Author

Mehak Singal
21f1006390
21f1006390@ds.study.iitm.ac.in
I'm passionate about exploring new concepts and ideas related to this field, and I like to read books and articles to deepen my knowledge.

Description

This project is a social networking web application that allows users to post pictures and browse through other user's activity feeds. It involves using various technologies such as Flask, SQLite, and Vuejs to create an intuitive interface that enables users to create and share engaging content and stay connected with their followers.

Technologies used

HTML: Used for structuring and displaying the web pages.

Bootstrap, CSS: Used for styling and designing the web pages.

Python: Used as the primary programming language to build the back-end of the web application.

JavaScript, Vue.js: for adding interactivity and dynamic behaviour to the application.

SQLite: Used as the database management system to store and retrieve data related to the web application.

Jinja2: Used as a template engine to render dynamic HTML pages with data from the database.

Flask: Used as the web framework to handle HTTP requests and responses, and to connect the front-end and back-end of the web application.

Flask-Security: Used as an extension to add authentication and authorization features to the web application, such as user registration, login, and logout.

Flask-SQLAlchemy: Used as an extension to provide an Object-Relational Mapping (ORM) layer between the Python code and the SQLite database, making it easier to manipulate the data.

Celery: Used as a distributed task queue to handle background tasks asynchronously, such as sending email notifications or generating PDF reports.

PDFKit: Used as a Python wrapper for the wkhtmltopdf tool to generate PDF files from HTML pages.

DB Schema Design

Table: Account

- Columns: account_id (primary key), email, name, password, profile_picture, active, fs_uniquifier
- Constraints: account_id, email, fs_uniquifier must be unique. All are non-nullable.

Table: UserPosts

- Columns: id (primary key), account_id (foreign key), title, caption, post_picture, archive, timestamp
- Constraints: account_id must exist in User table

Table: Follow

- Columns: id (primary key), account_id (foreign key), following_others
- Constraints: account_id and following_others must exist in User table

This schema allows for the storage of user information, posts, and follows. The foreign key constraints ensure referential integrity, while the use of primary keys ensures each record can be uniquely identified.

API Design

RESTful API was created using Python and Flask. The API provides endpoints for creating, updating, and deleting blog posts, managing user accounts, and retrieving post/follows data. API was also used for celery tasks like importing and exporting csv files. SQLAlchemy was used to create a scalable and maintainable API. Additionally, token-based authentication was used to secure user data. HTTP requests - GET, POST, PUT, DELETE.

Architecture and Features

The controllers are located in the main.py file, which contains all the APIs and controllers for the project. The database models are defined in the models.py file, and the templates and static files are located in their respective folders.
The project has a variety of features implemented to enhance the user experience. These include token-based authentication for security, CRUD operations for posts and accounts, the ability to follow and unfollow other users, daily notifications via webhooks and Celery, and monthly reports delivered via email in both PDF and HTML formats using PDFkit and Celery. The application is designed using Bootstrap for a responsive layout, ensuring that it can be used on any screen size. Additionally, users can import and export posts in CSV format using Celery and Redis, further enhancing the flexibility of the application.

Video

https://drive.google.com/file/d/1zCKcYhSgyUr7YYXKfMFB-Hdgq7eFbPvS/view?usp=sharing