Лицей Академии Яндекса  
 Образовательная площадка «Супермалыш»

Проект WEB

Личный сайт с клипами GameClip studio

Выполнили:  
 Минниахметова Ева

Саитов Кирилл

Проверил:  
 Гильдин А. Г.

Уфа – 2025

Задача:

Написать сайт с использованием фреймворка flask.

Цель проекта:

Создать функциональное веб-приложение "MyClips" на базе Flask, обеспечивающее пользователю удобный интерфейс для организации, просмотра и базового управления локальной коллекцией видеоклипов, и просмотра их удаленно с других устройств.

Задачи, решенные в проекте:

* Разработка серверной части на Flask: Создание базовой архитектуры приложения, обработка HTTP-запросов и маршрутизация.
* Реализация механизма сканирования локальной директории: Обеспечение возможности указания папки с видеофайлами и их динамического отображения.
* Интеграция с FFmpeg для генерации превью: Автоматическое создание визуальных миниатюр для видеофайлов в формате .webp с целью быстрой идентификации контента и оптимизации загрузки.
* Разработка пользовательского интерфейса: Создание HTML-шаблонов для отображения списка клипов, страницы просмотра и элементов управления с использованием Jinja2.
* Реализация функции поиска: Предоставление пользователю возможности поиска клипов по имени файла.
* Обеспечение функционала просмотра видео: Интеграция HTML5 видеоплеера для воспроизведения выбранных клипов.
* Реализация операций управления файлами: Добавление функций удаления и переименования видеофайлов и связанных с ними превью непосредственно через веб-интерфейс.
* Внедрение системы "избранного": Предоставление возможности отмечать понравившиеся клипы и просматривать их отдельно, с сохранением состояния в cookies браузера.

2. Архитектура и описание работы приложения (версия 0.1)

Приложение "MyClips" построено на клиент-серверной архитектуре, где серверная часть реализована на Python с использованием фреймворка Flask.

Серверная часть (Flask):

* Маршрутизация (app.py): Определяет эндпоинты для различных действий пользователя (главная страница, просмотр видео, удаление, переименование, добавление в избранное, смена вида).
* Логика обработки: Функции-обработчики маршрутов выполняют основную логику:
* Сканирование директории CLIPS\_FOLDER для получения списка видеофайлов.
* Вызов функции generate\_thumbnail для создания превью. Эта функция, в свою очередь, использует модуль subprocess для запуска внешней утилиты ffmpeg с необходимыми параметрами.
* Взаимодействие с файловой системой для чтения, удаления и переименования файлов (модуль os).
* Рендеринг HTML-шаблонов с передачей необходимых данных (список клипов, информация о конкретном клипе и т.д.) с помощью шаблонизатора Jinja2.
* Обработка POST-запросов для операций изменения данных (удаление, переименование, избранное).
* Управление cookies для сохранения пользовательских предпочтений (избранное, режим отображения).
* Конфигурация: Основные параметры, такие как путь к папке с клипами (CLIPS\_FOLDER), папка для превью (THUMBNAILS\_FOLDER) и разрешенные расширения файлов, задаются в конфигурации приложения.

Клиентская часть (Браузер):

* HTML-шаблоны (templates/): Определяют структуру веб-страниц. Используется наследование шаблонов (например, base.html) для переиспользования общих элементов интерфейса.
* CSS (пользовательские стили): Отвечают за визуальное оформление приложения.
* JavaScript: Используется для добавления интерактивности на стороне клиента:
* Отправка AJAX-запросов (например, при добавлении в избранное или смене вида) без перезагрузки всей страницы.
* Динамическое изменение классов CSS для визуального отображения состояния (например, активная кнопка избранного).
* Обработка событий пользователя (клики по кнопкам).

Процесс работы пользователя:

1. Пользователь запускает Python-скрипт app.py. Flask запускает локальный веб-сервер.
2. Пользователь открывает в браузере адрес сервера (например, http://localhost:5000).
3. Приложение сканирует указанную в конфигурации папку CLIPS\_FOLDER.
4. Для каждого найденного видеофайла (с разрешенным расширением) генерируется превью (если оно отсутствует) и сохраняется в папку static/thumbnails.
5. На главной странице отображается список клипов с их превью.
6. Пользователь может выполнять поиск, переключать вид, добавлять клипы в избранное, переименовывать или удалять их, а также просматривать видео. Все операции, связанные с изменением файлов, производятся непосредственно на локальном диске пользователя.

3. Функциональные возможности (версия 0.1)

* Генерация превью в формате WebP: Для каждого видеофайла создается легковесное превью, что ускоряет загрузку страницы и улучшает визуальное восприятие каталога. Используется команда ffmpeg с параметрами для извлечения кадра и конвертации в .webp.
* Поиск: Реализован простой поиск по части имени файла.
* Воспроизведение видео: Интегрированный HTML5-видеоплеер для просмотра выбранных клипов.
* Управление файлами:
* Переименование: Изменение имени видеофайла и автоматически связанного с ним файла превью.
* Удаление: Удаление видеофайла с диска и его превью.
* Система "Избранное":
* Возможность отметить клипы как "избранные".
* Отдельная страница для просмотра только избранных клипов.
* Статус избранного сохраняется в cookies браузера.

Безопасность (базовая): Используется secure\_filename для предотвращения проблем с именами файлов при переименовании. Проверки на выход за пределы CLIPS\_FOLDER при операциях с файлами.

4. Использованные инструменты и технологии (версия 0.1)

Python 3: Основной язык разработки серверной части.

* Flask (версия, указанная в requirements.txt или последняя стабильная на момент разработки): Микрофреймворк для создания веб-приложения. Выбран из-за своей легковесности, гибкости и соответствия ТЗ.
* Jinja2: Шаблонизатор, интегрированный во Flask, для генерации динамических HTML-страниц.
* Werkzeug: WSGI-утилита, на которой основан Flask, используется для обработки HTTP-запросов и маршрутизации.
* FFmpeg: Внешняя кроссплатформенная утилита для обработки видео и аудио. Используется для извлечения кадров из видео и создания превью. Вызывается из Python через модуль subprocess.
* HTML5, CSS3: Для структурирования и стилизации веб-страниц.
* JavaScript (Vanilla JS): Для реализации интерактивных элементов на стороне клиента (переключение вида, добавление в избранное, отправка асинхронных запросов).
* Cookies браузера: Используются для сохранения пользовательских настроек (статус избранного для клипов, выбранный режим отображения).

Модули Python:

* os: Для взаимодействия с файловой системой (листинг директорий, проверка существования файлов, удаление, переименование).
* subprocess: Для запуска внешних процессов (ffmpeg).
* datetime: Для работы с датами и временем (например, дата создания файла).
* Система контроля версий: Git.
* Хостинг кода: GitHub (github.com/kirillsaitow/MyClips).