# Assignment Restful API & Flask

***1. What is a RESTful API?***
***Ans : Interface for client-server communication.***
***Uses HTTP methods like GET, POST, PUT, DELETE.***
***Follows REST principles for stateless service***

***2. Explain the concept of API specification.***
***Ans : Defines endpoints and methods.***
***Specifies request and response formats.***
***Ensures consistent API behavior.***

***3. What is Flask, and why is it popular for building APIs?***
***Ans : Lightweight python framework.***
***Used for web apps and APIs.***
***Simple, flexible, and easy to extend.***

***4. What is routing in Flask?***
***Ans : Maps URLs to python functions***
***Determines which function runs***
***Handles request for specific URLs.***

***5. How do you create a simple Flask application?***
***Ans : Import flask and create app instance.***
***Define routes using @app.route().***
***Run the app with app.run()***

***6. What are HTTP methods used in RESTful APIs?***
***Ans : GET retrieves data.***
***POST sends data***
***PUT/PATCH update , DELETE removes***

***7. What is the purpose of the @app.route() decorator in Flask?***
***Ans : Links URL to function***
***Executes function when URL is visited***
***Defines route behavior.***

***8. What is the difference between GET and POST HTTP methods?***
***Ans : GET retrieves data in URL.***
***POST sends data securely in body.***
***POST used for creating/updating resources.***

***9. How do you handle errors in Flask APIs?***
***Ans : Use @app.errorhandler()***
***Return custom message***

*Use proper HTTP status codes.*

***10. How do you connect Flask to a SQL database?***
***Ans : Use SQLAlchemy or connector***
*Initialize in Flask app.*
***11. What is the role of Flask-SQLAlchemy?***
***Ans : Simplifies database operation***
*Uses ORM for python classes*
*Avoid raw SQL queries*

***12. What are Flask blueprints, and how are they useful?***
***Ans : organize app into modules***
*Separate routes, templates, static files*

***13. What is the purpose of Flask's request object?***
***Ans : Access form data***
*Access JSON and query parameters*
*Handle client data in routes*

***14. How do you create a RESTful API endpoint using Flask?***
***Ans : define route with @app.route()***
*Return data in JSON formate*
*Use jsonify() for response*

***15. What is the purpose of Flask's jsonify() function?***
***Ans : Convert python data to JSON***
*Suitable for API responses*

***16. Explain Flask's url_for() function.***
***Ans : Generates dynamic URLs***
*Avoid hardcoding*
*Useful in templates and redirects*

***17. How does Flask handle static files (CSS, JavaScript, etc.)?***
***Ans : Place files in static folder***
*Access via/static/filename*
*Flask serves them automatically.*

***18. What is an API specification, and how does it help in building a Flask API?***
***Ans : Defines endpoint and formats***
*Ensures consistent behaviors*
*Helps developers integrate easily*

***19. What are HTTP status codes, and why are they important in a Flask API?***
***Ans : Indicate success, error, or failure***
*Inform client about request result*
*Standardize API responses*

***20. How do you handle POST requests in Flask?***
***Ans : use request.form for data***
***Use request.json for JSON payloads***
***Validate and process the data***

***21. How would you secure a Flask API?***
***Ans : Use authentication and authorization***
***Enable HTTPS***
***Validate input to prevent attacks***

***22. What is the significance of the Flask-RESTful extension?***
***Ans : Simplifies building REST APLs***
***Provides classes and resources***
***Organizes endpoint efficiently***

***23. What is the role of Flask's session object?***
***Ans : Stores user-specific data***
***Maintain state across requests***
***Useful for loging information***