ASSIGNMENT-3

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

ANS): Flask is a lightweight and flexible web framework for Python. It differs from other frameworks like Django by being more minimalistic, giving developers more freedom to choose components and libraries according to their needs.

2. Describe the basic structure of a Flask application.

ANS): A basic Flask application typically consists of a single Python file containing routes, templates directory for HTML files, and optionally static files like CSS and JavaScript. It follows the WSGI (Web Server Gateway Interface) protocol.

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

ANS): You can install Flask using pip: pip install Flask. To set up a Flask project, create a directory, and within it, create a Python file for your application logic. You can also create directories for templates and static files if needed.

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

ANS): Routing in Flask is done using decorators. You define routes with @app.route('/url'), where /url is the URL endpoint. When a

request matches this URL, Flask calls the associated Python function, which returns the response.

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

ANS): A template in Flask is an HTML file with placeholders for dynamic content. Flask uses Jinja2 templating engine to render these templates. Python variables and expressions can be passed to the template for dynamic content generation.

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

ANS): You can pass variables from Flask routes to templates by including them as arguments when rendering the template using the render_template() function. For example: render_template('template.html', variable_name=variable).

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

ANS): Form data submitted by users can be retrieved in Flask using the request object. You can access form data using request.form['fieldname'] or request.form.get('fieldname').

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

ANS): Jinja templates are a templating engine for Python, used in Flask for generating dynamic HTML content. They offer advantages

over traditional HTML by allowing for dynamic content insertion, logic implementation, and template inheritance, enhancing code reusability and maintainability.

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

ANS): Values from templates in Flask are fetched using Jinja templating syntax, such as {{ variable }}. Arithmetic calculations can be performed directly within the template using Jinja expressions like {{ variable1 + variable2 }}.

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

ANS): - Use the blueprint pattern to organize routes into logical components.

- Separate concerns by dividing code into modules for models, views, and forms.

- Utilize Flask extensions for commonly needed functionalities.

- Implement error handling and logging for better debugging.

- Use virtual environments to manage dependencies and version control.

- Document code and adhere to PEP 8 style guide for readability.

- Consider using a factory pattern for creating the Flask application instance to facilitate testing and configuration management.