Authentication Website with Python

1. Install a framework python called flask:
2. What is flask and how to install it.

Flask is a lightweight, open-source web framework for Python used to build web applications. It is classified as a microframework because it keeps things simple and minimal, providing the essential tools to create web apps without imposing strict rules or dependencies.

How to install flask: pip install flask

Notes: It installs all collected packages: blinker-1.9.0 , click-8.2.1 , colorama-0.4.6 , flask-3.1.1 , itsdangerous-2.2.0 , jinja2-3.1.6 , markupsafe-3.0.2 , werkzeug-3.1.3

1. There are popular extensions for the Flask web framework in Python, designed to add specific functionality to my web applications.

* pip install flask-login

Flask-Login is a Flask extension that simplifies user authentication and session management in web applications. It handles tasks related to logging users in, logging them out, and managing user sessions securely.

Notes: Successfully installed flask-login-0.6.3

* pip install flask-sqlalchemy

Flask-SQLAlchemy is a Flask extension that simplifies integration with SQLAlchemy, a powerful Python library for working with relational databases (e.g., SQLite, PostgreSQL, MySQL). It provides an easy-to-use interface for defining database models, querying data, and managing database connections in Flask applications.

Notes: Successfully installed flask-sqlalchemy-3.1.1 greenlet-3.2.3 sqlalchemy-2.0.41 typing-extensions-4.14.0

1. Create some folder and file to begin the project:

Authentication Website with Python ( Main folder ) contains :

1. Read\_Me folder contains ReadMe file that contains all notes of this project
2. Main.py file that I called all function I do
3. website folder contains :

* static folder
* templates folder

I create many files to organize my code as base, home, login, sign\_up et logout

* \_\_init\_\_.py file contains:
* In this file, I import the class Flask from flask and I create an instance of it
* The function create\_app: permits to create an app configurate a key, call views or auth and return this app to be used elsewhere in the code.
* auth.py file
* In this file, I import the class Blueprint from flask and I create an instance of it
* I create a view
* I import the class render\_template from flask and I use it in my view to show what I did in frontend
* I import request from flask to recycle all information from the input of the user, also I create some loops to verify if the information the user has put is well or not .
* models.py file
* view.py file
* In this file, I import the class Blueprint from flask and I create an instance of it
* I create a path
* I import the class render\_template from flask and I use it in my view to show what I did in frontend