**SOFTWARE TESTING & AUTOMATION**
**ASSINGMENT**
**MODULE- 3**
**PREPARED BY : BHARAT V CHAUDHARY**

**1. What is RDBMS**

Relational Database Management System (RDBMS) is a more advanced version of a DBMS system that allows access to data in a more efficient way. It is used to store or manage only the data that are in the form of tables

The software used to store, manage, query, and retrieve data stored in a relational database is called a relational database management system (RDBMS)

The RDBMS provides an interface between users and applications and the database, as well as administrative functions for managing data storage, access, and performance

**2. What is SQL**

SQL stands for Structured Query Language SQL is a standard language for storing, manipulating and retrieving data in databases

SQL allows you to access and manipulate the databases

The use of SQL is in: MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres, and other database systems

**3. Write SQL Commands**

SQL commands are mainly categorized into five categories:

DDL - Data Definition Language
DML - Data Manipulation Language
TCL - Transaction Control Language
DQL - Data Query Language
DCL - Data Control Language

**4 What is priority?**

**Priority**

- If you are raising any bug for any application, how soon you want the developer to fix that bug is called priority.
- Priority is considered as customer's point of view. But priority can be set by the QA tester. Later on it can be changed by project manager.
- The high priority indicates that the bug to fix it first.

**P r i o r i t y**

**High Severity Low Priority**
Flipkart home page has a "Privacy" link at the bottom side of the page. That link is not working so it is high Severity-because link is not working. This link is used rarely so on low priority.

**Low Severity High Priority**
You can find the defect of spelling mistake of Flipkart as "Flipkrt". So this spelling mistake is on low severity, It does not impact the functionality or customer. But it impacts tothe brans as spell mistake so it is on high priority.

**High Severity High Priority**
Flipkarthas the "Add to cart" & "Buy now" option. When you want to purchase the product, but "Add to cart" or "Buy now" button not working. Then you can't use that functionality. It is on high severity because need to solve, and high priority because it has to besolve first.Core functionality is broken.

**Low Severity Low Priority**
Flipkart open with IE browser. "About us" information is overlapping text.

**5 What is severity?**

**Severity**

 The impact of Defect /bug on the customer business workflow is known as Severity.

**Critical**

The main functionality is not working.

For E.g Login by username & password, it will show a blank page then the next step is critical to deal with.

For E.g If you have sent Rs. 10000 to account "X". Then you login to check balance and it shows only Rs. 5000 as balance.

**Minor (Low)**

Defect is minor here. Usability of functionality not affected much but must be fixed.

For example, Runthe same application with both the browsers Firefox & Chrome. Look & feel can be different but not much affected to its usability.

**Moderate (Medium)**

Result is not coming consistency.

For example, Run the application 5 times, then 3 times its working fine but 2 times it shows some errors. So result is inconsistent and error is also inconsistent.

**Major (High)**

One of the functionality expected from the software which is not happening.

For example, If you booked Ola cab, and you got the message for cab booking confirmation. Your cab came but you still not receiving any OTP. Then how the next step of your ride will be possible?

For example, Youare sending any mail but you didn't get any pop up message for mail sent. You need to go to "Sent" to check that mail.

**Cosmetic**

Related to GUI issue like spell mistake, alignment problem.

For e.g Username & password textfield is not formatted with proper alignment.

**4. Difference between priority and severity.**

| | Priority | Severity |
|---|---|---|
| **1** | How urgent it is to fix a bug | How much a bug affects the software's functionality or end-user experience |
| **2** | Priority is subjective and may vary between users or businesses. | Severity is an objective measure that should be the same across organizations. |
| **3** | It takes into account factors like user impact | A QA engineer determines the severity level of a bug |
| **4** | It refers to how quickly the defect should be rectified. | Severity means how severe the defect is affecting the functionality. |

**5. Bug categories are…**

6. **What are the different Methodologies in Agile Development Model?**

There are three main methodologies. : Scrum,Kanban,XP

**SCRUM:** Scrum is a framework through which we build software product by following Agile principles.
SCRUM is an agile development method which concentrates particularly on how to manage tasks within a team based development environment.
Scrum includes group of people called a scrum team. Normally contains 5 to 9 members. Scrum team can involve the people like product owner, scrum master, DEV team, QA team etc.

**Roles:**

Product Owner

Scrum Master

Scrum Team

**Artifacts:**

Product Backlog

Sprint Backlog

Burn-down Charts

**Ceremonies:**

Spint Planning

Sprint Review

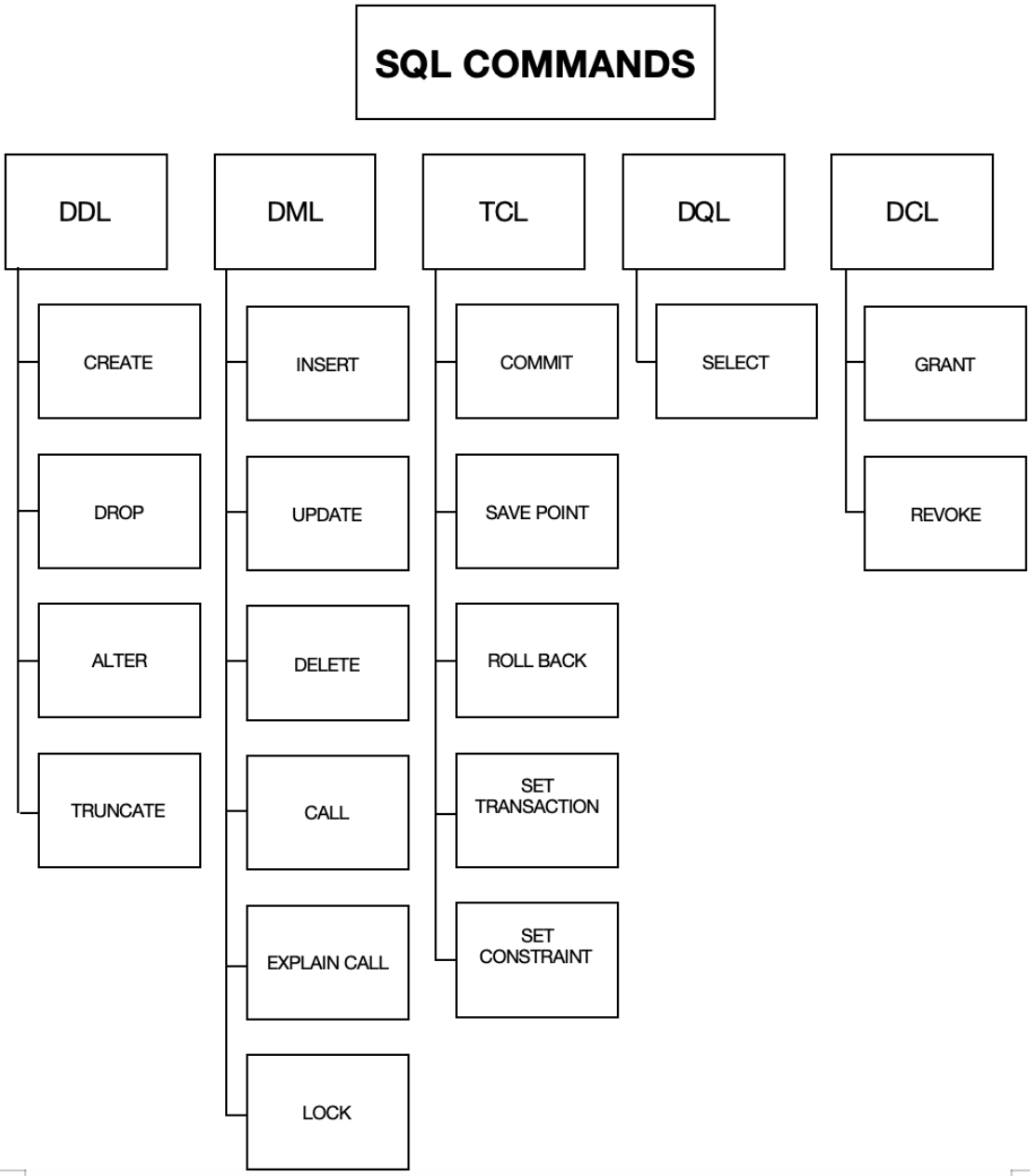
Sprint Retrospective

Daily Scrum Meeting

**KANBAN:**

Kanban is a very popular framework for development in the agile software development methodology.

- It provides a transparent way of visualizing the tasks and work capacity of a team.
- It mainly uses physical and digital boards to allow the team members to visualize the current state of the project they are working on.
-
- Kanban's meaning in Japanese is "billboards."
- The columns are nothing, but workflow states and cards are nothing but a demonstration of the actual task a team member is performing.
- Kanban should be used when you want to visualize your work, and you want to see the progress of your tasks visually.s




**7. What is join?**

A JOIN is a clause used in RDBMS to combine rows from two or more tables, based on a related column between them

The JOIN keyword merges two or more tables and creates a temporary image of the merged table, then according to the conditions provided, it extracts the required data from the image table, and once data is fetched, the temporary image of the merged tables is dumped

**8. Write types of JOIN**

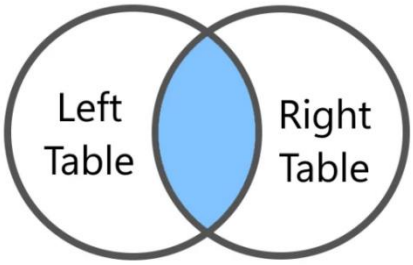
departments

| dept_id | dept_name |
|---|---|
| 10 | purchase |
| 20 | IT |
| 30 | finance |
| 40 | marketing |

employees

| emp_id | emp_name | dept_id |
|---|---|---|
| 1 | ram | 10 |
| 2 | shyam | 20 |
| 3 | rohan | 12 |
| 4 | soham | 30 |

**(INNER) JOIN:** A JOIN clause which returns records that have matching values in both tables




```
SELECT e.emp_id, e.emp_name, e.dept_id, d.dept_name FROM employees e
JOIN departments d
ON e.dept_id = d.dept_id;
```

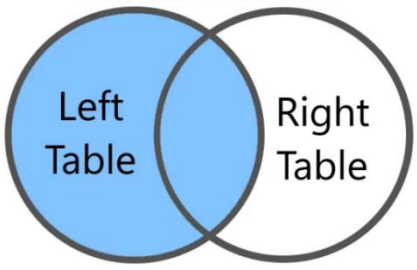
OR

```
SELECT e.emp_id, e.emp_name, e.dept_id, d.dept_name
FROM employees e
INNER JOIN departments d
ON e.dept_id = d.dept_id;
```

**Result:**

| emp_id | emp_name | dept_id | dept_name |
|---|---|---|---|
| 1 | ram | 10 | purchase |
| 2 | shyam | 20 | IT |
| 4 | soham | 40 | marketing |

**LEFT (OUTER) JOIN:** A JOIN clause which returns all records from the left table, and the matched records from the right table

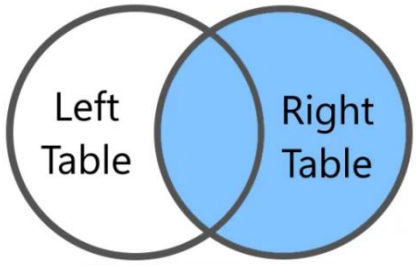



```
SELECT e.emp_id, e.emp_name, e.dept_id, d.dept_name FROM employees e
LEFT OUTER JOIN departments d ON e.dept_id = d.dept_id;
```

**Result:**

| emp_id | emp_name | dept_id | dept_name |
|---|---|---|---|
| 1 | ram | 10 | purchase |
| 2 | shyam | 20 | IT |
| 4 | soham | 40 | marketing |
| 3 | rohan | 50 | *NULL* |

**RIGHT (OUTER) JOIN:** A JOIN clause which returns all records from the right table, and the matched records from the left table

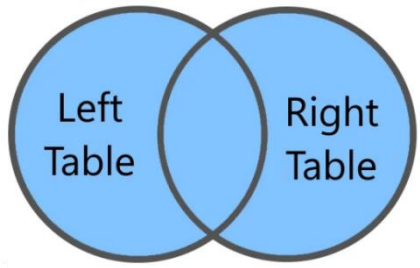



```
SELECT e.emp_id, e.emp_name, e.dept_id, d.dept_name FROM employees e
RIGHT OUTER JOIN departments d ON e.dept_id = d.dept_id;
```

**Result:**

| emp_id | emp_name | dept_id | dept_name |
|---|---|---|---|
| 1 | ram | 10 | purchase |
| 2 | shyam | 20 | IT |
| 4 | soham | 40 | marketing |
| *NULL* | *NULL* | *NULL* | finance |

**FULL (OUTER) JOIN:** A JOIN clause which returns all records when there is a match in either left or right table




```
SELECT e.emp_id, e.emp_name, e.dept_id, d.dept_name FROM employees e
FULL OUTER JOIN departments d ON e.dept_id = d.dept_id;
```

OR

```
SELECT e.emp_id, e.emp_name, e.dept_id, d.dept_name FROM employees e
LEFT OUTER JOIN departments d
ON e.dept_id = d.dept_id
UNION
SELECT e.emp_id, e.emp_name, e.dept_id, d.dept_name FROM employees e
RIGHT OUTER JOIN departments d
```

```
ON e.dept_id = d.dept_id;
```

**Result:**

| emp_id | emp_name | dept_id | dept_name |
|---|---|---|---|
| 1 | ram | 10 | purchase |
| 2 | shyam | 20 | IT |
| 4 | soham | 40 | marketing |
| 3 | rohan | 50 | *NULL* |
| *NULL* | *NULL* | *NULL* | finance |

**9. Howmany types of constraints in SQL. Describe them**

The following types of constraints are commonly used in SQL:

**NOT NULL:** Ensures that a column cannot have a NULL value

```
CREATE TABLE department
(
   dept_id INT NOT NULL,
   dept_name VARCHAR (20) NOT NULL,
       branch VARCHAR (20) NOT NULL
);

INSERT INTO department VALUES (1,'Purchase','Delhi');
INSERT INTO department VALUES (2,'Legal','Mumbai');
INSERT INTO department VALUES (3,'Finance','Chennai');
INSERT INTO department VALUES (4,'HR','Kolkata');
```

| dept_id | dept_name | branch |
|---|---|---|
| 1 | Purchase | Delhi |
| 2 | Legal | Mumbai |
| 3 | Finance | Chennai |
| 4 | HR | Kolkata |

**UNIQUE:** Ensures that all values in a column are different

```
CREATE TABLE department
(
   dept_id INT NOT NULL UNIQUE,
   dept_name VARCHAR (20) NOT NULL,
       branch VARCHAR (20) NOT NULL
);

INSERT INTO department VALUES (1,'Purchase','Delhi');
INSERT INTO department VALUES (2,'Legal','Mumbai');
INSERT INTO department VALUES (3,'Finance','Chennai');
INSERT INTO department VALUES (4,'HR','Kolkata');
```

| dept_id | dept_name | branch |
|---|---|---|
| 1 | Purchase | Delhi |
| 2 | Legal | Mumbai |
| 3 | Finance | Chennai |
| 4 | HR | Kolkata |

**PRIMARY KEY:** A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table

```
CREATE TABLE department
(
   dept_id INT PRIMARY KEY,
   dept_name VARCHAR (20),
       branch VARCHAR (20)
);

INSERT INTO department VALUES (1,'Purchase','Delhi');
INSERT INTO department VALUES (2,'Legal','Mumbai');
INSERT INTO department VALUES (3,'Finance','Chennai');
INSERT INTO department VALUES (4,'HR','Kolkata');
```

| dept_id | dept_name | branch |
|---|---|---|
| 1 | Purchase | Delhi |
| 2 | Legal | Mumbai |
| 3 | Finance | Chennai |
| 4 | HR | Kolkata |

**FOREIGN KEY:** Prevents actions that would destroy links between tables

```
CREATE TABLE employee
(
   emp INT,
   emp_name VARCHAR (20),
   salary VARCHAR (20),
   dept_id INT,

   PRIMARY KEY (emp),
       FOREIGN KEY (dept_id) references department(dept_id)
);

INSERT INTO employee VALUES (101,'ram',30000,1);
INSERT INTO employee VALUES (102,'shyam',32000,2);
INSERT INTO employee VALUES (103,'soham',35000,3);
INSERT INTO employee VALUES (104,'rohan',37000,4);
```

| emp | emp_name | salary | dept_id |
|---|---|---|---|
| 101 | ram | 30000 | 1 |
| 102 | shyam | 32000 | 2 |
| 103 | soham | 35000 | 3 |
| 104 | rohan | 37000 | 4 |

**CHECK:** Ensures that the values in a column satisfies a specific condition

```
CREATE TABLE employees
(
emp_id INT PRIMARY KEY,
emp_name VARCHAR (100),
age INT CHECK (age >= 18),
salary INT CHECK (salary > 0)
);

INSERT INTO employees (emp_id, emp_name, age, salary) VALUES (1, 'samar', 30, 50000);
```

| emp_id | emp_name | age | salary |
|---|---|---|---|
| 1 | samar | 30 | 50000 |

Here, age column has a CHECK constraint to ensure that only values 18 or older can be inserted and salary column has a CHECK constraint to ensure that the salary is greater than 0

**DEFAULT:** Sets a default value for a column if no value is specified

```
CREATE TABLE products
(
   product_id INT PRIMARY KEY,
   product_name VARCHAR(255),
       quantity INT DEFAULT 0,
       price INT (10) DEFAULT 20
);
INSERT INTO products (product_id, product_name) VALUES (1, 'Laptop');
```

| product_id | product_name | quantity | price |
|---|---|---|---|
| 1 | Laptop | 0 | 50000 |

Here, quantity and price are set default initially then added afterwards

**10. Difference between RDBMS vs DBMS**

| RDBMS | DBMS |
|---|---|
| In RDBMS, data stored is in table format | In DBMS data stored is in the file format |
| In RDBMS, multiple data elements are accessible together | In DBMS, individual access of data elements |
| In RDBMS, data in the form of a table are linked together | In DBMS, there is no connection between data |
| RDBMS supports distributed database | In DBMS, there is no support for distributed database |
| In RDBMS, data is stored in a large amount | In DBMS, data stored is a small quantity |
| RDBMS supports multiple users | DBMS supports a single user |
| In RDBMS, the software and hardware requirements are higher | In DBMS, the software and hardware requirements are low |
| Example: Oracle, SQL Server | Example: XML, Microsoft Access |

**11. What is API Testing**

API testing is a software testing method that verifies the functionality, security, performance, and reliability of an application programming interface (API)

API is the mediator which helps to applications to communicate with each other. It is kind of logic written by developers using any programming language to perform something

API is a Software Interface that allows two applications to interact with each other without any user intervention

Testing the business logic of any application is called API, QA will test the same logic and called API testing

API testing is a part of back end testing like database


**12. Types of API Testing**

**Open APIs:** These types of APIs are publicly available to use. It has also not given any restriction to use them. So, they are also known as Public APIs. e.g. OAuth APIs from Google

**Partner APIs:** Specific rights or licenses needed to access this type of API because they are not available to the public

**Internal APIs:** Internal or private. These APIs are developed by companies to use in their internal systems It helps you to enhance the productivity of your teams

**13. What is Responsive Testing?**

To check the responsiveness of our website on multiple devices is simply called responsive testing

When user switches to one device to another device the contents of responsive websites adapt themselves according to the device UI, resolution etc factors

Responsive testing involves how a website or web application looks and behaves on different devices, screen sizes, and resolutions

The goal of responsive testing is to ensure that the website or web application can be used effectively on various devices, including desktops, laptops, tablets, and smartphones

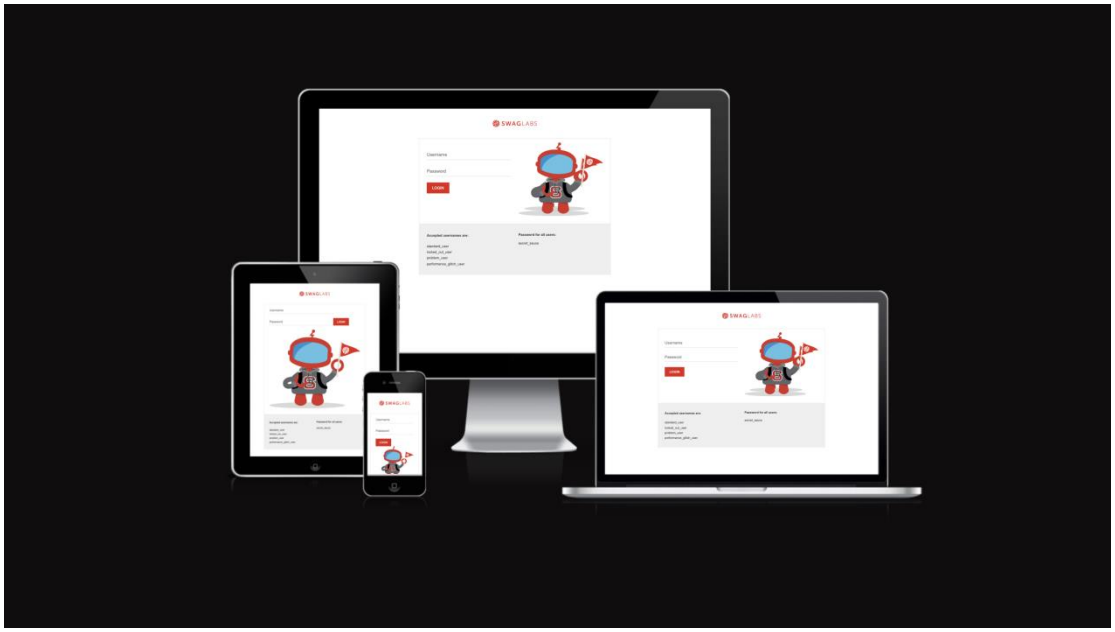
**14. Which types of tools are available for Responsive Testing**

Lembda Testing
Google Resizer
am I responsive
Pixel tuner

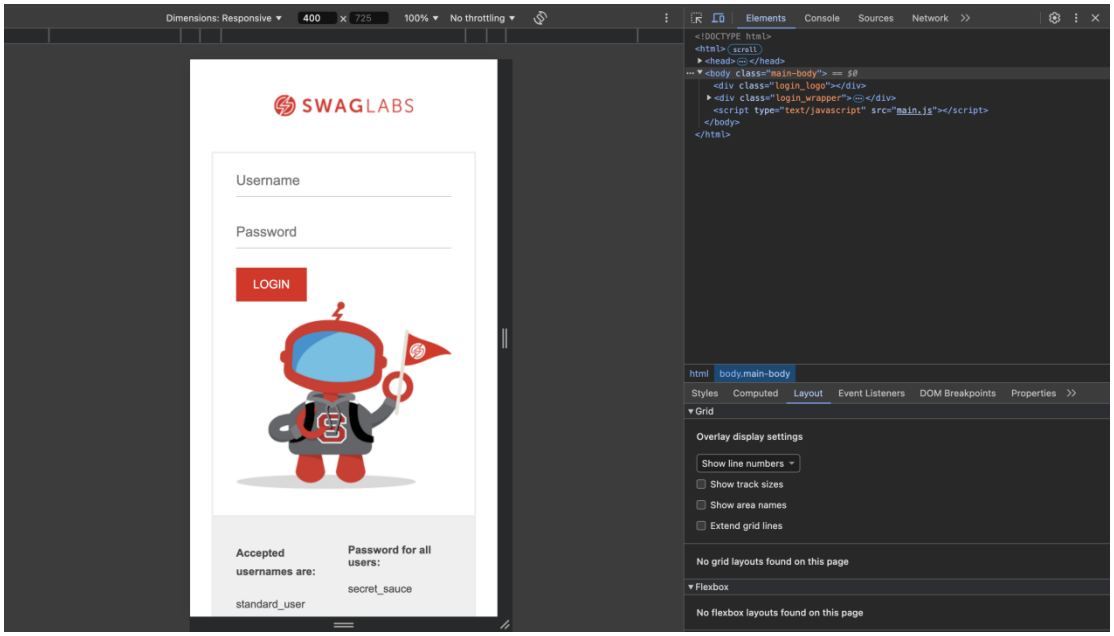
e..g., https://ui.dev/amiresponsive

Enter your URL in this box to search


we get different response of the website on different devices



For Google chrome, you can right click in anywhere in the browser and select “Inspect”, you can check the screen where you can set the screen with multiple dimensions

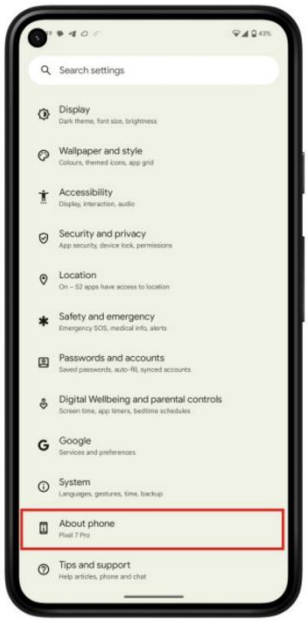



**15. What is the full form of .ipa, .apk**

**.ipa** stands for iOS package, App international phonetic alphabet
**.apk** stands for Android Application Package

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

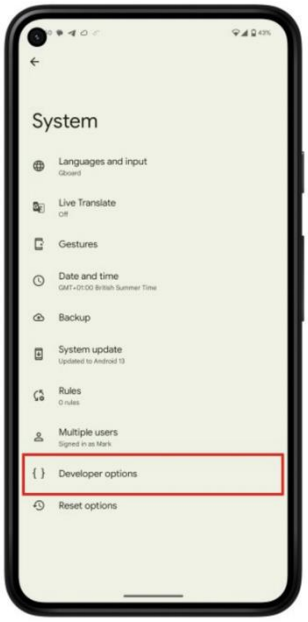
The following example uses a Google Pixel 7 Pro running Android 13 shows steps to open the developer option mode ON



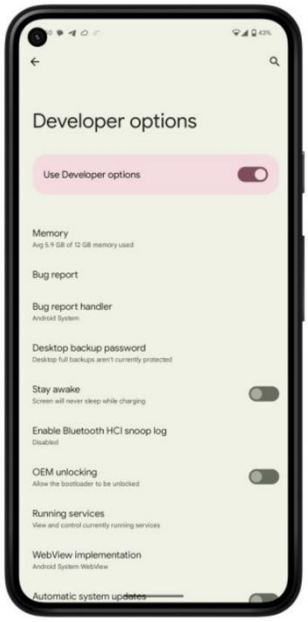

Step: 1


Step: 2


Step: 3 & 4




Step: 5




Step: 6

**Step 1:** Go to *Settings* > *About phone*

**Step 2:** Scroll down to *Build number*

**Step 3:** Tap *Build number* seven times. After the first few taps, you should see the steps counting down until you unlock the developer options. You may also have to tap in your PIN for verification

**Step 4:** Once developer options are activated, you will see a message that reads, *You are now a developer*

**Step 5:** Go back to the *Settings* pane and head to *System*, where you will now find *Developer options* as an entry

**Step 6:** Tap it and toggle the switch on if it is not already, and from there, you can proceed to make adjustments to your phone

You can unlock the developer options on any Android smartphone or tablet by locating the build number in your *Settings* menu and tapping it multiple times. However, the exact location of the aforementioned build number may differ depending on your phone's manufacturer