

GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
Seshadrirao Knowledge Village, Gudlavalleru – 521356

DEPARTMENT OF INFORMATION TECHNOLOGY

WEB TECHNOLOGIES LAB
III Year I Semester

2019 - 20



Prepared by

Sri Y.K.Viswanadham
Associate Professor
Professor

Mr K.Srikanth
Assistant

GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institution with Permanent Affiliation to JNTUK, Kakinada)
Seshadri Rao Knowledge Village, Gudlavalleru – 521356
DEPARTMENT OF INFORMATION TECHNOLOGY

COLLEGE VISION & MISSION

Vision:

To be a leading institution of engineering education and research, preparing students for leadership in their fields in a caring and challenging learning environment.

Mission:

1. To produce quality engineers by providing state-of-the-art engineering education.
2. To attract and retain knowledgeable, creative, motivated and highly skilled individuals whose leadership and contributions uphold the college tenets of education, creativity, research and responsible public service.
3. To develop faculty and resources to impart and disseminate knowledge and information to students and also to society that will enhance educational level, which in turn, will contribute to social and economic betterment of society.
4. To provide an environment that values and encourages knowledge acquisition and academic freedom, making this a preferred institution for knowledge seekers.
5. To provide quality assurance.
6. To partner and collaborate with industry, government, and R&D institutes to develop new knowledge and sustainable technologies and serve as an engine for facilitating the nation's economic development.
7. To impart personality development skills to students that will help them to succeed and lead.
8. To instil in students the attitude, values and vision that will prepare them to lead lives of personal integrity and civic responsibility.
9. To promote a campus environment that welcomes and makes students of all races, cultures and civilizations feel at home.
10. Putting students face to face with industrial, governmental and societal challenges.

DEPARTMENT VISION & MISSION

VISION

To be a centre of innovation by adopting changes in Information Technology, imparting quality education, research to produce visionary computer professionals and entrepreneurs.

MISSION

- To provide an academic environment in which students are given the essential resources for solving real-world problems and work in multidisciplinary teams.
- To impart value based education and research among students, particularly belonging to rural areas, for their sustained growth in technological aspects and leadership.
- To collaborate with the industry for making the students adoptable to evolving changes in Information Technology and related areas.

PROGRAMME EDUCATIONAL OBJECTIVES(PEOs):-

PEO1:To exhibit analytical skills in modeling and solving computing problems by applying mathematical, scientific and engineering knowledge and to pursue their higher studies.

PEO2: To communicate effectively with multi-disciplinary teams to develop quality software systems with an orientation towards research and development for lifelong

PEO3: To address industry and societal needs for the growth of global economy using emerging technologies by following professional ethics.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES

Student will be able to

1. Organize, monitor and protect IT Infrastructural resources.
2. Design & Develop software solutions to the real world problems in the form of web, mobile and smart apps.

Course Objectives

- To introduce concepts of designing dynamic web pages using HTML, CSS and Javascript.
- To familiarize with JSP programming and master database access using JSP and JDBC.
- To impart PHP programming and master database access using PHP and MySQL.

Learning Outcomes

Upon successful completion of the course, the students will be able to

- design dynamic web pages using HTML, CSS and JavaScript.
- access and Validate form data using JavaScript and PHP.
- connect to database using JSP and JDBC and perform various operations.
- connect to MySQL using PHP and perform various operations

WEB TECHNOLOGIES LAB	1	2	3	4	5	6	7	8	9	10	11	12	PSO 1	PSO 2
CO1. design dynamic web pages using HTML, CSS and JavaScript.	3	2	2					2	2	2		2	2	3
CO2: access and Validate form data using JavaScript and PHP.	2	2	2					2	2	2		2	2	3
CO3: connect to database using JSP and JDBC and perform various operations.	2	2	2					2	2	2		2	2	3
CO4: connect to MySQL using PHP and perform various operations.	2	2	2					2	2	2		2	2	3

INDEX

S.NO	List of Experiments	PAGE .NO
1	Design the following static web pages required for an online book store web site. i. Home Page ii. Login Page iii. Catalogue Page	6-9
2	Design the following static web pages required for an online book store web site. i. Registration Page ii. Cart Page	10-14
3	Design a web page using CSS which includes the following: i. Use different font and text styles ii. Set a background image for both the page and single element on the page. iii. Define styles for links iv. Working with layers v. Adding a Customized cursor	15-22
4	i. Write a JavaScript to validate the fields of the login page. ii. Write a JavaScript to validate the fields of the Registration page	23-32
5	Write an XML file which will display the Book information which includes the following: Title of the book, Author Name, ISBN number, Publisher name, Edition and Price. Validate the above document using DTD and XML Schema.	33-39
6	i. Write a PHP program to validate the fields of the login page. ii. Write a PHP program to validate the fields of the Registration page	40-45
7	Write a JSP to connect to the database and extract data from the tables and display them to the user.	46-47
8	Design a JSP to insert the details of the users who register through the registration page and store the details in to the database.	48-50
9	Write a PHP program to connect to MySQL database which retrieves the data from the tables and display them to the user.	51-58
10	Write a PHP program to insert the details entered by the user in the Registration form into MySQL database.	59-60

ADDITIONAL EXPERIMENTS

S.NO	List of Experiments	PAGE .NO
1	Design a web page to show course Time Table using Table tag.	61-
2	Create a web page to display India map which navigates you to state web page when you click on state.	62

EXP NO: 1

AIM: Design the following static web pages required for an online book store web site.

i. Home Page ii. Login Page iii. Catalogue Page

1) HOME PAGE:

The static home page must contain three **frames**.

Top frame : Logo and the college name and links to Home page, Login page, Registration page, Catalogue page and Cart page (the description of these pages will be given below).

Left frame : At least four links for navigation, which will display the catalogue of respective links.

For e.g.: When you click the link “CSE” the catalogue for CSE Books should be displayed in the Right frame.

Right frame: The pages to the links in the left frame must be loaded here. Initially this page contains description of the web site.

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE ECE EEE CIVIL	Description of the Web Site			

Fig 1.1

2) LOGIN PAGE:



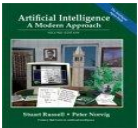





This page looks like below:

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE ECE EEE CIVIL	<p>Login : <input type="text"/></p> <p>Password: <input type="password"/></p> <p><input type="button" value="Submit"/> <input type="button" value="Reset"/></p>			

3) CATALOGUE PAGE:

The catalogue page should contain the details of all the books available in the web site in a table. The details should contain the following:

1. Snap shot of Cover Page.
2. Author Name.
3. Publisher.
4. Price.
5. Add to cart button.

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE		Book : XML Bible Author : Winston Publication : Wiely	\$ 40.5	
ECE				
EEE		Book : AI Author : S.Russel Publication : Princeton hall	\$ 63	
CIVIL		Book : Java 2 Author : Watson Publication : BPB publications	\$ 35.5	
		Book : HTML in 24 hours Author : Sam Peter Publication : Sam publication	\$ 50	

Source Code

Index.html

```
<frameset rows="200px,*">
  <frame src="top.html" />
  <frameset cols="150px,*">
    <frame src="left.html" />
    <frame src="home.html" name="body" id="body" />
  </frameset>
</frameset>
```

Top.html

```
<html>
```



```

    <head>
      <title>Top Page</title>
    </head>
    <body>
      <table width="100%">
        <tr><td></td><td
colspan="4"><h1>ONLINE BOOK STORE </h1></td>
        </tr><tr><td><br><br></td></tr>
        <tr ><td><a href="home.html" target="body">HOME</a></td>
        <td><a href="login.html" target="body">Login</a></td><td><a
href="registration.html" target="body">Registration</a></td>
        <td><a href="catalogue.html" target="body">Catalogue</a></td>
        <td><a href="cart.html" target="body">Cart</a></td>
        </tr>
      </table>
    </body>
  </html>

```

Left.html

```

<html>
<head>
<title>Top Page</title>
</head>

<body>
<a href="cse.html" target="body">CSE</a><br />
<a href="ece.html" target="body">ECE</a><br />
<a href="eee.html" target="body">EEE</a><br />
<a href="civil.html" target="body">CIVIL</a><br />
</body>
</html>

```

Login.html

```

<html>
<head>
<title>Login Page</title>
</head>
<body>
Username: <input name="uname" type="text" /><br />
Password: <input type="text" name="password" /><br />
<input type="submit" value="Submit" />&nbsp;&nbsp; <input type="reset" value="Reset" />
</body>
</html>

```

Catalogue.html

```

<html>
<head>
<title>Catalogue Page</title>
</head>
<body>
<table>
<tr><td><br /><br /></td>
<td>Book : XML Bible<br />Author : Winston<br />Publication : Wiely<br /></td>
<td>$ 40.5</td>

```

```

<td><input type="image" src="images/cart.jpg" /></td></tr>
<tr><td><br /><br /></td>
<td>Book : AI<br />Author : S.Russel<br />Publication : Princeton hall</td><td>$
63</td><td><input type="image" src="images/cart.jpg" /></td></tr>
<tr><td><br /><br /></td>
<td>Book : Java 2<br />Author : Watson<br />Publication : BPB publications</td>
<td>$ 35.5</td><td><input type="image" src="images/cart.jpg" /></td></tr>
<tr><td><br /><br /></td>
<td>Book : HTML in 24 hours<br />Author : Sam Peter<br />Publication : Sam
publication</td>
<td>$ 50</td><td><input type="image" src="images/cart.jpg" /></td></tr>
</table>
</body>
</html>

```

Home.html

```

<html>
<head>
<title>Home Page</title>
</head>
Description of Online Book Store
</body>
</html>

```

Execution Procedure

- Step1: type all html programs in notepad or any text editor and save .html format.
- Step2: open web browser and open index.html
- Step3: traverse all pages from index.html.

EXP NO: 2

Aim: Design the following static web pages required for an online book store web site.

i. Registration Page ii. Cart Page

CART PAGE:

The cart page contains the details about the books which are added to the cart.

The cart page should look like this:

Logo	Web Site Name			
Home	Login	Registration	Catalogue	Cart
CSE ECE EEE CIVIL	Book name	Price	Quantity	Amount
	Java 2	\$35.5	2	\$70
	XML bible	\$40.5	1	\$40.5
	Total amount -			\$130.5

5) REGISTRATION PAGE:

Create a “registration form “with the following fields

- 1) Name (Text field)
- 2) Password (password field)
- 3) E-mail id (text field)
- 4) Phone number (text field)
- 5) Sex (radio button)
- 6) Date of birth (3 select boxes)
- 7) Languages known (check boxes – English, Telugu, Hindi, Tamil)
- 8) Address (text area)

Description

Through table and form tag we can design pages.

The form tag you can add to your web pages a guestbook, order forms, surveys, get feedback or whatever. The basic construction of a html form is this...

- <FORM>** begin a form
- <INPUT>** ask for information in one of several different ways...
- <INPUT>** ...there can be as many input areas as you wish
- </FORM>** end a form

<TABLE>

The main tag. Used to tell the browser "this is a table", along with some attributes like size, border width and a few other things.

<TR>

TableRow defines a horizontal row of **<TD>** (TableData) *cells*.

<TD>

Specifies an individual block or *cell* in a table row

Source Code

Registraion.html

```

<html>
<head>

<title>Registration Page</title>
</head>
<body>
Name: <input type="text" name="fname" id="fname" /><br />
Password: <input type="text" name="password" id="password" /><br />
E-mail Id: <input type="text" name="email" id="email" /><br />
Phone Number: <input type="text" name="phone" id="phone" /><br />
Sex: &nbsp;Male<input type="radio" name="sex" id="sex" value="male" /> Female<input type="radio"
name="sex" id="sex" value="female" /><br />
Date of Birth: Day <select name="day" id="day">
<option>1</option>
<option>2</option>
<option>3</option>
<option>4</option>
<option>5</option>
<option>6</option>
<option>7</option>
<option>8</option>
<option>9</option>
<option>10</option>
<option>11</option>
<option>12</option>
<option>13</option>
<option>14</option>
<option>15</option>
<option>16</option>
<option>17</option>
<option>18</option>
<option>19</option>
<option>20</option>
<option>21</option>
<option>22</option>
<option>23</option>
<option>24</option>
<option>25</option>
<option>26</option>
<option>27</option>
<option>28</option>
<option>29</option>
<option>30</option>
<option>31</option>
</select>

Month <select name="month" id="month" >
<option>Jan</option>
<option>Feb</option>
<option>Mar</option>
<option>Apr</option>

```

```
<option>May</option>
<option>Jun</option>
<option>Jul</option>
<option>Aug</option>
<option>Sep</option>
<option>Oct</option>
<option>Nov</option>
<option>Dec</option>
</select>
```

```
Year <select name="year" id="year" >
```

```
<option>1971</option>
<option>1972</option>
<option>1973</option>
<option>1974</option>
<option>1975</option>
<option>1976</option>
<option>1977</option>
<option>1978</option>
<option>1979</option>
<option>1980</option>
<option>1981</option>
<option>1982</option>
<option>1983</option>
<option>1984</option>
<option>1985</option>
<option>1986</option>
<option>1987</option>
<option>1988</option>
<option>1989</option>
<option>1990</option>
<option>1991</option>
<option>1992</option>
<option>1993</option>
<option>1994</option>
<option>1995</option>
<option>1996</option>
<option>1997</option>
<option>1998</option>
<option>1999</option>
<option>2000</option>
<option>2001</option>
<option>2002</option>
<option>2003</option>
<option>2004</option>
<option>2005</option>
<option>2006</option>
<option>2007</option>
<option>2008</option>
<option>2009</option>
<option>2010</option>
<option>2011</option>
<option>2012</option>
</select><br />
```

```
Languages known: English <input type="checkbox" name="english" id="english" />
```

```
Telugu <input type="checkbox" name="telugu" id="telugu" />
```

```
Hindi <input type="checkbox" name="hindi" id="hindi" />
Tamil <input type="checkbox" name="tamil" id="tamil" /><br />
```

```
Address: <textarea name="address" id="address" rows="5" cols="20" ></textarea><br /><br />
```

```
<input type="submit" value="Submit" /><input type="reset" value="Reset" />
</body>
</html>
```

Cart.html

```
<html>
<head>

<title>Cart Page</title>
</head>
<body>
<table ><tr><th>Book name</th><th>Price</th><th>Quantity</th><th>Amount</th></tr>
<tr><td>Java 2</td><td>$35.5</td><td>2</td><td>$70</td></tr>
<tr><td>XML bible</td><td>$40.5</td><td>1</td><td>$40.5</td></tr>
<tr><td colspan="3" align="right"><b>Total amount -</b></td><td>$130.5</td></tr>
</table>
</body>
</html>
```

CSE.html

```
<html>
<head>

<title>CSE Page</title>
</head>

<p>Department of CSE Description</p>
</body>
</html>
```

Ece.html

```
<html>
<head>

<title>ECE Page</title>
</head>

<p>Department of ECE Description</p>
</body>
</html>
```

Eee.html

```
<html>
<head>

<title>EEE Page</title>
</head>

<p>Department of EEE Description</p>
```

```
</body>  
</html>
```

Civil.html

```
<html>  
<head>
```

```
<title>CIVIL Page</title>  
</head>
```

```
<p>Department of CIVIL Description</p>  
</body>  
</html>
```

Execution Procedure

Step1: type all html programs in notepad or any text editor and save .html format.

Step2: open web browser and open index.html, i.e previous week.

Step3: traverse all pages from index.html.

EXP NO: 3

Design a web page using CSS which includes the following:

- i. Use different font and text styles
- ii. Set a background image for both the page and single element on the page.
- iii. Define styles for links
- iv. Working with layers
- v. Adding a Customized cursor

1) Use different font, styles:

In the style definition you define how each selector should work (font, color etc.). Then, in the body of your pages, you refer to these selectors to activate the styles.

For example:

```
<HTML>
<HEAD>
<style type="text/css">
B.headline {color:red, font-size:22px, font-family:arial, text-decoration:underline}
</style>
</HEAD>
<BODY>
<b>This is normal bold</b><br>
Selector {cursor:value}
```

For example:

```
<html>
<head>
<style type="text/css">
.xlink {cursor:crosshair}
.hlink{cursor:help}
</style>
</head>
<body>
<b>
<a href="mypage.htm" class="xlink">CROSS LINK</a>
<br>
<a href="mypage.htm" class="hlink">HELP LINK</a>
</b>
</body>
</html>
<b class="headline">This is headline style bold</b>
</BODY>
</HTML>
```


2) Set a background image for both the page and single elements on the page. You can define the background image for the page like this:

```
BODY {background-image:url(myimage.gif),}
```

3) Control the repetition of the image with the background-repeat property. As background-repeat: repeat Tiles the image until the entire page is filled, just like an ordinary background image in plain HTML.

4) Define styles for links as

```
A:link
A:visited
A:active
A:hover
```

Example:

```
<style type="text/css">
A:link {text-decoration: none}
A:visited {text-decoration: none}
A:active {text-decoration: none}
A:hover {text-decoration: underline, color: red,}
</style>
```

5) Work with layers:

For example:

LAYER 1 ON TOP:

```
<div style="position:relative, font-size:50px, z-index:2,">LAYER 1</div> <div
style="position:relative, top:-50, left:5, color:red, font-size:80px, z-
index:1">LAYER 2</div>
```

LAYER 2 ON TOP:

```
<div style="position:relative, font-size:50px, z-index:3,">LAYER 1</div> <div
style="position:relative, top:-50, left:5, color:red, font-size:80px, z-
index:4">LAYER 2</div>
```

6) Add a customized cursor:

```
Selector {cursor:value}
```

For example:

```
<html>
<head>
<style type="text/css">
.xlink {cursor:crosshair}
.hlink{cursor:help}
</style>
</head>

<body>
<b>
<a href="mypage.htm" class="xlink">CROSS LINK</a>
<br>
<a href="mypage.htm" class="hlink">HELP LINK</a>
</b>
</body>
</html>
```

Description

- **CSS** stands for **Cascading Style Sheets**
- Styles define **how to display** HTML elements
- Styles are normally stored in **Style Sheets**
- Styles were added to HTML 4.0 **to solve a problem**
- **External Style Sheets** can save you a lot of work
- External Style Sheets are stored in **CSS files**
- Multiple style definitions will **cascade** into one

HTML tags were originally designed to define the content of a document.

As the two major browsers - Netscape and Internet Explorer - continued to add new HTML tags and attributes (like the tag and the color attribute) to the original HTML specification, it became more and more difficult to create Web sites where the content of HTML documents was clearly separated from the document's presentation layout.

To solve this problem, the World Wide Web Consortium (W3C) - the non profit, standard setting consortium, responsible for standardizing HTML - created STYLES in addition to HTML 4.0.

Source Code

1) Use different font, styles:

```
<html>
<head>
<style type="text/css">
.xlink{cursor:crosshair}
.hlink{cursor:help}
b.headline
{
color:red;
font-size:24px;
font-family:arial;
text-decoration:underline;
}
</style>
</head>
<body>
<b><a href="mypage.html" class="xlink"> crosslink</a><br>
<a href="mypage.html" class="hlink">helpink</a><br></b>
<b class="headline">this is heading style bold</b>
</body>
</html>
```

2) Set a background image

```
<html>
<head>
<style type="text/css">
```

```

body
{
background-image:url(logo.jpg);
}
.paragraph
{
background-image:url(logo.jpg);
}
</style>
</head>
<body>
<p>Gudlavalleru Engineering College</p>
</body></html>

```

3. Control the repetition of the image with the background-repeat property.

```

<html>
<head>
<style type="text/css">
body
{
background-image:url(logo.jpg);
background-repeat:no-repeat;
}
.pragraph
{
background-image:url(logo.jpg);
}
</style>
</head>
<body>
<p> MITS, madanapalle</p>
</body>
</html>

```

4) Define styles for links

```

<html>
<head>
<style type="text/css">
A:link
{
color:black;
text-decoration:none;
}
A:visited
{
color:black;
text-decoration:none;
}
A:hover
{

```

```

color:red;
text-decoration:underline;
}
A:active
{
color:red;
text-decoration:underline;
}
</style>
</head>
<body>
<a href="cse.html">cse</a>
<br>
<a href="it.html">it</a><br>
<a href="ece.html">ece</a><br>
<a href="me.html">me</a><br>
</body>
</html>

```

5) Work with layers

```

<html>
<head>
<body>
<strong>LAYER1 ON TOP</strong>
<div style="position:relative;
font-size:30px;
z-index:2;
color:orange">
TOPLAYER</div>
<div style="position:relative; font-size:30px;top:-50px;left:10px;
z-index:1;color:orange">BOTTOMLAYER</div>
<strong> layer2 on top</strong>
<div style="position:relative;
font-size:30px;
z-index:3">TOPLAYER</div>
<div style="position:relative;
font-size:30px;
top:-50px;
left:10px;
z-index:4">
BOTTOMLAYER</div>
</body>
</html>

```

6) Add a customized cursor

```

<html>
<head>
<style type="text/css">
.xlink{cursor:crosshair}
.hlink{cursor:help}
</style>

```

```

</head>
<body>
<b><a href="homepage.html" class="xlink">CROSSLINK</a><br>
<a href="homepage.html" class="hlink">HELPLINK</a><br>
</b>
</body>
</html>

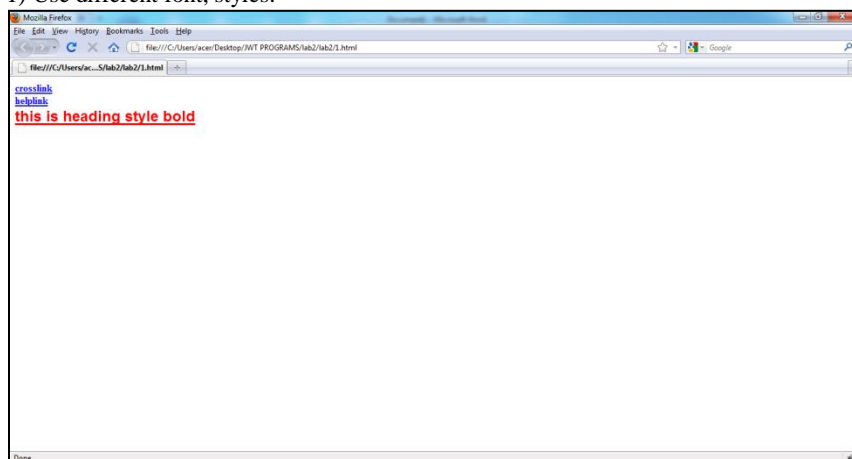
```

Execution procedure

Step1: type all html programs in notepad or any text editor and save .html format.
 Step2: open web browser and open .html files.

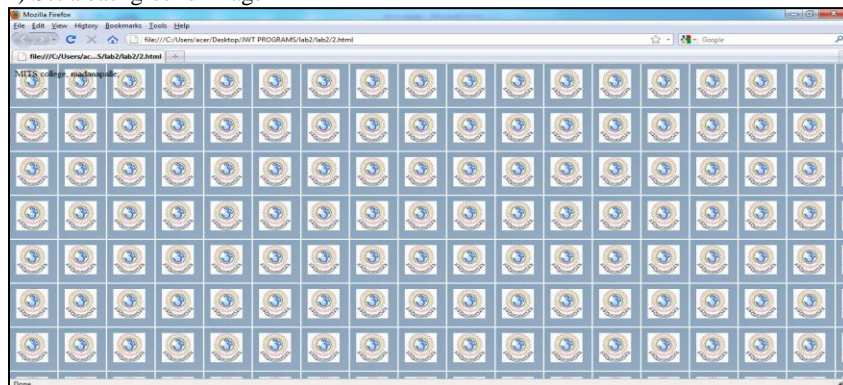
Output

1) Use different font, styles:



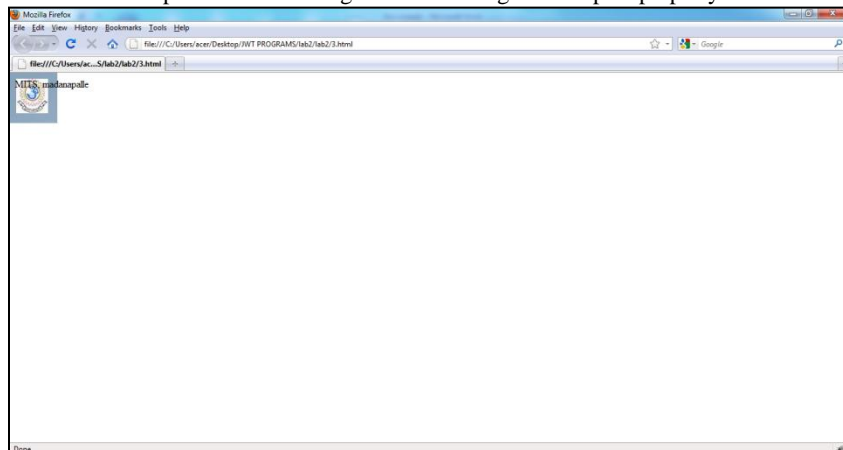
Use different font, styles

2) Set a background image



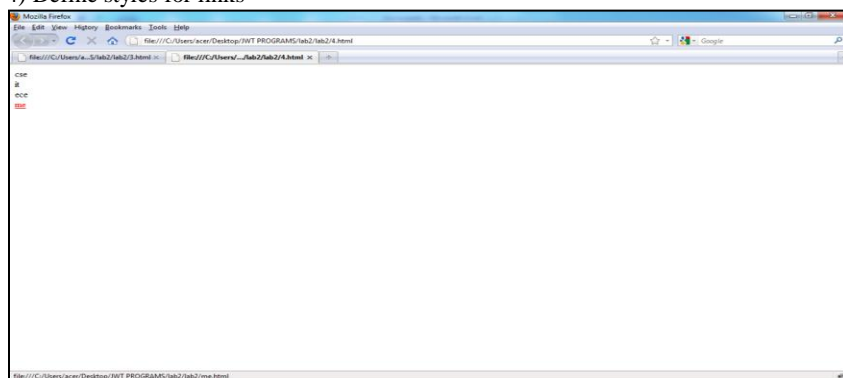
Set a background image

3. Control the repetition of the image with the background-repeat property.



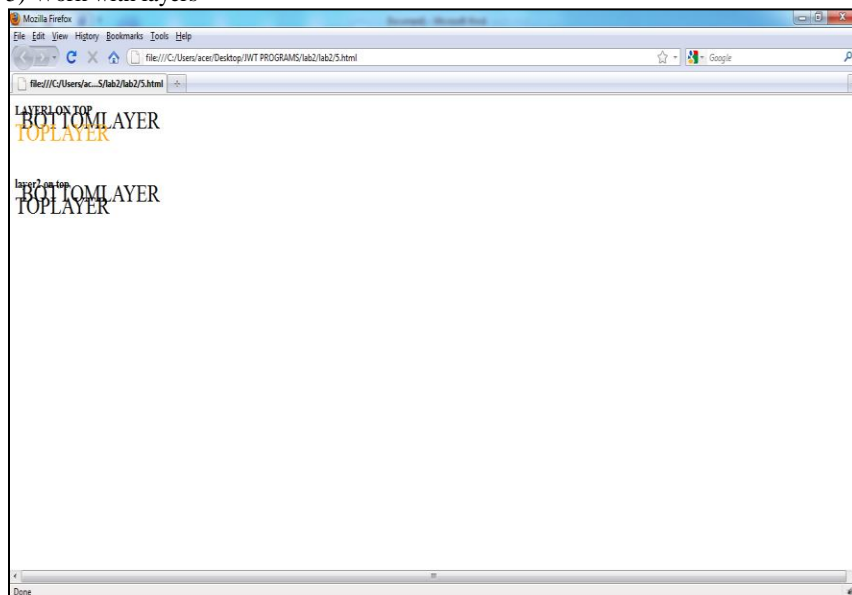
Control the repetition of the image with the background-repeat property

4) Define styles for links



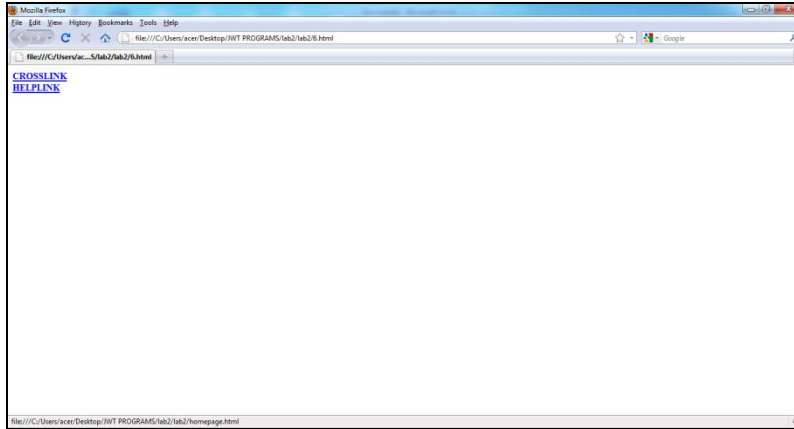
Define styles for links

5) Work with layers



Work with layers

6) Add a customized cursor



Add a customized cursor

EXP NO: 4**AIM:**

- i. Write a JavaScript to validate the fields of the login page.
- ii. Write a JavaScript to validate the fields of the Registration page

Description

JavaScript is a lightweight interpreted programming language with rudimentary object-oriented capabilities. Syntactically, the core JavaScript language resembles C, C++ and Java, with programming constructs such as the `if` statement, the `while` loop, and the `&&` operator. The similarity ends with this syntactic resemblance, however. JavaScript is an untyped language, which means that variables do not have to have a type specified.

The `<SCRIPT>` and `</SCRIPT>` tags are used to embed JavaScript code within an HTML file.

Source CodeRegistration.html

```

<html>
<head>
<title>Registration Page</title>
<script type="text/javascript">
varerrorFound=false;
functioncontainAlphabets(checkString)
{
//To validate Name Field characters
    var result=true;
    varregExp=/^[A-Z a-z]$/;
    if(checkString!=null &&checkString!="")
    {
        for(var i=0;i<checkString.length;i++)
        {
            if(!checkString.charAt(i).match(regExp))
            {
                result=false;
            }
        }
    }
    else
    {
        result=false;
    }
    return result;
}
functionvalidateNameLength(checkString)
{

```



```
//To validate Name Field length
if(checkString.length>=6) {
    return true;
}
else
{
    return false;
}
```

```
}  
function validatePasswordLength(password)  
{  
//To validate Password Field length  
    if(password.length>=6)  
    {  
        return true;  
    }  
    else  
    {  
        return false;  
    }  
}  
function validateEmail(email)  
{  
var x=document.regform.email.value;  
var atpos=x.indexOf("@");  
var dotpos=x.lastIndexOf(".");  
if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length)  
    {  
alert("Not a valid e-mail address");  
return false;  
    }  
  
return true;  
}  
  
function validatePhoneNumber(phone)  
{  
    var validDigits="0123456789";  
    var digits;  
    var result;  
    if(phone.length==0)  
        return false;  
    for(var i=0;i<phone.length;i++)  
    {  
        digits=phone.charAt(i);  
        if(validDigits.indexOf(digits)==-1)  
        {  
            return false;  
        }  
        result=true;  
    }  
    return result;  
}  
function validatePhoneNoLength(phone)
```

```
{
    if(phone.length==10)
        return true;
    else
        return
}

function validate()
{
    errorFound=false;
    if(!containAlphabets(document.regform.userName.value))
    {
        alert("Please Enter only alphabets for name field");
    }
    if(!validateLength(document.regform.passwordField.value))
    {
        alert("Please Enter minimum 6 characters for password");
    }
    if(!validateEmail(document.regform.email.value))
    {
        alert("Please Enter email");
    }
    if(!validatePhoneNumber(document.regform.phone.value))
    {
        alert("Please Enter valid phone number");
    }
    if(!validatePhoneNoLength(document.regform.phone.value))
    {
        alert("Please Enter 10 digit phone number");
    }
}

function checkName()
{
    if(!containAlphabets(document.regform.userName.value))
    {
        alert("Please Enter only alphabets for name field");
        document.regform.userName.value="";
        document.regform.userName.focus();
    }
    else
    if(!validateNameLength(document.regform.userName.value))
    {
        alert("Please Enter atleast 6 characters for name field");
        document.regform.userName.focus();
    }
}
```

```

    }
}
functioncheckPassword()
{
    if(!validatePasswordLength(document.regform.passwordField.value))
    {
        alert("Please Enter minimum 6 characters for password");
        document.regform.passwordField.value="";
        document.regform.passwordField.focus();
    }
}
functioncheckEmail()
{
    if(!validateEmail(document.regform.email.value))
    {
        //alert("Please Enter only valid Email ID");
        //document.regform.email.value="";
        document.regform.email.focus();
    }
}
functioncheckPhone()
{
    if(!validatePhoneNumber(document.regform.phone.value))
    {
        alert("Please Enter valid phone number");
        document.regform.phone.value="";
        document.regform.phone.focus();
    }else
    if(!validatePhoneNoLength(document.regform.phone.value))
    {
        alert("Please Enter 10 digit phone number");
        document.regform.phone.focus();
    }
}
</script>
</head>
<body>
<h1>Registration Here</h1>
<form name="regform">
    Name:
<input type="text" name="userName" />
<br />
    Password:
<input type="password" name="passwordField" onFocus="checkName()" />
<br />
    Email:

```

```

<input type="text" name="email" onFocus="checkPassword()" />
<br />
phone:
<input type="text" maxlength="10" name="phone" onFocus="checkEmail()" />
<br />
Gender:
<input type="radio" name="gender" value="male" onFocus="checkPhone()" />
Male &nbsp;
<input type="radio" name="gender" value="female" onFocus="checkPhone()" />
Female <br />
Date of Birth: Day
<select name="dd" style="width:75px" >
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>
<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
<option value="13">13</option>
<option value="14">14</option>
<option value="15">15</option>
<option value="16">16</option>
<option value="17">17</option>
<option value="18">18</option>
<option value="19">19</option>
<option value="20">20</option>
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
</select>
Month
<select name="mm" style="width:75px">

```

```

<option value="1">January</option>
<option value="2">February</option>
<option value="3">March</option>
<option value="4">April</option>
<option value="5">May</option>
<option value="6">June</option>
<option value="7">July</option>
<option value="8">August</option>
<option value="9">September</option>
<option value="10">October</option>
<option value="11">November</option>
<option value="12">December</option>
</select>

```

Year

```

<select name="yy" style="width:75px">
<option value="2000">2000</option>
<option value="2001">2001</option>
<option value="2002">2002</option>
<option value="2003">2003</option>
<option value="2004">2004</option>
<option value="2005">2005</option>
<option value="2006">2006</option>
<option value="2007">2007</option>
<option value="2008">2008</option>
<option value="2009">2009</option>
<option value="2010">2010</option>
<option value="2011">2011</option>
<option value="2012">2012</option>
<option value="2013">2013</option>
</select>

```


Languages Known:

```

<input type="checkbox" name="language" value="english" />

```

English

```

<input type="checkbox" name="language" value="hindi" />

```

Hindi

```

<input type="checkbox" name="language" value="telugu" />

```

Telugu

```

<input type="checkbox" name="language" value="tamil" />

```

Tamil

Address:

```

<textarea rows="5" cols="20"></textarea>

```



```

<input type="submit" value="Submit" onFocus=validate() />

```



```

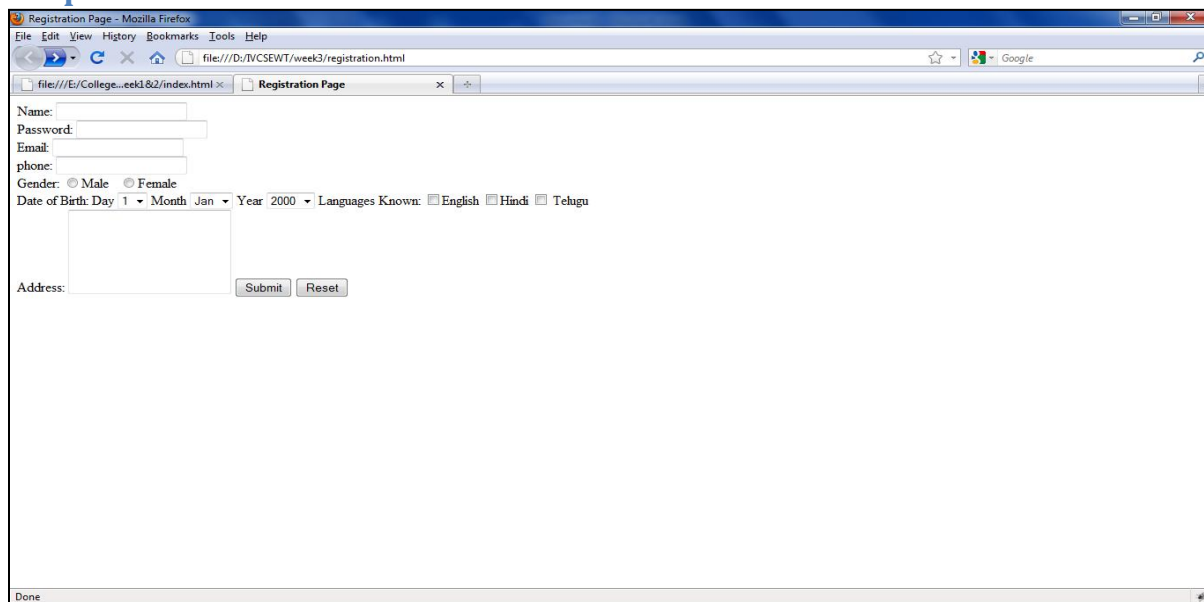
<input type="reset" value="Reset" />
</form>
</body>
</html>

```

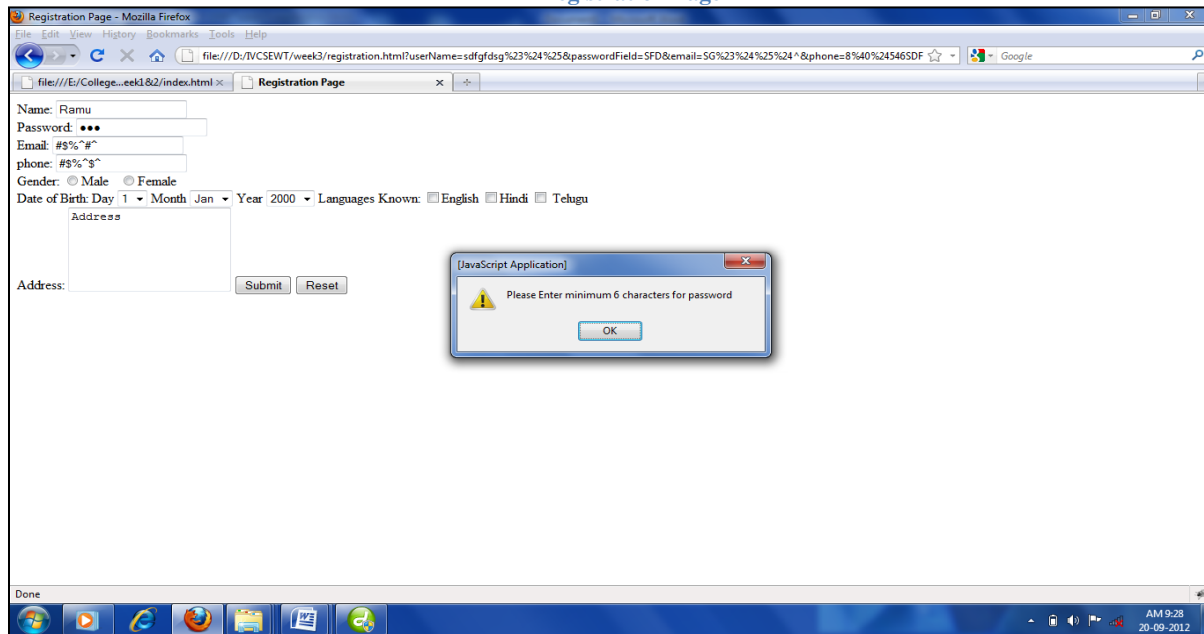
Execution Procedure

- Step1: type the html program in notepad or any text editor and save .html format.
- Step2: open web browser and open .html.

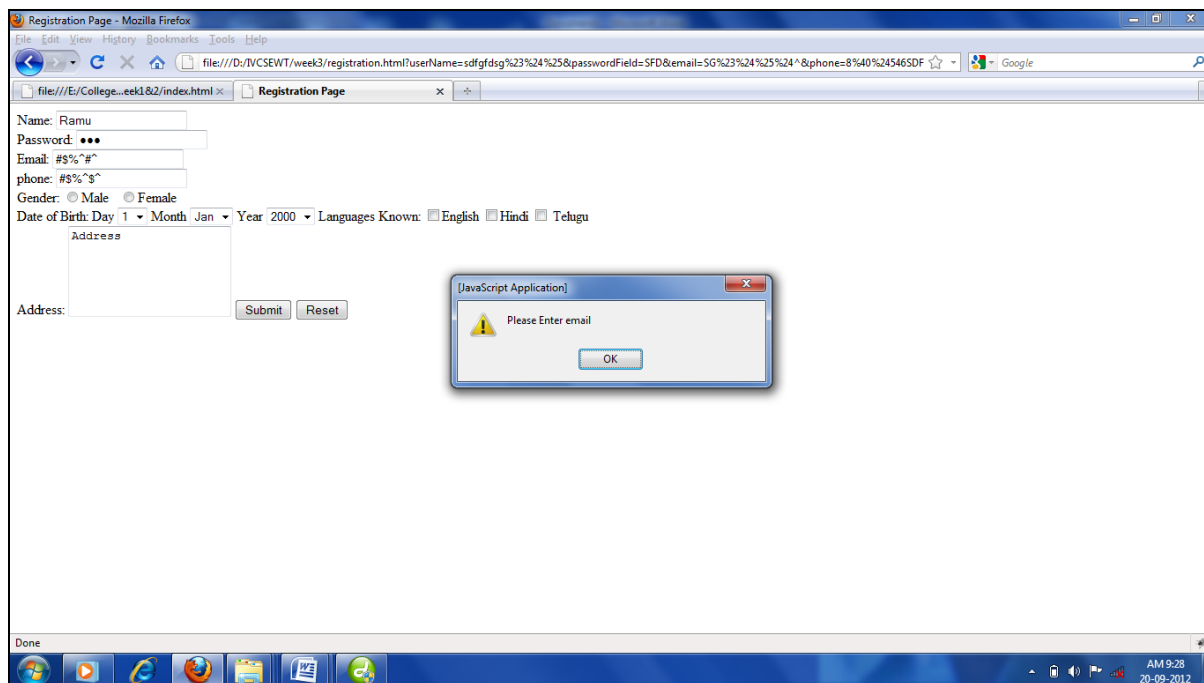
Output



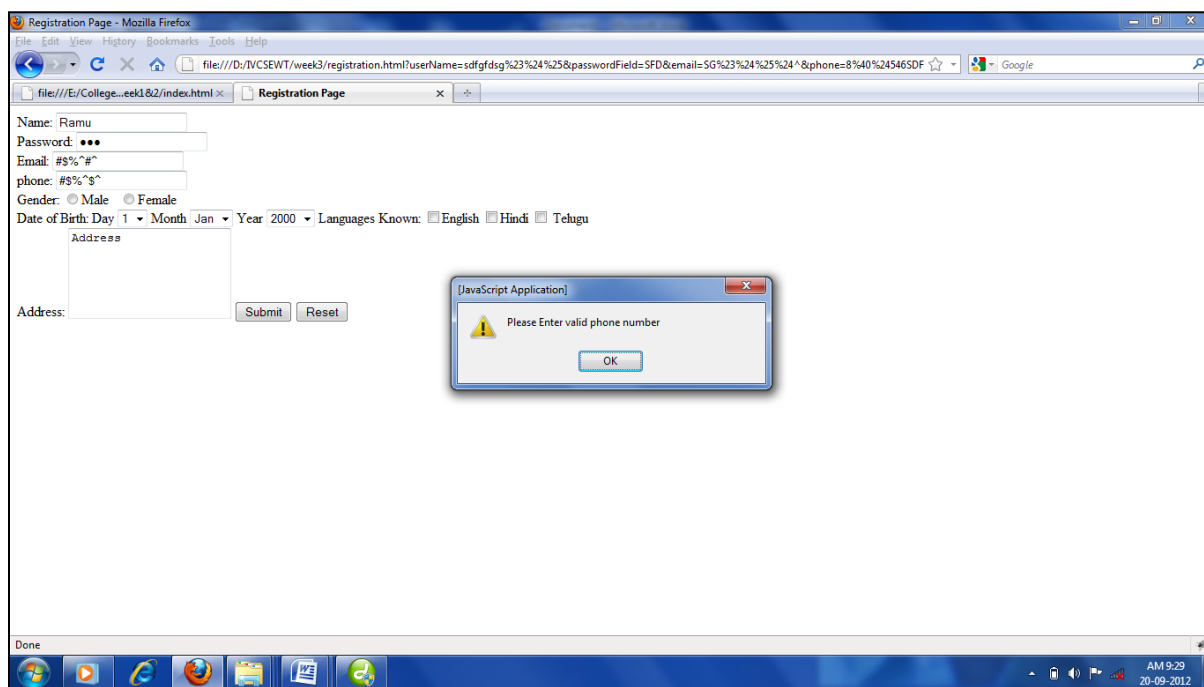
Registration Page



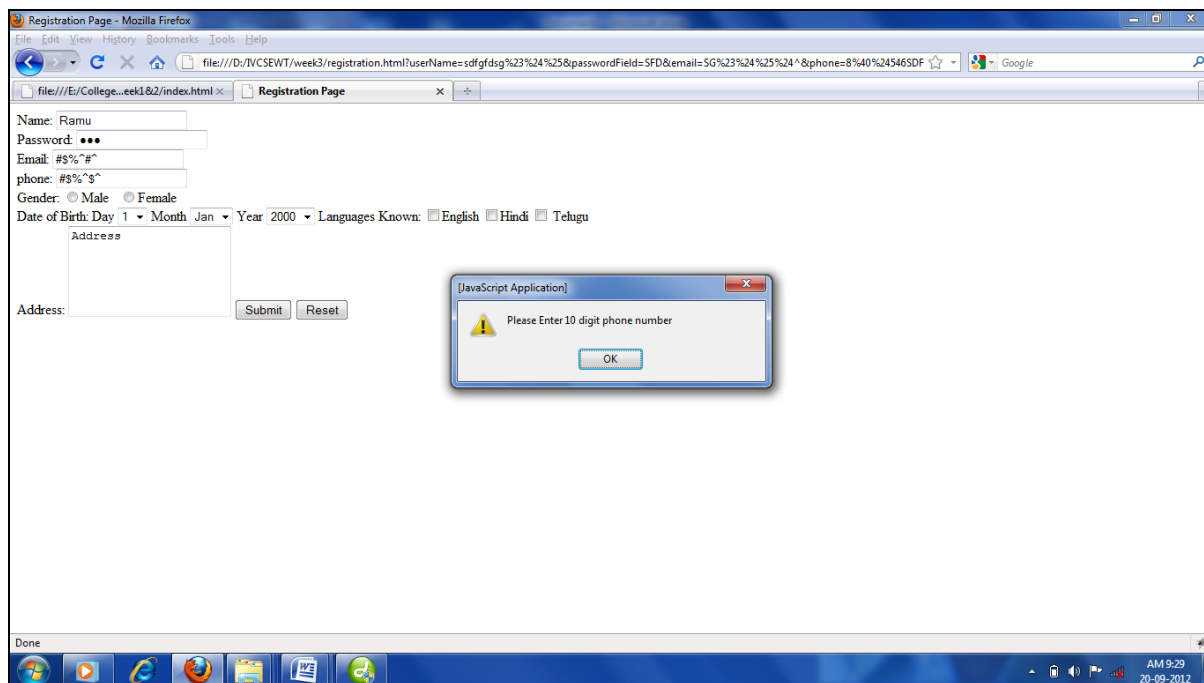
Registration Page – Password Validation



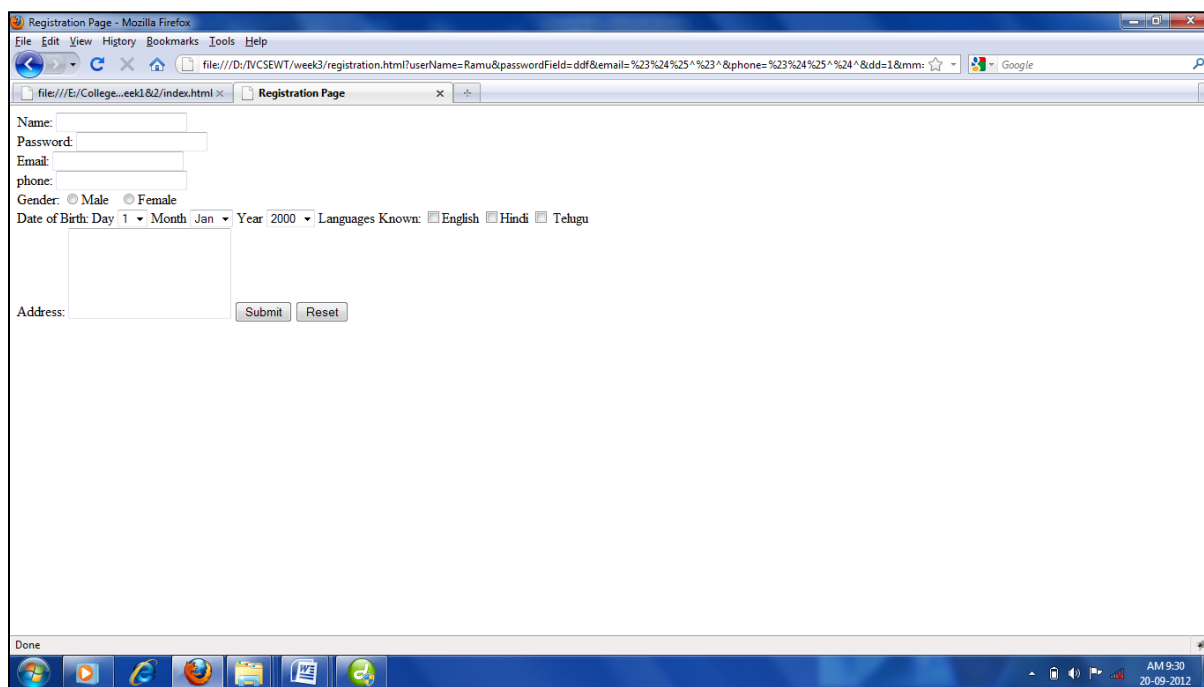
Registration Page – Email Validation



Registration Page – Phone Number Validation



Registration Page – Phone Number Validation



Registration Page – After Validation if any errors then automatically go to empty registration page

EXP NO: 5

AIM: Write an XML file which will display the Book information which includes the following: Title of the book, Author Name, ISBN number, Publisher name, Edition and Price. Validate the above document using DTD and XML Schema.

Description

Extensible Markup Language (XML) is a meta-markup language that provides a format for describing structured data. This facilitates more precise declarations of content and more meaningful search results across multiple platforms. In addition, XML is enabling a new generation of Web-based data viewing and manipulation applications.

In the HTML you use tags to tell the browser to display data as bold or italic; in XML you use tags only to describe data, such as city name, temperature, and barometric pressure. In XML, you use style sheets such as Extensible Stylesheet Language (XSL) and Cascading Style Sheets (CSS) to present the data in a browser. XML separates the data from the presentation and the process, enabling you to display and process the data as you wish by applying different style sheets and applications.

Source Code

a) Demonstrating DTD file

book.dtd

```
<!ELEMENT BOOKS (book)>
<!ELEMENT book (title,auther,ISBNnumber,publishername,editor,price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT auther (#PCDATA)>
<!ELEMENT ISBNnumber (#PCDATA)>
<!ELEMENT publishername (#PCDATA)>
<!ELEMENT editor (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

Books1.xml

```
<?xml version="1.0" ?>
<!DOCTYPE books SYSTEM "books.dtd">
<books>
<book>
<title>java server page</title>
<author>hans bergsten</author>
<isbn>81-265</isbn>
<publisher>O'relilly</publisher>
<edition>second</edition>
<price>200</price>
</book>
<book>
<title>c</title>
```

```
<author>balaguruswamy</author>
<isbn>81-695</isbn>
<publisher>tata</publisher>
<edition>second</edition>
<price>400</price>
</book>
<book>
<title>web program</title>
<author>chirs bates</author>
<isbn>81-265</isbn>
<publisher>wiley</publisher>
<edition>second</edition>
<price>300</price>
</book>
</books>
```

b) Demonstrating XSD

books.css

```
books
{
color:orange;
}
book
{
color:Red;
}
title
{
color:blue;
font-weight:bold;
margin-left:10pt;
display:block;
}
author
{
color:red;
font-weight:bold;
margin-left:10pt;
}
isbn
{
color:green;
font-weight:bold;
margin-left:10pt;
```

```

}
edition
{
color:red;
font-weight:bold;
margin-left:10pt;
}
publisher
{
color:green;
font-weight:bold;
margin-left:10pt;
}

```

Books.xsd

```

<?xml version="1.0" encoding="iso-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="books">
<xs:complexType>
<xs:sequence>
<xs:element name="book" minOccurs="1" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="book">
<xs:complexType>
<xs:sequence>
<xs:element ref="title" minOccurs="1" maxOccurs="1"/>
<xs:element ref="auther" minOccurs="1" maxOccurs="1"/>
<xs:element ref="isbn" minOccurs="1" maxOccurs="1"/>
<xs:element ref="publisher" minOccurs="1" maxOccurs="1"/>
<xs:element ref="edition" minOccurs="1" maxOccurs="1"/>
<xs:element ref="price" type="xs:string" minOccurs="1" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="title" type="xs.string"/>
<xs:element name="auther" type="xs.string"/>
<xs:element name="isbn" type="xs.string"/>
<xs:element name="publisher" type="xs.string"/>
<xs:element name="edition" type="xs.string"/>
<xs:element name="price" type="xs.string"/>
</xs:schema>

```

Books.xsl

```

<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="books.css"?>
<books xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="books.xsd">
<book>
    <title>web program</title>
    <author>chirs bates</author>
</book>
<book>
<title>java server page</title>
<author>hans bergsten</author>
<isbn>81-265</isbn>
<publisher>O'relilly</publisher>
<edition>second</edition>
<price>200</price>
</book>
<book>
<title>c</title>
<author>balaguruswamy</author>
<isbn>81-695</isbn>
<publisher>tata</publisher>
<edition>second</edition>
<price>400</price>
</book>
</books>

```

c) Demonstrating XSL

books.xsl

```

<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
    <head>
    <title> books</title></head>
    <body>
    <table border="1">
    <tr><th>title</th>

```

```

        <th> auther</th>
        <th> isbn</th>
        <th>publisher</th>
        <th>edition</th>
        <th>price</th>
    </tr>
<xsl:for-each select="/books/book">
<tr>
<td bgcolor="green"><xsl:value-of select="title"/></td>
<td bgcolor="red"><xsl:value-of select="author"/></td>
<td bgcolor="cyan"><xsl:value-of select="isbn"/></td>
<td bgcolor="yellow"><xsl:value-of select="publisher" /></td>
<td bgcolor="silver"><xsl:value-of select="edition" /></td>
<td bgcolor="blue"><xsl:value-of select="price" /></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Books3.xml

```

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="books.xsl"?>
<books>
<book>
    <title>java server page</title>
    <author>hans bergsten</author>
    <isbn>81-265</isbn>
    <publisher>O'relilly</publisher>
    <edition>second</edition>
    <price>200</price>
</book>

<book>
    <title>c</title>
    <author>balaguruswamy</author>
    <isbn>81-695</isbn>
    <publisher>tata</publisher>
    <edition>second</edition>
    <price>400</price>
</book>
<book>
    <title>web program</title>
    <author>chirs bates</author>

```

```

<isbn>81-265</isbn>
<publisher>wiley</publisher>
<edition>second</edition>
<price>300</price>
</book>
</books>

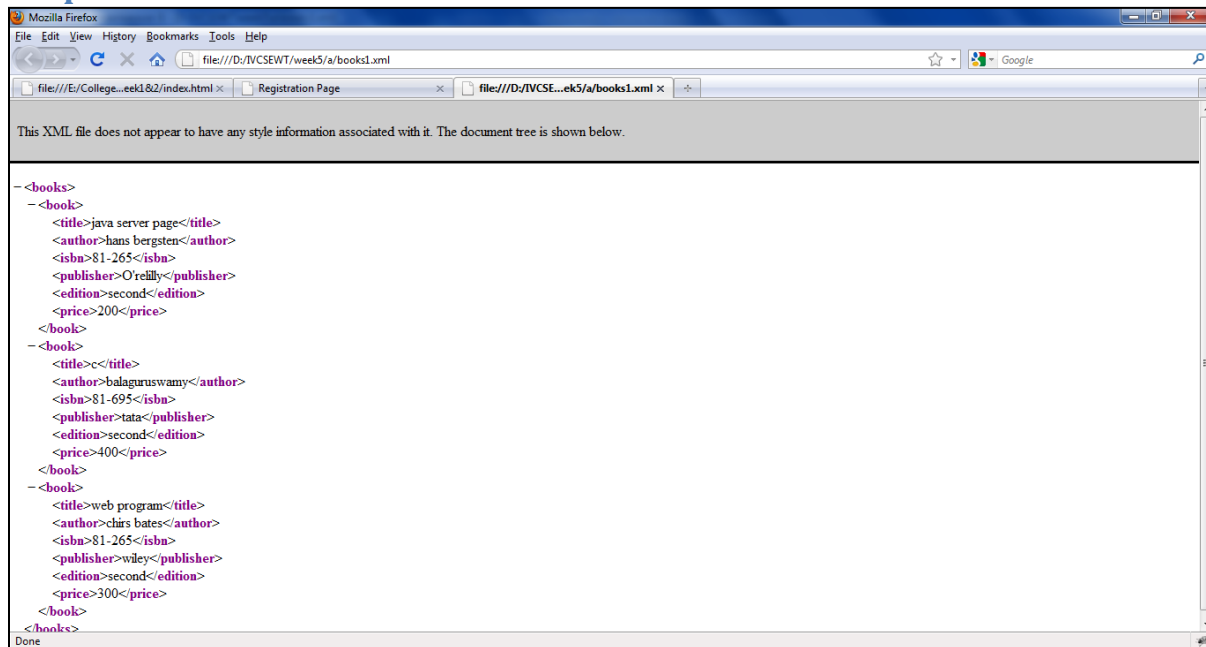
```

Execution Procedure

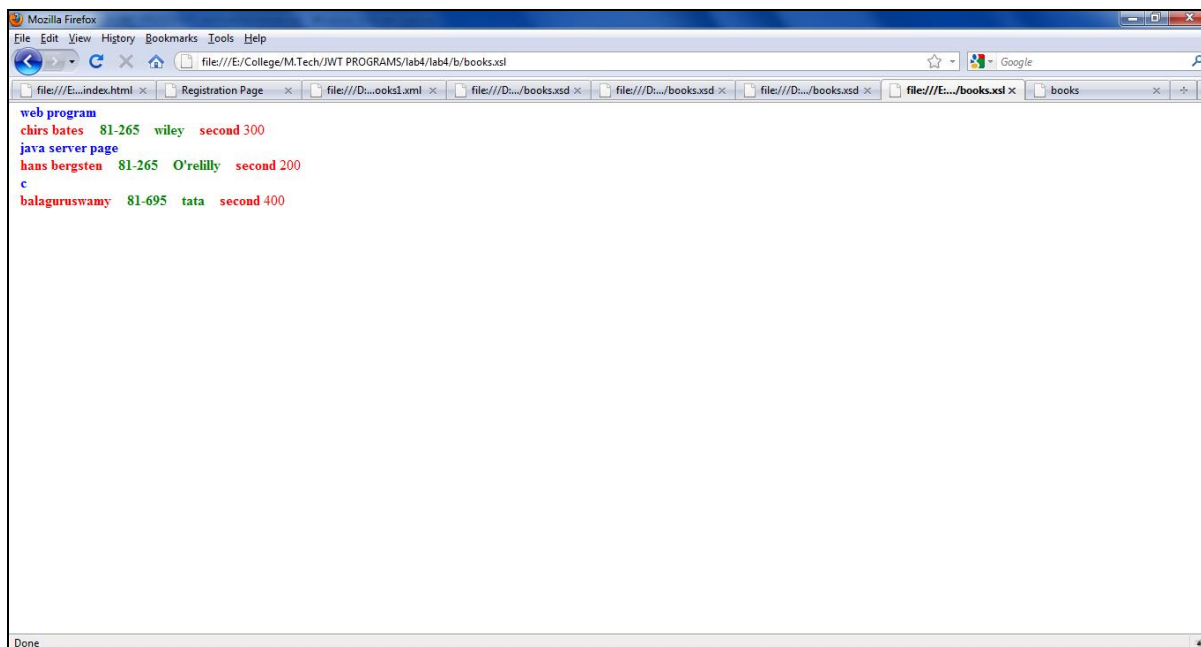
Step1: Type all html programs in notepad or any text editor and save the files.

Step2: open appropriate xml files in web browser for getting output.

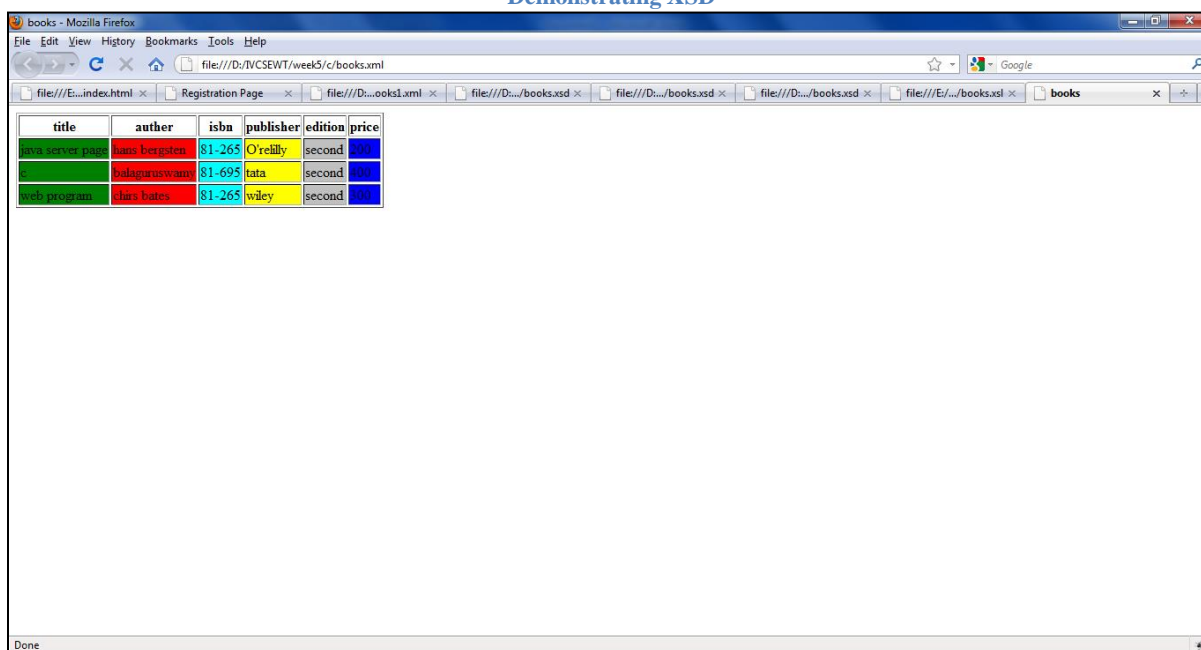
Output



Demonstrating DTD



Demonstrating XSD



Demonstrating XSL

EXP NO: 6**AIM:**

- i. Write a PHP program to validate the fields of the login page.
- ii. Write a PHP program to validate the fields of the Registration page

DESCRIPTION:

This PHP login application uses MySQL database to store user information and all the input parameters are validated with javascript. This PHP login page using MySQL database connections contains PHP 7.0 methods only. Many methods like `mysql_real_escape_string()`, `mysql_query()`, `mysql_num_rows()`, `mysql_connect()`, `mysql_select_db()`, `mysql_close()` were deprecated in PHP 5 and removed from the latest version of PHP i.e. PHP 7.0.

ALGORITHM:**LoginForm.php**

- Step 1:-Open the html tag and the title as LoginForm.php.
- Step 2:-Declare the `validate()` function.
- Step 3:-Design login form with username, password, submit and reset values.
- Step 4:-if anyone not enter user name or password it shows alert message.
- Step 5:-close the html tag.

Login.php

- Step 1:-Open the html and body tag.
- Step 2:-Open the MySQL database connection.
- Step 3:-if the `$_SERVER['REQUEST_METHOD'] == 'POST'` then
- Step 4:-enter user name and password
- Step 5:-check the user name and password is valid or not
- Step 6:-if valid then display the login page
- Step 7:-otherwise it shows incorrect username and password.
- Step 8:-close body and html tags.

PROGRAM:**Loginform.php**

```
<html>
<head>
  <title>LoginForm.php</title>
<!-- Javascript validation foruser inputs -->
<script type="text/javascript">
functionvalidate()
{
  vara = document.login.username.value;
  if(a=="")
  {
    Alert("Please enter your username");
    document.login.username.focus();
    returnfalse;
  }
}
```

```

    varb = document.forms["login"]["password"].value;
    if(b == "")
    {
    alert("Please enter your password");
    returnfalse;
    }
    }
</script>
</head>
<body>

<!-- Make a note that the method type used is post, action page is Login.php andvalidate()
functionwill get called on submit -->
<center><h1>PHP login application using MySQL</h1></center>
<br>
<form name="login"method="post"action="Login.php"onsubmit="return validate();">

<table border="3"width="355"align="center"bgcolor="#affeff">
<tr>
<td height="50"bgcolor="#aa0eff">
<center><b>Login</b></center>
</td>
</tr>
<tr>
<td width="219">
<table width="240"align="center">
<tr></tr>
<tr>
<td width="71"><span style="font-size:12pt;">Username:</span></td>
<td width="139"><input
type="text"name="username"minlength="4"required="required"placeholder="Your
username"></td>
</tr>
<tr>
<td width="71"><span style="font-siz:12pt;">Password:</span></td>
<td width="139"><input
type="password"name="password"minlength="4"required="required"placeholder="Your
Password"></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td width="71">&nbsp;</td>
<td width="139">
<input type="submit"name="submit"value="Submit">

```

```

<input type="reset" name="reset" value="Reset">
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td width="219" bgcolor="#aa0eff" align="center">
</td>
</tr>
</table>
</form>
</body>
</html>

```

Login.php

```

<html>
<body>
<?php

include("DBConnection.php");
session_start(); //It is advised to open a session in the beginning
if($_SERVER['REQUEST_METHOD'] == 'POST')
{
    if(empty($_POST['username']) || empty($_POST['password'])) //This is the way to
    validate inputs using PHP code but here we are using javascript validations so it is not
    necessary
    {
        echo "Please enter the correct Username and Password";
        //header("location: LoginForm.php");//You will be sent to Login.php for re-login
    }
    $usernameLogin= $_POST["username"]; // as the method type in the form is "post" we
    are using $_POST otherwise it would be $_GET[]
    $passwordLogin= $_POST["password"];
    $myusername= stripslashes($usernameLogin); // stripslashes() is used to clean up data
    retrieved from an HTML form
    $mypassword= stripslashes($passwordLogin);
    $myusername= mysqli_real_escape_string($db,$_POST['username']); // Escapes special
    characters in a string for use in an SQL statement
    $mypassword= mysqli_real_escape_string($db,$_POST['password']);
    $query= "SELECT * FROM profile WHERE username = '$myusername' and password =
    '$mypassword'"; //Fetching all the records with input credentials
    $result= mysqli_query($db,$query);
    $count= mysqli_num_rows($result);
    if($count== 1)
    {

```

```
$_SESSION['username']=$myusername; //Storing the username value in session
variable so that it can be retrieved on other pages
```

```
//Optional code - if you want to maintain records of logged in users, use following
two lines
```

```
$qry= "UPDATE profile SET loggedin=1 WHERE username='$username'"; // to
maintain Logged-in Status
```

```
$res= mysqli_query($db,$qry);
```

```
mysqli_close($connection); //To close MySQL database connection
```

```
header("location: UserProfile.php"); // user will be taken to profile page
```

```
    }
    else
    {
        echo "<br>Incorrect username or password";
        ?>
        <br>
        <a href="LoginForm.php">Try Login</a>
        <?php
        }
    }
}
```

```
?>
</body>
</html>
```

dbconnections.php

```
<?php
//Establishing connection with the database
define('DB_SERVER', 'localhost:3306');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', 'root123');
define('DB_DATABASE', 'customers'); //where customers is the database
$db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
?>
```

Userprofile.php

```
<html>
<title>userProfile.php</title>
<body>
<?php
session_start();
$username= $_SESSION['username']; //retrieve the session variable
?>
<center><h1>User Profile </h1></center>
<br/>
<b>Welcome <?php echo$username?></b>
```

```

<div style="text-align: right"><a href="Logout.php">Logout</a></div><!-- calling Logout.php
to destroy the session -->
<?php
if(!isset($_SESSION['username'])) //If user is not logged in then he cannot access the profile
page
{
//echo 'You are not logged in. <a href="login.php">Click here</a> to log in.';
header("location:LoginForm.php");
}
?>
</body>
</html>

```

Logout. php

```

<?php
include("DBConnection.php");
session_start();
$username= $_SESSION['username']; //retrieve the session variable

//Optional code - to set the logged in user's status to zero
$query= "UPDATE profile SET loggedin=0 WHERE username='$username'"; // to maintain
Logged-in Status
$result= mysqli_query($db,$query);
mysqli_close($connection);
unset($_SESSION['username']); //to remove session variable
session_destroy(); //destroy the session
header("location: LoginForm.php"); //to redirect back to "Login.php" after logging out
exit();
if(!isset($_SESSION['username'])) //If user is not logged in then he cannot access the profile
page
{
//echo 'You are not logged in. <a href="login.php">Click here</a> to log in.';
header("location:LoginForm.php");
}
?>

```

Output:



PHP login application using MySQL

A screenshot of a login form. The form has a purple header with the word 'Login'. Below the header, there are two input fields: 'Username: Your username' and 'Password: Your Password'. At the bottom of the form, there are two buttons: 'Submit' and 'Reset'.

VIVA QUESTIONS:

EXP NO: 7

AIM: Write a JSP to connect to the database and extract data from the tables and display them to the user.

DESCRIPTION**Source Code:****Retrieve.Html**

```
<html>
<head><title>loginpage</title></head>
<body>
<center><br><br><br><h1><font size="22" color="pink"><i>Login Page</font></h1></i>
<form action="retrieve.jsp" name="log" id="li" onSubmit="return validate( )" action="#"
method="post">
<table><tr><td>Login Id:</td><td><input type="text" name="login" size="10"><span
id="s1"></span></td></tr>
<tr><td>Password:</td><td><input type="password" name="pwd" size="10"><span
id="s2"></span></td></tr>
<tr><td><input type="submit" value="Login"></td><td><input type="reset"
value="reset"></td></tr></table>
</form>
</body>
</html>
```

retrieve.jsp

```
<% @page import="java.sql.*"%>
<html>
<body>
<%
String uname=request.getParameter("login");
String pwd=request.getParameter("pwd");
try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/srimanth","root","");
    Statement st=con.createStatement();
    PreparedStatement p=con.prepareStatement("select E_mail,Password from student");
    p.setString(1,uname);
    p.setString(2,pwd);
    if(uname==E_mail && pwd==Password) {
        response.sendRedirect("welcome.jsp");
    }
}
```

OUTPUT:

VIVA QUESTION

EXP NO: 8

AIM: Design a JSP to insert the details of the users who register through the registration page and store the details in to the database.

PROGRAM:**Registration.html**

```

<html>
<head><title>register</title></head>
<body>
<center><br><br><br><h1><font size="22" color="pink"><i>Registration</font></h1></i>
<form action="registration.jsp" method="post" name="reg" id="rn" onSubmit="return validate(
)">
<table><tr><td>Name:<td><input type="text" name="na" size="10"></td></tr>
<tr><td>Gender:</td><td><input type="Radio" name="gender" >Male<input type="Radio"
name="gender" >Female</td></tr>
<tr><td>Date of birth:</td><td><input type="date" name="dob" size="10"></td></tr>
<tr><td>Rollno:</td><td><input type="text" name="roll" size="10"></td></tr>
<tr><td>Phone no:</td><td><input type="text" name="pno" size="10"></td></tr>
<tr><td>Email id:</td><td><input type="text" name="mail" size="30"></td></tr>
<tr><td>Password:</td><td><input type="password" name="pwd" size="10"></td></tr>
<tr><td>Re-enter Password:</td><td><input type="password" name="rpwd"
size="10"></td></tr>
<tr><td>Branch:</td><td><select name="br"><option value="default">Select
branch</option><option value="IT">Information Technology</option><option
value="CSE">Computer Science Engineering</option><option value="ECE">Electronics and
Communication Engineering</option><option value="EEE">Electrical and Electronics
Engineering</option><option value="CIVIL">Civil Engineering</option><option
value="MECH">Mechanical Engineering</option></td></tr>
<tr><td>College:</td><td><input type="text" name="col" size="10"></td></tr>
<tr><td>Address:</td><td><textarea name="add" rows="4" cols="30"></textarea></td></tr>
<tr><td><input type="submit" value="Register"></td><td><input type="reset"
value="reset"></td></tr></table>
</form>
</body>
</html>

```

registration.jsp

```

<% @page import="java.sql.*"%>
<html>
<body>
<%
String Name=request.getParameter("na");

```

```

String Gender=request.getParameter("gender");
String Dob=request.getParameter("dob");
String Rollno=request.getParameter("roll");
String Phoneno=request.getParameter("pno");
String E_mail=request.getParameter("mail");
String Password=request.getParameter("pwd");
String Branch=request.getParameter("br");
String College=request.getParameter("col");
String Address=request.getParameter("add");
try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/srimanth","root","");
    Statement st=con.createStatement();
    PreparedStatement ps=con.prepareStatement("insert into
student(Name,Gender,Dob,Rollno,Phoneno,E_mail>Password,Branch,College,Address)values(?,
?,?,?, ?, ?, ?, ?, ?)");
    ps.setString(1,Name);
    ps.setString(2,Gender);
    ps.setString(3,Dob);
    ps.setString(4,Rollno);
    ps.setString(5,Phoneno);
    ps.setString(6,E_mail);
    ps.setString(7>Password);
    ps.setString(8,Branch);
    ps.setString(9,College);
    ps.setString(10,Address);
    int x=ps.executeUpdate();
%><a href="welcome.jsp">Register</a><%
}
catch(Exception e) {
out.print(e.getMessage());
}
%>
</body>
</html>

```

welcome.jsp

```

<html>
<body bgcolor="green"><center>
<font size="22" color="red">
<%
out.print("Welcome to the Java Server Page Programming");
%>
</font>
</center>

```

```
</body>  
</html>
```

OUTPUT:

VIVA QUESTIONS:

EXP NO: 9

AIM:

Write a PHP program to connect to MySQL database which retrieves the data from the tables and display them to the user.

Description

Create a Connection to a MySQL Database

Before you can access data in a database, you must create a connection to the database.

In PHP, this is done with the `mysqli_connect()` function.

Syntax

```
mysqli_connect (servername, username, password, dbname) ;
```

Parameter	Description
Servername	Optional. Specifies the server to connect to. Default value is "localhost:3306"
Username	Optional. Specifies the username to log in with. Default value is the name of the user that owns the server process
Password	Optional. Specifies the password to log in with. Default is ""
Dbname	Database Name

Example

In the following example we store the connection in a variable (`$con`) for later use in the script. The "die" part will be executed if the connection fails:

```
<?php
$con = mysqli_connect ("localhost", "root", "", 'srikanth');
if (!$con)
{
    die('Could not connect: ' . mysqli_error());
}

// some code
?>
```

Closing a Connection

The connection will be closed automatically when the script ends. To close the connection before, use the `mysqli_close()` function:

```
<?php
$con = mysqli_connect ("localhost", "root", "", "srikanth");
if (!$con)
{
    die('Could not connect: ' . mysqli_error());
}

// some code

mysqli_close ($con);
?>
```

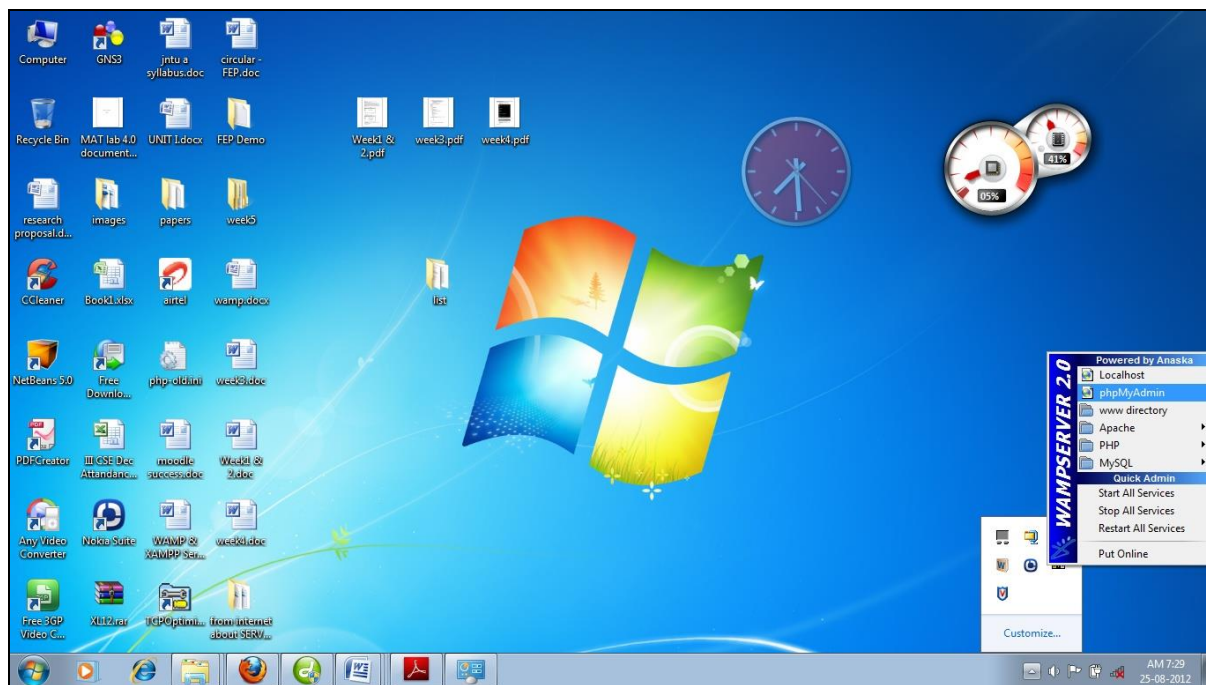
Source code

Step by step procedure for creating table in php my admin

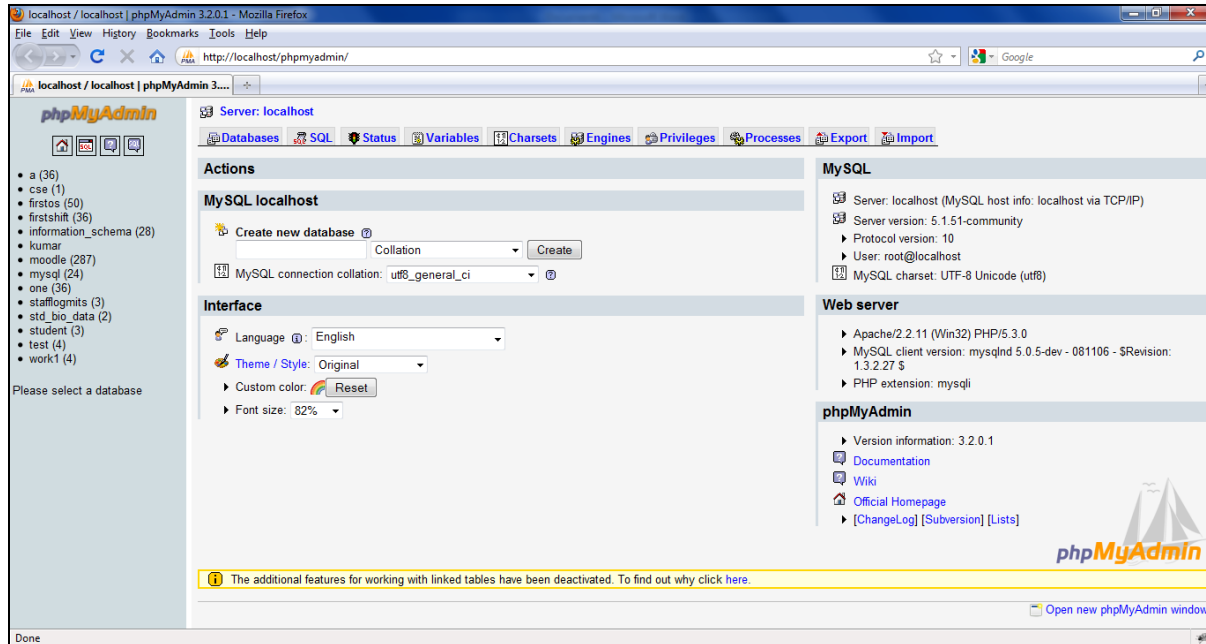
MySQL is automatically installed along with WAMP server installation.
User environment for MySQL in WAMP is “phpMyAdmin”.

Step1: Start WAMP server

Step2: Click WAMP server icon on taskbar and select “phpMyAdmin” option.

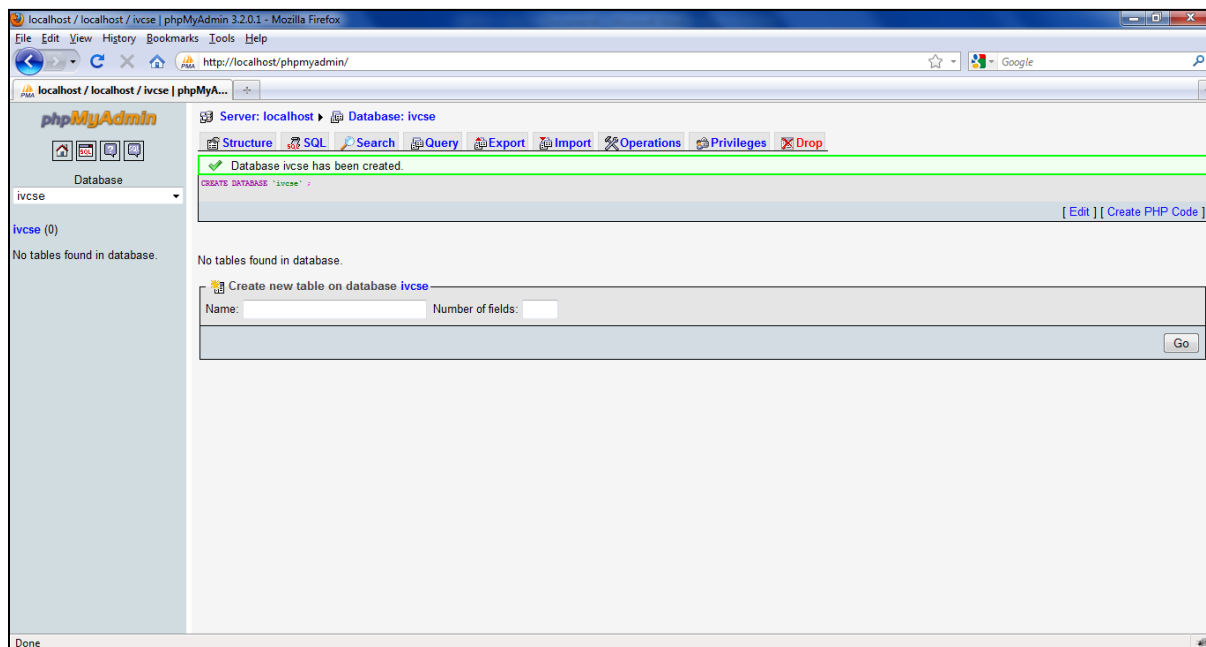


It automatically opens “phpMyAdmin” screen.

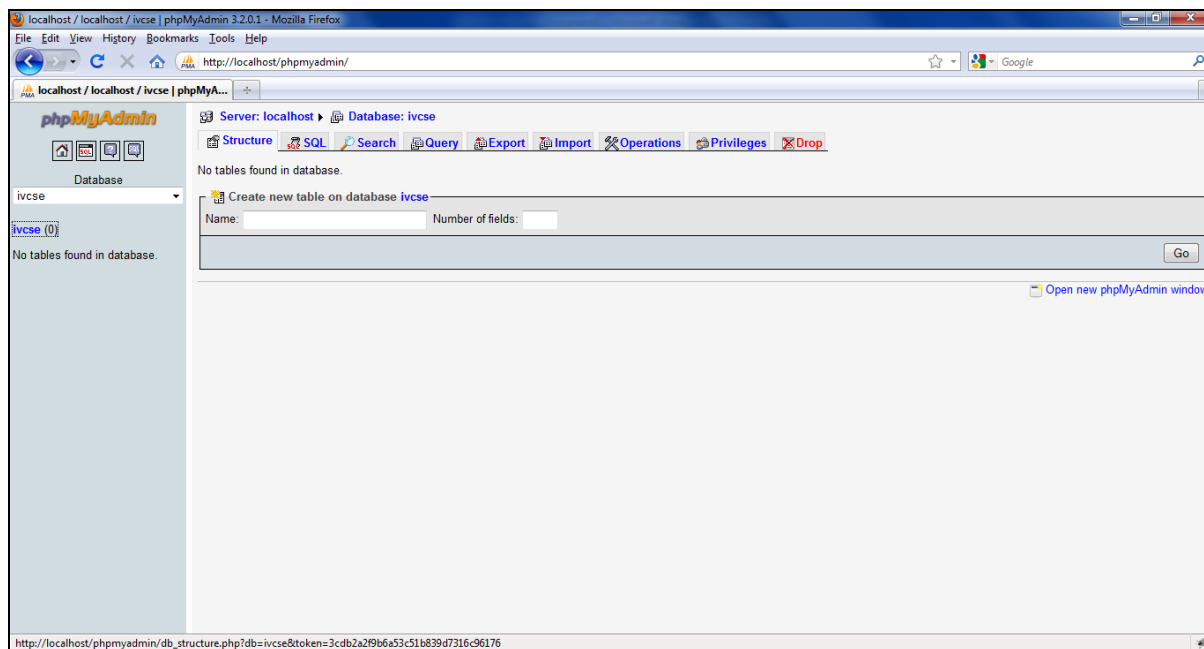


Step3: create New Database.

Provide database name on “create new database” field and click “create” button. The following screen displays in your computer



Step 4: Click database “ivcse” i.e appeared left panel of phpMyAdmin Screen. It displays following screen

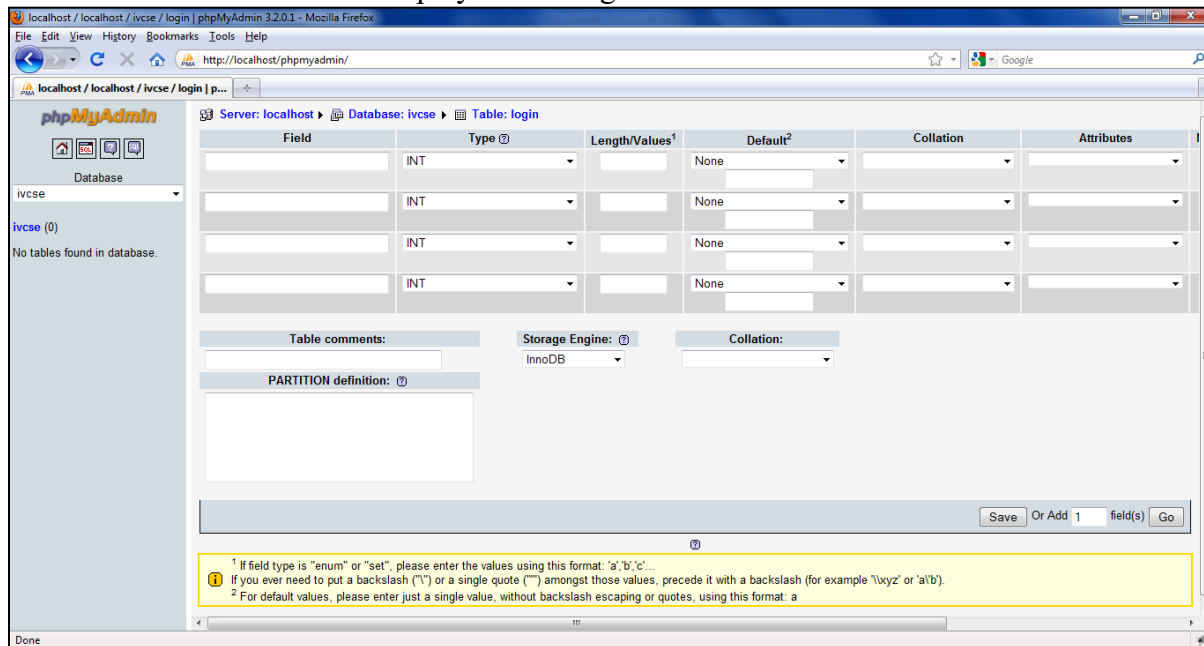


Step5: Creating a table.

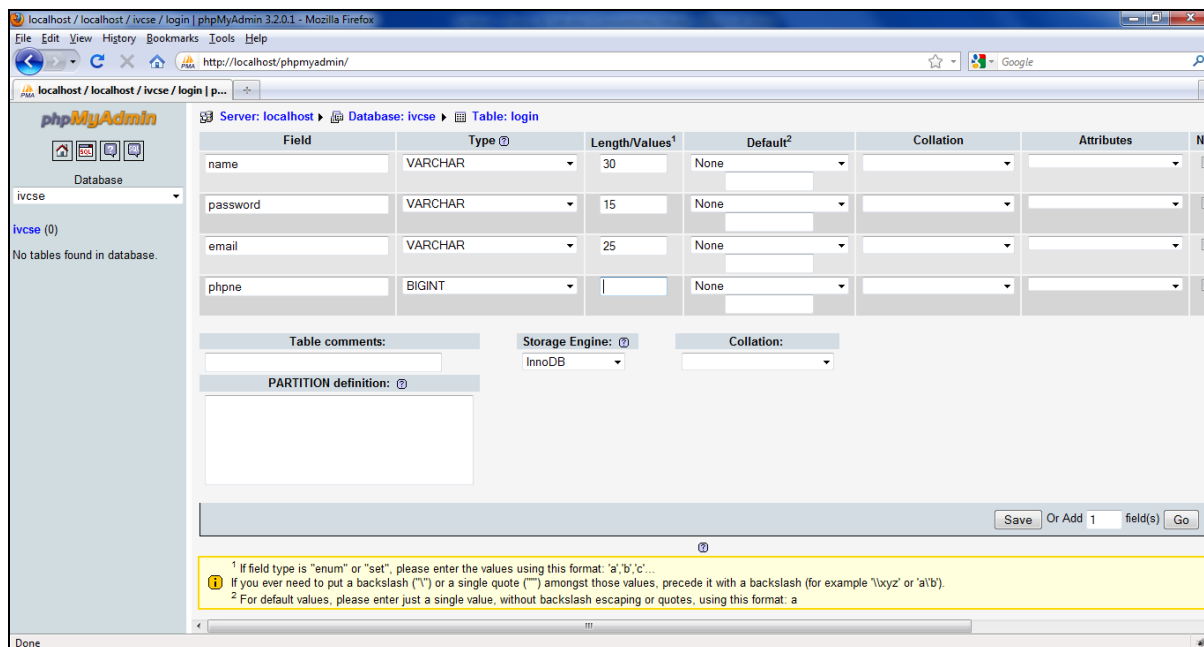
The Aim indicates that you can create which should contain at least the following fields: name, password, email-id, phone number. So here 4 fields are there.

Provide table name “login” and “number of fields:” 4

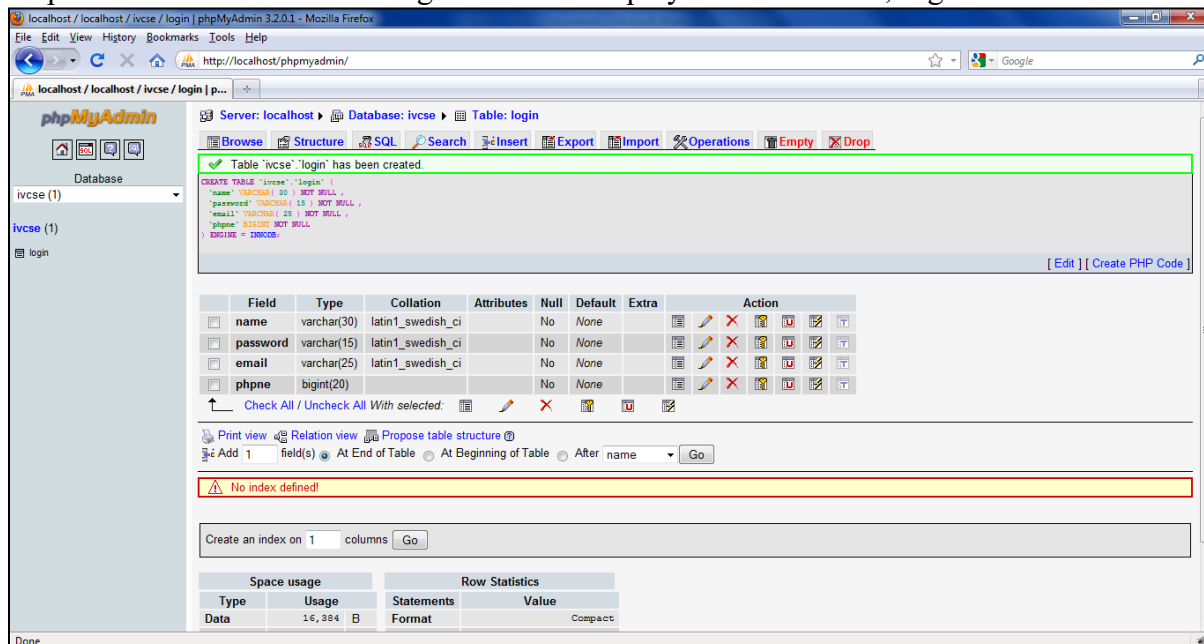
Click “Go” button. It displays following screen



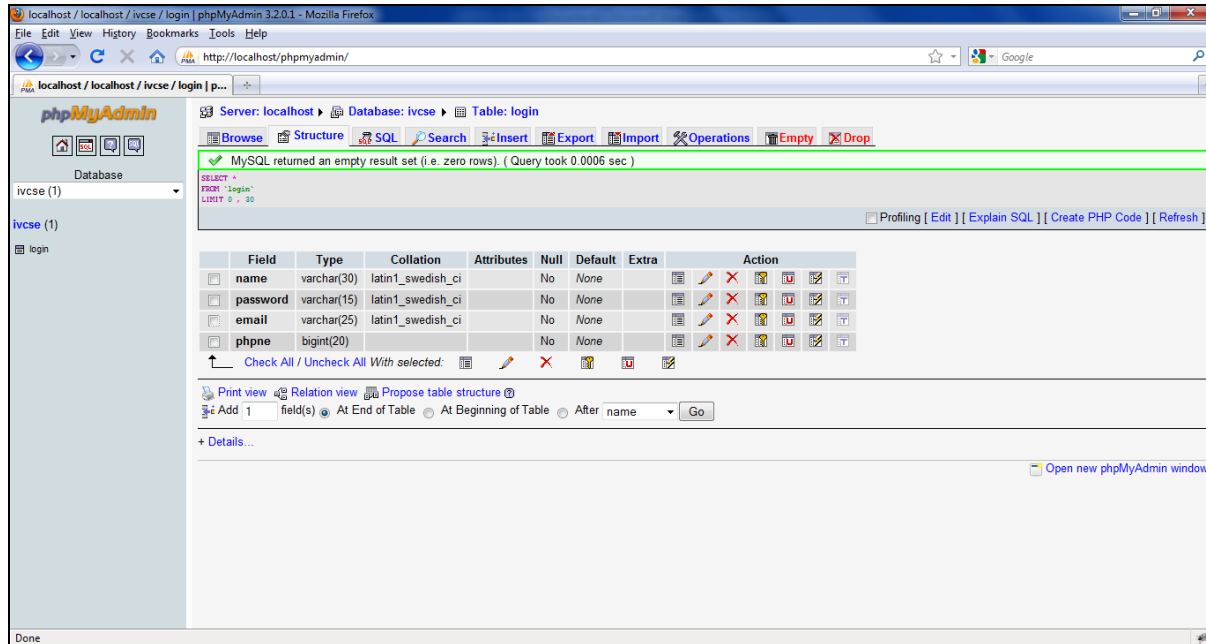
Step 6: Provide fields name and its datatype, size like the following screen



Step 7: click “Save”. It create login table and displays “Table ‘ivcse’,’login’ has been created.”



Step 8: click on table name on left panel of phpMyAdmin, you can see structure of table i.e field name datatype and other parameters



Display.php

```
<?php
```

```
$con = mysqli_connect("localhost","root","","srikanth");
```

```
if (!$con)
```

```
{
    die('Could not connect: ' . mysqli_error());
}
```

```
mysql_select_db("ivcse", $con);
```

```
$result = mysqli_query($con, "SELECT * FROM login");
```

```
echo "<table border='1'>
```

```
<tr>
```

```
<th>Name</th>
```

```
<th>Password</th>
```

```
<th>Email</th>
```

```
<th>Phone Number</th>
```

```
</tr>";
```

```
while($row = mysqli_fetch_array($result))
```

```
{
    echo "<tr>";
    echo "<td>" . $row['name'] . "</td>";
    echo "<td> Not shown</td>";
    echo "<td>" . $row['email'] . "</td>";
    echo "<td>" . $row['phno'] . "</td>";
    echo "</tr>";
}
```

```
}  
echo "</table>";  
  
mysqli_close($con);  
?>
```

Execution procedure

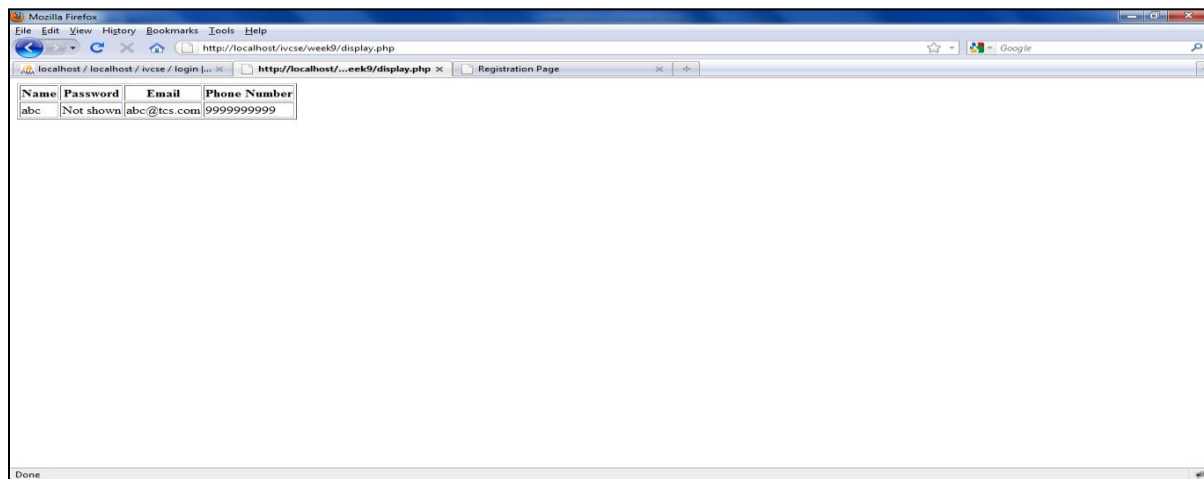
Step1: Type all programs and save with .php extension in c:\WAMP\WWW directory.

Step2: Run WAMP Server.

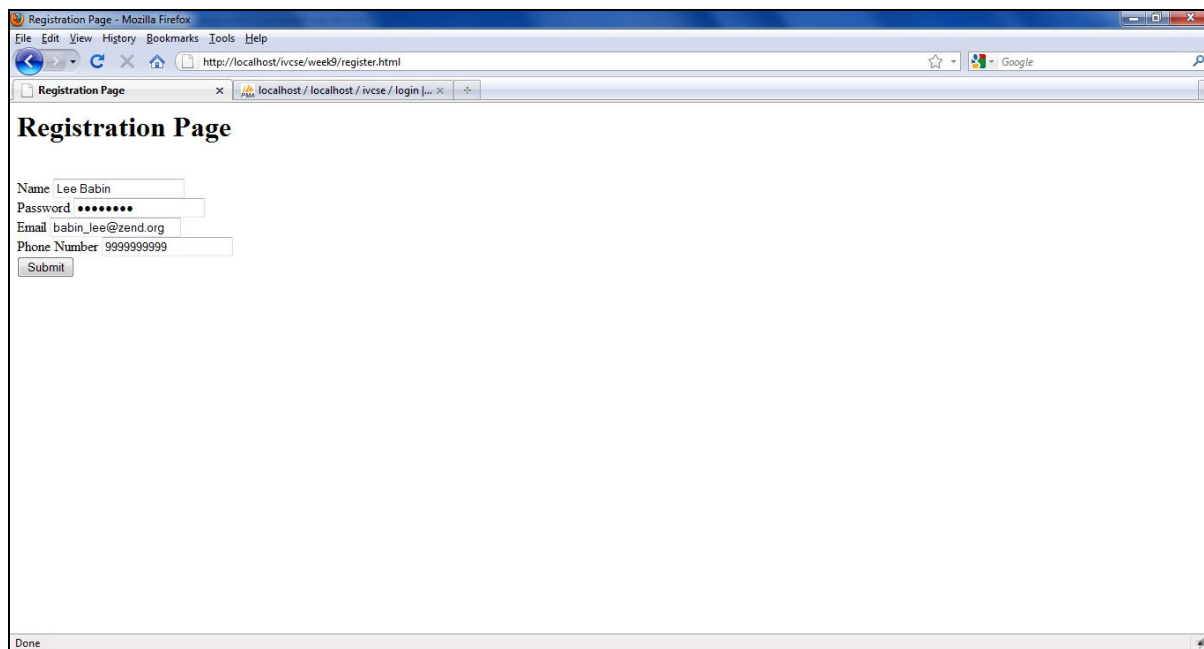
Step3: Through localhost execute your programs.

url's are: <http://localhost/display.php>&&<http://localhost/insert.php>

Output



Displaying data from Database



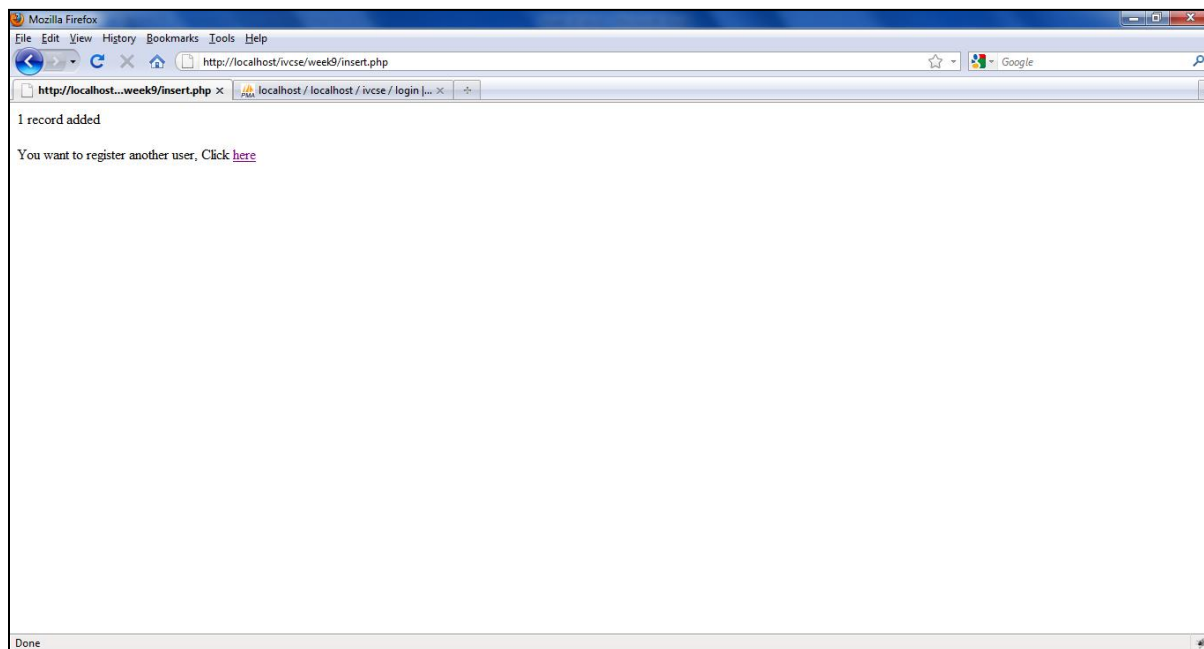
The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://localhost/ivcse/week9/register.html`. The page title is "Registration Page". The form contains the following fields and values:

- Name: Lee Babin
- Password: ••••••••
- Email: babin_lee@zend.org
- Phone Number: 999999999

A "Submit" button is located below the form fields. The status bar at the bottom of the browser window shows "Done".

Registering New User

After providing all fields of data and Click on Submit button



After successfully added 1 user

EXP NO: 10**AIM:**

Write a PHP program to insert the details entered by the user in the Registration form into MySQL database.

DESCRIPTION:**PROGRAM-1:****Register.php**

```
<html>
<title>Registration Page</title>
</head>
<body>
<h1> Registration Page</h1>
<br />
<form action="insert.php" method="post">
Name <input type="text" name="fname" id="fname" value="name" /><br />
Password <input type="password" name="pwd" id="pwd" value="password" /><br />
Email <input type="text" name="email" id="email" value="email@example.com"/><br />
Phone Number <input type="text" name="ph" id="ph" value="9999999999"
maxlength="10"/><br />
<input type="submit" value="Submit" />
</form>
</body>
</html>
```

Insert .php

```
<?php
$con = mysqli_connect("localhost","root","root","srikanth");
if (!$con)
{
    die('Could not connect: ' . mysqli_error());
}

mysql_select_db("ivcse", $con);

$sql="INSERT INTO login (name, password, email, phno)
VALUES
('$_POST[fname]','$_POST[pwd]','$_POST[email]','$_POST[ph]')";

if (!mysqli_query($con,$sql))
{
    die('Error: ' . mysqli_error());
}
echo "1 record added";
```

```
echo "<br><br>";  
echo "You want to register another user, Click <a href='register.html'>here</a>";
```

```
mysqli_close($con);  
?>
```

Output: