A SUMMER TRAINING PROJECT REPORT

ON

"MERN Stack Development"

Submitted By

Ayush Garg

13214803119

Submitted in the partial fulfillment for the award of Degree of

Bachelor of Technology

In

Information Technology

Maharaja Agrasen Institute of Technology, PSP Area, Plot No 1, Sector-22, Rohini, New Delhi-110086

(Affiliated to Guru Gobind Singh Indraprastha University, New Delhi)

CERTIFICATE

udemy

Certificate no: UC-c8dfe3e4-3279-4256-bc1b-804ce6e3eb1a
Certificate url: ude.my/UC-c8dfe3e4-3279-4256-bc1b-804ce6e3eb1a
Reference Number: 0004

CERTIFICATE OF COMPLETION

The Complete JavaScript Course 2021: From Zero to Expert!

Instructors **Jonas Schmedtmann**

Ayush Garg

Date **Sept. 27, 2021**
Length **68.5 total hours**

udemy

Certificate no: UC-96575742-6ad5-439b-98bd-17d02e8c9bdf
Certificate url: ude.my/UC-96575742-6ad5-439b-98bd-17d02e8c9bdf
Reference Number: 0004

CERTIFICATE OF COMPLETION

MERN Stack Front To Back: Full Stack React, Redux & Node.js

Instructors **Brad Traversy**

Ayush Garg

Date **Sept. 14, 2021**
Length **12 total hours**

DECLARATION

I, **Ayush Garg**, roll no-**13214803119**, B. Tech (Semester- 5th) of Maharaja Agrasen Institute of Technology, New Delhi hereby declare that the Project Report is based on my own work carried out during the course of the Training. This report has not been submitted to any other Institute for the award of any other degree/diploma/certificate. I further certify that I have followed the guidelines provided by the college in writing the report.

This project is submitted in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology.

Ayush Garg
Roll No – 13214803119
(I – 7)

ACKNOLEDGEMENT

I avail this Opportunity to express my profound sense of sincere and deep gratitude to many people who are responsible for the knowledge and experience I have gained during the Project Work.

I have great pleasure in expressing my deep sense of gratitude to guide I-Skills for their valuable and prompt guidance without which this project would not have been a successful one. I extend My overwhelming gratitude to **Prof. (Dr.) M L Sharma (HOD IT**) for his valuable guidance and meticulous supervision during the preparation of this Project Report.

My hearty and inevitable thanks to all the respondents who helped me to bring out the project in a successful manner. Last but not the least I extend my gratitude towards my parents, faculties and friends who extended their wholehearted support towards the successful completion of this Project Work/ Training.

Ayush Garg
Roll No – 13214803119
(I – 7)

ABSTRACT

Web app Development is no longer just about publishing your brochure on the Internet. Whether it's about establishing your identity, developing/enhancing an existing identity, producing a periodical online publication, producing an entertainment site, or exchanging data with your partners/clients, the web app must do something for its audience.
The goal of the project was to create a web app "**ProShop**" which is an e-commerce platform. It can be used to showcase products and allow users to buy products from the web app. Technology used for frontend is React and Redux for state management, and Express for backend.

In this build I've followed agile methodology that relies on incremental development. In first iteration I've built basic frontend of my shopping website, in second iteration I've added backend functionality to load data from server and more features are built into the product in subsequent sprints.

Table of Contents

Chapter One

Introduction

1.1 Overview

The 'Online E-commerce Web application' Services department strives to provide solutions to develop and transfer easy and efficient way in the digital age and to help reduces the human pressure and time. To help support shop collections, the digital initiatives, and external partner institution digital projects, It provide services that include the digitization of analog objects, metadata management, digital preservation, and discovery and access of digital collections. **"ProShop"** is a web application written for all operating systems, designed to help users maintain and organize shop virtually. This software is easy to use for both beginners and advanced users. It features a familiar and well thought out, an attractive user interface, combined with strong searching Insertion

1.2 Background Study

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace.

The objective of this project is to develop a general-purpose e-commerce store where any product can be bought from the comfort of home through the Internet. However, for implementation purposes, this paper will deal with an online ecommerce store.

An online store is a virtual store on the Internet where customers can browse the catalogue and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction.

Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as a credit card number.

1.3 Project Planning

Project planning is part of project management, which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to

complete the work are listed and grouped into a work breakdown structure. The logical dependencies between tasks are defined using an activity network diagram that enables identification of the critical path. Float or slack time in the schedule can be calculated using project management software. Then the necessary resources can be estimated and costs for each activity can be allocated to each resource, giving the total project cost. At this stage, the project plan may be optimized to achieve the appropriate balance between resource usage and project duration to comply with the project objectives. Once established and agreed, the plan becomes what is known as the baseline. Progress will be measured against the baseline throughout the life of the project.

1.4 Purposes

The project is about to handle all the information of the shop regarding members. Also it manages resources which were managed and handled by manpower previously. The main purpose of the project is to integrate distinct sections of the shop into consistent manner so that complex functions can be handled smoothly. The project aims at the following matters:

- Automation of product manipulation.
- Buying products.
- To manage information of different types of items.
- Consistently update information of all the item.
- Managing security by providing authorized email & password.
- Manages database efficiently.

Chapter Two

System Design

2.1 Design

The system is divided into some parts these are Register system, Login System, Search System, Buying System, Order Received System, Viewing System side with database represent the server using Express and MongoDB. System diagram and system database diagram illustrated in figure.

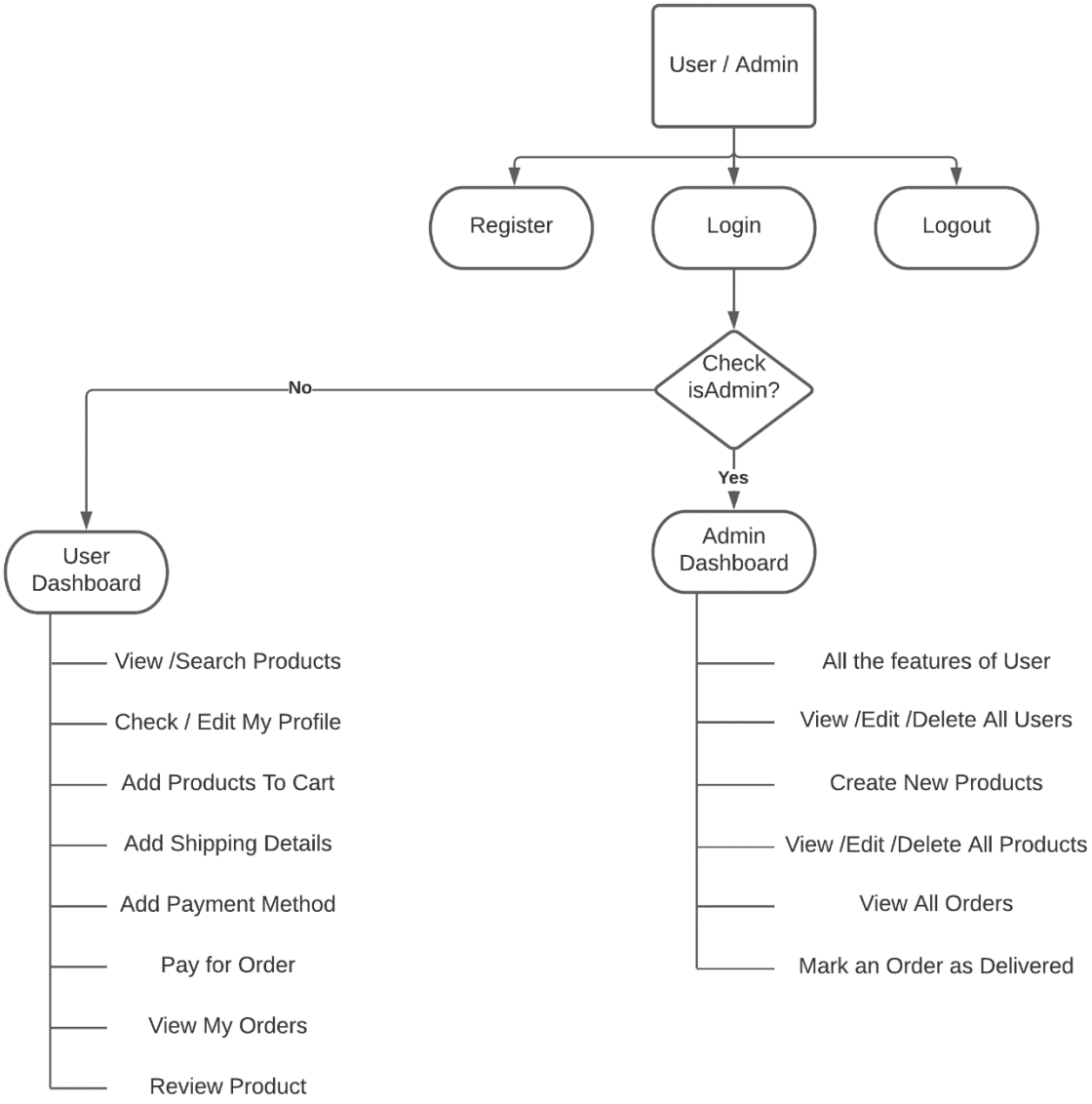


FIGURE: ADMIN / USER FEATURES

2.2 User Characteristic

Admin has all the rights to access the system. He is the one who has all rights to view the members and product details, modify those details. He can add various product based on the category. He can also set the available quantity of a product and its reasonable price. Admin can also view and edit the details of a user. Also, he can delete a user.

Users can log in to the system by using his specific email and password and register for a new account. User can view the products and order the products according to their own needs. He can view his profile and update his details. He can update his personal information by logging into the system. User can find various product by using search option easily. A product carousel is also shown for the top-rated products.

2.3 System Information

This system is an automated Shop Management System. Through the software user can add members, add product, search product, update information, edit information, buy the product in quick time. The system has the following advantages:

- User friendly interface
- Fast access to database
- Search facility
- Look and Feel Environment

2.4 Feasibility Analysis

Whatever we think need not be feasible. It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

Technical Feasibility

It is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance.

Economical Feasibility

Development of this application is highly economically feasible. The organization needed not spend much more for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. If we are doing so, we can attain the maximum usability of the corresponding resource. Even after the development, the organization will not be in a condition to invest more in the organization. Therefore, the system is economically feasible.

2.5 Project Structure

The Project i.e., ProShop is divided into multiple files. The project structure is as follows.

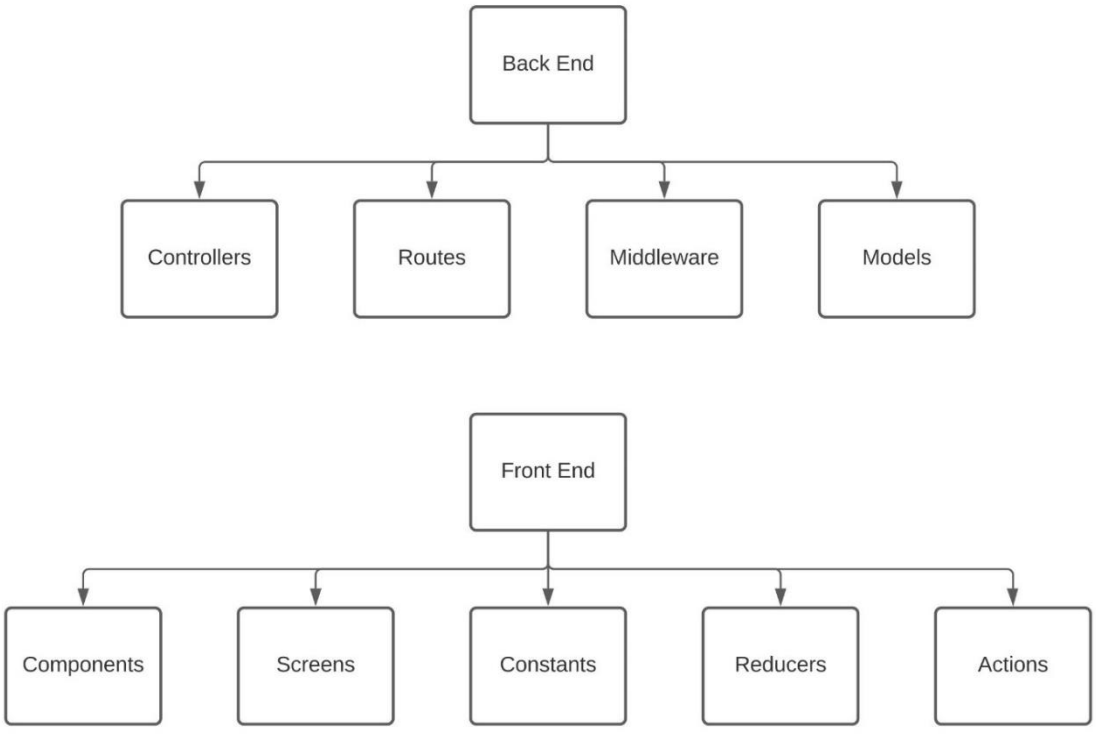


Figure: Structure of Project

Backend

1. Controllers: It includes all the controller function such as listProductDetails , listProductsById , etc.
2. Routes: It includes all the routes / endpoints the server can take and process on.
3. Middleware: It includes custom error handler and function to check if the request made to the server is done by a logged in user and if the user is Admin or not.
4. Models: It contains all the custom function related to database such as hashing password before storing and has basic schema of each database.

Frontend

1. Components: All the reusable components are present here such as Header, Footer, Message, Loader, etc.
2. Screens: Every file here corresponds to a Screen in client, eg. HomepageScreen, CartScreen, etc.
3. Constants: Every constant for reducer is stored here as switch case works on strings and we want to make them constant all over the environment.
4. Reducers: It contains all the reducer functions which tells what to do when a particular request is dispatched.
5. Actions: Contains all the functions which make call to the backend server and serve data to our redux state to load up in frontend components.

2.6 Data Flow Diagram

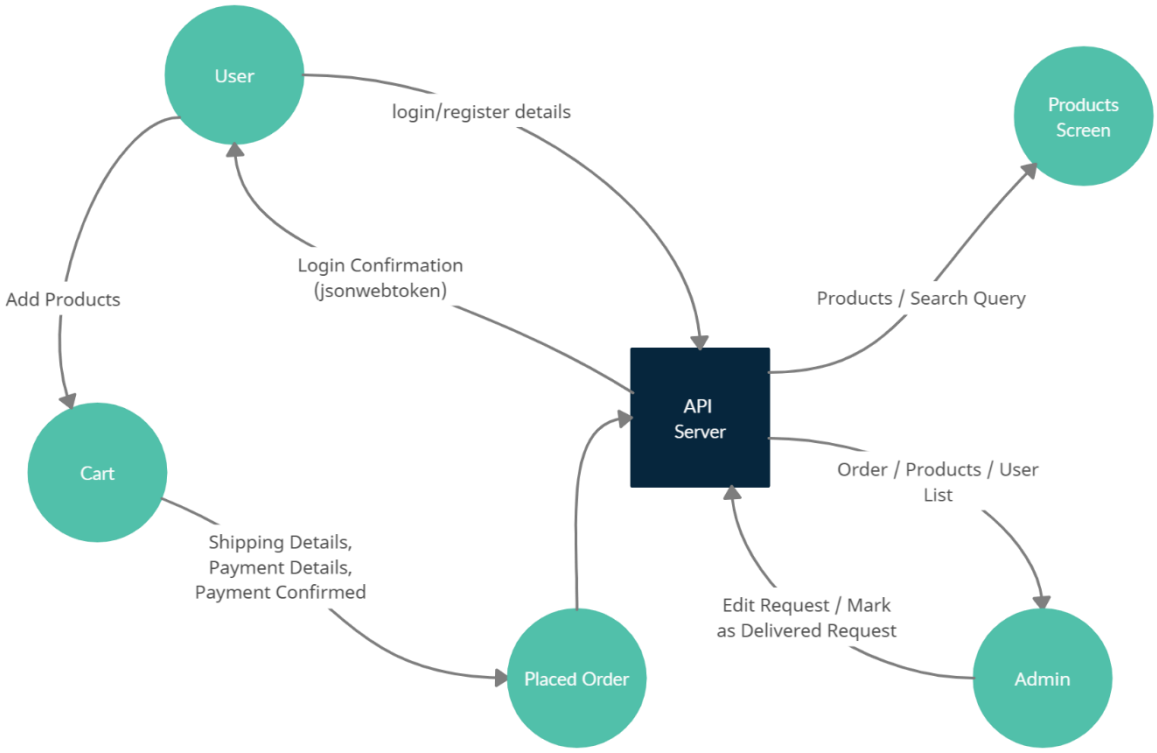


Figure: Data Flow Diagram

Chapter Three

Hardware and System Requirement

3.1 Hardware Required

- **Processor** : Pentium 4 (or equivalent)
- **RAM** : 2 GB RAM
- **Hard Disk Space** : 1 GB
- **Input Devices** : Keyboard, Mouse
- **Output Devices** : Monitor
- **Connection to the internet**

3.2 Software Required

- ✓ **Operating System** : Linux, Ubuntu, Mac, Windows XP,7,8,8.1,10
- ✓ **Browser** : Latest version of Chrome, Firefox or similar browser
- ✓ **Frontend** : React, Redux, Bootstrap
- ✓ **Backend** : Express, MongoDB
- ✓ **Node.js version** : v16.x.x and above
- ✓ **NPM version** : v7.x.x and above

Chapter Four

Implementing Tools for the Project

4.1 Tools

- ✓ **M**ongoDB
- ✓ **E**xpress.js
- ✓ **R**eact
- ✓ **N**ode.js
- ✓ Bootstrap

4.2 What is MERN

MERN is an acronym used to describe a specific set of JavaScript based technologies that are used in the web application development process. It is designed with an idea to make the development process as smooth as possible. MERN includes the following open-source components:

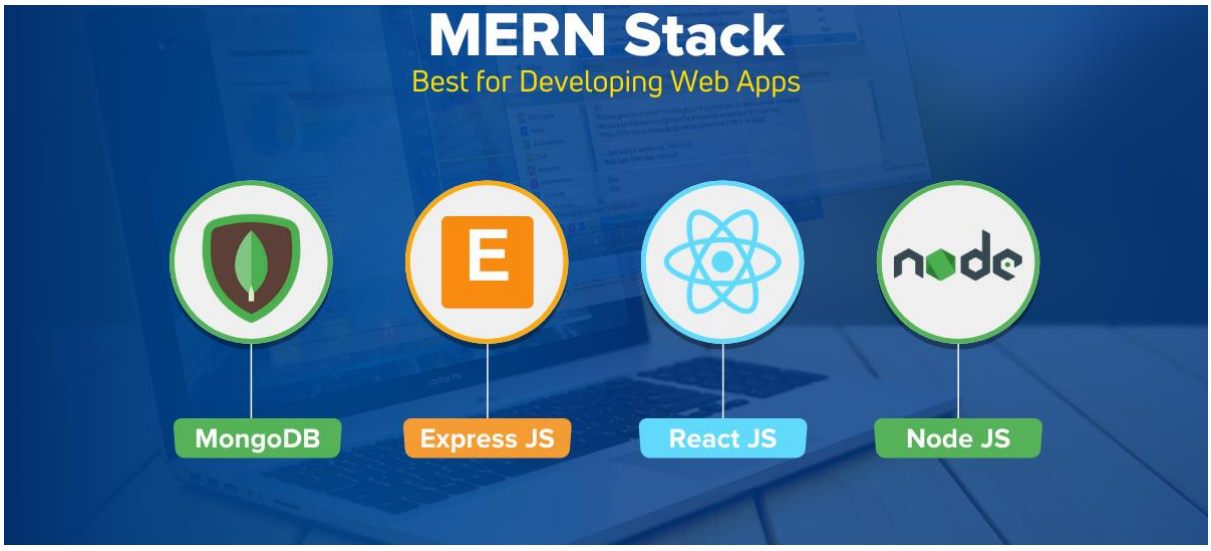


Figure: MERN Stack

- Mongo DB
- Express JS
- React JS/ Redux
- Node JS

Every single one of these components plays a crucial role in the web app development process. All of these provide an end-to-end framework for developers to work in. For instance, Mongo DB is a database system, Node JS is a back-end runtime environment, Express JS is a back-end web framework and React is a front-end framework.

So, let's take a bit closer look at every single one of these components:

1. Mongo DB

In case you're not familiar with Mongo DB, this is a free open-source, cross-platform document-oriented database program. It is classified as a No SQL database program, which means that data is stored in flexible documents with JSON-based query language. This also means that the size of the content number of fields in the documents tends to vary. The whole data is structured in a way to be prone to change over time.

MongoDB is known as a flexible solution that is always easy to scale. This specific database system is developed by Mongo DB Inc., and is published under a combination of the GNU Affero General Public License and the Apache License.

How It Works

Figure: What is MongoDB

MongoDB makes use of records which are made up of documents that contain a data structure composed of field and value pairs. Documents are the basic unit of data in MongoDB. The documents are similar to JavaScript Object Notation, but use a variant called Binary JSON (BSON). The benefit of using BSON is that it accommodates more data types. The fields in these documents are similar to the columns in a relational database. Values contained can be a variety of data types, including other documents, arrays and arrays of documents, according to the MongoDB user manual. Documents will also incorporate a primary key as a unique identifier.

Sets of documents are called *collections*, which function as the equivalent of relational database tables. Collections can contain any type of data, but the restriction is the data in a collection cannot be spread across different databases.

The mongo shell is a standard component of the open-source distributions of MongoDB. Once MongoDB is installed, users connect the mongo shell to their running MongoDB instances. The mongo shell acts as an interactive JavaScript interface to MongoDB, which allows users to query and update data, and conduct administrative operations.

The NoSQL DBMS uses a single master architecture for data consistency, with secondary databases that maintain copies of the primary database. Operations are automatically replicated to those secondary databases for automatic failover.

Pros and Cons

Figure: What is MongoDB

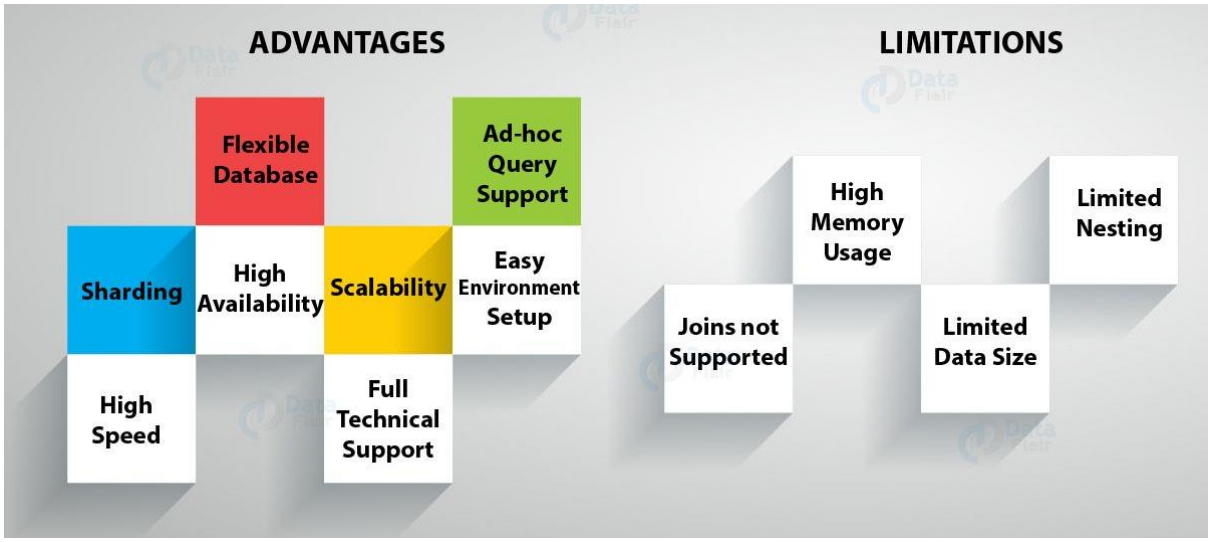


Figure: Advantages and Limitations

Like other NoSQL databases, MongoDB doesn't require predefined schemas. It stores any type of data. This gives users the flexibility to create any number of fields in a document, making it easier to scale MongoDB databases compared to relational databases.

One of the advantages of using documents is that these objects map to native data types in a number of programming languages. Also, having embedded documents reduces the need for database joins, which can reduce costs.

A core function of MongoDB is its horizontal scalability, which makes it a useful database for companies running big data applications. In addition, sharding allows the database to distribute data across a cluster of machines. Newer versions of MongoDB also support the creation of zones of data based on a shard key.

2. Express JS

Express.js is a free and open-source web application framework for Node.js. It is used for designing and building web applications quickly and easily. Web applications are web apps that you can run on a web browser. Since Express.js only requires JavaScript, it becomes easier for programmers and developers to build web applications and API without any effort.

Express.js is a framework of Node.js which means that most of the code is already written for programmers to work with. You can build a single page, multi-page, or hybrid web application using Express.js. Express.js is lightweight and helps to organize web applications on the server-side into a more organized MVC architecture.

It is important to learn JavaScript and HTML to be able to use Express.js. Express.js makes it easier to manage web applications. It is a part of a JavaScript based technology called MEAN software stack which stands for MongoDB, ExpressJS, AngularJS, and Node.js. Express.js is the backend part of MEAN and manages routing, sessions, HTTP requests, error handling, etc.

The JavaScript library of Express.js helps the programmers to build efficient and fast web apps. Express.js enhances the functionality of the node.js. In fact, if you don't use Express.js, then you have to do a lot of complex programming to build an efficient API. It has made programming in node.js effortless and has given many additional features.

What is Express.js?

Express.js supports JavaScript which is a widely used language that is very easy to learn and is also supported everywhere. Therefore, if you already know JavaScript, then it will be really easy for you to do programming using Express.js.

With the help of Express.js, you can easily build different kinds of web applications in a short period of time. Express.js provides a simple routing for requests made by clients. It also provides a middleware that is responsible for making decisions to give the correct responses for the requests made by the client

Without Express.js, you have to write your own code to build a routing component which is a time consuming and tedious task. Express.js offers simplicity, flexibility, efficiency, minimalism, and scalability to the programmers. It also has the advantage of powerful performance as it is a framework of Node.js.

Node.js carries all the executions really fast with the help of Event Loop that avoids any kind of inefficiency. The powerful performance of Node.js and the ease of coding using Express.js are the most popular features loved by web application developers. Since Express.js is written in JavaScript, you can build websites, web applications, or even mobile apps using it.

Features of Express.js

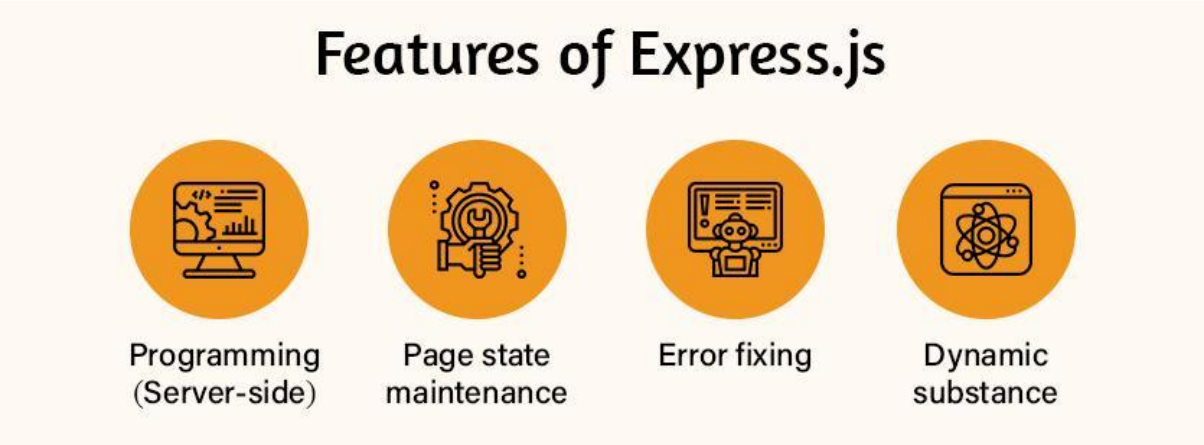


Figure: Features of Express.js

1. *Faster Server-side development*

Express.js provides many commonly used features of Node.js in the form of functions that can be readily used anywhere in the program. This removes the need to code for several hours and thus saves time.

2. *Middleware*

Middleware is a part of the program that has access to the database, client request, and the other middlewares. It is mainly responsible for the systematic organization of different functions of Express.js.

3. *Routing*

ExpressJS provides a highly advanced routing mechanism which helps to preserve the state of the webpage with the help of their URLs.

4. *Templating*

ExpressJS provides templating engines that allow the developers to build dynamic content on the web pages by building HTML templates on the server-side.

5. *Debugging*

Debugging is crucial for the successful development of web applications. ExpressJS makes debugging easier by providing a debugging mechanism that has the ability to pinpoint the exact part of the web application which has bugs.

3. React JS/ Redux

React.js is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It's used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components. React was first created by Jordan Walke, a software engineer working for Facebook. React first deployed on Facebook's newsfeed in 2011 and on Instagram.com in 2012.

React allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple. It works only on user interfaces in the application. This corresponds to the view in the MVC template. It can be used with a combination of other JavaScript libraries or frameworks, such as Angular JS in MVC.

With React, the same code can run on both the server and the browser. Because of these, most devs call React the "heart and soul of the MERN stack".

React.Js Features

Let us take a closer look at some important features of React.

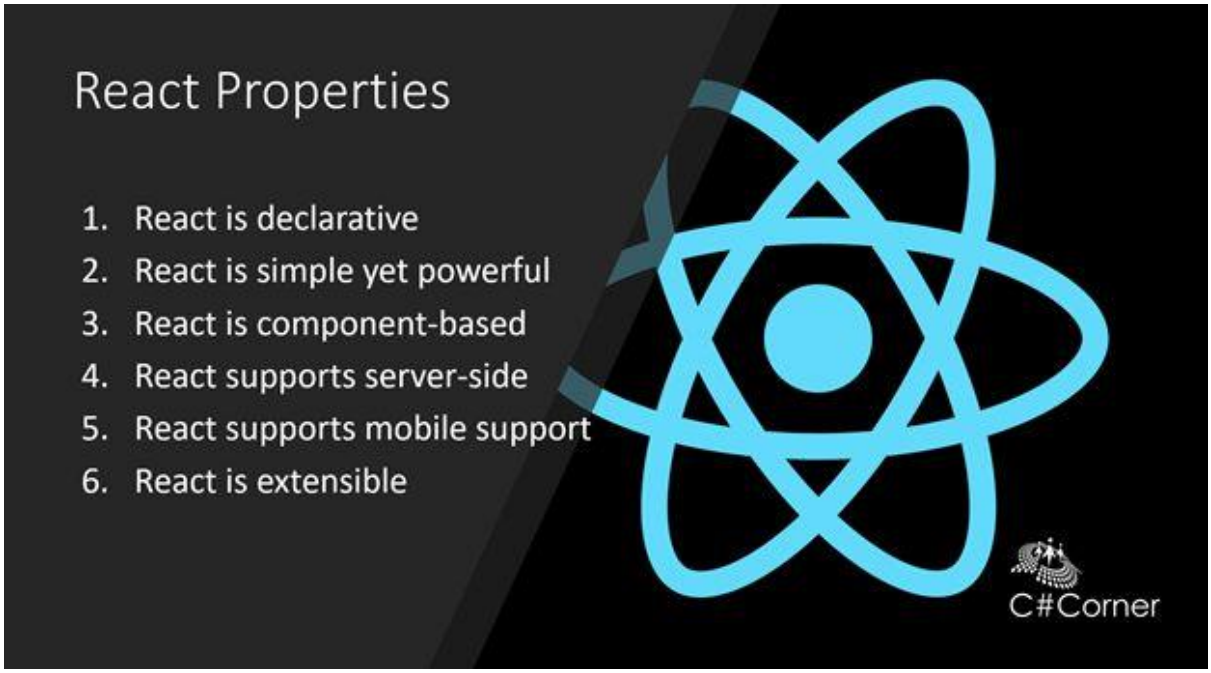


Figure: Properties of React

React.js properties include the following

- React.js is declarative
- React.js is simple
- React.js is component based

- React.js supports server side
- React.js is extensive
- React.js is fast
- React.js is easy to learn

Why React?

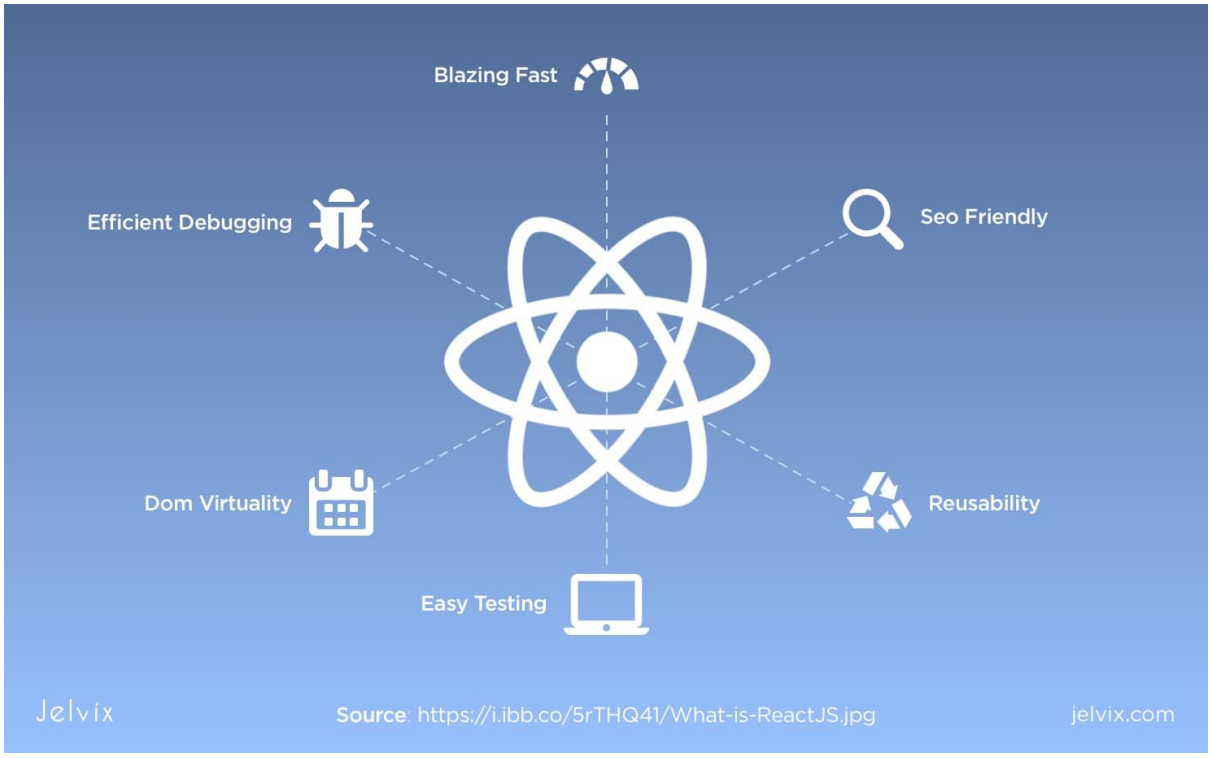


Figure: Why React?

Now, the main question arises in front of us is why one should use React. There are so many open-source platforms for making the front-end web application development easier, like Angular. Let us take a quick look on the benefits of React over other competitive technologies or frameworks. With the front-end world-changing on a daily basis, it's hard to devote time to learning a new framework – especially when that framework could ultimately become a dead end. So, if you're looking for the next best thing but you're feeling a little bit lost in the framework jungle, I suggest checking out React.

1. *Simplicity*

ReactJS is just simpler to grasp right away. The component-based approach, well-defined lifecycle, and use of just plain JavaScript make React very simple to learn, build a professional web (and mobile applications), and support it. React uses a special syntax called JSX which allows you to mix HTML with JavaScript. This is not a requirement; Developer can still write in plain JavaScript but JSX is much easier to use.

2. *Easy to learn*

Anyone with a basic previous knowledge in programming can easily understand React while Angular and Ember are referred to as 'Domain-specific Language', implying that it is difficult to learn them. To react, you just need basic knowledge of CSS and HTML.

3. *Native Approach*

React can be used to create mobile applications (React Native). And React is a diehard fan of reusability, meaning extensive code reusability is supported. So at the same time, we can make IOS, Android and Web applications.

4. *Data Binding*

React uses one-way data binding and an application architecture called Flux controls the flow of data to components through one control point – the dispatcher. It's easier to debug self-contained components of large ReactJS apps.

5. *Performance*

React does not offer any concept of a built-in container for dependency. You can use Browserify, Require JS, EcmaScript 6 modules which we can use via Babel, ReactJS-di to inject dependencies automatically.

6. *Testability*

ReactJS applications are super easy to test. React views can be treated as functions of the state, so we can manipulate with the state we pass to the ReactJS view and take a look at the output and triggered actions, events, functions, etc.

4. Node JS

Originally built for Google Chrome and later on open-sourced, Node JS is a cross-platform run-time JavaScript environment used for executing JavaScript code outside of a browser.

As you know, JavaScript was originally used for front-end scripting, but Node JS has enabled devs to use it to write command line tools and back-end scripts for the purpose of creating dynamic web page content before the page is sent to the user's web browser.

Node JS was designed with an idea of allowing devs to build scalable network applications.

The Main Benefit of a MERN Stack

At Share IT, we primarily use this stack because everything is done in JavaScript. As you know, JavaScript is everywhere. It is used both on the front-end and back-end side. Because of this, there's no need for context switching.

Tech stack that utilize multiple coding languages, force devs to figure out how to mix them together. Since MERN is JavaScript-based, devs only need to master a single coding language, which makes things a million times easier.

What is Node.js?

- Node.js is an open-source server environment
- Node.js is free
- Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- Node.js uses JavaScript on the server
-

Why Node.js?

A common task for a web server can be to open a file on the server and return the content to the client.

Here is how PHP or ASP handles a file request:

1. Sends the task to the computer's file system.
2. Waits while the file system opens and reads the file.
3. Returns the content to the client.
4. Ready to handle the next request.

Here is how Node.js handles a file request:

1. Sends the task to the computer's file system.
2. Ready to handle the next request.

3. When the file system has opened and read the file, the server returns the content to the client.

Node.js eliminates the waiting, and simply continues with the next request.

Node.js runs single-threaded, non-blocking, asynchronously programming, which is very memory efficient

4.3 NPM packages

- **react:** The react package contains only the functionality necessary to define React components. It is typically used together with a React renderer like react-dom for the web, or react-native for the native environments.
- **react-scripts:** This package includes scripts and configuration used by Create React App
- **react-bootstrap**: Bootstrap 5 components built with React.
- **bcryptjs**: Used to hash password to store in database to maintain security.
- **concurrently:** Run multiple commands concurrently. Like npm run dev& npm run client. Helps us to start backend and frontend with single command.
- **dotenv:** Loads environment variables from a .env file into process.env.
- **express**: This is the express.js package that we used to make backend server.
- **express-async-handler:** Middleware for handling exceptions inside of async express routes and passing them to our custom express error handlers.
- **jsonwebtoken:** It provides a web token to the logged in user and helps us to authenticate the user.
- **mongoose:** Mongoose is a MongoDB object modelling tool designed to work in an asynchronous environment. Mongoose supports both promises and callbacks.
- **morgan:** Dev dependency - HTTP request logger middleware for node.js
- **multer:** Multer is a node.js middleware for handling multipart/form-data, which is primarily used for uploading files. It is written on top of busboy for maximum efficiency.
- **nodemon:** Dev dependency – it is a tool that helps develop node.js based applications by automatically restarting the node application when file changes in the directory are detected.
- **react-helmet:** Reusable component to make changes with the document head. It is beginner friendly and used in the project to change meta accordingly.
- **react-paypal-button-v2:** An easy and simple to use React button component to implement PayPal's Checkout with Smart Payment Buttons V2
- **redux:** Redux is a predictable state container for JavaScript apps. It helps you write applications that behave consistently, run in different environments (client, server, and native), and are easy to test.
- **react-redux:** official react binding for redux

- **redux-thunk:** Thunk middleware for Redux. It allows writing functions with logic inside that can interact with a Redux store's dispatch and getState methods.
- **redux-devtools-extension:** Used to make the chrome extension of Redux Devtools to work.
- **axios:** Promise based HTTP client for the browser and node.js
- **react-router-dom:** The react-router-dom package contains bindings for using React Router in web applications
- **react-router-bootstrap:** The react-router-bootstrap package is used to make react-bootstrap Container clickable and attached to a link.
- **colors:** Dev dependency – Used to add colors in console to stand out certain errors and warnings.

Chapter Five

Project Database and Table

5.1 Database Design

Database is critical for all businesses. A good database does not allow any form of anomalies and stores only relevant information in an ordered manner. If a database has anomalies, it is affecting the efficiency and data integrity. For example, delete anomaly arise upon the deletion of a row which also forces other useful data to be lost. As such, the tables need to be normalized. This fulfils the last objective of ensuring data are accurate and retrieved correctly.

Database files are the key source of information into the system. It is the process of designing database files, which are the key source of information to the system. The files should be properly designed and planned for collection, accumulation, editing and retrieving the required information.

The organization of data in database aims to achieve three major objectives:

- ✓ Data integration
- ✓ Data integrity
- ✓ Data independence

5.2 Order Model

```
const orderSchema = mongoose.Schema(
{
    user: { type: mongoose.Schema.Types.ObjectId, required: true, ref: 'User' },
    orderItems: [
      {
        name: { type: String, required: true },
        qty: { type: Number, required: true },
        image: { type: String, required: true },
        price: { type: Number, required: true },
        product: {type: mongoose.Schema.Types.ObjectId,required: true,ref:
'Product'},
      },
    ],
    shippingAddress: {
      address: { type: String, required: true },
      city: { type: String, required: true },
      postalCode: { type: String, required: true },
      country: { type: String, required: true },
    },
    paymentMethod: { type: String, required: true },
    paymentResult: {
      id: { type: String },
      status: { type: String },
      update_time: { type: String },
      email_address: { type: String },
    },
    taxPrice: {type: Number,required: true,default: 0.0},
    shippingPrice: {type: Number,required: true,default: 0.0},
    totalPrice: {type: Number,required: true,default: 0.0},
    isPaid: {type: Boolean,required: true,default: false},
    paidAt: {type: Date},
    isDelivered: {type: Boolean,required: true,default: false},
    deliveredAt: {type: Date},
  },{timestamps: true});
```

5.3 Product Model

```
const reviewSchema = mongoose.Schema(
  {
    name: {type: String,required: true},
    rating: {type: Number,required: true},
    comment: {type: String,required: true},
    user: {type: mongoose.Schema.Types.ObjectId,required: true,ref: 'User'},
  }, { timestamps: true });

const productSchema = mongoose.Schema(
  {
    user: {type: mongoose.Schema.Types.ObjectId,required: true,ref: 'User'},
    name: {type: String,required: true},
    image: {type: String,required: true},
    brand: {type: String,required: true},
    category: {type: String,required: true},
    description: {type: String,required: true},
    reviews: [reviewSchema],
    rating: {type: Number,required: true,default: 0},
    numReviews: {type: Number,required: true,default: 0},
    price: {type: Number,required: true,default: 0},
    countInStock: {type: Number,required: true,default: 0},
  },{ timestamps: true }
);
```

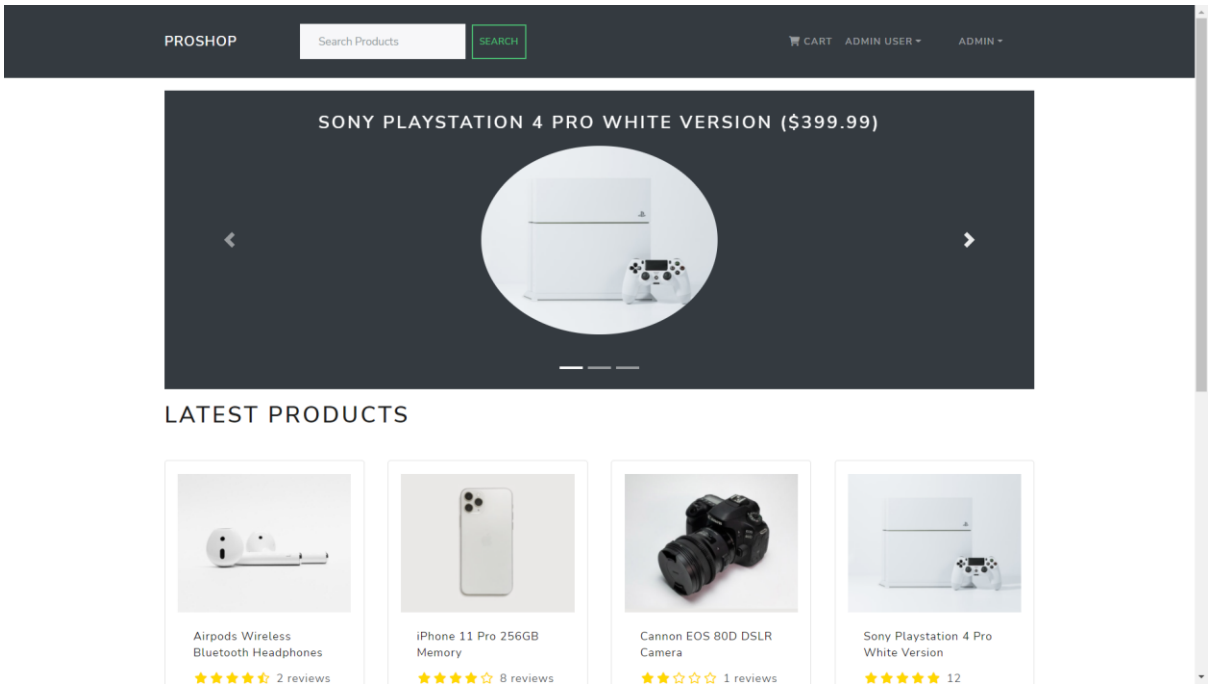
5.4 User Model

```
const userSchema = mongoose.Schema(
  {
    name: {type: String,required: true},
    email: {type: String,required: true,unique: true},
    password: {type: String,required: true},
    isAdmin: {type: Boolean,required: true,default: false},
  },
  { timestamps: true }
);
```

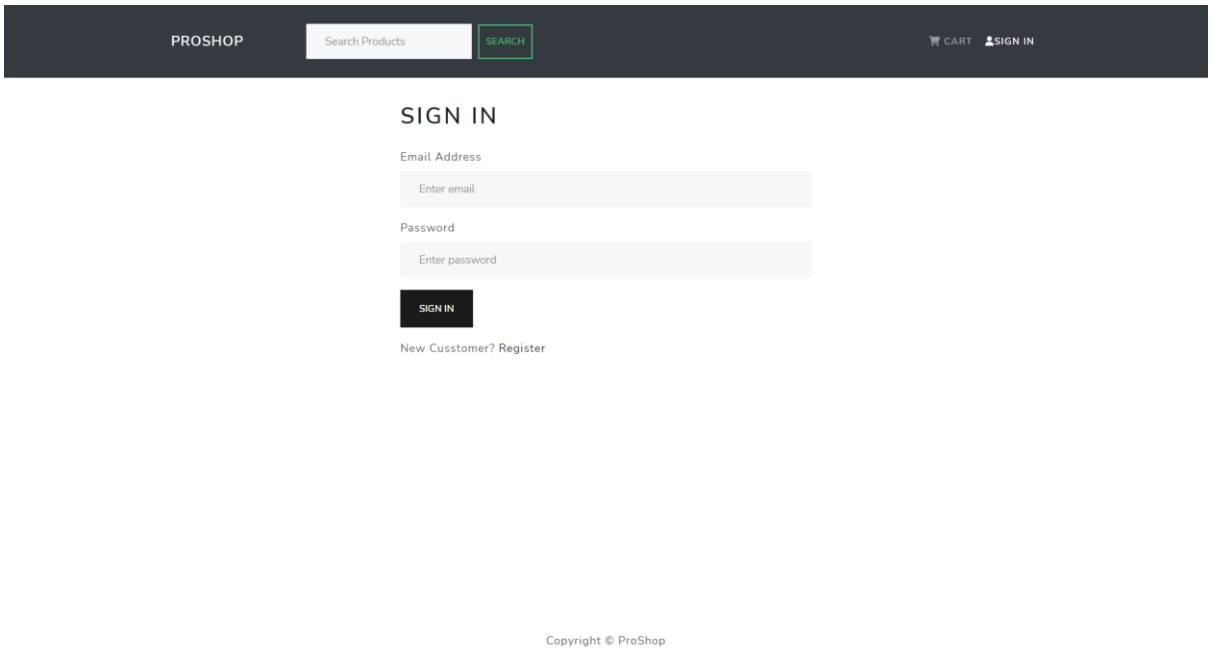
Chapter Six

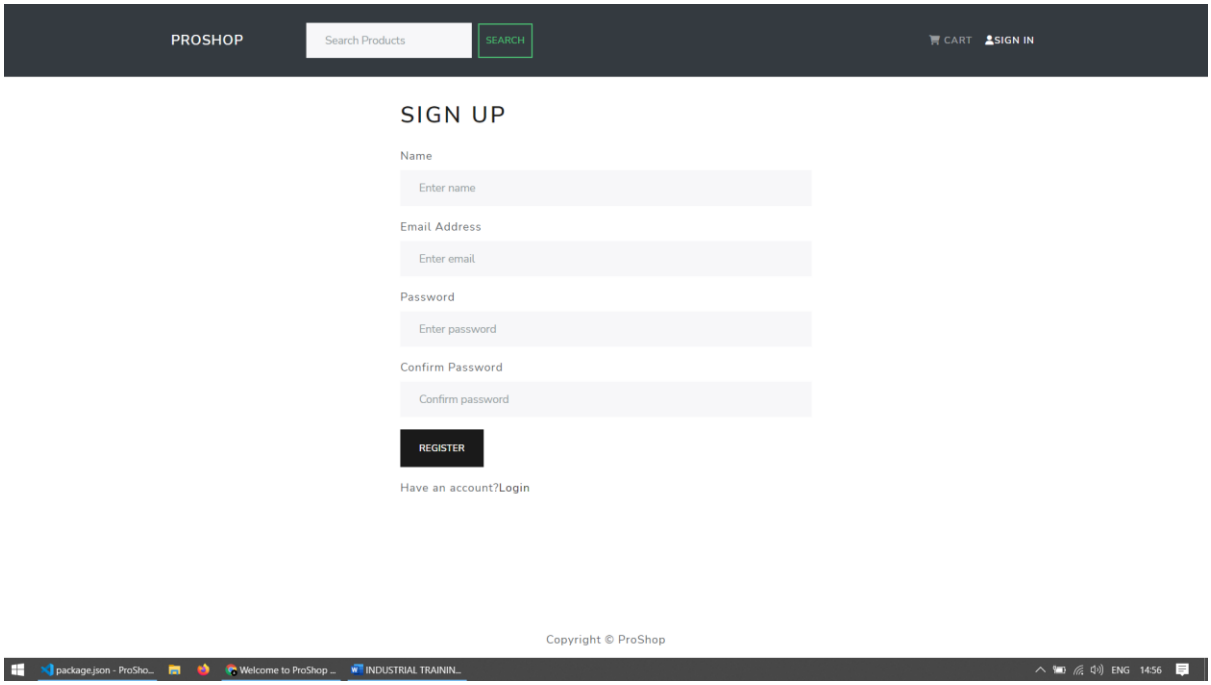
Project Model View

6.1 Home Page

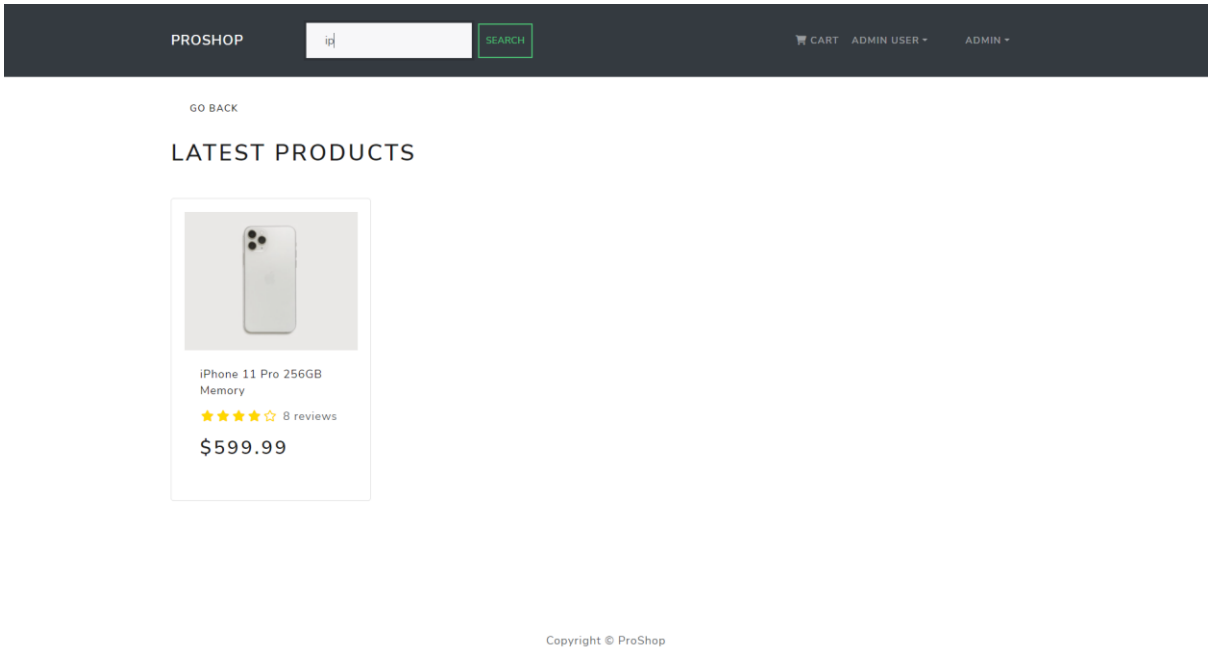


6.2 Login Screen/ Register Screen

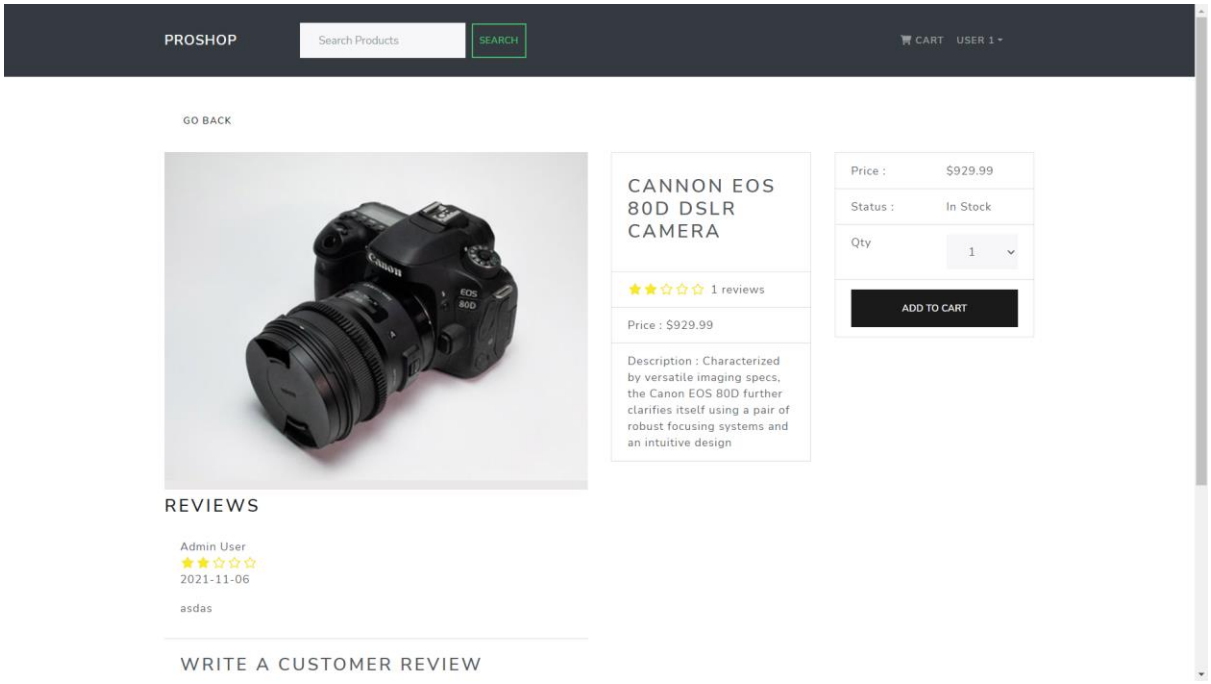




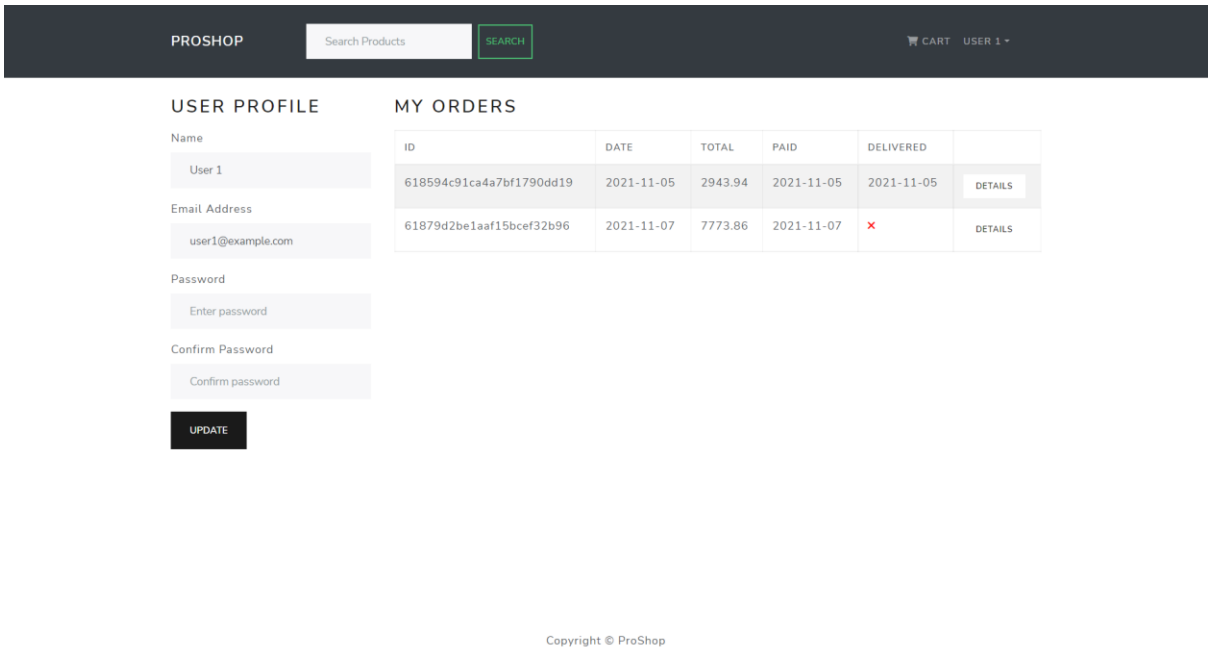
6.3 Search Screen



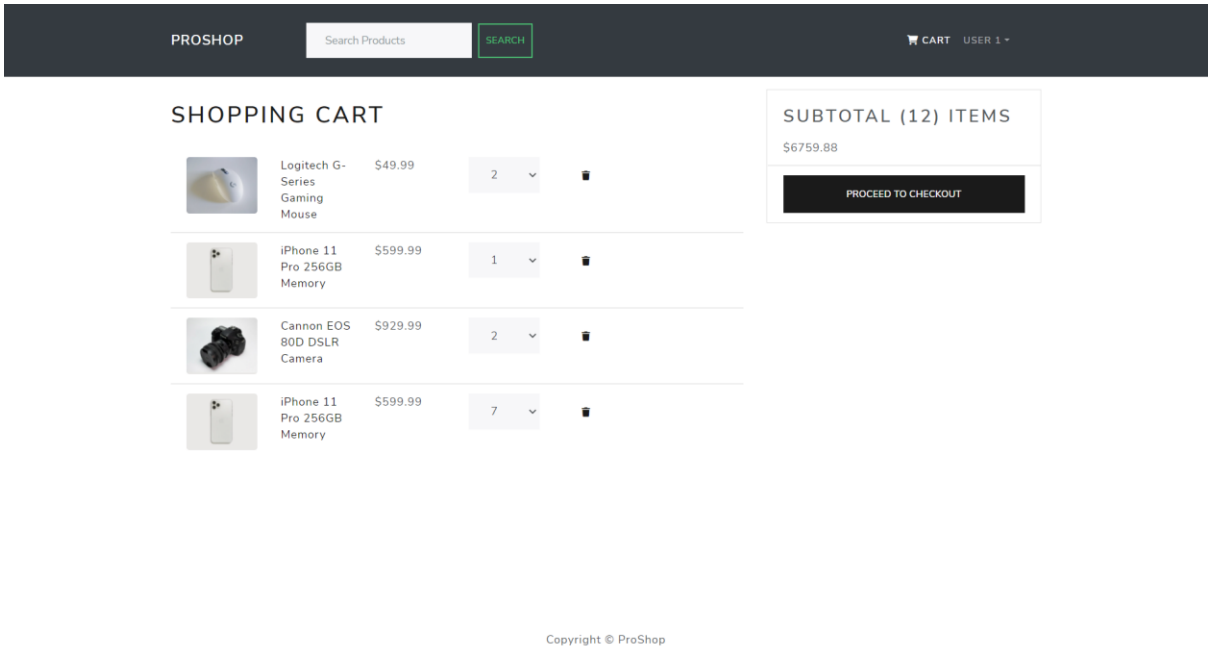
6.4 Single Product Screen



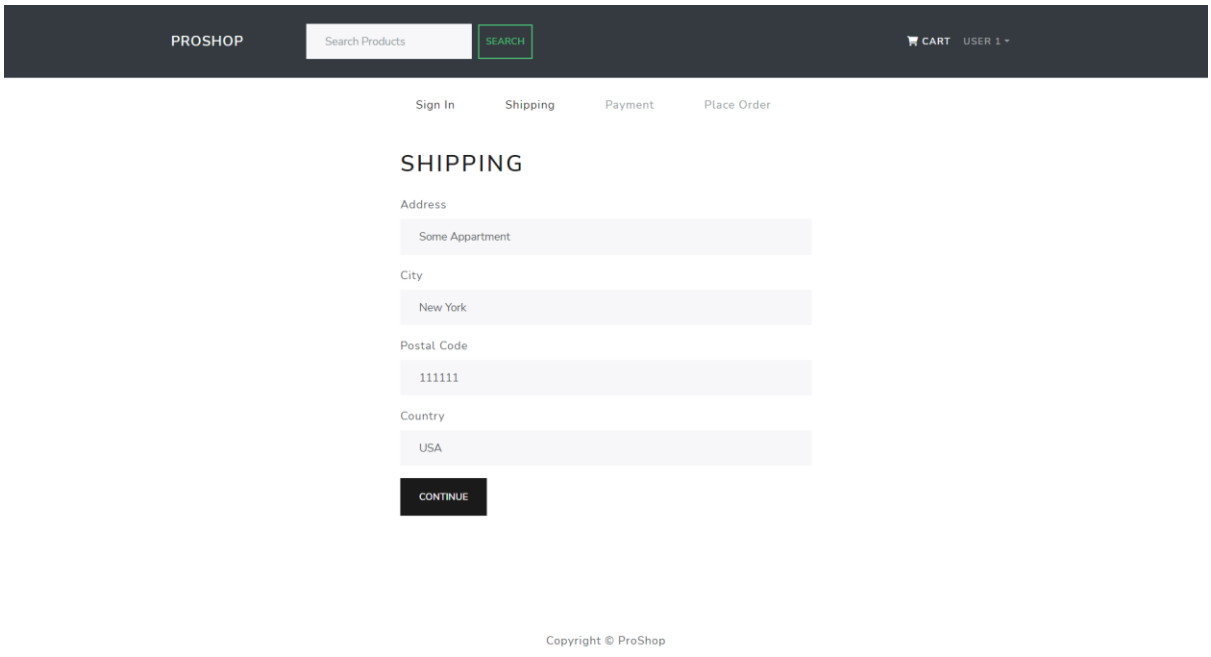
6.5 Profile Screen



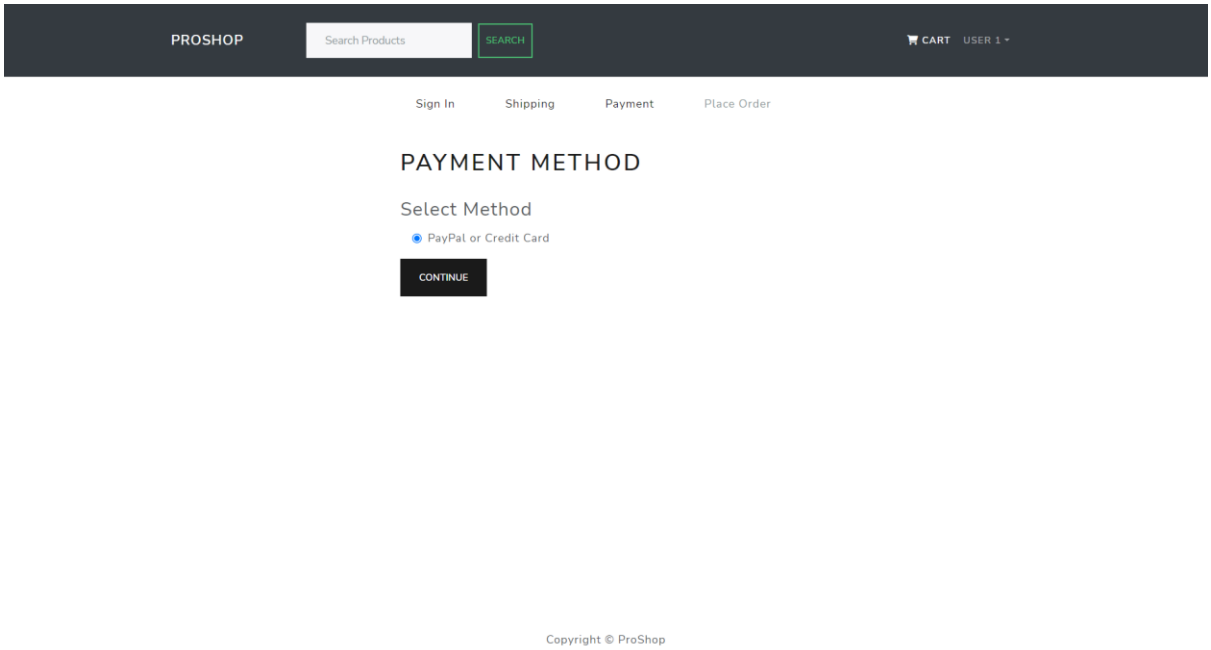
6.6 Cart Screen



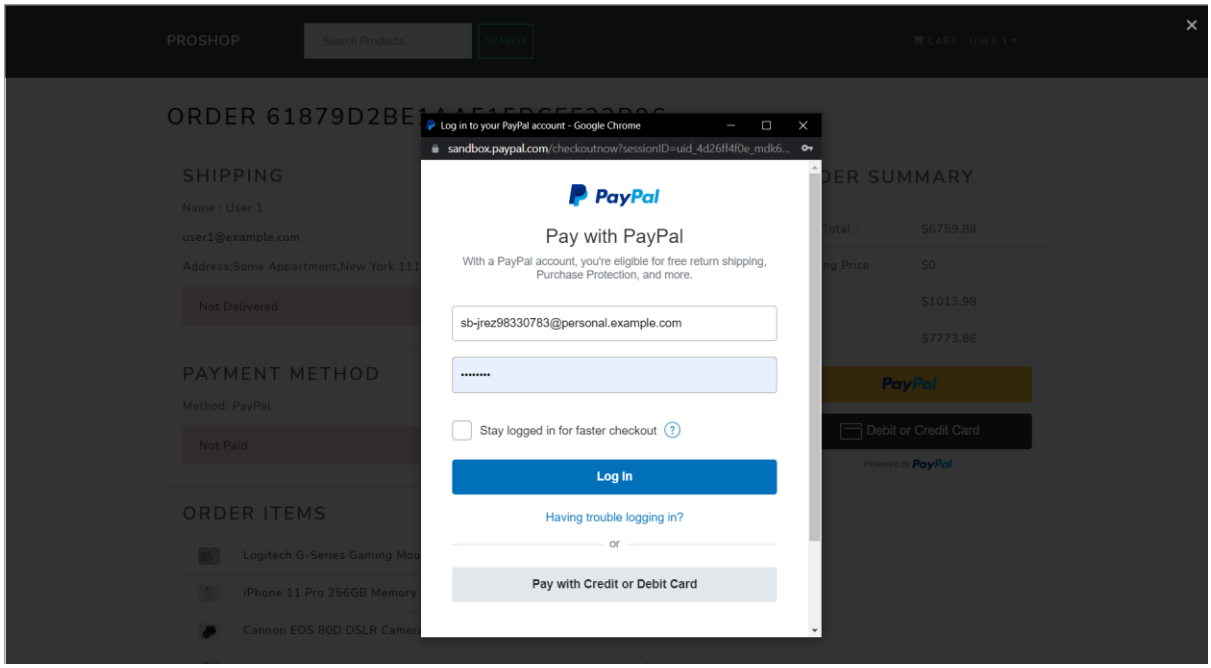
6.7 Shipping Screen



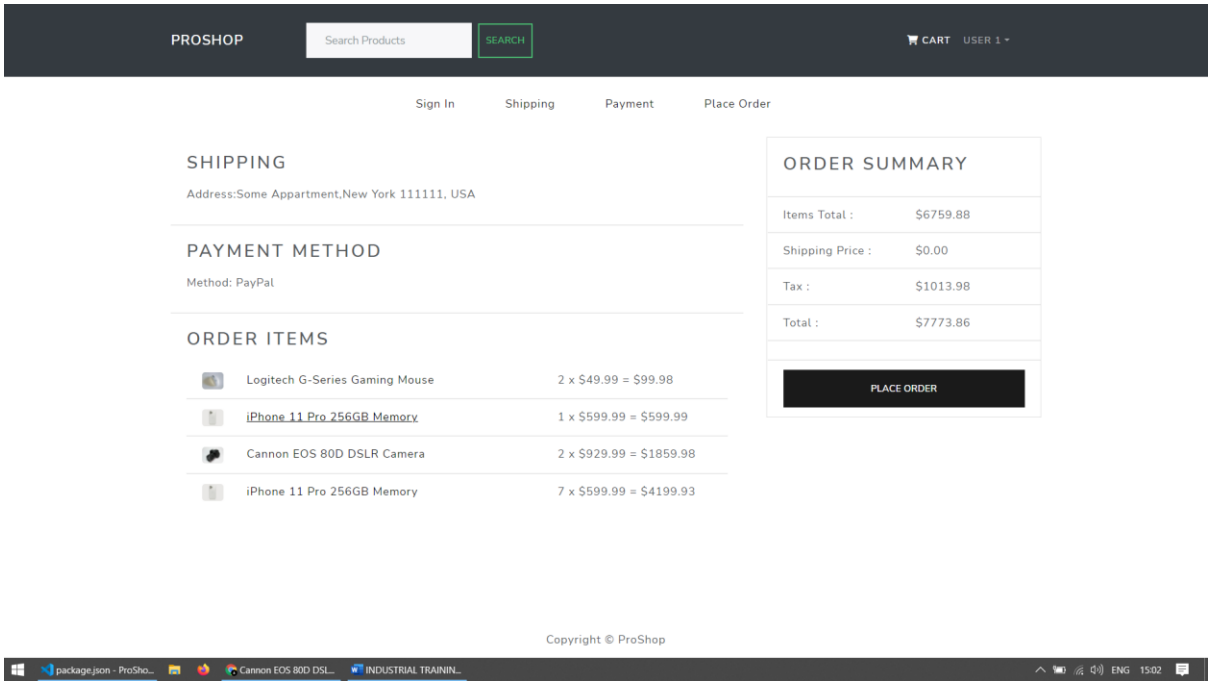
6.8 Payment Method Screen



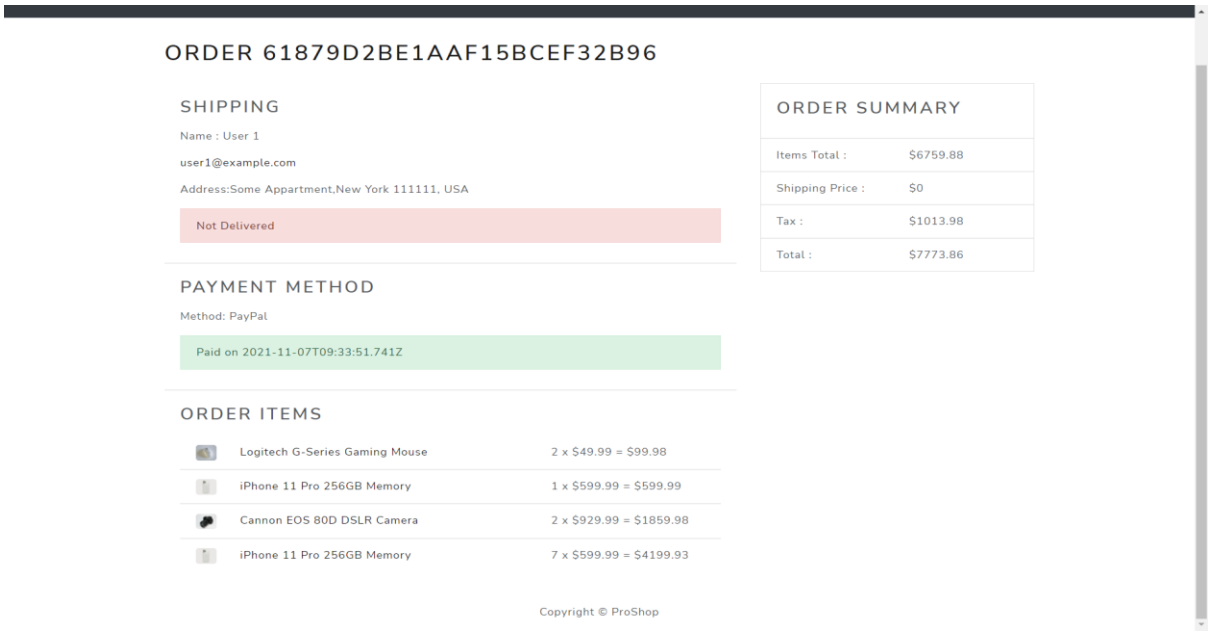
6.9 Payment Screen



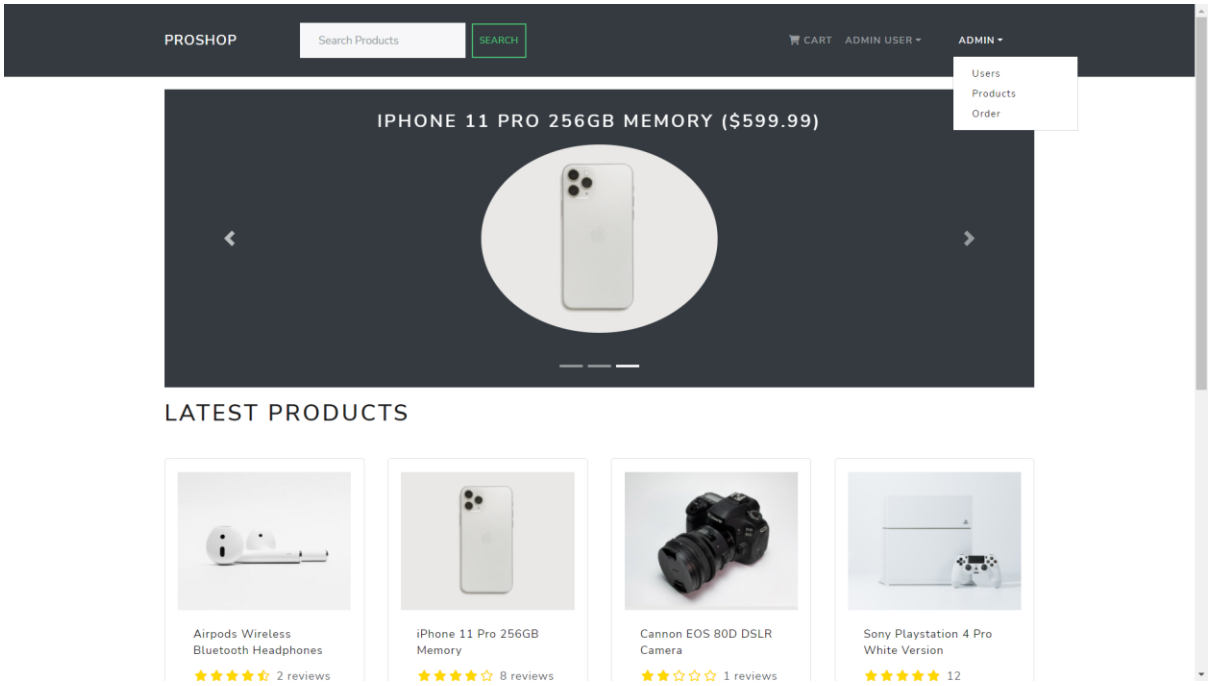
6.10 Place Order Screen



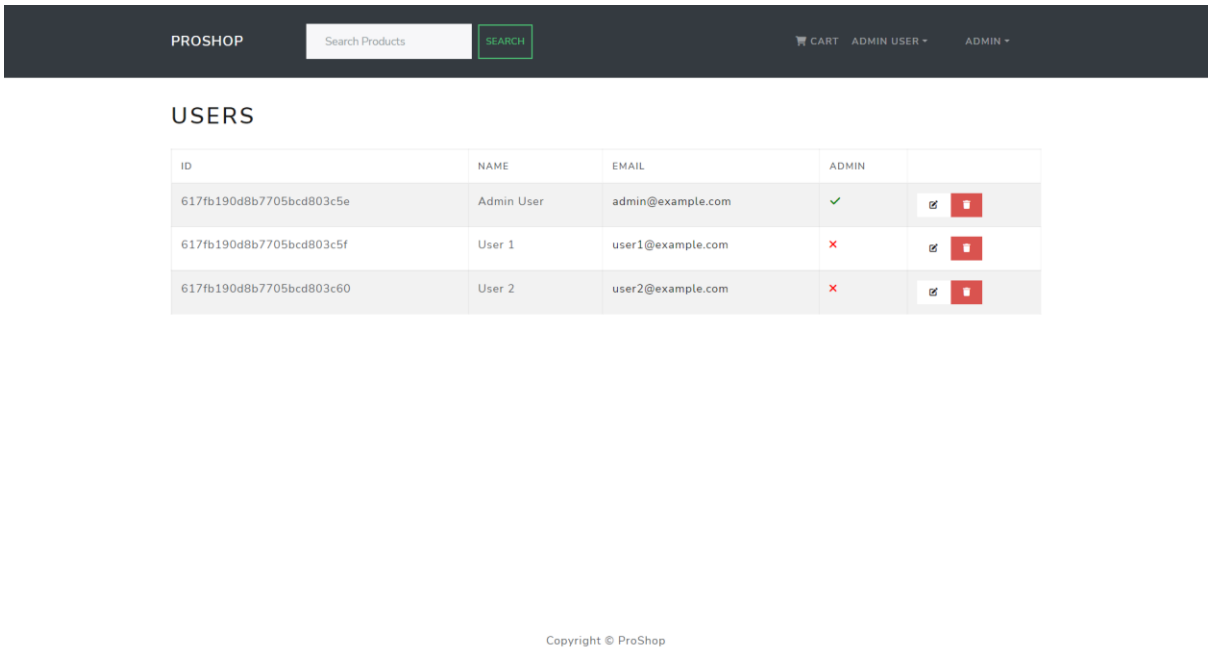
6.11 Order Details Screen



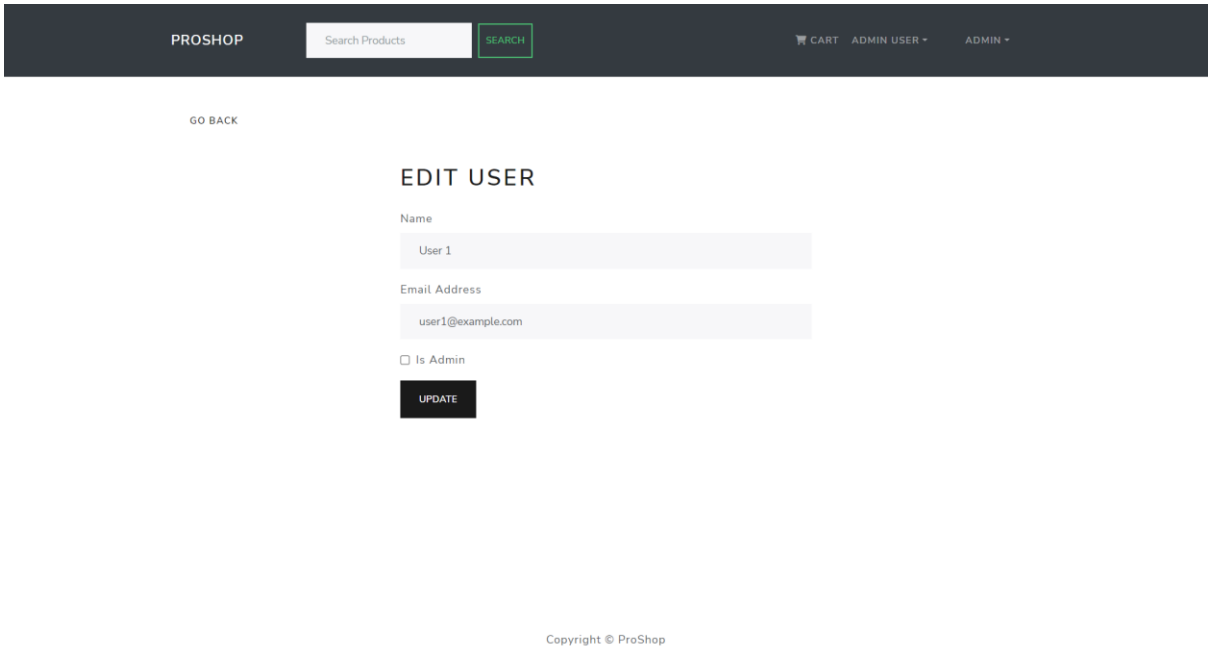
6.12 Admin Dashboard



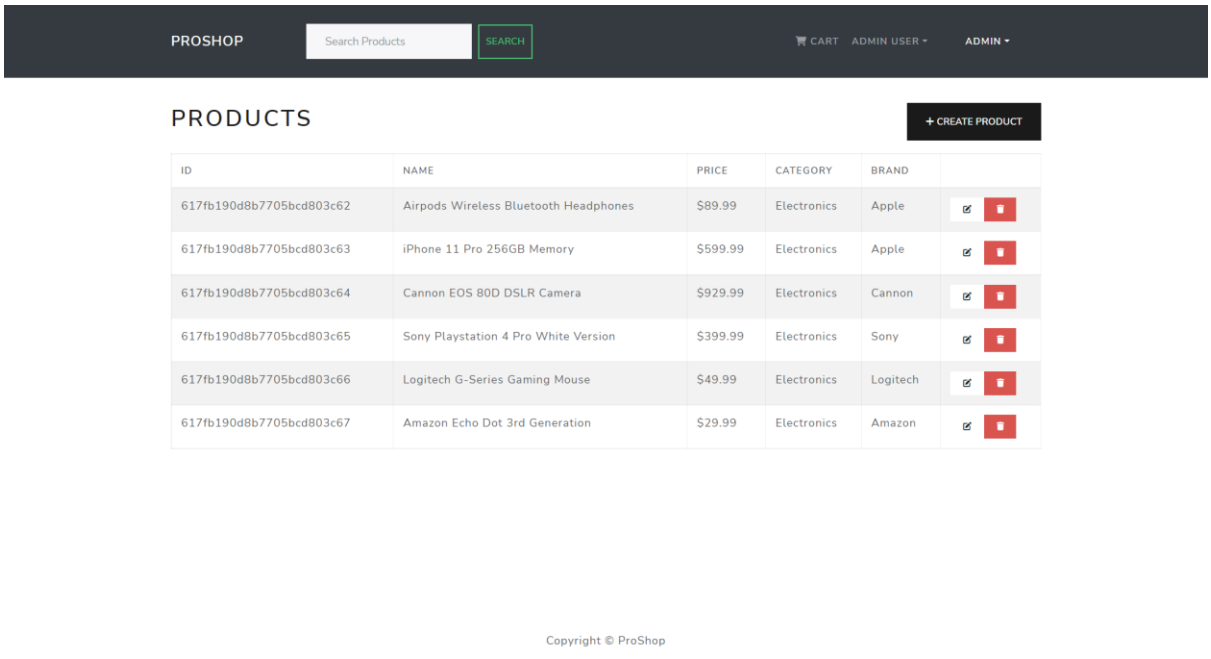
6.13 User List Screen



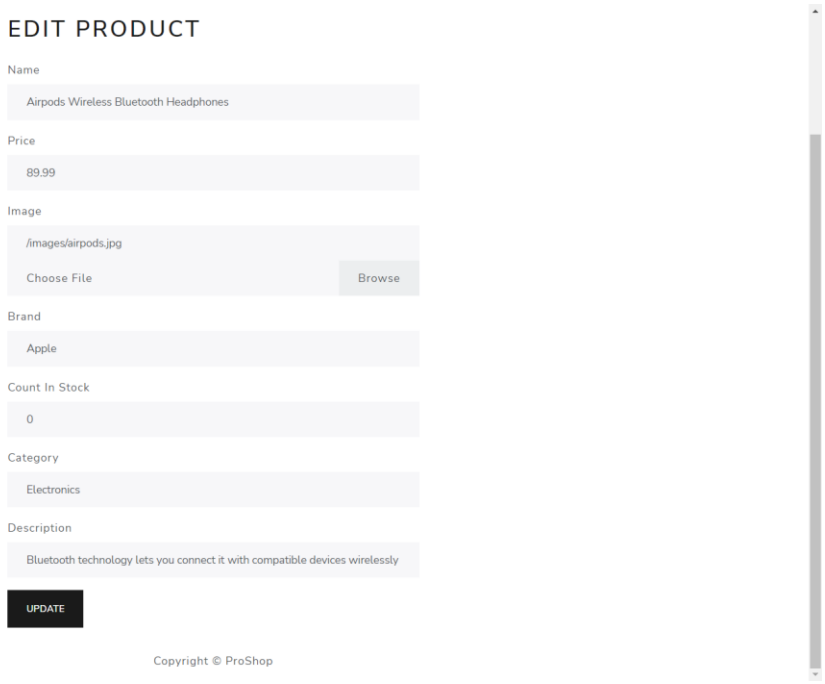
6.14 User Edit Screen



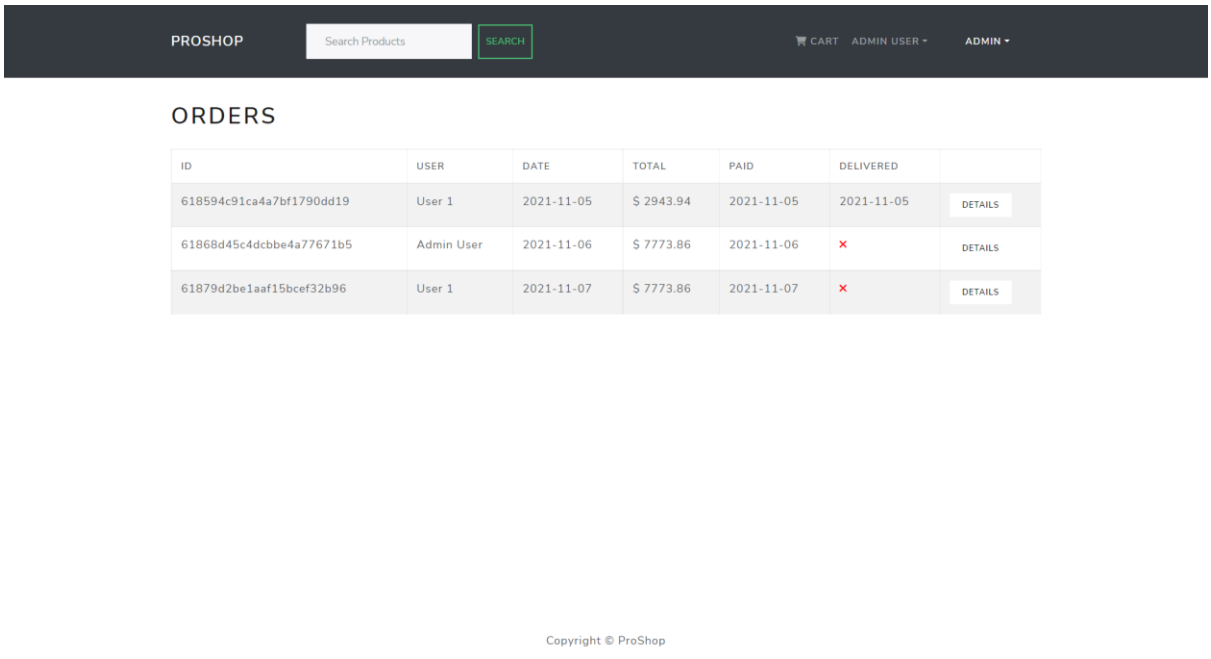
6.15 Product List Screen



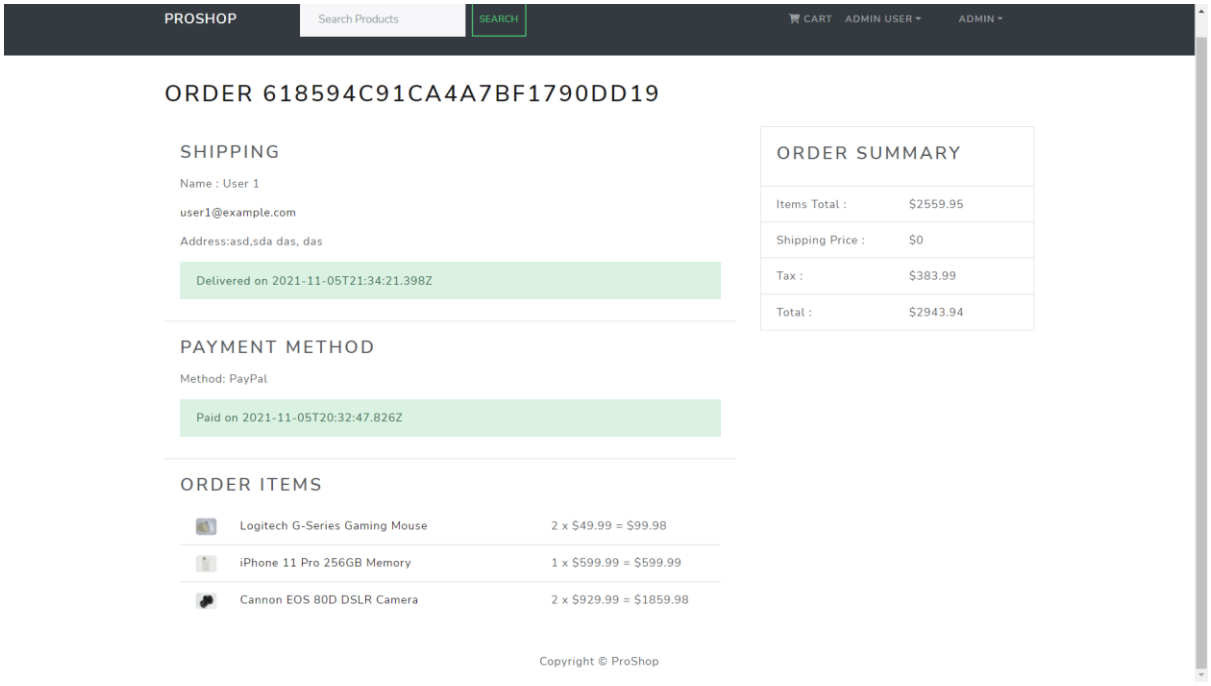
6.16 Product Edit Screen

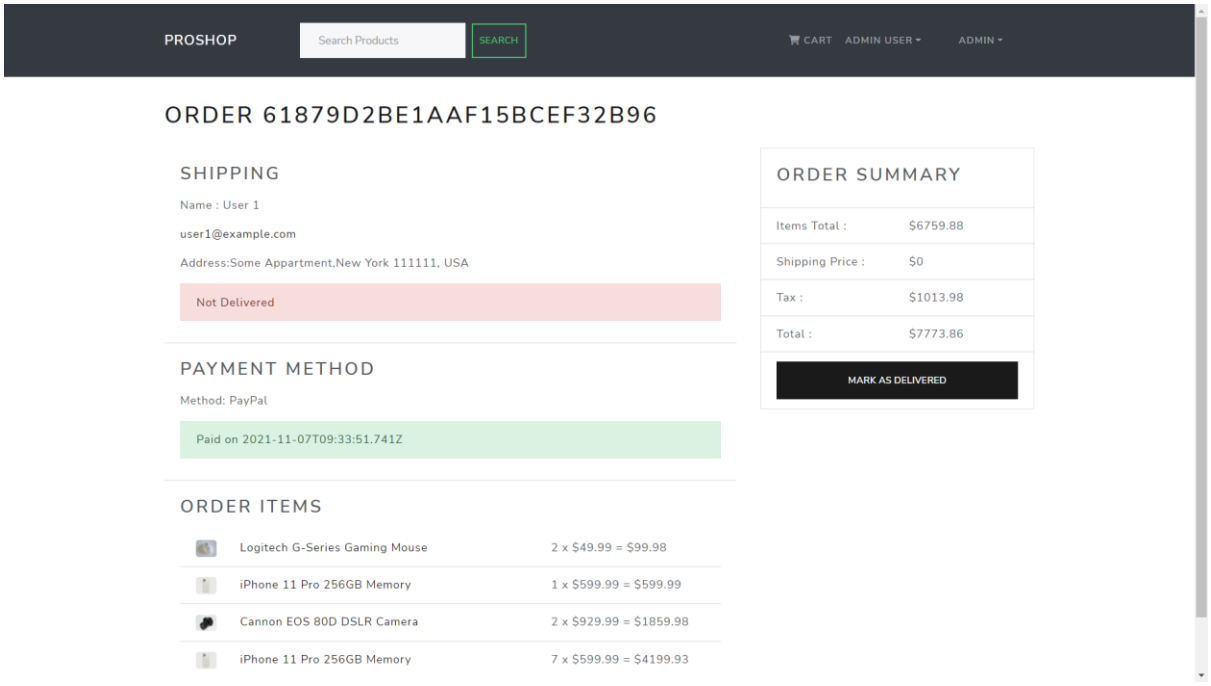


6.17 Order List Screen



6.18 Order Deliver Screen





Chapter Seven

Software Testing

7.1 Why Software Testing is Needed

Tool-bars work properly? Are all menu function and pull-down sub function properly listed? Is it possible to invoke each menu function using a logical assumption that if all parts of the system are correct, the goal will be successfully achieved? In adequate testing or non-testing will leads to errors that may appear few months later. Testing represents an interesting anomaly for the software engineer. During earlier software engineering activities, the engineer attempts to build software from an abstract concept to a tangible product. Now comes testing. The engineer creates a series of test cases that are intended to "demolish" the software that has been built. In fact, testing is the one step in the software process that could be viewed (psychologically, at least) as destructive rather than constructive. Testing requires that the developer discard preconceived notions of the "correctness" of software just developed and overcome a conflict of interest that occurs when errors are uncovered.

If testing is conducted successfully (according to the objectives stated previously) it will uncover errors in the software. As a secondary benefit, testing demonstrates that software functions appear to be working according to specification, that behavioural and performance requirements appear to have been met. In addition, data collected as testing is conducted provide a good indication of software reliability and some indication of software quality as a whole. But testing cannot show the absence of errors and defects, it can show only that software errors and defects are present. It is important to keep this (rather gloomy) statement in mind as testing is being conducted.

7.2 Testing Strategy

There are types of testing that we implement. They are as follows:

While deciding on the focus of testing activities, study project priorities. For example, for an online system, pay more attention to response time. Spend more time on the features used frequently. Decide on the effort required for testing based on the usage of the system. If the system is to be used by a large number of users, evaluate the impact on users due to a system failure before deciding on the effort.

This creates two problems

- ✓ Time delay between the cause and appearance of the problem.
- ✓ The effect of the system errors on files and records within the system.

The purpose of the system testing is to consider all the likely variations to which it will be suggested and push the systems to limits. The testing process focuses on the logical intervals of the software ensuring that all statements have been tested and on functional interval is conducting tests to uncover errors and ensure that defined input will produce actual results that agree with the required results. Program level testing, modules level testing integrated and carried out.

There are two major types of testing they are:

- ✓ White Box Testing.
- ✓ Black Box Testing.

7.3 White Box Testing

White box sometimes called "Glass box testing" is a test case design uses the control structure of the procedural design to drive test case. Using white box testing methods, the following tests were made on the system

a) All independent paths within a module have been exercised once. In our system, ensuring that case was selected and executed checked all case structures. The bugs that were prevailing in some part of the code where fixed.
b) All logical decisions were checked for the truth and falsity of the values.

7.4 Black Box Testing

Black box testing focuses on the functional requirements of the software. This is black box testing enables the software engineering to derive a set of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box testing rather it is complementary approach that is likely to uncover a different class of errors that white box methods like.

- ✓ Interface errors.
- ✓ Performance in data structure.
- ✓ Performance errors.
- ✓ Initializing and termination errors

Chapter Eight

Conclusion & Future Enhancement

8.1 Conclusion

This project is only a humble venture to satisfy the needs in a shop. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the organization. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

This website provides a computerized version of shop manipulate system which will benefit the users as well as the visitor of the shop. It makes entire process online where users can search product, and buy various product. It also has a facility for common user by login into the system where user can login and can see status of ordered. It provides the facility of admin's features where admins can add various item, review users' and send orders for delivery.

8.2 Future aspect

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner.

The following are the future scope for the project.

- ✓ Add more payment gateways (PayPal already added)
- ✓ Sales and Discount Offers can be added
- ✓ Bulk purchase discount can be added
- ✓ Filter using Brand, Category, Price can be added
- ✓ Multilingual can be added to this site
- ✓ And many features can be added this project to make it more robust

Bibliography

- Certificate: "*https://www.udemy.com/"*
- Flowcharts: *https://www.app.creatly.com*
- https://www.shareitsolutions.com/blog/mern-technology-stack
- https://www.oreilly.com/live-events/beginning-frontend-development-with-react/0636920189800/0636920246619/
- https://www.educba.com/what-is-expressjs/
- https://www.besanttechnologies.com/what-is-expressjs
- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction
- *"www.slideshare.com"*
- NPM packages and related information: *"https://www.npmjs.com/"*