**PROFESSIONAL TRAINING REPORT**

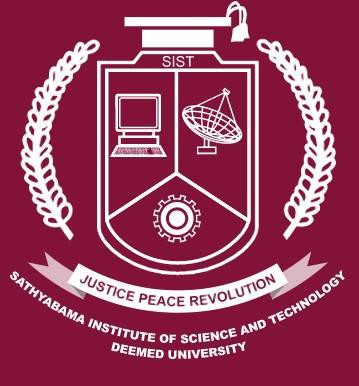
**at**

**Sathyabama Institute of Science and Technology (Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering By

**CHINTHLACHERUVU GOVARDHAN REDDY**

**(Reg.No.39110235)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

**CHENNAI - 600119, TAMILNADU**

**NOVEMBER 2021**

Text

Description automatically generated

###### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafied work of **CHINTHALACHERUVU** **GOVARDHAN REDDY39110235)**

who carried out the project entitled

“ **ONLINE ELECTRONIC STORES** ”

under my supervision from September 2021 to November 2021.

**Internal Guide Ms. Vinodhini ,**

**Head of the Department**

**Dr. L. Lakshmanan, M.E., PH.D.,**

###### Submitted for Viva voce Examination held on

**Internal Examiner External Examiner**

###### DECLARATION

I, **CHINTHALACHERUVU GOVARDHAN REDDY** hereby declare that the project report entitled **YOU & ME ONLINE ELECTRONIC STORE** done by me under the guidance of **Ms.Vinodhini** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

**DATE:**

**PLACE: SIGNATURE OF THE**

**CANDIDATE**

**ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, **Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms.Vinodhini** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

**TRAINING CERTIFICATE**

### ABSTRACT

This project is a web based shopping system for an existing

shop. The project objective is to deliver the online shopping application .

This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet.

Thus the customer will get the service of online shopping and

home delivery from his favorite shop. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains.

If shops are providing an online portal where their customers

can enjoy easy shopping from anywhere, the shops won’t be long any more customers to the trending online shops such as flipcart or ebay.

## 

## TABLE OF CONTENTS

### CHAPTE TITLE PAGE No

**R No.**

ABSTRACT 6

LIST OF FIGURES 9

1. **INTRODUCTION 12**

* 1. PROJECT OBJECTIVES 12
  2. PROJECT IMPLEMENTATION 12

**AIM AND SCOPE OF THE PRESENT**

#### 2. 14

**INVESTIGATION**

2.1 PROJECT AIM 13

2.2 PROJECT OVERVIEW 13

2.3 PROJECT SCOPE 13

1. **EXPERIMENTAL OR MATERIALS AND**

**METHODS, ALGORITHMS USED**

* 1. MODULES 14

* 1. EXISTING SYSTEM 14

* 1. PROPOSED SYSTEM 14

* 1. SYSTEM REQURIMENT SPECIFICATION 14

3.4.1GENERAL DESCRIPTION 15

3.4.2SYSTEM OBJECTIVE 15

3.4.3SYSTEM REQUIRMENT 15

3.4.1.1NON FUNCTIONAL REQURIMENT 15

3.4.1.2FUNCTIONAL REQURIMENT 15

* 1. INPUT AND OUTPUT DESIGN 16

3.5.1 INPUT DESIGN 16

3.5.2 OUTPUT DESIGN 16

* 1. DATABASE (JSON) 16
  2. SYSTEM TOOLS 17
     1. FRONT END 17
     2. BACK END 18

#### 4. 19

**RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS**

4.1 CONTROL FLOW DIAGRAM 19

4.2 E-R DIAGRAM 21

**5 APPENDIX 23**

**A.SCREEN SHOT 23**

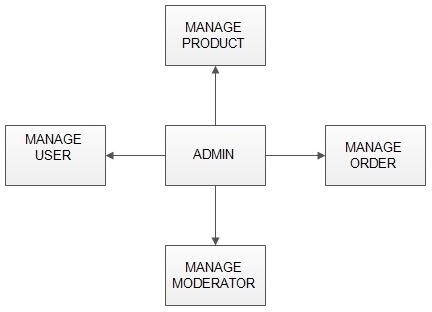
**B.SOURCE CODE 27**

REFERENCES 48

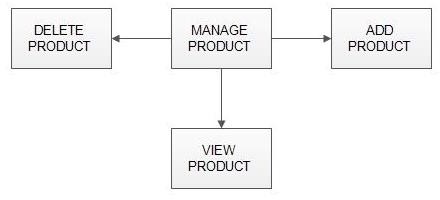
CONCLUSION 47

**LIST OF FIGURES**

* + **1.1 ADMIN MODULE**



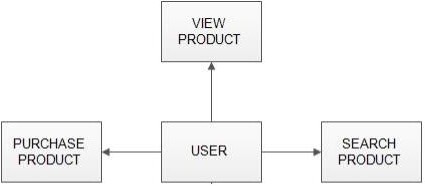
* **1.2 MANAGE PRODUCT**



* + **1.3 MANAGE ORDER**



* **1.4 USER MODULE**



# CHAPTER-1

### INTRODUCTION

This project is a web based shopping system for an existing shop. The project objective is to deliver the online shopping application. Online shopping is the process whereby consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet. Thus the customer will get the service of online shopping and home delivery from his favorite shop .

* 1. **PROJECT OBJECTIVES**

The objective of the project is to make an application in web platform to purchase items in an existing shop. In order to build such an application complete web support need to be provided. A complete and efficient web application which can provide the online shopping experience is the basic objective of the project.

* 1. **PROJECT IMPLEMENTATION**
     + Let's first install the Angular CLI. It's a neat tool that automates many development tasks. To install it, open a terminal and type the following command:
     + npm install -g @angular/cli
     + Once installed, create a new project with the following command: ng new YOU & ME -angular
     + A prompt will appear asking you if you want to enable strict mode. Select yes. This will enable a few other settings that will help to catch bugs ahead of time.
     + In your directory, you now see your project repo. Type the following commands to go in it:
     + ng serve --open
     + ng serve will build the app, while the --open option will open up a browser to http://localhost:4200/. You should now see Angular's generic template page.
     + Divide the components into header , cart , product , product-info and login
     + And finally for the backend to get the product details using json

**CHAPTER 2**

**AIM AND SCOPE OF THE PRESENT INVESTIGATION**

* 1. **PROJECT AIM**

The main aim of the Online Shopping System is to manage the details of Shopping, Internet, Customer. It manages all the information about Shopping, Products, Customer, Shopping

* 1. **PROJECT OVERVIEW**

Online shopping is a form of electronic commerce which allows consumers to buy directly from the website using web browser. consumers find a products of interest by visiting the website of the retailer directly or by searching among alternative vendor using a topic search engine. The online store usally enable shopper to use “search” features to find specific model, brand or models etc. online customers must have access internet to visit the website.

It provides requirements specific to the products , such as the requirement for the external interface , functional and behavioural requriements.

Professors can go through the entire document while reviewing it to get the general overviewof the product and to check the requirements of the application whereas clients may skipviewing the specific requirements and concentrate on reading the quality assurances and the performance and safety requirements.

###### PROJECT SCOPE

This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. The system recommends a facility to accept the orders 24\*7 and a home delivery system which can make customers happy. If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won’t be losing any more customers to the trending online shops such as flipcart or ebay. Since the application is available in the Smartphone it is easily accessible and always available.

# CHAPTER 3

**EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED**

* 1. **MODULES**

The system after careful analysis has been identified to be presented with the following modules and roles.

The modules involved are:

* + - Administrator
    - Users
* **ADMINISTRATOR :**

The administrator is the super user of this application. Only admin have access into this admin page. Admin may be the owner of the shop. The administrator has all the information about all the users and about all products. This module is divided into different sub-modules.

1. Manage Moderators
2. Manage Products
3. Manage Users

###### EXISTING SYSTEM

The current system for shopping is to visit the shop manually and from the available product choose the item customer want and buying the item by payment of the price of the item .

1. It is less user-friendly.
2. User must go to shop and select products.
3. It is difficult to identify the required product.
4. Description of the product limited.
5. It is a time consuming process
6. Not in reach of distant users.

###### PROPOSED SYSTEM

In the proposed system customer need not go to the shop for buying the products. He can order the product he wish to buy through the application in his Smartphone. The shop owner will be admin of the system. Shop owner can appoint moderators who will help owner in managing the customers and product orders. The system also recommends a home delivery system for the purchased products.

###### SYSTEM REQUIREMENT SPECIFICATION

* + 1. **GENERAL DESCRIPTION**

Product Description: The system consists of two parts .A web application which can provide the online shopping service and an android application for the customer to access the web service from his Smartphone. Web application should be able to help the customer for selecting his item and to help the owner in managing the orders from the customers.

###### SYSTEM OBJECTIVES

* To provide an android application for online shopping of products in an existing shop.
* To provide a online shopping web site for the same shop.

**4.SYSTEM REQUIREMENTS**

* + - 1. **NON FUNCTIONAL REQUIREMENTS**
         1. EFFICIENCY REQUIREMENT When an online shopping cart android application implemented customer can purchase product in an efficient manner.
         2. RELIABILITY REQUIREMENT The system should provide a reliable environment to both customers and owner. All orders should be reaching at the admin without any errors.
         3. USABILITY REQUIREMENT The android application is designed for user friendly environment and ease of use.
         4. IMPLEMENTATION REQUIREMENT Implementation of the system using css and html in front end with jsp as back end and it will be used for database connectivity. And the database part is developed by mysql. Responsive web designing is used for making the website compatible for any type of screen.
         5. DELIVERY REQUIREMENT The whole system is expected to be delivered in four months of time with a weekly evaluation by the project guide.

###### FUNCTIONAL REQUIREMENTS USER

* USER LOGIN Description of feature This feature used by the user to login into system. A user must login with his user name and password to the system after registration. If they are invalid, the user not allowed to enter the system.
* Functional requirement
* Username and password will be provided after user registration is confirmed.
* Password should be hidden from others while typing it in the field

###### LOGIN NEW USER

**Description of feature**

A new user will have to login in the system by providing essential details in order to view the product details.

**Functional requirement**

- System must ensure that, only a registered customer can purchase items

**Description of feature** : The user can add the desired product into his cart by clicking add to cart option on the product. He can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove.On successful submitting the cart will become empty.

-Admin account should be secured so that only owner of the shop can access that account

* 1. **INPUT AND OUTPUT DESIGN**
     1. **INPUT DESIGN**

Input design is the link that ties the information system into the world of its users. The input design involves determining the inputs, validating the data, minimizing the data entry and provides a multi-user facility. Inaccurate inputs are the most common cause of errors in data processing. Errors entered by the data entry operators can be controlled by input design. The user-originated inputs are converted to a computer based format in the input design. Input data are collected and organized into groups of similar data. Once identified, the appropriate input media are selected for processing. All the input data are validated and if any data violates any conditions, the user is warned by a message.

* + 1. **OUTPUT DESIGN**

Computer output is the most important and direct source of information to the user. Output design is a very important phase since the output needs to be in an efficient manner. Efficient and intelligible output design improves the system relationship with the user and helps in decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. The output module of this system is the selected notifications.

* 1. **DATABASE (JSON)**
* Json stands for java script object notation it is the way to transmit json data over text.
* Json can be written in a simple way just by specifying their characterstics.
* Example : -

{

name: 'Peter',

age: 29, 10xEnigneer: true,

}

* Json server acts as backend allowing REST API’S repeat your website and database you configure ngrok allows you to tunnel network request over the internet in the specific address.
* Json is mostly used in send data via http in the contest of web applications.
  1. **SYSTEM TOOLS**
     + - The various system tools that have been used in developing both the front end and the back end of the project are being discussed in this chapter.
     1. **FRONT END:**

HTML, CSS, JAVA SCRIPT , JSP are utilized to implement the frontend.

##### Java Server Pages (JSP)

Different pages in the applications are designed using jsp. A Java Server Pages component is a type of Java servlet that is designed to fulfil the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands. Using JSP, one can collect input from users through web page.

##### HTML (Hyper Text Markup Language)

HTML is a syntax used to format a text document on the web.

[HTML elements](https://en.wikipedia.org/wiki/HTML_element) are the building blocks of HTML pages.

With HTML constructs, [images](https://en.wikipedia.org/wiki/HTML_element#Images_and_objects) and other objects such as [interactive](https://en.wikipedia.org/wiki/Fieldset) [forms](https://en.wikipedia.org/wiki/Fieldset) may be embedded into the rendered page.

##### CSS (Cascading Style Sheets)

CSS is a style sheet language used for describing the look and formatting of a document written in a markup language. CSS stands for Cascading Style Sheets.

CSS describes how HTML elements are to be displayed on screen, paper, or in other media CSS saves a lot of work. It can control the layout of multiple web pages all at once External stylesheets are stored in CSS files.

##### Java Script

JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages.

It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document.

It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser.

Since then, it has been adopted by all other graphical web browsers.

**3.7.2 BACK END**

The back end is implemented using JSON & NODE JS which is used to design the databases.

* + - * **JSON**

JSON or JavaScript Object Notation is a lightweight text-based open standard designed for human-readable data interchange. The JSON format was originally specified by Douglas Crockford, and is described in RFC 4627. The official Internet media type for JSON is application/json. The JSON filename extension is .json.

* + - * **NODE JS**

Node.js is a server-side platform built on Google Chrome's JavaScript Engine.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Node.js = Runtime Environment + JavaScript Library

# CHAPTER 4

* 1. **DATA CONTROL FLOW**

Diagram

Description automatically generated

* A Data Flow Diagram (DFD) is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system.

This network is constructed by using a set of symbols that do not imply any physical implementation. It has the purpose of clarifying system requirements and identifying major transformations.

So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. DFD can be considered to an abstraction of the logic of an information-oriented or a process- oriented system flow-chart. For these reasons DFD’s are often referred to as logical data flow diagrams.

* + Diagram

    Description automatically generated**ADMIN DFD**
  1. **E-R DIAGRAMS**
* **LOGIN**

Diagram

Description automatically generated

* **USER DETAILS**

Diagram

Description automatically generated

* **PRODUCT DETAILS**

Diagram

Description automatically generated

**CHAPTER 5**

* + **APPENDIX**
* **SCREEN SHOTS**
* **HOME PAGE**

Graphical user interface, application, PowerPoint

Description automatically generated

-When we visit the website first we can able to see the home page which consists of all product ,searchbar and the cart symbol with the functionality.

-From the backend we can able to get the product details which are divided into the categories . the search bar will be provided the user can able to select the specific categories such as laptops,mobile,tv,airpods etc….

-And also the cart functionality will be provided such as emptycart,shop more and place order.

-When we click on add to cart then it will to shown as below figure

### CART PAGE

### Graphical user interface, website Description automatically generated

-The functionality such as the empty cart,shop more and the check out

-But when we can able to see according to the json data we are written in the code such as the sno,item,item image,item description and the item price will be provide according to it

### LOGIN PAGE

Graphical user interface, application

Description automatically generated

-when we click on any item then the page will be redirected to the login page according to specific item or the product we had selected because it is the statement given by the problem

-In the login page we can able to see the email id,password and the login button.in this no need to reg we can directly login in the page and see the product description of the product we had selected

-We can also able to follow our website on the twitter and the facebook

* **PRODUCT DESCRIPTION**

Graphical user interface, text

Description automatically generated

-If we selected a product and we had logined successfully then we can able to see the specific product description

### SOURCE CODE

**To make the project easier and flexible we had divided the project into components such as cart , header , login , products & product- info.**

**And collied all components in index.html and the app component.html**

* **index.html**

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>You & Me</title>

<base href="/">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="icon" type="image/x-icon" href="favicon.ico">

<link href="[https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css](https://cdn.jsdelivr.net/npm/bootstrap%405.0.2/dist/css/bootstrap.min.css)" rel="stylesheet" integrity="sha384- EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<script src="https://kit.fontawesome.com/b7926f3752.js" crossorigin="anonymous"></script>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/5.15.2/css/all.min.css"/>

</head>

<body>

<script src="[https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js](https://cdn.jsdelivr.net/npm/bootstrap%405.0.2/dist/js/bootstrap.bundle.min.js)" integrity="sha384- MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<app-root></app-root>

</body>

</html>

#### There are four types of components

* Header
* Cart
* Product
* Login
* Product-info

#### Header component

**header.html**

<header>

<div class="navbar navbar-dark shadow-sm">

<div class="navbar-brand align-item-center d-flex">

<i style="font-size: 40px;" class="fas fa-shipping-fast">&nbsp;

</i>

<strong style="font-size: 30px;">You & Me</strong>

<div class="form-group">

<input type="text" class="form-control" [(ngModel)]="searchTerm" (keyup)="search($event)" placeholder="search products">

</div>

<span class="fas fa-search search-icon"></span>

</div>

<button routerLink="cart" class="btn " (click)="cart()">

<i style="font-size: 30px;" class="fas fa-cart-plus"></i>

<div style="font-size: 10px;" class="badge bg-danger">{{totalItem}}</div>

</button>

</div>

</header>

* **header.scss**

.form-control {

border-radius: 3px;

width: 600px;

margin-left: 20px;

}

.search-icon {

position: absolute;

z-index: 10;

right: 47%;

top: 20px;

color:rgb(136, 79, 5);

}

.navbar{

background:rgb(136, 79, 5);

}

.btn

{

background:rgb(136, 79, 5);

color: white;

}

* **header.ts**

import { Component, OnInit } from '@angular/core';

import { CartService } from 'src/app/service/cart.service';

import { Router, ActivatedRoute } from '@angular/router';

@Component({

selector: 'app-header',

templateUrl: './header.component.html',

styleUrls: ['./header.component.scss']

})

export class HeaderComponent implements OnInit {

public totalItem : number = 0;

public searchTerm !: string;

constructor(private cartService : CartService,private router: Router) { }

ngOnInit(): void {

this.cartService.getProducts()

.subscribe(res=>{

this.totalItem = res.length;

})

}

search(event:any){

this.searchTerm = (event.target as HTMLInputElement).value;

console.log(this.searchTerm);

this.cartService.search.next(this.searchTerm);

}

cart()

{

this.router.navigate(['/cart']);

}

}

#### Cart component

* **Cart.html**

<div><app-header></app-header></div>

<ng-container \*ngIf="products.length !=0">

<div class="container">

<div class="card-table">

<div class="cart-product">

<table class="table table-responsive">

<thead>

<tr>

<th>Sr.No</th>

<th>Product Name</th>

<th>Product Image</th>

<th>Description</th>

<th>Price</th>

<th>Quantity</th>

<th>Total</th>

<th>Action</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let item of products; let i = index">

<td>{{i+1}}</td>

<td>{{item.title}}</td>

<td><img style="width: 120px;" src="{{item.image}}" alt=""></td>

<td style="width: 25%;">{{item.description}}</td>

<th style="width: 12%;">{{item.price}}</th>

<td style="width: 12%;">{{item.quantity}}</td>

<td style="width: 12%;">{{item.total}}</td>

<td>

<button (click)="removeItem(item)" class="btn btn-danger"><i class="fas fa-trash-

alt"></i></button>

</td>

</tr>

<tr>

<td colspan="4"></td>

<td><button (click)="emptycart()" class="btn btn-danger">Empty Cart</button></td>

<td><button routerLink="/products" class="btn btn-primary">Shop More</button></td>

<td><button class="btn btn-success">Checkout</button></td>

<td><strong>Grand Total : ${{grandTotal}}</strong></td>

</tr>

</tbody>

</table>

</div>

</div>

</div>

</ng-container>

<ng-container \*ngIf="products.length ==0">

<div class="container">

<div class="card">

<h5 class="card-title">My Cart</h5>

</div>

<div class="center">

<img src="https://rukminim1.flixcart.com/www/800/800/promos/16/05/2019/d438a32e-765a- 4d8b-b4a6-520b560971e8.png?q=90" alt="">

<h4>Your cart is empty!</h4>

<h6>Add item to it now</h6>

<button routerLink="/products" class="btn ">Shop Now</button>

<!---------<button class="btn btn-primary" (click)="loginpage()">Shop Now</button>--->

</div>

</div>

</ng-container>

##### Cart.scss

.card {

height: 60vh; margin: 25px; padding: 25px;

}

.center img {

text-decoration: none; color: inherit;

border: none; outline: none; height: 162px; width: 250px; margin: 20px 0px;

}

h4,

h6 {

font-weight: 400;

}

.center {

position: absolute; top: 50%;

left: 50%;

transform: translate(-50%, -50%); text-align: center;

}

.card-table { position: relative; display: flex;

flex-direction: column; min-width: none;

word-wrap: break-word; background-color: #fff; background-clip: border-box; border: 1px solid rgba(0, 0, 0, 0.2); border-radius: .25rem;

}

.center .btn {

font-size: 14px !important; margin-top: 20px !important; font-weight: 400;

padding: 12px 72px;

border-radius: 3px !important;

}

.btn{

background:rgb(136, 79, 5); color: white;

}

## Cart.ts

import { Component, OnInit } from '@angular/core'; import { CartService } from 'src/app/service/cart.service';

//import { Router, ActivatedRoute } from '@angular/router';

@Component({ selector: 'app-cart',

templateUrl: './cart.component.html', styleUrls: ['./cart.component.scss']

})

export class CartComponent implements OnInit {

public products : any = []; public grandTotal !: number;

//router: any;

constructor(private cartService : CartService,

// private router: Router,

) { }

ngOnInit(): void { this.cartService.getProducts()

.subscribe(res=>{ this.products = res;

this.grandTotal = this.cartService.getTotalPrice();

})

}

removeItem(item: any){ this.cartService.removeCartItem(item);

}

emptycart(){ this.cartService.removeAllCart();

}

//add new code for login page just for example

// loginpage(){

// console.log(32, "hello world")

// this.router.navigate(['/products']);

// }

}

## Login component

##### Login.html

<!-- <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/5.15.2/css/all.min.css"/> -->

<div class="rizwan">

<div class="container">

<form action="#">

<div class="title">Login</div>

<div class="input-box underline">

<input type="text" placeholder="Enter Your Email" required>

<div class="underline"></div>

</div>

<div class="input-box">

<input type="password" placeholder="Enter Your Password" required>

<div class="underline"></div>

</div>

<div class="input-box button">

<button class="log" (click)="login()">Login</button>

</div>

</form>

<div class="option">or Connect With Social Media</div>

<div class="twitter">

<a href="#"><i class="fab fa-twitter"></i>Sign in With Twitter</a>

</div>

<div class="facebook">

<a href="#"><i class="fab fa-facebook-f"></i>Sign in With Facebook</a>

</div>

</div>

</div>

#### Login.scss

.rizwan{

background-image:

url(https://png.pngtree.com/thumb\_back/fw800/back\_our/20190619/ourmid/pngtree-taobao-vector- creative-technology-online-shopping-illustration-computer-finger-poster-image\_131803.jpg); background-size: 100%;

background-repeat: no-repeat; html, body{

display: grid; height: 100vh; width: 100%;

place-items: center;

background: linear-gradient(to right, #9ac0cc 0%, #3E8DA8 100%);

}

::selection{ background: #9ac0cc;

}

.container{

//background:lavender;

//background-image: url(https://i.pinimg.com/736x/06/26/13/062613231a04bcea7bdbe83a8ad1c63c.jpg);

background-size: cover; max-width: 350px; width: 100%;

padding: 25px 30px; border-radius: 5px;

box-shadow: 0 10px 10px rgba(0, 0, 0, 0.15); margin-left: 200px;

margin-top: 70px;

}

.container form .title{ font-size: 30px; font-weight: 600;

margin: 20px 0 10px 0; position: relative; color:black;

font-size: xx-large; font-weight: bolder;

}

.container form .input-box{ width: 100%;

height: 45px; margin-top: 25px; position: relative;

}

.container form .input-box input{ width: 100%;

height: 100%; outline: none; font-size: 16px; border: none;

}

.container form .underline::before{ content: '';

position: absolute; height: 2px;

width: 100%; background: #ccc; left: 0;

bottom: 0;

}

.container form .underline::after{ content: '';

position: absolute; height: 2px;

width: 100%;

background: linear-gradient(to right, #3E8DA8 0%, #9ac0cc 100%); left: 0;

bottom: 0;

transform: scaleX(0); transform-origin: left; transition: all 0.3s ease;

}

.container form .input-box input:focus ~ .underline::after,

.container form .input-box input:valid ~ .underline::after{ transform: scaleX(1);

transform-origin: left;

}

.container form .button{ margin: 40px 0 20px 0;

}

.container .input-box input[type="submit"]{

background: linear-gradient(to right, #3E8DA8 0%, #9ac0cc 100%); font-size: 17px;

color: #fff;

border-radius: 5px; cursor: pointer; transition: all 0.3s ease;

}

.container .input-box input[type="submit"]:hover{ letter-spacing: 1px;

background: linear-gradient(to left, #99004d 0%, #ff0080 100%);

}

.container .option{ font-size: 14px; text-align: center;

}

.container .facebook a,

.container .twitter a{ display: block; height: 45px; width: 100%;

font-size: 15px;

text-decoration: none; padding-left: 20px; line-height: 45px; color: #fff;

border-radius: 5px; transition: all 0.3s ease;

}

.container .facebook i,

.container .twitter i{ padding-right: 12px; font-size: 20px;

}

.container .twitter a{

background: linear-gradient(to right, #00acee 0%, #1abeff 100%); margin: 20px 0 15px 0;

}

.container .twitter a:hover{

background: linear-gradient(to left, #00acee 0%, #1abeff 100%); margin: 20px 0 15px 0;

}

.container .facebook a{

background: linear-gradient( to right, #3b5998 0%, #476bb8 100%); margin: 20px 0 50px 0;

}

.container .facebook a:hover{

background: linear-gradient( to left, #3b5998 0%, #476bb8 100%); margin: 20px 0 50px 0;

}

.log{

display: block; height: 45px; width: 100%; font-size: 15px;

text-decoration: none; padding-left: 20px; line-height: 45px; color:black;

border-radius: 5px; transition: all 0.3s ease; text-align: center;

}

}

.log{

background:linear-gradient(rgba(12, 148, 189, 0.5),rgba(0,0,30,0));

}

## Login.ts

import { Component, OnInit } from '@angular/core'; import { Router, ActivatedRoute } from '@angular/router'; @Component({

selector: 'app-login',

templateUrl: './login.component.html', styleUrls: ['./login.component.scss']

})

export class LoginComponent implements OnInit {

constructor(private router: Router) { } login(){

console.log(32, "hello world") console.log(history.state.sts); // GET DATA

//if(history.state.sts=="1"){

//this.router.navigate(['/products-info']); this.router.navigate(['/product-info'],{ state: { sts:history.state.sts } });

// ,

//}

}

ngOnInit(): void {

}

}

* **Product-info**
* **Product-info.html**

<div><app-header></app-header></div>

<!-- {{n1}} -->

<!-- <div class="container-fluid">

<div class="row">

<ng-container >

<div class="card col-md-3">

<img src="{{product\_image}}" alt="" style="width: 120px;height: 120px;">

<h5 style="white-space: nowrap;overflow: hidden;text-overflow: ellipsis;max-width:

100ch;">{{product\_item}}</h5>

<p style="white-space: nowrap;overflow: hidden;text-overflow: ellipsis;max-width:

43

43

100ch;">{{product\_des}}</p>

<p><strong>Price:</strong> ${{product\_price}}</p>

</div>

</ng-container>

* **Product-info.ts**

</div>

</div> -->

<ng-container >

<div class="container">

<div class="card-table">

<div class="cart-product">

<table class="table table-responsive">

<thead>

<tr>

<th>Product Name</th>

<th>Product Image</th>

<th>Description</th>

<th>Price</th>

</tr>

</thead>

<tbody>

<tr>

<td>{{product\_item}}</td>

<td><img style="width: 120px;" src="{{product\_image}}" alt=""></td>

<td style="width: 25%;">{{product\_des}}</td>

<td style="width: 12%;">{{product\_price}}</td>

</tr>

</tbody>

</table>

</div>

</div>

</div>

</ng-container>

Product-info.ts

import { Component, OnInit } from '@angular/core';

@Component({

selector: 'app-product-info',

templateUrl: './product-info.component.html',

styleUrls: ['./product-info.component.scss']

})

export class ProductInfoComponent implements OnInit {

name: any;

n1: any;

product\_image: any;

product\_item: any;

product\_des: any;

44

44

product\_price: any;

constructor() { }

ngOnInit(): void {

console.log(history.state.sts)

this.name = history.state.sts

console.log(16,this.name)

this.product\_image = this.name.image

this.product\_item = this.name.title

this.product\_des = this.name.description

this.product\_price = this.name.price;

}

}

#### There are two services used in it they are

#### 1. api.service.ts

#### 2. cart.service.ts

#### Api.service.ts

#### import { HttpClient } from '@angular/common/http';

#### import { Injectable } from '@angular/core';

#### import {map} from 'rxjs/operators';

#### @Injectable({

#### providedIn: 'root'

#### })

#### export class ApiService {

#### constructor(private http : HttpClient) { }

#### getProduct(){

#### return this.http.get<any>("http://localhost:3000/posts")

#### .pipe(map((res:any)=>{

#### return res;

#### }))

#### }

#### }

#### Cart.service.ts

#### import { Injectable } from '@angular/core';

#### import { BehaviorSubject } from 'rxjs';

#### @Injectable({

#### providedIn: 'root'

#### })

#### export class CartService {

#### public cartItemList : any =[]

#### 45

#### 45

#### public productList = new BehaviorSubject<any>([]);

#### public search = new BehaviorSubject<string>("");

#### constructor() { }

#### getProducts(){

#### return this.productList.asObservable();

#### }

#### setProduct(product : any){

#### this.cartItemList.push(...product);

#### this.productList.next(product);

#### }

#### addtoCart(product : any){

#### this.cartItemList.push(product);

#### this.productList.next(this.cartItemList);

#### this.getTotalPrice();

#### console.log(this.cartItemList)

#### }

#### getTotalPrice() : number{

#### let grandTotal = 0;

#### this.cartItemList.map((a:any)=>{

#### grandTotal += a.total;

#### })

#### return grandTotal;

#### }

#### removeCartItem(product: any){

#### this.cartItemList.map((a:any, index:any)=>{

#### if(product.id=== a.id){

#### this.cartItemList.splice(index,1);

#### }

#### })

#### this.productList.next(this.cartItemList);

#### }

#### removeAllCart(){

#### this.cartItemList = []

#### this.productList.next(this.cartItemList);

#### }

#### }

#### And finally we include them all in the app components

##### 1.app-routing.module.ts

import { NgModule } from '@angular/core'; import { RouterModule, Routes } from '@angular/router'; import { CartComponent } from './component/cart/cart.component'; import { LoginComponent } from './component/login/login.component'; import { ProductsComponent } from './component/products/products.component'; import { ProductInfoComponent } from './product-info/product-info.component'; //import { LoginComponent } from './component/login/login.component'; const routes: Routes = [

{path:'', redirectTo:'products',pathMatch:'full'},

{path:'products', component: ProductsComponent},

{path:'cart', component: CartComponent},

{path:'product-info', component: ProductInfoComponent},

{path:'login',component:LoginComponent}

];

@NgModule({

imports: [RouterModule.forRoot(routes)], exports: [RouterModule]

})

export class AppRoutingModule { }

**2.app.component.html**

<router-outlet></router-outlet>

##### 3.app.component.ts

import { Component } from '@angular/core';

@Component({

selector: 'app-root', templateUrl: './app.component.html',

styleUrls: ['./app.component.scss']

})

export class AppComponent {

title = 'add-to-cart';

}

##### 4.app.module.ts

import { NgModule } from '@angular/core';

import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module'; import { AppComponent } from './app.component'; import { HeaderComponent } from './component/header/header.component'; import { CartComponent } from './component/cart/cart.component'; import { ProductsComponent } from './component/products/products.component'; import { HttpClientModule } from '@angular/common/http';

import { FilterPipe } from './shared/filter.pipe';

import { FormsModule, ReactiveFormsModule } from '@angular/forms'; import { LoginComponent } from './component/login/login.component'; import { ProductInfoComponent } from './product-info/product-info.component';

//import { LoginComponent } from './login/login.component';

//import { LoginComponent } from './login/login.component';

@NgModule({

declarations: [

AppComponent,

HeaderComponent,

CartComponent,

ProductsComponent,

FilterPipe,

LoginComponent,

ProductInfoComponent,

//LoginComponent,

//LoginComponent

],

imports: [

BrowserModule,

AppRoutingModule,

HttpClientModule,

FormsModule,

ReactiveFormsModule,

//LoginComponent

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

##### CONCLUSION

Based on the implementation of the research that has been done, can be concluded that this research produces e-commerce on the You & me.

The project entitled “You & me” Online shopping system was completed successfully. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming.

The purpose of this project was to develop a web application for purchasing items from a shop. This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using html & css, usage of responsive templates, designing of web applications, and management of database using json.

The entire system is secured. Also the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project.

This project has given us great satisfaction in having designed an application which can be implemented to any nearby shops or branded shops selling various kinds of products by simple modifications. There is a scope for further development in our project to a great extend. A number of features can be added to this system in future like providing moderator more control over products so that each moderator can maintain their own products. Another feature we wished to implement was providing classes for customers so that different offers can be given to each class.

System may keep track of history of purchases of each customer and provide suggestions based on their history. These features could have implemented unless the time did not limited us.

##### REFERENCES

1. K. C. Laudon and C. G. Traver, E-commerce: business, technology, society, 10th ed., Upper Saddle River: Pearson, 2014.
2. A. Q. Haviv, MEAN Web Development, Birmingham: Packt Publishing, 2014.
3. V. Waghade and B. V. Chaudhari, "Study Of AngularJS with Other Frameworks", International Journal of Research in Computeabr & Information, vol. 1, no. 2, pp. 151154, 2016.
4. H. D. Purnomo, D. A. Saputro, R. Somya and C. Fibriani, 'The Application of Restful Web Service and JSON for Poultry Farm Monitoring System", Journal of Electrical Engineering and Computer Sciences, pp. 25-30, 2016.
5. D. Jacobson, G. Brail and D. Woods, APIs: A Strategy Guide, California: O’Reilly Media, 2011.
6. B. Nugroho, Latihan Membuat Aplikasi Web PHP dan dengan Dreamweaver, Yogyakarta: Gava Media, 2009.
7. K. Kendall and J. Kendall, System Analysis and Design 8th Edition, Upper Saddle River: Prentice Hall, 2011.
8. B. Hambling and P. Goethem, User Acceptance Testing: a step-by-step guide, Chippenham: BCS, 2013.
9. A. Dennis, B. H. Wixom and R. M. Roth, System Analysis and Design, New Jersey: John Wiley & Sons, Inc., 2012.
10. R. Pressman, “Rekayasa Perangkat Lunak Buku 1,” in Pendekatan Praktisi Edisi 7, Yogyakarta, Andi, 2010, p. 668.
11. G. D. Everett and R. McLeod, Software Testing: testing across the entire software development life cycle, New Jersey: John Wiley & Sons, Inc., 2007. [12] L.G.Vandenbosch, M.Nicoletti, “An industry case study: a mobile-based business strategy to improve the customer care service in a major retail company”, 5º Simposio Argentino de Informática Industrial, 2016

[13] W.Chansuwath, T.Senivongse, “A Model-Driven Development of Web

Applications Using AngularJS Framework”, ICIS, June 26-29, 2016, Okayama, Japan [14] M.A. Jadhav, B.R. Sawant, A.Deshmukh, ” Single Page Application using

AngularJS”, (IJCSIT)

**THANK YOU**