Author

- Arshi Khan
- 21f3002806
- 21f3002806@student.onlinedegree.iitm.ac.in

- I am a full time BS student and a Data Science enthusiast and have started my programming journey with python, with time I have developed interest in coding and would love to see a career out of it.

Description

Basically it is similar to social media applications we use, having login and register, with the ability to post and view others and follow/unfollow, including search other users.

View other user's profile and ours as well, in the end we have logout and delete account permanently option as well.

Technologies used

1. Flask - Being the framework
2. Flask-restful - for creating API's
3. Flask SQLAlchemy - for database in flask
4. SQLalchemy- as database
5. Werkzeug - for handling exceptions
6. Json - for accessing and then showing data on frontend
7. Datetime- for posts having date and time creation
8. Base64- for image data

DB Schema Design

User Table

1. id = db.Column(db.Integer, primary_key=True)
2. name=db.Column(db.String(250), nullable=False)
3. username=db.Column(db.String(250),unique=True, nullable=False)
4. email = db.Column(db.String(250), nullable=False)
5. password=db.Column(db.String(250), nullable=False)

BlogPost Table

1. post_id = db.Column(db.Integer, primary_key=True)
2. author_username = db.Column(db.String, db.ForeignKey(User.username))
3. title = db.Column(db.String(250))
4. date = db.Column(db.DateTime, default=datetime.datetime.utcnow)
5. body = db.Column(db.Text)
6. img = db.Column(db.String(250))

Comment Table

1. comment_id = db.Column(db.Integer, primary_key=True)
2. text = db.Column(db.Text, nullable=False)

Like Table

1. like_id = db.Column(db.Integer, primary_key=True)
2. like_username = db.Column(db.String, db.ForeignKey(User.username))
3. post_id = db.Column(db.Integer, db.ForeignKey(BlogPost.post_id))

Follow Table

1. id = db.Column(db.Integer, primary_key=True)
2. follower_username = db.Column(db.String, db.ForeignKey(User.username))
3. followed_username = db.Column(db.String, db.ForeignKey(User.username))

API Design

- User - CRUD on user
- Posts - CRUD on posts
- Like/Dislike - ability to like and dislike
- Follow/Unfollow - ability to follow and unfollow others
- News feed - having all posts on user’s feed
- Comment- ability to comment on posts
- Own Profile- ability to view own profile
- Other’s Profile- ability to view other’s profile
- All users - ability to view all users of this site
- Search - ability to search

Architecture and Features

1. Api
    a. Instance
        - api_database.sqlite3
    b. App.py
    c. models.py
2. Frontend
    a. Static
        - Css
    b. Templates
    c. main.py

Features

Completed core requirements

- ☑ ~~Login/Sign up page~~
- ☑ ~~User’s feed - posts, follow information etc.~~
- ☑ ~~User’s profile~~
    - ☑ ~~No. of posts~~
    - ☑ ~~following~~
    - ☑ ~~followers~~
- ☑ ~~Blog/Post management~~

Completed Recommended requirements

- ☑ ~~APIs for~~
    - ☑ ~~User~~
    - ☑ ~~Blogs~~
    - ☑ ~~Others~~
- ☑ ~~Validation on input fields~~
- ☑ ~~Blogs engagement~~

Completed optional requirements

- ☑ ~~Styling and Aesthetics~~
- ☑ ~~Proper login system~~
- ☑ ~~Exporting blog engagements~~

Video

Project presentation video