

Mini Flask Blog Application - Full Documentation

This documentation provides a detailed explanation of the codebase for the Mini Blog Application built with Flask. It covers the folder structure, each Python file, the authentication system, templates, static files, and database models. This guide is meant for developers and team members who want to understand how the application works and extend it further.

Folder Structure:

- flask_blog/
- app.py (main application entry point)
- models.py (database models)
- forms.py (WTForms definitions)
- auth.py (authentication routes)
- blog.py (blog post routes)
- __init__.py (application factory and setup)
- static/
- style.css (custom CSS for styling)
- templates/
- base.html (main base template with navbar)
- index.html (home page listing posts)
- login.html (login form)
- register.html (registration form)
- create_post.html (form for creating/editing posts)
- post_detail.html (view individual post)

app.py

- Entry point of the application. - Creates Flask app, initializes database and blueprints. - Runs the app when executed directly.

models.py

- Contains SQLAlchemy models. - User model: handles authentication details (username, email, password). - Post model: stores blog posts with title, content, timestamp, and author relation.

forms.py

- Contains WTForms classes. - RegistrationForm: for new users to sign up. - LoginForm: for logging into the app. - PostForm: for creating and editing posts.

auth.py

- Handles authentication (register, login, logout). - Passwords are hashed using Werkzeug. - User session management with Flask-Login.

blog.py

- Handles blog features (create, read, update, delete posts). - Uses PostForm for forms. - Ensures only logged-in users can manage posts.

templates/

- base.html: shared layout with navbar and Bootstrap setup. - index.html: displays all posts. - login.html: login form UI. - register.html: registration form UI. - create_post.html: form for new/edit post. - post_detail.html: view single post in detail.

static/style.css

- Contains custom styles for the blog. - Styles navbar, buttons, forms, and layout.

Authentication Features

Authentication Features: - Registration: Users create an account with username, email, and password. - Login: Users log in with email and password. - Logout: Ends user session. - Access Control: Only logged-in users can create, edit, or delete posts. - Password Security: Passwords stored as secure hashes using Werkzeug.