ASSIGNMENT-3

1. What is Flask, and how does it differ from other web frameworks?

Flask is a micro web framework for Python, designed to be lightweight and flexible. It differs from other web frameworks by its minimalistic approach, allowing developers to have more control over the components they use and how they structure their applications.

2. Describe the basic structure of a Flask application?

A basic Flask application typically consists of a Python script that defines the application instance, routes, and configuration. It also includes templates for HTML rendering and possibly static files like CSS and JavaScript.

3. How do you install Flask and set up a Flask project?

To install Flask, you can use pip, Python's package manager, by running pip install Flask. After installation, you can set up a Flask project by creating a directory for your project, creating a virtual environment, and then creating a Python script to define your Flask application.

4. Explain the concept of routing in Flask and how it maps URLs to Python functions?

Routing in Flask involves mapping URLs to Python functions. Developers use decorators like @app.route('/some/url') to define routes in their Flask applications. When a user navigates to a specific URL, Flask invokes the corresponding Python function to handle the request.

5. What is a template in Flask, and how is it used to generate dynamic HTML content?

In Flask, a template is an HTML file that includes placeholders for dynamic content. These placeholders are typically defined using Jinja templating syntax, allowing Flask to inject dynamic data into the HTML before rendering it to the user.

6. Describe how to pass variables from Flask routes to templates for rendering?

To pass variables from Flask routes to templates for rendering, developers can simply pass the variables as arguments to the render_template() function. Within the template, these variables can then be accessed using Jinja syntax.

7. How do you retrieve form data submitted by users in a Flask application?

Form data submitted by users in a Flask application can be retrieved using the request object provided by Flask. Developers can access form data using request.form['field_name'] for POST requests or request.args['field_name'] for GET requests.

8. What are Jinja templates, and what advantages do they offer over traditional HTML?

Jinja templates are a powerful feature of Flask that allow for dynamic content generation in HTML files. They offer advantages over traditional HTML by providing features like template inheritance, macros, and filters, which make it easier to maintain and reuse code.

9. Explain the process of fetching values from templates in Flask and performing arithmetic calculations.

To fetch values from templates in Flask and perform arithmetic calculations, developers can pass variables containing numeric values from the Flask routes to the templates. Within the templates, these variables can then be used in expressions to perform calculations using Jinja syntax.

10. Discuss some best practices for organizing and structuring a Flask project to maintain scalability

readablity

Some best practices for organizing and structuring a Flask project include separating concerns by using a modular structure with blueprints for different parts of the application, using a consistent naming convention for routes and templates, and leveraging Flask extensions for common functionality like database integration and authentication. Additionally, using configuration files to manage environment-specific settings and version control systems for collaboration and versioning can improve scalability and readability of Flask projects.