**Backend Project Report**

**1. Introduction and Objectives**

This project is a full-stack blog application built with Flask. Its main objective is to demonstrate the end-to-end development of a dynamic, interactive, secure, and data-driven web application, including user authentication, CRUD operations, file uploads, session management, modular coding with Blueprints, and Object-Oriented Programming (OOP) principles.

**2. Description of Features**

* User registration and login with secure password hashing.
* Profile page with avatar upload and profile editing.
* Creating, viewing, and deleting blog posts.
* Image uploads for posts with secure file handling.
* Responsive user interface with Bootstrap 5.
* Flash messaging for feedback.
* Only post creators can delete their posts.
* Session-based authentication.

**3. ER**

**Entities and Relationships:**

* **User**
  + id (Primary Key)
  + username (String)
  + email (String)
  + password\_hash (String)
  + avatar (String)
* **Post**
  + id (Primary Key)
  + title (String)
  + content (Text)
  + image\_filename (String)
  + user\_id (Foreign Key referencing User.id)

**Relationship:**

* One User can have many Posts.

**4. Explanation of Code Structure (OOP & Blueprints)**

* **OOP Usage:**
  + Models are defined as Python classes (User, Post) using SQLAlchemy ORM.
  + Forms (RegistrationForm, LoginForm, PostForm, EditProfileForm) use WTForms.
* **Blueprints:**
  + auth\_bp handles authentication (register, login, logout, profile, edit profile).
  + posts\_bp handles blog post operations (create, view, delete posts).
* **Separation of Concerns:**
  + models.py for database models.
  + forms.py for form structures.
  + routes/auth.py and routes/posts.py for route handlers.
  + templates/ for HTML templates.
  + static/uploads/ for user-uploaded files.

**5. Screenshots of Key Pages**

* Registration Page

Изображение выглядит как снимок экрана, текст, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

* Login Page

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

* Dashboard with Posts

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Контент, сгенерированный ИИ, может содержать ошибки.

* View Post Page

Изображение выглядит как текст, снимок экрана, Веб-сайт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

* Profile Page

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

* Edit Profile Page

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**6. Challenges Faced and Solutions**

| **Challenge** | **Solution** |
| --- | --- |
| File upload errors | Implemented file type validation and used secure filenames with secure\_filename. |
| Session updates after avatar change | Updated session variables immediately after saving the new avatar. |
| Restricting delete access | Ensured only the post creator can delete their posts by checking user\_id against session['user\_id']. |
| Layout issues on small screens | Used Bootstrap responsive grid and object-fit: cover for consistent image sizing. |
| Maintaining text formatting | Used white-space: pre-line and text-align: justify for post content. |