

Ex.No.1: Creation of Static web pages using HTML

1.a. Design the following web page using HTML.

The screenshot shows a Microsoft Internet Explorer window displaying a web page titled "Table Example Page". The page contains a heading "Camelid comparison" and a caption "Approximate as of 9/2002". It features a table with two rows. The first row has a blue silhouette of a camel in the left column. The second row contains the text "Camels (bactrian)" in the first column, followed by the values "2", "Africa/Asia", "Llama", and "Llama" in the subsequent columns respectively. The table has four columns with headers "# of Humps", "Indigenous region", "Spits?", and "Produces Wool?".

	# of Humps	Indigenous region	Spits?	Produces Wool?
Camels (bactrian)	2	Africa/Asia	Llama	Llama
Llamas	1	Andes Mountains		

1.b. Design a web page using HTML to display your profile. Use your digital image as a link. When the image link is clicked, a new page opens with the following links.

- Academic profile
- List of Mini Projects done
- Personal Profile

In each of the above links, provide a link "HOME" that will return to your first page.

1.c. Design the following web page using the concept of navigation frames.

<p>MENU</p> <ul style="list-style-type: none">• Rose• Jasmine• Lotus	<p>Click on the choice of your favorite flower links in the left frame in order to view the details of your favorite flower in the right frame.</p>
--	---

1.d. Design the following web page using HTML to demonstrate the concept of nested lists.

- Coffee
- Tea
 - Black Tea
 - Green Tea
 - China
 - Africa
- Milk

1.e. Design a web page using HTML with the following.

- i. To embed an image map in a web page
- ii. To fix the hot spots in that map
- iii. Show all the related information when the hot spots are clicked.

1.f. Design a Drop-down List in HTML with the following options.

- i. Brinjal
- ii. Potato
- iii. Tomato
- iv. Carrot
- v. Raddish

1.g. Reverse the following text using HTML text direction tag.

“Web Technology”

Design a web page using HTML that contains a link – NITT. When the link is clicked, it will open the website “<http://www.nitt.edu>” in a new browser window.

1.h. Design a Form as shown below to implement the Checkbox and Radio button.

Choose any ONE of your favorite Car from the following.

- BMW
- BENZ
- MARUTI

Choose your favorite colors from the following.

- RED
- GREEN
- VIOLET
- NAVY

1.i. Design a Marquee that scrolls the text “WELCOME” from right to left across the browser window. Set the background color to Marquee as RED and Text color as WHITE. Also set the background color to the Web page as BLUE.

1.j. Write XHTML markups to accomplish each of the following:

- a. Insert a framed Web page, with the first frame extending 500 pixels across the page from the left side.
 - b. Insert a table with a border of 7.
 - c. Indicate alternate content to a frameset.
 - d. Insert an image map in a page using deitel.gif as an image and map with name = "hello" as the image map, and set the alt text to “hello”.
-

Ex.No.2:Creation of Web pages using Cascading Style Sheets (CSS) - Part I

- 2.a. Write a CSS rule that makes all text 1.5 times larger than the base font of the system and colors the text red.
- 2.b. Write a CSS rule that removes the underlines from all links inside list items (li) and shifts all list items left by 3 ems.
- 2.c. Write a CSS rule that places a background image halfway down the page, tiling it horizontally. The image should remain in place when the user scrolls up or down.
- 2.d. Write a CSS rule that displays h1 elements in blue. In addition, create a rule that displays all links in blue without underlining them. When the mouse hovers over a link, change the link's background color to yellow.
- 2.e. Write a CSS rule that gives all **h1** elements a padding of 0.8 **ems**, a **dashed** border style and a margin of 0.8 **ems**.

Write a CSS rule that gives all **h2** elements a padding of 0.5 **ems**, a **dotted** border style and a margin of 0.5 **ems**.

Ex.No.2:Creation of Web pages using Cascading Style Sheets (CSS) - Part II

2.a. Write the CSS rules to display the following borders.

Thick Groove
Medium Groove
Thin Groove
Thin Red Inset
Medium Blue Outset

2.b Design an external CSS document with two headers contained in <h1>and <h2>tags in which <h1>and <h2>elements both are in green, 22pt, Times New Roman font. Add a paragraph of text also in black, Times New Roman font, but 12pt in size. No declaration may be repeated. Include comments to signify what each rule does.

2.c. Compose a web page with a heading of <h1>and a paragraph of text <p>. Specify the font size in the selector for the <body>element as 12 pixels. Specify the heading font size as 1.2em. Give the paragraph a font size of 0.9em. Give the heading a thin, black, solid border and a pink background. Give the paragraph a thin, solid, black border, white text, and a gray background. Give both the heading and the paragraph a padding of 1em.(use External style sheets)

2.d. Write your own user Content.css file. Make all backgrounds white and all text black. Make all unvisited links blue, visited links purple, and active links red. Have all input fields use a white background with a black border and black text. Don't display any images.

2.e. Using the list-style-type property and other properties, create a tabbed navigation menu using only lists (and) and CSS. Include four links, one in each list-item and lay out the tabs horizontally. Put a thin, solid border below all the tabs and around each individual tab, make each tab 100 pixels wide, center the text of each tab, and space each tab from the other.

Ex.No.2: Creation of web pages using Cascading Style Sheets (CSS) – Part III

2.a. Design the following web page using XHTML with conflicting styles. The Left margin for unordered list elements is set as 75 pixels and for nested list elements, it is set as 15 pixels whose text is underlined. The emphasized text is bold whose color is red. Initially, the text format for the link text is none. When the mouse is placed over the link, the text is underlined whose color changes to red and the background color is set to “aqua”.

Shopping list for Monday:

- Milk
- Bread
 - White bread
 - Rye bread
 - Whole wheat bread
- Rice
- Potatoes
- Pizza *with mushrooms*

[Go to the Grocery store](#)

2.b. Design the web page with the CSS styles as in Ex.2 using XHTML. The styles (CSS rules) are coded in a separate document(style.css). Link the external CSS file to the corresponding XHML file. Also validate the CSS file using W3C CSS validation service.

2.c. Design a web page that contains the following paragraph.

Thanks for visiting my web site. I hope you enjoy a lot.

Please Note: This site will be moving soon. Check periodically for updates.

Write a separate User Style Sheet with the following styles. The body text is formatted as follows:

`font-size: 20pt; color: yellow; backgroundcolor: blue`

Set this user style sheet in your browser's options and apply it to the XHTML file containing the above paragraph.

2.d. Write an XHTML document that shows the results of a color survey. The document should contain a form with radio buttons that allows users to vote for their favorite color. One of the colors should be selected as a default. The document should also contain a table showing various colors and the corresponding percentage of votes for each color. Each row should be displayed in the color to which it is referring. Use inline style sheets for formatting.

2.e.(i)

Write a CSS rule that gives all **h1** and **h2** elements a padding of 0.5 **ems**, a **grooved** border style and a margin of 0.5 **ems**.

(ii)

Write a CSS rule that changes the color of all elements containing attribute **class = "greenMove"** to green and shifts them down 25 pixels and right 15 pixels.

Ex.No: 3-Part I : Validation of XML Documents using DTD, Schema and Conversion of XML to XHTML using XSLT

a. Write the XML Schema for the following XML document.

```
<?xml version="1.0" encoding="UTF-8" ?>

<bank>
<accounts>
    <savings_accounts>
        <savings_account id="a1" interest="0.03">
            <balance>2500</balance>
        </savings_account>
        <savings_account id="a2" interest="0.03">
            <balance>15075</balance>
        </savings_account>
    </savings_accounts>
    <checking_accounts>
        <checking_account id="a3">
            <balance>4025</balance>
        </checking_account>
        <checking_account id="a4">
            <balance>-125</balance>
        </checking_account>
        <checking_account id="a5">
            <balance>325</balance>
        </checking_account>
    </checking_accounts>
</accounts>
<customers>
    <customer id="c1">
        <name>Ben Richardson</name>
        <address>Park Drive 2</address>
    </customer>
    <customer id="c2">
        <name>Marc Wetcher</name>
        <address>Mill Drive 75</address>
    </customer>
<customer id="c3">
    <name>Angel Steady</name>
    <address>Lake Sight 15</address>
</customer>
</customers>
    <customer_accounts>
        <customer_account c_id="c1" ac_id="a2" />
        <customer_account c_id="c1" ac_id="a3" />
        <customer_account c_id="c2" ac_id="a4" />
        <customer_account c_id="c3" ac_id="a1" />
        <customer_account c_id="c3" ac_id="a5" />
    </customer_accounts>
</bank>
```

b. Write the DTD for the following XML document.

```
<Books>
  <Book>
    <Isbn> 0 - 8447 - 768 </Isbn>
    <Title> XML in a Nutshell </Title>
    <Author> Harold, Elliotte Rusty </Author>
  </Book>
  <Book>
    <Isbn> 978 - 81 - 317 - 6235 - 6 </Isbn>
    <Title> Service - Oriented Architecture </Title>
    <Author> Thomas Erl </Author>
  </Book>
  <Book>
    <Isbn> 978 - 81 - 317 - 1869 - 8 </Isbn>
    <Title> XML and WebServices </Title>
    <Author> Ron Schmelzer </Author>
  </Book>
</Books>
```

c. Convert the following XML document into an XHTML table using XSLT.

```
<? Xml version = "1.0" ?>
<? Xml : stylesheet type = "text/xsl" href = "Students.xsl" ?>
<Students>
  <Student>
    <name>John</name>
    <rollno>CS101</rollno>
    <department>CSE</department>
    <cgpa>9.5</cgpa>
  </Student>
  <Student>
    <name>Williams</name>
    <rollno>CE034</rollno>
    <department>CIVIL</department>
```

```

<cgpa>7.4</cgpa>
</Student>

<Student>
<name>Rohit</name>
<rollno>EC087</rollno>
<department>ECE</department>
<cgpa>8.7</cgpa>
</Student>

</Students>

```

d. Write the DTD and XML Schema for the following XML document.

```

<? Xml version = "1.0" ?>
<? Xml : stylesheet type = "text/xsl" href = "Students.xsl" ?>

<Employee>

<Emp>
<name>Roshan</name>
<eid>E101</eid>
<department>HR</department>
<doj>09-SEP-1999</doj>
<salary>68,000</salary>
</Emp>

<Emp>
<name>Mary</name>
<eid>E233</eid>
<department>Accounts</department>
<doj>06-AUG-2001</doj>
<salary>24,000</salary>
</Emp>

<Emp>
<name>Sunil</name>
<eid>E521</eid>
<department>Engineering</department>
<doj>04-NOV-2009</doj>
<salary>42,000</salary>
</Emp>

</Employee>

```

e. Convert the XML document given in Ex.b into an XHTML document for tabular presentation
Using XSLT.

Ex.No: 3 – Part II: Validation of XML Documents using DTD, Schema and Conversion of XML to XHTML using XSLT

- a. Given the following DTD, Generate the corresponding XML document. Also write the XML schema for that document.

```
<!ELEMENT bib (book*)>
<!ELEMENT book (title, (author+ | editor+ ), publisher, price)>
<!ATTLIST book year CDATA #REQUIRED >
<!ELEMENT author (last, first)>
<!ELEMENT editor (last, first, affiliation)>
<!ELEMENT title (#PCDATA)> <!ELEMENT last (#PCDATA)>
<!ELEMENT first (#PCDATA)> <!ELEMENT affiliation (#PCDATA)>
<!ELEMENT publisher (#PCDATA)> <!ELEMENT price (#PCDATA)>
```

- b. Write the DTD for the following XML document.

```
<pets>
  <pet>
    <name>Tilly</name>
    <age>14</age>
    <type>cat</type>
    <color>silver</color>
  </pet>
  <pet>
    <name>Amanda</name>
    <age>10</age>
    <type>dog</type>
    <color>brown</color>
  </pet>
  <pet>
    <name>Jack</name>
    <age>3</age>
    <type>cat</type>
    <color>black</color>
  </pet>
  <pet>
    <name>Blake</name>
    <age>12</age>
    <type>dog</type>
    <color>blue</color>
  </pet>
  <pet>
    <name>Loosey</name>
    <age>1</age>
    <type>cat</type>
    <color>brown</color>
  </pet>
  <pet>
    <name>Stop</name>
    <age>5</age>
    <type>pig</type>
    <color>brown</color>
  </pet>
</pets>
```

- c. Create an XML document for the following letter as per the element names – [letter, date, addressee, name, address_one, address_two, greeting, paragraph, italics, list, item, closing]. Also validate the XML file generated by writing the appropriate DTD.

December 11, 2002
Melville Dewey
Columbia University
New York, NY

Dear Melvile,

I have been reading your ideas concerning the nature of librarianship, and I find them very intriguing. I would love the opportunity to discuss with you the role of the card catalog in today's libraries considering the advent to World Wide Web. Specifically, how are things like Google and Amazon.com changing our patrons' expectations of library services? Mr. Cutter and I will be discussing these ideas at the next Annual Meeting, and we are available at the following dates/times:

- * Monday, 2-4
- * Tuesday, 3-5
- * Thursday, 1-3

We hope you can join us.

Sincerely,
James K Robert

- d. Write the DTD and XML Schema for the following XML document.

```
<bank>
  <account account-number="A-101" branch-name="Downtown"
    balance="500">
  </account>
  <account account-number="A-102" branch-name="Perryridge"
    balance="400">
  </account>
  <account account-number="A-201" branch-name="Brighton"
    balance="900">
  </account>
  <customer customer-name="Johnson" customer-street="Alma"
    customer-city="Palo Alto">
  </customer>
  <customer customer-name="Hayes" customer-street="Main"
    customer-city="Harrison">
  </customer>
  <depositor account-number="A-101" customer-name="Johnson">
  </depositor>
  <depositor account-number="A-201" customer-name="Johnson">
  </depositor>
  <depositor account-number="A-102" customer-name="Hayes">
  </depositor>
</bank>
```

- e. Convert the XML document given in Ex.b into an XHTML document for tabular presentation Using XSLT.
-

Ex.No.4: Programs using Java Script – Part I

4.a. Write a JavaScript program to prompt the user for the radius and call function Sphere_Volume to calculate and display the volume of the sphere. Also invoke the function Circle_Area to calculate and display the area of the circle. The user should input the radius through an XHTML text field in a form and click an XHTML button to initiate the calculation.

4.b. Write a function Compute_Distance that calculates the distance between two points (x_1, y_1) and (x_2, y_2). All numbers and return values should be floating-point values. Incorporate this function into a Java script that enables the user to enter the Coordinates of the points through an XHTML form.

4.c. Write a function Integer_Power(base, exponent) that returns the value of $\text{Base}^{\text{exponent}}$. Assume that exponent and base are integers. Function Integer_Power should use a for or while statement to control the calculation. Do not use any math library functions.

Incorporate this function into a Java script that reads integer values from an XHTML form for base and exponent and performs the calculation with the Integer_Power function. The XHTML form should consist of two text fields and a button to initiate the calculation. The user should interact with the program by typing numbers in both text fields and then clicking the button.

4.d.

- a) Write a function that determines whether a number is prime or Not.
- b) Use this function in a Java script that determines and prints all the prime numbers between 1 and 1000. How many of these 1000 numbers do you really have to test before being sure that you have found all the primes? Display the results in a <textarea>.
- c) Initially, you might think that $n/2$ is the upper limit for which you must test to see whether a number is prime, but you only need go as high as the square root of n . Why? Rewrite the program, and run it both ways. Estimate the performance improvement.

4.e. Write a function that determines whether a number is Armstrong or Not. Use this function in a Java script that determines and prints all the Armstrong numbers between 100 and 5000.

4.f. Write a user defined function in Java script to design and simulate a Scientific Calculator that includes all the arithmetic and scientific operations.

Ex.No.4: Programs using Java Script – Part II

- 4.a. Write a JavaScript program to perform the following String Manipulation Operations with and without using the methods of String Object.
- I. To search for the last occurrence of a substring in a given string
 - II. To extract a substring from a given string
 - III. To split the given string into various substrings
- 4.b. i. Write a JavaScript to display the Date (as a number), Day (as a string), Month (as a string) and Year (as a four digit number) that is seven days before and after the Current Date.
- ii. Write a JavaScript that inputs the date of birth from the user dynamically and display the current age of the user in the following format : (----Years, ----Months, ----Days).
- 4.c. Write a User-Defined function in Java Script to perform Binary search. Get the ‘N’ number of elements dynamically from the user.
The user have to enter the key value to be searched through the HTML form as designed below. When the user clicks the OK button, the search result should be displayed in the corresponding text field as either FOUND at “Index” – if the search key is present or NOT FOUND – if the search key is not present.

BINARY SEARCH

Enter the Search Key :

Search Result :

4.d. Write a Java Script program to get N number of strings dynamically from the user and arrange them in alphabetical order. Display the strings in sorted order and also display those strings that begin with ‘e’ / ‘E’.

4.e. write a JavaScript that reads several lines of text and prints a table indicating the number of one-letter words, two-letter words, three-letter words, etc. appearing in the text, in a tabular form.

For example, the sentence **Welcome to “Java” Lab.** contains:

Word Length	Occurrences
1	0
2	1
3	0
4	1
5	0
6	1
7	1

Ex.No.4: Programs using Java Script – Part III

4.a. Write a JavaScript program to perform the following String Manipulation Operations with and without using the methods of String Object.

- I. To search for the first occurrence of a substring in a given string
- II. To return the character at a specified index
- III. To concatenate two string objects

4.b. Write a JavaScript that inputs text from an XHTML form and outputs it in Upper case and Lower case letters.

4.c. Write a JavaScript that reads a Five-Letter word from the user and produces all possible Three-Letter words that can be derived from the letters of the Five-Letter word.

For example, the three-letter words produced from the word “bathe” include the commonly used words : ate bat bet tab hat the tea

4.d. Write a JavaScript to display the current time and print the message according to the time. For example, Good Morning in Morning, Good After Noon in Noon, Good Evening in evening and Good Night in Night.

4.e. Write a JavaScript program to perform Merge Sort for a given set of N numbers dynamically prompted from the user. (Use user defined function to perform Merge Sort).

Ex.No.5: Creation of Dynamic Web pages DHTML – Part I

- 6.a. Write a DHTML code to move the text randomly on the web page where the color and font-size of the text vary dynamically.
 - 6.b. Add an image to the web page. Write a DHTML code that corresponds to the movement of mouse anywhere on the page. Display the element name along with its (X,Y) co-ordinates, whenever the mouse is moved over the page.
 - 6.c. Design a web page using DHTML such that one image gets loaded when the mouse is moved over the page and another image gets loaded when the mouse loses its control inside the page.
 - 6.d. Write a DHTML code to display all the XHTML elements in a document, in the order in which they appear using all and child collections.
 - 6.e. Write a DHTML code that responds to a click anywhere on the page by displaying an alert dialog. Display the event name if the user held SHIFT during the mouse click. Display the element name that triggered the event if the user held CTRL during the mouse click.
-

Ex.No.5: Creation of Dynamic Web pages DHTML – Part II

6.a. Create a web page with two horizontal frames. Write a DHTML code to display three images in the upper frame. When any one of the images is clicked, it must be displayed in the lower frame. On clicking the displayed image, it must be cleared.

6.b. Create a web page with two vertical frames. Write a DHTML code to add a list on the left frame along with items available in a departmental store. When the user clicks on an item in the list box, its image will be displayed on the right frame. Also the price of the selected item must be dynamically retrieved in the left frame and transferred to the right frame through an alert dialog.

6.c. Create a web page that enables the user to play the game of 15. There is a 4-by-4 board implemented as an XHTML table for a total of 16 slots. One of the slots is empty. The other slots are occupied by 15 tiles, randomly numbered from 1 through 15. Any tile next to the currently empty slot can be moved into the currently empty slot by clicking on the tile. Your program should create the board with the tiles out of order. The user's goal is to arrange the tiles in sequential order row by row. Using the DHTML model and the onclick event, write a script that allows the user to swap the positions of the open position and adjacent tile.

6.d. Write a DHTML code to add an image to the web page. The image is then divided into 16 equally sized pieces. Discard one of the pieces and randomly place the other 15 pieces in the XHTML table.

6.e. Add two elements to the web page that the users can click. Use an image as the first element. When the user clicks the image, display an alert dialog box with the text "You clicked the image". For the second element, create a one-row table containing a text string. Set the table border to 1. When the user clicks the table, display an alert dialog box containing "You clicked the table". In the two accompanying functions, set each event object to true. (Use Event Bubbling)

6.f. Write a DHTML code to load an image on the webpage and perform the following.

- Rotate the image horizontally and vertically using Flip Filters.
 - Apply transparency effects dynamically using Chroma Filter.
 - Produce the Negative image effect using Invert Filter.
 - Produce the Gray Scale image effect using Gray Filter.
 - Produce the X-Ray image effect using Xray Filter.
-

Ex.No: 6: Programs using Java Servlets

- a. Write a Generic Servlet program using Java Servlet to print the message -“HELLO WORLD” in the Servlet window.
- b. Write a Servlet program to demonstrate Http Get Request that retrieves the content (XHTML document) of a specified URL. When the user clicks the Get HTML Document button of Welcome.html, a get request is sent to the servlet WelcomeServlet.java. The servlet responds to the request by generating dynamically an XHTML document for the client that displays “ Welcome to Servlets ! ”.
- c. Design an HTML Form that consists of the following fields and a SUBMIT button.

Emp Name	Emp ID	DOB
Department	Salary	Email ID

Write a Generic Servlet Program to retrieve all the Form Parameter Names and their corresponding values and display them in the Servlet window.

- d. Design an HTML Form that consists of the following options in a Drop-down List and a SUBMIT button.

Red	Green	Blue
-----	-------	------

Write a Servlet Program using Http Servlet to retrieve the selected option from the drop-down list (use doGet() method) and display it in the Servlet window.

- e. Create a Web application that allows the user to select a favorite programming language and post the choice to the server. The response is a Web page in which the user can select another favorite language or click a link to view a list of book recommendations. When the user selects the list of book recommendations, a get request is sent to the server. The cookies previously stored on the client are read by the servlet and used to form a Web page containing the book recommendations. CookieServlet.java handles both the get and the post requests. The CookieSelectLanguage.html document contains four radio buttons (**C, C++, Java** and **VB 6**) and a **Submit** button. When the user presses **Submit**, the **CookieServlet** is invoked with a **post** request. The servlet adds a cookie containing the selected language to the response header and sends an XHTML document to the client. Each time the user clicks **Submit**, a cookie is sent to the client.

- f. Create a Web application for dynamic FAQs. The application should obtain the information to create the dynamic FAQ Web page from a database that consists of a Topics table and an FAQ table. The Topics table should have two fields—a unique integer ID for each topic (topicID) and a name for each topic (topicName). The FAQ table should have three fields—the topicID (a foreign key), a string representing the question (question) and the answer to the question (answer). When the servlet is invoked, it should read the data from the database and return a dynamically createdWeb page containing each question and answer, sorted by topic.
- g. Create a Web application to demonstrate basic session-tracking techniques, in that the servlet use HttpSession objects. Once again, the servlet handles both get and post requests. The document SessionSelectLanguage.html contains four radio buttons (C, C++, Java and VB 6) and a Submit button. When the user presses Submit, SessionServlet.java is invoked with a post request. The servlet responds by creating an object of type HttpSession for the client (or using an existing session for the client) and adds the selected language and an ISBN number for the recommended book to the HttpSession object. Then, the servlet sends an XHTML page to the client. The display information contains Session information such as Session ID, Whether new or old session, creation time, last accessed time and maximum inactivate interval. Each time the user clicks Submit, a new language/ISBN pair is added to the HttpSession object.
- h. The SurveyServlet.java and the Survey.html document demonstrate a three-tier distributed application that displays the user interface in a browser using XHTML. The middle tier is a Java servlet that handles requests from the client browser and provides access to the third tier—a Cloudscape database accessed via JDBC. The servlet in this example is a survey servlet that allows users to vote for their favorite animal. The document Survey.html contains five radio buttons (Dog, Cat, Parrot, Cow and None) and a Submit button. When the servlet receives a post request from the Survey.html document, the servlet updates the total number of votes for that animal in the database and returns a dynamically generated XHTML document containing the survey results to the client. The result should display the number of responses and vote percentage for each pet. Also display the total number of responses obtained.
- i. Design an HTML Form that consists of the following Checkbox options and a SUBMIT button.

Apple Orange Grapes Mango Lemon

Write a Servlet Program using Http Servlet to retrieve the selected options

(use doPost() method) and display them in the Servlet window.

Ex.No: 7: Programs using Java Server Pages (JSP)

- a. Write a program using JSP to print the message -“HELLO WORLD” in the browser window. Also insert the current date and time into the web page using the JSP Expression.
- b. Write a JSP program to sort the given list of names in alphabetical order and display the sorted list in the browser window.
- c. Design an HTML Form that consists of the following fields and a SUBMIT button.

Emp Name	Emp ID	DOB
Department	Salary	Email ID

Write a JSP Program to retrieve all the Form Parameter Names and their corresponding values and display them in the browser window.

- d. Write a JSP program to print the number of times a particular website is visited using the Session object.
- e. Create a web application using Java Server Pages to retrieve the Employee table (Eid, Ename, Dept, Doj, Salary) from SQL database via JDBC and display the retrieved table entries in the browser window. Also include options to insert, update and delete the database entries.
- f. Create a Web application for dynamic FAQs. The application should obtain the information to create the dynamic FAQ Web page from a database that consists of a Topics table and an FAQ table. The Topics table should have two fields—a unique integer ID for each topic (topicID) and a name for each topic (topicName). The FAQ table should have three fields—the topicID (a foreign key), a string representing the question (question) and the answer to the question (answer). When the JSP is invoked, it should read the data from the database and return a dynamically createdWeb page containing each question and answer, sorted by topic.

- g. Create a three-tier distributed application that displays the user interface in a browser using XHTML. The middle tier is a JSP that handles requests from the client browser and provides access to the third tier—a Cloudscape database accessed via JDBC. The JSP in this example is a survey that allows users to vote for their favorite animal. The document Survey.html contains five radio buttons (Dog, Cat, Parrot, Cow and None) and a Submit button. When the JSP receives a post request from the Survey.html document, the JSP updates the total number of votes for that animal in the database and returns a dynamically generated XHTML document containing the survey results to the client. The result should display the number of responses and vote percentage for each pet. Also display the total number of responses obtained.
- h. Create a web application using Java Server Pages to retrieve the BOOK table (bid, bname, author, isbn, price) from SQL database via JDBC and display the retrieved table entries in the browser window. Also include options to insert, update and delete the database entries.
-

Ex.No: 8: Programs using PHP

- a. Create a Web application using PHP. The application should obtain the information from the SQL database that consists of the following tables.

STUDENT (Rollno, Name, Branch, Year, Cgpa, DOB, E-mailID)
COURSES (Cid, Cname, FacultyName)
COURSE_TAKEN (Rollno, Cid)

Perform the following operations in PHP using Database Connectivity in the above tables.

- i. Add a Student Information
- ii. Add a Course
- iii. Assign a Course to a Student
- iv. Update a Student Entry
- v. Delete a Student Entry

- b. Create a Web application using PHP. The application should obtain the information from the SQL database that consists of the following tables.

PATIENT (Pid, Pname, DOB, ContactNo, Address)
DIAGNOSIS (Did, Dname, Medication, Department)
TREATMENT (Pid, Type, Did, DoctorName)

Perform the following operations in PHP using Database Connectivity in the above tables.

- vi. Add a Patient Information
 - vii. Add a Diagnosis
 - viii. Assign a Treatment to a Patient
 - ix. Update a Patient Entry
 - x. Delete a Patient Entry
-

Ex: 9: Programs on Mobile Applications

- a. Design restaurant data entry form using Table Layout and show different events using activity class.
- b. Write a program to capture image using built in camera and store it in database.
- c. Develop a banking application that registers the user by verifying OTP.
- d. Develop a native application that uses GPS location information and convert into speech.
- e. Develop a real-time application that uses any academic lecture video / audio and convert into text.