Blogify

A Project Report Submitted to

MANIPAL ACADEMY OF HIGHER EDUCATION

For Partial Fulfillment of the Requirement for theof the Degree

Of

Bachelor of Technology

in

Computer and Communication Engineering

by

Rishabh, Shouvik, Bhavya

210953080, 210953174, 210953176

Under the guidance of

Ms. Chetana Pujari
Assistant Professor Senior-Scale
Department of I&CT
Manipal Institute of Technology
Manipal, Karnataka, India

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Unit of MAHE, Manipal

November 2023

ABSTRACT

Blogging has evolved from its humble beginnings as personal online diaries into a powerful platform for sharing knowledge, expressing creativity, and building communities. Our website explores the essence of blogging as a digital medium for communication and self-expression. It delves into the multifaceted world of blogging, highlighting its significance, evolution, and impact on society.

The application underscores the democratizing potential of blogging, empowering individuals and marginalized voices to share their stories and perspectives. It recognizes the role of blogs in knowledge dissemination and opinion formation, fostering a global dialogue.

In conclusion, the enduring relevance of blogging as a powerful tool for self-expression, community-building, and societal change. It invites readers to delve into the ever-expanding blogosphere, where the convergence of technology, creativity, and connectivity continues to redefine the way we communicate and connect in the digital age.

INTRODUCTION

This project introduces a user classification system, dividing users into two distinct categories: anonymous users and logged-in users. Anonymous users are granted the privilege to read posts, access individual blog pages for reading, and utilize basic reading features. In contrast, logged-in users enjoy extended privileges, including the ability to create posts, access all their personal posts, engage with posts through interactions, and subscribe to bloggers. Additionally, a comprehensive search functionality has been implemented to cater to the needs of all user types.

Blogify prides itself on providing an immersive and user-centric environment for diving into the captivating realm of blogging. Irrespective of their user type, all visitors can seamlessly explore a diverse array of blogs, delve into the nuanced realm of likes and dislikes associated with individual posts, and employ a powerful keyword-based search functionality to unearth posts of particular interest. For those seeking an elevated experience, Blogify extends the opportunity to create a logged-in user account through a straightforward registration process using their Email ID. Once logged in, users unlock an array of enhanced features, including access to a personalized dashboard where they can fine-tune their profile details, peruse their own collection of blogs, and delve into comprehensive blog analytics. The power to craft and edit posts, specifically those they own, adds to the allure of the platform. Moreover, users can actively engage with the vibrant blogging community by expressing their appreciation through likes or subscribing to their favorite bloggers, fostering a sense of connection and community within the Blogify ecosystem.

BACKGROUND INFORMATION

A Flask-based blogging website has been developed to serve as a dynamic and interactive platform for users interested in sharing and exploring programming-related content. This website provides a user-friendly interface and incorporates various features to enhance the overall blogging experience. Below is the background information for the programming laboratory report:

1. Purpose and Scope

The primary objective of Blogify is to create a space where individuals passionate about expressing their feelings can share their knowledge, insights, and experiences with a like-minded community. The website aims to foster collaboration, learning, and discussion on a wide range of topics.

2. Technology Stack

The website is built using Flask, a micro web framework for Python, known for its simplicity and flexibility. Flask facilitates the rapid development of web applications and allows seamless integration with other technologies. The backend utilizes MySQL as the database service for storing and retrieving blog-related information.

3. Future Developments

The blogging website is designed to be scalable, and future developments may include additional features, improved user interfaces, and optimizations based on user feedback and evolving technology trends.

Developments like integration of text-to-speech for visually impaired an be done, user analytics and post interaction prediction can be a sector inculcated into the project, also network security aspects along with various testing modules can be implemented to regulate the flow and maintenance of workspace.

Furthermore the website can be deployed in real-time for users from across the globe to access our creation and explore the vivid world of blogging with Blogify.

PYTHON CONCEPTS USED

Designing a Flask-based blogging application involves leveraging various Python concepts and libraries. Here's a list of key Python concepts commonly used in building Blogify.

VANILLA PYTHON BASED CONCEPTS

1. **Loops:**
    - Python loops have been utilized to make sure iterations over database objects is smoothly operated upon.

2. **Control Flow using Conditionals:**
    - Proper use of branching statements have made sure that only under circumstances an action would take place otherwise flow shifts to something else.

3. **Data Structures:**
    - Flask library makes use of requests which in turn acts as a JSON renderer which in turn is changed to dictionaries, similarly under many other scenarios proper use of data structures has been accounted for.

4. **Functions:**
    - Any routes generated using Flask are a result of functions which help in rendering respective templates.

5. **Object Oriented Programming:**
    - SQLAlchemy, WTForms and Flask Models all work under the basic concept of classes wherein the objects of these classes are mapped to relations in a database using ORM or act as form fields. All of these concepts fundamentally operate on the concept of OOP in python.

6. **Database:**
    - MySQL is used as a database connection with Python in this context along with relational mapping using ORM but primarily all the data is fetched using a Python-MySQL connection.

7. **Regular Expression in Python:**
    - Not a huge contributor in the success of Blogify but helps in email validation while ensuring validity of entered addresses.

8. **Exception Handling:**
    - Any sort of operation that may lead to runtime errors have been accounted for using necessary try-except blocks wherever applicable. For example while making use of database connections, if no records found to delete, the exception raised is properly handled.

9. **Virtual Environments (venv or virtualenv):**
    - Virtual environments allow isolating Python environments for different projects, preventing dependency conflicts. This application solely runs on virtual environments to cater for the same needs.

10. **Modules and Packages**
    - The entire application has been split into user-defined packages for access of information and modularization of code improving efficiency and readability of code.

FLASK BASED CONCEPTS

11. **Flask Framework:**
    - Flask is a micro web framework for Python, used to develop web applications. It facilitates routing, handling HTTP requests and responses, and structuring the application.

12. **Routing:**
    - In Flask, routing is used to map URLs to specific functions (view functions) that handle HTTP requests.

13. **Templates (Jinja2):**
    - Jinja2 is a templating engine that allows embedding dynamic content within HTML templates. It is used in his application for rendering dynamic web pages.

14. **HTTP Methods (GET, POST):**
    - Flask utilizes HTTP methods like GET and POST to handle different types of requests. GET is often used for retrieving data, while POST is used for submitting data.

15. **SQLAlchemy (or other ORM):**
    - SQLAlchemy is an Object-Relational Mapping (ORM) library for Python. It provides a way to interact with databases using Python objects, making it easier to manage database operations in Flask applications.

16. **Database Migration (Alembic):**
    - Alembic is used with Flask and SQLAlchemy to manage database migrations. It helps to version and apply changes to the database schema over time.

17. **WTForms:**
    - WTForms is a library for handling web forms in Flask applications. It simplifies form creation, validation, and rendering.

18. **User Authentication (Flask-Login):**
    - Flask-Login is an extension for managing user authentication in Flask applications. It helps handle user sessions and login functionality.

19. **Password Hashing (Werkzeug):**
    - Werkzeug is a utility library that includes password hashing functions. It's commonly used in Flask applications to securely store and verify user passwords.

20. **Middleware:**
    - Middleware components, such as those provided by the Flask framework, can be used to intercept and process requests and responses. This can include tasks like logging, security checks, etc.

21. **RESTful API Concepts:**
    - If the blogging application includes a RESTful API, concepts like resource endpoints, HTTP methods, and serialization/deserialization of data become important.
    -

22. **Testing (pytest):**
    - The pytest framework is often used for testing Flask applications. It allows developers to write unit tests and test the application's functionality.

23. **Security Best Practices:**
    - Concepts like securing against Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and other security best practices are crucial in Flask application development.

24. **Email Integration (Flask-Mail):**
    - Flask-Mail is a Flask extension for sending emails. It's often used for features like email verification or notifications in a blogging application.

25. **Environment Variables (python-dotenv):**
    - The python-dotenv library is commonly used to manage environment variables in Flask applications, ensuring sensitive information is not hard-coded. For example, in our application, all the public/private keys along with file locations have been assigned as environment variables to prevent exposure in version control softwares like GitHub.

METHODOLOGY

The methodology to create this blogging website involves series of systematic steps to ensure the successful development, launch and ongoing management of the blog:

1. DEFINING THE PURPOSE OF THE BLOG:

We clearly identified our target audience for Blogify which varies from Industry or Niche Enthusiasts to Lifestyle Enthusiasts to parenting and family.

2. MARKET RESEARCH:

Blogging these days is very popular as people want to express their views to the world and also share their opinion for a particular thing or an occasion. From fashion to make-up to amazing vacation spots along with stories and poetries, our audience will have a platform to share their opinions without any filters.

3. DESIGN AND USER EXPERIENCE:

Our design for Blogify is user-friendly and anybody can understand the working of the website with ease. Blogify offers many features like; sort by categories, sort by tags, search; these features make it easy for the user to find what they want. We also have a key feature which is anonymity, our audience can share their opinion without being discovered or judged for it.

The comment section along with like helps in motivating our users and also help them get better in case of any lacking.

4. EMAIL MARKETING:

We've set up a standalone email verification portal to confirm and send a verification email when users sign up on our website. Additionally, any special campaigns or important updates on our platform are routed through this portal to reach users. This approach not only prevents spam but also assures users of the authenticity of our emails. Moreover, we use emails to receive any feedback users may have, making it a convenient channel for communication and improvements.

IMPLEMENTATIONS

Creating a blogging website involves several components, including front-end development, back-end development, and database management. Here's a basic outline of the steps and technologies used to implement Blogify:

1. **Frontend (HTML, CSS, JavaScript)**

- *HTML Structure*

- HTML pages are utilized to render all routes, aiding in structuring the website for user accessibility and simplified styling.

- Integration of semantic HTML tags is implemented to improve both accessibility and SEO.

- *CSS Styling*

- Styling for the pages involves a combination of inline and external CSS to achieve an impressive website appearance.

- The design is responsive, seamlessly accommodating users on various devices and adapting to different preferences.

- *JavaScript*

- JavaScript is employed to integrate diverse functionalities into the website, such as modal launches and alerts.

- Client-side validation for forms has been implemented to offer users prompt feedback.

2. Backend

- *Server-Side Logic*

- Utilizing Flask as the server-side logic framework, we manage HTTP requests and responses.

- We meticulously address all edge cases in the server-side logic, incorporating version control mechanisms for effective error handling and recovery.

- *Data Storage*

- The database service employed is MySQL, serving as a storage and manipulation tool for information retrieval.

- The database is structured to facilitate the extraction of functionalities based on its attributes.

- Migration processes have been implemented to manage database version control, ensuring a structured approach for recovery purposes.

- *Template Engine*

- Jinja2 serves as the template engine, allowing the injection of Python into HTML for dynamic templating needs.

- Templates have been crafted to showcase blog posts, post previews, and the form for creating new posts.

3. Routing

Proper routing has been ensured to make sure that all the functionalities are aided by well defined routes for example for CRUD operations and even for login/register pages. Care has been taken to make sure proper handling of HTTP requests and URLs.

4. User Authentication

- User registration and login features have been incorporated, enabling multiple users to engage and interact on Blogify.
- Passwords are secured through encryption using the SHA256 algorithm to mitigate unauthorized access.
- Unauthorized access to actions has been limited, ensuring that only the rightful owner can make alterations to their assigned tasks.

5. Object Relational Mapping

- To map objects dynamically, we have made use of SQLAlchemy as ORM to map Python with Postgres.

6. Comments and Interactivity

- Interactivity with blog posts is facilitated through user actions such as liking and commenting, fostering constructive benefits and the exchange of information.
- Social media integration is offered, allowing users to share posts on external platforms.
- A subscription feature has been implemented, enabling users to subscribe to their favorite bloggers. Subscribers receive email notifications, keeping them informed about new posts and ensuring they never miss any updates.

7. SEO Considerations

- Optimized website for search engines by including relevant metadata.
- Clean and descriptive URLs have been used for blog posts.
- Sitemap has been created for search engines to optimize and index website.

8. Security

- To safeguard against SQL injection or Cross-Site Scripting (XSS) attacks, user inputs undergo thorough sanitization.
- Secure methods for password storage have been implemented, particularly when managing user authentication.
- Data in transit is encrypted through the use of HTTPS.
- Email verifications are fortified with OTP services to deter automated bot spam accounts.
- Comprehensive measures have been adopted to prevent CSRF and DDoS attacks on the website.

9. Blogging-Specific Concepts

- *Categories and Tags*
  - Users allowed to categorize and tag their posts for better organization and searchability.

- *RSS Feeds*
  - RSS feed for users to subscribe to updates.

- *Author Profiles*
  - Authors authorized to have profiles with information about them and links to their posts.

- *Editing Features*
  - Users are given a feature to edit their posts and interaction along with profile.

- *Post Search*
  - Users are enabled to search for posts based on any keywords.

- *Blogger Subscriptions*
  - Blogger subscription feature has been implemented, enabling users to subscribe to their favorite bloggers. Subscribers receive email notifications, keeping them informed about new posts and ensuring they never miss any updates.

- *Post Filtering*
  - Posts can be filtered according to various timelines, based on past week, month or year. This helps in keeping the users up-to-date with all the blogs.

10. Analytic Tools

- *ReCaptcha Integration*
  - Integrated ReCaptcha and Google Analytics to prevent automated traffic and also provide a deeper insight into logging and network details.

RESULT

I. AUDIENCE ENGAGEMENT

Our blogging website will attract and engage a targeted audience. This includes regular visitors who read, comment on, and share the blog posts. High engagement indicates that a specific post or content is resonating with the audience.

II. TRAFFIC GROWTH

Increasing the number of visitors to the blog over time. This can be achieved through various means, such as search engine optimization(SEO), social media promotions, and email marketing in the deployment phase of our website.

III. CONTENT QUALITY

Bloggers can often aim to create high-quality, informative, and valuable content that resonates with their readers. This can help establish authority and credibility in a particular niche.

IV. BRAND BUILDING

Blogging can be a powerful tool for building a personal or business brand. The expected result may be increased brand awareness, reputation, and trust among the target audience after Blogify gets deployed.

V. FEEDBACK AND IMPROVEMENT

Bloggers often seek feedback from their audience to improve their content and website. The expected result may be a continuous cycle of improvement based on feedback.

CONCLUSION

In conclusion, the Flask-based blogging application, Blogify, has been successfully developed and implemented, showcasing a user-friendly interface, essential blogging functionalities, and a responsive design for multi-device accessibility. Security measures, scalability considerations, and performance optimizations contribute to a reliable and secure environment. The project emphasizes testing, documentation, and code quality, while future development opportunities and community involvement remain pivotal for ongoing success. Blogify stands as a testament to the versatility of Flask in creating a feature-rich and robust web application, with thanks extended to the development team and stakeholders for their contributions to the success of this project.

REFERENCES

Building Blogify has been an eventful journey with the help of a lot of online resources along the way to innovate and refer to. The entire comprehensive GitHub repository for this project can be found at:

- https://github.com/RampageousRJ/Blogify

Some of the websites referred to for building this application are:

- https://stackoverflow.com/
- https://flask.palletsprojects.com/en/3.0.x/
- https://docs.python.org/3/
- https://www.youtube.com/@Codemycom
- https://git-scm.com/
- https://www.pythonanywhere.com/