

MINI PROJECT (2020-21)

TRUE CART

REPORT



Institute of Engineering & Technology

Submitted by-
Ayush Gupta
(181500172)
Kshitij Gupta
(181500337)

Supervised By

Dr. Anand Parkash Gupta, Technical Trainer
Mrs. Ruchi Gupta, Assistant Professor

Department of Computer Engineering & Applications



Department of Computer Engineering and Applications GLA University, 17 km. Stone NH#2, Mathura-Delhi Road, Chaumuha, Mathura – 281406 U.P (India)

DECLARATION

We hereby declare that the work which is being presented in the Mini Project “**TrueCart**”, in fulfillment of the requirements for Mini Project in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of our own work carried under the supervision of Dr. Anand Prakash Gupta, Technical Trainier and Mrs. Ruchi Gupta, Assistant professor.

Name of Candidate: Kshitij Gupta

Name of Candidate: Ayush Gupta

University Roll No.: 181500337

University Roll No.: 181500172



Department of Computer Engineering and Applications GLA University, 17 km. Stone NH#2, Mathura-Delhi Road, Chaumuha, Mathura – 281406 U.P (India)

CERTIFICATE

This is to certify that the project entitled “**TrueCart**” carried out in Mini Project – I is a bonafide work done by

Ayush Gupta (181500172)

Kshitij Gupta (181500337)

And is submitted in fulfillment of the requirements for the award of the Bachelor of Technology (Computer Science and Engineering).

Signature of Supervisor:

Name of Supervisors: Dr. Anand Parkash Gupta (Technical Trainer) and Mrs. Ruchi Gupta (Assistant Professor)

Date:

Acknowledgement

It gives us a great sense of pleasure to present the report of the B. Tech Mini Project undertaken during B. Tech. Third Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

Our heartiest thanks to Dr. (Prof). Anand Singh Jalal, Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.

We owe a special debt of gratitude to Dr. Anand Parkash Gupta (Technical Trainer) and Mrs. Ruchi Gupta (Assistant Professor)

for their constant support and guidance throughout the course of our work. Their sincerity, thoroughness and perseverance have been a constant source of inspiration for us. They have showered us with all his extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.

We also do not like to miss the opportunity to acknowledge the contribution of all instructors who are available on YouTube and Stackoverflow. I would like to thank all my friends who helped me in making this project.

Last but not the least, I would like to express our deep sense of gratitude and earnest thanks giving to our dear parents for their moral support and heartfelt cooperation during the project.

Ayush Gupta

Kshitij Gupta

Table of Contents

Declaration	ii
Certificate	iii
Acknowledgments	iv
Abstract	v
Table of Contents	vi
1. Introduction	7
1.1 General Introduction to the topic	7
1.2 Problem Definition & Objective	8
2. Software Requirement Analysis	8
3. Software Design	13
4. Implementation	15
4.1 Implementation Details	15
4.2 Some Screenshots	18
5. Testing	27
6. Appendices	29
7. References	32

ABSTRACT

Building an E-Commerce (Electronic commerce) Web Application for the users who want to search the product or buy the different products online. In this application users can search the different products, filter the products, check the product in stock / out of stock and create cart after signup / sign in.

We have divided our project into four modules –

Fist Module –

In first module, we have used JavaScript, Node JS, and Express JS as a framework for Node JS. We have also used Postman which is a development tool. It enables us to test calls for API's and simplifies each step of building API and streamline collaboration.

By using these technologies, we are covering database setup, user signup or sign in, apply validations, creation of tokens, creating categories and CRUD (create, read, update and delete) operations on products.

Second Module --

In second module, we have created a user interface for the user to ease user's interaction with our application and for this we have used HTML-5 (Hypertext Markup Language), React JS which is a JavaScript library. This interface is split up into components, instead of having one huge page you break it up into smaller pieces known as components. .

We have our home page in such a manner so that users will render each functionality of our web application. They can see the available products, filter / search the products, create their cart, add the products into cart and many more.

Third Module --

In third module, we have worked on our database part and with the help of MongoDB we have stored all the data. MongoDB is a highly scalable, flexible, and distributed NoSQL database. With this database we will define some schema's and apply some validations for the data. After combining all three modules we will be able to make a user-friendly web application.

Fourth Module --

In fourth module, we have deployed our application over the internet so that user can use this application from anywhere in the world.

This application that we have built using front-end as well as back-end web technologies and we have provided a user-friendly interface so that users get never stuck.

INTRODUCTION

1.1 General Introduction to the topic

An Ecommerce website is an information technology method in which traders, businesses/distributors/marketers can sell products/services and the customer can purchase on that website electronically by using the internet on the mobile and computer. It means an e-commerce website is an online shop. e means electronic. The website means the group of HTML web pages and that is created to market/sell information/products/services.

From a bigger perspective, every website on the internet is an ecommerce Website. It can be the platform, it can be a marketplace, it can be a portal, it can be apps, it can be an entertainment website, shopping website, online courses website, and online degree college.

Building an E-Commerce (Electronic commerce) Web Application for the users who want to search the product or buy the different products online. In this application users can search the different products, filter the products, check the product in stock / out of stock and create cart after signup / sign in. By creating cart they are able to add the items, remove the items, see their total price and check the status of the order. This application will be build using front-end as well as back-end web technologies. We will provide a user-friendly interface so that users get never stuck.

PROBLEM DEFINITION

E-commerce provides an easy way to sell products to a large customer base. However, there is a lot of competition among multiple e-commerce sites. When users land on an e-commerce site, they expect to find what they are looking for quickly and easily. Also, users are not sure about the brands or the actual products they want to purchase. They have a very broad idea about what they want to buy. Many customers nowadays search for their products on Google rather than visiting specific e-commerce sites.

The purpose of any e-commerce website is to help customers narrow down their broad ideas and enable them to finalize the products they want to purchase. For example, suppose a customer is interested in purchasing a mobile. His or her search for a mobile should list mobile brands, operating systems on mobiles, screen size of mobiles, and all other features as facets. As the customer selects more and more features or options from the facets provided, the search narrows down to a small list of mobiles that suit his or her choice. If the list is small enough and the customer likes one of the mobiles listed, he or she will make the purchase.

OBJECTIVE

Although, there are many existing web applications are available but we are trying to reach the customers who are not able to buy the things online because of the unavailability of the product when they want, there is a high possibility that the product is not deliverable in their area or they lost their trust in online purchasing because of some providers who only rob them. And we can also use this project to run the local businesses online that will help in the growth and development of the person as well as for the country also. We are providing original and genuine products to the users with the best price

SOFTWARE AND REQUIREMENT ANALYSIS

2.1 Software Analysis

VS Code-

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI.

MongoDB Compass-

MongoDB Compass is a GUI to explore, analyze, and interact with the content stored in a MongoDB database without knowing or using queries.

MongoDB Compass is a much better alternative for the Mongo shell. Compass can carry out all the operations that Mongo Shell does and more, including:

- Visualize and explore data stored in your database.
- Create databases and Insert, update, and delete data in your database.
- Get immediate real-time server statistics.
- Understand performance issues with visual explain plans.
- Manage your indexes.
- Validate your data with JSON schema validation rules.
- Extendable via plugins.

2.2 Language and Frameworks Requirements

➤ HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web Browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML Elements are the building blocks of HTML pages.

➤ CSS (Cascading Style Sheets):

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other effects. CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

➤ **Bootstrap:**

Bootstrap is a free and open front-end framework for designing websites and web applications. It contains HTML - and CSS -based design templates for typography, forms, buttons, and navigation and other interface components, as well as optional JavaScript extensions. Unlike many earlier web frameworks, it concerns itself with front end development only. Bootstrap is the second most-starred project on GitHub, with more than 129,000 stars. Bootstrap comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels.

➤ **JavaScript:**

JavaScript, often abbreviated as JS, is a high-level, interpreted programming language that conforms to the ECMAScript specification. It is a programming language that is characterized as dynamic, weakly typed, prototype-based and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it, and major web browsers have a dedicated JavaScript engine to execute it.

➤ **React JS:**

React is a JavaScript library for building user interfaces. React is a JavaScript library and React applications built on it run in the browser, NOT on the server. Applications of this kind only communicate with the server when necessary, which makes them very fast compared to traditional websites that force the user to wait for the server to re-render entire pages and send them to the browser.

React is used for building user interfaces - what the user sees on their screen and interacts with to use your web app. This interface is split up into components, instead of having one huge page you break it up into smaller pieces known as components. In more general terms, this approach is called Modularity.

➤ **Node JS and Express JS:**

Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside of a browser. You need to remember that Node JS is not a framework and it's not a programming language. Most of the people are confused and understand it's a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It's used in production by large companies such as PayPal, Uber, Netflix, and Walmart and so on. Express is a small framework that sits on top of Node.js web server functionality to simplify its APIs and add helpful new features. It makes it easier to organize your application's functionality with middle ware and routing; it adds helpful utilities to Node.js HTTP objects; it facilitates the rendering of dynamic HTTP objects. Express is a part of MEAN stack, a full stack JavaScript solution used in building fast, robust, and maintainable production web applications.

➤ **MongoDB:**

MongoDB, the most popular NoSQL database, is an open-source document-oriented database. The term 'NoSQL' means 'non-relational'. It means that MongoDB isn't based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data. This format of storage is called JSON format. SQL databases store data in tabular format. This data is stored in a predefined data model which is not very much flexible for today's real-world highly growing applications. Modern applications are more networked, social and interactive than ever. Applications are storing more and more data and are accessing it at higher rates. Relational Database Management System (RDBMS) is not the correct choice when it comes to handling big data by the virtue of their design since they are not horizontally scalable. If the database runs on a single server, then it will reach a scaling limit. NoSQL databases are more scalable and provide superior performance. MongoDB is such a NoSQL database that scales by adding more and more servers and increases productivity with its flexible document model.

2.2 Software and Hardware Requirement

1. Hardware Requirements:

- Processor: Minimum 1 GHz; Recommended 2GHz or more.
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi).
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more.
- Memory (RAM): Minimum 4 GB; Recommended 8 GB or above.

2. Software Requirements:

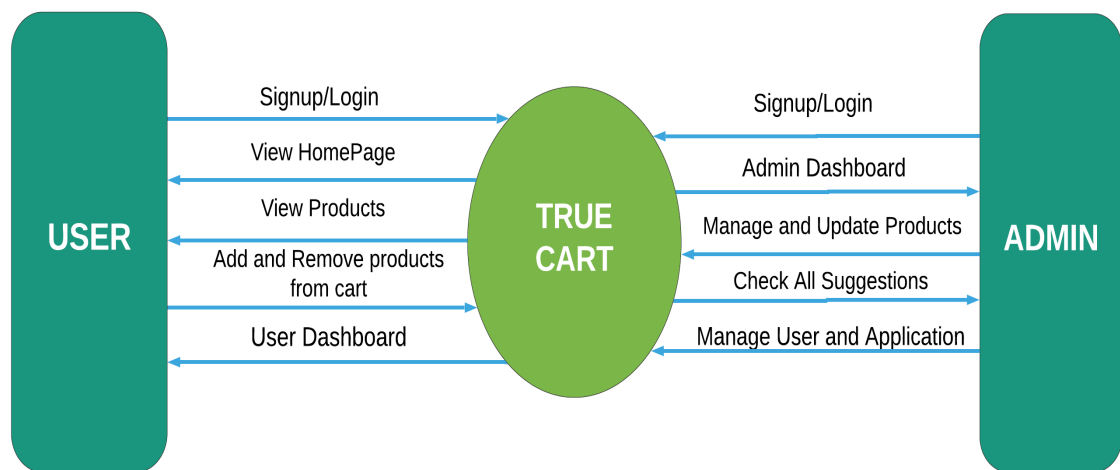
- Windows Operating System.
- Integrated Development Environment (IDE) -Visual Studio Code 1.47 / Atom Editor.
- GitHub Desktop 2.5.3.
- MongoDB Compass.

3. Language and Framework Requirements:

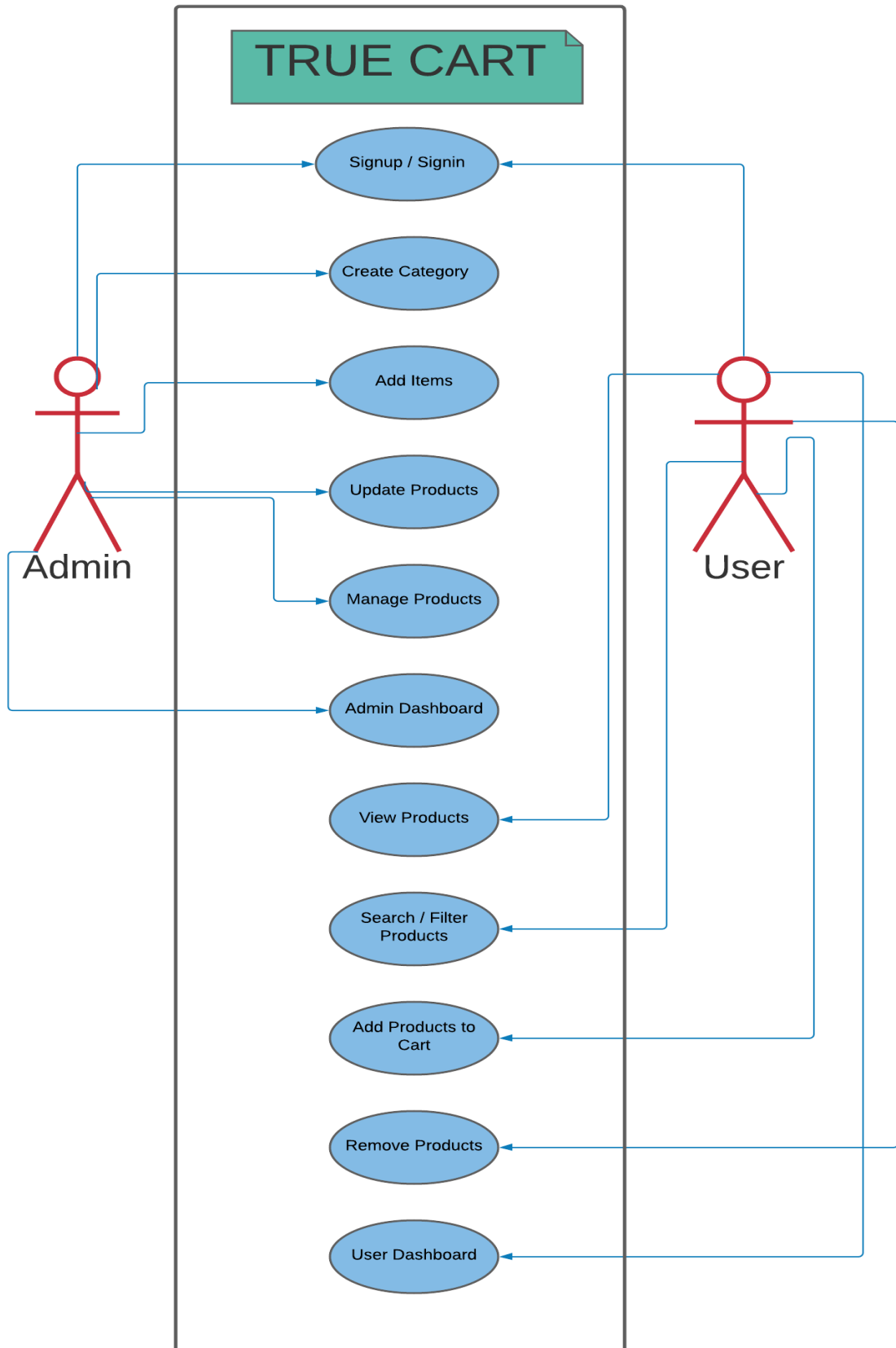
- HTML 5
- CSS 3
- Bootstrap 4 / 5
- JavaScript
- Node JS 12.18.3
- Express JS 4.17.1
- MongoDB 4.2.8
- React JS

SOFTWARE DESIGN

3.1 DATA FLOW DIAGRAM



3.2 USE CASE DIAGRAM



IMPLEMENTATIONS DETATILS

We have divided our project into four modules:

- a. Building API
- b. User Interface
- c. Back-end
- d. Deploying our application

Part (i):

We have started our project with the first module i.e. API. In this module, we are using JavaScript, Node JS, and Express JS as a framework for Node JS (It is used for easier creation of web applications and services). We are using Postman which is a development tool. It enables us to test calls to API's and simplifies each step of building API and streamline collaboration so that we can create better API's.

By using these technologies, we are covering database setup, user signup or sign in, apply validations, creation of tokens, creating categories and CRUD (create, read, update and delete) operations on products. Basically, first we are making an API to do all these stuffs then we are going to provide all these functionalities to the user.

Part (ii):

In this part, we are going to use HTML-5 (Hypertext Markup Language), React JS which is a JavaScript library for building user interfaces - [what the user sees on their screen and interacts with to use your web app](#).

This interface is split up into components, instead of having one huge page you break it up into smaller pieces known as components. In more general terms, this approach is called Modularity, CSS-3 (Cascading Style Sheets) for describing the presentation of Web pages, including colors, layout, and many more, using bootstrap for designing purposes because Bootstrap is a potent front-end framework used to create modern websites and web apps. It's open-source and free to use, yet features numerous HTML and CSS templates for UI interface elements such as buttons and forms. Bootstrap also supports JavaScript extensions. With the help of bootstrap we can easily make our website that adjusts on mobile devices, tablets and desktops.

We will design registration page with the help of HTML, React JS, CSS and Bootstrap. In this page, users can easily register themselves by simply filling the required details. We have also

made a sign in page that will help users to log in if they are already registered to our web application. We will create restrictions for some features and those features will show only if the users are logged in.

Now, our homepage will help users to interact with our web application in a very easy and efficient manner. We will create our home page in such a manner so that users will render each functionality of our web application. They can see the available products, filter / search the products, create their cart, add the products into cart and many more.

Part (iii):

In the third part, we are considering database also. With the help of MongoDB we are going to store all data. MongoDB is an open-source document-based database management tool that stores data in JSON-like formats. It is a highly scalable, flexible, and distributed NoSQL database. With this database we will define some schema's and apply some validations for the data. After combining all three modules we will be able to make a user-friendly web application.

Part (iv):

In the fourth part, we have deployed our application over the internet so that user can use this application from anywhere in the world.

➤ Module 1:

It consists of:

- Database Setup
- User Signup
- User Sign in
- Validations
- User Sign in using JWT
- Creating Categories with help of Postman.
- CRUD operations on product with help of Postman.

➤ Module 2.

It consists of:

- Sign in / Sign up on client side.
- Creating Categories
- Creating Products
- Shop Page
- Filter / Search Products
- Cart Page

- Add products to cart
- Remove products from cart

➤ Module 3:
It consists of:

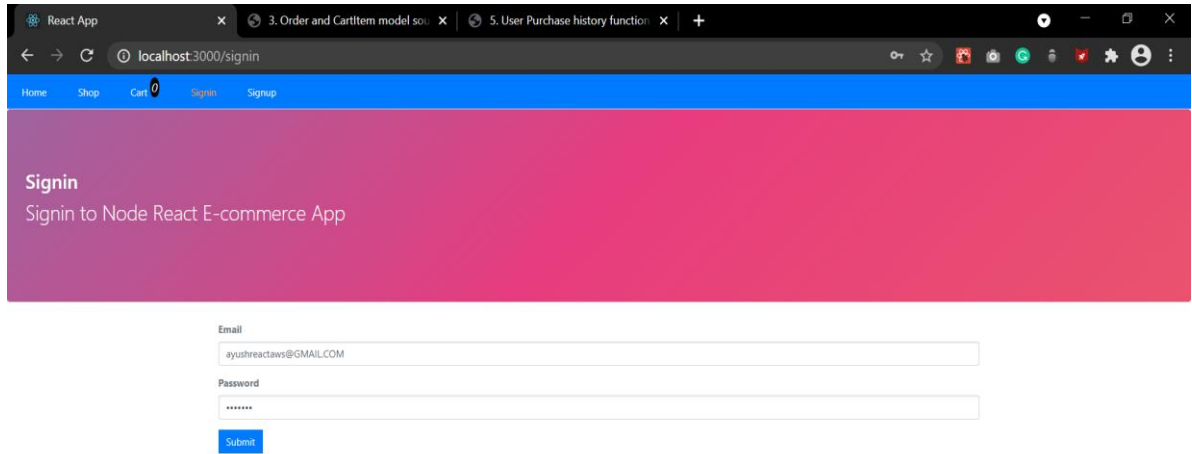
- All about databases.
- Save all the register and sign in page data in mongo DB database.
- Send order details to back-end.
- Save the orders.
- Profile Update.
- Product Update.

➤ Module 4:

- Deploy our application over the internet so that user can use this application from anywhere in the world.

Some Screenshots

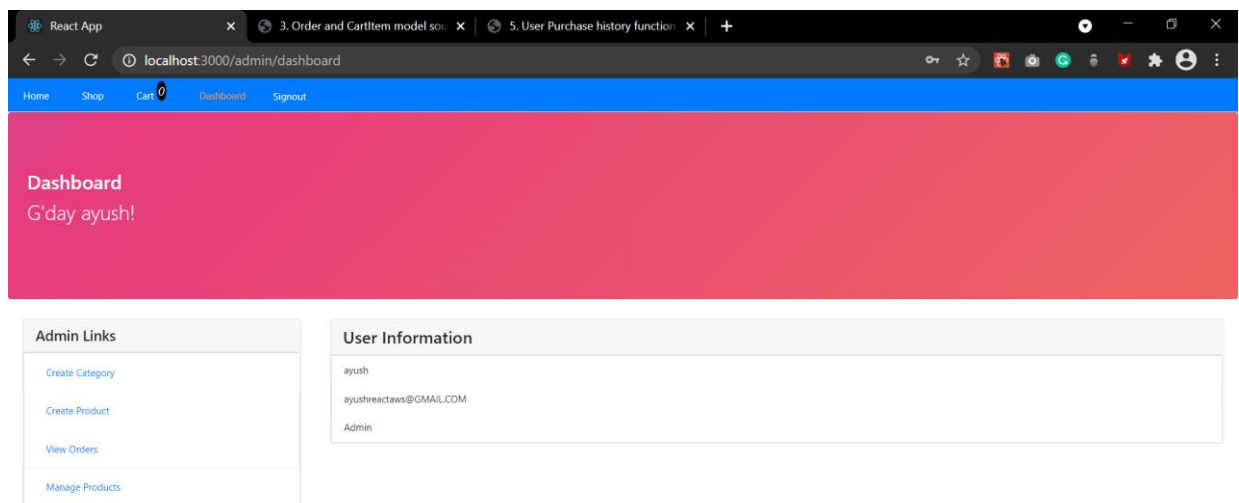
Admin Login



A screenshot of a web browser showing the Admin Login page. The browser has three tabs: 'React App', '3. Order and CartItem model sou', and '5. User Purchase history function'. The address bar shows 'localhost:3000/signin'. The page has a blue header with navigation links: 'Home', 'Shop', 'Cart 0', 'Signin', and 'Signup'. The main content area has a pink-to-purple gradient background. It features the text 'Signin' and 'Signin to Node React E-commerce App'. Below this is a form with two input fields: 'Email' (containing 'ayushreactaws@GMAIL.COM') and 'Password' (containing '*****'). A blue 'Submit' button is at the bottom of the form.

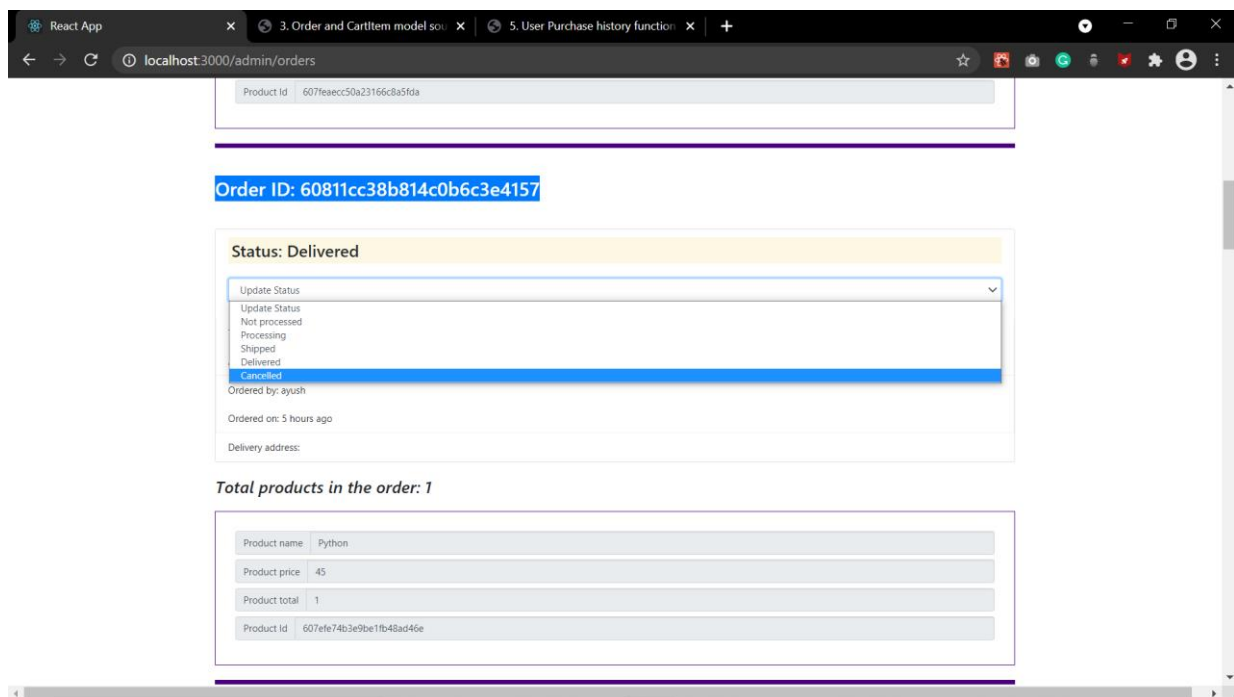
Admin Dashboard

Admin has different dashboard as he or she can create category, create product, view orders, manage products.

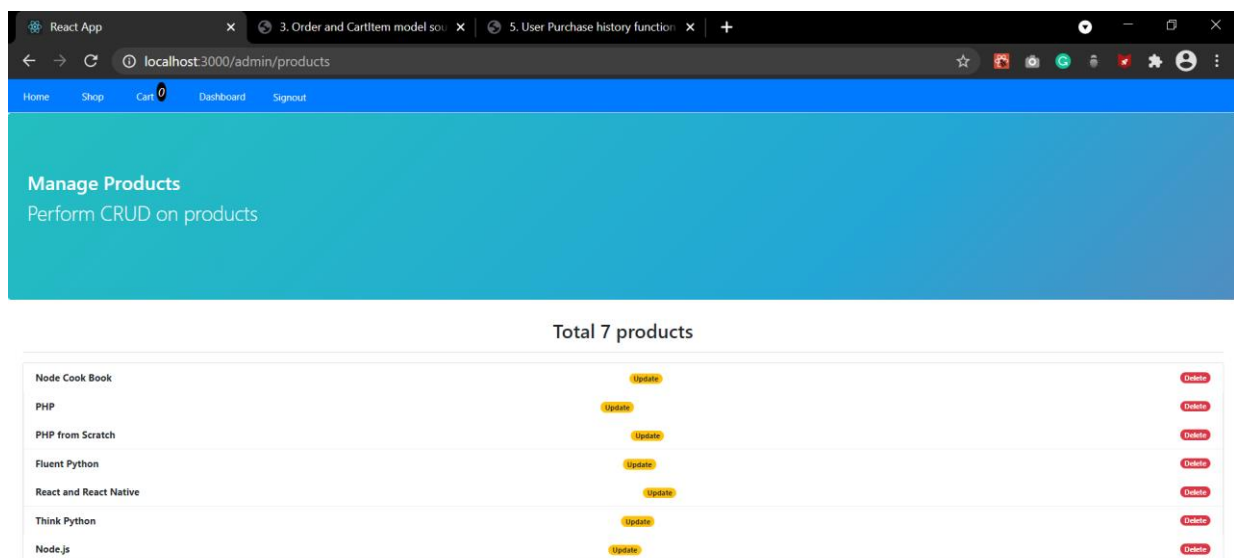


A screenshot of a web browser showing the Admin Dashboard. The browser has three tabs: 'React App', '3. Order and CartItem model sou', and '5. User Purchase history function'. The address bar shows 'localhost:3000/admin/dashboard'. The page has a blue header with navigation links: 'Home', 'Shop', 'Cart 0', 'Dashboard', and 'Signout'. The main content area has a pink-to-red gradient background. It features the text 'Dashboard' and 'G'day ayush!'. Below this are two white boxes. The left box is titled 'Admin Links' and contains four links: 'Create Category', 'Create Product', 'View Orders', and 'Manage Products'. The right box is titled 'User Information' and contains three lines of text: 'ayush', 'ayushreactaws@GMAIL.COM', and 'Admin'.

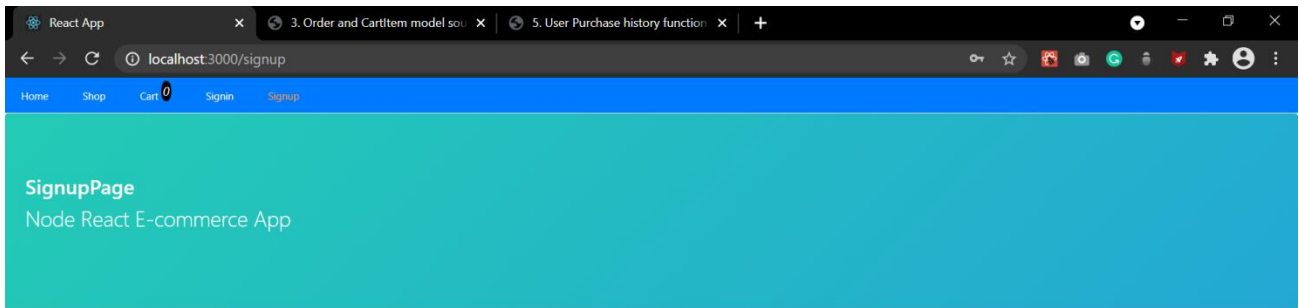
Admin can check users orders, purchase history, etc.



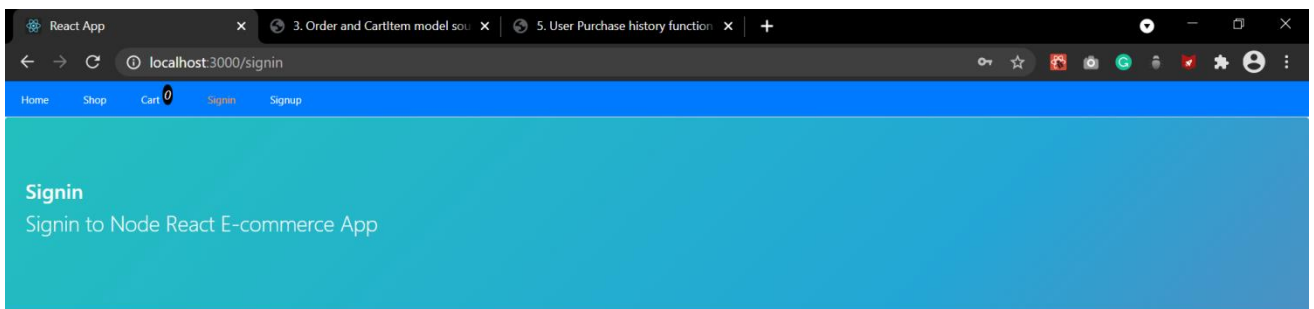
Admin can update, delete and add products.



New User Sign up

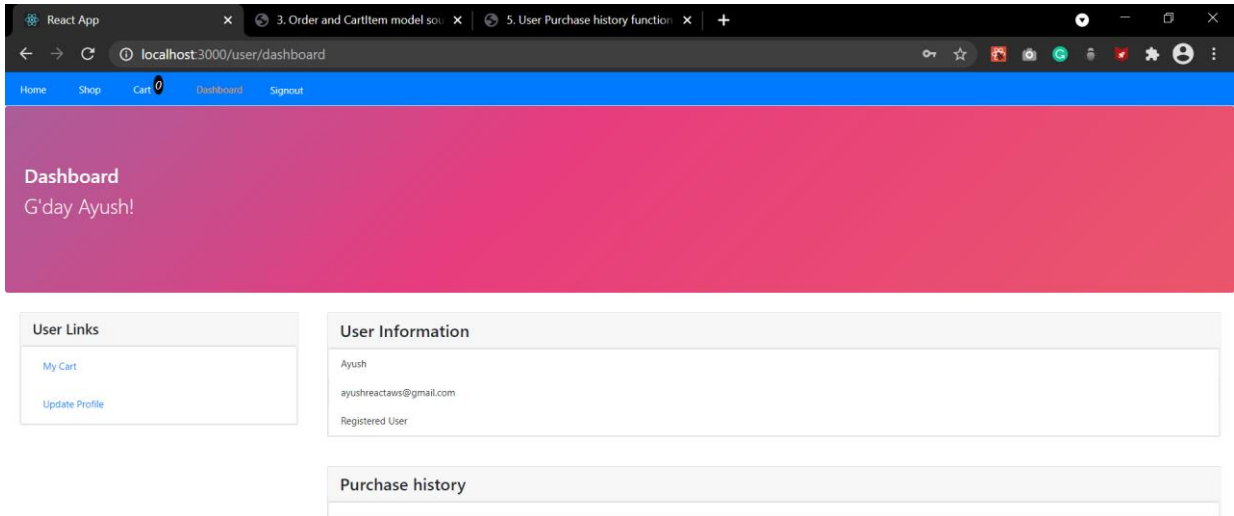


New User Sign in

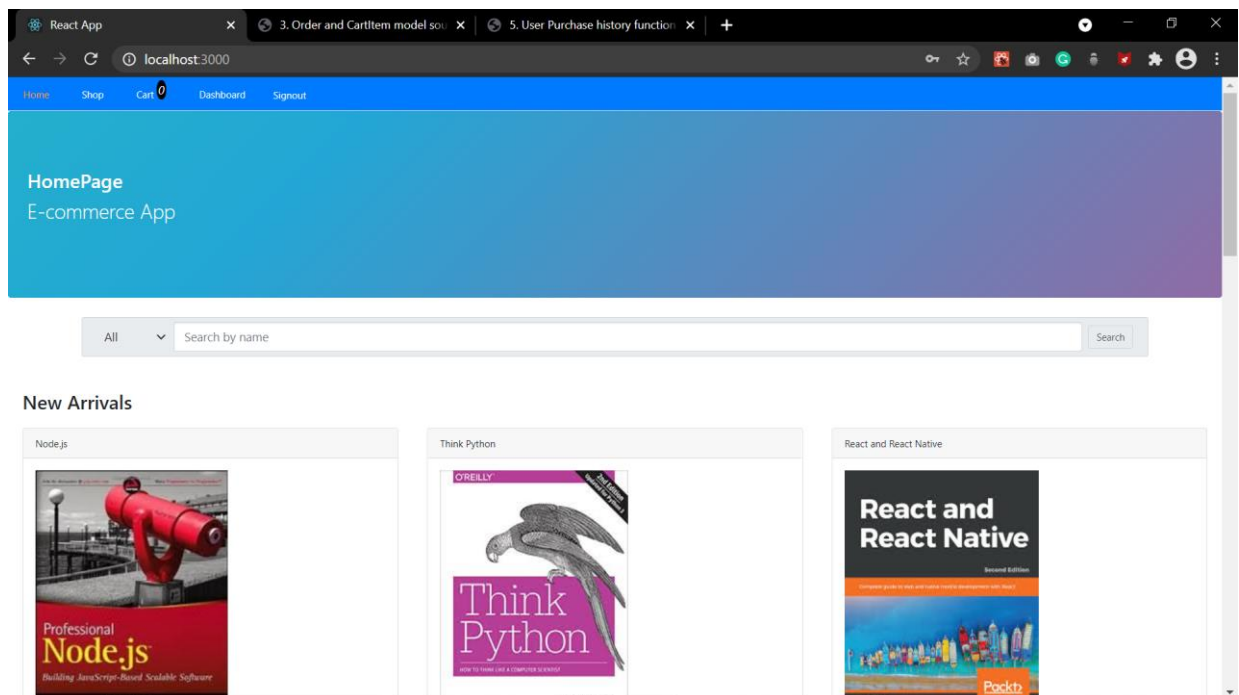


User Dashboard

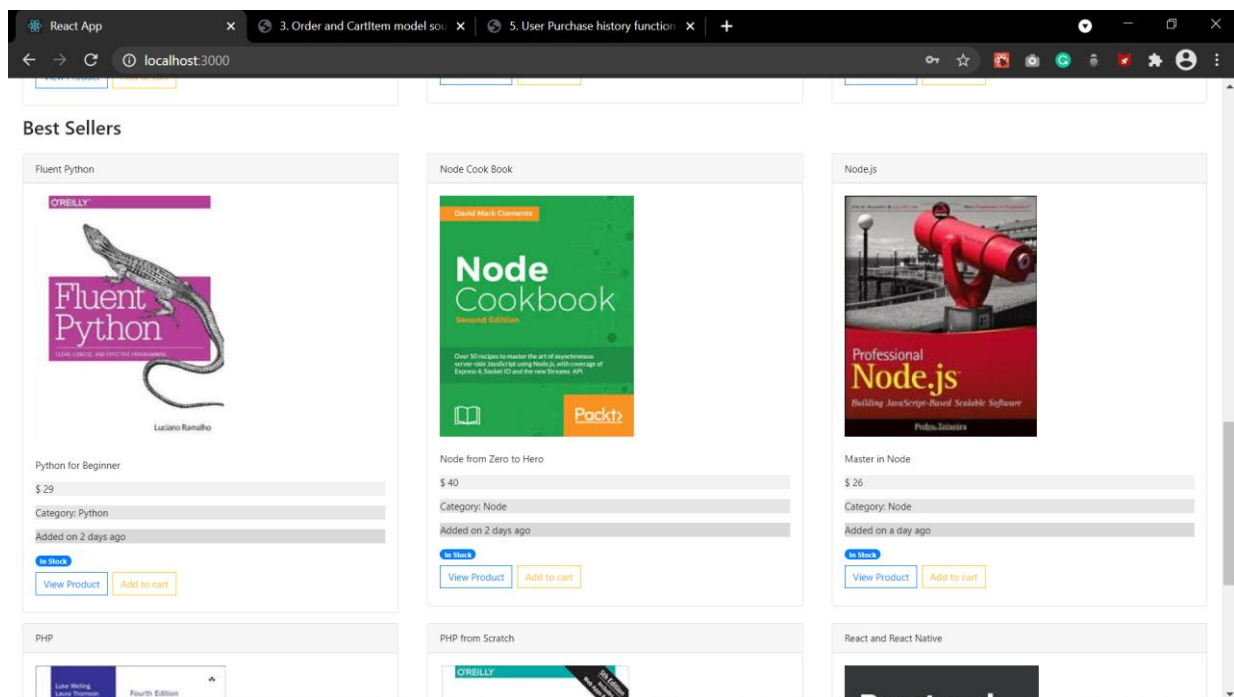
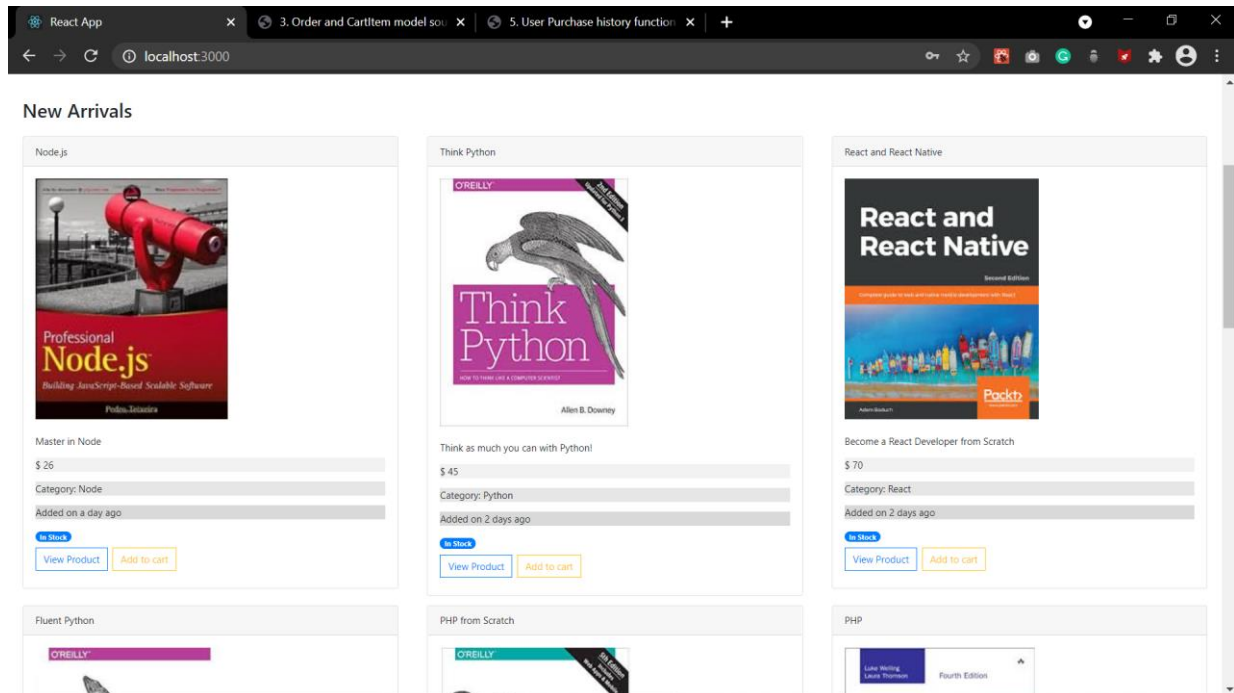
Here, user can see his or her purchase history and also update profile.



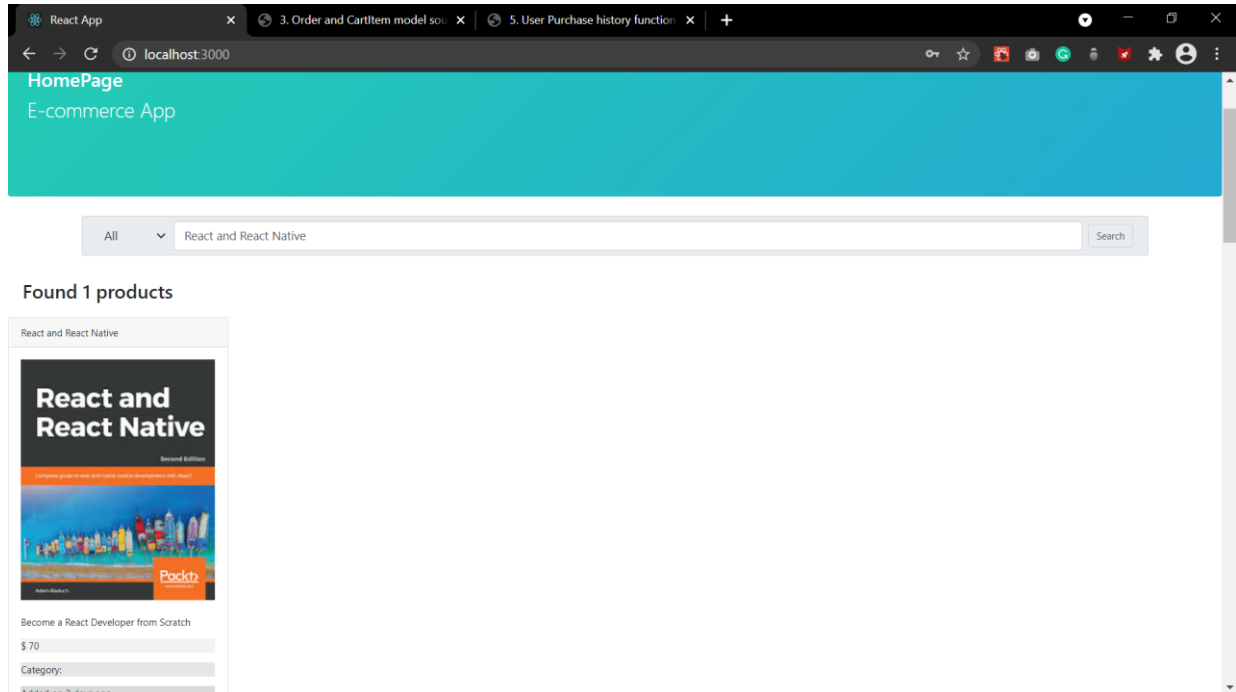
Homepage of our web application



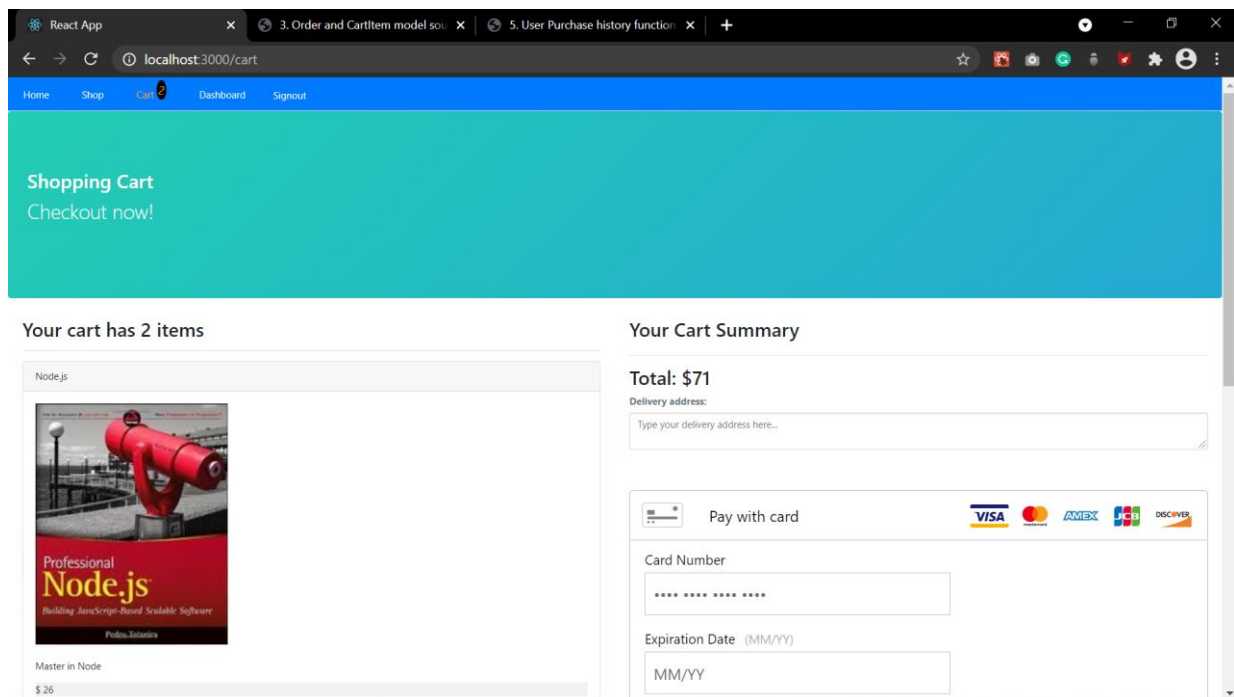
New Arrivals and Best Sellers on our WebApp



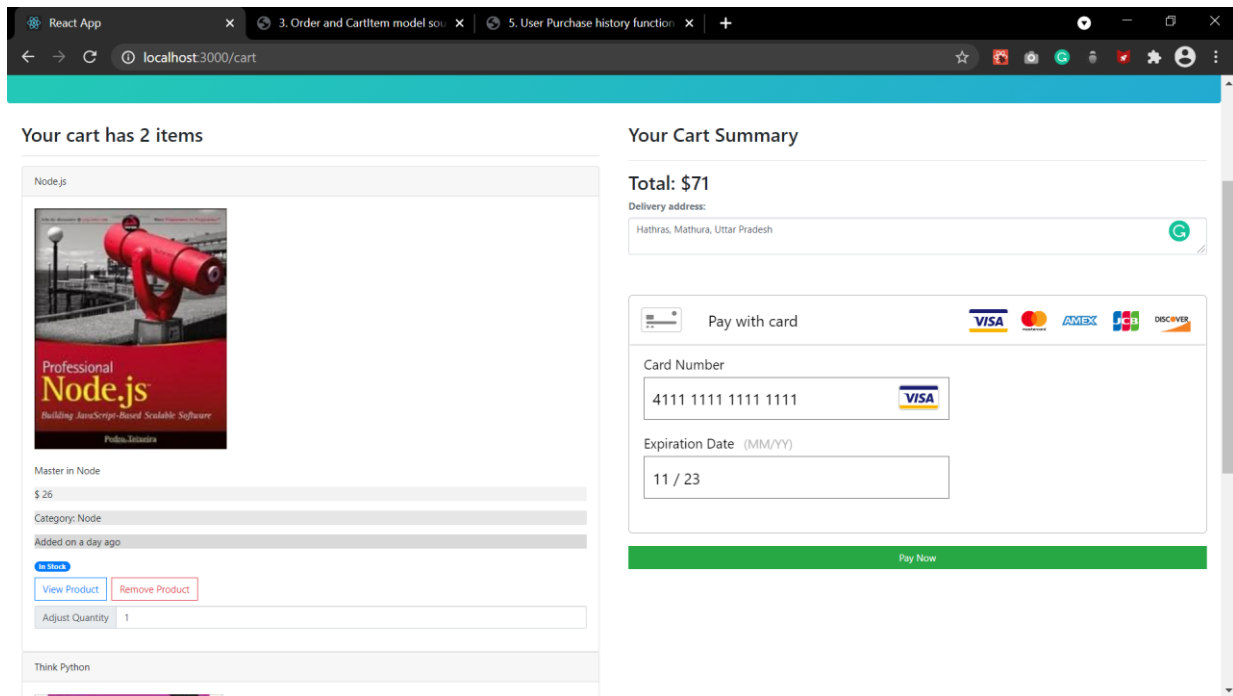
User can find products by searching it on the search bar



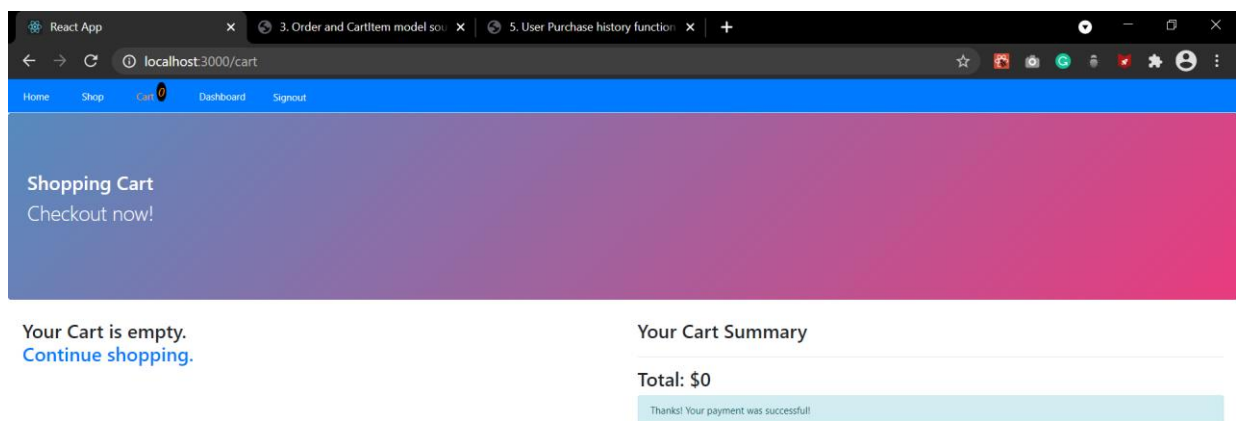
Adding product into cart



Payment gateway of our web application

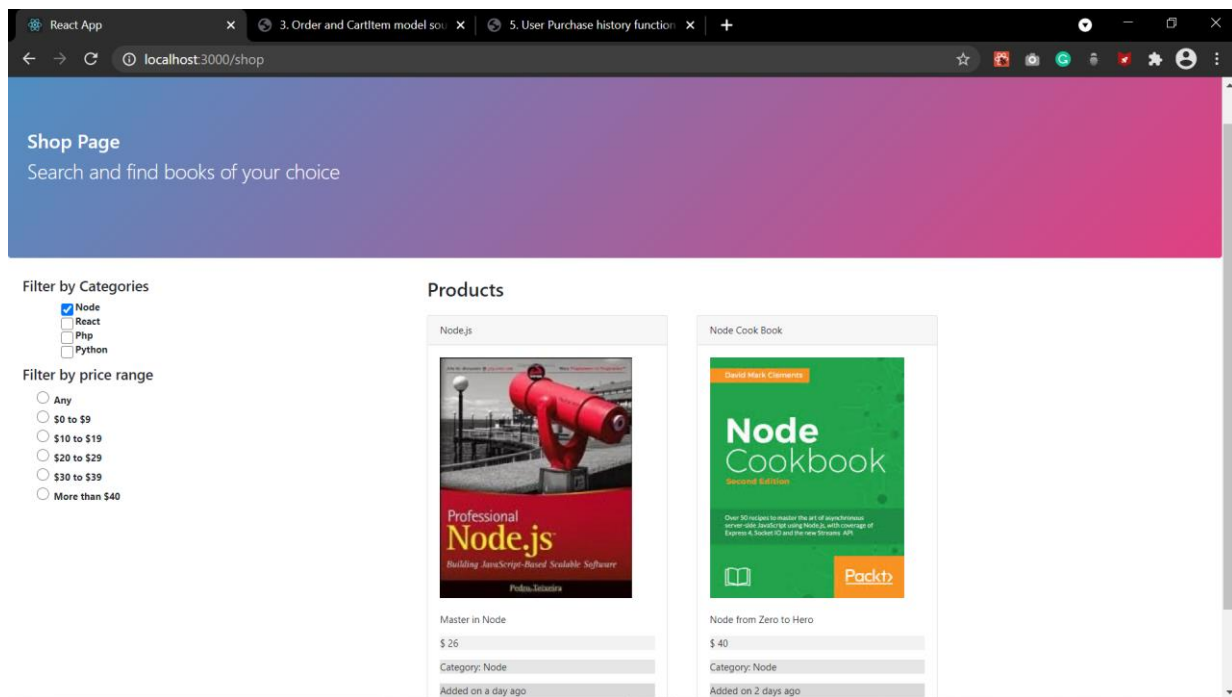


After completing the payment the cart size reduces to zero again.

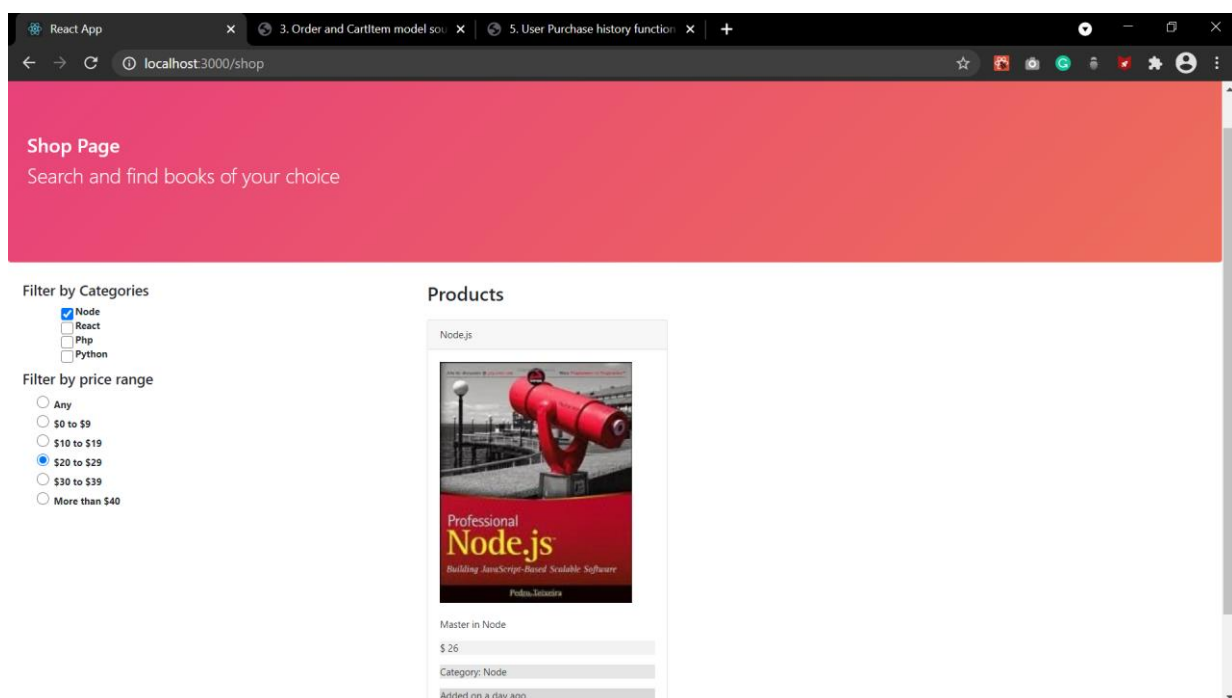


User can also search and find books of his or her choice:-

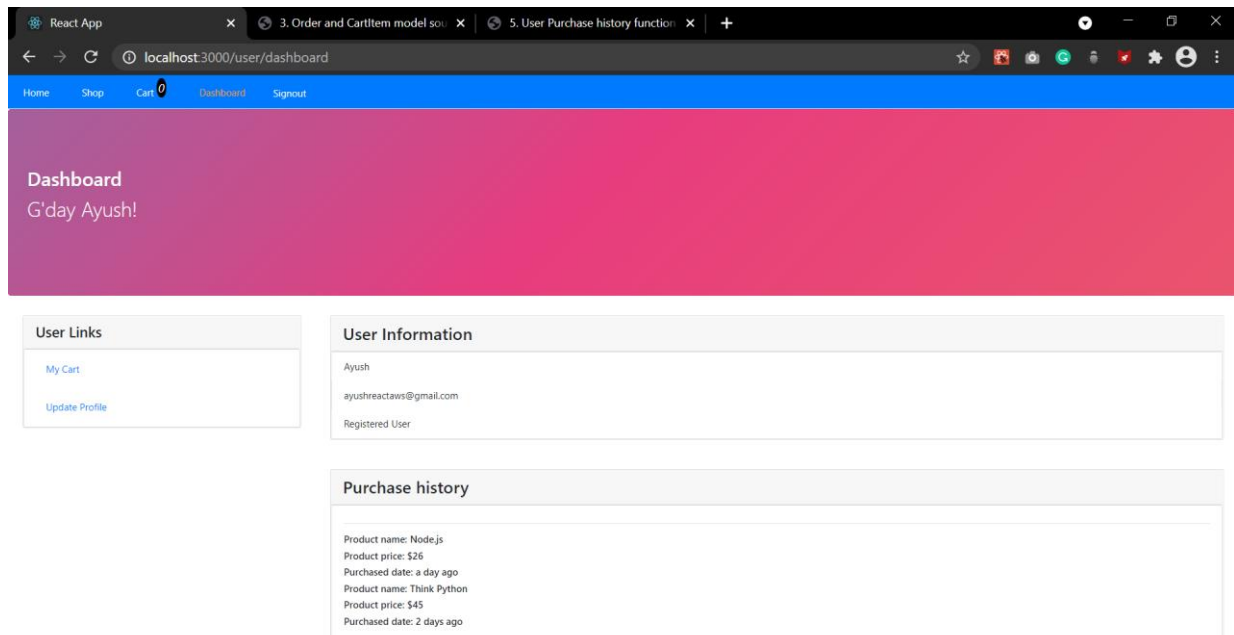
Filter by categories



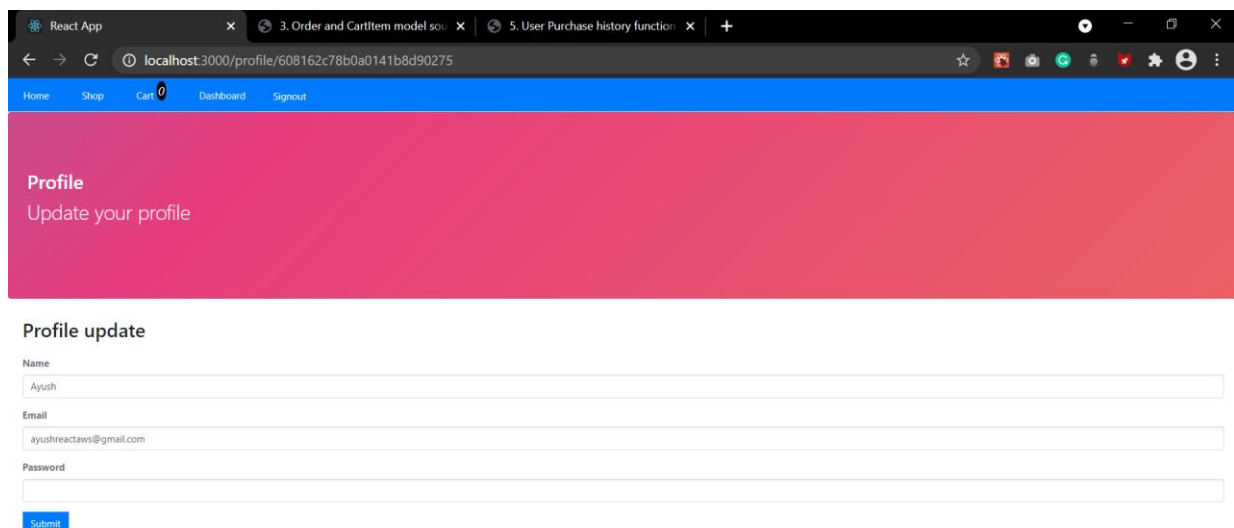
Filter by price range:



Users can also check their purchase history

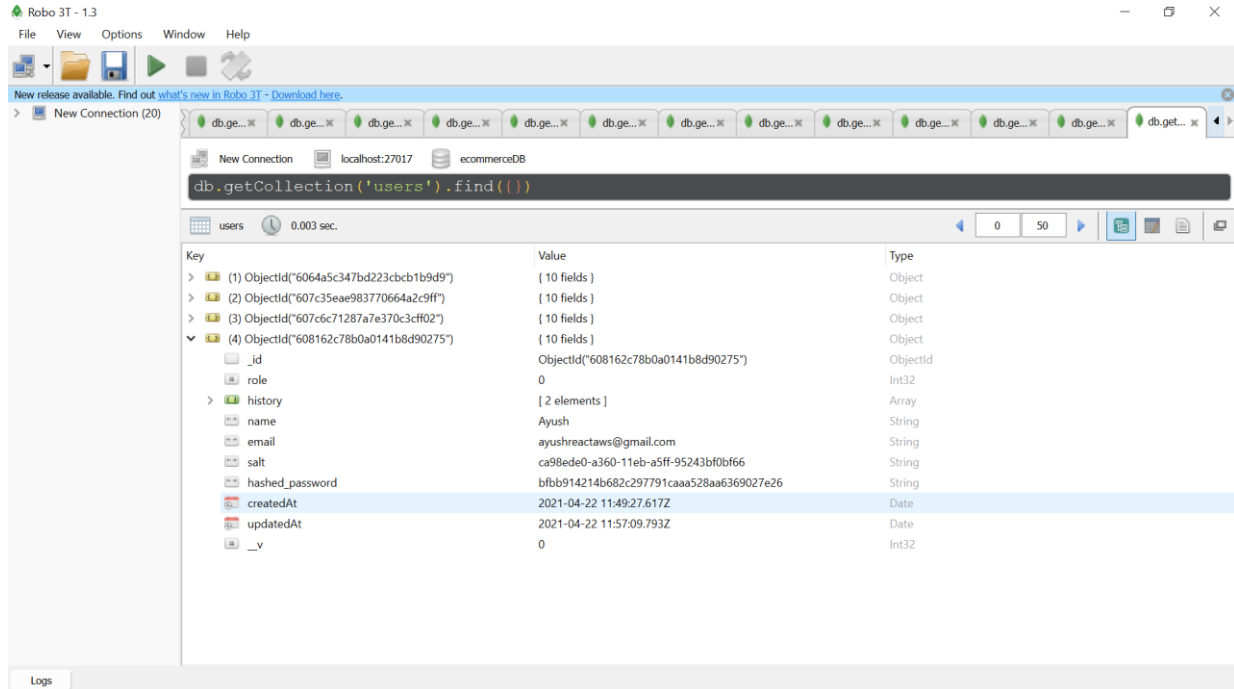


Users can also change update their profiles.

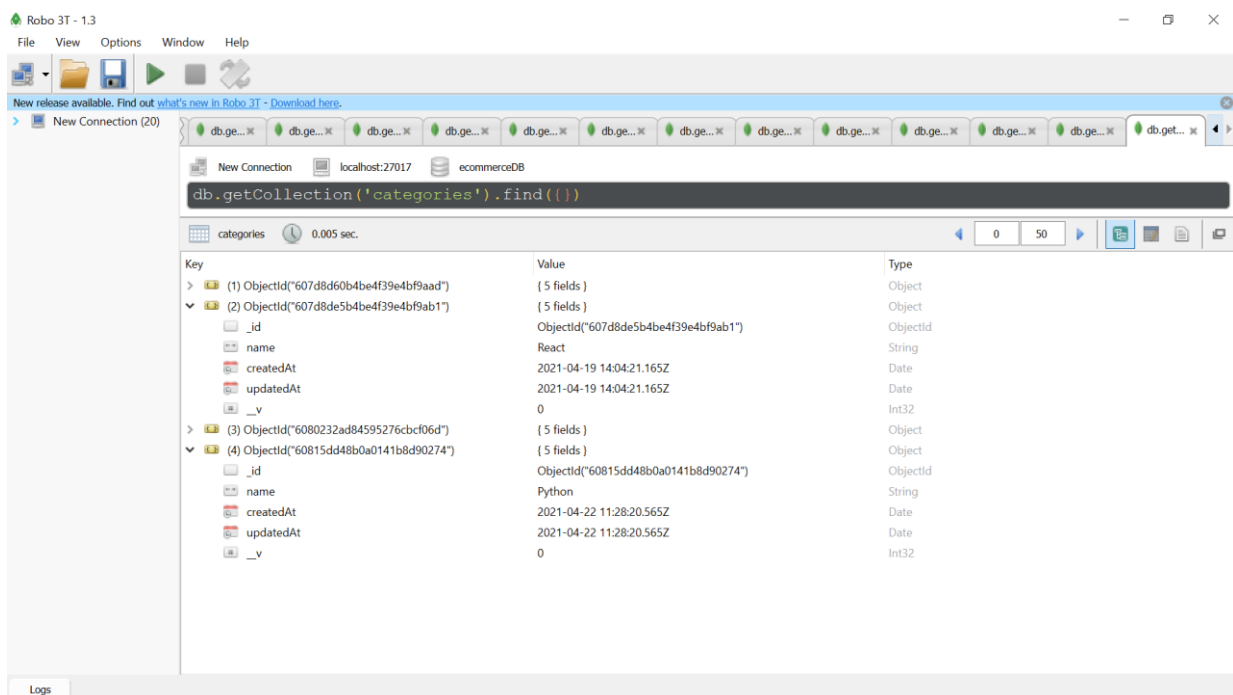


Testing

Users are saving in database after signup.



Categories saved in database



Products save in the database

Robo 3T - 1.3

File View Options Window Help

New release available. Find out [what's new in Robo 3T](#) - Download here.

> New Connection (20)

New Connection localhost:27017 ecommerceDB

```
db.getCollection('products').find({})
```

products 0.003 sec.

Key	Value	Type
> (1) ObjectId("607e9f15267ca02d78f833fd")	{ 12 fields }	Object
> (2) ObjectId("607ea0bd267ca02d78f833fe")	{ 12 fields }	Object
▼ (3) ObjectId("607ea0c5267ca02d78f833ff")	{ 12 fields }	Object
_id	ObjectId("607ea0c5267ca02d78f833ff")	ObjectId
photo	{ 2 fields }	Object
sold	0	Int32
name	PHP from Scratch	String
description	PHP for Experts	String
price	87	Int32
category	ObjectId("6080232ad84595276cbcf06d")	ObjectId
shipping	true	Boolean
quantity	54	Int32
createdAt	2021-04-20 09:37:09.033Z	Date
updatedAt	2021-04-22 11:27:16.487Z	Date
_v	0	Int32
> (4) ObjectId("607ea0c6267ca02d78f83400")	{ 12 fields }	Object
> (5) ObjectId("607ea4c0267ca02d78f83402")	{ 12 fields }	Object
> (6) ObjectId("607efe74b3e9be1fb48ad46e")	{ 12 fields }	Object
▼ (7) ObjectId("6080234dd84595276cbcf06e")	{ 12 fields }	Object

Logs

Orders of users in database

Robo 3T - 1.3

File View Options Window Help

New release available. Find out [what's new in Robo 3T](#) - Download here.

> New Connection (20)

New Connection localhost:27017 ecommerceDB

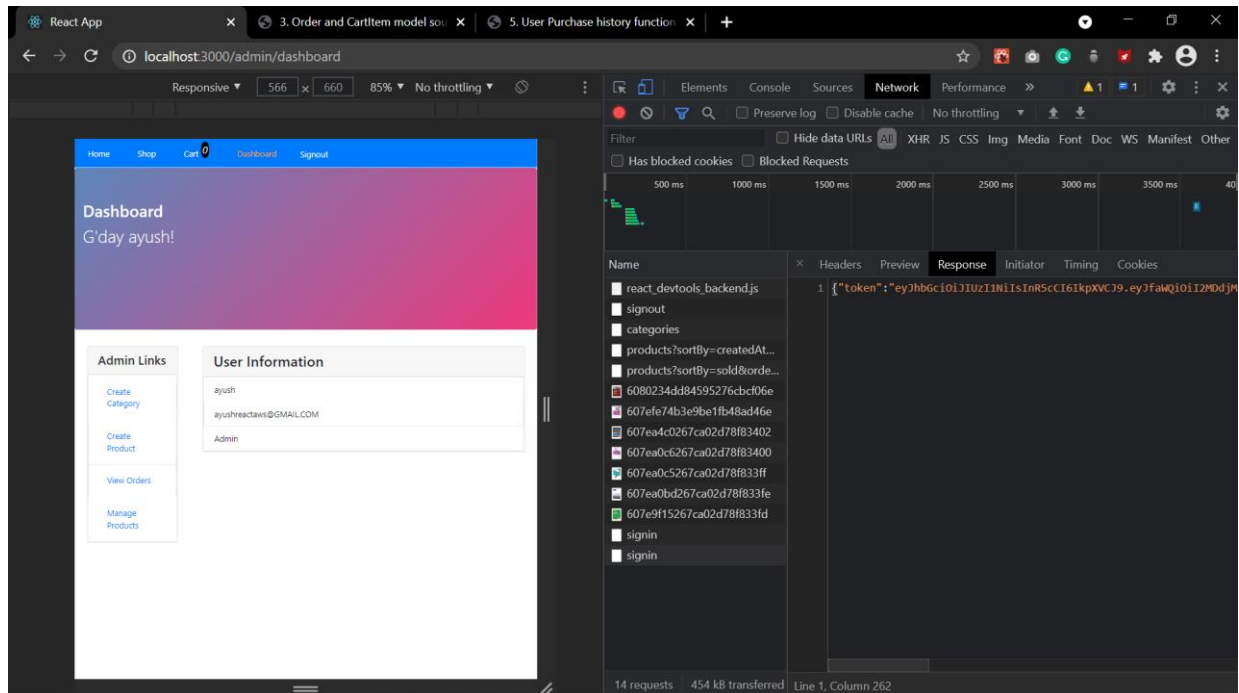
```
db.getCollection('orders').find({})
```

orders 0.005 sec.

Key	Value	Type
> (1) ObjectId("60811ab88b814c0b6c3e4155")	{ 9 fields }	Object
> (2) ObjectId("60811cc38b814c0b6c3e4157")	{ 9 fields }	Object
> (3) ObjectId("60811d4e8b814c0b6c3e4159")	{ 9 fields }	Object
> (4) ObjectId("60811ec38b814c0b6c3e415c")	{ 10 fields }	Object
> (5) ObjectId("608122bbe4923d0b88d198ac")	{ 10 fields }	Object
> (6) ObjectId("6081259e81f37e18e4a21e55")	{ 10 fields }	Object
> (7) ObjectId("60814eb71243691764eb2b41")	{ 10 fields }	Object
▼ (8) ObjectId("608164958b0a0141b8d90276")	{ 10 fields }	Object
_id	ObjectId("608164958b0a0141b8d90276")	ObjectId
status	Not processed	String
products	[2 elements]	Array
transaction_id	d1tafmw	String
amount	71	Int32
address	Hathras, Mathura, Uttar Pradesh	String
user	ObjectId("608162c78b0a0141b8d90275")	ObjectId
createdAt	2021-04-22 11:57:09.901Z	Date
updatedAt	2021-04-22 11:57:09.901Z	Date
_v	0	Int32

Logs

Token is generated when user or admin login



Appendices

Category model

```
// Requirring mongosse
const mongoose = require('mongoose');

// Creating user schema
const categorySchema = new mongoose.Schema(
  {
    name: {
      type: String,
      unique: true,
      trim: true,
      required: true,
      maxlength: 32,
    },
  },
  { timestamps: true }
);

module.exports = mongoose.model('Category', categorySchema);
```

```

const mongoose = require('mongoose');
const { ObjectId } = mongoose.Schema;

const productSchema = new mongoose.Schema(
  {
    name: {
      type: String,
      trim: true,
      required: true,
      maxlength: 32,
    },
    description: {
      type: String,
      required: true,
      maxlength: 2000,
    },
    price: {
      type: Number,
      trim: true,
      required: true,
      maxlength: 32,
    },
    category: {
      type: ObjectId,
      ref: 'Category',
      required: true,
    },
  },

```

```

    quantity: {
      type: Number,
    },
    sold: {
      type: Number,
      default: 0,
    },
    photo: {
      data: Buffer,
      contentType: String,
    },
    shipping: {
      required: false,
      type: Boolean,
    },
  },
  { timestamps: true }
);

module.exports = mongoose.model('Product', productSchema);

```

App.js

```
const express = require('express');
const mongoose = require('mongoose');
const jwt = require("jsonwebtoken"); // To generate signed token
const expressJwt = require("express-jwt"); // For authorization check
// import routes
const authRoutes = require('./routes/auth');
const userRoutes = require('./routes/user');
const categoryRoutes = require('./routes/category');
const productRoutes = require('./routes/product');
const braintreeRoutes = require('./routes/braintree');
const orderRoutes = require('./routes/order');

// It gives all the route sort of roads that have been requested in the console
const morgan = require('morgan');

const bodyParser = require('body-parser');
const cookieParser = require('cookie-parser'); // We are saving the user credentials in the cookie
const expressValidator = require('express-validator');
const cors = require("cors");

require('dotenv').config();
// app
const app = express();
```

```
    .connect(process.env.DATABASE, {
      useNewUrlParser: true,
      useCreateIndex: true,
      useFindAndModify: false,
      useUnifiedTopology: true,
    })
    .then(() => console.log('DATABASE CONNECTED'));
```

```
// middlewares
app.use(morgan('dev'));
app.use(bodyParser.json());
app.use(cookieParser());
app.use(expressValidator());
app.use(cors());

// routes middleware
app.use('/api', authRoutes);
app.use('/api', userRoutes);
app.use('/api', categoryRoutes);
app.use('/api', productRoutes);
app.use('/api', braintreeRoutes);
app.use('/api', orderRoutes);
```

You, 2 weeks ago • Fix some issues

```
const port = process.env.PORT || 8000;

app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

References

- ✓ [Stack Overflow - Where Developers Learn, Share, & Build Careers](#)
- ✓ [YouTube](#)
- ✓ [W3Schools Online Web Tutorials](#)
- ✓ [Udemy: Online Courses - Learn Anything, On Your Schedule](#)
- ✓ [DevDocs API Documentation](#)

We are using these resources as a reference to build our project. These all resources are good e-learning platforms and give us a lot of information to resolve troubles and doubts in the making of this project.