**Web Application Documentation**

**Architecture:**

The web application follows a simple Model-View-Controller (MVC) architecture.

**Model** (models/user\_model.py):

* Defines the User model class with SQLAlchemy for database interaction.
* Uses Flask-Bcrypt for secure password hashing.

**View** (templates):

* Contains HTML templates for different pages (index.html, register.html, dashboard.html, password\_reset.html).
* Utilizes CSS for basic styling, including responsive design and color-coded input validation.

**Controller** (controllers/auth\_controller.py):

* Implements routes for various functionalities such as user registration, login, password reset, and dashboard.
* Integrates with the User model to interact with the database.
* Uses Flask session for user authentication.

**Design Choices:**

User Registration:

Implemented with a registration form in register.html.

Uses email as the username for simplicity.

Password and confirm password fields with client-side and server-side validation.

Login:

Login form in index.html.

Validates user credentials against stored hashed passwords.

Displays error messages for unsuccessful login attempts.

Dashboard:

Accessible only after successful login.

Placeholder dashboard.html for future content.

Password Reset:

Password reset functionality in password\_reset.html.

Requires entering the username (email) and setting a new password.

**Security Measures:**

Password Hashing:

Utilizes Flask-Bcrypt to securely hash user passwords before storing in the database.

Session Management:

Uses Flask session for user authentication.

Sessions are secure with options like 'SESSION\_COOKIE\_SECURE'.

HTTPS and Content Security Policy (CSP):

Enforces HTTPS using Flask SSLify extension.

Implements Content Security Policy using Flask-Talisman to mitigate XSS attacks.

Input Validation:

Client-side validation for email format and password length.

Server-side validation to ensure data integrity and security.

**Usage Instructions:**

Installation:

Install required packages using pip install flask flask-bcrypt flask\_sqlalchemy flask\_sslify flask-talisman.

Run the Application:

Execute python app.py to run the Flask development server.

Access the application at http://localhost:5000.

User Registration:

Navigate to the registration page (/register).

Enter a valid email and password.

Click the "Register" button.

Login:

Visit the login page (/ or /login).

Enter the registered email and password.

Click the "Login" button.

Password Reset:

Access the password reset page (/reset\_password).

Provide the username (email) and set a new password.

Dashboard:

After successful login, access the dashboard (/dashboard).

Logout:

Logout by accessing the /logout route.