Author

- **Name :** J S RANJITH
- **Roll Number :** 21f1002171
- **Email :** 21f1002171@student.onlinedegree.iitm.ac.in
- **About me :** I am self motivated and a hardworking person , I will not settle until I achieve my goal.
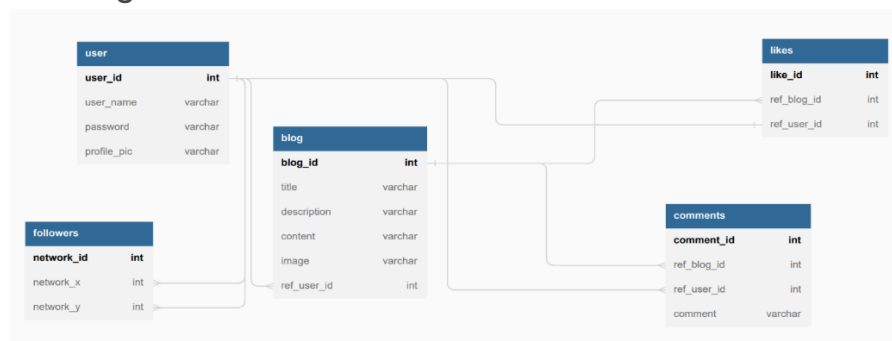
Description

- **Blog Lite** is a multiuser application , where each user should be able to create , view , and update their account.
- Each user should be able to create , read , update and delete their blogs.

Technologies used

- **Vue**
    Used as a framework for building user interface.
- **flask**
    Used for developing the application.
- **flask_restful**
    Used for developing the REST API.
- **Flask SQLAIchamey**
    Used for creating database models.
- **Flask Security**
    Used for login authentication (Token based Authentication).
- **Sqlite3**
    Used for storing the data.
- **HTML**
    Used for creating interactive webpages.
- **CSS (Bootstrap)**
    Used for styling the webpages.

DB Schema Design

- In my DB schema user and blog table have a **one-many** relationship because the application allows user to create multiple blogs individually.
- Columns **network_x** and **network_y** (**followers** tables) are foreign key constrained to column **user_id** (**user** table) , because the followers table is used for storing the followers and following information of users.

API Design

- API was created using **flask_restful** for performing CRUD operations on blogs and to retrieve all sort of data from the database.
- Some of the example API endpoints are
  - **GET , PUT and DELETE** operations are done on endpoint : **/api/blog/<blog_id>**
  - **POST** operation is done on endpoint : **/api/create_blog**

Architecture and Features

- **Project folder**
  - **Backend**
    - **api.py file (**API resources**)**
    - **config.py (**configurations are specified in this file**) and models.py (**contains the DB models**)**
    - **instance folder**
      - **bloglite.sqlite3** (database of the application)

  - **Frontend**
    - **templates folder** (contains index.html file)
    - **static folder** (contains CSS files)
      - **blog_images folder** (stores images of blogs)
      - **profile_pic folder** (stores profile pictures of users)
      - **components folder** (stores Vue components)
    - **app.py file** (controllers and models)
  - **Project Documentation file** (pdf file)
  - **readme.md file** (markdown file)
- User can create a account using sign up.
- User is allowed to sign in only if the user has a account , done by checking if the user detail entered for sign in is present in the database or not.
- User can edit his/her profile.
- One can search for other users within the application , done by querying for the usernames which contains the search value.
- Users can follow or unfollow other users.
- Users are allowed to create , edit , read and delete the blogs or posts.
- User's feeds are updated as they follow other users in the application.
- User's get reminders and reports which are scheduled jobs of the application.

Video

https://drive.google.com/file/d/157Wg8cRzZFY8wKebp9WyuNTWcphHLT9R/view?usp=share_link