README

This program consists of four files: blog, instance, orders.txt, run.py.

Blog: Contains the following files: main, static, templates, init.py, and models.py. The main file stores: .env, forms.py, routes.py, settings.py, tests.py. .env is a configuration file for the application that uses Flask and SQLAlchemy to work with the database. forms.py - This file contains a description of the login form for the Flask application. The form includes the following fields: email, password, remember, submit. routes.py - This file contains routes and functions for the Flask application related to the blog. Key functions and routes include: index(), login(), account(), logout(), restaurant(restaurant_name), order_menu_item(menu_item_id). This file also includes other necessary imports and settings for the proper operation of the Flask application, including database operations, form handling, and user authentication. settings.py - This file loads configuration settings for the application from the .env file, including the secret key and the SQLAlchemy database connection string. tests.py - This file contains a set of tests to check the functionality of the blog application. The tests check the creation of users, restaurants, menu items, page access, and the ability to create orders.

The static and templates files store HTML and CSS files. I use them to demonstrate the results in the form of a simple UI design.

__init__.py contains functions for creating test data in the database for our blog application: create_users(), create_restaurants(), create_menu(). These functions help initialize our database with test data that can be used for testing and developing your application.

models.py creates the database structure for managing restaurants, menus, and orders. The main classes and functions include: User, Restaurant, Menu, Order, load_user. This database structure helps us store and manage data about users, restaurants, menus, and orders in your web application.

The instance folder stores our database (test1), which contains four tables.

order.txt - a file that will store orders for restaurants and who they are from. This file will be cleared and populated with new data daily.

run.py - This code creates and runs the Flask application for managing our blog, helps initialize our database with test data, and schedules regular clearing of the 'orders.txt' file. Key actions include: schedule_clear_orders(), create_users(), create_restaurants(), create_menu(). Run schedule_clear_orders() in a separate thread for regular file clearing.

Additional Information: The database already contains information about company users, so anyone won't be able to log into the system. To test the program's functionality, you can use one of the authorized users:

{'username': 'Mike', 'email': 'mike_admin@email.com', 'password': '1234'},

{'username': 'John', 'email': 'john_user@email.com', 'password': '5678'},

{'username': 'Alice', 'email': 'alice_user@email.com', 'password': 'abcd'},

{'username': 'Tom', 'email': 'tom_user@email.com', 'password': '4444'},

{'username': 'Io', 'email': 'io_user@email.com', 'password': 'ab12'},