

Python Web Development Assignment using Flask

1. What is Flask and what are its main features?
2. How does Flask differ from other Python web frameworks like Django?
3. What is routing in Flask and why is it important?
4. How do you define a route in Flask? Provide an example.
5. What is the purpose of the `render_template()` function in Flask?
6. Explain how Jinja2 templates work in Flask applications.
7. What are template inheritance and blocks in Jinja2?
8. Describe how to handle form data submission in Flask.
9. What is the difference between GET and POST HTTP methods?
10. How can you access form data sent via POST in a Flask route?
11. Explain the use of global variables (like lists or dictionaries) to store data in Flask apps. What are their drawbacks?
12. How can you implement a delete functionality in a Flask app using URL parameters?
13. What are static files in Flask and how do you serve them?
14. How do you run a Flask application locally?
15. Why is it important to organize your Flask project using folders like templates and static?

Instructions:

- Use Flask as the web framework.
- Store data in Python (using a list or dictionary). Database is not mandatory.
- HTML pages must be created using Jinja2 templates.
- Organize your code cleanly with a proper folder structure.

Assignment Questions (Tasks)

Q1. Setup Flask Project and Create a Home Page

- Create a basic Flask project.
- Set up routing for '/' (home).
- Display a welcome message using an HTML template.

File: app.py, templates/index.html

Q2. Create a Form to Add New Contact

- Add a new route /add-contact.
- Create an HTML form that accepts:
 - Name
 - Email
 - Phone Number
- On form submission (POST request), display a success message.

File: templates/add_contact.html

Q3. Store Submitted Contact Details in a List

- Use a global Python list to store contacts temporarily.
- Each contact should be stored as a dictionary in the list.
- Ensure new contacts are appended correctly.

Example:

```
contacts = [  
    {"name": "Amit", "email": "amit@gmail.com", "phone": "9876543210"}  
]
```

Q4. Display All Contacts on a New Page

- Create a new route /view-contacts.
- Display all saved contacts in an HTML table.
- Use Jinja2 for-loop to render the data dynamically.

File: templates/view_contacts.html

Q5. Add a Delete Contact Feature

- Next to each contact in the table, add a Delete button.
- When clicked, it should delete that contact using its index or email.
- Refresh the contacts list after deletion.

Q6. Add Navigation and Basic Styling

- Use a base.html template to create a common layout.
- Include navigation links: Home, Add Contact, View Contacts.
- Add simple CSS styling using an internal or external stylesheet.

Files: templates/base.html, static/style.css