E COMMERCE WEB APPLICATION

A project submitted in partial fulfilment of the

requirements for the award of the degree of

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by: Anupreet Singh

Roll number:11911063

Group Number: - 118

Supervised by:

Dr. MUKESH MANN

Assistant professor

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, SONEPAT – 131201, HARYANA, INDIA

ACKNOWLEDGEMENTS

The success and the outcome of this project required ceaseless guidance and assistance, my team members and I are extremely privileged to have got this all along the project.

We would like to take this opportunity to acknowledge all the people who have helped us whole heartedly in every stage of this project.

We are indebtedly grateful to (Dr.) Mukesh Mann, Assistant professor, CSE, IIIT SONEPAT for providing this opportunity in the first place and giving us all the support and guidance possible, in spite of having a busy schedule.

We would also like to thank Md Arquam sir, Assistant professor, CSE, IIIT SONEPAT for guiding us through web development. We are indebtedly grateful for his help which played a very foremost part in the project and for providing us all the indispensable information for developing the web application.

Anupreet Singh

SELF DECLARATION

I hereby state that work contained in the project titled “E commerce web application” is original. I have followed the standards of the project ethics to the best of my abilities. I have acknowledged all the sources of knowledge which I have used in the project.

Name: Anupreet Singh

Roll no: 11911063

Department of Computer Science and Engineering,

Indian Institute of Information technology,

Sonepat-131201, Haryana, India

CERTIFICATE

This is to certify that Mr. Anupreet Singhhas worked on the project entitled **"**E- Commerce Web Application**"** under my supervision and guidance.

The contents of the project, being submitted to the Department of Computer Science and Engineering, IIIT SONEPAT, HARYANA, for the award of the degree of B. Tech in Computer Science and Engineering, are original and carried out by candidate himself. This project has not been submitted in full or part for award of any other degree or diploma to this or any other university.

Dr. Mukesh Mann

Supervisor

Department of Computer Science and Engineering,

Indian Institute of Information technology,

Sonepat-131201, Haryana, India

ABSTRACT

Name of the student: Anupreet Singh

Roll No: 11911063

Degree for which submitted: BTech

Department of Computer Science And Engineering

Indian Institute of Information Technology, Sonepat

Project Title**:** E Commerce web Application

Name of the project supervisor**:** Dr. Mukesh Mann

Month and year of the project submission**:** December, 2020

Our project is an E Commerce website which sells the top selling books of all time irrespective of their linguistic ties i.e., the language in which they have been printed and sold all over the world. The project is meant to focus on W/A and S/w related Development purposes rather than focusing on research point of view. The prime objective behind this topic is that there is a lot of untapped potential in the Indian market of books as there are a lot of individuals who want to buy and read only the best literature but there aren't many W/A's that provide a user-friendly platform for browsing, viewing information about and purchasing the best-selling books of all time. Users waste quite a lot of their time in finding blend of books which are both critically acclaimed and commercially successful and also to buy in real time, a problem that our web-app will solve. The platforms and programming languages used till now are CSS, HTML, SASS, Js, React, Babel, Strapi, Stripe.

The prime Results which wished to achieve are: -

That users in most parts of the country can browse the best of best books and order to their door step according to their convenience. To provide a user-friendly interface and platform for the above objectives and to challenge the E giants in the business and establishing ourselves as a profound name.

LIST OF ABBREVIATIONS

S/W	Software
Js	JavaScript
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
SASS	Syntactically Awesome Style Sheets
AUT	Application Under Test
W/A	Web Application
JSON	JavaScript Object Notation
PUW	Project Under Work
API	Application Programming Interface
SQL	Structure Query Language
RDBMS	Relational Database Management System
UI	User Interface
CDN	Content Delivery Network

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

Chapter 1
Introduction

1.1 INTRODUCTION

Brief Introduction to the Topic:

The project under work (PUW) is an E Commerce web Application for sale of the top selling Books of all time. The books put up for sale presently and in the future would be linguistically independent. The project is purely meant to focus on Web-Application and S/w related Development purposes rather than focusing on research or publication point of the view. The basic reason in choosing this topic for our practicum project is that we will provide an easier way for the novice and VR for exposure to literature which is both critically acclaimed and commercially successful. Thus, making the work a lot easier for the user which would be wasted in web surfing for finding a worthy book to read and later for purchase of the same, because there are not a lot of good websites, which would be providing the services rivaling our website.

1.2 PROBLEM OUTLINE

The sole purpose of choosing this problem is that the number of average book readers in India is sky rocketing and putting India in comparison with other countries such as China and United states. And books are beginning to be an enormous business in this new generation of India.

People are now accepting reading as a passion and a useful hobby as well. Since, the number of readers is increasing day by day so our W/A which will not only provide user with critically acclaimed and commercially successful books**,** with their description and other critical information and also provide an option to purchase them side by side and not wasting their time in:

1. Firstly, searching in some of the books which are acclaimed by both critics and common audiences.

2. Then, finding a web site which both sells and deliver these books.

If we are pursuing a hobby or a passion, why not do it the right and the best possible way which in essence are top sellers. In some foreign country's literature, history, art etc. are provided as an option by undertaking these studies as minors in educational courses. People who read a lot, display a greater knowledge of how things work and tend to be more practical than others. Also, majority of most popular movies and video games are inspired by books e.g. The Harry Potter movie series, The Witcher game and web series and also a lot of other excellent works in history (such as live plays) are inspired by book collections.
So, purchase and sale of books are turning into multi- million-dollar business.
Our website is making it easy for the users by providing a user-friendly platform which covers both tiers of book-business including browsing top selling books of all time with their description and buying them at the same time.

1.3 PROJECT OBJECTIVES

The main project objective is to deliver the application as a platform where we can sell our product and services to an already existing and upcoming gigantic audience. The prime objectives of the project other than revenue generation are an easy to use or user-friendly interface with only the best of the best books to offer. To overcome the limitations of space, time, language and location for the users.

1.4 PROJECT METHODOLOGY

The methodology used to achieve the upcoming objectives are as follows: -

The PUW will a user-friendly interface and it would be achieved by making interface a bit modern and different than the existing interfaces and it is achieved by using Js, React, CSS, Strapi and React-stripe.
Predominantly all the top selling books are not available at our local book sellers, so many a times readers tend to download these books from pirated sites and we will be overcoming this obstacle by providing these books to extensive locations throughout the country.
Our platform would be providing every minuscule detail about the product thus overcoming space, time and language barriers. Also, the books would be linguistically independent.

1.5 SCOPE OF PROJECT WORK

The E Commerce Sector has been widely and positively accepted by the Indians all over the country.

These are some stats of the retail market:

Table - 1.1 The Table shows the Indian Retail Market in the past years

YEAR	INDIAN RETAIL MARKET
2011	$470 BILLION DOLLARS
2016	$500 BILLION DOLLARS
2020 (Expected)	$600 BILLION DOLLARS

Table 1.2 This Table shows the Indian Book market in the past years

YEAR	INDIAN BOOK INDUSTRY
2011	$10 BILLION DOLLARS
2015	(UPTO) $100 BILLION DOLLARS
2020	(EXPECTED) $190 BILLION DOLLARS

From the figures, it is quite clear that book sale industry is sky rocketing year by year, the scope of E book-business escalating and a startup business for the same would be an asset rather than a liability.

Our project will be selling top selling books of all time and now the people have been indulging in reading some good art and literature and the same would be provided by our W/A.

1.5.1 LIMITATIONS: -

But every project has some Limitations and here are some for our project:

There are already some giants in the book selling business as well which are Amazon and Flipkart, and it is very difficult for a new player to quickly overthrow these giants and establish itself.

There is not very proficient investment in the Indian market for online book business. Many consumers for book purchase still prefer personal touch rather than online and people prefer face to face interaction more for to check book condition.

The particular limitation for the project is it only sells top selling books of all time and not regional art and literature such as works of Prem Chand Khushwant Singh, etc.

1.6 ORGANISATION OF PROJECT

The organization of the project mentioned is being done by dividing the work into several phases or chapters to make the project work more organized. The chapters have been as follows: -

- Chapter-1 Introduction: This is the initial phase of the project. This briefly explains the project, the objectives of the project, methodology, scope of project.

- Chapter-2 UI: In this chapter, the complete front-end work will be done, and practice will be carried out to enhance the UI and to make it interactive.

- Chapter-3 API: In this chapter, the part of the backend along with the API will be developed, and the initial phase testing will be done.

- Chapter-4 Finishing Touch: This would be the Last phase, in this the remaining part of the backend will be completed, and the system flaws will be identified and will be fixed and the project will be deployed.

1.7 SUMMARY

In the summary of chapter 1 i.e., Introduction, we have talked about the Problem Outline in subchapter 1.2 which explains why we chose this topic over others, Project Objectives which is explained in the subchapter 1.3 which briefly mentions the main objectives of the project and the methodology used to follow these objectives in subchapter 1.4.
The Scope and Limitations of the project have been discussed in subchapter 1.5 and followed by Organization of the complete project in subchapter 1.6 which mentions all the chapters in which the report work is categorized.

.

Chapter 2

Study and Review of Literature

2.1 INTRODUCTION

An e-commerce website, by definition, is a website that allows the user to buy and sell goods, digital products or services online. Trade, whether being exchange or buying and selling of goods and services has been prevalent for centuries. No one can be self-sufficient. And this brings out the need for demand and supply of goods and services. Transactions have been going on all over the world for centuries, locally, and across locations. And before user begin, user should be well aware of the tools and technologies required to build such a system or a web application. This has evolved more with the emergence of smartphones, where now, user can shop from anywhere and anytime, with a wireless device connected to the Internet. Now user can search for almost any product or service online, without having to go anywhere physically. In 2020, eCommerce sales are expected to account for 15.5 percent of retail sales worldwide.

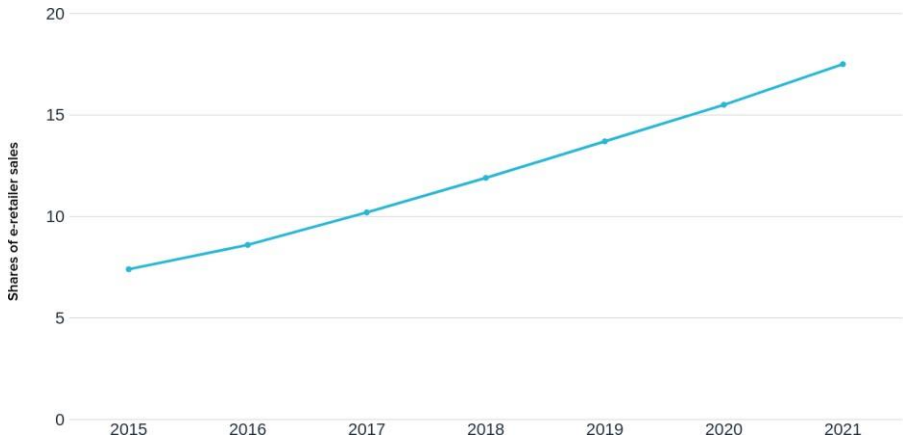

Fig. 2.1 E-commerce sales

This above figure shows the pictorial representation of e commerce sales shares of the total retail sales increasing in the past few years. Now with a single website, anything and everything can be facilitated online with a single click over any required web application. But user should be familiar with the technologies or services required to create a web-application.
This chapter presents some discussions about the relevant tools and technologies used to develop an e-commerce web application.
Some of the tools and technologies or the Pre-requisite which are used in development of front-end, UI and back-end are JavaScript, React, HTML, CSS, SASS, JSX technology, Node.js, Webpack, SQLite. Others are Babel and Webpack, Strapi, React- Strapi-Element, React-Router-Dom and Stripe Payment Processor.

2.2 FRONT-END

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a

user can see and interact with them directly. The front end of a website is the part that users interact with. Everything that user see when user navigating around the Internet, from fonts and colors to dropdown menus and sliders, is a combo of HTML, CSS, and JavaScript being controlled by user's computer's browser. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

HTML AND CSS

HTML and CSS HTML stands for Hyper Text Markup Language. It is used as the standard markup language for building web applications and web pages. It was originally developed by Tim Berners-Lee in 1991. HTML5 is the latest version of HTML.

CSS stands for Cascading Style Sheet. It is used to style web pages for different kinds of devices and screen sizes. It also saves a lot of time and stress since it can be used to style multiple web pages simultaneously. Its latest version is CSS3.

SASS

Sass stands for Syntactically awesome style sheets. It was designed by Hampton Catlin and SASS first appeared in November 2006 but its stable version was released was on March 23' 2018. Sass is a pre-processor scripting language that is interpreted or compiled into Cascading Style Sheets. Sass Script is the scripting language itself. Sass consists of two syntaxes. The original syntax, called "the indented syntax,"uses a syntax similar to Haml.

2.3 UI DEVELOPMENT

A Web user interface or Web app allows the user to interact with content or software running on a remote server through a Web browser. The content or Web page is downloaded from the Web server and the user can interact with this content in a Web browser, which acts as a client. Simply put, a good User Interface is important because it can turn potential visitors to buyers as it facilitates interactions between the user and user website or web application. An interface is a point where a user interacts with the website they're using. UI is a main part of building an engaging website.

- **Virtual design**
  Visual design improves a site's value by strategically implementing elements such as fonts, colors, and images among other things. When professionally done, visual design makes a page elegant without compromising on its function or content.

- **Interactive design**

  The interactive design looks at how users interact with technology. It uses the understanding of such interactions to create an interface with behaviors that are well thought-out. Excellent interactive design not only anticipates how a person interacts with a system but also antedates and fixes problems in good time. It may also invent new ways through which a system interacts and responds to users.

- **Information Architecture**

  Information architecture is designed to help users find the info they need to complete various tasks. It, therefore, involves labelling, structuring, and organizing the web content in a manner that makes it easily accessible and sustainable. The services and technologies used in the building of the web user interface (UI) are mentioned below.

JAVASCRIPT

JavaScript is the world's most popular programming language. JavaScript is the most popular programming language in the world. JavaScript helps user in developing great front-end as well as back-end software's using different JavaScript based frameworks like jQuery, Node.JS, React etc.

REACT JS

It is an open-source JavaScript library used in web development for building user interfaces and interactive elements and was first deployed in 2011. React makes it easier to create interactive UIs. React helps to create Declarative views which make the code more predictable and easier to debug. React is a front-end library and it runs on a browser like apache and rails. We used react to create the basic front-end framework of the project.

JSX

JSX stands for JavaScript XML. JSX allows us to write HTML in React. JSX makes it easier to write and add HTML in React. JS is standard JavaScript while JSX is an HTML-like syntax that user can use with React to (theoretically) make it easier and more intuitive to create React components. Without JSX, creating large, nested HTML documents using JS syntax would be a difficult task. JSX simply makes that process easier.

BABEL

Babel is used to make sure that everyone will be able to run user code. Babel was stably released on 27th February' 2020. Babel is a JavaScript transpiler that converts edge JavaScript into plain old ES5 version of JavaScript that can run in any browser (even the old ones). It makes available all the options that were added to JavaScript with the new ES6 specification, including classes, fat arrows and multiline strings.

Basically, Babel will convert +ES6 code into ES5 without waiting for browser support. We use babel to use the newest react and JavaScript syntax with old technologies and in old browsers.

WEBPACK

Webpack is an open-source JavaScript module bundler. It was released on 10th of march in 2012. It was primarily made for JavaScript but can also transform HTML, CSS and images etc. It is nowadays impossible for us to create web without module bundlers like webpack. Webpack is a JavaScript build tool and it basically groups the code of the program and other assets together. We use webpack in front end also because there are a number of assets in front-end like CSS, SASS, images, fonts. So, to bundle all these components we use module bundlers like webpack. As an example, Webpack can collect all user inline CSS styles in user JavaScript files and bundle them into one.

REACT-ROUTER-DOM

A router allows the application to navigate between different components, changing the browser URL, modifying the browser history, and keeping the UI state in sync. React is a popular library for building SPAs. However, as React focuses only on building user interfaces, it doesn't have a built-in solution for routing. React Router is the most popular routing library for React. In other words, React-router-dom is the router component for websites and webapp. For installing react-router-dom the system needs to have Nodejs version-6 or above installed.

2.4 BACK-END DEVELOPMENT

Backend website development consists of those tasks that allow user to optimize the website. The database, content, plugins, and other elements that make the website operate are all part of its backend. The backend ensures that the website is fully functional and applies to business websites, blogs, ecommerce sites, and others. The needs of the backend development will depend on the nature of the website and the functionality it offers to its users. This is distinct from the website's frontend, which consists of the coding languages used to create the site's pages, navigation, and other parts that result in the visual display with which users interact when engaging content. The backend and frontend of the website play different roles, but they work together to create the sum total experience of the website. Backend development utilizes the applications that access data and deliver it to the frontend for the audiences. The services and tools which were utilized to prepare the back end of the web application have been briefly mentioned below.

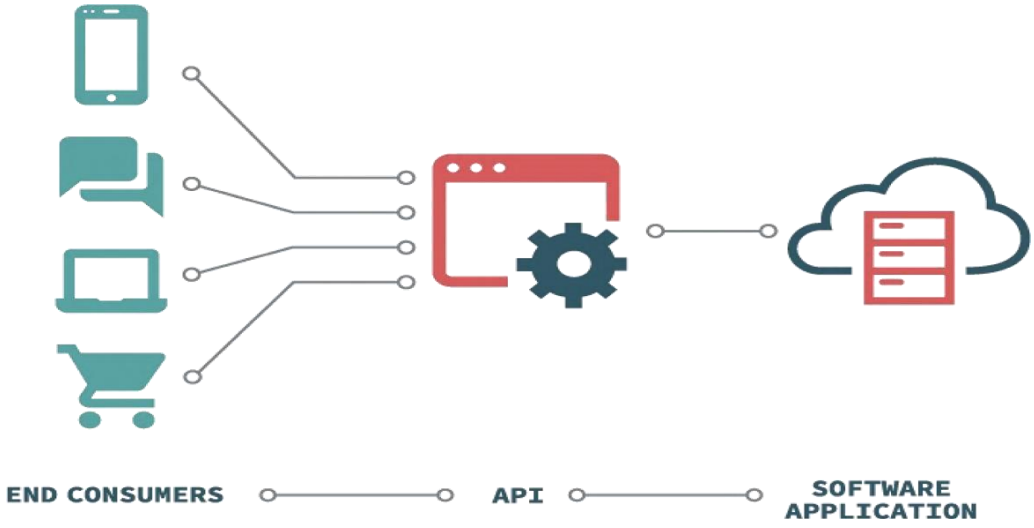

Fig. 2.2 API Working Diagram

NODE.JS

Node.js is an open-source, cross-platform, back-end, JavaScript runtime environment that executes JavaScript code outside a web browser. It was initially released on May 27th, 2009 and is written in c, c++, JavaScript. The main developers or authors are Ryan Dahl and Bryan Cantrill. Node.js is a lot faster, lightweight and less complex from its competitors like Django and php. Node.js excels from python in back-end in web services and real time applications. Node.js is used by users mainly in server-side scripting to produce dynamic web page content before the page is sent to the user's browser window. Node.js represents a "JavaScript everywhere" architecture, unifying web-application development around a single programming language, rather than using different languages for server- side and client-side scripting. Corporate users of Node.js software includes companies like Netflix, PayPal, LinkedIn, Walmart, UBER, Twitter, Yahoo, IBM, Rakuten and Amazon web services.

STRAPI

Strapi is an open source, self-hosted, 100% JavaScript, customizable CMS. Strapi was launched as a beta version past year. It consumes API from clients (React, Angular, Vue), mobile apps and now even from IoT devices. Using Strapi, we can easily customize the admin panel and the API as well. It gives us the option to use database of our choice and is currently supporting MongoDB, MySQL, MariaDB, SQLite, PostgreSQL. We have used SQLite in the project. Advantage of using Strapi is that it will fit in any web project which requires API and we can deploy the API in meagre 10 minutes. Thousands of companies and enterprises such as Société-Générale, IBM, Discovery Channel, NASA, Walmart, Delivery Hero and ASOS are already using Strapi to build and manage their blogs, editorial, corporate websites and mobile applications and other purposes.

SQLite

SQLite is a relational database system contained in a C library. SQLite was developed by D. Richard Hipp and was initially released on 17th August 2000. In contrast to many other database systems, SQLite is not a client-server database engine but rather it is embedded into end program i.e., in our case Strapi. SQLite is much faster than MySQL database. SQLite is the default database used in Strapi, while we have four other choices as well.

BABEL

Babel is used to make sure that everyone will be able to run the website. Babel was stably released on 27th February' 2020. Babel is a JavaScript compiler that converts edge JavaScript into plain old ES5 version of JavaScript that can run in any browser (even the old ones). It makes available all the options that were added to JavaScript with the new ES6 specification, including classes, fat arrows and multiline strings. Basically, Babel will convert +ES6 code into ES5 without waiting for browser support. We use Babel along with Node.js and react also. It reduces the code and makes it easy to understand and browser compatible.

WEBPACK

Webpack is an open-source JavaScript module bundler. It was released on 10th of march in 2012. It was primarily made for JavaScript but can also transform HTML, CSS and images etc. It is nowadays impossible for us to create web without module bundlers like webpack. Webpack is a JavaScript build tool and it basically groups the code of the program and other assets together. But the problem is on server when user is using the react modules since node doesn't support these (ES6) Modules. User can use babel in node server side but it transpile code in runtime which is not effective. The most common way to solve this problem is pack backend by webpack. Webpack can also collect all the inline CSS styles in the JavaScript files and bundle them into one.

STRIPE

Stripe is an American financial services and software as a service company headquartered in San Francisco, California, United States. The company primarily offers payment processing software and application programming interfaces for e- commerce websites and mobile applications. In the project we are using Stripe as a payment processor by using react-stripe-elements. Stripe can be used a payment processor for e commerce's and businesses of every scale from individual start-up's to companies like Amazon, Zoom etc. these all-use stripe in their payment processor.

We can use Elements with any Stripe product to collect online payments. The Stripe.js / Stripe Elements API reference goes into more detail on the various customization options for Elements (e.g., styles, fonts). The element is a thin React wrapper around Stripe.js and Stripe Elements. It allows user to add Elements to any React app, and manages the payment elements for user.

2.5 DEPLOYMENT

Software deployment includes all of the steps, processes, and activities that are required to make a software system or update available to its intended users. The general deployment process consists of several interrelated activities with possible transitions between them. These activities can occur at the server side or at the client side or both. Today, most IT organizations and software developers deploy software updates, patches and new applications with certain technologies mentioned below. Software deployment is one of the most important aspects of the software development process. Deployment is the mechanism through which applications, modules, updates, and patches are delivered from developers to users. The methods used by developers to build, test and deploy new code will impact how fast a product can respond to changes in customer preferences or requirements and the quality of each change.

HEROKU

Heroku is a cloud platform as a service supporting several programming languages. Heroku was founded by James Linden Baum, Adam Wiggins, Orion Henry in 2007. One of the first cloud platforms, it initially supported only the Ruby programming language, but now supports Java, Node.js, Scala, Python, PHP, and Go. Developers use Heroku to deploy, manage, and scale modern apps. The platform is elegant, flexible, and easy touse, offering developers the simplest path to getting their apps to market.

NETLIFY

Netlify is a San Francisco-based cloud computing company that offers hosting and serverless backend services for web applications and static websites. Netlify was founded by Mathias Biilmann along with Christian Bach on 7th February, 2014. Netlify is a platform user can use to automatically build, deploy, serve, and manage web apps. It also provides a variety of other features like form processing, serverless functions, and split testing. We can deploy modern static websites with Netlify and get CDN and continuous deployment.

GITHUB

GitHub provides hosting for software development and version control using Git. The best and most used features of GitHub are project management, collaborative coding and public repositories. It offers distributed control and source code management functionality of Git, plus other features. We can easily assign work to our team members on a particular issue or problem. GitHub also helps us to keep track of the individual work and progress.

2.6 Summary

This chapter presents some discussions about the relevant tools and technologies used to develop an e-commerce web application.
JavaScript helps user in developing great front-end as well as back-end software's using different JavaScript based frameworks like jQuery, Node.JS, React etc. that allows user to make web pages interactive. Whereas HTML and CSS are languages that give structure and style to web pages. React is an open-source JavaScript library used in web development for building user interfaces and interactive elements. React makes it easier to create interactive UIs. React helps in making the code more predictable and easier to debug and is used along with JSX. While for back-end Node.js was used for executing JavaScript code outside a web browser and for browser independency Babel was used so that everyone can use the web application. For API purposes, Strapi was used which also additionally provides the security to the application while Stripe was used for payment purposes. GitHub, Heroku and Netlify were used for collaborative coding, project management and hosting, managing and deployment of the web application.

Chapter 3
Implementation

3.1 INTRODUCTION

This chapter will describe the key implementation processes, the UML diagrams and some code snippets of this e-commerce web Application. Implementation in software development is the process of realizing an application's requirements and design. It mainly involves mapping the design into coding in order to achieve the specifications stated for the application. The following Sections define the different modules of the implementation.

3.2 UML ANALYSIS MODEL

Modeling involves the designing of software systems before coding takes place. Modeling plays an important role in any software development project. It guarantees the completeness and correctness of a software system and the fulfillment of end-user's expectations. In addition, modeling serves as the only reference point to cross-check requirements before coding.

A Unified Modeling Language (UML) based tool was used to model this application. UML diagrams give both static and dynamic views of an application and it is well suited for object-oriented languages like Java and C#. The following sub-sections present the UML diagrams used to model this application

3.2.1 USE CASE DIAGRAMS

The use case diagrams for this application illustrate the interactions that exist between users (actors) and use cases (actions) within the application. There are two actors identified for this application – administrator (admin) and customer actors. As a result, there are two use case diagrams for the software application – admin use casediagram and customer use case diagram. The admin is the owner of the e-commerce store who performs various administrative tasks such as add products, view orders, and update order status while the customer is any individual who buys a product or products from the online store.

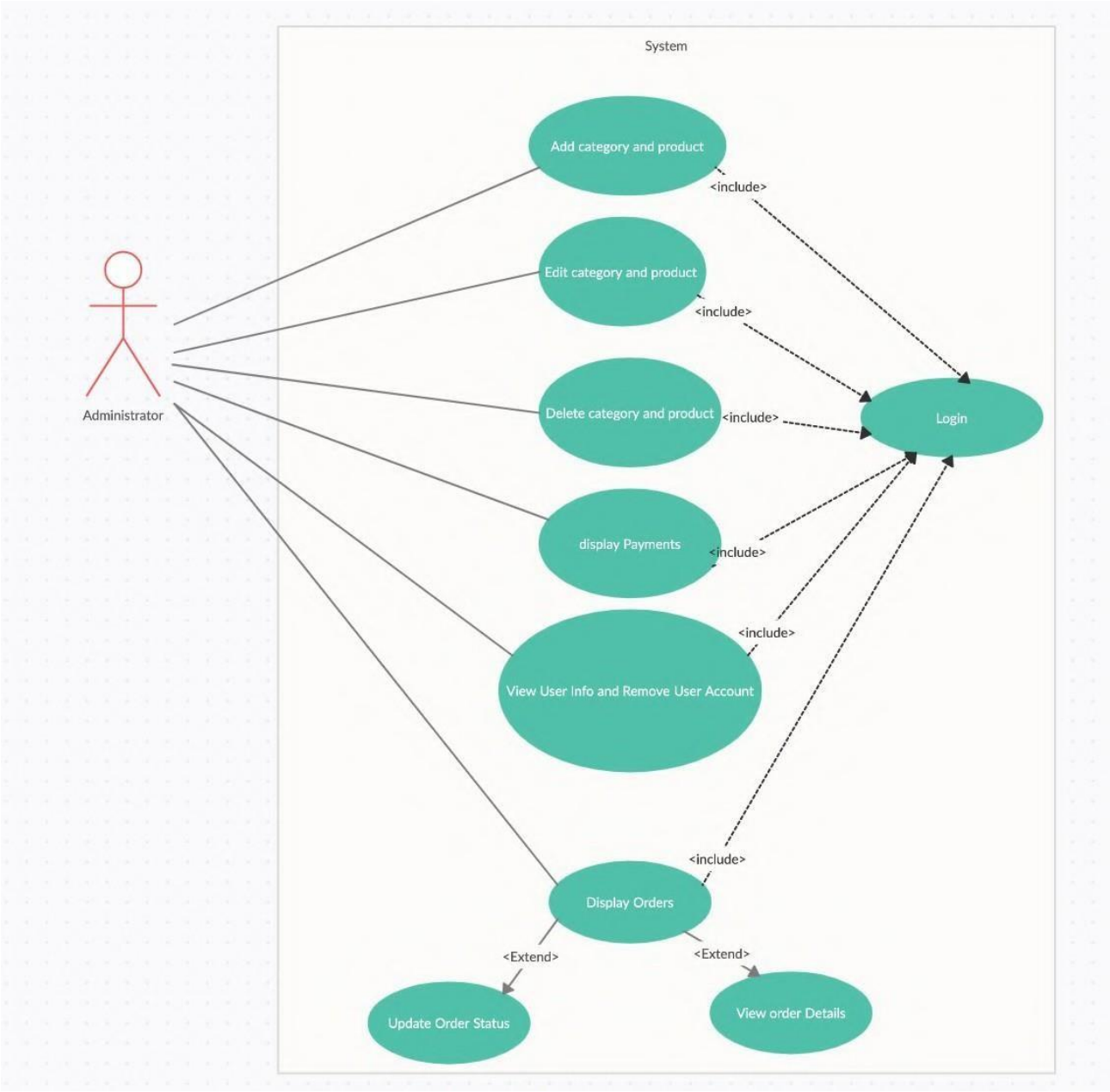

Fig 3.1 -Admin Use Case Diagram

Figure 3.1 shows the admin use case diagram. The diagram depicts how the admin communicates with the application. More so, it shows all the actions that the admin can perform on the application. As can be seen in the diagram, before any of these actions could be executed the admin will have to login in order to be authenticated.

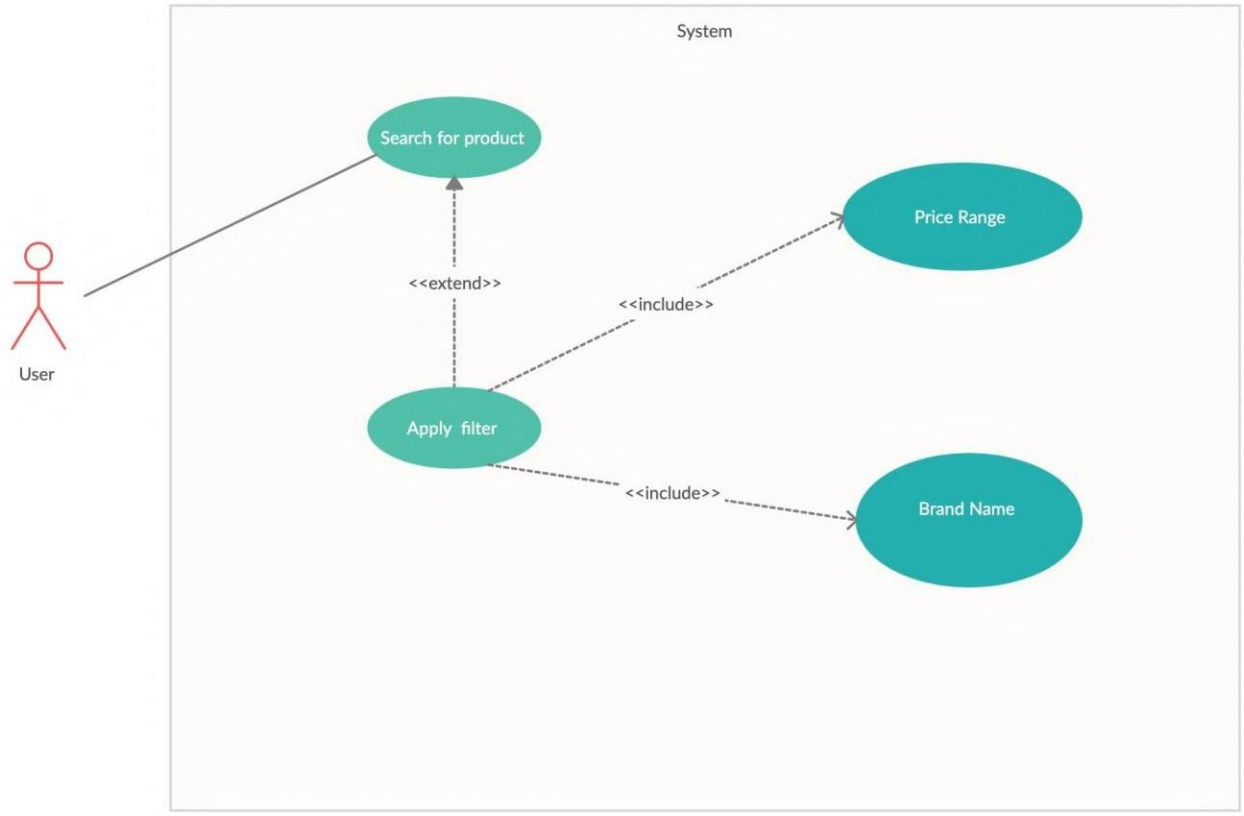

Fig 3.2 User Use Case Diagram

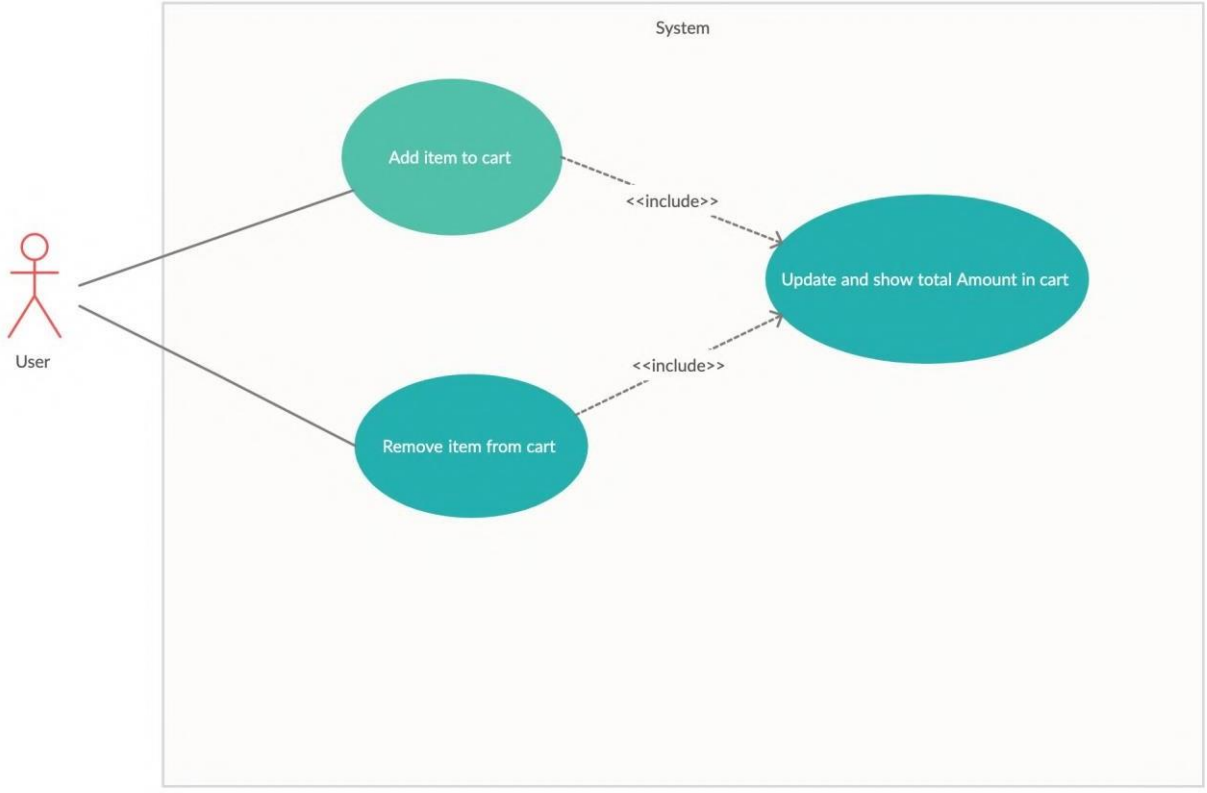

Fig 3.3 User Case Diagram -2

Fig 3.4 User Use Case Diagram - 3

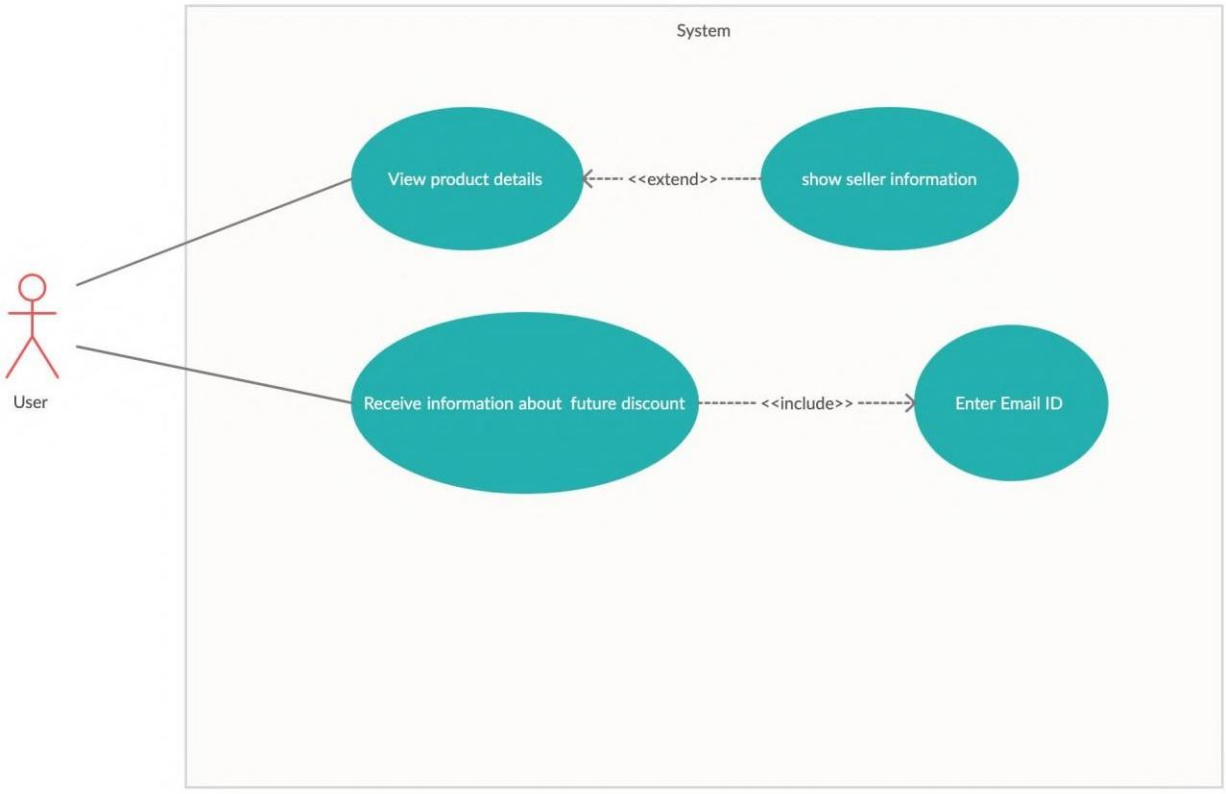

Fig 3.5 User Use Case Diagram - 4

Fig 3.2 shows the Filtering and search functionality for the user. This allows the user to limit Search Results for quicker identification and buying process

Fig 3.3 shows the cart Functionality of the user in which the user can add/remove items from the cart for buying them later. The user can also see the total Payable amount for purchasing all items in the cart

Fig3.4 shows the data storage functionality of user who has created an account on the web-application. The items in the user's cart and user credentials remain stored in the user's cloud account after the web-application is closed

Fig3.5 shows the functionality for the option to see the details of the product and information about the seller. The user is also notified in the apps if there is change in availability of the product or any discounts/change in price

3.2.2 CLASS DIAGRAMS

A class diagram depicts the classes in a software system and how they interact with each other. Also, the class attributes and functions are illustrated in a class diagram.

Figure 3.6 shows the class diagram for this application. It shows the relationships between classes in the application and constraints applied to these relationships.

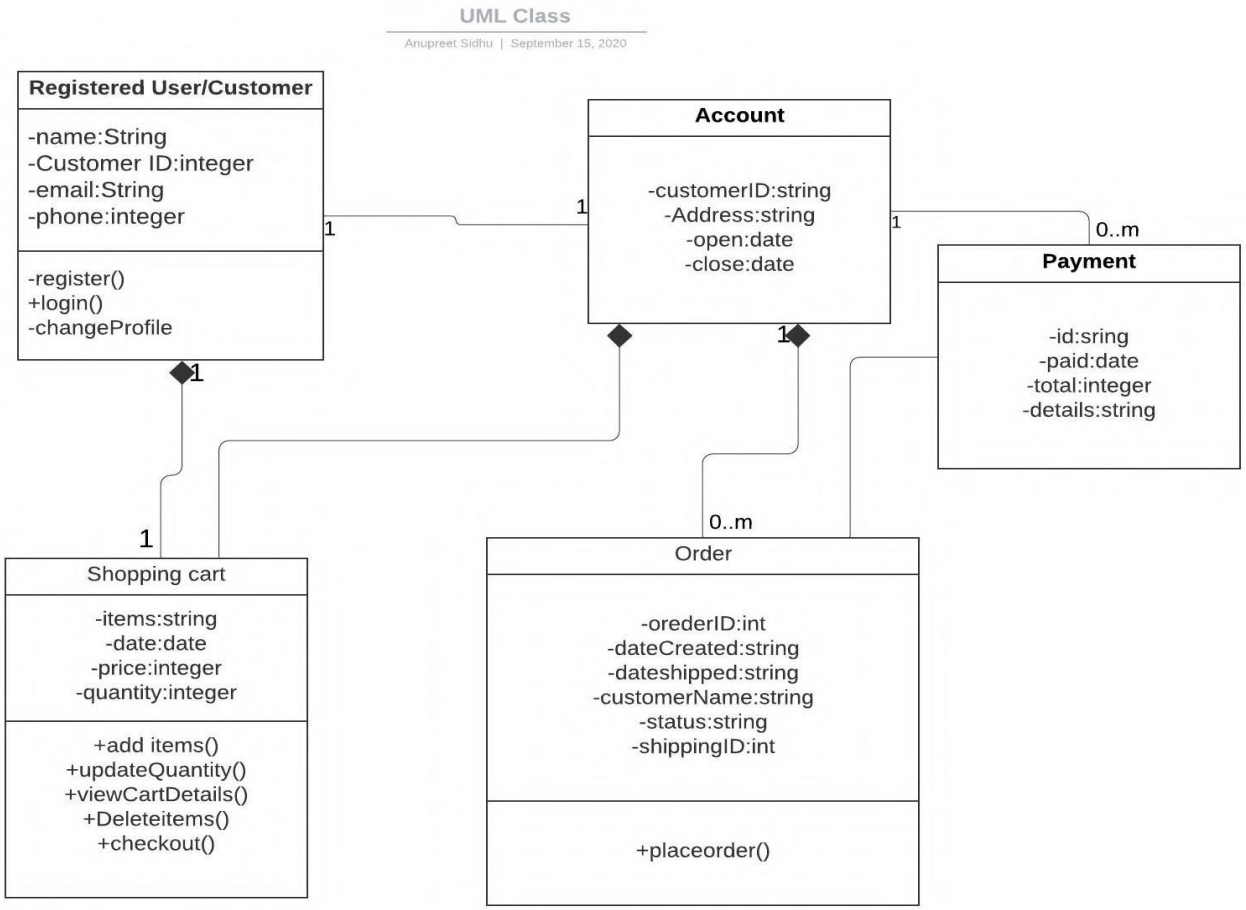

Fig 3.6 Class Diagram

3.2.3 SEQUENCE DIAGRAMS

A sequence diagram gives a detailed visual description of how the various classes in a system interact with each other. Also, it depicts the order in which different objects exchange messages with one another in a system. The sequence diagrams for this application are presented in the following sub-sections.

Admin login Sequence Diagram

Figure 3.7 gives a detailed sequence of events needed for the admin to login to the admin page of the 1application. After providing the login credentials on the admin login page,

the admin is authenticated through a form-based authentication method provided by the Apache Tomcat web server. After a successful authentication process, the admin is forwarded to the admin menu page, which contains links to several administrative functions. However, if the authentication process fails, the admin is redirected to the admin login page with displayed error message. For the remaining sequence diagrams required for the administrative tasks, see Appendices.

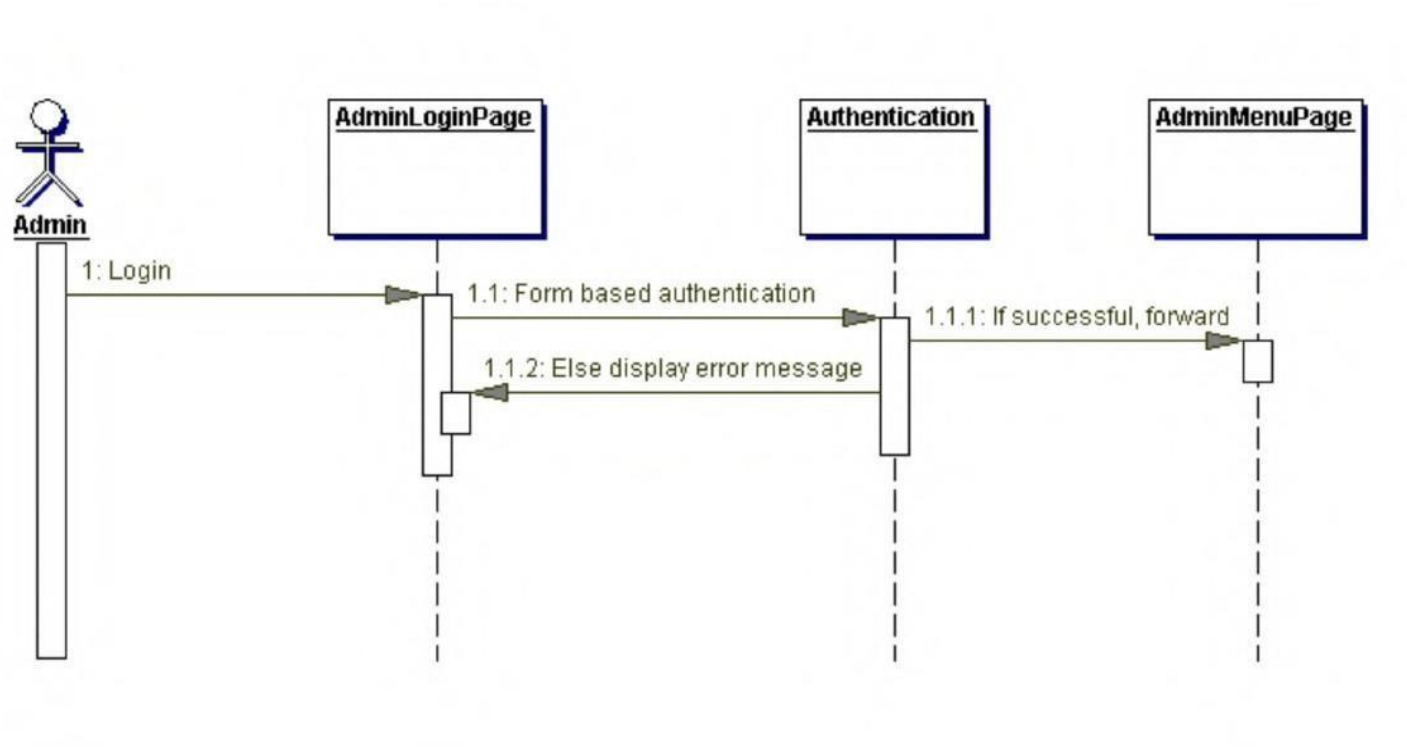

Fig 3.7 Admin Login sequence Diagram

3.3 CODE SNIPPETS

The code for the e-commerce web-application was written in JavaScript using Visual Studio code. The snippets are taken as Screen Shots of the code open in Visual Studio Code to explain the main concepts of implementation.

3.3.1 Home Page

Home Page of the W/A gives the user option for browsing the entire collection of products or see the featured/most popular products on the home page directly.

Fig 3.8 shows the base code of home page Fig 3.9 shows the component for hero section
Fig 3.10 shows the component for featured section for displaying featured products

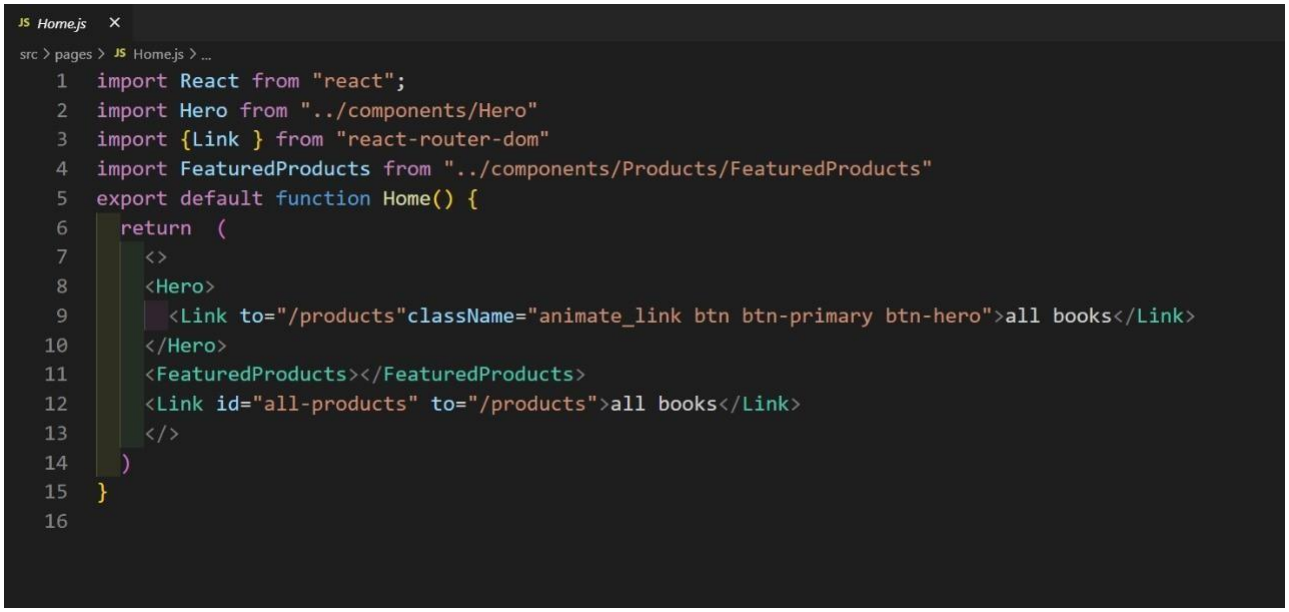

Fig 3.8 Code Snippet for home Page

Hero.js

src > components > Hero.js > ...

```
1  import React from "react";
2
3  export default function Hero({children}) {
4
5    return  (
6    <div className="hero">
7      <div className="banner">
8        <h2>looking for a special gift for <br/><h3> the book lover</h3></h2>
9        <p>
10         here  is your destination
11       </p>
12       {
13         children
14       }
15     </div>
16     <div className="picture-container">
17
18     </div>
19   </div>
20   )
21 }
22
```

Fig 3.9 Code Snippet for Hero Section of home Page

FeaturedProducts.js

src > components > Products > FeaturedProducts.js > ...

```
1  import React from "react";
2  import ProductList from "./ProductList"
3  import {ProductContext} from "../../context/products"
4  import Loading from "../Loading"
5  export default function FeaturedProducts() {const{loading,featured}=React.useContext(ProductContext)
6
7  if(loading){
8    return <Loading/>
9  }
10   return <ProductList nullValue={"null"} title="featured products" products={featured}/>
11 }
12
```

Fig 3.10 Code Snippet for Featured Section of home Page

3.3.2 Add product to Cart

In Fig 3.11 the code snippet is for the function when the user clicks add to cart in any item then that item will be saved in the user's cart.

```
const addToCart=(product)=>{

    const{
    id,
    name,
    price,
    image,

  }=product

  const newProduct={id,name,price,image:image[0],amount:1}
   const item=[...cart ].find(item=>item.id===id)
   if(item){
       increaseAmount(id)
       return
   }
   else{
       const newItems=[...cart,newProduct]
       setCart(newItems)}


}
```

Fig 3.11 Code Snippet for Add Item to Cart

3.3.3 Remove product from Cart

Fig 3.12 shows the code for the instance when user presses the remove button on any item in the cart and the item is removed from the cart

```
export const CartProvider = ({children}) => {const [cart,setCart]=useState(getCartFromLocalStorage())
const [total,setTotal]=useState(0)
const [cartItems,setCartItems]=useState(0)

const removeItem=(id)=>{
    setCart([...cart].filter(item=>item.id!==id))
}
```

Fig 3.12 Code Snippet for Removing item from the cart

3.3.4 Changing Item amount in the Cart

Fig 3.13 shows the code for increasing or decreasing quantity of item from the cart by pressing the button below any item in the cart.

```
const increaseAmount=(id)=>{
    const item=[...cart].map(item=>{
        return item.id===id?{...item,amount:item.amount+1}:{...item}
    })
    setCart(item)
}
const decreaseAmount=id=>{
    const item=[...cart].map(item=>{
        return item.id===id?{...item,amount:item.amount-1}:{...item}
    }).filter(item=>item.amount!==0)
    setCart(item)
}
```

Fig 3.13 Code Snippet for Changing item amount in cart

3.3.5 Clearing the Cart

Fig 3.14 shows the code for removing all items from the cart of the user by pressing the "Clear-Cart" button in the cart.

```
const clearCart=()=>{
    setCart([])
}
React.useEffect(()=>{
    // local storage
    localStorage.setItem("cart",JSON.stringify(cart))
    let newCartItems=cart.reduce((total,cartItem)=>{
        total+=cartItem.amount
        return total
    },0)
    setCartItems(newCartItems)
    let newTotal=cart.reduce((total,cartItem)=>{
        return total+=((cartItem.amount)*(cartItem.price))
    },0)
    newTotal=parseFloat(newTotal.toFixed(2))
    setTotal(newTotal)
},[cart])
```

Fig 3.14 Code Snippet for Clearing the Cart

3.3.6 Searching and filtering Products

Fig3.15 Shows the function to type product name and do a general search of a Product. Fig 3.16 Shows the function to apply filter on the search by selecting various criteria such as genre, author, price range, Copies sold, or language

```
products.js
src > context > products.js > ...
53          })
54      }
55          if(shipping!==false){
56              newProducts=newProducts.filter(({free_shipping})=>free_shipping===true)
57          }
58
59          if(search!==""){
60              newProducts=newProducts.filter(({name})=>name.toLocaleLowerCase().trim().includes(search.
                toLocaleLowerCase().trim()))
61          }
62          if(price!=="all"){newProducts=newProducts.filter((item)=>{
63                  if(price==0){
64                      return item.price<300
65                  }
66                  if(price===300){
67                      return item.price<650 && item.price >300
68                  }
69                  return item.price>650
70              })
71
72          }
```

Fig 3.15 Code Snippet for Searching Product using Name

```
products.js
src > context > products.js > ...
33      useEffect(()=>{
34          // sort wrt to price
35      let newProducts=[...products].sort((a,b)=>{
36          return a.price-b.price})
37
38          const {search,category,shipping,price,author,sale}=filters
39          if(author!=="all"){
40              newProducts=newProducts.filter(item=>{
41                  return item.author.toLocaleLowerCase()===author.toLowerCase()
42
43          })
44          }
45          if(category!=="all"){
46
47              newProducts=newProducts.filter(item=>{
48                  const categories=item.category.map(cat=>{
49                      return cat.toLocaleLowerCase()
50                  })
51                  return categories.includes(category.toLowerCase())
52
53          })
54      }
55          if(shipping!==false){
56              newProducts=newProducts.filter(({free_shipping})=>free_shipping===true)
57          }
```

Fig 3.16 Code Snippet for Filtering Search

3.3.7 User Sign Up and Sign in

Fig 3.17 shows the code for first time user Sign Up, Fig3.18 shows the code for registered user to sign in and Fig 3.19 shows the functionality for registered and signed in user to Log out.

```
// register user
import axios from "axios"
import url from "../utils/URL"

export default async function registerUser({email,password,username}){

const response=await axios.post(`${url}/auth/local/register`,{
    username,email,password
}).catch(error=>console.log(error))
return response
}
```

Fig 3.17 Code Snippet for User Sign Up

loginUser.js

```
src > strapi > loginUser.js > ...
//login user
import url from "../utils/URL"
import axios from "axios"
export default async function loginUser({password,email}){
    console.log(email,password);
const response =await axios.post(`${url}/auth/local`,{
    identifier:email,
    password
}).catch(error=>{
    console.log(error);
})
return response
}
```

Fig 3.18 Code Snippet for Registered User Sign In

```
// user context
import React,{useState,useContext,createContext} from 'react'
const UserContext=createContext()
function getUserFromLocalStorage(){
    return localStorage.getItem("user")?JSON.parse(localStorage.getItem("user")):{username:null,token:null}
}
const UserProvider = ({children}) => {
    const [user,setUser]=useState(getUserFromLocalStorage())
    const [height,setHeight]=useState(0)
    React.useEffect(()=>{
        window.addEventListener("scroll",()=>{
            setHeight(window.pageYOffset)
        })
        return ()=>{
            return window.removeEventListener("scroll",()=>{})
        }
    })
    const userLogin=user=>{
        setUser(user)
        localStorage.setItem("user",JSON.stringify(user))
    }
    const userLogout=()=>{
        setUser({username:null,token:null})
        localStorage.removeItem("user")
    }
    const [alert,setAlert]=useState({
```

Fig 3.19 Code Snippet for Logging Out of account

3.3.8 Payment Request and Storing Order Data

Fig 3.20 shows the code for filling payment information and sending payment request through stripe. Fig3.21 shows the code for storing Data of confirmed order int he database.

```
const CardForm=injectStripe(Checkout);
const StripeWrapper=()=>{
  return <StripeProvider
    apiKey="pk_test_51HlbLoK1ED9U7Hukekyy8LgHl4b6b9Gq4o9DcWAX5GsN6sqsLIOoyban9WUbZtd8e0BJkj6N24bJhqsk3SosCmKa00tPP4DmqY"
    >
    <Elements>
      <CardForm></CardForm>
    </Elements>
  </StripeProvider>
}
export default StripeWrapper
```

Fig 3.20 Code Snippet for sending Payment request

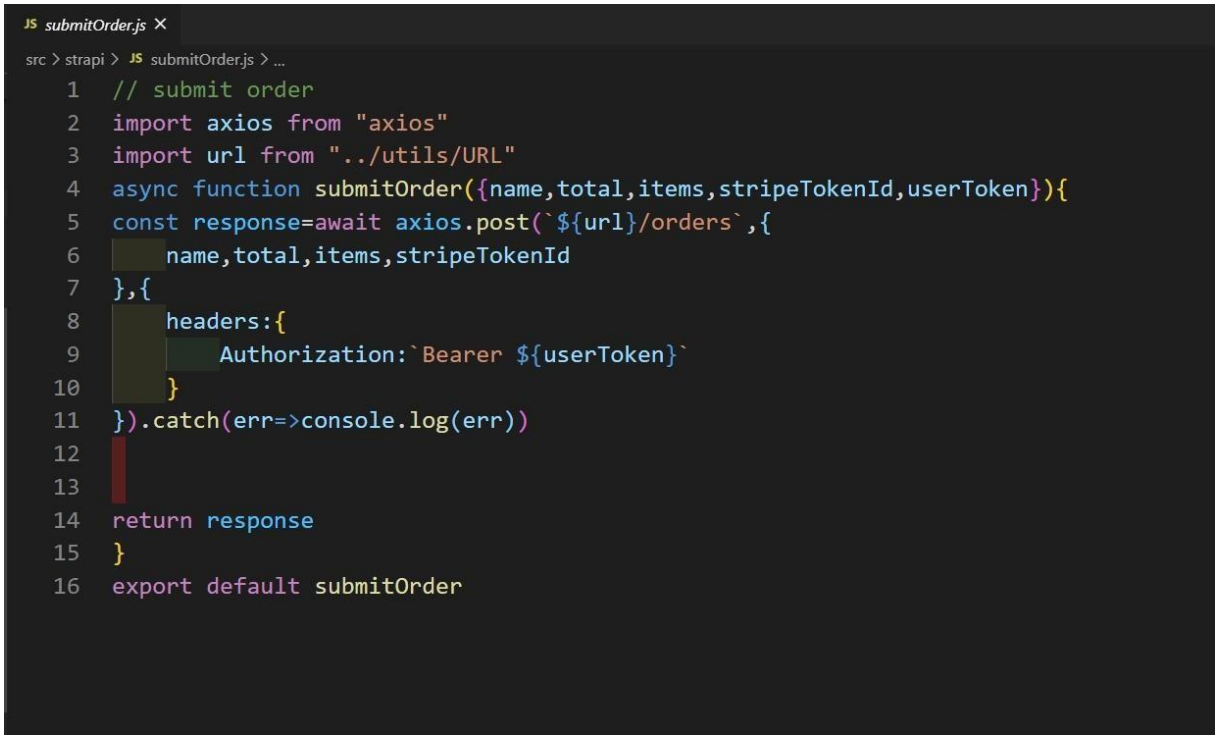

Fig 3.21 Code Snippet for storing data for confirmed order data

3.4 Summary

This chapter was about showing and Explaining the Use Case diagrams for the administrator as well as the general user, the Class Diagrams to show the interaction between different parts of system, also the admin login sequence diagram. Code Snippets have been provided for further showing the implementation of various functionalities.

CHAPTER 4
RESULT AND CONCLUSION

4.1 TESTING

Software testing is carried out on a software application mainly to detect software bugs or missing requirements of the application. It involves the process of investigating and evaluating a software product or application in order to make sure that its business and technical requirements are fulfilled. For the purpose of testing, this application was deployed and run-on Apache Tomcat servlet container and accessed on a web browser. For simplicity, the testing template in Table 4.1 was used for the testing.

Table 4.1 Software testing template and results.

S/N	Test Description	Steps	Expected System Response	Status
1.	Access home page of the application to view available products.	Enter the home page URL of the application on a web browser.	The home page of the application is displayed with available products.	Pass.
2.	View available products by category.	Click any of the category links on the home page.	The products for the clicked category are displayed.	Pass.
3.	Add product to the shopping cart.	Click the add to cart button on the displayed product on the home page.	The application navigates on the cart page with the number of items in the cart updated and shown on the top of the home page	Pass.

4.	View the shopping cart.	Click the view cart link on the home page of the application.	The shopping cart of the application is displayed with its items.	Pass.
5.	Update an item quantity in the shopping cart.	Edit the quantity of an item in the cart and click the update button.	The item quantity, its price and the total price of all items in the cart are updated accordingly.	Pass.
6.	Remove an item from the shopping cart.	Click the remove button of the item to be removed.	The item is removed from the cart and the total price of all items in the cart is updated accordingly.	Pass.
7.	Clear all items from the shopping cart.	Click the clear cart button.	The cart is empty.	Pass.
8.	Continue shopping from the shopping cart page.	Click the continue shopping link.	The user is redirected to the products page of the application to continue shopping.	Pass.
9.	Check out with Stripe.	Click the Stripe check out button on the shopping cart page.	The application redirects to the login page of Stripe.	Pass.
10.	Confirm order after a successful payment authorization on Stripe.	Click the confirm order button on the order review page of the application.	The order confirmation page is displayed with details of the transaction. Also, an email order confirmation is sent to the email address of the buyer.	Pass.

11.	Access the admin page of the application to carry administrative tasks.	Enter the admin page URL of the application on a web browser.	The admin login page of the application is displayed.	Pass.
12.	Check for empty login data on the admin login form.	Click the login button with one empty field.	The application displays a prompt message telling the user to enter data in the empty field.	Pass.
13.	Check for incorrect login credentials on the admin login form.	Enter the incorrect login details. Click the login button.	The login form with an error message is displayed.	Pass.
14.	Check for correct login credentials on the admin login form.	Enter the correct login details. Click the login button.	The admin menu page is displayed.	Pass.
15.	Access the admin menu items on the admin menu page.	Click any of the menu links on the page.	The appropriate page is displayed.	Pass.
16.	Check for empty data inputs on forms accessible by the admin.	Enter the required data on the form with one or more empty inputs and submit the form.	The application displays a message prompt telling the user to enter data in the empty field(s).	Pass.
17.	Add/edit category/ product.	Enter the required data on the appropriate form and submit.	An appropriate page is displayed to show the added/edited category or product.	Pass.
18.	Delete category/product.	Click the appropriate delete button.	The application displays an appropriate page to confirm the deleted category or product.	Pass.

4.2 RESULT AND SCREENSHOTS

After successfully compiling the Final Code and Deploying the E-commerce W/A the final Results achieved are shown below.

4.2.1 Home Page

The Home page of the W/A acts as a passage to different Pages on the Site

Fig 4.1 shows the buttons to go to browse Products page, login page, User Cart, etc. Fig 4.2 shows the lower half of the home page which includes a list of featured products.

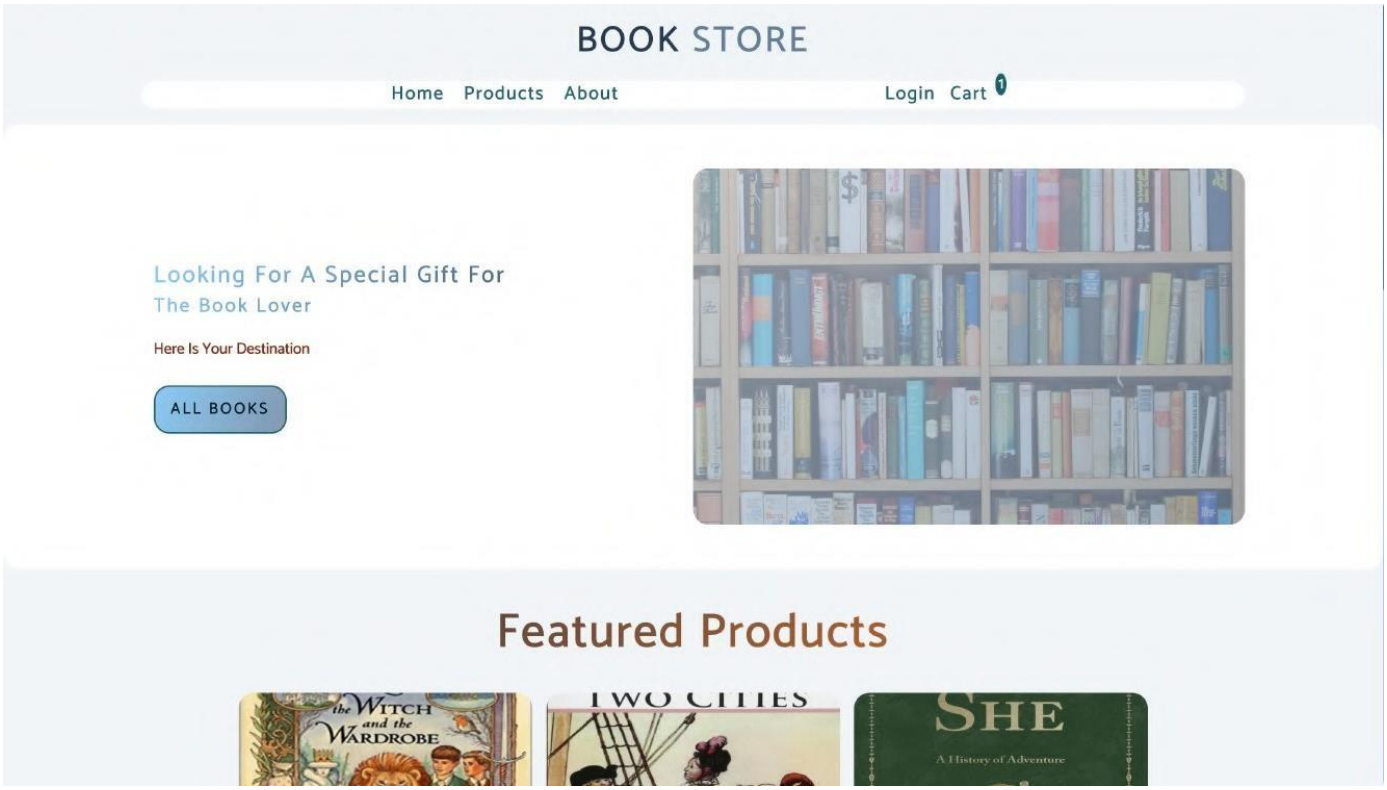

Fig 4.1 Upper half of Home Page

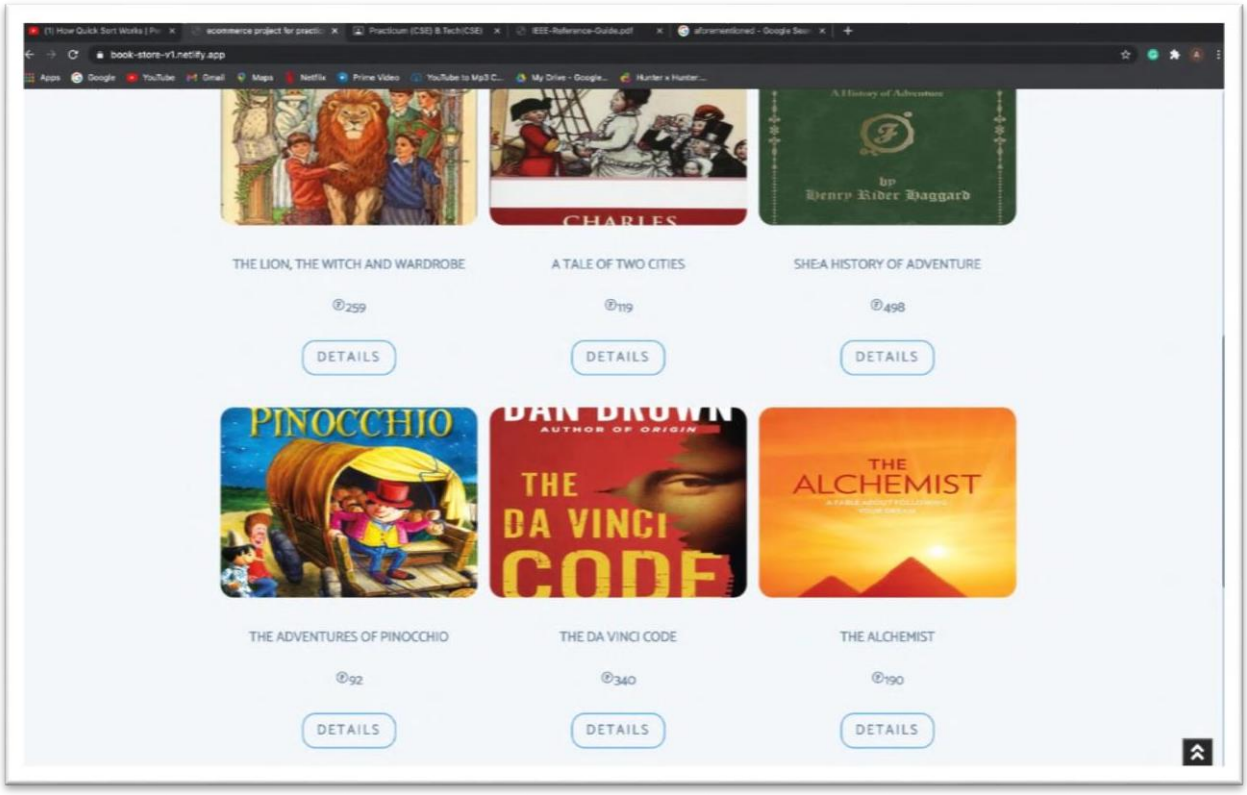

Fig 4.2 Lower Half of Home Page

4.2.2 Shopping Cart Page

As shown in Fig 4.3 the shopping cart shows the total items in the cart, each items price, total payable amount for checkout, options to change each items quantity etc.

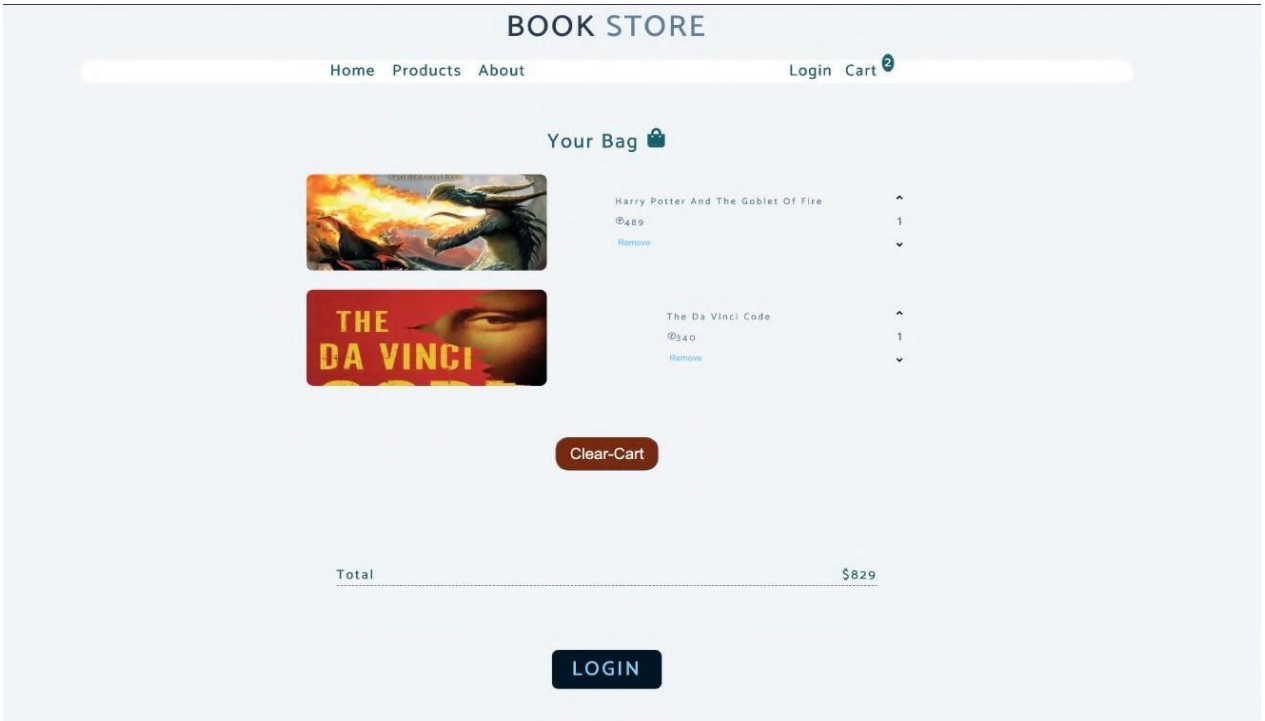

Fig 4.3 Shopping Cart Page

4.2.3 Sign Up and Sign-in Page

Fig 4.4 Shows the Sign-up page for non-registered User and Fig 4.5 shows the Sign-in page for Signed Up users.

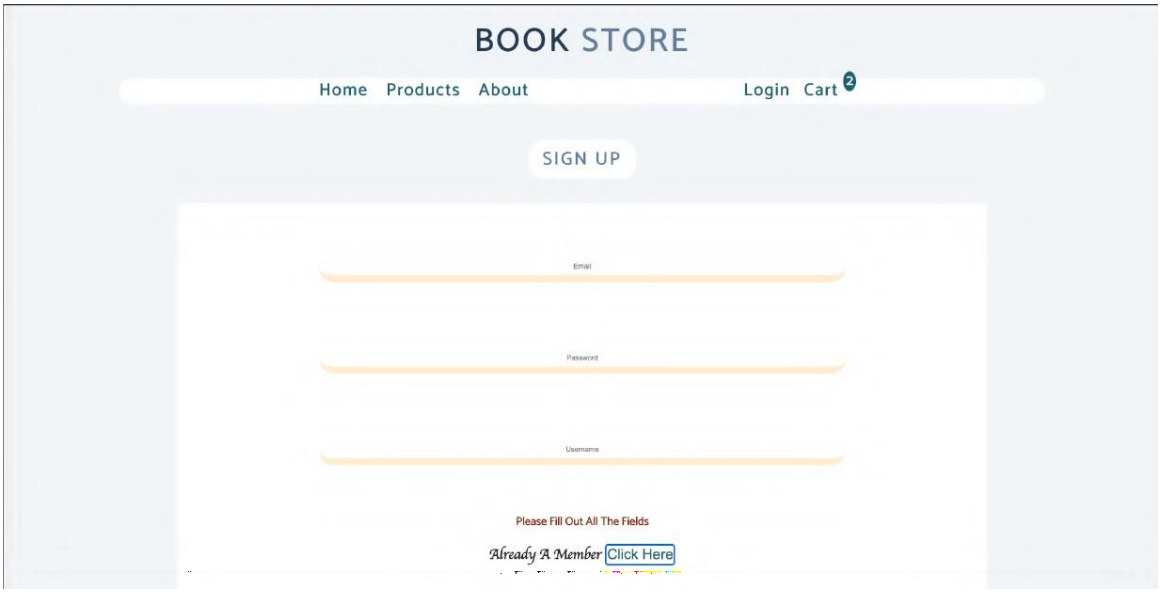

Fig 4.4 Sign up Page

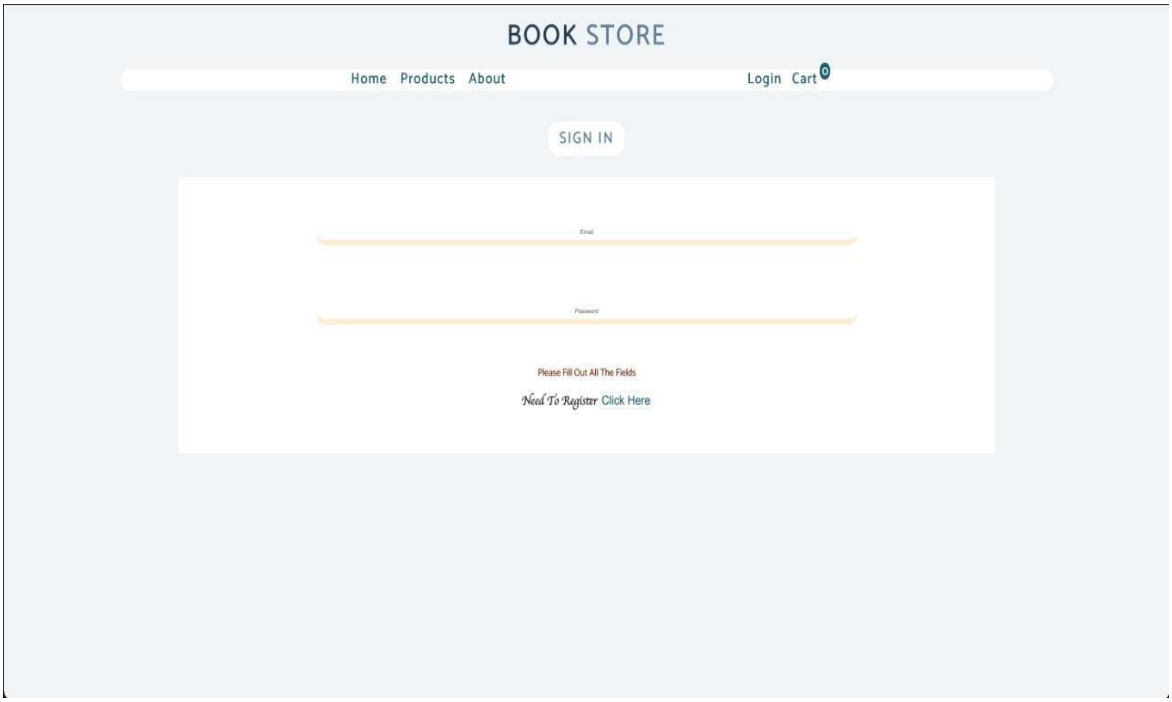

Fig 4.5 Sign-in Page

4.2.4 Filter and Search Products Page

Fig4.6 Shows Search by name option and also the drop-Down menus for filtering search based on Genre, Author, Price, Sales, etc.Fig.4.7 Shows the search Results for “Fantasy” Genre.

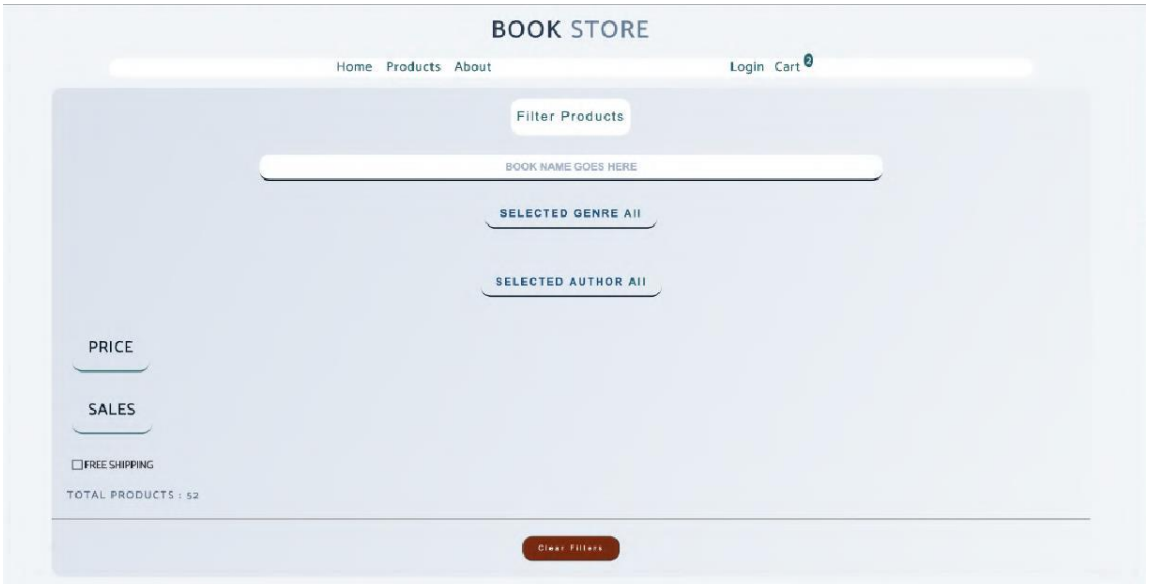

Fig 4.6 Search Option and Types of Filters

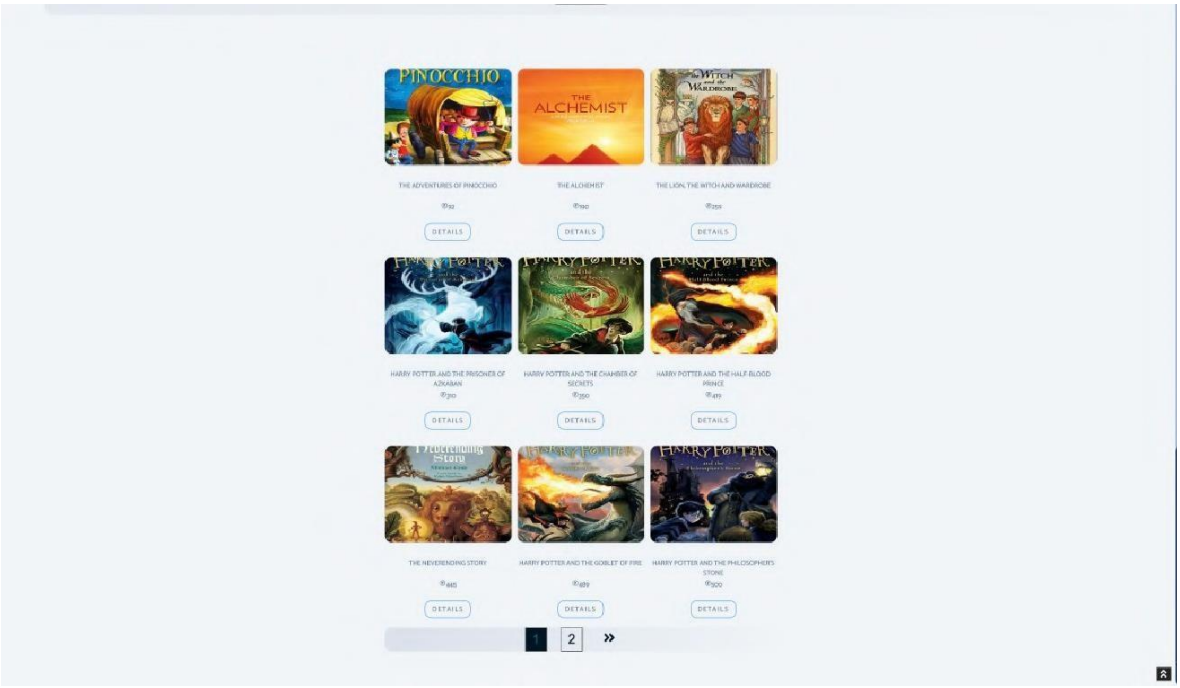

Fig 4.7 Fantasy genre Filter Result

4.2.5 Admin Log-in Page

Fig4.8 shows Admin login page where a user can enter and submit login credentials for authentication purpose

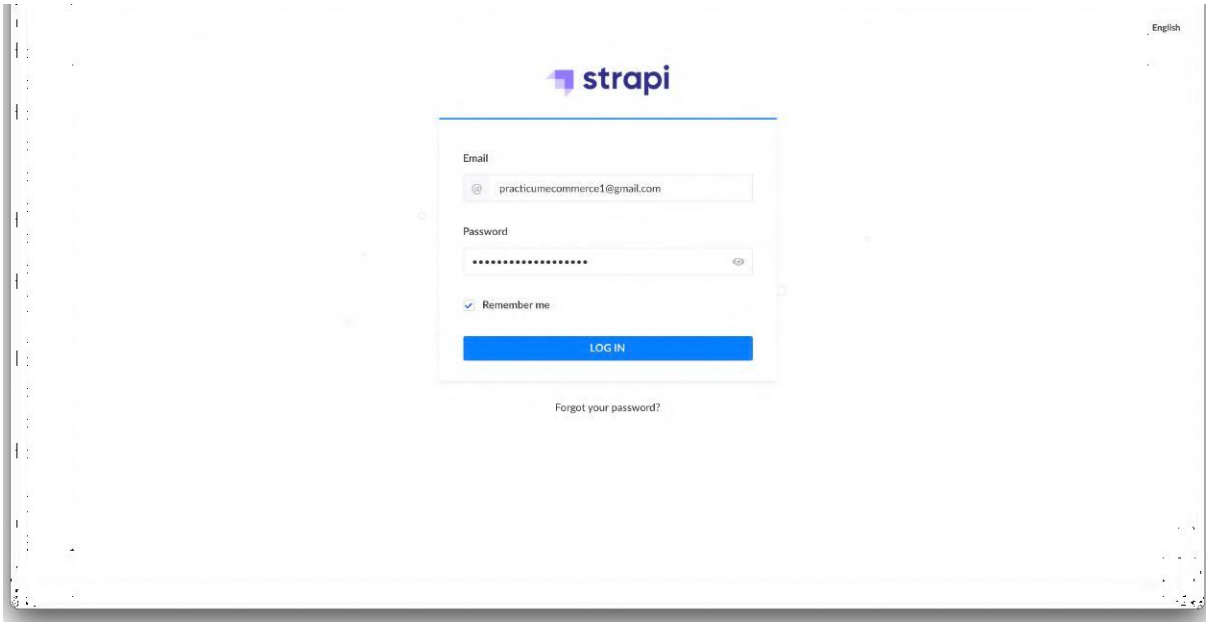

Fig 4.8 Admin Login Page

4.2.6 Admin Accessible Pages

Fig4.9 Shows the Admin Product page where Admin can Edit available products

Fig4.10 Shows the Admin Order Page where the Admin can view all orders placed and information like order date, payment amount, user address, etc.

Fig4.11 Shows the Admin User Page where admin can view all registered users and even block cheating users.

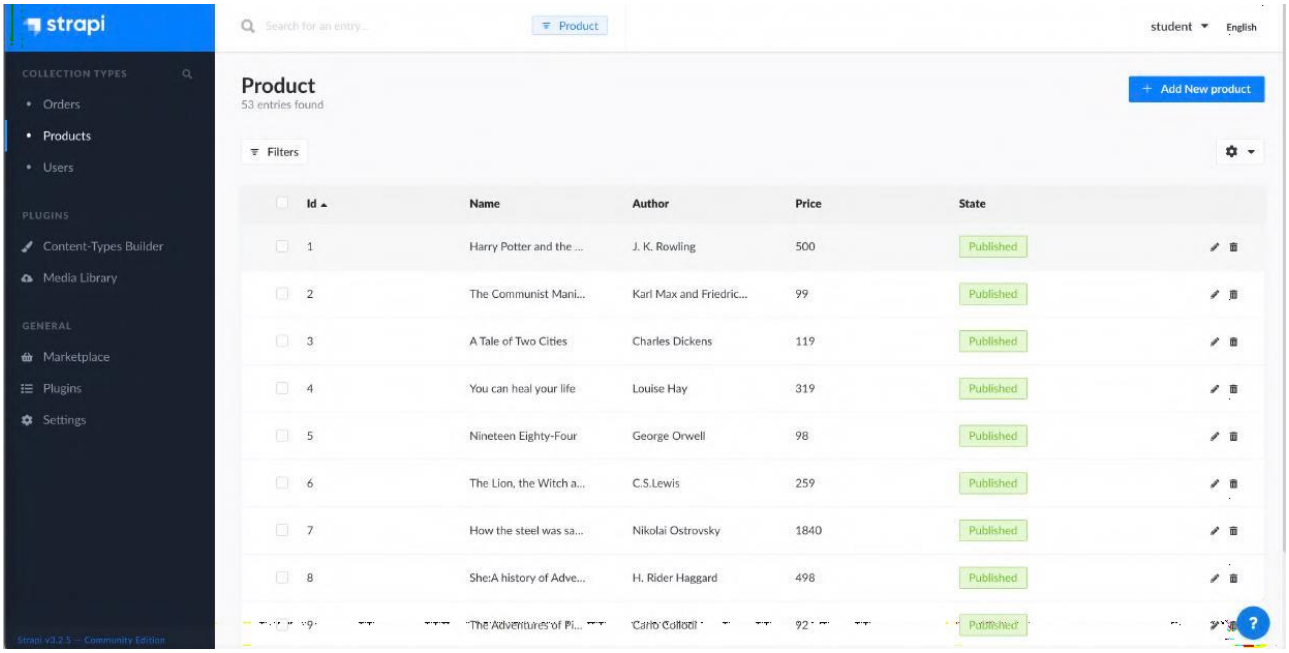

Fig 4.9 Admin Product Page

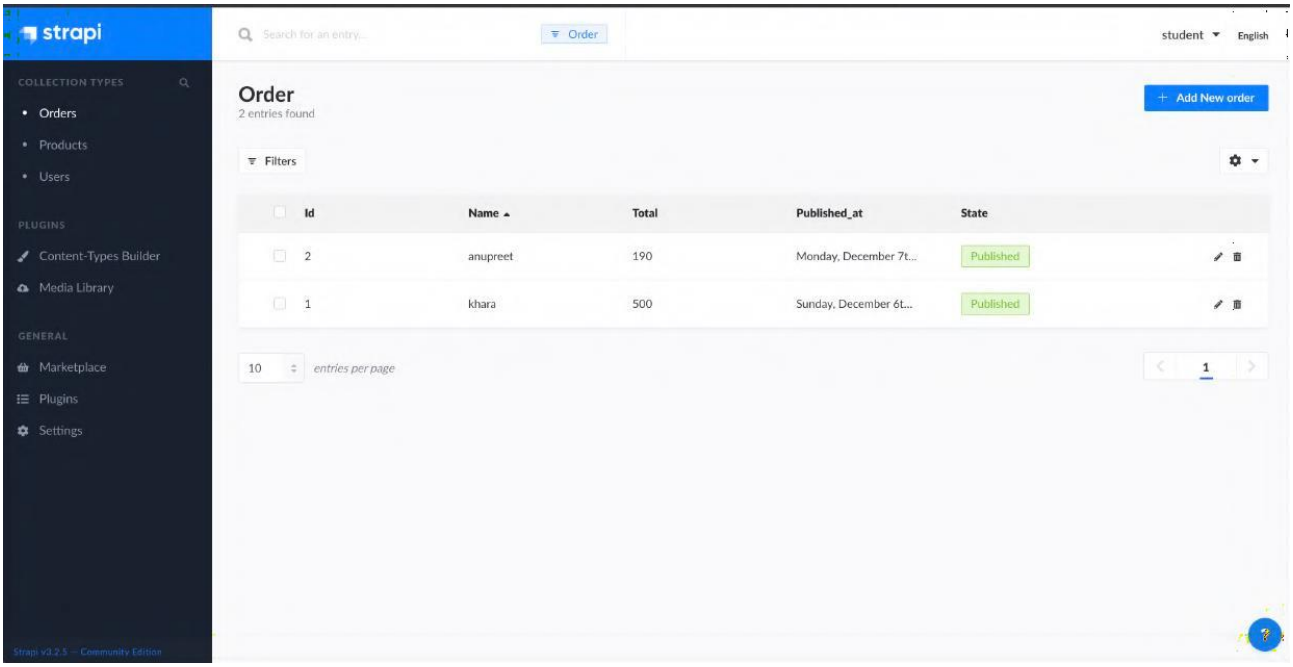

Fig 4.10 Admin Orders Page

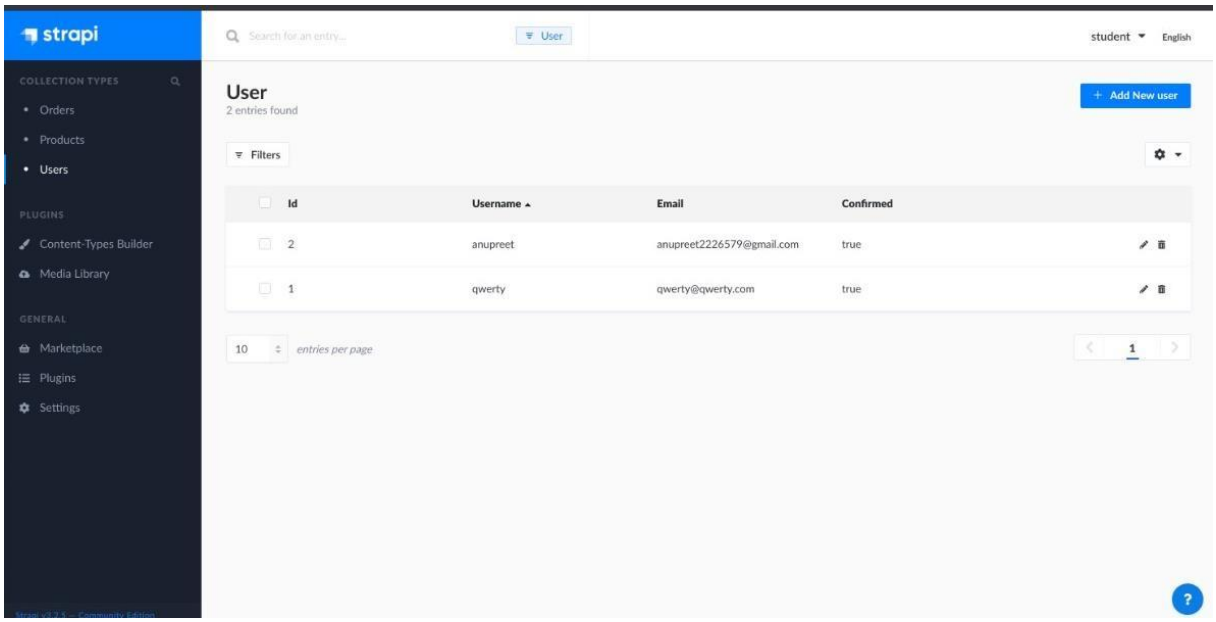

Fig 4.11 Admin Users Page

4.3 CONCLUSION AND FUTURE SCOPE

The main objective of this Practicum project was to develop fully functioning project using the existing technologies and the knowledge obtained so far and if needed then further increasing that knowledge. The Project intention was to build a full stack e – commerce web application with the theme of selling best seller books online to the Indian market where the user can sign in their account, add their accounts to their accounts, add products to their products to their carts, remove products from their cart, have their information stored on the cloud after their application is closed, search and filter products and make payments during their checkout.

Challenges

One of the biggest challenges faced during the development of the W/A was integrating stripe payment processor with react and Strapi. Earlier React used class-based components with life cycle methods in order to manage component state but the latest

version of react uses react hooks with functional components so the learning curve during development was steep.

Future Scope

Although the requirements set out for the web application have been met, while there are still some future scopes and areas which can be improved later on. A mobile version can be developed for the application so that the users can have access to the application from their mobile phones as well. Also, other online payments like credit/debit cards, UPI payments and other payment methods can be implemented for the application. Products recommendation system can also be implemented for the application.

References

[1] React Documentation, ReactJS Facebook, Silicon Valley, California [Online] https://reactjs.org/docs/getting-started.html

[2] Pandit, N. (2020). What and Why React.js? . c-sharpcorner. Available at: https://www.c-sharpcorner.com/article/what-and-why-reactjs/

[3] Piotr, B. (2019). What is the React.js Framework? When and why should I use React.js in my project? . Netguru. Available at: https://www.netguru.com/blog/what-is-react-js

[4] Kutlu, S. (2017). 7 Reasons why you should use React. Jotform. Available at: https://stories.jotform.com/7-reasons-why-you-should-use-react-ad420c634247

[5] Scott Morris, skill crush. Tech 101: What is React.js? . skillcrush. Available at: https://skillcrush.com/blog/what-is-react-js/#jsx

[6] React enlightenment. What is JSX? Available at: https://www.reactenlightenment.com/react-jsx/5.1.html

[7] Hamedani, M. (2018) React Virtual Dom Explained in Simple English. Programmingwithmosh. Available at: https://programmingwithmosh.com/react/react-virtual-dom-explained/

[8] Chris Minnick (2016) The real benefits of the Virtual DOM in React.js. Available at: https://www.accelebrate.com/blog/the-real-benefits-of-the-virtual-dom-in-react-js/

[9] Valentino, G. (2020). React Hooks Tutorial: useState, useEffect, useReducer. Valentinog. Available at : https://www.valentinog.com/blog/hooks/

[10] Hamedani, M. (2018) React Functional or Class Components: Everything you need to know. Available at: https://programmingwithmosh.com/react/react-functional-components/

[11] Kagga, J. (2018) Understanding React Components. Medium. Available at: https://medium.com/the-andela-way/understanding-react-components37f841c1f3bb

[12] Facebook Inc. Components and Props. Available at: https://reactjs.org/docs/components-and-props.html

[13] Create React App , ReactJS Facebook, Silicon Valley, California [Online] https://create-react-app.dev/docs/getting-started

[14] JS Documentation, Mozilla Corporation, Mountain View, California [Online] https://developer.mozilla.org/en-US/docs/Web/JavaScript.

[15] Stripe Documentation, Stripe, San Francisco, California [Online] https://stripe.com/docs/stripe-js/react.

[16] Strapi Documentation, Strapi, Paris, France [Online] https://strapi.io/documentation/v3.x/getting-started/introduction.html

[17] Stackoverflow (2016). What is the difference between Axios and Fetch? . Available at: https://stackoverflow.com/questions/40844297/what-is-difference-between-axios-and-fetch

[18] SASS Documentation, SASS lang, California [Online] https://sass-lang.com/documentation

[19] React Router DOM, React training, Carlsbad, California [Online] https://reactrouter.com/

[20] Margret, R. (2007). What is requirements analysis (requirements engineering)?. http://searchsoftwarequality.techtarget.com/definition/requirements-analysis

Appendix

GITHUB: https://github.com/sahilssj03/practicum-s3

WEBSITE: https://react-book-store-v1.netlify.app/

BACK-END: https://practicum-backend.herokuapp.com/