1.What is Flask?

Flask is a **lightweight Python web framework** used to build web applications and APIs.

- It is **minimalistic**, meaning it doesn't include extra tools like ORM or form validation by default.
- Ideal for small to medium web apps or APIs.
- Key features: Routing, templates, request handling, and extensions for databases or authentication.

2.What is REST?

REST (Representational State Transfer) is an **architecture for designing networked applications**, especially web APIs.

- Uses standard **HTTP methods**: GET, POST, PUT, DELETE.
- Stateless: Each request contains all info for the server to process.
- Common in web services and API development.

3.Difference between GET and POST?

Method	Purpose	Parameters	Visibility	Idempotent
GET	Retrieve data	URL query params	Visible in URL	Yes
POST	Send data / create	Body (JSON/form)	Not visible in URL	No

4.How does a Flask route work?

- Routes define **URL endpoints** and link them to **Python functions**.
- Syntax:

```
@app.route('/users', methods=['GET', 'POST'])
def users():
    return "Hello Users"
```

- When a client accesses `/users`, Flask calls the linked function and returns a response.

5.What is request.json?

- In Flask, `request` represents the client's request.
- `request.json` is used to **get JSON data sent in the request body** (usually for POST/PUT APIs).

```
from flask import request
data = request.json
name = data['name']
```

6.What are status codes like 200, 404?

HTTP status codes indicate the result of a request:

Code	Meaning
200	OK – request succeeded
201	Created – new resource created
400	Bad Request – invalid data
404	Not Found – requested resource doesn’t exist
500	Internal Server Error – server problem

7.How do you run a Flask app?

Make sure Flask is installed:
pip install flask
Run the app:
python app.py

8.What is JSON?

SON (JavaScript Object Notation) is a **lightweight format for storing and exchanging data**.

- Easy to read for humans, easy to parse for machines.

```
{
  "name": "Alice",
  "email": "alice@example.com"
}
```

9.How to test an API?

- **Postman**: GUI tool to send GET/POST/PUT/DELETE requests
- **cURL**: Command-line tool

10.Can we use a database instead of memory?

Yes! Using a database is preferred for **persistent storage**.

- Instead of storing data in-memory (which disappears when the app stops), you can use:
    - **SQLite** (simple file-based DB)
    - **MySQL / PostgreSQL** (for production)

    - **MongoDB** (NoSQL)
- Flask can interact with databases using **SQLAlchemy**, **Peewee**, or raw database drivers.