A

REPORT

On

WEB DEVELOPMENT

Submitted

In partial fulfilment

For the award of the Degree of

Submitted By:

S. Avinash (221710301052)

G. Varshith Reddy (221710301020)

V. Ajay Kumar Reddy(221710301062)

S. Sai Charitha Sanjana

(221710301049)

Submitted To:

Dhyanahitha

School of

Professional Studies

Department of Computer Science and Engineering

Candidate's Declaration

I, hereby declare that the project report "HOME TUTORIAL POINT" is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the certificate of internship. The work has not been submitted to any other college or University for the award of any degree or diploma.

(Name and Signature of Candidate) **Counter Signed by**

S. Avinash (221710301052)

G. Varshith Reddy (221710301020)

V. Ajay Kumar Reddy (221710301062)

S. Sai Charitha Sanjana (221710301049)

ACKNOWLEDGEMENT

My project would not have been successful without the help of several people. I would like to thank the personalities who were part of my project in numerous ways, those who gave me outstanding support from the birth of the project.

I am extremely thankful to our honorable Pro-Vice-Chancellor, Prof. N. Siva Prasad for providing necessary infrastructure and resources for the accomplishment of my project. I am highly indebted to Prof. N. Seetha Ramaiah, Principal, School of Technology, for his support during the tenure of the project.

I am very much obliged to our beloved Prof. S. Phani Kumar, Head of the Department of Computer Science & Engineering for providing the opportunity to undertake this project and encouragement in the completion of this project.

I hereby wish to express my deep sense of gratitude to Mr. Anil Kumar, for the esteemed guidance, moral support and invaluable advice provided by him for the success of the project. I am also thankful to all the staff and mentors of the Dhyanahitha School of Professional Studies who have cooperated in making my project a success. I would like to thank my parents and friends who extended their help, encouragement and moral support either directly or indirectly in my project work.

Sincerely,

S. Avinash(221710301052)

G. Varshith Reddy(221710301020)

V. Ajay Kumar Reddy (221710301062)

S. Sai Charitha Sanjana (221710301049)

ABSTRACT

People expect to be able to work, learn, and study whenever and wherever they want to. Since the present pandemic situation has disrupted the normal lifestyle of people across the globe, the virtual world has come to the rescue. Many institutions have also shifted their base to virtual platforms to conduct classes online,with the distinctive rise of e-learning, whereby teaching is undertaken remotely and on digital platforms. Consequently, catering to the needs of all stages of education from pre-primary to university level, online education has emerged as an alternative to ordinary face to face classes. Moreover, efforts are being made by both government and non-government organisations and edtech companies to support the school system to make a smooth transition to the virtual world.In view of this pandemic situation we came up with an online 'HOME TUTOTIAL POINT' webite, which helps each and every student to learn everything free online and help them continue to pursue their respective courses through our tutorial. Our tutorial provides courses such as from class-11 to class-12, C, C++, Java, Python and languages like German, Spanish.

Table of Contents

List of Figures

CHAPTER-1

INTRODUCTION

The industry definition of a Full Stack Developer is an engineer who can work on different levels of an application stack. The term stack refers to the combination of components and tools that make up the application. The components could be in the front-end or the back-end of the system.

The main objective of a full stack engineer is to keep every part of the system running smoothly. A Full Stack Developer can perform tasks ranging from resizing an image or text in a webpage to patching the kernel.

Full stack development: It refers to the development of both front end (client side) and back end (server side) portions of web application.

Full stack web Developers: Full stack web developers have the ability to design complete web applications and websites. They work on the frontend, backend, database and debugging of web applications or websites.

2.4 FRONT-END

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

The objective of designing a site is to ensure that when the users open up the site they see the information in a format that is easy to read and relevant. This is further complicated by the fact that users now use a large variety of devices with varying screen sizes and resolutions thus forcing the designer to take into consideration these aspects when designing the site. They need to ensure that their site comes up correctly in different browsers (cross-browser), different operating systems (cross-platform) and different devices (cross-device), which requires careful planning on the side of the developer.

Front end development manages everything that users visually see first in their browser or application. Front end developers are responsible for the look and feel of a site. It is the visible part of a website or web application which is responsible for user experience. The user directly interacts with the front-end portion of the web application or website.

1.2 BACK-END

Back end development refers to the server side of an application and everything that communicates between the database and the browser. It is responsible for managing the database through queries and APIs by client-side commands. Back end development refers to the server side of development where you are primarily focused on how the site works. Making updates and changes in addition to monitoring functionality of the site will be your primary responsibility. This type of web development usually consists of three parts: a server, an application, and a database. Code written by back end developers is what communicates the database information to the browser. Anything you can't see easily with the eye such as databases and servers is the work of a back-end developer.
Back end developer positions are often called programmers or web developers.

CHAPTER-2

WEB-DEVELOPMENT

Web development is a broad term for the work involved in developing a website for the Internet (World Wide Web) or an intranet (a private network). Web development can range from developing the simplest static single page of plain text to the most complex web-based internet applications, electronic businesses, and social network services. A more comprehensive list of tasks to which web development commonly refers, may include web engineering, web design, web content development, client liaison, client-side/side scripting, web server and network security configuration, and e-commerce development. Among web professionals, "web development" usually refers to the main non-design aspects of building web sites: writing markup and coding. Most recently Web development has come to mean the creation of content management systems or CMS. These CMS can be made from scratch, proprietary or open source. In broad terms the CMS acts as middleware between the database and the user through the browser. A principle benefit of a CMS is that it allows non-technical people to make changes to their website without having technical knowledge.

For larger organizations and businesses, web development teams can consist of hundreds of people (web developers) and follow standard methods like Agile methodologies while developing websites. Smaller organizations may only require a single permanent or

contracting developer, or secondary assignment to related job positions such as a graphic designer or information systems technician. Web development may be a collaborative effort between departments rather than the domain of a designated department. There are three kinds of web developer specialization: front-end developer, back-end developer, and full-stack developer.

2.1 WEB-SITE

A **website** is a collection of related web pages, including multimedia content, typically identified with a common domain name, and published on at least one web server. A website may be accessible via a public Internet Protocol (IP) network, such as the Internet, or a private local area network (LAN), by referencing a uniform resource locator (URL) that identifies the site.

Websites have many functions and can be used in various fashions; a website can be a personal website, a commercial website for a company, a government website or a non-profit organization website. Websites are typically dedicated to a particular topic or purpose, ranging from entertainment and social networking to providing news and education. All publicly accessible websites collectively constitute the World Wide Web, while private websites, such as a company's website for its employees, and are typically a part of an intranet.

Web pages, which are the building blocks of websites, are documents, typically composed in plain text interspersed with formatting instructions of Hypertext Markup Language (HTML, XHTML). They may incorporate elements from other websites with suitable markup anchors. Web pages are accessed and transported with the Hypertext Transfer Protocol (HTTP), which may optionally employ encryption (HTTP Secure, HTTPS) to provide security and privacy for the user. The user's application, often a web browser, renders the page content according to its HTML markup instructions onto a display terminal.

Hyperlinking between web pages conveys to the reader the site structure and guides the navigation of the site, which often starts with a home page containing a directory of the site web content. Some websites require user registration or subscription to access content. Examples of subscription websites include many business sites, news websites, academic journal websites, gaming websites, file-sharing websites, message boards, web-based email, social networking websites, websites providing real-time stock market data, as well as sites providing various other services. As of 2016 end users can access websites on a range of devices, including desktop and

laptop computers, tablet computers, smartphones and smart TVs. A web site consists of web pages which are interconnected to each other and contain various data and functionalities.

2.2 WEB-PAGE

A **web page**, or **webpage**, is a document that is suitable for the World Wide Web and web browsers. A web browser displays a web page on a monitor or mobile device. The web page is what displays, but the term also refers to a computer file, usually written in HTML or comparable markup language. Web browsers coordinate the various web resource elements for the written web page, such as style sheets, scripts, and images, to present the web page.

Typical web pages provide hypertext that includes a navigation bar or a sidebar menu to other web pages via hyperlinks, often referred to as links.

On a network, a web browser can retrieve a web page from a remote web server. On a higher level, the web server may restrict access to only a private network such as a corporate intranet or it provides access to the World Wide Web. On a lower level, the web browser uses he Hypertext Transfer Protocol (HTTP) to make such requests.

A static web page is delivered exactly as stored, as web content in the web server's file system, while a dynamic web page is generated by a web application that is driven by server- side software or client-side scripting. Dynamic website pages help the browser (the client) to enhance the web page through user input to the server.

2.3 WEB BROWSER

A **web browser** (commonly referred to as a **browser**) is a software application for accessing information on the World Wide Web. When a user requests a web page from a particular website, the web browser retrieves the necessary content from a web server and then displays the page on the user's device.

A web browser is not the same thing as a search engine, though the two are often confused. For a user, a search engine is just a website that stores searchable data about other websites. However, to connect to a website's server and display its web pages, a user must have a web browser installed.

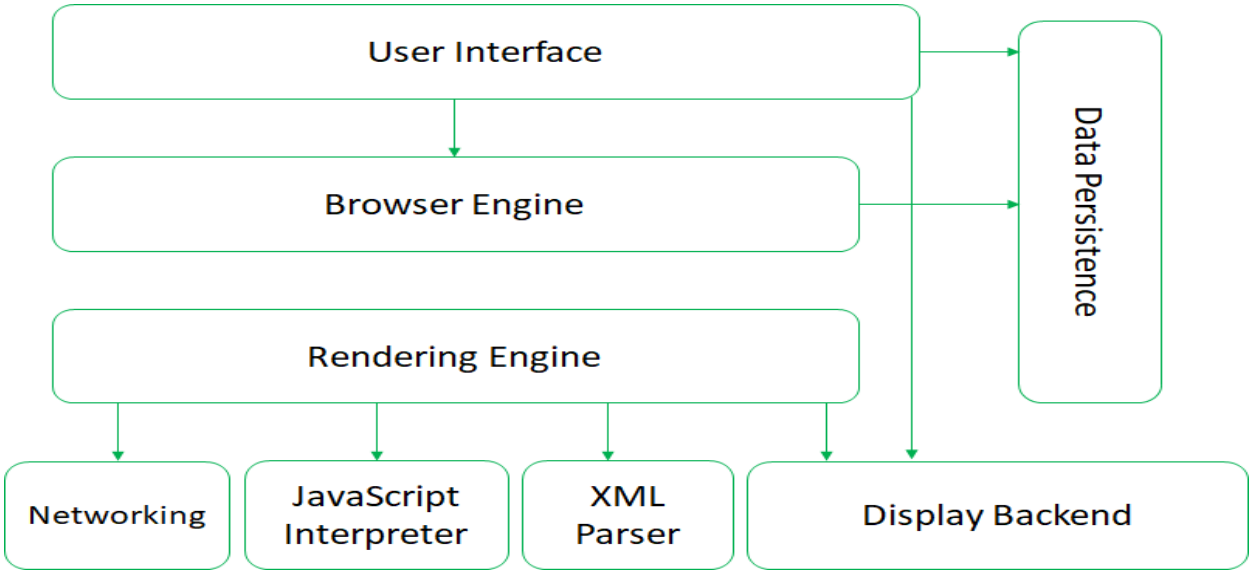
Web browsers are used on a range of devices, including desktops, laptops, tablets, and

smartphones. In 2019, an estimated 4.3 billion people used a browser. The most used browser is Google Chrome, with a 64% global market share on all devices, followed by Safari with 18%.

2.4 MOBILE WEB BROWSER

A **mobile browser** is a web browser designed for use on a mobile device such as a mobile phone or PDA. Mobile browsers are optimized so as to display Web content most effectively for small screens on portable devices. Mobile browser software must be small and efficient to accommodate the low memory capacity and low-bandwidth of wireless handheld devices. Typically, they were stripped-down web browsers, however, some recent mobile browsers can handle latest technologies also such as CSS 3, JavaScript, and Ajax.

Websites designed so that they may be accessed from these browsers are referred to as wireless portalsor collectively as the Mobile Web. They may automatically create "mobile" version of each page, for example the Wikipedia website.



CHAPTER-3

THE STEPS TO CREATE A WEBSITE

Creating a web site requires multiple steps which includes the following:

- Creating a UI (User interface)
- Scripting (Both at server end and client end)
- Creating a backend or the database

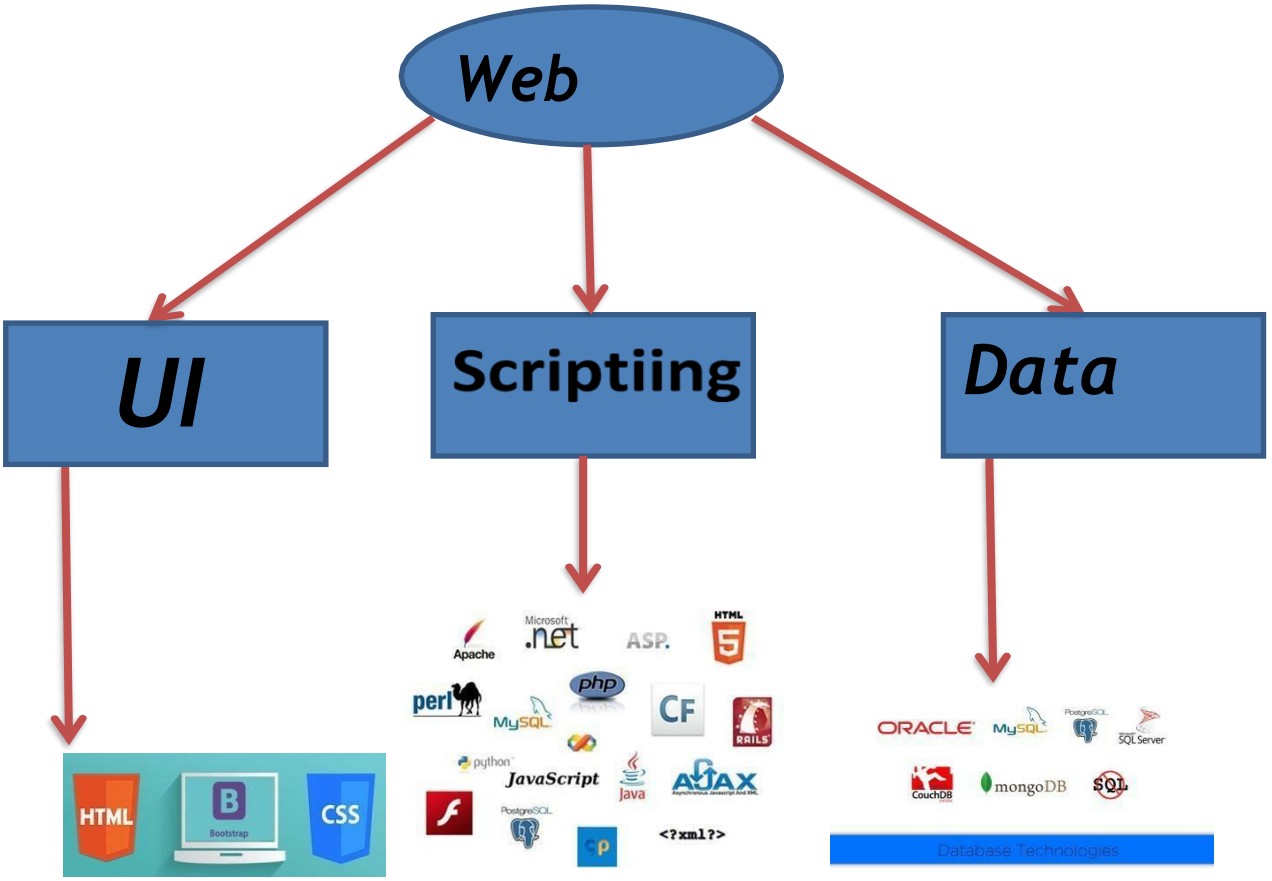


Fig 3.1

3.1 UI DEVELOPMENT

Technologies that are mostly used to develop a User Interface are:

- HTML
- CSS

- Bootstrap

3.1.1 HTML

Hypertext Markup Language (**HTML**) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Others such as ... surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

HTML markup consists of several key components, including those called tags (and their attributes), character-based data types, character references and entity references. HTML tags most commonly come in pairs like <h1> and </h1>, although some represent empty elements and so are unpaired, for example <img>. The first tag in such a pair is the start tag, and the second is the end tag (they are also called opening tags and closing tags).Another important component is the HTML document type declaration, which triggers standards mode rendering.

The following is an example of the classic Hello world program, a common test employed for comparing programming languages, scripting languages and markup languages. This example is made using 9 lines of code:

General Syntax ofHTML

```
<!DOCTYPE html>
<html>
<head>
<title>This is a title</title>
</head>
<body>
<p>Hello world! </p>
</body>
</html>
```

(The text between <html> and </html> describes the web page, and the text between <body> and </body> is the visible page content. The markup text "<title>This is a title</title>" defines the browser page title.)

The Document Type Declaration <!DOCTYPE html> is for HTML5. If a declaration is not included, various browsers will revert to "quirks mode" for rendering.

3.1.2 CSS

Cascading Style Sheets (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS

in a separate .css file, and reduce complexity and repetition in the structural content

Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified.

Changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in thedocuments.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/CSS is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

Types of CSS:

- **Inline CSS**: In this CSS is applied in between the tags

  E.g.: <tag style=" styling">Hello World</tag>

- **Internal CSS:**

  In this code is defined inside the style tag in the head section of the HTML page.

General Syntax:

```
<html>

    <head>

    <style>
```

```
        <! -- CSS STYLING -- >

    </style>

    </head>
</html>
```

- **External CSS:**

In this the CSS code is written on another page and is linked to the HTML page. It is advantageous to use this type of styling as we can use the same file to style various HTML pages.

External CSS uses the extension .CSS and is applied using the following syntax:

```
<html>

    <head>

        <link relation=" stylesheet" type="CSS" href="URL to the page">

    </head>

</html>
```

All the CSS style types are important but can be used in different situations.

- Inline CSS is used when only small changes are to be done to the HTML tag and the changes are to be reflected only to that specific tag
- Internal CSS is used when the individual HTML pages have to be designed differently. This also slows the page load system if the internal styling is long.
- External CSS files are maintained to design multiple pages and use common styles over various pages. It is useful as it helps in managing the resources in an easy manner.

Both HTML and CSS are used to create a UI but CSS behaves like a makeup on the face of

an actress which makes her look even more beautiful than she is in reality.

And here is the difference: **Before using CSS in HTML page**:

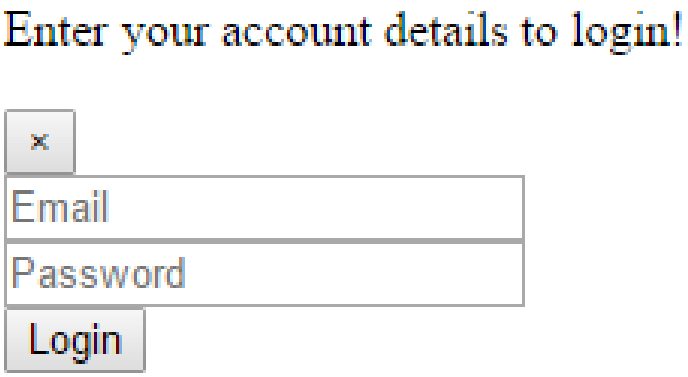


Fig 3.2

After using CSS in HTML Page:

Fig 3.3

3.1.3 BOOTSTRAP

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap is the second most-starred project on GitHub, with more than 107,000 stars and 48,000 forks.

Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to inconsistencies and a high maintenance burden. According to twitter developer Mark Otto:

"A super small group of developers and I got together to design and build a new internal tool and saw an opportunity to do something more. Through that process, we saw ourselves build something much more substantial than another internal tool. Months later, we ended up with an early version of Bootstrap as a way to document and share common design patterns and assets within thecompany."

After a few months of development by a small group, many developers at Twitter began to contribute to the project as a part of Hack Week, a hackathon-style week for the Twitter development team. It was renamed from Twitter Blueprint to Bootstrap, and released as an open source project on August 19, 2011. It has continued to be maintained by Mark Otto, Jacob Thornton, and a small group of core developers, as well as a large community of contributors.

On January 31, 2012, Bootstrap 2 was released, which added a twelve-column responsive grid layout system, inbuilt support for Glyph icons, several new components, as well as changes to many of the existing components.

On August 19, 2013, Bootstrap 3 was released, which redesigned components to use Flat design, and a mobile first approach.

On October 29, 2014, Mark Otto announced that Bootstrap 4 was in development. The first alpha version of Bootstrap 4 was released on August 19, 2015.

Bootstrap 3 supports the latest versions of Google Chrome, Firefox, Internet Explorer, Opera, and Safari (except on Windows). It additionally supports back to IE8 and the latest Firefox Extended Support Release(ESR).

Since 2.0, Bootstrap supports responsive web design. This means the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (desktop, tablet, mobile phone).

Starting with version 3.0, Bootstrap adopted a mobile-first design philosophy, emphasizing responsive design bydefault.

The version 4.0 alpha release added Sass and flexbox support.

Installing and linking bootstrap to the HTML page:

Install bootstrap from https://getbootstrap.com/

- Copy the bootstrap.min.css file to your CSS folder and link it to the HTML page in the similar manner to how any other CSS file is linked.
- Link the bootstrap.min.js file which is present in the JS folder of the bootstrap. It can be linked using ascript tag.

E.g.: <script src =”URL to bootstrap.min.js”></script>

- Now use bootstrap classes to reduce the work of designing which was earlier done through CSS.

3.2 SCRIPTING

There are two scripting methodologies.

1. Server-side scripting: This scripting is done at the server end
2. Client-side scripting: This scripting is done at the client end or the browser

3.2.1 SERVER-SIDE SCRIPTING

Server-side scripting is a technique used in web development which involves employing scripts on a web server which produce a response customized for each user's (client's) request to the website. The alternative is for the web server itself to deliver a static web page. Scripts can be written in any of a number of server-side scripting languages that are available (see below). Server-

side scripting is distinguished from client- side scripting where embedded scripts, such as JavaScript, are run client-side in a web browser, but both techniques are often used together.

Server-side scripting is often used to provide a customized interface for the user. These scripts may assemble client characteristics for use in customizing the response based on those characteristics, the user's requirements, access rights, etc. Server-side scripting also enables the website owner to hide the source code that generates the interface, whereas with client-side scripting, the user has access to all the code received by the client. A down-side to the use of server-side scripting is that the client needs to make further requests over the network to the server in order to show new information to the user via the web browser. These requests can slow down the experience for the user, place more load on the server, and prevent use of the application when the user is disconnected from the server.

When the server serves data in a commonly used manner, for example according to the HTTP or FTP protocols, users may have their choice of a number of client programs (most modern web browsers can request and receive data using both of those protocols). In the case of more specialized applications, programmers may write their own server, client, and communications protocol that can only be used with one another.

Programs that run on a user's local computer without ever sending or receiving data over a network are not considered clients, and so the operations of such programs would not be considered client-side operations.

3.2.1.1 Server-Side scripting Languages

There are several languages that can be used for server-side programming:

PHP

ASP.NET (C# OR Visual

Basic) C++

Java and

JSP Python

Ruby on Rails and so on.

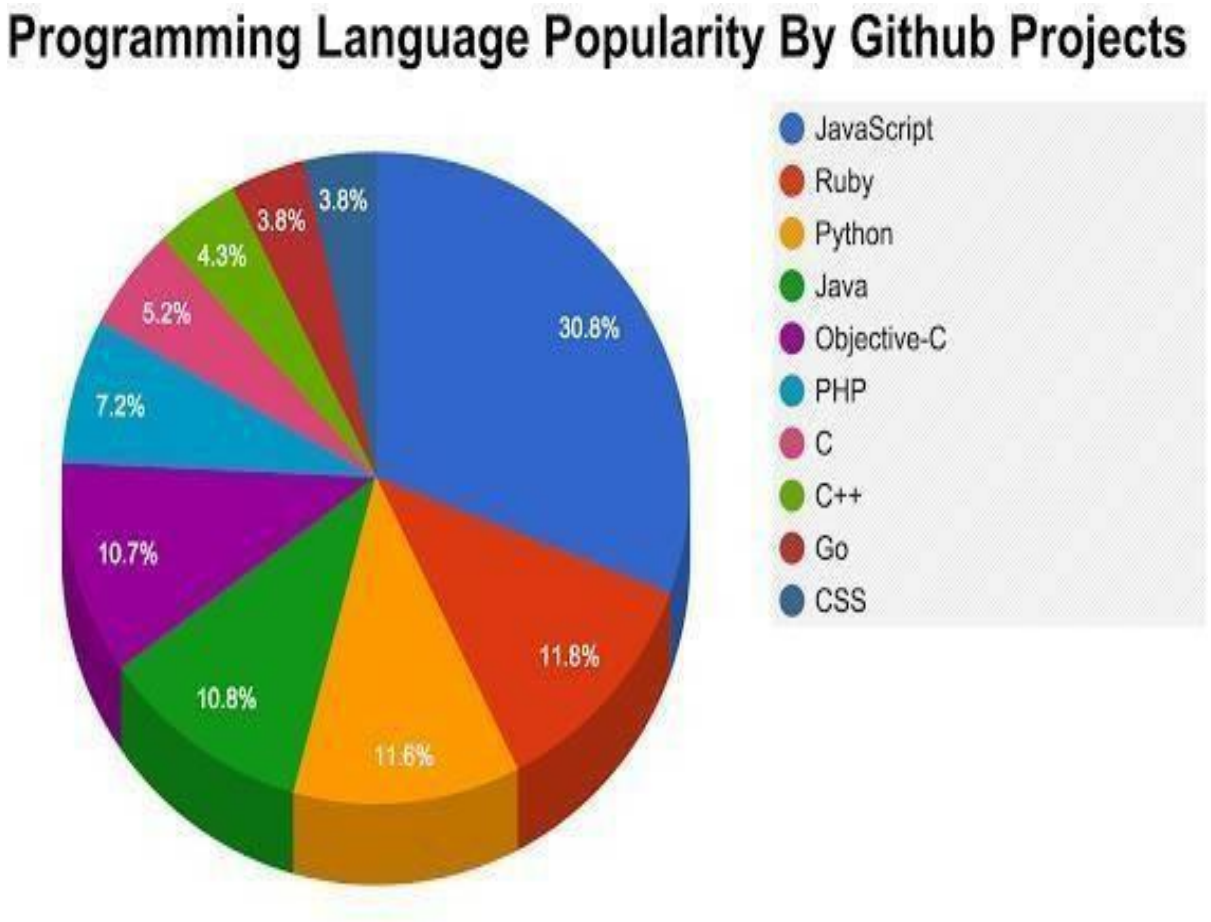


Fig 3.4

3.2.2 CLIENT-SIDE SCRIPTING

Client-side scripting is changing interface behaviors within a specific web page in response to mouse or keyboard actions, or at specified timing events. In this case, the dynamic behavior occurs within the presentation. The client-side content is generated on the user's local computer system.

Such web pages use presentation technology called rich interfaced pages. Client- side scripting languages like JavaScript or ActionScript, used for Dynamic HTML (DHTML) and Flash technologies respectively, are frequently used to orchestrate media types (sound, animations, changing

text, etc.) of the presentation. Client-side scripting also allows the use of remote scripting, a technique by which the DHTML page requests additional information from a server, using a hidden frame, XML Http Requests, or a Web service.

The first widespread use of JavaScript was in 1997, when the language was standardized as ECMAScript and implemented in Netscape 3.

Example:

The client-side content is generated on the client's computer. The web browser retrieves a page from the server, then processes the code embedded in the page (typically written in JavaScript) and displays the retrieved page's content to the user.

The most popularly used client-side scripting languages is **JavaScript**. Flow of request from browser toserver:

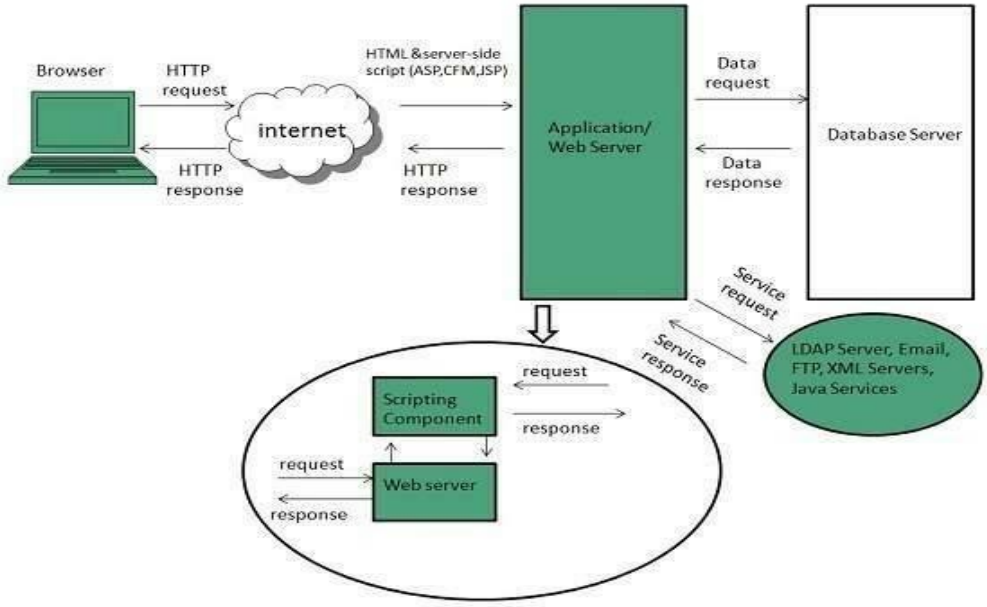


Fig 3.5

3.3 DATABASE

A **database** is an organized collection of data. It is the collection of schemas, tables, queries, reports, views, and other objects. The data are typically organized to model aspects of reality in a way that supports processes requiring information, such as modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

A **database management system** (**DBMS**) is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data. A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, PostgreSQL, MongoDB, MariaDB, Microsoft SQL Server, Oracle, Sybase, SAP HANA, MySQL and IBM DB2. A database is not generally portable across different DBMSs, but different DBMS can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Database management systems are often classified according to the database model that they support; the most popular database systems since the 1980s have all supported the relational model as represented by the SQL language. Sometimes a DBMS is loosely referred to as a "database".

3.4 SQL

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

SQL was one of the first commercial languages for Codd's relational, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks." Despite not entirely adhering to the relational model as described by Codd, it became the most widely used databaselanguage.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987. Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments.

3.5 QUERIES

The most common operation in SQL, the query, makes use of the declarative SELECT statement. SELECT retrieves data from one or more tables, or expressions. Standard SELECT statements have no persistent effects on the database. Some non-standard implementations of SELECT can have persistent effects, such as the SELECT INTO syntax provided in some databases.

Queries allow the user to describe desired data, leaving the database management system (DBMS) to carry out planning, optimizing, and performing the physical operations necessary to produce that result as it chooses.

A query includes a list of columns to include in the final result, normally immediately following the SELECT keyword. An asterisk ("*") can be used to specify that the query should return all columns of the queried tables. SELECT is the most complex statement in SQL, with optional keywords and clauses that include:

The FROM clause, which indicates the table(s) to retrieve data from. The FROM clause can include optional JOIN subclauses to specify the rules for joining tables.

The WHERE clause includes a comparison predicate, which restricts the rows returned by the query. The WHERE clause eliminates all rows from the result set where the comparison predicate does not evaluate to True.

The GROUP BY clause projects rows having common values into a smaller set of rows. GROUP BY is often used in conjunction with SQL aggregation functions or to eliminate duplicate rows from a result set. The WHERE clause is applied before the GROUP BY clause.

The HAVING clause includes a predicate used to filter rows resulting from the GROUP BY clause. Because it acts on the results of the GROUP BY clause, aggregation functions can be used in the HAVING clause predicate.

The ORDER BY clause identifies which column[s] to use to sort the resulting data, and in which direction to sort them (ascending or descending). Without an ORDER BY clause, the order of rows returned by an SQL query is undefined.

The DISTINCT keyword eliminates duplicate data.

CHAPTER-4 SCRIPTING LANGUAGES

4.1 PHP

Paradigm	Imperative, functional, object-oriented, procedural, reflective
Designed by	RasmusLerdorf
Developer	The PHP Development Team, ZendTechnologies
First appeared	June 8, 1995; 21 years ago[1]
Stable release	7.1.5 / May 11, 2017; 16 days ago,
Typing discipline	Dynamic, weak, gradual (as of PHP 7.0.0)
Implementatio n language	C (primarily; some componentsC++)
OS	Unix-like, Windows
License	PHP License (most of Zend Engine under Zend Engine License& The TSRMLicense)
Filename extension s	.php, .phtml, .php3, .php4, .php5, .php7,. phps
Website	php.net
Major implementations	
Zend Engine, HHVM, Phalanger, Quercus, Project Zero, Parrot	
Influenced by	
C, C++, Java, Perl, Tcl[1]	
Influenced	
Falcon, Hack	

PHP is a server-side scripting language designed primarily for web development but also used

as a general-purpose programming language. Originally created by RasmusLerdorf in 1994, the PHP reference implementation is now produced by ThePHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML or HTML5 markup, or it can be use in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.

4.2 Installing PHP

I. Step 1: download the files. Download the latest PHP 5 ZIP package from www.php.net/downloads.php. ...
II. Step 2: extract the files. ...
III. Step 3: configure php.ini. ...
IV. Step 4: add C: php to the path environment variable. ...
V. Step 5: configure PHP as an Apache module. ...
VI. Step 6: test a PHP file.
VII. Or we can install **Xampp** which have inbuilt php, mysql,apache server

We have used xampp to run the php files.

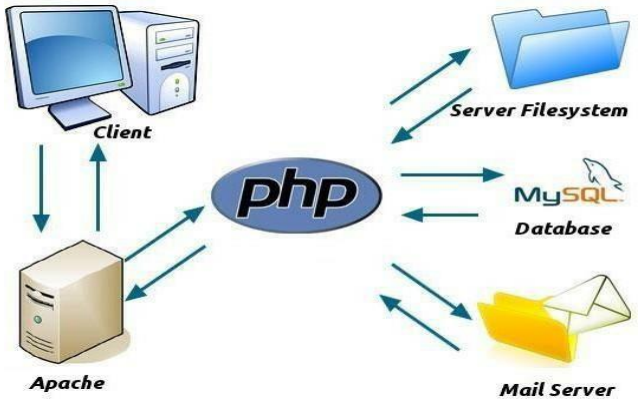


Fig 4.1

4.3 JAVA SCRIPT

JavaScript, often abbreviated as "JS", is a high-level, dynamic, untyped, and interpreted run-time language. It has been standardized in the ECMAScript language specification. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production; the majority of websites employ it, and all modern Web browsers support it without the need for plug-ins. JavaScript is prototype-based with first- class functions, making it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. It has an API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as networking, storage, or graphics facilities, relying on these upon the host environment in which it is embedded.

Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two are distinct languages and differ greatly in their design. JavaScript was influenced by programming languages such as self and Scheme.

JavaScript is also used in environments that are not Web-based, such as PDF documents, site-specific browsers, and desktop widgets. Newer and faster JavaScript virtual machines (VMs) and platforms built upon them have also increased the popularity of JavaScript for server-side Web applications. On the client side, developers have traditionally implemented JavaScript as an interpreted language, but more recent browsers perform just-in- time compilation. Programmers also use JavaScript in video-game development, in crafting desktop and mobile applications, and in server-side

network programming with run-time environments such as Node.js.

4.4 JQUERY

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT license. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin.

jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications.

The set of jQuery core features—DOM element selections, traversal and manipulation—enabled by its selector engine (named "Sizzle" from v1.3), created a new "programming style", fusing algorithms and DOM data structures. This style influenced the architecture of other JavaScript frameworks like YUI v3 and Dojo, later stimulating the creation of the standard Selectors API.

Microsoft and Nokia bundle jQuery on their platforms. Microsoft includes it with Visual Studio for use within Microsoft's ASP.NET AJAX and ASP.NET MVC frameworks while Nokia has integrated it into the Web Run-Time widget development PLATFORM

4.5 AJAX

Ajax (also **AJAX** short for "asynchronous JavaScript and XML") is a set of Web development techniques using many Web technologies on the client side to create asynchronous Web applications. With Ajax, Web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behavior of the existing page. By decoupling the data interchange layer from the presentation layer, Ajax allows for Web pages, and by extension Web applications, to change content dynamically without the need to reload the entire page. In practice, modern implementations commonly substitute JSON for XML due to the advantages of being native to

JavaScript.

Ajax is not a single technology, but rather a group of technologies. HTML and CSS can be used in combination to mark up and style information. The DOM is accessed with JavaScript to dynamically display — and allow the user to interact with — the information presented. JavaScript and the XML HTTP Request object provide a method for exchanging data asynchronously between browser and server to avoid full page reloads.

4.6 JSON

In computing, **JavaScript Object Notation** or **JSON** is an open- standard file format that uses human-readable text to transmit data objects consisting of attribute—value pairs and array data types (or any other serializable value). It is a very common data format used for asynchronous browser/server communication, including as a replacement for XML in some AJAX-style systems.

JSON is a language-independent data format. It was derived from JavaScript, but as of 2017 many programming languages include code to generate and parse JSON-format data. The official Internet media type for JSON is application/json. JSON file names use the extension

.json.

Douglas Crockford originally specified the JSON format in the early 2000s; two competing standards, RFC 7159 and ECMA-404, defined it in 2013. The ECMA standard describes only the allowed syntax, whereas the RFC covers some security and interoperability considerations.[3]

A restricted profile of JSON, known as **I-JSON** (short for "Internet JSON"), seeks to overcome some of the interoperability problems with JSON. It is defined in RFC 7493.

4.7 XAMPP

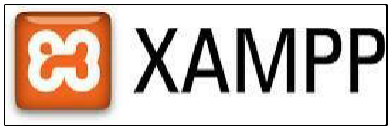


Fig 4.2

Xampp is a free and open source cross platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server — server application (Apache), database (MariaDB), and scripting language (PHP) — is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

4.8 FEATURES

XAMPP is regularly updated to the latest releases of Apache, MariaDB, PHP and Perl. It also comes with a number of other modules including OpenSSL, phpMyAdmin, Media Wiki, Joomla, WordPress and more. Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another. XAMPP is offered in both a full and a standard version (Smaller version).

4.9 USAGE

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. XAMPP has the ability to serve web pages on the World Wide Web. A special tool is provided to password-protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in MariaDB and SQLite among others. Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing a content management system (CMS) like Joomla or WordPress. It is also possible to connect to localhost via FTP with an HTML editor.

CHAPTER-5

SOFTWARE REQUIREMENT SPECIFICATION

5.1 Hardware Requirements

The selection of hardware is very important in the existence and proper working of any software. When selecting hardware, the size and requirements are also important.

Processor	Intel CORE i5
RAM	**4.0 GB**
Hard Disk Drive	**500 GB**

5.2 Software Requirements

Number	**Description**
1	Windows 10
2	HTML/CSS/Ajax/JavaScript/ Bootstrap.
3	Apache server/ XAMPSERVER
4	PHP 5.5.38
4	MySQL
5	Compiler: MSVC11 (Visual C++ 2012)
6	Apache version: Apache/2.4.23 (Win32) OpenSSL/1.0.2h PHP/5.5.38

CHAPTER-6

DATA FLOW DIAGRAMS

Data Flow Diagrams show the flow of data from external entities into the system, and from one process to another within the system. There are four symbols for drawing a DFD:

i. Rectangles representing external entities, which are sources or destinations of data. Ellipses representing processes, which take data as input, validate and process it and output it.

ii. Arrows representing the data flows, which can either, be electronic data or physical items.

iii. Open-ended rectangles or a Disk symbol representing data stores, including electronic stores such as databases or XML files and physical stores such as filing cabinets or stacks of paper. Figures below are the Data Flow Diagrams for the current system. Each process within the system is first shown as a Context Level DFD and later as a Detailed DFD.

iv. The Context Level DFD provides a conceptual view of the process and its surrounding input, output and data stores. The Detailed DFD provides a more detailed and comprehensive view of the interaction among the sub-processes within the system.

Figures below are the Data Flow Diagrams for the current system. Each process within the system is first shown as a Context Level DFD and later as a Detailed DFD. The Context Level DFD provides a conceptual view of the process and its surrounding input, output and data stores. The Detailed DFD provides a more detailed and comprehensive view of the interaction among the sub-processes within the system.

Context Level Diagram

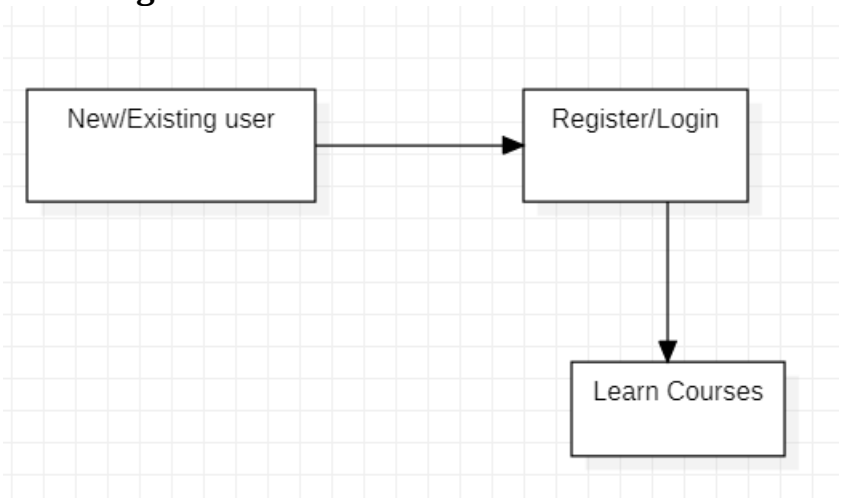


Fig 6.1

LEVEL-I DFD:

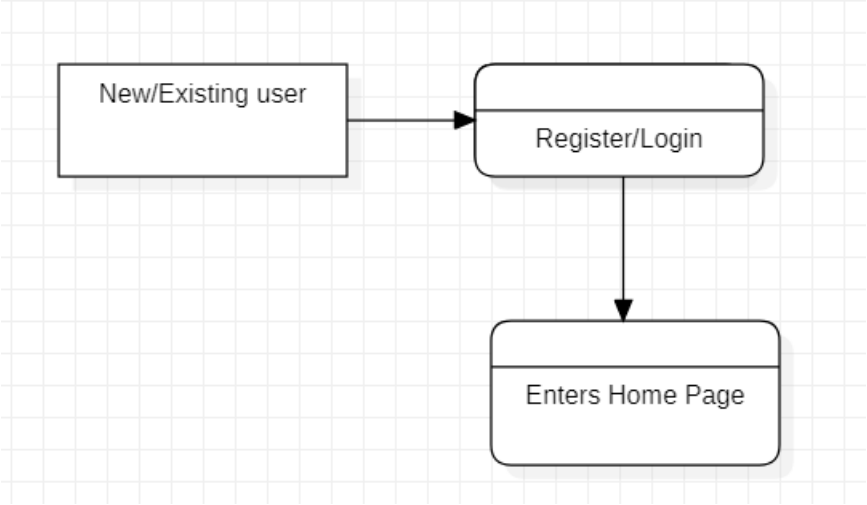


Fig 6.2

LEVEL-II DFD:

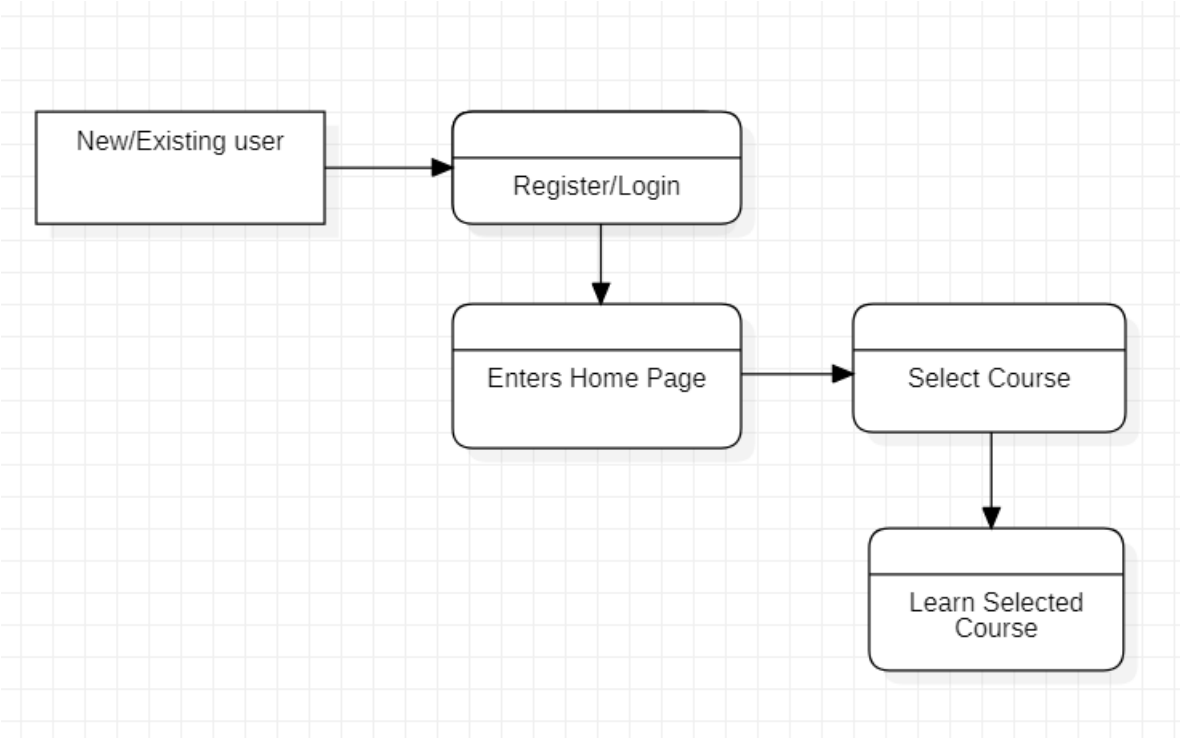


Fig 6.3

UML-USE CASE DIAGRAM

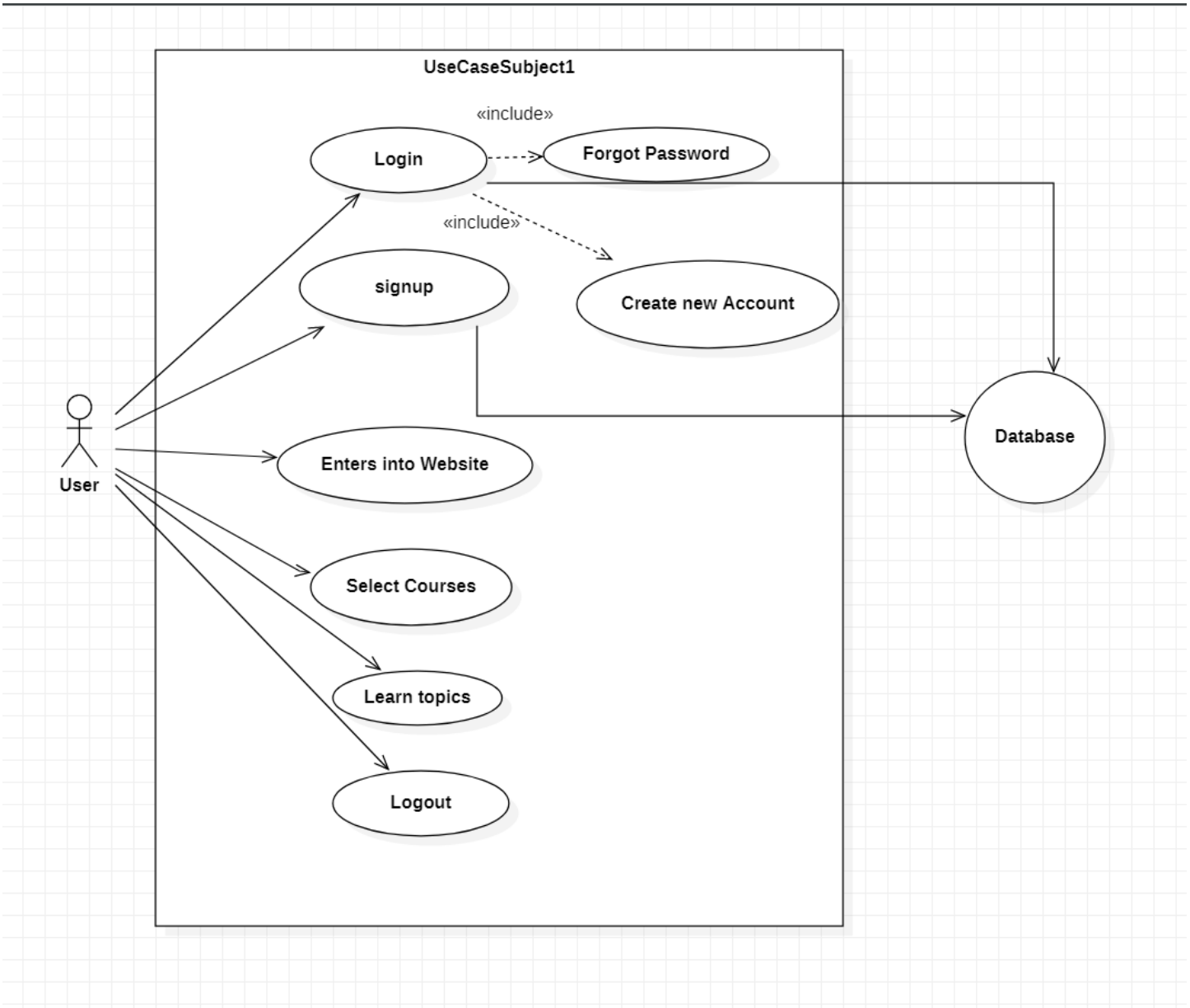


Fig 6.4

ENTITY-RELATIONSHIP Diagram:

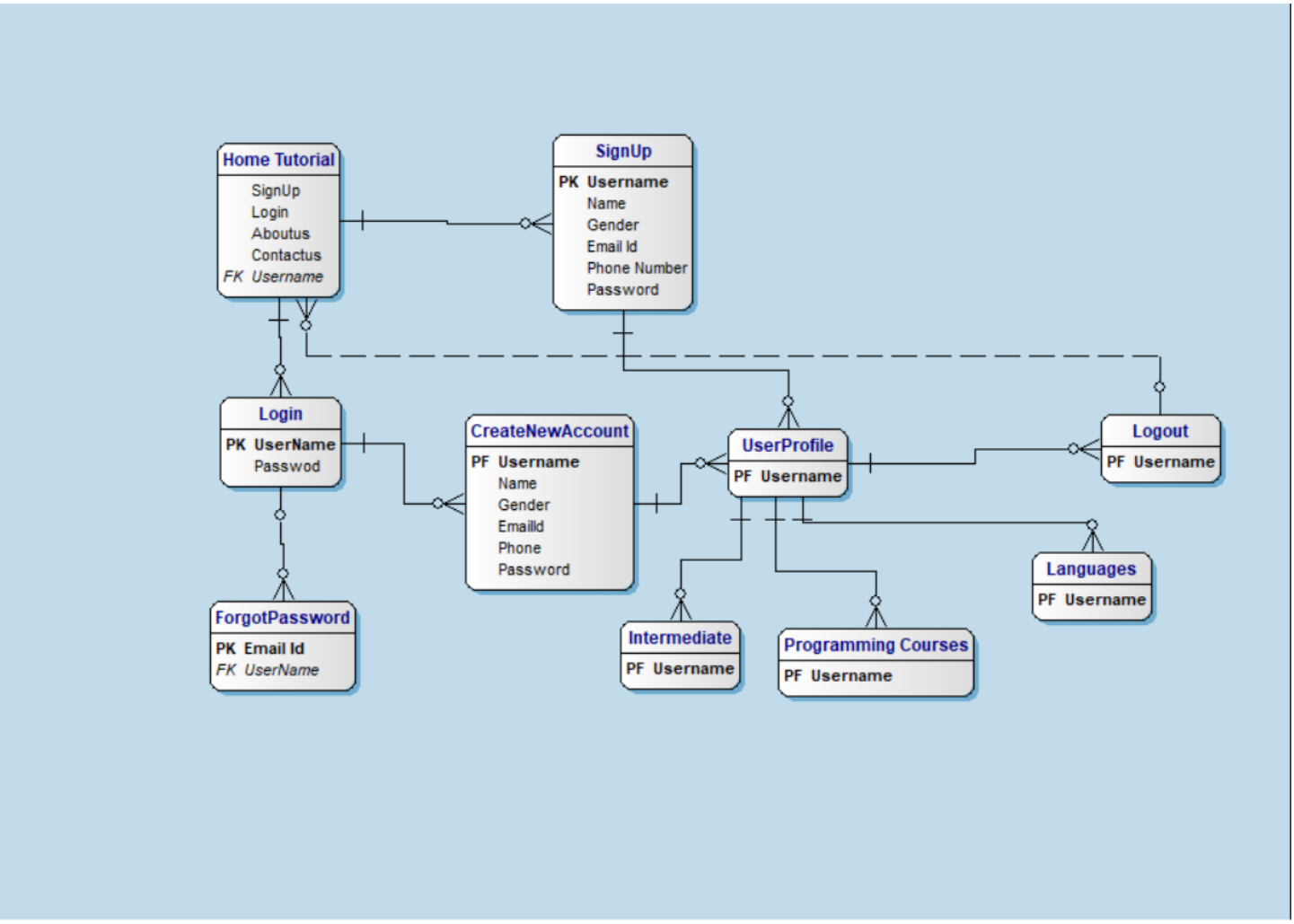


Fig 6.5

CHAPTER-7

PROJECT

Name: HOME TUTORIAL POINT

7.1 Technologies Used:

HTML

CSS

Bootstrap

SQL

Java Script

JSON

jQuery

Node.JS

AJAX

Server: Local Host

Database: MySQL

Operating System: Windows7/8/8.1/10

Wireframing tool: Paint

Team Size: 4

7.2 TECHNICAL DETAILS

Front end is designed using HTML, CSS and Bootstrap. Ajax used to perform behind the screen requests and JavaScript used to perform client-side scripting

Backend is based on MySQL based RDB (Relational Database) model. The SQL queries are run using the CI SQL library functions

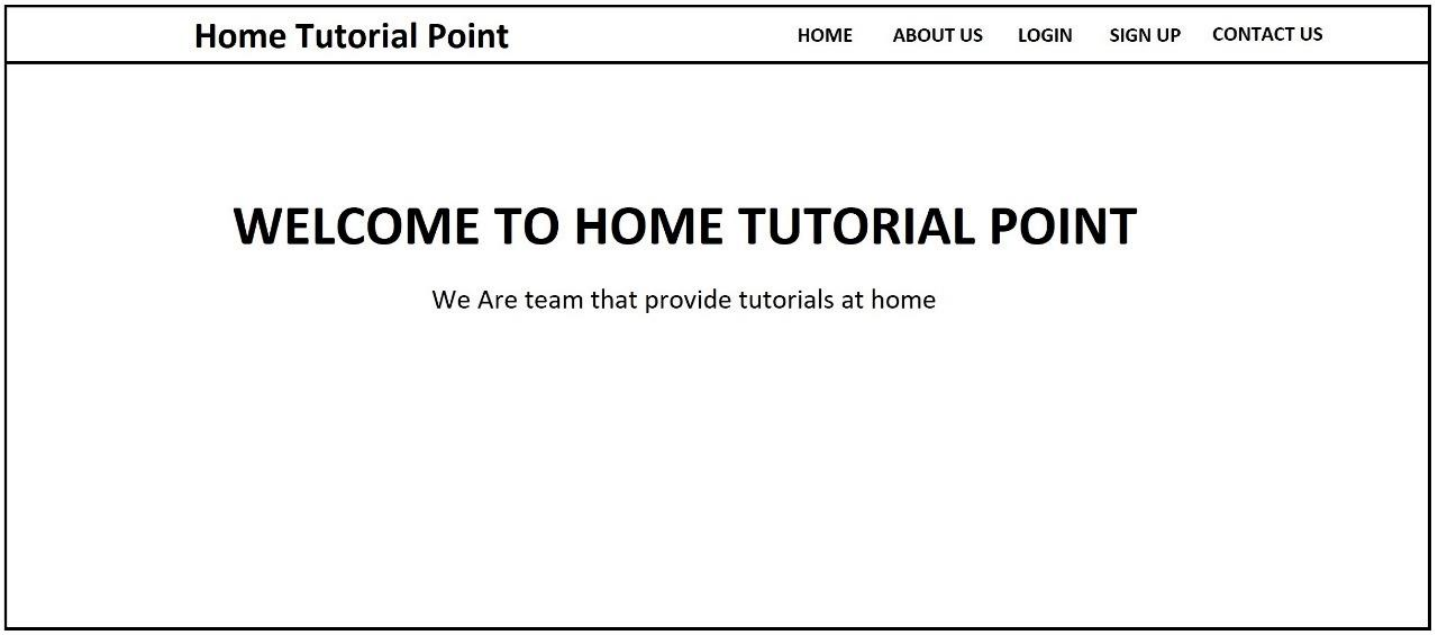
Backend online host includes a centralized database resident on the server, the script which is built in PHP used to SQL query the database on user's request for transaction of data

The forms are made using the HTML, Bootstrap for designing and SQL for back- end JavaScript, AJAX and jQuery used for client-side scripting and PHP for the server- side development

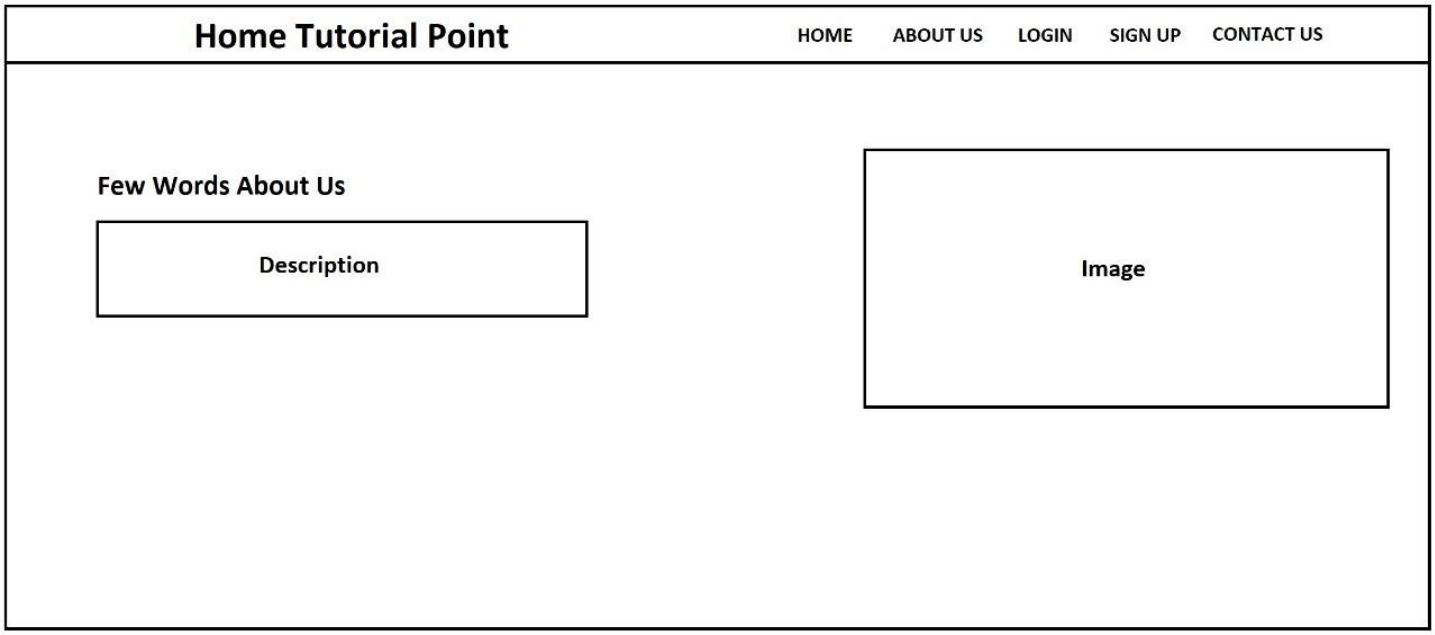
CHAPTER-8

WIREFRAMES

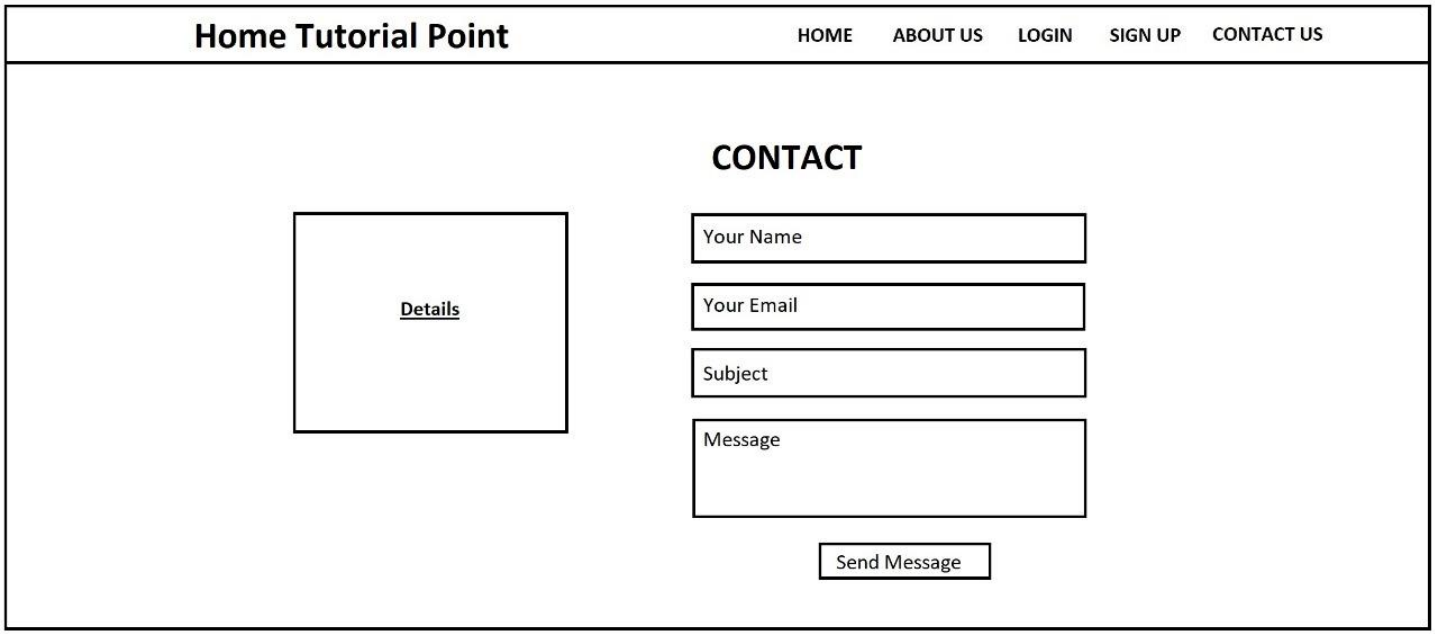
- **HOME PAGE**



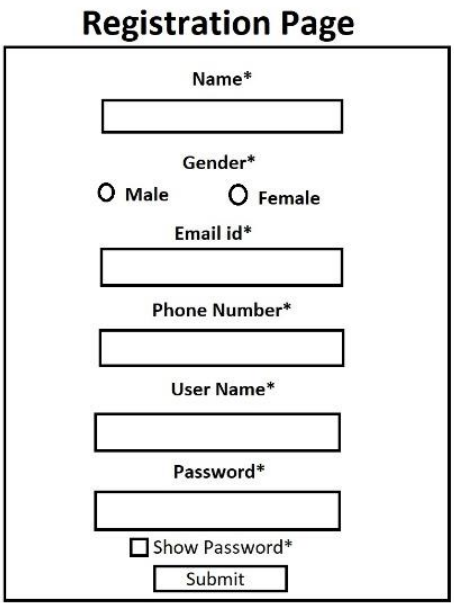
- **ABOUT US**



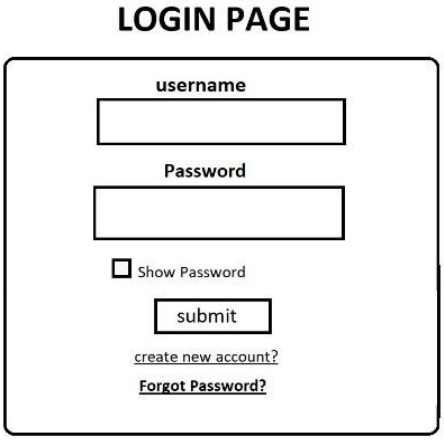
- **CONTACT US**



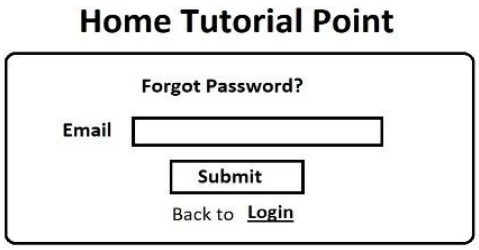
- SIGN UP



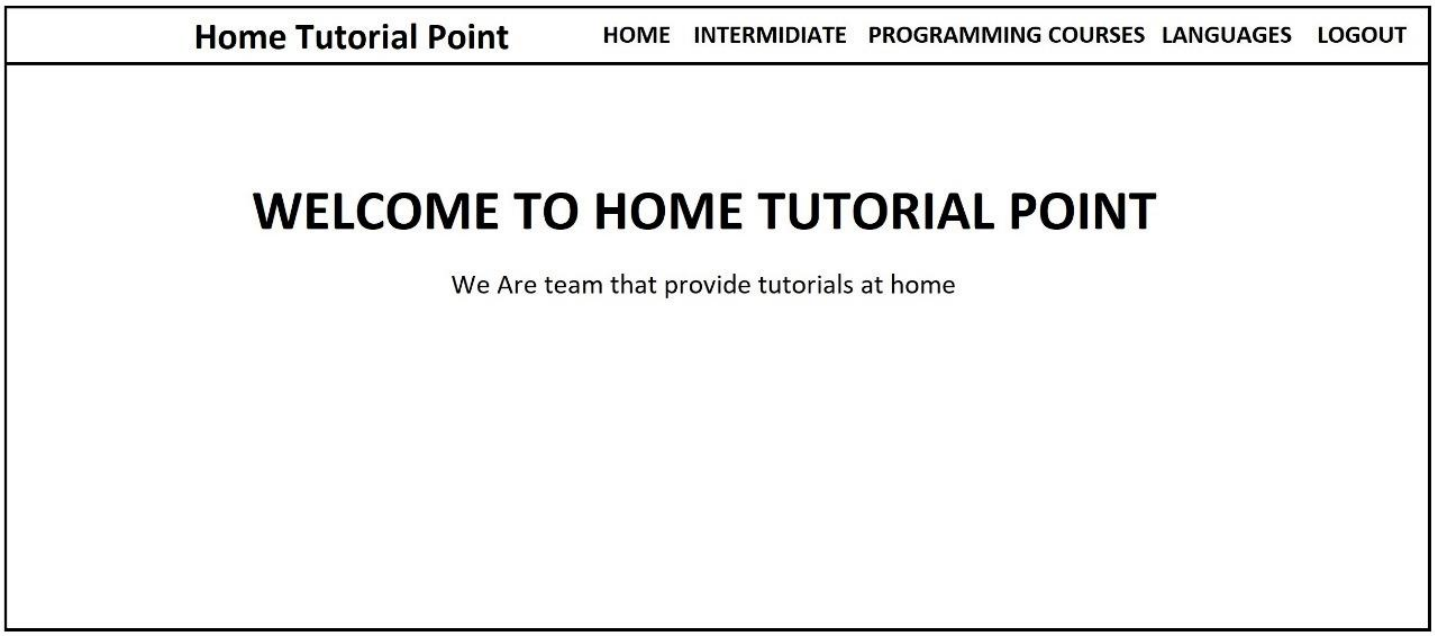
- **LOGIN**



- **FORGOT PASSWORD**



- **USER PAGE**


Home Tutorial Point
HOME
INTERMIDIATE
PROGRAMMING COURSES
LANGUAGES
LOGOUT
WELCOME TO HOME TUTORIAL POINT
We Are team that provide tutorials at home

SCREENSHOTS

HOME PAGE:

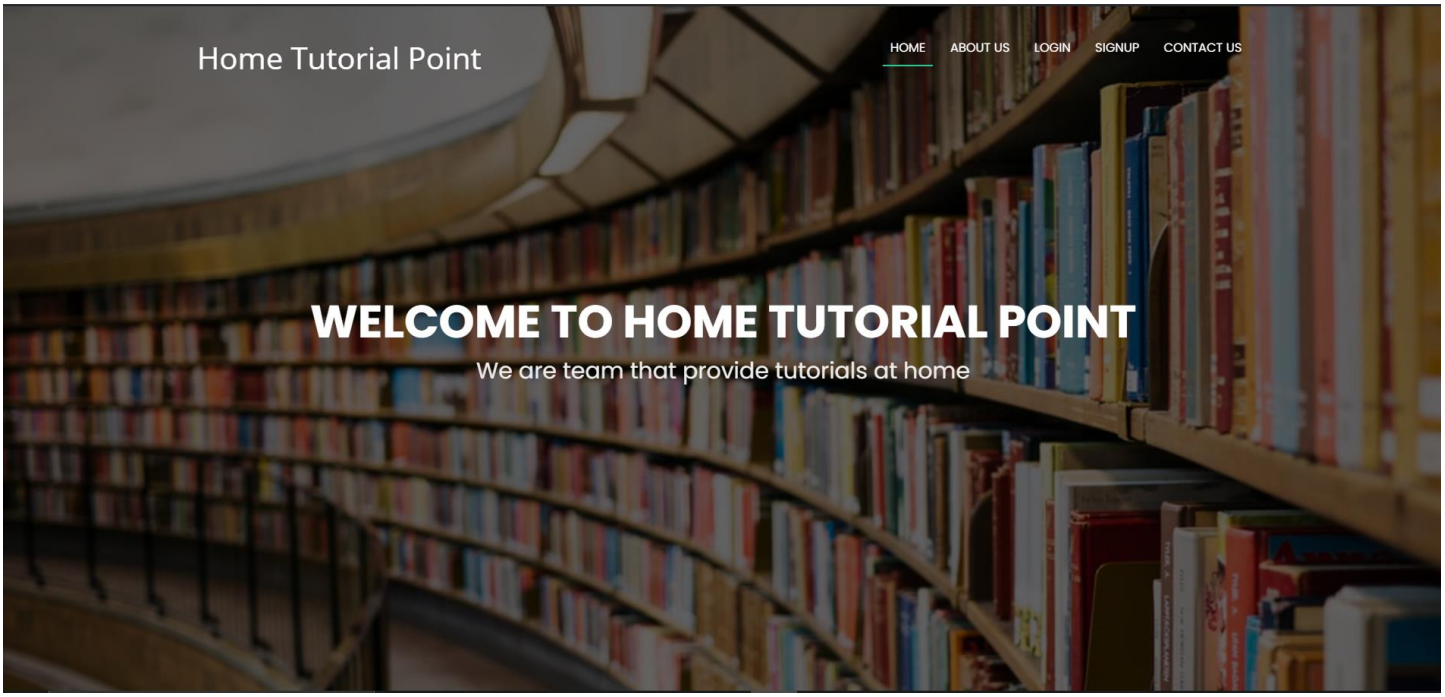


Fig 8.1

ABOUT US:

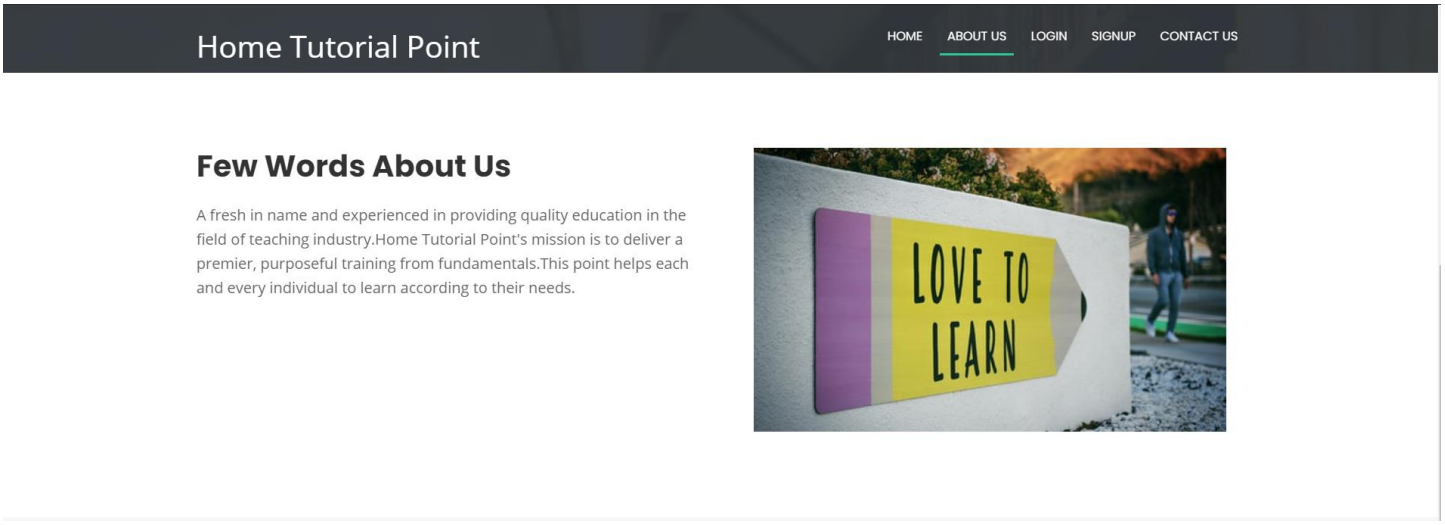


Fig 8.2

CONTACT US:

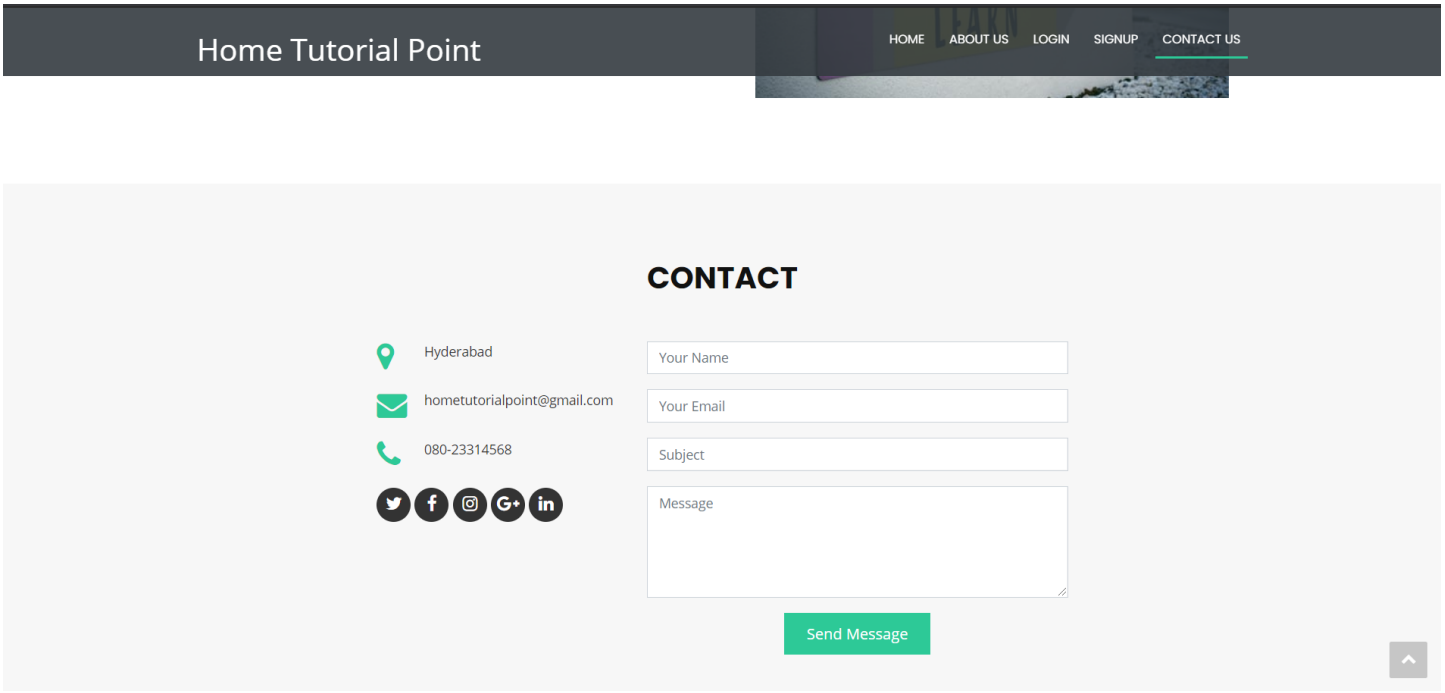


Fig 8.3

SIGNUP:

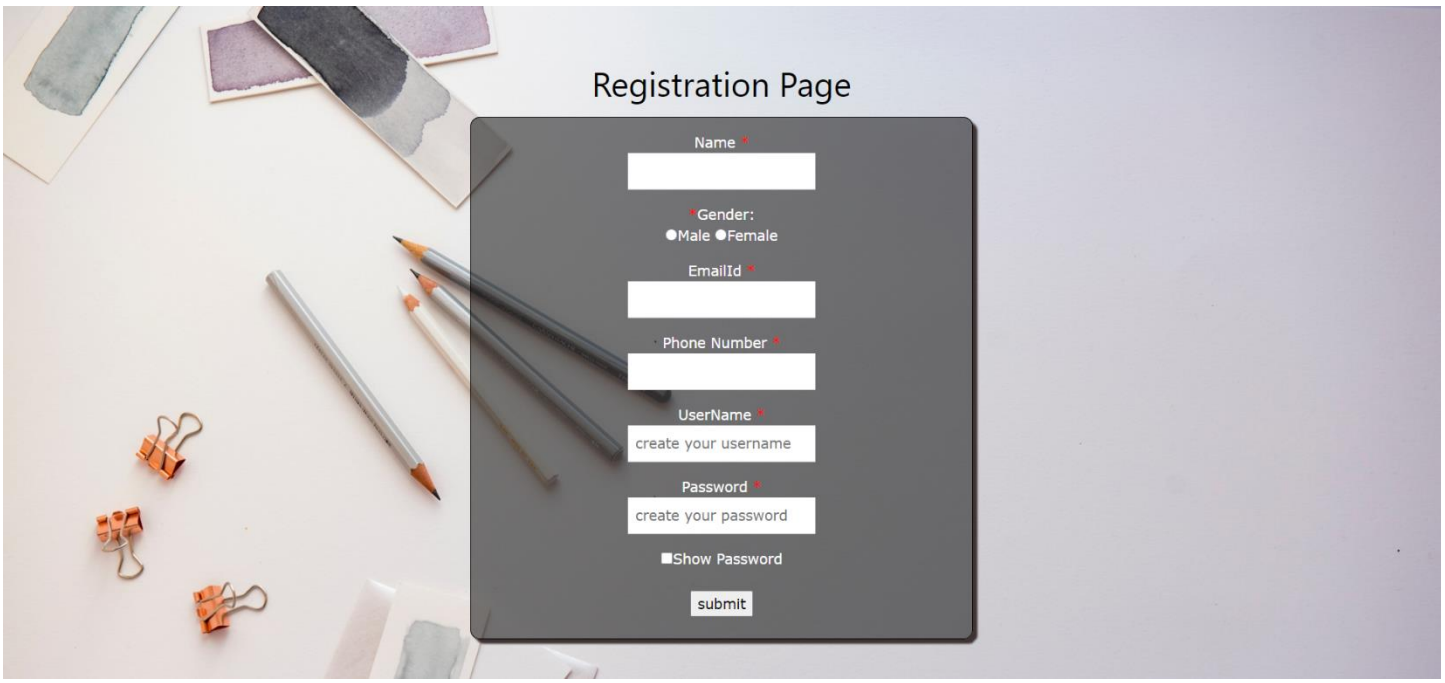


Fig 8.4

LOGIN:

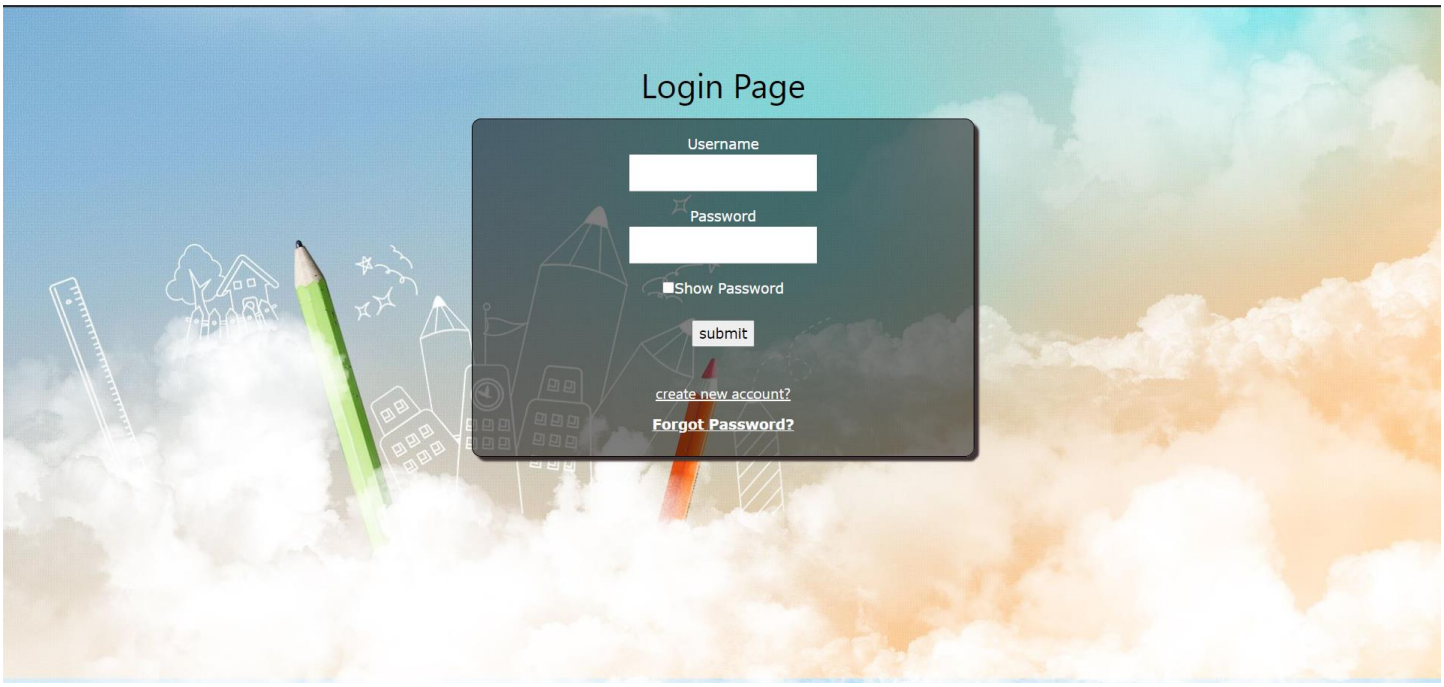


Fig 8.5

FORGOT PASSWORD:

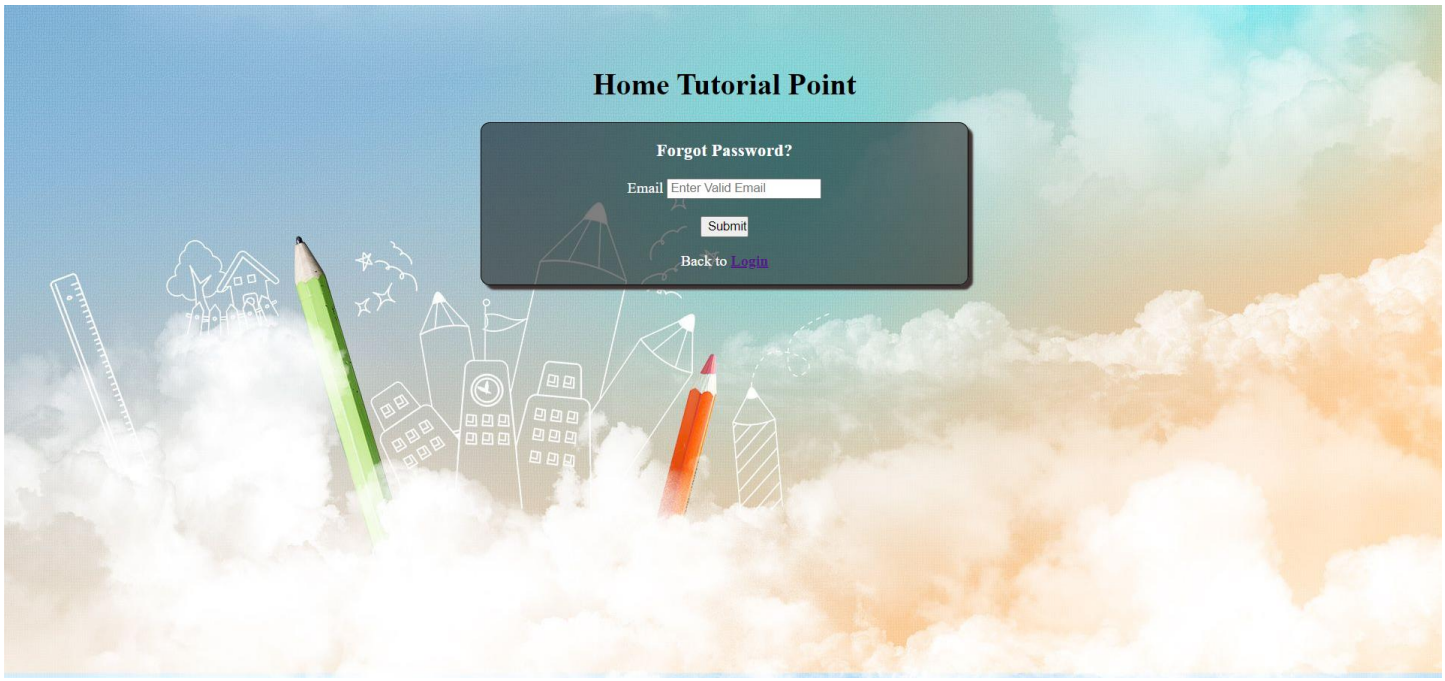


Fig 8.6

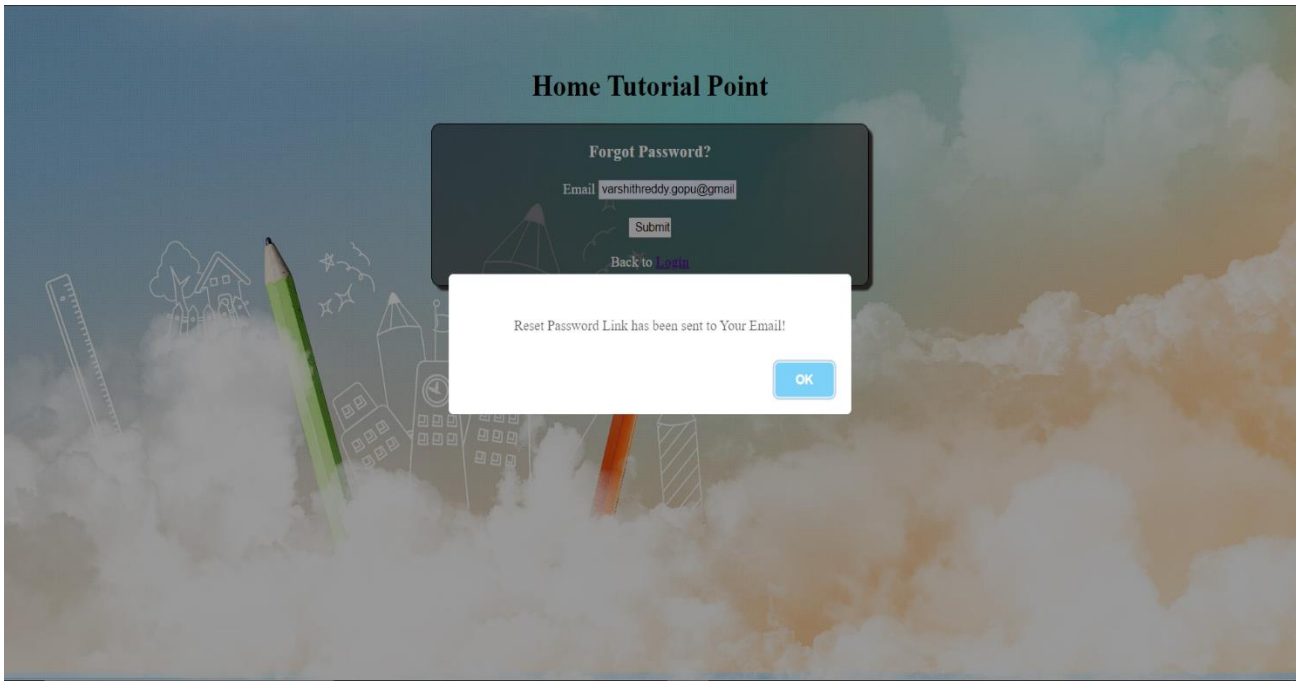


Fig 8.6

USER PAGE:

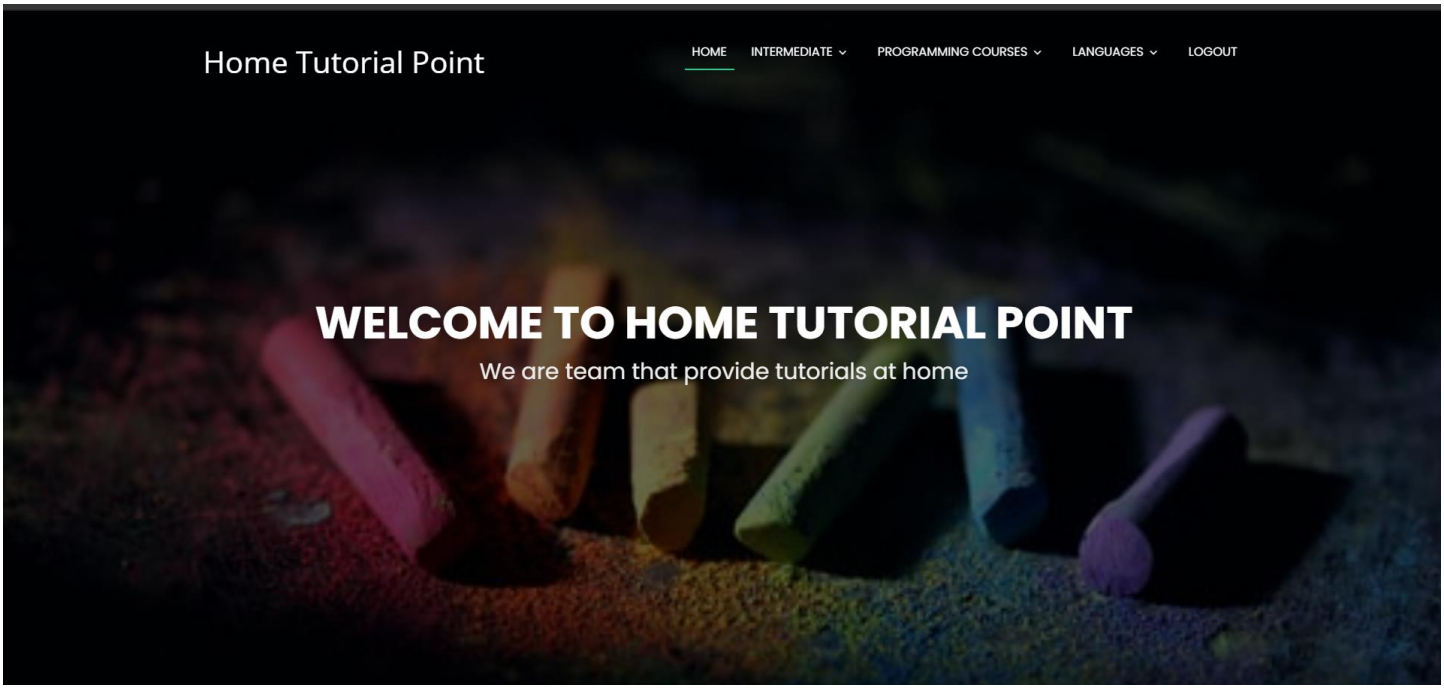
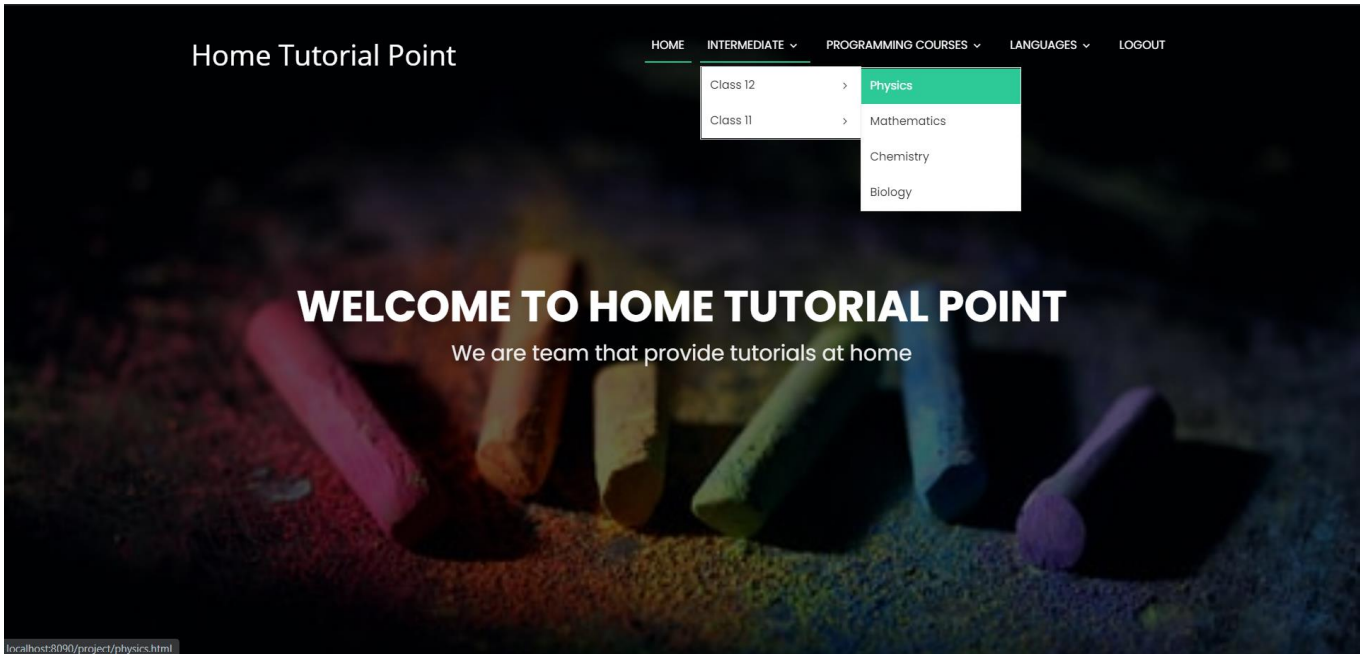
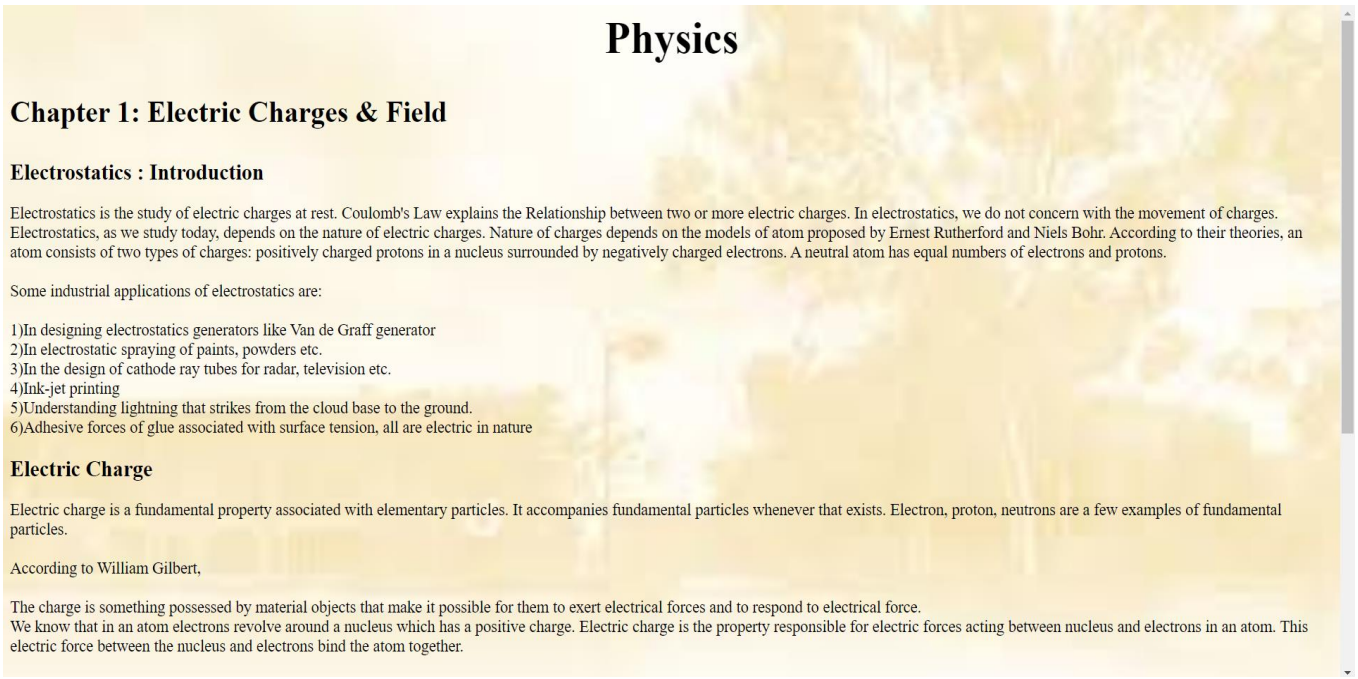


Fig 8.7

COURSE SELECTION:





Physics

Chapter 1: Electric Charges & Field

Electrostatics : Introduction

Electrostatics is the study of electric charges at rest. Coulomb's Law explains the Relationship between two or more electric charges. In electrostatics, we do not concern with the movement of charges. Electrostatics, as we study today, depends on the nature of electric charges. Nature of charges depends on the models of atom proposed by Ernest Rutherford and Niels Bohr. According to their theories, an atom consists of two types of charges: positively charged protons in a nucleus surrounded by negatively charged electrons. A neutral atom has equal numbers of electrons and protons.

Some industrial applications of electrostatics are:

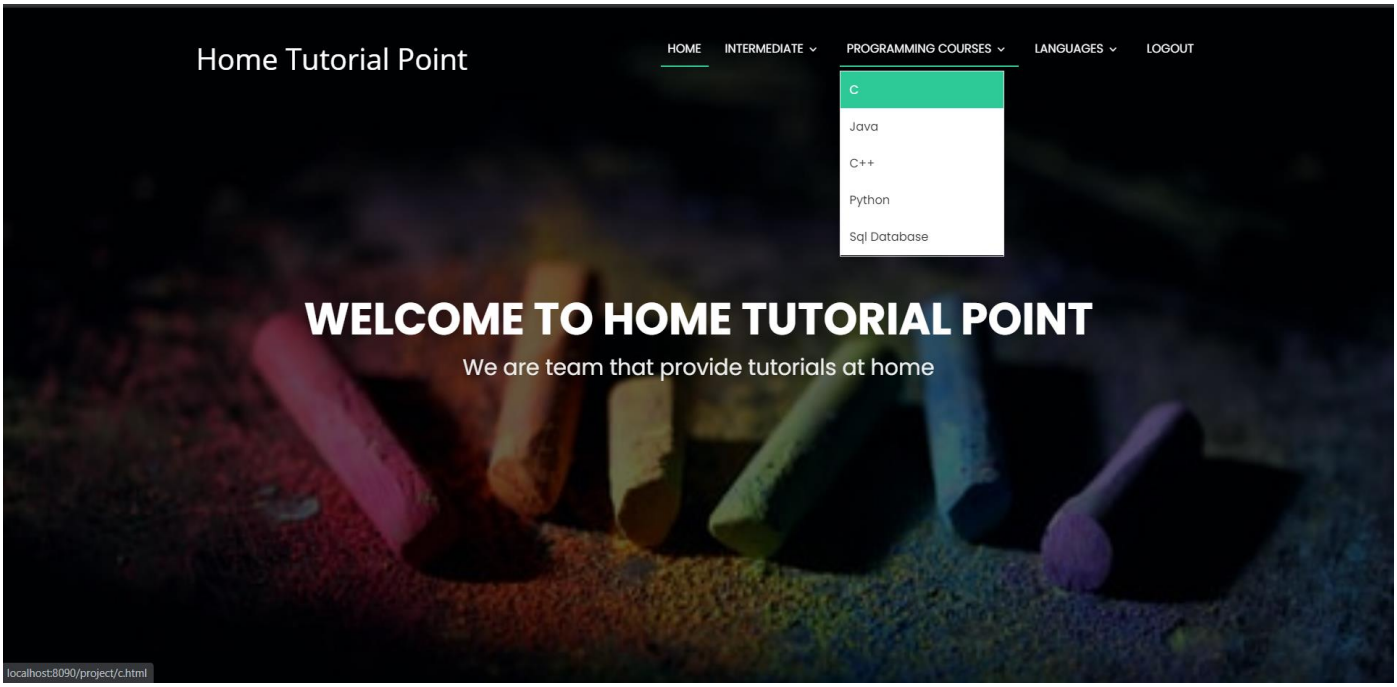
1)In designing electrostatics generators like Van de Graff generator
2)In electrostatic spraying of paints, powders etc.
3)In the design of cathode ray tubes for radar, television etc.
4)Ink-jet printing
5)Understanding lightning that strikes from the cloud base to the ground.
6)Adhesive forces of glue associated with surface tension, all are electric in nature

Electric Charge

Electric charge is a fundamental property associated with elementary particles. It accompanies fundamental particles whenever that exists. Electron, proton, neutrons are a few examples of fundamental particles.

According to William Gilbert,

The charge is something possessed by material objects that make it possible for them to exert electrical forces and to respond to electrical force.
We know that in an atom electrons revolve around a nucleus which has a positive charge. Electric charge is the property responsible for electric forces acting between nucleus and electrons in an atom. This electric force between the nucleus and electrons bind the atom together.



C Programming Language

Audience

This C tutorial series has been designed for those who want to learn C programming; whether you are beginners or experts, tutorials are intended to cover basic concepts straightforwardly and systematically.

Required Knowledge
To learn C Programming language you haven't required any previous programming knowledge, but the basic understanding of any other programming languages will help you to understand the C programming concepts quickly.

C Example

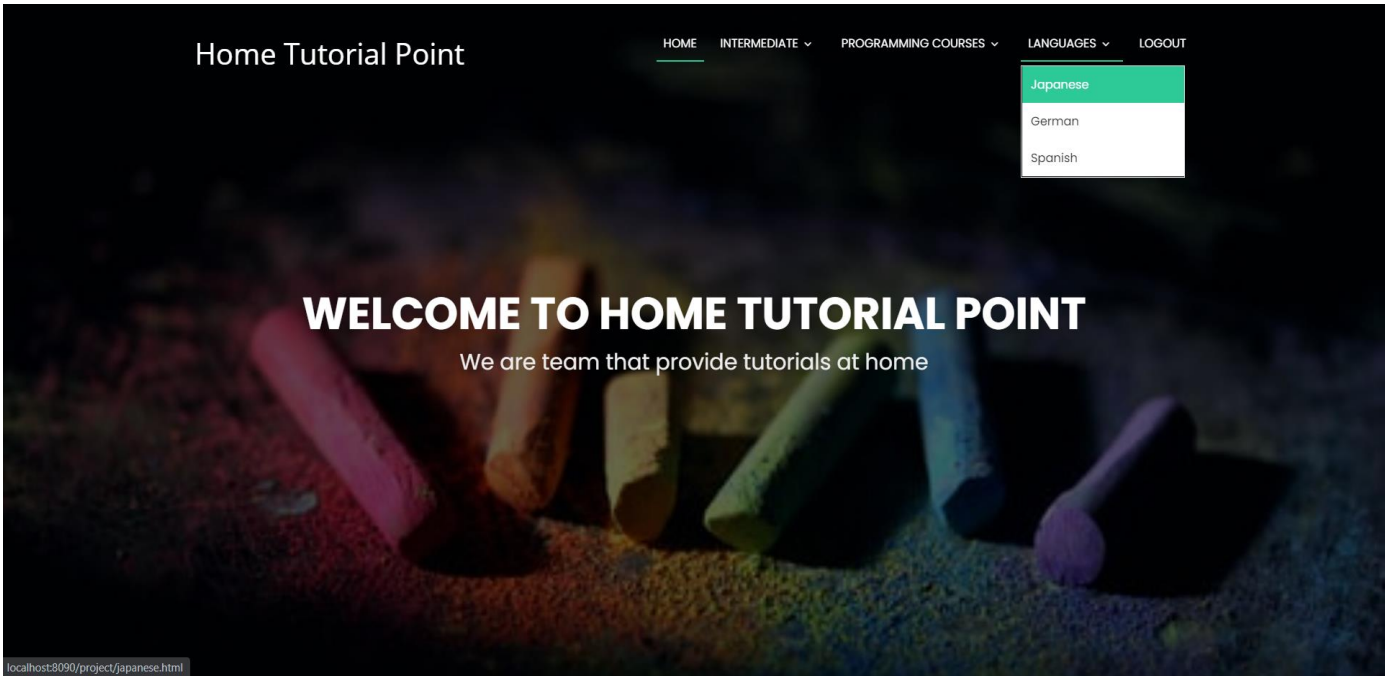
A quick look at the example of Hello, World! In C programming, and detailed description is given in the C Program Structure page.

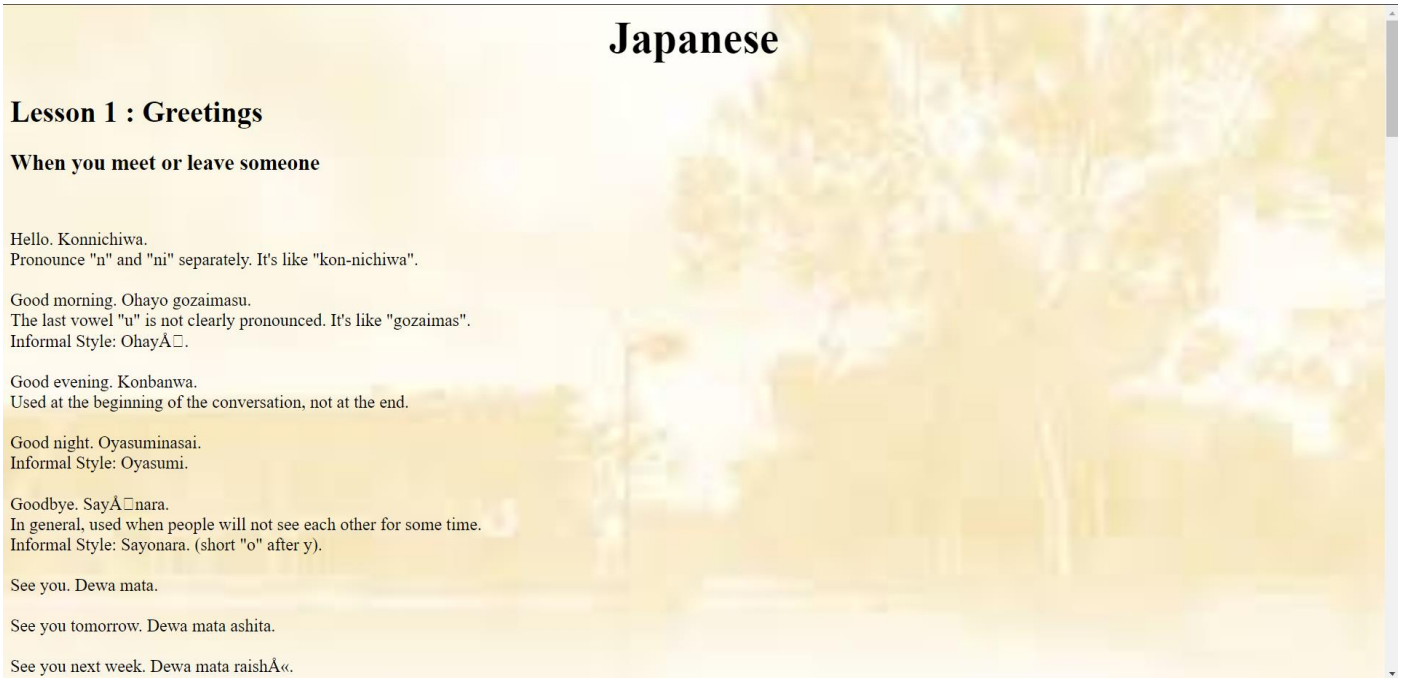
```
#include
int main()
{
printf("Hello, World!\n"); getch(); //Use to get one character input from user, and it will not be printed on screen.
return 0;
```

Some Facts About Programming language

In 1988, the American National Standards Institute (ANSI) had formalized the C language.
1) C was invented to write UNIX operating system.
2) C is a successor of 'Basic Combined Programming Language' (BCPL) called B language.
3) Linux OS, PHP, and MySQL are written in C.
4) C has been written in assembly language.

Uses of C programming Language





Japanese

Lesson 1 : Greetings

When you meet or leave someone

Hello. Konnichiwa.
Pronounce "n" and "ni" separately. It's like "kon-nichiwa".

Good morning. Ohayo gozaimasu.
The last vowel "u" is not clearly pronounced. It's like "gozaimas".
Informal Style: OhayÅ□.

Good evening. Konbanwa.
Used at the beginning of the conversation, not at the end.

Good night. Oyasuminasai.
Informal Style: Oyasumi.

Goodbye. SayÅ□nara.
In general, used when people will not see each other for some time.
Informal Style: Sayonara. (short "o" after y).

See you. Dewa mata.

See you tomorrow. Dewa mata ashita.

See you next week. Dewa mata raishÅ«.

Fig 8.8

HTML CODES

Index Code:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Home Tutorial Point</title>
  <meta content="" name="descriptison">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="assets/img/favicon.png" rel="icon">
  <link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">

  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,700i|Poppins:300,400,500,700" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="assets/vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet">
  <link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
  <link href="assets/vendor/owl.carousel/assets/owl.carousel.min.css" rel="stylesheet">
  <link href="assets/vendor/venobox/venobox.css" rel="stylesheet">
  <link href="assets/vendor/aos/aos.css" rel="stylesheet">

  <!-- Template Main CSS File -->
  <link href="assets/css/style.css" rel="stylesheet">

  <!-- =======================================================
  * Template Name: Regna - v2.1.0
  * Template URL: https://bootstrapmade.com/regna-bootstrap-onepage-template/
  * Author: BootstrapMade.com
  * License: https://bootstrapmade.com/license/
  ======================================================= -->
</head>
<style>
```

```
</style>

<body>

  <!-- ======= Header ======= -->
  <header id="header" class="header-transparent">
    <div class="container">

      <div id="logo" class="pull-left">
        <FONT SIZE=+3 COLOR="WHITE">Home Tutorial Point</font>
        <!-- Uncomment below if you prefer to use a text logo -->
        <!--<h1><a href="#hero">Bitcoin price prediction</a></h1>-->
      </div>

      <nav id="nav-menu-container">
        <ul class="nav-menu">
          <li class="menu-active"><a href="index.html">Home</a></li>
          <li><a href="#about">About Us</a></li>
          <li><a href="loginstu.html">Login</a></li>
          <li><a href="regstu.html">SignUp</a></li>
          <li><a href="#contact">Contact Us</a></li>
        </ul>
      </nav><!-- #nav-menu-container -->
    </div>
  </header><!-- End Header -->

  <!-- ======= Hero Section ======= -->
  <section id="hero">
    <div class="hero-container" data-aos="zoom-in" data-aos-delay="100">
      <h1>Welcome to Home Tutorial Point</h1>
      <h2>We are team that provide tutorials at home</h2>
    </div>
  </section><!-- End Hero Section -->
```

```
<main id="main">

  <!-- ======= About Section ======= -->
  <section id="about">
    <div class="container" data-aos="fade-up">
      <div class="row about-container">

        <div class="col-lg-6 content order-lg-1 order-2">
          <h2 class="title">Few Words About Us</h2>
          <p>
            A fresh in name and experienced in providing quality education in the field of teaching industry.Home Tutorial Point's mission is to deliver a premier,
          </p>
        </div>

        <div class="col-lg-6 background order-lg-2 order-1" data-aos="fade-left" data-aos-delay="100"></div>
      </div>

    </div>
  </section><!-- End About Section -->
```

```
<!-- ======= Contact Section ======= -->
<section id="contact">
  <div class="container">
    <div class="section-header">
      <h3 class="section-title">Contact</h3>
    </div>
  </div>



  <div class="container mt-5">
    <div class="row justify-content-center">

      <div class="col-lg-3 col-md-4">

        <div class="info">
          <div>
            <i class="fa fa-map-marker"></i>
            <p>Hyderabad</p>
          </div>

          <div>
            <i class="fa fa-envelope"></i>
            <p>hometutorialpoint@gmail.com</p>
          </div>

          <div>
            <i class="fa fa-phone"></i>
            <p>080-23314568</p>
          </div>
        </div>

        <div class="social-links">
          <a href="#" class="twitter"><i class="fa fa-twitter"></i></a>
          <a href="#" class="facebook"><i class="fa fa-facebook"></i></a>
```

```
  <a href="#" class="google-plus"><i class="fa fa-google-plus"></i></a>
  <a href="#" class="linkedin"><i class="fa fa-linkedin"></i></a>
/div>

iv>

v class="col-lg-5 col-md-8">
div class="form">
 <form action="forms/contact.php" method="post" role="form" class="php-email-form">
   <div class="form-group">
     <input type="text" name="name" class="form-control" id="name" placeholder="Your Name" data-rule="minlen:4" data-msg="Please enter at least 4 chars" />
     <div class="validate"></div>
   </div>
   <div class="form-group">
     <input type="email" class="form-control" name="email" id="email" placeholder="Your Email" data-rule="email" data-msg="Please enter a valid email" />
     <div class="validate"></div>
   </div>
   <div class="form-group">
     <input type="text" class="form-control" name="subject" id="subject" placeholder="Subject" data-rule="minlen:4" data-msg="Please enter at least 8 chars of subject" />
     <div class="validate"></div>
   </div>
   <div class="form-group">
     <textarea class="form-control" name="message" rows="5" data-rule="required" data-msg="Please write something for us" placeholder="Message"></textarea>
     <div class="validate"></div>
   </div>
   <div class="mb-3">
     <div class="loading">Loading</div>
     <div class="error-message"></div>
     <div class="sent-message">Your message has been sent. Thank you!</div>
   </div>
   <div class="text-center"><button type="submit">Send Message</button></div>
 </form>
/div>
iv>
```

REGISTRATION HTML CODE:

```
<html>
    <head>
        <title>Registration</title>
        <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
    </head>
    <style>
      body{

        background-image:url('images/img3.jpg');
        background-repeat: no-repeat;
        background-size:cover;
        background-attachment: fixed;
}

.box {
    background: transparent;
    border: none;
    border-bottom: 1px solid #000000;
}
.required:after {
    content:" *";
    color: red;
  }
  .form{
    margin-left: 500px;
    margin-right: 500px;
    border: solid black 1px;
    border-radius: 10px;
    box-shadow: 5px 5px 2px rgb(63, 52, 52);
    background-color: rgb(0,0,0,0.5);
    color: white;

}

        </style>
```

```
<body>
    <script>
        function myFunction() {
          var x = document.getElementById("myInput");
          if (x.type === "password") {
            x.type = "text";
          } else {
            x.type = "password";
          }
        }
        function abc()
        {
          window.alert("you are succesfully registered");
        }
        </script>
    <br><br><center><h1>Registration Page</h1></center>
    <center>

    <form action="regstu1.php" method="POST" class="form">
        <p>
            <label>Name</label><label class="required"></label>
            <input class="w3-input" type="text" style="width:200px;" name="name" required="required"></p>
            <label class="required"></label>Gender:<br>
            <input type="radio" name="gender" value="male">Male
            <input type="radio" name="gender" value="female">Female<br>
            <p>
            <label>EmailId</label><label class="required"></label>
            <input class="w3-input" type="text" style="width:200px;" name="emailid"  required="required"></p>
            <p >
             <label >Phone Number </label><label class="required"></label>
            <input class="w3-input" type="text" style="width:200px;" name="phone" minlength="1" maxlength="10" required="required"></p>
            <p>
              <label>UserName</label><label class="required"></label>
              <input class="w3-input" type="text" style="width:200px;" name="username" placeholder="create your username" required="required"></p>
            <p>
            <label>Password</label><label class="required"></label>
```

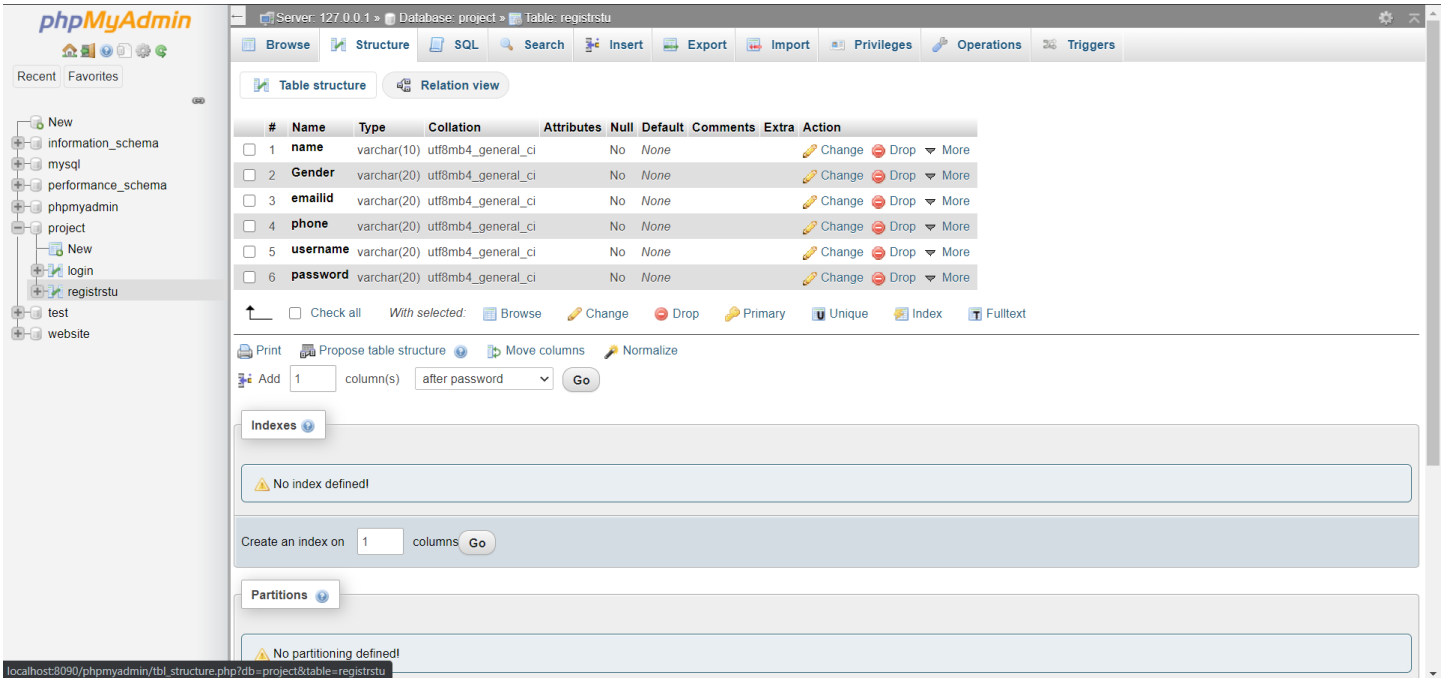
```
            <p>
            <label>Password</label><label class="required"></label>
            <input class="w3-input" type="password" style="width:200px;" name="password" id="myInput" placeholder="create your password" required="required"></p>
            <input type="checkbox" onclick="myFunction()">Show Password

        <br><br>
        <input type="submit" name="submit" value="submit" ><br><br>


    </form>

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

REGISTRATION PHP CODE:

```
<html>
        <head>
                <title>Registration</title>
</head>
<body>
        <?php
        $name = $_POST['name'];
        $gender=$_POST['gender'];
        $emailid=$_POST['emailid'];
        $phone=$_POST['phone'];
        $username1=$_POST['username'];
        $password = $_POST['password'];

        $conn = mysqli_connect("localhost","project","project");
        $db = mysqli_select_db($conn,"project");
        function function_alert($message) {

                // Display the alert box
                echo "<script>alert('$message');";
                echo 'window.location.href = "Home.html";';
                echo"</script>";
            }

            $query1="SELECT * from login  where username='$username1'";
            $result1 = mysqli_query($conn,$query1) or die("Query failed: ".mysqli_error($conn));
            if(mysqli_num_rows($result1)!=0){
                    echo "<script>alert('there exists a user with this username');";
                    echo 'window.location.href = "regstu.html";';
                   echo " </script>";
             }
```

```
            else
               {
                 $query = "INSERT INTO login VALUES ('$username1', '$password')" ;
                 $query2=   "INSERT INTO registrstu VALUES ('$name','$gender','$emailid','$phone','$username1','$password')" ;
                 $result = mysqli_query($conn,$query) or die("Query failed: ".mysqli_error($conn));
                 $result2 = mysqli_query($conn,$query2) or die("Query failed: ".mysqli_error($conn));
                 }
                 function_alert("You are Successfully registered");
                 //header('Location: hii.html');

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

REGISTRATION TABLE:



LOGIN HTML CODE:

```
<html>
    <head>
        <title>LoginPage</title>
        <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

    <style>
      body {
        background-image:url('images/img12.jpg');
        background-repeat: no-repeat;
        background-size:cover;
        background-attachment: fixed;
      }

.box {
    background: transparent;
    border: none;
    border-bottom: 1px solid #000000;
}
.form{
    margin-left: 500px;
    margin-right: 500px;
    border: solid black 1px;
    border-radius: 10px;
    box-shadow: 5px 5px 2px rgb(63, 52, 52);
    background-color: rgb(0,0,0,0.5);
    color: white;

}

        </style>
        </head>
    <body>
        <script>
            function myFunction() {
              var x = document.getElementById("myInput");
              if (x.type === "password") {
                x.type = "text";
```

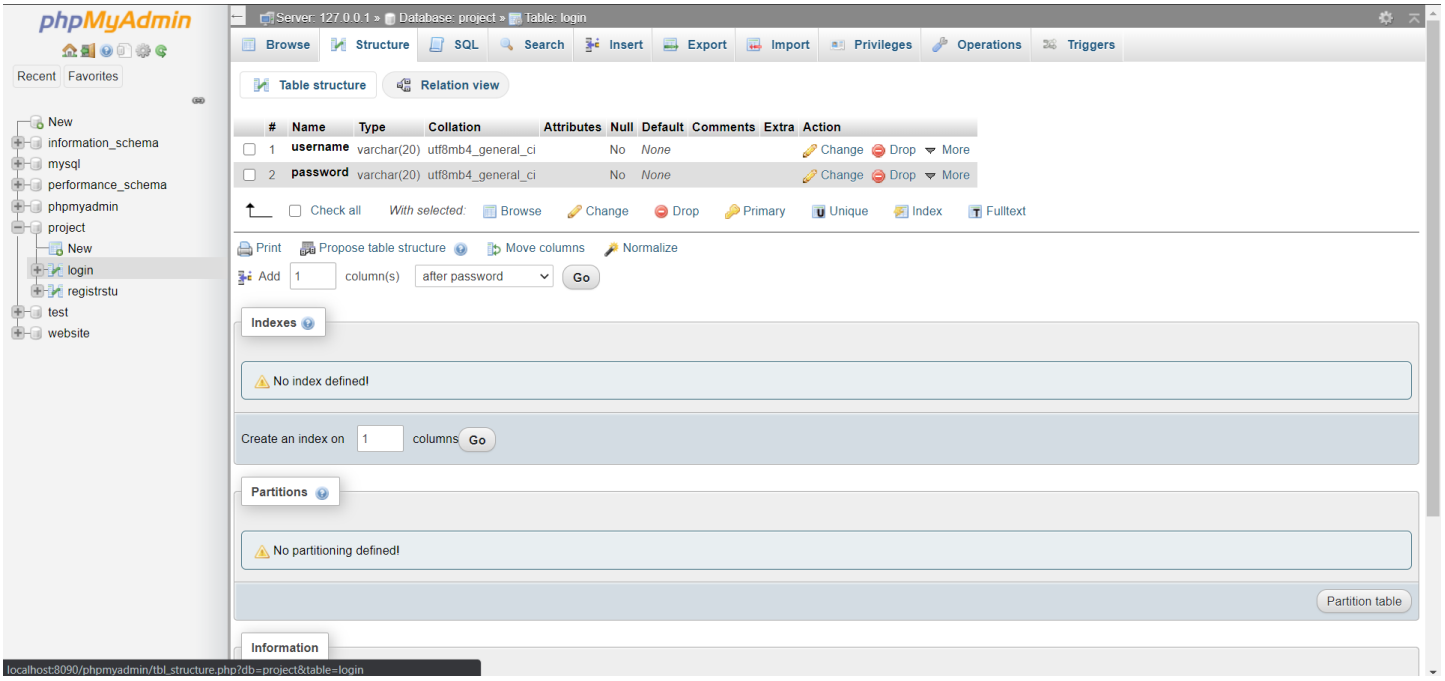
```
            } else {
              x.type = "password";
            }
          }
          </script>
        <br><br><center><h1>Login Page</h1> </center>
      <center>
        <form action="loginstu1.php" method="POST" class="form">
          <p>
            <label>Username</label>
            <input class="w3-input" type="text" style="width:200px;" name="name"></p>
            <p>
            <label>Password</label>
            <input class="w3-input" type="password" style="width:200px;" name="pwd" id="myInput"></p>
            <p>
              <input type="checkbox" onclick="myFunction()">Show Password


          <br><br>
          <input type="submit" name="submit" value="submit"><br><br>
          <a href="regstu.html"><h6>create new account?</h6></a>
          <a href="forgotpassword.html" class="a1"><b>Forgot Password?</b></a><br /><br />
        </form>
      </center>
    </body>
</html>
```

LOGIN PHP CODE:

```
<html>
        <head>
                <title>Login</title>
</head>
<style>
.alert {
  padding: 20px;
  background-color: #f44336;
  color: white;
  opacity: 1;
  transition: opacity 0.6s;
  margin-bottom: 15px;
}

.alert.success {background-color: black;}
.alert.info {background-color: #2196F3;}
.alert.warning {background-color: #ff9800;}

.closebtn {
  margin-left: 15px;
  color: white;
  font-weight: bold;
  float: right;
  font-size: 22px;
  line-height: 20px;
  cursor: pointer;
  transition: 0.3s;
}

.closebtn:hover {
  color: black;
}
</style>
<body>
        <?php
        $name = $_POST['name'];
        $pwd = $_POST['pwd'];
```

```
</style>
<body>
        <?php
        $name = $_POST['name'];
        $pwd = $_POST['pwd'];

        $conn = mysqli_connect("localhost","project","project");
        $db = mysqli_select_db($conn,"project");
        $query = "SELECT * FROM login WHERE username='$name' and password='$pwd'";
        $result = mysqli_query($conn,$query) or die("Query failed: ".mysqli_error($conn));
        if(mysqli_num_rows($result)!=0)
        {
                echo "<script>";
                echo 'window.location.href = "Home.html";';
                echo "</script>";
        }
        else{
              echo "<script>alert('username or password is incorrect please enter valid details');";
              echo 'window.location.href = "loginstu.html";';
              echo "</script>";
        }
        ?>
</body>
        </html>
```

LOGIN TABLE:



FORGOT PASSWORD CODE:

```
<!DOCTYPE html>
<html>
    <head>
        <title>forgotpassword</title>
        <link href="bootstrap-4.5.0-dist/bootstrap-4.5.0-dist/css/bootstrap.min.css" rel="StyleSheet"/>
        <link href="forgotpassword.css" rel="StyleSheet" />
        <script src="bootstrap-4.5.0-dist/bootstrap-4.5.0-dist/js/bootstrap.min.js"></script>
        <script src="bootstrap-4.5.0-dist/bootstrap-4.5.0-dist/js/jquery.min.js"></script>
        <script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
        <script src="fpvalid.js"></script>
        <style>


body{
    background-image: url(images/img12.jpg);
    background-size: cover;
    background-repeat: no-repeat;

}
.form{
    margin-left: 500px;
    margin-right: 500px;
    border: solid black 1px;
    border-radius: 10px;
    box-shadow: 5px 5px 2px rgb(63, 52, 52);
    background-color: rgb(0,0,0,0.5);
    color: white;

}
.tit{
    text-align: left;
    vertical-align: middle;
    height:100px ;
    font-size: xx-large;
    line-height: 100px;
}
```

```
.title{
    background-color: black;
    color: whitesmoke;
}
.log{
    float: left;
    padding: 10px;
}
.row{
    padding-top: 50px;
}
        </style>
    </head>
    <body>
        <script>
            function validate(){
            var em=document.getElementById('email').value;
            var e=/^[a-zA-Z0-9.-_]{6,20}[@][a-z]{3,15}[.][a-z]{2,6}$/;
            if(em=='')
                {
                swal('Please enter the Email');
                return false;
                }
            else{
                if(e.test(em)){
                    swal('Reset Password Link has been sent to Your Email!')
                    }
                else{
                    swal('Please enter the valid Email.... ');
                    return false;
                    }
                }
                }
        </script>
                    <br><br><center><h1>Home Tutorial Point</h1> </center>
                    <form action="" method="POST"  class="form" >
```

```
        </script>
                    <br><br><center><h1>Home Tutorial Point</h1> </center>
                    <form action="" method="POST"  class="form" >
                        <h3 style="text-align: center;">Forgot Password?</h3>
                        <div class="form-group" >
                          <center> <label for="Email">Email</label>
                            <input type="email" class="form-control" id="email" placeholder="Enter Valid Email" required />
                        </div>
                        <br>
                    </center>
                        <center><button type="button" class="btn btn-primary" id="btn" style="width: 10%;" onclick="validate()" >Submit</button></center>

                      <center> <p>Back to <a href="loginstu.html"><b>Login</b></a></p></center>
                    </form>

    </body>
</html>
```

USER PAGE CODE:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Home Tutorial Point</title>
  <meta content="" name="descriptison">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="assets/img/favicon.png" rel="icon">
  <link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">

  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700,700i|Poppins:300,400,500,700" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="assets/vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet">
  <link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
  <link href="assets/vendor/owl.carousel/assets/owl.carousel.min.css" rel="stylesheet">
  <link href="assets/vendor/venobox/venobox.css" rel="stylesheet">
  <link href="assets/vendor/aos/aos.css" rel="stylesheet">

  <!-- Template Main CSS File -->
  <link href="assets/css/style.css" rel="stylesheet">

  <!-- =======================================================
  * Template Name: Regna - v2.1.0
  * Template URL: https://bootstrapmade.com/regna-bootstrap-onepage-template/
  * Author: BootstrapMade.com
  * License: https://bootstrapmade.com/license/
  ======================================================= -->
</head>
```

```
<style>
  .button1 {
  font-family: "Poppins", sans-serif;
  text-transform: uppercase;
  font-weight: 500;
  font-size: 16px;
  letter-spacing: 1px;
  display: inline-block;
  padding: 8px 28px;
  text-align: center;
  border-radius: 50px;
  transition: 0.5s;
  margin-left: 500px;
  margin-top:300px;
  margin-bottom:300px;
  border: 2px solid #fff;
}
.button1:hover {
  background: #2dc997;
  border: 2px solid #2dc997;
}
.button2
{
  font-family: "Poppins", sans-serif;
  text-transform: uppercase;
  font-weight: 500;
  font-size: 16px;
  letter-spacing: 1px;
  display: inline-block;
  padding: 8px 28px;
  text-align: center;
  border-radius: 50px;
  transition: 0.5s;
  margin-left: 500px;
  margin-top:300px;
  margin-bottom:300px;
  border: 2px solid #fff;
```

```
}
.button2:hover{
  background: #2dc997;
  border: 2px solid #2dc997;
}

#hero{
        background-image:url('images/img5.jpg');
        background-repeat: no-repeat;
        background-size:cover;
        background-attachment: fixed;
      }

</style>

<body>

  <!-- ======= Header ======= -->
  <header id="header" class="header-transparent">
    <div class="container">

      <div id="logo" class="pull-left">
        <FONT SIZE=+3 COLOR="WHITE">Home Tutorial Point</font>
        <!-- Uncomment below if you prefer to use a text logo -->
        <!--<h1><a href="#hero">Bitcoin price prediction</a></h1>-->
      </div>

      <nav id="nav-menu-container">
        <ul class="nav-menu">
          <li class="menu-active"><a href="Home.html">Home</a></li>
          <li class="menu-has-children"><a href="">Intermediate</a>
            <ul>
              <li class="menu-has-children"><a href="#">Class 12</a>
               <ul>
                 <li><a href="physics.html">Physics</a></li>
                 <li><a href="Maths.html">Mathematics</a></li>
```

```
                <li><a href="chemistry.html">Chemistry</a></li>
                <li><a href="biology.html">Biology</a></li>
              </ul>
            </li>
            <li class="menu-has-children"><a href="#">Class 11</a>
              <ul>
                <li><a href="physics.html">Physics</a></li>
                <li><a href="Maths.html">Mathematics</a></li>
                <li><a href="chemistry.html">Chemistry</a></li>
                <li><a href="biology.html">Biology</a></li>
              </ul>
            </li>
          </ul>
        </li>
        <li class="menu-has-children"><a href="">Programming Courses</a>
          <ul>
            <li><a href="c.html">C</a></li>
            <li><a href="java.html">Java</a></li>
            <li><a href="c++.html">C++</a></li>
            <li><a href="python.html">Python</a></li>
            <li><a href="sql Database.html">Sql Database</a></li>
          </ul>
        </li>
        <li class="menu-has-children"><a href="">Languages</a>
          <ul>
            <li><a href="japanese.html">Japanese</a></li>
            <li><a href="german.html">German</a></li>
            <li><a href="spanish.html">Spanish</a></li>
          </ul>
        </li>
        <li><a href="index.html">Logout</a></li>
      </ul>
    </nav><!-- #nav-menu-container -->
  </div>
</header><!-- End Header -->
```

```
            </ul>
          </li>
          <li><a href="index.html">Logout</a></li>
        </ul>
      </nav><!-- #nav-menu-container -->
    </div>
  </header><!-- End Header -->

  <!-- ======= Hero Section ======= -->
  <section id="hero">
    <div class="hero-container" data-aos="zoom-in" data-aos-delay="100">
      <h1>Welcome to Home Tutorial Point</h1>
      <h2>We are team that provide tutorials at home</h2>

    </div>
  </section><!-- End Hero Section -->
```

INTERMEDIATE COURSES:
PHYSICS CODE:

```
<html>
    <style>
        body {
        background-image:url('images/img11.jpg');
        background-repeat: no-repeat;
        background-size:cover;
        background-attachment: fixed;
      }
    </style>
    <body>

<h2>
  <center><h1>Physics</h1></center>
   <h1><p>Chapter 1: Electric Charges & Field</p></h1>
    <p><h2>Electrostatics : Introduction</h2></p>
    <p style="font-size: large;">Electrostatics is the study of electric charges at rest. Coulomb's Law explains the Relati
        <br>Electrostatics, as we study today, depends on the nature of electric charges. Nature of charges depends on the
        <br><br>Some industrial applications of electrostatics are:<br>
        <br>1)In designing electrostatics generators like Van de Graff generator
        <br>2)In electrostatic spraying of paints, powders etc.
        <br>3)In the design of cathode ray tubes for radar, television etc.
        <br>4)Ink-jet printing
        <br>5)Understanding lightning that strikes from the cloud base to the ground.
        <br>6)Adhesive forces of glue associated with surface tension, all are electric in nature </p>
        <p><h2>Electric Charge</h2></p>
        <p style="font-size: large;">Electric charge is a fundamental property associated with elementary particles. It acc
        <br><br>According to William Gilbert,
        <br><br>The charge is something possessed by material objects that make it possible for them to exert electrical fo
        <br>We know that in an atom electrons revolve around a nucleus which has a positive charge. Electric charge is the
<br><br>Charges are of two kinds
<br>(i) negative charge
<br>(ii) positive charge
<br><br>In an atom electron are particles having a negative charge? The nucleus consists of protons and neutrons. In a nucl

<br>(1) like charges repel,like charges repel each other
```

```
<br>(1) like charges repel,like charges repel each other

<br>(2) unlike charges attract,unlike charges attract each other

<br><br>The electric force between two electrons is the same as the electric force between two protons kept at the
The electric force between an electron and proton placed at the same distance apart is not repulsive but attractive
Assignment of a negative charge on the electron and a positive charge on a proton is only a convention. It does not
<br><br><b>Unit of charge</b><br>
1)The charge on a system can be measured by comparing it with the charge on a standard body.<br>
2)SI unit of charge is Coulomb written as C.<br>
3)Coulomb is the charge flowing through the wire in 1 second if the electric current in it is 1A.<br>
4)Charge on electron is e=−1.602×10−19C and charge on proton is positive of this value.<br>
        </p>
</h2>
</body>
</html
```

MATHEMATICS CODE:

```
<html>
    <style>
        body {
        background-image:url('images/img11.jpg');
        background-repeat: no-repeat;
        background-size:cover;
        background-attachment: fixed;
      }
    </style>
    <body>

<h2>
    <h1><center>Maths</center></h1>
   <h1><p>Chapter 1:Relations and Funtions</p></h1>
    <p><h2>Relation:</h2></p>
    <p style="font-size: large;">A relation R from set X to a set Y is defined as a subset of the cartesian product X × Y. We can also write it as R ⊆ {(x, y)

      <br> Note: If n(A) = p and n(B) = q from set A to set B, then n(A × B) = pq and number of relations = 2pq.
    <br><br><b>Types of Relation</b>
    <br><br><b>Empty Relation:</b> A relation R in a set X, is called an empty relation, if no element of X is related to any element of X,
    i.e. R = ϕ ⊂ X × X
    <br><br><b>Universal Relation: </b>A relation R in a set X, is called universal relation, if each element of X is related to every element of X,
       i.e. R = X × X
      <br><br><b> Reflexive Relation:</b> A relation R defined on a set A is said to be reflexive, if
       (x, x) ∈ R, ∀ x ∈ A or
       xRx, ∀ x ∈ R

      <br><br><b> Symmetric Relation: </b>A relation R defined on a set A is said to be symmetric, if
       (x, y) ∈ R ⇒ (y, x) ∈ R, ∀ x, y ∈ A or
       xRy ⇒ yRx, ∀ x, y ∈ R.
      <br><br><b> Transitive Relation:</b> A relation R defined on a set A is said to be transitive, if
       (x, y) ∈ R and (y, z) ∈ R ⇒ (x, z) ∈ R, ∀ x, y, z ∈ A
       or xRy, yRz ⇒ xRz, ∀ x, y,z ∈ R.

      <br><br><b> Equivalence Relation: </b>A relation R defined on a set A is said to be an equivalence relation if R is reflexive, symmetric and transitive.
```

```
      of xRy, yRz ⇒ xRz, ∀ x, y,z ∈ R.

      <br><br><b> Equivalence Relation: </b>A relation R defined on a set A is said to be an equivalence relation if R is ref

       <br><br><b>Equivalence Classes:</b> Given an arbitrary equivalence relation R in an arbitrary set X, R divides X into
</p>
<h1><p>Chapter 2:Inverse Trignometric functions</p></h1>
<p><h2>Domain and Range of Inverse Trignometric functions:</h2></p>
<p>
    <img src="images/inverse.jpg" height="300px" width="500px"/>
</p>
<p><h2>Properties of Inverse Trignometric functions:</h2></p>
<p>
    <img src="images/propertiesinverse.jpg" height="500px" width="700px"/>
</p>
<p style="font-size: large;">Following substitutions are used to write inverse trigonometric functions in simplest form:<br>
    <img src="images/simple.jpg" height="500px" width="700px"/>
</p>
</p>
<h1><p>Chapter 3:Continuity and Differentiality</p></h1>
<p style="font-size: large;"><b>Continuity at a Point:</b> A function f(x) is said to be continuous at a point x = a, if
    Left hand limit of f(x) at(x = a) = Right hand limit of f(x) at (x = a) = Value of f(x) at (x = a)
    i.e. if at x = a, LHL = RHL = f(a)
    where, LHL = \lim _{ x\rightarrow { a }^{ - } }{ f(x) } and RHL = \lim _{ x\rightarrow { a }^{ + } }{ f(x) }
    Note: To evaluate LHL of a function f(x) at (x = o), put x = a – h and to find RHL, put x = a + h.
<br><br><b>Standard Results of Limits</b><br>
<img src="images/limits.jpg" height="300px" width="500px"/>
<br><br><b>Some Standard Derivatives</b><br>
<img src="images/derivatives.jpg" height="300px" width="500px"/>
</p>
</p>
</p>
</h2>
</body>
</html
```

CHEMISTRY CODE:

```
<html>
    <style>
        body {
        background-image:url('images/img11.jpg');
        background-repeat: no-repeat;
        background-size:cover;
        background-attachment: fixed;
      }
    </style>
    <body>

<h2>
  <h1><center>Chemistry</center></h1>
   <h1><p>Chapter1: Solid State</p></h1>
   <p style="font-size: large;">1)Solids are substances which have fixed shape
    and volume. 1'hey are characterised by rigidity, incompressibility, slow diffusion and mechanical strength. <br>They are cla
    <br>(a) Crystalline solids
    <br>(b) Amorphous solids .
<br><br>2)The crystalline solids are further classified as:<br><br>
(a) Metallic solids
<br>(b) Ionic solids
<br>(c) Covalent solids
<br>(d) Molecular solids
<br><br>3)A regular three dimensional arrangement of points in space is called a space lattice or crystal lattice. There are onl
<br><br>4)A unit cell is the smallest unit of the crystal which when repeated again and again gives the crystal of the given sub
<br><br>5)There are three types of unit cells based on the cube. These are:
<br>(a) Primitive or simple cube which has one constituent at each comer.
<br>(b) Body-centred cube in which one constituent at the centre of the cube as well as one at each comer.
<br>(c) Face-centred cube in which there is one constituent at the centre of each face as well as one at each comer.
<br><br>6)A pure metal in the solid crystalline state is composed of atoms that are identical in shape and size. The identical s
<br><br>7)The number of nearest neighbours of an atom, ion or a molecule is called its coordination number.
<br><br>8) In the hcp and ccp structures, about 74 percent of the available space is occupied by the spheres. In bcc arrangement
<br><br>9) The density of the unit cell,
<br><img src="images/img20.jpg" height="70px" width="70px"/>
</p>
```

```
<h1><p>Chapter2:Solutions</p></h1>
<p style="font-size: large;">
  <ol style="font-size:larger;">
<li>
  A solution is a homogeneous mixture of two or 9.
  more chemically non-reacting substances.
  The components of a solution generally cannot be separated by filtration, settling or centrifuging.</li>
 <li>A solution may be classified as solid, liquid or a gaseous solution.</li>
 <li>Solubility is defined as the amount of solute in a saturated solution per 100g of a solvent.</li>
 <li>The solubility of a gas in a liquid depends upon
  <br>(a) the nature of the gas and the nature of the liquid,
  <br>(b) the temperature of the system, and
 <br> (c) the pressure of the gas.</li>
 <li>The effect of pressure on the solubility of a gas in a liquid is governed by Henry's Law. It states that the solubility of a g
 <li>The vapour pressure of a liquid is the pressure exerted by its vapour when it is in dynamic equilibrium with its liquid, in a
 <li>According to Raoults Law, the vapour pressure of a solution containing a non-volatile solute is directly proportional to the m
<li>A solution which obeys Raoult's Law at all concentrations and temperatures is known as an ideal solution.</li>
<li>Characteristics of an ideal solution:
  <br>(a) Δsol V = 0, i.e., there is no change in volume when an ideal solution is formed.
  <br>(b) Δsol H= 0; i.e., heat is neither evolved nor absorbed during the formation of an ideal solution.</li>
  <li>(a) The solution shows positive deviation from Raoult's Law if its vapour pressure is higher than that predicted by Raoult's
    <br>(b) The solution shows negative deviation if its vapour pressure is lower than that predicted by Raoult's Law</li>
    <li>Colligative properties of solutions are those properties which depend only upon the number of solute particles in the solut
     <br> (a) Relative lowering in vapour pressure,
      <br>(b) Elevation of boiling point,
      <br>(c) Depression of freezing point and
      <br>(d) Osmotic pressure.</li>
</ol>
</p>
```

```
 <h1><p>Chapter3: Polymers</p></h1>
 <p>
   <ol style="font-size: larger;">
   <li>A polymer is a large molecule of high molecular mass formed by the repetitive bonding of many small molecules called monomers.
      The process by which the monomers are transformed into polymers is called polymerisation. As polymers are single big size molecules,
      they are also called  macromolecules.</li>
    <li>Classification of polymers on the basis of source:
     <br>(a) Natural polymers: Proteins, cellulose, starch, resins and rubber.
     <br>(b) Semi-synthetic polymers: Cellulose derivatives as cellulose acetate (rayon) and cellulose nitrate, etc.
     <br>(c) Synthetic polymers: Plastic (polythene), synthetic fibres (nylon 6, 6) and synthetic rubbers (Buna-S).</li>
     <li> Classification based on structure of polymers:
      <br> (a) Linear polymers: They consists of long and straight chains, e.g., high density polythene and PVC.
      <br> (b) Branched chain polymers: They contain linear chains having some branches, e.g., low density polyethene.
       <br>(c) Cross linked or Network polymers: Those are formed from bifunctional and tri-functional monomers and contain strong
       covalent bonds between various linear polymer chains e.g, bakelite and melamine.</li>
  <li>Classification based on mode of polymerisation:
   <br>(a) Addition polymers: They are formed by the addition reactions between monomers having multiple bonds, e.g., polythene.
   <br>(b) Condensation polymers: They are formed by the condensation reactions between two monomers, each monomers having two functional
    groups, with the elimination of small molecules such as water, alcohol and ammonia, e.g., Nylon 6,6.</li>
 <li>Classification based on molecular forces:
  <br> (a) Elastomers
  <br> (b) Fibres
   <br>(c) Thermoplastics.
   <br>(d) Thermosetting plastics</li>
   <li>
     There are two broad types of polymerisation reaction:
     <br>(i) Addition or chain growth polymerisation: It is a polymersation in which monomers having one or more double bonds undergo repeated addi
    <br> (ii) Condensation polymerisation or Step growth polymerisation: It occurs when monomers condense in a stepwise manner with elimination of
   </li>
   <li>Copolymerisation: It is a process in which a mixture of more than one monomeric species polymerize to form a copolymer. A copolymer contains
   <li> Natural rubber is cis 1,4-polyisoprene. It is a linear 1,4-polymer of isoprene. It is manufactured from rubber latex which is a colloidal s
   <li>The process of heating a mixture of raw rubber and sulphur at 373 K to 415 K is known as vulcanisation of rubber. The process of vulcanisati
   <li> Biodegradable polymers: PHBV and Nylon 2- Nylon-6 are developed to minimise the environmental hazards of synthetic polymeric wastes.</li>
  </ol>
 </p>
 </h2>
 </body>
 </html
```

BIOLOGY CODE:

```
<html>
    <style>
        body {
        background-image:url('images/img11.jpg');
        background-repeat: no-repeat;
        background-size:cover;
        background-attachment: fixed;
      }
    </style>
    <body>
        <h2>
<center><h1>Biology</h1></center>
<h1>Chapter 1:Evolution</h1>
<p style="font-size: larger;">
<ol style="font-size: larger;">
<li>The study of history of life forms on earth is called evolutionary biology.</li>
<br>
<li>Evolution is a process that results in heritable changes in population spread over many generations leading to diversity of organisms on earth.</li>
<br>
<li> Origin of life is considered a unique event in the history of universe.
    <br><b><u>(i) The Universe</u></b>
    <br> (a) It is very old-almost 20 billion years ago. It contains huge galaxies.
    <br>(b) Galaxies contain stars and clouds of gas and dust.
    <br>(c) The origin of universe is explained by Big Bang theory.
    <br>(d) The Big Bang theory states that a huge explosion occurred, the universe expanded, temperature came down and hydrogen and helium were formed l
    <br><b><u>(ii) The earth was supposed to have been formed about 4.5 billion years back in the solar system of the milkyway galaxy.</u></b>
    <br>(a) Water vapour, methane, carbon dioxide and ammonia released from molten masses covered the surface.
    <br>(b) UV rays from the sun broke up water molecule into hydrogen and oxygen and lighter hydrogen escaped.
    <br>(c) Oxygen combined with ammonia and methane to form water, carbon dioxide and others.
    <br>(d) Ozone layer formed, as it cooled, the water vapour fell as rain to fill depression and form oceans.
    <br>(e) Life appeared 500 million (about 4 billion years back) years after the formation of earth.</li>
    <br>
    <li>Theories of origin of life were given by different thinkers and scientists.
        <br>(i) Theory of special creation states that God created life by his divine act of creation.
        <br> (ii) Theory of panspermia/cosmozoic theory, given by early Greek thinkers states that the spores or panspermia came from outer space and dev
        <br> (iii) Theory of spontaneous generation states that life originated from decaying and rotting matter like straw, mud, etc.
```

```
    <br>(a) Water vapour, methane, carbon dioxide and ammonia released from molten masses covered the surface.
    <br>(b) UV rays from the sun broke up water molecule into hydrogen and oxygen and lighter hydrogen escaped.
    <br>(c) Oxygen combined with ammonia and methane to form water, carbon dioxide and others.
    <br>(d) Ozone layer formed, as it cooled, the water vapour fell as rain to fill depression and form oceans.
    <br>(e) Life appeared 500 million (about 4 billion years back) years after the formation of earth.</li>
    <br>
    <li>Theories of origin of life were given by different thinkers and scientists.
        <br>(i) Theory of special creation states that God created life by his divine act of creation.
        <br> (ii) Theory of panspermia/cosmozoic theory, given by early Greek thinkers states that the spores or panspermia came from o
        <br> (iii) Theory of spontaneous generation states that life originated from decaying and rotting matter like straw, mud, etc.
        <br>(a) Louis Pasteur rejected the theory of spontaneous generation and demonstrated that life came from pre-existing life.
        <br>(b) In his experiment, he kept killed yeast cells in pre-sterilised flask and another flask open into air. The life did not
        <br> (iv) Theory of chemical evolution or Oparin-Haldane theory states that life originated from pre-existing non-living organi
        <br>The conditions on the earth that favoured chemical evolution were very high temperature, volcanic storms and reducing atmos
<br>
        <li> Millers experiment provided experimental evidence for chemical evolution.
          <br> (i) The experiment was carried out by SL Miller and HC Urey in 1953.
          <br> (ii) He took a closed flask containing CH4,H2,NH3 and water vapour at 800°C and created electric discharge. These condi
          <br> (iii) After a week, formation of amino acids were observed. Complex molecules like sugars, nitrogen bases, pigments and
           <br>(iv) Analysis of the meteorite also revealed the presence of similar compounds.
           <br>(v) Chemical evolution of life was more or less accepted.
        <br><img src="images/img21.jpg" height="500px" width="700px"/></li>
    </ol>
</p>
<h1><p>Chapter2:Human Health and disease</p></h1>
<p style="font-size: larger;">
<ol style="font-size: larger;">
<li><b>Health </b>is defined as a state of complete physical, mental and social well-being.
    <br>(i) The factors which affect human health are:
    <br>(a) Genetic disorders (b) Infections (c) Lifestyle.
    <br> (ii) Balanced diet, personal hygiene and regular exercise are very important to maintain good health.
    <br> (Hi) Awareness about diseases and their effect on different body functions, vaccination against infectious diseases, proper di
<br>
    <li>Disease is a state when functioning of one or more organs or systems of the body is adversely affected, characterised by variou
    <br>Diseases can be divided broadly into the following two categories depending on the
    modes of transmission
```

```
    <br>II. Non-infectious diseases which does not transmit from one person to another, e.g. cancer, diabetes, etc.
    <br>(i) Disease causing organisms are called pathogens, e.g. bacteria, viruses, fungi, protozoans, helminthes, etc.
    <br> (ii) The above described pathogens enter the body by direct contact, contaminated food and water, droplet infection, etc.
    <br>(iii)The pathogens multiply in body cells, interfere with normal vital activities, cause morphological and functional damage.
    <br>(iv) Infectious diseases can be divided into certain categories, based on the type of pathogen
    Infectious Diseases
    Bacterial Viral Protozoan Fungal Helminthic
    <br>e.g. pneumonia, e.g. common cold, e.g. amoebiasis, e.g. ringworm, e.g. ascariasis, ,
    plague, typhoid, polio, etc. malaria, etc. athelete's foot, etc. filariasis, taeniasis, etc.
    diphtheria, etc
<br>    (c) Life cycle of Plasmodium is given in the figure.</li>

<br>
<li>Some of the common infectious diseases are:
  <br> <b> I. Bacterial Diseases</b>
  <br> <u>(i) Typhoid is caused by bacterium (Salmonella typhi).</u>
  <br> (a) S. typhi enters the small intestine through food and water contaminated with them and migrate to other organs through blood.
  <br> (b) Intestinal perforation and death may occur in severe cases.
  <br>  (c) Widal test is a confirmation test for typhoid.
  <br> (d) Symptoms are high fever (39-40°C), weakness, stomach pain, constipation, headache and loss of appetite.
  <br> <u>(ii) Pneumonia is caused byStreptococcus pneumoniae and Haemophilus influenzae.</u>
  <br> (a) These bacteria infect alveoli of the lungs. The alveoli get filled with fluid causing decrease in respiratory efficiency of the lungs
  <br> (b) Pneumonia spreads by inhaling droplets/aerosol from infected individuals or even by sharing glasses and utensils with patients.
  <br>  (c) Symptoms of pneumonia are fever, chills, cough, headache, etc.
  <br><u> (iii) Dysentery, plague, diphtheria, etc., are some other examples of bacterial diseases.</u>
  <br> <b> II. Viral Disease</b>
  <br> Common cold occurs due to a group of viruses called rhino viruses.
  <br> (a) These viruses infect the nose and respiratory passage but not the lungs.
  <br> (b) Common cold is characterised by nasal congestion and discharge, sore throat, hoarseness, cough, headache, tiredness, etc., which gene
  <br>(c) The infection occurs due to cough or sneezes of an infected person, either inhaled directly or transmitted through contaminated object
  <br> <b>III. Protozoan Diseases</b>
  <br> (i) Malaria is caused by a protozoan, Plasmodium sp. (P. vivax, P. malariae and P. falciparum).
  <br>(a) P. falciparum causes most serious kind of malaria, i.e. malignant malaria which can be fatal.
  <br>(b) Female Anopheles mosquito is the vector of Plasmodium, which transfer the sporozoites (infectious form) in human body.
   <br> (c) Life cycle of Plasmodium is given in the figure.
```

```
  <br>(c) The infection occurs due to cough or sneezes of an infected person, either inhaled directly or transmitted through conta
  <br> <b>III. Protozoan Diseases</b>
  <br> (i) Malaria is caused by a protozoan, Plasmodium sp. (P. vivax, P. malariae and P. falciparum).
  <br>(a) P. falciparum causes most serious kind of malaria, i.e. malignant malaria which can be fatal.
  <br>(b) Female Anopheles mosquito is the vector of Plasmodium, which transfer the sporozoites (infectious form) in human body.
   <br> (c) Life cycle of Plasmodium is given in the figure.
<br><img src="images/img23.jpg" height="500px" width="700px"/>
</li>
</ol>
</p>

        </h2>
</body>
</html>
```

PROGRAMMING COURSES:
C :

```
<html>
    <style>
        body {
        background-image:url('images/img11.jpg');
        background-repeat: no-repeat;
        background-size:cover;
        background-attachment: fixed;
      }
    </style>
    <body>

<h2>
    <h1><center>C Programming Language</center></h1>
    <h2>Audience</h2>
   <p style="font-size: large;"> This C tutorial series has been designed for those who want to learn C programming; whether you are b

    Required Knowledge<br>
    To learn C Programming language you haven't required any previous programming knowledge, but the basic understanding of any other
</p>
    <h2>C Example</h2>
   <p style=" font-size: large;"> A quick look at the example of Hello, World! In C programming, and detailed description is given in
    #include<stdio.h><br>
int main()<br>
{<br>
    printf("Hello, World!\n");
    getch(); //Use to get one character input from user, and it will not be printed on screen.<br>
    return 0;</p><br>
    <p>
    <h2>Some Facts About Programming language</h2></p>
    <p style="font-size: large;">
    In 1988, the American National Standards Institute (ANSI) had formalized the C language.<br>
  1) C was invented to write UNIX operating system.<br>
  2) C is a successor of 'Basic Combined Programming Language' (BCPL) called B language.<br>
  3) Linux OS, PHP, and MySQL are written in C.<br>
 4) C has been written in assembly language.<br></p>
 <p>
```

```
<h2>Uses of C programming Language</h2></p>
<p style="font-size: large;">
In the beginning, C was used for developing system applications, e.g. <br>
1) Database Systems<br>
2) Language Interpreters<br>
3) Compilers and Assemblers<br>
4) Operating Systems<br>
4) Network Drivers<br>
5) Word Processors<br></p>
<p>
   <h2> C Has Become Very Popular For Various Reasons</h2> </p>
   <p style="font-size: large;">
    1) One of the early programming languages.<br>
    2) Still, the best programming language to learn quickly.<br>
    3) C language is reliable, simple, and easy to use.<br>
    4) C language is a structured language.<br>
    5) Modern programming concepts are based on C.<br>
    6) It can be compiled on a variety of computer platforms.<br>
    7) Universities preferred to add C programming in their courseware.<br>
</p>
<p>
   <h2>Features Of C Programming Language</h2> </p>
   <p style="font-size: large;">
   1) C is a robust language with a rich set of built-in functions and operators.<br>
2) Programs written in C are efficient and fast.<br>
3) C is highly portable; programs once written in C can be run on other machines with minor or no modification.<br>
4) C is a collection of C library functions; we can also create our function and add it to the C library.<br>
5) C is easily extensible.<br>
</p>
<p>
   <h2>Advantages Of C</h2></p>
   <p style="font-size: large;">
   1) C is the building block for many other programming languages.<br>
2) Programs written in C are highly portable.<br>
3) Several standard functions are there (like in-built) that can be used to develop programs.<br>
4) C programs are collections of C library functions, and it's also easy to add functions to the C library.<br>
```

```
    <h2>Disadvantages Of C</h2></p>
    <p style="font-size: large;">
    1) C does not provide Object Oriented Programming (OOP) concepts.<br>
2) There are no concepts of Namespace in C.<br>
3) C does not provide binding or wrapping up of data in a single unit.<br>
4) C does not provide Constructor and Destructor.<br>

</p>
<p>
<h2>The limitations of C programming language</h2></p>
<p style="font-size: large;">
1) Difficult to debug.<br>
2) C allows a lot of freedom in writing code, and that is why you can put an empty line or white space anywhere in the program<br>and
3) C compilers can only identify errors and are incapable of handling exceptions (run-time errors).<br>
4) C provides no data protection.<br>
5) It also doesn't feature reusability of source code extensively.<br>
6) It does not provide strict data type checking (for example an integer value can be passed for floating datatype).<br>
</p>

<p>
    <h2>Operators</h2></p>
    <p style="font-size: large;">
<br> Main article: Operators in C and C++<br>
C supports a rich set of operators, which are symbols used within an expression to specify the manipulations to be performed while ev

arithmetic: +, -, *, /, %<br>
assignment: =<br>
augmented assignment: +=, -=, *=, /=, %=, &=, |=, ^=, <<=, >>=<br>
bitwise logic: ~, &, |, ^<br>
bitwise shifts: <<, >><br>
boolean logic: !, &&, ||<br>
conditional evaluation: ? :<br>
equality testing: ==, !=<br>
calling functions: ( )<br>
increment and decrement: ++, --<br>
member selection: ., -><br>
```

```
object size: sizeof<br>
order relations: <, <=, >, >=<br>
reference and dereference: &, *, [ ]<br>
sequencing: ,<br>
subexpression grouping: ( )<br>
type conversion: (typename)<br>
C uses the operator = (used in mathematics to express equality) to indicate assignment, following the precedent of Fortran and PL/I, but
The C operator precedence is not always intuitive. For example, the operator == binds more tightly than (is executed prior to) the opera
</p>
<p>

<h2>"Hello, world" example</h2></p><br>
<p style="font-size: large;">
<b>See also:</b> Hello, world<br>
The "hello, world" example, which appeared in the first edition of K&R, has become the model for an introductory program in most program

The original version was:<br><br>

<b>main()<br>
{<br>
    printf("hello, world\n");<br>
}</b><br><br>
A standard-conforming "hello, world" program is:[a]<br><br>

<b>#include <stdio.h><br>

int main(void)<br>
{<br>
    printf("hello, world\n");<br>
}</b><br><br>
The first line of the program contains a preprocessing directive, indicated by #include.<br><br>This causes the compiler to replace that

The next line indicates that a function named main is being defined. The main function serves a special purpose in C programs; the run-t

The opening curly brace indicates the beginning of the definition of the main function.<br>
```

JAVA :

```
<html>
    <style>
        body {
        background-image:url('images/img11.jpg');
        background-repeat: no-repeat;
        background-size:cover;
        background-attachment: fixed;
     }
    </style>
    <body>
     <h2>
        <h1><center>JAVA</center></h1>
    <h2>Audience</h2>
   <p style="font-size: large;">  This Java tutorial series has been designed for those who want to learn Java programming; wheth
<p>
    <h2>Required Knowledge</h2></p>
   <p style="font-size: large;">  Basic Knowledge of C Programming and C++ will help you to understand Java Programming quickly,
</p>
<p>
    <h2>Java Example</h2></p>
   <p style="font-size: large;">  A quick look at the example of Hello Java program and the detailed description is given in Java
    Example<br>
    public class Hello<br>
{   <br>
    public static void main(String[] args)<br>
    {<br>
        System.out.println("Hello Java");<br>
    }<br>
}<br>
Program Output<br>
Hello Java<br>
The above example has been used to print Hello Java text on the screen.<br>
</p>
<p>
    <h2>INTRODUCTION</h2>
    <h2>
        What is Java?</h2></p>
```

```
        What is Java?</h2></p>
      <p style="font-size: large;">  1) Java is an object-oriented programming language developed by Sun Microsystems, and it was r
        2) James Gosling initially developed Java in Sun Microsystems (which was later merged with Oracle Corporation).<br>
        3) Java is a set of features of C and C++. It has obtained its format from C, and OOP features from C++.<br>
        4) Java programs are platform independent which means they can be run on any operating system with any processor as long as
        5) Java code that runs on one platform does not need to be recompiled to run on another platform; it's called write once, r
        6) Java Virtual Machine (JVM) executes Java code, but it has been written in platform-specific languages such as C/C++/ASM,
</p>
<p>
    <h2>Where is Java Being Used?</h2></p>
   <p style="font-size: large;"> Earlier Java was only used to design and program small computing devices, but it was later adopted
    Java is one of the most important programming languages in today's IT industries.<br>
    1) JSP - In Java, JSP (Java Server Pages) is used to create dynamic web pages, such as in PHP and ASP.
2) Applets - Applets are another type of Java programs that are implemented on Internet browsers and are always run as part of a we
3) J2EE - Java 2 Enterprise Edition is a platform-independent environment that is a set of different protocols and APIs and is used
4) JavaBeans - This is a set of reusable software components that can be easily used to create new and advanced applications.
5) Mobile - In addition to the above technology, Java is widely used in mobile devices nowadays, many types of games and applicatio
</p>
<p>
    <h2>Types Of Java Applications</h2></p>
 <p style="font-size: large;">   1) Web Application - Java is used to create server-side web applications. Currently, Servlet, JSP,
2) Standalone Application - It is also known as the desktop application or window-based application. An application that we need to
3) Enterprise Application - An application that is distributed in nature, such as banking applications, etc. It has the advantage o
4) Mobile Application - Java is used to create application software for mobile devices. Currently, Java ME is used for building app
</p>
<p>
    <h2>Features Of Java</h2></p>
   <p style="font-size: large;"> 1) Object-Oriented - Java supports the features of object-oriented programming. Its object model i
2) Platform independent - C and C++ are platform dependency languages hence the application programs written in one Operating syste
3) Simple - Java has included many features of C / C ++, which makes it easy to understand.<br>
4) Secure - Java provides a wide range of protection from viruses and malicious programs.  It ensures that there will be no damage
5) Portable - Java provides us with the concept of portability. Running the same program with Java on different platforms is possib
6)Robust - During the development of the program, it helps us to find possible mistakes as soon as possible.<br>
7) Multi-threaded - The multithreading programming feature in Java allows you to write a program that performs several different ta
8) Distributed - Java is designed for distributed Internet environments as it manages the TCP/IP protocol.<br>
</p>
```

```
</p>
<p>
    <h2>Basic Structure Of Java Programs</h2></p>
  <p style="font-size: large;">  A Java program involves the following sections:<br>
    1) Documentation Section
2) Package Statement<br>
3) Import Statements<br>
4) Interface Statement<br>
5) Class Definition<br>
6) Main Method Class<br>
7) Main Method Definition<br>
</p>
        </h2>
        </body>
</html>
```

C++ :

```
<html>
    <style>
        body {
       background-image:url('images/img11.jpg');
       background-repeat: no-repeat;
       background-size:cover;
       background-attachment: fixed;
     }
   </style>
   <body>
    <h2>
        <h1><center>C++ Programming Language</center></h1>
<h2>Audience</h2>
<p style="font-size: large;">C++ is a general-purpose, middle-level programming language, with high and low-level programming capabilit
<p>
    <h2>Required Knowledge</h2></p>
   <p style="font-size: large;"> It is important to understand the concepts of C before learning C++, and the basic Knowledge of C Prog
</p>
<p>
    <h2>C++ Example</h2></p>
   <p style="font-size: large;"> A quick look at the example of  C++ Program and detailed description is given in the C++ Program Struc
    #include <iostream> <br>

        int main()<br>
        {<br>
         std::cout<<"This is my first C++ Program.";<br>
         std::cout<<"and its very easy to learn";<br>
        }<br>
</p>
<p>
    <h2>INTRODUCTION</h2></p>
   <p style="font-size: large;"> C++ is a multi-paradigm programming language that supports object-oriented programming (OOP), created
</p>
<p>
    <h2>Uses of C++</h2></p>
  <p style="font-size: large;">  C++ is used by programmers to create computer software. It is used to create general systems software,
    C++ is used by many programmers of different types and coming from different fields. C++ is mostly used to write device driver prog
```

```
    <h2>Object-Oriented Programming and C++</h2></p>
  <p style="font-size: large;">  C++ supports object-oriented programming (OOP), with four significant principles of object-oriented dev
    1) Abstraction<br>
2) Encapsulation<br>
3) Inheritance<br>
4) Polymorphism<br>
</p>
<p>
    <h2>Features of Object Oriented C++</h2></p>
  <p style="font-size: large;"> 1) The main focus remains on data rather than procedures.<br>
2) Object-oriented programs are segmented into parts called objects.<br>
3) Data structures are designed to categorize the objects.<br>
4) Data member and functions are tied together as a data structure.<br>
5) Data can be hidden and cannot be accessed by external functions using access specifier.<br>
6) Objects can communicate among themselves using functions.<br>
7) New data and functions can be easily added anywhere within a program whenever required.<br>
8) Since this is an object-oriented programming language, it follows a bottom up approach, i.e. the execution of codes starts from the m
        The object-oriented approach is a recent concept among programming paradigms and has various fields of progress. Object-orie
</p>
<p>
    <h2>Standard Libraries in c++</h2></p>
    <p style="font-size: large;">
    C++ standard library was created after many years and it has three important parts:<br>
1) C++ core language provides all the building blocks including data types, variables, and literals etc.<br>
2) The C++ Standard Library has a rich set of methods for manipulating files and strings.<br>
3) The STL(Standard Template Library) provides a rich set of template classes for manipulating data structures.<br>
</p>
<p>
    <h2>ANSI Standard for C++</h2></p>
    <p style="font-size: large;">
    ANSI stands for American National Standard Institute & the ANSI standard began an attempt to ensure that C++ codes become portable -
</p>
    </h2>
</body>
</html>
```

PYTHON:

```
<html>
    <style>
        body {
      background-image:url('images/img11.jpg');
      background-repeat: no-repeat;
      background-size:cover;
      background-attachment: fixed;
     }
   </style>
   <body>
    <h2>
        <center><h1>Phython</h1></center>
<h2>Audiene</h2>
<p style="font-size: large;"> This Python tutorial series has been designed for those who want to learn Python programming; whether
<p>
    <h2>Required Knowledge</h2></p>
    <p style="font-size: large;">
    To learn the Python programming language, you do not need any previous programming knowledge. Nevertheless, a basic understandi
</p>
<p>
    <h2>Python Example</h2></p>
    <p style="font-size: large;">
    A quick look at the example of Hello, World! In Python programming, and a detailed description is given in the Python Basics pa
    Example:<br>
    print "Hello, World!"<br><br>
    Python 3 requires adding brackets around the value to be printed:<br>
    print ("Hello, World!")<br><br>
    Output:<br>
    Hello, World!<br>
    The above example has been used to print Hello, World! Text on the screen.
</p>
<p>
    <h2>Python Overview</h2></p>
    <p style="font-size: large;">
    Python is a general-purpose, object-oriented programming language with high-level programming capabilities It has become famous
    Python is a programming language that includes features of C and Java. It provides the style of writing an elegant code like C,
```

```
    <h2>Some Facts About Python</h2></p>
    <p style="font-size: large;">
    1) Python was developed in the late eighties, i.e., the late 1980's by Guido van Rossum at the National Research Inst
2) Python is derived from programming languages such as ABC, Modula 3, small talk, Algol-68.<br>
3) Van Rossum picked the name Python for the new language from a TV show, Monty Python's Flying Circus.<br>
4) Python page is a file with a .py extension that contains could be the combination of HTML Tags and Python scripts.<br>
5) In December 1989, the creator developed the 1st python interpreter as a hobby, and then on 16 October 2000, Python 2.0
6) On 3rd December 2008, Python 3.0 was released with more testing and included new features.<br>
7) Python is an open-source scripting language.<br>
8) Python is open-source, which means that anyone can download it freely from www.python.org and use it to develop progra
9) Python is one of the official languages at Google.<br>
</p>
<p>
    <h2>Python Why? Characteristics and Features Of Python</h2></p>
    <p style="font-size: large;">
    Python is gaining good popularity in the programming community; there are many reasons behind this.<br>

1) Interpreted Language:Python is processed at runtime by Python Interpreter.<br>
2) Object-Oriented Language: It supports object-oriented features and techniques of programming.<br>
3) Interactive Programming Language: Users can interact with the python interpreter directly for writing programs.<br>
4) Easy language: Python is easy to learn, especially for beginners.<br>
5) Straightforward Syntax: The formation of python syntax is simple and straightforward, which also makes it popular.<br>
6) Easy to read: Python source-code is clearly defined and visible to the eyes.<br>
7) Portable: Python codes can be run on a wide variety of hardware platforms having the same interface.<br>
8) Extendable: Users can add low level-modules to Python interpreter.<br>
9) Scalable: Python provides an improved structure for supporting large programs then shell-scripts.<br>
</p>
<p>
    <h2>What Can You Do with Python</h2></p>
    <p style="font-size: large;">
    Python is used to create web and desktop applications, and some of the most popular web applications like Instagram,
</p>
    </h2>
    </body>
</html>
```

SQL DATABASE:

```
<html>
    <style>
        body {
        background-image:url('images/img11.jpg');
        background-repeat: no-repeat;
        background-size:cover;
        background-attachment: fixed;
       }
    </style>
    <body>
     <h2>
        <h1><center>SQL</center></h1>
      <p> SQL (Structured Query Language) is a standard database programming language used for accessing and manipulating dat
    </p>
<p>
    <h2>What Wil You Get Learning SQL?</h2></p>
    <p style="font-size: large;">
    SQL gives unique learning and database handling techniques on Structured Query language and will help you make better com
</p>
<p>
    <h2>Required Knowledge</h2></p>
    <p style="font-size: large;">
    The basic understanding of computer programming and RDBMS Concepts will help you to understand the SQL quickly.</p>
    <h2>SQL Example</h2>
  <p style="font-size: large;">  The SQL query to select all records from the users' table:<br>
    SELECT * FROM users;<br>
    The SQL query to delete single records from users table by using where clause:<br>
DELETE FROM users WHERE user_id=299;<br>
</p>
<h2>SQL Commands</h2>
<p style="font-size: large;">
    These are the some important SQL command:<br><br>

    SELECT: it extracts data from a database.<br>
    UPDATE: it updates data in database.<br>
    DELETE: it deletes data from database.<br>
    CREATE TABLE: it creates a new table.<br>
```

```
    CREATE TABLE: it creates a new table.<br>
    ALTER TABLE: it is used to modify the table.<br>
    DROP TABLE: it deletes a table.<br>
    CREATE DATABASE: it creates a new database.<br>
    ALTER DATABASE: It is used to modify a database.<br>
    INSERT INTO: it inserts new data into a database.<br>
    CREATE INDEX: it is used to create an index (search key).<br>
    DROP INDEX: it deletes an index.<br>

</p>
<h2>SQL CREATE TABLE</h2>
<p style="font-size: large;">
    The SQL CREATE TABLE statement is used to create a new table.
<br><br><b>Syntax</b>
<br><i>CREATE TABLE table_name(
<br>column1 datatype,
<br>column2 datatype,
<br>column3 datatype,
<br>.....
<br>columnN datatype,
<br>PRIMARY KEY( one or more columns )
 <br>);</i>
 <br><br>CREATE TABLE is the keyword telling the database system what you want to do. In this case, you want to create a ne

<br> Then in brackets comes the list defining each column in the table and what sort of data type it is. The syntax becomes

 <br>A copy of an existing table can be created using a combination of the CREATE TABLE statement and the SELECT statement.
<br><br><b>Example</b>
<br><br>The following code block is an example, which creates a CUSTOMERS table with an ID as a primary key and NOT NULL ar
</p>
<i>SQL> CREATE TABLE CUSTOMERS(
<br>ID   INT              NOT NULL,
<br> NAME VARCHAR (20)     NOT NULL,
<br>AGE  INT              NOT NULL,
<br>ADDRESS  CHAR (25) ,
<br>SALARY   DECIMAL (18, 2),
```

```
 <i>SQL> CREATE TABLE CUSTOMERS(
 <br>ID   INT              NOT NULL,
 <br> NAME VARCHAR (20)     NOT NULL,
 <br>AGE  INT              NOT NULL,
 <br>ADDRESS  CHAR (25) ,
 <br>SALARY   DECIMAL (18, 2),
 <br> PRIMARY KEY (ID)
  <br>);</i>
  <p style="font-size: large;">
     You can verify if your table has been created successfully by looking at the message displayed by the SQL server, o
     SQL> DESC CUSTOMERS;
    <br> +---------+---------------+------+-----+---------+-------+
    <br> | Field   | Type          | Null | Key | Default | Extra |
    <br> +---------+---------------+------+-----+---------+-------+
     <br>| ID      | int(11)       | NO   | PRI |         |       |
    <br> | NAME    | varchar(20)   | NO   |     |         |       |
    <br> | AGE     | int(11)       | NO   |     |         |       |
    <br> | ADDRESS | char(25)      | YES  |     | NULL    |       |
     <br>| SALARY  | decimal(18,2) | YES  |     | NULL    |       |
    <br> +---------+---------------+------+-----+---------+-------+
    <br> 5 rows in set (0.00 sec)
   <p style="font-size: large;">Now, you have CUSTOMERS table available in your database which you can use to store the
 </p>
     </h2>
     </body>
 </html>
```

LANGUAGES:

JAPANESE:

```
<html>
    <style>
        body {
       background-image:url('images/img11.jpg');
       background-repeat: no-repeat;
       background-size:cover;
       background-attachment: fixed;
     }
   </style>
   <body>
    <h2>
        <center><h1>Japanese</h1></center>
        <h1>  Lesson 1 : Greetings </h1>
<p>
  <h2>When you meet or leave someone</h2><br></p>
<p style="font-size: larger;">Hello.    Konnichiwa. <br>
Pronounce "n" and "ni" separately. It's like "kon-nichiwa".<br><br>
Good morning.   Ohayo gozaimasu.    <br>
The last vowel "u" is not clearly pronounced. It's like "gozaimas".<br>
Informal Style: Ohayō.<br><br>
Good evening.   Konbanwa.   <br>
Used at the beginning of the conversation, not at the end.<br><br>
Good night. Oyasuminasai.   <br>
Informal Style: Oyasumi.<br><br>
Goodbye.    Sayōnara.   <br>
In general, used when people will not see each other for some time.<br>
Informal Style: Sayonara. (short "o" after y).<br><br>
See you.    Dewa mata.  <br><br>
See you tomorrow.   Dewa mata ashita.<br><br>
See you next week.  Dewa mata raishū.<br><br>
dewa: well, now or so<br><br>
mata: again<br>
</p>
<p>
    <h2>When You Thank Or Apologize Someone</h2></p>
    <p style="font-size: larger;"> Thank you.   Arigatō gozaimasu.  <br><br>
    Thank you very much.    Dōmo arigatō gozaimasu.<br><br>
```

```
    Thank you. (past)   Arigatō gozaimashita.<br><br>
    Thanks. Dōmo.<br>
    The last vowel "u" is not pronounced ly. gozaimas(u).<br>
    Dōmo arigatō gozaimasu.: To emphasize.<br><br>
    Arigatō gozaimashita.: To thank for something in the past.<br><br>
    Dōmo.: To thank for a small favor.<br>
    Informal Style: Arigatō.<br><br>
    You're welcome. Dōitashimashite.    <br>
    Don't mention it.   Iie.<br>
    Iie. (literally means "No.") also can be used with the rising accent if it's not a big deal.<br><br>
    Excuse me.  Sumimasen.  <br>
    Yes.    Hai.<br>
    Used when you talk to someone, and to apologize for small faults (Sorry.).<br><br>
    Hai. (Yes.) can be used to answer someone's call.<br><br>
    I'm sorry.  Gomennasai. <br><br>
    I'm sorry.  Dōmo sumimasen.<br><br>
    Don't be sorry. Iie.<br>
    Dōmo sumimasen. is also used. It sound<br>s more polite.<br><br>

    As a response, Iie. can be used if it's not a big deal.<br>
</p>

<p>
 <h1>Lesson 2 : Useful Expressions</h1>
<h2>How are you?</h2></p>
<p style="font-size: larger;">Are you well? Ogenki desuka?  <br><br>
Yes, I'm well.  Hai, genki desu.<br><br>
No, I'm not well.   Iie, genki dewa (ja) arimasen..<br><br>
Yes, I'm very well. Hai, totemo genki desu.<br><br>
So so.  Māmā desu.<br><br>
genki: well<br><br>

"o" in front of genki is a prefix which makes the word more polite.<br><br>

Genki desu.: I'm well.<br><br>

Genki dewa (ja) arimasen.: I'm not well.<br><br>
```

```
"ja arimasen" sounds more casual, and is used more in daily conversation.<br><br>

Informal: Genki?
</p>
<p>
    <h2>Understand?</h2></p>
    <p style="font-size: larger;">    Do you understand?    Wakari masuka?  <br><br>
Yes, I understand.  Hai, wakari masu.<br><br>
No, I don't understand. Iie, wakari masen.<br><br>
I understand a little.  Sukoshi wakari masu.<br><br>
Do you understand Japanese? Nihongo ga wakari masuka?<br><br>
Do you understand English?  Eigo ga wakari masuka?<br><br>
wakaru: to understand, to know<br><br>

Wakari masu.: I understand.<br><br>

Wakari masen.: I don't understand.<br><br>

Wakari mashita.: I understood. (I got it. All right.)<br><br>

[Nihongo] ga wakari masu.: I understand [Japanese].<br><br>

ga: Particle which comes after the object. Used with the verb "wakaru".<br><br>
</p>
<p>
    <h2>Please. (to request)</h2></p>
    <p style="font-size: larger;">   Please.    Onegaishimasu.  <br><br>
One more time, please.  Mōichido onegaishimasu.<br><br>
Slowly please.  Yukkuri onegaishimasu.<br><br>
Menu please.    Menyū o onegaishimasu.<br><br>
Used to ask some favor.<br><br>

[Menyū] o onegaishimasu.: [Menu] please.<br><br>

o: Particle which comes after the object.<br><br>
</p>
```

```
<h1>Lesson 3: Introducing Yourself</h1>
<p>
    <h2>I'm from ...</h2></p>
    <p style="font-size: larger;"> I'm from Brazil. Burajiru kara kimashita.    <br><br>
South Korea Kankoku kara kimashita.<br><br>
China   Chūgoku kara kimashita.<br><br>
Australia   Ōsutoraria kara kimashita.<br><br>
U. S. A.    Amerika kara kimashita.<br><br>
kara: from<br><br>

kimashita: came
</p>
<p>
    <h2>Nationality</h2></p>
    <p style="font-size: larger;">   I am Japanese.<br>
Watashi wa Nihon-jin desu.<br><br>
Is Ms. Wang Australian? Wan-san wa Ōsutoraria-jin desuka?<br><br>
He is not American  Kare wa Amerika-jin dewa arimasen.<br><br>
What nationality is she?    Kanojo wa Nani-jin desuka?<br><br>
watashi: I / anata: you / kare: he / kanojo: she<br><br>

Country name + jin = nationality<br><br>

Nihon-jin: Japanese<br><br>

Nani-jin desuka?: What nationality?<br><br>
</p>
    </h2>
    </body>
</html>
```

GERMAN:

```
<htm>
    <style>
        body {
       background-image:url('images/img11.jpg');
       background-repeat: no-repeat;
       background-size:cover;
       background-attachment: fixed;
     }
    </style>
   <body>
    <h2>
        <center><h1>German</h1></center>
<h1>Basic German Phrases</h1>
<p style="font-size: larger;">
    Guten Morgen<br>
goot-en mor-gen<br>
Good Morning    <br><br>Guten Tag<br>
goot-en tahk<br>
Hello/Good Day<br><br>  Guten Abend<br>
goot-en ah-bent<br>
Good Evening<br><br>
Gute Nacht<br>
goot-eh nakht<br>
Good Night<br><br>  Tag / Hallo / Servus<br>
tahk / hah-loh / sair-voohs<br>
Hi / Hello / Hi & Bye (Southern Germany & Austria)<br><br>  Auf Wiedersehen<br>
owf vee-dair-zayn<br>
Goodbye<br><br>
Grüß dich / Grüß Gott!<br>
Hello! / Greetings! (Southern Germany & Austria)<br><br>    Tschüs / Tschau<br>
tchews / chow<br>
Bye!<br><br>    Gehen wir!<br>
geh-en veer<br>
Let's go!<br><br>
Bis später<br>
biss shpay-ter<br>
See you later   <br><br>Bis bald<br>
```

```
Let's go!<br><br>
Bis später<br>
biss shpay-ter<br>
See you later   <br><br>Bis bald<br>
biss bahlt<br>
See you soon<br><br>    Bis morgen<br>
biss mohr-gen<br>
See you tomorrow<br><br>
Bitte<br>
bih-tuh<br>
Please  <br><br>Danke (schön / sehr)<br>
dahn-kuh shurn/zair<br>
Thank you   <br><br>Bitte schön<br>
bih-tuh shurn<br>
You're welcome<br><br>
Es tut mir leid.<br>
ehs toot meer lite<br>
I'm sorry<br><br>   Entschuldigen Sie<br>
ehnt-shool-dih-gun zee<br>
Excuse me<br><br>   Verzeihung<br>
Pardon me<br><br>
Wie geht es Ihnen?<br>
vee gayt es ee-nen<br>
How are you? (formal)   <br><br>Wie geht's?<br>
vee gayts<br>
How are you? (informal)<br><br> (Sehr) Gut / So lala<br>
zair goot / zo lahlah<br>
(Very) Good / OK<br><br>
Schlecht / Nicht Gut<br>
shlekht / nisht goot<br>
Bad / Not good<br><br>  Es geht.<br>
ess gate<br>
I'm ok. (informal)<br><br>  Ja / Nein<br>
yah / nine<br>
Yes / No<br><br>
Wie heißen Sie?<br>
vee hie-ssen zee<br>
```

```
What's your name? (formal)<br><br>  Wie heißt du?<br>
vee hiesst doo<br>
What's your name? (informal)<br><br>    Ich heiße...<br>
ikh hie-ssuh<br>
My name is... [I am called...]<br><br>
Es freut mich.<br>
froyt mikh<br>
Pleased to meet you.<br><br>    Gleichfalls.<br>
glykh-fals<br>
Likewise.   <br><br>Herr / Frau / Fräulein<br>
hair / frow / froi-line<br>
Mister / Misses / Miss<br><br>
Woher kommen Sie?<br>
vo-hair koh-men zee<br>
Where are you from? (formal)    <br><br>Woher kommst du?<br>
vo-hair kohmst doo<br>
Where are you from? (informal)<br><br>  Ich komme aus...<br>
ikh koh-muh ows...<br>
I'm from...<br><br>
Wo wohnen Sie?<br>
vo voh-nen zee<br>
Where do you live? (formal)<br><br> Wo wohnst du?<br>
vo vohnst doo<br>
Where do you live? (informal)   <br><br>Ich wohne in...<br>
ikh voh-nuh in<br>
I live in...<br><br>
Wie alt sind Sie?<br>
vee alt zint zee<br>
How old are you? (formal)   <br><br>Wie alt bist du?<br>
vee alt bisst doo<br>
How old are you? (informal)<br><br> Ich bin ____ Jahre alt.<br>
ikh bin ____ yaa-reh alt<br>
I am ____ years old.<br><br>
Sprechen Sie deutsch?<br>
shpreck-en zee doytch<br>
Do you speak German? (formal)<br><br>   Sprichst du englisch?<br>
shprikhst doo eng-lish<br>
```

```
Do you speak English? (informal)<br><br>    Ich spreche (kein)...<br>
ikh shpreck-uh kine<br>
I (don't) speak...<br><br>
Verstehen Sie? / Verstehst du?<br>
fehr-shtay-en zee / fehr-shtayst doo<br>
Do you understand? (formal / informal)<br><br>  Ich verstehe (nicht).<br>
ikh fehr-shtay-eh nikht<br>
I (don't) understand.   <br><br>Ich weiß (nicht).<br>
ikh vise nikht<br>
I (don't) know.<br><br>
Können Sie mir helfen?<br>
ker-nen zee meer hell-fen<br>
Can you help me? (formal)   <br><br>Kannst du mir helfen?<br>
kahnst doo meer hell-fen<br>
Can you help me? (informal) <br><br>Natürlich / Gerne<br>
nah-tewr-likh / gair-nuh<br>
Of course / Gladly<br><br>
Kann ich Ihnen helfen?<br>
kahn ikh ee-nen hell-fen<br>
May I help you? (formal)<br><br>    Kann ich dir helfen?<br>
kahn ikh deer hell-fen<br>
May I help you? (informal)<br><br>  Wie bitte?<br>
vee bih-tuh<br>
What? Pardon me?<br><br>
Wie heißt ___ auf Deutsch?<br>
vee heist ___ owf doytch<br>
How do you say ___ in German?<br><br>   Wo ist / Wo sind... ?<br>
voh ist / voh zint<br>
Where is / Where are... ?   <br><br>Es gibt...<br>
ess geept<br>
There is / are...<br><br>
Was ist los?<br>
vahs ist lohs<br>
What's the matter?<br><br>  Das macht nichts.<br>
dass makht nikhts<br>
It doesn't matter.<br><br>  Das ist mir egal.<br>
dass ist meer eh-gahl<br>
```

```
It doesn't matter.<br><br>  Das ist mir egal.<br>
dass ist meer eh-gahl<br>
I don't care.<br><br>
Keine Angst!<br>
ky-nuh ahngst<br>
Don't worry!<br><br>    Ich habe es vergessen.<br>
ikh hah-buh ess fehr-geh-sen<br>
I forgot.   <br><br>Jetzt muss ich gehen.<br>
yetz mooss ikh geh-en<br>
I must go now.<br><br>
Ich habe Hunger / Durst.<br>
ikh hah-buh hoong-er / dirst<br>
I'm hungry / thirsty.<br><br>   Ich bin krank / müde.<br>
ikh bin krahnk moo-duh<br>
I'm sick / tired.   <br><br>Ich habe Langeweile.<br>
ikh hah-buh lahn-guh-vy-luh<br>
I'm bored.<br><br>
Ich möchte / Ich hätte gern...<br>
ikh merkh-tuh / ikh heh-tuh gairn<br>
I'd like... <br><br>Das gefällt mir.<br>
dahs geh-fehlt meer<br>
I like it.<br><br>  Prima / Toll / Super!<br>
pree-mah / tohl / zoo-pair<br>
Great / Fantastic!<br><br>
Gesundheit!<br>
geh-soont-hyt<br>
Bless you!  <br><br>Herzlichen Glückwunsch!<br>
herts-likh-en glewk-voonsh<br>
Congratulations!<br><br>    Sei ruhig!<br>
zy roo-hikh<br>
Be quiet! (informal)<br><br>
Willkommen!<br>
vil-koh-men<br>
Welcome!<br><br>    Viel Glück!<br>
feel glewk<br>
Good luck!<br><br>  Schauen Sie mal! / Schau mal!<br>
```

```
Congratulations!<br><br>    Sei ruhig!<br>
zy roo-hikh<br>
Be quiet! (informal)<br><br>
Willkommen!<br>
vil-koh-men<br>
Welcome!<br><br>    Viel Glück!<br>
feel glewk<br>
Good luck!<br><br>  Schauen Sie mal! / Schau mal!<br>
show-en zee mal / show mal<br>
Look! (formal / informal)<br><br>
Bitte schön?<br>
Yes? / What would you like to order?<br><br>    Was darf's sein?<br>
What can I get you? / How can I help you?   <br><br>Sonst noch etwas?<br>
Anything else?<br><br>
Bitte schön.<br>
Here you go. (handing something to someone) <br><br>Zahlen bitte!<br>
The check, please!<br><br>  Stimmt so.<br>
Keep the change.<br><br>
Ich bin satt.<br>
I'm full.   <br><br>Mir ist schlecht.<br>
I feel sick.<br><br>    Es tut mir weh.<br>
It hurts.<br><br>
Ich liebe dich.<br>
ikh leeb-uh dikh<br>
I love you. (informal)<br><br>  Du fehlst mir.<br>
I miss you. (informal)<br><br>  Alles ist in Ordnung.<br>
Everything is fine.<br><br>
Wie wäre es mit ... ?<br>
How about...?   <br><br>Was für ein...?<br>
What kind of (a)...?<br><br>    Nicht wahr?<br>
[general tag question]
</p>
    </h2>
</htm>
```

SPANISH:

```
<html>
    <style>
        body {
       background-image:url('images/img11.jpg');
       background-repeat: no-repeat;
       background-size:cover;
       background-attachment: fixed;
     }
    </style>
   <body>
    <h2>
        <center><h1>Spanish</h1></center>
<h1>Basic Spanish Phrases</h1>
<p style="font-size: larger;">
    ¡Buenos días! <br>
bway-nohs dee-ahs <br>
Hello! / Good morning!<br><br>  ¡Buenas tardes! <br>
bway-nahs tard-ays <br>
Good afternoon! <br><br>¡Buenas noches! <br>
bway-nahs noh-chays <br>
Good evening! / Good night!<br><br>
¡Hola! / ¡Chao!<br>
oh-lah / chow<br>
Hi! / Bye!<br><br>  Adiós. <br>
ah-dee-ohs <br>
Good bye.<br><br>   Por favor. <br>
por fah-bor <br>
Please.<br><br>
Hasta la vista / Hasta luego. <br>
ah-stah lah vees-tah / ah-stah loo-ay-go <br>
See you / See you later.<br><br>    Hasta pronto. <br>
ah-stah prohn-toh <br>
See you soon.   <br><br>Hasta mañana. <br>
ah-stah mahn-yahn-ah <br>
See you tomorrow.<br><br>
(Muchas) Gracias. <br>
(moo-chahs) grah-see-ahs <br>
```

```
Thank you (very much).<br><br>  De nada.<br>
day nah-dah<br>
You're welcome. <br><br>Bienvenidos<br>
byen-veh-nee-dohs<br>
Welcome<br><br>
Lo siento<br>
loh see-ehn-toh<br>
I'm sorry<br><br>   Con permiso / Perdón / Disculpe<br>
kohn pehr-mee-soh / pehr-dohn / dees-kool-peh<br>
Excuse me / Pardon me<br><br>   ¡Vamos!<br>
bah-mohs<br>
Let's go!<br><br>
¿Cómo está usted? <br>
koh-moh ay-stah oo-sted <br>
How are you? (formal)<br><br>   ¿Cómo estás? <br>
koh-moh ay-stahs <br>
How are you? (informal)<br><br> ¿Qué tal? <br>
kay tahl <br>
How's it going?<br><br>
Bien / Muy bien <br>
bee-ehn / moy bee-ehn <br>
Good / Very good<br><br>    Mal / Muy mal / Más o menos <br>
mahl / moy mahl / mahs oh may-nohs <br>
Bad / Very bad / OK <br><br>Sí / No<br>
see / noh<br>
Yes / No<br><br>
¿Cómo se llama usted? <br>
koh-moh say yah-mah oo-sted <br>
What is your name? (formal) <br><br>¿Cómo te llamas? <br>
koh-moh tay yah-mahs <br>
What is your name? (informal)<br><br>   Me llamo...  / Mi nombre es...<br>
may yah-moh  / mee nohm-breh ess<br>
My name is...<br><br>
Mucho gusto. / Encantado.<br>
moo-choh goo-stoh / en-cahn-tah-doh<br>
Nice to meet you.<br><br>   Igualmente.<br>
ee-guahl-mehn-tay<br>
```

Same here. / Same to you.

  Señor / Señora / Señorita
sayn-yor / sayn-yor-ah / sayn-yor-ee-tah
Mister / Mrs. / Miss

¿De dónde es usted?
day dohn-day ehs oo-sted
Where are you from? (formal)

  ¿De dónde eres?
day dohn-day eh-rehs
Where are you from? (informal)

  Yo soy de...
yoh soy day
I'm from...

¿Cuántos años tiene usted?
quahn-tohs ahn-yohs tee-ay-nay oo-sted
How old are you? (formal)

  ¿Cuántos años tienes?
quahn-tohs ahn-yohs tee-ayn-ays
How old are you? (informal)

 Yo tengo _____ años.
yoh tayn-goh _____ ahn-yohs
I am _____ years old.

¿Habla usted español?
ah-blah oo-sted eh-spahn-yol
Do you speak Spanish? (formal)

  ¿Hablas inglés?
ah-blahs een-glehs
Do you speak English? (informal)

  (No) Hablo...
noh ah-bloh
I (don't) speak...

¿Entiende usted? / ¿Entiendes?
ehn-tyen-deh oo-sted / ehn-tyen-dehs
Do you understand? (formal / informal)

  (No) Entiendo.
noh ehn-tyen-doh
I (don't) understand.

  Yo (no lo) sé.
yoh noh loh seh
I (don't) know.

¿Puede ayudarme?
pweh-deh ah-yoo-dar-meh
Can you help me? (formal)

  Claro / Claro que sí
klah-roh / klah-roh keh see
Sure / Of course

  ¿Cómo?
koh-moh

```
What? Pardon me?<br><br>
¿Dónde está / Dónde están... ?<br>
dohn-deh eh-stah / dohn-deh eh-stahn<br>
Where is ... / Where are ... ?  <br><br>Aquí / Ahí<br>
ah-kee / ah-ee<br>
Here / There    Hay / Había...<br>
eye / ah-bee-ah<br>
There is / are... / There was / were...<br><br>
¿Cómo se dice ____ en español?<br>
koh-moh seh dee-seh ___ en eh-spahn-yol<br>
How do you say ____ in Spanish? <br><br>¿Qué es esto?<br>
keh ehs ehs-toh<br>
What is that?   <br><br>¿Qué te pasa?<br>
keh teh pah-sah<br>
What's the matter (with you)?<br><br>
No importa.<br>
noh eem-por-tah<br>
It doesn't matter.<br><br>  ¿Qué pasa?<br>
keh pah-sah<br>
What's happening?<br><br>   Sin novedad.<br>
seen noh-veh-dahd<br>
Nothing much.<br><br>
No tengo ninguna idea.<br>
noh tehn-goh neen-goo-nah ee-deh-ah<br>
I have no idea.<br><br> ¡Buena idea!<br>
bweh-nah ee-deh-ah<br>
Good idea!<br><br>  ¡Pase!<br>
pah-seh<br>
Go ahead!<br><br>
Estoy cansado / enfermo.<br>
eh-stoy kahn-sah-doh / ehn-fehr-moh<br>
I'm tired / sick.<br><br>   Tengo hambre / sed.<br>
tehn-goh ahm-breh / sed<br>
I'm hungry / thirsty.<br><br>   Tengo calor / frío.<br>
tehn-goh kah-lohr / free-oh<br>
I'm hot / cold.<br><br>
Estoy aburrido.<br>
```

```
ehs-tah bee-ehn<br>
That's alright. / It's ok.<br><br>  Me olvidé.<br>
meh ohl-vee-deh<br>
I forgot.<br><br>   Tengo que ir ahora.<br>
tehn-goh keh eer ah-oh-rah<br>
I must go now.<br><br>
¿Listo?<br>
lees-toh<br>
Ready?<br><br>  Quizás / Depende.<br>
kee-sahs / deh-pehn-deh<br>
Maybe / It depends.<br><br> Todavía no.<br>
toh-dah-vee-ah noh<br>
Not yet.<br><br>
¡Qué chistoso!<br>
keh chees-toh-soh<br>
How funny!<br><br>  ¡Que le vaya bien!<br>
keh leh vah-yah bee-ehn<br>
Have a nice day!<br><br>    ¡Nos vemos!<br>
nohs veh-mos<br>
We'll see you!<br><br>
¡Salud!<br>
sah-lood<br>
Bless you!<br><br>  ¡Felicitaciones!<br>
feh-lee-see-tah-see-oh-nehs<br>
Congratulations!<br><br>    ¡Buena suerte!<br>
bweh-nah swehr-teh<br>
Good luck!<br><br>
Te toca a ti.<br>
teh toh-kah ah tee<br>
It's your turn. (informal)<br><br>  ¡Callate!<br>
Shut up!<br><br>    Te amo. <br>
tay ah-moh <br>
I love you. (informal and singular)
</p>

    </h2>
</html>
```

CSS CODE:
STYLES.CSS CODE:

```css
body {
  background: #fff;
  color: #666666;
  font-family: "Open Sans", sans-serif;
}

a {
  color: #2dc997;
}

a:hover, a:active, a:focus {
  color: #2dca98;
  outline: none;
  text-decoration: none;
}

p {
  padding: 0;
  margin: 0 0 30px 0;
}

h1, h2, h3, h4, h5, h6 {
  font-family: "Poppins", sans-serif;
  font-weight: 400;
  margin: 0 0 20px 0;
  padding: 0;
}

/* Prelaoder */
#preloader {
  position: fixed;
  left: 0;
  top: 0;
  z-index: 999;
  width: 100%;
  height: 100%;
```

```
  overflow: visible;
  background: #fff url("../img/preloader.svg") no-repeat center center;
}

/* Back to top button */
.back-to-top {
  position: fixed;
  display: none;
  background: rgba(0, 0, 0, 0.2);
  color: #fff;
  padding: 6px 12px 9px 12px;
  font-size: 16px;
  border-radius: 2px;
  right: 15px;
  bottom: 15px;
  transition: background 0.5s;
}

@media (max-width: 768px) {
  .back-to-top {
    bottom: 15px;
  }
}

.back-to-top:focus {
  background: rgba(0, 0, 0, 0.2);
  color: #fff;
  outline: none;
}

.back-to-top:hover {
  background: #2dc997;
  color: #fff;
}
```

```css
/*--------------------------------------------------------------
# Disable AOS delay on mobile
--------------------------------------------------------------*/
@media screen and (max-width: 768px) {
  [data-aos-delay] {
    transition-delay: 0 !important;
  }
}

/*--------------------------------------------------------------
# Header
--------------------------------------------------------------*/
#header {
  padding: 30px 0;
  height: 92px;
  position: fixed;
  left: 0;
  top: 0;
  right: 0;
  transition: all 0.5s;
  z-index: 997;
  background: rgba(52, 59, 64, 0.9);
}

#header #logo {
  float: left;
}

#header #logo h1 {
  font-size: 36px;
  margin: 0;
  padding: 6px 0;
  line-height: 1;
  font-family: "Poppins", sans-serif;
  font-weight: 700;
  letter-spacing: 3px;
  text-transform: uppercase;
```

```css
}

#header #logo h1 a, #header #logo h1 a:hover {
  color: #fff;
}

#header #logo img {
  padding: 0;
  margin: 0;
}

@media (max-width: 768px) {
  #header #logo h1 {
    font-size: 26px;
  }
  #header #logo img {
    max-height: 40px;
  }
}

#header.header-transparent {
  background: transparent;
}

#header.header-fixed {
  background: rgba(52, 59, 64, 0.9);
  padding: 20px 0;
  height: 72px;
  transition: all 0.5s;
}

/*--------------------------------------------------------------
# Hero Section
--------------------------------------------------------------*/
#hero {
  width: 100%;
  height: 100vh;
```

```
  height: 100vh;
  background: url(../img/img3.jpg) top center;
  background-size: cover;
  position: relative;
}

@media (min-width: 1024px) {
  #hero {
    background-attachment: fixed;
  }
}

#hero:before {
  content: "";
  background: rgba(0, 0, 0, 0.6);
  position: absolute;
  bottom: 0;
  top: 0;
  left: 0;
  right: 0;
}

#hero .hero-container {
  position: absolute;
  bottom: 0;
  top: 0;
  left: 0;
  right: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  flex-direction: column;
  text-align: center;
}

#hero h1 {
  margin: 30px 0 10px 0;
```

```
  font-size: 48px;
  font-weight: 700;
  line-height: 56px;
  text-transform: uppercase;
  color: #fff;
}

@media (max-width: 768px) {
  #hero h1 {
    font-size: 28px;
    line-height: 36px;
  }
}

#hero h2 {
  color: #eee;
  margin-bottom: 50px;
  font-size: 24px;
}

@media (max-width: 768px) {
  #hero h2 {
    font-size: 18px;
    line-height: 24px;
    margin-bottom: 30px;
  }
}

#hero .btn-get-started {
  font-family: "Poppins", sans-serif;
  text-transform: uppercase;
  font-weight: 500;
  font-size: 16px;
  letter-spacing: 1px;
  display: inline-block;
  padding: 8px 28px;
  border-radius: 50px;
```

```
  transition: 0.5s;
  margin: 10px;
  border: 2px solid #fff;
  color: #fff;
}

#hero .btn-get-started:hover {
  background: #2dc997;
  border: 2px solid #2dc997;
}

/*--------------------------------------------------------------
# Navigation Menu
--------------------------------------------------------------*/
/* Nav Menu Essentials */
.nav-menu, .nav-menu * {
  margin: 0;
  padding: 0;
  list-style: none;
}

.nav-menu ul {
  position: absolute;
  display: none;
  top: 100%;
  left: 0;
  z-index: 99;
}

.nav-menu li {
  position: relative;
  white-space: nowrap;
}

.nav-menu > li {
  float: left;
}
```

```
.nav-menu li:hover > ul,
.nav-menu li.sfHover > ul {
  display: block;
}

.nav-menu ul ul {
  top: 0;
  left: 100%;
}

.nav-menu ul li {
  min-width: 180px;
}

/* Nav Menu Arrows */
.sf-arrows .sf-with-ul {
  padding-right: 30px;
}

.sf-arrows .sf-with-ul:after {
  content: "\f107";
  position: absolute;
  right: 15px;
  font-family: FontAwesome;
  font-style: normal;
  font-weight: normal;
}

.sf-arrows ul .sf-with-ul:after {
  content: "\f105";
}

/* Nav Meu Container */
#nav-menu-container {
  float: right;
  margin: 0;
}
```

```css
@media (max-width: 768px) {
  #nav-menu-container {
    display: none;
  }
}

/* Nav Meu Styling */
.nav-menu a {
  padding: 0 8px 10px 8px;
  text-decoration: none;
  display: inline-block;
  color: #fff;
  font-family: "Poppins", sans-serif;
  font-weight: 400;
  text-transform: uppercase;
  font-size: 13px;
  outline: none;
}

.nav-menu > li {
  margin-left: 10px;
}

.nav-menu > li > a:before {
  content: "";
  position: absolute;
  width: 100%;
  height: 2px;
  bottom: 0;
  left: 0;
  background-color: #2dc997;
  visibility: hidden;
  transform: scaleX(0);
  transition: all 0.3s ease-in-out 0s;
}

.nav-menu a:hover:before, .nav-menu li:hover > a:before, .nav-menu .menu-active > a:before {
```

```
  visibility: visible;
  transform: scaleX(1);
}

.nav-menu ul {
  margin: 4px 0 0 0;
  border: 1px solid #e7e7e7;
}

.nav-menu ul li {
  background: #fff;
}

.nav-menu ul li:first-child {
  border-top: 0;
}

.nav-menu ul li a {
  padding: 10px;
  color: #333;
  transition: 0.3s;
  display: block;
  font-size: 13px;
  text-transform: none;
}

.nav-menu ul li a:hover {
  background: #2dc997;
  color: #fff;
}

.nav-menu ul ul {
  margin: 0;
}

/* Mobile Nav Toggle */
#mobile-nav-toggle {
```

```
  position: fixed;
  right: 0;
  top: 0;
  z-index: 999;
  margin: 20px 20px 0 0;
  border: 0;
  background: none;
  font-size: 24px;
  display: none;
  transition: all 0.4s;
  outline: none;
  cursor: pointer;
}

#mobile-nav-toggle i {
  color: #fff;
}

@media (max-width: 768px) {
  #mobile-nav-toggle {
    display: inline;
  }
}

/* Mobile Nav Styling */
#mobile-nav {
  position: fixed;
  top: 0;
  padding-top: 18px;
  bottom: 0;
  z-index: 998;
  background: rgba(52, 59, 64, 0.9);
  left: -260px;
  width: 260px;
  overflow-y: auto;
  transition: 0.4s;
}
```

```css
#mobile-nav ul {
  padding: 0;
  margin: 0;
  list-style: none;
}

#mobile-nav ul li {
  position: relative;
}

#mobile-nav ul li a {
  color: #fff;
  font-size: 16px;
  overflow: hidden;
  padding: 10px 22px 10px 15px;
  position: relative;
  text-decoration: none;
  width: 100%;
  display: block;
  outline: none;
}

#mobile-nav ul li a:hover {
  color: #fff;
}

#mobile-nav ul li li {
  padding-left: 30px;
}

#mobile-nav ul .menu-has-children i {
  position: absolute;
  right: 0;
  z-index: 99;
  padding: 15px;
  cursor: pointer;
  color: #fff;
```

```
}

#mobile-nav ul .menu-has-children i.fa-chevron-up {
  color: #2dc997;
}

#mobile-nav ul .menu-item-active {
  color: #2dc997;
}

#mobile-body-overly {
  width: 100%;
  height: 100%;
  z-index: 997;
  top: 0;
  left: 0;
  position: fixed;
  background: rgba(52, 59, 64, 0.9);
  display: none;
}

/* Mobile Nav body classes */
body.mobile-nav-active {
  overflow: hidden;
}

body.mobile-nav-active #mobile-nav {
  left: 0;
}

body.mobile-nav-active #mobile-nav-toggle {
  color: #fff;
}
```

```
/*--------------------------------------------------------------
# Sections
--------------------------------------------------------------*/
section {
  overflow: hidden;
}

/* Sections Header
--------------------------------*/
.section-header .section-title {
  font-size: 32px;
  color: #111;
  text-transform: uppercase;
  text-align: center;
  font-weight: 700;
  margin-bottom: 5px;
}

.section-header .section-description {
  text-align: center;
  padding-bottom: 40px;
  color: #999;
}

/*--------------------------------------------------------------
# Breadcrumbs
--------------------------------------------------------------*/
.breadcrumbs {
  padding: 20px 0;
  background-color: whitesmoke;
  min-height: 40px;
  margin-top: 92px;
}

.breadcrumbs h2 {
  font-size: 24px;
  font-weight: 300;
```

```
  margin: 0;
}

@media (max-width: 992px) {
  .breadcrumbs h2 {
    margin: 0 0 10px 0;
  }
}

.breadcrumbs ol {
  display: flex;
  flex-wrap: wrap;
  list-style: none;
  padding: 0;
  margin: 0;
  font-size: 14px;
}

.breadcrumbs ol li + li {
  padding-left: 10px;
}

.breadcrumbs ol li + li::before {
  display: inline-block;
  padding-right: 10px;
  color: #6c757d;
  content: "/";
}

@media (max-width: 768px) {
  .breadcrumbs .d-flex {
    display: block !important;
  }
  .breadcrumbs ol {
    display: block;
  }
  .breadcrumbs ol li {
```

```
  }
}

/* About Us Section
------------------------------*/
#about {
  background: #fff;
  padding: 80px 0;
}

#about .about-container .background {
  min-height: 300px;
  background: url(../img/img1.jpg) center top no-repeat;
  margin-bottom: 10px;
}

#about .about-container .content {
  background: #fff;
}

#about .about-container .title {
  color: #333;
  font-weight: 700;
  font-size: 32px;
}

@media (max-width: 768px) {
  #about .about-container .title {
    padding-top: 15px;
  }
}

#about .about-container p {
  line-height: 26px;
}

#about .about-container p:last-child {
```

```
  margin-bottom: 0;
}

#about .about-container .icon-box {
  background: #fff;
  background-size: cover;
  padding: 0 0 30px 0;
}

#about .about-container .icon-box .icon {
  float: left;
  background: #fff;
  width: 64px;
  height: 64px;
  display: flex;
  justify-content: center;
  align-items: center;
  flex-direction: column;
  text-align: center;
  border-radius: 50%;
  border: 2px solid #2dc997;
}

#about .about-container .icon-box .icon i {
  color: #2dc997;
  font-size: 24px;
}

#about .about-container .icon-box .title {
  margin-left: 80px;
  font-weight: 500;
  margin-bottom: 5px;
  font-size: 18px;
  text-transform: uppercase;
}
```

CHAPTER-9
MAINTENANCE

The maintenance phase involves making changes to hardware, software, and documentation to support its operational effectiveness. It includes making changes to improve a system's performance, correct problems, enhance security, or address user requirements. To ensure modifications do not disrupt operations or degrade a system's performance or security, organizations should establish appropriate change management standards and procedures.

Routine changes are not as complex as major modifications and can usually be implemented in the normal course of business. Routine change controls should include procedures for requesting, evaluating, approving, testing, installing, and documenting website modifications. Maintaining accurate, up-to-date hardware and software inventories is a critical part of all change management processes. Management should carefully document all modifications to ensure accurate system inventories. Management should coordinate all technology related changes through an oversight committee and assign an appropriate party responsibility for administering software patch management programs. Quality assurance, security, audit, regulatory compliance, network, and end-user personnel should be appropriately included in change management processes. Risk and security review should be done whenever a system modification is implemented to ensure controls remain in place.

For maintenance of thewebsite:

1. The database has to be updated regularly according to new available information.
2. Redundant and false information must be removed from the database.
3. Newer versions of PHP and MYSQL can be used for up gradation of website and to improve the overall performance of the system.

CHAPTER-10

FUTURE SCOPE AND FUTUREENHANCEMENT

PROJECT NAME: HOME TUTORIAL POINT

With technological advancements at their peak, Indian Education System went through some much-needed modifications in the recent past. One of them being the introduction of online education. The ease and opportunity to attain additional knowledge has boosted the scope of online education in India. As the majority of the students generally pursue their graduation and other level courses through on-campus education and prefer to take additional courses online. Online education in India is at a very early stage. Yet, India is filled with passionate youth who can enhance the possibilities of Indian Education System. Looking at the current trends, we can expect growth in enrollment rates in distance online education programs. It will evolve the continuous learning process. According to a recent study in a global level online learning program, India is reported to have the second highest number of online course enrollments.

CONCLUSION

CHAPTER-11

We have successfully implemented the site 'HOME TUTORIAL POINT'. With the help of various links and tools, we have been able to provide a site which will be live soon and running on the web. We have been successful in our attempt to take care of the needs of both the user as well as the administrator. Finally, we hope that this will go a long way in popularizing.

BIBLIOGRAPHY

1. www.javatutpoint.com
2. www.w3schools.com
3. www.getbootstrap.com
4. www.codeigniter.com
5. www.stackoverflow.com
6. www.fontawesome.io
7. www.sweetalert.js.org
8. www.codepen.io
9. Learn HTML and CSS faster (Mark Myers)
10. Wikipedia