

Libraries and Modules Used

1. **Flask:**

A lightweight WSGI web application framework in Python. It provides tools and libraries to build web applications quickly and easily, allowing you to create routes, handle requests, and render HTML templates.

2. **render_template:**

A function from Flask that helps in rendering HTML templates. It allows you to separate your application logic from presentation by using templates with Jinja2.

3. **request:**

An object in Flask that contains all the information about the incoming HTTP request, allowing you to access form data, query parameters, and other request-related information.

4. **redirect:**

A Flask function used to redirect the user to a different endpoint or URL. It is commonly used after form submissions to prevent duplicate submissions.

5. **session:**

A Flask object that allows you to store information about the user across multiple requests. It can be used for session management, like keeping track of logged-in users.

6. **url_for:**

A Flask function that generates URLs for the specified function. It helps in creating links that are dynamic and can adapt if the route changes.

7. **flash:**

A function in Flask used to display messages to users. It is typically used to show success or error messages after a form submission.

8. **jsonify:**

A Flask function that converts Python dictionaries into JSON responses. It is useful for building APIs or returning data in a structured format.

9. **pymysql:**

A library used to connect and interact with MySQL databases. It allows you to execute SQL queries and manage your database.

10. **sqlite3:**

A built-in Python library for interacting with SQLite databases. It is lightweight and used for simple database management tasks.

11. MySQLdb:

A Python interface for MySQL that allows you to connect and execute SQL queries on MySQL databases.

12. mysql.connector:

A MySQL driver that allows Python applications to connect to a MySQL database using the MySQL Connector/Python API.

13. werkzeug.security:

A module that provides utilities for securely hashing passwords and verifying hashed passwords, enhancing the security of user authentication.

14. Flask-Mail:

An extension for Flask that provides easy email sending capabilities. It supports various email backends and simplifies the process of sending emails from your application.

15. uuid:

A built-in Python module for generating universally unique identifiers (UUIDs). It is often used for creating unique tokens or identifiers.

16. random:

A built-in Python module that implements pseudo-random number generators for various distributions. It can be used for generating random numbers, choices, and more.

17. logging:

A built-in Python module that provides a flexible framework for emitting log messages from Python programs. It can be used for tracking events and debugging.

18. smtplib:

A built-in Python module for sending emails using the Simple Mail Transfer Protocol (SMTP). It is often used in conjunction with the email library to send email messages.

19. datetime:

A built-in Python module for manipulating dates and times. It provides classes for working with dates and times, including formatting and arithmetic.

20. email.mime.multipart:

A module from the email library that helps create email messages with multiple parts, such as plain text and HTML.

21. email.mime.text:

A module that provides functionality to create text-based email messages, enabling you to send formatted emails.

22. **APScheduler:** A library that allows you to schedule tasks to run at specific intervals or times. It is useful for automating background jobs in web applications.