WEBSITE DESIGNING FOR NIELIT ITANAGAR LIBRARY USING HTML AND CSS

Technical Report · January 2018

CITATIONS
0

READS
13,844

1 author:

G. Stephen
St. Xavier's University, Kolkata
73 PUBLICATIONS **86** CITATIONS

SEE PROFILE

WEBSITE DESIGNING FOR NIELIT ITANAGAR LIBRARY USING HTML AND CSS

A project report submitted in partial fulfillment of the requirements for the award of the DEOACC 'O' Level IT Course

Submitted by

Dr.G.STEPHEN

(Reg.No- 1050049)

Project Supervisor

Mr. Anil Kumar Shaw,

Scientist –C & Academic Head,

NIELIT –Itanagar Centre.

NATIONAL INSITITUTE OF ELECTRONICS AND INFORMATION TECHNOLOGY-ITANAGAR CENTRE,

ARUNACHAL PRADESH-791110.

MAY 2018

Shri. Anil Kumar Shaw,
Scientist –C & Academic Head,
NIELIT –Itanagar Centre.
Arunachal Pradesh.

CERTIFICATE

This is to certify that the Project work done at **Website Designing for Nielit Itanagar Library Using HTML and CSS** By ***Dr.G.Stephen*** (NIELIT Registration No. ***1050049***) in partial fulfillment of ***NIELIT 'O' Level*** Examination has been found satisfactory. This report has not been submitted for any other examination and does not form part of any other course undergone by the candidate. It is further certifies that he/she has appeared in all the four modules of NIELIT 'O' Level Examination.

Date :

Place: Naharlagun **Signature of the Supervisor**

Declaration

I hereby declare that the dissertation entitled "**Website Designing for Nielit Itanagar Library Using HTML and CSS**" submitted for the ***NIELIT 'O' Level*** IT Software Examination is my original work and the project report has not formed the basis for the award of any degree, associateship, fellowship or any other similar title.

Date:

Place: Naharlagun **Signature of the Student**

Acknowledgements

First of all, I offer up my sincere thanks to the Almighty, for providing me the tremendous strength and courage to undertake this project work successfully. I express my humble sentiments and gratitude to my worthy Supervisor **Shri.Anil Kumar Shaw**, Scientist-C & Academic Head, NIELIT-Itanagar Centre for patiently supervising me, and always guiding me in the right direction. His encouragements, personal and professional guidance, help and immense support throughout this work has led to the successful completion of this project work.

I express my sincere thanks to **Shri. N Debachandra Singh**, Director i/c, **Shri. Gordon Kynsai Nongkynrih ,** Sci-c & Admin i/c, Shri.Rajendra Singh, Finance Officer, **Shri Shammi Sharma**, Jr.Tech.Asst, **Shri. Wanpli Coelho Synnah** Jr.Asst, (Itanagar Centre), **Shri.Anup Kumar,** Sci-c (Tezu Centre), **Shri.Pawan Kumar Mall**, Sci-C, **Shri.Kumar Harsh**, Sr.Tech.Asst (Pasighat centre) for their encouragement at all stages of this work.

My enormous debt of gratitude can hardly be repaid to my beloved wife **Mrs. S. Rosali**, who always inspired me and provided moral support that I needed. I am thankful to my son **Master S.Sterosen Bernard** for his sacrifice, as I was not able to attend him fully during the course of study. As the space limitation does not allow mentioning the name of all the persons who have helped me during this project work, I would like to express my warmest thanks to all of them.

(G.STEPHEN)

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

In 2019 the Internet will be celebrated its 50th anniversary, and the World Wide Web had been in existence for over 25 years. The concepts of computer networks and hypertext on which these technologies rely are only a little older. And yet the speed of development of these technologies, the speed of uptake by companies, and the speed of acceptance by consumers is unlike anything mankind has witnessed. Although both the Internet and the Web are firmly rooted in academic, altruistic Endeavour, there is no doubt that the commercial interests are currently driving much of the technological development. This project aims for contributing to this Endeavour the basic ideas and technologies behind the Internet, and giving the opportunity to design and write Web pages using HTML5 and JavaScript.

1.1.Hypertext

Take a dictionary and observe how its content is linked together. How do you search for the meaning of a word? How can you find another word synonymous with that word? The dictionary is a paper example of a hypertext system. So are encyclopedias, product catalogues, user help books, technical documentation and many other kinds of books. Information is obtained by searching through some kind of index - the dictionary is arranged in alphabetical order, and each word is its own index. Readers are then pointed to the page of any other related information. They can read the information they are interested in without having to read the document sequentially from beginning to end. Hypertext systems allow for non-sequential or non-linear reading. This is the underlying idea of a hypertext system. The result is a multidimensional document that can be read by following different paths through it. In this section we will look into the application of hypertext in computer systems, mainly the World Wide Web hypertext system. The main use of hypertext is in information retrieval applications. The ease of linking different pieces (fragments) of information is the important aspect of hypertext information retrieval. The information can be of various media: it may be fragments of textual documents, structured data from databases, or list of terms and their definitions. Any of these, or a mixture thereof, can make up the contents of a hypertext document.

Therefore, in a hypertext system it is possible to:

- link with a term that represents aspects of the content of a document;
- connect two related documents;
- relate a term to a fragment containing its definition and use; and
- link two related terms.

Such a hypertext system can store a large collection of textual and multimedia documents. Such a hypertext system gives the end-user access to a large repository of knowledge for reading, browsing and retrieving. This is a "database" of sorts, and is the reason why such a hypertext system is called a digital library. The Web started as an extensively large digital library. As it has grown in

popularity, it has offered the possibility of interactive applications and commerce on the Internet, making it much more than a digital library.

1.2 Project Scope

The scope of library Website and Online Library Management System includes:

- Create distinct product users based on their roles and permissions.
- Authenticate users at their login.
- Provide the list of books the users can borrow.
- Facility to reserve books that are available.
- A status page for all users to view books reserved by them.
- Facility to cancel the reservation for a book made earlier.
- A status page for all users to view books borrowed by them, their individual due dates and their individual penalties if any.
- An interface to view and edit the own profile.
- Provide method for adjusting account settings such as passwords.
- Mechanism to reset the password in case user forgets it.
- Providing interface to add or delete books to staffs.

1.3. **The Web as a Digital Library**

The Web as a vast digital library is becoming what is known as a 'Global Information Structure'. It will have a profound effect on how we live, work and play. We shall now look into a few of the social implications of the Web as a digital library and a marketplace.

Different Literacy The hypermedia concept includes not only text and illustrations, but also music, animation, digital movies, video games and computer software. This diversity changes the form of literacy required when using the Web. The literacy needed when listening to music and watching a movie may be different from that used when reading a book. Less literacy may be required with innovative ways of using the digital library. For example, software that reads text aloud can assist people with visual handicaps.

Indeterminate Quality and Value Editors and publishers employing traditional methods of publishing have little to gain from this type of publishing. As digital works can be copied at low costs, stored in almost no space and transported instantly anywhere in the world, writers can be their own publishers. Therefore, the works published are of indeterminate quality and value. Web publishing may provide no evaluation of work published.

Specialist Audiences An article may perhaps interest a group of specialists in a particular field. With the Web, an average reader may browse through the article according to their degree of interest in the field. He or she may not want to be burdened with an additional flood of technicalities,

or perhaps would navigate further to extract more in-depth information to supplement a deeper interest in the field.

Copyright Issues and Ease of Purchasing The ease of copying digital works causes difficulties in protecting copyrights. It may be tempting to make illegal copies rather than finding the rightful owners and paying them a fee. On the other hand, the non-issue of distance and the 24-hour, 365-day activity on the Web means that much can be easily bought through on-line shops. Consumers may come from distant areas or different time zones. With the Web, this market place is open at all times and can serve a very large global region. New technology even allows computational agents to staff the market place rather than people. Therefore, businesses are not constrained by distance or time. v. Sense of Place Despite the irrelevance of distance, an electronic marketplace may be attractive as it goes to the consumers instead of them physically moving to the business environment. Its sense of place is created as an illusion for the benefit of the consumers.

CHAPTER II

WEB TERMINOLOGIES

2.1 Network Protocols

A network protocol is a standard way of regulating data transmission between computers. Just as diplomats adhere to protocols — rules of behavior — when in foreign lands, network communications do the same. They have to obey agreed rules if they are to communicate and 'get on with each other'. After many years of both public and private research and development, two network protocols are now dominant:

TCP (Transaction Control Protocol) and IP (Internet Protocol), together known as TCP/IP. These were actually unlikely protocols to be so widely accepted, as faster, standardized protocols had been agreed upon, but none had the same robustness and extensibility as TCP/IP.) Very often protocols were implemented without any formal acceptance and, because they worked most of the time, they became standards by default. Although TCP/IP is an accepted, de facto standard, work on Internet protocols continue in order to improve communication quality and support the continued growth of the Internet.

There is no dictating authority for the Internet. Without a controlling authority, interim proposals about protocol changes are made by groups of interested individuals and then opened up for discussion. Documents containing the various proposed standards are published as Requests For Comment documents (RFCs). You may see references to a specific RFC as the best description of a protocol!

2.2. Web Application (Webapp)

A web application (or webapp), unlike standalone application, runs over the Internet. Examples of webapps are google, amazon, ebay, facebook and the UCT website. A webapp is typically a 3-tier (or multi-tier) client-server database application run over the Internet and it comprises five components:

HTTP Server: E.g., Apache HTTP Server, Apache Tomcat Server, Microsoft Internet Information Server (IIS), nginx, Google Web Server (GWS), and others. You will learn how to install Apache HTTP and Tomcat web servers in the next chapter.

• HTTP Client (or Web Browser): E.g., Internet Explorer (MSIE), FireFox, Chrome, Safari, and others.

• Database: E.g., Open-source MySQL, MariaDB, Apache Derby, mSQL, SQLite, PostgreSQL, OpenOffice's Base; Commercial Oracle, IBM DB2, SAP SyBase, MS SQL Server, MS Access; and others. You will learn how to install MySQL in the next chapter.

• Client-Side Programs: could be written in HTML Form, JavaScript, VBScript, Flash, and others. You will learn how to writer client-side programs using HTML and JavaScript in this course.

• Server-Side Programs: could be written in Java Servlet/JSP, ASP, PHP, Perl, Python, CGI, and others.

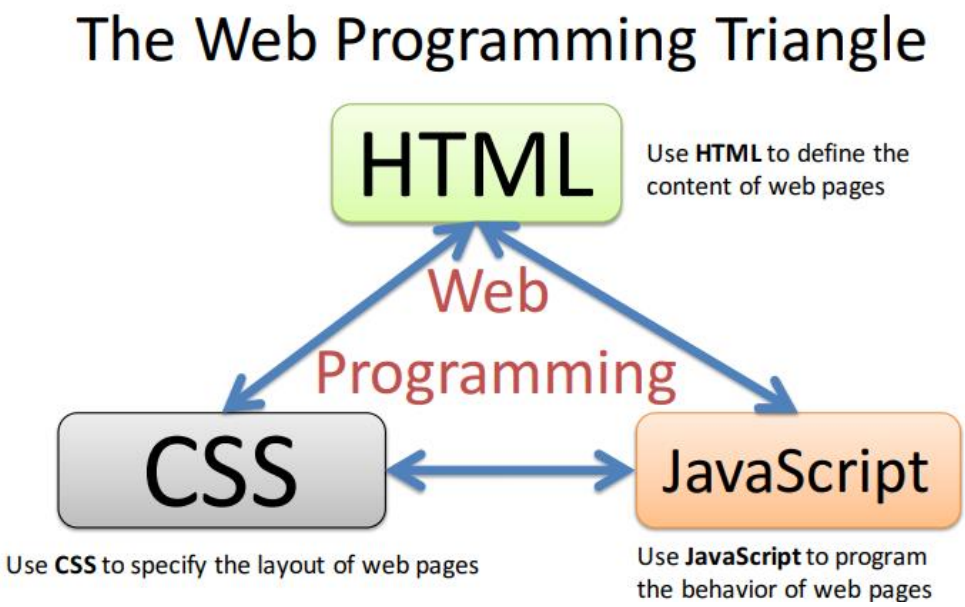
2.3. Uniform Resource Locator (URL)

An URL is needed to locate any resources on the Web. It is an address format that specifies how and where to find a document. The general format is as follows, where the various items in italics must be substituted with part of a real URL, or omitted altogether. http://machine_name:port/path/file_name.file_extension machine_name is either an IP address, for example 137.234.33.89, or a Fully Qualified Domain Name (also known as a DNS name, because Domain Name Servers map between Domain Names and IP addresses), for example, www.apple.com [http://www.apple.com].

In the machine name http is the protocol identifier, while www.apple.com is the resource name. port is the TCP port to connect to; this is an entry point to software on the server; an optional part of a URL. path is a relative file path from the server's document root; the server will start looking for a file in a specific directory and paths are relative to this file_name is the name of the file to be browsed, e.g. welcome file_extension is one of a number of suffixes which, by convention and operating system setup, indicate the type of data contained within the file, e.g. htm,html,txt.Forexample,intheURLbelow,http://www.apple.com/retail/business/jointventure/terms.html ‘terms.html’ is a file with the html extension.

2.4. Hyper Text Markup Language (HTML)

This language provides the format for specifying simple logical structure and links in a hypertext document. As a markup language, special formatting commands are placed in the text describing how the final version should appear. These formatted documents are interpreted by a Web browser which usesthe HTML code to format the page being displayed. Although most professionals use special authoring tools to write HTML documents and to manage sites, developers of e-commerce sites and applications need to know the nitty-gritty detail of HTML, and this is what you will study. HTML has had several versions over the years. "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML 5 version which is an extension to HTML 4.01, and this version was published in 2012 . This course will take you through website creation using HTML5.



2.5. Hypertext Transfer Protocol (HTTP)

HTTP is a network protocol used to retrieve documents from a variety of machines in a minimum amount of time. It was invented by Tim Berners-Lee to support a project in developing a distributed hypertext system. Distributed hypertext requires the retrieval of documents from many different machines. File Transfer Protocol (FTP), which predates the Web, would be too slow for this purpose as it is a connection-oriented protocol that requires a permanent connection to a server, thus requiring a connection-maintenance overhead when accessing different machines.

Therefore, to support browsing, HTTP has the following characteristics: connection-less; a connection is established only for the period of transfer, and the connection need not be maintained after thereafter;

• stateless; the server has no 'history' of client visits (although the implementation of cookies overcomes this);

• comprehensive addressing; diverse files on any HTTP server world-wide can be referenced via URLs;

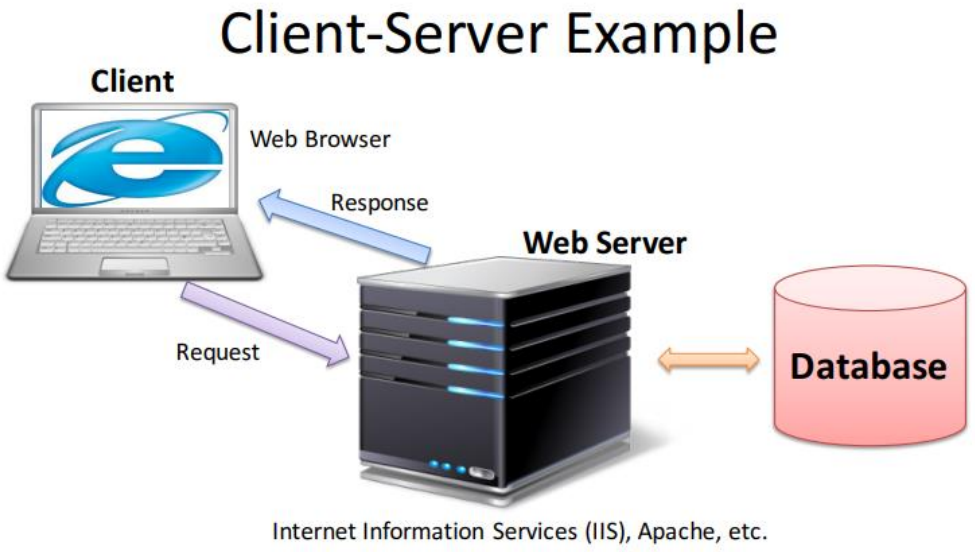
• diverse data; using extensible MIME-types (see later), HTTP servers can supply information of every possible data types; and

• rapid; allows request-response cycles of less than 100 milliseconds HTTP is not mandatory for distributed hypertext; there are other techniques and protocols that can be used to access or transfer information. However, like TCP/IP and HTML, HTTP is ubiquitous and so enables development in e-commerce.

2.6. Client-server Computing Model

A software partitioning paradigm in which a distributed system is split between one or more server tasks which accept requests, according to some protocol, from (distributed) client tasks, asking for information or action. There may be either one centralized server or several distributed ones. This model allows clients and servers to be placed independently on nodes in a network. Client-server computing is mainly about the client computer possessing its own computing power. In the days of mainframes, all the processing power took place on central computers.

The client 'terminals' were little more than a television that could send and receive characters. When microprocessors became available, it was possible to make the terminals more powerful so that they could handle some of the processing. Over time this has meant that mainframes have been replaced by smaller server machines and terminals have been replaced by more powerful client workstations. The client-server model provides a good division of processing power, since the server primarily provides information to the client, which is responsible for interpreting and displaying it. This means that servers do not have to be powerful machines, allowing more people to become service providers.



2.7. Functionality

In the context of the Web, users run client programs (i.e. Web browsers) which provide the following functionality:

- They allow the user to send a request for information to the server.
- They format the request so that the server can understand it.

• They format the response from the server in a way that the user can read it. Server programs carry out the following:

• They receive a request from a client and process the request.

• They respond by sending the requested information back to the client. In summary, the typical functionality of a client-server model is:

• A user, via a web browser (HTTP client), issues a URL request to an HTTP server to start a webapp.

• A client-side program (such as an HTML form) is loaded into client's browser.

• The user fills up the query criteria in the form.

• The client-side program sends the query parameters to a server-side program.

• The server-side program receives the query parameters, queries the database and returns the query result to the client.

• The client-side program displays the query result on the browser.

• The process repeats.

CHAPTER III

HTML BASICS

This language provides the format for specifying simple logical structure and links in a hypertext document. As a markup language, special formatting commands are placed in the text describing how the final version should appear. These formatted documents are interpreted by a Web browser which uses the HTML code to format the page being displayed. Although most professionals use special authoring tools to write HTML documents and to manage sites, developers of e-commerce sites and applications need to know the nitty-gritty detail of HTML, and this is what you will study. HTML has had several versions over the years. "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML 5 version which is an extension to HTML 4.01, and this version was published in 2012 . This course will take you through website creation using HTML5.

3.1. HTML Markup

HTML pages are created by tagging textual information with HTML markup. HTML markup consists of tags, which appear inside angled brackets < and > . An example of an HTML tag is , which causes text to appear in bold. only notes where text should begin to appear in bold, while the tag marks the end of the emboldening. Most HTML tags have a corresponding end tag, which is specified by the name of the tag preceded by the / character.

3.2. Nesting HTML Tags

Text may be both bold and italicized. This is done by using both the and tags. When doing so, it is important to remember not to overlap HTML tags Overlapping tags is a common mistake. Although Web browsers are usually smart enough to work out what is meant, it can lead to problems. Furthermore, for an HTML page to be considered valid HTML, it must contain no overlapping tags.

3.3. Creating HTML Text using Notepad++

This section covers the creation of an HTML page. You will need a Web browser and a text editor. Use HTML: Basics 3 any text editor you wish to, but the following Activity descriptions will use Notepad++. Notepad++ is a free Windows editor that also supports several programming languages. For example, you will notice that HTML keywords are highlighted in different colors.

1. Open your Web browser. This sections' goal is to create a Web document that can be opened with your browser.

2. Open Notepad++. It can be found by selecting Start, then All Programs, then Notepad++.

3. Type the following text into Notepad++: your name and the module number (CSC5003). Save this file as start.txt.

4. Now load start.txt into the browser by dragging start.txt onto your browser.

5. The browser should now display the text contained in start.txt. (If it does not, make sure that you have saved start.txt and that this is the file you are opening).

6. Once you have displayed start.txt, return to Notepad. Add the text "Internet Commerce", and save the file again.

7. Return to the Web browser and reload the document (by using either by using the Refresh or Reload toolbar buttons, or by selecting File/Open once again).

8. If you are able to see the new piece of text, you have successfully used Notepad to create your first Web page.

3.4. Processing Forms

Although forms could simply be used to display information, HTML provides them in order to supply a way for the user to interact with a Web server. The most widely used method to process the data submitted through a form is to send it to server-side software typically written in a scripting language, although any programming language can be used. The figure below outlines the kind of processing that takes place.

1. The user retrieves a document containing a form from a Web server.

2. The user reads the Web page and interacts with the form it contains.

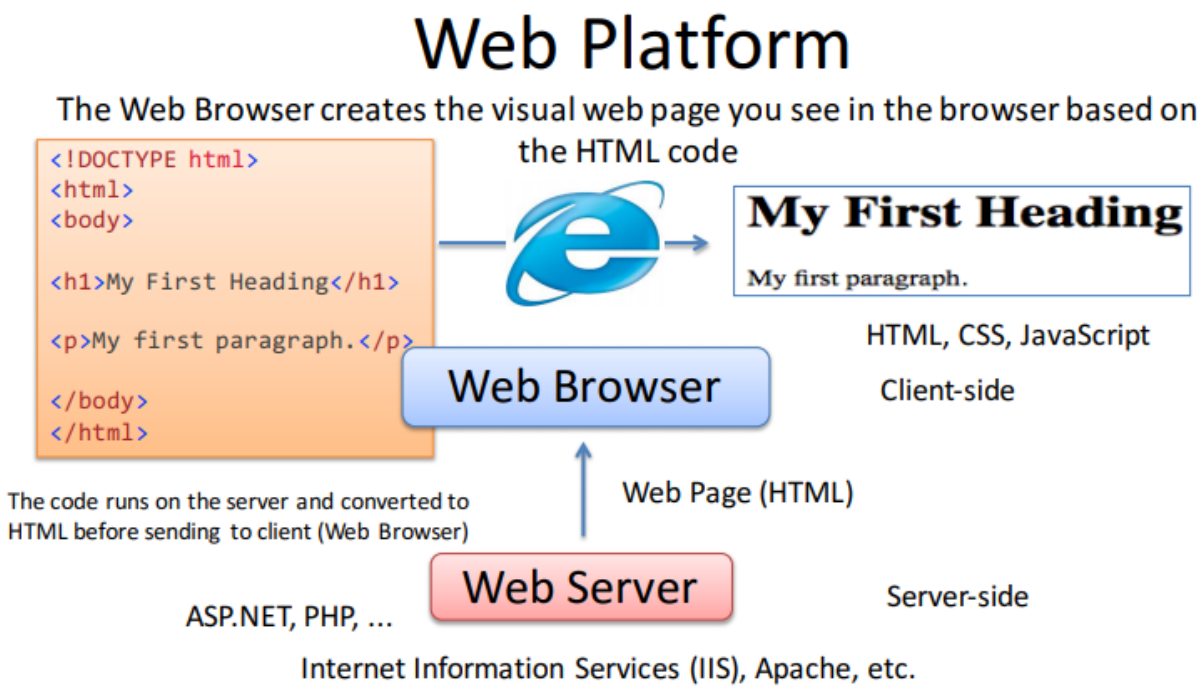
3. Submitting the form sends the form data to the server for processing.

4. The Web server passes the data to a CGI programme.

5. The CGI software may use database information or store data in a server-side database.
HTML Forms

6. The CGI software may generate a new Web page for the server to return to the user.

7. The user reads the new Web document and may interact with it.



3.5. List of used HTML codes

- Basic HTML
- Formatting
- Forms and Input
- Frames
- Images
- Audio / Video
- Links
- Lists
- Tables
- Styles and Semantics
- Meta Info

3.6. Basic HTML Tags and Description

Tag	Description
<!DOCTYPE>	Defines the document type
<html>	Defines an HTML document
<head>	Defines information about the document
<title>	Defines a title for the document
<body>	Defines the document's body
<h1> to <h6>	Defines HTML headings
<p>	Defines a paragraph
** **	Inserts a single line break
<hr>	Defines a thematic change in the content
<!--...-->	Defines a comment
<abbr>	Defines an abbreviation or an acronym
<address>	Defines contact information for the author/owner of a document/article
<b>	Defines bold text
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it
<bdo>	Overrides the current text direction
<big>	Not supported in HTML5. Use CSS instead. Defines big text
<blockquote>	Defines a section that is quoted from another source
** **	Not supported in HTML5. Use CSS instead. Defines centered text
<cite>	Defines the title of a work
<code>	Defines a piece of computer code
<del>	Defines text that has been deleted from a document
<dfn>	Represents the defining instance of a term
****	Defines emphasized text
<font>	Not supported in HTML5. Use CSS instead. Defines font, color, and size for text
<i>	Defines a part of text in an alternate voice or mood
<ins>	Defines a text that has been inserted into a document
<kbd>	Defines keyboard input
<mark>	Defines marked/highlighted text
<meter>	Defines a scalar measurement within a known range (a gauge)
<pre>	Defines preformatted text

<progress>	Represents the progress of a task
<q>	Defines a short quotation
<rp>	Defines what to show in browsers that do not support ruby annotations
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<ruby>	Defines a ruby annotation (for East Asian typography)
<s>	Defines text that is no longer correct
<samp>	Defines sample output from a computer program
<small>	Defines smaller text
<strike>	Not supported in HTML5. Use <del> or <s> instead. Defines strikethrough text
****	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<template>	Defines a template
<time>	Defines a date/time
<tt>	Not supported in HTML5. Use CSS instead. Defines teletype text
<u>	Defines text that should be stylistically different from normal text
<var>	Defines a variable
<wbr>	Defines a possible line-break
Forms and Input	
<form>	Defines an HTML form for user input
<input>	Defines an input control
<textarea>	Defines a multiline input control (text area)
<button>	Defines a clickable button
<select>	Defines a drop-down list
<optgroup>	Defines a group of related options in a drop-down list
<option>	Defines an option in a drop-down list
<label>	Defines a label for an <input> element
<fieldset>	Groups related elements in a form
<legend>	Defines a caption for a <fieldset> element
<datalist>	Specifies a list of pre-defined options for input controls
<output>	Defines the result of a calculation
Frames	
<frame>	Not supported in HTML5. Defines a window (a frame) in a frameset
<frameset>	Not supported in HTML5.

	Defines a set of frames
<noframes>	Not supported in HTML5. Defines an alternate content for users that do not support frames
<iframe>	Defines an inline frame
Images	
<img>	Defines an image
<map>	Defines a client-side image-map
<area>	Defines an area inside an image-map
<canvas>	Used to draw graphics, on the fly, via scripting (usually JavaScript)
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content
<picture>	Defines a container for multiple image resources
<svg>	Defines a container for SVG graphics
Audio/Video	
<audio>	Defines sound content
<source>	Defines multiple media resources for media elements (<video>, <audio> and <picture>)
<track>	Defines text tracks for media elements (<video> and <audio>)
<video>	Defines a video or movie
Links	
<a>	Defines a hyperlink
<link>	Defines the relationship between a document and an external resource (most used to link to style sheets)
<nav>	Defines navigation links
Lists	
** **	Defines an unordered list
** **	Defines an ordered list
** **	Defines a list item
<dir>	Not supported in HTML5. Use instead. Defines a directory list
<dl>	Defines a description list
<dt>	Defines a term/name in a description list
<dd>	Defines a description of a term/name in a description list
<menu>	Defines a list/menu of commands
<menuitem>	Defines a command/menu item that the user can invoke from a popup menu
Styles and Semantic	
<style>	Defines style information for a document
** **	Defines a section in a document

****	Defines a section in a document
<header>	Defines a header for a document or section
<footer>	Defines a footer for a document or section
<main>	Specifies the main content of a document
<section>	Defines a section in a document
<article>	Defines an article
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<summary>	Defines a visible heading for a <details> element
<data>	Links the given content with a machine-readable translation
Meta Info	
<head>	Defines information about the document
<meta>	Defines metadata about an HTML document
<base>	Specifies the base URL/target for all relative URLs in a document
<basefont>	Not supported in HTML5. Use CSS instead. Specifies a default color, size, and font for all text in a document

CHAPTER IV

INTRODUCTION TO CASCADING STYLE SHEETS

4.1. CSS

There is no format to follow for teaching the aesthetics of style - most people, though, can recognize something that follows a classical design. But some things can be said about the style of a website. For instance, when Web pages belong to the same website, each page should have a consistent look in order to provide familiarity for the user. Style sheets (sometimes referred to as templates) are used in desktop publishing to provide consistency when formatting text. The format applied by the stylesheet could be to indent every first line of a paragraph by 2cm, insert a page break at the end of every chapter, and so on. Naturally, due to multimedia, Web pages not only have to consider text formatting, but also visual and sound presentation, and various multimedia formats in general. Before we continue, let us briefly discuss the advantages and disadvantages of using style sheets.

4.2. Advantages of Style Sheets

1. Multiple Styles - A single document can be presented in multiple styles by using multiple style sheets.

2. Re-styling - The use of style sheets (which are separate to the HTML files) allows the quick re- styling of any document, without modifying the original HTML.

3.Document maintenance - The ability to re-style many documents allows usto easily make changes to the appearance of many Web pages without separately editing each one.

4. Consistency - Style sheets guarantee consistency throughout website.

5. Optimal file size - The smaller the files the faster the download. Using style sheets can help minimize file sizes, since, for example, every < font > tag, is defined in one place in a style sheet, rather than in multiple places in the HTML file.

6. Style and structure - When first developed, HTML was only concerned with document markup and not with the document's formatting. This eventually changed, with more and more functionality being added to HTML to allow for formatting. With the introduction of style sheets, the HTML document is again concerned only with structural document markup — all formatting is now placed in the style sheet.

4.3. Minimum Hardware and Software requirements

Below is a list of the minimum Hardware and Software requirements to access the basic website content.

Operating System:

- Windows 7, Windows 8 or Windows 10
- Mac OSX 10.8, 10.9, 10.10 or 10.11

Hardware:

- Processor (CPU) with 2 gigahertz (GHz) frequency or above
- A minimum of 2 GB of RAM
- Monitor Resolution 1024 X 768 or higher
- A minimum of 20 GB of available space on the hard disk
- Internet Connection Broadband (high-speed) Internet connection with a speed of 4 Mbps or higher
- Keyboard and a Microsoft Mouse or some other compatible pointing device
- Sound card
- Speakers or headphones
- Strongly Recommended - Microphone and Webcam

Browsers:

- Chrome* 36+
- Edge* 20+
- Mozilla Firefox 31+
- Internet Explorer 11+ (Windows only)
- Safari 6+ (MacOS only)

*Google Chrome version 42+ and Microsoft Edge do not support NPAPI-type plug-ins, including Java plug-ins and many media browser plug-ins. Blackboard does not support these browsers for use with media that require NPAPI plug-ins for viewing.

Browser Configuration:

Browser must be configured as follows:

- **Strongly Recommended** to trusted sites.
- JavaScript must be enabled
- Cookies must be enabled.
- Pop-up windows must be enabled.

Software:

- Microsoft Office--a free copy of Office 265
- Java -- To view and interact with all available blackboard applications.
- Real Player--for audio/video content.
- Quick Time--for audio/video content.
- PowerPoint viewer--PowerPoint Viewer lets you view full-featured presentations created in PowerPoint 97 and later versions.
- Adobe Shockwave--for Macromedia content.
- Adobe Acrobat Reader --for PDF files Adobe® Reader® software is the global standard for electronic document sharing. It is the only PDF viewer that can open and interact with all PDF documents. Use Adobe Reader to view, search, digitally sign, verify, print, and collaborate on Adobe PDF files.
- Adobe Flash Player--Flash Player allows you to enjoy content with video, graphics and animation.

CHAPTER V

IMPORTANCE OF WEBSITE IN A LIBRARY

5.1. Library Portal

The Library Portal is a gateway to its resources and services. The purpose of an information gateway of this type is to help users discover high quality, relevant web-based information quickly and effectively. The portal besides providing information about the staff, collection and services, allows access to the OPAC, and provides direct link to e resources on public sites. User interaction is encouraged through a number of e-mail links. Website performs the role of an effective help desk and promotional tool for libraries. As practiced in many college libraries abroad, audio tours of libraries, detailed maps and brochures serves as a great help for libraries. A user friendly environment should make the users more confident and efficient library users without requiring help from the library staff.

Importance of Website for Libraries.

5.2 Roles the Library Web Site Can Play

The library Web site can and by default does play a variety of roles. First, a library Web site often serves the role of a library workstation, both for the users and for the librarians serving them. This necessitates a presentation and organization that allows users to know all that the library has to offer electronically, and in a way that makes sense. It is a tool that will help to speed up or slow down the reference librarians' work in assisting patrons to find information. It is a tool that will help or hinder the user in expanding research or in finding the answer to a very simple question, such as how determine if the library has a particular book and where it might be found.

In the role of library workstation, a library Web site serves as a delivery mechanism for databases, electronic texts and journals, and often for the library catalog. In delivering these resources, the Web necessitates dialog between public and technical service librarians to determine how and where to represent access to all this information. We saw a revolution in the interaction between these units when OPACs first came on the scene, and we are now witnessing a further entwining of these relationships.

Should electronic journals be listed on the Web site? Should reference librarians attempt to categorize them? Should categorization be done instead by catalogers who will assign call numbers and standardized subject headings? Can our systems people create mechanisms for browsing these titles if, in fact, they are all simply listed in the catalog? Can we create links between catalog records, the journals themselves, and the sources that index them? Do we continue to single out and bring to the Web site electronic databases ... or again ask the user to rely on the catalog for finding them? We might have one answer when we have access to 20 such electronic resources, and another answer when we have access to 120 or 520.

The very core of what our OPACs are now to accomplish and what types of materials are to be represented in them is coming into question and is causing us to consider the need for alternative and supplementary databases to track the information. Next, a library Web site is a way of making internal resources or products available. These might be digitized copies of special collections, including manuscripts, images, or even locally created databases. They might be products, such as instructional tools, class assignments, guides, and finding aids. The option of placing electronic reserve material in a copyright secure environment is being undertaken by many libraries. The Web has allowed librarians to find new roles as information generators, as well as to continue the functions of information gathering, organization, and access.

Additionally, a library Web site can become an agent for archiving and retaining information that comes and goes on other sites. Some consortia, such as the CIC (Committee on Institutional Cooperation or the Big-Ten-plus-one) libraries, are looking at ways of retaining copies of free electronic journal sites that are not maintained by any publisher. Such activity requires agreements on copyright and access rules, beyond the technical issues of gathering and ensuring complete runs of journals. Other projects are being developed between libraries, consortia, and publishers for access to proprietary information that a library has in fact contracted.

How can libraries ensure availability of materials without "housing" them directly? Sometimes license agreements can include guarantees of efforts by publishers, and sometimes library server space can archive material the publisher doesn't care to retain. My library maintains copies of encyclopedias from the 1800s, which are often called upon for academic research. However, are ever-evolving editions of encyclopedias on the Web also being archived as articles are revised, dropped, and changed? Do libraries play a role in retaining records of this sort? Importantly, the library Web site is also a window to and a component of the Web at large. Libraries are making use of the great wealth of resources available on the Web at no cost and are examining the reliability, accuracy, and completeness of these vast resources.

Where trust in a resource is engendered, and as it relates to the needs of a library's users, these resources are being incorporated into library sites as part of catalogs or in separate lists or databases created by bibliographers and reference librarians. As a component of the Web at large, libraries are increasing their user base by making services and resources available to a potentially worldwide audience. Policy decisions may have to be made to determine the extent of services and the definitions of users in this easy-to-access environment, where internal resources may be stretched. Additionally, libraries might wish to consider what uniqueness they have to offer in this vast network when placing priorities on what is to be accomplished on the Web site. Finally, an all-important function of the library's Web site is to serve as a communication tool for a library and its users. It is a way to advertise when the physical presence is available and where it is located, as well as to tell users who works there and what sort of services those people provide.

A library site can let people know organizational structures and missions. It can tell people how to use interlibrary loan, when there are research workshops, or what exhibits are on

display. It can even encourage them to come to book sales or other fundraisers. Even better, the Web site can allow us to hear from the users through interactive forms, chat rooms, and e-mail links. But we can also do so through examination of our log files to learn how many people visit us, where they come from, what they use, and what sort of problems they encounter. A colleague of mine once eloquently stated that libraries are not becoming virtual, the users are. And while the Web is a great instrument in causing that to happen, it can also be the tool that keeps us in touch.

CHAPTER VI
WEBSITE DESIGNING HTML CODE AND CSS CODES

6.1. Home Menu

6.1.1. HTML Code for Home Menu

```
<!DOCTYPE html>
<html>
<head>
<title></title>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
<div class="bgimage">
<div class="menu">
<div class="leftmenu">
<h4> NIELIT ITA LIBRARY</h4>
</div>
<div class="rightmenu">
<ul>
<li id="firstlist"> HOME </li>
<li> <a href="ABOUT.html">ABOUT </a> </li>
<li> <a href="services.html"> SERVICES </a> </li>
<li> <a href="blog.html">BLOGS </a> </li>
<li> <a href="CONTACT.html">CONTACT </a> </li>
<li> <a href="Regform.html" class="header-signup"> SIGNUP </a> </li>
</ul>
</div>
</div>
<div class="text">
<h4>National Institute of Electronics and Information Technology</h4>
<h1>ITANAGAR LIBRARY - Arunachal Pradesh</h1>
<h3>Small library but Hybrid Library</h3>
<button id="buttonone"> Like Share </button>
<button id="buttontwo"> Subscribe </button>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<!-- The form -->
<form class="example" action="action_page.php">
<input type="text" placeholder="Search.." name="search">
<button type="submit"><i class="fa fa-search"></i></button>
```

```
</form>
</div>
</div>
</body>
</html>
```

6.1.2. Css sheet for Home Menu

```
* {
margin: 0px;
padding: 0px
}

.bgimage{
background-image: url('LibPhoto.jpg');
background-size: 100% 110%;
width: 100%;
height: 100vh;
}
menu{

width: 100%;
height: 100%;
background-color: inherit;(0.5,0.5,0.5,0.5);
}

.leftmenu {
width: 25%;
line-height: 100px;
float: left;
/*background-color: yellow;*/
}

.leftmenu h4{
padding-left: 70px;
font-weight: bold;
color: green;
font-size: 25px;
font-family: 'montserrat', sans-serif;
}
```

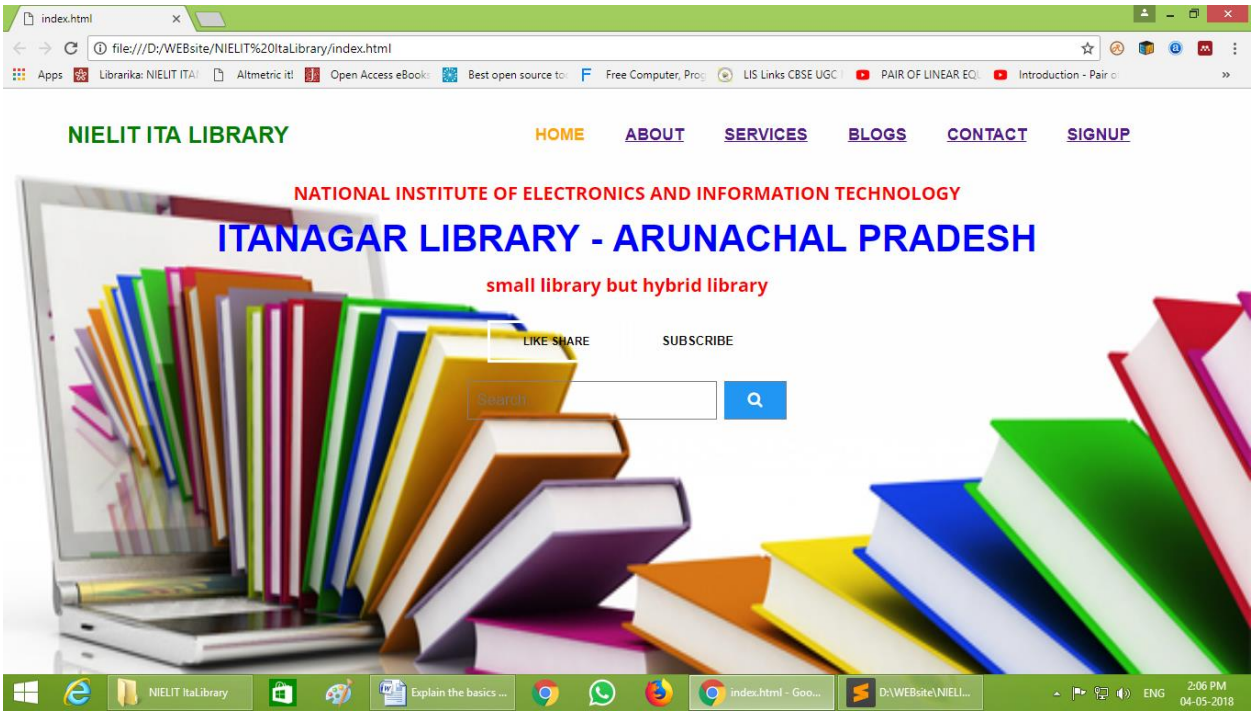
```
.rightmenu{
width: 75%;
height: 100%;
float: right;
/*background-color: red;*/

}
.rightmenu ul{
margin-left: 200px;
}

.rightmenu ul li {
font-family: 'montserrat', sans-serif;
display: inline-block;
list-style: none;
font-size: 18px;
color: green;
font-weight: bold;
line-height: 100px;
margin-left: 40px


}

#firstlist{
color: orange;
}
.rightmenu ul li:hover{
background: rgba(105, 105, 135, .5);
color: orange;
}

.text {
width: 100%;
margin-bottom: 185px;
text-transform: uppercase;
text-align: center;
color: red;
}
```

```
.text h4 {
font-size: 20px;
font-family: 'open sans', sans-serif;
text-transform: uppercase;

}

.text h1 {
font-size: 40px;
font-family: 'montserrat', sans-serif;
color: blue;
font-weight: 400px;
margin:14px 0px;
}


.text h3 {
font-size: 20px;
font-family: 'open sans', sans-serif;
text-transform: lowercase;
}

#buttonone {
background-color: transparent;
border: none;
font-size: 12px;
font-weight: bold;
text-transform: uppercase;
line-height: 40px;
width: 150px;
font-family:'montserrat', sans-serif;
margin-top: 25px;
border: 3px solid white;

}

#buttontwo {
background-color: transparent;
border: none;
```

```
font-family:'montserrat', sans-serif;
text-transform: uppercase;
font-weight: bold;
line-height: 40px;
border: 3px solid white;
width: 150px;

}
.example {
background-color: transparent;
font-size: 30px;
font-family: 'open sans', sans-serif;
margin:14px 0px;
}
* {
box-sizing: border-box;
}

/* Style the search field */
form.example input[type=text] {
padding: 10px;
font-size: 17px;
border: 1px solid grey;
float: center;
width: 20%;
background: transparent;
}

/* Style the submit button */
form.example button {
float: center;
width: 5%;
padding: 10px;
background: #2196F3;
color: white;
font-size: 17px;
border: 1px solid grey;
border-left: none; /* Prevent double borders */
cursor: pointer;
text-align: center;
```

```
}

form.example button:hover {
background: #0b7dda;

}

.background  {
background-image: url(LS.png);
background-repeat: no-repeat;
}
```

6.1.3.Webpage Output of Home Menu



6.2.About Menu

6.2.1.HTML Code for About Menu

```
<!DOCTYPE html>
<html>
<head>
    <title>NIELIT ITA LIBRARY SERVICES</title>
    <link rel="stylesheet" type="text/css" href="about.css">
</head>
<body>
    <section class="hero_image">
            <div class="hero_wrapper">
                    <header> <h1><p align="justify">       ABOUT NIELIT - ITANAGAR LIBRARY</p></h1></header>
                    <p> <h3> The NIELIT Itanagar Centre is aimed to facilitate the youth of Arunachal Pradesh to have easy access to education and training in the field of Computer Science and Information Technology in a well equipped state-of-the-art training infrastructure with qualified & trained faculty members, resulting in generation of quality and employable manpower. Books are acquired according to the requirement of the student, requisition obtained from faculty members and other staff on regular basis. There is a library committee for administering, organizing and maintaining the Library and Library services with the approval of Director In-charge. Library holdings 1689 books, 500 ebooks. Library Subscribed 5 daily Newspaers and 06 Magazines. Library Timing 8.30am to 5.30 pm </h3> </p>
            </div>
    </section>
    <section class="OPAC">
            <div class="product_list_item">
                    <img  class="img-responsive" src="opac.png" alt="">
                    <h2 class="Library Web Opac"> OPAC </h2>
                    <p align="justify"> OPAC is the Online Public Access Catalogue or, in other words, the library catalogue. It is an online database of all of the resources held in the library.It lists the number of the items, whether they are in the library or out on loan, and their call number. You can search OPAC to locate books in the library. <br> OPAC is web-based and will work on world wide.This Library Using Librariaka ILS... <a href="https://nielititanagar.librarika.com/search/alphabetical">click here</a> to go to NIELIT - Itanagar OPAC.</p>
            </div>
    </section>
```

```
    <section class="InstitutionalRepository">
        <div class="product_list_item">
            <img class="img-responsive" src="repository.png" alt="">
            <h2 class="product_heading"> Institutional Repository </h2>
            <p align="justify">An institutional repository is an archive for collecting,
preserving, and disseminating digital copies of the intellectual output of an institution,
particularly a research institution.The content of an institutional repository depends on the focus
of the institution. Higher education institutions conduct research across multiple disciplines, thus
research from a variety of academic subjects.<a
href="http://192.168.1.106:8383/greenstone3/library">click here</a> to go to NIELIT - Itanagar
Institutional Repostitory.</p>
        </div>
    </section>
    <section class="Ebook">
        <div class="product_list_item">
            <img class="img-responsive" src="ebook.png" alt="">
            <h2 class="product_heading"> Ebook Accessing Portal </h2>
            <p align="justify">This  Library provides access to several collections of full-
text ebooks covering a wide range of Computer Science and electronics areas. An eBook is an
electronic version of a traditional print book that can be read by using a personal computer or by
using an eBook reader.For collecting and managing  ebook using  Calibre Software... <a
href="http://192.168.1.106:443/">click here</a> to go to NIELIT - Itanagar ebook.</p>
        </div>
    </section>
    <section class="ELMS">
        <div class="product_list_item">
            <img class="img-responsive" src="elms.png" alt="">
            <h2 class="product_heading"> eLMS </h2>
            <p align="justify"> <a href="https://nielititanagarlibrary.neolms.com/">click
here</a> to go to NIELIT - Itanagar elms.
            </p>
        </div>
    </body>
</html>
```

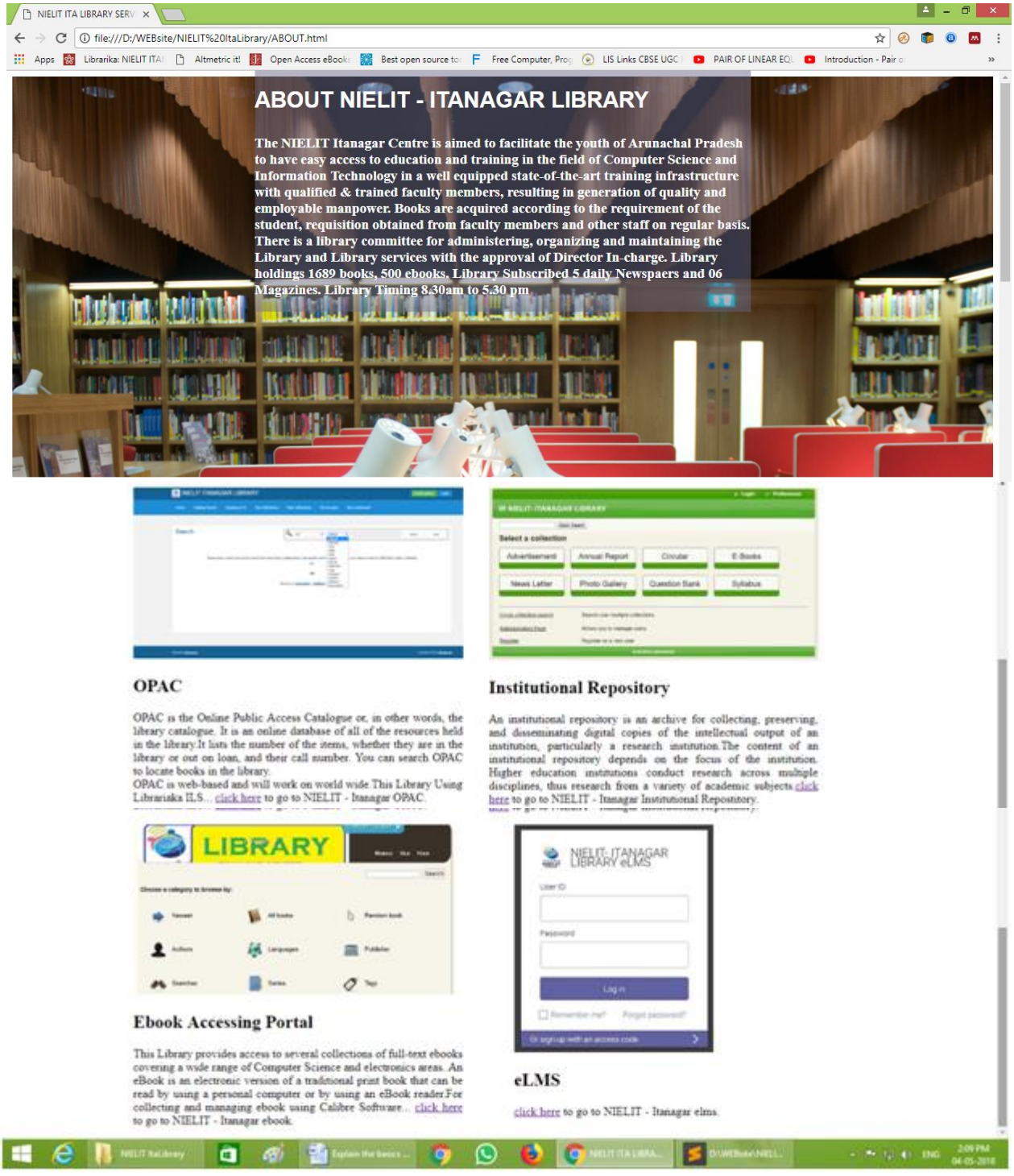
6.2.2. Css sheet for About Menu

```
*body{
    font-family: 'montserrat', sans-serif;
}
.img-responsive{
    max-width: 100%
}

.hero_image{
    background: url(library.jpg) no-repeat;
    height: 85vh;
    background-size: cover;
    position: center;

}
.hero_wrapper{
    color: white;
    background: rgba(105, 105, 135, .5);
    position: absolute;
    top: 25%;
    left: 50%;
    transform: translate(-50%,-50%);
}
.hero_wrapper h1{
    font-size: 30px;
    font-family: 'montserrat', sans-serif;
    color: white; position: center;
    font-weight: 200px;
    margin:12px 0px;
}
.text h2{
    font-size: 30px;
    font-family: 'montserrat', sans-serif;
    color: blue;
    font-weight: 200px;
    margin:12px 0px;
}
.text h3 {
    font-size: 30px;
```

```
    font-family: 'montserrat', sans-serif;
    color: blue;
    font-weight: 200px;
    margin:12px 0px;
}
.text h4 {
    font-size: 10px;
    font-family: 'open sans', sans-serif;
    text-transform: uppercase;

}
.OPAC {
    width: 80%;
    margin-top: 1rem;
    margin-left: auto;
    margin-right: auto;
}
.InstitutionalRepository {
    width: 80%;
    margin-right: 1rem;
    margin-left: auto;
}
.Ebook {
    width: 80%;
    margin-left: auto;
    margin-right: auto;
}
.ELMS{
    width: 50%;
    margin-right: 1rem;
    margin-left: auto;
}

.product_list_item {
    float: left;
    width: 42%;
    margin-left: 1rem;
}
.product_list_item:first-of-type{
    margin-left: 35px;
```

```
}

.h1{
    font-size: 40px;
    font-family: 'montserrat', sans-serif;
    color: blue;
    font-weight: 400px;
    margin:14px 0px;
}
.footer{
    height: 70px;
    width: 100%;
    color: #fff;
    background-color: #004d40;
    text-align:center;
    position: absolute;
    bottom: 0;

}
```

6.2.3. Webpage Output of About Menu.



6.3.Services Menu

6.3.1. HTML code for Service Menu

```
<!DOCTYPE html>
<html>
<head>
    <title>NIELIT ITA LIBRARY SERVICES</title>
    <link rel="stylesheet" type="text/css" href="serstyle.css">
</head>
<body>
    <div class="leftmenu">
                        <h4> NIELIT ITA LIBRARY</h4>
        </div>
    <div class = "background">
    <div class="nav">
        <a class="a active" href="#">Library Services </a>
        <a class= "b" href="#"> For Staff</b>
        <a class= "c" href="#"> For Students</b>
        <a class= "d" href="#"> For Vistors </b>
    </div>
    <div class="loginbox">
        <img src="login-blue-hi.png" class="login-blue-hi">
        <form>
            <p>user name </p>
            <input type="text" name="User name" placeholder="Enter user name">
            <p>password</p>
            <input type="Password" name="" placeholder="Enter password">
            <input type="submit" name="" value="Login">
            <a href="#"> Lost your Password? </a> <br>
            <a href="#"> Dont have an Account? </a>
        </form>
    </div>
</body>
</html>
```

6.3.2.CSS sheet for Service menu

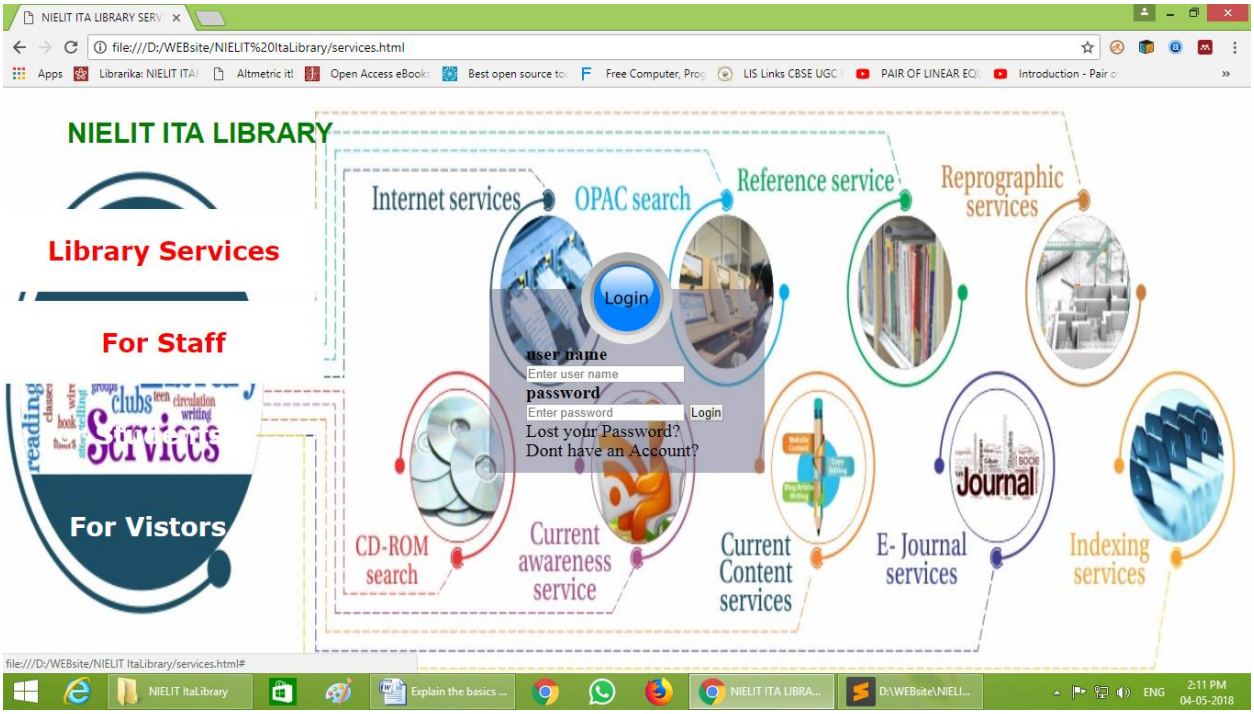
```
*{
    margin: 0px;
    padding: 0px;
}
html, body {
    width: 100%;
    height: 100%;
    font-size: 14px;
    font-weight: normal;
}
.leftmenu {
    width: 33%;
    line-height: 100px;
    float: left;
/*background-color: yellow;*/
}
.leftmenu h4{
    padding-left: 70px;
    font-weight: bold;
    color: green;
    font-size: 30px;
    font-family: 'montserrat', sans-serif;
}
.background{
    background-image: url('LS.jpg');
    background-size: 100% 110%;
    width: 100%;
    height: 100vh;
}
.nav {
    width: 350px;
    height: auto;
    position: absolute;
    top: 20%;
    left: 0;
}
.nav a{
    text-decoration: none;
    font-size: 28px;
```

```
        font-family: verdana;
        font-weight: bold;
        height:90px;
        margin: 5px 0;
        float: left;
        display: inline-block;
        color: white;
        line-height: 90px;
        text-align: center;
        transition: 300ms;
    }
    a.active, .nav a:hover {
        background-color: white;
        color: red;
        width: 100%;
    }
    .a {
        width: 95%
    }
    .b {
        width: 85%
    }
    .c {
        width: 80%
    }
    .d {
        width: 90%
    }
    .content {
        position: absolute;
        width: 930px;
        min-height: 500px;
        height: auto;
        background-color: transparent;
        right: 0;
        top: 10%;
    }
    .body {
        margin: 0;
        padding: 0;
```

```
        background: url(login-blue-hi);
        background-size:cover;
        background-blend-mode: center;
        font-family: sans-serif;
    }
    .loginbox {
        width: 300px;
        height: 200px;
        background: rgba(105, 105, 135, .5);
        color: #fff;
        top: 50%;
        left: 50%;
        position: absolute;
        transform: translate(-50%, -50%);
        box-sizing: border-box;
        border: 10px green;
        border-color: green;
        padding: 60px 40px;
    }
    .login-blue-hi{
        width: 100px;
        height: 100px;
        border-radius: 50%;
        position: absolute;
        top: -40px;
        left: calc(50% - 50px);
    }
    .loginbox p{
        margin: 0%;
        padding: 0%;
        font-weight: bold;
    }
    .loginbox input {
        width: 100%
        margin-bottom: 20px;
    }
    .loginbox input [type="text"], input [type="password"] {

        border: none;
        border-bottom: 1px solid #fff;
```

```
        background: transparent;
        outline: none;
        height: 100px;
        color: #fff;
        font-size: 20px;
    }
    .longinbox input [type="submit"] {
        border: red;
        outline: green;
        background: #fb2525;
        color: #fff;
        font-size: 18px;
        border-radius: 30px;
    }
    .loginbox input [type="submit"]: hover {
        cursor: pointer;
        background: #ffc107;
        color: #000;
    }
    .loginbox a{
        text-decoration: none;
        font-size: 20px;
        line-height: 20px;
        color: black;
    }
    .loginbox a:hover{
        color: #ffc107;
    }
```

6.3.3. Webpage Output of Service Menu



6.4.Blogs Menu

6.4.1. HTML code for Blogs Menu

```
<!DOCTYPE html>
<html>
<head>
  <title>ItaLibBlog</title>
  <link rel="stylesheet" type="text/css" href="Blog.css">
</head>
<body>
   <header>
     <h1> Nielit-Ita-Library 24x7 </h1>
     <nav>
       <ul>
         <li> <a href="http://nielititalibrary.wixsite.com/nielititalibrary"> Website </a></li>
         <li> <a href="https://nielititanagar.librarika.com/search"> OPAC </a> </li>
         <li> <a href="https://nielititalibrary.wixsite.com/academic-depository"> Academic
Depository </a> </li>
         <li> <a href="http://winksite.com/xhtml/ms_main?susid=52662"> Mobile Site </a> </li>
       </ul>
       <a href="Regform.html" class="header-signup"> signup </a>
     </nav>
   </header>
   <main>
     <section class="index-banner">
       <div class="vertical-center">
         <h2> Small Library But Hybrid Library </h2>
         <h1> Knowledge Park of Computer Science and IT </h1>
         </div>
     </section>
     <div class="Wrapper">
        <section class="index-links">
          <a href="http://nielit.gov.in/">
            <div class="index-boxlink-square">
              <h3> About NIELIT </h3>
            </div>
            </a>
            <a href="http://nielititalibrary.wixsite.com/nielititalibrary">
              <div class="index-boxlink-rectancle">
                <h3> Library Website </h3>
```

```
          </div>
        </a>
        <a href="https://nielititanagar.librarika.com/search">
          <div class="index-boxlink-square">
            <h3> OPAC </h3>
          </div>
        </a>
        <a href="http://192.168.1.106:8383/greenstone3/library">
          <div class="index-boxlink-rectancle">
            <h3>Institute Repository </h3>
          </div>
        </a>
        <a href="http://192.168.1.106:443/">
          <div class="index-boxlink-square">
            <h3> E-Books </h3>
          </div>
        </a>
        <a href="http://stephenlisp.webs.com">
          <div class="index-boxlink-square">
            <h3> Librarian </h3>
          </div>
        </a>
      </section>
    </div>
    </main>
    <div class="Wrapper">
      <footer>
        <ul class="footer-links-main">
        <li> <p>Library Online </p> <a href="https://nielititanagar.librarika.com/search">
Catalog search </a> </li>
        <li> <a href="https://nielititanagar.librarika.com/search/newCollections"> New Arrivals
</a> </li>
        <li> <a href="https://nielititanagar.librarika.com/search/topCollections"> Top
Collections </a> </li>
        <li> <a href="https://nielititanagar.librarika.com/search/alphabetical"> A-Z Database
</a> </li>
        <li> <a href="https://nielititanagar.librarika.com/libraries/contact"> Ask Librarian </a>
</li>
        </ul>
        <ul class="footer-links-main">
```

```
        <li><p> Students Links</p> <a href="https://www.jagranjosh.com/current-affairs">
Current Affairs </a> </li>
        <li> <a href="http://student.nielit.in/"> Student.Nielit </a> </li>
        <li> <a href="https://www.india.gov.in/my-government/schemes"> Students Welfare
</a> </li>
        <li> <a href="https://www.naukri.com/"> Job Search </a> </li>
        <li> <a href="http://nielit.gov.in/nielit-news"> Nielit News </a> </li>
        </ul>
        <ul class="footer-links-main">
        <li> <p> E-Learning </p> <a href="https://onlinecourses.nptel.ac.in/"> NPTEL </a>
</li>
        <li> <a href="https://swayam.gov.in/"> Swayam </a> </li>
        <li> <a href="https://www.khanacademy.org/"> Khan Academy </a> </li>
        <li> <a href="https://www.edx.org/"> edx </a> </li>
        <li> <a href="https://www.coursera.org/"> Coursera </a> </li>
        </ul>
        <ul class="footer-links-main">
        <li> <p> Contact </p> <p> Dr.G.Stephen, </p></li>
        <li> <p> Assistant Librarian, </p> </li>
        <li> <p>NIELIT-Itanagar Library, </p></li>
        <li> <p>Naharlagun,</p>  </li>
        <li> <p>Arunachal Pradesh.</p></li>
        </ul>
       <div class="footer-sm">
        <a href="https://www.facebook.com/nielititanagarlibrary/">
          <img src="fb.png" alt="facebook icon">
        </a>
        <a href="https://twitter.com/nielititalib?lang=es">
          <img src="Twitter_Icon.png" alt="Twitter icon">
        </a>
        <a href="https://www.youtube.com/channel/UCuaLHgjqMUbXIMglar0UTcQ">
          <img src="YouTube.png" alt="You Tube icon">
        </a>
        <a href="https://www.trepup.com/nielititanagarlibrary/timeline">
          <img src="trepup.jpg" alt="Trepup icon">
        </a>
      </div>
     </footer>
    </div>
</body>
```

```
</html>
```

6.4.3.CSS sheet for Blog menu

```
*body{
  font-family: 'montserrat', sans-serif;
  text-decoration: none;
  padding: 20px;
}
header{
  background-color: #fff;
  width: 100%
  height:100px;
}
header h1{
  font-family: sans-serif;
  font-size: 24px;
  font-weight: 900;
  color: #111;
  text-transform: uppercase;
  display: block;
  text-align: center;
  padding: 10px 0;
}

header nav ul{
  display: block;
  margin: 0 auto;
  width: fit-content;

}
header nav ul li{
  display: inline-block;
  float: left;
  list-style: none;
  padding: 0 10px;
}
header nav ul li a {
  font-family: sans-serif;
  font-size: 16px;
  color: #111;
```

```
  text-transform: uppercase;
}
header .header-signup {
  display: none;
}
 @media only screen and (min-width: 1000px){
  header h1{
  margin:31px 0;
  text-align: left;
  line-height: 28px;
  padding: 0px 20px 0px 30px;
  border-right: 5px solid #111;
  float: left;
}
header nav ul{
  margin: 20px 0px 0px 20px;
  float:left;
}
header nav ul li a {
  line-height: 50px;
}
header .header-signup {
  display: block;
  font-family: sans-serif;
  font-size: 16px;
  color: #111;
  text-transform: uppercase;
  line-height: 30px;
  border: 2px solid blue;
  float: right;
  margin-right: 30px;
  margin-top: 30px;
  padding: 0 20px;
}
}
/*INDEX*/
.index-banner{
  width: 100%;
  height: 100vh;
  background-image: url('lib Photo.jpg');
```

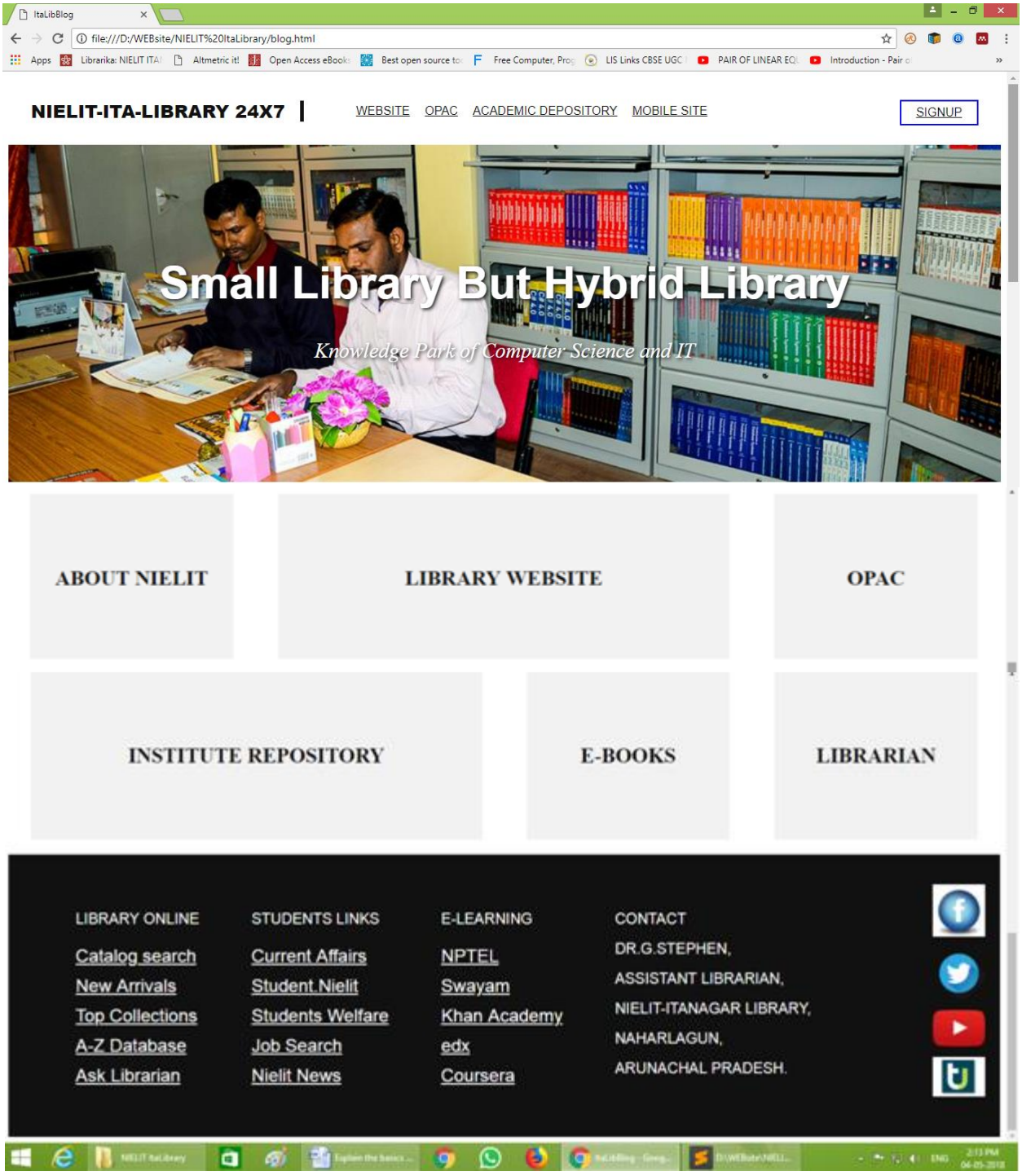
```
  background-repeat: no-repeat;
  background-position: center;
  background-size: cover;
  display: table;
}
.vertical-center{
  display: table-cell;
  vertical-align: middle;
}
.index-banner h2{
  font-family: sans-serif;
  font-size: 60px;
  font-weight: 600;
  line-height: 40px;
  color: #fff;
  text-align: center;
  text-shadow: 2px 2px 8px #111;
}
.index-banner h1{
  font-family: Times New Roman;
  font-size: 28px;
  font-weight: 100;
  line-height: 40px;
  font-style: italic;
  color: #fff;
  text-align: center;
  text-shadow: 2px 2px 8px #111;
}
.index-links h1{
  font-family: Times New Roman;
  font-size: 28px;
  font-weight: 100;
  line-height: 40px;
  font-style: italic;
  color: #fff;
  text-align: center;
  text-shadow: 2px 2px 8px #111;
}
.index-links div {
  margin: 16px 16px 0px;
```

```
  width: calc(100%-32px);
  height: 100px;
  background-color: #f2f2f2;
}
.index-links div h3{
  font-family: Times New Roman;
  font-size: 28px;
  font-weight: 300px;
  line-height: 100px;
  color: #111;
  text-align: center;
  text-transform: uppercase;
}
@media only screen and (min-width: 1000px)
{
.wrapper{
    width: 800px;
    margin: 0 auto;
  }
.index-banner{
  height: 450px;
  }
.index-banner h1{
  display: block;
  width: 560px;
  margin: 40px auto;
  }
  .index-links  {
  padding-bottom: 20px;
  overflow: hidden;
  }
.index-links div {
  margin: 20px 30px 0px;
  height: 230px;
  background-color: #f2f2f2;
  float: left;
  }
.index-boxlink-square{
  width: calc(25% - 60px) !important;
  }
```

```
.index-boxlink-rectancle{
  width: calc(50% - 60px) !important;
  }
  .index-links div h3{
  line-height: 180px;
  }
}
/*FOOTER*/
footer {
  width: calc(100%-20px);
  padding: 40px;
  background-color: #111;
  overflow: hidden;
}
footer ul {
  width: fit-content;
  float: left;
  padding-left: 50px;
}
footer ul li {
  display: block;
  list-style: none;
}
footer ul li a {
  font-family: sans-serif;
  font-size: 24px;
  color: #fff;
  line-height: 40px;
}
.footer-links-cases{
  display: none;
}
.footer-sm {
  width: 50px;
  float: right;
}
.footer-sm img {
  width: 70px;
  margin-bottom: 10px;
}
```

```
@media only screen and (min-width: 1000px){
.footer-links-cases{
  display: block;
  }
  footer ul {
  padding-right: 20px;
}
  footer ul li p {
  font-family: sans-serif;
  font-size: 20px;
  color: #fff;
  line-height: 20px;
  text-transform: uppercase;
  }
}
```

6.5.3. Webpage output for blog Menu



6.5. Contact menu

6.5.1. HTML code for Contact menu

```
<html>
<head>
    <title> Contact Form Design</title>
    <link rel="stylesheet" type="text/css" href="contact.css">
</head>
<body>

    <div class="contact-title">
            <h1> Say Hello to NIELIT-ITANAGAR Library</h1>
            <h2> *** We are always ready to serve you! ***</h2>
    </div>
    <div class="contact-Form">
            <div class="container">
  <form action="action_page.php">

    <label for="fname">First Name</label>
    <input type="text" id="fname" name="firstname" placeholder="Your name..">

    <label for="lname">Last Name</label>
    <input type="text" id="lname" name="lastname" placeholder="Your last name..">

    <label for="country">Country</label>
    <select id="country" name="country">
      <option value="australia">India</option>
      <option value="canada">Sri Lanka</option>
      <option value="usa">Bangladesh</option>
    </select>

    <label for="subject">Subject</label>
    <textarea id="subject" name="subject" placeholder="Write something.."
style="height:100px"></textarea>

    <input type="submit" value="Submit">

  </form>
  <footer>
  <p> Dr.G.Stephen, Assistant Librarian, NIELIT-Itanagar Library will reply your message</p>
```

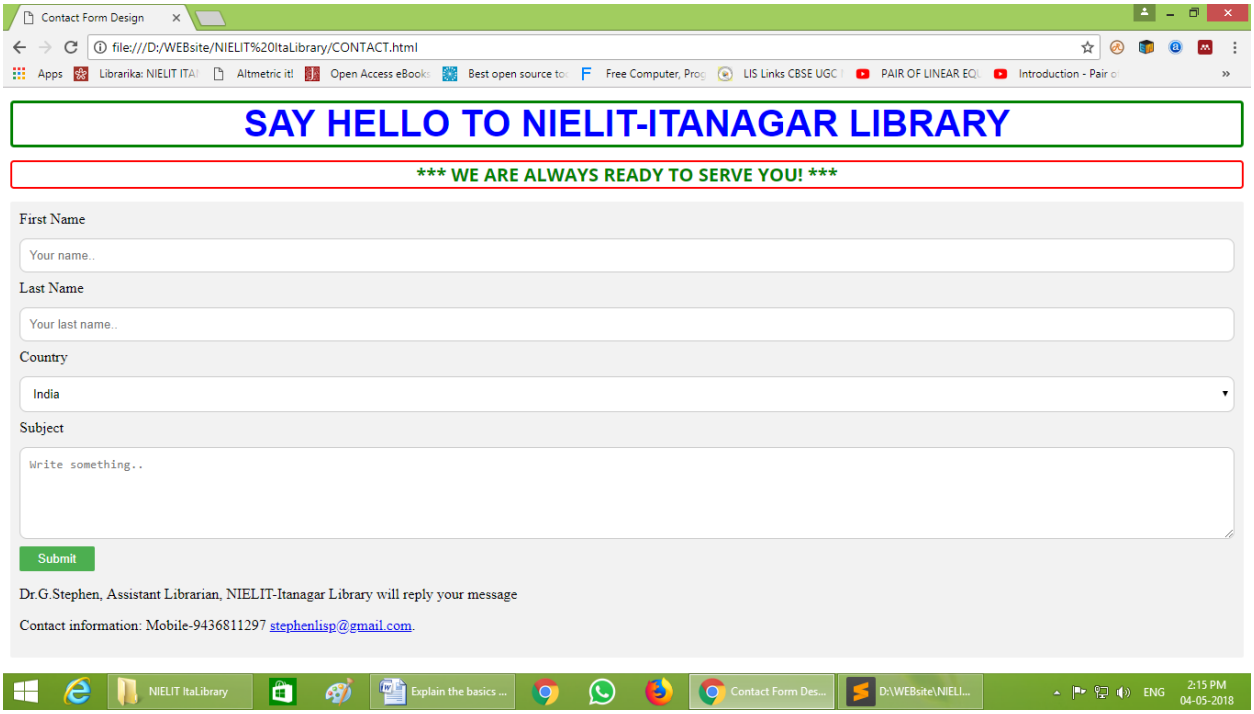
```
  <p>Contact information: Mobile-9436811297 <a href="stephenlisp@gmail.com">
  stephenlisp@gmail.com</a>.</p>
</footer>
</div>
     </div>
</body>
</html>
```

6.5.2. CSS sheet for Contact menu

```
   input[type=text], select, textarea {
   width: 100%;
   padding: 10px;
   border: 1px solid #ccc;
   border-radius: 8px;
   box-sizing: border-box;
   margin-top: 12px;
   margin-bottom: 8px;
   resize: vertical
}
.contact-title h1{
     font-size: 30px;
     font-family: 'open sans', sans-serif;
     text-transform: uppercase;
     text-align: center;
     border: 3px solid green;
   border-radius: 4px;
   font-size: 35px;
     font-size: 40px;
     font-family: 'montserrat', sans-serif;
     color: blue;
     font-weight: 400px;
     margin:14px 0px;
}
.contact-title h2{
     font-size: 20px;
     font-family: 'open sans', sans-serif;
     text-transform: uppercase;
     text-align: center;
   border: 2px solid red;
```

```
    border-radius: 4px;
    color: green;
      font-weight: 200px;
      margin:14px 0px;
    }
input[type=submit] {
    background-color: #4CAF50;
    color: white;
    padding: 6px 20px;
    border: none;
    border-radius: 2px;
    cursor: pointer;
}
input[type=submit]:hover {
    background-color: #45a049;
}
.container {
    border-radius: 2.5px;
    background-color: #f2f2f2;
    padding: 10px;
}
```

6.5.3.Webpage output for Contact Menu



6.6. Signup

6.6.1. HTML code for Signup Menu

```
<!DOCTYPE html>
<html>
    <head>
        <title>Registration Form</title>
        <link rel="stylesheet" type="text/css" href="Regform.css">
    </head>
    <body>
        <div class="Simple-form">
            <form id="Registration">
                <input type="text" name="Fname" id="button" placeholder="Enter UR First name"> <br>
                <br>
                <input type="text" name="Lname" id="button" placeholder="Enter UR Last name"> <br>
                <br>
                <input type="email" name="Email" id="button" placeholder="Enter email id"> <br>
```

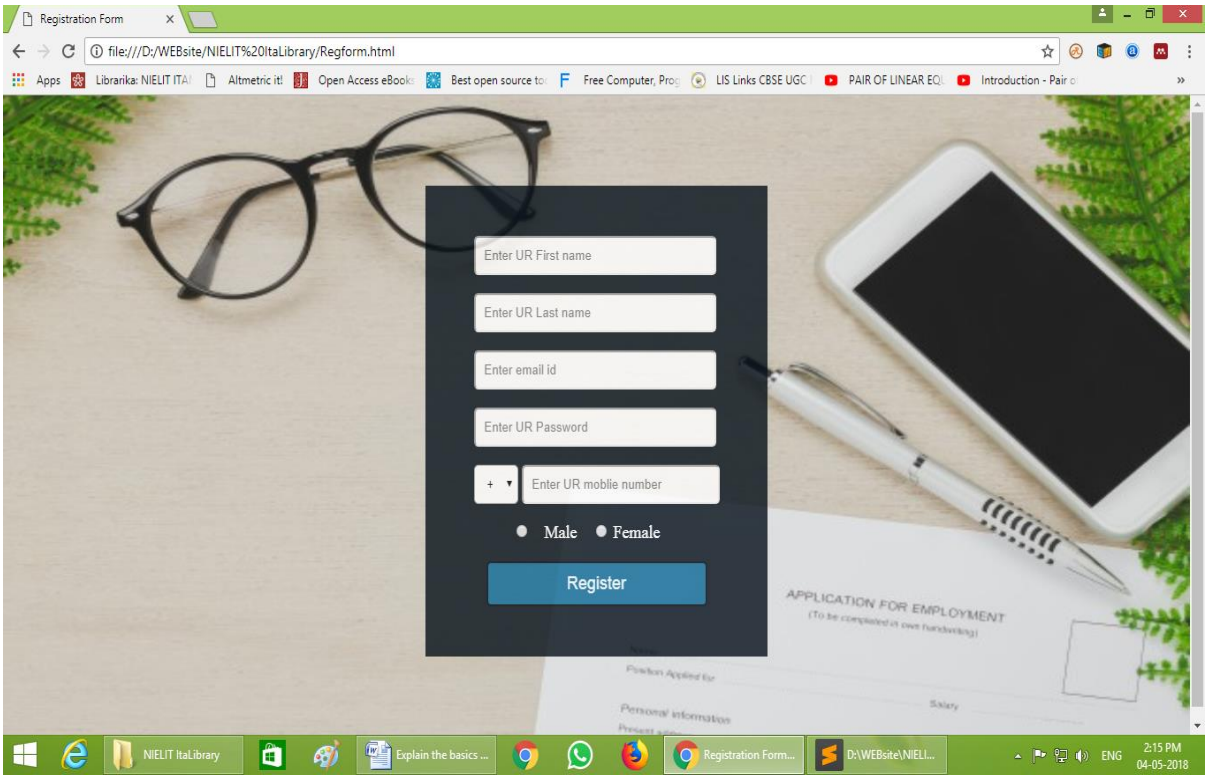
```
                        <br>
                        <input type="password" name="Password" id="button" placeholder="Enter UR Password"> <br>
                        <br>
                        <select id="Ph"> <option>+91</option> <option>+92</option> <option>+93</option> <option>+94</option><option>+95</option></select>
                        <input type="number" name="num" placeholder="Enter UR moblie number" id="Phone"><br>
                        <br>
                        <input type="radio" name="gender" id="rd">    <span id="but">Male</span>    <input type="radio" name="gender"> <span id="but">Female</span>     
                        <br> <br>
                        <input type="submit" value="Register" id="butt">
                    </form>
            </div>
    </body>
    </html>
```

6.6.2.CSS sheet for Regform

```
*body{
    background-image: url(form.jpg);
    background-repeat: no-repeat;
    background-size: 100% 700px;
}
.Simple-form{
    text-align: center;
    margin: 90px 470px;
}
#Registration{
    width: 100%;
    background-color: #051019;
    opacity: 0.8;
    padding: 50px 0px;
}
#button{
    width: 250px;
    padding: 10px;
```

```
    border-radius: 5px;
    outline: 0px;
}
#Ph{
    outline: 0px;
    width: 50px;
    padding: 10px;
    border-radius: 5px;
}
#Phone{
    width: 200px;
    padding: 10px;
    border-radius: 5px;
    outline: 0px;
}
#rd{

}
#but{
    color: white;
    font-size: 18px;
}
#butt{
    width: 250px;
    padding: 10px;
    border-radius: 5px;
    outline: 0px;
    background-color: #0c6996;
    border: 2px solid #01010c;
    color: aliceblue;
    font-size: 18px;
}
```

6.6.2. Webpage output for Registration form (Signup)



6.7. Useful External links connected with this website

Library Online

- Catalague search
- New arrivals
- Top collections
- A-Z database
- Ask librarian

Student links

- Current Affiars
- Student.Nielit
- Students welfare
- Job search
- Nielit News

E-learning

- NPTEL

Swayam
Khan Academy
Edx
Coursera

Social media

Facebook
Twitter
Trepup

CHAPTER VII

CONCLUSION

Few Benefits of HTML and CSS

- **Cost effective and Multi-Platform Development:** A single piece of code can be used across platforms, devices and markets, this an added advantage because it results in lower development and maintenance costs, allowing you to use your resources elsewhere.

- **Good page ranking**: If your page is not structured well then the page itself will not attain a good rank within search engines. HTML5's new elements can be anticipated when site are being re-indexed on search engines.

- **Consistency across multiple web browsers:** The implementation of HTML and CSS helps the designer to create a compatible site or system within all browsers.

- **User Experience:** HTML5 offers wide range of design and presentation tools which gives the developers greater scope to produce better sites and web applications. This is very important from business point of view as user engagement is the key to increase site and system use and conversion.

Conclusion

In order to accomplish the goal of providing a complete and coherent Web presence, every unit in the organization needs to be tapped. In a Webbed world, duplication of effort is not only wasteful, it is confusing to users. There is a need to ensure that the same information is not being maintained in several different places with several different rates of upkeep. Issues of preferences in design need to be hammered out so that our users need not understand our political structure in order to understand our Web site. Yet we need to be flexible enough to realize that librarians serving very specific segments of users know those segments best and know what is going to be most effective in communicating with that population. To solve problems when users might not be able to connect to a resource, public service librarians are learning the technical realities of such things as network structure, proxy servers, browser differences, and what licenses may be restricting access.

Bibliographers can now select networked resources that assure representation but not duplication of all necessary subjects. Systems people continue to refine and develop search engines that will enhance the design layout created by those who maintain content on the Web. Reference and user service people need to be queried about what interfaces are most effective and least problematic. Administration needs to be aware of changing traffic patterns to know where best to allocate funds and personnel.

To ensure a Web presence that is coherent, deliberate, and a step beyond making deli sandwiches, libraries need to examine these potential roles and determine which are priorities for that organization. The ingenuity and ideas of librarians from all facets of the organization coupled with a vision from an institutional perspective will give meaningful direction to a Web presence.

Classic HTML has been a major component in the emergence of a word wide content communication system. It is now time to move HTML forward so that it becomes one of several tools for presentation of content. Classic HTML should be transformed into modern HTML which is a namespace in XML and associated canonical semantics. The semantics of canonical HTML should be available to all tagsets in XML via the DOM along with other standardized semantics. In this way a set of focused rich vertical vocabularies can arise that make communication efficient and fast while at the same time permit widespread understanding and dynamic re-presentation to achieve the widest possible access. To achieve this we have suggested a plan that moves HTML from accommodation to integration to assimilation until it is finally subsumed.

Web References.

1. https://www.homeandlearn.co.uk/WD/WebDesign.html.
2. http://bradfrostweb.com/blog/post/five-responsive-webdesign-pitfalls-to-avoid.
3. 3. http://blog.teamtreehouse.com/modern-field-guideresponsive-web-design.
4. “Implementing Responsive Web Design for Enhanced Web Presence” by S. Mohorovičić, University of Rijeka, Faculty of Maritime Studies, Rijeka, Croatia.
5. “Responsive web design from worldwide consortium” http://en.wikipedia.org/wiki/Responsive_web_design. “Basic fundamentals of Responsive Designing by Pete.
6. LePage”,https://developers.google.com/web/fundamentals/layouts/ rwd-fundamentals/. http://www.nngroup.com/articles/responsive-web-designdefinition/, “Responsive Web Design and User Experience by Amy Schade.”
7. http://learn.shayhowe.com/advanced-htmlcss/responsive-web-design/, by Ethan Marcotte.
8. http://www.lynda.com/Dreamweaver-tutorials/Dreamweaver-CS6- WordPress-38-Core-Concepts/161459-
9. html http://www.lynda.com/Web-Interactive-CSS-tutorials/CSSFundamentals/80436-html?srchtrk=index:1%0Alinktypeid:2%0Aq:css%0Apage:1%0As:relevance%0Asa:true%0Aproducttypeid:2
10. "Namespaces in XML 1.0 (third edition)", T. Bray, D. Hollander, A. Layman, R. Tobin, H. S. Thompson, 8 December 2009Available at: http://www.w3.org/TR/2009/REC-xml-names-20091208/.
11. "CSS3 Speech Module", David Raggett, Daniel Glazman, Claudio Santambrogio, Daniel Weck, 19 April 2011, W3C Working Draft (work in progress). Available at http://www.w3.org/TR/2011/WD-css3-speech-20110419/
12. "Cascading Style Sheets, level 2, CSS2 Specification", B. Bos, H. W. Lie, C. Lilley and I. Jacobs, 12 May 1998 (revised 11 April 2008), http://www.w3.org/TR/2008/REC-CSS2-20080411/
13. "HTML 4.01 Specification", D. Raggett, A. Le Hors, I. Jacobs, 24 December 1999. The latest version of the specification is available at http://www.w3.org/TR/html4/. The Recommendation defines three document type definitions: Strict, Transitional, and Frameset, all reachable from the Recommendation.