**Experiment : 10**

# PRACTICAL STATEMENT OF PRACTICAL:

Write a program to show the usage of frames in HTML.

# OBJECTIVE OF PRACTICAL

To show the implementation of frames in html.

# THEORY

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

# IMPLEMENTATION

<html>

<head>

    <title>Frames</title>

</head>

<frameset rows="50%,50%">

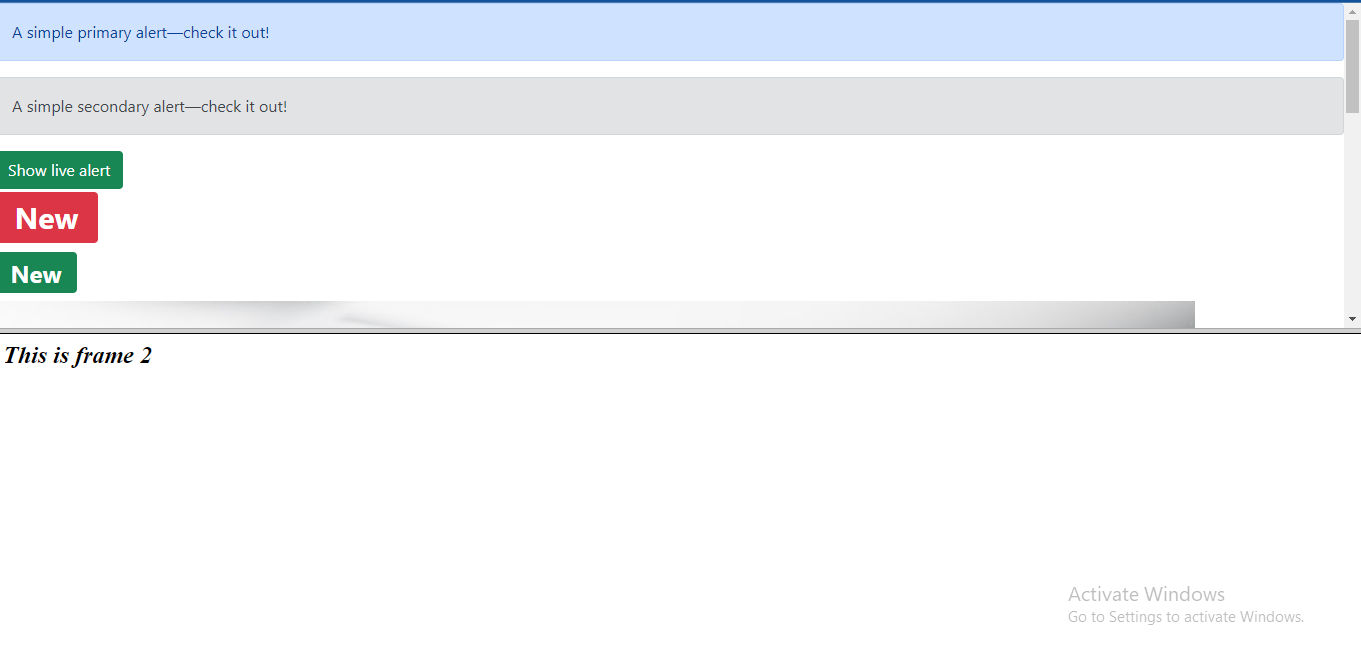
    <frame name="frame1" src="myOwn.html">

        <frame name="frame2" src="frame2.html">

</frameset>

</html>

# RESULT /OUTPUT



**Experiment : 11**

1. **PRACTICAL STATEMENT OF PRACTICAL:**

Write a program to print date using JavaScript.

# OBJECTIVE OF PRACTICAL

To show today’s date on webpage using JavaScript

# THEORY

JavaScript is a scripting language designed primarily for adding interactivity to Web pages and creating Web applications. The language was first implemented by Netscape Communications Corp. in Netscape Navigator 2 beta (1995). JavaScript is different from the Java language (developed in the 1990s at Sun Microsystems). However, the two languages can interoperate well. Client-side JavaScript programs, or scripts, can be embedded directly in HTML source of Web pages. (Note: There is also server-side JavaScript, but it's beyond the scope of this FAQ collection.) Depending on the Web developer's intent, script code may run when the user opens the Web page, clicks or drags some page element with the mouse, types something on the keyboard, submits a form, or leaves the page.

JavaScript is an object-oriented language with prototypal inheritance.

The language supports several built-in objects, and programmers can create or delete their own objects. Prototypal inheritance makes JavaScript very different from other popular programming languages such as C++, C#, or Java featuring classes and classical inheritance. JavaScript does not have classes in the C++ or Java sense. In JavaScript, objects can inherit properties directly from each other, forming the object prototype chain.

JavaScript is widely supported. It is available in the following browsers:

• Netscape Navigator (beginning with version 2.0)

• Microsoft Internet Explorer (beginning with version 3.0)

• Firefox

• Opera

• Google Chrome

Any other browser whose vendor licensed or implemented JavaScript.

Date objects are created with the new Date() constructor.

# IMPLEMENTATION

<!DOCTYPE html>

<html>

<body>

    <h1>Today's Date</h1>

    <p>Click the button to display the date.</p>

    <p id="demo"></p>

    <script>

        var d = new Date();

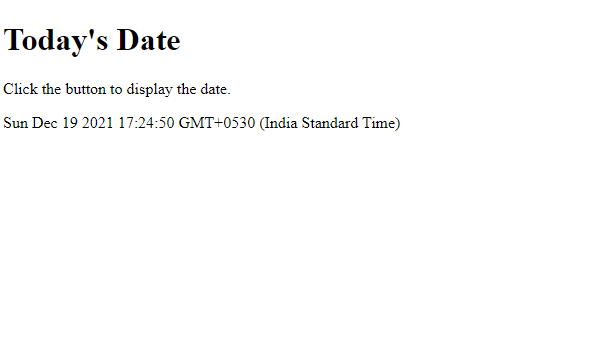
            document.getElementById("demo").innerHTML = d;

    </script>

</body>

</html>

1. **RESULT /OUTPUT**

****

**Experiment : 12**

# PRACTICAL STATEMENT OF PRACTICAL:

Create a style sheet in CSS/ XML & display the document in internet explorer

# OBJECTIVE OF PRACTICAL

To print and style XML content on webpage using JavaScript

# THEORY

Extensible Markup Language (**XML**) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

# IMPLEMENTATION :

# Hello.xml

<!DOCTYPE student[

<!ELEMENT student\_information (ad+)>

<!ELEMENT ad (usn,name,collegename,branch,year,email)>

<!ELEMENT usn (#PCDATA)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT collegename (#PCDATA)>

<!ELEMENT branch (#PCDATA)>

<!ELEMENT year (#PCDATA)>

<!ELEMENT email (#PCDATA)>

]>

<?xml-stylesheet type="text/css" href="student.css"?>

<student\_information>

<h3>STUDENT DATABASE</h3>

<h2>Rohan Sharma</h2>

<ad><label>rno:<usn> 1 </usn></label></ad>

<ad><label>Age:<name> 25 </name></label></ad>

<ad><label>College Name:<college>ABESEC </college></label></ad>

<ad><label>Branch:<branch> Ceit </branch></label></ad>

<ad><label>Year of Joining:<year> 2019 </year></label></ad>

<ad><label>Email -id:<email> rohan@gmail.com </email></label></ad>

<h2>Samarpit Dua</h2>

<ad><label>usn:<usn> 2  </usn></label></ad>

<ad><label>Age:<name> 20 </name></label></ad>

<ad><label>College Name:<college> ABESEC  </college></label></ad>

<ad><label>Branch:<branch> CEIT </branch></label></ad>

<ad><label>Year of Joining:<year> 2018 </year></label></ad>

<ad><label>Email -id:<email> samarpit@gmail.com </email></label></ad>

<h2>Roshan Lal</h2>

<ad><label>usn:<usn> 6  </usn></label></ad>

<ad><label>Age:<name> 19 </name></label></ad>

<ad><label>College Name:<college> ABESEC </college></label></ad>

<ad><label>Branch:<branch> CS </branch></label></ad>

<ad><label>Year of Joining:<year> 2018 </year></label></ad>

<ad><label>Email -id:<email> roshan@gmail.com </email></label></ad>

</student\_information>

**Student.css**

ad {

    display: block;

    margin-top: 15px;

    color: blue;

    font-size: 13pt;

}

usn {

    color: red;

    font-size: 12pt;

    margin-left: 15px;

}

name {

    color: red;

    font-size: 12pt;

    margin-left: 15px;

}

college {

    color: red;

    font-size: 12pt;

    margin-left: 15px;

}

branch {

    color: red;

    font-size: 12pt;

    margin-left: 15px;

}

year {

    color: red;

    font-size: 12pt;

    margin-left: 15px;

}

email {

    color: red;

    font-size: 12pt;

    margin-left: 15px;

}

h3 {

    color: red;

    font-size: 18pt;

}

h2 {

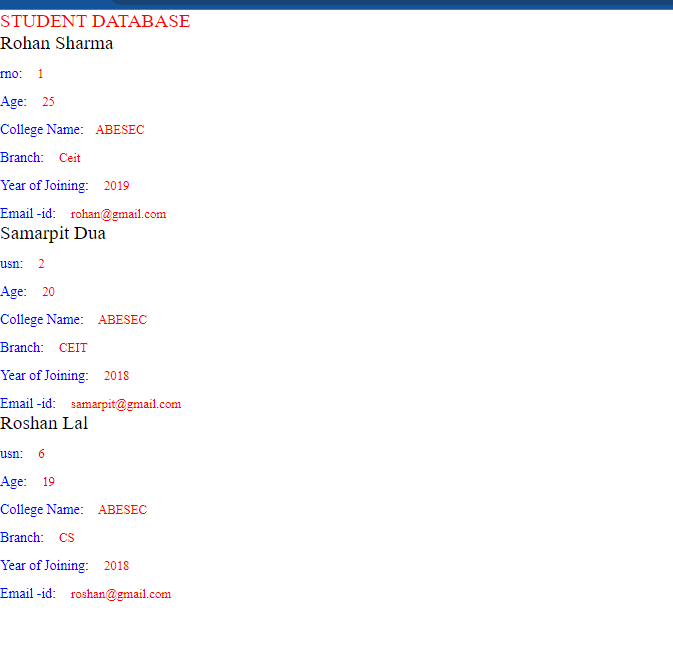
    display: block;

    color: black;

    font-size: 18pt;

}

1. **RESULT /OUTPUT**

****

**Experiment : 13**

# PRACTICAL STATEMENT OF PRACTICAL:

Write a program to display Fibonacci Series in Java Script

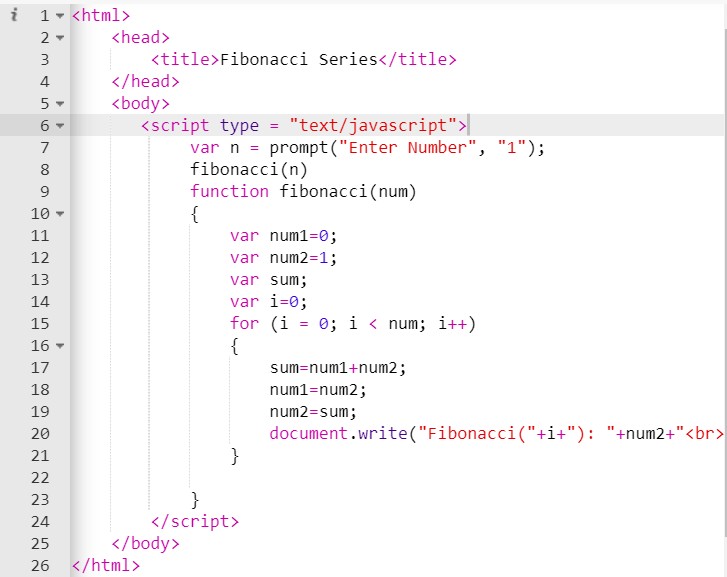
# OBJECTIVE OF PRACTICAL

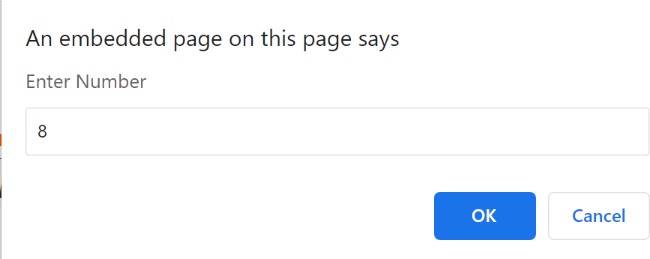
To print Fibonacci Series on webpage using JavaScript

# THEORY

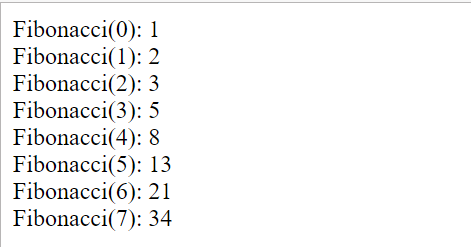
JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

# IMPLEMENTATION





1. **RESULT /OUTPUT**



**Experiment : 14**

## PRACTICAL STATEMENT OF PRACTICAL:

Write a program to Sum and Multiply two numbers using JavaScript.

## OBJECTIVE OF PRACTICAL:

Make a program to Sum and Multiply two numbers using JavaScript.

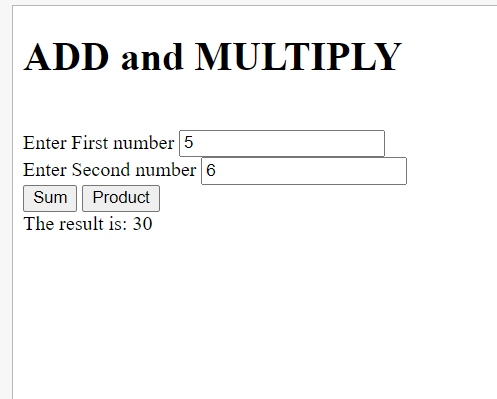
## THEORY:

<script>: it is used to embed a client-side script using java-script. Function: used to define a function in Java-Script.

## IMPLEMENTATION:



1. Result /Output:



**Experiment : 15**

## PRACTICAL STATEMENT OF PRACTICAL:

Create validation Form in JavaScript

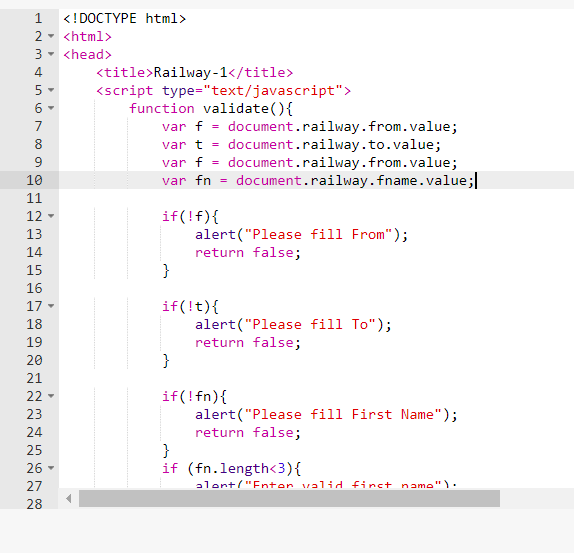
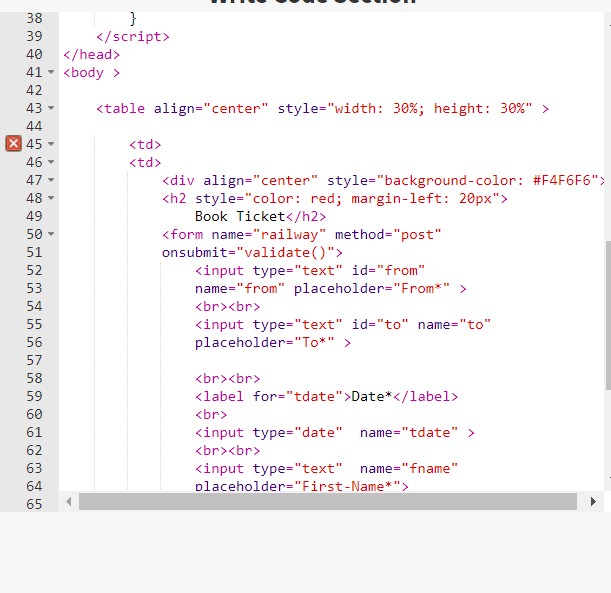
## OBJECTIVE OF PRACTICAL:

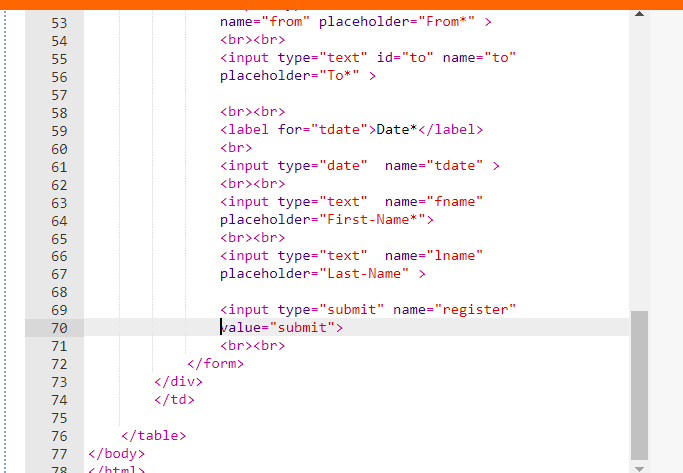
Create validation Form in JavaScript.

## THEORY:

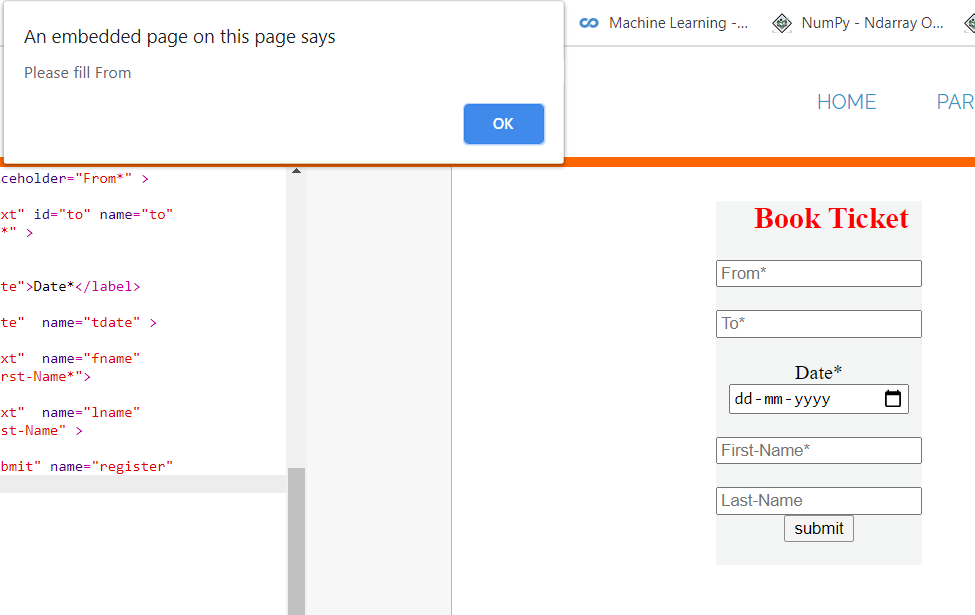
<script>: it is used to embed a client-side script using java-script. Function: used to define a function in Java-Script.

## IMPLEMENTATION:



1. Result /Output:



Practical Name : Write a program to change content of web page using AJAX. Practical No : 16

**Experiment : 16**

## PRACTICAL STATEMENT OF PRACTICAL:

Write a program to change content of web page using AJAX.

## OBJECTIVE OF PRACTICAL:

To display the working of AJAX using web server.

## THEORY:

**AJAX:** Asynchronous JavaScript and XML. AJAX is a technique for creating fast and dynamic web pages. AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

* AJAX is not a new programming language, but a new way to use existing standards.
* AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.
* AJAX is a technique for creating fast and dynamic web pages.
* AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
* Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.
* Examples of applications using AJAX: Google Maps, Gmail, Youtube, and Facebook tabs.
* In essence, Ajax is an efficient way for a web application to handle user interactions with a web page - a way that reduces the need to do a page refresh or full page reload for every user interaction. This enables rich behavior (similar to that of a desktop application or plugin-based web application) using a browser. Ajax interactions are handled asynchronously in the background. As this happens, a user can continue working with the page. Ajax interactions are initiated by JavaScript code. When the Ajax interaction is complete, JavaScript updates the HTML source of the page. The changes are made immediately without requiring a page refresh. Ajax interactions can be used to do things such as validate form entries (while the user is entering them) using server-side logic, retrieve detailed data from the server, dynamically update data on a page, and submit partial forms from the page.
* AJAX is Based on Internet Standards
* AJAX is based on internet standards, and uses a combination of:
  + XMLHttpRequest object (to exchange data asynchronously with a server)
  + JavaScript/DOM (to display/interact with the information)
  + CSS (to style the data)
  + XML (often used as the format for transferring data)
* Google Suggest
* AJAX was made popular in 2005 by Google, with Google Suggest.
* Google Suggest is using AJAX to create a very dynamic web interface: When you start typing in Google's search box, a JavaScript sends the letters off to a server and the server returns a list of suggestions.
* As you type in the search box, you can find information quickly by seeing searches that might be similar to the one you're typing. For example, as you start to type [google], you may see searches for other popular google-related searches.

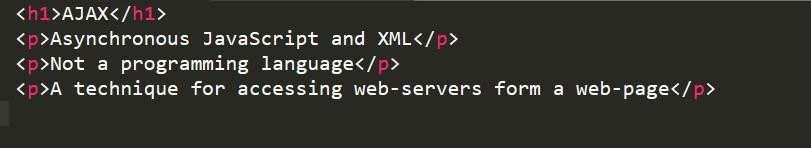
Syntax for creating an XMLHttpRequest object:

variable=new XMLHttpRequest();

Old versions of Internet Explorer (IE5 and IE6) uses an ActiveX Object: variable=new ActiveXObject("Microsoft.XMLHTTP");

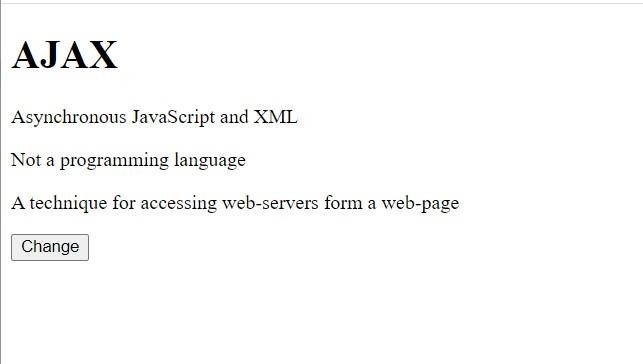
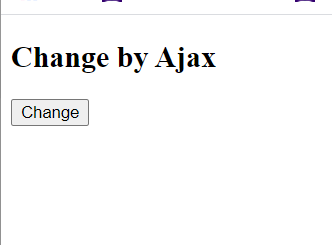
## IMPLEMENTATION:





Practical Name : Write a program to change content of web page using AJAX.

1. Result /Output:



Practical Name : XMLHttpRequest Object Practical No : 17

**Experiment : 17**

1. **PRACTICAL STATEMENT OF PRACTICAL:**

Write a program to create XMLHttpRequest.

1. **OBJECTIVE OF PRACTICAL**

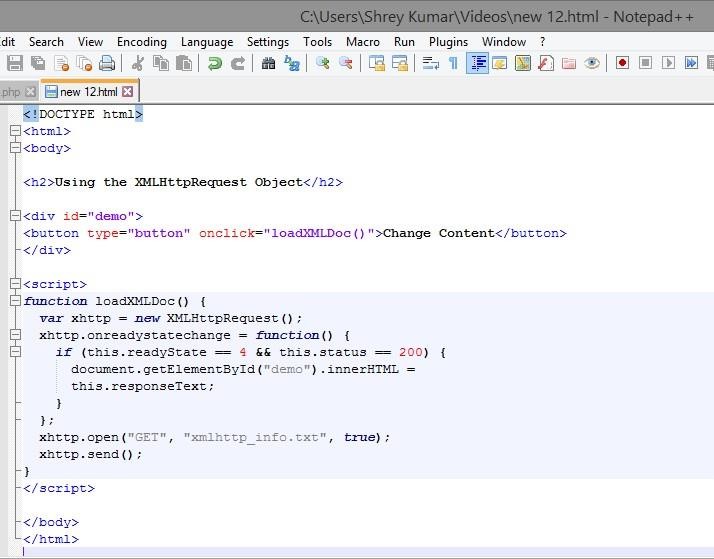
Write a program to create XMLHttpRequest.

1. **THEORY**

The XMLHttpRequest object can be used to request data from a web server. The XMLHttpRequest object is **a developers dream**, because you can:

* + Update a web page without reloading the page
  + Request data from a server - after the page has loaded
  + Receive data from a server - after the page has loaded
  + Send data to a server - in the background

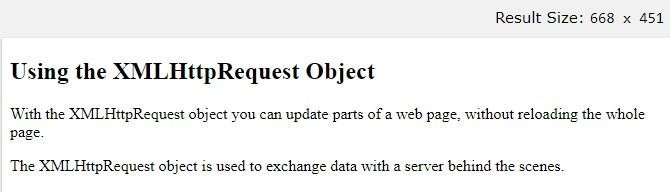
## IMPLEMENTATION



Practical Name : XMLHttpRequest Object Practical No : 11

## Result /Output





Practical Name: Write a program to connect and insert to database. Practical No : 18

**Experiment : 18**

## PRACTICAL STATEMENT OF PRACTICAL:

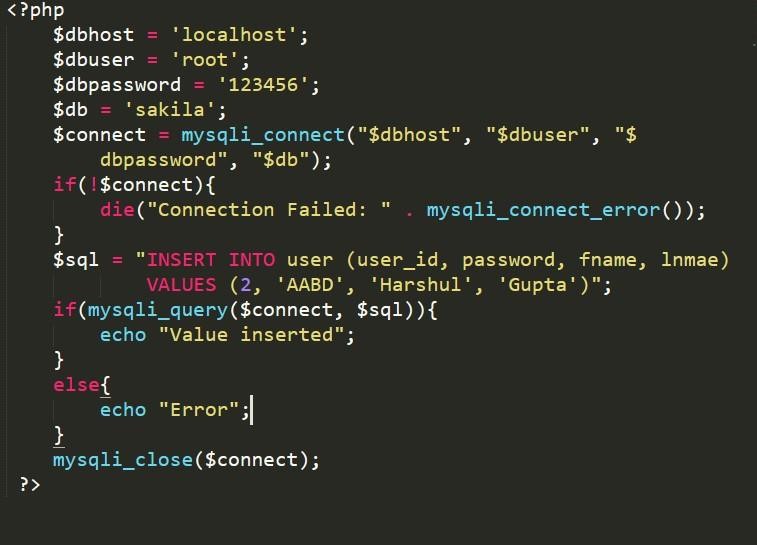
Write a program to connect and insert to database.

## OBJECTIVE OF PRACTICAL:

To connect to database and insert records.

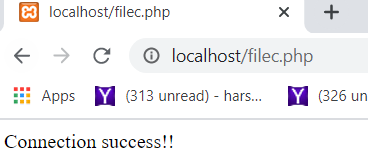
## IMPLEMENTATION:

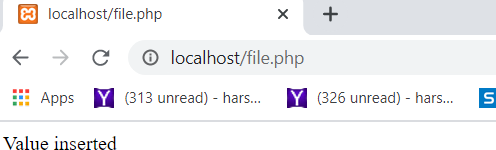


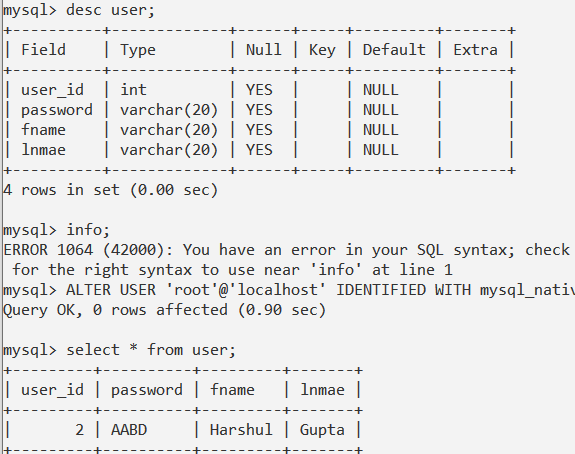


Practical Name : Write a program to connect and insert to database. Practical No : 18

1. Result /Output:







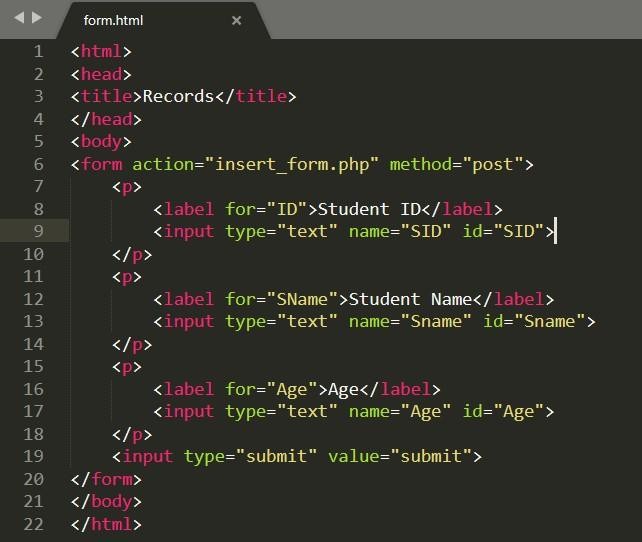
Practical Name : Connect and insert to database via HTML Practical No : 13

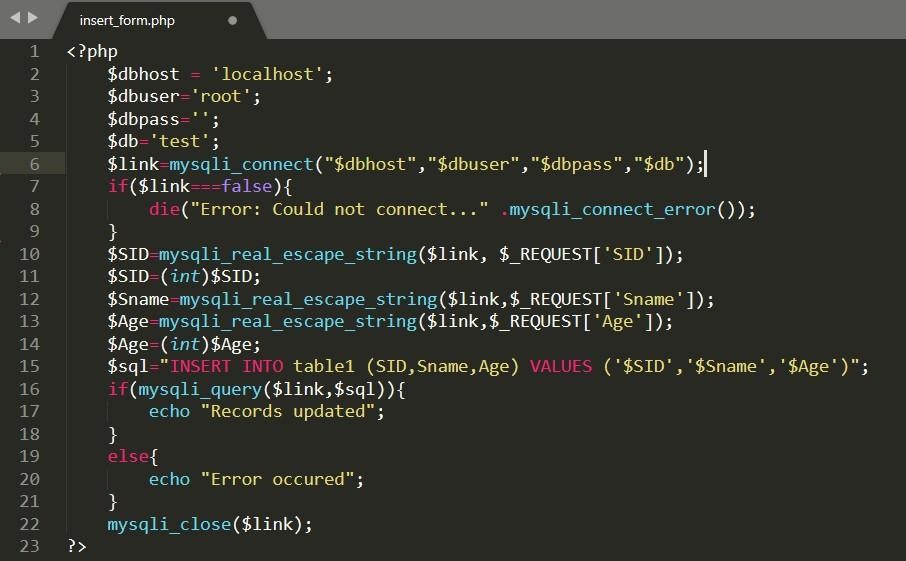
**Experiment : 19**

1. **PRACTICAL STATEMENT OF PRACTICAL:** Write a program to generate login control
2. **Theory:**

PHP is used for the login controls. It is used to link with the database and update it.

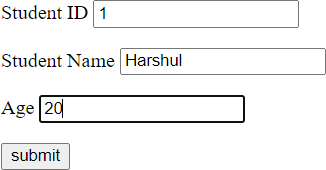
## IMPLEMENTATION:





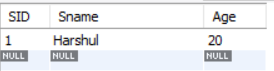
Practical Name : Connect and insert to database via HTML

## RESULT/OUTPUT:



**Mayankesh**





**Mayankesh**

Practical No : 20

**Experiment : 20**

1. **PRACTICAL STATEMENT OF PRACTICAL:** Write a program to connect and insert into database via Java.
2. **OBJECTIVE OF PRACTICAL:** To connect and insert record into database via Java file.
3. **Theory:**

**JDBC:**

|  |
| --- |
| **Java DataBase Connectivity**, commonly referred to as **JDBC**, is an [API](http://en.wikipedia.org/wiki/Application_programming_interface) for the [Java programming language](http://en.wikipedia.org/wiki/Java_(programming_language)) that defines how a client may access a [database](http://en.wikipedia.org/wiki/Database). It provides methods for querying and updating data in a database. JDBC is oriented towards [relational databases](http://en.wikipedia.org/wiki/Relational_database_management_system). A JDBC-to-[ODBC](http://en.wikipedia.org/wiki/Open_Database_Connectivity) bridge enables connections to any ODBC-accessible data source in the JVM host environment. |

## Functionality

JDBC allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the JDBC Driver Manager. The Driver Manager is used as a connection factory for creating JDBC connections.

JDBC connections support creating and executing statements. These may be update statements such as [SQL](http://en.wikipedia.org/wiki/SQL)'s CREATE, INSERT, UPDATE and DELETE, or they may be query statements such as SELECT. Additionally, [stored procedures](http://en.wikipedia.org/wiki/Stored_procedures) may be invoked through a JDBC connection. JDBC represents statements using one of the following classes:

* [Statement](http://download.oracle.com/javase/6/docs/api/java/sql/Statement.html) – the statement is sent to the database server each and every time.
* [PreparedStatement](http://download.oracle.com/javase/6/docs/api/java/sql/PreparedStatement.html) – the statement is cached and then the execution path is pre determined on the database server allowing it to be executed multiple times in an efficient manner.
* [CallableStatement](http://download.oracle.com/javase/6/docs/api/java/sql/CallableStatement.html) – used for executing [stored procedures](http://en.wikipedia.org/wiki/Stored_procedures) on the database.

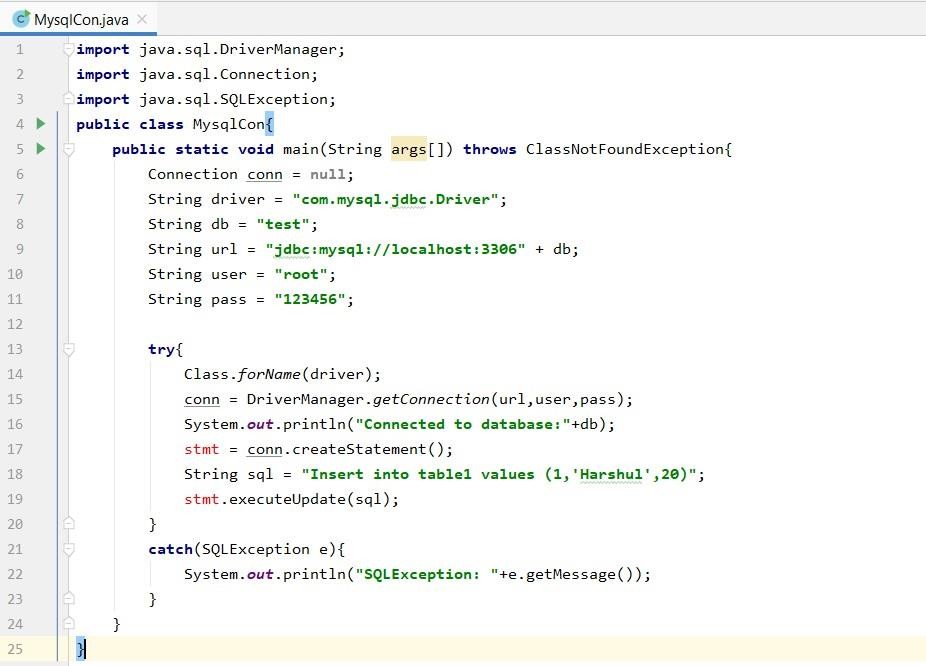
Update statements such as INSERT, UPDATE and DELETE return an update count that indicates how many rows were affected in the database. These statements do not return any other information.

Query statements return a JDBC row result set. The row result set is used to walk over the result set. Individual columns in a row are retrieved either by name or by column number. There may be any number of rows in the result set. The row result set has metadata that describes the names of the columns and their types.

There is an extension to the basic JDBC API in the [javax.sql](http://download.oracle.com/javase/6/docs/api/javax/sql/package-summary.html).

JDBC connections are often managed via a connection pool rather than obtained directly from the driver. Examples of connection pools include [BoneCP](http://jolbox.com/), [C3P0](http://sourceforge.net/projects/c3p0) and [DBCP](http://commons.apache.org/dbcp)

## IMPLEMENTATION:



Practical No :14

## RESULT/OUTPUT:

