January 4, 2021

BlogApp Report

Nishanth Moolam

This application was created as a means to familiarize myself to various concepts within the Flask framework. These concepts include:

* Class Based Views
* SQLAlchemy
* Table Model Relations
* Pagination
* Sessions

# Overview

BlogApp is an application that stores users, blogs, and comments in an SQLAlchemy database. Users can be added, and logged into using created credentials, and users can create blogs, or comment on existing blogs. All data is in the form of JSON, with CORS access enabled, intended to then be sent to a frontend application.

## Database Structure

There are three models in the database structure:

|  |  |
| --- | --- |
| User |  |
| id | Primary Key |
| username | String |
| email | Unique String |
| password | String (Hashed) |
| about | String |
| blogs | A query of Blogs this user authored |
| comments | A query of Comments this user authored |

|  |  |
| --- | --- |
| Blog |  |
| id | Primary Key |
| title | String |
| content | String |
| author\_id | Foreign Key (User) |
| date\_added | Datetime Object |
| comments | A query of Comments under this blog |

|  |  |
| --- | --- |
| Comment |  |
| id | Primary Key |
| content | String |
| author\_id | Foreign Key (User) |
| blog\_id | Foreign Key (Blog) |
| date\_added | Datetime Object |

These models form many to one relationships among each other. A user may author several blogs and comments, and a blog may contain several comments.

# Authentication & Security

New users can be created using a unique email address, and a self-made password. This password is never stored in plain text in the database, and uses ‘sha256’ hashing in the werkzeug security module to encrypt stationary data in the database. Users may then login by verifying their email and password. Once logged in, user data is stored in a permanent session.

# Class Based Views & Pagination

There is currently a user page, and blog page endpoint. These query the database and return all users or all blogs, and use pagination to prevent large query times. These views are within a flask blueprint, that is then registered in the main app to allow for more organization and file structure.

# Future Features

There are a few features, or revisions that I had intended to include, that may be added in the future. These include: Better Exception Handling, and Unit Testing. I have included exception handling for the most common of errors, however implementing unit testing could better protect from any error the program encounters.