

Contents:

| S.No | Description | Page No |
|------|---|----------------|
| | <i>Vision, Mission</i> | <i>iv</i> |
| | <i>PEOs and POs</i> | <i>v-vi</i> |
| | <i>Preface</i> | <i>vii</i> |
| | <i>Lab objectives, Outcomes & Guidelines to the students</i> | <i>viii-ix</i> |
| | <i>University Syllabus</i> | <i>x-xii</i> |
| | <i>List of Experiments to be conducted for this semester</i> | <i>xiii-xv</i> |
| 1. | Design the following static web pages required for an online book store web site. 1. Home Page 2. Login Page 3. Catalogue Page | 22-30 |
| 2. | Design the following static web pages required for an online book store web site. 4. Cart Page 5. Registration Page | 31-37 |
| 3. | Design a web page using CSS | 38-43 |
| 4. | Write an XML file which will display the Book information, DTD, Schema, XSLT. | 44-47 |
| 5. | Write Ruby program reads a number and calculates the factorial value of it and prints the Same, counts number of lines in a text files using its regular Expressions facility, iterator to find out the length of a string, arrays in Ruby. | 48-53 |
| 6. | Write Ruby programs for associative arrays, uses Math module to find area of a triangle, tk module to display a window, operations on complex objects. | 54-60 |
| 7. | Write a program which illustrates the use of associative arrays in perl, takes set names along the command line and prints whether they are regular files or special files, implement UNIX `passed' program, connect to a MySQL database table and executing simple commands. | 61-67 |
| 8. | Example PHP program for contact us page, User Authentication | 68-78 |
| 9. | Example PHP program for registering users of website and login, Install a database (Mysql or Oracle). | 79-90 |
| 10. | Insert the details of the 3 or 4 users who register with the web site by using registration form and authenticate them. Create tables in the database which contain the details of items (books in our case like Book name , Price, Quantity, Amount) of each category. Modify your catalogue page, extract data and also display it. | 91-100 |
| 11. | Session Management | 101-108 |
| 12. | Mini Project | |

Vision & Mission

Vision of the Institute

Confect as a premier institute for professional education by creating technocrats who can address the society's needs through inventions and innovations.

Mission of the Institute

- Partake in the national growth of technological, industrial arena with societal responsibilities.
- Provide an environment that promotes productive research.
- Meet stakeholders expectations through continued and sustained quality improvements.

Vision of the Program

To create Technical & Managerial expertise, ethically strong global manpower by providing top class Information Technology education to deliver the needs of the society to excel in real life situation at all levels.

Mission of the Program

To provide best supporting learning environment and development of knowledge well trained, confident, industry ready professionals and an inquisitive mind, who are ready to contribute to the industry, economy & the society

PEOs & POs

Program Educational Objectives

This education is meant to prepare our students to thrive and to lead. In their careers,
our graduates

| | |
|----|--|
| P1 | To develop the ability among the students to gain knowledge about core and Application domain subjects in Information Technology |
| P2 | To prepare the students for successful careers in industry and make them compatible Information Technology professionals to meet the global needs. |
| P3 | To promote students' awareness for life-long learning and to introduce them to professional ethics. |

Program Outcomes

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, 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 modelling 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

1. **Mobile & Web Application Development:** Ability to develop mobile & web applications using J2EE, Android and J2ME.
2. **Cloud Services:** To deploy virtualized and cloud based services in the organization

Preface:

The Web Technologies Laboratory manual for III year II Semester is strictly written as per R-16 regulation of Jawaharlal Nehru Technological University, Kakinada.

For thorough understanding, viva questions are included at the end of each experiment. We hope that this manual will be useful for the students of Information Technology program of various universities for gaining deep knowledge in the Web Technologies Lab. It also would be useful for faculty as a ready reference.

The authors are thankful to the Principal and Head of the IT Department of Sasi Institute of Technology & Engineering, Tadepalligudem for their continuous encouragement in completing this manual.

Any suggestions for further improvement of the manual will be acknowledged and appreciated.

Authors

OBJECTIVES :

This course will enable the students

1. To create web pages using HTML and Cascading Styles sheets.[BT6 - Creating]
2. To make use of XML and their concepts to store, navigate, transport, manipulate, parse the content.[BT3 - Applying]
3. To design dynamic web pages for writing simple client-side scripts using AJAX.[BT6 - Creating]
4. To build small to large scale real time web applications using PHP.[BT6 - Creating]
5. To develop programs using PERL.[BT3 - Applying]
6. To develop programs using RUBY.[BT3 - Applying]

OUTCOMES :

After the completion of this course student will be able

1. Create web pages using HTML and Cascading Styles sheets.[BT6 - Creating]
2. Make use of XML and their concepts to store, navigate, transport, manipulate, parse the content.[BT3 - Applying]
3. Design dynamic web pages for writing simple client-side scripts using AJAX.[BT6 - Creating]
4. Build small to large scale real time web applications using PHP.[BT6 - Creating]
5. Develop programs using PERL.[BT3 - Applying]
6. Develop programs using RUBY.[BT3 - Applying].

RECOMMENDED SYSTEMS / SOFTWARE REQUIREMENTS :

- Intel based desktop PC with P-IV or above, minimum of 166 MHZ or faster processor with at least 512 MB RAM and 10 GB free disk space.
- Any platform with Browser, XAMPP, Ruby support 60 systems.

Guidelines to the Students

- Be regular to the Lab. After entering into the Lab, Sign in the Log-register.
- Students should come with observation and record.
- Lab attendance will play a part of the internal assessment marks
- The students should come to the lab with a prior preparation to complete the programs within time.
- Students should get their observation books corrected before leaving the lab and should submit the records in next lab session.
- Follow the dress-code while coming to the Lab. Strict discipline should be maintained inside the laboratory.
- At the mid and end of the semester, lab exams will be conducted.
- Shut down the system properly, turn-off the monitor, Arrange the chair properly and then leave. Handle the systems carefully.

➤ JAHAWARLAL NEHRU TECHNOLOGICAL UNIVERSITY
KAKINADA
II Year B.Tech.-I Sem.

WEB TECHNOLOGIES LAB

List of experiments as per the university

PROGRAMS LIST:

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

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 “**MCA**” the catalogue for MCABooks 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.

| Web Site Name | | | | | |
|-------------------|-----------------------------|-------|--------------|-----------|------|
| Logo | Home | Login | Registration | Catalogue | Cart |
| mca mba BCA | Description of the Web Site | | | | |

2. LOGIN PAGE

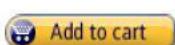
| Web Site Name | |
|-------------------|---|
| Logo | Home |
| MCA MBA BCA | <input type="button" value="Login"/> Login : <input type="text" value="11a51f0003"/> Password: <input type="password" value="*****"/> |
| | <input type="button" value="Submit"/> <input type="button" value="Reset"/> |

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 |
| MCA | | | | | |
| MBA |  | | | Book : XML Bible Author : Winston Publication : Wiely | \$ 40.5 |
| BCA |    | | | Book : AI Author : S.Russel Publication : Princeton hall | \$ 63 |
| | | | | Book : Java 2 Author : Watson Publication : BPB publications | \$ 35.5 |
| | | | | Book : HTML in 24 hours Author : Sam Peter Publication : Sam | \$ 50 |
| | | | | |  |
| | | | | |  |
| | | | | |  |
| | | | | |  |

4. 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 | Book name | Price | Quantity | Amount | |
| ECE | Java 2 | \$35.5 | 2 | \$70 | |
| EEE | XML bible | \$40.5 | 1 | \$40.5 | |
| CIVIL | | | Total amount - | \$130.5 | |

5. REGISTRATION PAGE

Create a “registration form” with 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)

5. Design a web page using CSS (Cascading Style Sheets) which includes the following:

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>
```

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; zindex: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; zindex: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>
```

6. Write an XML file which will display the Book information which includes the following:

1. Title of the book
2. Author Name
3. ISBN number
4. Publisher name
5. Edition
6. Price

Write a Document Type Definition (DTD) and XML schemas to validate the above XML file. It should be valid syntactically. Hint: You can use some xml editors like XML-spy Display the XML file as follows : The contents should be displayed in a table. The header of the table should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold. Use your own colors for remaining columns. Use XSL and CSS for the above purpose. Note: Give at least for 4 books.

7. Write Ruby program reads a number and calculates the factorial value of it and prints the Same.
8. Write a Ruby program which counts number of lines in a text files using its regular Expressions facility.
9. Write a Ruby program that uses iterator to find out the length of a string.
10. Write simple Ruby programs that uses arrays in Ruby.
11. Write programs which uses associative arrays concept of Ruby.
12. Write Ruby program which uses Math module to find area of a triangle.

- 13.** Write Ruby program which uses tk module to display a window
- 14.** Define complex class in Ruby and do write methods to carry operations on complex objects.
- 15.** Write a program which illustrates the use of associative arrays in perl.
- 16.** Write perl program takes set names along the command line and prints whether they are regular files or special files
- 17.** Write a perl program to implement UNIX `passwd' program
- 18.** An example perl program to connect to a MySQL database table and executing simple commands.
- 19.** Example PHP program for contact us page.
- 20.** User Authentication:

Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a PHP for doing the following.

 1. Create a Cookie and add these four user id's and passwords to this Cookie.
 2. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords) available in the cookies.
If he is a valid user (i.e., user-name and password match) you should welcome him by name (user-name) else you should display "You are not an authenticated user ". Use init-parameters to do this.
- 21.** Example PHP program for registering users of a website and login.
- 22.** Install a database(Mysql or Oracle).

Create a table which should contain at least the following fields: name, password, email-id, phone number(these should hold the data from the registration form).

Write a PHP program to connect to that database and extract data from the tables and display them. Experiment with various SQL queries.

Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page (week2).
- 23.** Write a PHP which does the following job:

Insert the details of the 3 or 4 users who register with the web site (week9) by using registration form. Authenticate the user when he submits the login form using the user name and password from the database (similar to week8 instead of cookies).
- 24.** Create tables in the database which contain the details of items (books in our case like Book name, Price, Quantity, Amount) of each category. Modify your catalogue page (week 2)in

such a way that you should connect to the database and extract data from the tables and display them in the catalogue page using PHP

25. HTTP is a stateless protocol. Session is required to maintain the state.

The user may add some items to cart from the catalog page. He can check the cart page for the selected items. He may visit the catalogue again and select some more items. Here our interest is the selected items should be added to the old cart rather than a new cart. Multiple users can do the same thing at a time (i.e., from different systems in the LAN using the ip-address instead of local host). This can be achieved through the use of sessions. Every user will have his own session which will be created after his successful login to the website. When the user logs out his session should get invalidated (by using the method session. Invalidate ()).

Modify your catalogue and cart PHP pages to achieve the above mentioned functionality using sessions.



**III Year B. Tech. –II Sem.
WEB TECHNOLOGIES LAB**

List of experiments to be conducted for this semester

No. of Credits: 2

PROGRAMS LIST:

Experiment No-1

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

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 “**MCA**” the catalogue for MCABooks 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.

| Web Site Name | | | | |
|---------------------|-----------------------------|--------------|-----------|------|
| Logo Home | Login | Registration | Catalogue | Cart |
| mca mba BCA | Description of the Web Site | | | |

2. LOGIN PAGE

| Web Site Name | | | | |
|---------------------|---|--------------|-----------|------|
| Logo Home | Login | Registration | Catalogue | Cart |
| MCA MBA BCA | Login : <input type="text" value="11a51f0003"/> Password: <input type="password" value="*****"/> <input type="button" value="Submit"/> <input type="button" value="Reset"/> | | | |

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:

- Snap shot of Cover Page.
- Author Name.
- Publisher.
- Price.
- Add to cart button.

| Logo | Web Site Name | | | |
|------|---|--|-----------|---|
| Home | Login | Registration | Catalogue | Cart |
| MCA |  | Book : XML Bible Author : Winston Publication : Wiely | \$ 40.5 |  Add to cart |
| MBA |  | Book : AI Author : S.Russel Publication : Princeton hall | \$ 63 |  Add to cart |
| BCA |  | Book : Java 2 Author : Watson Publication : BPB publications | \$ 35.5 |  Add to cart |
| |  | Book : HTML in 24 hours Author : Sam Peter Publication : Sam | \$ 50 |  Add to cart |

EXPERIMENT 2:

4. 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 | Book name | Price | Quantity | Amount |
| ECE | Java 2 | \$35.5 | 2 | \$70 |
| EEE | XML bible | \$40.5 | 1 | \$40.5 |
| CIVIL | Total amount - | | | \$130.5 |

5. REGISTRATION PAGE

Create a “registration form” with following fields

- Name (Text field)
- Password (password field)
- E-mail id (text field)
- Phone number (text field)
- Sex (radio button)

- f. Date of birth (3 select boxes)
- g. Languages known (check boxes – English, Telugu, Hindi, Tamil)
- h. Address (text area)

EXPERIMENT 3:

5. Design a web page using CSS (Cascading Style Sheets) which includes the following:

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>
```

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; zindex: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; zindex: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>
```

EXPERIMENT 4:

- 6.** 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
- Price

Write a Document Type Definition (DTD) and XML schemas to validate the above XML file. It should be valid syntactically. Hint: You can use some xml editors like XML-spy

Display the XML file as follows: The contents should be displayed in a table. The header of the table should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold. Use your own colors for remaining columns.

Use XSL and CSS for the above purpose. Note: Give at least for 4 books.

EXPERIMENT 5:

7. Write Ruby program reads a number and calculates the factorial value of it and prints the same.
8. Write a Ruby program which counts number of lines in a text files using its regular Expressions facility.
9. Write a Ruby program that uses iterator to find out the length of a string.
10. Write simple Ruby programs that uses arrays in Ruby.

EXPERIMENT 6:

11. Write programs which uses associative arrays concept of Ruby.
12. Write Ruby program which uses Math module to find area of a triangle.
13. Write Ruby program which uses tk module to display a window
14. Define complex class in Ruby and do write methods to carry operations on complex objects.

EXPERIMENT 7:

15. Write a program which illustrates the use of associative arrays in perl.
16. Write perl program takes set names along the command line and prints whether they are regular files or special files
17. Write a perl program to implement UNIX `passwd' program
18. An example perl program to connect to a MySQL database table and executing simple commands.

EXPERIMENT 8:

19. Example PHP program for contact us page.

20. User Authentication:

Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a PHP for doing the following.

1. Create a Cookie and add these four user id's and passwords to this Cookie.
2. Read the user id and passwords entered in the Login form (EXPERIMENT1) and authenticate with the values (user id and passwords) available in the cookies.

If he is a valid user (i.e., user-name and password match) you should welcome him by name (user-name) else you should display "You are not an authenticated user ". Use init-parameters to do this.

EXPERIMENT 9:

21. Example PHP program for registering users of a website and login.

22. Install a database(Mysql or Oracle).

Create a table which should contain at least the following fields: name, password, email-id, phone number(these should hold the data from the registration form).

Write a PHP program to connect to that database and extract data from the tables and display them. Experiment with various SQL queries.

Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page (EXPERIMENT2).

EXPERIMENT 10:

23. Write a PHP which does the following job:

Insert the details of the 3 or 4 users who register with the web site (EXPERIMENT9) by using registration form. Authenticate the user when he submits the login form using the user name and password from the database (similar to EXPERIMENT8 instead of cookies).

24. Create tables in the database which contain the details of items (books in our case like Book name, Price, Quantity, Amount) of each category. Modify your catalogue page (EXPERIMENT 2)in such a way that you should connect to the database and extract data from the tables and display them in the catalogue page using PHP

EXPERIMENT 11:

25. HTTP is a stateless protocol. Session is required to maintain the state.

The user may add some items to cart from the catalog page. He can check the cart page for the selected items. He may visit the catalogue again and select some more items. Here our interest is the selected items should be added to the old cart rather than a new cart. Multiple users can do the same thing at a time (i.e., from different systems in the LAN using the ip-address instead of local host). This can be achieved through the use of sessions. Every user will have his own session which will be created after his successful login to the website. When the user logs out his session should get invalidated (by using the method session. Invalidate ()). Modify your catalogue and cart PHP pages to achieve the above mentioned functionality using sessions.

EXPERIMENT 12 & 13:**MINI PROJECT:**

Design any web application of your choice (Eg. student information management, super market management, mobile store etc.)

INDEX**Academic Year 2018 - 2019**

| Prog. No. | Date of Completion | Name of Program/Experiment | Marks / Grade | Signature of faculty |
|----------------------|-------------------------------|-----------------------------------|------------------------------|---------------------------------|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| Prog. No. | Date of Completion | Name of Program/Experiment | Marks / Grade | Signature of faculty |
|----------------------|-------------------------------|-----------------------------------|------------------------------|---------------------------------|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

WEEK-1:**1. Design the following static web pages required for an online book store web site.****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 “**MCA**” the catalogue for MCABooks 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 |
| mca mba BCA | Description of the Web Site | | | | |

2.LOGIN PAGE

| Logo | Web Site Name | | | | |
|-------------------|---|---------------------------------------|--------------------------------------|-----------|------|
| | Home | Login | Registration | Catalogue | Cart |
| MCA MBA BCA | Login : <input type="text" value="11a51f0003"/> Password: <input type="password" value="*****"/> | <input type="button" value="Submit"/> | <input type="button" value="Reset"/> | | |

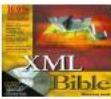
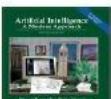
3. CATALOGUE PAGE

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:

- 23. Snap shot of Cover Page.
- 24. Author Name.
- 25. Publisher.
- 26. Price.

27.Add to cart button.

| Logo | Web Site Name | | | |
|------|---|--|--------------|---|
| | Home | Login | Registration | Catalogue |
| MCA |  | Book : XML Bible Author : Winston Publication : Wiley | \$ 40.5 |  Add to cart |
| MBA |  | Book : AI Author : S.Russel Publication : Princeton hall | \$ 63 |  Add to cart |
| BCA |  | Book : Java 2 Author : Watson Publication : BPB publications | \$ 35.5 |  Add to cart |
| | | Book : HTML in 24 hours Author : Sam Peter Publication : Sam | \$ 50 |  Add to cart |

AIM:

To write a program to implement functions of Dictionary using Hashing (linear probing, quadratic probing, double probing - division method, Multiplication method, Universal hashing)

DESCRIPTION:

HTML stands for Hyper Text Markup Language. HTML is a markup language. A markup language is a set of markup tags. The tags describe document content. HTML documents contain HTML tags and plain text. HTML documents are also called web pages.

HTML markup tags are usually called HTML tags. HTML tags are keywords (tag names) surrounded by angle brackets like <html>. HTML tags normally come in pairs like and . The first tag in a pair is the start tag, the second tag is the end tag. The end tag is written like the start tag, with a forward slash before the tag name. Start and end tags are also called opening tags and closing tags.

Syntax:**Frames:**

```
<frameset cols="values"> <frame src="filename">
</frameset>
```

Image:

```

```

Font:

```
<font color="color" size="number">
```

Link:

```
<a href="URL">
```

Backgroundcolor:

```
<body bgcolor="color_name|hex_number|rgb_number">
```

Form tag:

```
<form >
```

```
First name: <input type="text" name="fname"><br>
Last name: <input type="text" name="lname"><br>
<input type="submit" value="Submit">
```

```
</form>
```

Button:

```
<input type="button" value="name">
```

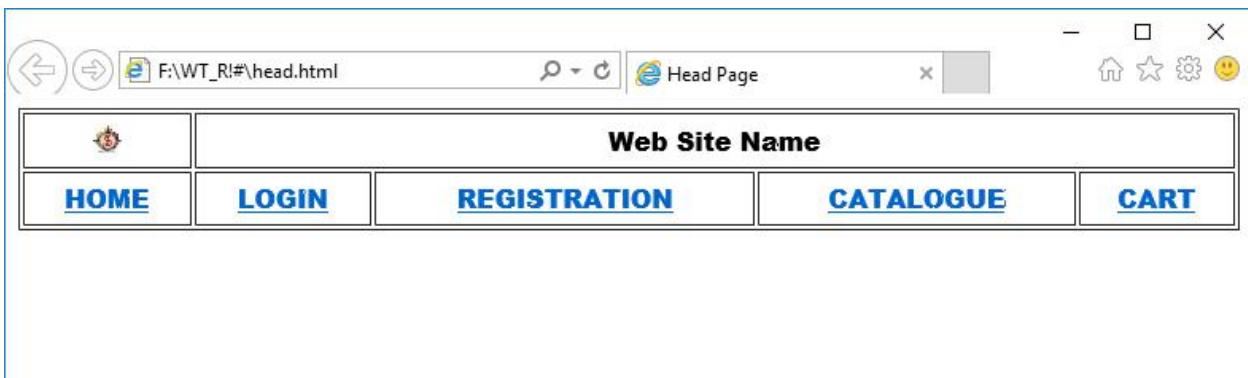
PROGRAM:**head.html**

```
<html>
<head>
    <title>Head Page</title>

</head>
<body>
    <font face="Arial Black" size="3">

        <table border="1" cellspacing="2" cellpadding="5"
            width="100%"> <tr>

            <td align="center"></td> <td colspan="4"
                align="center">Web Site Name</td>
        </tr>
        <tr>
            <td align="center"><a href="description.html"
                target="des_page">HOME</a></td> <td align="center"><a
                href="login.html" target="des_page">LOGIN</a></td>
            <td align="center"><a href="registration.html"
                target="des_page">REGISTRATION</a></td> <td align="center"><a
                href="catalogue.html" target="des_page">CATALOGUE</a></td> <td
                align="center"><a href="cart.html" target="des_page">CART</a></td>
        </tr>
    </table>
    </font>
</body> </html>
```

OUTPUT:**dept.html**

```

<html>
<head>
    <title>Departments Page</title>

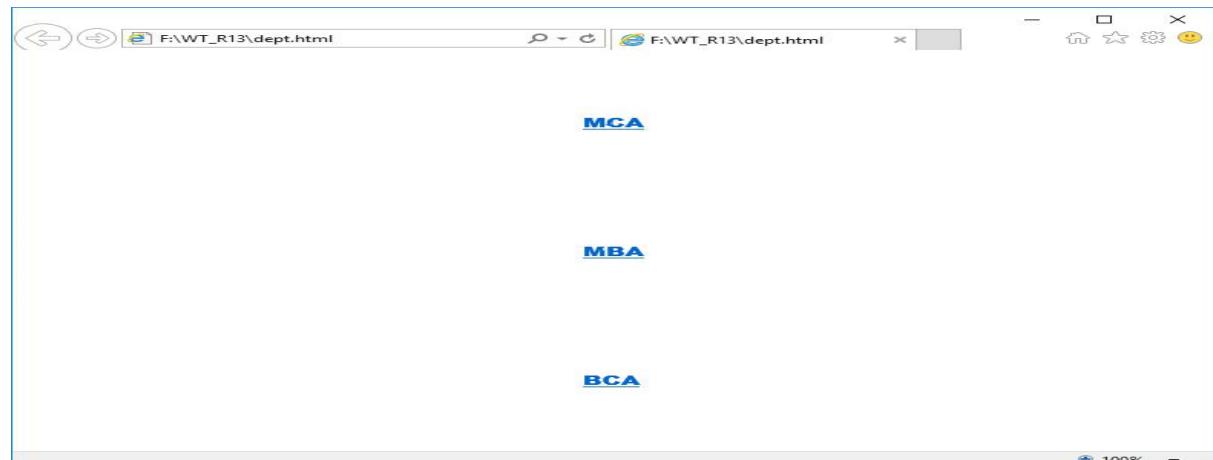
</head>
<body>
<font face="Arial Black" size="4">
<table align="center" height="100%">

    <tr>
        <td><a href="cat_mca.html" target="des_page">MCA</a></td>
    </tr>
    <tr>

        <td><a href="cat_mba.html" target="des_page">MBA</a></td>
    </tr>
    <tr>
        <td><a href="cat_bca.html" target="des_page">BCA</a></td>

    </tr>
</table>
</font>
</body>
</html>

```

OUTPUT:

desc.html

```
<html>
  <head>
    <title> Description page</title>

  </head>
  <body>
    <br><br><br><br><br><br><br><br><br><br>
    <font face="TIMES NEW ROMAN" size="5">

    <center>
      Description of the Website
    </center>
    </font>

  </body>
</html>
```

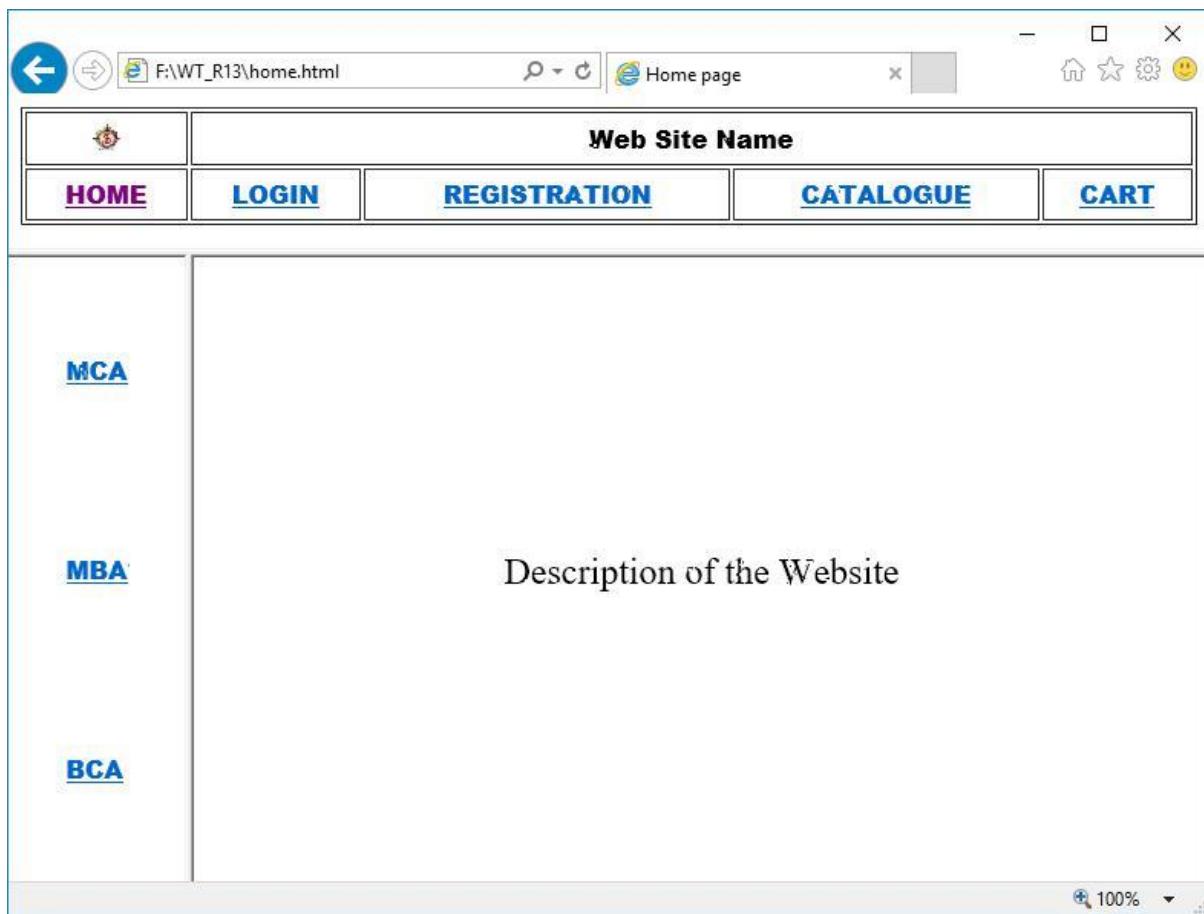
OUTPUT:**home.html**

```
<html>
  <head>
    <title>Home page</title>

  </head>
  <frameset rows="20,80">
    <frame src="head.html" name="head_page">
    <frameset cols="15,85">

      <frame src="dept.html" name="dept_page">
      <frame src="desc.html" name="des_page">
    </frameset>
  </frameset>

</html>
```

OUTPUT:

2. LOGIN PAGE: Login page must contain Login field, Password field, Submit and reset buttons.

login.html

```
<html>
  <head>
    <title>Login page</title>
  </head>
  <body>
    <center>
      <font face="Arial Black" size="4"><u><b>LOGIN
      FORM</b></u></font> <br><br>
      <form name="f1" action="" method="post">
        <table frame="box" cellspacing="10">
          <tr>
```

```

<td>Login</td>
<td><input type="text" size="25"></td>

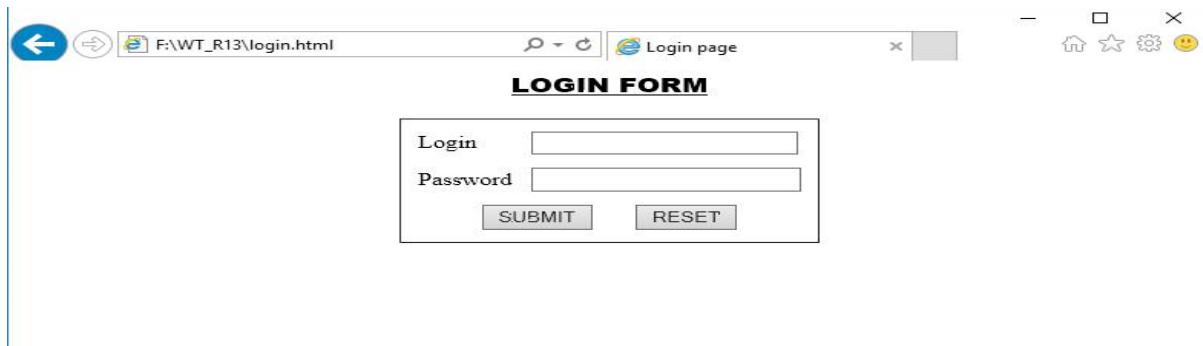
</tr>
<tr>
<td>Password</td>
<td><input type="password" size="25"></td>

</tr>
<tr>
<td colspan="2" align="center"><input type="submit" value="SUBMIT">
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<input type="reset" value="RESET"></td>
</tr>
</table>
</form>

</center>
</body>
</html>

```

OUTPUT:



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.

catalogue.html

```

<html>
<head>
```

```
<title> Catalogue Page</title>
</head>
<body>
<table width="100%" height="100%" border="1">

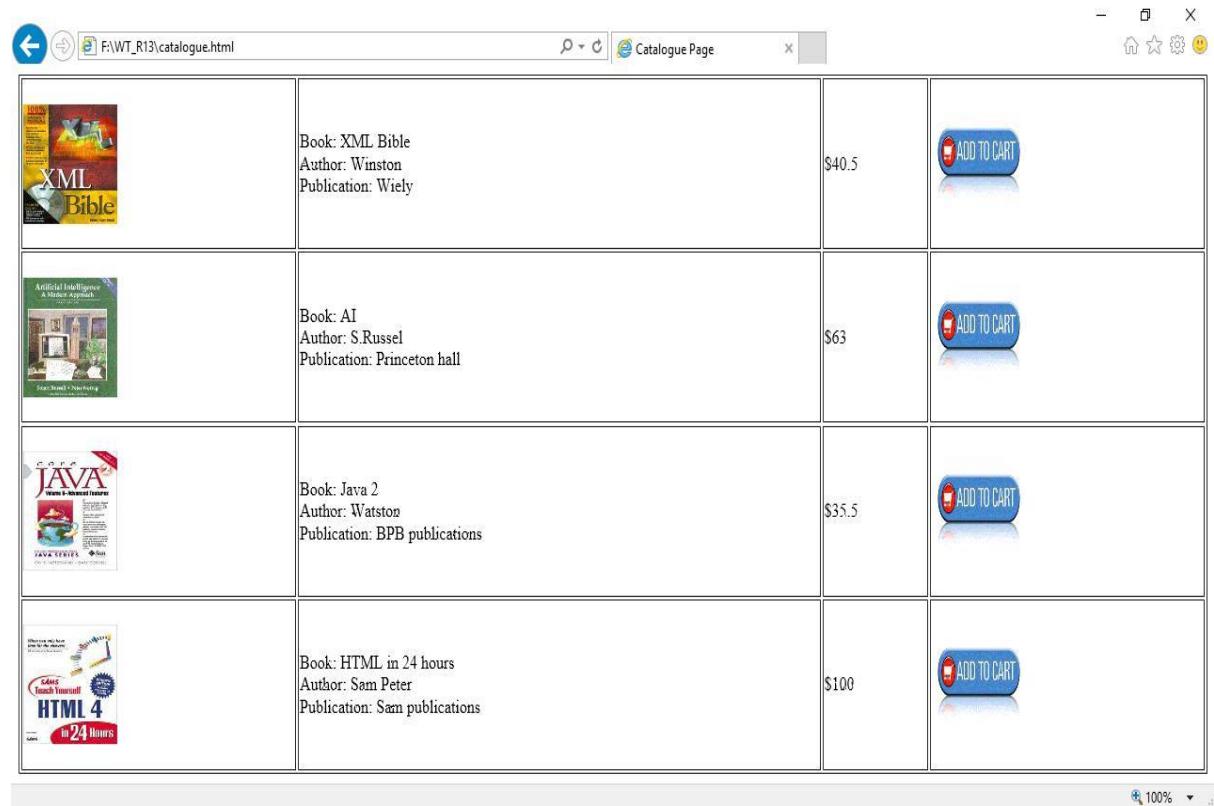
<tr>
<td></td>
<td>Book: XML Bible
<br>Author: Winston
<br>Publication: Wiely</td>
<td>$40.5</td>
<td></td>
</tr>

<tr>
<td></td>
<td>Book: AI
<br>Author: S.Russel
<br>Publication: Princeton hall</td>
<td>$63</td>
<td></td>
</tr>

<tr>
<td></td>
<td>Book: Java 2
<br>Author: Watston
<br>Publication: BPB publications</td>
<td>$35.5</td>
<td></td>
</tr>
<tr>

<td></td>
<td>Book: HTML in 24 hours
<br>Author: Sam Peter
<br>Publication: Sam publications</td>
<td>$100</td>
<td></td>
</tr>
</table>

</body>
</html>
```

OUTPUT:


| | | | |
|--|--|--------|--|
| | Book: XML Bible Author: Winston Publication: Wiley | \$40.5 | |
| | Book: AI Author: S.Russel Publication: Princeton hall | \$63 | |
| | Book: Java 2 Author: Watson Publication: BPB publications | \$35.5 | |
| | Book: HTML in 24 hours Author: Sam Peter Publication: Sam publications | \$100 | |

- 1. What is HTML?**
- 2. Explain different types of lists.**
- 3. Give different types of form elements.**
- 4. How to insert frames into web page? Give its importance.**
- 5. Give syntax for creating a form.**

WEEK-2

- 4. 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 | | | |
|----------------------------|------------------|-----------------------|-----------------|---------------|
| | 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 |

cat_mca.html

```

<html>
<head>
    <title> MCA Catalogue Page</title>

</head>
<body>
    <table width="100%" height="100%" border="1">
        <tr>

            <td></td>
            <td>Book: XML Bible
                <br>Author: Winston
                <br>Publication: Wiely</td>

            <td>$40.5</td>
            <td></td>
        </tr>
        <tr>

            <td></td>
            <td>Book: HTML in 24 hours
                <br>Author: Sam Peter
                <br>Publication: Sam publications</td>

            <td>$100</td>
            <td></td>
        </tr>
    </table>

</body>
</html>

```

OUTPUT:

| | | | |
|--|--|--------|--|
| | Book: XML Bible Author: Winston Publication: Wiley | \$40.5 | |
| | Book: HTML in 24 hours Author: Sam Peter Publication: Sam publications | \$100 | |

cat_bca.html

```

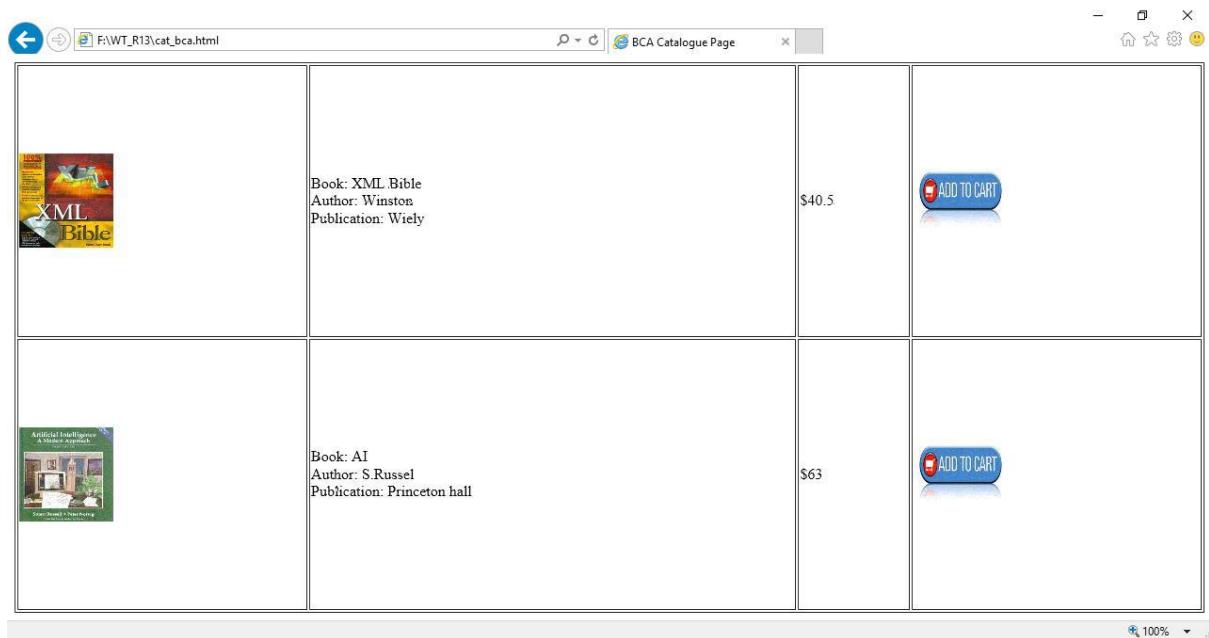
<html>
<head>
    <title> BCA Catalogue Page</title>
</head>
<body>
    <table width="100%" height="100%" border="1">
        <tr>
            <td></td>
            <td>Book: XML Bible
                <br>Author: Winston
                <br>Publication: Wiley</td>
            <td>$40.5</td>
            <td></td>
        </tr>
        <tr>
            <td></td>
            <td>Book: AI
                <br>Author: S.Russel
                <br>Publication: Princeton hall</td>
            <td>$63</td>
            <td></td>
        </tr>
    </table>
</body>

```

```
</table>
```

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

OUTPUT:



The screenshot shows a web browser window displaying a catalog page titled "BCA Catalogue Page". The page contains two entries for books, each in a separate table row.

| | | | |
|---|---|--------|---|
|  | Book: XML Bible Author: Winston Publication: Wiley | \$40.5 |  |
|  | Book: AI Author: S.Russel Publication: Princeton hall | \$63 |  |

VIVA-QUESTIONS

- 1. Give the difference between check box and radio button.**
- 2. Give the difference between text field and text area.**
- 3. Give the syntax for drop down list.**
- 4. Give the syntax for check box group.**
- 5. Give the syntax for radio button group.**

5. REGISTRATION PAGE:

Create a “registration form” with 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)

registration.html

```
<html>
<head>
    <title>Registration Page</title>
</head>
<body>
<center>
<h3 align="center"><u>REGISTRATION PAGE</u></h3>
<table border="3">
<tr><td>
<form name="f1" action="" method="post" onsubmit="">
<table cellspacing="10" cellpadding="5">
    <tr><td>NAME</td><td><input type="text" size="30" name="uname"/></td></tr>
    <tr><td>PASSWORD</td><td><input type="password" size="30" name="pass"/></td></tr>
    <tr><td>E-MAIL ID</td><td><input type="text" size="30" name="email"/></td></tr>
    <tr><td> PHONE NUMBER</td><td><input type="text" size="15" name="phone"/></td></tr>
    <tr><td>GENDER</td><td><input type="radio" name="gen" value="m" />MALE
        <input type="radio" name="gen" value="f" />FEMALE </td></tr>
    <tr><td>DATE OF BIRTH</td>
        <td><select name="day">
            <option value="day">DAY</option>
            <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>
<select name="month">
    <option value="month">MONTH</option>
    <option value="jan">JANUARY</option>
    <option value="feb">FEBRUARY</option>
    <option value="mar">MARCH</option>
    <option value="apr">APRIL</option>
    <option value="may">MAY</option>
    <option value="jun">JUNE</option>
    <option value="jul">JULY</option>
    <option value="aug">AUGUST</option>
    <option value="sep">SEPTEMBER</option>
    <option value="oct">OCTOBER</option>
    <option value="nov">NOVEMBER</option>
    <option value="dec">DECEMBER</option>
</select>
<select name="year">
    <option value="year">YEAR</option>
    <option value="1986">1986</option>
    <option value="1987">1987</option>
    <option value="1988">1988</option>
    <option value="1989">1989</option>
    <option value="1990">1990</option>
    <option value="1991">1991</option>
    <option value="1992">1992</option>
    <option value="1993">1993</option>
    <option value="1994">1994</option>
    <option value="1995">1995</option>
    <option value="1996">1996</option>
    <option value="1997">1997</option>
    <option value="1998">1998</option>
    <option value="1999">1999</option>
    <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">1980</option>
<option value="2012">1981</option>
<option value="2013">1982</option>
<option value="2014">1983</option>
</select></td></tr>
<tr><td>LANGUAGES KNOWN</td>
<td>
    <input type="checkbox" value="eng"
    name="lang" />ENGLISH <input
    type="checkbox" value="tel" name="lang"
    />TELUGU <input type="checkbox" value="hin"
    name="lang" />HINDI <input type="checkbox"
    value="tam" name="lang" />TAMIL
</td></tr>
<tr>
<td>ADDRESS</td>
<td><textarea name="addr" cols="25" rows="5"></textarea></td></tr>
<tr><td colspan="2" align="center"><input type="submit" value="SUBMIT"/>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="RESET"
    /></td>
</tr>
</table>
</form>
</td>
</tr>

</table>
</center>
</body>
</html>

```

OUTPUT:

The screenshot shows a web browser window with the title "Registration Page". The page itself is titled "REGISTRATION PAGE" in bold capital letters. It contains the following fields:

- NAME: A text input field.
- PASSWORD: A text input field.
- E-MAIL ID: A text input field.
- PHONE NUMBER: A text input field.
- GENDER: Two radio buttons labeled "MALE" and "FEMALE".
- DATE OF BIRTH: Three dropdown menus labeled "DAY", "MONTH", and "YEAR".
- LANGUAGES KNOWN: A group of four checkboxes labeled "ENGLISH", "TELUGU", "HINDI", and "TAMIL".
- ADDRESS: A text area with scroll bars.

At the bottom of the form are two buttons: "SUBMIT" and "RESET".

VIVA-QUESTIONS

1. Give the difference between check box and radio button.
2. Give the difference between text field and text area.
3. Give the syntax for drop down list.
4. Give the syntax for check box group.
5. Give the syntax for radio button group.

WEEK 3:

5.Design a web page using CSS (Cascading Style Sheets) which includes the following:

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>
```

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; zindex: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; zindex: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>
```

AIM:

A program to design a web page using CSS.

DESCRIPTION:

CSS stands for Cascading Style Sheets .Styles define **how to display** HTML elements. CSS is a breakthrough in Web design because it allows developers to control the style and layout of multiple Web pages all at once. As a Web developer we can define a style for each HTML element and apply it to as many Web pages as we want. To make a global change, simply change the style, and all elements in the Web are updated automatically. There are 3 ways of inserting a style sheet:

1. External Style sheets
2. Internal Style sheets
3. Multiple Style sheets

SYNTAX:

inline style sheet:

```
<html>
<head>
</head>
<body>
<p style="styles to be applied">
</p>
</body>
</html>
```

internal style sheet:

```
<html>
<head>
<style type="text/css">
//styles to be applied
</style>
</head>
<body>
</body>
</html>
```

external style sheet:

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="externalstylesheet.css"/> </head>
<body class="center">
</body>
</html>
```

externalstylesheet.css

```
body.center
{
//Styles to be applied
}
```

PROGRAM:**styles.css**

```
h1
{
    color:red;
    font-family:caStellar;
    font-size:22pt;
    text-decoration:underline;

}
h2
{
    color:blue;
    font-family:Chiller;
    font-size:18pt;
    text-decoration:overline;
}
p
{
    color:magenta;
    font-family:Trebuchet MS;
    font-size:14pt;
```

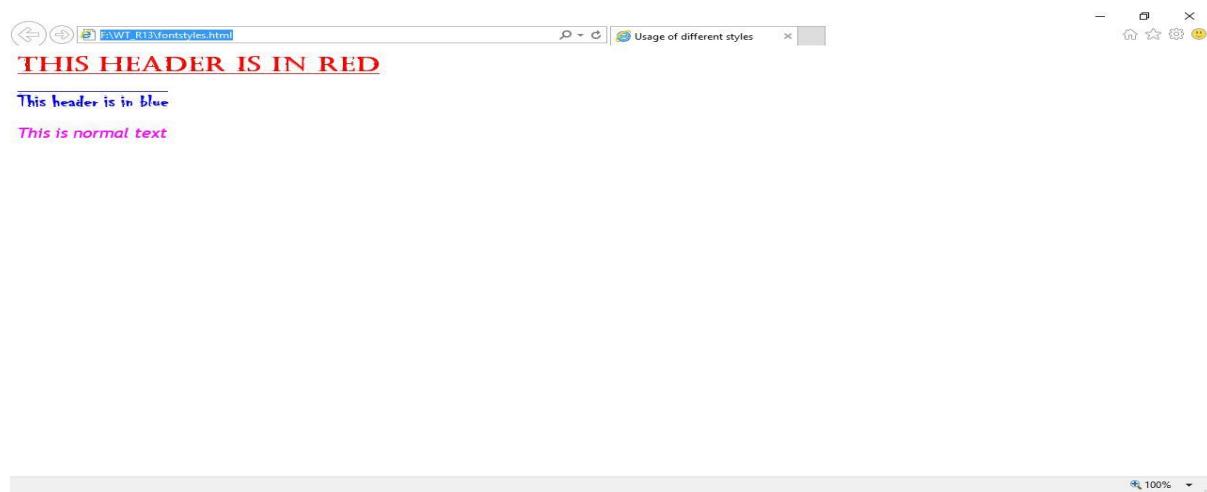
```

        font-style:italic;
    }
fontstyles.html

<html>
<head>
    <title>Usage of different font,styles and colors </title>
    <link rel="stylesheet" type="text/css" href="styles.css"/>
</head>
<body>
    <h1>This header is in red</h1>

    <h2>This header is in blue</h2>
    <p>This is normal text</p>
</body>
</html>

```

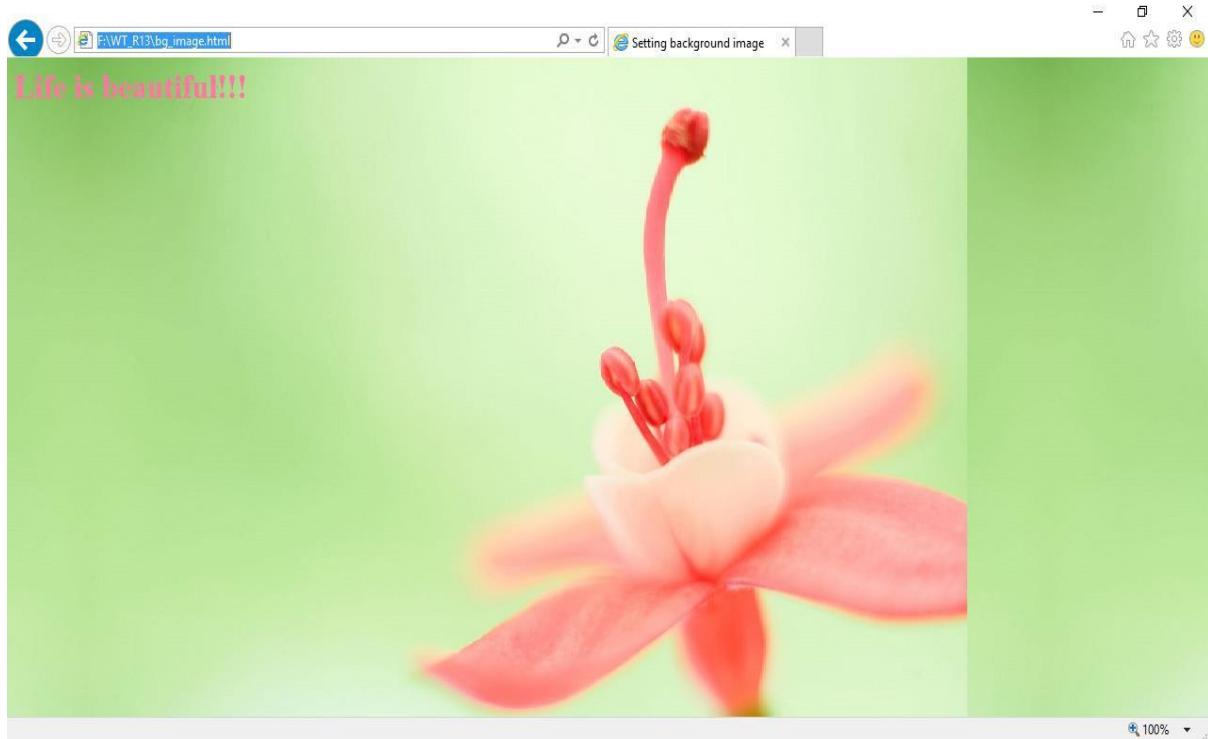
OUTPUT:**B. Set a background image for both the page and single elements on the page.****bg_image.html**

```

<html>
<head>
    <title>Setting background image</title>
    <style type="text/css">
        body
        {
            background-image:url("img11.jpg");
        }
    </style>
</head>
<body text="#ee78a2">
    <h1>Life is beautiful!!!</h1>

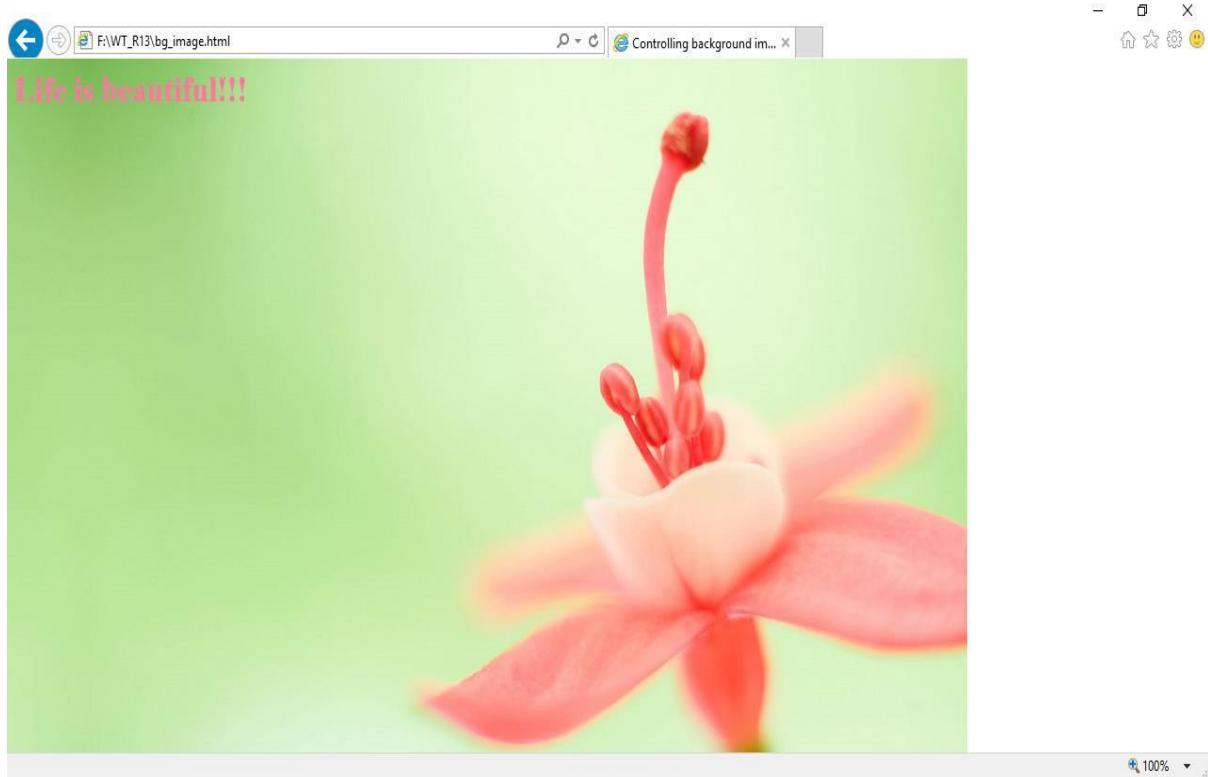
```

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

OUTPUT:**C. Control the repetition of the image with the background-repeat property.****bg_repeat.html**

```
<html>
<head>
    <title>Controlling background image</title>
    <style type="text/css">
        body
        {
            background-image:url("img11.jpg");
            background-repeat:no-repeat
        }
    </style>
</head>
<body text="#fe78a2">
    <h1>Life is beautiful!!!</h1>
</body>
</html>
```

OUTPUT



VIVA QUESTIONS:

1. Why CSS?
2. Explain the different ways for applying styles.
3. Give the syntax for inline styles.
4. Give the syntax for internal styles.
5. Give the syntax for external styles.

WEEK 4:

6. Write an XML file which will display the Book information which includes the following:

- a. Title of the book
- b. Author Name
- c. ISBN number
- d. Publisher name
- e. Edition
- f. Price

Write a Document Type Definition (DTD) and XML schemas to validate the above XML file. It should be valid syntactically. Hint: You can use some xml editors like XML-spy
Display the XML file as follows : The contents should be displayed in a table. The header of the table should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold. Use your own colors for remaining columns.
Use XSL and CSS for the above purpose. Note: Give at least for 4 books.

AIM:

To write an XML file which will display the book information and write a DTD and Schema to validate that xml file.

DESCRIPTION:

XML stands for eXtensible Markup Language. It is designed to transport and store data. XML tags are not predefined. We must define your own tags. XML is designed to be self-descriptive. XML is a W3C Recommendation. XML data is stored in plain text format. This provides a software and hardware independent way of storing data. This makes it much easier to create data that can be shared by different applications.

SYNTAX:**Xml:**

```
<root>
<child>
<subchild>.....</subchild>
</child>
</root>
```

DTD:

```
<!DOCTYPE root-element [element-declarations]>
```

Xml schema:

```
<?xml version="1.0"?>
<xsschema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xselement name="root"> <xstype of the xml file>
<xselement name="child" type="xstype of the child"/> </xstype of the xml file>
</xselement>
</xsschema>
```

PROGRAM:**catalog.dtd**

```
<!ELEMENT Catalog (Book)*>
<!ELEMENT Book (Title, Author, Publication ,Edition, ISBN, Price)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Publication (#PCDATA)>
<!ELEMENT Edition (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
<!ELEMENT Price (#PCDATA)>
```

library.css

Catalog

```
{
    font-family:arial;
    color:blue;

    font-size:16pt
}
```

Book

```
{
    font-family:times new roman;
    color:blue;
    table-layout:auto;

    font-size:14pt
}
```

Title

```
{
    font-family:arial;
    color:green;
    margin-left:20pt;
    font-size:12pt
}
```

}

Author

```
{
    font-family:arial;
    color:magenta
}
```

Publication,Edition,ISBN,Price

```
{
    display:block;
    font-family:arial;
    color:black;
    font-size:10pt;
    margin-left:40pt
}
```

Cataloguedemo.xml

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?> <?xml-stylesheet
type="text/css" href="library.css" ?>
<!DOCTYPE Catalog SYSTEM "catalog.dtd">
<Catalog>
<Book>

<Title>XML Bible</Title>
<Author>Winston</Author>
<Publication>Wiely</Publication>
<Edition>Fifth Edition</Edition>
<ISBN>0-7645-4760-7</ISBN>
<Price>$40.5</Price >
</Book>

<Book>
<Title>Artificial Intelligence</Title>
<Author>S. Russel</Author>
<Publication> Princeton Hall </Publication>
<Edition> Sixth Edition</Edition>
<ISBN> 0-13-1038-5-2 </ISBN>
<Price>$63</Price>

</Book>

<Book>
<Title>Java 2</Title>

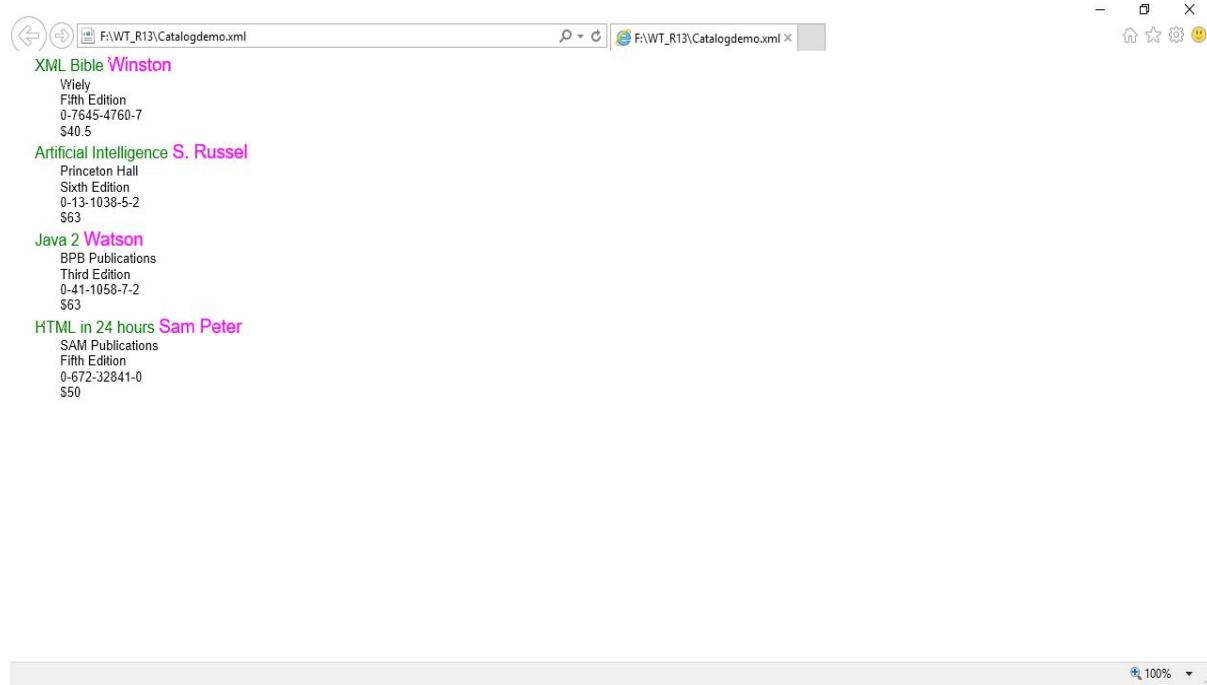
<Author>Watson</Author>
<Publication>BPB Publications</Publication>
<Edition>Third Edition</Edition>
<ISBN>0-41-1058-7-2</ISBN>

<Price>$63</Price>
</Book>

<Book>

<Title>HTML in 24 hours </Title>
<Author> Sam Peter</Author>
<Publication> SAM Publications </Publication>
<Edition>Fifth Edition</Edition>
<ISBN> 0-672-32841-0 </ISBN>
<Price> $50 </Price >
</Book>
</Catalog>
```

OUTPUT:



VIVA QUESTIONS:

1. Explain about DTD?
2. What is XML schema?
3. What is XML Name Space?
4. What is parser?
5. What is XSLT ?

WEEK 5:

7. Write Ruby program reads a number and calculates the factorial value of it and prints the same.

AIM: To write Ruby program reads a number and calculates the factorial value of it and prints the same.

DESCRIPTION: To find factorial of a given number, start multiplying from 1 to given number.

ALGORITHM:

Step 1. Start
step 2. Read the number n
step 3. [Initialize]
 i=1, total=1
step 4. Repeat step 4 through 6 until i=n
step 5. total=total*i
step 6. i=i+1
step 7. Print total
step 8. stop

PROGRAM:**fact.rb**

```
puts "Enter a number>>"  
n=gets.to_i  
f=1  
i=1  
while i<=n  
do  
    f=f*i  
    i=i+1  
end  
puts "factorial of #{n} is #{f}"
```

OUTPUT:

```
C:\WINDOWS\system32\cmd.exe  
C:>ruby -w fact.rb  
Enter a number>>5  
factorial of 5 is 120  
  
C:>ruby -w fact.rb  
Enter a number>>10  
factorial of 10 is 3628800  
  
C:>ruby -w fact.rb  
Enter a number>>15  
factorial of 15 is 1307674368000  
C:>
```

VIVA QUESTIONS:

1. What are ruby gems?
2. What is the difference between a symbol and string?
3. What is the purpose of yield?
4. What are class variables? How do you define them?
5. How do you define instance variables?

8. Write a Ruby program which counts number of lines in a text file using its regular expressions facility

AIM: To write a Ruby program which counts number of lines in a text file using its regular expressions facility.

DESCRIPTION: Some regular expression symbols are

/ - start of regex
 ^ - start of input
 / - end of input, so, escape it:
 \
 . - any character
 * repeated any number of times
 \$ - end of input
 / - end of regex

ALGORITHM:

Step 1: start
 Step 2: var count = 0;
 Step 3: for (var line in file)
 Step 4: if (line.match(/\^/.*/))
 Step 5: count += 1;
 Step 6: end loop
 Step 7: stop

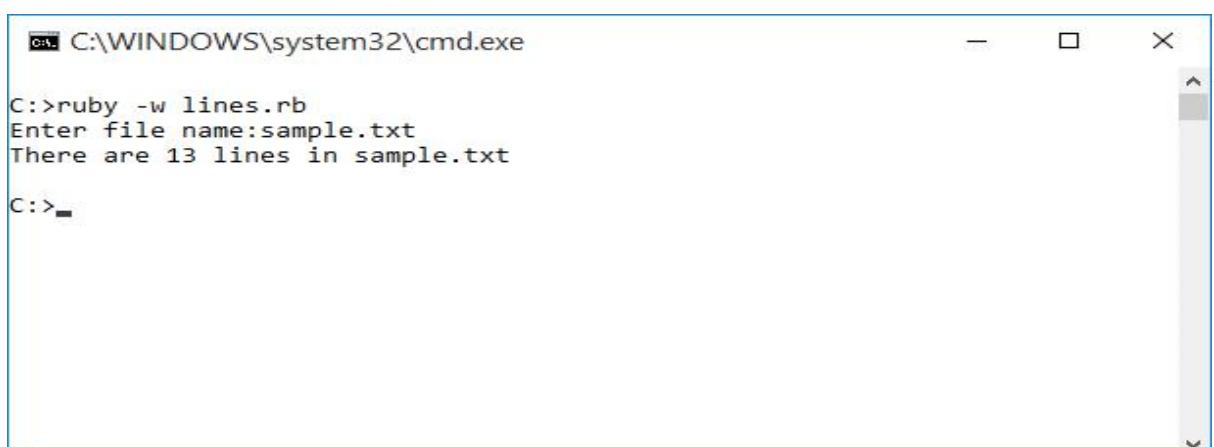
PROGRAM:

```
print "Enter file name:";  

fname = gets.chomp;  

count = IO.readlines(fname).size  

puts "There are #{count} lines in #{fname}";
```

OUTPUT:


The screenshot shows a Windows Command Prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The command entered is 'C:>ruby -w lines.rb'. The user then types 'Enter file name:' followed by 'sample.txt'. The output shows 'There are 13 lines in sample.txt'. The command prompt ends with 'C:>'.

VIVA QUESTIONS:

1. What is regular expression?
2. Does ruby support constructors? How are they declared?
3. How do you define global variables?
4. How can you dynamically define a method body?

9. Write a Ruby program that uses iterates to find out the length of a string?

AIM: To write a Ruby program that uses iterates to find out the length of a string.

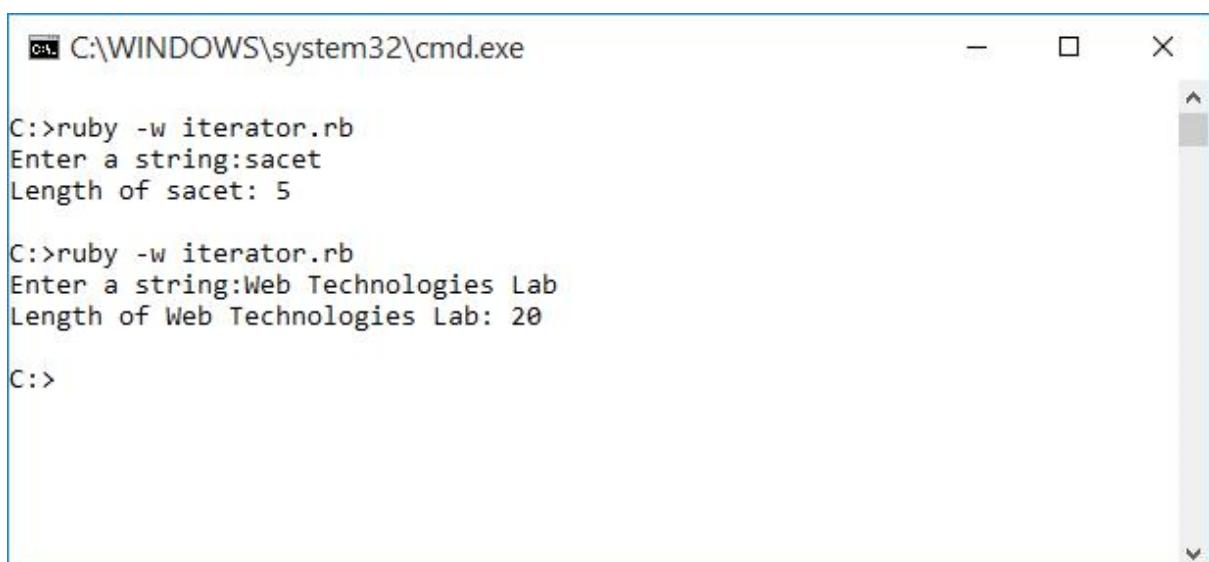
DESCRIPTION: Using loops, length of a string can be found by iterating.

ALGORITHM:

Step1: start
Step 2: declare string s="one two three four"
Step 3: n=s.split(" ") //split this string on space
Step 3: for (i=0;i<n;i++)
Step 4: puts [s,s.length] //display each value of length
Step 5: end loop
Step 6: stop

PROGRAM:**iterator.rb**

```
print "Enter a string:\n\nstr = gets.chomp
c=0
str.each_char do |i|
  c=c+1
end
puts "Length of #{str}: #{c}"
```

OUTPUT

The screenshot shows a Windows Command Prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The window contains the following text output:

```
C:\>ruby -w iterator.rb
Enter a string:sacet
Length of sacet: 5

C:\>ruby -w iterator.rb
Enter a string:Web Technologies Lab
Length of Web Technologies Lab: 20

C:>
```

VIVA-QUESTIONS:

1. Does ruby support multiple inheritances?
2. What is a module?
3. What is the difference between a symbol and string?
4. How can you implement method overloading?

10. Write simple Ruby programs that uses arrays in Ruby

AIM: To write simple Ruby programs that uses arrays in Ruby

DESCRIPTION: There are many ways to create or initialize an array. One way is with the *new* class method: names = Array.new

You can set the size of an array at the time of creating array: names = Array.new(20)

| | ARGUMENT | RETURNS | |
|---------|---|--|---|
| push | Appends its argument to the end of the array. | Element(s) to be appended to end of the array. | A string consisting of the concatenation of all non-nil elements in the array AFTER the action was taken. |
| pop | Returns the last element in the array and deletes that element. | None. | The last element of the array. |
| shift | Returns the first element of the array, deletes that element, and shifts all other elements down one location to fill its empty spot. | None. | The first element in the array. |
| unshift | Shifts all elements of the array up one, and places its argument at the beginning of the array. | Element(s) to be prepended to start of array. | A string consisting of the concatenation of all non-nil elements in the array AFTER the action was taken. |

ALGORITHM:

- Step 1: start
- Step 2: create an array as presidents
- Step 3: pop the last element as president.pop();
- Step 4: pop again and see if the array is empty as president.pop();
- Step 6: Loop over presidents and display them
- Step 7: stop

PROGRAM:

[bsort.rb](#)

```
puts "Bubble Sort"
puts "====="
print "Enter the size of the array:"
n=gets.to_i

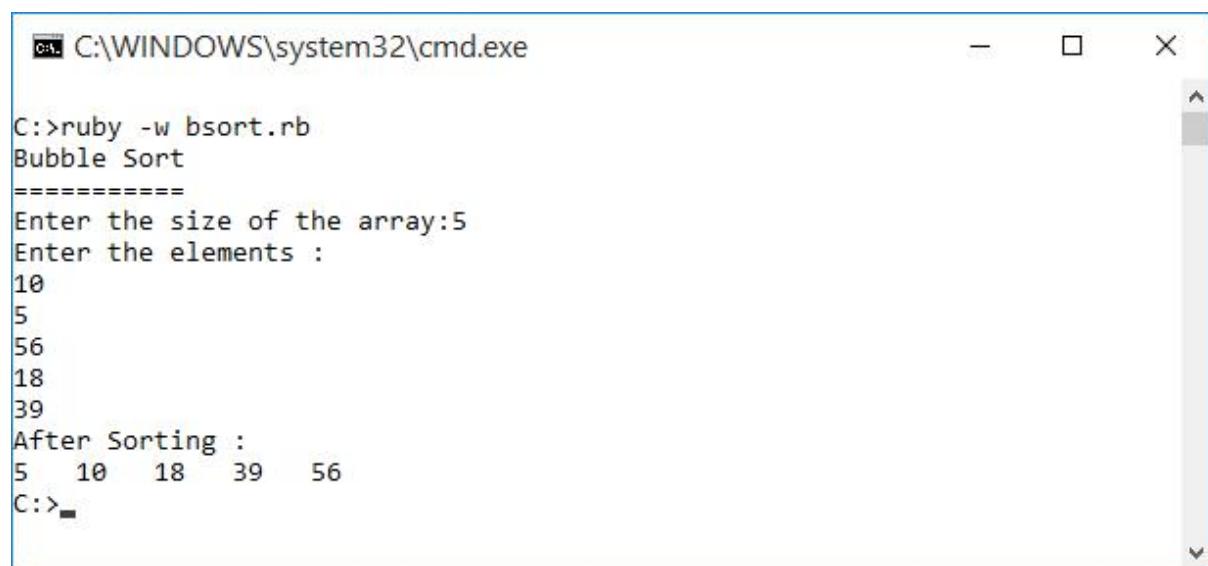
a=Array.new(n)
puts "Enter the elements :"
for i in 0..n-1
  a[i]=gets.to_i

end

#Bubble sort Algorithm
t=0

for i in 0 .. n -1
  for j in i .. n-1
    if (a[i] > a[j])
      t=a[j]
      a[j]=a[i]
      a[i]=t
    end
  end
end
puts "After Sorting :"
for i in 0..n-1
  print "#{a[i]}      "
end
```

OUTPUT:



The screenshot shows a Windows Command Prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The window contains the following text output from a Ruby script:

```
C:\>ruby -w bsort.rb
Bubble Sort
=====
Enter the size of the array:5
Enter the elements :
10
5
56
18
39
After Sorting :
5   10   18   39   56
C:>_
```

Viva-Questions:

1. What is array?
2. Explain about string concatenation?
3. What is the convention for using '!' at the end of a method name?
4. How can you achieve the same effect as multiple inheritances using ruby? What is mixin?

WEEK 6:**11. Write programs which uses associative arrays concept of Ruby.**

AIM: To write programs which uses associative arrays concept of Ruby.

DESCRIPTION: An associative array has elements that are accessed not by sequential index numbers, but by *keys* which can have any sort of value. Such an array is sometimes called a *hash* or *dictionary*; in the ruby world, we prefer the term *hash*. A hash can be constructed by quoting pairs of items within curly braces ({}). You use a key to find something in a hash, much as you use an index to find something in an array.

ALGORITHM:

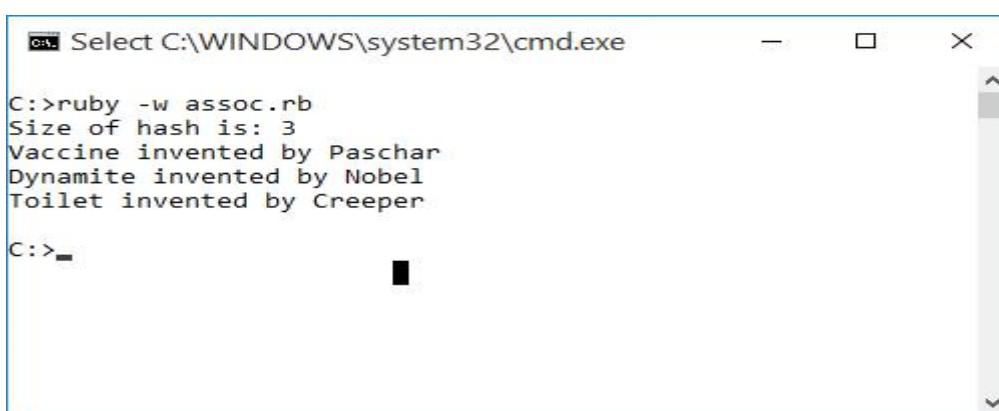
- Step 1: start
- Step 2: name=hash.new // this is creates an hash with a default value of an empty array
- Step 3: names.store(1, "string1")/insert into hash table
- Step 4: names.store(2, "string2")
- Step 5: names.store(3, "string3")
- Step 6: print values

PROGRAM:

```
creators = Hash.new;

creators =
{"Vaccine"=>"Paschar","Dynamite"=>"Nobel","Toilet"=>"Cre
epper"}; size = creators.length;
puts "Size of hash is: #{size}";
creators.each do |key,val|

    puts "#{key} invented by #{val}"
end
```

OUTPUT:


```
C:\>ruby -w assoc.rb
Size of hash is: 3
Vaccine invented by Paschar
Dynamite invented by Nobel
Toilet invented by Creeper
C:>_
```

VIVA-QUESTIONS:

1. What is associative array?
2. Explain sequential indexing?
3. How will you implement an observer pattern?
4. What is the purpose of environment.rb and application.rb file?

12. Write Ruby program which uses Math module to find area of a triangle.

AIM: To write Ruby program which uses Math module to find area of a triangle.

DESCRIPTION: Area of a triangle can be calculated as $\frac{1}{2} \times b \times h$. In math module direct methods Math.sqrt(to_a.zip(oth.to_a).inject(0){|s,(a,b)|s+(a-b)**2}) are available.

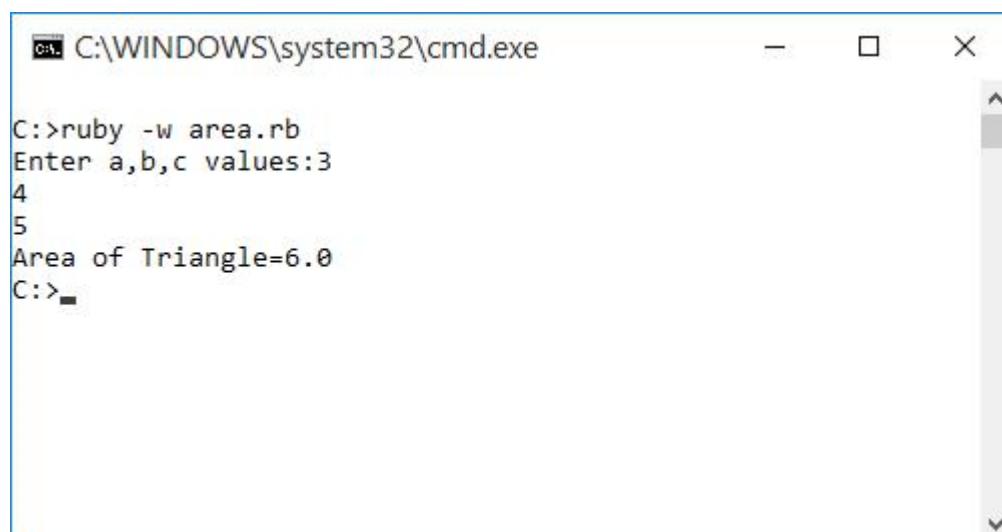
ALGORITHM:

1. ab = @a.distance(@b)
2. bc = @b.distance(@c)
3. ac = @a.distance(@c)
4. s = (ab+bc+ac)/2
5. Math.sqrt(s*(s-ab)*(s-bc)*(s-ac))
6. For distance Math.sqrt(to_a.zip(oth.to_a).inject(0){|s,(a,b)|s+(a-b)**2})

PROGRAM:**area.rb**

```
print "Enter a,b,c values:"
a=gets.to_i
b=gets.to_i
c=gets.to_i
s = (a+b+c)/2

area = Math.sqrt(s*(s-a)*(s-b)*(s-c)).round(2)
print "Area of Triangle=#{area}"
```

OUTPUT:


```
C:\WINDOWS\system32\cmd.exe
C:>ruby -w area.rb
Enter a,b,c values:3
4
5
Area of Triangle=6.0
C:>
```

VIVA-QUESTIONS:

1. Give some of the methods available in Math module.
2. How will you implement a singleton pattern?
3. What is the difference between '&&&', 'and' and '&' operators?
4. How can you create setter and getter methods in ruby?

13. Write Ruby program which uses tk module to display a window

AIM: To write Ruby program which uses tk module to display a window

DESCRIPTION: Standard graphical user interface (GUI) for Ruby is Tk. Tk started out as the GUI for the Tcl scripting language developed by John Ousterhout.

Tk has the unique distinction of being the only cross-platform GUI. The basic component of a Tk-based application is called a widget. A component is also sometimes called a window, since, in Tk, "window" and "widget" are often used interchangeably.

Tk applications follow a widget hierarchy where any number of widgets may be placed within another widget, and those widgets within another widget, ad infinitum. The main widget in a Tk program is referred to as the root widget and can be created by making a new instance of the TkRoot class.

Most Tk-based applications follow the same cycle: create the widgets, place them in the interface, and finally, bind the events associated with each widget to a method.

There are three geometry managers; *place*, *grid* and *pack* that are responsible for controlling the size and location of each of the widgets in the interface.

Installation:

The Ruby Tk bindings are distributed with Ruby but Tk is a separate installation. Windows users can download a single click Tk installation from ActiveState's ActiveTcl.

Mac and Linux users may not need to install it because there is a great chance that its already installed along with OS but if not, you can download prebuilt packages or get the source from the Tcl Developer Xchange.

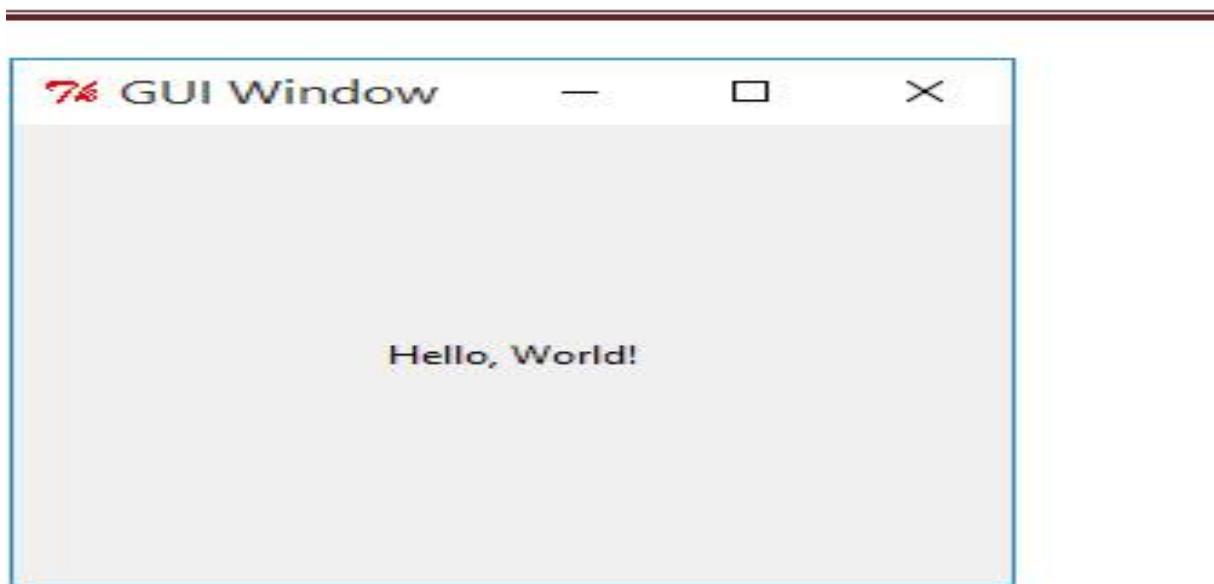
Simple Tk Application:

A typical structure for Ruby/Tk programs is to create the main or root window (an instance of TkRoot), add widgets to it to build up the user interface, and then start the main event loop by calling Tk.mainloop.

PROGRAM:**window.rb**

```
require 'tk'
```

```
root = TkRoot.new { title "GUI Window" }
TkLabel.new(root) do
  text 'Hello, World!'
  pack("side" => "right", "padx"=> "100", "pady"=> "100")
end
Tk.mainloop
```

OUTPUT:**VIVA QUESTIONS:**

1. Explain GUI?
2. What is interface?
3. How can you call the base class method from inside of its overridden method?
4. Define Package?

14.Define complex class in Ruby and do write methods to carry operations on complex objects.

AIM: To define complex class in Ruby and do write methods to carry operations on complex objects.

DESCRIPTION: A complex number is of the form $x+iy$ form. Different operations that can be carried out are addition, subtraction etc.

ALGORITHM:

Addition:

```
t.real = (p.real + q.real)
t.imag = (p.imag + q.imag)
```

Subtraction:

```
t.real = (p.real - q.real)
t.imag = (p.imag - q.imag)
```

Multiplication:

```
t.real=(p.real * q.real) - (p.imag * q.imag)
t.imag=(p.real * q.imag) + (p.imag * q.real)
```

Division:

```
t.real = ((p.imag * q.real) - (p.real * q.imag)) / ((q.real * q.real) + (q.imag * q.imag))
t.imag = ((p.real * q.real) + (p.imag * q.imag)) / ((q.real * q.real) + (q.imag * q.imag))
```

PROGRAM:

Cmplx.rb

```
attr_accessor :real, :imag
def read
    @real=gets.to_i
    @imag=gets.to_i
end
def add(other)
    ob = Cmplx.new
    ob.real = @real+other.real;
    ob.imag = @imag+other.imag;
    return ob
end
def subtract(other)
    ob = Cmplx.new;
    ob.real = @real-other.real;
    ob.imag = @imag-other.imag;
    return ob;
end
```

```

def multiply(other)
    ob = Cmplx.new
    ob.real = (@real * other.real)-(imag*other.imag);
    ob.imag = (@real * other.imag)+(imag*other.real);
    return ob;
end
def divide(other)
    t = Cmplx.new;
    ob = Cmplx.new;
    t.imag = -other.imag;
    r=(other.real).abs;
    i=(other.imag).abs;
    d=(r*r)+(i*i);
    ob.real = ((@real *other.real)-(imag*t.imag))/d;
    ob.imag=((@real*t.imag)+(imag*other.real))/d;
    return ob;
end

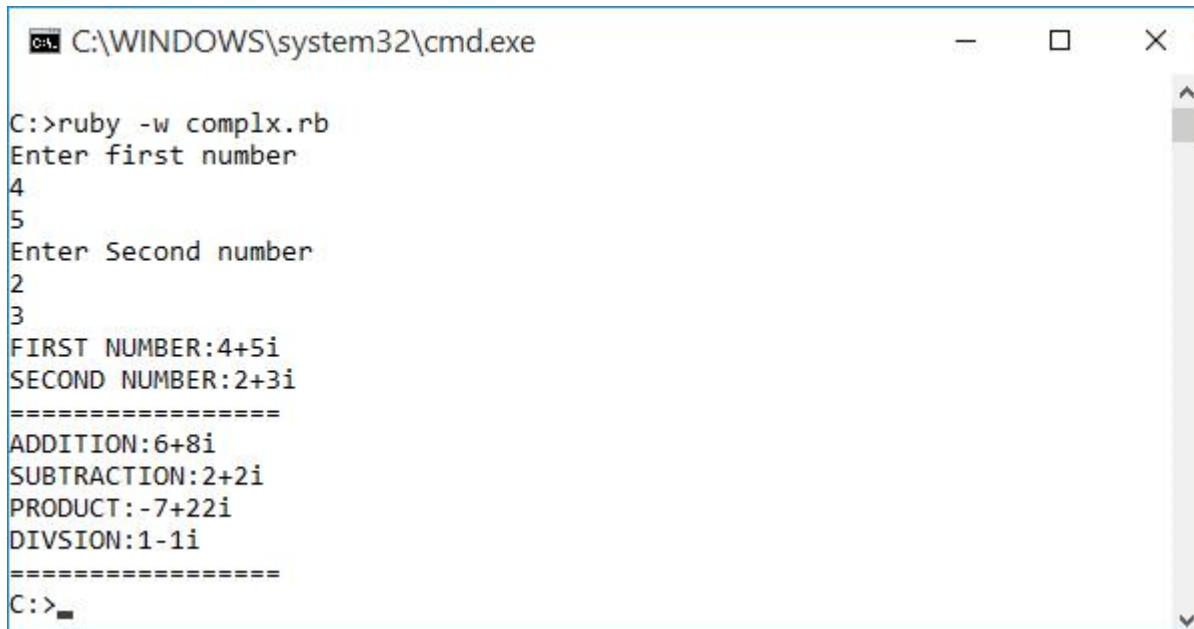
def disp
    if @imag<0
        puts "#{@real}#{@imag}i"
    else
        puts "#{@real}+#{@imag}i"
    end
end
end

t1 = Cmplx.new
t2 = Cmplx.new
t3 = Cmplx.new
t4 = Cmplx.new
t5 = Cmplx.new
t6 = Cmplx.new
puts "Enter first number"
t1.read
puts "Enter Second number"
t2.read
print "FIRST NUMBER:"
t1.disp
print "SECOND NUMBER:"

t2.disp
print "=====\\n";
print "ADDITION:"
t3 = t1.add(t2)
t3.disp
print "SUBTRACTION:"
t4 = t1.subtract(t2)
t4.disp

```

```
print "PRODUCT:"  
t5 = t1.multiply(t2)  
t5.disp  
print "DIVISION:"  
t6 = t1.divide(t2)  
t6.disp  
print "=====";
```

OUTPUT:

The screenshot shows a Windows Command Prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The command 'ruby -w complx.rb' is run, followed by user input for two complex numbers: '4+5i' and '2+3i'. The script then outputs the results of addition ('6+8i'), subtraction ('2+2i'), multiplication ('-7+22i'), and division ('1-1i').

```
C:\>ruby -w complx.rb  
Enter first number  
4  
5  
Enter Second number  
2  
3  
FIRST NUMBER:4+5i  
SECOND NUMBER:2+3i  
=====  
ADDITION:6+8i  
SUBTRACTION:2+2i  
PRODUCT:-7+22i  
DIVSION:1-1i  
=====  
C:\>
```

VIVA QUESTIONS:

1. Write a single line of Ruby code that prints the Fibonacci sequence of any length as an array.
2. Can you call a private method outside a Ruby class using its object?
3. What is the difference between extend and include in ruby?
4. How do you remove nil values in array using ruby?

WEEK 7:**15. Write a program which illustrates the use of associative arrays in Perl.**

AIM: To write a program that illustrates the use of associative arrays in Perl.

DESCRIPTION: A Hash is a collection of key-value pairs like this: "employee" => "salary". It is similar to an Array, except that indexing is done via arbitrary keys of any object type, not an integer index. The order in which you traverse a hash by either key or value may seem arbitrary and will generally not be in the insertion order. If you attempt to access a hash with a key that does not exist, the method will return *nil*.

ALGORITHM:

```
H = Hash["a" => 100, "b" => 200]
puts "#{H['a']}"
puts "#{H['b']}
```

PROGRAM:**Assoc.pl**

```
%ages = ('kiran'=>19,'vijay'=>21,'raju'=>20);

print"Original Array:\n";
print"=====|\n";
while (($key) = each %ages)
{
    print "$key is $ages{$key} years old\n";
}
$ages{'mayur'} = 24;
print"\nAfter adding element:\n";

print"=====|\n";
while(($key) = each %ages)
{
    print "$key is $ages{$key} years old\n";
}

delete( $ages{'vijay'});
print"\nAfter deleting element:\n";
print"=====|\n";

@all_keys = keys(%ages);
print "Keys are: @all_keys\n";
@all_values = values(%ages);
print "Values are: @all_values";
```

OUTPUT:

```
C:\>perl -w pAssoc.pl
Original Array:
=====
raju is 20 years old
vijay is 21 years old
kiran is 19 years old

After adding element:
=====
raju is 20 years old
mayur is 24 years old
vijay is 21 years old
kiran is 19 years old

After deleting element:
=====
Keys are: raju mayur kiran
Values are: 20 24 19
C:>_
```

VIVA QUESTIONS:

1. What are data types that perl supports?
2. What are scalar data types in perl?
3. What are Arrays in perl?
4. What are Hashes in perl?

16. write perl program takes set of file names along the command line and prints whether they are regular files or special files.

AIM: To write perl program takes set of file names along the command line and prints whether they are regular files or special files.

DESCRIPTION: Perl command line arguments stored in the special array called @ARGV . The array @ARGV contains the command-line arguments intended for the script. \$#ARGV is generally the number of arguments minus one, because \$ARGV[0] is the first argument, not the program's command name itself.

ALGORITHM:

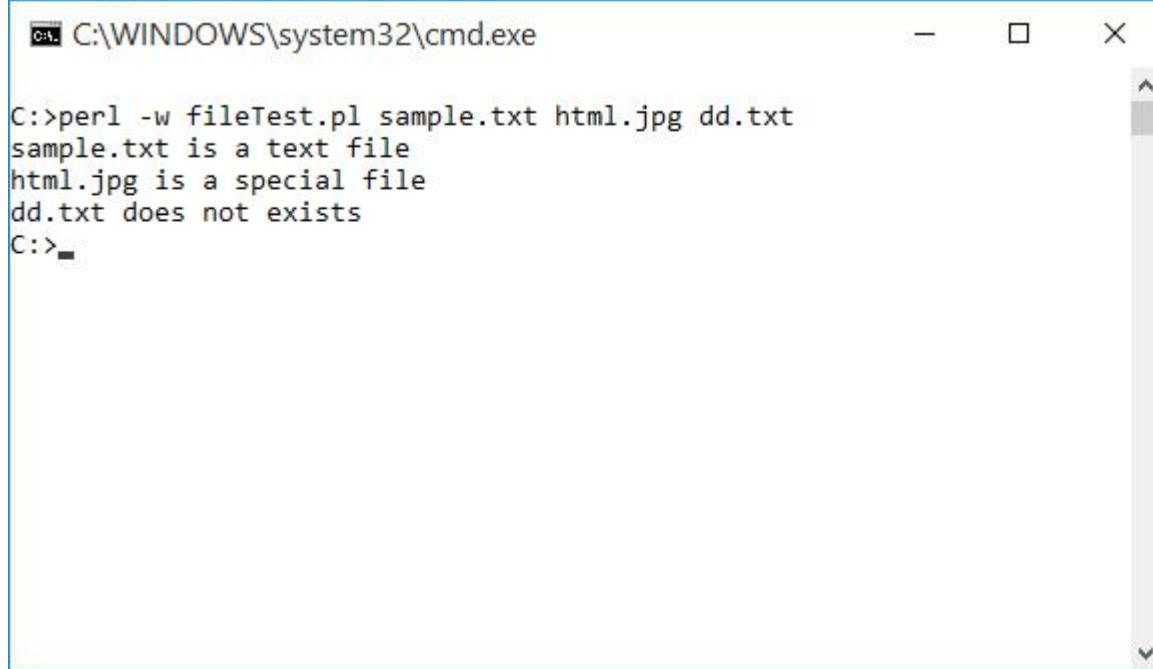
```
foreach(@ARGV) {
    print
    print((-f) ? " Regular File\n" : "Special File\n")
}
```

PROGRAM:**fileTest.pl**

```
$len = @ARGV;
for ($i=0;$i<$len;$i++)
{
    if(-e $ARGV[$i])

    {
        if(-T $ARGV[$i])
        {
            print "$ARGV[$i] is a text file\n";

        }
        else
        {
            print "$ARGV[$i] is a special file\n";
        }
    }
    else
    {
        print "$ARGV[$i] does not exists";
    }
}
```

OUTPUT:

```
C:\WINDOWS\system32\cmd.exe
C:>perl -w fileTest.pl sample.txt html.jpg dd.txt
sample.txt is a text file
html.jpg is a special file
dd.txt does not exists
C:>
```

VIVA QUESTIONS:

1. What is command line argument?
2. What does @ARGV represents?
3. What does \$#ARGV represents?
4. What does \$ARGV[0] represents?

17. Write a perl program to implement UNIX `passwd' program

AIM: To write a perl program to implement UNIX `passwd' program

DESCRIPTION:

ALGORITHM:

```
$pipe="data";
$username="venkat";
$password="venkat";
open my $pipe, '|chpasswd' or die "can't open pipe: $!";
print {$pipe} "$username:$password";
close $pipe
```

PROGRAM:**unixPw.pl**

```
my $salt="";
my $encrypted="";
my $password="";
my $use = 'Usage: Please provide password for encrypt';
my @saltchars=('.', '/', 0..9, 'A'..'Z', 'a'..'z');

my $args=@ARGV;
if ( $args < 1 || $args > 2 )
{
    print "$use\n";
    exit;
}
$password=$ARGV[0];
if( $args == 1 )

{
    $salt = join(",@saltchars[rand(64),rand(64)]");
}
else

{
    $salt=$ARGV[1];
```

```
}
```

```
$encrypted=crypt($password,$salt);
```

```
print "$password --> $encrypted\n";
```

OUTPUT:

The screenshot shows a Windows Command Prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The window contains the following text:

```
C:>perl -w unixPw.pl sacet123 9i
sacet123 --> 9i32G9/C74z/g

C:>perl -w unixPw.pl sacet123
sacet123 --> 29R2r5QMmpNLY

C:>perl -w unixPw.pl
Usage: Please provide password for encrypt

C:>
```

VIVA QUESTIONS:

1. What does passwd command do?
2. What is interpolative context?
3. What is V-Strings?
4. How will you define and call a subroutine in perl?

18.An example perl program to connect to a MySQL database table and executing simple commands.

AIM: To write an example perl program to connect to a MySQL database table and executing simple commands.

DESCRIPTION:

1. First, you need to tell DBI where to find the MySQL database server. This information is called data source name or DSN. The data source name specifies which driver to use, what database that you want to connect to. Perl requires the data source name to begin with dbi: and the name of the driver, in this case, it is mysql , followed by another colon : e.g., dbi:mysql: , and then the database name e.g., dbi:mysql:perlmysqldb .
2. Second, you need to provide the username and password of the MySQL account that you connect to the database.

3. Third, the optional connection attributes specify the way DBI handles exceptions that may occur when it connects to the MySQL database.

ALGORITHM:

For connecting to DB

```
$dbh = DBI->connect($dsn,$username,$password,\%attr);
```

LIST DBs

```
@databases = $connect->listdbs;
foreach $database (@databases) {
    print "$database<br />";
}
```

Inserting Data to DB

```
my $sth = $dbh->prepare("INSERT INTO TEST_TABLE (FIRST_NAME, LAST_NAME,
SEX, AGE, INCOME ) values (?,?,?,?,?)");
$sth->execute($first_name,$last_name,$sex, $age, $income)
```

Fetch Data from DB

```
while (my @row = $sth->fetchrow_array()) {
    my ($first_name, $last_name ) = @row;
    print "First Name = $first_name, Last Name = $last_name\n";
}
```

PROGRAM:

Note: Create the following table in MySQL and insert some data in to the table.

students table

```
create table students
(
    roll varchar(3) not null,
    name varchar(3) not null,
    branch varchar(3) not null
);
```

dbConn.pl

```
use DBI;
#definition of variables
$db="test";
$host="localhost";
$user="root";
$password="";
```

```
#connect to MySQL database
```

```

my $dbh = DBI->connect ("DBI:mysql:database=$db:host=$host",$user,$password) or
die "Can't connect to database: $DBI::errstr\n";

#prepare the query
my $sth = $dbh->prepare( "SELECT *FROM students");

#execute the query
$sth->execute();

## Retrieve the results of a row of data and print
print "\tQuery results:\n=====\n";
print("ROLL\tNAME\tBRANCH\n=====\n");
while (my @row = $sth->fetchrow_array()) {
    print "@row\n";

}

warn "Problem in retrieving results",sth->errstr( ), "\n" if $sth->err();
exit;

```

OUTPUT:

| ROLL | NAME | BRANCH |
|------|--------|--------|
| 554 | SUJITH | CSE |
| 514 | KIRAN | CSE |
| 503 | RAJU | CSE |
| 533 | KAMAL | CSE |

VIVA QUESTIONS:

1. How Do You Connect To Database In Perl?
2. Give syntax for deleting a record from DB using PERL.
3. Give syntax to update data in DB using PERL.
4. Give syntax for alter fields in table using PERL.

WEEK 8:

19. Example PHP program for contactus page.

AIM: To write an example PHP program for contactus page.

DESCRIPTION: Contactus page contains fields whom to contact, who is contacting, reason for contact etc.

Sample Contactus Page

Name

Email

Company Name

Message

Send Message

ALGORITHM:

1. Create text fields for Name, email, Company Name, Message
2. Create a Send Message button
3. Give Data and on submission data need to be stored in DB using PHP

PROGRAM:contacts.php

```
<?php

if(isset($_REQUEST['vname']))
{
    $vname=$_REQUEST['vname'];
    $email=$_REQUEST['email'];

    $message=$_REQUEST['message'];
    $from="From: $vname<$email>\r\nReturn-path: $email";
    $subject="Message sent using your contact form";

    $m = mail("youremail@yoursite.com", $subject,
    $message, $from); if($m)
        echo "<h1>Email sent!";
    else

```

```
echo "Send Failed:".mysql_error();  
  
echo "&nbsp;&nbsp;&nbsp;<a href='contacts.php'>[Back to Contacts]</a>";  
}  
else { ?>  
<html>  
  
<head>  
    <title>Contacts Page</title>  
</head>  
<body>  
  
<font face="Trebuchet ms" size="6">  
<table border="1">  
    <tr><th>Contact Us</th></tr>  
    <tr><td>  
  
        <form action="contacts.php" method="POST"  
            enctype="multipart/form-data"> Your name<br/>  
  
        <input type="text" name="vname" value=""  
            size="30"/><br> Your email<br>  
  
        <input type="text" name="email" value=""  
            size="30"/><br> Your message<br>  
  
        <textarea name="message" rows="7"  
            cols="30"></textarea><br> <input type="submit"  
            value="Send email"/>  
    </form>  
  
</td></tr>  
</table>  
</body>  
</html>  
  
<?php } ?>
```

OUTPUT:

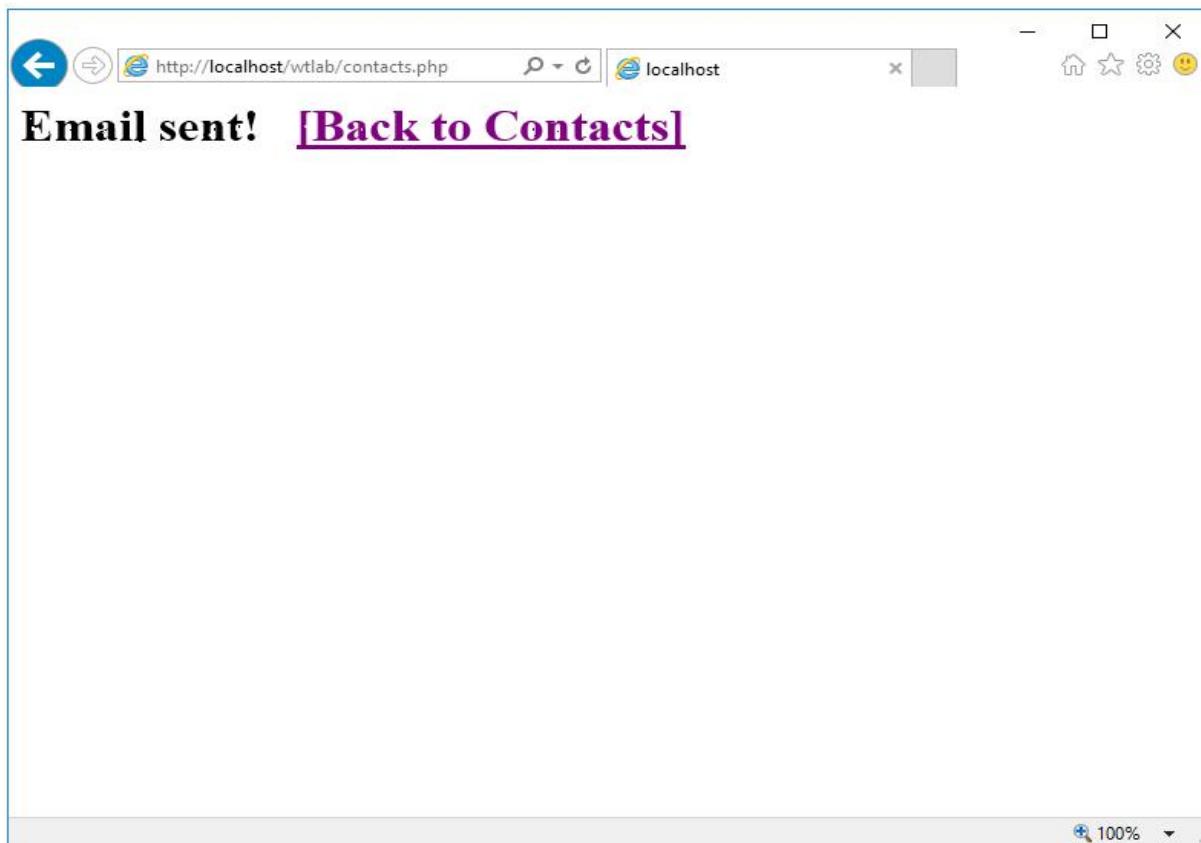
Contact Us

Your name
vijayanand

Your email
vijaych@gmail.com

Your message
Hi..
This test mail

Send email



VIVA QUESTIONS:

- 1. How Do You Connect To Database in PHP?**
- 2. Give syntax for fetching records from DB using PHP.**
- 3. Give syntax to update data in DB using PHP.**
- 4. Give syntax for alter fields in table using PHP.**

20. User Authentication: Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a PHP for doing the following.

- 1. Create a Cookie and add these four user id's and passwords to this Cookie.**
- 2. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords) available in the cookies. If he is a valid user (i.e., user-name and password match) you should welcome him by name (user-name) else you should display 'You are not an authenticated user'. Use init-parameters to do this.**

AIM: To write a program for User Authentication.

DESCRIPTION: A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values. A session is a way to store information (in variables) to be used across multiple pages. Unlike a cookie, the information is not stored on the user's computer.

ALGORITHM:

1. Create variables for 4 usernames & passwords
2. Initialize usernames & passwords
3. Declare cookie variable
4. Set Cookie names & values as usernames & passwords
5. Read username and password from login form
6. Compare username & passwords
 - 6.1 If Username & password match → Display welcome user
 - 6.2 else Display " You are not authenticated user"

PROGRAM:**login.html**

```
<html>
<head>

<title>Login Page</title>
```

```

</head>
<body>
<center>

<h3> Login Page</h3>

<form name="f1" action="setcookie.php"
method="post"> <table frame="border">

<tr>
<td>User</td>

<td><input type="text" name="user" size="20"
value="" /></td> </tr>
<tr>
<td>Password</td>
<td><input type="password" name="pwd" size="20"
value="" /></td> </tr>

<tr>
<td align="center" colspan="2"><input type="submit"
value="Sign in" /> <input type="reset" /></td>
</tr>
</table>
</form>
</body></center>
</html>

```

setcookie.php

```

<html>
<head>
    <title>Set Cookie</title>

</head>
<body>
    <?php
    $user = $_POST['user'];

    $pwd = $_POST['pwd'];
    setcookie("user",$user,time()+3600*2);
    setcookie("pwd",$pwd,time()+3600*2);
    echo "<h3>The Cookie Added...</h3>";

    ?>
</body>
</html>

```

getcookie.php

```

<html>
<head>

```

```

<title>Get Cookie</title>
</head>
<body>
<?php

    $user = $_COOKIE['user'];
    $pwd = $_COOKIE['pwd'];
    if($user=="user1")

    {
        if($pwd == "pwd1")
            echo "<h2>Welcome User1 </h2>";
        else

            echo "<h2>You are not an authenticated user.</h2>";
    }
    elseif($user=="user2")
    {

        if($pwd == "pwd2")
            echo "<h2>Welcome User2</h2>";
        else
            echo "<h2>You are not an authenticated user.</h2>";

    }
    elseif($user=="user3")
    {
        if($pwd == "pwd3")

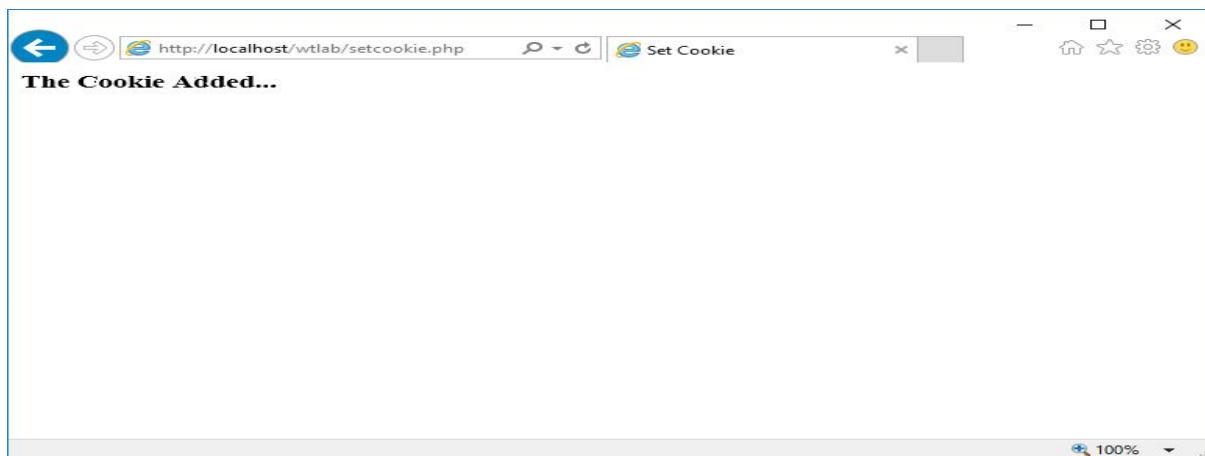
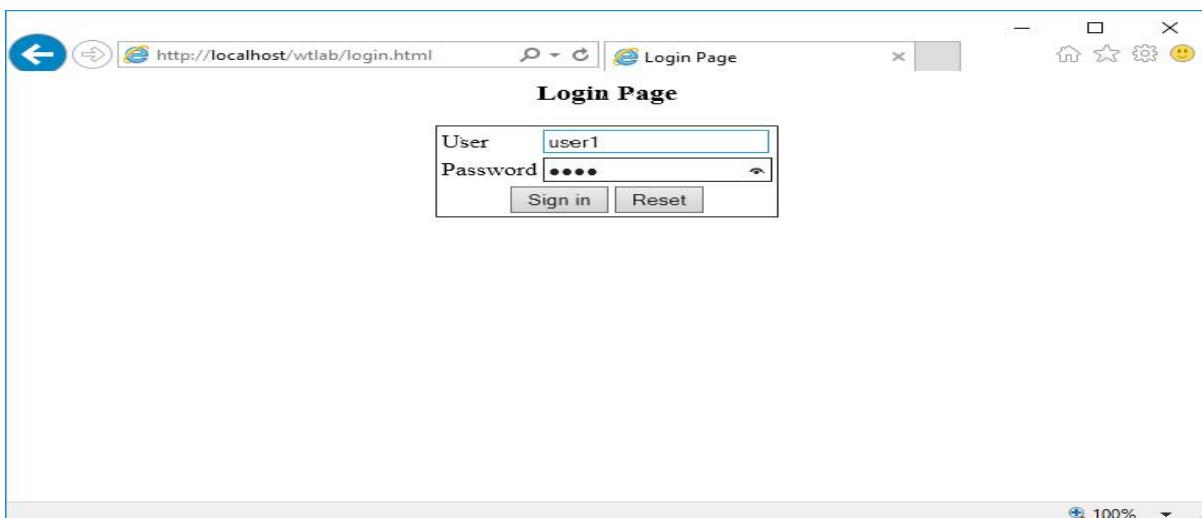
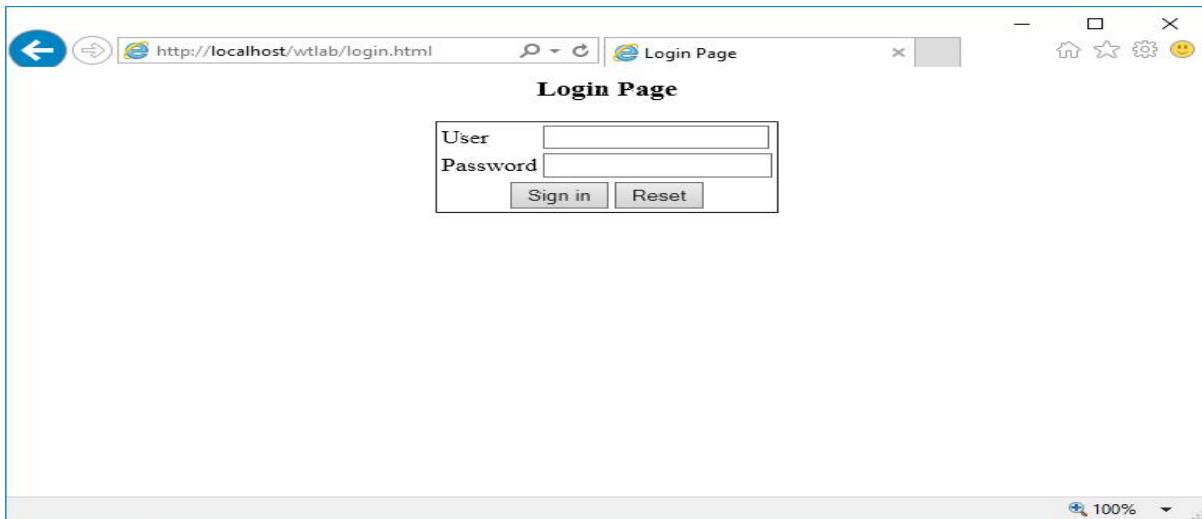
            echo "<h2>Welcome User3</h2>";
        else
            echo "<h2>You are not an authenticated user.</h2>";
    }

    elseif($user=="user4")
    {
        if($pwd == "pwd4")
            echo "<h2>Welcome User4</h2>";
        else

            echo "<h2>You are not an authenticated user.</h2>";
    }
    else
        echo "<h2>Invalid Username/Password</h2>";

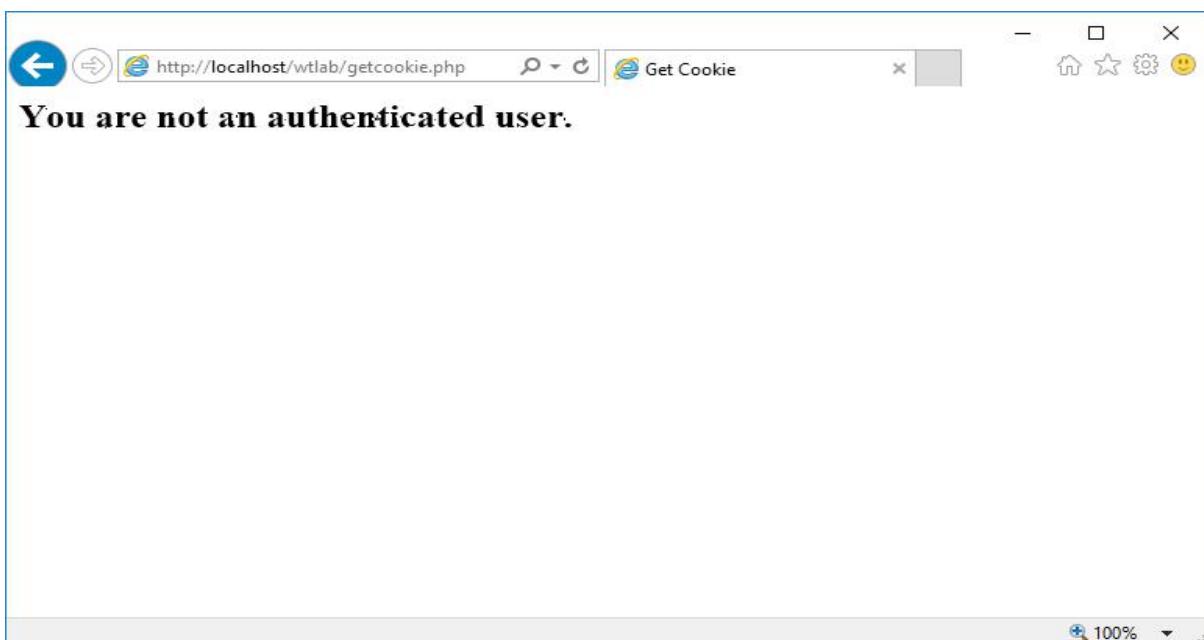
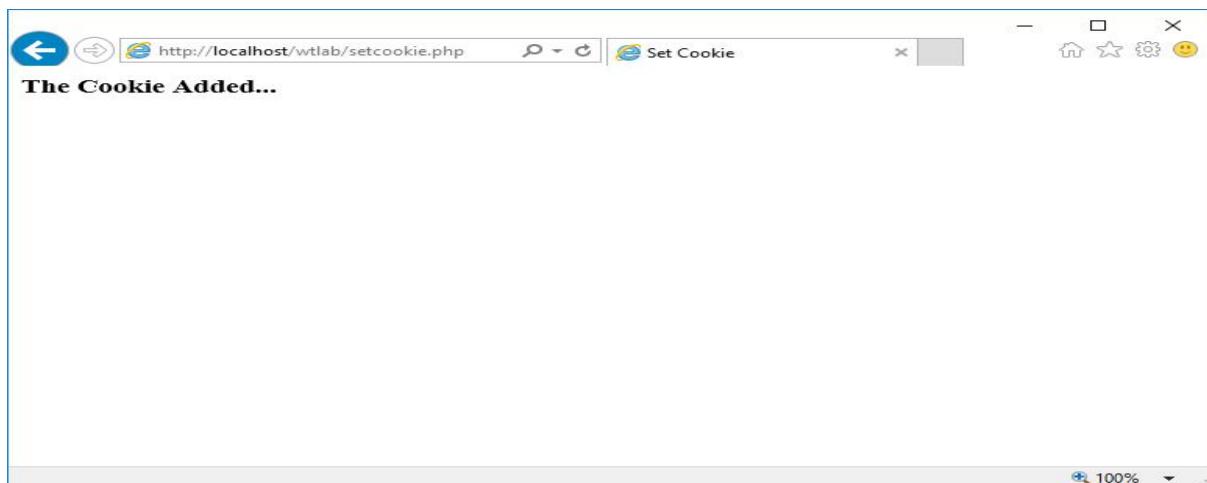
    ?>
</body>
</html>

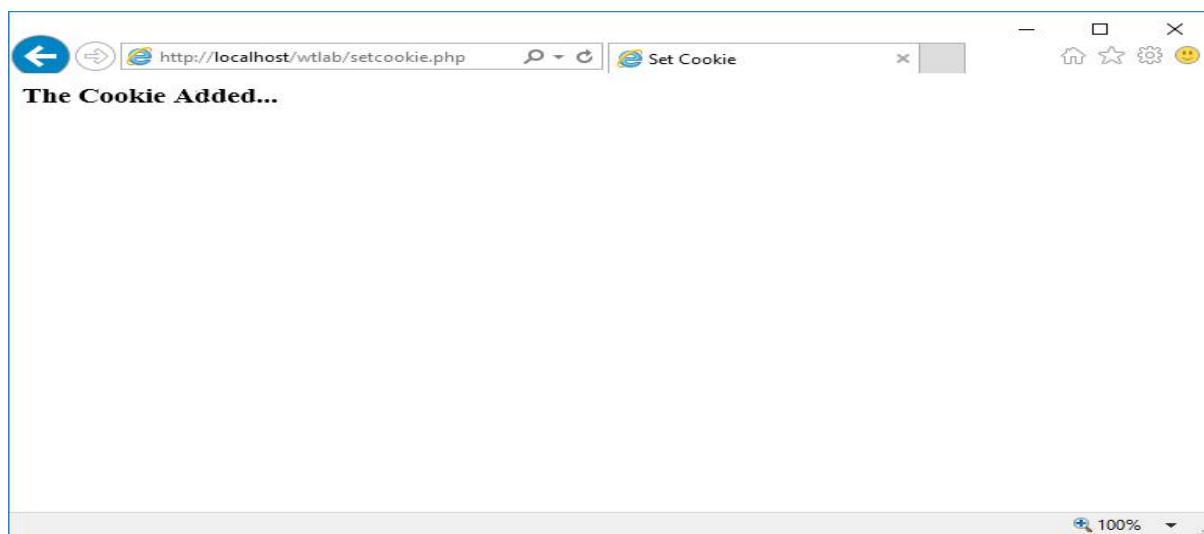
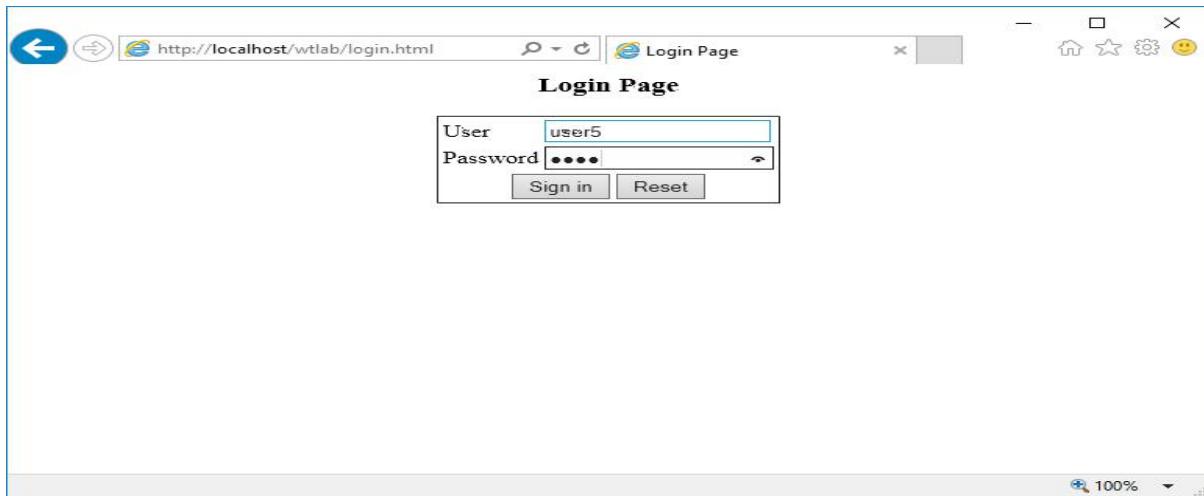
```

OUTPUT 1 for Valid User:

OUTPUT 2 for Invalid User:

A screenshot of a web browser window titled "Login Page". The address bar shows the URL "http://localhost/wtlab/login.html". The page content is a form with two fields: "User" containing "user4" and "Password" containing "*****". Below the form are two buttons: "Sign in" and "Reset".



OUTPUT 3 for Non-existing user:

b) AIM: Use init-parameters to do the above task. Store the user-names and passwords in the “users.ini” file and access them in php page by using the “parse_ini_file()” method.

mylogin.html

```
<html>
<head>
    <title>Login Page</title>
</head>
<body>
    <center>
```

```

<h3> Login Page</h3>

<form name="f1" action="initparam.php" method="post">
<table border="1">
<tr>
<td>User</td>

<td><input type="text" name="user" size="20"
value="" /></td> </tr>

<tr>
<td>Password</td>

<td><input type="password" name="pwd" size="20"
value="" /></td> </tr>

<tr>
<td align="center" colspan="2"><input type="submit"
value="Sign in" /> <input type="reset" /></td>
</tr>
</table>
</form>
</body></center>
</html>

```

users.ini

;This is a users .ini file

```

[first_section]
u1=user1
p1=pwd1

```

initparam.php

```

<html>
<head>

<title>Init Parameters</title>
</head>
<body>

<?php

$user = $_POST['user'];
$pwd = $_POST['pwd'];
$temp = parse_ini_file("users.ini");
foreach($temp as $value)

```

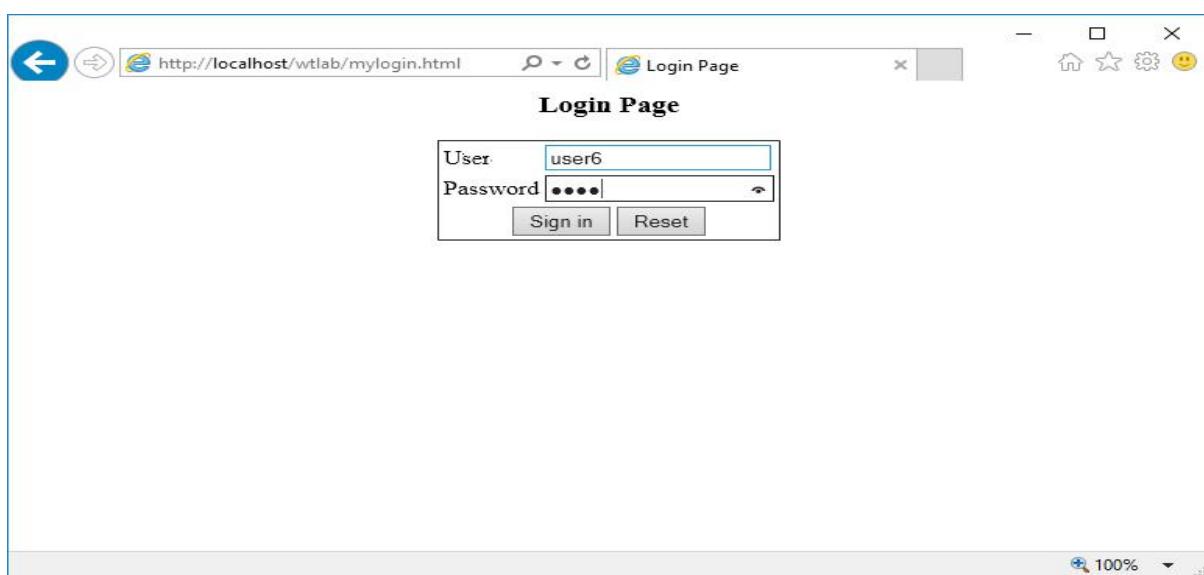
```

{
    $users[] = $value;
}
if($user==$users[$i])
{
    if($pwd==$users[$i+1])
        echo "<h3>Welcome $users[$i]</h3>";
    else

        echo "<h3>You are not an authenticated user</h3>";
}
else
    echo "<h3>Invalid Username/Password</h3>";

}
?>
</body>
</html>

```

OUTPUT for Valid user:**OUTPUT for Invalid user:**

WEEK 9:**21.Example PHP program for registering users of a website and login.**

AIM: To write an example PHP program for registering users of a website and login.

DESCRIPTION: Create a registration page to register users for a website and store their details in DB. Create a Login page to check either the user is authorized or not.

ALGORITHM:

1. Display Registration & Login
2. If user selects Registration
 - 2.1 Display Registration form
 - 2.2 Read the values from registration form
 - 2.3 Store the values in cookies
3. If user selects login
 - 3.1 Read username & Password
 - 3.2 Compare usernames & password using cookies.

PROGRAM:**registration1 table:**

```
CREATE TABLE registration1
(
    uname varchar(25) NOT NULL,
    pass varchar(25) NOT NULL,
    email varchar(25) NOT NULL,
    phno bigint(10) NOT NULL,
    gender varchar(6) NOT NULL,
    dob char(10) NOT NULL,
    languages varchar(25) NOT NULL,
    address varchar(75) NOT NULL,
    UNIQUE KEY (uname)
);
```

A) Registration: Add the following code to registration.html in <head> section:

```
<script language="javascript">
    function validate()
```

```

{
    var nam = document.f1.uname.value;
    if(nam=="")
    {
        alert("Please enter name");
        document.f1.uname.focus();
        return false;
    }

    var pwd = document.f1.pass.value;
    if(pwd=="")
    {
        alert("Please enter Password");
        document.f1.pass.focus();
        return false;
    }

    var email = document.f1.email.value;
    if(email=="")
    {
        alert("Please enter youe email");

        document.f1.email.focus();
        return false;
    }

    var phno = document.f1.phone.value;
    len=phno.length
    if(phno=="" || len != 10)
    {
        alert("Please enter phno or should be strictly 10 digits");
        document.f1.phone.focus();
        return false;
    }

    var genders = document.getElementsByName("gen");
    var selectedGender="";
    for(var i = 0; i < genders.length; i++)
    {
        if(genders[i].checked == true)
            selectedGender = genders[i].value;
    }
    if(selectedGender=="")
    {
        alert("Please select your gender");
        document.f1.gen.focus();
        return false;
    }
}

```

```

var day = document.f1.day.value;
var month = document.f1.month.value;

var year = document.f1.year.value;
if(day=="day" || month=="month" || year=="year")
{
    alert("Please select your DOB");

    document.f1.day.focus();
    return false;
}

var languages = document.getElementsByName("lang[]");
var selected = new Array();
var j=0;
for(var i = 0; i < languages.length; i++)
{
    if(languages[i].checked == true)
        selected[j++] = languages[i].value;
}
var len = selected.length;
if(len==0)
{
    alert("Please Choose atleast one language");

    document.f1.lang.focus();
    return false;
}

var addr = document.f1.addr.value;
if(addr=="")
{
    alert("Please enter your address");
    document.f1.addr.focus();
    return false;
}

}

}

</script>
```

Then, Modify the <form> tag attributes:

```
<form name="f1" action="insert.php" method="post" onsubmit="javascript:return validate()">
```

insert.php

```
<?php
$conn = mysql_connect("localhost","root","");
if (!$conn)
die("Failed to connect:".mysql_error());
mysql_select_db("test",$conn);
if (!mysql_select_db("test"))
die("No Database existing:".mysql_error());
```

```

if(isset($_POST['uname']))
{
    $uname=$_POST['uname'];

    $pass=$_POST['pass'];
    $email=$_POST['email'];
    $phno=(float)$_POST['phone'];
    $gender=$_POST['gen'];

    $day=(int)$_POST['day'];
    $month=(int)$_POST['month'];
    $year=(int)$_POST['year'];
    $dob = "$day-$month-$year";

    $str=$_POST['lang'];
    $lang="";
    foreach ($str as $val)

    {
        $lang = $lang."
        ".$val;
    }
    $text=$_POST['addr'];
    $addr = str_replace("\n","<br>",$text);

$query = "INSERT INTO registration1 VALUES ('$uname', '$pass', '$email',
'$phno', '$gender', '$dob', '$lang', '$addr')";

$res =
mysql_query($query);
if($res)
{
    $script = "<script>alert('User Registration Successful');";
    $script .= "document.writeln('<center><h4><u>User Details
</u></h1>');";
    $script .= "document.writeln('Your username is:<b> $uname
</b><br/>');";
    $script .= "document.writeln('Your password is:<b> $pass
</b></center>');";
    $script .= "document.writeln('<h4 align=right><a
href=catalogue.html>[View Books]</a> </h4>');</script>";
    echo $script;
}
else
    echo "<script>alert('User Registration Failed'); window.location =
'registration.html'; </script>";
}

?>

```

OUTPUT:

The screenshot shows a web application interface. At the top, there's a header with a logo, a back button, a forward button, and a search bar containing 'http://localhost/wtlab/'. The title bar says 'Home page'. Below the header is a navigation menu with tabs: HOME, LOGIN, REGISTRATION (which is currently active), CATALOGUE, and CART. To the left of the main content area, there are three vertical buttons labeled 'MCA', 'MBA', and 'BCA'. The main content area is titled 'REGISTRATION PAGE'. It contains several input fields: NAME (with value 'admin'), PASSWORD (with value '*****'), E-MAIL ID (with value 'admin@gmail.com'), and PHONE NUMBER (with value '9876543210'). There are also gender selection buttons ('MALE' is selected), date of birth dropdowns (set to '15 JUNE 1986'), and language checkboxes (ENGLISH, TELUGU are checked, while HINDI and TAMIL are not). An ADDRESS field contains the text 'vetapalem', 'chirala', and 'prakasam'. At the bottom of the form are 'SUBMIT' and 'RESET' buttons.

b.) Login Verification**login.html**

```
<html>
    <head>
        <title> Sign-in Page </title>
        <script language="javascript">
            function validate()
            {
                var nam = document.f1.uname.value;
                if(nam=="")
                {
                    alert("Please enter name");
                    document.f1.uname.focus;
                    return false;
                }
                var pwd = document.f1.pass.value;
                if(pwd=="")
                {
                    alert("Please enter Password");
                    document.f1.pass.focus;
                    return false;
                }
            }
        </script>
    </head>
    <body>
        <form name="f1">
            <table border="1">
                <tr>
                    <td>User Name:</td>
                    <td><input type="text" name="uname"/></td>
                </tr>
                <tr>
                    <td>Password:</td>
                    <td><input type="password" name="pass"/></td>
                </tr>
                <tr>
                    <td colspan="2" style="text-align: center; padding-top: 10px;">
                        <input type="submit" value="Login" /> <input type="button" value="Cancel" />
                    </td>
                </tr>
            </table>
        </form>
    </body>
</html>
```

```

        }
    </script>
</head>
<body>
<br/><br/><br/>
<center>
<form name="f1" action="validation.php" method="post"
onsubmit="javascript:return
validate()">
<table border="3" cellpadding="0" cellspacing="0">

<tr>
<td>
<table cellspacing="10">
<tr>
<td colspan="2" align="center"><h2><u>Login Page<u></h2></td>

</tr>
<tr>
<td> User Name</td>
<td><input type="text" name="uname" size="15"></td>

</tr>
<tr>
<td> Password</td>

<td><input type="password" name="pass"
size="15"></td> </tr>
<tr>

<td align="center"><input type="submit"
value="Sign in"></td> <td align="center"><input
type="reset" value="Clear"></td>
</tr>
</table>
</td>
</tr>

</table>
</form>
</center>
</body>
</html>

```

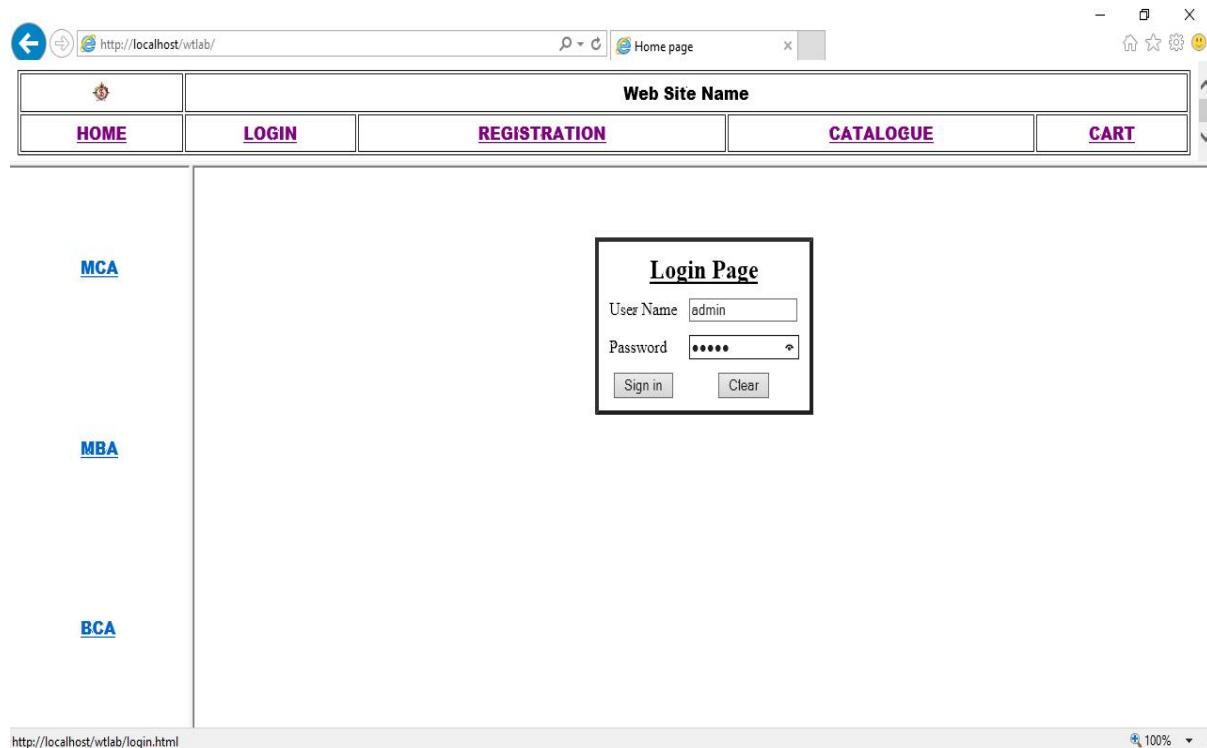
validation.php

<?php

```

$conn = mysql_connect("localhost","root","");
if(mysql_error())
{
    die("Failed to Connect: ".mysql_error());
}
mysql_select_db("test",$conn) or die("No Database existing:".mysql_error());
if(isset($_POST['uname']))
{
    $uname=$_POST['uname'];
    $pass=$_POST['pass'];
    $res = mysql_query("select * from registration1 where uname='".$uname' and
    pass ='$pass'"); $count=mysql_num_rows($res);
    if($count)
        echo "<script>alert('Welcome
        $uname');window.location='catalogue.html';</script>";
    else
        echo "<br/><br/><br/><h2 align='center'>Invalid Username
        /Password/Try again!!!!</h2>";
}
?>

```

OUTPUT:**VIVA QUESTIONS:**

1. How be the result set of Mysql handled in PHP?
2. Differentiate between mysqli_fetch_object() and mysqli_fetch_array()?
3. Which function gives us the number of affected entries by a query?
4. How is it possible to know the number of rows returned in the result set?

22. Install a database (Mysql or Oracle).

Create a table which should contain at least the following fields: name, password, email-id, phone number (these should hold the data from the registration form). Write a PHP program to connect to that database and extract data from the tables and display them. Experiment with various SQL queries.

Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page (week2).

AIM: To install a DB, create a table and insert the details of users.

DESCRIPTION: MySQL is the most popular database system used with PHP. Create a table in MYSQL, insert data into DB and retrieve data from DB.

ALGORITHM:

1. Create connection to Database using server name, username & password
2. Read the values from registration form
3. Create insert query with the registration values
4. Execute the query
5. Create a select query which selects registration details
6. Execute the query
7. Display the values

PROGRAM:

registration table:

```
CREATE TABLE registration
(
    uname varchar(25) NOT NULL,
    pass varchar(25) NOT NULL,
    email varchar(25) NOT NULL,
    phno bigint(10) NOT NULL,
    UNIQUE KEY (uname)
)
```

registration.html

```
<html>
    <head>
        <title> User Registration Page</title>
        <script language="javascript">
            function validate()
            {
                var nam = document.f1.uname.value;
                if(nam=="")
```

```

{
    alert("Please enter name");
    document.f1.uname.focus;
    return false;
}
var pwd = document.f1.pass.value;
if(pwd=="")
{
    alert("Please enter Password");
    document.f1.pass.focus;
    return false;
}
var email = document.f1.email.value;
if(email=="")
{
    alert("Please enter youe email");
    document.f1.email.focus;
    return false;
}
var phno = document.f1.phone.value;
len=phno.length
if(phno=="" || len != 10)
{
    alert("Please enter phno or should be strictly 10 digits");
    document.f1.phone.focus;
    return false;
}
}
</script>
</head>
<body>
<br/><br/><br/>
<center>
<form name="f1" action="insertData.php" method="post" onsubmit=
"javascript:return validate()">
<table border="3" cellpadding="0" cellspacing="0">
<tr>
<td>

<table cellspacing="10">
<tr>
<td colspan="2" align="center"><h2><u>User Registration
Form</u></h2></td>
</tr>
<tr>
<td> User Name</td>
<td><input type="text" name="uname" size="50"></td>
</tr>

```

```

<tr>
    <td> Password</td>
    <td><input type="password" name="pass" size="50"> </td> </tr>

<tr>
    <td> E-mail</td>
    <td><input type="text" name="email" size="50"></td>
</tr>
<tr>

    <td> Phone</td>
    <td><input type="text" name="phone" size="15"></td>
</tr>
<tr>

    <td colspan="2" align="center"><input type="submit" value="submit">
    </td>
</tr>
</table>
</td>

</tr>
</table>
</form>
</center>

</body>
</html>

```

insertData.php

```

<?php
$conn = mysql_connect("localhost","root","");
if($conn)
    echo "Connected to database!!!";
else
    echo "Failed to Connect:".mysql_error();
mysql_select_db("test",$conn) or die("No Database existing:".mysql_error());

if(isset($_REQUEST['uname']))
{
    $uname=$_REQUEST['uname'];
    $pass=$_REQUEST['pass'];
    $email=$_REQUEST['email'];
    $phno=(float)$_REQUEST['phone'];

```

```

$query = "INSERT INTO registration
VALUES('$uname','$pass','$email','$phno')";
mysql_query($query);
$result = mysql_query("select * from registration");
?>
<html>
<body>

<br/><br/><br/>

<p align="right"><a
href="registration.html">[Registration Page]</a></p>
<center>

<font face="verdana" size="4">

<table border="1" cellpadding="0" cellspacing="0">
<tr>
    <th colspan="4" align="center">User List</td>
</tr>

<tr>
    <th> S.No.</th>
    <th> User Name</th>
    <th> Email</th>

    <th>Phone</th>
</tr>
<?php $num=1; while($row = mysql_fetch_array($result))
?> <tr>

    <td><?php echo $num++; ?>
    </td> <td><?php echo
    $row['uname']; ?> </td>
    <td><?php echo $row['email'];
    ?> </td>
    <td><?php echo $row['phno']; ?> </td>
</tr>
<?php }?>
</table>
</center>

</body>
</html>
<?php } ?>

```

OUTPUT:

A screenshot of a web browser window titled "User Registration Page". The address bar shows the URL "http://localhost/wtlab/registration.html". The main content area displays a form titled "User Registration Form" with four input fields: "User Name", "Password", "E-mail", and "Phone", each with its own text input box. Below the input fields is a "submit" button.

A screenshot of a web browser window titled "User Registration Page". The address bar shows the URL "http://localhost/wtlab/registration.html". The main content area displays a form titled "User Registration Form" with four input fields: "User Name" (value "user1"), "Password" (value "****"), "E-mail" (value "user1@gmail.com"), and "Phone" (value "9876543210"). Below the input fields is a "submit" button.

VIVA QUESTIONS:

1. Give syntax for creating a table using PHP in MySQL DB.
2. Give syntax for inserting data into a table using PHP in MySQL DB.
3. Give syntax for fetching data from a table using PHP in MySQL DB.
4. Give syntax for deleting data from a table using PHP in MySQL DB.

WEEK 10:

23. Write a PHP which does the following job: Insert the details of the 3 or 4 users who register with the web site (week9) by using registration form. Authenticate the user when he submits the login form using the user name and password from the database (similar to week8 instead of cookies).

AIM: To write a PHP which insert the details of the 3 or 4 users who register with the web site by using registration form and authenticate the user.

DESCRIPTION: Use registration form which already created in previous sessions. Register details of 3 to 4 users using this form. Later use login form to authenticate the users.

ALGORITHM:

1. Create connection to Database using serve name, username & password
2. Display login and Registration to user
3. If user selects registration Display Registration form
4. Read the values from registration form
5. Create insert query with the registration values
6. Execute the query
7. Repeat steps 3-6 for 3 or 4 times
8. If user selects login Display Login page
9. Read username and password from login page into PHP variables
10. Select usernames and password form Database
11. Match usernames & passwords
 - 11.1 If Username & password match Display "Login successful"
 - 11.2 else Display "Login fail"

PROGRAM:**login.html**

```
<html>
  <head>
    <title> Sign-in Page </title>

    <script language="javascript">
      function validate()
      {
        var nam = document.f1.uname.value;
        if(nam=="")
        {
          alert("Please enter name");
        }
      }
    </script>
  </head>
  <body>
    <form name="f1">
      <table border="1">
        <tr>
          <td>User Name</td>
          <td><input type="text" name="uname" /></td>
        </tr>
        <tr>
          <td>Password</td>
          <td><input type="password" name="pass" /></td>
        </tr>
        <tr>
          <td colspan="2" style="text-align: center;">
            <input type="button" value="Submit" onclick="validate()"/>
            <input type="button" value="Cancel" />
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```

        document.f1.uname.focus;
        return false;
    }
    var pwd = document.f1.pass.value;
    if(pwd=="")
    {
        alert("Please enter Password");
        document.f1.pass.focus;
        return false;
    }
}
</script>
</head>
<body>
<br/><br/><br/>
<center>
<form name="f1" action="validation.php" method="post" onsubmit
="javascript:return validate()">
<table border="3" cellpadding="0" cellspacing="0">
<tr>
<td>
<table cellspacing="10">

<tr>
    <td colspan="2" align="center"><h2><u>Login Page<u></h2></td>
</tr>
<tr>

    <td> User Name</td>
    <td><input type="text" name="uname" size="15"></td>
</tr>
<tr>

    <td> Password</td>

    <td><input type="password" name="pass" size="15"> </td> </tr>
<tr>

    <td colspan="2" align="center"><input type="submit" value="submit">
    </td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</center>
</body>
</html>

```

validation.php

```

<?php
$conn = mysql_connect("localhost","root","");
if($conn)

    echo "<h4>Connected to Database.....!!";
else
    echo "Failed to Connect:".mysql_error()."</h4>";
mysql_select_db("test",$conn) or die("No Database existing:".mysql_error());
if(isset($_POST['uname']))
{
    $uname=$_POST['uname'];
    $pass=$_POST['pass'];

    echo "<br/><br/><br/><p align='right'><a href='login.html'>[Login
Page]</a></p>";
$res = mysql_query("select * from registration where uname='".$uname' and
pass ='$pass'");
$count=mysql_num_rows($res);
if($count)

    echo "<br/><br/><br/><h2 align='center'>Welcome
$uname!!!!</h2>";
else
    echo "<br/><br/><br/><h2 align='center'>Invalid
Username/Password/Try again!!!!</h2>";

}
?>

```

OUTPUT:



VIVA QUESTIONS:

1. How can we create a data base using PHP and MYSQL?
2. How can execute a PHP script using command line?
3. How can register a variable into a session?
4. Explain command line arguments?

24. Create tables in the database which contain the details of items (books in our case like Book name, Price, Quantity, Amount) of each category. Modify your catalogue page (week 2) in such a way that you should connect to the database and extract data from the tables and display them in the catalogue page using PHP

AIM : To create tables in the database, modify catalogue page, extract data and display.

DESCRIPTION: Using create command Create table in the DB with name books with fields like Book name, Price, Quantity, Amount of each category. Later using update command modify catalogue page. Using select command, extract data from tables and display it.

ALGORITHM:

1. Create connection to Database using server name, username & password
2. Using create command Create table in the DB with name books with fields like Book name, Price, Quantity, Amount of each category.
3. Using update command modify catalogue page.
4. Using select command, extract data from tables and display it.

PROGRAM:

Catalogue table

```
CREATE TABLE catalogue
(
```

```

        bname varchar(50) NOT NULL,
        auth varchar(25) NOT NULL,
        publ varchar(25) NOT NULL,
        isbn varchar(20) NOT NULL,
        edi varchar(10) NOT NULL,
        cost int(5) NOT NULL,
        qty int(3) NOT NULL,
        UNIQUE KEY(bname)
    )

```

bkEntryForm.html

```

<html>
    <head>
        <title> Catalogue Page</title>
        <script language="javascript">
            function validate()
            {
                var nam = document.f1.bname.value;
                if(nam=="")
                {
                    alert("Please enter Book name");
                    document.f1.bname.focus();
                    return false;
                }
                var auth = document.f1.auth.value;
                if(auth=="")
                {
                    alert("Please enter Author Name");
                    document.f1.auth.focus();
                    return false;
                }
                var publ = document.f1.publ.value;
                if(publ=="")
                {
                    alert("Please enter Publisher name");
                    document.f1.publ.focus();
                    return false;
                }
                var isbn = document.f1.isbn.value;
                if(isbn=="")
                {
                    alert("Please enter ISBN Number");
                    document.f1.isbn.focus();
                    return false;
                }
                var ed = document.f1.edi.value;
                if(ed=="")
                {
                    alert("Please enter book edition");

```

```

        document.f1.edi.focus();
        return false;
    }
    var cost = document.f1.cost.value;
    if(cost=="")
    {
        alert("Please enter cost of the book");
        document.f1.cost.focus();
        return false;
    }
    var qty = document.f1.qty.value;
    if(qty=="")
    {
        alert("Please enter number of books you want!!!\"");
        document.f1.qty.focus();
        return false;
    }
}
</script>
</head>
<body>
<br/><br/><br/>
<center>
<form name="f1" action="Catalogue.php" method="post" onsubmit=
"javascript:return validate()">
<table border="3" cellpadding="0" cellspacing="0">
<tr>

<td>
<table cellspacing="5">
<tr>
<td colspan="2" align="center"><h2><u>Book Details Entry
Form</u></h2></td>

</tr>
<tr>
<td> Book Name</td>
<td><input type="text" name="bname" size="50"></td>

</tr>
<tr>
<td> Author</td>
<td><input type="text" name="auth" size="50"></td>

</tr>
<tr>
<td> Publication</td>
<td><input type="text" name="publ" size="50"></td>

</tr>

```

```

<tr>
<td> ISBN</td>
<td><input type="text" name="isbn" size="15"></td>
</tr>

<tr>
<td> Edition</td>
<td><input type="text" name="edi" size="15"></td>
</tr>

<tr>
<td> Cost</td>
<td>$<input type="text" name="cost" size="5"></td>
</tr>

<tr>
<td> Quantity</td>
<td><input type="text" name="qty" size="3"></td>
</tr>

<tr>
<td colspan="2" align="center"><input type="submit" value="submit">
</td>
</tr>
</table>

</td>
</tr>
</table>
</form>

</center>
</body>
</html>

```

Catalogue.php

```

<?php
$conn = mysql_connect("localhost","root","");
if($conn)
    echo "Connected to database!!!";
else
    echo "Failed to Connect:".mysql_error();
mysql_select_db("test",$conn) or die("No Database existing:".mysql_error());
if(isset($_POST['bname']))
{
    $bname=$_POST['bname'];
    $auth=$_POST['auth'];
    $publ=$_POST['publ'];
    $isbn=$_POST['isbn'];

```

```

$edi=$_POST['edi'];
$cost=(float)$_POST['cost'];
$qty=(int)$_POST['qty'];
echo "<br/><br/><br/><p align='right'><a href='bkEntryForm.html'>[Book
Entry Page]</a></p>"; $query = "INSERT INTO catalogue VALUES
('$bname','$auth','$publ','$isbn','$edi','$cost','$qty')";
mysql_query($query);
$result = mysql_query("select * from catalogue");
?>
<html>
<body><center>
<font face="verdana" size="4">
<table border="1" cellpadding="0" cellspacing="0">
<tr>
    <th colspan="8" align="center">Books List</td>
</tr>
<tr>
    <th>S.No.</th>
    <th>Title</th>
    <th>Author</th>
    <th>Publication</th>
    <th>ISBN</th>
    <th>Edition</th>
    <th>Cost</th>
    <th>Quantity</th>
</tr>
<?php $num=1; while($row = mysql_fetch_array($result))

{ ?> <tr>
    <td align="center"><?php echo $num++; ?>
    </td> <td align="left"><?php echo $row['bname']; ?> </td>
    <td align="left"><?php echo $row['auth']; ?> </td>
    <td align="left"><?php echo $row['publ']; ?> </td>
    <td align="center"><?php echo $row['isbn']; ?> </td>
    <td align="center"><?php echo $row['edi']; ?> </td>
    <td align="center"><?php echo $row['cost']; ?> </td>
    <td align="center"><?php echo $row['qty']; ?> </td>
</tr> <?php } ?> </table>
</center>
</body>
</html>

<?php } ?>

```

OUTPUT:

The screenshot shows a web browser window with the URL <http://localhost/wtlab/bkEntryForm.htm> in the address bar. The title bar says "Catalogue Page". The main content is a form titled "Book Details Entry Form" with the following fields:

| | |
|-------------|-------------------------|
| Book Name | <input type="text"/> |
| Author | <input type="text"/> |
| Publication | <input type="text"/> |
| ISBN | <input type="text"/> |
| Edition | <input type="text"/> |
| Cost | \$ <input type="text"/> |
| Quantity | <input type="text"/> |

At the bottom right of the form is a "submit" button.

The screenshot shows a web browser window with the URL <http://localhost/wtlab/bkEntryForm.htm> in the address bar. The title bar says "Catalogue Page". The main content is a form titled "Book Details Entry Form" with the following fields filled in:

| | |
|-------------|---------------|
| Book Name | XML Bible |
| Author | Winston |
| Publication | Wiely |
| ISBN | 0-7645-4760-7 |
| Edition | Fifth |
| Cost | \$ 40.5 |
| Quantity | 3 |

At the bottom right of the form is a "submit" button.

Connected to database!!!

[\[Book Entry Page\]](#)

| Books List | | | | | | | |
|------------|-------------------------|-----------|------------------|---------------|---------|------|----------|
| S.No. | Title | AUthor | Publication | ISBN | Edition | Cost | Quantity |
| 1 | Artificial Intelligence | S. Russel | Princeton Hall | 0-13-1038-5-2 | Sixth | 63 | 2 |
| 2 | HTML in 24 Hours | Sam Peter | Sam Publications | 0-672-32841-0 | Fifth | 50 | 2 |
| 3 | Java 2 | Watson | BPB Publications | 0-41-1058-7-2 | Third | 63 | 6 |
| 4 | XML Bible | Winston | Wiely | 0-7645-4760-7 | Fifth | 40.5 | 3 |

VIVA QUESTIONS:

1. What is the default session time in PHP?
2. Is it possible to extend the execution time of a PHP script?
3. \$_SERVER is an array including information created by the web server such as paths, headers, and script locations.
4. How to initiate a session in PHP?

WEEK 11:

25.HTTP is a stateless protocol. Session is required to maintain the state. The user may add some items to cart from the catalog page. He can check the cart page for the selected items. He may visit the catalogue again and select some more items. Here our interest is the selected items should be added to the old cart rather than a new cart. Multiple users can do the same thing at a time (i.e., from different systems in the LAN using the ip-address instead of local host). This can be achieved through the use of sessions. Every user will have his own session which will be created after his successful login to the website. When the user logs out his session should get invalidated (by using the method session. Invalidate ()). Modify your catalogue and cart PHP pages to achieve the above mentioned functionality using sessions.

AIM: To write a program to maintain state using sessions.

DESCRIPTION: HTTP is a stateless protocol. Session is required to maintain the state. Every user will have his own session which will be created after his successful login to the website. Until user logout session will be maintained. Upon logout his session will get invalidated.

ALGORITHM:

1. Create a catalogue page
2. If user login successful
 - 2.1. Define Session variable.
 - 2.2 If user selects an item add the item to the session variable
 - 2.3 If user un select the item remove the item from session variable
3. If user logout
 - 3.1 Invalidate the session by using session.invalidate ()

PROGRAM:**style.css**

```
body
{
    width:800px;
}
.txt-heading
{
    padding: 5px 10px;
    font-size:1.1em;
    font-weight:bold;
    color:#999;
```

```
}

.btnAddAction
{
    color:#D60202;
    border:0;

    padding:2px 10px;
    font-size:0.9em;
}

#btnEmpty

{
    background-color:#D60202;
    border:0;
    padding:1px 10px;

    color:#FFF;
    font-size:0.8em;
    font-weight:normal;
    float:right;

    text-decoration:none;
}

.btnAddAction
{
    background-color:#79b946;
    border:0;
    padding:3px 10px;
    color:#FFF;

    margin-left:1px;
}

#shopping-cart
{
    border-top: #79b946 2px solid;
    margin-bottom:30px;
}

#shopping-cart .txt-heading
{
    background-color: #D3F5B8;
}

#shopping-cart table
{
    width:100%;
    background-color:#F0F0F0;
```

```
}

#shopping-cart table td
{

    background-color:#FFFFFF;
}

.cart-item
{

    border-bottom: #79b946 1px dotted;
    padding: 10px;
}

#product-grid
{

    border-top: #F08426 2px solid;
    margin-bottom:30px;
}

#product-grid .txt-heading
{
    background-color: #FFD0A6;
}

.product-item {
    float:left;
    background:#F0F0F0;
    margin:15px;

    padding:5px;
}

.product-item div
{

    text-align:center;
    margin:10px;
}

.product-price
{

    color:#F08426;
}

.product-image {
    height:100px;
    background-color:#FFF;
}
```

dbController.php

```

<?php
class DBController {
    private $host = "localhost";
    private $user = "root";
    private $password = "";
    private $database = "test";
    function __construct() {
        $conn = $this->connectDB();
        if(!empty($conn)) {
            $this->selectDB($conn);
        }
    }
    function connectDB() {
        $conn = mysql_connect($this->host,$this->user,$this-
        >password); return $conn;
    }
    function selectDB($conn) {
        mysql_select_db($this->database,$conn);
    }
    function runQuery($query) {
        $result = mysql_query($query);
        while($row=mysql_fetch_assoc($result)) {
            $resultset[] = $row;
        }
        if(!empty($resultset))
            return $resultset;
    }
}

```

index.php

```

<?php
session_start();
require_once("dbcontroller.php");
$db_handle = new DBController();
if(!empty($_GET["action"])) {
switch($_GET["action"]) {
    case "add":
        if(!empty($_POST["quantity"])) {
            $productByCode = $db_handle->runQuery("SELECT * FROM
tblproduct WHERE code='" . $_GET["code"] . "'");
            $itemArray=array($productByCode[0]["code"]=>array('name'=>$productByCode[0]["
name"],'code'=>$productByCode[0]["code"], 'quantity'=>$_POST["quantity"],
'price'=>$productByCode[0]["price"]));
            if(!empty($_SESSION["cart_item"]))
            {

```

```

        if(in_array($productByCode[0]["code"],$_SESSION["cart_item"]))
        {
            foreach($_SESSION["cart_item"] as $k => $v)
            {
                if($productByCode[0]["code"] == $k)
                    $_SESSION["cart_item"][$k]["quantity"] = $_POST
                    ["quantity"];
            }
        } else {
            $_SESSION["cart_item"]
            =array_merge($_SESSION["cart_item"],$itemArray);
        }
    } else {
        $_SESSION["cart_item"] = $itemArray;
    }
}
break;
case "remove":
    if(!empty($_SESSION["cart_item"])){
        foreach($_SESSION["cart_item"] as $k => $v) {
            if($_GET["code"] == $k)
                unset($_SESSION["cart_item"][$k]);
            if(empty($_SESSION["cart_item"]))
                unset($_SESSION["cart_item"]);
        }
    }
}
break;
case "empty":
    unset($_SESSION["cart_item"]);
break;

}
}
?>
<HTML>
<HEAD>
<TITLE>Simple PHP Shopping Cart</TITLE>
<link href="style.css" type="text/css" rel="stylesheet" />
</HEAD>
<BODY>
<div id="shopping-cart">
<div class="txt-heading">Shopping Cart <a id="btnEmpty"
href="index.php?action=empty">Empty Cart</a></div> <?php
if(isset($_SESSION["cart_item"])){
    $item_total = 0;
?>
<table cellpadding="10" cellspacing="1">
<tbody>

```

```

<tr>
<th><strong>Name</strong></th>
<th><strong>Code</strong></th>
<th><strong>Quantity</strong></th>
<th><strong>Price</strong></th>
<th><strong>Action</strong></th>
</tr>
<?php
foreach ($_SESSION["cart_item"] as $item){
    ?
    <tr>

        <td><strong><?php echo $item["name"]; ?></strong>
        </td>
        <td><?php echo $item["code"]; ?></td>
        <td align=right><?php echo $item["quantity"]; ?></td>
        <td align=right><?php echo "$".$item["price"]; ?></td>
        <td><a href="index.php?action=remove&code=<?php echo
            $item["code"]; ?>" class="btnRemoveAction">Remove
            Item</a></td>
        </tr>
        <?php
    $item_total += ($item["price"]*$item["quantity"]);
    ?
}
<tr>
<td colspan="5" align=right><strong>Total:</strong> <?php echo "$".$item_total; ?></td>
</tr>
</tbody>
</table>

<?php }?>
</div>
<div id="product-grid">
    <div class="txt-heading">Products</div>
    <?php
    $product_array = $db_handle->runQuery("SELECT * FROM tblproduct
    ORDER BY id ASC"); if (!empty($product_array)) {
        foreach($product_array as $key=>$value){
    ?
        <div class="product-item">
            <form method="post" action="index.php?action=add&code =<?php
            echo $product_array[$key]["code"]; ?>">

                <div class="product-image">" height="100" width="75"></div>
                <div><strong><?php echo $product_array[$key]["name"]; ?></strong></div>
                <div class="product-price">
                    <?php echo "$".$product_array[$key]["price"]; ?>
                </div>

```

```

<div><input type="text" name="quantity" value="1" size="2"
/><input type="submit" value="Add to cart" class="btnAddAction"
/></div>

</form>
</div>
<?php } }?>
</div>
</BODY>
</HTML>

```

OUTPUT:**Empty Cart**
After adding items to cart

| Name | Code | Quantity | Price | Action |
|-------------------------|---------------|----------|---------|-----------------------------|
| XML Bible | 0-7645-4760-7 | 10 | \$40.50 | Remove Item |
| Artificial Intelligence | 0-13-1038-5-2 | 3 | \$63.00 | Remove Item |

Total: \$594

After modification

The screenshot shows a web browser window with the URL <http://localhost/scart/index.php?action=remove&id=1>. The title bar says "Simple PHP Shopping Cart". The main content area has a green header "Shopping Cart" with a red "Empty Cart" button. Below it is a table with columns: Name, Code, Quantity, Price, and Action. Three items are listed:

| Name | Code | Quantity | Price | Action |
|-------------------------|---------------|----------|---------|-----------------------------|
| Atrificial Intelligence | 0-13-1038-5-2 | 3 | \$63.00 | Remove Item |
| HTML in 24 Hours | 0-672-32841-0 | 5 | \$50.00 | Remove Item |
| Java 2 | 0-41-1058-7-2 | 3 | \$63.00 | Remove Item |

Total: \$628

Below the cart is a section titled "Products" with four items:

- XML Bible** (\$40.50) - Image: XML Bible book cover.
- Atrificial Intelligence** (\$63.00) - Image: Artificial Intelligence book cover.
- Java 2** (\$63.00) - Image: Core Java book cover.
- HTML in 24 Hours** (\$50.00) - Image: Sams Teach Yourself HTML 4 in 24 Hours book cover.

Each product item has a quantity input field (set to 1) and a green "Add to cart" button.

VIVA QUESTIONS:

1. What is the difference between include() and require()?
2. What is PHP sessions default timeout?
3. How would you find the number of repeated characters in a given string "abcabcxyz"?
4. What is PHP sessions default timeout?