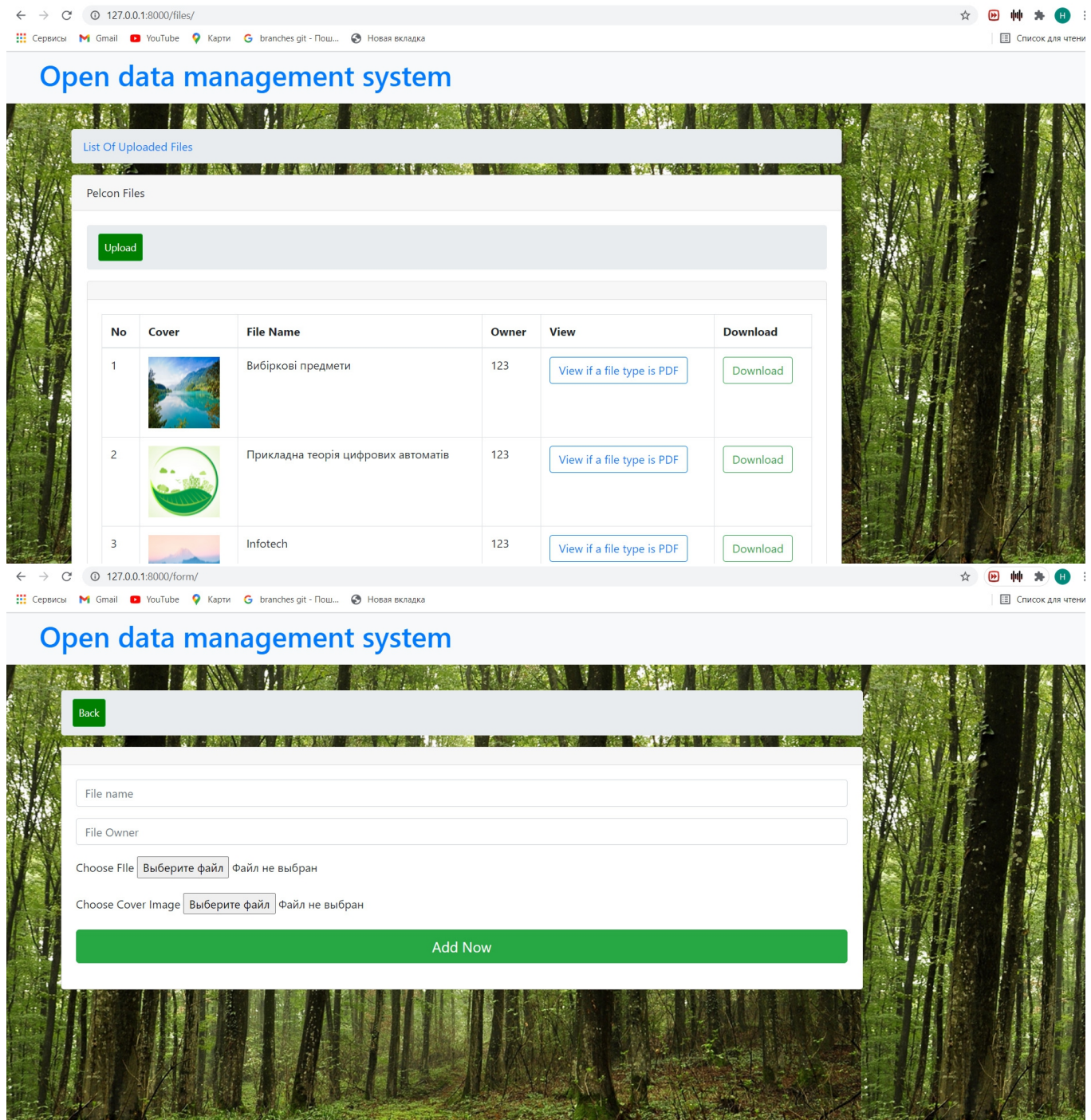
-  account\_user
-  account\_user\_groups
-  account\_user\_user\_permissions
-  auth\_group
-  auth\_group\_permissions
-  auth\_permission
-  comment\_files
-  comment\_pelcon
-  comment\_youtube
-  django\_admin\_log
-  django\_content\_type
-  django\_migrations
-  django\_session
-  sqlite\_sequence

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),  
    }  
}
```

Ініціалізація бази даних в файлі settings.py



## 5. Скриншоти готового проекту



### Висновки:

На мою думку я зміг реалізувати всі поставлені задачі. Сайт має приємний дизайн та гарну бекенд частину з можливістю розширювати функціонал сайту. Ми змогли багато чому навчитися та покращили свої навички програмування, отримали досвід створення свого проекту та пройшли всі його етапи.

