Two-Factor Authentication (2FA) System

Project Created By : MADHAN K
Institution : TECH VEDHU
GitHub Project Link : https://github.com/Madhan-Cs-Study/Madhan.git

Project Overview

This project implements a secure Two-Factor Authentication (2FA) system using Python Flask, SQLite, and Email-based OTP verification. The goal of this project is to improve login security by adding an extra verification step after username and password authentication

Technologies Used

1 Python (Flask Framework)

2 SQLite Database

3 HTML & CSS (Frontend)

4 SMTP (Gmail) for OTP Email Service

5 VS Code (Development Environment)

Project Workflow

- User opens the application and accesses the login page.
- User enters username and password.
- Server verifies credentials from SQLite database.
- If credentials are valid, a random OTP is generated.
- OTP is sent to the registered email using SMTP.
- User enters the OTP on the verification page.
- System verifies OTP within a time limit.
- On success, user is redirected to the dashboard.

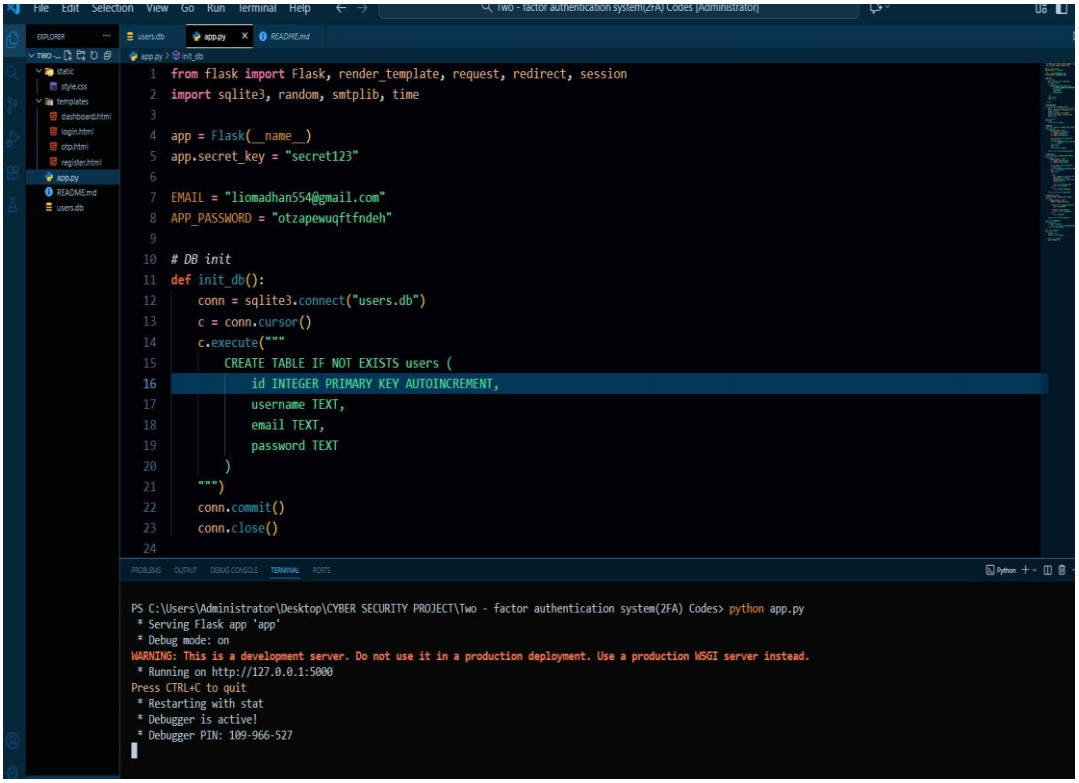
Security Features

- Two-step authentication (Password + OTP)
- Time-based OTP validation
- Session management using Flask
- Email verification ensures authorized access

Stage 1: Project Startup

The Flask application is started using the command python app.py. Once executed, the server runs locally on http://127.0.0.1:5000. This confirms that the backend server is active and ready to handle requests.

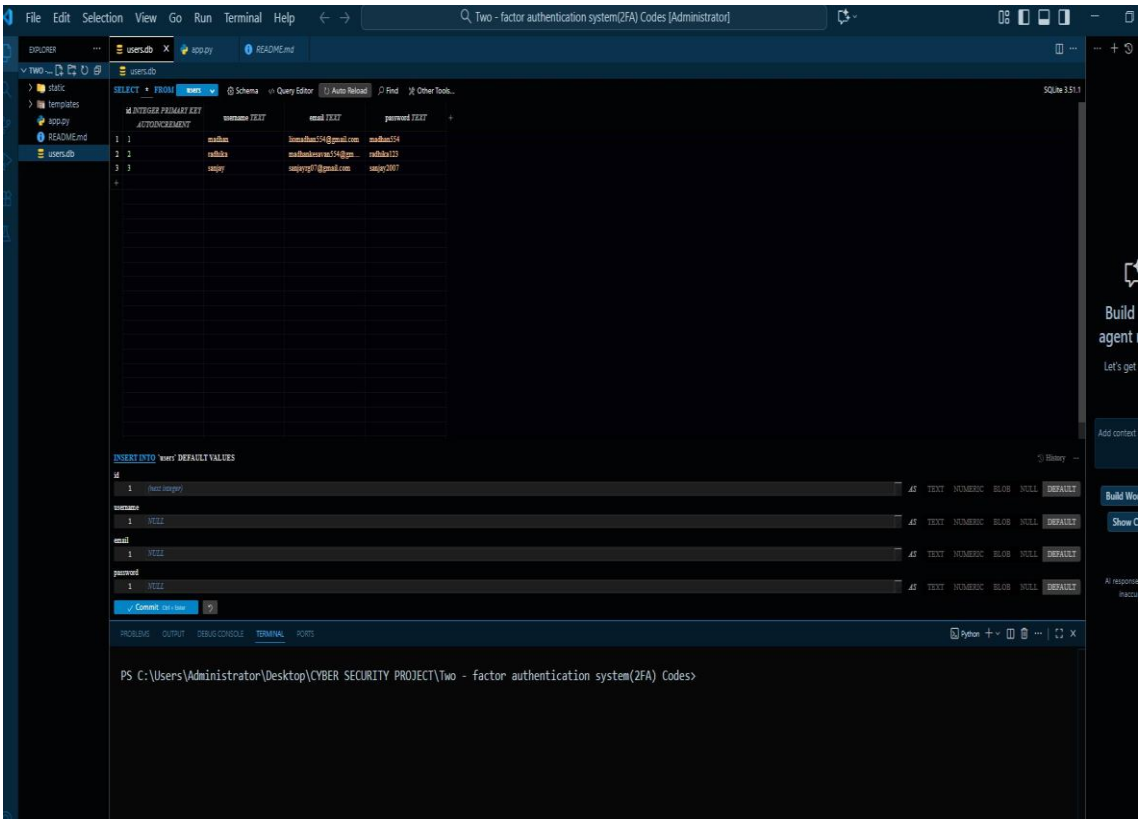
- The Flask web application is started using the command `python app.py`.
- This command runs the backend server in development mode.
- Once executed, Flask initializes all routes, database connections, and configurations.
- The server runs locally at `http://127.0.0.1:5000`.
- This local address allows users to access the application through a web browser.
- The backend handles user login, registration, and OTP verification processes.
- Flask manages communication between the frontend pages and the database.
- Successful server startup confirms the application is ready to handle user requests.



Stage 2: Database Creation

SQLite database is used to store user details such as username, email, and password. The users table is automatically created when the application starts. This database stores registered users securely for authentication.

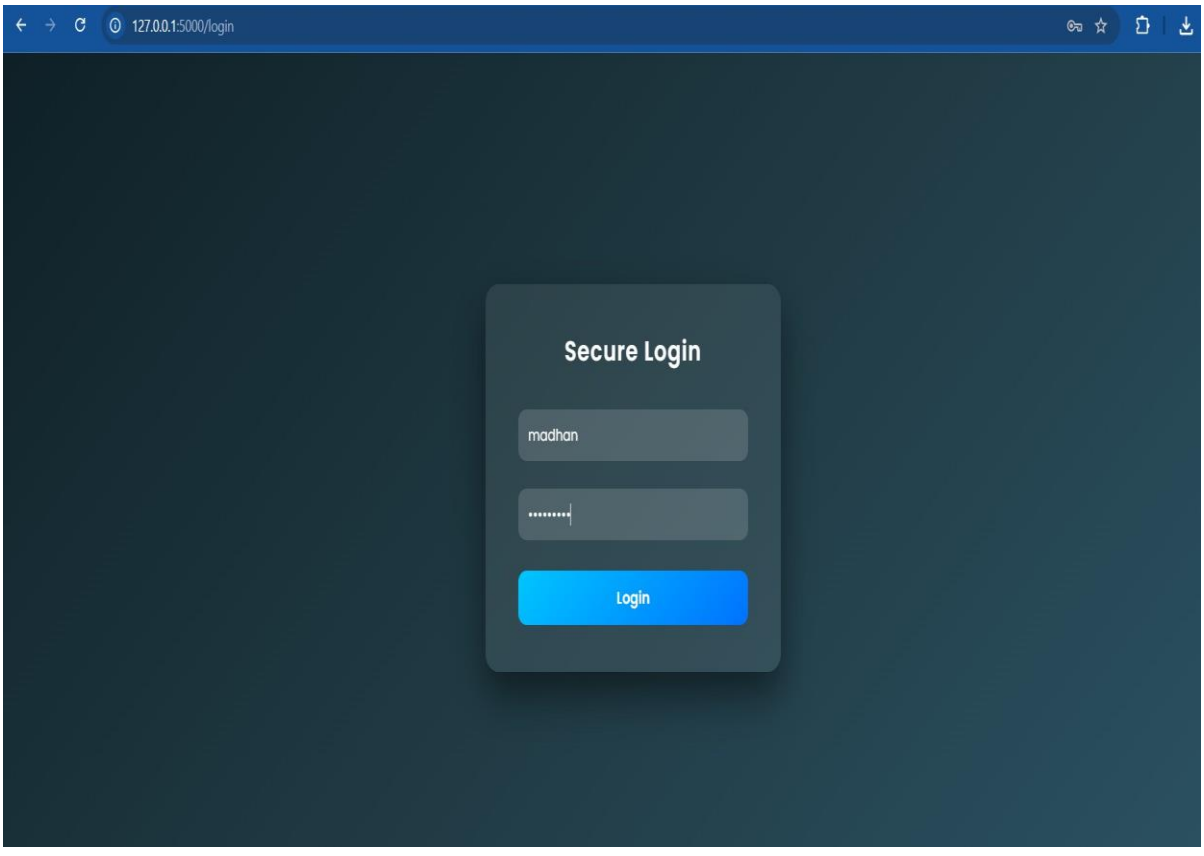
- SQLite is used as the database to store user registration details.
- The database stores important information such as username, email, and password.
- A `users` table is created to organize and manage user data.
- This table is automatically created when the application starts.
- The database ensures persistent storage of registered user information.
- Stored user data is used during login and authentication processes.
- SQLite provides a lightweight and efficient solution for small web applications.
- This stage ensures secure and structured data management for the system.



Stage 3: User Login

The user enters their username and password on the secure login page. The system validates these credentials against the database. If correct, the user is redirected to OTP verification.

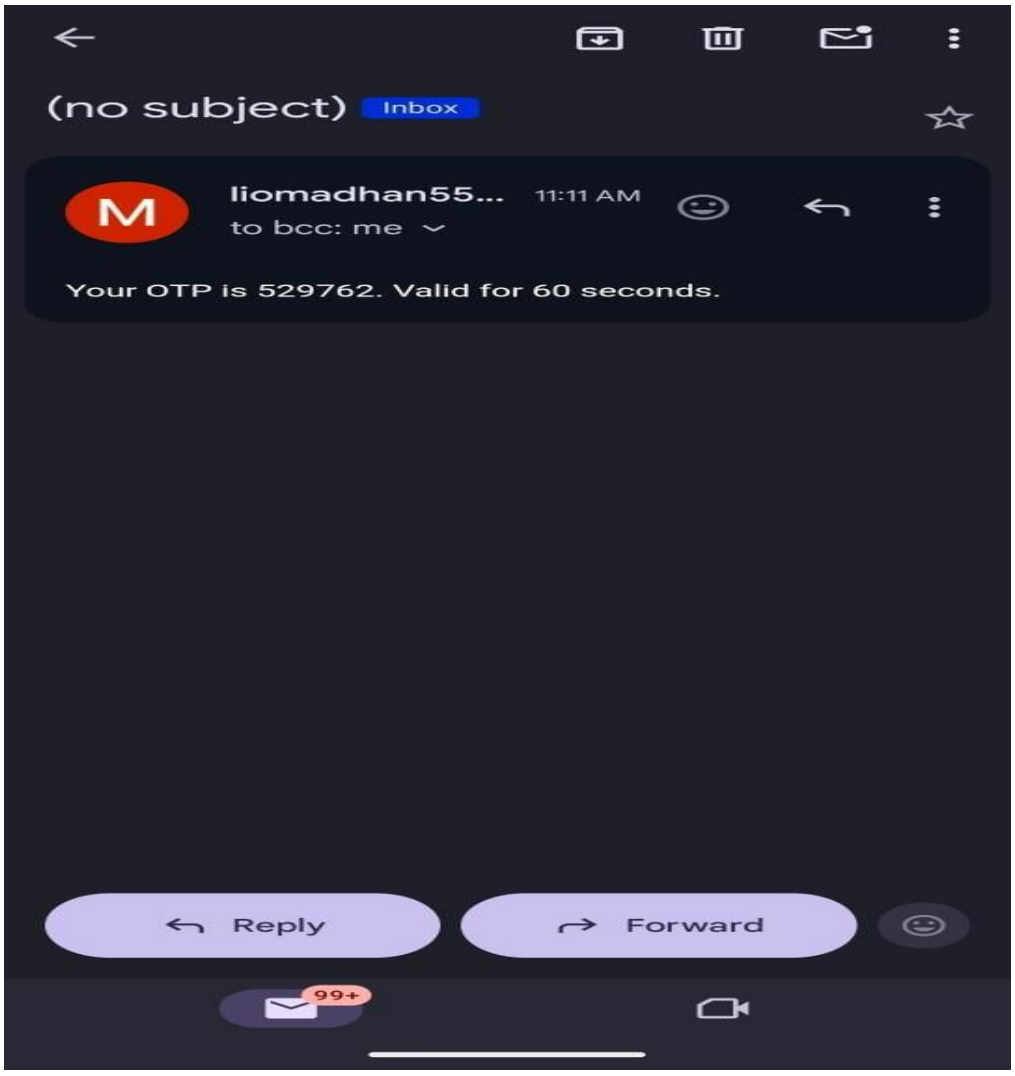
- The user accesses the secure login page of the application.
-
- The user enters their registered username and password.
- These credentials are sent to the backend for verification.
- The system checks the entered details against the database records.
- If the credentials are valid, the login process is approved.
- Invalid credentials result in an error message to the user.
- Upon successful login, the system initiates the second authentication step.
- The user is redirected to the OTP verification page.
- Session management is used to track the logged-in user.
- This stage ensures only authorized users proceed to OTP verification.



Stage 4: OTP Verification

After successful login, a One-Time Password (OTP) is generated. The OTP is sent to the registered email address of the user. The OTP is valid for 60 seconds to enhance security.

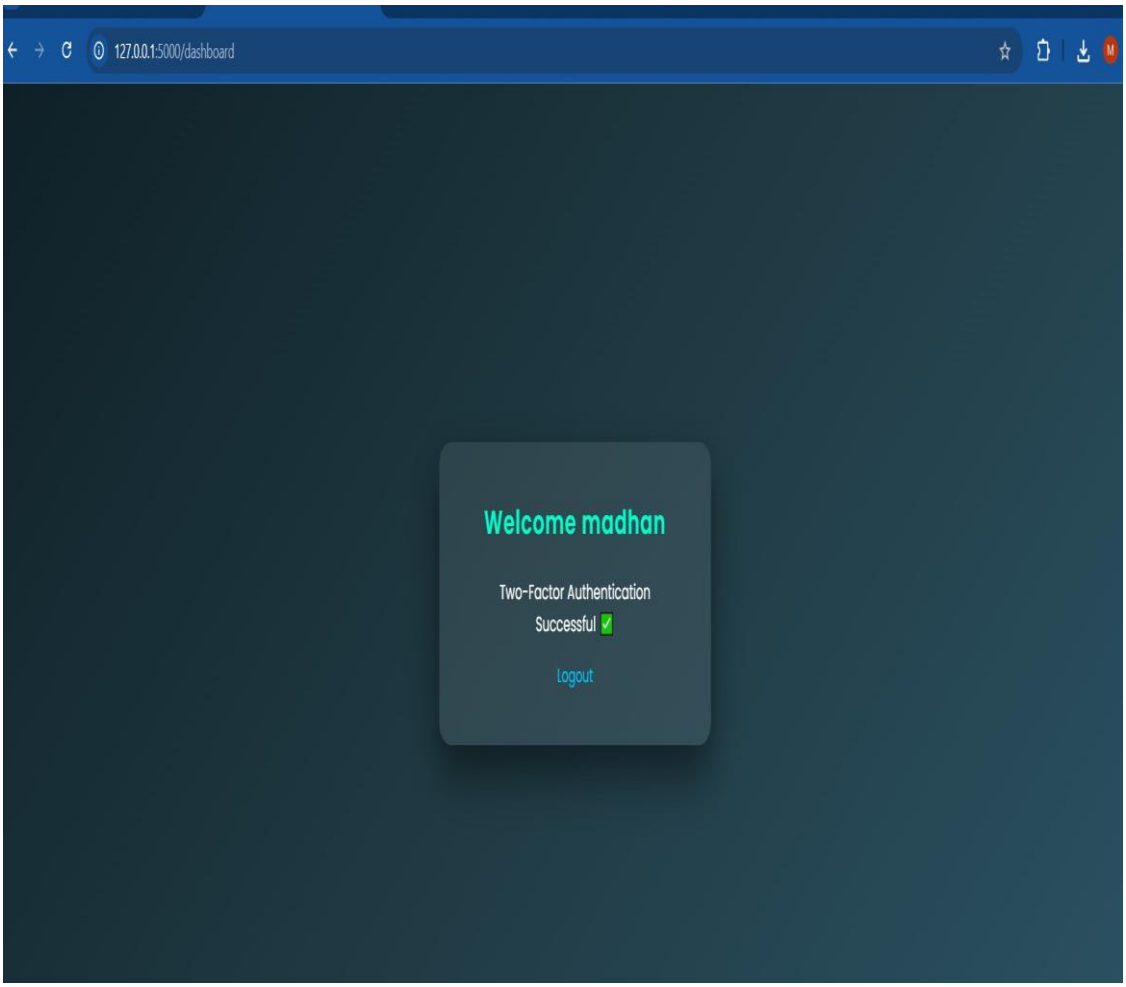
- The system generates a random One-Time Password (OTP).
- This OTP is unique for each login session.
- The generated OTP is sent to the user’s registered email address.
- Email delivery ensures only the legitimate user receives the OTP.
- The user is prompted to enter the received OTP on the verification page.
- The OTP is valid only for 60 seconds to improve security.
- If the OTP expires, the user must request a new one.
- The system verifies the entered OTP with the generated value.
- Successful OTP validation completes the authentication proces



Stage 5: Successful Authentication

When the correct OTP is entered, authentication is completed successfully. The user is redirected to the dashboard page. This confirms that Two-Factor Authentication is working correctly.

- After entering the correct OTP, the system verifies it successfully.
- OTP validation confirms the user's identity.
- Both login credentials and OTP are now authenticated.
- This completes the Two-Factor Authentication process.
- The system marks the user as fully authenticated.
- The user is redirected to the dashboard page.
- The dashboard displays authorized user content.
- Unauthorized access is prevented at this stage.
- Session security is maintained throughout the process.
- This stage confirms that the 2FA system is working correctly.



Final Result

This project successfully demonstrates a secure Two-Factor Authentication system. It prevents unauthorized access even if passwords are compromised. The project enhances application security using real-time OTP verification.

1. his project successfully demonstrates a secure Two-Factor Authentication (2FA) system.
2. It uses both password authentication and OTP verification.
3. Unauthorized access is prevented even if passwords are compromised.
4. The OTP provides an additional layer of security.
5. OTPs are generated dynamically for each login session.
6. Real-time OTP verification improves authentication reliability.
7. OTPs are sent only to registered user email addresses.
8. Expired or incorrect OTPs are automatically rejected.
9. The system follows secure authentication best practices.
10. Overall, the project significantly enhances application security.