ECommerce Website – Angular

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Ishan Gupta (161261)

Under the supervision of

(Mr. Bobin Sondhi)

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat,
Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **" ECommerce Website – Angular "** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from Feb 2020 to May 2020 under the supervision of **Mr. Bobin Sondhi**.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Ishan

(Student Signature)

Ishan Gupta, (161261)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Sondhi

(Supervisor Signature)

Mr. Bobin Sondhi

Dated: 26/05/2020

ACKNOWLEDGMENT

First and foremost, I would like to express my gratefulness to Head, Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology and Mr. Pankaj Kumar, Training and Placement Officer, Jaypee University of Information Technology for providing me the opportunity to work with such passionate organization like **Paxcom India Pvt Limited** in my final semester of bachelors course. As always said hard work is the key to success but it's not just hard work that pays off but what matters most is the right guidance and also the right attitude to grab the opportunity. It gives me immense pleasure to express my gratitude and respect to Mr. Bobin Sondhi for always being a great manager and support whenever required. I would also like to thank entire team for not only imparting their knowledge but also their constant supervision, advice and guidance whenever required, without which this internship wouldn't have been possible.

I would also like to thank all other department faculty at Jaypee University of Information Technology. Not only did they taught me and made me capable enough to undergo this industrial experience but were always there at the need of the hour and provided with all the help, facilities and co-operation, which was much required to reach this level.

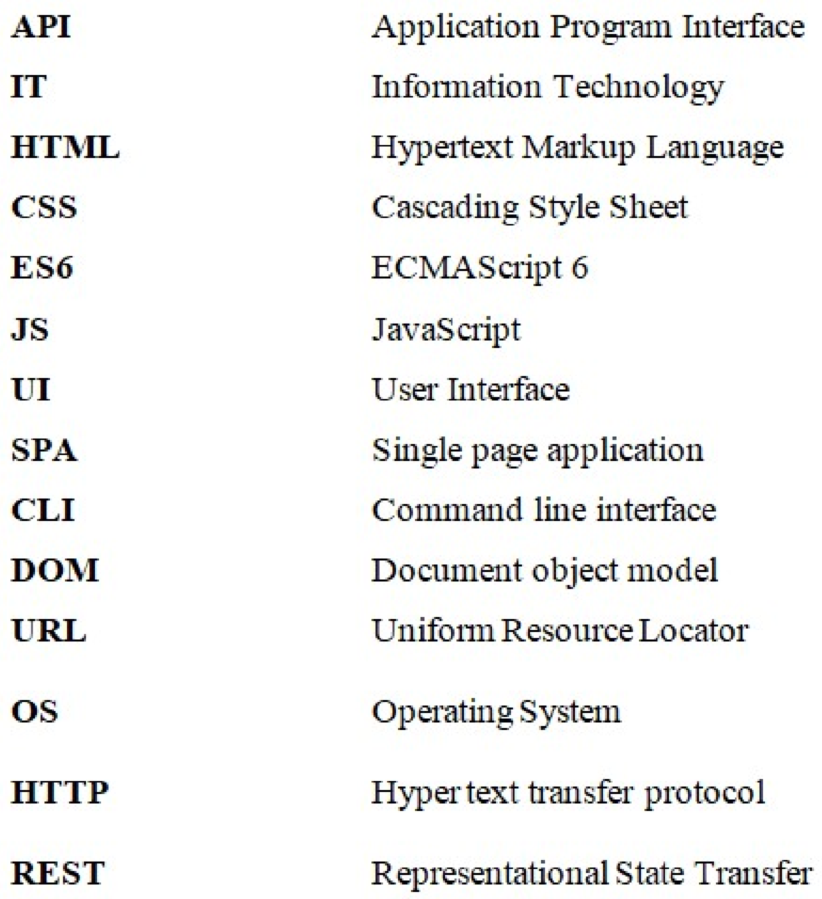
Last but not least I would like to thank my parents, who taught me the value of hard work by their own example.

TABLE OF CONTENT

TABLE OF CONTENT

Chapter No **Page No**

ABBREVATIONS LIST



API	Application Program Interface
IT	Information Technology
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
ES6	ECMAScript 6
JS	JavaScript
UI	User Interface
SPA	Single page application
CLI	Command line interface
DOM	Document object model
URL	Uniform Resource Locator
OS	Operating System
HTTP	Hyper text transfer protocol
REST	Representational State Transfer

LIST OF FIGURES

ABSTRACT

Innovative headways have affected the eCommerce world, changing the manner in which customers connect with items and empowering them to purchase all the more productively. With the comfort of getting items conveyed to your entryway, eCommerce has now become a piece of regular day to day existence. Clients can purchase things from their homes, their or working environment according to their comfort. Today web shopping has made it simpler and helpful for the client as it spares their time and exertion. Clients can check an assortment of things, can without much of a stretch drop exchanges and As the present innovation has gained a quick and speedy advancement of applications and website pages, so front end includes a fundamental layman design for clients to utilize the highlights and capacity making everyday work simple. Remembering this, utilizing the essential innovation of Front End Development an Ecommerce Website in Angular was planned as a major aspect of a quarter of a year of preparing at Paxcom India Pvt Ltd, Gurgaon.

As a development trainee student we exceeded expectations in JavaScript, TypeScript, Angular and Native Scripts alongside important GIT and Routing conventions. Spilling of live venture and conveyance of items to clients incorporated wide information on DOM control and specialized execution of the front end advancement dialects and apparatuses learned.

CHAPTER – 1 INTRODUCTION

1.1Introduction

Advancements in technology have a huge impact on the e-commerce world, which has changed the way consumers nowadays interact with products which enables them to buy the products more efficiently. With the ease of delivering products to your door, e-commerce nowadays has become an integral part of a person's everyday life.

Today e-commerce organizations expected to regularly update themselves with the latest technologies with the help of effective strategies, competitive marketing, and skilled laborers aside. These advancements in technology have helped people to meet their shopping needs easily, and as a result, it helps in growth and strengthing of the e-commerce sector.

Nowadays e-commerce is faster providing a smooth shopping experience to its customers, using the latest technology buyers can find anything with just the click of a button. Also, customers can keep track of their orders, find the best deals, check out the latest offers, and much more. With all this development, new business opportunities are emerging, which is a boon to the e-commerce sector.

Fig 1.1: Omni channel

Modern methods of purchasing include the process of channel research, speculation, and acquisition. According to the latest data approximately 90% of customers want a smooth and interactive experience across multiple channels and screens, highlighting the importance of Omnichannel in today's business.

Using the right technology means giving consumers not only what they want when they want it, but also when they want it.

*1.2**Problem Statement***

The framework we choose was Angular 9 together with the MDC web. This was a great idea but the challenge here was that the framework we were going to learn was completely new and developing a whole new product from scratch was also a big challenge. The entire project was divided into three modules and each of the interns was working on a different module. This was a team task and the main problem which we faced as a team was to learn how to work in a team task. The application was regularly synced and updated using Git. Before starting the project development we planned the different phases of the project, its database, backend server to be used, project flow, etc.

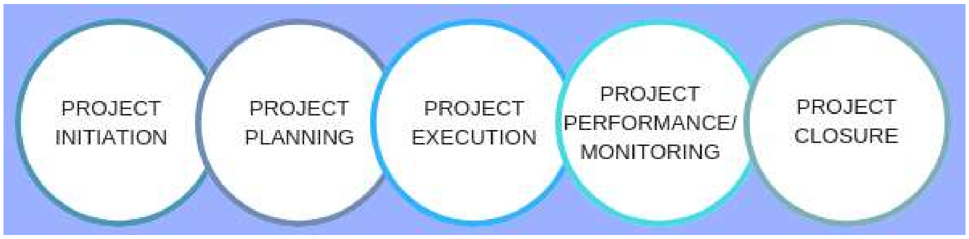


Fig 1.2 Phases of Project Management

1.3Objectives

The basic objective of this project was to get familiar with concepts such as:

- User Authentication
- Angular Modules
- Angular Components
- Angular Services
- Use of database
- Http Requests – GET, PUT, POST, DELETE
- Auth Guards in Angular
- Dynamic Header
- Angular Http Interceptor
- Angular Material
- MDC Framework

1.4Methodology

The various methodologies involved in this project are, frameworks such as Angular 9, NativeScript. The different technologies and languages used are HTML5, CSS3, Bootstrap4, JS, Angular material library, Angular Web MDC library. For interaction with database JSON Server is used. The code was regularly updated and synced with git.

1.4.1 *Frameworks*

In the computer, the software framework is Abstraction software to supply standard functionality which will be selected by modifying the extra code written by the user, thus providing direct use of application software. It provides the user with a standard way of designing and deploying applications. It is a valid and efficient software platform that provides specific functionality as part of a larger software platform to provide performance improvements, different products, and software solutions. These advanced frameworks include support systems, compilers, toolsets, libraries, and API's that compile all the various components to start a project or program development.

To build and deploy any application software we use frameworks as they are universal in working and their standards are maintained. To develop different software-products, applications, etc frameworks reused for developing high-end applications providing a particular functionality as a part of high-end and large software platforms. Today there are various frameworks used while I am using Angular for frontend development.

1.4.1.1 Angular

Angular is an open-source web application framework designed by a group of people and companies and the Angular Team at Google. It is based on typescript which is superset of JavaScript and is rewritten by the same team that built AngularJS i.e. angular version 1.0. AngularJS was a popular client-side framework used by JavaScript engineers for a long time. Angular is nothing new but is an improved version Angular 1 i.e. AngularJS. All other angular versions are named as Angular 2, Angular 3, and so on. Angular 9 is the latest version.

Fig 1.3 : Angular Logo

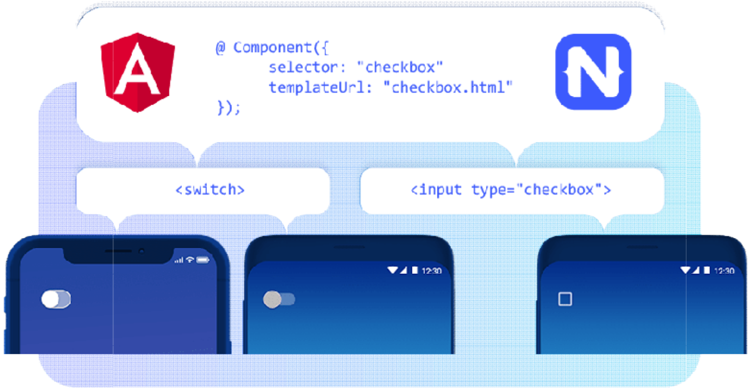
The latest angular version being used today is Angular 9.0. Version 9 moves all applications to use the Ivy compiler and run time by default. Angular 9.0 uses the Angular CLI to create a new project, generate new modules, components, services, and automatically links and declare the create modules, components, etc. The user needs to write one simple command, and entire work from creating a new component to declaring that component is done by Angular CLI. It also provides the user with an option to enable angular routing while creating a new project. If the user does not enable routing while project creation, routing can be enabled manually from within the project. Applications can be made browser and device friendly with the help of NativeScript that works with Angular and other frameworks.

As mentioned one among the feature of Angular CLI is code reusability, one code is often used for both mobile and desktop application development. Angular unlike AngularJS, works on the concept of Promises, Observables which make data fetch from API even faster and hence increase the performance of the SPA. Angular provides it's plugins for nearly every ide and hence also allows performing on a local server with an auto-refresh feature. As AngularJS it also extends the HTML vocabulary but now we make use of our components and also make use of several other existing components.

1.4.1.2 __NativeScript:__

NativeScript is an open-source cross-platform framework employed for building native iOS and Android apps using JavaScript, Typescript, Angular, or Vue.js. It makes use of Typescript when used with Angular and JavaScript also can be used for development, but just in case of JavaScript the developer should be well known worth XML. NativeScript with Angular integration has all the features of

Angular CLI from routing support to code generation via CLI, web packs, etc.



```
@ Component({
      selector: "checkbox"
      templateUrl: "checkbox.html"
});
```

```
<switch>
```

```
<input type="checkbox">
```

Fig 1.4 : Angular + NativeScript

Why we should use Angular together with NativeScript?

- Application Structure and Lumbering - Angular concepts such as binding of data, dependency injection, angular services, and routing to build native mobile apps are reused. Also code and skill is reused.
- Native Mobile Performance - Create mobile apps without any compromise using Angular that provides truly native UI and performance.
- Angular Community - Huge Angular developer community, to get extra help.

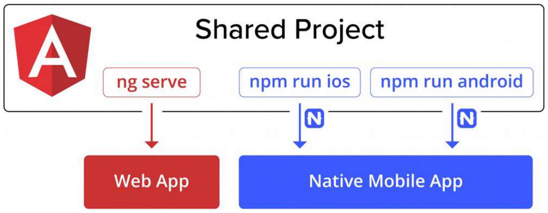


Fig 1.5: NativeScript with Angular

1.4.2 Technology and Languages:

As in regular everyday practice, to speak with one another dialect is a significant medium, correspondingly to cooperate with the framework; programming dialects are a significant vehicle for the engineer. As, all things considered, there, individuals speak with various dialects, correspondingly, for driving with gadgets there are various dialects accessible, for example, C, C#, C++, Java, Python, and so on.

The first and significant stage an individual or a group needs to perform before dealing with a particular venture is to choose a lot of advancements that will be utilized for building up the product or an application. For creating, the backend innovation stack incorporates web structures and after that the programming dialects come and servers to send the product application on the overall web.

1.4.2.1 HTML

HTML stands for Hypertext Markup Language. It is the standard markup language that describes the structure of a web page. It consists of a series of elements that tell the browser how to display the content and these elements are represented by tags. The browser doesn’t display the tags on the screen but uses the HTML tags to render the content of the web page. HTML is the root of any web-based application which is easy to learn and use, therefore it is not always considered as an important language. As many other applications work in a standard format, therefore it is very important to use the language in a standard format.

Fig 1.6: HTML 5 Logo

HTML5 is the latest evolution of the standard that defines HTML. The term represents two different concepts. It is a new version of the language HTML, with new elements, attributes, and behaviors, and a larger set of technologies that allows the building of more diverse and powerful Web sites and applications.

1.4.2.2 CSS

Cascading **S**tyle **S**heets (CSS) are used for designing and presenting the HTML page, describes how HTML elements are to be displayed on different devices, and directly controls the layout of site on multiple screen sizes. It manipulates the text colors, fonts, background colors, layout, etc. The latest flexbox and grid concept in css has made it easier to layout the document. CSS can be embedded in HTML in three ways, i.e. inline css, internal css and external css. To add css internally `<style>` tag is used. IT is recommended to make an external css file as it keeps the code clean and makes it understandable.

```
body {
  color: #000;
  font: 16px Times, serif;
}
h1 {
  color: #f09;
  font-size: 24px;
}
```

Fig 1.7: CSS illustration

1.4.2.3 JS

JavaScript (JS) is a structured language that is lightweight, translated or has recently been integrated into functional tasks in the first grade. Js is popular and well known for writing Web pages; many non-browser sites also use it, as Node.js, Apache-CouchDB, and Adobe Acrobat. It is a style-based, elementary, series, language based, supported object based, essential, and dictionary (e.g. functional programs). Toda JavaScript latest level is ECMAScript. ECMAScript 5.1 is supported by all modern browsers and at least ECMAScript 3 is supported by older browsers. With the help of JavaScript one can manipulate Dom from manage static forms to dynamic, the language is so vast and widely used that it is not only supported by all browsers but also helps debugging it.

```
<script>
if(age > 19){
    alert("Adult");
} else{
    alert("Teenager");
}
</script>
```

Fig 1.8: JavaScript illustration

1.4.2.4 Bootstrap

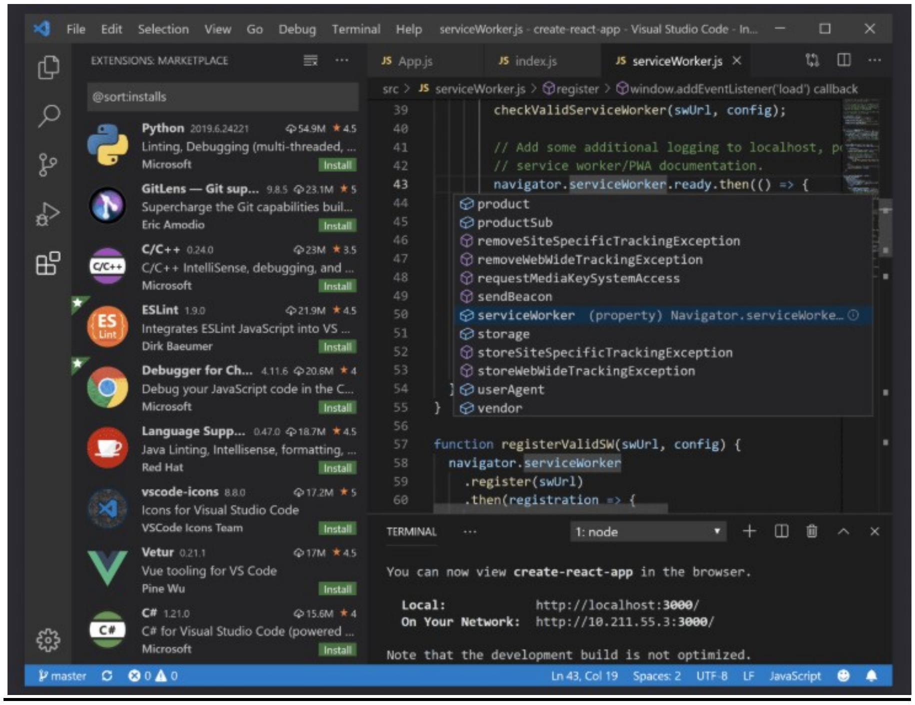
Bootstrap is a collection of free and open-source tools for building responsive websites and web applications. An immense framework for HTML, CSS, and JavaScript for developing responsive websites. It solves many problems we used to have, one of which is a browser compatibility problem. Today, websites are ready for all browsers (IE, Firefox, and Chrome) and in all screen sizes (Desktop, Tablets, Phones). To include bootstrap in the website include it from bootstrap CDN link or download it form bootstrap site. Bootstrap 4 is the latest stable version of bootstrap.

Fig 1.9: Bootstrap Logo

1.4.3 IDE, Softwares , OS, Libraries and Plugins

1.4.3.1 Visual Studio Code

VS code is a free & open source code editor made by Microsoft for Windows, Linux and macOxS. Its features include debugging support, syntax highlighting, intelligent code elimination, captions, code retrieval, and Git embedding. In visual studio code users can change the display theme, update keyboard shortcuts, preferences, and can install extensions that adds extra functionality.


File
Edit
Selection
View
Go
Debug
Terminal
Help
serviceWorker.js - create-react-app - Visual Studio Code - In...
EXTENSIONS: MARKETPLACE
@sort:installs
Python 2019.6.24221 54.9M ★4.5
Linting, Debugging (multi-threaded, ...
Microsoft Install
GitLens — Git sup... 9.8.5 23.1M ★5
Supercharge the Git capabilities buil...
Eric Amodio Install
C/C++ 0.24.0 23M ★3.5
C/C++ IntelliSense, debugging, and ...
Microsoft Install
ESLint 1.9.0 21.9M ★4.5
Integrates ESLint JavaScript into VS ...
Dirk Baeumer Install
Debugger for Ch... 4.11.6 20.6M ★4
Debug your JavaScript code in the C...
Microsoft Install
Language Supp... 0.47.0 18.7M ★4.5
Java Linting, Intellisense, formatting, ...
Red Hat Install
vscode-icons 8.8.0 17.2M ★5
Icons for Visual Studio Code
VSCode Icons Team Install
Vetur 0.21.1 17M ★4.5
Vue tooling for VS Code
Pine Wu Install
C# 1.21.0 15.6M ★4
C# for Visual Studio Code (powered ...
Microsoft Install
App.js
index.js
serviceWorker.js
src > serviceWorker.js > register > window.addEventListener('load') callback
product
productSub
removeSiteSpecificTrackingException
removeWebWideTrackingException
requestMediaKeySystemAccess
sendBeacon
serviceWorker (property) Navigator.serviceWorke...
storage
storeSiteSpecificTrackingException
storeWebWideTrackingException
userAgent
vendor
TERMINAL
1: node
master
0 0
Ln 43, Col 19
Spaces: 2
UTF-8
LF
JavaScript

```
        checkValidServiceWorker(swUrl, config);

        // Add some additional logging to localhost, po
        // service worker/PWA documentation.
        navigator.serviceWorker.ready.then(() => {
```

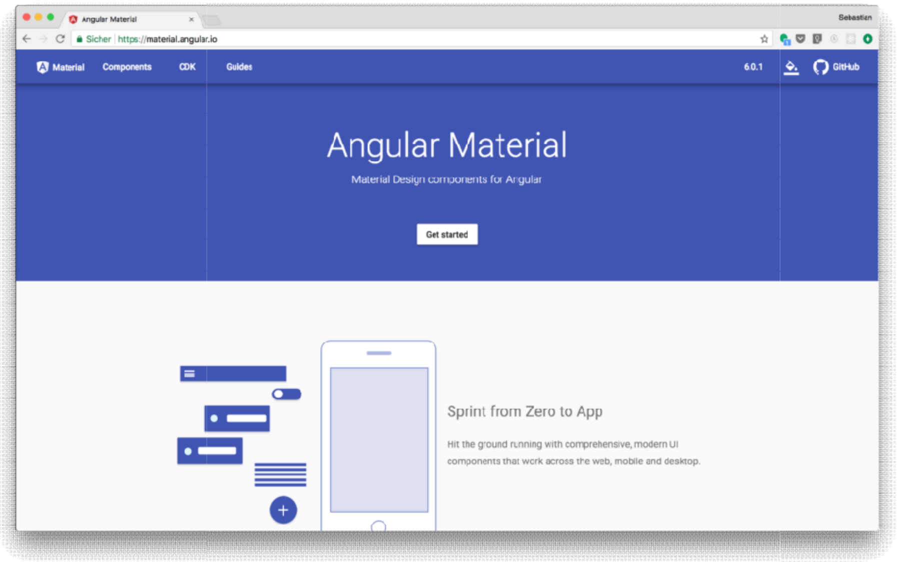
```
    }
  }

function registerValidSW(swUrl, config) {
  navigator.serviceWorker
    .register(swUrl)
    .then(registration => {
```

```
You can now view create-react-app in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://10.211.55.3:3000/

Note that the development build is not optimized.
```

Fig 1.10: Visual Studio code

1.4.3.2 Angular Material

Angular Material is a library that is part of the UI for Angular developers. While adhering to modern web design principles such as browser portability, application independence and good scalability, angular Material components help to create attractive, consistent and effective web pages and web applications It helps to create fast, beautiful and responsive websites.



Angular Material

Material Design components for Angular

Sprint from Zero to App

Hit the ground running with comprehensive, modern UI components that work across the web, mobile and desktop.

Fig 1.11 Angular Material

1.4.3.3 Git

Git is popularly used system software where there are large numbers of people work together on one project in parallel. It is a distributed version control system that tracks all code changes over time process entire software development. The motivation behind this program is to speed the development process across the team, supporting non-linear workflow & maintaining data integrity. It is mind-based & therefore easy to use and understand & work on, working with the mind branches, reunions, reunions and more. It keeps the work local on our system and global as well as at the origin.

Fig 1.12 Git

1.4.3.4 Json-Server

A common task for front – End developers is to use the REST backend service to deliver some data in the JSON format to the previous application and make sure everything is working as expected. You can also set up a full backend server, e.g. using Node.js, Express and MongoDB. However this takes time and a very simple method can help speed up the development time before the end.

JSON Server is a simple project that helps you set up a REST API with CRUD functions very quickly. It is available as NPM Package. It can be installed using the Node.js package Manager

```
$ npm install -g json-server
```

1.5Organisation

The way in which this project report is organized is that:

1. In 1st Chapter problem statement is covered along with basic introduction of entire system, major technologies, libraries and frameworks given.
2. In 2nd Chapter about the literature survey and the study that was done before starting the development process. Brief introduction is given about rendering of HTML, working of web browsers, angular observables and interceptors.
3. Chapter 3 describes about software and hardware requirements, architecture of the application and flow chart.
4. Chapter 4 describes everything about project development i.e. its database schema, modules , component , services , auth guards that are used in the entire project
5. Chapter 5 includes all the screenshot of the project. The entire project completed till date is concluded in this chapter and also throws light on the project's future scope of the project and other features that are in the process of being development.

CHAPTER – 2 LITERATURE SURVEY

It's consistently a smart thought to know precisely what subject to chip away at. A similar practice is to be considered on account of improvement. Further turn of events and information on the underlying foundations of innovation and sees how it functions, the more gifted the architect will have the option to play out, the simpler the designer will have the option to discover an answer for the issues they are looking, in actuality, work. For this reason all understudies who joined the organization must be taught with the goal that experts know about the issues that engineers are looking in their day by day lives. Not exclusively was this however was the kind of issues the learners are approached to discover an answer for and to unravel as assignments. In this section I portrayed all the exploration work I did during preparing and at the hour of advancement experience when I involved with a mistake, or creation issue.

So here the primary spotlight is on a couple of essential thoughts that one neglects to perceive in light of theoretical layers of new structures. Commonly it is discovered that the issue happens on the grounds that the designer doesn't know about how the stream functions and how things are coming in the image. Consequently it is first critical to concentrate on fundamental dialects and essential ideas of those dialects, so that regardless of whether we don't coexist well with the system we're chipping away at, we despite everything can take care of the issue with the information and comprehension of the essential indigenous dialects. Along these lines the preparation procedure depended on handy information and comprehension.

How web browsers work? [12]

The main function of a web browser is to request resources on the web and display them inside a browser window. Usually the browser will request HTML, CSS, JavaScript and image content on the server and translate it based on web standards and specifications. Browsers follow standards because it allows websites to behave in the same way across browsers, and creates less performance and fewer headaches for web developers.

7 key components of any web browser are:

- **Layout Engine –** It takes input from browser in the form of URL bar, search box, mouse click, key presses etc and transfers it to rendering engine.
- **Rendering Engine –** This takes HTML code and translates it to what you see. For example, a tag was interpreted by a translator engine as a set of instructions for making text within a part bold**.**
- **User interface -** This is a visual presentation of the browser controls, for example the back and front buttons, bookmarks, and all the chrome that appears around the browser window where web pages display.
- **JavaScript Engine -** This engine takes JavaScript code executes it, and returns results.
- **Network Layer -** This is a browser function that occurs behind the scenes and handles network functions such as encryption, http and ftp requests, and all network settings such as expiry and HTTP status codes.
- **Storage -** The browser must store specific information that may include cached files, cookies, and newly updated browsers to be able to store data and objects in JavaScript.
- **Operating System Operations -** The browser must work with the operating system to draw several page objects such as boxes and window elements (close, expand, and minimize buttons).

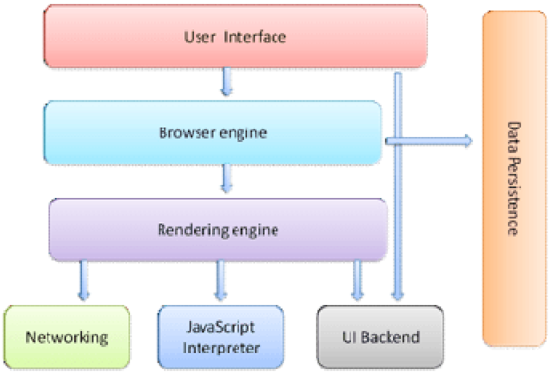


Fig 2.1: 7 Key Components of Web Browser

2.1.1 ***Rendering Engine***

The rendering engine plays a very important role as it reflects what you see on your screen. It communicates with the browser's browsing network to hold HTML code and other objects passed to the remote server.

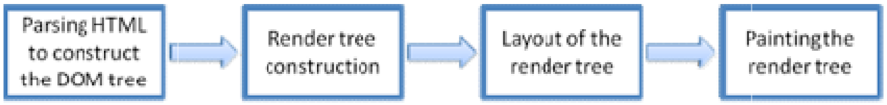

Parsing HTML to construct the DOM tree
Render tree construction
Layout of the render tree
Paintingthe render tree

Fig 2.2 Steps followed by rendering engine

Rendering engine follow these steps:

1. Parsing HTML to construct the DOM tree – HTML elements are passed and converted into DOM tree.
2. Render tree construction – Parsing of CSS attributes is done and combined with DOM tree to create render tree.
3. Layout of the render tree – After construction of render tree HTML elements and attributes are mapped to screen coordinates, starting from position 0, 0.
4. Painting the render tree – With the help of OS interface each node of the render tree is drawn out on the screen.

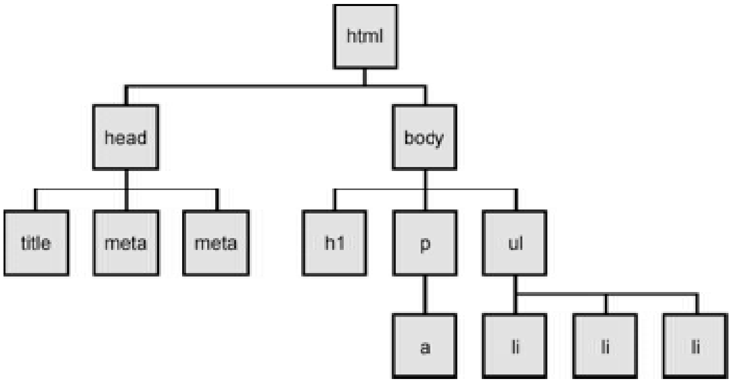


Fig 2.3 HTML Tree

2.2 *Angular Observables [9][13]*

Observables are widely used in Angular for handling a variety of common asynchronous operations. To understand why we need to use asynchronous code instead of empty JavaScript, consider the code waiting for a response from a remote server. There is no way to know when the answer will come back, we don't know whether we will get any reply or not, or when will the reply come. So it is not a good idea to stop executing the script until the response is received. Asynchronous code is the best solution, and the way Angular chooses to handle asynchronous code is obvious. Angular's Http Client returns observables from HTTP method calls.

Observables Working

1. **Observer and Observable**
   - Data is fired by the observable in response to an event. Event can be a button clicked by the user or new page load or response to the data that is received from the remote server.
   - Observer subscribes to receive the data sent by the observable. To use the data Observer has three handle:
     - onNext that handles the requested data.
     - onError to handle errors if any.
     - onComplete - used on process completion

2. **HTTP**

   Angular's Http Client returns observables from HTTP method calls. For instance, http.get('/api') returns an observable. To cancel HTTP requests unsubscribe() method is used.

3. **Async Pipe**

   The Async Pipe subscribes to an observable and returns the latest value it has emitted. Async pipe marks the component to be checked for changes, every time when a new value is emitted.

4. **Router**

To provide events as observables use Router.events. The filter() operator from RxJS can also be used to look for events of interest, and subscribe to these events in order to make decisions based on the sequence of events in the navigation process. To get information about a route path use an injected route service that is ActivatedRoute that uses observables

5. **Reactive Forms**

Reactive forms have FormControl properties such as valueChanges and statusChanges that use observables to monitor form control values and raise events. Subscribing to an angular observable form-control property is a way of triggering app. logic within the component class.

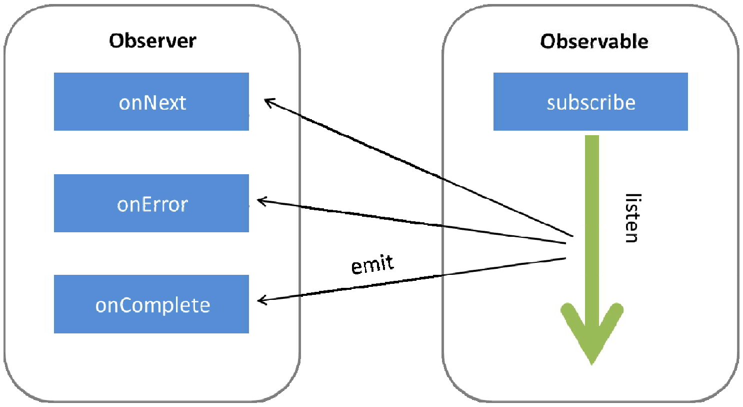


Fig 2.4 Observer and Observable

2.3 Angular Interceptors [14]

Interceptors on Angular, as the name suggests, is an easy way given the framework to extract and modify global http requests before they are deployed to the server. That comes with the utility, which allows us to configure authentication tokens, add application logs, add customization topics that may need to be used and much more.

Interceptors can perform a variety of tasks, from authentication to encryption, in a standard, standard, to every HTTP request / response. Without exception, developers will need to use this function explicitly for each HttpClient call.

Since interceptors are HttpClient dependent, you should add them in the same provider (or parent) that provides HttpClient. For example, if you think you have your HttpClientModule imported into AppModule, you should add patches to providers there too. The multi: true option provided tells Angular that you provide multiple interceptors and that is necessary if so.

It is also important to note that Angular uses interceptors in such a way that you assign them to your module providers.

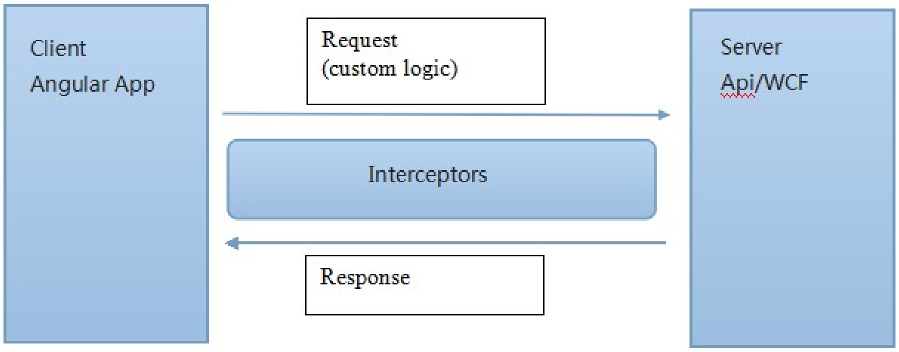


Fig 2.5: Interceptor Flow Diagram

CHAPTER – 3 SYSTEM DEVELOPMENTS

The entire website was designed and developed using Angular Framework for frontend and fake rest API called Json Server was used for backend. This project is done by splitting it into 3 modules. The tool used was Visual Studio Code Version 1.45 and database was in the form of Json format. Agile Methodology was used for website development.

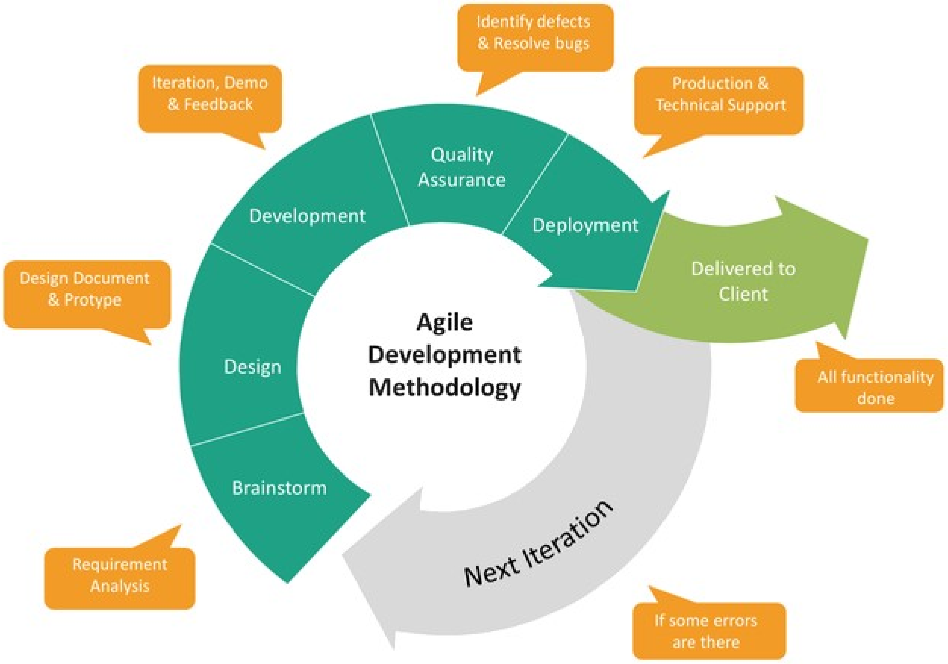


Fig 3.1 Agile Methodology

3.1 Software Requirements

Client Side	: Internet Explorer, Google Chrome, Operating system (Windows)
Developer Side	: Angular CLI, VS Code (Recommended), Operating system (Windows)
Servers	: Json Server, Microsoft Windows

3.2 Hardware Requirements

Processor/RAMs (Client Side)	: 2 GB (Min)
Processor/RAM (Developer Side)	: 4 GB (Recommended)
Web Server	: Any suitable sever
Primary Memory	: 2GB or Higher RAM

3.3 Functional Requirements

3.3.1 Register Panel

- User can register using Google or registration form
- Only registered users can do shopping.
- Every user has to provide name, email id, phone No, address details before adding products to cart.

3.3.2 User (Buyer)

- Buyer can login through Google.
- Buyer can filter products depending on category.
- Buyer can add products to cart. The product quantity for a particular product is capped to 5.
- Buyer can move products from cart to save for later items wish list.
- Buyer can send products to different address with same or different billing address.
- Buyer can reorder a particular order from orders history page.

3.3.3 User Interface(UI)

- The website is user friendly
- The website is device friendly i.e. interface adapts to different screen sizes.

3.3.4 Sprint Work

The website contains following pages

1. Login Page
2. Registration Page
3. Products Dashboard Page
4. View Cart Page
5. Shipping Address Page
6. Billing Address Page
7. Add/Edit Address Page
8. Review Order Page
9. Orders History Page

In any project it is important to fulfill all the requirements of the user. Following are the User tasks that need to be accomplished in this project.

- As a customer, I should be able to register.
- As a customer, I should be able to login.
- As a customer, I should be able to view all products and their categories
- As a customer, I should be able to add product to cart.
- As a customer, I should be able to increase product quantity.
- As a customer, I should be able to modify cart items, add more products to cart
- As a customer, I should be able to add cart items to save for later items WishList.
- As a customer, I should be able to select delivery and billing address.
- As a customer, I should be able to add new address or edit existing address.
- As a customer, I should be able to review my order, check total price.

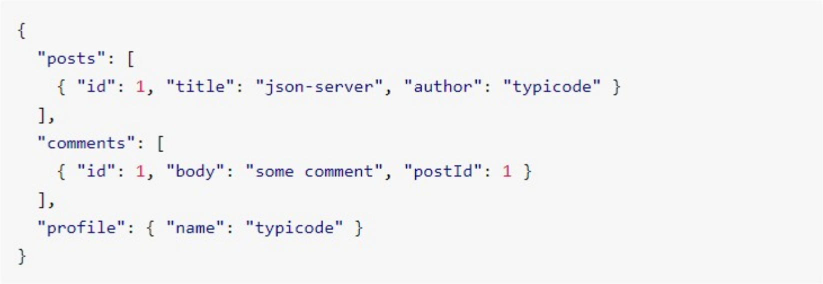
- As a customer, I should be able to check my previous orders.

*3.4**Setting JSON-Server [16]***

Json-server is a part of npm package. To install Json-server, a fake rest API locally on your system run this command.

```
npm install -g json-server
```

- Create database.json file with some data



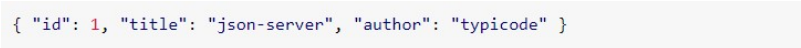
```
{
  "posts": [
    { "id": 1, "title": "json-server", "author": "typicode" }
  ],
  "comments": [
    { "id": 1, "body": "some comment", "postId": 1 }
  ],
  "profile": { "name": "typicode" }
}
```

Fig 3.2 Sample Json Data

- Start json server with this command.

  ```
  json-server -watch database.json
  ```

- Now on your system browse to “ http://localhost:3000/posts/1, you” get



```
{ "id": 1, "title": "json-server", "author": "typicode" }
```

Fig 3.2 Sample Json Output

- If POST, PUT, GET, DELETE http requests are made then changes will automatically be saved to database.json file.

CHAPTER – 4 PROJECT DEVELOPMENT

4.1 Implementing Database

JSON allows developers store data as JSON documents in a database which is a great deal of simplicity and flexibility, and in some cases it may be more practical to. JSON documents required a NoSQL database in the past. Class and interface is made as a part of software design pattern for better layout of application code. Interface is an abstract type, it does not contain any code and it is used with a class to define custom data types. Data security and validation can be maintained with the help of classes and interfaces.

1. Users Interface

- userId – It is of integer type and is a primary key used for indentifying each user.
- Name – It is of string type and is used to get name of the user.
- Email - It is of string type and is used to get email id of the user.
- Address - It is an array of addresses and is used to get address details of the user.

2. Products Interface

- productId - It is of integer type and is a primary key used for indentifying each product.
- productName - It is of string type and is used to get name of the product.
- productPrice - It is of integer type and is used to get price of the product.
- productQuantity – It is of integer type and is used to get maximum quantity of particular product.

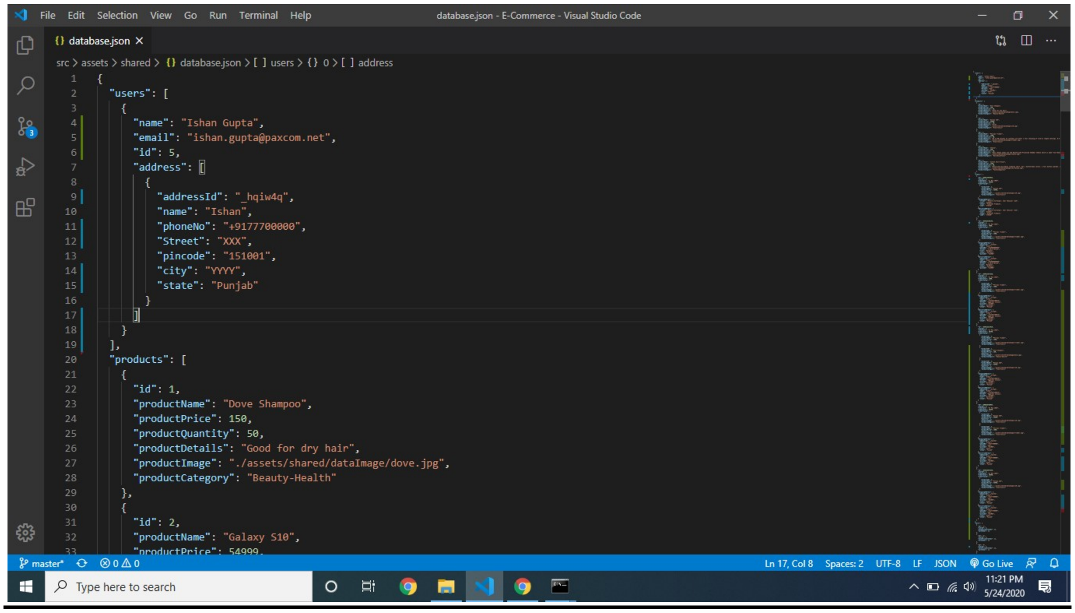
- productDetails – It of string type and is used to get details of the product.
- productImage – It is of string type and is used to get product image for display.
- Product category – It is of string type and is used to define category of the product.

3. Cart Interface

- cartId – Unique integer cart id.
- userId – Id of registered user.
- saveForLaterItems – It is an array of products saved for later.
- cartOrder – It is an array of items selected for buying.

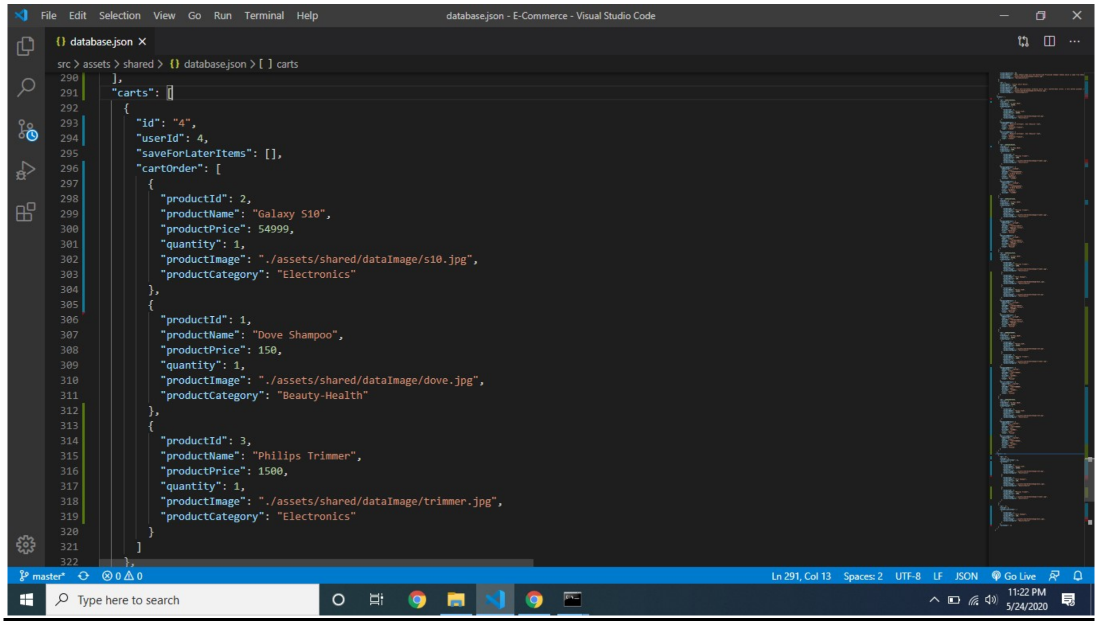
4. Address Interface

- addressed – It is string type unique address id used for identifying each project.
- Name – It is of string type and is used to get name of the user.
- Email - It is of string type and is used to get email id of the user.
- phoneNo – It is of integer type and must of 10 digits starting with +91.
- Street – It is of string type.
- Pincode - It is of integer type and must of 6 digits.
- City - It is of string type.
- State - It is of string type.



```json
{
  "users": [
    {
      "name": "Ishan Gupta",
      "email": "ishan.gupta@paxcom.net",
      "id": 5,
      "address": [
        {
          "addressId": "_hqiw4q",
          "name": "Ishan",
          "phoneNo": "+9177700000",
          "Street": "XXX",
          "pincode": "151001",
          "city": "YYYY",
          "state": "Punjab"
        }
      ]
    }
  ],
  "products": [
    {
      "id": 1,
      "productName": "Dove Shampoo",
      "productPrice": 150,
      "productQuantity": 50,
      "productDetails": "Good for dry hair",
      "productImage": "./assets/shared/dataImage/dove.jpg",
      "productCategory": "Beauty-Health"
    },
    {
      "id": 2,
      "productName": "Galaxy S10",
      "productPrice": 54999,
```

Fig 4.1: Screenshot of Users and products data



```json
  ],
  "carts": [
    {
      "id": "4",
      "userId": 4,
      "saveForLaterItems": [],
      "cartOrder": [
        {
          "productId": 2,
          "productName": "Galaxy S10",
          "productPrice": 54999,
          "quantity": 1,
          "productImage": "./assets/shared/dataImage/s10.jpg",
          "productCategory": "Electronics"
        },
        {
          "productId": 1,
          "productName": "Dove Shampoo",
          "productPrice": 150,
          "quantity": 1,
          "productImage": "./assets/shared/dataImage/dove.jpg",
          "productCategory": "Beauty-Health"
        },
        {
          "productId": 3,
          "productName": "Philips Trimmer",
          "productPrice": 1500,
          "quantity": 1,
          "productImage": "./assets/shared/dataImage/trimmer.jpg",
          "productCategory": "Electronics"
        }
      ]
    },
```

Fig 4.2: Screenshot of Cart data

4.2 Modules

In Angular, the module is a way of grouping components, directories, pipes, and related services, in a way that can be integrated with other application programming modules. The Angular app can be thought of as a puzzle where each piece (or module) is required to be able to see the full picture. The application is divided into three modules and different components.

1. **Login Module** – This module contains login component and register component.
2. **View Product Module** – This module contains header component, products component and product-detail component.
3. **Cart Module** – This module contains viewCart/ReviewOrder component, selectAddress Component, AddNewAddress component, OrderHistory component

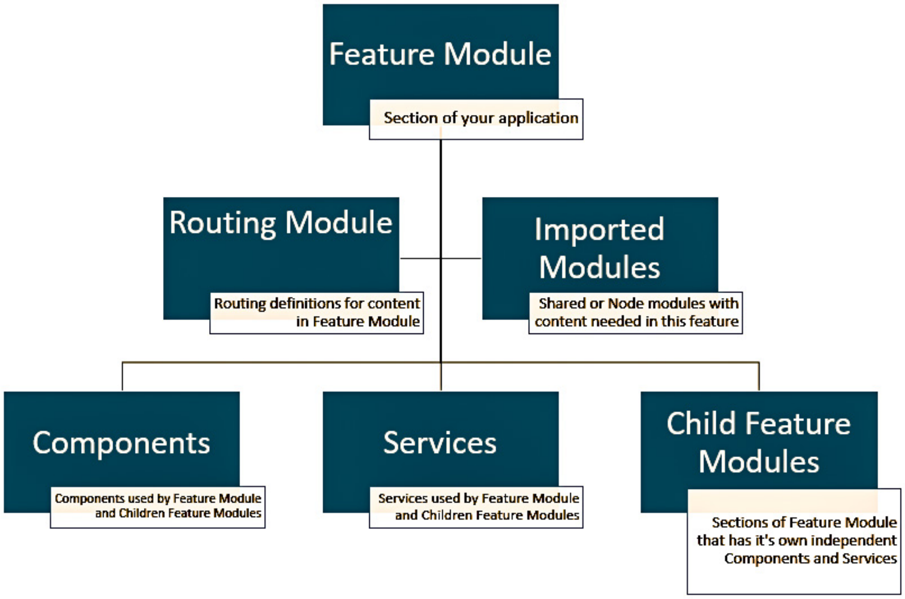

Feature Module
Section of your application
Routing Module
Routing definitions for content in Feature Module
Imported Modules
Shared or Node modules with content needed in this feature
Components
Components used by Feature Module and Children Feature Modules
Services
Services used by Feature Module and Children Feature Modules
Child Feature Modules
Sections of Feature Module that has it's own independent Components and Services

Fig 4.3 Angular Module Tree

To create a new module we need to make use of angular CLI. The command for the same is “ `ng g module ModuleName` ”.

4.3 Components

In Angular Components are the basic building blocks of UI. It has a selector, template (.html), style (.css, .sass, .scss) and other properties and method using which it specifies the metadata required for processing.

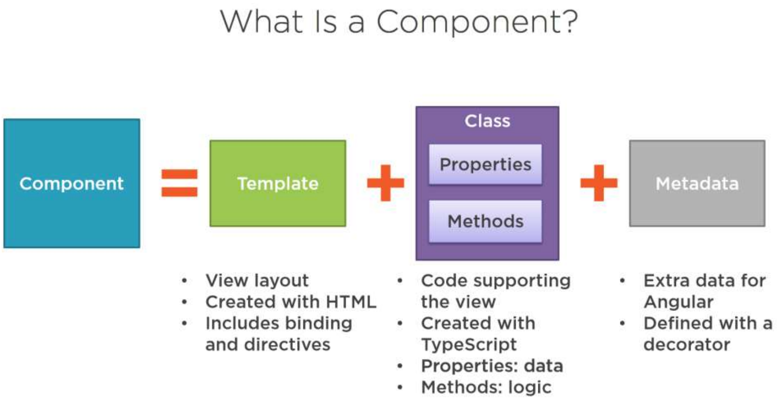


Fig 4.4: Angular Component

1. **Login Component -** This component is part of the login Module. It allows us to either log in through goggle or log in with email id and password. If the User is not registered he/she can register himself using the register button.

2. **Register Component** – This component is part of the login Module. It allows us to register by adding user details.

3. **View Products Component** – This component is part of the View Product Module. It shows all the products and different categories of products. Users can select a particular category to filter products according to the category. Users can also add a product to the cart from the product dashboard.

4. **View Product Details Component** – This component is part of the View Product Module. It shows detail of a particular product selected from the View Products Component. Users can increase the quantity of a product and add it to the cart.

5. **Header Component** – This component is part of the View Product Module. It shows the top header bar which dynamically changes its layout depending on the current page on which the user is.

6. **View Cart/Review Order Component** – This component is part of the Cart Module. It dynamically changes its layout depending on the current URL. The DOM renders itself based on the current URL and the user will see only the required data. If the URL is:-

   - ViewCart - The user can add more items to cart, move items to Save For Later WishList, and can proceed further to buy items.

   - ReviewOrder – The user can review his order, Shipping Address, Billing Address, order total and precede further to place the order.

7. **Select Address Component –** This component is part of the Cart Module. It allows the user to select his delivery address and billing address.

8. **Add New Address/ Edit Address Component -** This component is part of the cart Module. It also dynamically changes its layout depending on the current URL. The DOM renders itself based on the current URL and the user will see only the required data. If the URL is:-

   - AddNewAddress - The user can add a new address by filling his address details according to the validations set.

- EditAddress – User's Address will be prefetch from the database and will be filled automatically. Then the user can edit a particular address filed and update his address.

9. **Orders History Component** - This component is part of the Cart Module. It allows the user to check his latest orders placed.

To create a new component we need to make use of angular CLI. The command for the same is " `ng g compoent ComponentName` ".

4.4 Services [8]

In angular when we want to reuse the same code or data that needs to be shared among different components we use services. Services are singleton objects that are instantiated only once during the lifetime of an angular application. It helps us to access methods and properties across other components in the entire project. Its main objective is to organize and share business logic, models, or data and functions with different components of an Angular application.

The separation of concerns is the main reason why Angular services came into being. Angular service is countless and offers some very useful functions. These functions can be requested from any Angular component, such as controllers, controls, etc. This helps in dividing the web application into smaller, different logical units that can be used again.

Services deals with GET, POST, PUT, DELETE operations by performing http requests with the server.

Services used in Ecommerce project are:

- **Authentication Service** – It contains logical functions like checkEmailAlreadyExists() , registerUser(), getRegisteredUserData().

- **View Product Service** - It contains logical functions like viewProducts() , viewProductDetails(), updateHeader().

- **Cart Service** - It contains logical functions like getCartData() , updateCartData(), deleteCartData(), getUserId(), getUserAddressData(), deleteUserAddressData(), getOrderDetails().

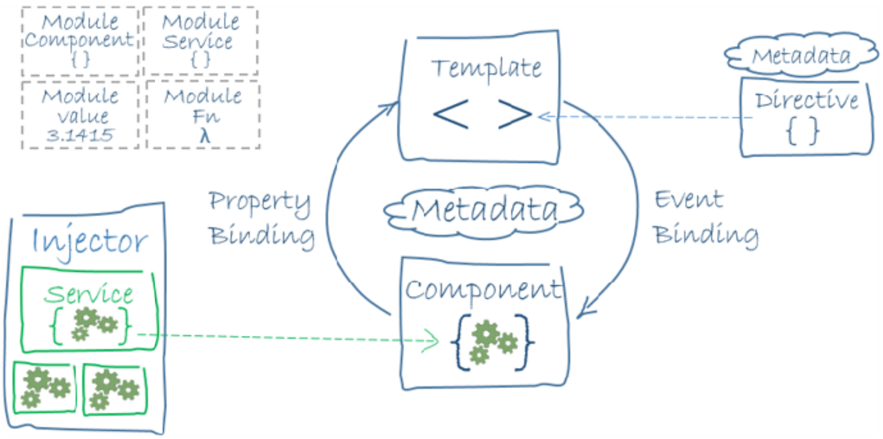


Fig 4.5: Service Illustration

To create a new service we need to make use of angular CLI. The command for the same is " `ng g service ServiceName ".`

4.5 ***Route Guards***

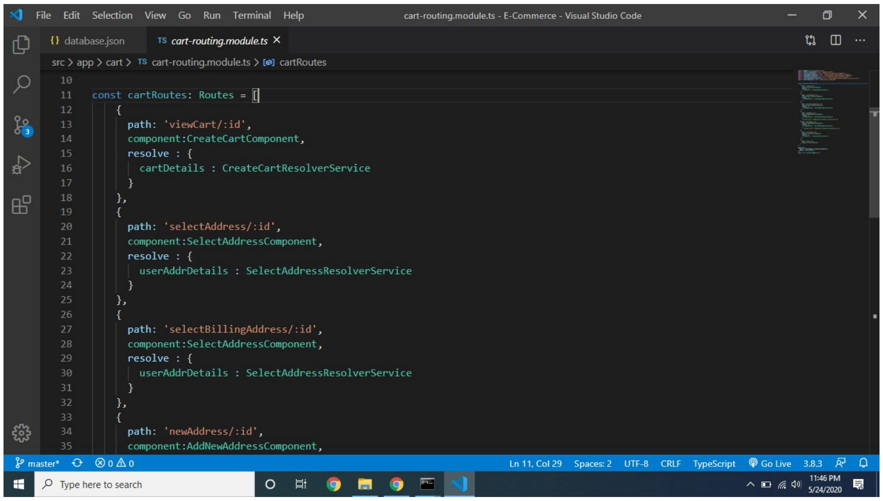
Angular route gurads are interfaces that can tell the router whether it should allow navigation on the requested route or not. They make this decision by looking for a true or false return value for the class using the given guard indicator.

There are five types of guards and each of them is named after a specific order. The behavior of the router is changed differently depending on which guard is used.

The guards are:

- canActivate
- canActivateChild
- canDeactivate
- canLoad
- canResolve

In this project only resolve guard was used. A service class is used with router to resolve data during navigation. A resolve() method is defined by the interface and it will be invoked when the navigation starts. Before the final activation of the route, the router waits for the data to be resolved.


File Edit Selection View Go Run Terminal Help
cart-routing.module.ts - E-Commerce - Visual Studio Code
database.json
cart-routing.module.ts
src > app > cart > cart-routing.module.ts > cartRoutes
master* ⊗ 0 ⚠ 0
Ln 11, Col 29 Spaces: 2 UTF-8 CRLF TypeScript Go Live 3.8.3
Type here to search
11:46 PM
5/24/2020

```

const cartRoutes: Routes = [
  {
    path: 'viewCart/:id',
    component:CreateCartComponent,
    resolve : {
      cartDetails : CreateCartResolverService
    }
  },
  {
    path: 'selectAddress/:id',
    component:SelectAddressComponent,
    resolve : {
      userAddrDetails : SelectAddressResolverService
    }
  },
  {
    path: 'selectBillingAddress/:id',
    component:SelectAddressComponent,
    resolve : {
      userAddrDetails : SelectAddressResolverService
    }
  },
  {
    path: 'newAddress/:id',
    component:AddNewAddressComponent,
```

Fig 4.6 Code Screenshot of resolve guard

CHAPTER – 5 RESULTS

5.1 Screenshots

5.1.1 Login Page

This page comprises of login with Google button, email, and password field which is set by the user while registering him as a new user and an option to create a new account. On this page header will show only register button.

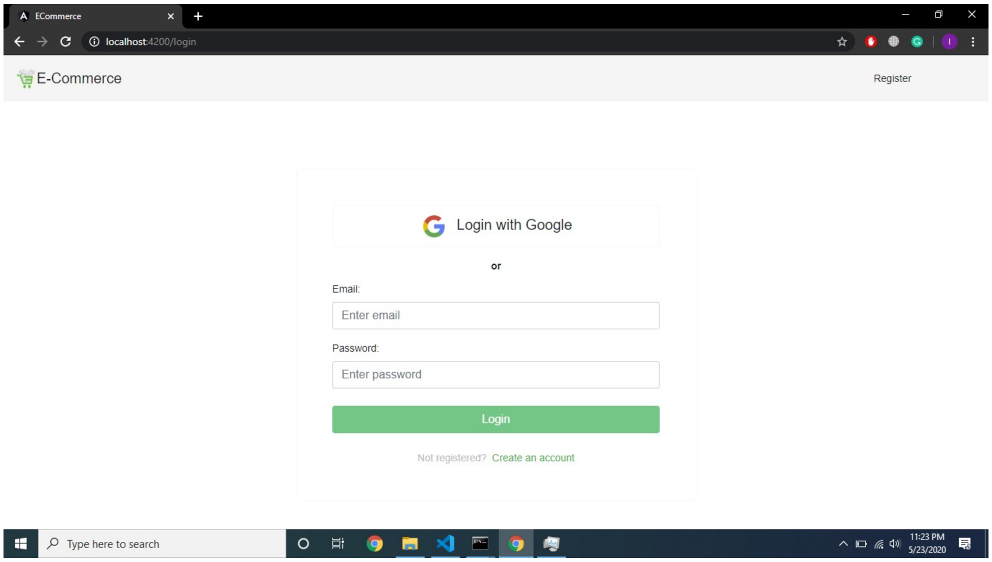
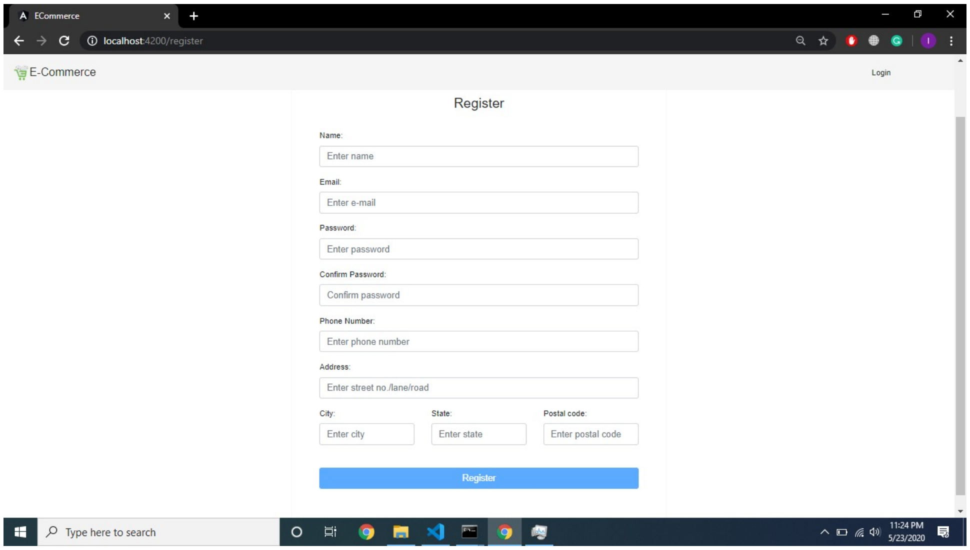


Fig 5.1: Login Profile

5.1.2 ***Register Page***

This page allows the user to register as a new user by filling the following details. After Successful registration user will be redirected to Login Page. On this page, the same header will dynamically change itself and will show only the Login button.



Register

Name:
Enter name

Email:
Enter e-mail

Password:
Enter password

Confirm Password:
Confirm password

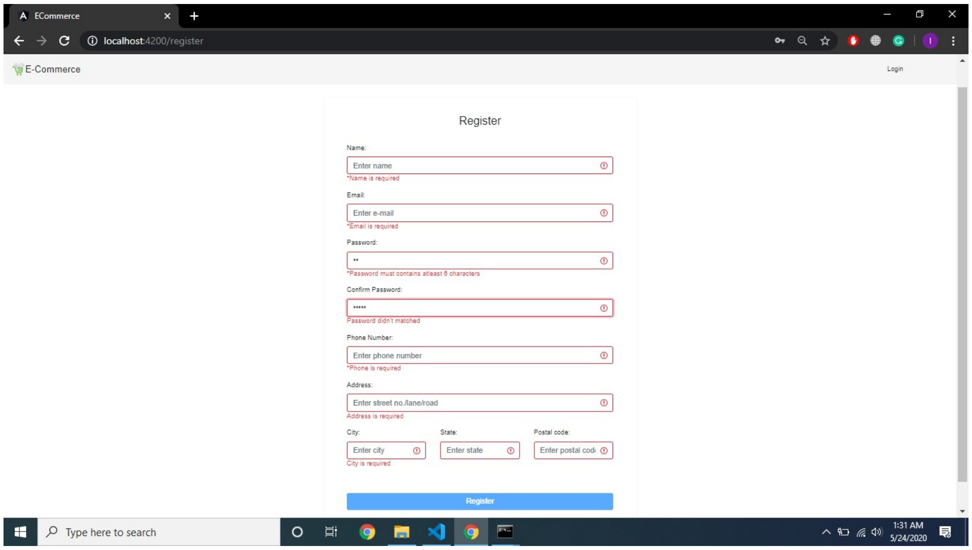
Phone Number:
Enter phone number

Address:
Enter street no./lane/road

City: Enter city
State: Enter state
Postal code: Enter postal code

Register

Fig 5.2: Register Profile



Register

Name:
Enter name
*Name is required

Email:
Enter e-mail
*Email is required

Password:
••
*Password must contains atleast 6 characters

Confirm Password:
•••••
Password didn't matched

Phone Number:
Enter phone number
*Phone is required

Address:
Enter street no./lane/road
Address is required

City: Enter city
City is required
State: Enter state
Postal code: Enter postal code

Register

Fig 5.3: Register Profile with validations.

5.1.3 ***Products Dashboard***

This page shows all the products and different categories of products. On this page, the header will dynamically change itself and will show only the name of user login, cart button, orders button and logout button. User can add products to cart with their quantity by default set to 1.

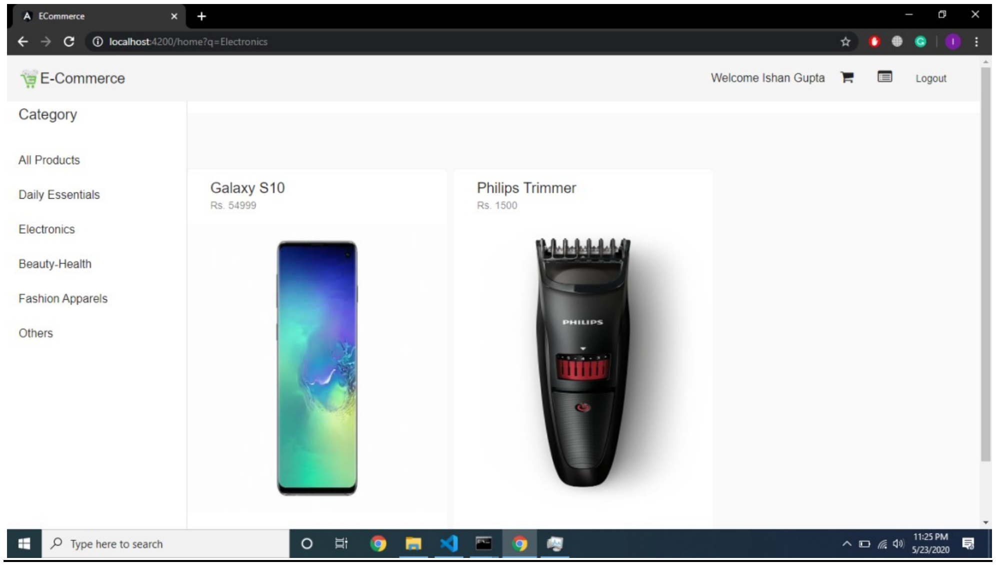
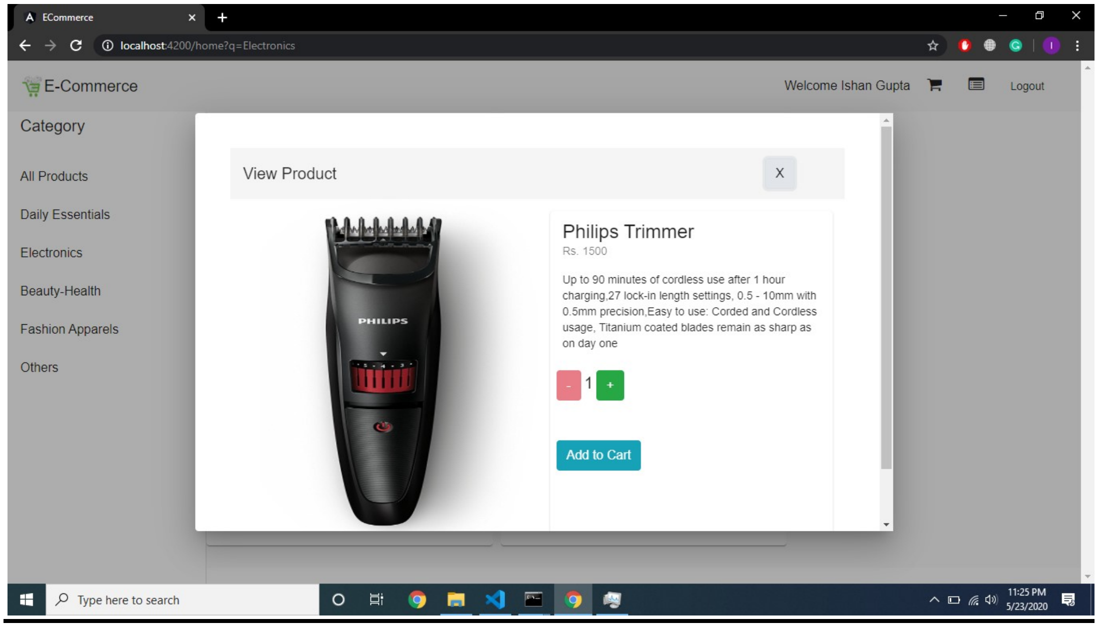


Fig 5.4: Product Dashboard

5.1.4 Products Details

Here Users can see product details, change the quantity, and add the product to cart. Maximum quantity for a product is capped to 5.



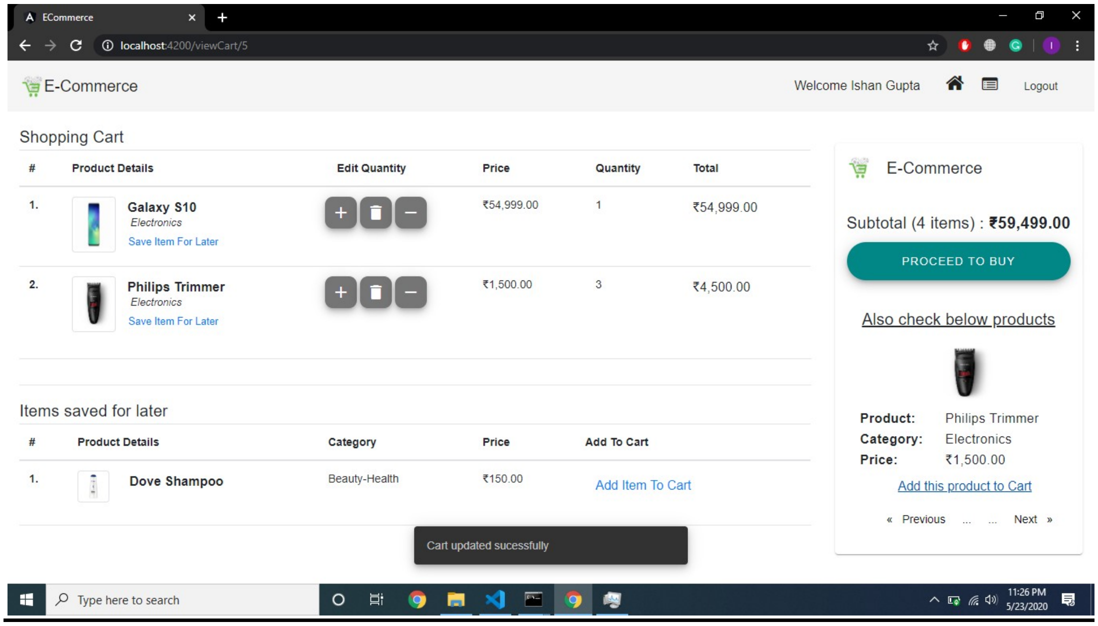
Philips Trimmer
Rs. 1500

Up to 90 minutes of cordless use after 1 hour charging,27 lock-in length settings, 0.5 - 10mm with 0.5mm precision,Easy to use: Corded and Cordless usage, Titanium coated blades remain as sharp as on day one

Fig 5.5: Product Details

5.1.5 *User Cart Details*

This page shows all the items in the cart added by a user. Here user can vary the item quantity, add it to Save For Later WishList, add more products to the cart, and at last, can proceed to buy. The data is shown based on user id set while user logins or registers himself. This page is part of a single component i.e. viewCart component explained above, that dynamically changes its view based on URL. If the shopping cart is empty, then proceed to buy button will be disabled. Pagination is used to show more products to the user.



Shopping Cart

#	Product Details	Edit Quantity	Price	Quantity	Total
1.	Galaxy S10 *Electronics* Save Item For Later	+ −	₹54,999.00	1	₹54,999.00
2.	Philips Trimmer *Electronics* Save Item For Later	+ −	₹1,500.00	3	₹4,500.00

Items saved for later

#	Product Details	Category	Price	Add To Cart
1.	Dove Shampoo	Beauty-Health	₹150.00	Add Item To Cart

Cart updated sucessfully

E-Commerce

Subtotal (4 items) : **₹59,499.00**

PROCEED TO BUY

Also check below products

Product:	Philips Trimmer
Category:	Electronics
Price:	₹1,500.00

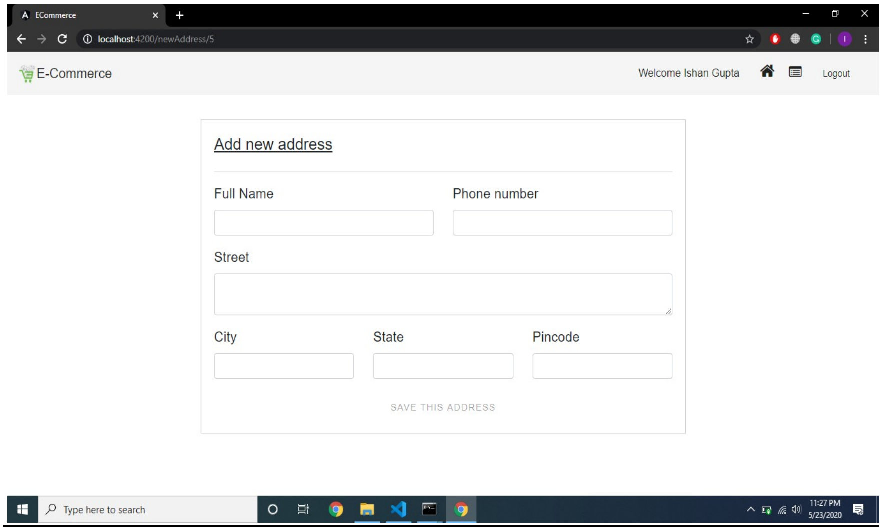
Add this product to Cart

« Previous ... ... Next »

Fig 5.6 User Cart

5.1.6 Add new Address / Edit Address

This page allows the user to add a new address or to edit the existence address. The DOM updates its view based on the URL. If the URL contains EditAddress then the address will be prefetch from the database based on address Id and the address details will be set in respective fields from where the user can update his address.



Add new address

Full Name

Phone number

Street

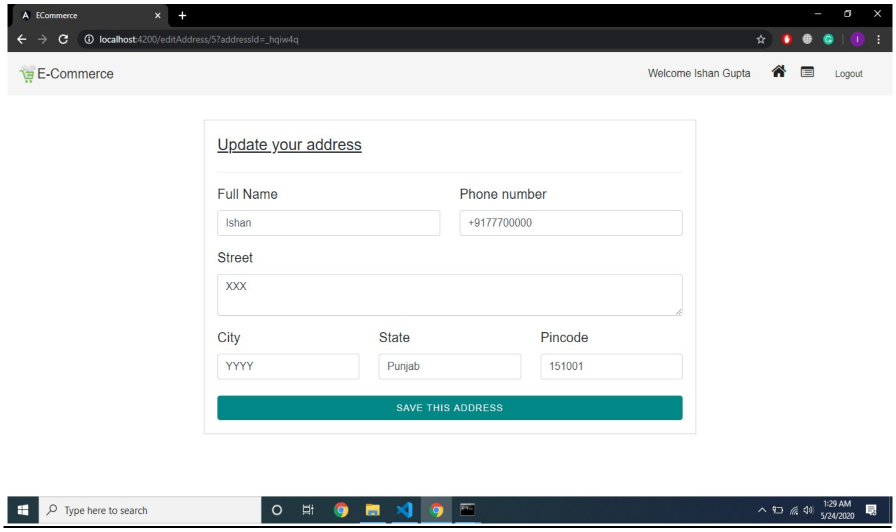
City

State

Pincode

SAVE THIS ADDRESS

Fig 5.7: Add New Address



Update your address

Full Name: Ishan

Phone number: +9177700000

Street: XXX

City: YYYY

State: Punjab

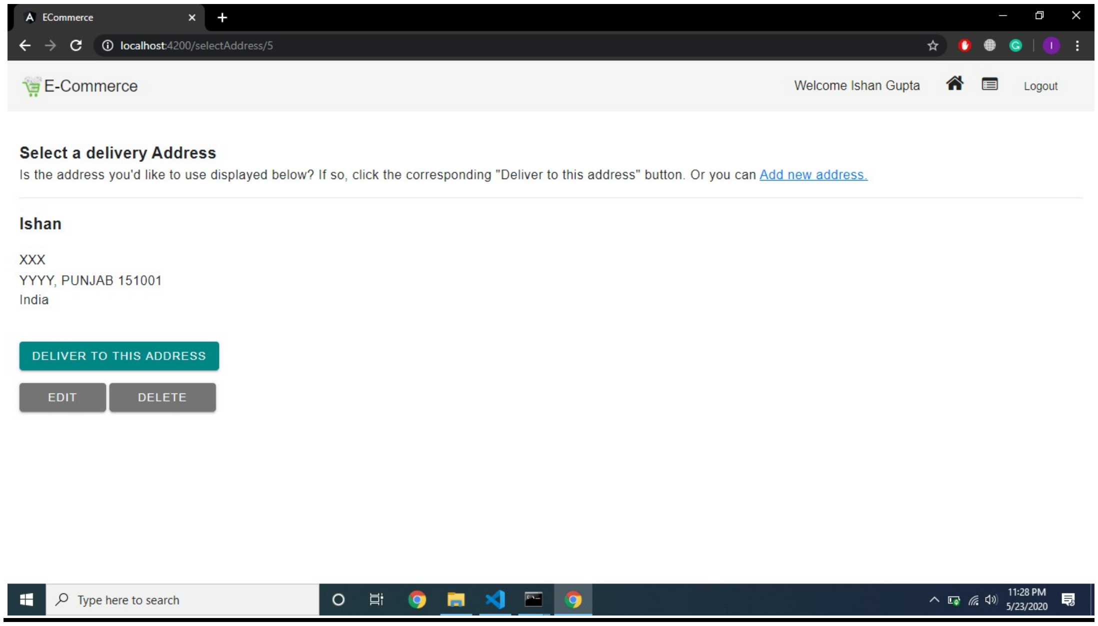
Pincode: 151001

SAVE THIS ADDRESS

Fig 5.8: Edit Address

5.1.7 Select Delivery / Billing Address

This page allows the user to select billing address and delivery address. The DOM updates its view based on the URL. If the URL contains deliveryAddress then the user will be able to select only delivery address.



Select a delivery Address

Is the address you'd like to use displayed below? If so, click the corresponding "Deliver to this address" button. Or you can Add new address.

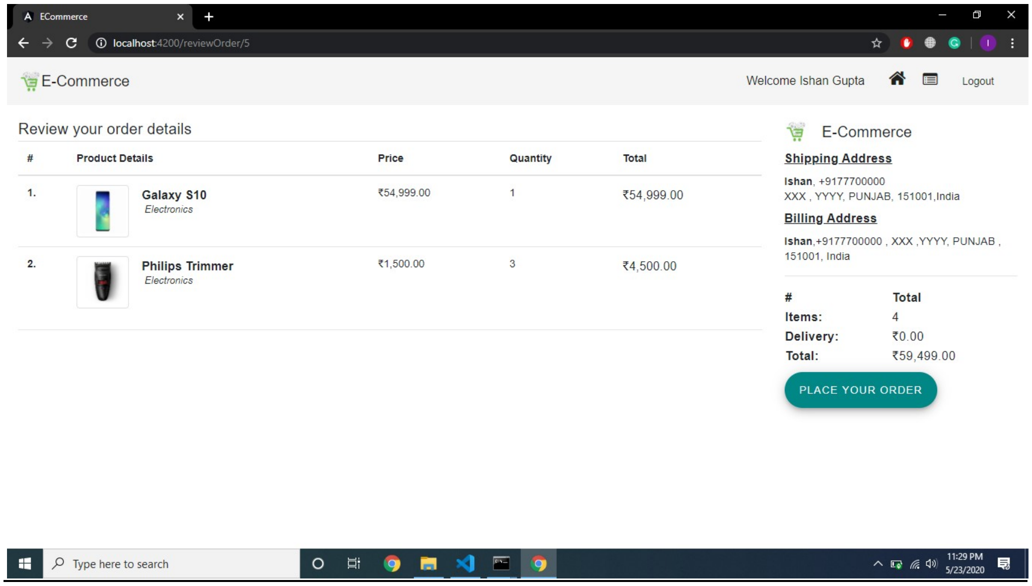
Ishan

XXX
YYYY, PUNJAB 151001
India

Fig 5.9: Select delivery Address

5.1.8 Review Order Details

This page shows all the items in the cart added by a user. Here user can review his order details, his shipping address, billing address, total no of items and order total. The data is shown based on user id set while user logins or registers himself. This page is part of a single component i.e. viewCart component explained above, that dynamically changes its view based on URL.



Review your order details

#	Product Details	Price	Quantity	Total
1.	Galaxy S10 *Electronics*	₹54,999.00	1	₹54,999.00
2.	Philips Trimmer *Electronics*	₹1,500.00	3	₹4,500.00

E-Commerce

Shipping Address

Ishan, +9177700000
XXX , YYYY, PUNJAB, 151001,India

Billing Address

Ishan,+9177700000 , XXX ,YYYY, PUNJAB , 151001, India

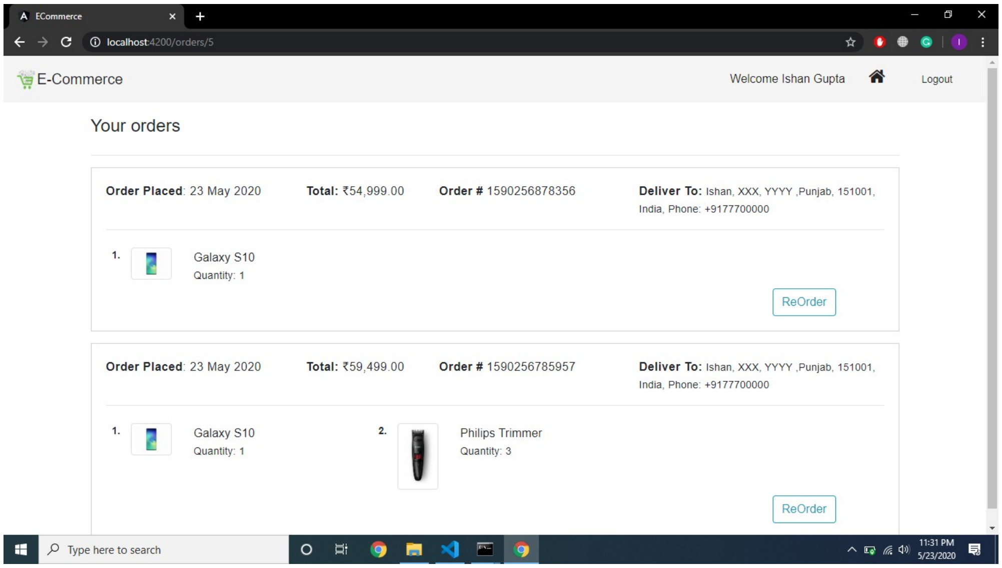
#	Total
Items:	4
Delivery:	₹0.00
Total:	₹59,499.00

PLACE YOUR ORDER

Fig 5.10: Review Order

5.1.9 *Orders History*

This page shows all the orders placed by the user. The most recent order placed will be shown on the top. From here user can reorder a particular order.



Your orders

Order Placed: 23 May 2020 **Total:** ₹54,999.00 **Order #** 1590256878356 **Deliver To:** Ishan, XXX, YYYY ,Punjab, 151001, India, Phone: +9177700000

1. Galaxy S10
Quantity: 1

ReOrder

Order Placed: 23 May 2020 **Total:** ₹59,499.00 **Order #** 1590256785957 **Deliver To:** Ishan, XXX, YYYY ,Punjab, 151001, India, Phone: +9177700000

1. Galaxy S10
Quantity: 1

2. Philips Trimmer
Quantity: 3

ReOrder

Fig 5.11: Orders History

5.1.10 Angular Loader

Loader is shown whenever there is any http request made to the server which indicates that server is fetching data from the database or making changes to the database.

If there is a button click event performed to proceed to next page, loader is shown which indicates that resolve guards used are resolving the data with the services used from database. After data is received the loader automatically hides and we move to the next page.

Preloading of data prevents rendering of html first while there is delay in receiving the data from the server.

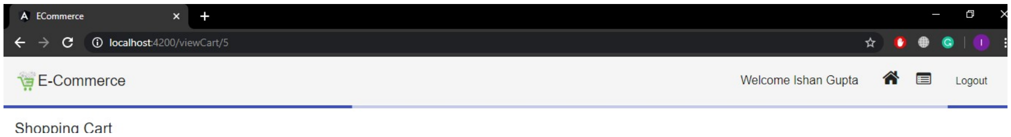


Fig 5.12: Angular Loader

CHAPTER 6 – CONCLUSIONS

6.1 Conclusion

The fundamental motivation behind the proposed task is to build up a web application that can give a smooth encounter to the client while shopping through this website. With the assistance of this venture I had the option to get profound information on Angular structure, how it works, different modules, segments, administrations included while building the application. The hands-on understanding on innovation and confronting genuine issues made my learning much all the more fascinating.

Generally, I will depict my preparation as a positive and educational experience. In the main stage I experienced a couple of issues, since it was an entirely different field for me. After some time I calculated my issues lastly begin with it. Continually evaluating this task by my guide and giving satisfactory criticisms helped me a ton and It further improved my learning experience.

The workplace was very spurring and it was a lot of rousing to perceive how a portion of the engineers had the option to autonomously execute their inclinations and plan, and perform and execute their own tests. It was an astonishing encounter to be a piece of building the application that we use and learning the foundation usefulness of how the client takes a gander at things and what the dynamic hand is returning to.

It was likewise energizing to locate another IT stage, where numerous engineers were effectively adding to one another and helping each other accomplish the organization's long and transient objectives. Then again, getting the chance to take a shot at a live venture was one of the most energizing and testing encounters. It makes me enormous pleased to introduce this task report which shows all the difficult work and endeavors that I had set up during my 3 months mechanical preparing. On the off chance that I needed to summarize my whole training period on a solitary line, I would state this was the start of reinforcing my profession.

6.2 Future Scope

- As per now the user must be logged in to view the product dashboard but there should be an option of login as a guest or skip sign which will allow users to add items to cart without signing in, as seen in other eCommerce applications.

- Currently, the admin panel is under construction which is very important to manage the site.

- The layout can be improved by integrating MDC which will help in building device-friendly sites compatible with devices of different screen sizes such as phones, laptops, etc.

- There must be a feature that keeps track of no of users who are logged in as guests and at a particular time how many users are accessing this site.

- Application's performance can be increased by using more route guards which will lead to better and smooth user experience.

- Different deals can be added on the website that will attract more no of customers, increasing the scalability of our site.

REFERENCES

1. https://acquire.io/blog/technology-drive-commerce-success/
2. https://en.wikipedia.org/wiki/Angular_(web_framework)
3. https://www.nativescript.org/nativescript-is-how-you-build-native-mobile-apps-with-angular
4. https://developer.mozilla.org/en-US/docs/Web/JavaScript
5. https://www.tutorialrepublic.com/css-tutorial/,
   https://www.tutorialrepublic.com/javascript-tutorial/
6. https://getbootstrap.com/
7. https://www.intertech.com/Blog/angular-module-tutorial-application-structure-using-modules/
8. https://dzone.com/articles/what-is-a-service-in-angular-js-why-to-use-it
9. https://angular.io/,
   https://material.angular.io/
10. https://www.whizlabs.com/blog/project-life-cycles-in-project-management/
11. https://code.visualstudio.com/
12. https://www.pathinteractive.com/blog/design-development/rendering-a-webpage-with-google-webmaster-tools/
13. https://phpenthusiast.com/blog/what-is-observable-in-angular
14. https://medium.com/compute-and-beyond/working-with-http-interceptors-angular-31067ffaeb2f ,
    https://itnext.io/understanding-angular-interceptors-405b84d7ad69
15. https://chercher.tech/jira/agile-methodology
16. https://www.npmjs.com/package/json-server#getting-started
17. https://medium.com/@ryanchenkie_40935/angular-authentication-using-route-guards-bf7a4ca13ae3

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 07/14/2020

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report [✓ B.Tech Project Report] Paper

Name: ISHAN GUPTA **Department:** CSE **Enrolment No** 161261

Contact No. 7837100474 **E-mail.** GUPTAISHAN32@GMAIL.COM

Name of the Supervisor: DR. JAGPREET SIDHU

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters):

ECOMMERCE WEBSITE - ANGULAR

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 53
- Total No. of Preliminary pages = 9
- Total No. of pages accommodate bibliography/references = 1

Ishan

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at 14%(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com