VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

WEB TECHNOLOGIES LAB MANUAL

LAB CODE: 19CS215 A

COURSE DESCRIPTION AND OBJECTIVES

This course offers the basic concepts used to develop static web pages and it also provides knowledge of Internet programming concepts, Web Servers and Web Application Servers, Design Methodologies with concentration on Object-Oriented concepts. The objective of this course is to build web applications using JSP and client side script technologies

COURSE OUTCOMES

The student will be able to:

- Understand the concepts of HTML, CSS and JAVA script
- Apply javascript features for form validation and jdbc concepts to perform database operations from web pages
- Analyze the suitability of servlet and jsp technologies to build solutions for real world problems
- Evaluate the performance of web application development using JSP, SERVLETS and PHP.
- Design and develop three tier web applications using JSP, SERVLETS and PHP.

SKILLS

- Perform client side validation using Java script.
- Store and retrieve data using JDBC.
- Generate dynamic contents using Servlets.
- Generate dynamic contents using JSPs and PHP.
- Overcome problems in Servlets and JSP using Struts Programs.
- Develop a working system of web application or web site.

LIST OF EXPERIMENTS

1. Create a HTML document to display your profile and Class timetable to demonstrate lists, links, images, pre, font, heading tags.
2. Create a HTML page having four frames named
    a. top
    b. center
    c. bottom
    d. left

The top frame should contain company logo and title. The bottom frame should contain Copy right information. The left frame should contain various links like Home, Products, Services, Branches, about us, etc. When clicked on those links, the contents should appear in the display on to center frame.

3. Create a HTML document to demonstrate Form Elements that includes Form, input-text, password, radio, checkbox, hidden, button, submit, reset, label, text area, select, option, file upload
4. Write a HTML program with at least two <h1>, two images, two buttons and appropriate

CSS to display

a) All <h1> with font-size 12pt, and bold in Verdana font using Inline CSS.

b) All <img> with border color yellow, thickness 10px using Document Level CSS

c) All <input type=’button’> should change background color to red on mouse over them using External CSS.

5. Design a HTML page having a text box and four buttons via Factorial, Fibonacci, Prime and Palindrome. When a button is pressed an appropriate java script function should be called to display the following:
    a. factorial of that number
    b. Fibonacci series up to that number
    c. prime numbers up to that number
    d. is it palindrome or not
6. Write Java script program to demonstrate the following objects with at least

five methods:

a. MATH
b. STRING
c. ARRAY
d. DATE

7. Write a JavaScript program to perform arithmetic operations when the corresponding button is pressed.
8. Write a Javascript program to validate user registration form.
9. Write a Java program to connect to a database server and insert employees data into employee table using JDBC.
10. Write a JDBC Program to retrieve and display data from a table employee.
11. Write a simple java servlet program to display Simple text content on screen.
12. Write a servlet program to demonstrate Life cycle of Servlet.
13. Develop a Servlet to validate user name and password with the data stored in Servlet configuration file. Display authorized user if she/he is authorized else display unauthorized user.
14. Write JSP Program to store student information sent from registration page into Database.
15. Write a JSP program to validate username and password through database.
16. Write an appropriate JSP page to insert, update and delete data in student table in a single application with proper linking of JSP pages.
17. Write a PHP script to store registration information sent from registration page into database.
18. Develop a program to validate username and password that are stored in database using PHP.
19. Write an appropriate PHP page to insert, update and delete data in registration table in a single application with proper linking of PHP pages and session.

EXPERIMENT 1

Create a HTML document to display your profile and Class timetable to demonstrate lists, links, images, pre, font, heading tags.

AIM: Create a profile and Class Timetable using the basic concepts of HTML.

ALGORITHM:

STEP 1. Open a document and save it with program name.html

STEP 2. By using li and ul tag create a ordered list and unordered list should be created.

STEP 3. To each list create a link by using href attribute.

STEP 4. In one of the page, create a separate html page which has one heading with some text followed by paragraph using p tag. Which will be followed for remaining headings also? Each para will be separated by hr tag.

STEP 5. An image should be displayed at the top right side. Different pages should be created and displayed by clicking the links which are displayed by using lists

STEP 6. By using table tag create a table.

STEP 7. For heading use the th tag and for data use the td tag

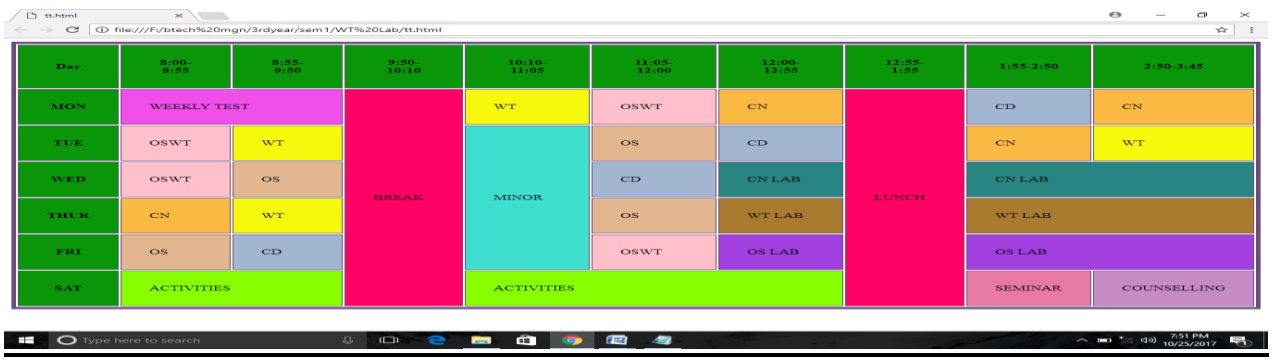
STEP 8. Background color should be applied for each table row.

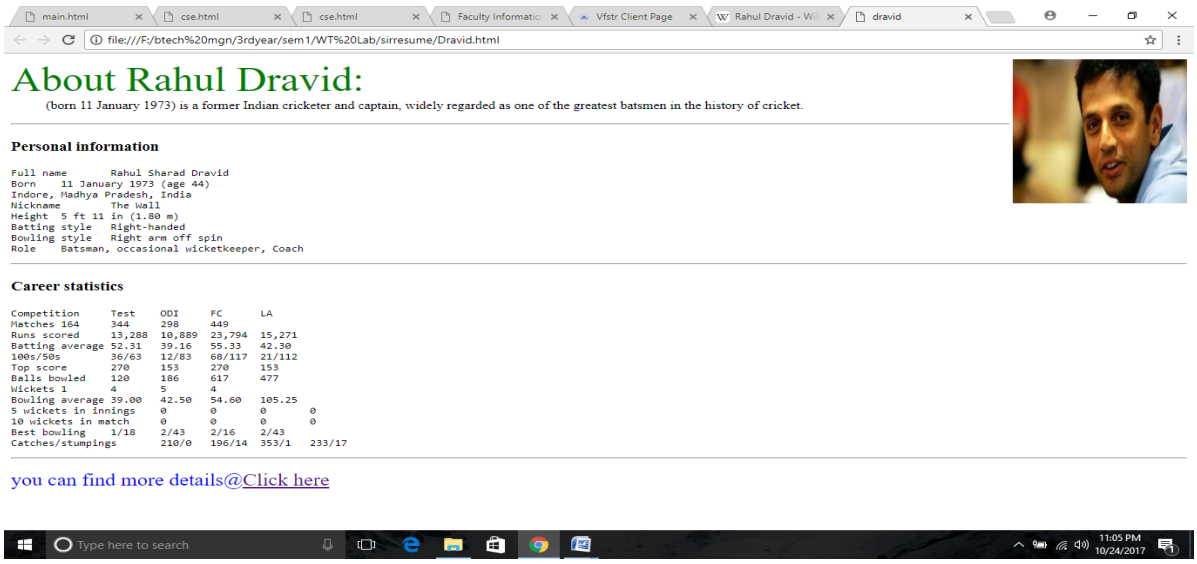
STEP 9. For merging rows or columns, use colspan and rowspan

STEP 10. A time table format table should be displayed at last as per the conditions given in the question.

SOURCE CODE:

SAMPLE OUTPUT:





TEST CASES:

1. Html page should be displayed with different kinds of lists in which one should be ordered list and unordered lists.
2. Each list name should be made as link. By clicking each link of list, another page should be displayed
3. Different pages to be displayed successfully by clicking each link.
4. A html page should be displayed which shows some clear description of time table.
5. Each row should be showed with different kinds of colors.
6. There should be few merging of rows and columns should be there.

RESULT: User profile page and class time table is designed and developed successfully.

EXPERIMENT 2

Create a HTML page having four frames named

a. top

b. center

c. bottom

d. left

The top frame should contain company logo and title. The bottom frame should contain copy right information. The left frame should contain various links like Home, Products, Services, Branches, About us, etc. When clicked on those links, the contents should appear in the display on to center frame.

AIM: To Create a HTML page having four frames named

a. top

b. center

c. bottom

d. left

The top frame should contain company logo and title. The bottom frame should contain copy right information. The left frame should contain various links like Home, Products, Services, Branches, About us, etc. When clicked on those links, the contents should appear in the display on to center frame.

ALGORITHM:

STEP 1. create a html document by saving it with program name.html

STEP 2. by using frame tag, frames should be created

STEP 3. partition of columns, rows will be made by frameset tag

STEP 4. Top of the frame should contain of different images

STEP 5. In the middle left frame, 3 links should be created by anchor tag and target should be set on which frame, that particular page should be displayed.

STEP 6. Each separate html pages should be created with different images by using img tag, lists by using li tag, paragraph by using p tag. When we click the links, pages should be properly displayed in centre of the page.

SOURCE CODE:

SAMPLE OUTPUT:

TEST CASES:

1. A html page should be displayed which shows the division of page by 3 columns.
2. On the 1st top frames. Different images to be displayed.
3. In the second frame, on the left corner 3 links should be displayed.
4. By clicking that links different pages should be displayed in the middle of the frame or right side of the frame.
5. At the last bottom footer frame, some copyright should be displayed.

RESULT: Frames are designed using HTML successfully.

EXPERIMENT 3

Create a html document to demonstrate form elements that includes form, input-text, password, radio, checkbox, hidden, button, submit, reset, label, textarea, select, option, file-upload.

AIM: To create application using html tags.

ALGORITHM:

STEP 1. Open a document and save it with .html extension.
STEP 2. Open html tag and write code for textboxes for username, password, phone number etc.
STEP 3. Create a radio button for gender.
STEP 4. Create a checkbox for hobbies, qualification etc.
STEP 5. Create a dropdown box for select statement.
STEP 6. Create a file button for uploading a files
STEP 7. Create a button for submit, reset and cancel.
STEP 8. Save the program and open the document in browser.

SOURCE CODE:

SAMPLE OUTPUT:

Registration Form

First name	enter your first name
Last name	
Enter your password	
Enter your email	
Enter your mobile	
Enter your address	
Select your gender	Male Female
Select your hobbies	Watching Movies Playing Games Reading Books
Select your DOB	dd - mm - yyyy
Select your Country	Select your country
Upload your File	Choose File No file chosen
	Submit Reset

TEST CASES:

1. Text boxes should be displayed for entering the information
2. Radio button should be displayed to select gender
3. For multiple selection of input checkbox should be created for selection course
4. Files should be successfully submitted by choose file button
5. After giving input data, submit button should be clicked to submit database
6. Reset button is displayed to reset the database
7. Cancel button should be displayed to cancel the data

RESULT: Application form is designed successfully using html tags.

EXPERIMENT 4

Write a HTML program with at least two <h1>, two images, two buttons and appropriate CSS to display

- All <h1> with font-size 12pt, and bold in Verdana font using Inline CSS.
- All <img> with border color yellow, thickness 10px using Document Level CSS
- All <input type='button'> should change background color to red on mouse over them using External CSS.

AIM: To Create and design a webpage using the concept of CSS in HTML.

ALGORITHM:

STEP 1. Create a html document and save it with program name .html

STEP 2. Open html tag and write code

STEP 3. Use bgcolor in body tag to add background color

STEP 4. Write a frameset tags code inside the html tag and close the tags.

STEP 5. Create one more document for one frame in which we have given some html name

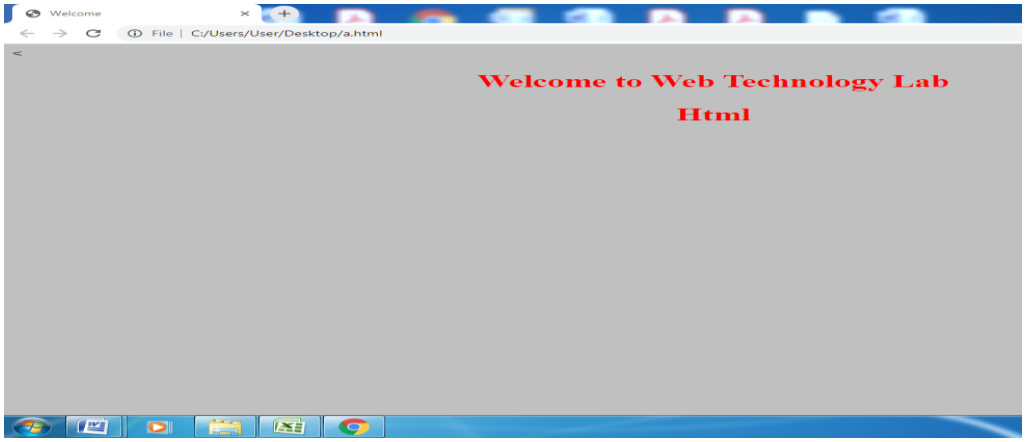
STEP 6. Write the html code and css code inside the document which we have given in the frame tag

STEP 7. Apply background color for the menu bar by using css or by using style tag

STEP 8. Save the file

SOURCE CODE:

SAMPLE OUTPUT:



TESTCASES:

1. Html page should be displayed with background menu bar as yellow which shows the division of page by 3 partition.
2. Html file should be displayed in frames.
3. If we click on the button, the corresponding page changed to red color
4. It should be displayed on same frame or next frame based on the code.

RESULT: A webpage is designed and developed by using CSS in HTML

EXPERIMENT 5

AIM: Design a HTML page having a text box and four buttons via Factorial, Fibonacci, Prime and Palindrome. When a button is pressed an appropriate java script function should be called to display the following:

factorial of that number

Fibonacci series up to that number

prime numbers up to that number

is it palindrome or not.

ALGORITHM:

STEP 1. HTML page should be created in a document by saving it with program name. html

STEP 2. Two textboxes should be created .one text box is for enter the data and another text box is for display the result

STEP 3. Four buttons should be created for four operations i.e.,Factorial,Fibonacci,prime numbers and palindrome.

STEP 4. After clicking the Factorial button, an appropriate java script factorial function should be displayed.

STEP 5. After clicking the Fibonacci button, an appropriate java script Fibonacci function should be displayed.

STEP 6. After clicking the Prime number button ,an appropriate java script prime number function should be displayed.

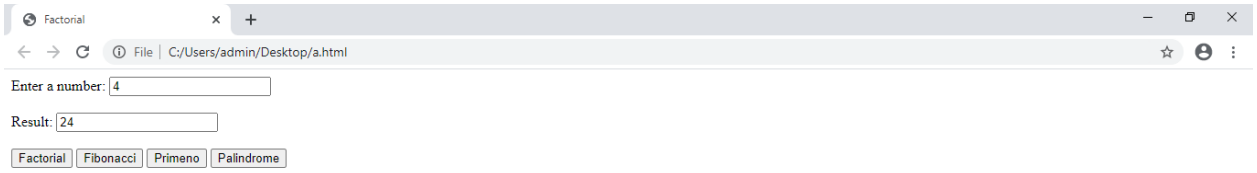
STEP 7. After clicking the palindrome button, an appropriate java script palindrome function should be displayed.

STEP 8. In the head tag, script tag should be opened

STEP 9. Inside the script tag,each function should be created for all four operations to write the code for factorial,Fibonacci,prime number and palindrome operations

STEP 10. Create variables in functions to store the number which user enters in the textbox by getElemntById function.

SAMPLE OUTPUT:



TEST CASES:

1. Html page should be displayed with two text boxes, one is for taking input from the user and other is for displaying result.

2. The user clicks on particular button , the appropriate java script function should be displayed.

EXPERIMENT 6

AIM: Write Java script program to demonstrate the following objects with at least five methods

MATH

STRING

ARRAY

DATE

ALGORITHM:

STEP 1. HTML page should be created in a document by saving it with program name. html

STEP 2. Four buttons should be created for four operations i.e.,Math,String,Array,Date.

STEP 3. After clicking the Math button ,the methods of math object should be displayed

STEP 4. After clicking the String button , the methods of string object should be displayed

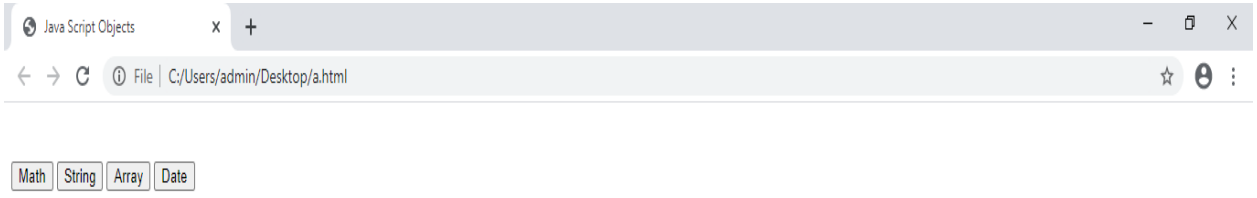
STEP 5. Next clicking on the Array button , the methods of Array object should be displayed

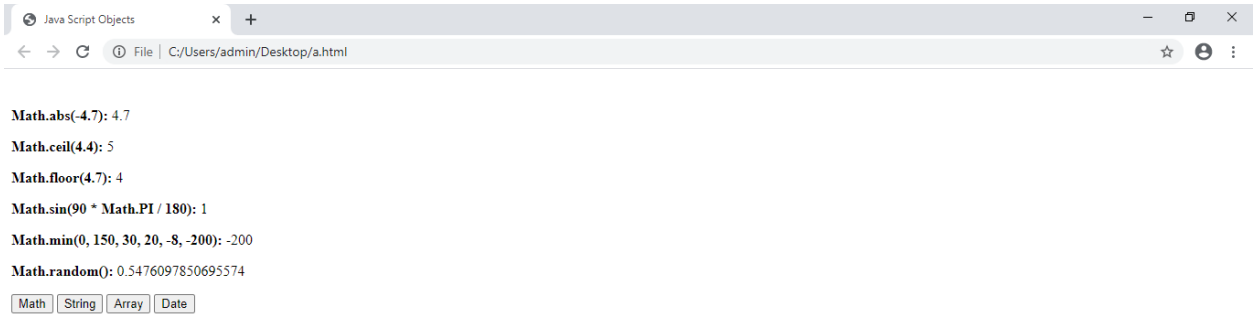
STEP 6. After clicking the Date button, the methods of Date object should be displayed

STEP 7. In the head tag, script tag should be opened

STEP 8. Inside the script tag,each function should be created for all four operations to write the code for Math,String,Array and Date operations.

SAMPLE OUTPUT:





TEST CASES:

1. Html page should be displayed with four buttons.

2. The user clicks on particular button,the appropriate methods of the function should be displayed.

RESULT: Successfully web page designed with given requirements.

EXPERIMENT 7

AIM : Write a JavaScript program to perform arithmetic operations when the corresponding button is pressed.

ALGORITHM:

STEP 1. HTML page should be created in a document by saving it with program name. html

STEP 2. Two textboxes should be created in order to enter the data

STEP 3. One more text box to be created in order to display the result inside it

STEP 4. Four buttons should be created for four operations i.e., addition, subtraction, division and multiplication.

STEP 5. After clicking the add button ,addition of two numbers operation should be done.

STEP 6. After clicking the sub button ,subtraction of two numbers operation should be done.

STEP 7. After clicking the mul button ,multiplication of two numbers operation should be done.

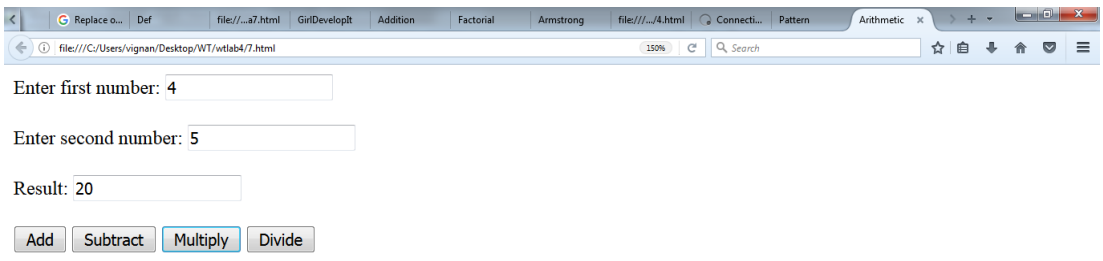
STEP 8. After clicking the division button ,division of two numbers operation should be done.

STEP 9. In the head tag, script tag should be opened

STEP 10. Inside the script tag,each function should be created for all four operations to write the code for adding,subtracting,multiplication and division of two numbers.

STEP 11. Create two variables. To each variable, store the number which user enters in the textbox by getElemntById function.

OUTPUT:



TEST CASES:

1. Three text boxes should be displayed, the first two text boxes for taking input and other is for displaying the output inside it.
2. Add button is displayed, after clicking it adding of two numbers should be done and successfully displayed the result inside the textbox.
3. Subtract button is displayed, after clicking it subtraction of two numbers should be done and successfully displayed the result inside the textbox.
4. Multiply button is displayed, after clicking it Multiplication of two numbers should be done and successfully displayed the result inside the textbox.
5. Division button is displayed, after clicking it division of two numbers should be done and successfully displayed the result inside the textbox.

RESULT: Successfully web page designed with given requirements.

EXPERIMENT 8

AIM : Write a Javascript program to validate user registration from.

ALGORITHM:

STEP 1. HTML page should be created in a document by saving it with program name. html.

STEP 2. Eight textboxes should be created in order to enter the data.

STEP 3. For each field you have to make validation.

STEP 4. For Name Field, the name should be a valid username ie., it contains only characters and it should not be null

STEP 5. Next for Password Field,the password should be a valid password i.e,. it contains combination of atleast one upper case letter,one lowercase letter,one digit and special characters. The field also contains length of characters i.e.,6-20 character size.

STEP 6. For email field,the email should be a valid email ie.,it contains @ and . symbol.

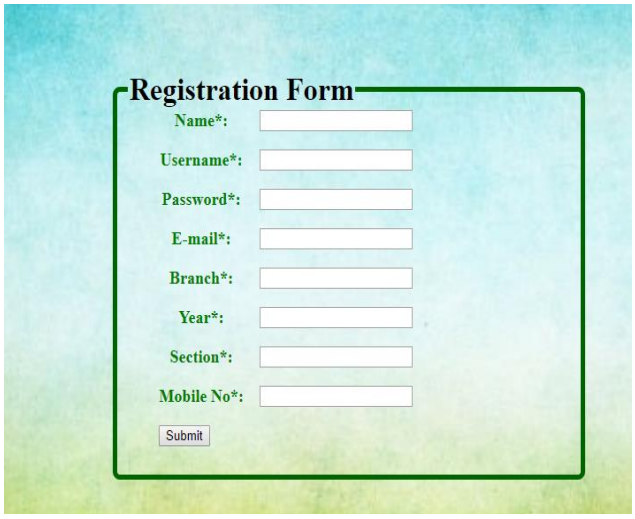
STEP 7. For branch field,the branch name should be valid.

STEP 8. For year Field, the year should be valid year.

STEP 9. Last the mobile number,valid mobile number should be entered i.e.,the mobile number should be 10 digit number.

STEP 10. Create one submit button to validate all the fields.

SAMPLE OUTPUT:



TEST CASES:

1. Html page should be displayed with eight text boxes, which is for taking input from the user.

2. The user click on particular textbox and enter the data; the data should be validated based upon the given conditions.

RESULT: Successfully web page designed with given requirements.

EXPERIMENT 9

Write a Java program to connect to a database server and insert employee's data into employee table using JDBC.

AIM: To create a JDBC program to connect with database and insert employee's information in database table.

ALGORITHM:

STEP 1. Import JDBC packages.

STEP 2. Load and register the JDBC driver.

STEP 3. Open a connection to the database.

STEP 4. Create a statement object to perform a query.

STEP 5. Execute the statement object and return a query resultset.

STEP 6. Process the resultset.

STEP 7. Close the resultset and statement objects.

STEP 8. Close the connection.

DESCRIPTION:

Fundamental Steps in JDBC

The fundamental steps involved in the process of connecting to a database and executing a query consist of the following:

- Import JDBC packages.
- Load and register the JDBC driver.
- Open a connection to the database.
- Create a statement object to perform a query.
- Execute the statement object and return a query resultset.
- Process the resultset.
- Close the resultset and statement objects.
- Close the connection.

These steps are described in detail in the sections that follow.

Import JDBC Packages

This is for making the JDBC API classes immediately available to the application program. The following import statement should be included in the program irrespective of the JDBC driver being used:

```
import java.sql.*;
```

Additionally, depending on the features being used, Oracle-supplied JDBC packages might need to be imported. For example, the following packages might need to be imported while using the Oracle extensions to JDBC such as using advanced data types such as BLOB, and so on.

```
import oracle.jdbc.driver.*;

import oracle.sql.*;
```

Load and Register the JDBC Driver

This is for establishing a communication between the JDBC program and the Oracle database. This is done by using the static registerDriver() method of the DriverManager class of the JDBC API. The following line of code does this job:

```
DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
```

JDBC Driver Registration

For the entire Java application, the JDBC driver is registered only once per each database that needs to be accessed. This is true even when there are multiple database connections to the same data server.

Alternatively, the forName() method of the java.lang.Class class can be used to load and register the JDBC driver:

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

However, the forName() method is valid for only JDK-compliant Java Virtual Machines and implicitly creates an instance of the Oracle driver, whereas the registerDriver() method does this explicitly.

Connecting to a Database

Once the required packages have been imported and the Oracle JDBC driver has been loaded and registered, a database connection must be established. This is done by using the getConnection() method of the DriverManager class. A call to this method creates an object instance of the java.sql.Connection class. The getConnection() requires three input parameters, namely, a connect string, a username, and a password. The connect string should specify the JDBC driver to be yes and the database instance to connect to.

The getConnection() method is an overloaded method that takes

- Three parameters, one each for the URL, username, and password.
- Only one parameter for the database URL. In this case, the URL contains the username and password.

The following lines of code illustrate using the getConnection() method:

```
Connection conn = DriverManager.getConnection(URL, username, passwd);

Connection conn = DriverManager.getConnection(URL);
```

where URL, username, and passwd are of String data types.

We will discuss the methods of opening a connection using the Oracle JDBC OCI and thin _drivers.

When using the OCI driver, the database can be specified using the TNSNAMES entry in the tnsnames.ora file. For example, to connect to a database on a particular host as user oratest and password oratest that has a TNSNAMES entry of oracle.world, use the following code:

```
Connection conn = DriverManager.getConnection("jdbc:oracle:oci8:
```

```
@oracle.world", "oratest", "oratest");
```

Both the ":" and "@" are mandatory.

When using the JDBC thin driver, the TNSNAMES entry cannot be used to identify the database. There are two ways of specifying the connect string in this case, namely,

- Explicitly specifying the hostname, the TCP/IP port number, and the Oracle SID of the database to connect to. This is for thin driver only.
- Specify a Net8 keyword-value pair list.

For example, for the explicit method, use the following code to connect to a database on host training where the TCP/IP listener is on port 1521, the SID for the database instance is Oracle, the username and password are both oratest:

```
Connection conn = DriverManager.getConnection

        ("jdbc:oracle:thin:@training:1521:Oracle",

 "oratest", "oratest");
```

For the Net8 keyword-value pair list, use the following:

```
Connection conn = DriverManager.getConnection
("jdbc:oracle:thin@(description=(address=
(host=training)(protocol=tcp)(port=1521))(connect_data=(sid=Oracle))) ", _"oratest",
"oratest");
```

This method can also be used for the JDBC OCI driver. Just specify oci8 instead of thin in the above keyword-value pair list.
Querying the Database
Querying the database involves two steps: first, creating a statement object to perform a query, and second, executing the query and returning a resultset.

Creating a Statement Object

This is to instantiate objects that run the query against the database connected to. This is done by the createStatement() method of the conn Connection object created above. A call to this method creates an object instance of the Statement class. The following line of code illustrates this:

```
Statement sql_stmt = conn.createStatement();
```

Executing the Query and Returning a ResultSet

Once a Statement object has been constructed, the next step is to execute the query. This is done by using the executeQuery() method of the Statement object. A call to this method takes as parameter a SQL SELECT statement and returns a JDBC ResultSet object. The following line of code illustrates this using the sql_stmt object created above:

```
ResultSet rset = sql_stmt.executeQuery

    ("SELECT empno, ename, sal, deptno FROM emp ORDER BY ename");
```

Alternatively, the SQL statement can be placed in a string and then this string passed to the executeQuery() function. This is shown below.

```
String sql = "SELECT empno, ename, sal, deptno FROM emp ORDER BY ename";

ResultSet rset = sql_stmt.executeQuery(sql);
```

Statement and ResultSet Objects

Statement and ResultSet objects open a corresponding cursor in the database for SELECT and other DML statements.

The above statement executes the SELECT statement specified in between the double quotes and stores the resulting rows in an instance of the ResultSet object named rset.
Processing the Results of a Database Query That Returns Multiple Rows
Once the query has been executed, there are two steps to be carried out:

- Processing the output resultset to fetch the rows
- Retrieving the column values of the current row

The first step is done using the next() method of the ResultSet object. A call to next() is executed in a loop to fetch the rows one row at a time, with each call to next() advancing the control to the next available row. The next() method returns the Boolean value true while rows are still available for fetching and returns false when all the rows have been fetched.
The second step is done by using the getXXX() methods of the JDBC rset object.
Here getXXX() corresponds to the getInt(), getString() etc with XXX being replaced by a Java datatype.
The following code demonstrates the above steps:

```
String str;

while (rset.next())

 {

 str = rset.getInt(1)+ " "+ rset.getString(2)+ "  "+rset.getFloat(3)+ " "rset.getInt(4)+ "\n";

 }

byte buf[] = str.getBytes();

OutputStream fp = new FileOutputStream("query1.lst");

fp.write(buf);

fp.close();
```

Here the 1, 2, 3, and 4 in rset.getInt(), rset.getString(), getFloat(), and getInt() respectively denote the position of the columns in the SELECT statement, that is, the first column empno, second column ename, third column sal, and fourth column deptno of the SELECT statement respectively.

Specifying get() Parameters

The parameters for the getXXX() methods can be specified by position of the corresponding columns as numbers 1, 2, and so on, or by directly specifying the column names enclosed in double quotes, as getString("ename") and so on, or a combination of both.

Closing the ResultSet and Statement

Once the ResultSet and Statement objects have been used, they must be closed explicitly. This is done by calls to the close() method of the ResultSet and Statement classes. The following code illustrates this:

```
rset.close();

sql_stmt.close();
```

If not closed explicitly, there are two disadvantages:

- Memory leaks can occur
- Maximum Open cursors can be exceeded

Closing the ResultSet and Statement objects frees the corresponding cursor in the database.

Closing the Connection

The last step is to close the database connection opened in the beginning after importing the packages and loading the JDBC drivers. This is done by a call to the close() method of the Connection class.

The following line of code does this:

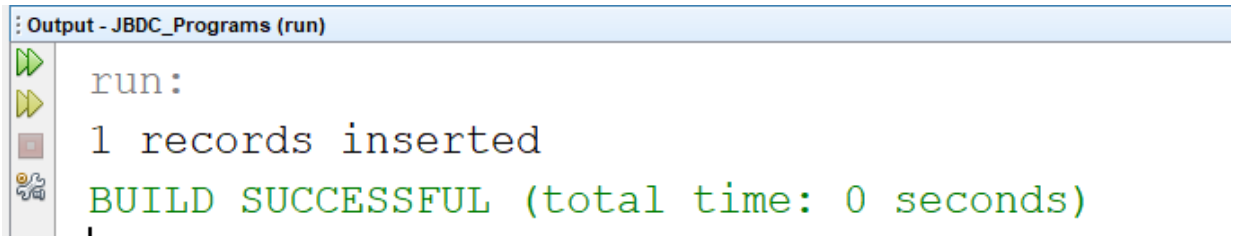
```
conn.close();
```

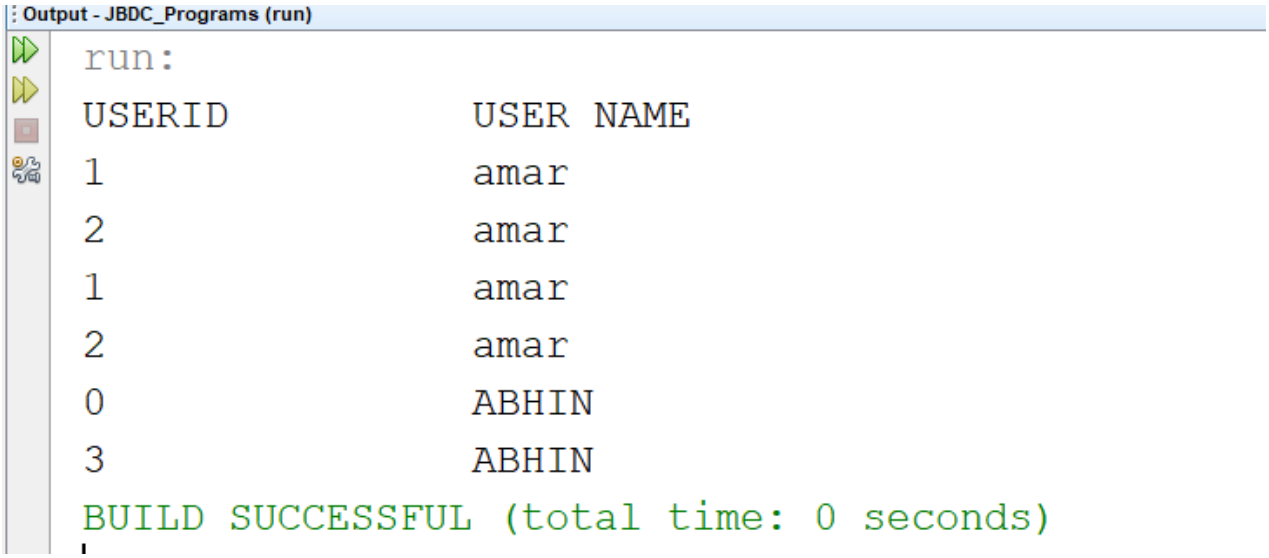
Explicitly Close your Connection

Closing the ResultSet and Statement objects does not close the connection. The connection should be closed by explicitly invoking the close() method of the Connection class.

SOURCE CODE:

SAMPLE OUTPUT:





RESULT: Successfully JDBC program is implemented and employee's information inserted into database table.

EXPERIMENT 10

Write a JDBC Program to retrieve and display data from a table employee.

AIM: To develop JDBC java program to retrieve and display user information from employee table.

ALGORITHM:

STEP 1. Import JDBC packages.

STEP 2. Load and register the JDBC driver.

STEP 3. Open a connection to the database.

STEP 4. Create a statement object to perform a query.

STEP 5. Execute the statement object and return a query resultset.

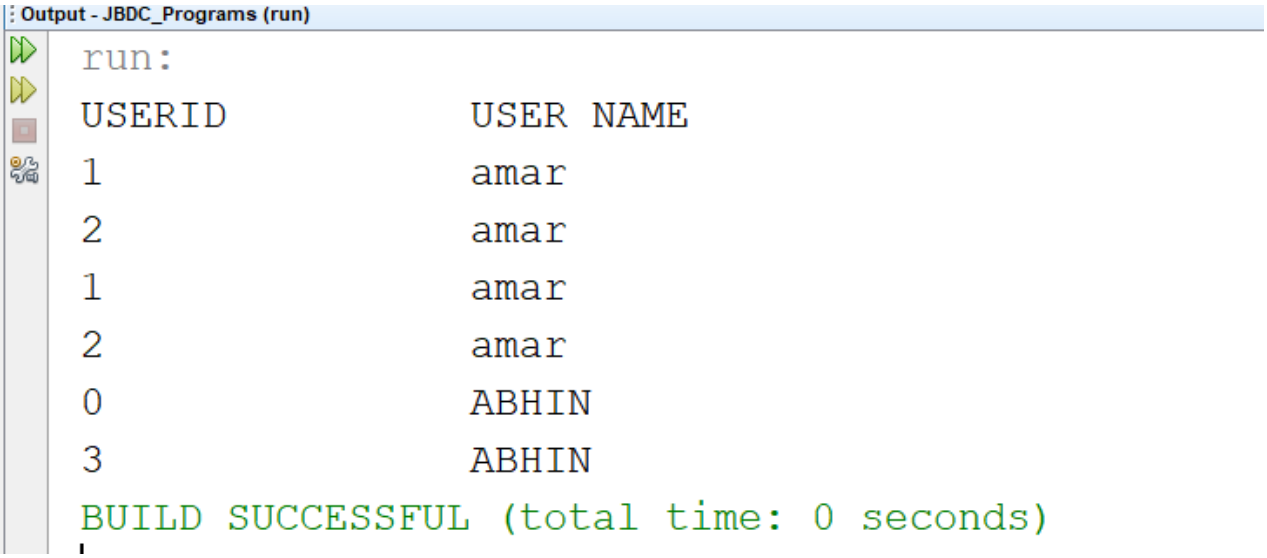
STEP 6. Process the resultset.

STEP 7. Close the resultset and statement objects.

STEP 8. Close the connection

SOURCE CODE:

SAMPLE OUTPUT:



TEST CASES:

RESULT: Successfully JDBC program is implemented and employee's information retrieved and displayed on screen.

EXPERIMENT 11

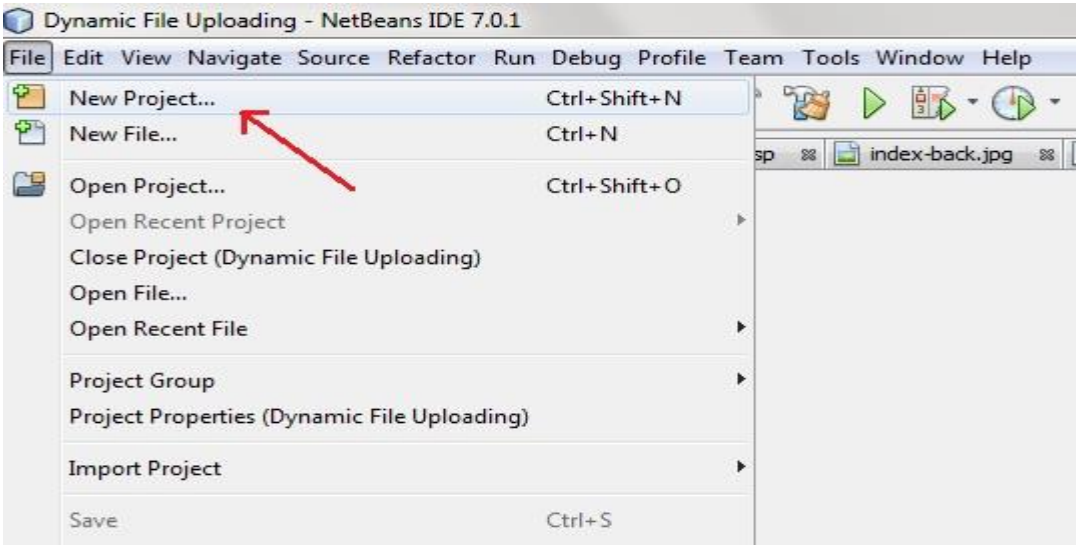
Write a simple java servlet program to display Simple text content on screen.

AIM: To develop java servlet program to display user simple text on screen.

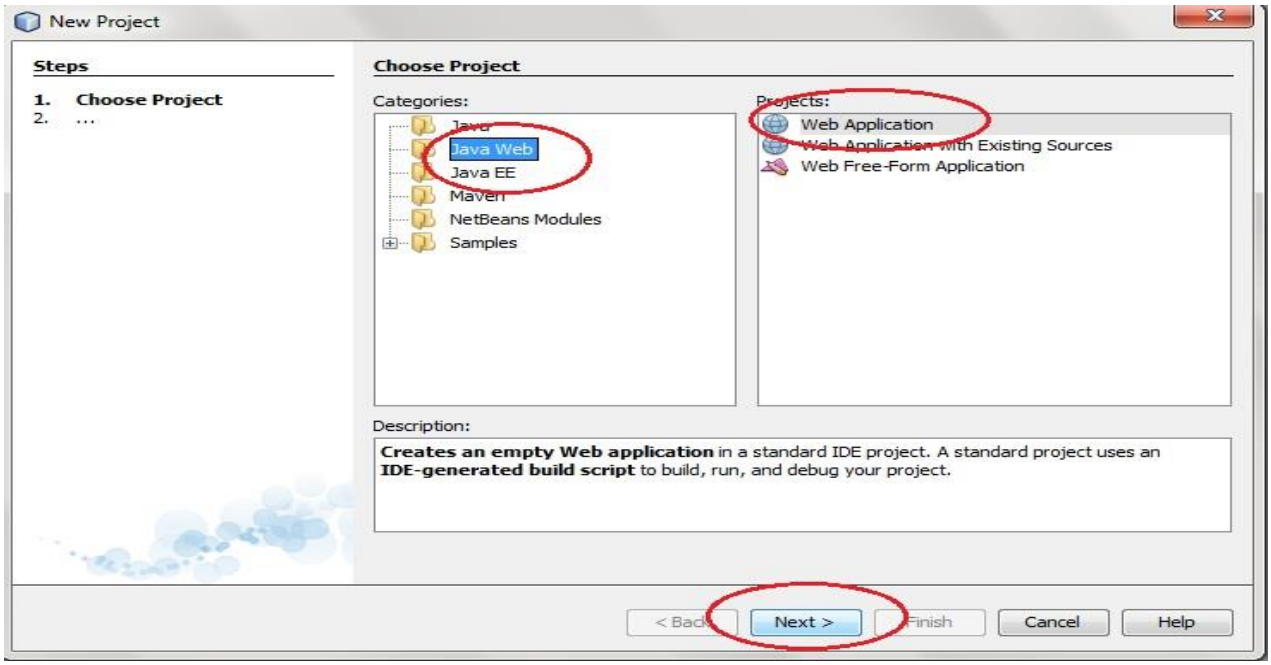
ALGORITHM:

To create a servlet application in Netbeans IDE, you will need to follow the following (simple) steps :

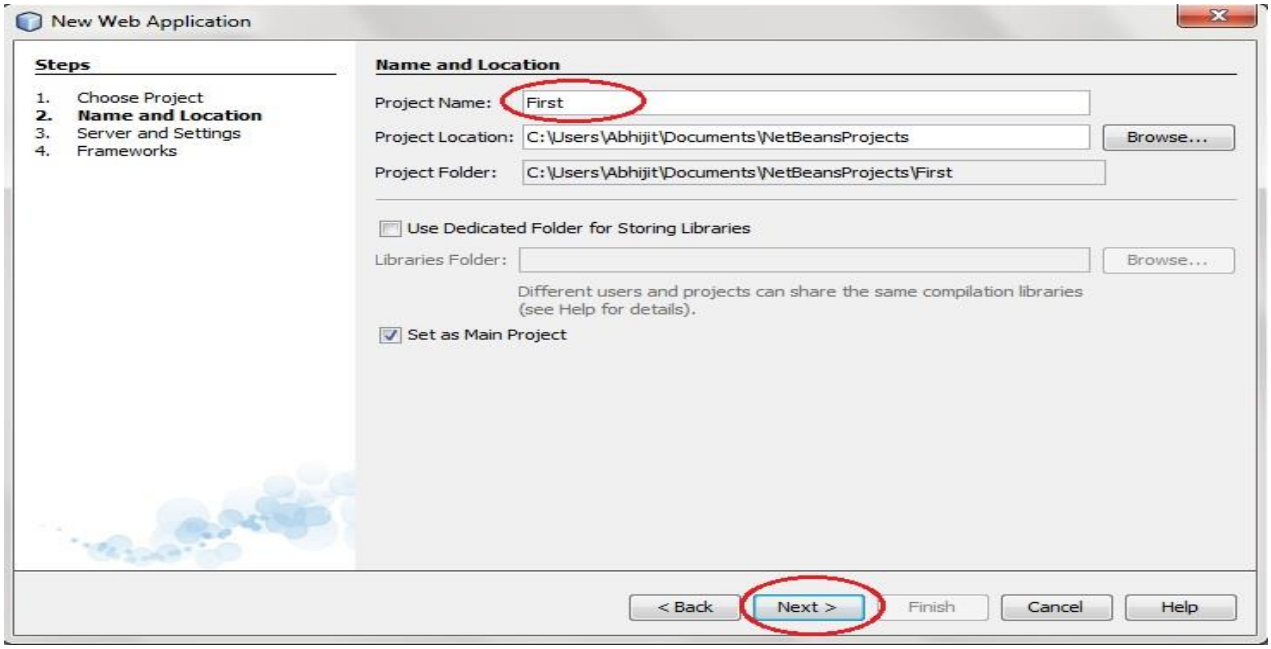
STEP 1. Open Netbeans IDE, Select **File** -> **New Project**



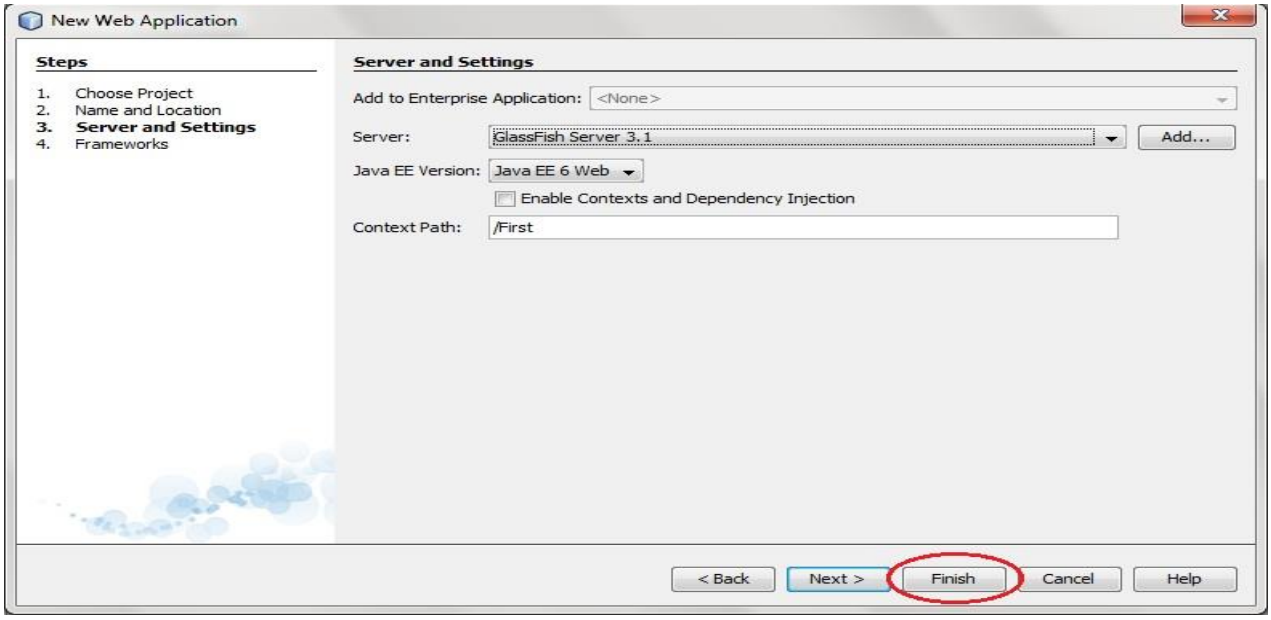
STEP 2. Select **Java Web** -> **Web Application**, then click on Next,



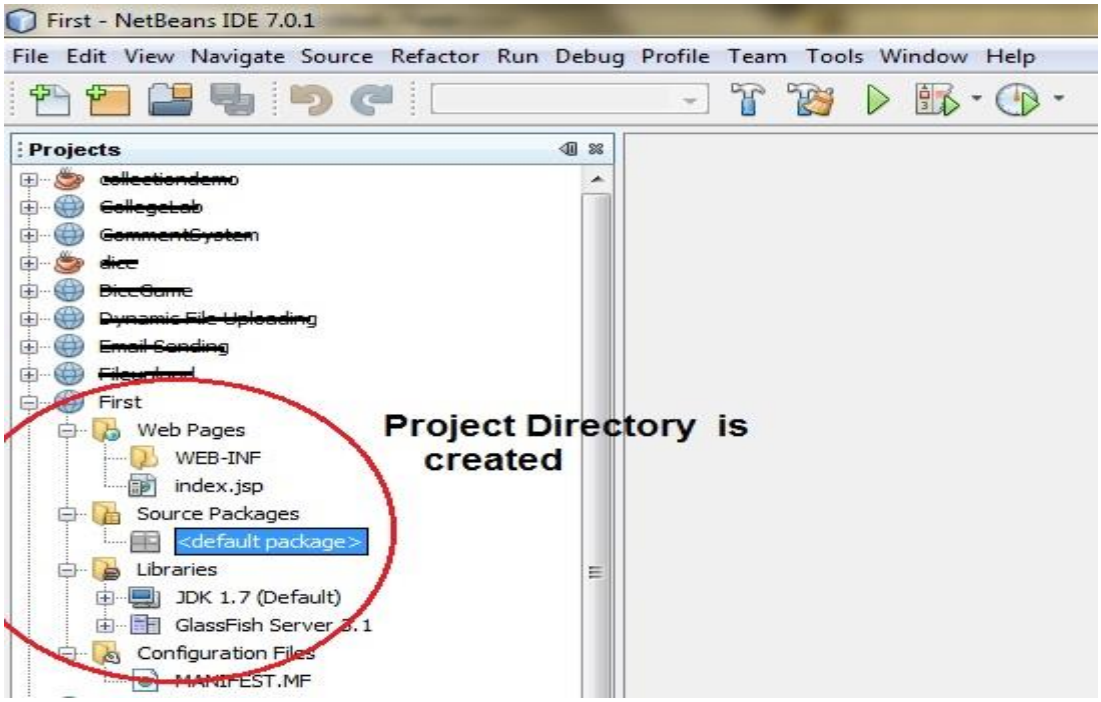
STEP 3. Give a name to your project and click on Next,



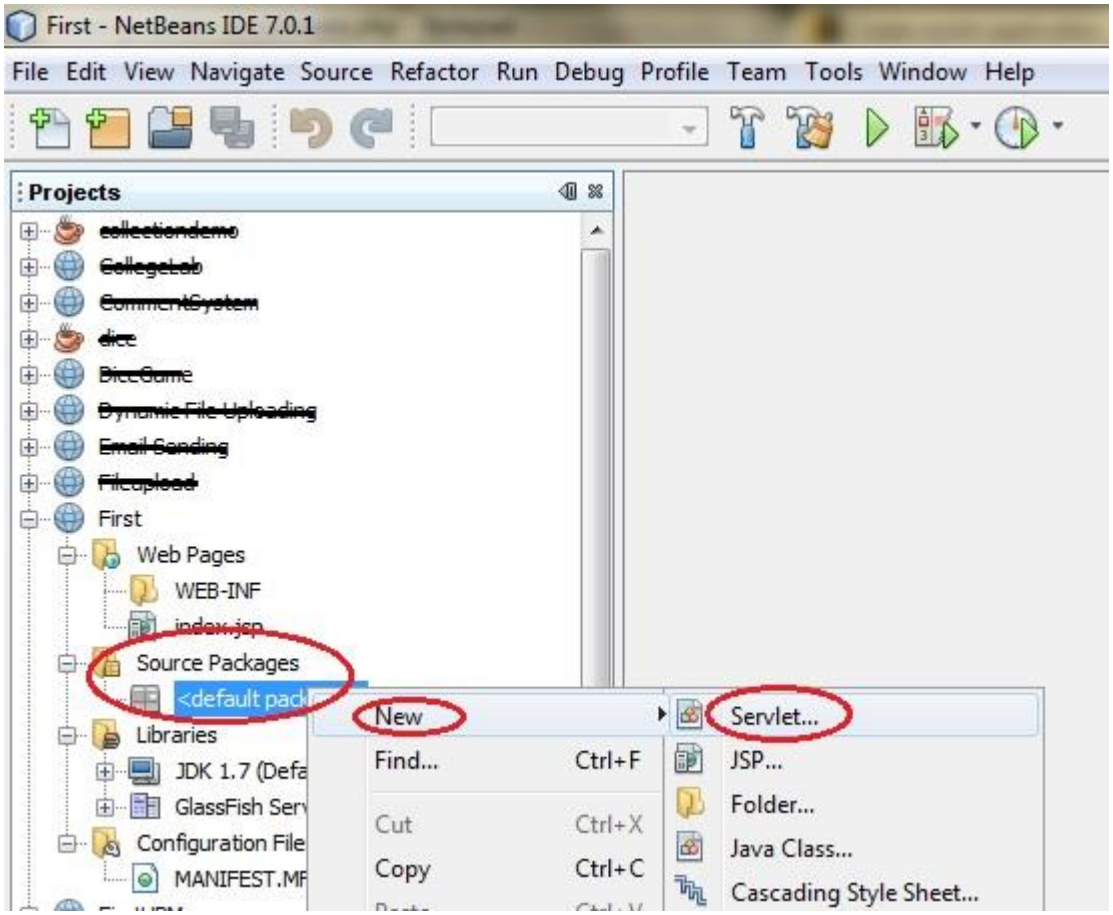
STEP 4. and then, Click **Finish**



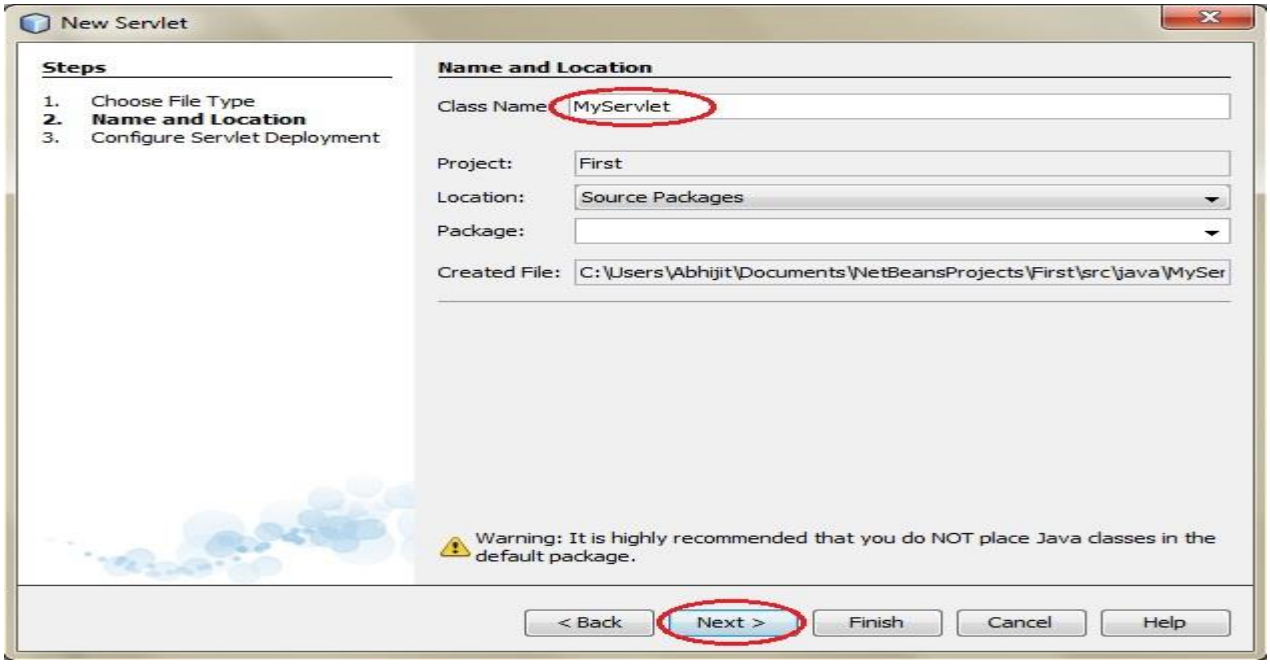
STEP 5. The complete directory structure required for the Servlet Application will be created automatically by the IDE.

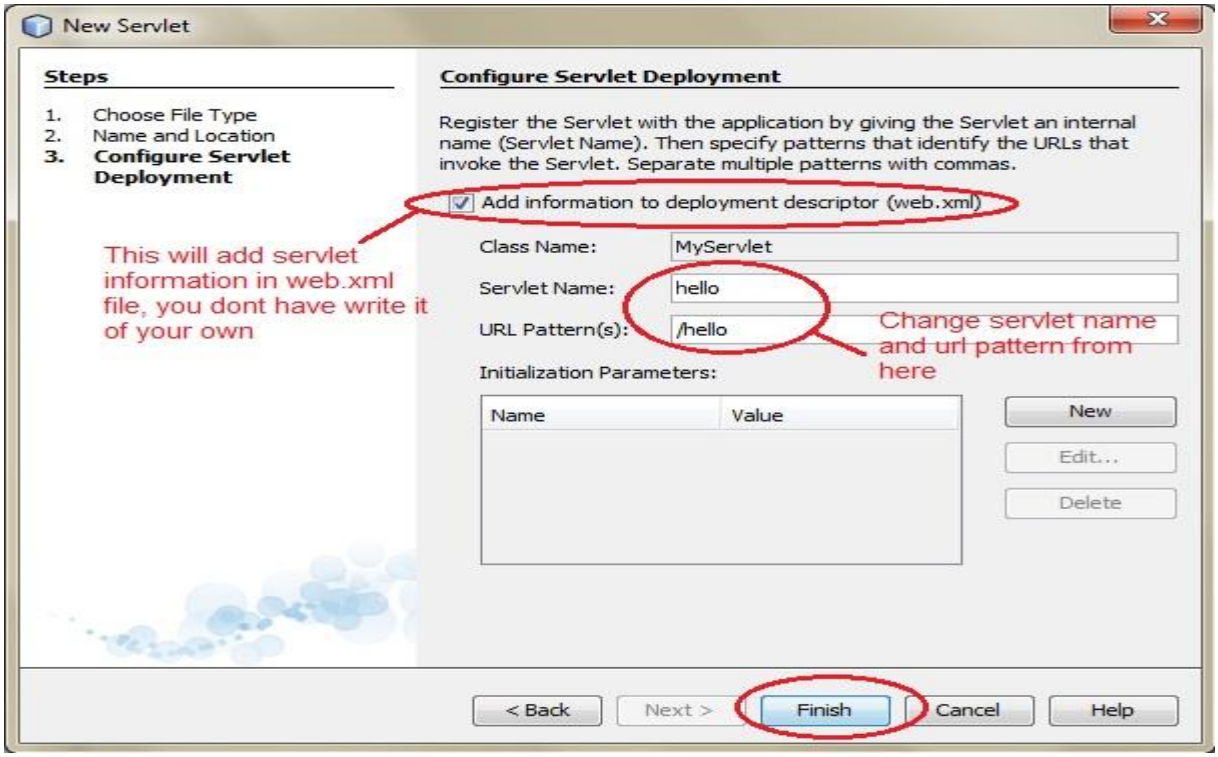


STEP 6. To create a Servlet, open **Source Package**, right click on **default packages** -> **New** -> **Servlet**.

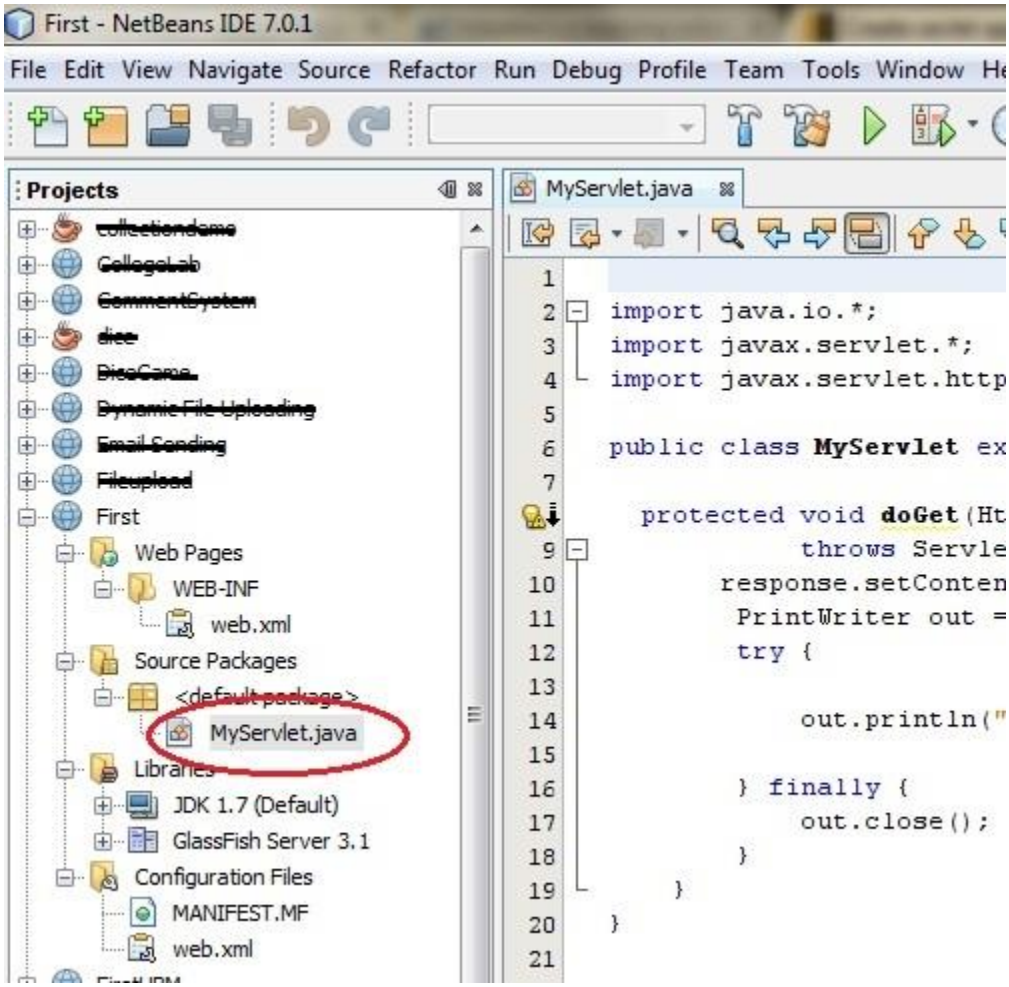


STEP 7. Give a Name to your Servlet class file,

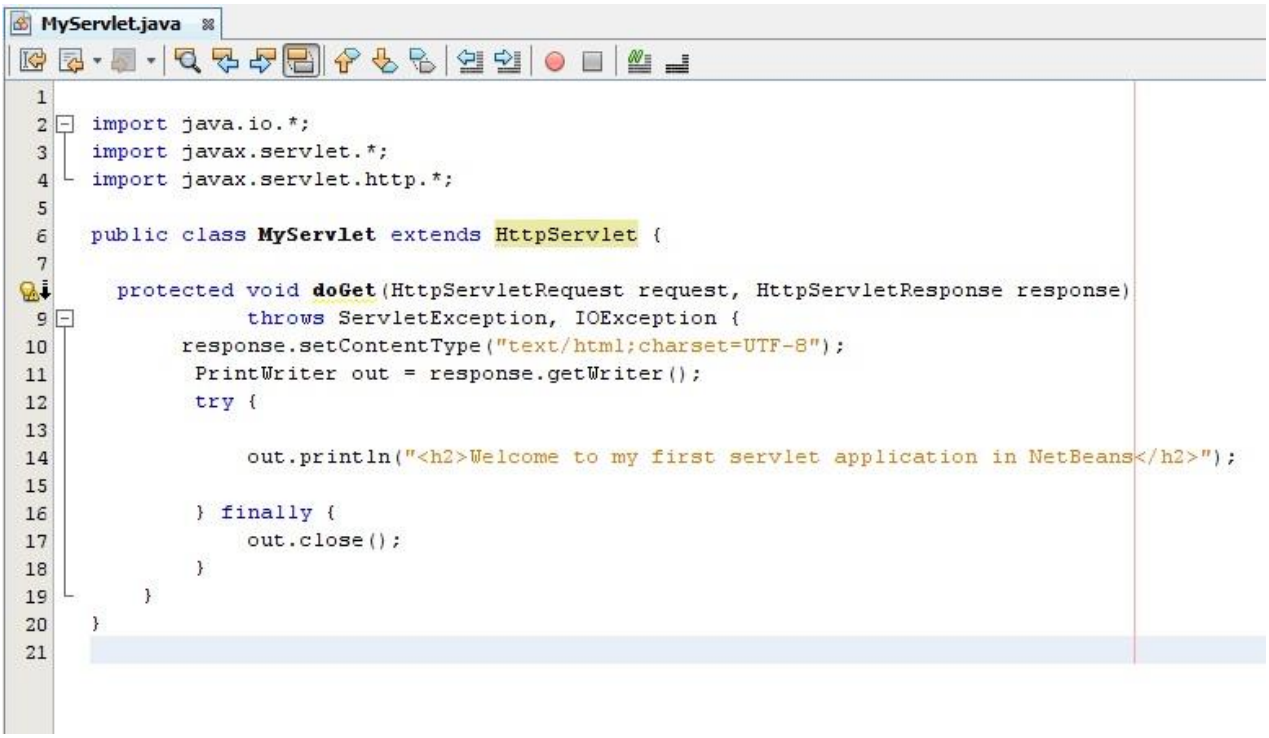




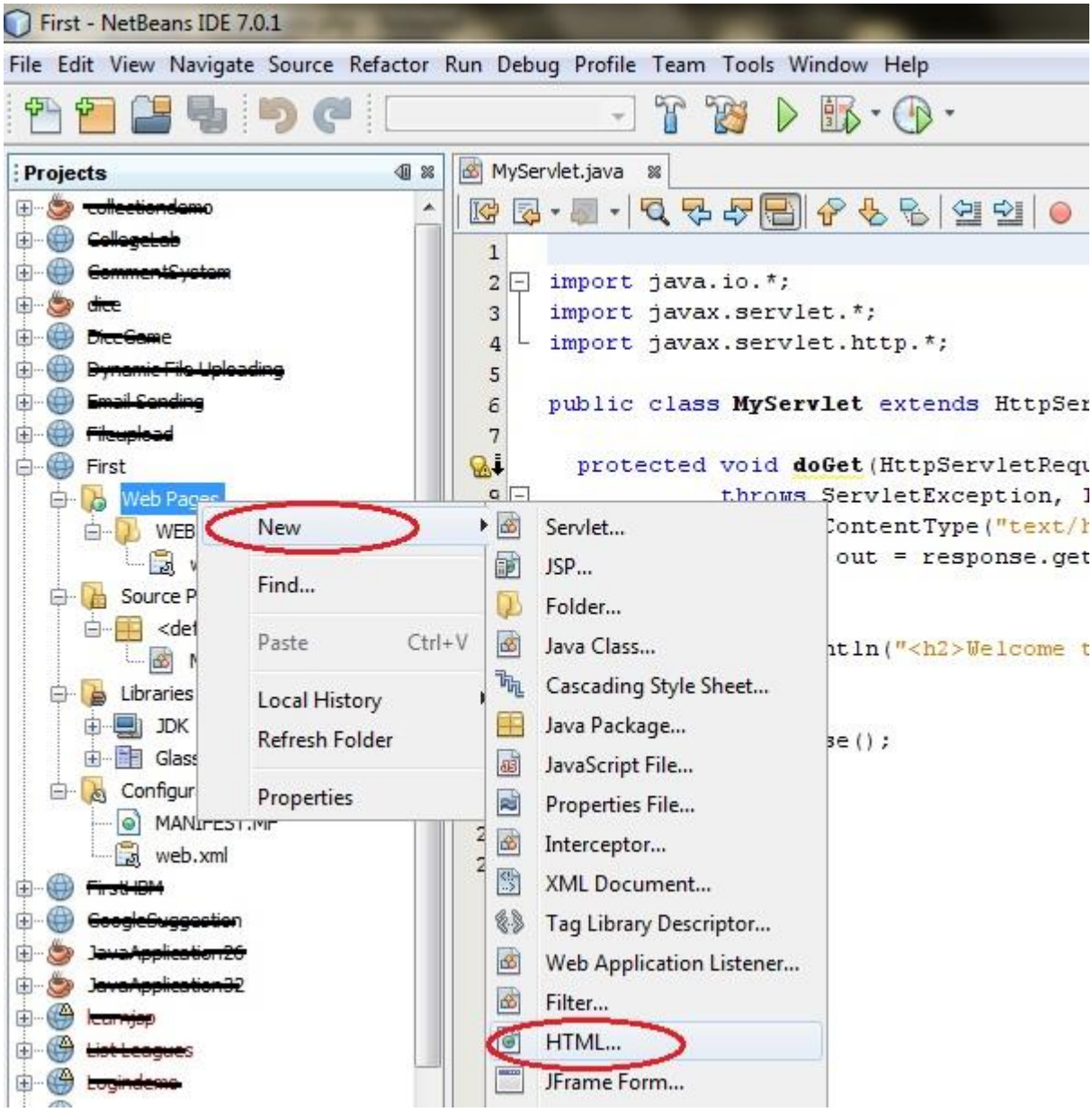
STEP 8. Now, your Servlet class is ready, and you just need to change the method definitions and you will good to go.



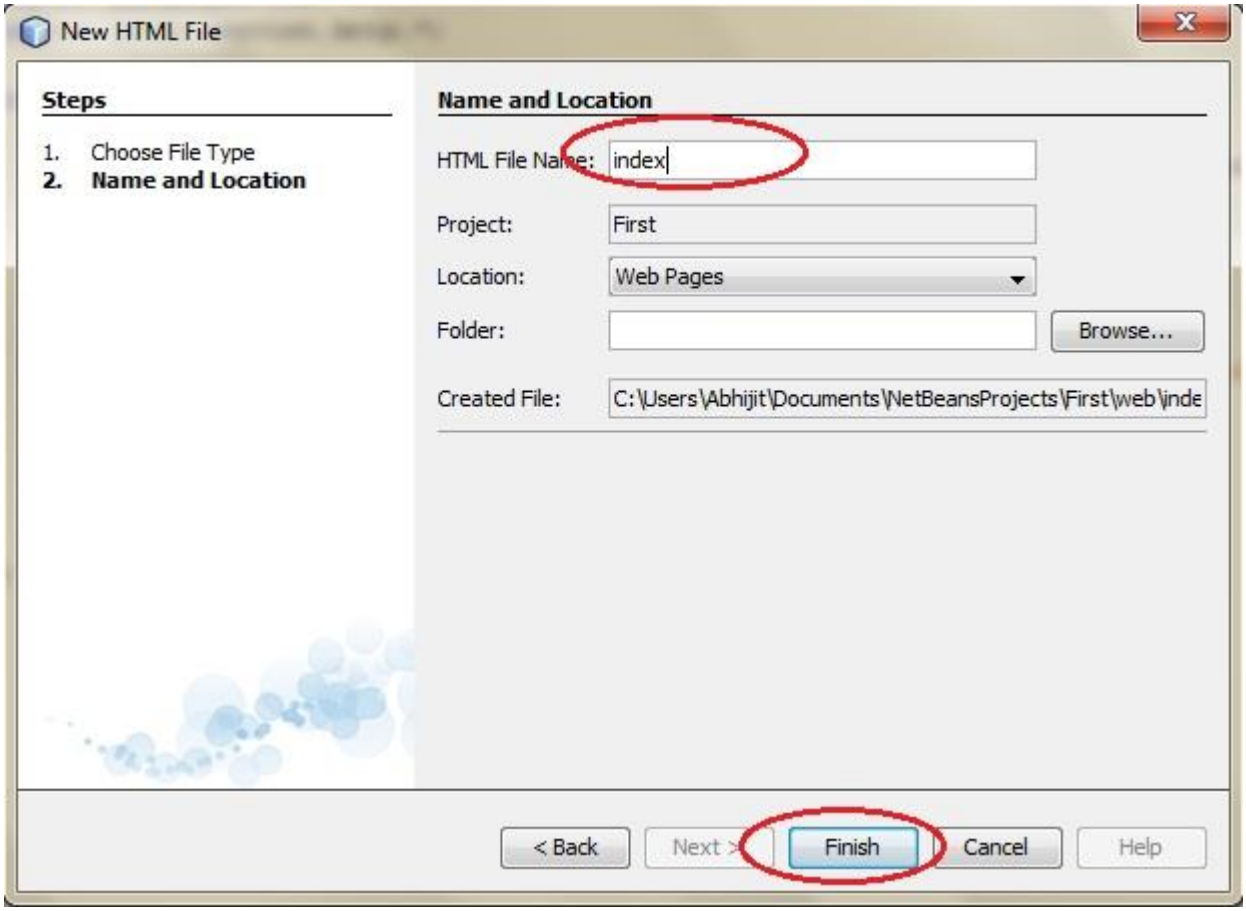
STEP 9. Write some code inside your Servlet class.



STEP 10. Create an HTML file, right click on **Web Pages** -> **New** -> **HTML**

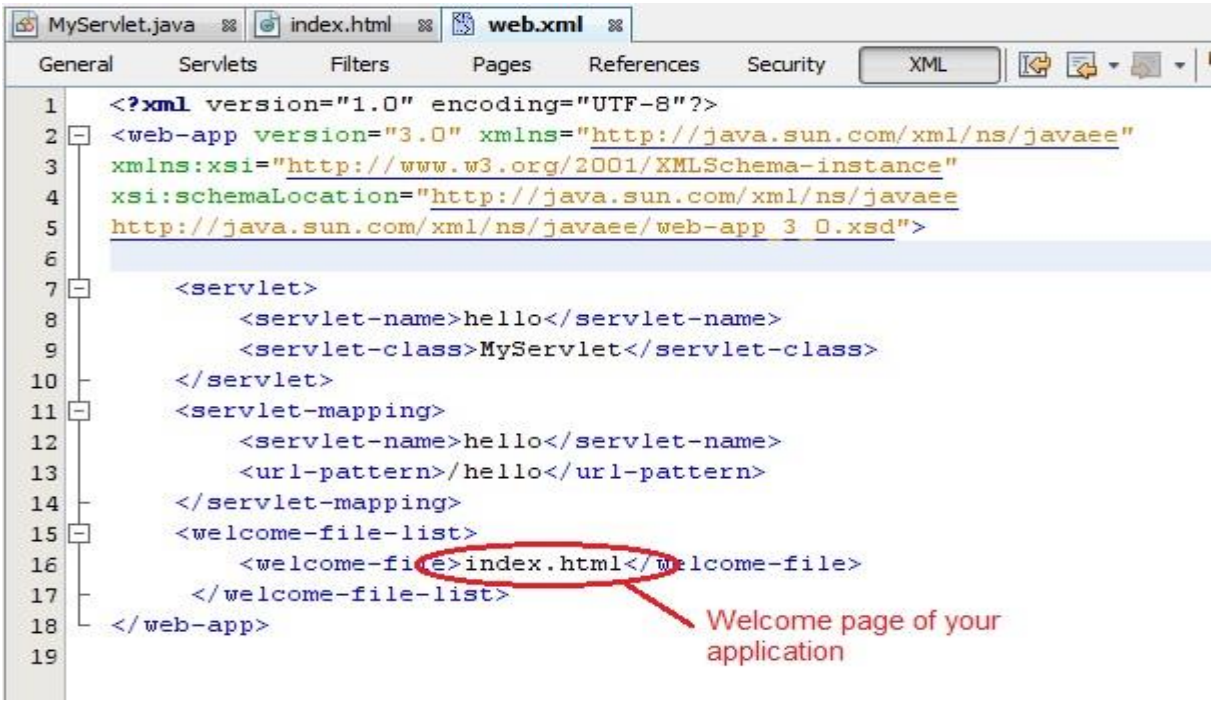


STEP 11. Give it a name. We recommend you to name it index, because browser will always pick up the index.html file automatically from a directory. Index file is read as the first page of the web application.

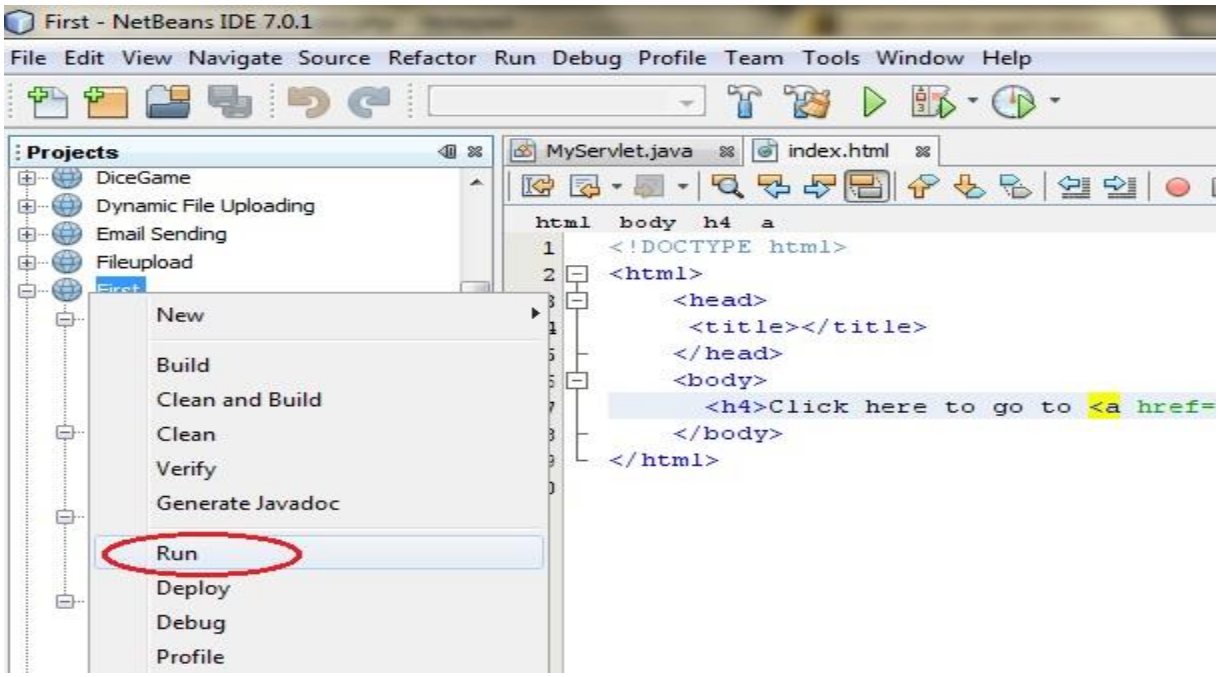


STEP 12. Write some code inside your HTML file. We have created a hyperlink to our Servlet in our HTML file.

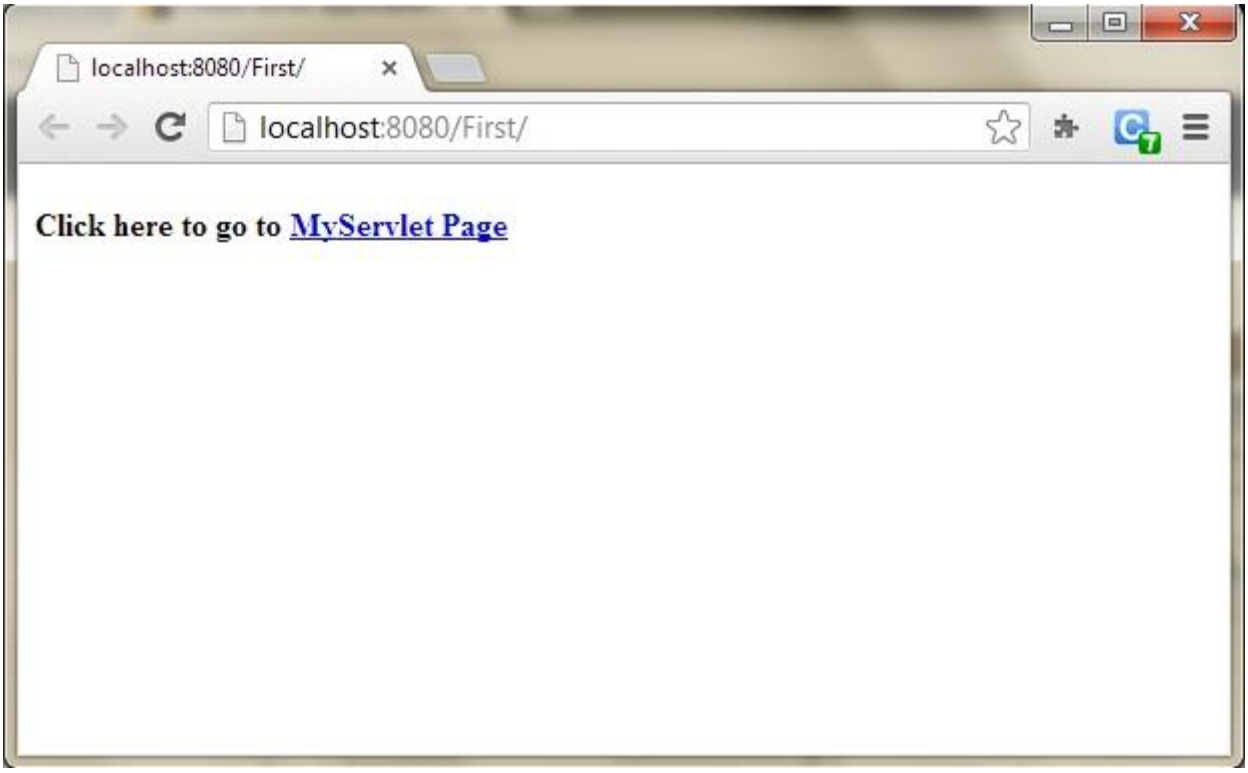
STEP 13. Edit **web.xml** file. In the web.xml file you can see, we have specified the **url-pattern** and the **servlet-name**, this means when hello url is accessed our Servlet file will be executed.



STEP 14. Run your application, right click on your Project and select **Run**



STEP 15. Click on the link created, to open your Servlet.



STEP 16. Hurray! Our First Servlet class is running.

SOURCE CODE:

SAMPLE OUTPUT:

TEST CASES:

RESULT: Successfully java servlet program is implemented and simple text information is displayed on screen.

EXPERIMENT 12

Write a servlet program to demonstrate Life cycle of Servlet.

AIM: To develop java servlet program to demonstrate Life Cycle of Servlet.

ALGORITHM:

STEP 1. Loading Servlet Class : A Servlet class is loaded when first request for the servlet is received by the Web Container.

STEP 2. Servlet instance creation :After the Servlet class is loaded, Web Container creates the instance of it. Servlet instance is created only once in the life cycle.

STEP 3. Call to the init() method : init() method is called by the Web Container on servlet instance to initialize the servlet.

Signature of init() method :

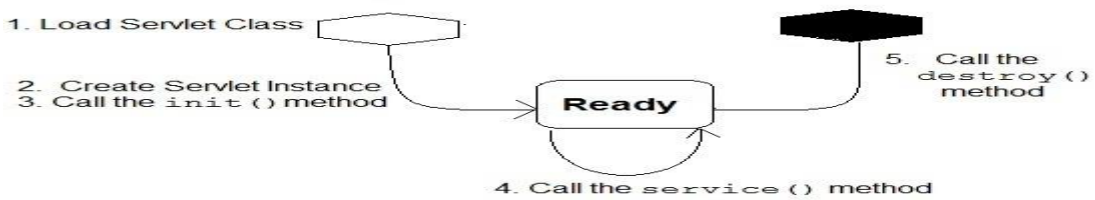
```
public void init(ServletConfig config) throws ServletException
```

STEP 4. Call to the service() method : The containers call the service() method each time the request for servlet is received. The service() method will then call the doGet() or doPost() methos based ont eh type of the HTTP request, as explained in previous lessons.

Signature of service() method :

```
public void service(ServletRequest request, ServletResponse response) throws ServletException, IOException
```

STEP 5. Call to destroy() method: The Web Container call the destroy() method before removing servlet instance, giving it a chance for cleanup activity.



SOURCE CODE:

SAMPLE OUTPUT:

TEST CASES:

RESULT: Successfully java servlet life cycle is demonstrated.

EXPERIMENT 13

Develop a Servlet to validate user name and password with the data stored in Servlet configuration file. Display authorized user if she/he is authorized else display unauthorized user.

AIM: To design and develop a servlet program to validate username and password and display corresponding messages on screen.

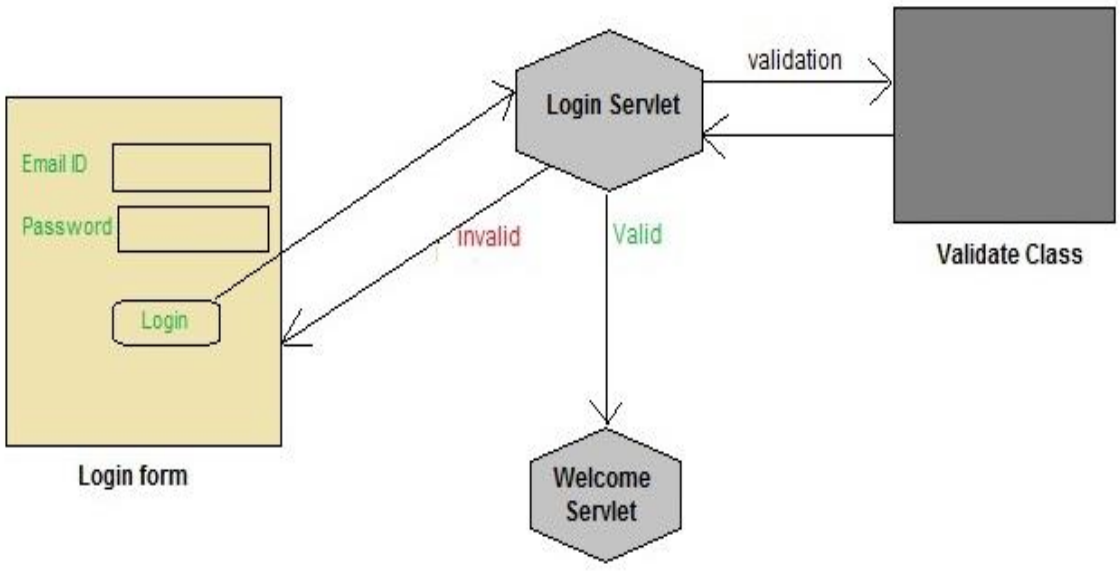
ALGORITHM:

STEP 1: Create a simple login page using HTML.

STEP 2: Use POST/GET method to navigate to validation servlet.

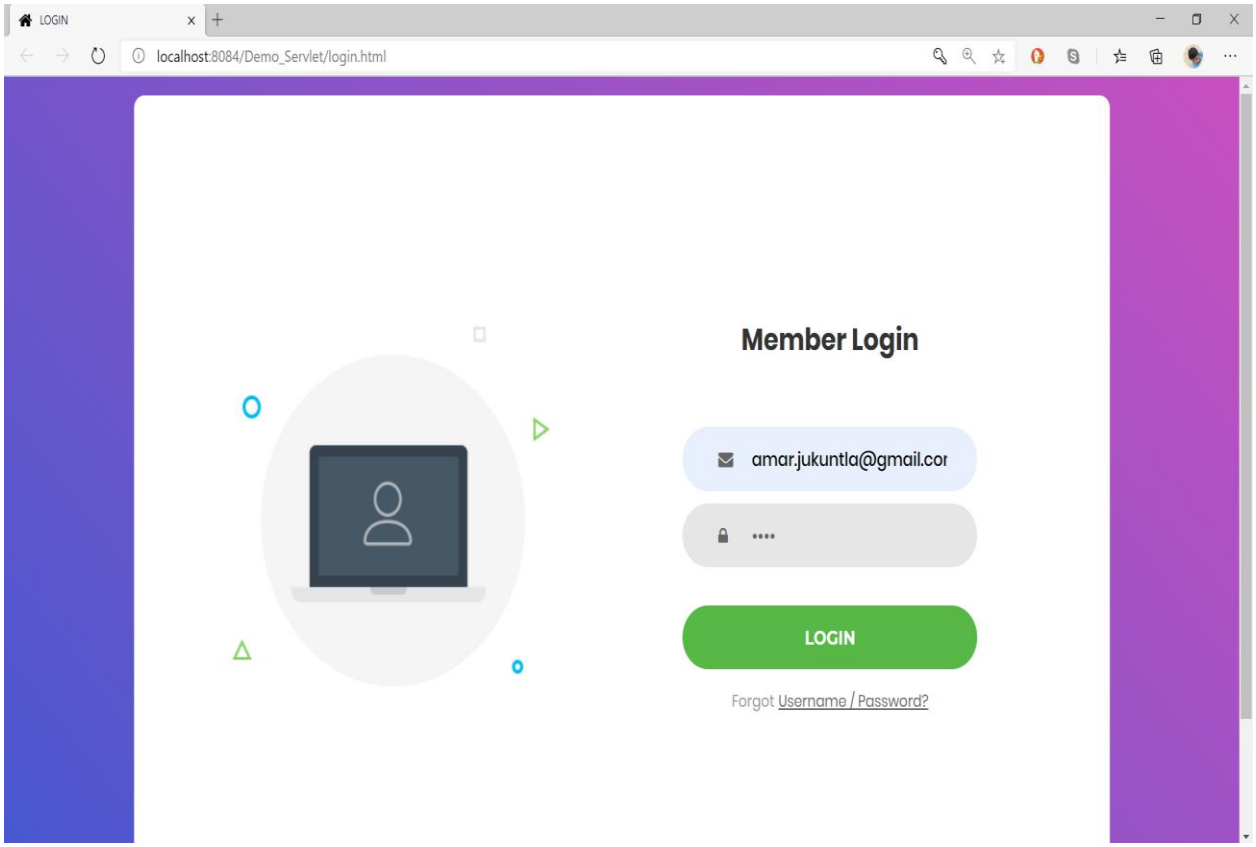
STEP 3: In validation servlet file retrieve data i.e., username and password from web.xml using getParameter() method.

STEP 4: Verify entered username and password is valid or not. If it is valid display authorized user or else display unauthorized user on screen using write buffer.



SOURCE CODE:

SAMPLE OUTPUT:



Login Successful

Invalid Username and Password

RESULT: Servlet is designed and developed for validating username and password through servlet config file.

EXPERIMENT 14

Write JSP Program to store student information sent from registration page into Database.

AIM: To design and develop a student registration page to store and display data from Database using JSP.

ALGORITHM:

STEP 1: Create a registration page using JSP.

STEP 2: Use POST/GET method to transfer student registration details from one page to another page.

STEP 3: Use JDBC connection for connecting to Database.

STEP 4: Retrieve student information from the registration page and store in database.

STEP 5: Once data is inserted into database successfully display entered data on screen.

SOURCE CODE:

SAMPLE OUTPUT:

Registration Form

First name	enter your first name
Last name	
Enter your password	
Enter your email	
Enter your mobile	
Enter your address	
Select your gender	Male Female
Select your hobbies	Watching Movies Playing Games Reading Books
Select your DOB	dd-mm-yyyy
Select your Country	Select your country
Upload your File	Choose File No file chosen
	Submit Reset

RESULT: Successfully JSP page created for storing and displaying information on screen.

EXPERIMENT 15

Write a JSP program to validate username and password through database.

AIM: To design and develop a JSP program to validate username and password and display corresponding messages on screen.

ALGORITHM:

STEP 1: Create a simple login page using JSP.

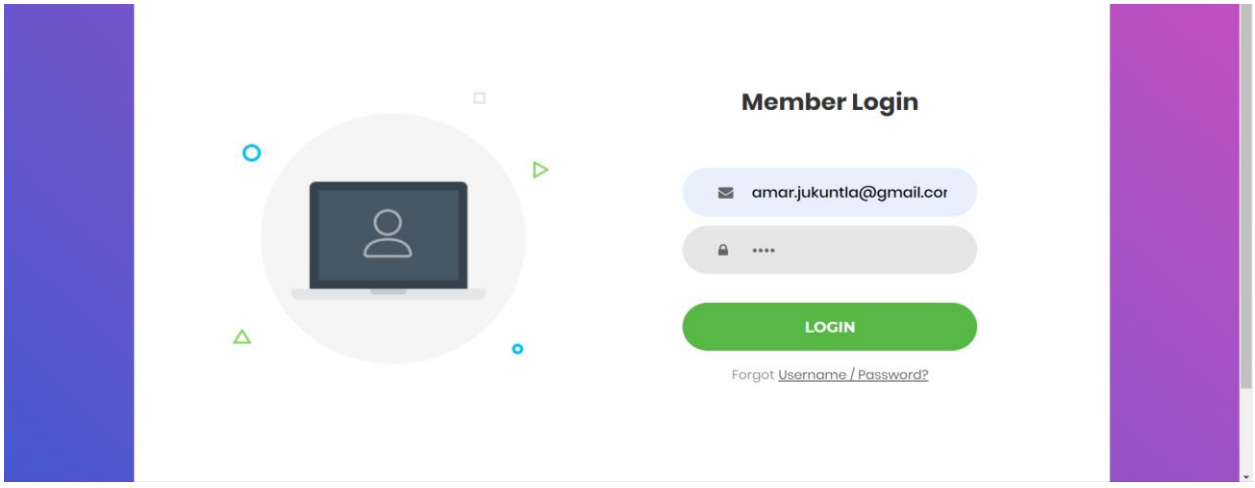
STEP 2: Use POST/GET method to navigate to validation servlet.

STEP 3: In validation JSP file retrieve data i.e., username and password from user login page.

STEP 4: Verify entered username and password is available in database or not. If it is valid display authorized user or else display unauthorized user on screen using write buffer.

SOURCE CODE:

SAMPLE OUTPUT:



Login Successful

Invalid Username and Password

RESULT: JSP is designed and developed for validating username and password through database.

EXPERIMENT 16

Write an appropriate JSP page to insert, update and delete data in student table in a single application with proper linking of JSP pages.

AIM: To design and develop a JSP program to data manipulation operation on student table.

ALGORITHM:

STEP 1: Create a simple JSP page that should contain three buttons.

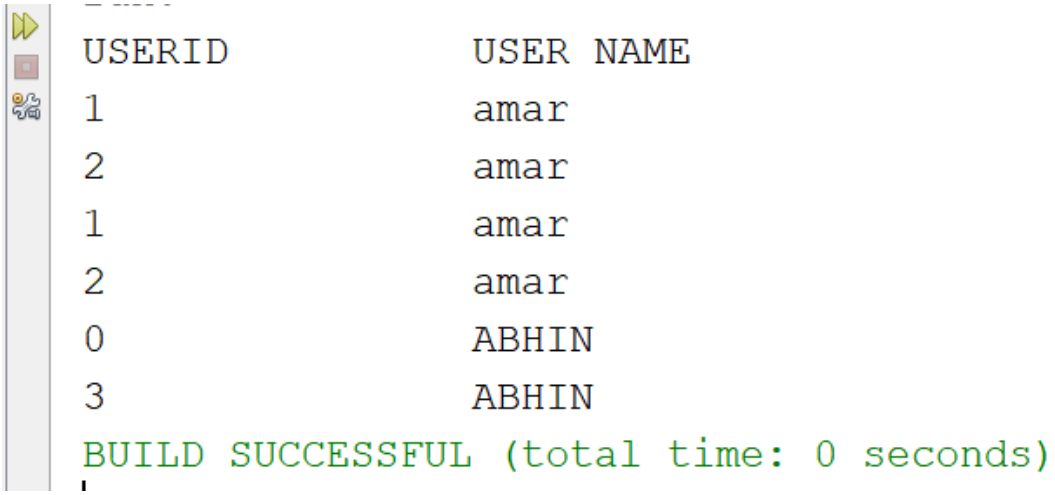
STEP 2: Use POST/GET method to navigate to validation JSP.

STEP 3: Based on your interaction with JSP page perform insert, delete and update operations in student table.

STEP 4: On user interaction use proper JSP page to perform corresponding operation.

SOURCE CODE:

SAMPLE OUTPUT:



RESULT: JSP is designed and developed for performing data manipulation operations on student table.

EXPERIMENT 17

Write PHP Program to store student information sent from registration page into Database.

AIM: To design and develop a student registration page to store and display data from Database using PHP.

ALGORITHM:

STEP 1: Create a registration page using HTML.

STEP 2: Use POST/GET method to transfer student registration details from one page to another page.

STEP 3: Use PHP mysql connection for connecting to Database.

STEP 4: Retrieve student information from the registration page and store in database.

STEP 5: Once data is inserted into database successfully display entered data on screen.

SOURCE CODE:

SAMPLE OUTPUT:

Registration Form

First name	enter your first name
Last name	
Enter your password	
Enter your email	
Enter your mobile	
Enter your address	
Select your gender	Male Female
Select your hobbies	Watching Movies Playing Games Reading Books
Select your DOB	dd-mm-yyyy
Select your Country	Select your country
Upload your File	Choose File No file chosen
	Submit Reset

RESULT: Successfully PHP page created for storing and displaying information on screen.

EXPERIMENT 18

Write a PHP program to validate username and password through database.

AIM: To design and develop a PHP program to validate username and password and display corresponding messages on screen.

ALGORITHM:

STEP 1: Create a simple login page using PHP.

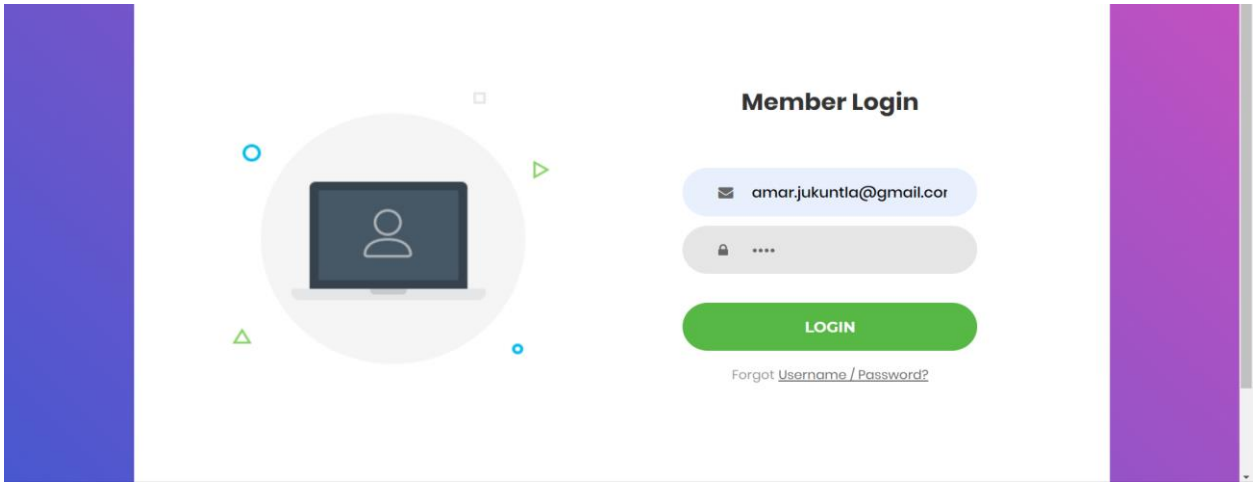
STEP 2: Use POST/GET method to navigate to validation servlet.

STEP 3: In validation PHP file retrieve data i.e., username and password from user login page.

STEP 4: Verify entered username and password is available in database or not. If it is valid display authorized user or else display unauthorized user on screen using write buffer.

SOURCE CODE:

SAMPLE OUTPUT:



Login Successful

Invalid Username and Password

RESULT: PHP login page is designed and developed for validating username and password through database.

EXPERIMENT 19

Write an appropriate PHP page to insert, update and delete data in student table in a single application with proper linking of PHP pages.

AIM: To design and develop a PHP program to data manipulation operation on student table.

ALGORITHM:

STEP 1: Create a simple PHP page that should contain three buttons.

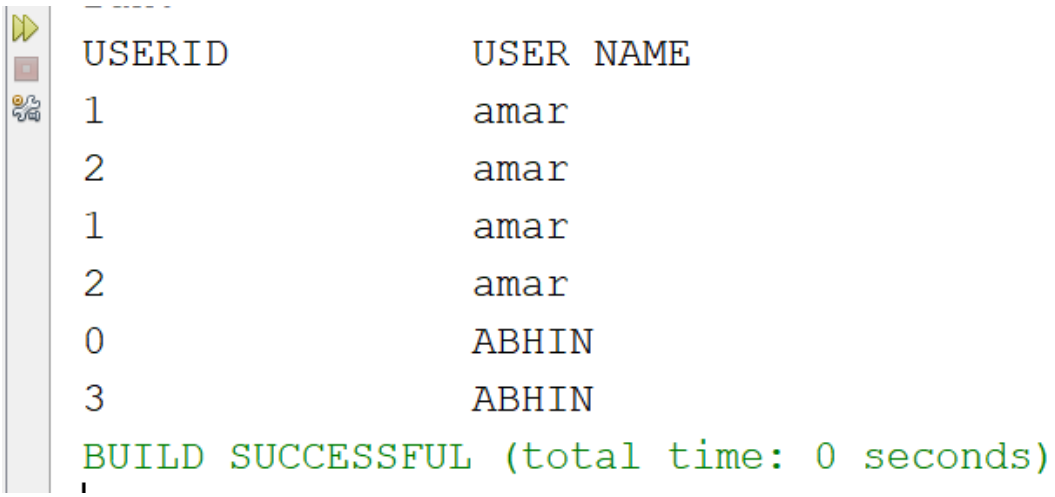
STEP 2: Use POST/GET method to navigate to validation PHP.

STEP 3: Based on your interaction with PHP page perform insert, delete and update operations in student table.

STEP 4: On user interaction use proper PHP page to perform corresponding operation.

SOURCE CODE:

SAMPLE OUTPUT:



RESULT: PHP page is designed and developed for performing data manipulation operations on student table.