1. What is an API?What are the main differences between API and Web Service?

API is the acronym for Application Programming Interface,that enable two software components to communicate with each other using a set of definitions and protocols.

Web Services	API
Web Services provide interaction between two machines over a network	APIs acts as an interface between two different applications for interoperability.
All Web services are APIs.	All APIs are not web services.
It provides supports only for the HTTP protocol.	It provides support for the HTTP/s protocol: URL Request/Response Headers, and so on.
Web service supports only XML.	API supports XML and JSON.

2. What is API Testing? What are the advantages of API Testing?

API testing is a software testing type that validates Application Programming Interfaces and the purpose is to check the functionality, reliability, performance, and security of the programming interfaces.

The advantages of API Testing,

- Language-independent
  Data is exchanged via XML and JSON formats, so any language can be used for test automation. XML and JSON are typically structured data, making the verification fast and stable. There are also built-in libraries to support comparing data using these data formats.
- GUI-independent
  API testing can be performed in the app prior to GUI testing. Early testing means early feedback and better team productivity. The app's core functionalities can be tested to expose small error
- Improved test coverage
  Most API/web services have specifications, allowing you to create automated tests with high coverage — including functional testing and non-functional testing
- Early Testing
  API testing helps identify early issues.Thus,developers can fix bugs before they affect the GUI.

3.What are the differences between API Testing and Unit Testing?

API Testing	Unit Testing
It is carried out by the QA team.	It is carried out by developers.

Only API functions are put to the test.	All functions and classes are put to test
Communication between different components are tested.	Every component is tested separately.
Performed post build is done.	Performed pre check in and hence before build
Frame/tools : Postman,REST Assured	Frame/tools : Junit, TestNG

4.What are the tools used for API testing?

- REST Assured
- Postman
- Jmeter
- Katalon Studio

5.What are the common tests performed on API’s?

API testing involves the following types of testing:

- Unit Testing
- Functional Testing
- Load Testing
- Runtime/Error Detection
- Security Testing
- Penetration Testing
- Fuzz Testing

6.Mention the key difference between UI level testing and API testing?

API testing	**UI testing**
This tests the functionality of the business logic	This tests the functionality of the UI and the business logic
API tests have high speed of execution	UI tests have low speed of execution
API tests are medium Maintenance	UI tests are High Maintenance
API test have medium coverage since it tests only single layer	UI tests have high coverage since it checks 2 layers
API tests are written by Developers and QA	UI tests are usually written by QA

7.What is API framework?

An API framework is more a set of tools, groups of classes, interfaces and other pre-compiled code, components aimed at helping the developer to develop his/her project in a given Frame. The framework usually sets some coding standards, provides useful components

8.Mention the steps for testing API ?

- Choose the suite to add the API test case
- Choose the test development mode
- Demand the development of test cases for the required API methods
- Configure the control parameters of the application and then test conditions
- Configure method validation
- Execute the API test
- Check test reports and filter API test cases
- Arrange all API test cases

9.What are the tools used for API documentation?

Here, are the various API documentation tools that make the whole process simple and easy. They are:

- Swagger
- Miredot
- Slate
- FlatDoc
- API blueprint
- RestDoc
- Web service API specification

10.What is Payload in REST API?

The Payload in REST API is the actual data pack that is sent with the POST/PUT method in HTTP. It's the crucial information that you submit to the server when making an API request.

The payload is denoted using "{}" in the Query string/body, and it can be sent or received in multiple formats.

11.What is Response Status code, Name HTTP status code classes

An HTTP status code is a server response to a browser’s request. When you visit a website, your browser sends a request to the site’s server, and the server then responds to the browser’s request with a three-digit code that is HTTP status code.
Common HTTP status code classes:
1xxs – Informational responses: The server is thinking through the request.

2xxs – Success! The request was successfully completed and the server gave the browser the expected response.

3xxs – Redirection: You got redirected somewhere else. The request was received, but there’s a redirect of some kind.

4xxs – Client errors: Page not found. The site or page couldn’t be reached. (The request was made, but the page isn’t valid — this is an error on the website’s side of the conversation and often appears when a page doesn’t exist on the site.)

5xxs – Server errors: Failure. A valid request was made by the client but the server failed to complete the request.

12.What are principles of API test design

There are the seven principles of API test design:

- Testing shows the presence of defects, not their absence.
- Exhaustive testing is impossible.
- Early testing saves time and money.
- Defects cluster together.
- Beware of the pesticide paradox.
- Testing is context dependent.
- Absence-of-errors is a fall

13.Why is API testing Prefered over UI Testing

API testing is better for testing the APIs underlying the UI. API testing is easy to automate, accurate, less time consuming and cost of execution is low hence it is preferred over UI Testing.
For example :
With API testing, we can hit the API endpoint directly and have control of what data we send to the API for testing purposes. e.g. invalid data, malformed requests, etc, but in UI level testing, we don’t have that level of flexibility because we are bound by the constraints of the UI.

14.What are different types of HTTP Methods and what are the main differences between PUT and POST

The primary or most commonly-used HTTP methods are POST, GET, PUT, PATCH, and DELETE. These methods correspond to create, read, update, and delete (or CRUD) operations, respectively. There are a number of other methods, too, but they are utilized less frequently.

PUT	POST
PUT method is call when you have to modify a single resource, which is already a part of resource collection.	POST method is call when you have to add a child resource under resources collection.
This method is idempotent.	This method is not idempotent.
PUT method syntax is PUT /questions/{question-id}	POST method syntax is POST /questions
PUT method answer can be cached.	You cannot cache PUT method responses.
If you send the same request multiple times, the result will remain the same.	If you send the same POST request more than one time, you will receive different results.
PUT works as specific.	POST work as abstract.
We use UPDATE query in PUT.	We use create query in POST.

15.Define Test Data?

Test Data is the input given to a software program during test execution. It represents data that affects or affected by software execution while testing. Test data is used for both positive testing to verify that functions produce expected results for given inputs and for negative testing to test software ability to handle unusual, exceptional or unexpected inputs.

16.What is an HTTP request? What are the core components of an HTTP request?

An HTTP request is made by a client, to a named host, which is located on a server. The aim of the request is to access a resource on the server.
When they make a request, clients use a URL (Uniform Resource Locator) that contains the information needed to access the server resources.

An HTTP request contains five core components:

- **Action showing HTTP method** like GET, PUT, POST, DELETE.
- **Uniform Resource Identifier (URI):** URI is the identifier for the resource on the server.
- **HTTP version:** Indicate the HTTP version like- HTTP V1.1.
- **Request Header:** Request Header carries metadata for the HTTP request message. Metadata could be a client type, format supported by the client, format of a message body, cache setting etc.
- **Request Body:** Resource body indicates message content or resource representation.

17. What does status code 302 stand for ? how can we find the url to be redirected to in case of a 302 status code?

The HTTP response status code 302 is a common way of performing URL redirection.It is used for the purpose of temporary redirection.

We can find the URL to be redirected in the HTTP response with status code it will additionally provided in the URL header field Location. This is an invitation to the user agent (e.g. a web browser) to make a second, otherwise identical, request to the new URL specified in the location field. The end result is a redirection to the new URL.

19. How can we pass input information through a GET API call ?

We can pass input information through GET API call using :

- Header parameters : Parameters included in the request header, usually related to authorization.
- Path parameters: Parameters within the path of the endpoint. In the documentation, they're denoted with { curly braces }.
- Query string parameters: Parameters in the query string of the endpoint, after the "?" separator.
- Request body parameters: Parameters included in the request body. Usually submitted as JSON.

20. What are the parts of a HTTP request ?

An HTTP request is divided into three parts:

- **Request line** : In the request line we place the HTTP method to be used, the URI of the request and the HTTP protocol to be used
- **Request Header** : The header of the request is where the headers of the request are located. Headers are metadata that are sent in the request to provide information about the request.

- **Request Body :** The Request Body is where we put additional information that we are going to send to the server.

21. What does status code 500 denote?
The HTTP status code 500 Internal Server Error server error response code indicates that the server encountered an unexpected condition that prevented it from fulfilling the request.

22. How do you debug a status code 500 response?
Steps to debug 500 status code :

- Refresh the page/Refresh the page.
- Delete your browser's cookies.
- Check File Permissions
- Ask developer to revert recent changes
- Check if the Admin Works
- To check server side logs

23. Which part of the HTTP request holds your authorization token
Request header holds Authorization token details

24. What does status code 100 denote?
The HTTP 100 Continue informational status response code indicates that everything so far is OK and that the client should continue with the request or ignore it if it is already finished.

POSTMAN Byte

1. **What is Postman? What are the advantages and disadvantages of Postman**
   Postman is an application used for API testing.
   In technical terms Postman is an HTTP client that tests HTTP requests, utilizing a graphical user interface, through which we obtain different types of responses that needs to be subsequently validated.

   Advantages:

   - User-friendliness
   - Accessibility
   - Various functionalities.
   - Request tracking capabilities

   Disadvantages:

   - Limited testing area. While Postman is ideal for RESTful API tests, it is not well designed for SOAP APIs and other APIs.

- Low script reusability. Postman users are unable to reuse their pre-written scripts or add more requests. This means testers have to create new test scripts over and over for each project.
- Constrained integration. While APIs enable the Agile process, the tool itself does not support much in integration capabilities.

2. **What are the various authorization methods provided by Postman?**
   Postman provides the following API request authorization options:

- API Key
- Oauth 1.0
- Oauth 2.0
- Bearer Token
- Basic auth
- Digest auth
- Hawk Authentication
- AWS Signature
- NTLM Authentication

3. **What are the collection,environments and Workspace in Postman**

Collections : A collection is used to group similar requests. It systematically arranges the requests into folders.
Environments : Environment is a set of key-value pairs. You can create multiple environments in Postman and switch among them quickly by pressing a button. There are 2 types of environment, global and local.
Workspace : A workspace is a collaborative environment for users to develop and test APIs. In the same way, a team workspace is a workspace that is shared by the whole team working on the same collections of requests.

4. **Can we have two global scope variables with the same name in Postman?Which one is preferred in Postman, a global or local variable?**
   The global variables are global, so we cannot set duplicate names for them without any environment as it creates confusion for the software. On the other hand, local variables can have the same name but in different environments.
   In Postman, if 2 variables have the same name (one being local, the other global), then the higher priority is of the local variable. The local variable will overwrite the global variable.

5. **Write Syntax to log variables in postman**

console.log(Variable name, Variable value);

6. **What is the difference between Query Params and Path Variables?**

Query Params or Query Parameters are used for sorting or filtering the resources. On the other hand, Path Variables are used for identifying specific resources.

7. **What is the use of the 301 status code in Postman?how to enable or Disable it in postman**
   The 301 status code is used to specify that the page has been permanently redirected from one website page to another. It tells the search engine that the old page is outdated, and the search engine has to index the new page URL.

To Enable/Disable 302 redirection in Postman:
Disable the automatic redirection for 3xx responses. To do this, open the Settings tab of your request and toggle off the Automatically follow redirects option.

8. **What are the various variable scopes provided by Postman?**
   There are 5 variable scopes provided by postman:

- **Global Variables:** Global variables allow data access between different collections, requests, and scripts. They are available throughout the workspace.
- **Local Variables:** Local variables are the temporary variables that can be accessed only within the scope of requests scripts. Depending on the requirements, these variables are either scoped to a single request or single collection. These variables are not available once the script execution is completed.
- **Environment Variables:** The Environment variables allow us to tailor the requests about different development environments such as local testing, stage testing, or prod testing.
- **Collection Variables:** The Collection variables are independent of the environment and scoped to be available for all the requests present within the collection.
- **Data Variables:** The Data variables come from external JSON or CSV files and define the datasets required to run the collection in Collection Runner or Newman.

9. **Where can we store query parameters in postman**

   The query parameters are stored in the URL in Postman.

10. **What is variable in Postman and write syntax to access Postman variable?When do we use global variables, collection variables, and local variables?**

The postman variable is similar to the programming language variable. As we know, a variable is an entity, which stores value. You can change the value of the variable. In Postman, a variable is a set of key-value pairs.

The Postman variables are always accessed by using the variable name: `{{variable name}}`

Global variables are all-purpose variables perfect for prototyping and quick results.
Collection variables are used to store constants that don't change while the Collection is being executed. They are used for constants that remain the same throughout execution
Local variables are only accessible for the duration of the request that set them or while using the Collection runner. They get used whenever you want to override all other variable scopes.

11. **Which programming language is used for Postman tests?Write a program to generate random numbers of a given range in postman**
    JavaScript is used for Postman tests.

Postman javascript code to generate random numbers
We can generate random/dynamic data in requests using the following functions.

- Math.random() :

```
const randomNumber = Math.floor((Math.random()*100 +1);
pm.globals.set("randomNumber", randomNumber);
```

- $randomInt : We can use Postman built-in variable randomInt to generate random integer value.

```
"randomInteger": {{$randomInt}},
}
```

12. **What is GUID?**
    Global Unique Identifier is referred to as GUID. It consists of hexadecimal digits separated by hyphens. This Postman identifier GUID fulfills the goal of uniqueness.
13. **Write a sample program to set same headers for all requests in postman Collection**
    In postman, we can add same HTTP headers to entire collections with pre-request scripts

- Go to left-menu and select pre-requisite tab
- And add below java script code

```
pm.request.headers.add({
    key: "Accept",
    value: "application/json"
});
```

- Before running the collection, we must make sure that we removed the pre-request scripts we initially added.
- Hit the Run button on the collection bar, and the Runner tab will open automatically:
- The Runner tab allows us to order our requests, select or deselect requests from our collection, and specify additional settings. Click on Run API Collection to execute our requests.
- Once the entire collection is finished, we can see the order of executions and test results, if any. However, we want to make sure our HTTP headers were part of our requests, and we can confirm that by opening the Postman console.

14. What are cookies ? Where do we define them on postman

A cookie typically has two pieces of data: a unique ID for each user and a site name. Cookies enable websites to retrieve this information when you revisit them, so that they can remember you and your preferences and tailor page content for you based on this information.

Steps to define a cookie :

- select the 'Add Cookie' button
- Enter the domain name in the URL text field and click on the Add button
- A new text box will open where some default values are already written. Edit those values as per requirement
- Click on the Save button.

REST-Assured Byte

1. What is REST Assured?How Rest Assured works internally?

Rest Assured is used to verify the REST APIs with the help of the Java library. Rest Assured acts like a headless client to act upon the Rest web services. The libraries based on the Rest Assured library are also capable of validating the HTTP responses from the server.

Rest assured uses groovy internally.It is t used to extract responses and it works for both XML and JSON.

2. What is Request Specification in Rest Assured and how to use it ?

Every Request in Rest-Assured library is represented by an interface called RequestSpecification. This interface allows to modify the request, like adding headers or adding authentication details.

A RequestSpecification with some specifications can be created as below:-

RequestSpecification requestSpecification = RestAssured.given();
requestSpecification.baseUri("https:booking.com")

requestSpecification.basePath("/path1");

3. **Why do we use static import in Rest Assured?**
   Static import is a Java language feature that allows fields and methods that have been scoped as public static within their container class to be used in Java code without mentioning the class in which the field has been defined.

4. **What is client server architecture?**
   The client-server architecture defines how a server gives resources and services to one or more clients. So, when the Client requests something the Server fulfils the request.

5. **Write Syntax/Code to test GET request in REST API?**

1. Create a new java class
2. Import required libraries

```

    import io.restassured.http.Method;
    import io.restassured.response.Response;
    import io.restassured.specification.RequestSpecification;
    import io.restassured.RestAssured;
```

3. Specify the base URL in RestAssured.baseURL method

```

   RestAssured.baseURI = "https://content-qkart-qa-backend.azurewebsites.net";
   RestAssured.basePath = "/api/v1/products";
```

4. Use the RestAssured class to generate a RequestSpecification and Get the RequestSpecification of the request to be sent to the server
5. Specify the HTTP Method type (GET method).

```

    RequestSpecification http = RestAssured.given();
     Response response = http.request(Method.GET);
```

6. Send the Request to the server.
7. Get the Response back from the server.
8. Validate the status code

```
```

```
int responseStatusCode = response.getStatusCode();
if (responseStatusCode == 200) {
    System.out.println("The API call was successful");
} else {
    System.out.println("The API call Failed");
}
```

6. **What is JSON and What is jsonPath in Rest Assured?**

   JavaScriptObjectNotation (full form of JSON) is a standard file format used to interchange data.The data objects are stored and transmitted using Key-Value pairs and array dataTypes .
   JSONPath is a query language for JSON, similar to XPath for XML. It allows to select and extract data from a JSON document. We use a JSONPath expression to traverse the path to an element in the JSON structure. In Rest Assured, It follows the Groovy dot notation syntax when getting an object from the document.

7. **How to use Path Variable with GET rest endpoint in Rest Assured?**
   pathParam() method can be used to pass Path variables and this method takes two string parameters. The first parameter is the parameter name and another parameter is the parameter value.
   Syntax is as below,

```
RestAssured .given()
            .baseUri("URL")
              .pathParam("name", "value")
            .when()
            .then()
```

8. **Write syntax/code to send Post request in REST Assured?**

- Create a new java class.
  Import required libraries
  ```
  import io.restassured.http.ContentType;
  import io.restassured.http.Method;
  import io.restassured.response.Response;
  import io.restassured.specification.RequestSpecification;
  import io.restassured.RestAssured;
  ```
- Create a Request pointing to the service Endpoint.
- Specify the base URL in RestAssured.baseURL method

- Use the RestAssured class to generate a RequestSpecification and Get the RequestSpecification of the request to be sent to the server

```
RequestSpecification http = RestAssured.given();
        http.contentType(ContentType.JSON);
```

- Create a JSON Request which contains all the fields. For ex,

```
 String jsonString =
"{\"username\":\"testUser\",\"password\":\"password12e\"}";
```

Add JSON body in the request and send the request.

```
  http.body(jsonString);

         Response response = http.request(Method.POST);
```

- Get the Response back from the server.

9. **What is Serialisation and Deserialization in Java?**

Serialization is the conversion of a Java object into a stream (sequence) of bytes, which we can then save to a database or transfer over a network.

Deserialization is the reverse process where the byte stream is used to recreate the actual Java object in memory.

10. **How to log in case of error in response in Rest assured?**

To print the response body if an error occur then we can use:
.log().ifError() .
It logs everything only if an error occurs (status code >= 400).

11. **Explain ways of extracting a single field from a response body.**

- By using JSONObject class.

```
 String jsonStr = "{\"name\":\"TestUser\",\"password\":\"Password123\"}";
JSONObject jsonObj = new JSONObject(jsonStr);
String name = jsonObj.getString("name");
System.out.println(name);
```

- By importing JsonParser

```
import org.json.simple.parser.JSONParser;
```

```
JSONParser parser = new JSONParser();
JSONObject jsonResponse = (JSONObject) parser.parse(response);

To get name,
String name = (String) jsonObject.get("name");
```

12. **Can we write RestAssured.with() instead of RestAssured.given()? What is the difference?**

    with() and given() are used to perform the same function. The package from which it is imported indicates that with() and given() do the same thing.

```
public static RequestSpecification with() {
  return given();
}
```

13. **Which protocol does RESTful Web Services use?**
    RESTful Web services are built on the HTTP protocol and implement operations that map to the HTTP/s methods, GET, POST, PUT, and DELETE to create, retrieve, update, and delete resources, respectively.

14. **How do you perform chaining in REST Assured?**

Chaining in RestAssured can be performed by RequestSpecification interface. This interface is mainly used to add headers , params, authentication , body, cookies , proxy etc to request body.
Syntax for chaining request is,

```
RequestSpecification req= RestAssured.given();
        req= req.accept(ContentType.JSON);
        req= req.header("headername", "headervalue");
        req= req.param("paramname", "paramvalue");
        req= req.cookie("cookieName", "value");
```

15. **How to handle parameters that are already URL encoded?**

In REST Assured we should use urlEncodingEnabled(false) method to handle parameters which are URL encoded.
For Example :

```
String uriPath="/rest/api/2.0.alpha1/search";
RestAssured.baseURI=""https://jira.atlassian.com:443";
```

```
String response =
given().urlEncodingEnabled(false).and().given().queryParam("","").and().given().get()
.asString();
```

Performance Testing Basics

1. **What is Performance Testing and what is the need for conducting performance testing?**
   Performance testing is a non-functional testing technique, which helps to determine how the stability, scalability, responsiveness and speed of an application hold up under a given workload.

   Performance testing is used to measure and analyze response times and potential errors and helps to clearly identify bugs and mistakes –and is used to guide to optimize the application, eliminating the problems found during testing.

2. **What are the types of Performance Testing?**
   Capacity Testing: Tests how many users the system can handle before performance dips below acceptable levels.

   Load Testing: Confirms that the system can handle the required number of users and still operate at a high level of performance.

   Volume Testing: Checks that the software can handle and process a large amount of data at once without breaking, slowing down, or losing any information.

   Stress Testing: Intentionally tries to break the software by simulating a number of users that greatly exceeds expectations. The launch day of a new phone and the sudden spike in user traffic on the website is a good example of a stress test in the real world.

   Soak Testing: Simulates high traffic for an extended period of time. Checks the software's ability to tolerate extended periods of high traffic.

3. **What are some of the commonly available tools for performance testing?**
   Apache JMeter
   LoadRunner[HP]
   LoadNinja
   WebLOAD
   LoadComplete
   NeoLoad
   LoadView
   Multi-mechanize

4. **What are some of the common problems that occur due to poor performance testing?**
   Poor response and loading times
   Poor scalability
   Bottlenecks like deadlocks, missing indexes

5. **What is performance testing Metrics, why is it important ?**
   Performance testing metrics are the parameters gathered during the performance and load testing processes.
   With the help of these metrics, performance test engineers or QA teams determine the success of the performance testing process and further identify the critical areas in the software that needs more attention/improvement.

6. **What are the key performance testing metrics?**
   CPU utilization:
   It is the percentage of CPU capacity utilized in processing the requests.

   Memory utilization:
   This metric measures the utilization of the primary memory of the computer while processing any work requests.

   Response times:
   It is the total time between sending the request and receiving the response. Better the response time, better the performance of website/application.

   Average load time:
   This metric measures the time taken by a webpage to complete the loading process and appear on the user screen.

   Throughput:
   It measures the number of transactions an application can handle in a second, or in other words, it is the rate at which a network or computer receives the requests per second.

   Average latency/Wait time:
   It is the time spent by a request in a queue before getting processed.

   Bandwidth:
   It is the measurement of the volume of data transferred per second.

   Requests per second:
   This metric refers to the number of requests handled by the application per second.

Error rate:
It is the percentage of requests resulting in errors compared to the total number of requests.

Transactions Passed/Failed:
It is the percentage of passed/failed transactions against the total number of transactions.

7. **What are the steps involved in conducting performance testing?**
The steps involved in the performance testing process are ,

- Develop the Right Testing Environment

Examine the software, hardware, and network configuration the performance test is going to be utilizing. Understanding the strengths and weaknesses of the host system is important later on for identifying potential problems. Make adjustments and upgrades to parts of the environment that may be ill-suited for the test. Lastly, ensure the testers are familiar with the tools that will be used to measure performance.

- Identify the Performance Acceptance Criteria

In this step, tester should ask how many simultaneous users the platform needs to support, and how quickly it needs to respond to those requests. You'll examine things like resource utilisation and platform throughput in this stage of the process.

- Plan and Design Performance Tests

During the planning and design phase, define how usage is likely to vary among end users, along with scenarios to test that showcase typical visitor use cases.

- Set Up the Performance Testing Environment

During this step, tester will configure and arrange the tools and other resources to perform the test. This step also includes designing the scripts the testing application will use to simulate user activity.

- Test the Design Implementation

The next step involves testing the test environment for potential bottleneck problems. If the test servers aren't capable of generating the number of virtual users required to run the test, the results will not be accurate. Run a pilot performance test to gauge CPU, memory, and network utilisation on the test server.

- Run the Test

While running the test, monitor and record platform performance data for analysis. This information not only identifies whether the infrastructure is capable of handling the traffic load efficiently, but also helps to determine where power should be added to address a growing user base.

- Analyze, Tune, and Retest

Examine the test results both during and after the test. If the testing platform is experiencing a problem like a CPU or network bottleneck, stop the test and address the issue before repeating the process.
Once you have completed test results, analyze the reports. Use this information to adjust the test to measure different performance levels and to identify ways to improve platform performance. Repeat the test and measure performance improvements to see if the recommended adjustments improve performance.

8. **What are Bench-marking,Reporting and Performance tuning?**
   A Benchmark in Performance Testing is a metric or a point of reference against which software products or services can be compared to assess the quality measures.

   Reporting provides an overall test result, test analysis and recommendations to the project team or client from application's performance perspective. The outcome of the Report document helps to take the decision for the whole application or specific business flow.

   Performance tuning is applied to correct the declared faults found during Performance Test

9. **What do we measure in Performance Testing ?**
   Excessive Load Times
   To start an application, you'll need a certain amount of load time. Delays should be kept to a minimum to provide the best possible user experience.

   Poor Response Times
   Latency is the time elapsed between a user entering information into an application and the response to its action. Long response times significantly reduce the interest of users within the application.

   Limited Scalability
   Limited scalability is a problem with an application's ability to scale up or scale down as needed to accommodate more or fewer users. While the appliance works well with a small number of users, it starts to struggle as the number of users grows.

   Bottlenecks
   Bottlenecks are performance-degrading impediments in a software system. They're usually the result of faulty hardware or bad code.

10. **What do you mean by throughput of an API server ?**

Throughput is the number of transactions produced over time during a test and it is also expressed as the amount of required capacity that a website or application can handle.

11. **How is load testing different from Performance testing ?**

Load testing	Performance testing
Load testing is the process of determining the behavior of the system when multiple users access it at the same time.	Performance testing is the process of determining the system's performance that includes speed, and reliability under varying loads.
In load testing peak load is used for testing.	In performance the load on which the system is tested is normal.
The load testing has the main objective of increasing the load on a web application.	The main goal of performance testing is to determine how an application will operate under typical conditions.
Only the sustainability of the system is tested during load testing.	Speed, scalability, stability, and reliability are tested during performance testing.

12. **What is an endurance test?**
Endurance testing is a type of performance testing of the software to check system performance under specific load conditions over an extended or longer amount of time.
13. **What is the difference between spike and stress test?**
Stress Testing is a testing process in which an application is tested beyond the anticipated/expected load condition.This is done in order to know the extreme level till which an application be loaded i.e to find the breakpoint.

Spike Testing is a testing process in which an application is tested sudden increment and decrement in the load.The system is suddenly loaded and unloaded.It is done to see how the system reacts with unexpected rise and fall of users.
14. **What is the entry criteria for performance testing?**
The following are the conditions for performance testing entry criteria,

- Clear and Approved Requirements
- Selection of Performance Test Type
- Assuring Stability of the Software
- A Dedicated Setup and Testing Environment
- Planning to Handle Problems

15. **What is the difference between benchmarking and baselining?**

benchmarking	baselining
Benchmark Testing metrics are often	Baseline metrics are recorded after the

pre-established to evaluate the performance.	applications undergo performance testing.
Benchmark is often applicable to all the software applications belong to an organisation.	Baseline Testing is specific to an individual software application.
Benchmark testing is done from business and SLA point of view.	Baseline Testing is done from the application and user experience point of view.
Benchmark Testing is done on a golden number established by team.	Baseline testing is done against previous or golden build.