ASSIGNMENT 2

DATE

TITLE — DESIGN WITH HTML and XML

PROBLEM STATEMENT — Design and develop a web application using HTML, CSS, XML from scratch for the application: Leave Management Application.

LEARNING OBJECTIVES
- To develop web pages using HTML.
- To optimize page styles and layout with CSS.
- To distinguish between HTML and XML.

LEARNING OUTCOMES
- Define the key terms relevant to coding HTML and CSS: including tag, attribute, element, entity, selector, header, table, ordered list, unordered list, link, heading, paragraph.
- Describe the function of common tags and styles in short snippets of code & predict the output of the same.

H/W & S/W REQUIREMENTS: OS open source Fedora 20, networked computer with internet access, IDE, Web Browser

THEORY

INTRODUCTION

- Files that travel across the largest netwok in the world, the Internet and carry information from Server to Client that requested them are called 'Web pages' HTML documents.
- Individual who develops these web pages is called 'Web Developer'.
- Web pages are created using HTML Syntax.
- The organization of web pages into directories and files stored on the HDD of a computer is called Web Site creation.

Computer runs special software called Web Server software that allows:

- Web Site Management
- Accept a client's request for information.
- Respond to a client's request by providing the page with the required information.

Computers that offer the facility to read information stored in web pages are called 'Web Clients'. Web Clients run special software called a 'Browser' that allows to:

- Connect to an appropriate Server
- Query the Server for the information to be read.
- Provides an interface to read the information retrived by the Server.

Requirements of a good web site

- First Impression
- Interface Design
- Corporate Milder
- Coriolis Effect

HTML

- The language used to develop web pages is called HyperText Markup Language which is interpreted by a browser.
- HTML Tags are instructions that are embedded directly into text of document.
- HTML tags can be of 2 types:
  - ↳ Paired Tags
  - ↳ Singular Tags

TAG	NAME	EXAMPLE
<!--	comment	<!-- This can be anywhere -->
<a-	anchor	 Visit PICT
	bold	Example
<big>	big (text)	<big> Example
<body>	body of HTML document	<body> Content of HTML page </body>
 	line break	Contents Index
	center	Aboli
<dd> <dl> <dt>	definition description definition list	<dl> <dt> def term </dt> <dd> def of the term </dd> </dl>
	Emphasis	This is an example of using emphasis tag
<embed>	embed object	<embed src="your file" width="100%" height='60'>

DHTML- Dynamic HTML-

It combines HTML with Cascading Style Sheets (CSS) and Scripting Languages. HTML specifies a web page's elements like table, frame, paragraph and CSS can be used to determine an element's size, color, position.

CSS - CASCADING STYLE SHEETS

Style sheets are powerful mechanism for adding styles to Web documents that enforces standards and uniformity throughout a website and provides numerous attributes to create dynamic effects.

Style information can be associated with the web page in several ways:

- embedding directly through STYLE attribute
- embedding directly through <STYLE> tag
- embedding through <LINK> Element.

Order of Importance for adding style sheets into the document:

i) Inline styles
ii) Embedded styles
iii) Linked styles
iv) Imported styles
v) Default browser styles.

Advantages

- Ability to make global changes
- Greater author control
- Reduced clutter
- Improvement of design potential

XML - Nuts & Bolts -

- DTD
- XSD - eXtensible Schema Definition
- XSL - eXtensible Style Languages
- XML linking Languages.
- XML Namespaces.

Advantages of Schemas

- easier to validate the correctness of data
- easier to work with data from database
- easier to define data facets and data patterns
- easier to convert data between different data types.

HTML vs XML

HyperText Markup Language	Extensible Markup Language
• used to display data	• used to transport and store data
• it is markup language	• provides framework to define markup languages.
• not case sensitive	• case sensitive
• predefined tags	• custom tag
• no strict tags	• strict rules
• static	• dynamic
• does not preserve white spaces	• preserves white spaces.

Algorithmic Steps

1. Finalize the scope of the web application.
2. Task distribution.
3. Design the respective application in Jlab notebook.
4. Write code snippet in the editor (HTML, CSS, XML) and visualize the same in browser
5. Testing.

TESTCASES

TEST CASE	EXPECTED O/P	ACTUAL O/P
1. All web pages must be properly linked	linked correctly	Success
2. Back button, home page button and logout.	Buttons exists and navigates	Success
3. Pop up confirmation on final button	Pop up appear	Success

Conclusion Thus we have successfully outlined the scope of the project and designed web pages using HTML, CSS and XML successfully.