Interview Questions

1. What is Flask?

Flask is a lightweight and flexible **Python web framework** used to build web applications and APIs.
It is called a **microframework** because it doesn't include built-in tools like database abstraction or form validation — developers add extensions as needed.

---

2. What is REST?

REST (Representational State Transfer) is an **architectural style** for designing web services.
It uses standard HTTP methods (GET, POST, PUT, DELETE) and represents data in formats like **JSON or XML**.
Each URL represents a **resource**, and operations are performed via HTTP methods.

---

3. Difference between GET and POST?

Feature	GET	POST
Purpose	Retrieve data	Send data to the server
Data in URL	Yes (visible in URL)	No (sent in body)
Idempotent	Yes	No
Use Case	Fetch data	Submit form data or upload files

---

4. How does a Flask route work?

A route in Flask maps a **URL path** to a **Python function**.
Example:

```
@app.route('/hello')
def hello():
    return "Hello, World!"
```

When a user visits /hello, Flask calls the hello() function and returns its response.

---

5. What is request.json?

request.json is used in Flask to **get JSON data** sent in a request body.
Example:

data = request.json

name = data['name']

It parses incoming JSON data into a Python dictionary.

---

6. What are status codes like 200, 404?

HTTP status codes indicate the **result of a request**:

- **200 OK** → Request was successful
- **201 Created** → Resource created successfully
- **400 Bad Request** → Invalid request from client
- **401 Unauthorized** → Authentication required
- **404 Not Found** → Resource not found
- **500 Internal Server Error** → Server-side error

---

7. How do you run a Flask app?

You can run a Flask app using:

flask run

Or in Python:

```
if __name__ == '__main__':
    app.run(debug=True)
```

This starts a local web server.

---

8. What is JSON?

JSON (JavaScript Object Notation) is a lightweight **data interchange format** used to transmit data between client and server.
It uses key-value pairs, similar to Python dictionaries.
Example:

{"name": "John", "age": 25}

---

9. How to test an API?

You can test an API using:

- Tools like **Postman** or **cURL**
- Writing test cases using **pytest** or **unittest**
- Example with Flask test client:
- with app.test_client() as client:
- response = client.get('/hello')
- assert response.status_code == 200

---

10. Can we use a database instead of memory?

Yes.
Instead of storing data in memory (temporary), we can use a **database** like SQLite, MySQL, or PostgreSQL for **permanent storage**.
Flask supports databases through libraries like **SQLAlchemy** or **Flask-SQLAlchemy**