

Assignment -3

Name : Cheruku Rahul
College : Sri Venkateswara Engineering College.
Roll No/Hall Ticket : 209E1A3313

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

- Flask is a micro web framework for Python.
- It differs from other frameworks by being lightweight and flexible, allowing developers to choose components rather than providing everything out of the box.

2. Describe the basic structure of a Flask application.

- Application setup involves importing Flask and creating an instance.
- Routes map URL patterns to Python functions.
- Views render templates or perform other tasks.
- Templates contain HTML with placeholders for dynamic content.
- Static files include CSS, JavaScript, etc.
- Configuration holds settings for the Flask application.

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

- Install Flask using pip: ``pip install Flask``
- Create a directory for the project and navigate into it.
- Create a Python script (e.g., ``app.py``) for the Flask application.
- Write the application code in ``app.py``.

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

- Routing maps URL patterns to Python functions using the ``@app.route()`` decorator.
- When a request is made to a URL, Flask invokes the corresponding Python function.

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

- Templates are HTML files with placeholders for dynamic content.
- They are rendered using the Jinja2 templating engine.
- Templates enable the generation of dynamic HTML content based on data provided by the Flask application.

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

- Pass variables as arguments to the ``render_template()`` function.

- In the template, use double curly braces `{{ variable_name }}` to display the variable's value.

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

- Use the `request` object to access incoming request data, including form data.
- For example, `request.form['key']` retrieves form data with the key `"key"`.

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

- Jinja templates are HTML files with placeholders for dynamic content.
- They offer advantages like template inheritance, variable interpolation, control structures, and filters over traditional HTML.

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

- Pass variables from routes to templates.
- Use Jinja syntax to perform calculations in the template, e.g., `{{ num1 + num2 }}`.

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

- Modularize the application into separate files.
- Use blueprints for related routes and views.
- Implement a configuration system.
- Keep business logic separate from presentation logic.
- Use Flask extensions for common tasks.
- Follow RESTful design principles for API endpoints.
- Document code and adhere to consistent naming conventions.