Secure API and JWT Authentication Project

Overview

This project demonstrates a secure API implemented using Flask (Python) that provides user registration, login, and profile access with JWT (JSON Web Tokens) based authentication. Passwords are securely hashed, and JWT tokens protect sensitive routes. The frontend uses HTML templates rendered by Flask.

Features

- User registration with password hashing (SHA256)
- User login with JWT token issuance stored in server session
- JWT-protected profile page accessible only by authenticated users
- Logout functionality clearing user session
- Simple UI with Flask-rendered HTML pages

Technologies Used

- Python 3.x
- Flask
- PyJWT
- HTML/CSS frontend templates

Project Structure

project_folder/ secure_api.py # Flask backend code templates/ # HTML templates home.html login.html profile.html register.html

---

Setup and Installation

1. **Install Python 3.x** if not already installed.
2. **Install dependencies:**

pip install flask pyjwt

3. **Ensure folder structure is as above, with templates inside `templates` directory.**
4. **Run the Flask application:**

python secure_api.py

Server starts at: `http://127.0.0.1:5000/`

Usage Instructions

- Visit `http://127.0.0.1:5000/` to access home page.
- Use "Register" to create new users (username & password).
- Login with valid credentials to receive a JWT stored in session.
- Access the protected "Profile" page only accessible with a valid token.
- Use logout to clear session.

Important Notes

- User data is stored only in memory (`users` dictionary) - data is lost after server restart.
- Password hash uses SHA-256 (consider stronger hashing for production).
- JWT tokens expire after 30 minutes.
- Session handles storing JWT token securely for the user.

Routes and Endpoints

Route	Method	Description	Protected
`/`	GET	Home page	No
`/register`	GET/POST	User registration page & submit	No
`/login`	GET/POST	User login page & submit	No
`/profile`	GET	Shows profile if authenticated	Yes
`/logout`	GET	Clears session and redirects home	No

Security Measures

- Passwords hashed before storing.
- JWT tokens signed with secret key and include expiry.
- Protected routes require verified JWT token.
- Session management for storing authentication token.
