**Manual\_Assignment\_3**

**Testing On Live Application**

**Submitted by Kuldeep Parmar**

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

1. **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

1. **Write SQL Commands**

SQL commands are mainly categorized into five categories:

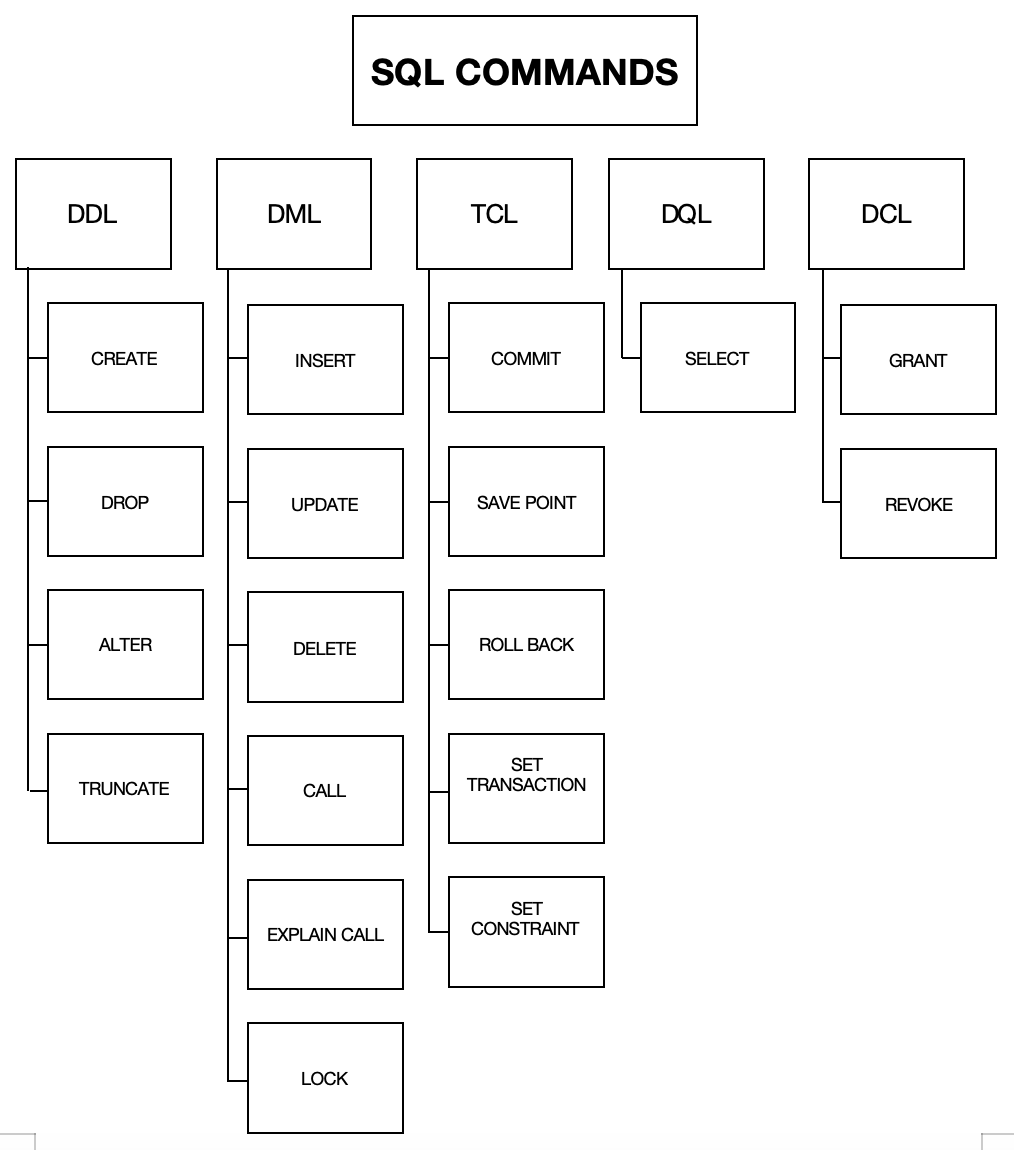
DDL - Data Definition Language

DML - Data Manipulation Language

TCL - Transaction Control Language

DQL - Data Query Language

DCL - Data Control Language

****

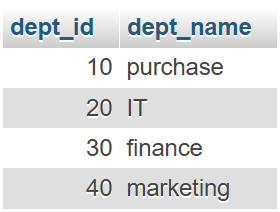
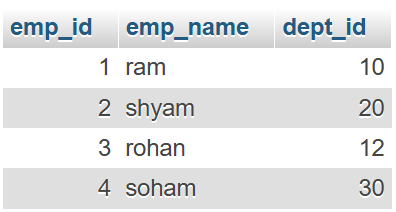
1. **What is join?**

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

1. **Write types of JOIN**

departments employees

** **

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

**Inner Join**

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

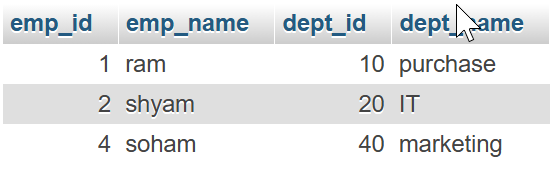
SELECT e.emp\_id, e.emp\_name, e.dept\_id, d.dept\_name

FROM employees e

INNER JOIN departments d

ON e.dept\_id = d.dept\_id;

**Result:**

****

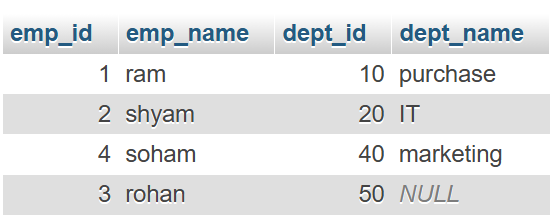
**LEFT (OUTER) JOIN:** A JOIN clause which returns all records from the left table, and the matched records from the right table

**Left Join**

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:**

****

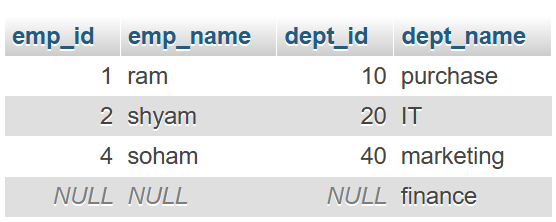
**RIGHT (OUTER) JOIN:** A JOIN clause which returns all records from the right table, and the matched records from the left table

**Right Join**

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:**

****

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

**Full Join**

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

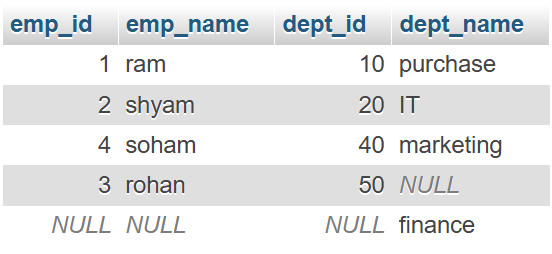
UNION

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:**

****

1. **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

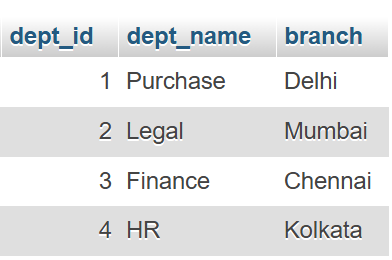
);

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');



**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

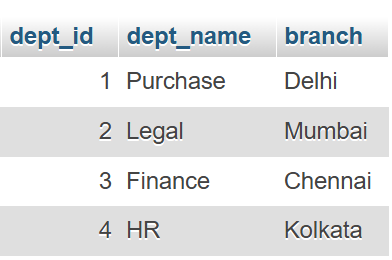
);

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');



**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)

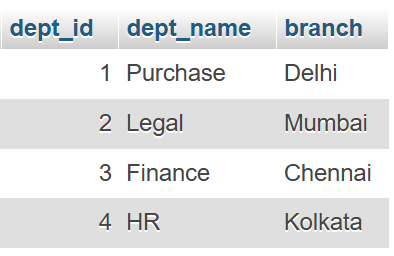
);

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');

****

**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)

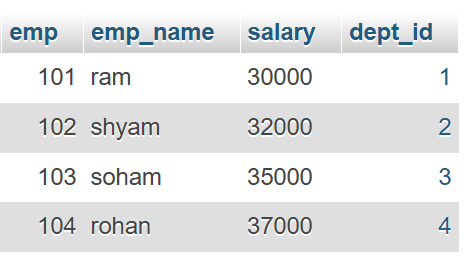
);

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);



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

CREATE TABLE employees

(

emp\_id INT PRIMARY KEY,

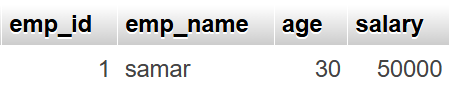
emp\_name VARCHAR (100),

age INT CHECK (age >= 18),

salary INT CHECK (salary > 0)

);

INSERT INTO employees (emp\_id, emp\_name, age, salary) VALUES (1, 'samar', 30, 50000);

****

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,

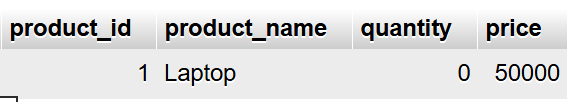
product\_name VARCHAR(255),

quantity INT DEFAULT 0,

price INT (10) DEFAULT 20

);

INSERT INTO products (product\_id, product\_name) VALUES (1, 'Laptop');

****

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

1. **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 |

1. **What is API Testing**

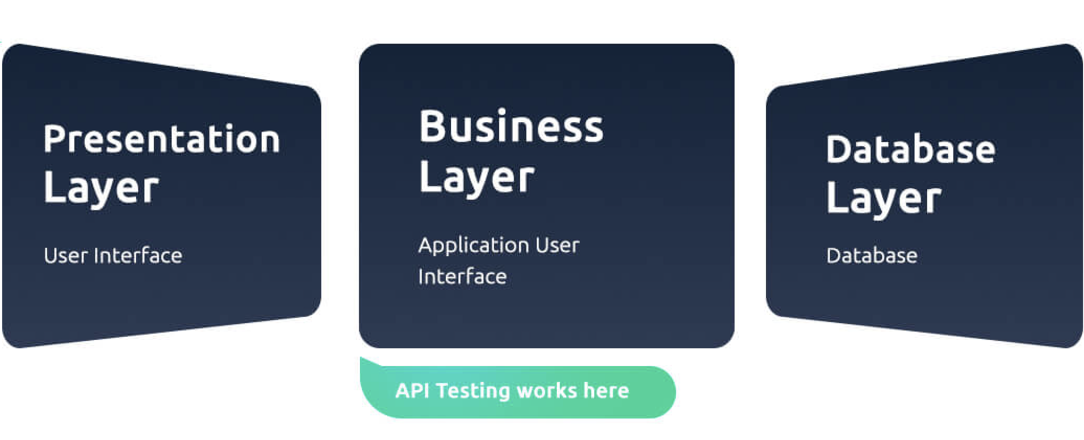
API testing is a software testing method that verifies the functionality, security, performance, and reliability of an application programming interface (API)

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

****

1. **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

1. **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

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

Lembda Testing

Google Resizer

am I responsive

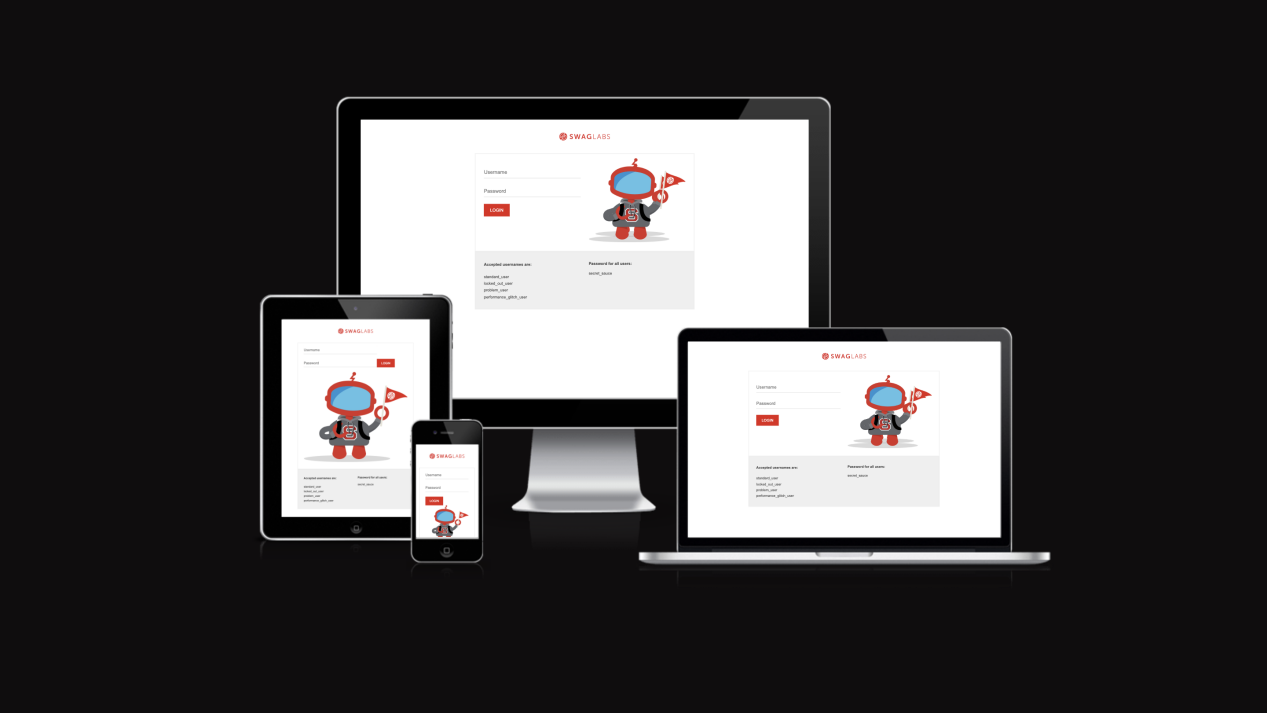
Pixel tuner

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

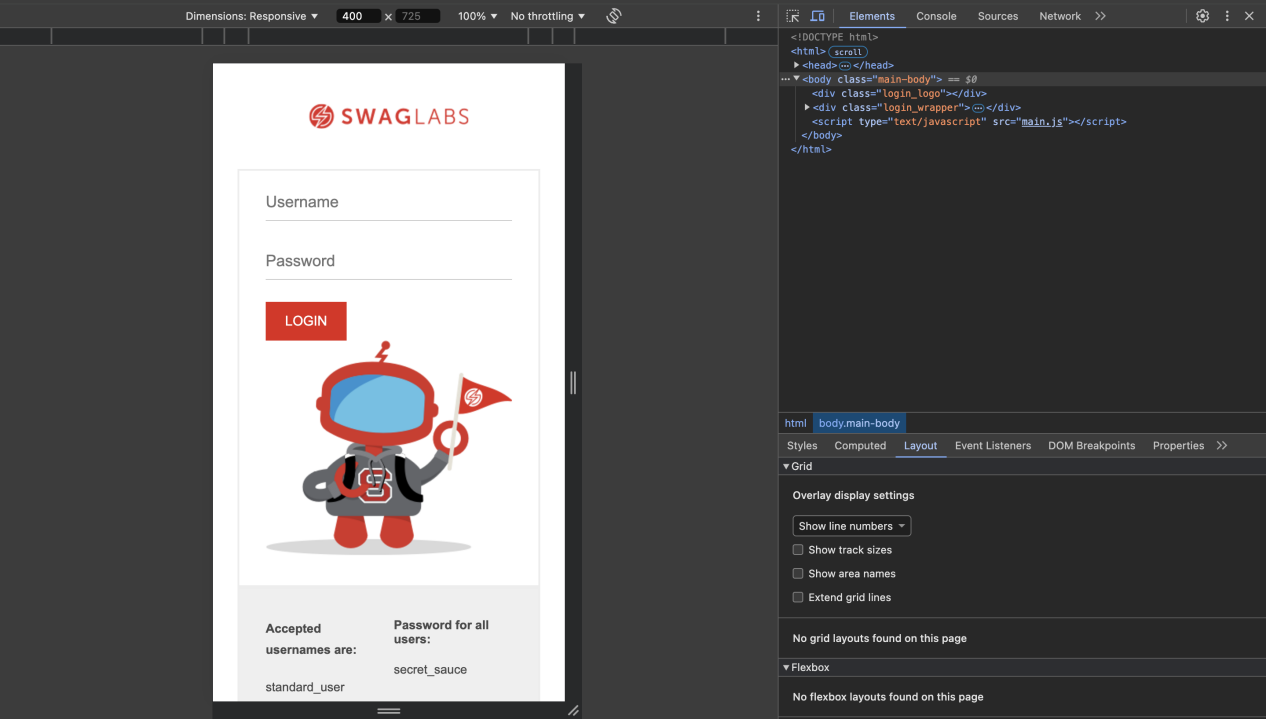
Enter your URL in this box to search



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

****

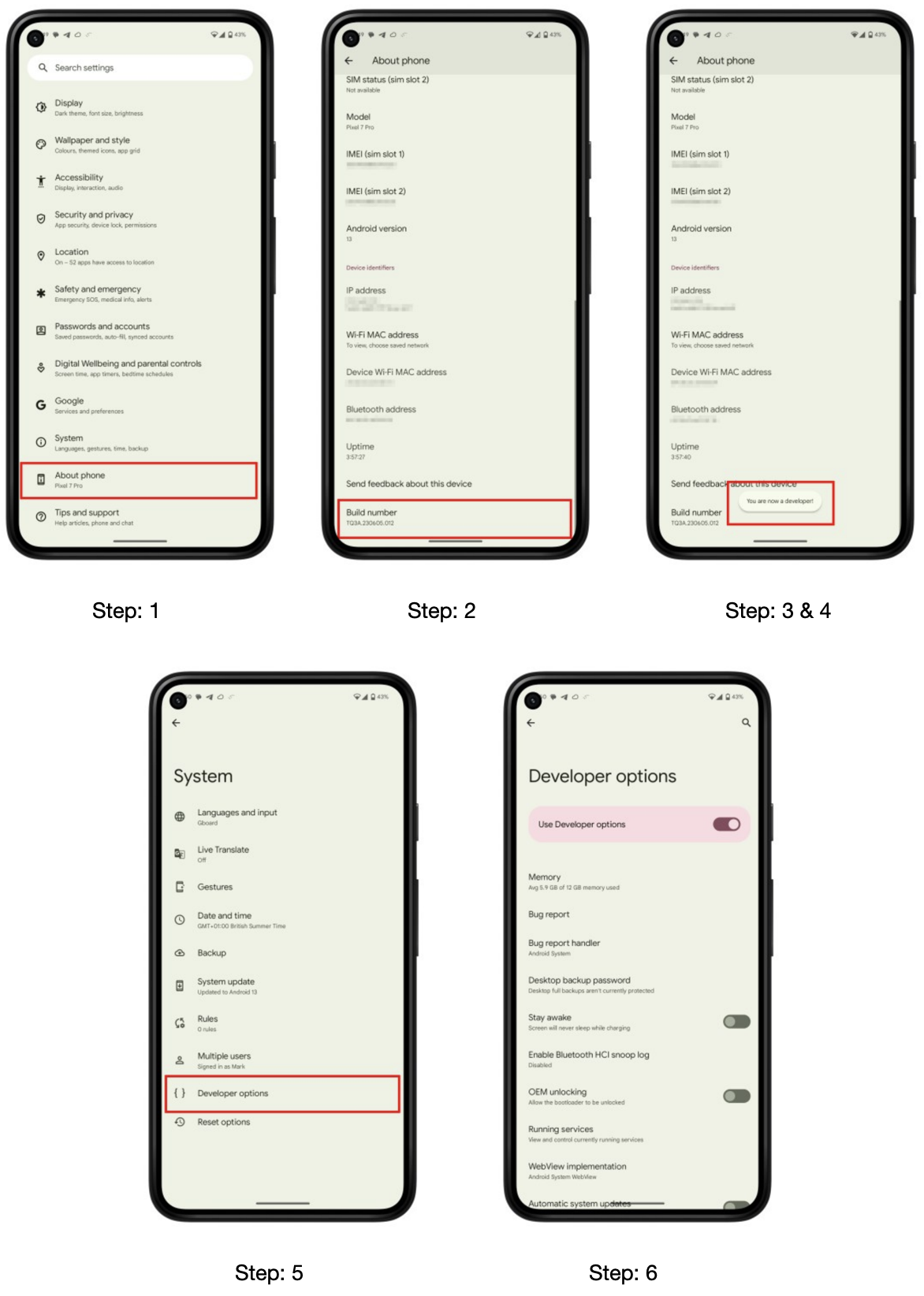
1. **What is the full form of .ipa, .apk**

**.ipa** stands for iOS package, App international phonetic alphabet

**.apk** stands for Android Application Package

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

The following example uses a [Google Pixel 7 Pro](https://www.digitaltrends.com/mobile/google-pixel-7-pro-review/) running [Android 13](https://www.digitaltrends.com/mobile/android-13-phones-list/) shows steps to open the developer option mode ON



****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