

# Project Report: Blog lite V2

## Author

Anirudh Murthy

21f1000931

[21f1000931@ds.study.iitm.ac.in](mailto:21f1000931@ds.study.iitm.ac.in)

Pursuing both online and offline degrees.

Passionate about learning and looking forward to growing as I move forward.

## Description

The Blog app is a web application built using the Flask web framework that allows users to create, read, update, and delete blog posts. Users can register for an account, log in, and view a list of all blog posts or view posts by individual authors. They can also create new blog posts, edit their own posts, or delete them. The application is built with a SQLite3 database to store user data and blog posts.

## Technologies used

This project is majorly divided into 2 parts: Frontend and Backend

### Frontend:

- Vue.js - it was used to build a SPA (Single Page Application).
- Bootstrap - it has been used to build different components of the application and improve its aesthetics.

### Backend:

- Python (FLASK) - it was used to build the api endpoints to interact with the backend and provide data to frontend
- SQLite3 - Database used to store data and interact with Flask API (Flask SQLAlchemy)
- Redis - it was used to store data and run backend job
- Celery - it was used as a message broker and as workers to complete jobs either triggered or auto.

## DB Schema Design

### 1.Users table

Column Name	Data Type	Constraints
id	Integer	Primary Key, Auto increment
username	String	Not Null
email	String	Not Null, Unique
password	String	Not Null
active	Boolean	—
fs_uniquifier	String(255)	Not Null, Unique

profile_pic	LargeBinary	—
roles	Relationship	Many to Many with Role
users_posts	Relationship	One to Many with Posts

## **2.Posts table**

Column Name	Data Type	Constraints
id	Integer	Primary Key, Auto Increment
title	String	Not Null
content	Text	Not Null
author_id	Integer	Not Null, Foreign Key to Users(id), ondelete=CASCADE
date_created	DateTime	Server Default=db.func.now()
date_modified	DateTime	—

## **3.Followers table**

Column Name	Data Type	Constraints
id	Integer	Primary Key, Auto Increment
user_id	Integer	Not Null, Foreign Key to Users(id), ondelete=CASCADE
following_id	Integer	Not Null, Foreign Key to Users(id), ondelete=CASCADE

## API Design

/get\_id: a route for getting the ID of the currently logged in user  
 /author: a route for getting information about the currently logged in author  
 /author/profile/<id>: a route for getting information about a specific author  
 /author/following: a route for getting the authors the current author is following  
 /author/followers: a route for getting the authors following the current author  
 /author/follow/<id>: a route for following an author  
 /author/unfollow/<id>: a route for unfollowing an author  
 /author/search: a route to search for an author  
 /author/post: a route to get the Logged in users posts  
 /author/delete: a route to delete current user  
 /post: a route to get all the posts  
 /author/post/<post\_id>/delete: a route to delete the current users post  
 /author/post/<post\_id>/edit: a route to edit the current users post  
 /DownloadCSV: a route to download the current users posts as csv  
 /report: a route to get the pdf report of the current user

## Architecture and Features

The project is split into two major parts namely Frontend and Backend. The backend includes the controllers in the controllers.py file, the database models, the celery jobs and workers, the template and the pdf generator file. The backend also includes the database instance. The frontend includes the Vue components and the javascript files.

Some features of the app include:

- Multi user support
- User can export all their posts as a csv file
- Daily reminders to check the blog or create a new post
- PDF reports can be emailed and downloaded as required
- User can search for other users
- Flask security-too enables user to login and signup securely along with token based authentication

## Video

<https://drive.google.com/file/d/1lhwUXZv2W75xiLlsS69OrOT5iYoxMgR3/view?usp=sharing>