SOFTWARE TESTING ASSIGNMENT

Module-3&4(Testing on live application & Automation core testing)

1. What is RDBMS?

RDBMS stands for Relational Database Management System. It is a type of database management system that uses a relational model to store and manage data. In an RDBMS, data is organized into tables with rows and columns, and the relationships between the tables are defined by keys. RDBMS provides a structured and organized way to store and retrieve data, with support for data integrity, consistency, and scalability. Popular examples of RDBMS include MySQL, Oracle Database, Microsoft SQL Server, and PostgreSQL.

2. What is SQL?

SQL stands for Structured Query Language. It is a domain-specific language used to manage and interact with relational databases. SQL allows users to define, manipulate, and retrieve data from databases using queries, making it a fundamental tool for managing and analyzing data in modern database systems.

3. Write SQL Commands?

- SELECT: Retrieve data from a database table. Example:

```
SELECT column1, column2 FROM table_name;
```

- INSERT: Insert new data into a database table. Example:

```
INSERT INTO table_name (column1, column2) VALUES ('value1', 'value2');
```

- UPDATE: Modify existing data in a database table. Example:

```
UPDATE table_name SET column1 = 'new_value1' WHERE condition;
```

- DELETE: Remove data from a database table. Example:

```
DELETE FROM table_name WHERE condition;
```

- CREATE: Create a new database table or other database objects. Example:

```
CREATE TABLE table_name (column1 data_type1, column2 data_type2);
```

- ALTER: Modify an existing database table, such as adding or deleting columns. Example:

```
ALTER TABLE table_name ADD column3 data_type3;
```

- DROP: Delete an existing database table or other database objects. Example:

```
DROP TABLE table_name;
```

- SELECT with JOIN: Retrieve data from multiple tables using JOIN operations. Example:

```
SELECT table1.column1, table2.column2 FROM table1 INNER JOIN
table2 ON table1.column_id = table2.column_id;
```

4. What is join?

A JOIN is an operation in SQL that combines data from two or more tables in a relational database based on related columns. JOINs are used to retrieve data from multiple tables and combine them into a single result set. JOINs are essential for combining data from related tables and are commonly used in SQL queries to retrieve and combine data from multiple tables for analysis or reporting purposes.

5. Write type of joins?

- INNER JOIN: Returns only the rows where the related columns between two tables match.
- LEFT JOIN (or LEFT OUTER JOIN): Returns all rows from the left table and matching rows from the right table, with NULL values for unmatched rows in the right table.
- RIGHT JOIN (or RIGHT OUTER JOIN): Returns all rows from the right table and matching rows from the left table, with NULL values for unmatched rows in the left table.
- FULL JOIN (or FULL OUTER JOIN): Returns all rows from both tables, with NULL values for unmatched rows in either table.
- CROSS JOIN: Returns the Cartesian product of two or more tables, with all possible combinations of rows from the tables, without any condition on related columns.

6. How Many constraint and describes it self?

- PRIMARY KEY: Uniquely identifies each row in a table.
- FOREIGN KEY: Refers to the primary key of another table to establish a relationship.
- UNIQUE: Ensures that values in a column are unique.
- NOT NULL: Requires a column to have a value for every row.
- CHECK: Defines a condition that must be true for data to be inserted or updated in a column.
- DEFAULT: Specifies a default value for a column if no value is provided during insertion.

7. Difference between RDBMS vs DBMS?

RDBMS	DBMS
Stores data in a structured manner using tables with rows and columns.	Stores data in a less structured manner, such as in hierarchical or network models.
Supports relationships between tables using keys.	May not have built-in features for enforcing data integrity rules.
Enforces data integrity rules, such as referential integrity and constraints.	May use proprietary query languages or none at all for querying and manipulating data.
Typically uses SQL (Structured Query Language) for querying and manipulating data.	May be more suitable for smaller databases with simpler requirements.
Provides better scalability for handling large databases and complex transactions.	May have limited flexibility in data retrieval and manipulation.
Allows for flexible data retrieval and manipulation using SQL queries.	May have limited or no security features.
Provides advanced security features, such as user authentication, authorization, and encryption.	May not provide the same level of data consistency as RDBMS.
Ensures data consistency through ACID properties (Atomicity, Consistency, Isolation, Durability).	

8. What is API Testing ?

API (Application Programming Interface) testing is a type of software testing that validates the functionality, reliability, performance, and security of APIs. It involves sending requests to APIs and validating the responses to ensure they meet expected behavior, data accuracy, and performance requirements. API testing helps identify issues early in the development process, ensures smooth communication between software components, and achieves robust API integrations in applications.

9. Types of API Testing?

- Unit Testing: Validates individual code units in the API for correct behavior.
- Functional Testing: Verifies if the API functions as intended and returns expected responses.
- Performance Testing: Measures API performance and scalability under load.
- Security Testing: Validates API security and identifies potential vulnerabilities.
- Compatibility Testing: Ensures API compatibility with different devices, browsers, OS, and platforms.
- Error Handling Testing: Validates API error handling capabilities and appropriate error responses.
- Load Testing: Tests API performance and stability under high load conditions.
- Regression Testing: Validates API behavior after changes or updates to existing functionality.

10. What is Responsive Testing?

Responsive testing is a type of software testing that ensures a web application or website is responsive and adapts correctly to different screen sizes, devices, and orientations. It involves verifying that the application or website displays and functions properly on desktops, laptops, tablets, and mobile devices, providing a consistent user experience across different devices. Responsive testing helps identify and fix any issues related to responsiveness, layout, and design inconsistencies, ensuring that the application or website is accessible and usable on various screens and devices.

11. Which types of tools are available for Responsive Testing?

- Browser Developer Tools: Built-in tools in modern web browsers for inspecting and emulating different screen sizes and devices.
- Responsive Testing Frameworks: Libraries and frameworks for building responsive web applications.
- Cross-Browser Testing Tools: Tools for cross-browser and cross-device testing, including responsive testing.
- Online Responsive Testing Tools: Web-based tools for testing web applications on different devices and screen sizes.
- Automated Testing Tools: Tools for automating responsive testing of web applications across devices and screen sizes.

12. What is the full form of .ipa, .apk?

- The full form of .ipa is "iOS App Store Package"
- The full form of .apk is "Android Package Kit".

13. How to create step for to open the developer option mode ON?

- Step 1: Go to the "Settings" app on your Android device.
- Step 2: Scroll down and select "About phone" or "About device" option.
- Step 3: Look for "Build number" or "Build version" in the list of options.
- Step 4: Tap on "Build number" or "Build version" multiple times (usually 7 times or more) in quick succession.
- Step 5: You will see a message indicating that you are now a developer.
- Step 6: Go back to the main Settings page, and you should see "Developer options" or "Developer mode" listed as an option.
- Step 7: Tap on "Developer options" or "Developer mode" to access and configure various developer settings on your Android device.

Module-4(& Automation core testing)

1. Which components have you used in Load Runner?

- Virtual User Generator (VUGen): Used for recording and creating virtual user scripts that simulate user actions on an application.
- Controller: Used for managing and controlling load testing scenarios, including defining test scenarios, scheduling tests, and monitoring test results.
- Analysis: Used for analyzing and interpreting the results of load tests, including generating reports and graphs for performance analysis.

2. How can you set the number of Vusers in Load Runner?

- Launch the LoadRunner Controller.
- Create a new scenario or open an existing one.
- In the "Scenario" section, go to the "Vuser Groups" tab.
- Select the Vuser group for which you want to set the number of Vusers.
- In the "Number of Vusers" field, enter the desired number of Vusers that you want to simulate during the load test.
- You can also set additional options like ramp-up time, duration, and iterations for the Vusers in the same tab.
- Save the scenario.
- Start the load test by clicking the "Run" button in the Controller.

3. What is Correlation?

Correlation is a technique used in software testing to capture and replace dynamic values in server responses with parameterized values in subsequent requests. It helps ensure that test scripts accurately replicate user interactions with web applications by handling changing data, such as session IDs or user tokens. Correlation is commonly used in performance testing and web application testing to achieve realistic and accurate test results.

4. What is the process for developing a Vuser Script?

- Recording: User interactions with the application are recorded to generate a Vuser script.
- Enhancing: The recorded script is enhanced with parameterization, correlation, and data manipulation.
- Validating: The script is validated for accuracy and correctness.
- Customizing: The script is customized to simulate different scenarios or load levels.
- Execution: The script is executed as part of a load testing scenario.
- Maintenance: The script may require periodic maintenance to keep it up-to-date with changes in the application or system under test.

5. How Load Runner interacts with the application?

- Recording: Load Runner records user interactions with the application during script recording process.
- Playback: Load Runner replays the recorded script, sending requests and receiving responses from the application.
- Parameterization: Load Runner allows for parameterization of dynamic values in the script to simulate realistic user behavior.
- Correlation: Load Runner automatically correlates dynamic values in the script for accurate replay.
- Data Manipulation: Load Runner allows for data manipulation in the script, such as extracting data from responses and modifying request data.
- Monitoring: Load Runner captures performance metrics of the application during script execution.
- Analysis: Load Runner provides built-in analysis tools to interpret captured performance metrics and identify performance bottlenecks.

6. How many VUsers are required for load testing?

The number of Vusers (virtual users) required for load testing depends on factors such as application complexity, performance goals, and available resources. Typically, load testing involves simulating realistic user loads, and the number of Vusers needed varies for each application. It is important to carefully analyze and plan for the appropriate number of Vusers to ensure meaningful and accurate load testing results.

7. What is the relationship between Response Time and Throughput?

- As response time increases, throughput tends to decrease, and vice versa.
- Longer response time can lead to lower throughput or capacity to handle concurrent requests.
- Shorter response time can lead to higher throughput or capacity to handle concurrent requests.
- Achieving a balance between response time and throughput is important for optimal application performance in load testing.

8. What is the difference between hits/second and requests/second?

hits/second	requests/second
Hits/second measures the rate at which cached content is served from the server's cache.	Requests/second measures the rate at which the server receives incoming requests from clients.
Hits/second is an indicator of the server's efficiency in serving cached content.	Requests/second reflects the server's ability to handle incoming requests and process them in a timely manner.
Hits/second can be significantly higher than requests/second in scenarios with effective caching mechanisms.	Requests/second is a critical performance metric that indicates the server's capacity to handle concurrent requests.
Hits/second provides insight into the server's ability to handle content-heavy websites or applications with significant caching.	Requests/second can be affected by various factors such as server configuration, hardware resources, network latency, etc.
Hits/second may not accurately represent the overall server workload, as it only considers cached content and does not account for non-cached content, redirects, errors, etc.	Requests/second provides insights into the server's scalability, performance, and ability to handle high levels of concurrent traffic.

Module-5(& Selenium IDE)

1. What is Automation Testing?

Automation testing is the use of software tools and scripts to perform repetitive and complex testing tasks in a software application, without human intervention. It involves writing test scripts, creating test cases, and using automation tools to execute tests, validate results, and identify defects or issues in the software system. Automation testing helps to improve efficiency, accuracy, and repeatability of testing processes, and enables faster and more reliable software releases.

2. Which Are The Browsers Supported By Selenium Ide?

- Mozilla Firefox
- Google Chrome
- Microsoft Edge (Chromium-based)
- Safari (Limited support)
- Opera (Limited support)

3. What are the benefits of Automation Testing?

- Increased efficiency and time savings.
- Improved test coverage and accuracy.
- Faster feedback on software quality.
- Cost-effective in the long run.
- Scalability for large and complex applications.
- Effective for regression testing.
- Reusability of test scripts.
- Better traceability and documentation of test results.
- Enhanced software quality and reliability.

4. What are the advantages of Selenium?

- Open source and free to use.
- Supports multiple programming languages.
- Cross-platform compatibility for testing on different operating systems.
- Large community and active support.
- Extensible with various frameworks and tools for enhanced testing capabilities.

5. Why testers should opt for Selenium and not QTP?

Testers should choose Selenium over QTP (UFT) due to the advantages of open-source nature, support for multiple programming languages, cross-platform compatibility, flexibility, customization, and a large active community for support.