

Description

Bloglite v2 is the MAD-2 project for January-2023 Term of IITM BS Degree. In this we need to create a blog application. Users create blogs and react to other user's blogs. Users follow users and see the following's blogs in the homepage.

Technologies used

- Flask – for backend server
- Jinja – for templating the pdf reports
- Vue - for frontend
- Flask-JWT-Extended – for user authentication token based
- Flask-Migrate (alembic) – for database versioning and migration
- Flask-RESTful – for API routes
- Flask-SQLAlchemy – for database Object Relational Mapper
- Redis – Caching, Message Broker
- Axios - for frontend to backend requests
- Python-dotenv – to load config environmental variables from .env file
- SQLite3 – for database storage
- Bash – for run script
- Flask-sse – for SSE events
- WeasyPrint - for PDFs
- Celery – Batch Jobs

DB Schema Design

User:

column	details
id	Integer, Primary Key
username	String(64), Unique
password_hash	String(512)
name	String(50)
about	String(256)
joined	DateTime
photoURL	String(512)

Post:

column	details
id	Integer, Primary Key
title	String(50)
caption	Text
imageURL	String(512)
author_id	Integer, Foreign Key
time	DateTime

Followers:

column	details
follower_id	Integer, Foreign Key, Primary Key
followed_id	Integer, Foreign Key, Primary Key

Comment:

column	details
id	Integer, Primary Key
content	Text
time	DateTime
post_id	Integer, Foreign Key
author_id	Integer, Foreign Key

Vote:

column	details
id	Integer, Primary Key
score	Integer
post_id	Integer, Foreign Key
author_id	Integer, Foreign Key

- Followers table is required as following is many to many relationship, so it is join table between Users.
- Vote stores foreign key to author (user) and post
- Post stores foreign key to author
- Comment stores foreign key to author (user) and post

API Design

API for CRUD of user and posts, and comments, and vote, and login are created as well as API for listing all the posts, and comments of each posts, and votes of each posts and all the users and all users being followed.. More details are present in the **openapi.yaml** file.

Architecture and Features

- The **CONTROLLERS** are located in the "**ticketshow/api**" directory, which contains Python files such as **auth.py**, **error.py**, **user.py**, **post.py**, **comment.py**, **vote.py**, and **photo.py**, responsible for handling the different aspects of the application's routing logic.
- The **VIEWS**, used for rendering the views, are stored in the "**ticketshow/frontend/src/pages/views**" directory, with subdirectories for each component, such as "**EditPostView.vue**," "**ExploreView.vue**," "**FollowingView.vue**," "**LoginView.vue**," and "**ProfileView.vue**." etc These files contain vue files specific to the various features of the application, like **login, registration, and managing blogs and users**.
- The project also contains a "**components**" folder, housing the components reused in the views.
- The main application file, **app.py**, is located in the root directory, along with other essential files like:
 - **config.py** for configuration,
 - **models.py** for **database models**,
 - **auth.py** for authentication
 - **cache.py** for caching
- Finally, there's a "**ticketshow/frontend/router**" directory for the frontend routes using VueRouter.

Features:

- **Authentication** - Login, and Register pages to perform necessary authentication. Users can register and login. Auth is handled using JWT Tokens.
- **User Homepage** - User can see all the posts of users they follow and also upload their own posts.
- **Search** - Users can search all the other users using their name and username.
- **Profile** - Users can see their own profile and change their details and profile photo and password.
- **Explore** - Users can see profile details of other users too.
- **Follow** - Users can follow or unfollow other users.
- **Like, Dislike, Comment** - users can like, dislike, comment on other's and their own posts.
- **Export** - User can export their own posts as csv which is user-triggered batch job and sent via SSE
- **Reminder and Report** - Everyday users are reminded to post. Every month the usage report is mailed to the admin.

Video

https://drive.google.com/file/d/1mb3peVl0Gs8lxs1qOpCFK5RbJtOAvoW_/view?usp=share_link