**FULL STACK DEVELOPMENT**

(RESTAURANT MANAGEMENT SYSTEM)

*Summer Internship Report Submitted in partial fulfillment*

*of the requirement for under graduate degree of*

**Bachelor of Technology**

In

**COMPUTER SCIENCE ENGINEERING**

By

**K.Pranavi**

**221710308028**

[**https://github.com/pranavipranitha/**](https://github.com/pranavipranitha/)

*Under the Guidance of*

Assistant Professor



Department Of Computer Science Engineering

GITAM School of Technology

GITAM (Deemed to be University)

Hyderabad-502329

                                                        July 2020

*Candidate’s Declaration*

I hereby declare that the project work entitled **“RESTAURANT MANAGEMENT SYSTEM”** submitted to the GITAM (deemed to be University) , Hyderabad, is record of an original work done by me under the guidance ofmy mentor **MR . ARAVIND**. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

(K.Pranavi)

(221710308028)

Counter Signed By

**ACKNOWLEDGEMENT**

I take this opportunity to express my profound gratitude and deep regards to our mentors for their exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by them time to time shall carry me a long way in the journey of life on which I am about to embark. The project helped me learn how to do proper Research and I learned about many new things while doing the project. Lastly, I thank almighty, my parents and friends for their constant encouragement without which this project would not be possible.

H.O.D

(Computer Science Department)

**ABSTRACT**

Running a restaurant is hectic enough as it is, so why not make the day-to-day processes easier by having a system that will help ease the workload for you? There are so many day-to-day processes that restaurants have to deal with. This can range from scheduling in employees, managing HR , Monitoring employee attendance to preparing for payroll and to keep record of transactions and database. In current marketplace ,there is a great value for food, restaurants and its management. There is day-by-day increment on the number of restaurants and food places that are emerging today. It can be considered as a rapid growth in the field of business and food restaurants and its management system. The management system applied for every restaurant is restaurant is different from the other one. Some restaurants may be bigger while the other may be smaller but every restaurant or hotel requires a management system and this is termed as Restaurant Management System.

RMS that is , Restaurant management systems are the crucial technologies that enables a single outlet or enterprise to better serve its customers and aid employees with food and beverage transactions and controls. Restaurant management system is database program that keeps record of all transactions carried out in the restaurant on daily basis. The restaurant management system helps the restaurant management to keep adequate record of all transactions carried out and does that will still be carried out by the restaurant and maintain the database of the restaurant.

While investing in elements such as marketing and décor will go a long way in advance in the growth of a restaurant , Investing in the right technology also plays a huge role, with restaurant management software being among the “must haves“. Every restaurant out there ,whether small ,medium-sized, or large ,will benefit greatly be switching from manual restaurant management processes to automated or software based ones.

**TABLE OF CONTENTS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | TITLE | | PG NO | |
| 1 | | INTRODUCTION | | 7 | |
|  | | 1.1 WEB DEVELOPMENT | | 7-8 | |
|  | | 1.2 WEB SITE | | 8 | |
|  | | 1.3 WEB PAGE | | 9 | |
| 2 | | FRONTEND AND BACKEND | | 10 | |
|  | | 2.1 TECHNOLOGY RELATED TO FULL STACK | | 10 | |
|  | | 2.1.1 FRONTEND | | 10 | |
|  | | 2.1.2 FRONTEND LANGUAGES | | 10 | |
|  | | 2.1.3 FRONTEND FRAMEWORKS AND LIABRARIES | | 11 | |
|  | | 2.2 BACKEND | | 12 | |
|  | | 2.2.1 BACKEND TOOLS | | 12 | |
| 3 | | STEPS TO CREATE WEBSITE | | 13 | |
|  | | 3.1 UI DEVELOPMENT | | 14 | |
|  | | 3.1.1 HTML | | 14-15 | |
|  | | 3.1.2 CSS | | 16-18 | |
|  | | 3.1.3 BOOTSTRAP | | 18-19 | |
|  | | 3.2 SCRPTING | | 19-20 | |
|  | | 3.2.1 SERVER SIDE SCRIPTING | | 20 | |
|  | | 3.2.3 CLIENT SIDE SCRIPTING | | 21 | |
|  | | 3.2.4 SCRIPTING LANGUAGES | | 22-26 | |
|  | | 3.3 DATABASE | | 26 | |
|  | | 3.4 MY SQL | | 27-28 | |
| 4 | | REQUIREMENTS | | 29 | |
|  | | 4.1 HARDWARE REQUIREMENTS | | 29 | |
|  | | 4.2 SOFTWARE REQUIREMENTS | | 29 | |
| 5 | | DATA FLOW DIAGRAMS | | 30-33 | |
| 6 | | PROJECT | | 34 | |
| 7 | | MODULES ASSIGNED | | 35 | |
|  | | 7.1 JAVASCRIPT | | 35 | |
|  | | 7.2 BOOKING FORM | | 35 | |
|  | | 7.3 MYSQL | | 35 | |
| 8 | | CODES AND OUTPUTS AND WIREFRAMES | | 36-58 | |
| 9 | | MAINTAINACE | | 59 | |
| 10 | | FUTURE SCOPE AND ENHANCMENT | | 60 | |
| 11 | | CONCLUSION | | 61 | |

**LIST OF FIGURES**

FIG -1 Full stack development------------------

FIG -2.1.2 Frontend languages----------------------

FIG -2.1.3 Frontend frameworks--------------------

FIG -3.1 Steps to create website-------------------

FIG -3.2.1 Client vs server scripting----------------

FIG -3.2.2 Client side scripting----------------------

FIG -5.1 Data flow diagrams-----------------------

FIG -5.2 Context level diagram--------------------

FIG -5.3 First level diagram------------------------

FIG -5.4 Second level diagram---------------------

FIG -5.5 Third level diagram-----------------------

FIG -5.6 Use case diagram-------------------------

**CHAPTER-1**

**INTRODUCTION**

* 1. **WEB DEVELOPMENT:**

Web development is 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 a simple single static page of plain text to complex web-based internet applications (web apps), 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/server-side scripting, web server and network security configuration, and e-commerce development.

There are three kinds of web developer specialization: front-end developer, back-end developer, and full-stack developer. Front-end developers are responsible for behaviour and visuals that run in the user browser, while back-end developers deal with the servers.

Front-end web development is the practice of converting data to a graphical interface, through the use of HTML, CSS, and JavaScript, so that users can view and interact with that data.

There are several tools and platforms (word press etc..) available that can be used to develop the front end of a website, and understanding which tools are best fit for specific tasks marks the difference between developing a hacked site and a well designed, scalable site . The developer of the front end keeps these points in mind, utilizing available tools and techniques to reach this end.

Accessibility Edit With continuing development for mobile devices, such as smart phones and tablets, designers need to ensure that their site comes up correctly in browsers on all devices. This can be done by creating a responsive web design using stylesheets in CSS. Performance Edit Performance goals are chiefly concerned with render time, manipulating the HTML, CSS, and JavaScript to ensure that the site opens up quickly.

**The advantage of being a full stack web developer is:**

* You can master all the techniques involved in a development project
* You can make a prototype very rapidly
* You can provide help to all the team members
* You can reduce the cost of the project
* You can reduce the time used for team communication
* You can switch between front and back end development based on requirements
* You can better understand all aspects of new and upcoming technologies

**Disadvantages**

* The solution chosen can be wrong for the project
* The solution chosen can be dependent on developer skills
* The solution can generate a key person risk
* Being a full stack developer is increasingly complex

## 1.2WEB – SITE

A website is a collection of related [web pages,](https://en.wikipedia.org/wiki/Web_page) including [multimedia c](https://en.wikipedia.org/wiki/Multimedia)ontent, typically identified with a common [domain name, a](https://en.wikipedia.org/wiki/Domain_name)nd published on at least one [web server.](https://en.wikipedia.org/wiki/Web_server) A website may be accessible via a public [Internet Protocol (](https://en.wikipedia.org/wiki/Internet_Protocol)IP) network, such as the [Internet,](https://en.wikipedia.org/wiki/Internet) or a private [local area network (](https://en.wikipedia.org/wiki/Local_area_network)LAN), by referencing a [uniform resource locator (](https://en.wikipedia.org/wiki/URL)URL) that identifies the site.

Websites have many functions and can be used in various fashions; a website can be a [personal website,](https://en.wikipedia.org/wiki/Personal_website) a commercial website for a company, a [government website o](https://en.wikipedia.org/wiki/E-Government)r a [non-profit organization w](https://en.wikipedia.org/wiki/Nonprofit_organization)ebsite. Websites are typically dedicated to a particular topic or purpose, ranging from entertainment and [social networking](https://en.wikipedia.org/wiki/Social_networking) to providing news and education. All publicly accessible websites collectively constitute the [World Wide Web,](https://en.wikipedia.org/wiki/World_Wide_Web) while private websites, such as a company's website for its employees, and are typically a part of an [intranet.](https://en.wikipedia.org/wiki/Intranet)

Web pages, which are the [building blocks](https://en.wikipedia.org/wiki/Building_block) of websites, are [documents,](https://en.wikipedia.org/wiki/Document) typically composed in [plain text](https://en.wikipedia.org/wiki/Plain_text) interspersed with formatting instructions of Hypertext Mark up Language ([HTML,](https://en.wikipedia.org/wiki/HTML) [XHTML)](https://en.wikipedia.org/wiki/XHTML). They may incorporate elements from other websites with suitable [mark up anchors.](https://en.wikipedia.org/wiki/HTML_anchor) Web pages are accessed and transported with the [Hypertext Transfer Protocol (](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)HTTP), which may optionally employ encryption [(HTTP Secure,](https://en.wikipedia.org/wiki/HTTP_Secure) HTTPS) to provide security and privacy for the user. The user's application, often a [web browser,](https://en.wikipedia.org/wiki/Web_browser) renders the page content according to its HTML mark up instructions onto a [display terminal.](https://en.wikipedia.org/wiki/Computer_monitor)

[Hyperlinking b](https://en.wikipedia.org/wiki/Hyperlink)etween web pages conveys to the reader the [site structure a](https://en.wikipedia.org/wiki/Site_map)nd guides the navigation of the site, which often starts with a [home page c](https://en.wikipedia.org/wiki/Home_page)ontaining a directory of the site [web content.](https://en.wikipedia.org/wiki/Web_content) Some websites require user registration or [subscription](https://en.wikipedia.org/wiki/Subscription) to access content. Examples of [subscription websites](https://en.wikipedia.org/wiki/Paywall) include many business sites, news websites, [academic journal w](https://en.wikipedia.org/wiki/Academic_journal)ebsites, gaming websites, file-sharing websites, [message boards,](https://en.wikipedia.org/wiki/Internet_forum) web-based [email,](https://en.wikipedia.org/wiki/Email) [social networking w](https://en.wikipedia.org/wiki/Social_networking)ebsites, websites providing real-time [stock market d](https://en.wikipedia.org/wiki/Stock_market)ata, as well as sites providing various other services. As of 2016 [end users](https://en.wikipedia.org/wiki/End_user) can access websites on a range of devices, including [desktop](https://en.wikipedia.org/wiki/Desktop_computer) and [laptop computers,](https://en.wikipedia.org/wiki/Laptop) [tablet computers,](https://en.wikipedia.org/wiki/Tablet_computer) [smartphones a](https://en.wikipedia.org/wiki/Smartphone)nd [smart TVs.](https://en.wikipedia.org/wiki/Smart_TV)

**1.3 WEB PAGE:**

A web page (or webpage) is a specific collection of information provided by a website and displayed to a user in a web browser. A website typically consists of many web pages linked together in a coherent fashion. The name "web page" is a metaphor of paper pages bound together into a book. The core element of a web page is one or more text files written in the Hypertext Mark up Language (HTML). Many web pages also make use of JavaScript code for dynamic behaviour and Cascading Style Sheets (CSS) code for presentation semantics .Images, videos, and other multimedia files are also often embedded in web pages. Each web page is identified by a distinct Uniform Resource Locator (URL). When the user inputs a URL into their browser, that page's elements are downloaded from web servers. The browser then transforms all of the elements into an interactive visual representation on the user's device. From the perspective of server-side website deployment, there are two types of web pages: static and dynamic. Static pages are retrieved from the web server's file system without any modification, while dynamic pages must be created by the web server on the fly, typically drawing from a database to fill out a web template, before being sent to the user's browser.



**Fig 1**

**CHAPTER-2**

**FRONTEND AND BACKEND**

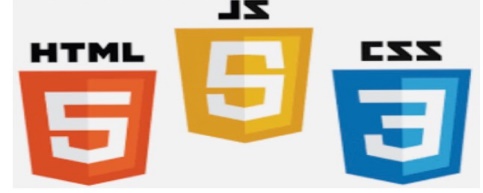
* 1. **Technology related to full stack development:**

**2.1.1 Frontend:**

It is the visible part of website or web application which is responsible for user experience. The user directly interacts with the front end portion of the web application or websites.

**2.1.2 Frontend Languages:** The front end portion is built by using some languages which are discussed below:

* **HTML:** HTML stands for Hyper Text Mark up Language. It is used to design the front end portion of web pages using mark up language. HTML is the combination of Hypertext and Mark up language. Hypertext defines the link between the web pages. The mark up language is used to define the text documentation within tag which defines the structure of web pages.
* **CSS:** Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.
* **JAVASCRIPT:** JavaScript is a famous scripting language used to create the magic on the sites to make the site interactive for the user. It is used to enhancing the functionality of a website to running cool games and web-based software.

****

**Fig2.1.2.**

**2.1.3 Front End Frameworks and Libraries:**

* **AngularJS:** Angular Js is a JavaScript open source front-end framework that is mainly used to develop single page web applications(SPAs). It is a continuously growing and expanding framework which provides better ways for developing web applications. It changes the static HTML to dynamic HTML. It is an open source project which can be freely used and changed by anyone. It extends HTML attributes with Directives, and data is bound with HTML.
* **React.js:** React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ReactJS is an open-source, component-based front end library responsible only for the view layer of the application. It is maintained by Facebook.
* **Bootstrap:** Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites.
* **jQuery:** jQuery is an open source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM), and JavaScript. Elaborating the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.
* **SASS:** It is the most reliable, mature and robust CSS extension language. It is used to extend the functionality of an existing CSS of a site including everything from variables, inheritance, and nesting with ease.

Some other libraries and frameworks are: Semantic-UI, Foundation, Materialize,

Backbone.js, Express.js, Ember.js etc.



**Fig 2.1.3.**

**2.2 Back End:**

While front end development centers on every aspect that users interact with, the back end development handles everything that is appearing behind the scenes. It provides the necessary support to the front end development. While PHP, Python, Ruby, etc. are included in back end programming languages. While you may not see many jobs seeking out especially for backend developers, rather you do see a job listed as PHP developers, Ruby developers, etc. This is because the knowledge of a particular language is an important factor, wholly deciding if a candidate is fit for the specific job title.

**2.2.1 Back end Tools:**

* **Web Servers**

Web servers are computer programs that store, process and deliver web pages to the users. The most popular ones include Apache (an open-source web server which is currently used by 50% of all websites) and NGINX which is good for reverse proxying, caching, load balancing and media streaming processes.

* **Database Management Systems**

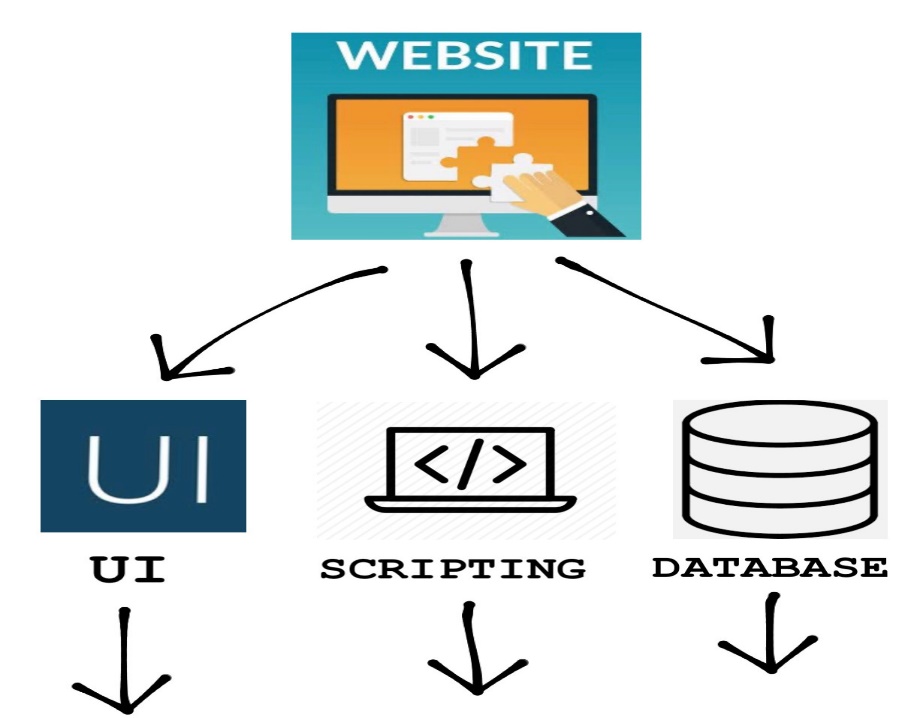
A Database Management System (DBMS) is a collection of programs that enables its users to access a database, manipulate, interpret and represent data. MySQL is the world’s most popular open-source relational database. It’s not only accessible but also free. Its ease of setup and speedy performances make it a favourite among many backend developers. On the other hand, MongoDB is an open-source NoSQL database system which is closely associated with a JavaScript-based set of technologies like Express JS, AngularJS, and NodeJS.

**CHAPTER-3**

**STEPS TO CREATE WEBSITE**

**STEPS INVOLVED:**

* Creating a User Interface
* Scripting of both server end and client end
* Creating a backed database

****



**Fig-3.1**

**3.1 UI DEVELOPMENT:**

**3.1.1** **HTML Hypertext Mark up Language (HTML) :**

HTML is the standard mark up language for Web pages. HTML elements are the building blocks of HTML pages. HTML elements are represented by <> tags

HTML can embed programs written in a [scripting language s](https://en.wikipedia.org/wiki/Scripting_language)uch as [JavaScript w](https://en.wikipedia.org/wiki/JavaScript)hich affect the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. The [World Wide Web Consortium (](https://en.wikipedia.org/wiki/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 mark up 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,](https://en.wikipedia.org/wiki/Document_type_declaration) which triggers [standards mode r](https://en.wikipedia.org/wiki/Standards_mode)endering.

An HTML element is a start tag and an end tag with content in between:

<h1>This is a Heading</h1>

Start tag Element content End tag

<h1> This is a Heading </h1>

<p> This is paragraph. </p>

HTML Attributes

HTML elements can have attributes

Attributes provide additional information about the element

Attributes come in name/value pairs like charset="utf-8"

A Simple HTML Document

<!DOCTYPE html>

<html lang="en">

<meta charset="utf-8">

<title>Page Title</title>

<body>

   <h1>This is a Heading</h1>

   <p>This is a paragraph.</p>

   <p>This is another paragraph.</p>

</body>

</html>

Example Explained

HTML elements are the building blocks of HTML pages.

The <!DOCTYPE html> declaration defines this document to be HTML5

The <html> element is the root element of an HTML page

The lang attribute  defines the language of the document

The <meta> element contains meta information about the document

The charset attribute defines the character set used in the document

The <title> element specifies a title for the document

The <body> element contains the visible page content

The <h1> element defines a large heading

The <p> element defines a paragraph

HTML Documents

All HTML documents must start with a document type declaration: <!DOCTYPE html>.

The HTML document itself begins with <html> and ends with </html>.

The visible part of the HTML document is between <body> and </body>.

HTML Document Structure

Below is a visualization of an HTML document (an HTML Page):

<html>

<head>

<title>Page title</title>

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

</body>

</html>

Note: Only the content inside the <body> section (the white area above) is displayed in a browser.

HTML Headings

HTML headings are defined with <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading:

Example

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

<h3>This is heading 3</h3>

**3.1.2 CSS Cascading Style Sheets (CSS) :**

CSS describes how HTML elements are to be displayed

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the [layout,](https://en.wikipedia.org/wiki/Page_layout) [colours,](https://en.wikipedia.org/wiki/Color) and [fonts.](https://en.wikipedia.org/wiki/Typeface) This separation can improve content [accessibility,](https://en.wikipedia.org/wiki/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.  Although most often used to set the visual style of [web pages a](https://en.wikipedia.org/wiki/Web_page)nd user interfaces written in [HTML a](https://en.wikipedia.org/wiki/HTML)nd [XHTML,](https://en.wikipedia.org/wiki/XHTML) the language can be applied to any [XML d](https://en.wikipedia.org/wiki/XML)ocument, including [plain XML,](https://en.wikipedia.org/wiki/Plain_Old_XML) [SVG a](https://en.wikipedia.org/wiki/Scalable_Vector_Graphics)nd [XUL,](https://en.wikipedia.org/wiki/XUL) and is applicable to rendering in [speech,](https://en.wikipedia.org/wiki/Speech_synthesis) or on other media. Along with HTML and [JavaScript,](https://en.wikipedia.org/wiki/JavaScript) CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for [web applications,](https://en.wikipedia.org/wiki/Web_applications) and user interfaces for many mobile applications.

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 (](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium)W3C). Internet media type text/css is registered for use with CSS by [RFC 2318 (](https://tools.ietf.org/html/rfc2318)March 1998). The W3C operates a free [CSS validation service f](https://en.wikipedia.org/wiki/W3C_Markup_Validation_Service#CSS_validation)or CSS documents.

CSS Example

<style>

body {background -color: light blue ; text-align:center;}

h1 {color:blue; font-size:40px;}

p {font-family:verdana; font-size:20px;}

</style>

Click on the "Try it Yourself" button to see how it works.

CSS Syntax

A CSS rule consists of a selector and a declaration block:

CSS selector: The selector points to the HTML element to style (h1).

The declaration block (in curly braces) contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

In the following example all <p> elements will be centre -aligned, red and have a font size of 32 pixels:

Example

<style>

p {font-size:32px; color:red; text-align:center;}

</style>

Same example can also be written like this:

<style>

p {

    font-size: 32px;

    color: red;

    text-align: center;

}

</style>

External Style Sheet

A CSS style sheet can be stored in an external file:

mystyle.css

body {background-color: orange; font-family:verdana}

h1 {color: white;}

p {font-size: 20px;}

External style sheets are linked to HTML pages with <link> tags:

Example

<!DOCTYPE html>

<html>

<link rel="stylesheet" href="mystyle.css">

<body>

<h1>My First CSS Example</h1>

<p>This is a paragraph.</p>

</body>

</html>

Inline Style

Example

<!DOCTYPE html>

<html>

<link rel="stylesheet" href="mystyle.css">

<body>

<h1>My First CSS Example</h1>

<p>This is a paragraph.</p>

<p style="font-size:25px">This is a paragraph.</p>

<p style="font-size:30px">This is a paragraph.</p>

</body>

</html>

Cascading Order

If different styles are specified for HTML elements, the styles will cascade into new styles with the following priority:

Priority 1: Inline styles

Priority 2: External and internal style sheets

Priority 3: Browser default

If different styles are defined on the same priority level, the last one has the highest priority.

Example

<!DOCTYPE html>

<html>

<link rel="stylesheet" href="mystyle.css">

<style>

body {background-color: lightblue;}

</style>

<body style="background-color: olivedrab">

<h1>Multiple Styles Cascades into One</h1>

<p>Try experimenting by removing styles to see how the cascading stylesheets work.</p>

<p>Try removing the inline first, then the internal, then the external.</p>

</body>

</html>

**3.1.3 BOOTSTRAP:**

**Bootstrap** is a [free and open-source f](https://en.wikipedia.org/wiki/Free_and_open-source_software)ront-end [web framework f](https://en.wikipedia.org/wiki/Web_framework)or designing [websites](https://en.wikipedia.org/wiki/Website) and [web applications.](https://en.wikipedia.org/wiki/Web_application) It contains [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS-](https://en.wikipedia.org/wiki/CSS)based design templates for [typography,](https://en.wikipedia.org/wiki/Typography) forms, buttons, navigation and other interface components, as well as optional [JavaScript](https://en.wikipedia.org/wiki/JavaScript) extensions. Unlike many web frameworks, it concerns itself with [front-end development o](https://en.wikipedia.org/wiki/Front-end_web_development)nly.

Bootstrap 4 is the newest version of Bootstrap. Bootstrap 4 supports all major browsers except Internet Explorer 9.If you require support for IE9 or IE8, you must use Bootstrap 3.Bootstrap Containers: The container class is one of the most important Bootstrap classes. It provides margins, padding, alignments, and more, to HTML elements.

Example

<div class="container">

<h1>This is a paragraph</h1>

<p>This is a paragraph</p>

<p>This is a paragraph</p>

<p>This is a paragraph</p>

<p>This is a paragraph</p>

</div>

Installing and linking bootstrap to the HTML page:

* Install bootstrap from [getbootstrap.com](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 script tag.

Eg: <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:**

Web script, a computer programming language for adding dynamic capabilities to World Wide Web pages. Web pages marked up with HTML (hypertext mark up language) or XML (extensible mark up language) are largely static documents. Web scripting can add information to a page as a reader uses it or let the reader enter information that may, for example, be passed on to the order department of an online business. CGI (common gateway interface) provides one mechanism; it transmits requests and responses between the readers Web browser and the Web server that provides the page. The CGI component on the server contains small programs called scripts that take information from the browser system or provide it for display. A simple script might ask the readers name, determine the Internet address of the system that the reader uses, and print a greeting. Scripts may be written in any programming language, but, because they are generally simple text-processing routines, computer scripting languages such as PERL are particularly appropriate.

Another approach is to use a language designed for Web scripts to be executed by the browser. JavaScript is one such language, designed by the Netscape Communications Corp.; it may be used with both Netscape’s and Microsoft Corporations browsers. JavaScript is a simple language, quite different from Java. A JavaScript program may be embedded in a Web page with the HTML tag <script language=JavaScript>. JavaScript instructions following that tag will be executed by the browser when the page is selected. In order to speed up display of dynamic (interactive) pages, JavaScript is often combined with XML or some other language for exchanging information between the server and the client’s browser. In particular, the XML Http Request command enables asynchronous data requests from the server without requiring the server to resend the entire Web page. This approach, or philosophy, of programming is called Ajax (asynchronous JavaScript and XML).

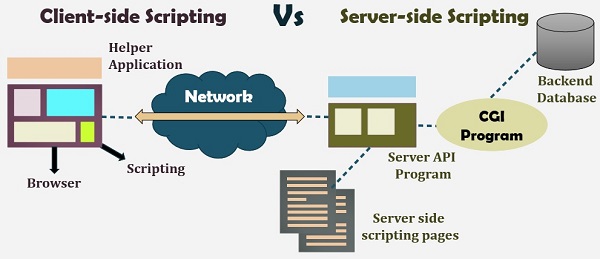
VB Script is a subset of Visual Basic. Originally developed for Microsoft Office suite of programs, it was later used for Web scripting as well. Its capabilities are similar to those of JavaScript, and it may be embedded in HTML in the same fashion.

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:**

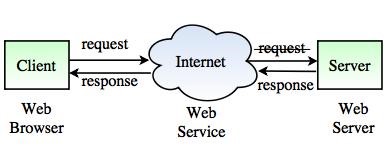
Server-side scripting is a technique used in web development which involves employing scripts on a web server which produce a response customized for each user’s (client’s) request to the website. The alternative is for the web server itself to deliver a static web page. Scripts can be written in any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



**FIG 3.2.1**

**3.2.3 CLIENT SIDE SCRIPTING:**

Client-side scripting is changing interface behaviours within a specific web page in response to mouse or keyboard actions, or at specified timing events. In this case, the dynamic behaviour 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 20 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 Java Script. Flow of request from browser to server:



**FIG 3.2.2**

**3.2.4 SCRIPTING LANGUAGES:**

* **PHP**

PHP is a [server-side scripting l](https://en.wikipedia.org/wiki/Server-side_scripting)anguage designed primarily for [web development b](https://en.wikipedia.org/wiki/Web_development)ut also used as a [general-purpose programming language.](https://en.wikipedia.org/wiki/General-purpose_programming_language) Originally created by [Rasmus Lerdorf](https://en.wikipedia.org/wiki/Rasmus_Lerdorf) in 1994, the PHP [reference implementation i](https://en.wikipedia.org/wiki/Reference_implementation)s now produced by The PHP Development Team. PHP originally stood for *Personal Home Page*, but it now stands for the [recursive acronym](https://en.wikipedia.org/wiki/Recursive_acronym) *PHP: Hypertext Pre processor*.

PHP code may be embedded into [HTML](https://en.wikipedia.org/wiki/HTML) or HTML5 [mark up,](https://en.wikipedia.org/wiki/Markup_language) or it can be used in combination with various [web template systems,](https://en.wikipedia.org/wiki/Web_template_system) [web content management systems a](https://en.wikipedia.org/wiki/Web_content_management_system)nd [web frameworks.](https://en.wikipedia.org/wiki/Web_framework) PHP code is usually processed by a PHP [interpreter i](https://en.wikipedia.org/wiki/Interpreter_%28computing%29)mplemented as a [module i](https://en.wikipedia.org/wiki/Plugin_%28computing%29)n the web server or as a [Common Gateway Interface (](https://en.wikipedia.org/wiki/Common_Gateway_Interface)CGI) [executable.](https://en.wikipedia.org/wiki/Executable) The [web server s](https://en.wikipedia.org/wiki/Web_server)oftware combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated [web page.](https://en.wikipedia.org/wiki/Web_page) PHP code may also be executed with a [command-line interface (](https://en.wikipedia.org/wiki/Command-line_interface)CLI) and can be used to implement [standalone graphical applications.](https://en.wikipedia.org/wiki/Computer_software)

The standard PHP interpreter, powered by the [Zend Engine,](https://en.wikipedia.org/wiki/Zend_Engine) is [free software r](https://en.wikipedia.org/wiki/Free_software)eleased under the [PHP License.](https://en.wikipedia.org/wiki/PHP_License) PHP has been widely ported and can be deployed on most web servers on almost every [operating system a](https://en.wikipedia.org/wiki/Operating_system)nd [platform,](https://en.wikipedia.org/wiki/Computing_platform) free of charge.

The PHP language evolved without a written [formal specification](https://en.wikipedia.org/wiki/Formal_specification) or standard until 2014, leaving the canonical PHP interpreter as a [*de facto* s](https://en.wikipedia.org/wiki/De_facto)tandard. Since 2014 work has gone on to create a formal PHP specification.

**Installing PHP**

Step1:download the files. Download the latest PHP 5 ZIP package from [www.php.net/downloads.php.](http://www.php.net/downloads.php)

Step 2: extract the files

Step 3: configure php.ini.

Step 4: add C: php to the path environment variable. ... V.

Step 5: configure PHP as an Apache module.

Step 6: test a PHP file.

Or we can install **Xampp** which have inbuilt php, mysql, apache server.

* **JAVASCRIPT**

  JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as Live Script, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name Live Script. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

Applications of Java script Programming. As mentioned before, Java script is one of the most widely used programming languages (Front-end as well as Back-end). It has it's presence in almost every area of software development. I'm going to list few of them here:

Client side validation - This is really important to verify any user input before submitting it to the server and Java script plays an important role in validating those inputs at front-end itself.

Manipulating HTML Pages – Java script helps in manipulating HTML page on the fly. This helps in adding and deleting any HTML tag very easily using java script and modify your HTML to change its look and feel based on different devices and requirements.

User Notifications - You can use Java script to raise dynamic pop-ups on the webpages to give different types of notifications to your website visitors.

Back-end Data Loading – Java script provides Ajax library which helps in loading back-end data while you are doing some other processing. This really gives an amazing experience to your website visitors.

Presentations - JavaScript also provides the facility of creating presentations which gives website look and feel. JavaScript provides Reveal JS and Bespoke JS libraries to build a web-based slide presentations.

Server Applications - Node JS is built on Chrome's Java script runtime for building fast and scalable network applications. This is an event based library which helps in developing very sophisticated server applications including Web Servers.

This list goes on, there are various areas where millions of software developers are happily using Java script to develop great websites and others software.

Advantages of JavaScript

The merits of using JavaScript are −

Less server interaction − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.

Immediate feedback to the visitors − They don't have to wait for a page reload to see if they have forgotten to enter something.

Increased interactivity − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.

Richer interfaces − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features −

Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.

JavaScript cannot be used for networking applications because there is no such support available.

JavaScript doesn't have any multi-threading or multiprocessor capabilities.

* **JQUERY:**

JQuery is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [JavaScript library](https://en.wikipedia.org/wiki/JavaScript_library) designed to simplify the [client-side scripting o](https://en.wikipedia.org/wiki/Client-side_scripting)f [HTML. I](https://en.wikipedia.org/wiki/HTML)t is [free, open-source software u](https://en.wikipedia.org/wiki/Free_and_open_source_software)sing the permissive [MIT license.](https://en.wikipedia.org/wiki/MIT_license) [Web](https://en.wikipedia.org/wiki/World_Wide_Web) analysis indicates that it is the most widely deployed JavaScript library by a large margin.  create [animations,](https://en.wikipedia.org/wiki/Animation) handle [events,](https://en.wikipedia.org/wiki/Event_%28computing%29) and develop [Ajax](https://en.wikipedia.org/wiki/Ajax_%28programming%29) applications. jQuery also provides capabilities for developers to create [plug-ins o](https://en.wikipedia.org/wiki/Plug-in_%28computing%29)n top of the JavaScript library. This enables developers to create [abstractions f](https://en.wikipedia.org/wiki/Abstraction_%28computer_science%29)or 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 a](https://en.wikipedia.org/wiki/Dynamic_web_page)nd Web applications.

  The set of [jQuery core features—](https://en.wikipedia.org/wiki/JQuery#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](https://en.wikipedia.org/wiki/Comparison_of_JavaScript_frameworks) like [YUI v3](https://en.wikipedia.org/wiki/YUI_Library) and [Dojo,](https://en.wikipedia.org/wiki/Dojo_Toolkit) later stimulating the creation of the standard Selectors API.

[Microsoft](https://en.wikipedia.org/wiki/Microsoft) and [Nokia](https://en.wikipedia.org/wiki/Nokia) bundle jQuery on their platforms. Microsoft includes it with [Visual Studio f](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio)or use within Microsoft's [ASP.NET AJAX a](https://en.wikipedia.org/wiki/ASP.NET_AJAX)nd [ASP.NET MVC f](https://en.wikipedia.org/wiki/ASP.NET_MVC)rameworks while Nokia has integrated it into the Web Run-Time widget development platform.

* **AJAX**

**Ajax** (also **AJAX** short for "asynchronous [JavaScript](https://en.wikipedia.org/wiki/JavaScript) and [XML"](https://en.wikipedia.org/wiki/XML)) is a set of [Web development](https://en.wikipedia.org/wiki/Web_development) techniques using many Web technologies on the [client side](https://en.wikipedia.org/wiki/Client_side) to create [asynchronous Web applications.](https://en.wikipedia.org/wiki/Asynchronous_I/O) With Ajax, Web applications can send data to and retrieve from a [server](https://en.wikipedia.org/wiki/Web_server) asynchronously (in the background) without interfering with the display and behaviour 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 f](https://en.wikipedia.org/wiki/JSON)or XML due to the advantages of being native to JavaScript.

  Ajax is not a single technology, but rather a group of technologies. [HTML a](https://en.wikipedia.org/wiki/Hypertext_Markup_Language)nd [CSS c](https://en.wikipedia.org/wiki/Cascading_Style_Sheets)an be used in combination to mark up and style information. The [DOM i](https://en.wikipedia.org/wiki/Document_Object_Model)s accessed with JavaScript to dynamically display – and allow the user to interact with – the information presented. JavaScript and the [XML Http Request](https://en.wikipedia.org/wiki/XMLHttpRequest) object provide a method for exchanging data asynchronously between browser and server to avoid full page reloads.

* **JSON:**

[p](https://en.wikipedia.org/wiki/Web_server)ackage developed by Apache Friends, consisting mainly of the [Apache HTTP Server,](https://en.wikipedia.org/wiki/Apache_HTTP_Server) [MariaDB](https://en.wikipedia.org/wiki/MariaDB) [database,](https://en.wikipedia.org/wiki/Database) and [interpreters f](https://en.wikipedia.org/wiki/Interpreter_%28computing%29)or scripts written in the [PHP a](https://en.wikipedia.org/wiki/PHP)nd [Perl programming languages.](https://en.wikipedia.org/wiki/Perl) 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 [In computing,](https://en.wikipedia.org/wiki/Computing) **JavaScript Object Notation** or **JSON** is an [open- standard file format](https://en.wikipedia.org/wiki/Open_standard) that uses [human-readable](https://en.wikipedia.org/wiki/Human-readable_medium) text to transmit data objects consisting of [attribute–value pairs a](https://en.wikipedia.org/wiki/Attribute%E2%80%93value_pair)nd [array data types (](https://en.wikipedia.org/wiki/Array_data_type)or any other [serializable v](https://en.wikipedia.org/wiki/Serialization)alue). It is a very common data format used for [asynchronous b](https://en.wikipedia.org/wiki/Asynchronous_I/O)rowser/server communication, including as a replacement for [XML i](https://en.wikipedia.org/wiki/XML)n some [AJAX-](https://en.wikipedia.org/wiki/Ajax_%28programming%29)style systems.

  JSON is [a language-independent d](https://en.wikipedia.org/wiki/Language-independent_specification)ata format. It was derived from [JavaScript,](https://en.wikipedia.org/wiki/JavaScript) but as of 2017 many [programming languages i](https://en.wikipedia.org/wiki/Programming_language)nclude code to generate and [parse J](https://en.wikipedia.org/wiki/Parsing)SON-format data. The official Internet [media type f](https://en.wikipedia.org/wiki/Media_type)or JSON is application/json. JSON filenames use the extension .json.  0s; two competing standards, [RFC 7159](https://tools.ietf.org/html/rfc7159) and [ECMA-404,](https://www.ecma-international.org/publications/standards/Ecma-404.htm) defined it in 2013. The ECMA standard describes only the allowed syntax, whereas the RFC covers some security and interoperability considerations.

  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.](https://tools.ietf.org/html/rfc7493)

* **FEATURES**

The latest releases of [Apache,](https://en.wikipedia.org/wiki/Apache_HTTP_Server) [MariaDB,](https://en.wikipedia.org/wiki/MariaDB) [PHP a](https://en.wikipedia.org/wiki/PHP)nd [Perl.](https://en.wikipedia.org/wiki/Perl) It also comes with a number of other modules including [OpenSSL,](https://en.wikipedia.org/wiki/OpenSSL) [phpMyAdmin,](https://en.wikipedia.org/wiki/PhpMyAdmin) [Media Wiki,](https://en.wikipedia.org/wiki/MediaWiki) [Joomla,](https://en.wikipedia.org/wiki/Joomla) [WordPress](https://en.wikipedia.org/wiki/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).

* **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](https://en.wikipedia.org/wiki/World_Wide_Web) special tool is provided to [password-protect t](https://en.wikipedia.org/wiki/Password)he most important parts of the package.

  XAMPP also provides support for creating and manipulating databases in [MariaDB](https://en.wikipedia.org/wiki/MariaDB) and [SQLite a](https://en.wikipedia.org/wiki/SQLite)mong others. Once XAMPP is installed, it is possible to treat a [localhost l](https://en.wikipedia.org/wiki/Localhost)ike a remote host by connecting using an [FTP c](https://en.wikipedia.org/wiki/File_Transfer_Protocol)lient. Using a program like [FileZilla h](https://en.wikipedia.org/wiki/FileZilla)as many advantages when installing a [content management system (](https://en.wikipedia.org/wiki/Content_management_system)CMS) like [Joomla o](https://en.wikipedia.org/wiki/Joomla)r [WordPress.](https://en.wikipedia.org/wiki/WordPress) It is also possible to connect to localhost via FTP with an [HTML editor.](https://en.wikipedia.org/wiki/HTML_editor)

**3.3 DATABASE**

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modelling techniques.

The database management system (DBMS) is the software that interacts with end users, applications, and the database itself to capture and analyse the data. The DBMS software additionally encompasses the core facilities provided to administer the database. The sum total of the database, the DBMS and the associated applications can be referred to as a "database system". Often the term "database" is also used to loosely refer to any of the DBMS, the database system or an application associated with the database.

Computer scientists may classify database-management systems according to the database models that they support. Relational databases became dominant in the 1980s. These model data as rows and columns in a series of tables, and the vast majority use SQL for writing and querying data. In the 2000s, non-relational databases became popular, referred to as NoSQL because they use different query languages.

Formally, a "database" refers to a set of related data and the way it is organized. Access to this data is usually provided by a "database management system" (DBMS) consisting of an integrated set of computer software that allows users to interact with one or more databases and provides access to all of the data contained in the database (although restrictions may exist that limit access to particular data). The DBMS provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organized.

Because of the close relationship between them, the term "database" is often used casually to refer to both a database and the DBMS used to manipulate it.

Outside the world of professional information technology, the term database is often used to refer to any collection of related data (such as a spreadsheet or a card index) as size and usage requirements typically necessitate use of a database management system.

**3.4 SQL**

SQL (pronounced "ess-que-el") stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database. This tutorial will provide you with the instruction on the basics of each of these commands as well as allow you to put them to practice using the SQL Interpreter.

SQL is case insensitive. But it is a recommended practice to use keywords (like SELECT, UPDATE, CREATE, etc.) in capital letters and use user defined things (liked table name, column name, etc.) in small letters.

We can write comments in SQL using — (double hyphen) at the beginning of any line. SQL is the programming language for relational databases (explained below) like MySQL, Oracle, Sybase, SQL Server, etc. Other non-relational databases (also called NoSQL) databases like MongoDB, Dynamo DB, etc. do not use SQL

Although there is an ISO standard for SQL, most of the implementations slightly vary in syntax. So we may encounter queries that work in SQL Server but do not work in MySQL.

The Most Important SQL Statements:

SELECT - extracts data from a database

UPDATE - updates data in a database

DELETE - deletes data from a database

INSERT INTO - inserts new data into a database

CREATE DATABASE - creates a new database

ALTER DATABASE - modifies a database

CREATE TABLE - creates a new table

ALTER TABLE - modifies a table

DROP TABLE - deletes a table

CREATE INDEX - creates an index (search key)

DROP INDEX - deletes an index

SQL keywords are NOT case sensitive: select is the same as SELECT**3.5 QUERIES**

The most common operation in SQL, the query, makes use of the declarative [SELECT](https://en.wikipedia.org/wiki/Select_%28SQL%29) statement. SELECT retrieves data from one or more [tables,](https://en.wikipedia.org/wiki/Table_%28database%29) 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)](https://en.wikipedia.org/wiki/Database_management_system) to carry out [planning,](https://en.wikipedia.org/wiki/Query_plan) [optimizing,](https://en.wikipedia.org/wiki/Query_optimizer) 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 c](https://en.wikipedia.org/wiki/From_%28SQL%29)lause, which indicates the table(s) to retrieve data from. The FROM clause can include optional [JOIN s](https://en.wikipedia.org/wiki/Join_%28SQL%29)ubclauses to specify the rules for joining tables.
* The [WHERE c](https://en.wikipedia.org/wiki/Where_%28SQL%29)lause 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 c](https://en.wikipedia.org/wiki/Having_%28SQL%29)lause 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 c](https://en.wikipedia.org/wiki/Order_by_%28SQL%29)lause 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**

**SOFTWARE REQUIREMENTS**

Software and hardware requirements are very essential 50% of work is done only

by choosing the efficient and essential software . So choosing the software makes

your work more easier and simpler. When it comes to hardware requirements we

need to have the fastest and need to have the disk with high space.

**4.1 HARDWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| Processor | Intel CORE i5 |
| RAM | 8.0 GB |
| Hard Disk Drive | 1 TB |

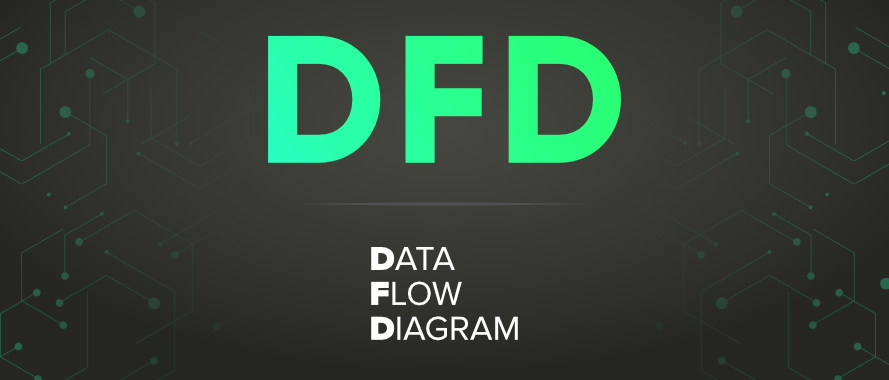
**4.2 SOFTWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| **NUMBER** | **Description** |
| 1 | Windows 10 |
| 2 | HTML, CSS, JS, BOOTSTRAP. |
| 3 | MYSQL, NODEJS |
| 4 | Compiler: (visual studio code) |

**CHAPTER-5**

**DATA FLOW DIAGRAMS**

A **data-flow diagram** is a way of representing a flow of data through a [process](https://en.wikipedia.org/wiki/Process) or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.



**Fig-5.1**

There are four symbols and components for drawing DFDs they are mainly

Rectangle Ellipse arrows open-ended rectangle

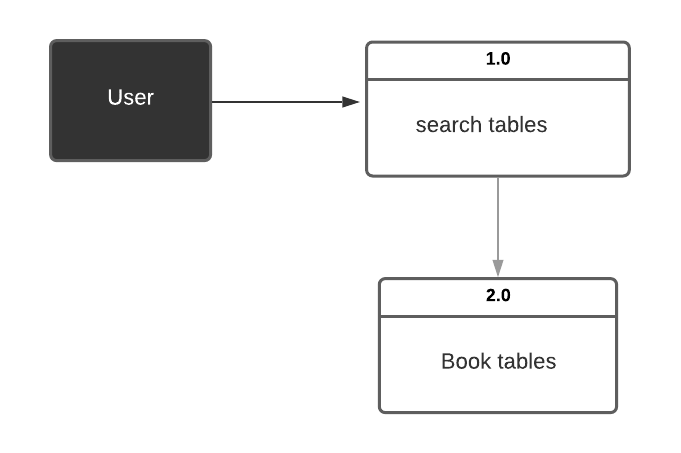
Process Warehouse Dataflow Terminator

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. Figures below are the Data Flow Diagrams for the current system.

CONTEXT LEVEL DIAGRAM

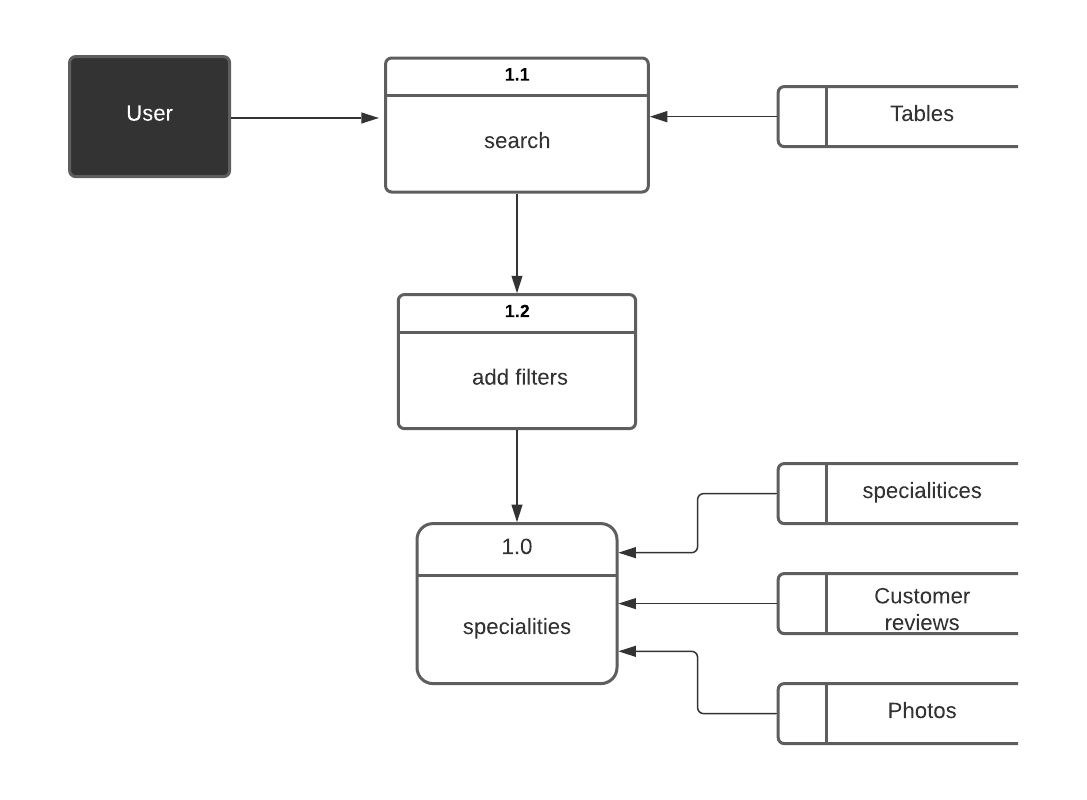
** fig-5.2**

FIRST LEVEL DFD

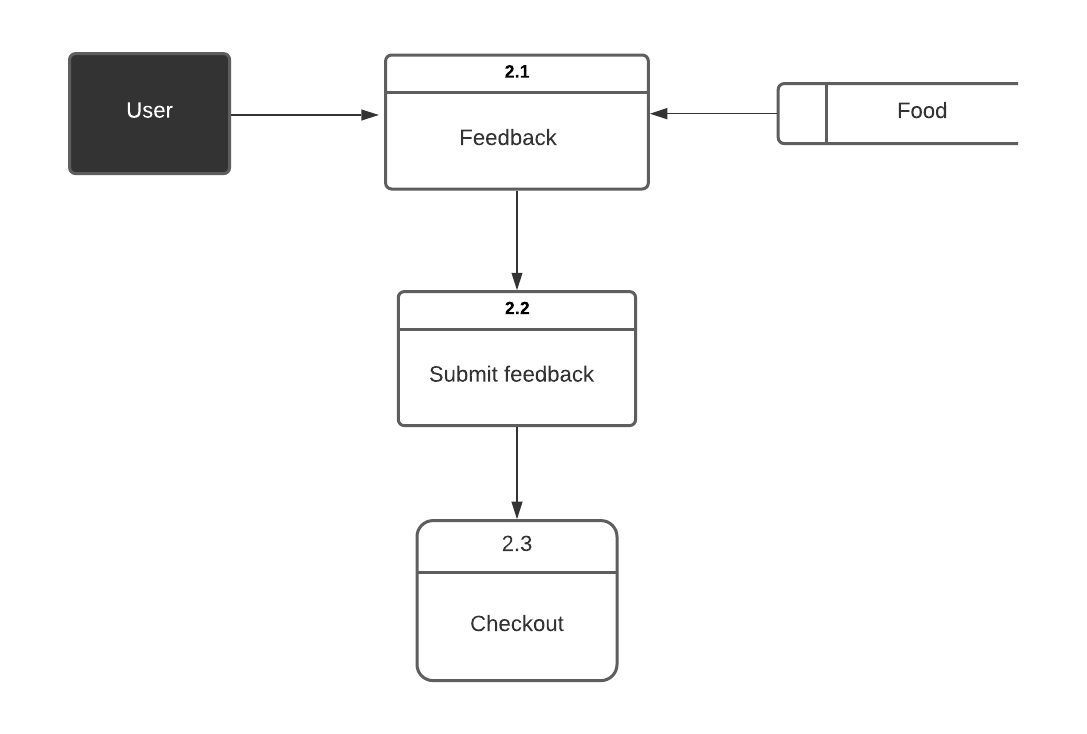


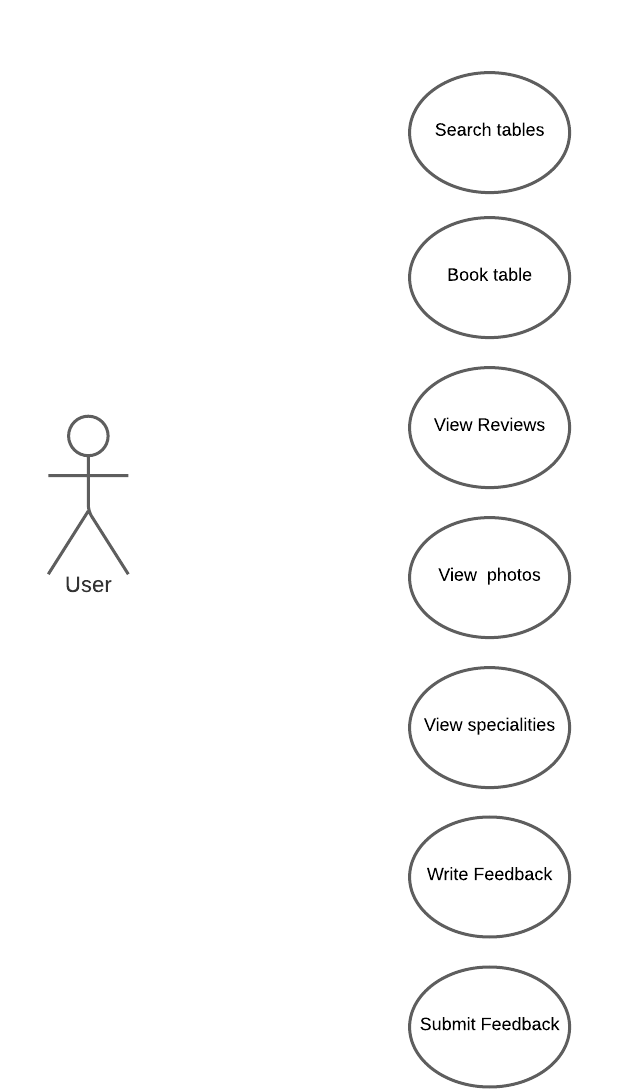
**Fig 5.3**

SECOND LEVEL DFD



**Fig 5.4**

 **Fig 5.5**



USECASE DIAGRAM (**fig-5.6)**

**CHAPTER 6**

**PROJECT**

NAME : RESTAURANT MANAGEMENT SYSTEM

USED TECHNOLOGIES:

* JAVASACRIPT
* MYSQL

TECHNICAL DETAILS :

* We used different technologies for designing our project i.e., both frontend and backend.
* To design frontend i.e., UI design we used html, css, bootstrap ,jquery.
* To design the backend of our project we used a database named mysql database and also we used node js to write the code for designing the backend .

**TEAM DETAILS:(size-3)**

|  |  |  |
| --- | --- | --- |
| NAME | ROLL-NO | WORKDONE |
| M. Laxmareddy | 221710308034 | Backend(Nodejs),wireframes |
| Sri sai Sanjana. Mettu | 221710308054 | Frontend (html , css), group documentation |
| K. Pranavi | 221710308028 | Frontend(Js,booking.html),backend(mysql tables), uml diagram. |

Database: MySql

Operating System: Windows7/8/8.1/10

Wire framing tool: ([wireframepro.mockflow.com/](https://wireframepro.mockflow.com/))

**CHAPTER-7**

**MODULES ASSIGNED**

**7.1 JAVASCRIPT:**

**JavaScript** often abbreviated as **JS**, is a [programming language](https://en.wikipedia.org/wiki/Programming_language) that conforms to the [ECMAScript](https://en.wikipedia.org/wiki/ECMAScript) specification. JavaScript is [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), often [just-in-time compiled](https://en.wikipedia.org/wiki/Just-in-time_compilation), and [multi-paradigm](https://en.wikipedia.org/wiki/Programming_paradigm). It has [curly-bracket syntax](https://en.wikipedia.org/wiki/List_of_programming_languages_by_type#Curly-bracket_languages), [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_typing), [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) [object-orientation](https://en.wikipedia.org/wiki/Object-oriented_programming), and [first-class functions](https://en.wikipedia.org/wiki/First-class_function).Alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/CSS), JavaScript is one of the core technologies of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web). JavaScript enables interactive [web pages](https://en.wikipedia.org/wiki/Web_page) and is an essential part of [web applications](https://en.wikipedia.org/wiki/Web_application). The vast majority of [websites](https://en.wikipedia.org/wiki/Website) use it for [client-side](https://en.wikipedia.org/wiki/Client-side) page behavior, and all major [web browsers](https://en.wikipedia.org/wiki/Web_browser) have a dedicated [JavaScript engine](https://en.wikipedia.org/wiki/JavaScript_engine) to execute it.

**7.2 BOOKING PAGE:**

This page generally contains the table booking section through which user can book a table. This page concentrate on the user details, this page contains a form through which user can book a table. Table booking related information is collected here such as name and email id,no.of tables then user can book the table .so that he can visit us with his mates to our restaurant.

**7.3 MYSQL:**

**MySQL**  is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). Its name is a combination of "My", the name of co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)'s daughter, and "[SQL](https://en.wikipedia.org/wiki/SQL)", the abbreviation for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language). A [relational database](https://en.wikipedia.org/wiki/Relational_database) organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an [operating system](https://en.wikipedia.org/wiki/Operating_system) to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

**CHAPTER 8**

**Codes and Output Screenshots**

Booking.html:-

<!DOCTYPE html>

<html lang="en">

<head>

    <title>project </title>

    <link rel="stylesheet" href="style.css">

    <link rel="stylesheet"  href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" ></script>

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

</head>

<body>

    <section id="nav-bar">

        <nav class="navbar navbar-expand-lg navbar-light">

            <a class="navbar-brand" href="#"><img src="logo1.jpg"></a>

            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

                <i class="fa fa-bars" aria-hidden="true"></i>

            </button>

            <div class="collapse navbar-collapse" id="navbarNav">

              <ul class="navbar-nav ml-auto">

                <li class="nav-item active">

                  <a class="nav-link" href="index.html">Home </a>

                </li>

                  <li class="nav-item">

                    <a class="nav-link " href="#booking" >Book your Table</a>

                  </li>

                <li class="nav-item">

                  <a class="nav-link" href="#footer">Contact</a>

                </li>

              </ul>

            </div>

          </nav>

    </section>

    <!-------------------banner section------------------>

      <!---booking--->

      <section id="booking" style="padding-top: 50px;">

        <div class="container text-center">

          <h1 class="title">Book Your Table</h1>

          <div class="feedback-form">

            <form action="" method="">

                <h2 class="text-center">Reserve Your Table</h2>

                <div class="form-group">

                    <input type="text" id="txtname" class="form-control" name="name" placeholder="Name" required="required">

                </div>

                <div class="form-group">

                    <input type="email" id="txtemail" class="form-control" placeholder="Email" required="required">

                </div>

                <div class="form-group">

                    <input type="number" id="number" class="form-control" placeholder="No of Tables" required="required">

                </div>

                <p>4 people are allowed in each table.</p>

                <div class="form-group">

                    <label class="checkbox-inline">

                        <input type="checkbox" required="required"> I accept the <a href="">Terms</a> &amp;

                        <a href="">Privacy Policy</a>

                    </label>

                </div>

                <div class="form-group">

                    <button type="submit" id="btnSubmit" class="btn btn-success btn-block" onclick="alert('your tables booked')">Book Now</button>

                </div>

            </form>

            </div>

        </div>

    </section>

      <!--social media--->

      <section id="social-media">

        <div class="container text-center">

          <p>FIND US ON SOCIAL MEDIA</p>

          <div class="social-icons">

            <a href="#"><img src="social.jpg"></a>

          </div>

        </div>

      </section>

      <!-----footer------>

      <section id="footer">

        <div class="container">

          <div class="row">

            <div class="col-md-4 fotter-box">

             <img src="logo1.jpg">

             <p>Tast the best food and drinks here.</p>

            </div>

            <div class="col-md-4 footer-box">

              <p><b>CONTACT US</b></p>

              <p><i class="fa fa-map-marker"></i>Foodies resturent in,Hyderabad</p>

              <p><i class="fa fa-phone"></i>+10738378717</p>

              <p><i class="fa fa-envelope"></i>foodie139@gmail.xom</p>

            </div>

          </div>

          <hr>

          <p class="copyright">website is designed by BE LIKE BRO</p>

        </div>

      </section>

      <script src="smooth-scroll.js"></script>

      <script>

       var scroll = new SmoothScroll('a[href\*="#"]');

     </script>

    <style type="text/css">

        .feedback-form {

            width: 340px;

            margin: 50px auto;

        }

        .feedback-form form {

            margin-bottom: 15px;

            background: #f7f7f7;

            box-shadow: 0px 2px 2px rgba(0, 0, 0, 0.3);

            padding:30px;

        }

        .feedback-form h2 {

            margin: 0 0 15px;

        }

        .form-control, .btn {

            min-height: 38px;

            border-radius: 2px;

        }

        .btn {

            font-size: 15px;

            font-weight: bold;

        }

        .a1{

            height: 100px;

            text-align: center;

         }

         .a2{

             height: 45px;

         }

    </style>

        <script src="js/jquery.min.js"></script>

        <script type="text/javascript" accesskey="">

            $(document).ready(function(){

            $("#btnSubmit").click(function(){

              debugger;

                        var api\_url = "http://localhost:3000/bookings";

                        var data = {

                            name : $('#txtname').val(),

                            email : $('#txtemail').val(),

                            number : $('#number').val(),

                        }

                        $.ajax({

                            url : api\_url,

                            type:"POST",

                            dataType:"json",

                            data:data,

                            success:function(data){

                                alert('values are submitted');

                                window.open("index.html");

                                return;

                            },

                            error:function(){

                                alert('wrong with insert');

                            }

                        });

                    });

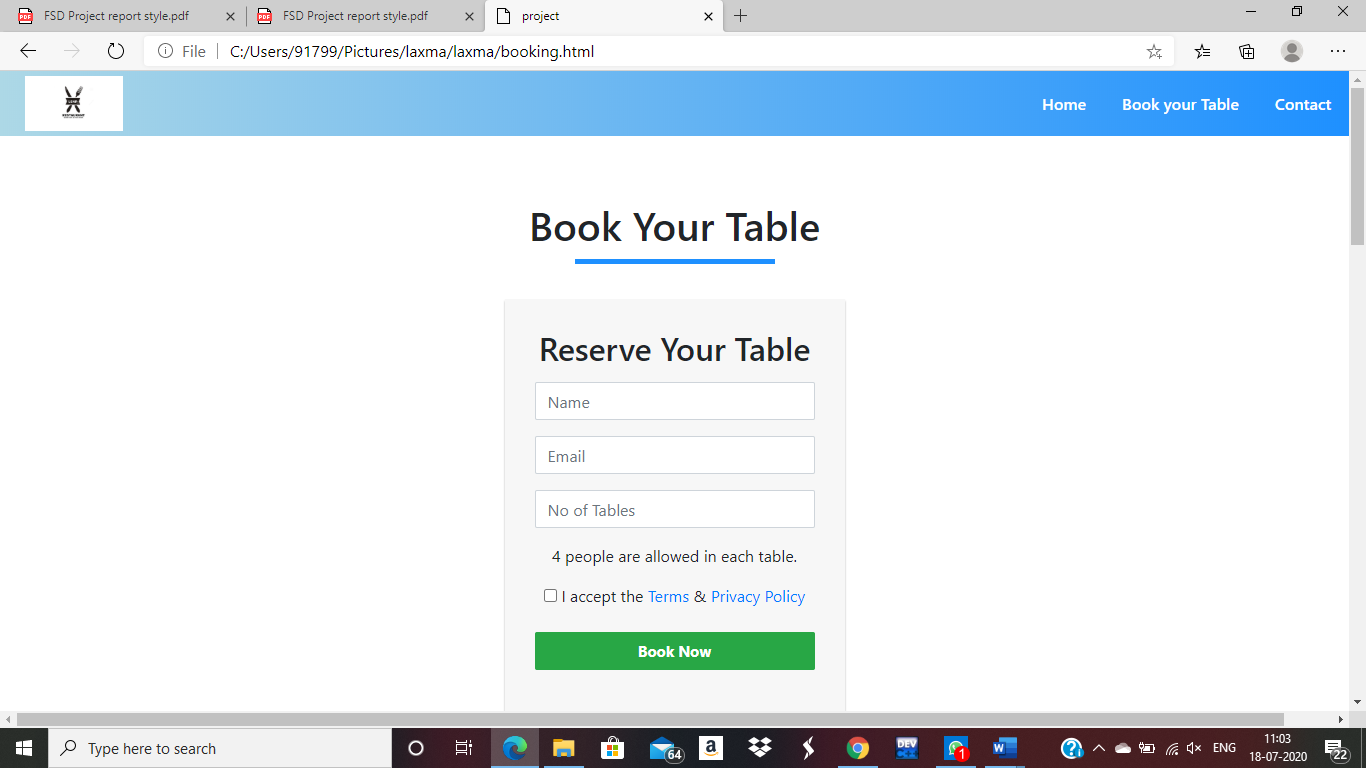
                  });

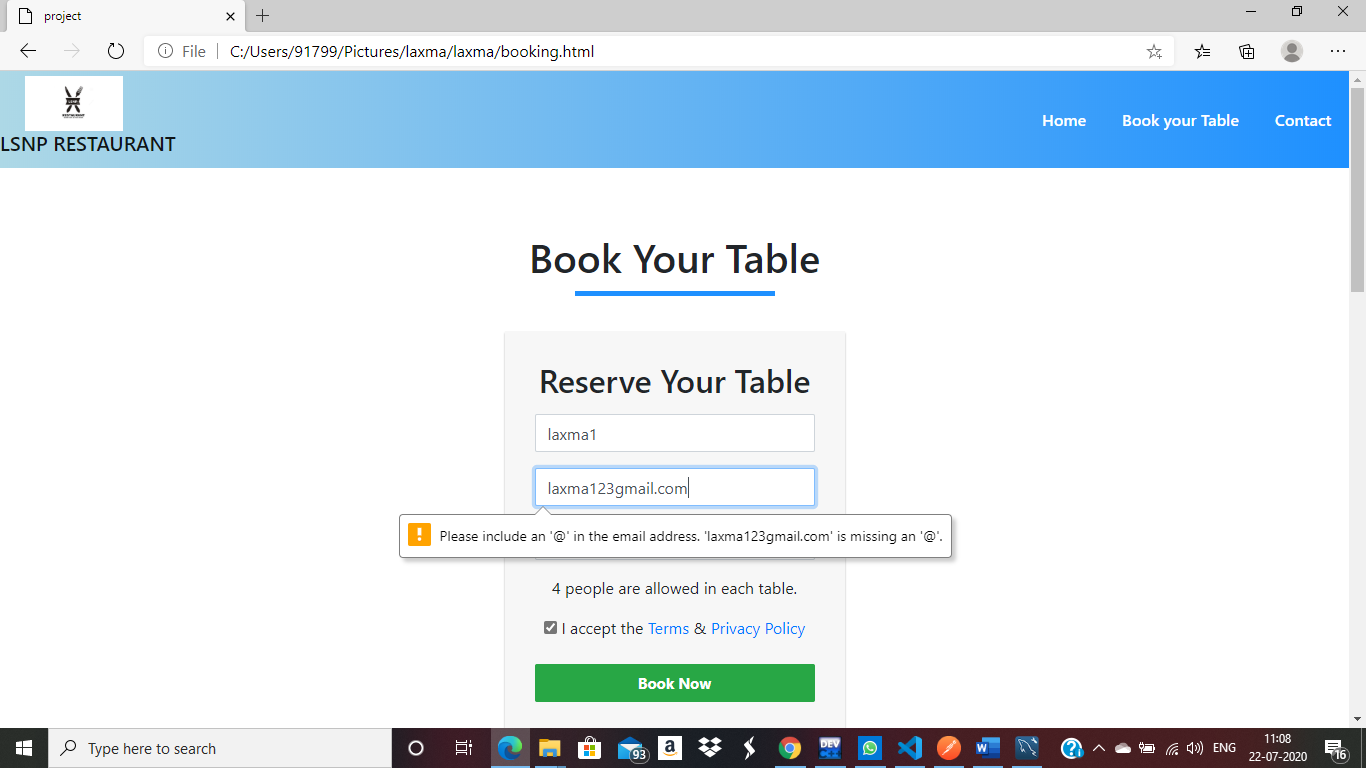
        </script>

 </body>

 </html>

Output:





Smoothscroll.js

/\*! SmoothScroll v16.1.4 | (c) 2020 Chris Ferdinandi | MIT License | http://github.com/cferdinandi/smooth-scroll \*/

(function (global, factory) {

    typeof exports === 'object' && typeof module !== 'undefined' ? module.exports = factory() :

    typeof define === 'function' && define.amd ? define(factory) :

    (global = global || self, global.SmoothScroll = factory());

}(this, (function () { 'use strict';

    //

    // Default settings

    //

    var defaults = {

        // Selectors

        ignore: '[data-scroll-ignore]',

        header: null,

        topOnEmptyHash: true,

        // Speed & Duration

        speed: 500,

        speedAsDuration: false,

        durationMax: null,

        durationMin: null,

        clip: true,

        offset: 0,

        // Easing

        easing: 'easeInOutCubic',

        customEasing: null,

        // History

        updateURL: true,

        popstate: true,

        // Custom Events

        emitEvents: true

    };

    //

    // Utility Methods

    //

    /\*\*

     \* Check if browser supports required methods

     \* @return {Boolean} Returns true if all required methods are supported

     \*/

    var supports = function () {

        return (

            'querySelector' in document &&

            'addEventListener' in window &&

            'requestAnimationFrame' in window &&

            'closest' in window.Element.prototype

        );

    };

    /\*\*

     \* Merge two or more objects together.

     \* @param   {Object}   objects  The objects to merge together

     \* @returns {Object}            Merged values of defaults and options

     \*/

    var extend = function () {

        var merged = {};

        Array.prototype.forEach.call(arguments, function (obj) {

            for (var key in obj) {

                if (!obj.hasOwnProperty(key)) return;

                merged[key] = obj[key];

            }

        });

        return merged;

    };

    /\*\*

     \* Check to see if user prefers reduced motion

     \* @param  {Object} settings Script settings

     \*/

    var reduceMotion = function () {

        if ('matchMedia' in window && window.matchMedia('(prefers-reduced-motion)').matches) {

            return true;

        }

        return false;

    };

    /\*\*

     \* Get the height of an element.

     \* @param  {Node} elem The element to get the height of

     \* @return {Number}    The element's height in pixels

     \*/

    var getHeight = function (elem) {

        return parseInt(window.getComputedStyle(elem).height, 10);

    };

    /\*\*

     \* Escape special characters for use with querySelector

     \* @author Mathias Bynens

     \* @link https://github.com/mathiasbynens/CSS.escape

     \* @param {String} id The anchor ID to escape

     \*/

    var escapeCharacters = function (id) {

        // Remove leading hash

        if (id.charAt(0) === '#') {

            id = id.substr(1);

        }

        var string = String(id);

        var length = string.length;

        var index = -1;

        var codeUnit;

        var result = '';

        var firstCodeUnit = string.charCodeAt(0);

        while (++index < length) {

            codeUnit = string.charCodeAt(index);

            // Note: there’s no need to special-case astral symbols, surrogate

            // pairs, or lone surrogates.

            // If the character is NULL (U+0000), then throw an

            // `InvalidCharacterError` exception and terminate these steps.

            if (codeUnit === 0x0000) {

                throw new InvalidCharacterError(

                    'Invalid character: the input contains U+0000.'

                );

            }

            if (

                // If the character is in the range [\1-\1F] (U+0001 to U+001F) or is

                // U+007F, […]

                (codeUnit >= 0x0001 && codeUnit <= 0x001F) || codeUnit == 0x007F ||

                // If the character is the first character and is in the range [0-9]

                // (U+0030 to U+0039), […]

                (index === 0 && codeUnit >= 0x0030 && codeUnit <= 0x0039) ||

                // If the character is the second character and is in the range [0-9]

                // (U+0030 to U+0039) and the first character is a `-` (U+002D), […]

                (

                    index === 1 &&

                    codeUnit >= 0x0030 && codeUnit <= 0x0039 &&

                    firstCodeUnit === 0x002D

                )

            ) {

                // http://dev.w3.org/csswg/cssom/#escape-a-character-as-code-point

                result += '\\' + codeUnit.toString(16) + ' ';

                continue;

            }

            // If the character is not handled by one of the above rules and is

            // greater than or equal to U+0080, is `-` (U+002D) or `\_` (U+005F), or

            // is in one of the ranges [0-9] (U+0030 to U+0039), [A-Z] (U+0041 to

            // U+005A), or [a-z] (U+0061 to U+007A), […]

            if (

                codeUnit >= 0x0080 ||

                codeUnit === 0x002D ||

                codeUnit === 0x005F ||

                codeUnit >= 0x0030 && codeUnit <= 0x0039 ||

                codeUnit >= 0x0041 && codeUnit <= 0x005A ||

                codeUnit >= 0x0061 && codeUnit <= 0x007A

            ) {

                // the character itself

                result += string.charAt(index);

                continue;

            }

            // Otherwise, the escaped character.

            // http://dev.w3.org/csswg/cssom/#escape-a-character

            result += '\\' + string.charAt(index);

        }

        // Return sanitized hash

        return '#' + result;

    };

    /\*\*

     \* Calculate the easing pattern

     \* @link https://gist.github.com/gre/1650294

     \* @param   {Object} settings Easing pattern

     \* @param   {Number} time     Time animation should take to complete

     \* @returns {Number}

     \*/

    var easingPattern = function (settings, time) {

        var pattern;

        // Default Easing Patterns

        if (settings.easing === 'easeInQuad') pattern = time \* time; // accelerating from zero velocity

        if (settings.easing === 'easeOutQuad') pattern = time \* (2 - time); // decelerating to zero velocity

        if (settings.easing === 'easeInOutQuad') pattern = time < 0.5 ? 2 \* time \* time : -1 + (4 - 2 \* time) \* time; // acceleration until halfway, then deceleration

        if (settings.easing === 'easeInCubic') pattern = time \* time \* time; // accelerating from zero velocity

        if (settings.easing === 'easeOutCubic') pattern = (--time) \* time \* time + 1; // decelerating to zero velocity

        if (settings.easing === 'easeInOutCubic') pattern = time < 0.5 ? 4 \* time \* time \* time : (time - 1) \* (2 \* time - 2) \* (2 \* time - 2) + 1; // acceleration until halfway, then deceleration

        if (settings.easing === 'easeInQuart') pattern = time \* time \* time \* time; // accelerating from zero velocity

        if (settings.easing === 'easeOutQuart') pattern = 1 - (--time) \* time \* time \* time; // decelerating to zero velocity

        if (settings.easing === 'easeInOutQuart') pattern = time < 0.5 ? 8 \* time \* time \* time \* time : 1 - 8 \* (--time) \* time \* time \* time; // acceleration until halfway, then deceleration

        if (settings.easing === 'easeInQuint') pattern = time \* time \* time \* time \* time; // accelerating from zero velocity

        if (settings.easing === 'easeOutQuint') pattern = 1 + (--time) \* time \* time \* time \* time; // decelerating to zero velocity

        if (settings.easing === 'easeInOutQuint') pattern = time < 0.5 ? 16 \* time \* time \* time \* time \* time : 1 + 16 \* (--time) \* time \* time \* time \* time; // acceleration until halfway, then deceleration

        // Custom Easing Patterns

        if (!!settings.customEasing) pattern = settings.customEasing(time);

        return pattern || time; // no easing, no acceleration

    };

    /\*\*

     \* Determine the document's height

     \* @returns {Number}

     \*/

    var getDocumentHeight = function () {

        return Math.max(

            document.body.scrollHeight, document.documentElement.scrollHeight,

            document.body.offsetHeight, document.documentElement.offsetHeight,

            document.body.clientHeight, document.documentElement.clientHeight

        );

    };

    /\*\*

     \* Calculate how far to scroll

     \* Clip support added by robjtede - https://github.com/cferdinandi/smooth-scroll/issues/405

     \* @param {Element} anchor       The anchor element to scroll to

     \* @param {Number}  headerHeight Height of a fixed header, if any

     \* @param {Number}  offset       Number of pixels by which to offset scroll

     \* @param {Boolean} clip         If true, adjust scroll distance to prevent abrupt stops near the bottom of the page

     \* @returns {Number}

     \*/

    var getEndLocation = function (anchor, headerHeight, offset, clip) {

        var location = 0;

        if (anchor.offsetParent) {

            do {

                location += anchor.offsetTop;

                anchor = anchor.offsetParent;

            } while (anchor);

        }

        location = Math.max(location - headerHeight - offset, 0);

        if (clip) {

            location = Math.min(location, getDocumentHeight() - window.innerHeight);

        }

            return location;

    };

    /\*\*

     \* Get the height of the fixed header

     \* @param  {Node}   header The header

     \* @return {Number}        The height of the header

     \*/

    var getHeaderHeight = function (header) {

        return !header ? 0 : (getHeight(header) + header.offsetTop);

    };

    /\*\*

     \* Calculate the speed to use for the animation

     \* @param  {Number} distance The distance to travel

     \* @param  {Object} settings The plugin settings

     \* @return {Number}          How fast to animate

     \*/

    var getSpeed = function (distance, settings) {

        var speed = settings.speedAsDuration ? settings.speed : Math.abs(distance / 1000 \* settings.speed);

        if (settings.durationMax && speed > settings.durationMax) return settings.durationMax;

        if (settings.durationMin && speed < settings.durationMin) return settings.durationMin;

        return parseInt(speed, 10);

    };

    var setHistory = function (options) {

        // Make sure this should run

        if (!history.replaceState || !options.updateURL || history.state) return;

        // Get the hash to use

        var hash = window.location.hash;

        hash = hash ? hash : '';

        // Set a default history

        history.replaceState(

            {

                smoothScroll: JSON.stringify(options),

                anchor: hash ? hash : window.pageYOffset

            },

            document.title,

            hash ? hash : window.location.href

        );

    };

    /\*\*

     \* Update the URL

     \* @param  {Node}    anchor  The anchor that was scrolled to

     \* @param  {Boolean} isNum   If true, anchor is a number

     \* @param  {Object}  options Settings for Smooth Scroll

     \*/

    var updateURL = function (anchor, isNum, options) {

        // Bail if the anchor is a number

        if (isNum) return;

        // Verify that pushState is supported and the updateURL option is enabled

        if (!history.pushState || !options.updateURL) return;

        // Update URL

        history.pushState(

            {

                smoothScroll: JSON.stringify(options),

                anchor: anchor.id

            },

            document.title,

            anchor === document.documentElement ? '#top' : '#' + anchor.id

        );

    };

    /\*\*

     \* Bring the anchored element into focus

     \* @param {Node}     anchor      The anchor element

     \* @param {Number}   endLocation The end location to scroll to

     \* @param {Boolean}  isNum       If true, scroll is to a position rather than an element

     \*/

    var adjustFocus = function (anchor, endLocation, isNum) {

        // Is scrolling to top of page, blur

        if (anchor === 0) {

            document.body.focus();

        }

        // Don't run if scrolling to a number on the page

        if (isNum) return;

        // Otherwise, bring anchor element into focus

        anchor.focus();

        if (document.activeElement !== anchor) {

            anchor.setAttribute('tabindex', '-1');

            anchor.focus();

            anchor.style.outline = 'none';

        }

        window.scrollTo(0 , endLocation);

    };

    /\*\*

     \* Emit a custom event

     \* @param  {String} type    The event type

     \* @param  {Object} options The settings object

     \* @param  {Node}   anchor  The anchor element

     \* @param  {Node}   toggle  The toggle element

     \*/

    var emitEvent = function (type, options, anchor, toggle) {

        if (!options.emitEvents || typeof window.CustomEvent !== 'function') return;

        var event = new CustomEvent(type, {

            bubbles: true,

            detail: {

                anchor: anchor,

                toggle: toggle

            }

        });

        document.dispatchEvent(event);

    };

    //

    // SmoothScroll Constructor

    //

    var SmoothScroll = function (selector, options) {

        //

        // Variables

        //

        var smoothScroll = {}; // Object for public APIs

        var settings, toggle, fixedHeader, animationInterval;

        //

        // Methods

        //

        /\*\*

         \* Cancel a scroll-in-progress

         \*/

        smoothScroll.cancelScroll = function (noEvent) {

            cancelAnimationFrame(animationInterval);

            animationInterval = null;

            if (noEvent) return;

            emitEvent('scrollCancel', settings);

        };

        /\*\*

         \* Start/stop the scrolling animation

         \* @param {Node|Number} anchor  The element or position to scroll to

         \* @param {Element}     toggle  The element that toggled the scroll event

         \* @param {Object}      options

         \*/

        smoothScroll.animateScroll = function (anchor, toggle, options) {

            // Cancel any in progress scrolls

            smoothScroll.cancelScroll();

            // Local settings

            var \_settings = extend(settings || defaults, options || {}); // Merge user options with defaults

            // Selectors and variables

            var isNum = Object.prototype.toString.call(anchor) === '[object Number]' ? true : false;

            var anchorElem = isNum || !anchor.tagName ? null : anchor;

            if (!isNum && !anchorElem) return;

            var startLocation = window.pageYOffset; // Current location on the page

            if (\_settings.header && !fixedHeader) {

                // Get the fixed header if not already set

                fixedHeader = document.querySelector(\_settings.header);

            }

            var headerHeight = getHeaderHeight(fixedHeader);

            var endLocation = isNum ? anchor : getEndLocation(anchorElem, headerHeight, parseInt((typeof \_settings.offset === 'function' ? \_settings.offset(anchor, toggle) : \_settings.offset), 10), \_settings.clip); // Location to scroll to

            var distance = endLocation - startLocation; // distance to travel

            var documentHeight = getDocumentHeight();

            var timeLapsed = 0;

            var speed = getSpeed(distance, \_settings);

            var start, percentage, position;

            /\*\*

             \* Stop the scroll animation when it reaches its target (or the bottom/top of page)

             \* @param {Number} position Current position on the page

             \* @param {Number} endLocation Scroll to location

             \* @param {Number} animationInterval How much to scroll on this loop

             \*/

            var stopAnimateScroll = function (position, endLocation) {

                // Get the current location

                var currentLocation = window.pageYOffset;

                // Check if the end location has been reached yet (or we've hit the end of the document)

                if (position == endLocation || currentLocation == endLocation || ((startLocation < endLocation && window.innerHeight + currentLocation) >= documentHeight)) {

                    // Clear the animation timer

                    smoothScroll.cancelScroll(true);

                    // Bring the anchored element into focus

                    adjustFocus(anchor, endLocation, isNum);

                    // Emit a custom event

                    emitEvent('scrollStop', \_settings, anchor, toggle);

                    // Reset start

                    start = null;

                    animationInterval = null;

                    return true;

                }

            };

            /\*\*

             \* Loop scrolling animation

             \*/

            var loopAnimateScroll = function (timestamp) {

                if (!start) { start = timestamp; }

                timeLapsed += timestamp - start;

                percentage = speed === 0 ? 0 : (timeLapsed / speed);

                percentage = (percentage > 1) ? 1 : percentage;

                position = startLocation + (distance \* easingPattern(\_settings, percentage));

                window.scrollTo(0, Math.floor(position));

                if (!stopAnimateScroll(position, endLocation)) {

                    animationInterval = window.requestAnimationFrame(loopAnimateScroll);

                    start = timestamp;

                }

            };

            /\*\*

             \* Reset position to fix weird iOS bug

             \* @link https://github.com/cferdinandi/smooth-scroll/issues/45

             \*/

            if (window.pageYOffset === 0) {

                window.scrollTo(0, 0);

            }

            // Update the URL

            updateURL(anchor, isNum, \_settings);

            // If the user prefers reduced motion, jump to location

            if (reduceMotion()) {

                adjustFocus(anchor, Math.floor(endLocation), false);

                return;

            }

            // Emit a custom event

            emitEvent('scrollStart', \_settings, anchor, toggle);

            // Start scrolling animation

            smoothScroll.cancelScroll(true);

            window.requestAnimationFrame(loopAnimateScroll);

        };

        /\*\*

         \* If smooth scroll element clicked, animate scroll

         \*/

        var clickHandler = function (event) {

            // Don't run if event was canceled but still bubbled up

            // By @mgreter - https://github.com/cferdinandi/smooth-scroll/pull/462/

            if (event.defaultPrevented) return;

            // Don't run if right-click or command/control + click or shift + click

            if (event.button !== 0 || event.metaKey || event.ctrlKey || event.shiftKey) return;

            // Check if event.target has closest() method

            // By @totegi - https://github.com/cferdinandi/smooth-scroll/pull/401/

            if (!('closest' in event.target)) return;

            // Check if a smooth scroll link was clicked

            toggle = event.target.closest(selector);

            if (!toggle || toggle.tagName.toLowerCase() !== 'a' || event.target.closest(settings.ignore)) return;

            // Only run if link is an anchor and points to the current page

            if (toggle.hostname !== window.location.hostname || toggle.pathname !== window.location.pathname || !/#/.test(toggle.href)) return;

            // Get an escaped version of the hash

            var hash;

            try {

                hash = escapeCharacters(decodeURIComponent(toggle.hash));

            } catch(e) {

                hash = escapeCharacters(toggle.hash);

            }

            // Get the anchored element

            var anchor;

            if (hash === '#') {

                if (!settings.topOnEmptyHash) return;

                anchor = document.documentElement;

            } else {

                anchor = document.querySelector(hash);

            }

            anchor = !anchor && hash === '#top' ? document.documentElement : anchor;

            // If anchored element exists, scroll to it

            if (!anchor) return;

            event.preventDefault();

            setHistory(settings);

            smoothScroll.animateScroll(anchor, toggle);

        };

        /\*\*

         \* Animate scroll on popstate events

         \*/

        var popstateHandler = function () {

            // Stop if history.state doesn't exist (ex. if clicking on a broken anchor link).

            // fixes `Cannot read property 'smoothScroll' of null` error getting thrown.

            if (history.state === null) return;

            // Only run if state is a popstate record for this instantiation

            if (!history.state.smoothScroll || history.state.smoothScroll !== JSON.stringify(settings)) return;

            // Get the anchor

            var anchor = history.state.anchor;

            if (typeof anchor === 'string' && anchor) {

                anchor = document.querySelector(escapeCharacters(history.state.anchor));

                if (!anchor) return;

            }

            // Animate scroll to anchor link

            smoothScroll.animateScroll(anchor, null, {updateURL: false});

        };

        /\*\*

         \* Destroy the current initialization.

         \*/

        smoothScroll.destroy = function () {

            // If plugin isn't already initialized, stop

            if (!settings) return;

            // Remove event listeners

            document.removeEventListener('click', clickHandler, false);

            window.removeEventListener('popstate', popstateHandler, false);

            // Cancel any scrolls-in-progress

            smoothScroll.cancelScroll();

            // Reset variables

            settings = null;

            toggle = null;

            fixedHeader = null;

            animationInterval = null;

        };

        /\*\*

         \* Initialize Smooth Scroll

         \* @param {Object} options User settings

         \*/

        var init = function () {

            // feature test

            if (!supports()) throw 'Smooth Scroll: This browser does not support the required JavaScript methods and browser APIs.';

            // Destroy any existing initializations

            smoothScroll.destroy();

            // Selectors and variables

            settings = extend(defaults, options || {}); // Merge user options with defaults

            fixedHeader = settings.header ? document.querySelector(settings.header) : null; // Get the fixed header

            // When a toggle is clicked, run the click handler

            document.addEventListener('click', clickHandler, false);

            // If updateURL and popState are enabled, listen for pop events

            if (settings.updateURL && settings.popstate) {

                window.addEventListener('popstate', popstateHandler, false);

            }

        };

        //

        // Initialize plugin

        //

        init();

        //

        // Public APIs

        //

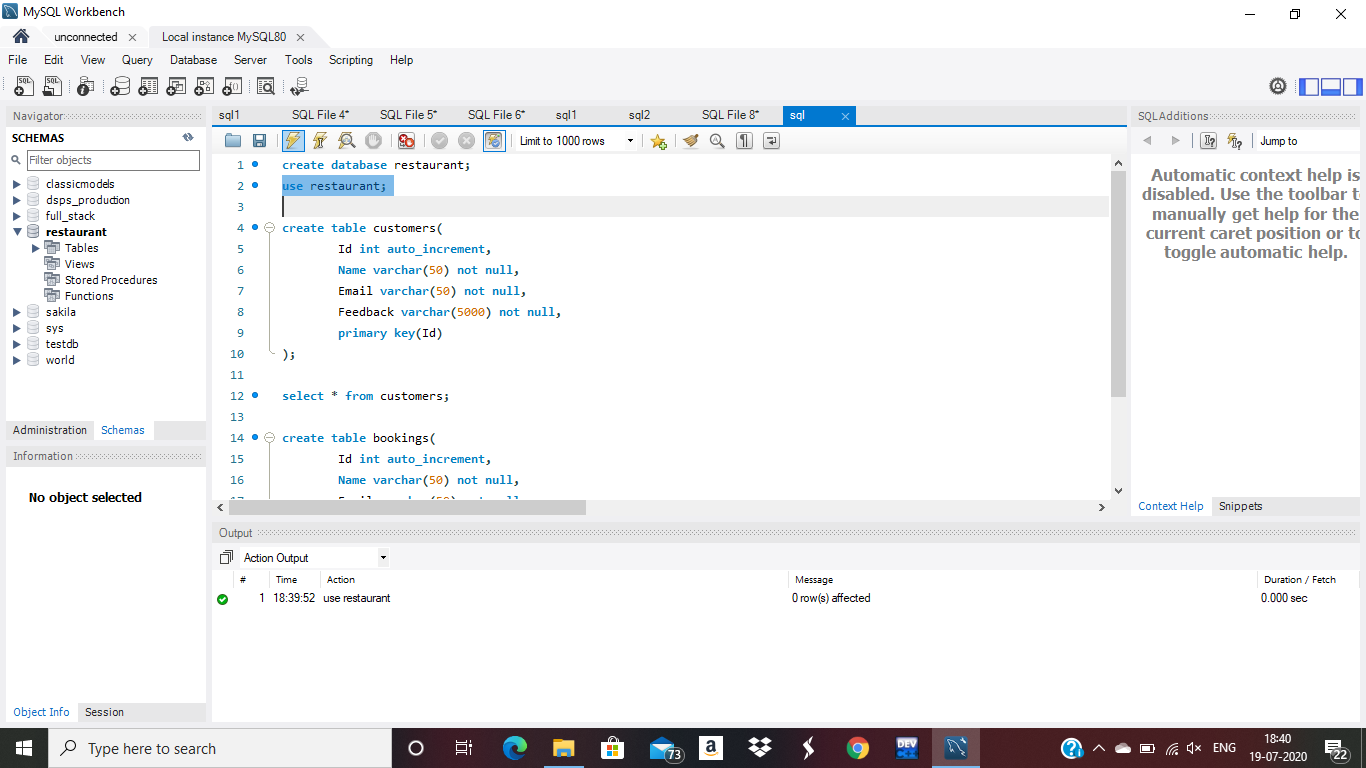
        return smoothScroll;

    };

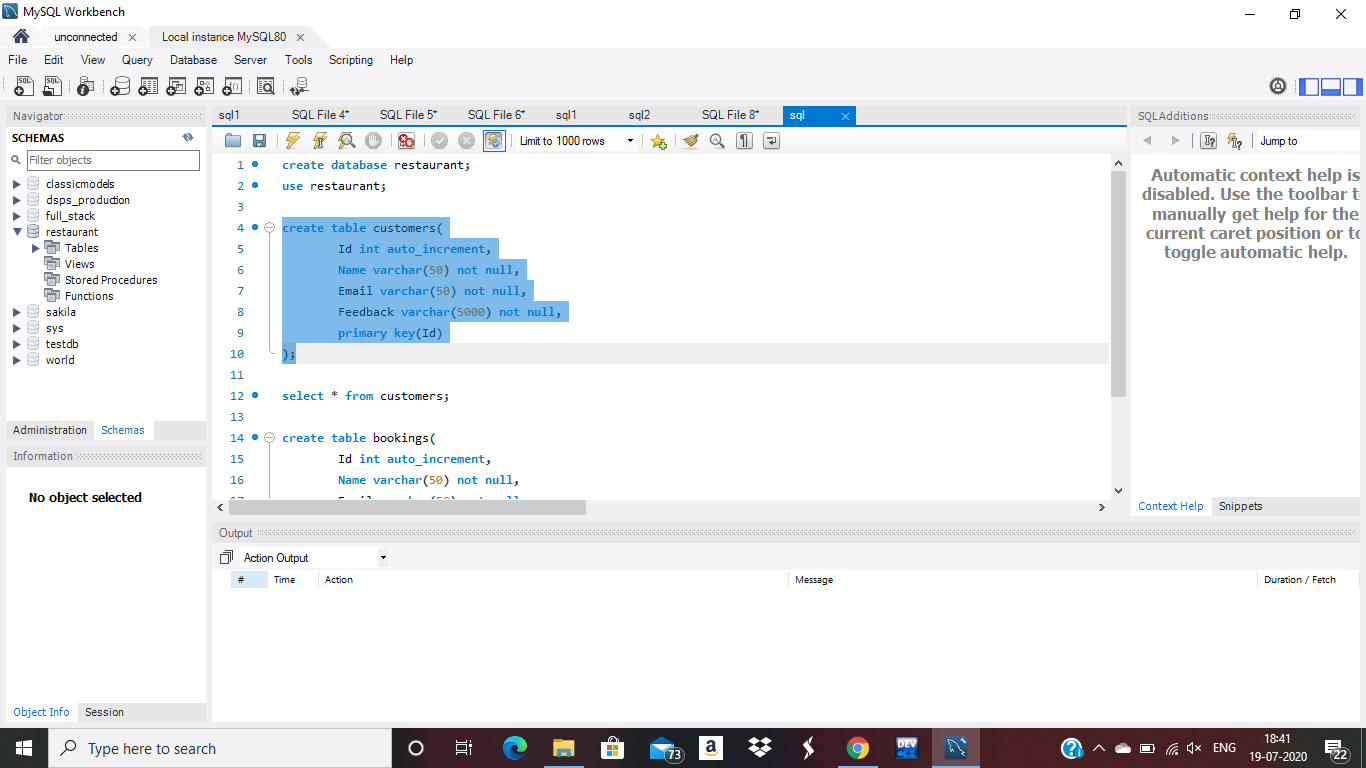
    return SmoothScroll;

})));

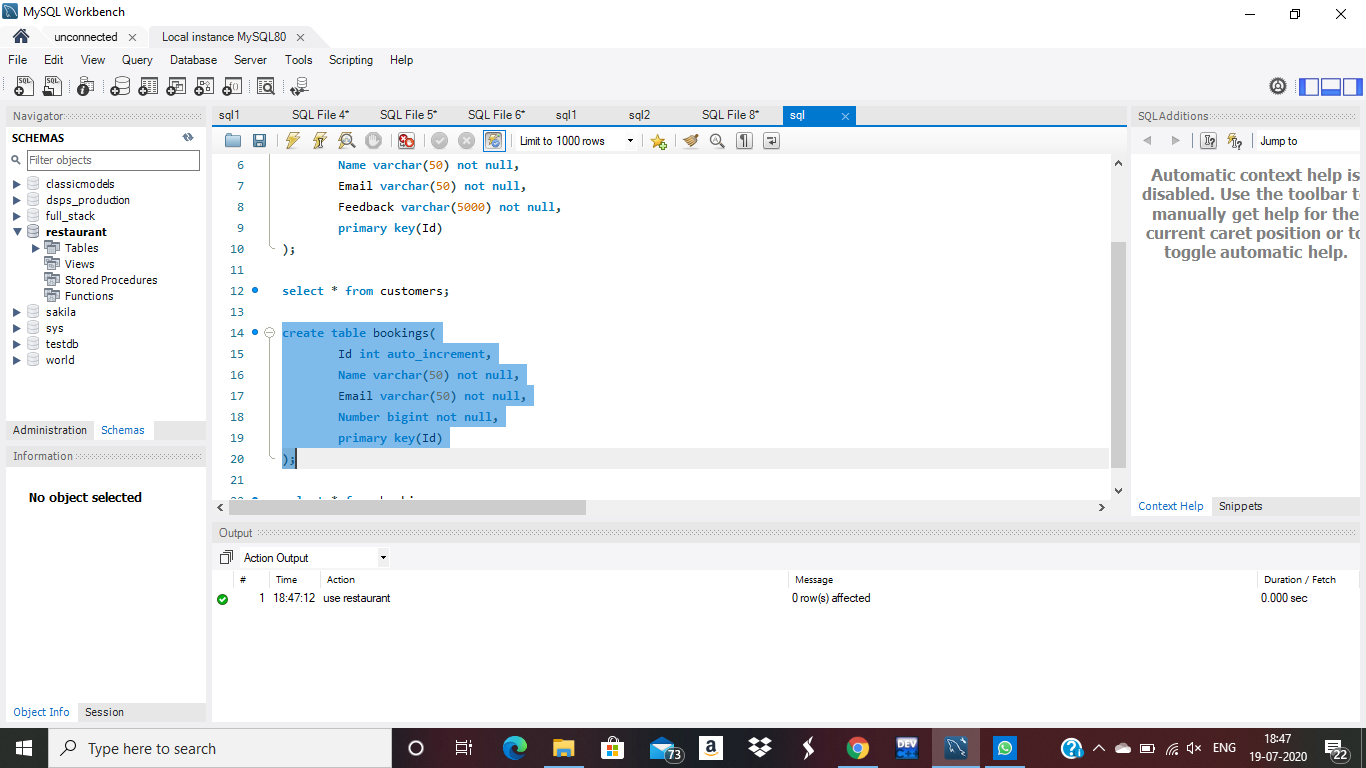
Using Database:



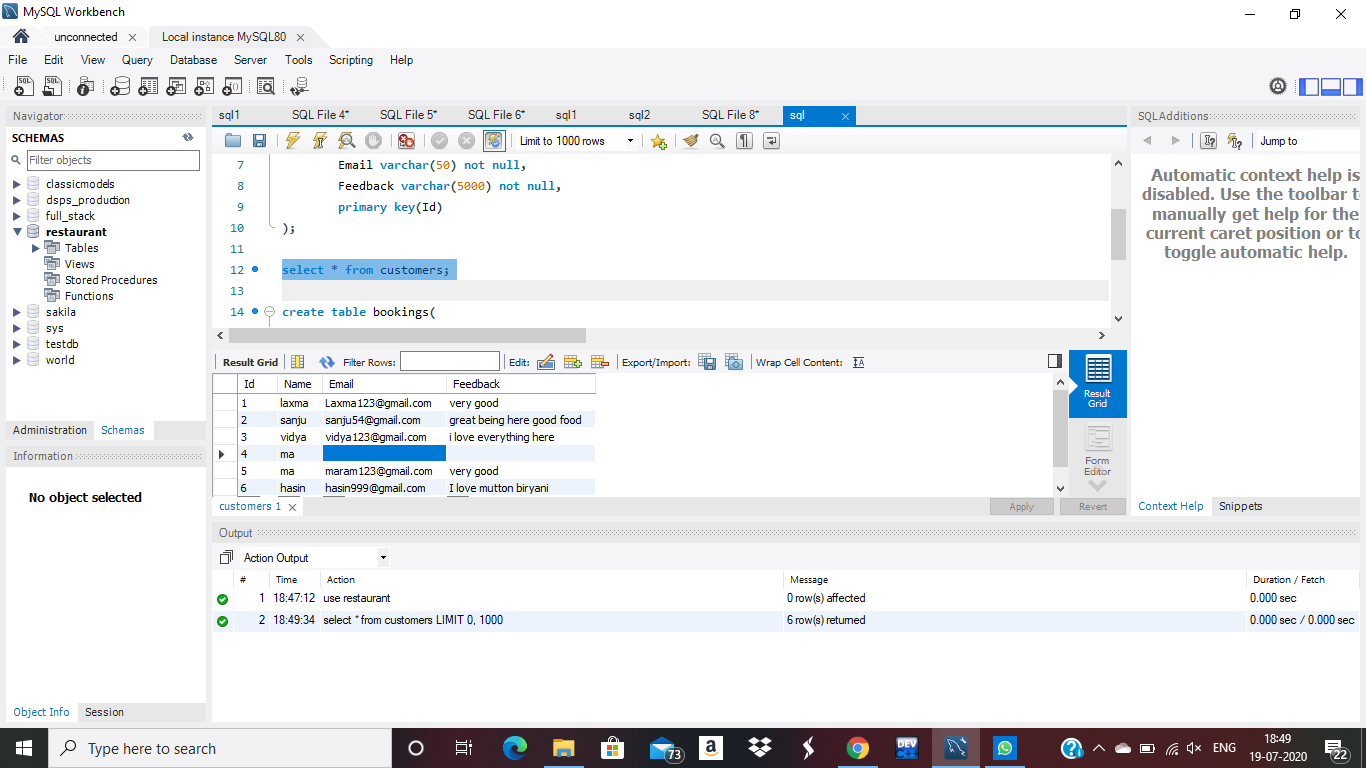
Creating a customer table:



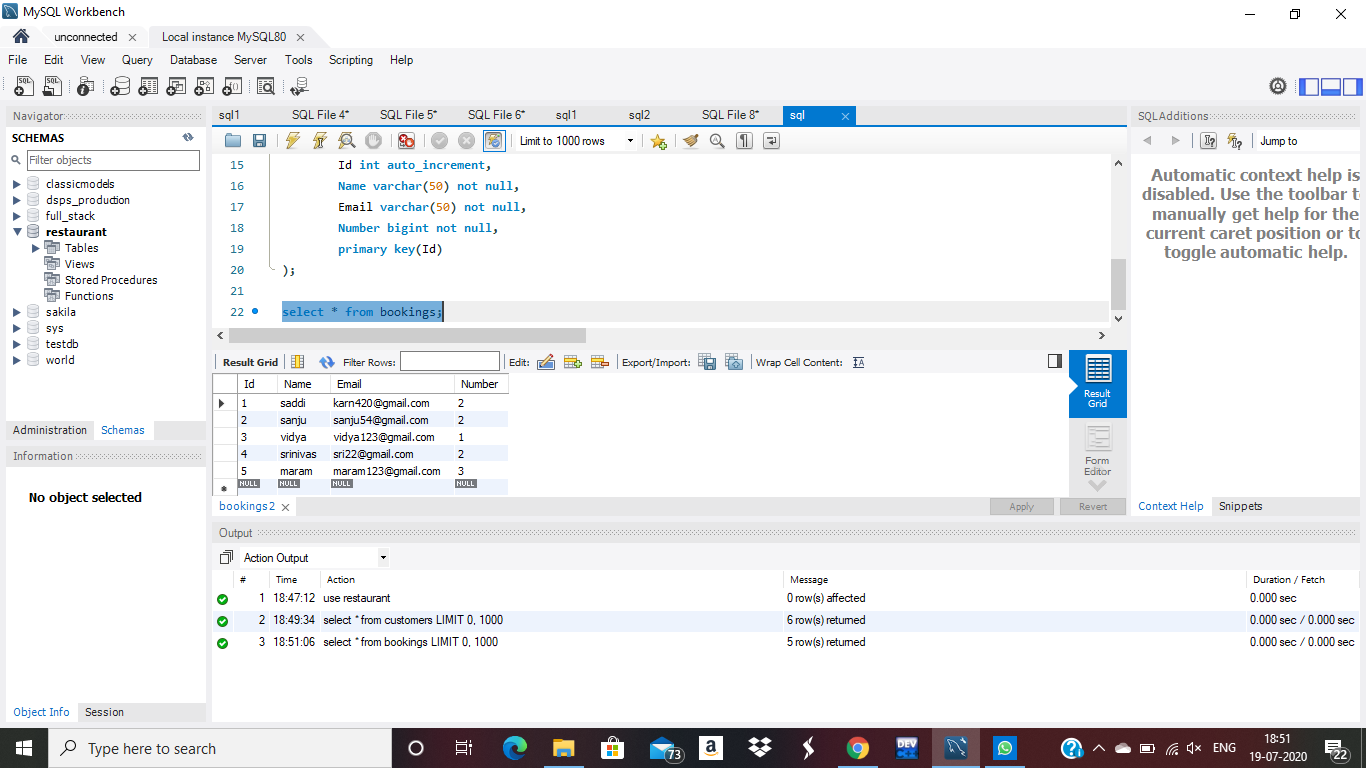
Creating Bookings Table:

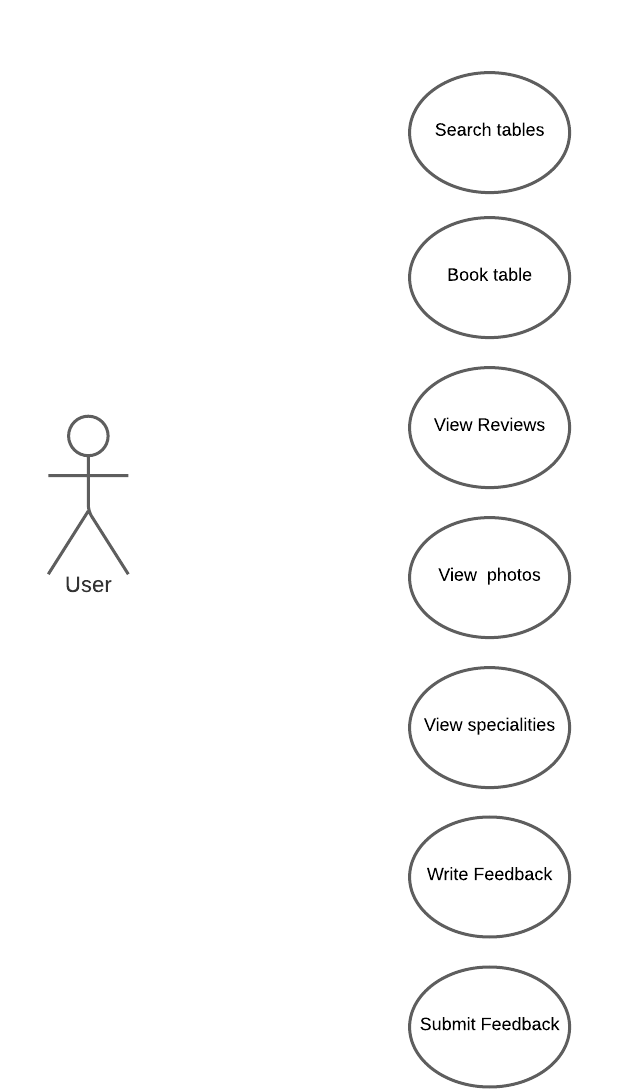


Selecting From Customers Table:



Selecting From Bookings Table:





**CHAPTER-9**

**MAINTAINANCE**

We made a lot of changes in regular intervals during the initial phase of our project to make the best we can do. The changes we did are mainly in creating the ui design as it requires the changes to better our design. We changed the whole ui design as its template is matching with some other project. We tried and made a lot of changes to make our design look fulfilled without missing anything. During the final stage of our project we made our backend .As backend is made in the final stage we didn’t changed anything because no changes are required. We don’t want to make anything complex so we didn’t change our backend. We also used the tools of the latest version by updating the available tool to make our work fast. We used the latest versions of mysql and visual studio code to get the advanced information about it. We also made minor changes in our project documentation to look unique. The one complex change we made is we changed entire ui design to be the best. We also used a application named postman to check for backend so that our user should not find any difficulty while he is using it. If we want users to like our software, we should design it to behave like a likeable person: respectful, generous and helpful.

**CHAPTER-10**

**FUTURE SCOPE & ENHANCEMENT**

PROJECT NAME:

RESTAURANT MANAGEMENT SYSTEM

* Our project will help each and every person to find the best food via our website.
* We are going to launch led screens in each table of our restaurant through which our customers do not require waiting for waiter to give the order.
* We are going to launch our online food delivery through our restaurant in near future.
* In near future you can have the most tastiest food at your home just by one click.

**CHAPTER -11**

**CONCLUSION**

We tried our best to design our website successfully. The main reason to complete this site is the tools we used. We wish we could take this site to the next level. We mainly concentrated on administration and end users because satisfying end user and administration is very important, but this is not enough we need make our restaurant food available to every person even the person is at home.

**BIBLIOGRAPHY**

* 1. www.getbootstrap.com
  2. [www.jquery-ui.com](http://www.jquery-ui.com)
  3. [www.fontawesome.com](http://www.fontawesome.com)
  4. [www.wikipedia.com](http://www.wikipedia.com)
  5. [www.w3schools.com](http://www.w3schools.com)
  6. [www.youtube.com/html](http://www.youtube.com/html)