**FULL STACK DEVELOPMENT**

**E-Commerce Cart Application**

*Summer Internship Report Submitted in partial fulfillment*

*of the requirement for under graduate degree of*

**Bachelor of Technology**

In

**Computer Science and Engineering**

By

**Kankanampati Nanda Kiran**

**221710301029**

<https://github.com/kankanampati/Full-Stack0>

*Under the Guidance of*

Assistant Professor



Department Of Mention your Branch Name

GITAM School of Technology

GITAM (Deemed to be University)

Hyderabad-502329

                                                   July 2020

**1**

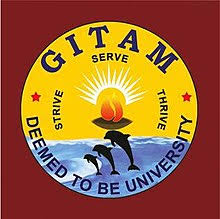
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF TECHNOLOGY**

**GITAM**

**(Deemed-to-be-University u/s 3 of UGC Act 1956)**

**HYDERABAD CAMPUS**

****

**DECLARATION**

I submit this industrial training work entitled **“E-Commerce Cart Application**”to GITAM (Deemed To Be University), Hyderabad in partial fulfillment of the requirements for the award of the degree of “**Bachelor of Technology**” in “**Computer Science and Engineering**”.I declare that it was carried out independently by me under the guidance of **(specify the name of the professor)**, Asst. Professor, GITAM (Deemed To Be University), Hyderabad, India.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Place: HYDERABAD                  K .Nanda Kiran

Date:                                      221710301029

**2**

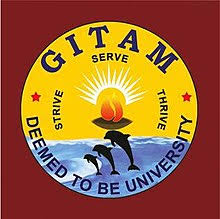
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM SCHOOL OF TECHNOLOGY**

**GITAM**

**(Deemed-to-be-University u/s 3 of UGC Act 1956)**

**HYDERABAD CAMPUS**

****

**CERTIFICATE**

  This is to certify that the Industrial Training Report entitled **“E-Commerce Cart Application ”** is being submitted by Student name (student roll number) in partial fulfillment of the requirement for the award of **Bachelor of Technology** **in Computer Science and Engineering** at GITAM (Deemed To Be University), Hyderabad during the academic year 2017-21

               It is faithful record work carried out by her at the **Computer Science and** **Engineering** , GITAM University Hyderabad Campus under my guidance and supervision.

**Dr.Phani Kumar**

Assistant Professor                                           Professor and HOD

Department of CSE                                           Department of CSE

**3**

**4**

**ACKNOWLEDGEMENT**

Apart from my effort, the success of this internship largely depends on the encouragement and guidance of many others. I take this opportunity to express my gratitude to the people who have helped me in the successful competition of this internship.

I would like to thank respected **Dr. N. Siva Prasad,** Pro Vice Chancellor, GITAM Hyderabad and **Prof.N.Seetharamaiah,** Principal, GITAM Hyderabad.

I would like to thank respected **Dr.Phani Kumar,** Head of the Department of Computer Science and Engineering for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present a internship report. It helped me a lot to realize of what we study for.

I would like to thank the respected faculties who helped me to make this internship a successful accomplishment.

I would also like to thank my friends who helped me to make my work more organized and well-stacked till the end.

                                                    K.Nanda Kiran

(221710301029)

**ABSTRACT**

1. commerce (Electronic Commerce or EC) is the buying and selling of goods and services on the Internet, especially the World Wide Web. In practice, this term and a newer term, e-business is often used interchangeably. For online retail selling, the term e-tailing sometimes used.

**5**

**Table Of Contents**

|  |  |  |
| --- | --- | --- |
|  | **Title** | **Page.No** |
|  | List Of Tables | 6-7 |
|  | List Of Figures | 8 |
| 1. | Full Stack Development  1.1 Introduction  1.2 Front-End Developement  1.3 Back-End Development | 9-10 |
| 2. | Web Development | 11-12 |
|  | 2.1 Web Site |  |
|  | 2.2 Webpage |  |
|  | 2.3 Web Server |  |
|  | 2.4 Web Browser |  |
| 3. | Steps To Create a Website |  |
|  | 3.1 UI Development | 13-23 |
|  | 3.1.1 HTML |  |
|  | 3.1.2 Versions of HTML |  |
|  | 3.1.3 Syntax of HTML |  |
|  | 3.1.4 Advantages of HTML |  |
|  | 3.1.5 Disadvantages of HTML |  |
|  | 3.1.6 CSS |  |
|  | 3.1.7 Types of CSS |  |
|  | 3.1.8 Advantages of CSS |  |
|  | 3.1.9 Disadvantages of CSS |  |
|  | 3.1.10 History of CSS |  |
|  | 3.1.11 Bootstrap |  |
|  | 3.1.12 Versions of Bootstrap |  |
|  | 3.1.13 Advantages of Bootstrap |  |
|  | 3.1.14 Disadvantages of Bootstrap |  |
|  | 3.2 Scripting | 24-31 |
|  | 3.2.1 Server-side Scripting |  |
|  | 3.2.2 Client-side Scripting |  |
|  | 3.2.3 Javascript |  |
|  | 3.2.4 Limitations of Javascript |  |
|  | 3.2.5 Angular JS |  |
|  | 3.2.6 Versions of Angular JS |  |
|  | 3.2.7 Advantages of Angular JS |  |
|  | 3.2.8 Disadvantages of Angular JS |  |

**6**

|  |  |  |
| --- | --- | --- |
|  | 3.2.9 React JS |  |
|  | 3.2.10 Features of React JS |  |
|  | 3.2.11 Advantages of React JS |  |
|  | 3.2.12 Limitations of React JS |  |
|  | 3.2.13 JQuery |  |
|  | 3.2.14 Advantages of JQuery |  |
|  | 3.2.15 Disadvantages of JQuery |  |
|  | 3.2.16 History of JQuery |  |
|  | 3.2.17 AJAX |  |
|  | 3.3 Back-End (or) Database | 32-38 |
|  | 3.3.1 Skill set required for Database |  |
|  | 3.3.2 NodeJS |  |
|  | 3.3.3 History of NodeJS |  |
|  | 3.3.4 Advantages of NodeJS |  |
|  | 3.3.5 Limitations of NodeJS |  |
|  | 3.3.6 Database |  |
|  | 3.3.7 Advantages of Database |  |
|  | 3.3.8 Disadvantages of Database |  |
|  | 3.3.9 MySQL(Tool for backend) |  |
|  | 3.3.10 History of MySQL |  |
|  | 3.3.11 Advantages of MySQL |  |
|  | 3.3.12 Disadvantages of MySQL |  |
|  | 3.3.13 Queries |  |
| 4. | Sotware Requirement Specification(SRS) | 39 |
|  | 4.1 Hardware Requirements |  |
|  | 4.2 Software Requirements |  |
| 5. | Project | 40-63 |
|  | 5.1 Technologies Used |  |
|  | 5.2 Technical Details |  |
|  | 5.3 Wireframes of the Project |  |
|  | 5.4 E-R-Diagrames |  |
|  | 5.5 Screenshots Of The Project |  |
|  | 5.6 Output of the Project |  |
| 6. | Maintanance | 64 |
| 7. | Future scope and future enhancement | 65 |
| 8. | Conclusion | 66 |
|  | Bibilography |  |

7

**List of Figures**

|  |  |
| --- | --- |
| Figures |  |
| 1.1 | Front-End VS Back-End Development |
| 2.1 | Web Browser Architecture |
| 3.1 | Steps to create website |
| 3.2 | CSS Level Structure |
| 3.3 | Before CSS |
| 3.4 | After CSS |
| 3.5 | Server side scripting |
| 3.6 | Client-side scripting |
| 3.7 | Javascript Applications |
| 3.8 | Back-End |
| 3.9 | Skill Set For Back-End |
| 3.10 | My Sql Work Bench |
| 5.1 | Home |
| 5.2 | Store |
| 5.3 | AboutUs |
| 5.4 | ContactUs |
| 5.5 | Login |
| 5.6 | Registration |
| 5.7 | Feedback |
| 5.8 | Forgot Password |

**8**

**CHAPTER-1**

**FULL STACK DEVELEOPMENT**

**1.1.Introduction**

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

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

**Full stack development**

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

**Full stack web Developers**

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

**1.2.Front-End Developement**

Front end development manages everything that users visually see first in their browser or application. Front end developers are responsible for the look and feel of a site.

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

The objective of designing a site is to ensure that when the users open up the site they see the information in a format that is easy to read and relevant. This is further complicated by the fact that users now use a large variety of devices with varying screen sizes and resolutions thus forcing the designer to take into consideration these aspects when designing the site. They need to ensure that their site comes up correctly

**9**

in different browsers (cross-browser), different operating systems (cross-platform)

and different devices (cross-device), which requires careful planning on the side of the developer.

**1.3.Back-End Development**

Back end development refers to the server side of an application and everything that communicates between the database and the browser.

Back-end development references a client server architecture common in ecommerce. Client side tends to have a strong user interface skill and the server side a strong API skill. But they overlap.

A backend developer could just be focused on a database and providing an API to it or on a server language like PHP and those mentioned so far and provide an API to it. Their work would need to be accessible to many other back end applications such as the database developer example and more to many front end applications. For example today many web backends service not only desktop and mobile web front ends but also device apps and even desktop apps. So the back end developer will not know how all of those front ends work but provide a server side interface such as a REST interface for those multiple front end technologies to access such as a login. Server developer can also make user interfaces that are probably less robust for administrative access to their code, functionality and data.



**Figure 1.1.Front-End VS Back-End Development**

**10**

**CHAPTER-2**

**WEB DEVELOPMENT**

**Web Development** refers to building, creating, and an maintaining websites. It includes aspects such as web design, web publishing, web programming, and database management.

While the terms "web developer" and "web designer" are often used synonymously, they do not mean the same thing. Technically, a web designer only designs website interfaces using HTML and CSS. A web developer may be involved in designing a website, but may also write web scripts in languages such as PHP and ASP. Additionally, a web developer may help maintain and update a database used by a dynamic website.

Web development includes many types of web content creation. Some examples include hand coding web pages in a text editor, building a website in a program like Dreamweaver, and updating a blog via a blogging website. In recent years, content management systems like WordPress, Drupal, and Joomla have also become popular means of web development. These tools make it easy for anyone to create and edit their own website using a web-based interface.

**2.1.WEB SITE**

A website (also written as web site) is a collection of web pages and related content that is identified by a common domain name and published on at least one web server.

All publicly accessible websites collectively constitute the World Wide Web. There are also private websites that can only be accessed on a private network, such as a company's internal website for its employees.Websites are typically dedicated to a particular topic or purpose, such as news, education, commerce, entertainment, or social networking. Hyperlinking between web pages guides the navigation of the site, which often starts with a home page.

Users can access websites on a range of devices, including desktops, laptops, tablets, and smartphones. The software application used on these devices is called a web browser.

**2.2.WEB PAGE**

A **web page** 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.

**11**

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 servers. The browser then transforms all of the elements into an interactive visual representation on the user's device.

A web page is often used to provide information to viewers, including pictures or videos to help illustrate important topics. A web page may also be used as a method to sell products or services to viewers. Multiple web pages make up a website, like our Computer Hope website.When you click a link provided by a search engine, you are accessing a web page. The Internet consists of millions of web pages, with more being added every day.

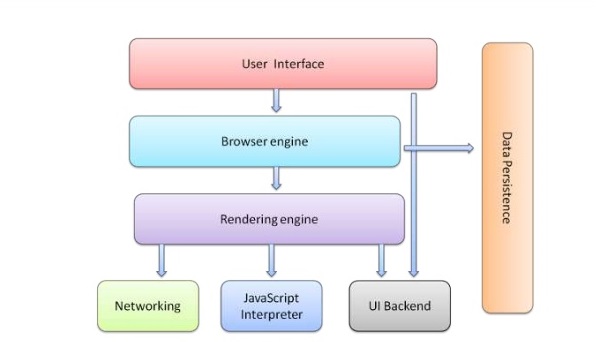
**2.3.WEB SERVER**

A computer that hosts a website on the Internet.

**2.4.WEB BROWSER**

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

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



**Figure 2.1.WEB BROWSER ARCHITECTURE**

**12**

**CHAPTER-3**

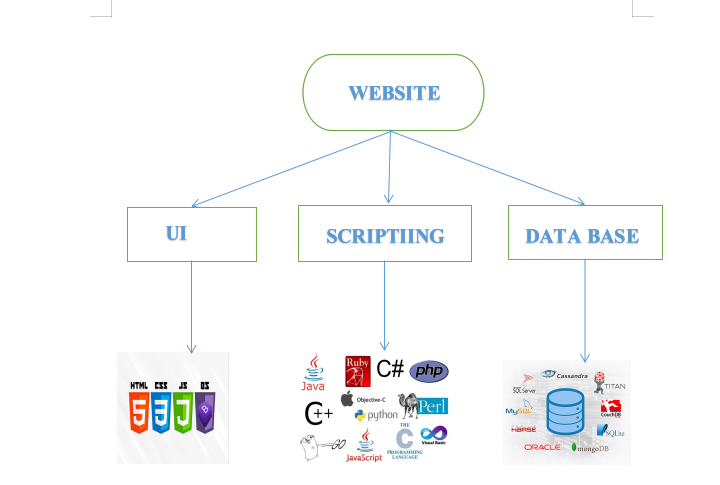
**STEPS TO CREATE A WEB SITE**

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

1.Creating a UI(User interface)

2.Scripting(Both at server end and client end)

3.Creating a backend or the database



**Figure 3.1.Steps to Create Web Site**

**13**

**3.1.USER INTERFACE DEVELOPMENT**

A user interface, also called a "UI" or simply an "interface," is the means in which a person controls a software application or hardware device. A good user interface provides a "user-friendly" experience, allowing the user to interact with the software or hardware in a natural and intuitive way.

Nearly all software programs have a graphical user interface, or GUI. This means the program includes graphical controls, which the user can select using a mouse or keyboard. A typical GUI of a software program includes a menu bar, toolbar, windows, buttons, and other controls. The Macintosh and Windows operating systems have different user interfaces, but they share many of the same elements, such as a desktop, windows, icons, etc. These common elements make it possible for people to use either operating system without having to completely relearn the interface. Similarly, programs like word processors and Web browsers all have rather similar interfaces, providing a consistent user experience across multiple programs.

Most hardware devices also include a user interface, though it is typically not as complex as a software interface. A common example of a hardware device with a user interface is a remote control. A typical TV remote has a numeric keypad, volume and channel buttons, mute and power buttons, an input selector, and other buttons that perform various functions. This set of buttons and the way they are laid out on the controller makes up the user interface. Other devices, such as digital cameras, audio mixing consoles, and stereo systems also have a user interface.

**3.1.1.HTML (HYPER TEXT MARKUP LANGUAGE)**

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.HTML uses "markup" to annotate text, images, and other content for display in a Web browser.

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way.

HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, <span>, <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, <ul>, <ol>, <li> and many others.

**14**

**3.1.2.VERSIONS OF HTML**

**HTML 1.0:** The original version of HTML was HTML 1.0. It had very limited features which greatly limited what you could do in designing your web pages.

**HTML 2.0:** HTML 2.0 then arrived and included all the features of HTML 1.0 plus several new features for web page design. Until January, 1997, HTML 2.0 was the standard in web page design.

**HTML 3.0:** HTML 2.0 served its purpose very well, but many people designing web pages (called HTML authors or webmasters) wanted more control over their web pages and more ways to mark up their text and enhance the appearance of their websites. Netscape, the leading browser at that time, introduced new tags and attributes called the Netscape Extension Tags. Other browsers tried to duplicate them but Netscape did not fully specify their new tags and so these extension tags did not work in most other browsers. It led to considerable confusion and problems when HTML authors used these tags and attributes and then saw that they didn't work as expected in other browsers.

At about that time, an HTML working group, led by Dave Raggett, introduced the HTML 3.0 draft which included many new and useful enhancements to HTML. However, most browsers only implemented a few elements from this draft. The phrase "HTML 3.0 enhanced" quickly became popular on the web but it more often than not referred to documents containing browser specific tags (discussed below in "The Netscape Problem" section), instead of referring to documents adhering to the HTML 3.0 draft. This was one of the reasons why the draft was abandoned. HTML 3.0 is now an expired draft. Another reason why HTML 3.0 did not make it was because it was so "big". Future versions were now to be introduced in a more "modular" way so that browsers can implement them modular by modular or bit by bit.

**HTML 3.2 (WILBUR):** As more browser-specific tags were introduced, it became obvious that a new standard was needed. For this reason, the Word Wide Web Consortium (W3C), founded in 1994 to develop common standards for the evolution of the World Wide Web, drafted the WILBUR standard, which later became known as HTML 3.2. HTML 3.2 captures the recommended practice as of early 1996 and became the official standard in January, 1997. Most, if not all, popular browsers in use today fully support HTML 3.2.

**HTML 4.0 (COUGAR):** In the early days, HTML 4.0 was code-named COUGAR. This version introduces new functionality, most of which comes from the expired HTML 3.0 draft. This version became a recommendation in December, 1997 and a standard as of April, 1998. Explorer has done a very good job in implementing the many features of HTML 4.0. Unfortunately, Netscape has not kept pace. The latest version of Netscape Communicator still does not recognize the many tags and attributes introduced with HTML 4.0. This means that a web page that involves HTML 4.0 specific tags will look great in Explorer but can look disastrous in Netscape.

**15**

**XHTML**

You would think that the next major version after HTML 4.0 would be HTML 5.0 and with it would come a bunch of new tags that would do all sorts of wonderful things. That would be a good guess - but it would also be a wrong guess. The next version of HTML after HTML 4 is XHTML.

XHTML stands for EXtensible HyperText Markup Language.

EXtensible

Hyper

Text

Markup

Language

XHTML is not bringing with it a lot of new tags. The purpose of XHTML is to address the new browser technologies that is sweeping the world. Today web pages are being viewed in browsers through cell/mobile phones, cars, televisions, plus a host of hand-held wireless devices and communicators. Alternate ways to access the internet are continually being introduced. In many cases, these devices will not have the computing power of a desktop or notebook computer and so will not be able to accommodate poor or sloppy coding practices. XHTML is designed to address these technologies. XHTML also begins to address the need for those with disabilities (such as the blind and visually impaired) to access the internet. Thus web pages written in XHTML will allow them to be viewed on a wide range of browsers and internet platforms.

XHTML is the result of the hard working World Wide Web Consortium (the W3C) to bring some sort of standard to provide rich high quality web pages through these varied devices. XHTML became an official W3C recommendation in January, 2000. XHTML is now a web standard and is the next generation of HTML.

### ****HTML 5****

### HTML 5 (usually written HTML5) is the new web standard. It follows HTML 4 (which came out way back in 1997) and XHTML. Since the introduction of HTML4, a lot has happened with the web and something needed to be done to address all the new technologies and latest multimedia. HTML5 is the result of cooperation that began in 2006 between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG). While HTML5 is still evolving (still under development), the latest browsers do support many of the new features and elements in this version. The basic aim of HTML5 is to provide two things - (1) to improve the language and (2) to support the latest multimedia. In order to accomplish this, some ground rules were established by the W3C and WHATWG. Among them were to reduce the need for external plug-ins (such as Flash plug-ins), better handling of errors, and more markup elements (tags) to replace scripting. HTML5 should also be device independent (that is, understood by computers and the many devices in existence today) while also keeping it easily readable by us humans.

**16**

**3.1.3.SYNTAX OF HTML**

<!DOCTYPE html>

<html>

<head>

<title>This is a title</title>

</head>

<body>

<div>

<p>Hello world!</p>

</div>

</body>

</html>

**<html> :** This is called HTML root element and used to wrap all the code.

**<head> :**  Head tag contains metadata, title, page CSS etc. All the HTML elements

thatcanbeusedinsidethe<head>elementare<style>,<title>,<script>,<meta>,<title>.

**<body>** : Body tag is used to enclosed all the data which a web page has from texts to links. All of the content that you see rendered in the browser is contained within this element.

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

**3.1.4.ADVANTAGES OF HTML**

1.**HTML** is widely used.

2.Every browser supports **HTML** Language.

3.Easy to learn and use.

4.Do not need to purchase any extra software because it is by default in every window.

**3.1.5.DISADVANTAGES OF HTML**

**HTML** can create only static and plain pages so if we need dynamic pages then **HTML** is not useful.

I need to write a lot of code for making a simple webpage.

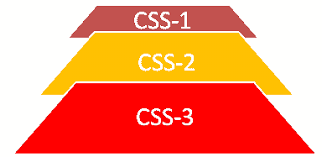
Security features are not good at **HTML**.

If we need to write long code for making a webpage then it produces some complexity.

**17**

**3.1.6. CSS(CASCADING STYLE SHEET)**

**Cascading Style Sheet(CSS)** is a separate language with its own syntax, which is enable you to lay out web pages exactly as you want them. As HTML grew, it came to encompass a wider variety of stylistic capabilities to meet the demands of web programmers. By the end of 1996, Cascading Style Sheets was ready to become official, and the CSS level 1 Recommendation was published in December, which is considered as first version of CSS. Each level of CSS builds upon the last, typically adding new features and typically denoted as CSS level 1 , CSS level 2 ,CSS level 3 , and CSS level 4 .



**Figure 3.2. CSS Level Structure**

**CSS Level 1**

CSS Level 1 (CSS1) was officially released in 1996, and included properties for adding font properties such as typeface and emphasis color of text, backgrounds, and other elements Text attributes such as spacing between words, letters, and lines of text. Unfortunately, lack of dependable **web browser** support prevented the popularity of CSS Level 1 for several years.

**CSS Level 2**

CSS level 2 (CSS2) specification was developed by the **W3C** and published as a recommendation in 1998. Its most notably added properties for positioning that allowed CSS to be used for **page layou**t .

**CSS Level 3**

The earliest **CSS level 3** (CSS3) drafts were published in 1999. CSS3 adds presentation-style properties, allowing you to effectively build presentations from Web documents. CSS level 3 is divided into several separate documents called modules. Due to the **modularization** , different modules have different stability and statuses.

**CSS Level 4**

There is no single **CSS4** specification and there is no standard which named as CSS4. But a few level 4 modules exist such as Image Values, Backgrounds & Borders, or Selectors etc., which build on the functionality of a preceding level 3 module. The level 4 modules can collectively be referred to as **CSS level 4** . There is only CSS standard, and each module can level up independently.

**18**

**3.1.7.Types of CSS**

**Inline CSS**

In this CSS is applied in between the tags.Inline CSS is used when only small changes are to be done to the HTML tag and the changes are to be reflected only to that specific tag

**Eg:** <tag style=”styling”>Hello World</tag>

**Internal CSS**

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

Internal CSS is used when the individual HTML pages have to be designed differently.This also slows the page load system if the internal styling is long

**General Syntax**

<html>

<head>

<style>

<! -- CSS STYLING -- >

</style>

</head>

</html>

**External CSS**

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

External CSS files are maintained to design multiple pages and use common styles overvarious pages. It is useful as it helps in managing the resources in an easy manner.External CSS uses the extension .css and is applied using the following syntax:

<html>

<head>

<link relation=”stylesheet” type=”css” href=”url to the page”>

</head>

</html>

**19**

**3.1.8.ADVANTAGES OF CSS**

**CSS saves time :** You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

**Easy maintenance :** To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

**Global web standards :** Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.

**Platform Independence :** The Script offer consistent platform independence and can support latest browsers as well.

**3.1.9.DISADVANTAGES OF CSS**

**Browser dependency :** Style sheets format things slightly differently on different browsers. Unfortunately, browsers have different support for HTML and style sheets. Newer browsers have largely converged on HTML support so that HTML documents look the same across different browsers. The state of style sheet support is somewhat worse, largely because style sheets are newer than HTML.

**Old Browsers :** Some very old browsers (such Netscape Navigator 2) do not support style sheets.

**3.1.10.HISTORY OF CSS**

Hakon Wium Lie first proposed CSS in 1994. He was working with Tim Berners-Lee, the father of HTML at CERN. Many other styling languages were also purposed at that time. CSS2 came out in 1998 with better features such as absolute, relative, fixed, positioning, z-index, the concept of media type, bidirectional text, and new font properties such as shadows. The development of CSS3 started later that year, but it was never completed.

**Syntax**

p {

color: red;

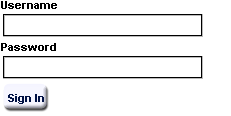
font-size: 20px;

}

<p> tag is the selector here. The declaration is written inside the curly brackets. In the example above, there are two declarations, separated by a semicolon. There can be any number of declarations.

**20**

**Before using CSS page of HTML display login page**



**Figure 3.3.Before CSS**

**After using CSS Page of HTML display for login page**

****

**Figure 3.4.After CSS**

**21**

**3.1.11.BOOTSTRAP**

Bootstrap is a free and open-source front-end web framework for designing websites

and web applications. It contains **HTML**- and **CSS**-based design templates for typography,forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

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

Bootstrap, originally named Twitter Blueprint, was developed by **Mark Otto ,Jacob Thornton** at Twitter as a framework to encourage consistency across internal tools.

**3.1.12.VERSIONS OF BOOTSTRAP**

**Bootstrap 2 and 3**

On January 31, 2012, Bootstrap 2 was released, which added built-in support for Glyphicons, several new components, as well as changes to many of the existing components. This version supports responsive web design, meaning the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (whether desktop, tablet, or mobile phone).

The next major version, Bootstrap 3, was released on August 19, 2013. It redesigned components to use flat design and a mobile first approach.

**Bootstrap 4**

Mark Otto announced Bootstrap 4 on October 29, 2014. The first alpha version of Bootstrap 4 was released on August 19, 2015. The first beta version was released on 10 August 2017.Mark suspended work on Bootstrap 3 on September 6, 2016, to free up time to work on Bootstrap 4. Bootstrap 4 was finalized on January 18, 2018.

**Bootstrap 5 Alpha**

Bootstrap 5 Alpha was officially released on 16 Jun 2020, although an experimental version of the package, created by Material Design for Bootstrap based on a developer version of the Alpha already surfaced the web weeks before.

Version 5 Alpha is currently the latest version of the package.

Among all **Bootstrap 4** supports the latest versions the Chrome, Firefox, Internet Explorer, Opera, and Safari (except on Windows). It additionally supports back to IE9 and the latest Firefox Extended Support Release (ESR).

**22**

**3.1.13.ADVANTAGES OF BOOTSTRAP**

1. Fewer Cross browser bugs.
2. A consistent framework that supports major of all browsers and CSS compatibility. Fixes
3. Lightweight and customizable.
4. Responsive structures and styles.
5. Several JavaScript plugins using the jQuery.
6. Good documentation and community support.
7. Loads of free and professional templates, WordPress themes and plugins.
8. Great grid system.

**3.1.14.DISADVANTAGES OF BOOTSTRAP**

1. There will be requirement of lots of style overrides or rewriting files that can thus lead to a lot of time spent on designing and coding the website if the design tends to deviate from the customary design used in Bootstrap.
2. You would have to go the extra mile while creating a design otherwise all the websites will look the same if you don’t do heavy customization.
3. Styles are verbose and can lead to lots of output in HTML which is not needed.
4. JavaScript is tied to jQuery and is one of the commonest library which thus leaves most of the plugins unused.
5. Non-compliant HTML.

**23**

**3.2.SCRIPTING**

There are two types of scripting methodologies:

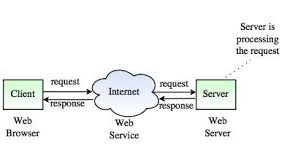
1. Server side Scripting: This scripting is done at the server end.
2. Client side Scripting: This scripting is done at the client end or the browser.

**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..

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.



**Figure 3.5.Server side scripting**

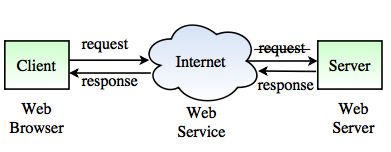
**24**

**3.2.2.CLIENT- SIDE SCRIPTING**

**Client-side Scripting** is utilized when the client’s (user’s) browser has all the code and the page is modified on the basis of the client’s (user’s) information. The Web Browser executes the client-side scripting that locates within the user’s computer. Client-side scripts are also known as the embedded script (as they are often embedded within an **HTM**L or XHTML document).

The browser gets the page sent by the server & executes the client-side scripts. Client-side scripting can’t be utilized to join with the databases on the web server. Client-side scripting cannot get the file system which lies at the web server.

The records and settings which are local at the client’s (user’s) computer can be approached employing Client-side scripting language. It’s generally observed the response from a client-side script is faster when compared to a server-side scripting language as the scripts are prepared on the local computers. JavaScript, VB scripts are some common examples of client-side scripting.



**Figure 3.6.Client-side scripting**

**25**

**3.2.3.JAVA SCRIPT**

**JavaScript (JS)** is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as **Node.js**, **Apache CouchDB** and **Adobe Acrobat**.

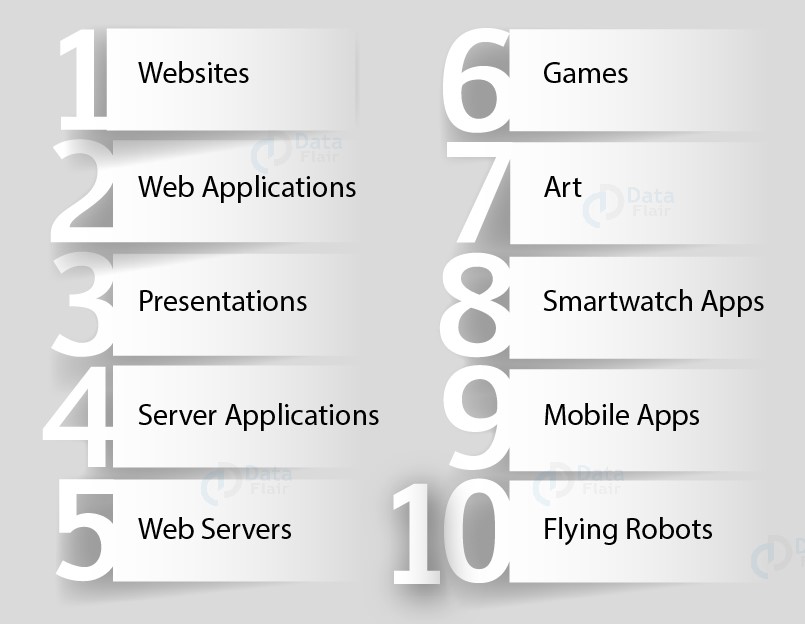
JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. Read more about JavaScript.

This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments. For information about API specifics to Web pages, please see Web **APIs** and **DOM**.

**3.2.4.LIMITATIONS OF JAVASCRIPT**

1. Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
2. JavaScript cannot be used for networking applications because there is no such 3.support available.

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



**Figure 3.7. Javascript Applications**

**26**

**FRONT END FRAME WORKS AND LIBRARIES**

**3.2.5.ANGULAR.JS**

**AngularJS** is a JavaScript-based open-source front-end web framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and **model–view–viewmodel** (MVVM) architectures, along with components commonly used in rich Internet applications.

**AngularJS** is the **frontend** part of the **MEAN** stack, consisting of **MongoDB** database, Express.js web application server framework, Angular.js itself, and Node.js server runtime environment. Version 1.7.x is on Long Term Support until July 1st 2021. After that date AngularJS will no longer be updated and Angular (2.0+) is suggested instead.

The AngularJS framework works by first reading the **Hypertext Markup Language** (HTML) page, which has an additional custom HTML attributes embedded into it. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic **JSON** resources.

AngularJS is built on the belief that declarative programming should be used to create user interfaces and connect software components, while imperative programming is better suited to defining an application's business logic. The framework adapts and extends traditional HTML to present dynamic content through two-way data-binding that allows for the automatic synchronization of models and views. As a result, AngularJS de-emphasizes explicit **Document Object Model** (DOM) manipulation with the goal of improving **testability** and **performance.**

**3.2.6.VERSIONS OF ANGULAR JS**

**Angular 2:** The release of Angular 2 brought numerous changes to the initial framework, as it was rewritten in TypeScript. The architectural style switched to component-based.

**Angular 4:** Angular CLI 1.0.0 was introduced with the fourth version, as a core element of the Angular project. With the release of Angular Universal, Angular applications could be rendered outside the browser.

**Angular 5-6:** The release of fifth and sixth version concentrated on optimizing Angular CLI and compiler work.

**27**

**Angular 7:** With Angular 7 CLI was enhanced with prompts which provided tips in CLI to explain functions and purposes of elements. Hence, using CLI became **more** intuitive. Applications received various improvements in the performance and size of the code-base.

**Angular 8:** the latest version of Angular: In Angular 8, two elements were introduced Ivy renderer, Bazel (build interface). Another major improvement is differential loading that is used to upload browser-specific bundles in order to support legacy browsers and upload content faster.

**3.2.7.ADVANTAGES OF ANGULAR.JS**

1. **Easy to work with**: All you need to know to work with AngularJs is basics of HTML,CSS and Javascript, not necessary to be an expert in these technologies.
2. **Time saving:** AngularJs allows us to work with components and hence we can use them again which saves time and unnecessary code.
3. **Ready to use template**: AngularJs is mainly plain HTML, and it mainly makes use of the plain HTML template and passes it to the DOM and then the AngularJS compiler. It traverses the templates and then they are ready to use.

**3.2.8.DISADVANTAGES OF ANGULAR.JS:**

1. **Complex at times:** At times AngularJS becomes complex to handles as there are multiple ways to do the same thing. This creates confusion and requires considerable efforts.
2. **Not Secure:** Its applications are not safe. Server side authentication and authorization is necessary to keep an application secure.
3. **Not Degradable:**If user of your application disables the JavaScript then it displays nothing except basic page.

**28**

**3.2.8. REACTJS**

**ReactJS** is JavaScript library used for building reusable UI components. According to React official documentation.

React is a library for building **composable** user interfaces. It encourages the creation of reusable UI components, which present data that changes over time. Lots of people use React as the V in MVC. React abstracts away the DOM from you, offering a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using **React Native**. React implements one-way reactive data flow, which reduces the boilerplate and is easier to reason about than traditional data binding.

**3.2.9.FEATURES OF REACT**

1.**JSX :**JSX is JavaScript syntax extension. It isn't necessary to use JSX in React development, but it is recommended.

2.**Components:** React is all about components. You need to think of everything as a component. This will help you maintain the code when working on larger scale projects.

3.**Unidirectional data flow and Flux :** React implements one-way data flow which makes it easy to reason about your app. Flux is a pattern that helps keeping your data unidirectional.

4.**License:** React is licensed under the Facebook Inc. Documentation is licensed under CC BY 4.0.

**3.2.10.ADVANTAGES OF REACT**

1.Uses **virtual DOM** which is a JavaScript object. This will improve apps performance, since JavaScript virtual DOM is faster than the regular DOM.

2.Can be used on client and server side as well as with other frameworks.

1. Component and data patterns improve readability, which helps to maintain larger apps.

**3.2.11.LIMITATIONS OF REACT**

1.Covers only the view layer of the app, hence you still need to choose other technologies to get a complete tooling set for development.

1. Uses inline templating and JSX, which might seem awkward to some developers.

**29**

**3.2.12.JQUERY**

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax.It is free, open-source software using the permissive MIT License.As of May 2019, jQuery is used by 73% of the 10 million most popular websites.Web analysis indicates that it is the most widely deployed JavaScript library by a large margin, having 3 to 4 times more usage than any other JavaScript library.

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

**3.2.13.ADVANTAGES OF JQUERY**

1. It is a very simple easy to use compared to ordinary javascript and further **javascript** libraries.

2. It allows you to execute hordes of functions in comparison to other **Javascript** libraries.

3. The **JQuery** site has an inclusive documentation and tutorials to obtain even an complete beginner in programming to get the ball rolling with this library.

**3.2.14.DISADVANTAGE OF JQUERY**

1.Functionality maybe limited

1. The **JQuery javascript** folder is necessary to run **JQuery** commands, while the size of this folder is moderately very small.

**3.2.15.HISTORY OF JQUERY**

JQuery was originally created in January 2006 at BarCamp NYC by **John Resig,** influenced by **Dean Edwards**' earlier **cssQuery library**. It is currently maintained by a team of developers led by **Timmy Willison** (with the jQuery selector engine, Sizzle, being led by Richard Gibson).

JQuery was originally licensed under the CC BY-SA 2.5, and relicensed to the MIT license in 2006. At the end of 2006, it was dual-licensed under GPL and MIT licenses. As this led to someconfusion, in 2012 the GPL was dropped and is now only licensed under the MIT license

**30**

**3.2.16.AJAX**

AJAX is an acronym that stands for Asynchronous **JavaScript and XML**, and it describes a set of development techniques used for **building websites** and web applications. According to web developer and Skillcrush WordPress instructor Ann Cascarano, the best way to understand AJAX is to start with identifying its specific purpose in the web development process. AJAX’s core function is to update web content asynchronously (the “A” of AJAX), meaning a user’s web browser doesn’t need to reload an entire web page when only a small portion of content on the page needs to change.

One of the most ubiquitous examples of asynchronous updating is Google’s “Google Suggest” feature. When you enter a search query into Google’s search bar and the Google website automatically begins offering auto-complete options while you type, that’s AJAX in action. The content on the page changes (in this case, the auto-complete options in the search bar) without having to manually refresh the page (something that would make Google Suggest impractical to use). Features like Google Suggest are a fundamental part of contemporary web browsing, which points to how essential AJAX is in web development. In addition to Google Suggest, Cascarano says that AJAX is commonly used to update features like status and notification bars, online forms, comments sections, and surveys and polls.

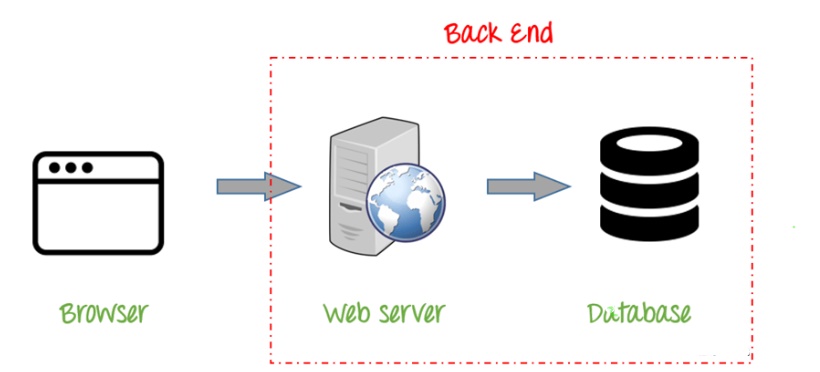
The popular JavaScript library**[jQuery](https://en.wikipedia.org/wiki/JQuery" \o "JQuery)** has implemented abstractions which enable developers to use Ajax more conveniently

**31**

**3.3.BACK-END OR DATABASE**

**Back-end Development** refers to the server-side development. It is the term used for the behind-the-scenes activities that happen when performing any action on a website. It can be logging in to your account or purchasing a watch from an online store.

Backend developer focuses on databases, scripting, and the architecture of websites. Code written by back-end developers helps to communicate the database information to the browser.



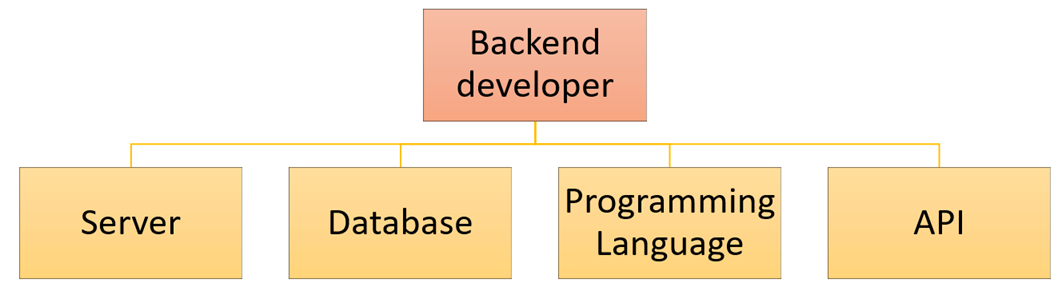
**Figure 3.8. Back-End**

**3.3.1.SKILL SET REQUIRED FOR BACKEND**

A **backend** developer works with the following:

1. Web Development Languages
2. Database and cache
3. Server
4. API (REST & SOAP)

**32**



**Figure 3.9.Skill Set For Back-End**

**3.3.2.NODE.JS: Node.js** belongs to JavaScript family, JavaScript is high-level interpreted programming language which is characterized as dynamic, weakly typed, prototype-based and **multi-paradigm**. Node.js is open-source, cross-platform JavaScript runtime environment provides a run-time environment with ease. It used to build application along with abstraction that makes it easier to create event-driven applications.

**3.3.3.HISTORY OF NODE.JS**

Node. js was written initially by Ryan Dahl in 2009, about thirteen years after the introduction of the first server-side JavaScript environment, Netscape's LiveWire Pro Web. The initial release supported only Linux and Mac OS X. Its development and maintenance was led by Dahl and later sponsored by Joyent.

Language used: JavaScript, CoffeeScript, C++.

**3.3.4.ADVANTAGES OF NODE.JS**

1. **Easy Scalability**: Node.JS is a lightweight technology and one of the key benefits of Node.JS is that it is considered to be the best option for micro-service architecture. Developers find it quite easy to scale the applications in horizontal as well as vertical directions. It becomes easier to add more micro-services on the top of the existing one.
2. **JS is Easy to Learn:** Since JavaScript is one of the most widely accepted and popular programming languages among developers, most of the front-end developers have a good hold on it. For them, to use Node.JS as backend becomes a lot easier.
3. **Used as Single Programming Language**: Node.JS offers the developers facility of writing server-side application in the JavaScript. This allows the developers to write front-end and back-end web application in JavaScript using a runtime environment.

**33**

As all the web browsers support JavaScript, so the deployment of web applications also becomes much easier.

1. **Benefit of Full Stack JS:** It is known as full stack JavaScript as it serves both client and server side applications. It adds up as one more benefit of Node.JS as you don’t need to hire separate developers for back-end and front-end development. It helps to save your money, time and effort.
2. **Offers High Performance:** As mentioned earlier, Node.JS runs the JavaScript code via Google V8 JavaScript engine and it compiles the JavaScript code directly into the machine code. This process facilitates easier and faster implementation of codes in an effective manner.

**3.3.5.Limitations of Node.js**

1.It doesn’t support **multi-threaded** programming.

2.It doesn’t support very high computational intensive tasks. When it executes long running task, it will queue all the incoming requests to wait for execution, since it follows JavaScript event loop which is single threaded.

3.Node good for executing synchronous and CPU intensive tasks.

**3.3.6.DATABASE**

**DATA: Data** is defined as facts or figures, or information that's stored in or used by a computer. An example of **data** is information collected for a research paper.

**DATABASE:** A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

**1.Oracle:** Oracle database is the collection of data which is treated as a unit. The purpose of this database is to store and retrieve information related to the query. It is a database server and used to manages information.

**2.MongoDB:** MongoDB, the most popular NoSQL database, is an open source document-oriented database. The term ‘NoSQL’ means ‘non-relational’. It means that

**34**

MongoDB isn’t based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data.

**3.Sql:** SQL is the standard language for dealing with Relational Databases. SQL can be used to insert, search, update, and delete database records. SQL can do lots of other operations, including optimizing and maintenance of databases. SQL stands for Structured Query language, pronounced as "S-Q-L" or sometimes as "See-Quel"... Relational databases like MySQL Database, Oracle, MS SQL Server, Sybase, etc. use ANSI SQL.

**DBMS(Data Base Management System)**

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

**SQL KEYWORDS**

**ADD -** Adds a column in an existing table

**ALL -** Returnstrue if all of the sub query values meet the condition

**AND -** Only includes rows where both conditions is true

**ANY -** Returns true if any of the sub query values meet the condition

**CREATE -** Creates a database, index, view, table, or procedure

**DELETE -**Deletes rows from a table

**DESC -**Sorts the result set in descending order

**DISTINCT -** Selects only distinct (different) values

**ALTER -** Adds, deletes, or modifies columns in a table, or changes the data type of a column in a table

**35**

**3.3.7.ADVANTAGES OF DATABASE**

1.Reduced data redundancy

2.Reduced updating errors and increased consistency

3.Greater data integrity and independence from applications programs

4.Improved data access to users through use of host and query languages

5.Improved data security

6.Reduced data entry, storage, and retrieval costs

1. Facilitated development of new applications program

**3.3.8.DISADVANTAGES OF DATABASE**

1.Database systems are complex, difficult, and time-consuming to design

2.Substantial hardware and software start-up costs

3.Damage to database affects virtually all applications programs

4.Extensive conversion costs in moving form a file-based system to a database system

1. Initial training required for all programmers and users

**3.3.9.MY SQL(TOOL FOR BACK-END)**

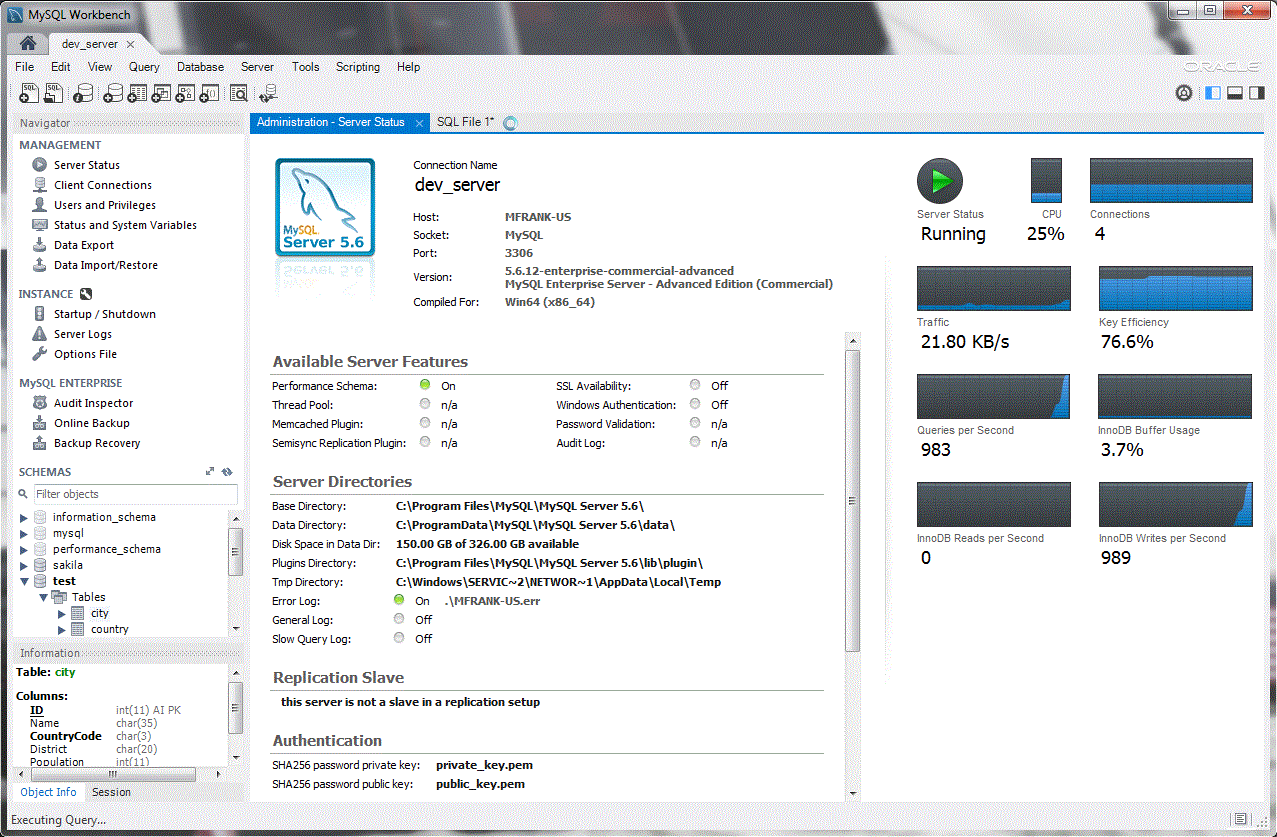
MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and Mac OS X.

**3.3.10.HISTORY OF MYSQL**

The first preview version of MySQL Workbench was released in September 2005, and was not included in the [MySQL GUI Tools Bundle](https://en.wikipedia.org/wiki/MySQL_Workbench" \l "MySQL_GUI_Tools_Bundle). Development was started again in 2007 and MySQL Workbench was set to become the MySQL GUI flagship product.

Version numbering was started at 5.0 to emphasise that MySQL Workbench was developed as the successor to DBDesigner4.

**36**

****

**Figure 3.10.My Sql Work Bench**

**3.3.11.ADVANTAGES OF MY SQL**

**Faster Query Processing :** Large amount of data is retrieved quickly and efficiently. Operations like Insertion, deletion, manipulation of data is also done in almost no time.

**No Coding Skills :** For data retrieval, large number of lines of code is not required. All basic keywords such as SELECT, INSERT INTO, UPDATE, etc are used and also the syntactical rules are not complex in SQL, which makes it a user-friendly language.

**Standardised Language :** Due to documentation and long establishment over years, it provides a uniform platform worldwide to all its users.

**Portable :**It can be used in programs in PCs, server, laptops independent of any platform (Operating System, etc). Also, it can be embedded with other applications as per need/requirement/use.

**Interactive Language :** Easy to learn and understand, answers to complex queries can be received in seconds.

**37**

**3.3.12.DISADVANTAGES OF MY SQL**

**Complex Interface :** SQL has a difficult interface that makes few users uncomfortable while dealing with the database.

**Cost :** Some versions are costly and hence, programmers cannot access it.

**Partial Control :** Due to hidden business rules, complete control is not given to the database.

**3.3.13.QUERIES**

A query is a question, often expressed in a formal way. A database query can be either a select query or an action query. A select query is a data retrieval query, while an action query asks for additional operations on the data, such as insertion, updating or deletion.

A query includes a list of columns to include in the final result, normally immediately

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

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

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

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

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

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

The **DISTINCT** keyword eliminates duplicate data.

**38**

**CHAPTER 4**

**SOFTWARE REQUIREMENT SPECIFICATION**

**4.1.Hardware Requirements**

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

|  |  |
| --- | --- |
| Processor | Intel CORE i5 |
| RAM | 4.0 GB |
| Hard Disk Drive | 500 GB |

**4.2 Software Requirements**

|  |  |
| --- | --- |
| **Number** | **Description** |
| 1 | Windows 10 |
| **2** | HTML/CSS/JavaScript/ Bootstrap/Node.js/JQuery. |
| **3** | Server : Local Host |
| **4** | MySQL |
| **5** | Compiler: MSVC11 (Visual C++ 201) |

**39**

**CHAPTER 5**

**PROJECT NAME:** **E-Commerce Cart Application**

**5.1.Technologies Used**



* HTML
* CSS
* Bootstrap
* Java Script
* Jquery
* JSon



**Database:** MySql

**Operating System:** Windows7/8/8.1/10

**Wire framing tool:** Paints

**Team Size: 3**

**Team Size : 3**

|  |  |
| --- | --- |
| **NAME** | **WORK DONE** |
| K.Nanda Kiran | Front End of The Project |
| V.Polasani Vikas | Front End of The Project |
| V.Sai Vikas Reddy | Back End of The Project |

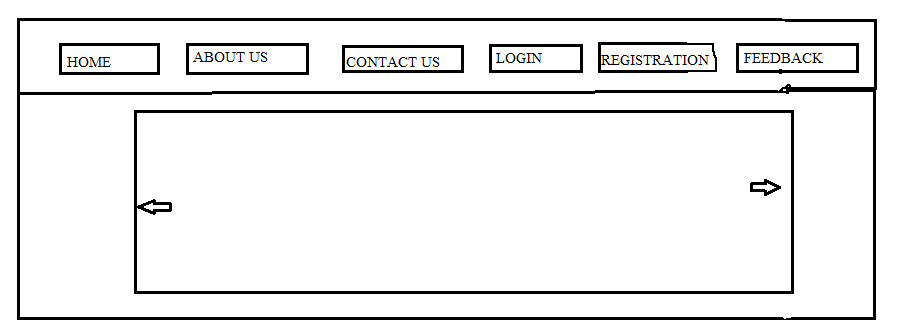
**5.2.TECHNICAL DETAILS**

Front end is designed using HTML, CSS and Bootstrap.JavaScript used to perform client side scripting.Backend is based on  MySql based RDB(Relational Data Base) model.The SQL queries are run using the CI SQL library functions.

Backend online host includes a centralized database resident on the server, the script is used to SQL query the database on user’s request for transaction of data.The forms are made using the HTML, Bootstrap for designing and sql for back-end.JavaScript, and JQuery used for client side scripting and Nodejs for the server side development.

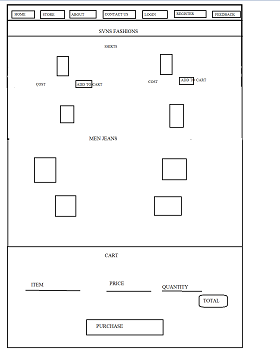
**40**

**5.3.WIRE FRAMES OF PROJECT**

**Home **

**Figure 5.1.Home**

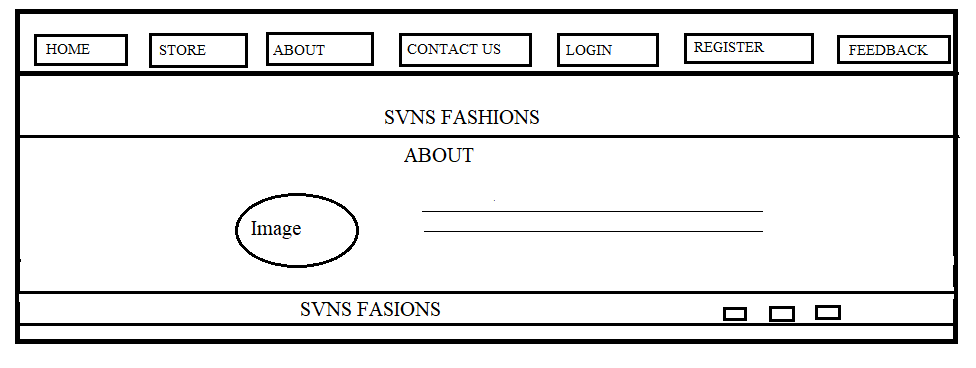
**STORE**

****

**Figure 5.2.Store**

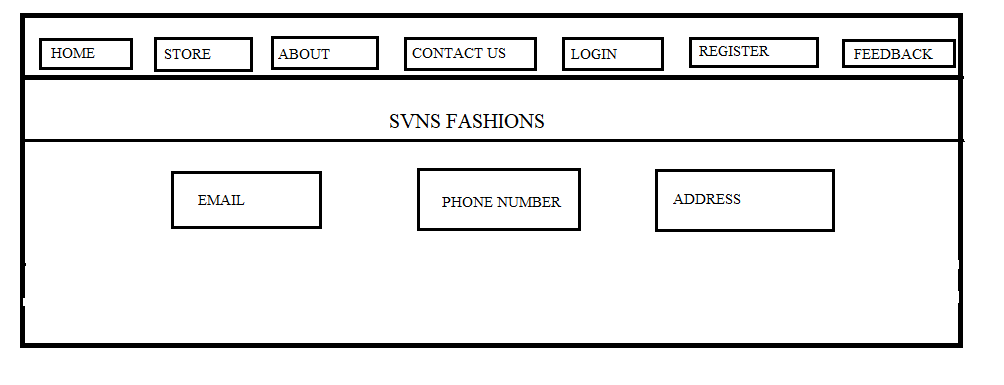
**41**

**ABOUT**

****

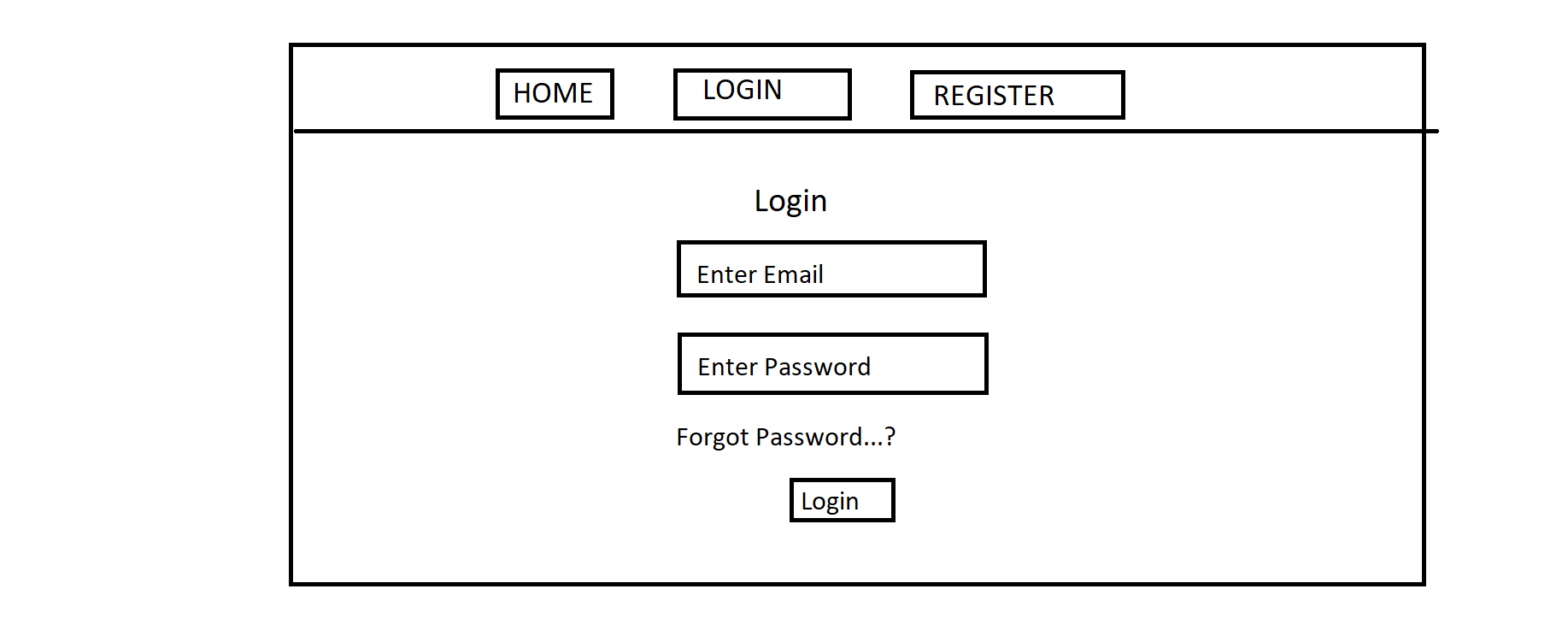
**Figure 5.3.About**

**CONTACT US**

****

**Figure 5.4.Contact Us**

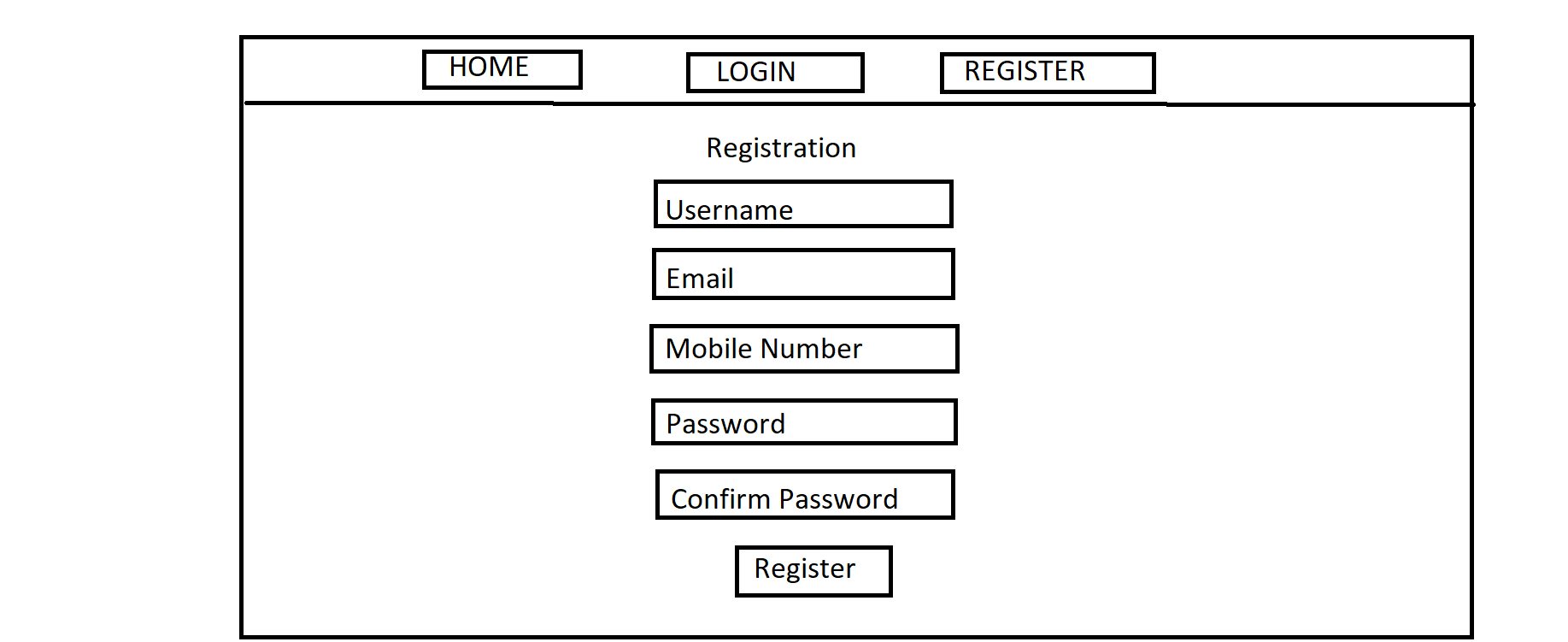
**LOGIN**

****

**Figure 5.5.Login**

**42**

**REGISTRATION**

****

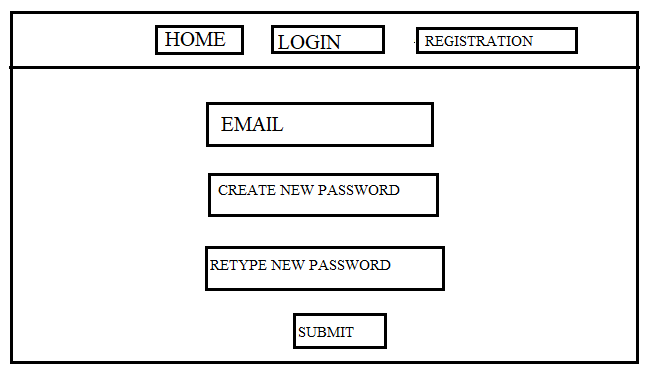
**Figure 5.6.Registration**

**FEEDBACK**

****

**Figure 5.7.FeedBack**

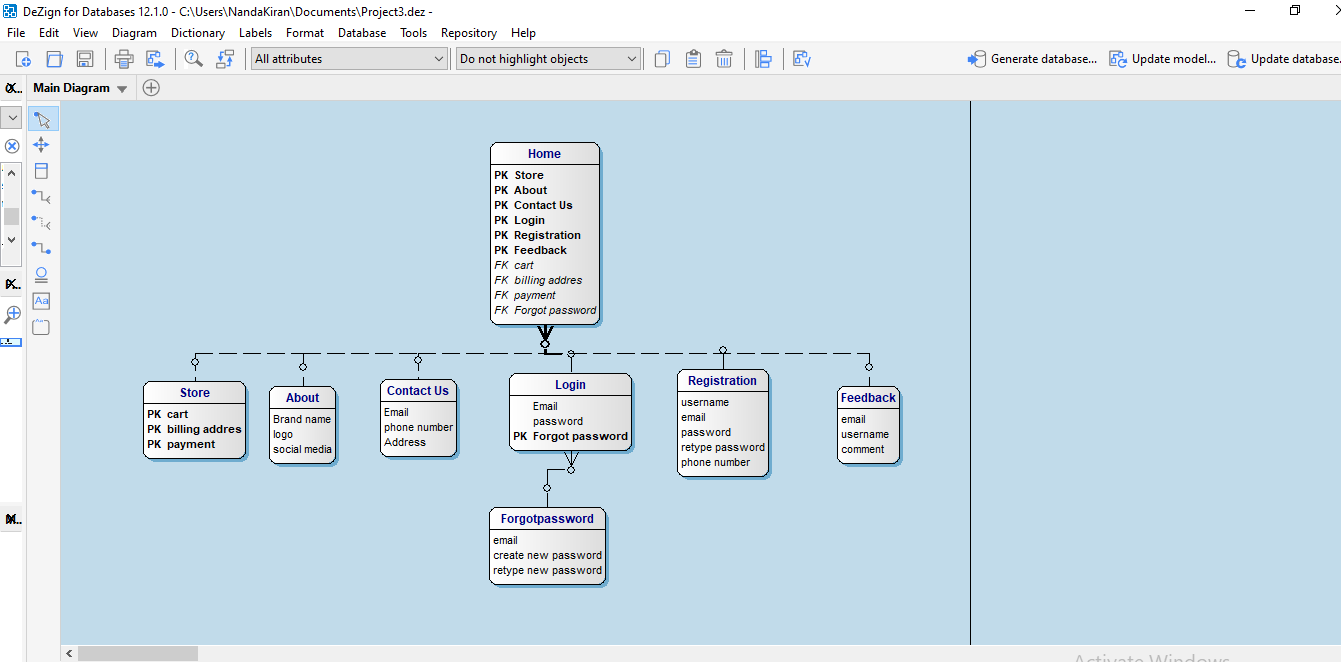
**FORGET PASSWORD**

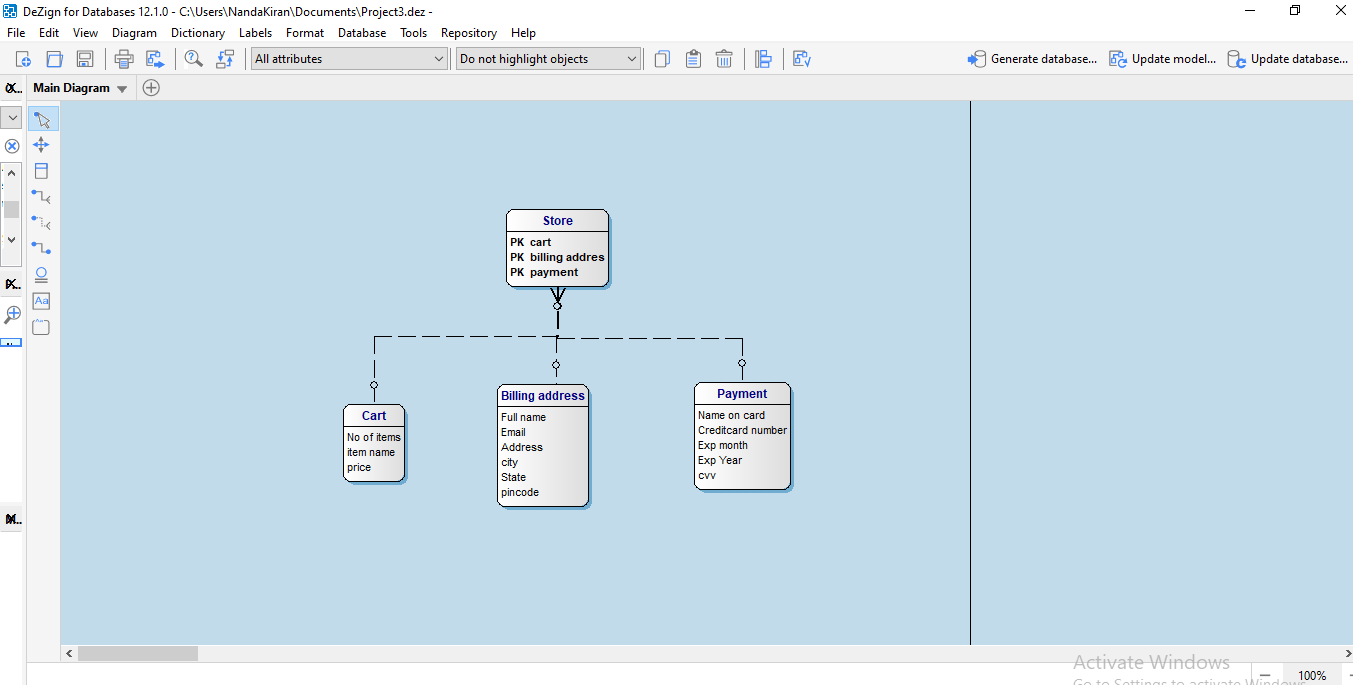
****

**Figure 5.8.Forget Password**

**43**

**5.4.E-R Diagrames**

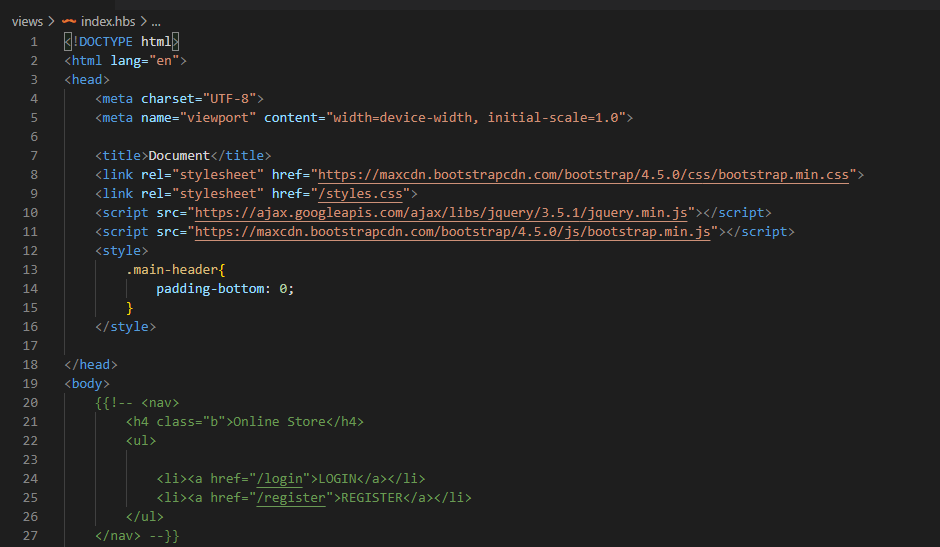
****

****

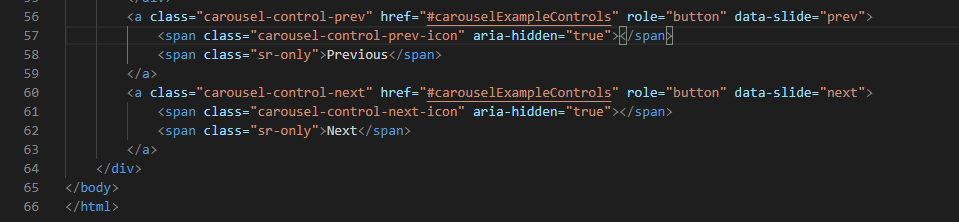
**44**

5.5 Screenshots of the project code

Index.hbs

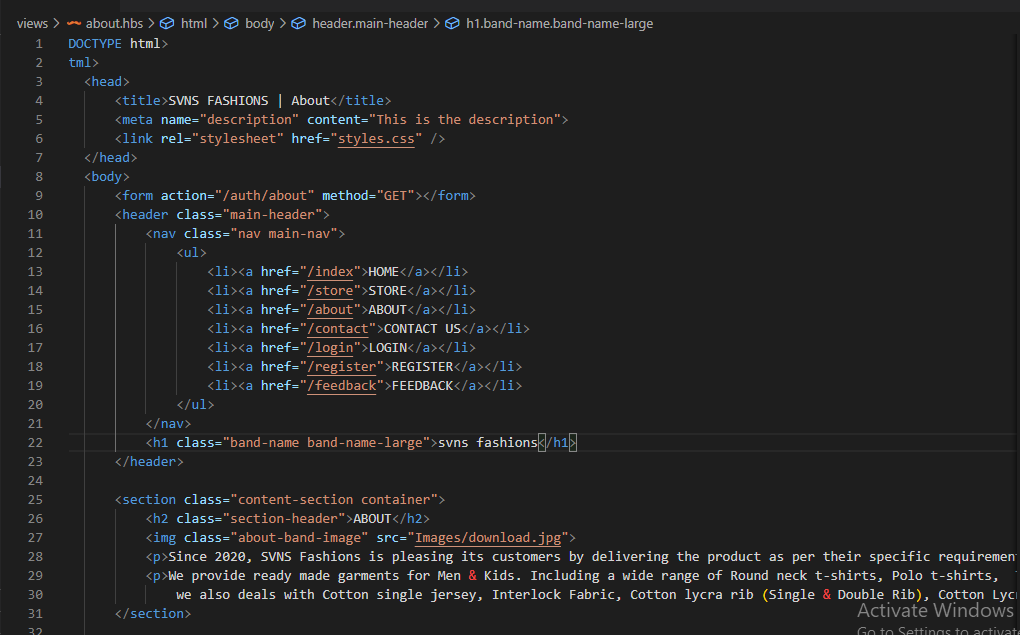


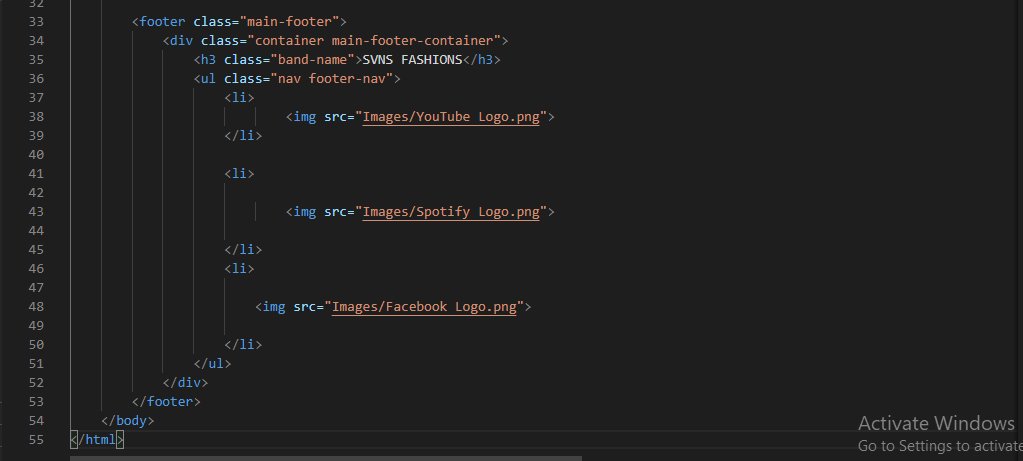




45

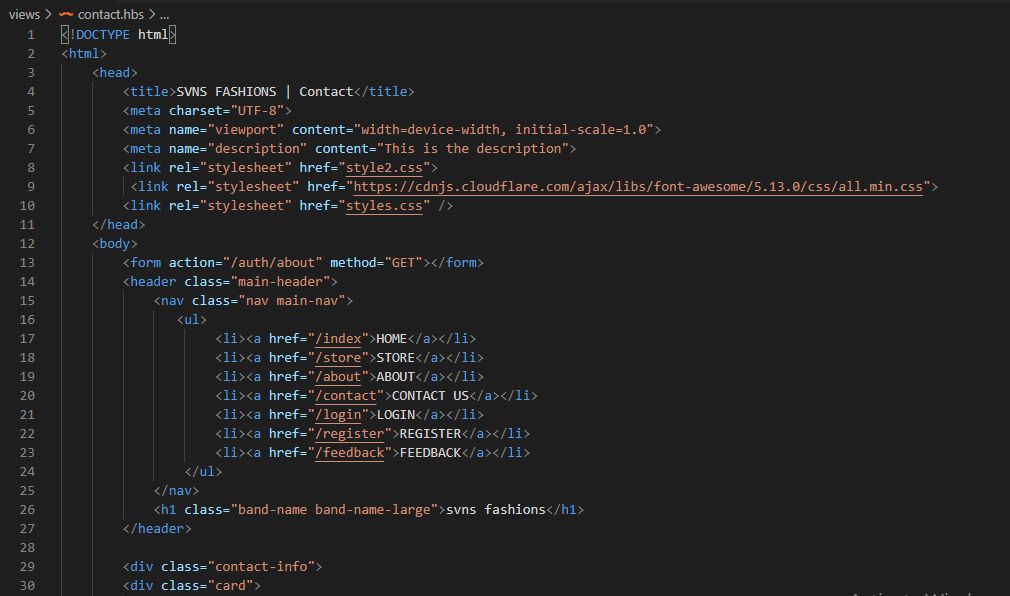
About.hbs

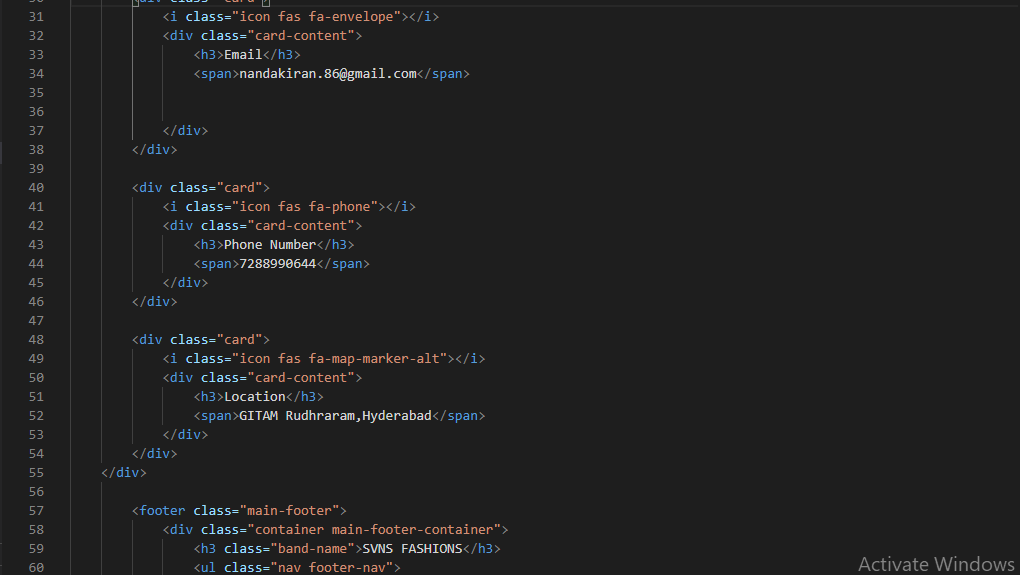


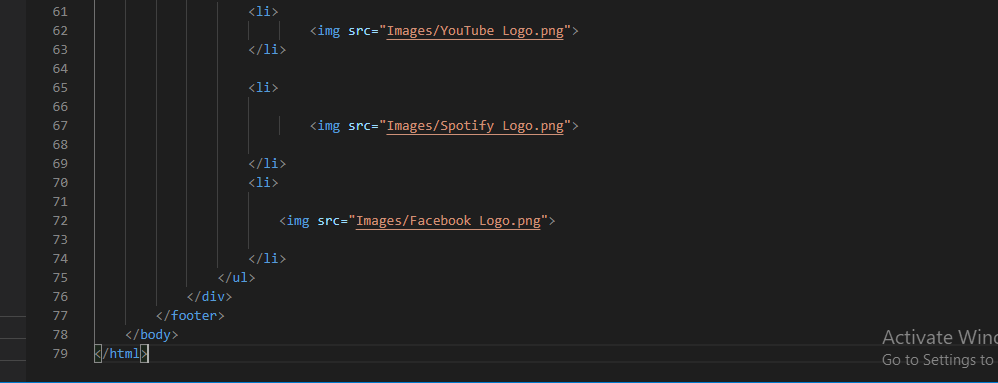


46

Contact.hbs

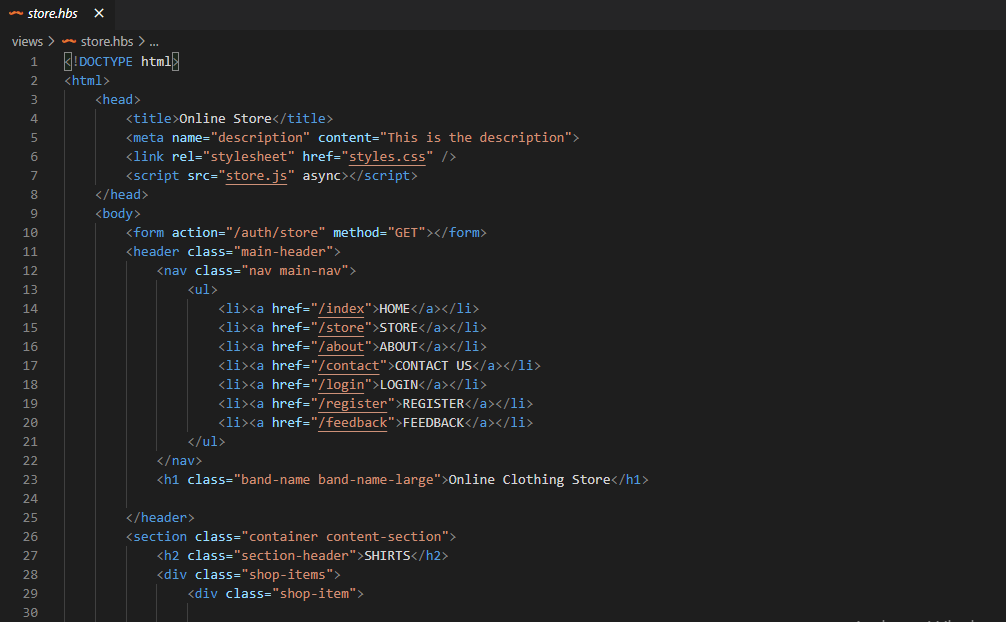


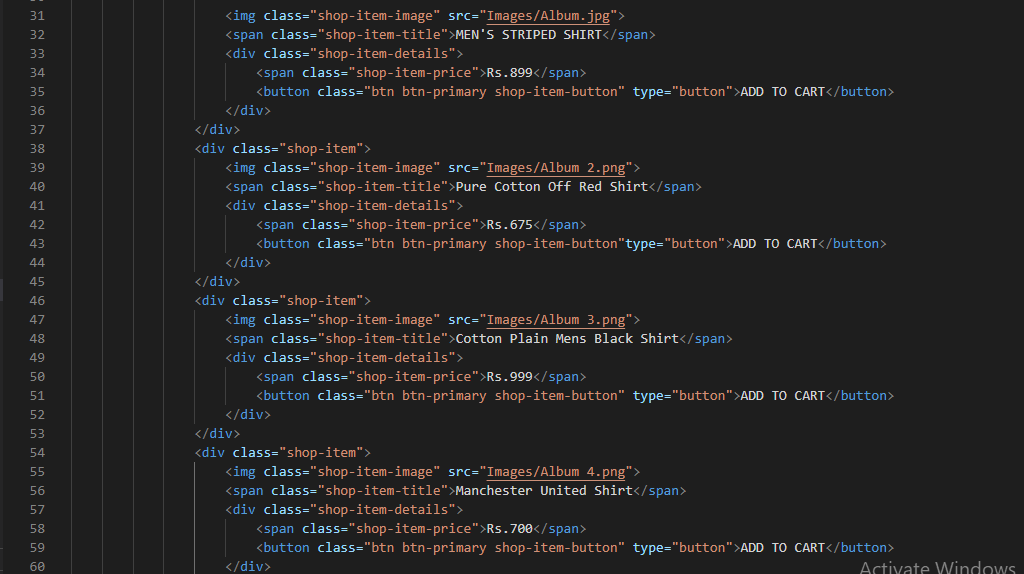




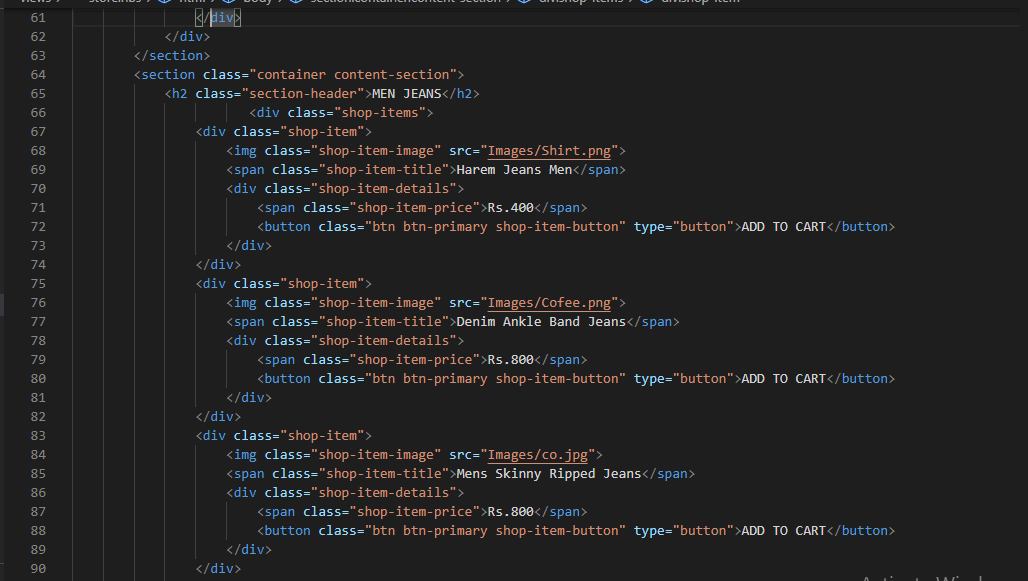
47

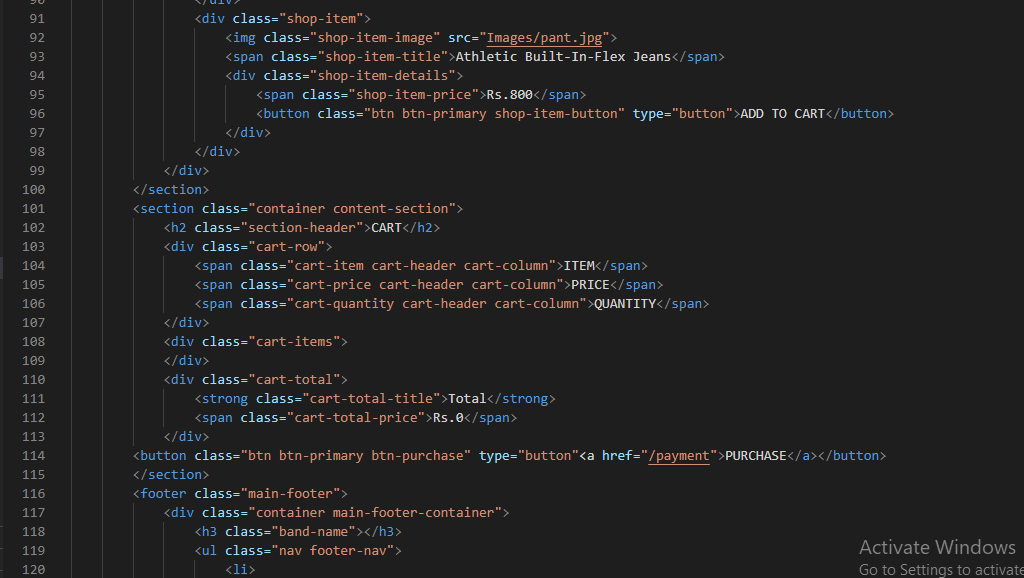
Store.hbs





48

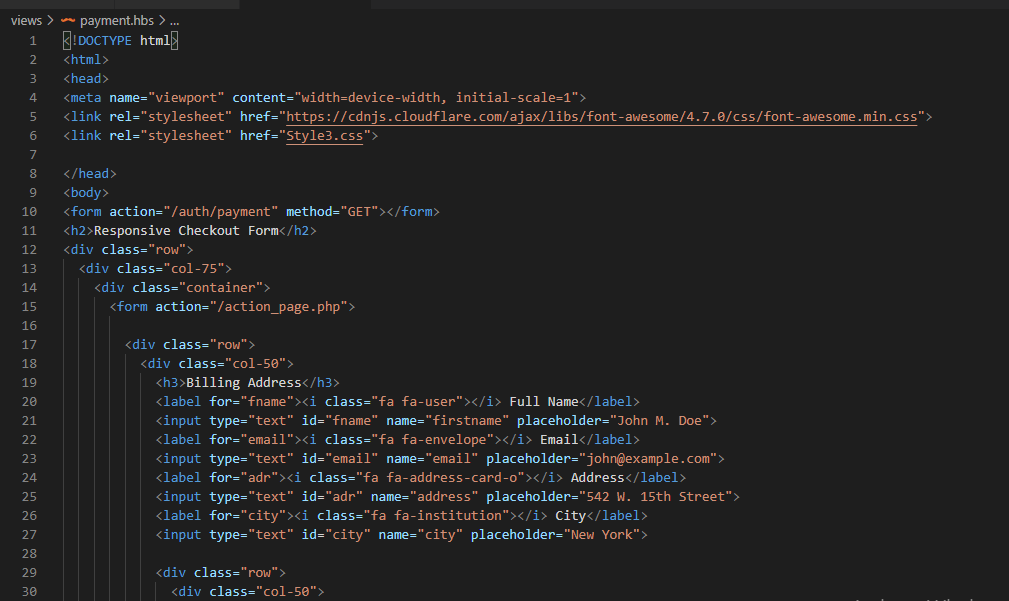




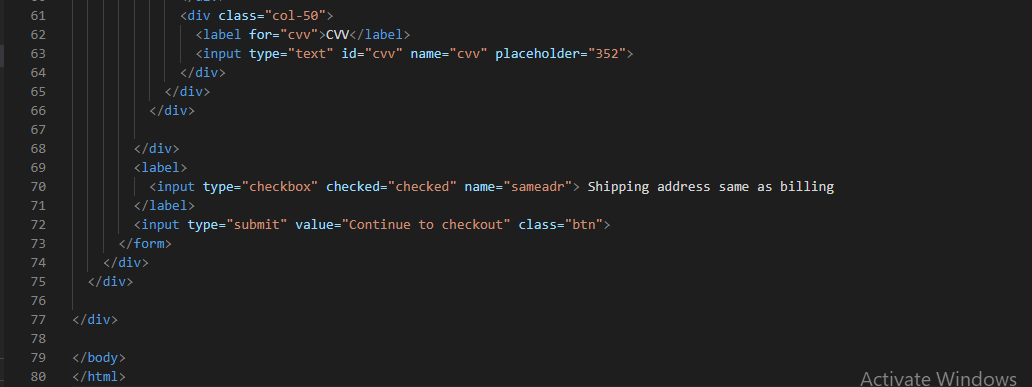


49

Payment.hbs

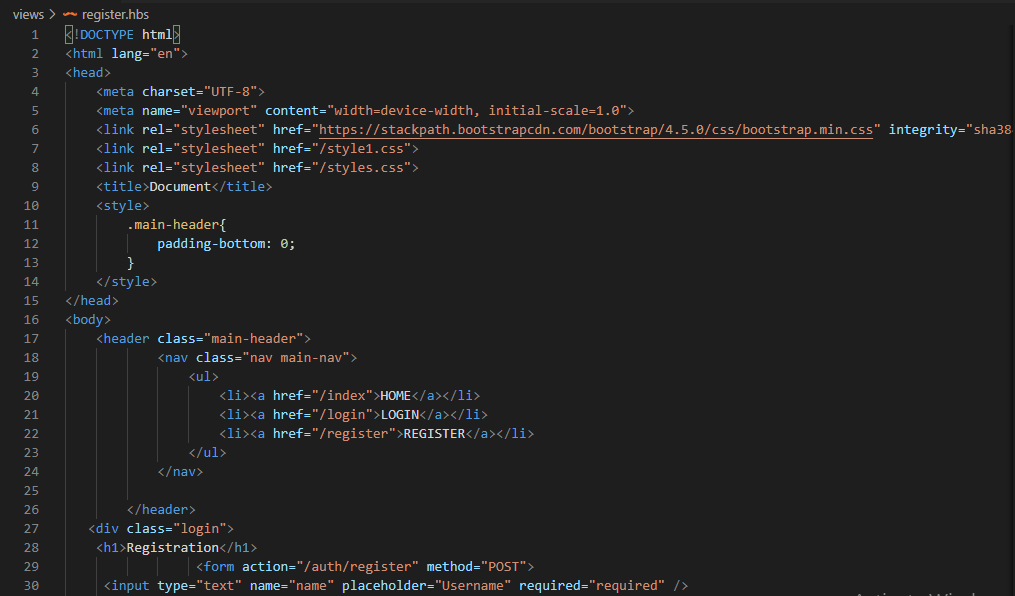


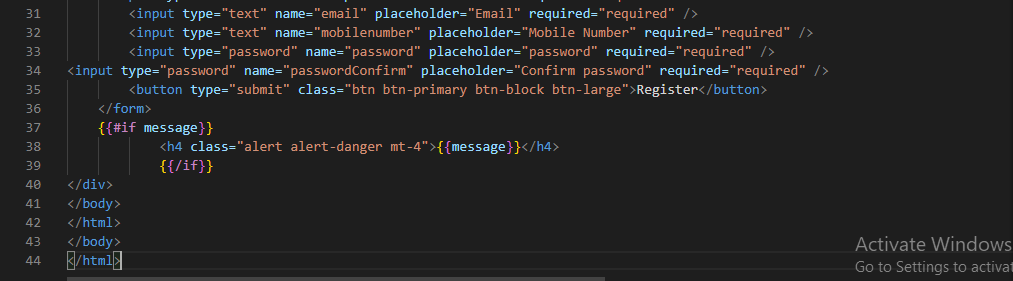




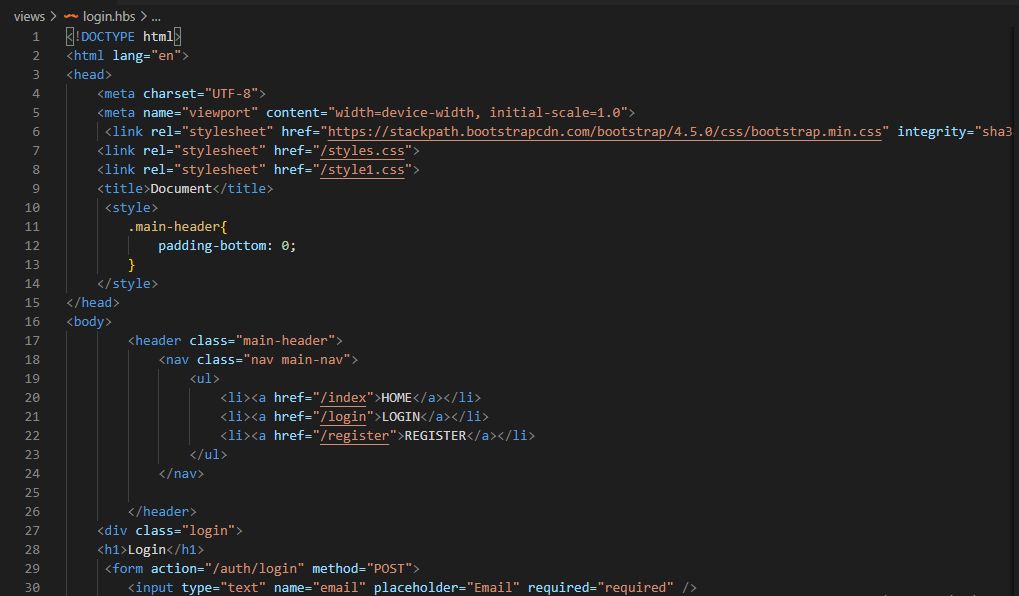
50

Registration.hbs

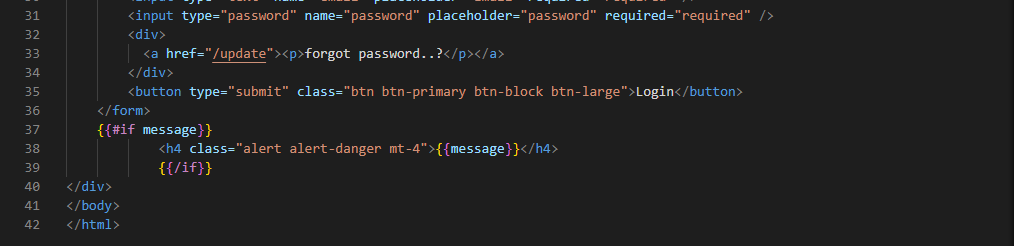




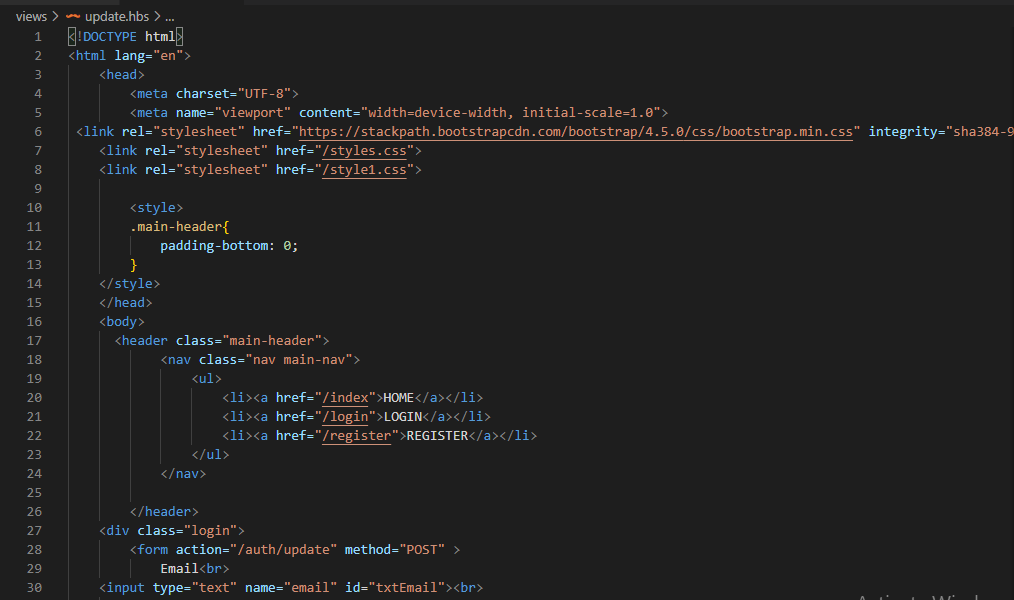
Login.hbs



51



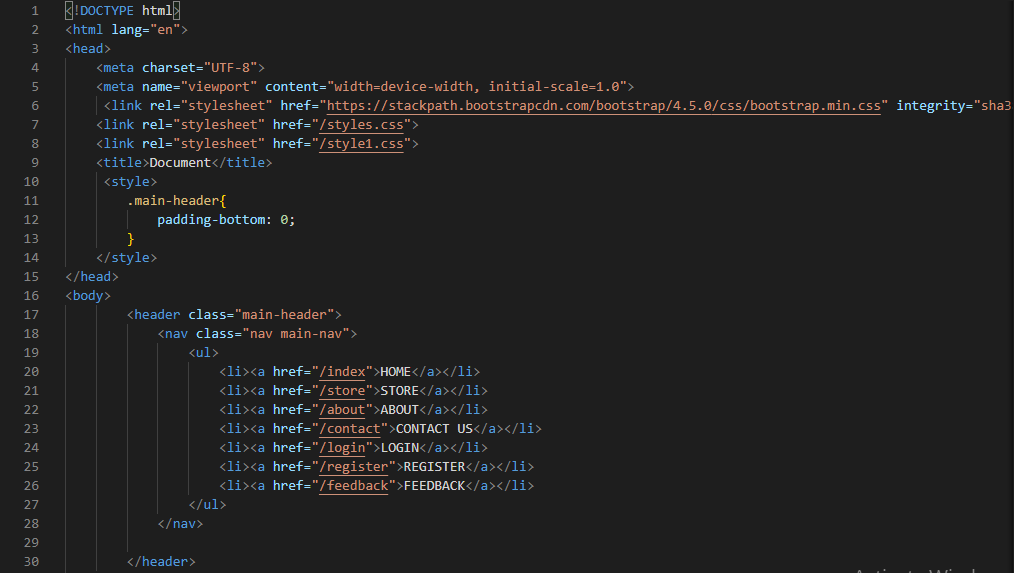
Update.hbs

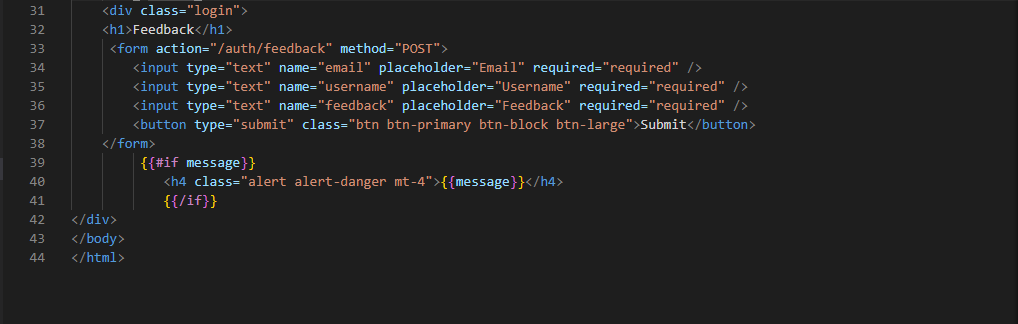




52

Feedback.hbs

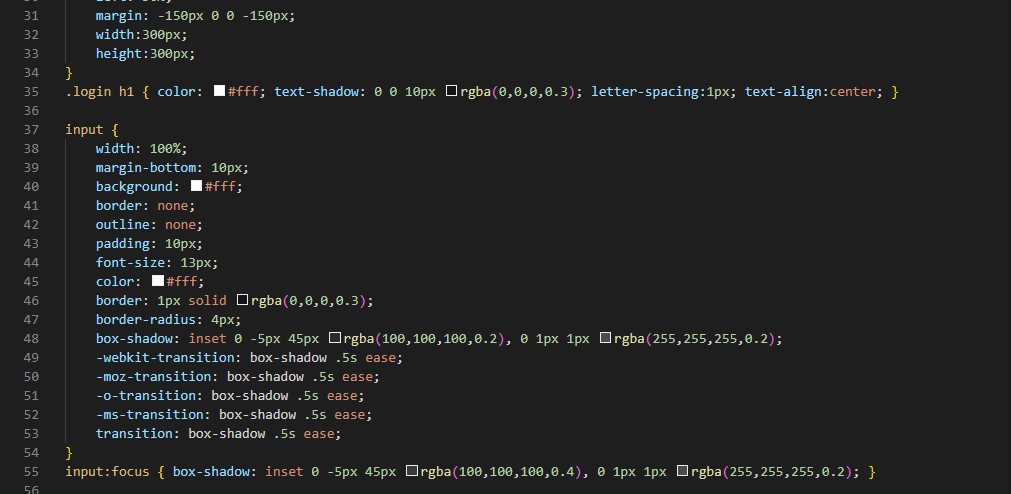




53

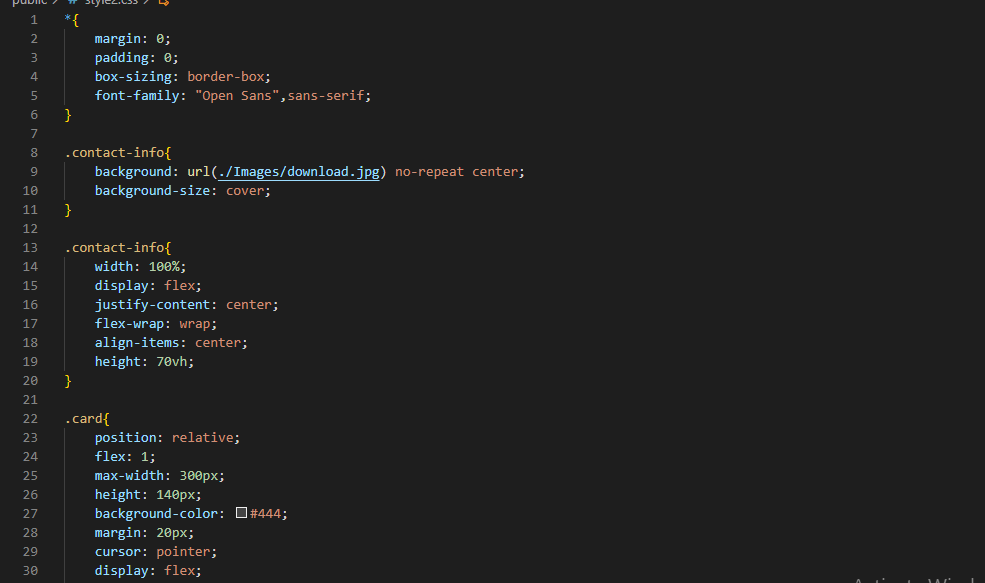
Style1.css

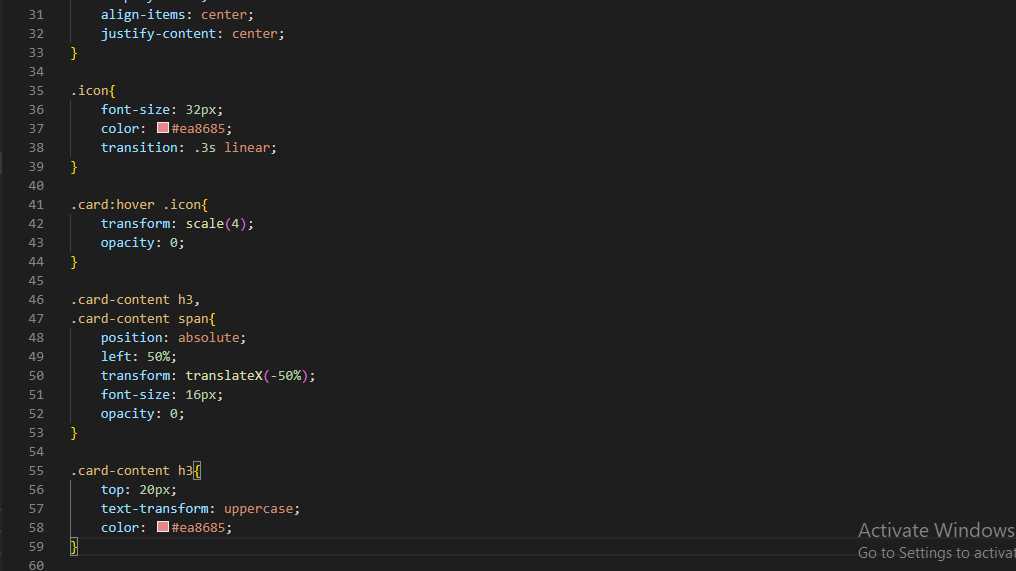


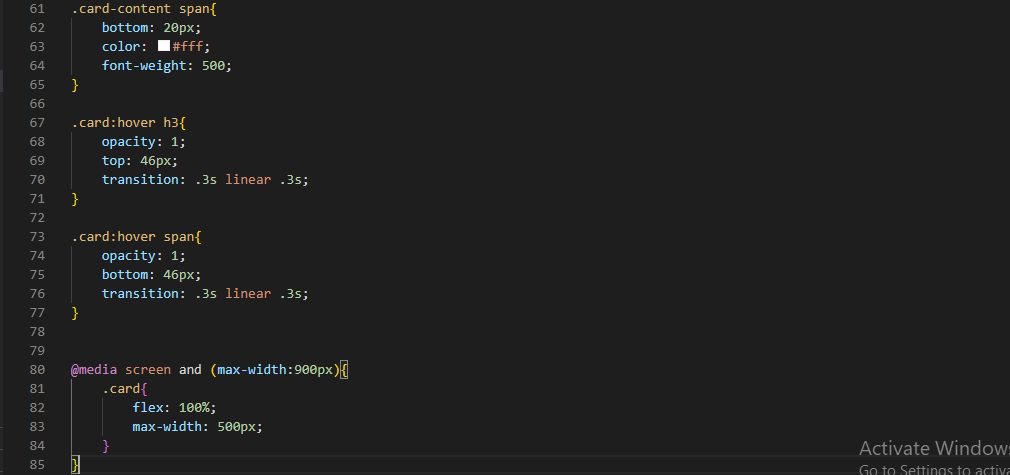


54

Style2.css

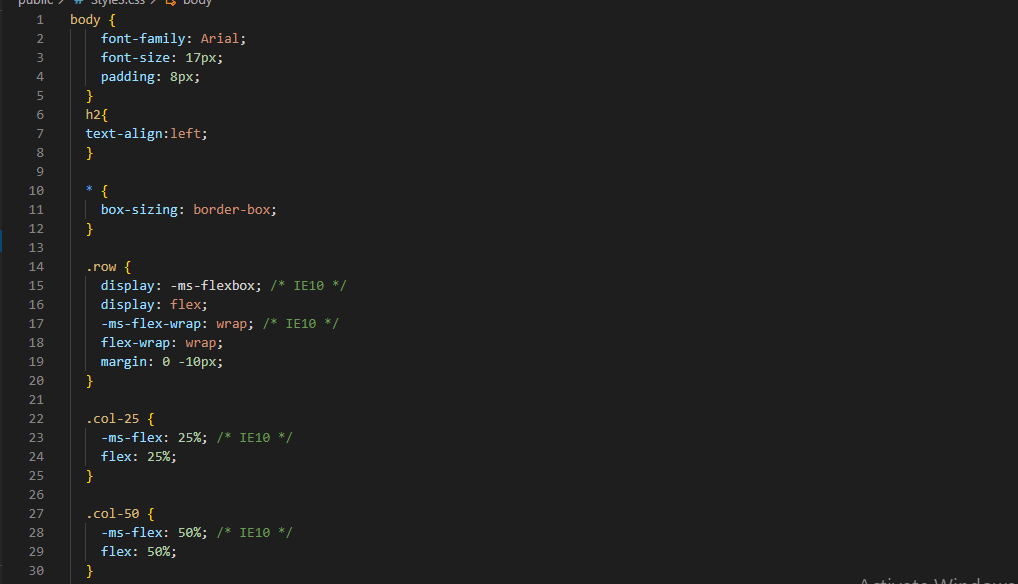


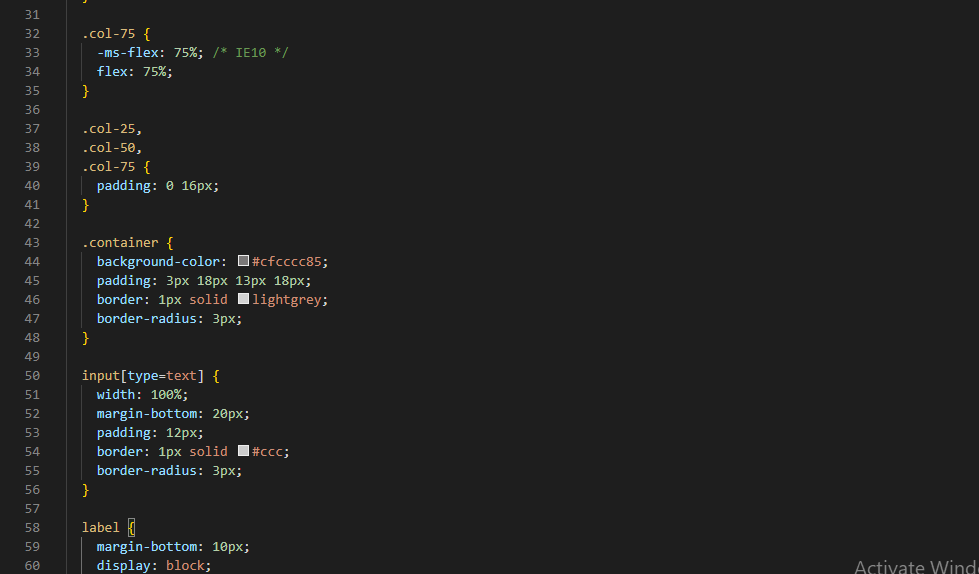




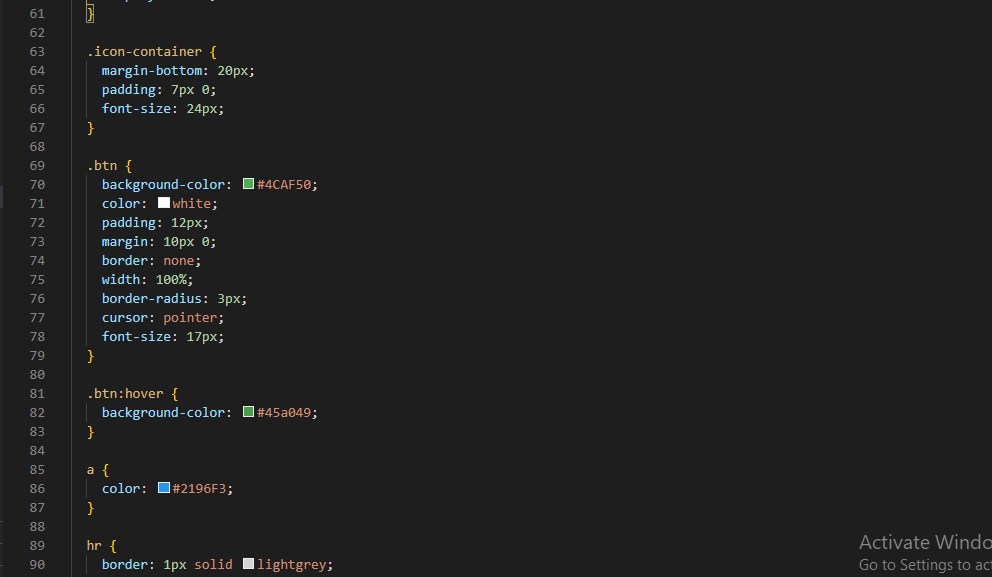
55

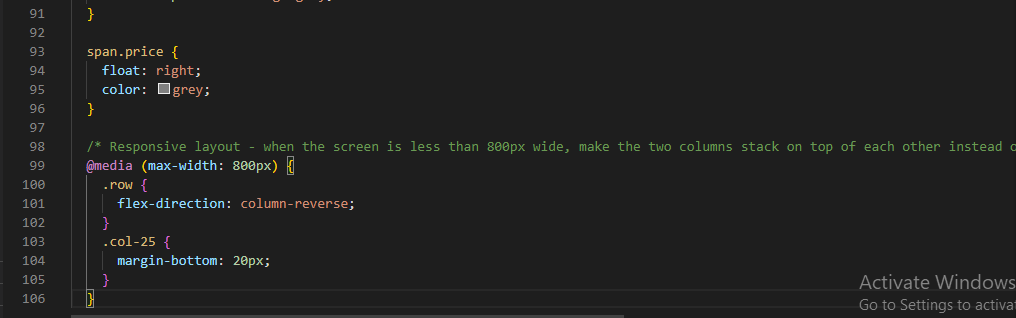
Style3.css



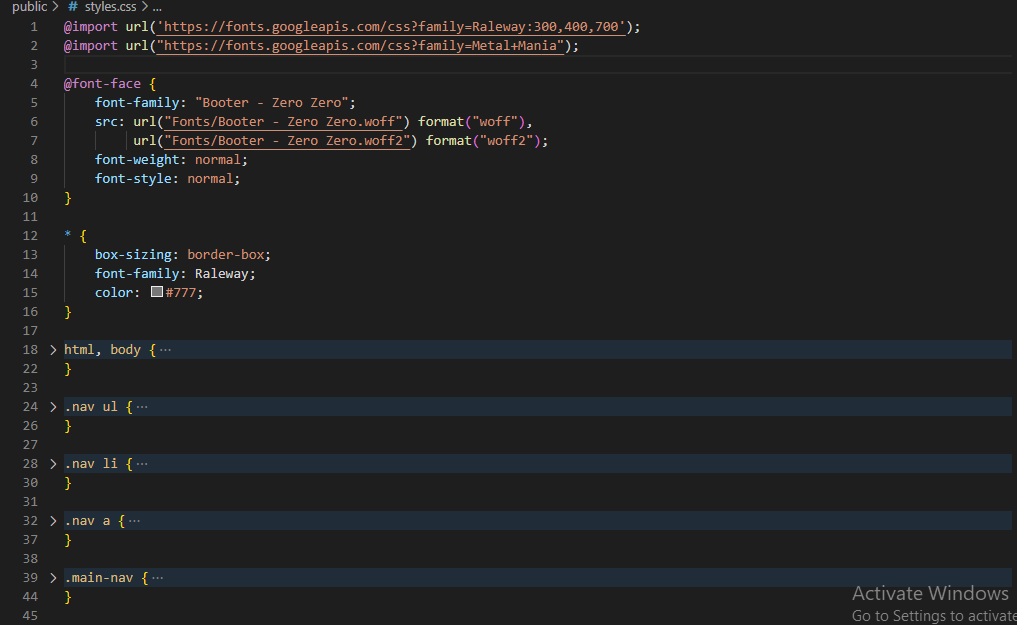


56





Styles.css

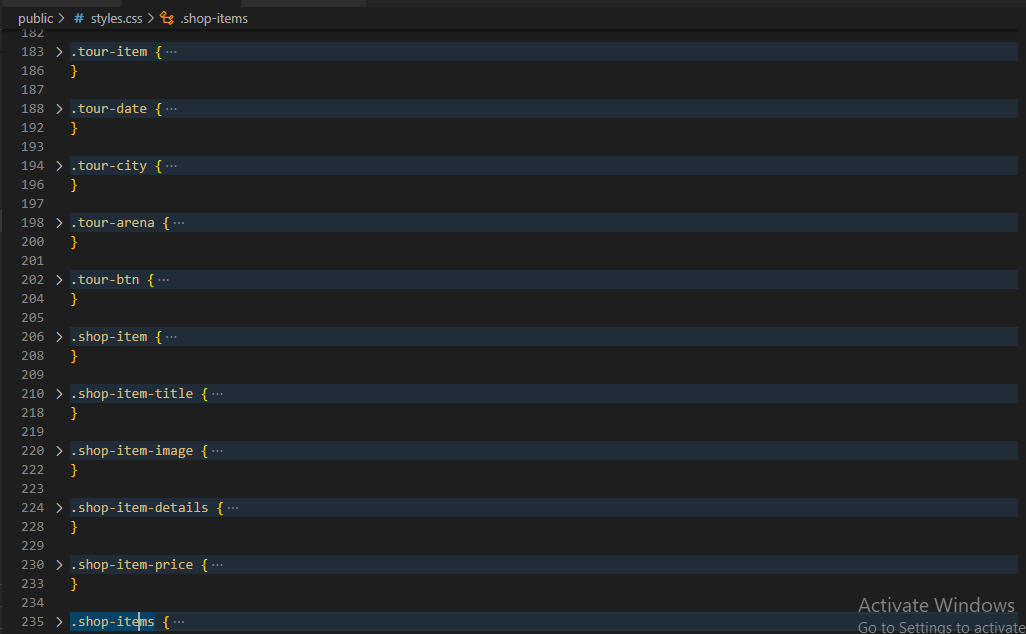


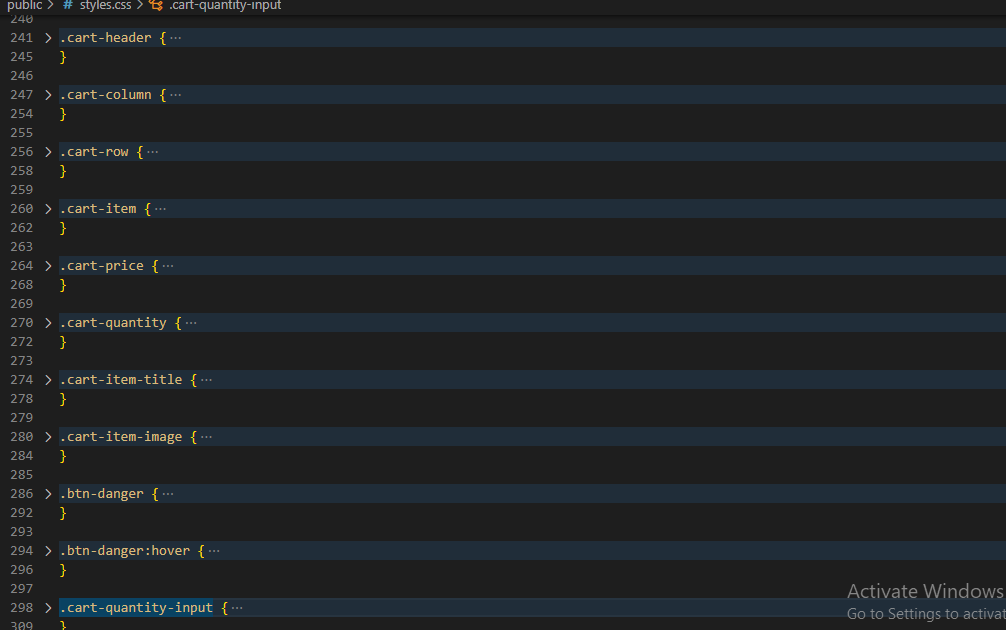
57

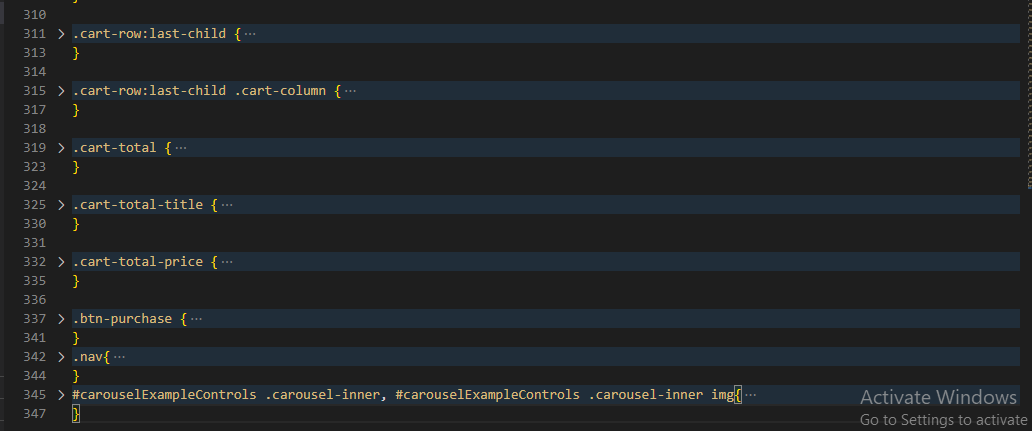




58



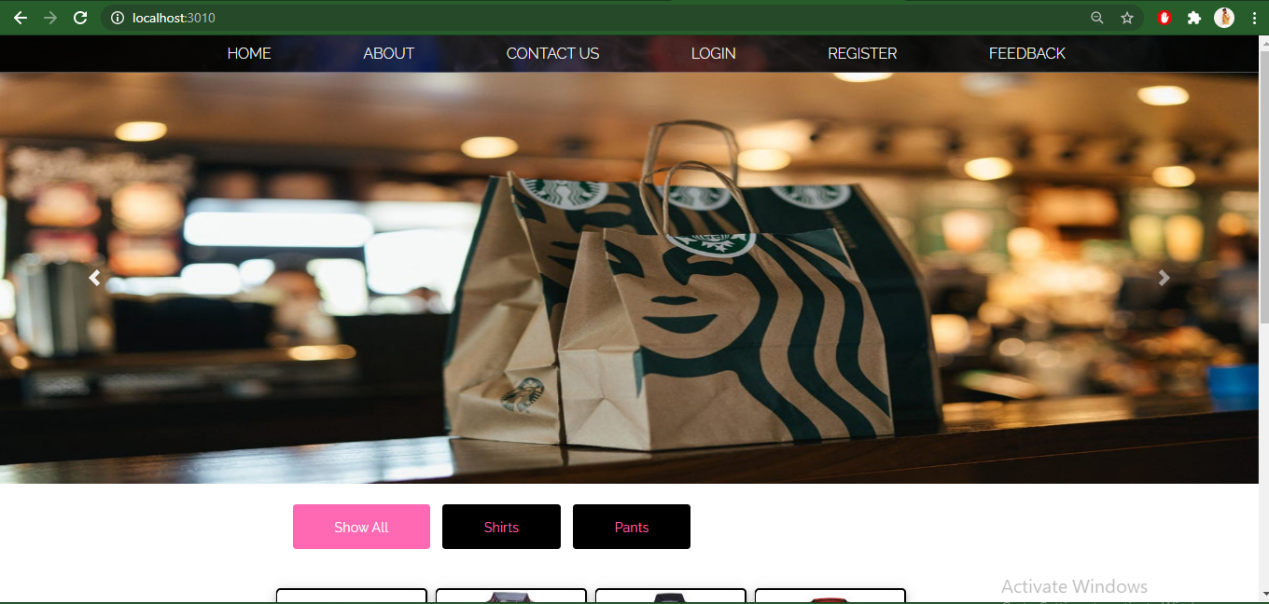


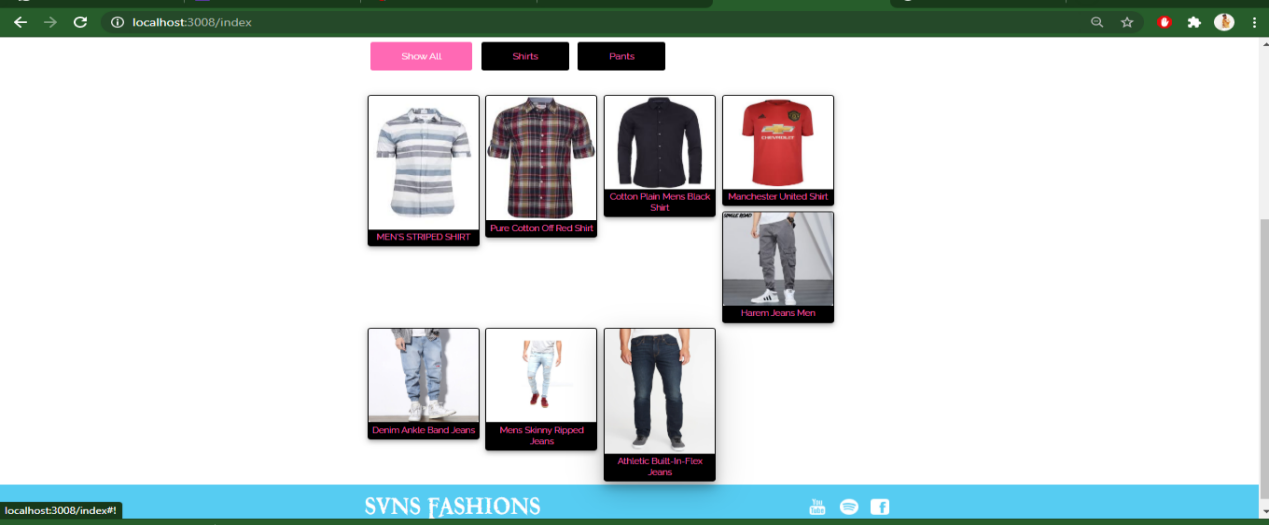


**59**

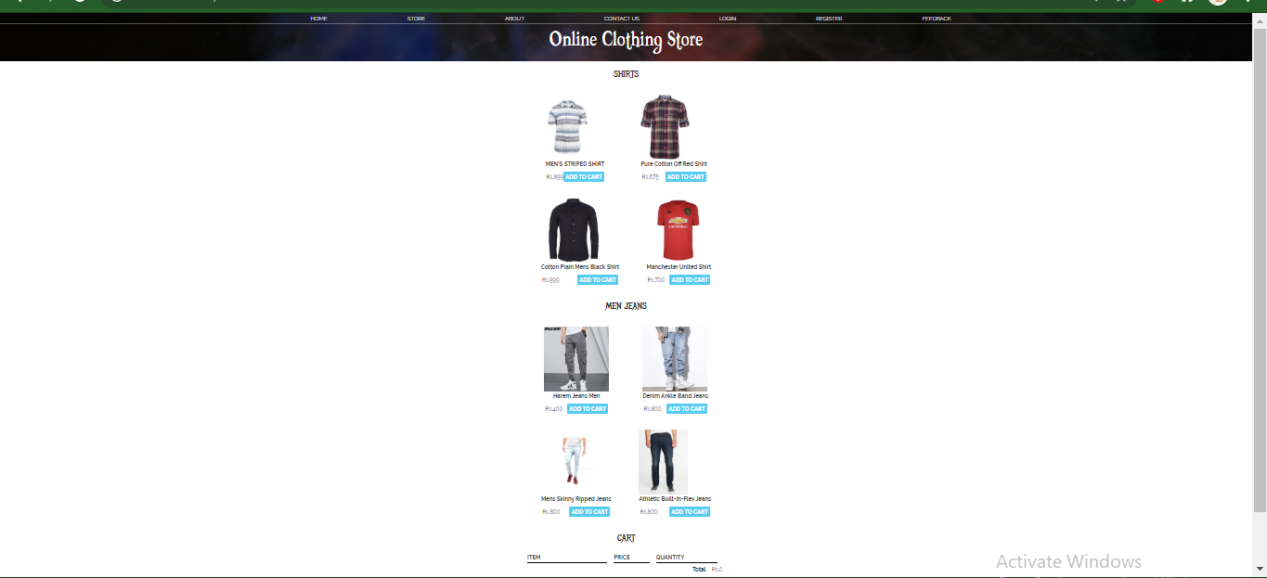
5.6 Screenshots of the project

**HOME**

****

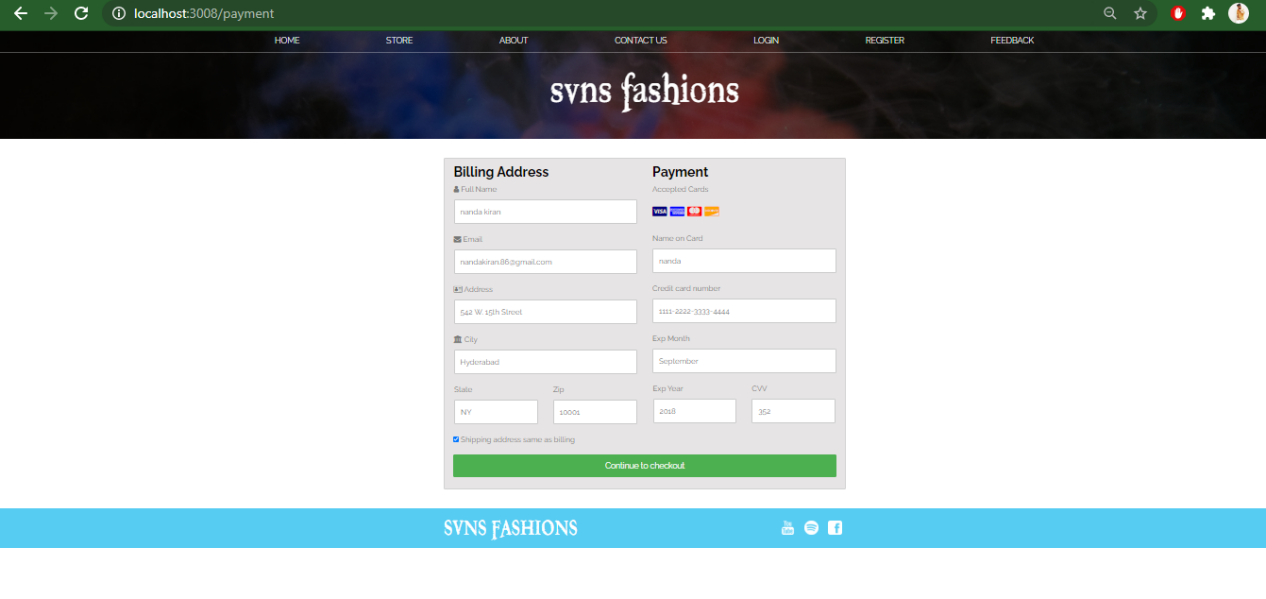
****

**STORE**

****

**60**

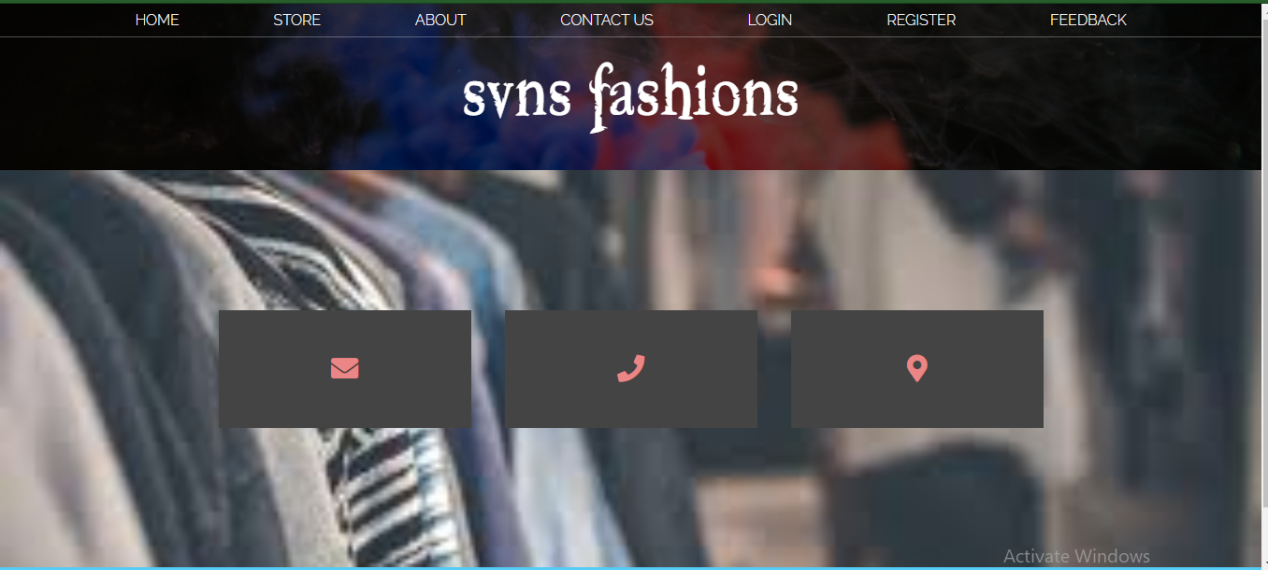
**PAYMENT**

****

**ABOUT**

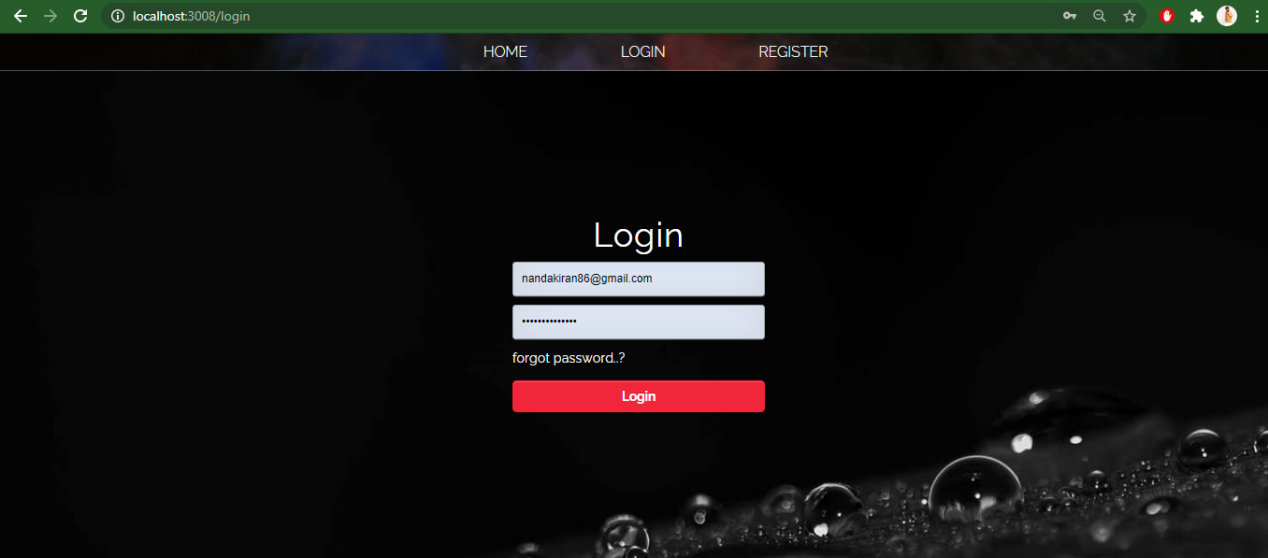
****

**CONTACT US**

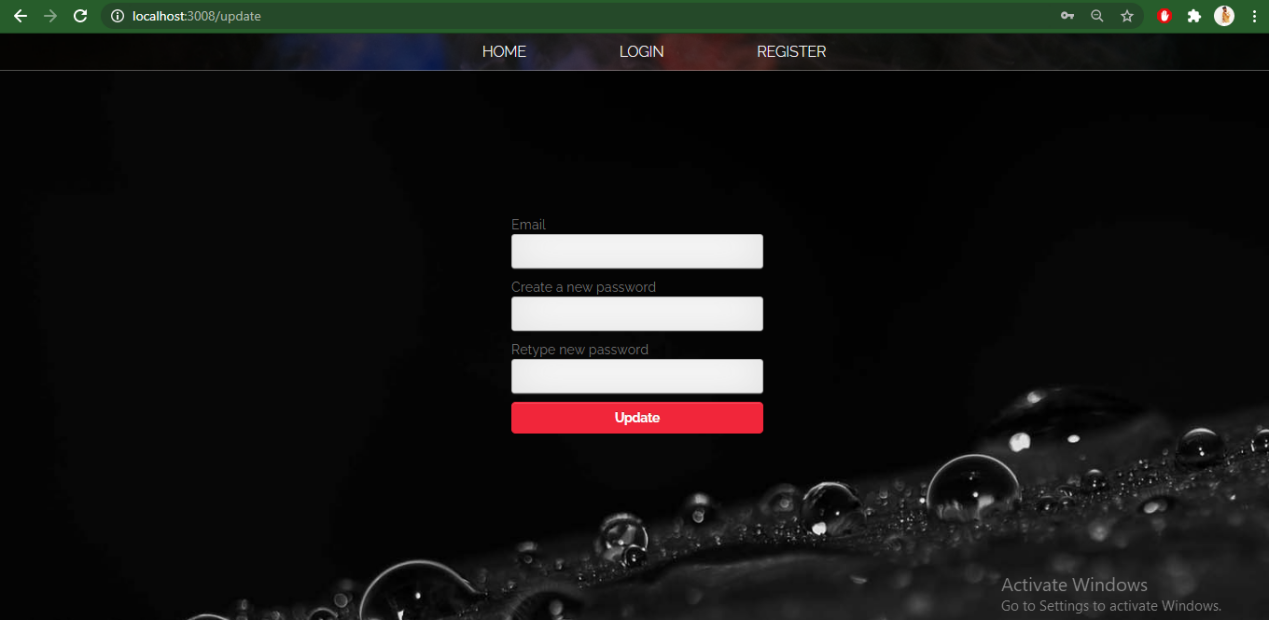
****

**61**

**LOGIN**

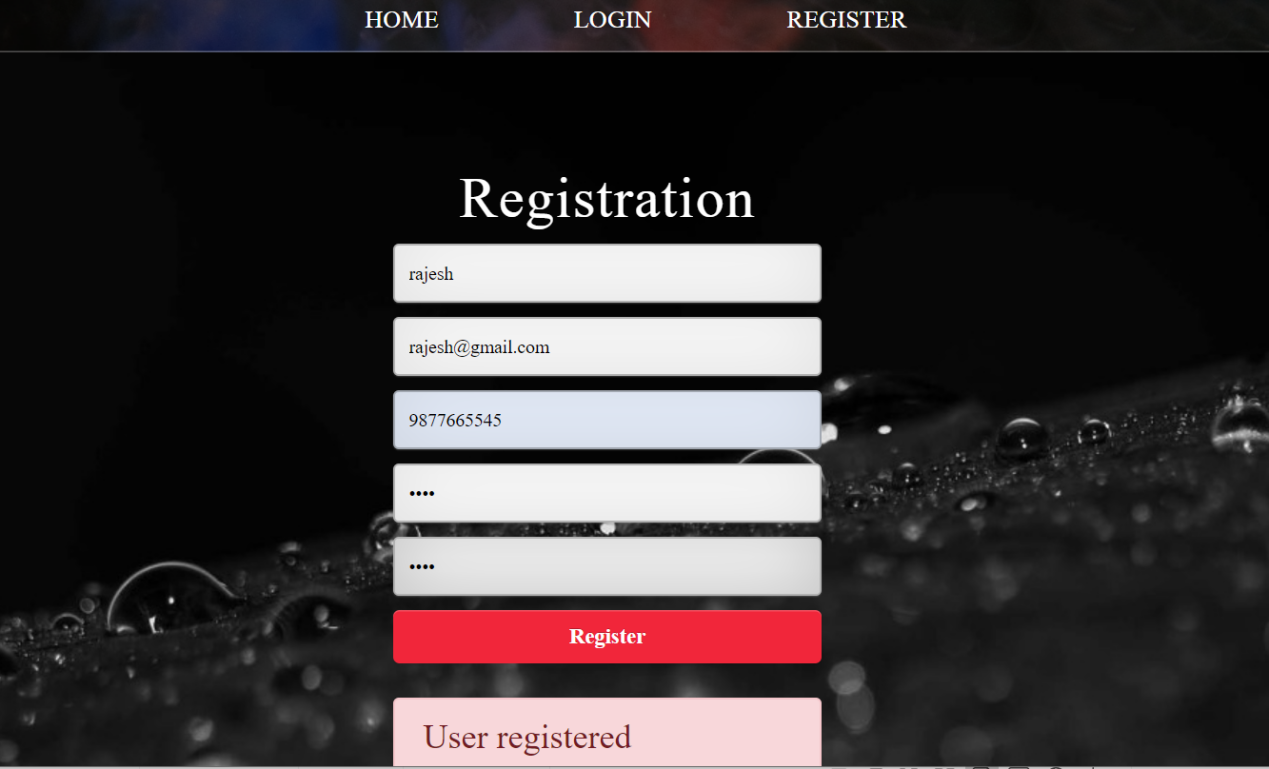
****

**FORGET PASSWORD**

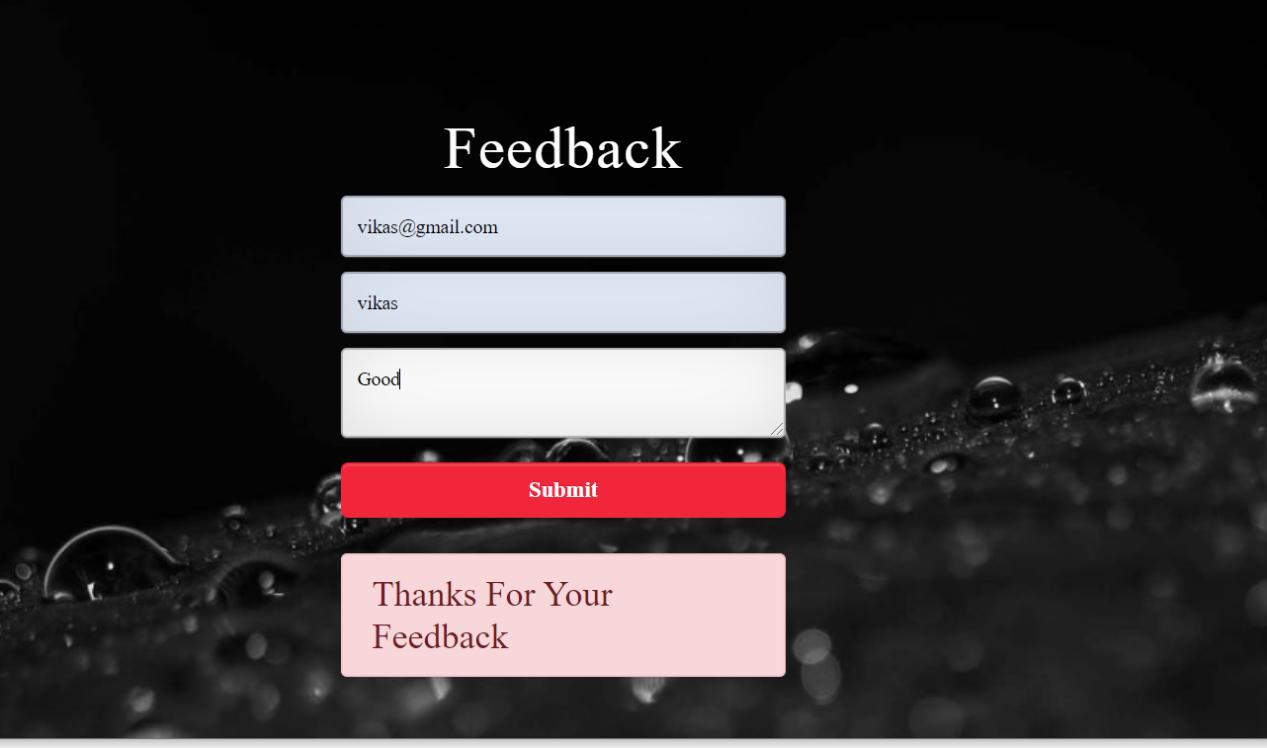
****

**62**

Registration



**Feedback**



63

**CHAPTER 6**

**MAINTENANCE**

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

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

For maintenance of the website:

1. The database has to be updated regularly according to new available information.
2. Redundant and false information must be removed from the database.

3. Newer version of MYSQL can be used for up gradation of website and to improve the overall performance of the system.

64

**CHAPTER 7**

**FUTURE SCOPE & FUTURE ENHANCEMENT**

**PROJECT NAME**: **E-Commerce Cart Application**

1. E-Commerce cart would help each and every person to find any clothing via our website and get it at home it will save their time.
2. It would provide huge collection of Dresses of all Ages.
3. Students will also get audio/video series of courses, i.e. they can learn online/offline.
4. We will be providing some special courses both online and offline.

65

**CHAPTER 8**

**CONCLUSION**

We have successfully implemented the site ‘**E-Commerce Cart Application’.**

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

**BIBLIOGRAPHY:**

1. www.jquery.com
2. [www.w3schools.com](http://www.w3schools.com)
3. [www.getbootstrap.com](http://www.getbootstrap.com)
4. [www.fontawesome.com](http://www.fontawesome.com)
5. Learn HTML and CSS faster(Mark Myers)
6. [www.w3scools.com](http://www.w3scools.com)
7. [www.codepen.com](http://www.codepen.com)
8. YouTube
9. Wikipedia

**66**