FLASK ASSIGNMENT

Q1. What is a Web API?

Ans : -

An API which is interface for Web is called as Web API. It consist of one or more endpoint to define request and response. API will communicate to web Application/ Database (if needed). API return Data to client.

Q2. How does a Web API differ from a web service?

Ans :-

API :- All API are not web services But some of them are web services. API can be hosted within the application or Web service. An API acts as an interface between two different applications. So that they are communicate each other.

Web Service :- All Web services are API. It can only hosted on web server. A web service uses limited communication like SOAP , REST etc. Web Service facilities interaction between two machines over network.

Q3. What are the benefits of using Web APIs in software development?

Ans :-

Web API allow developers to use pre-build function and service, saving time and effort. API enable different software system to communicate and work together. This allows different part of an application to be updated or replaced independently.

Q4. Explain the difference between SOAP and RESTful APIs.

Ans :-

SOAP (Simple object Access Protocol) :- SOAP is rigid and highly standardization use XML for message format require strict protocol. It includes built-in error handling and security features. It is mostly used in transaction oriented Application.

RESTful APIs (Representational state transfer) :- It is more flexible and use standard HTTP methods (GET,POST,HEAD,PUT etc.) to perform operation. It can handle multiple format like JSON, XML,HTML and Plain Text.

Q5. What is JSON and how is it commonly used in Web APIs?

Ans :-

It is lightweight data interchange format. It is used to store and exchange data. It is easy of human to parse and generated. It is commonly used as a format for transferring data between client and server.

Q6. Can you name some popular Web API protocols other than REST?

Ans :-

Here are some popular Web API protocol other than REST. These are following :-

a. SOAP
b. GraphQL
c. RPC Api

Q7. What role do HTTP methods (GET, POST, PUT, DELETE, etc.) play in Web API development.

Ans :-

GET -> Retrieve data from the Server.

POST -> Submit data to be processed by the Server.

PUT -> Updates existing data on the server.

DELETE -> Remove data from the Server.

OPTION -> Retrieve the HTTP method Support by the Server for a specific Resources.

HEAD -> Retrieve metadata about a resources without fetching the resources itself.

TRACE -> Perform a message look-back test along the path to the target resources.

Q8. What is the purpose of authentication and authorization in Web APIs?

Ans :-

Authentication :- It ensure that only the right people or system can access the API.

Authorization :- Decide what each person or system can do once they are in.

Q9. How can you handle versioning in Web API development?

Ans :-

Q10. What are the main components of an HTTP request and response in the context of Web APIs?

Ans :-

Component of HTTP Request :-

1. **Request Line :-** In this we have Method (GET,PUT,DELETE,POST) ,URL the endpoint of resources being accessed. HTTP version The version of the HTTP Protocol being used.
2. **Header :-** Key Value Pair that provided metadata about the request (eg :- Content type ,authorization etc.).

Component of HTTP Response :-

1. **Status Line :-** Status code indicate the result of the request.
2. **Status Message :-** Textual description of the status code.

Q11. Describe the concept of rate limiting in the context of Web APIs.

Ans :-

Rate limiting is a method used int web API to control the number off request a user can make to the server in a specific time period it ensure the API remain responsive. If limit is crossed then server responds with an error message like 429.

Q12. How can you handle errors and exceptions in Web API responses?

Ans :-

We have to used standard HTTP status code to indicate the type of Error.

Status Code :-

400 (Bad Request) :- Request is invalid.

401 (Understand) :- Authentication is required.

404 (Not Found) :- The request resources does not exist.

500 (Internal Server Error) :- The server encountered an unexpected conditions.

Error Message :-

provide clear and descriptive error message in the response body to explain what went wrong.

Q13. Explain the concept of statelessness in RESTful Web APIs.

Ans :-

It means the each request from a client to the server must contain all the information needed to understand and processed. The server does not store any context or session information between request.

Q14. What are the best practices for designing and documenting Web APIs?

Ans :-

Designing and Document web API involve ensuring they are easy to use , maintain and understand.

Designing :-

It Stick to clear and consistent naming for endpoint and parameter. It include a version number in your API. It return appropriate status codes and helpful error message.

Documentation :-

Provide detail information on how to use each endpoint.it show request and response example end point. It explain how to authenticate and authorize request.

Q15. What role do API keys and tokens play in securing Web APIs?

Ans :-

API Keys :-

API keys and tokens are essential tool for securing API. They help ensure that only authorized users can access or interact with API.

They help control who can access the API by validating the Key against a list of authorized Key.

Tokens :-

Token authenticate the user making the request often issued after logging in or providing Credential. They determine what actions a user is allowed to perform.

Q16. What is REST, and what are its key principles?

Ans :-

Its full form is Representational state Transfer. It is an architectural style for designing networked Application, especially web APIs.

Its Key Principle:-

1. **Client-Server Seperation :-** Divide the user interface from data storage and Processing.

2. **Statelessness :-** Each request from the client to the server contain all information.
3. **Uniform Interfaces :-** Use standard method (GET, POST,PUT,DELETE)
4. **Cachability :-** Response from the server can be marked as cacheable or non- cacheable to improved performances.

Q17. Explain the difference between RESTful APIs and traditional web services.

Ans :-

RESTful API :- It follow the principle of REST, Each request contain all necessary information utilizes standard HTTP method typically use JSON and XML for data exchange.

Traditional Web API :- It follow SOAP . It can be stateful or stateless involve more complex XML based message. message can be larger and more verbose than RESTful APIs.

Q18. What are the main HTTP methods used in RESTful architecture, and what are their purposes?

Ans :-

In Restful Architecture the main method also known as CRUD operation are used to perform action or resources.

1. GET -> Retrieve Data or information from the server.
2. POST -> Create new Data or information from the server.
3. PUT -> Updating existing data on the server.
4. DELETE -> Remove Data or information from the server.

Q19. Describe the concept of statelessness in RESTful APIs.

Ans :-

Q20. What is the significance of URIs (Uniform Resource Identifiers) in RESTful API design?

Ans :-

URI help client find specific resources on the server. They act like the addresses for resource guiding client to the right location in the API. Client can use URIs to navigate through different part of API.

Q21. Explain the role of hypermedia in RESTful APIs. How does it relate to HATEOAS.

Ans :- Hypermedia plays a significant role in the restful API by enabling dynamic navigation and discovery of resources. Hypermedia include links within API and responses so that client can dynamically navigate through the API.

HATEOAS (Hypermedia as the Engine of Application state)

With this client are not required to have prior knowledge of how to interact with the API. It is useful for making API as self – discovered.

Q22. What are the benefits of using RESTful APIs over other architectural styles?

Ans :-

Simplicity :- It is based on simple and standard HTTP method making them easy to understand and use.

Flexibility:- They work well across different platform and technologies. Allowing client and server to communicate seamlessly.

Caching:- Improve performance and reduce server load.

Q23. Discuss the concept of resource representations in RESTful APIs.

Ans :- Resources representation in RESTful API are the various format in which resources are present in response. Such as JSON or XML. Client can request or accept different representation based on their need providing flexibility in how they interact with the API.

Q24. How does REST handle communication between clients and servers?

Ans :- Using standard web protocol ,primarily HTTP**.**

Resources :- In REST , Data and functionality are considered resources. Each resources is identified by a URL.

Method :- Client communicate with server by sending HTTP request. (GET,POST,PUT,DELETE,HEAD)

Stateless :- Each request from a client to a server must contain all the information the server need to understand and process -the request.

Representation :- When a client request a resource the server send back representation of the resources. This can in various Format Such as JSON,XML or HTML.

Q25. What are the common data formats used in RESTful API communication.

Ans :- Commonly used for data formats in RESTful API communication is JSON and XML and HTML and Plain Text and YAML and CS.

Q26. Explain the importance of status codes in RESTful API responses.

Ans :- Status codes in RESTful API responses are crucial because they inform clients about the outcome of their requests. They quickly indicate success, errors, or the need for further actions, helping clients understand and handle responses appropriately without examining the entire message body. This enhances communication efficiency and reliability between clients and servers.

Q27. Describe the process of versioning in RESTful API development.

Ans :-

The process of versioning in RESTful API development involves several step to ensure that changes or improvement can be made without disrupting existing user.

Identity Changes :- Determine what changes or enhancement are needed in the API . These could be new features, changes in data format or improvement in functionality.

Implement Changes :- Develop the new version of the API based on the identified changes ensuring it co exist with the previous version.

Update Documentation :- Clearly document the changes and new version of the API provide details on how client can access different version.

Notify :- Inform client about the new version and guide them on how to upgrade if needed.

Q28. How can you ensure security in RESTful API development? What are common authentication methods?

Ans :-

Ensuring security in RESTful API development involves protecting data and preventing unauthorized access.

1. Use HTTP :- Encrypt data to protected it during transmission.
2. Validate Input :- Ensure that all data received from client is validated to prevented injection attack.
3. Use strong Authentication and Authorization :- Ensure only Authorized Users access the API.
4. Rate Limiting :- Limit the no of request a user can make in a given time period.
5. Data Encryption :- Encrypt sensitive data both in transit and at rest to protect it from unauthorized access.

Common Authentication Method :-

1. API Keys :- The client include a unique key in the request which the server verified.
2. Basic Authentication :- Uses a Username and password encode in Base64.
3. OAuth :- Provide access token instead of credential

Q29. What are some best practices for documenting RESTful APIs?

Ans :-

Developer understand how to use the API and can integrate it into their application Smoothly.

1. Use a standard Format
2. Provide clear and concise Description
3. List all endpoint
4. Explain Request and Response Format
5. Include HTTP Method
6. Authentication Detail
7. Error code and message
8. Versioning Information
9. Updata Regularly.

Q30. What considerations should be made for error handling in RESTful APIs?

Ans :-

Error handling in RESTful API involve consideration how to communicate error effectively to client. Provide information error message with clear explanation. Use Appropriate HTTP status code (eg -400 for client error, 500 for server error).

Q31. What is SOAP, and how does it differ from REST?

Ans :-

SOAP(simple object Access Protocol):- SOAP is rigid and highly standardization use XML for message format require strict protocol. It

includes build in error handling and security features.it is mostly used in transaction oriented Application.

RESTful APIs (Representational state transfer) :- it is more flexible and use standard HTTP methods (GET,POST,HEAD,PUT etc.) to perform operation. It can handle multiple formats like JSON, HTML and Plain Text.

Q32. Describe the structure of a SOAP message.

Ans :-

A SOAP message structure is like letter. It consists of an envelope is like the outer packaging, containing the message detail. The header is like the letter head with optional information such as Authentication or routing details. The body is the main content of the message containing the actual data being sent similar to the main text of letter.

Q33. How does SOAP handle communication between clients and servers?

Ans :-

SOAP handle communication between client and server by using XML based message sent over protocol like HTTP when a client sends a SOAP request to a server, it pakages data into a structured XML format and sends it over the chosen protocol. The server then receives this XML message, processes it and sent back a response in a similar XML format this process allows for standardization.

Q34. What are the advantages and disadvantages of using SOAP-based web services?

Advantages :-

1. **Standardization :-** SOAP follow a standardization protocol, ensuring interoperability between different platform and language**.**
2. **Security :-** It support built in security feature like Ws-Security.
3. **Formal Contracts :-** SOAP web services define clear contract using WSDL making it easier for client to understand how to interact with the services.

Disadvantages :-

1. Complexity :- Soap message are verbose and require parsing XML making them more complex compared to the other.
2. Performance Overhead :- Due to its XML based nature and additional layers for security and reliability SOAP can have higher performance overhead.

Q35. How does SOAP ensure security in web service communication?

Ans :-

1. **Transport-level security :-** SOAP message can be encrypted and transmitted over secure protocol like HTTP.
2. **WS-Security** :- This is widely used extension to SOAP that define a set of standard for securing message at the message level .
3. **Authentication**
4. **Authorization**

Q36. What is Flask, and what makes it different from other web frameworks?

Ans :- Flask is different from other web framework because its like a basic toolkit instead of a full toolbox.

1. **Lightweighted :-** Flask is like a small toolbox with just the basic tool you need to build a web application**.**
2. **Flexibility :-** Flask is so minimal you have more freedom to build your application**.**
3. **No Build Database :-** Flask does not come with built in tool for working with database . This means you to add this tool yourself.

Q37. Describe the basic structure of a Flask application.

Ans :-

Root :- At the root of the tree is your Flask Application file usually python file . This file is where your flask app start.

Branches :- Each branch represent a different URL. You define these branches using route decorators like @ app . route('/').

Leave :- At the end of each branch at leave which represent HTML file user see**.**

Q38. How do you install Flask on your local machine?

Ans:- At first we have to create a virtual python environment .

a. Create a folder
b. In folder we create an environment
c. Then we have to activate this environment
d. Install dependence

For creating Environment**,**

Command :- python -m venv env_name

For activate environment ,

Command :- env_name\scripts\activate

For installing the Flask :-

Command :- pip install flask

Q39. Explain the concept of routing in Flask.

Ans:-

Routing in flask is like giving direction to visitor on your website. In simple word, we can say that routing is like a roadmap that direct user to the right pages on your website when they type in different URL.

Q40. What are Flask templates, and how are they used in web development

Ans :-

Flask templates are HTML files used to create dynamic web pages.

Uses :-

1. Create a folder name ‘templates’ in the same directory
2. Create a HTML file inside the folder.
3. In your Flask app use render_template(‘filename.html’ ,variable=value) to send data to the template and generate dynamic content.

Thank You