

## Author

- **Name** : J S RANJITH
- **Roll Number** : 21f1002171
- **Email** : [21f1002171@student.onlinedegree.iitm.ac.in](mailto: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.

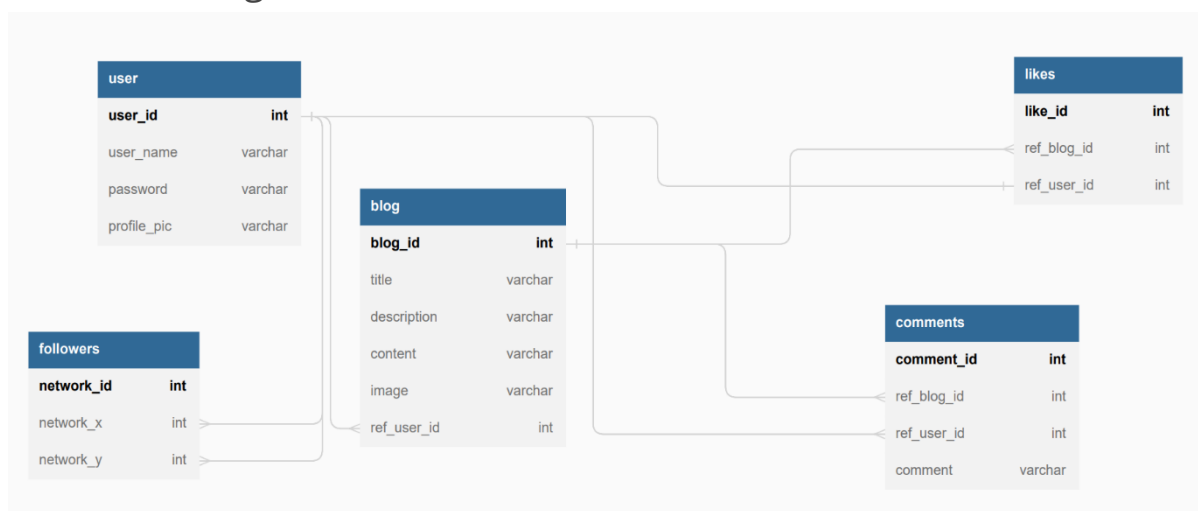
## Description

- **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.
- Users should also be allowed to follow and unfollow other users within the application.

## Technologies used

- **flask**  
Used for developing the application.
- **flask\_restful**  
Used for developing the REST API.
- **Flask SQLAlchemy**  
Used for creating database models.
- **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.
- Columns **ref\_blog\_id** and **ref\_user\_id** (**likes** and **comments** tables) are foreign key constrained to column **blog\_id** (**blog** table) and column **user\_id** (**user** table) respectively , because the likes and comments table is used for storing the likes and comments information of users.

## API Design

- API was created using **flask\_restful** for performing CRUD operations on blogs.
- **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**
  - **templates folder** (contains html files)
  - **static folder** (contains css files)
    - **blog\_images folder** (stores images of blogs)
    - **profile\_pic folder** (stores profile pictures of users)
  - **instance folder**
    - **bloglite.sqlite3** (database of the application)
  - **app.py file** (controllers and models)
  - **api.py file** (api resources)
  - **Blog\_API file** (yaml file)
  - **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.
- Users' feeds are updated as they follow other users in the application.
- A blog or post can be liked and commented by the users.
- Each of the above features are implemented as controllers with different endpoints

## Video

[https://drive.google.com/file/d/1ze\\_hYEVZC-NG9zOLXc33k3HEu4mvuevP/view?usp=sharing](https://drive.google.com/file/d/1ze_hYEVZC-NG9zOLXc33k3HEu4mvuevP/view?usp=sharing)