**COURSE PROJECT REPORT**

**AY: 2021-22 semester I**

**Subject:** Web Technology

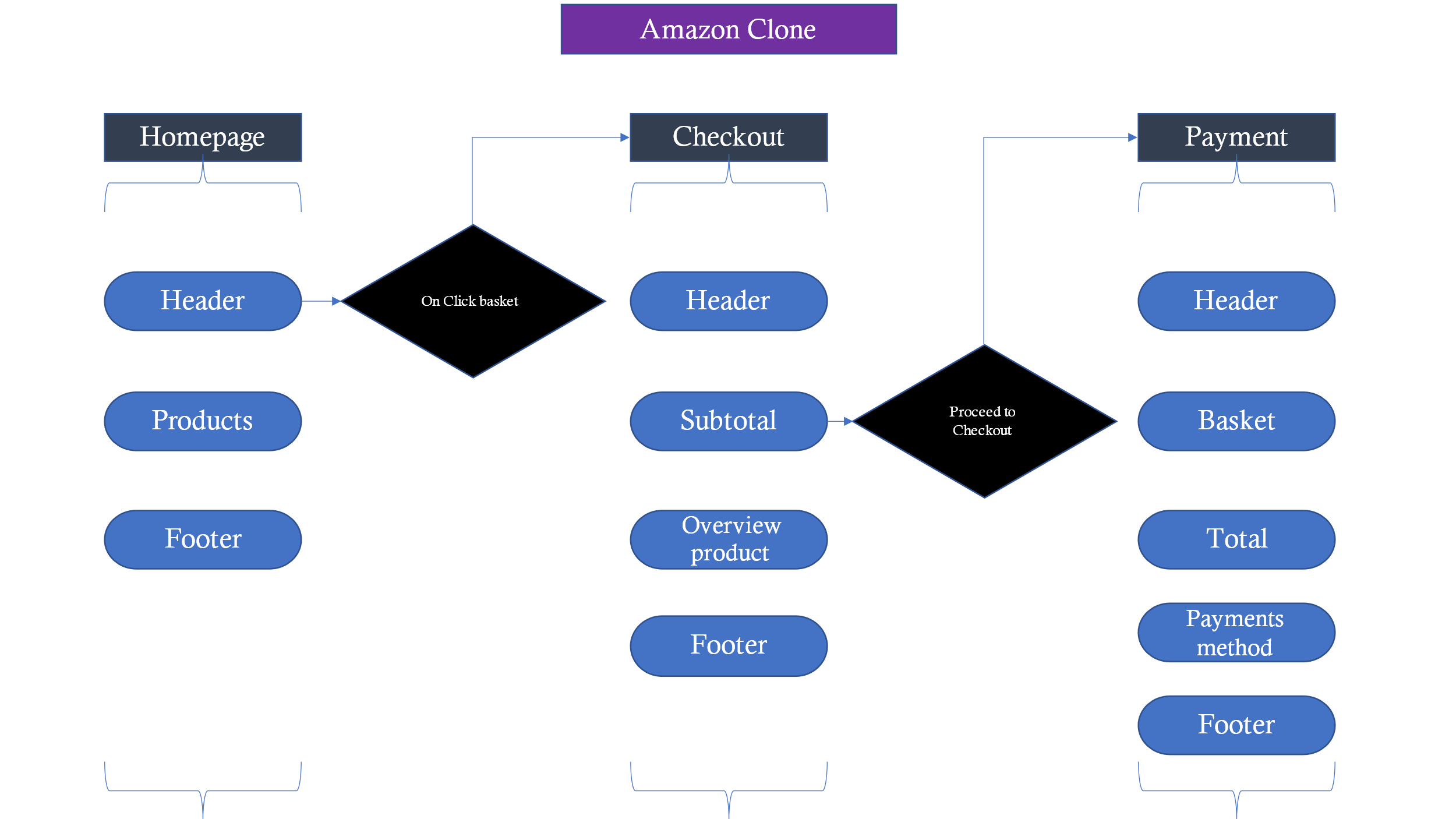
**Subject code:** ET3273

Prepared by:

|  |  |  |  |
| --- | --- | --- | --- |
| **Division** | **Roll no**: | **GR No:** | **Name** : |
| C | 52 | 11910241 | Amitesh Patil |
| C | 54 | 11910706 | Pranit Patil |
| C | 56 | 11911215 | Kshitij Patil |
| C | 77 | 11910860 | Vivek Pundkar |

**Project Title: Amazon Clone**

**Proposed system:**

* Block diagram
* Description:

It’s a great idea to start your own ecommerce platform. But it required a lot of manpower, enough time as well as capital to develop it from scratch. Amazon is a vast Internet-based enterprise that sells books, music, movies, housewares, electronics, toys, and many other goods. Amazon is the self-styled “Greatest Store on Earth”. It’s been said that Amazon aim is to sell everything to everyone as buyers differ in their need, wants, their buying behavior and attitude. Amazon take care of all these by providing over 80 million products, which are categorized into 35 departments. OVER; 6 million Clothing department, 20 million Sports & outdoors, 4 million office products, 20 million Electronic, 25 million books services.

We aim - to make an amazon clone which is similar to the original amazon web, we will focus more on the look and feel as it is clone application, i.e., it should be more similar to <https://www.amazon.in/>. In here we also aimed to create virtual card transactions and purchase listed items. Our clone will allow users to register, login, add products to the shopping cart, remove product from the shopping cart. Our clone will permit only an authenticated user to access the home page.

We will also be using Firebase for Authentication, Database, Functions and Hosting. Firebase is a great tool to get started with projects quickly. We will also be using HTML, CSS and React Context API. Please keep in mind, you need to have Visual Studio Code and NodeJS installed.

1. Setting up the React Router

A very important thing to consider in a React app is the navigation (moving from one page to another) of the users. Since React is a single page application, it doesn’t support multiple routes by default. But the node packages come to our save. There’s a package named react-router-dom which allows us to create routes for our React project. Not a lot complicated. Setting up is one time, and then whenever you add a new page, you just need to inform the Router.

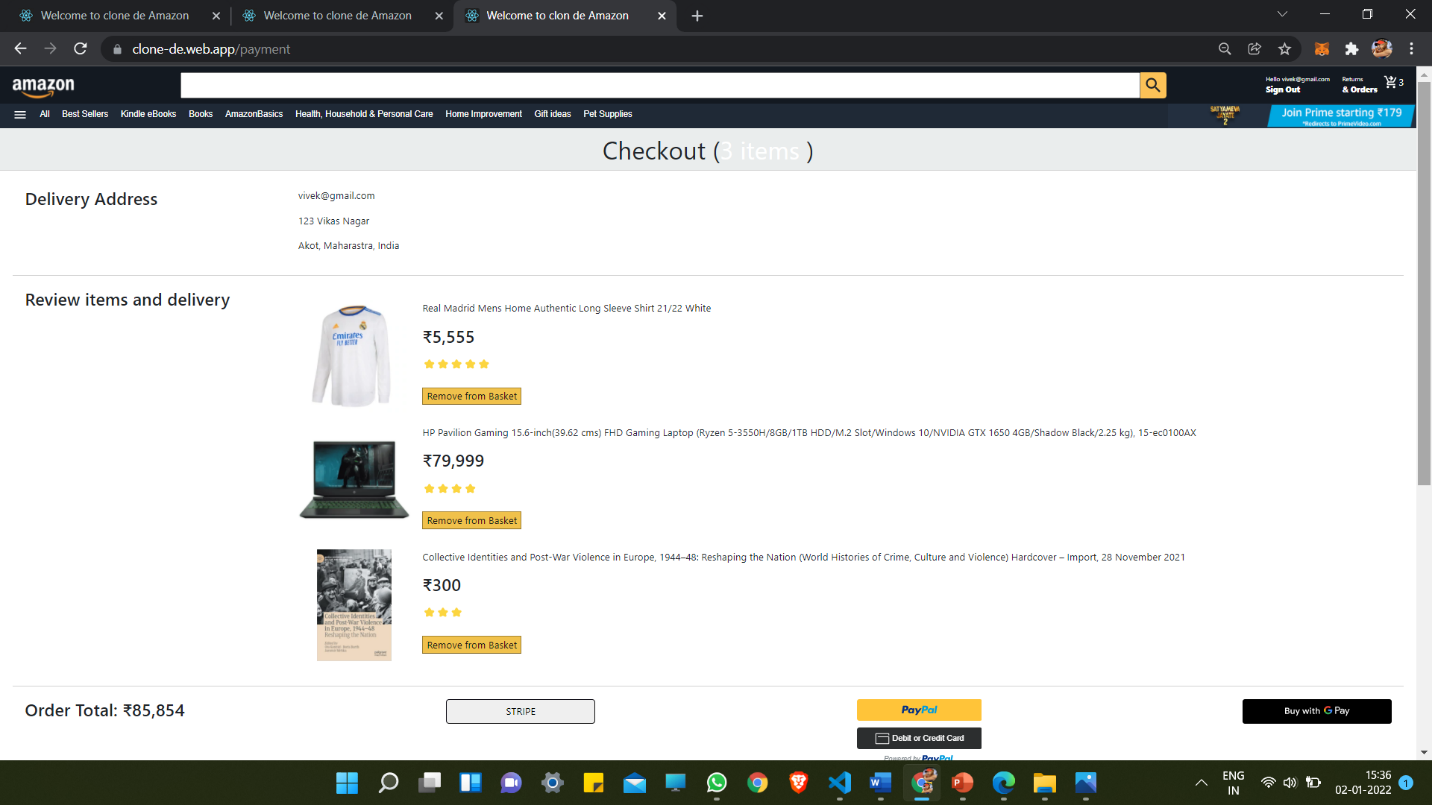
* Our motive is to have Home component to be rendered on the default route that is “/”.
* To use the Home component, we need to import it, so we imported it at the top.
* We need to wrap the entire app into the Router component, so that every component is a part of Router and has access to the Router.
* The switch specifies that the components beneath it are to be rendered only under certain routes. So if a route is meant for “/”, you cannot see the information on “/hello” route.
* The Route component is used under the Switch component. The Route component specifies the rules of the Routes and components to render at specific route. For example in this case we have the Home component render at “/” route. We will be adding more routes to this.

1. The Navbar

This component has functionalities like- on clicking amazon user will be redirected to Homepage, the login features and basket.

We are going to use a package for icons, and we need Material Icons to use them. So open your terminal and write the following command. npm install @material-ui/core @material-ui/icons

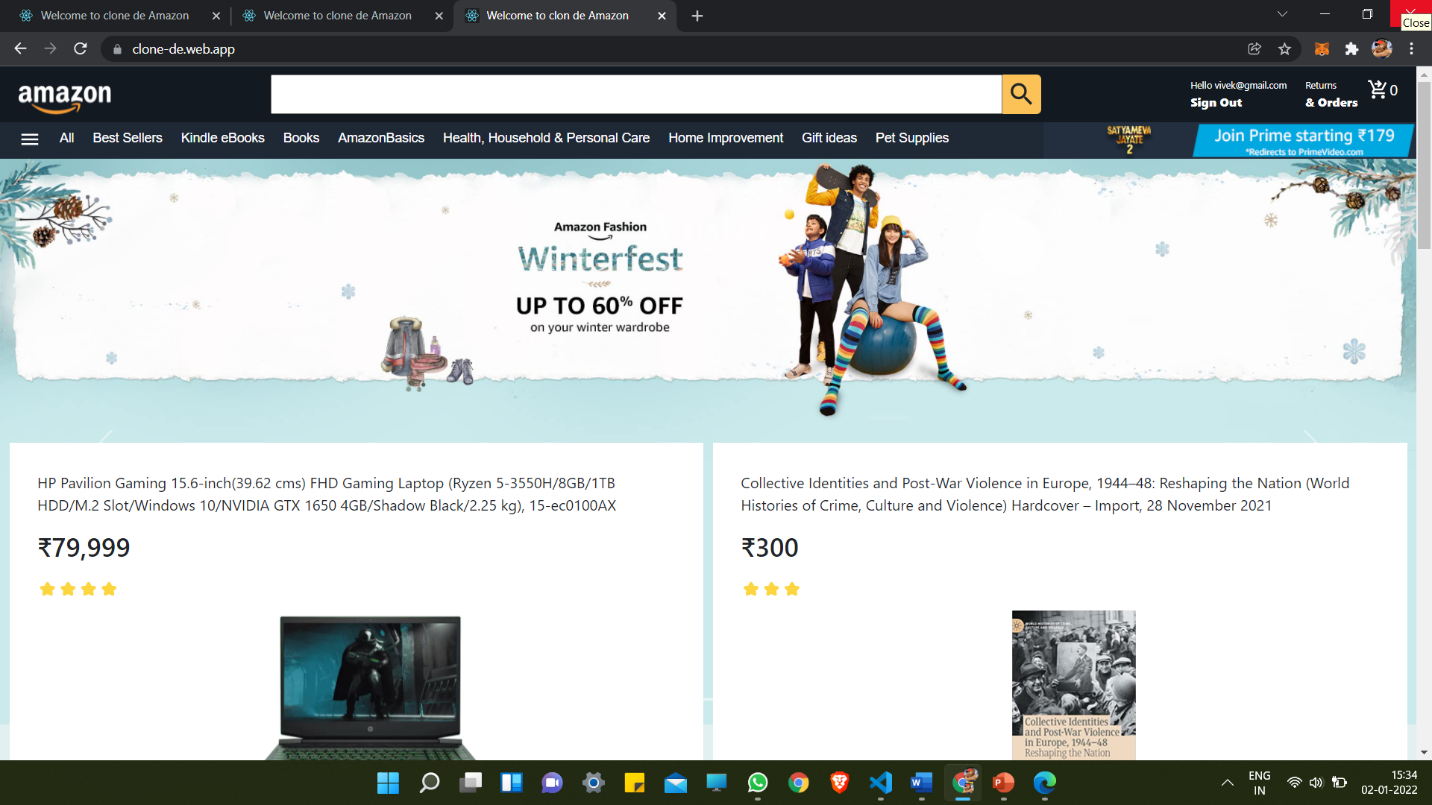
Once you installed the dependencies, you can use it to display SVG icons which are provided by Material UI. Material UI is a very popular UI library for React which has a lot of prebuilt components just as icons which makes life easier. Now let’s make a new component called Header.js and let’s make a new file and call it Header.css just as we did with the home component.



1. Creating the Amazon Homepage

We already made the Home component which is rendered by default because we configured it to be rendered it on default route “/” in the React Router. Now let’s go and do something creative in the Home component. We will start with the image which Amazon shows at the top of its landing page.

* We are passing some props through the component, which can be used by the components to render. These props are — Product ID, Title, Price, Image, Ratings.
* These props are usually variables and objects that are passed by parent components. We can render props on the screen in this case because all of the props here are viewable. So, we render the data on the screen by enclosing the variables with {} at appropriate places.
* To render the ratings, we use a bit different way here. We create an array of length of the ratings passed and loop through them and render out the star emoji the number of times the ratings passed through the props.
* The reducer is a place where you declare all the application level states which can be used later by the application.
* In initialState, we declare the states we are going to use. In this case, basket and user. We would cover them in future sections.
* The actual reducer function takes in a state and action. The state and action is provided so that the reducer can perform operations on the state.
* We export the reducer.
* The reducer is a place where you declare all the application level states which can be used later by the application.
* In initialState, we declare the states we are going to use. In this case, basket and user. We would cover them in future sections.
* The actual reducer function takes in a state and action. The state and action is provided so that the reducer can perform operations on the state.
* We export the reducer.

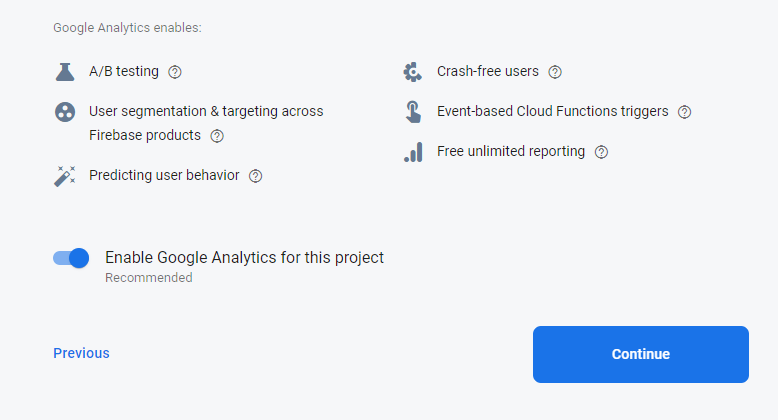
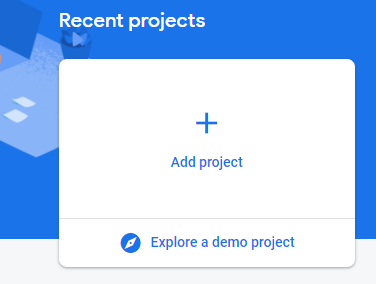


1. Adding the Basket functionality

So basically, we need to import the state whenever we feel the need and we can dispatch events. We will go in depth about that in this section. To start with things, let’s add the Add to Basket functionality to our Product.js. First of all, we import the state and dispatch using useStateValue() hook and then it’s all easy to work with stuff, so we imported it. Add this to the top of Product.js

1. Setting up Firebase

To set up firebase we first need to add a project in firebase https://firebase.google.com. After this we need to add API keys in firebase.js file inside our source folder.



1. Setting up Authentication

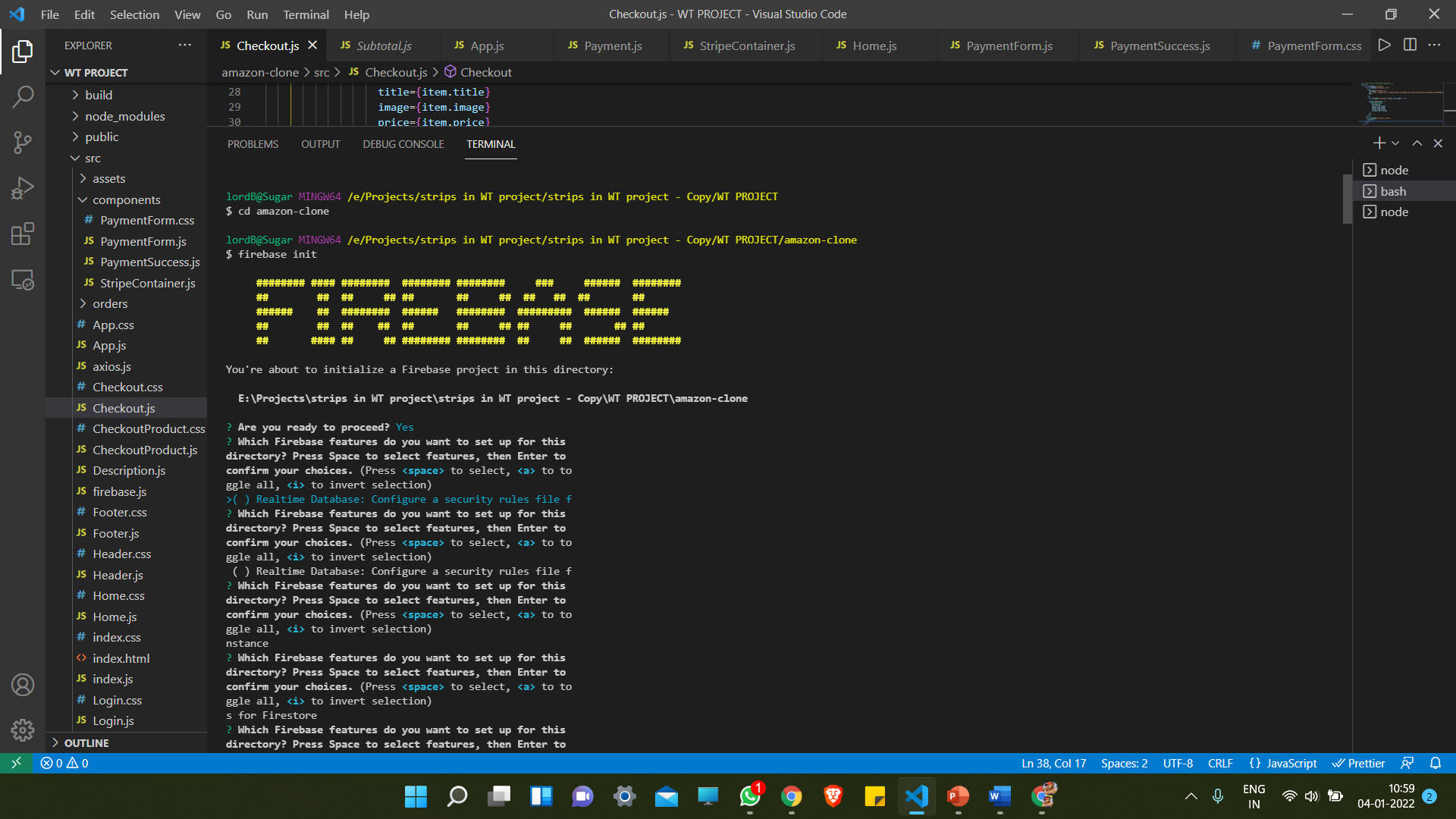
For authentication we used firebase auth. First we enabled the email authentication and all the authentication will be done on basis of email provided by user. If one doesnt have a account he can create account on same page. This authentication parameter is useful before payment and checkout product functionalities.

1. Basket

Initially basket count is 0 and after adding product to cart the count will be incremented and after clicking on it we will be redirected to checkout page inside which the product overview will be visible along with product subtotal and payment options.

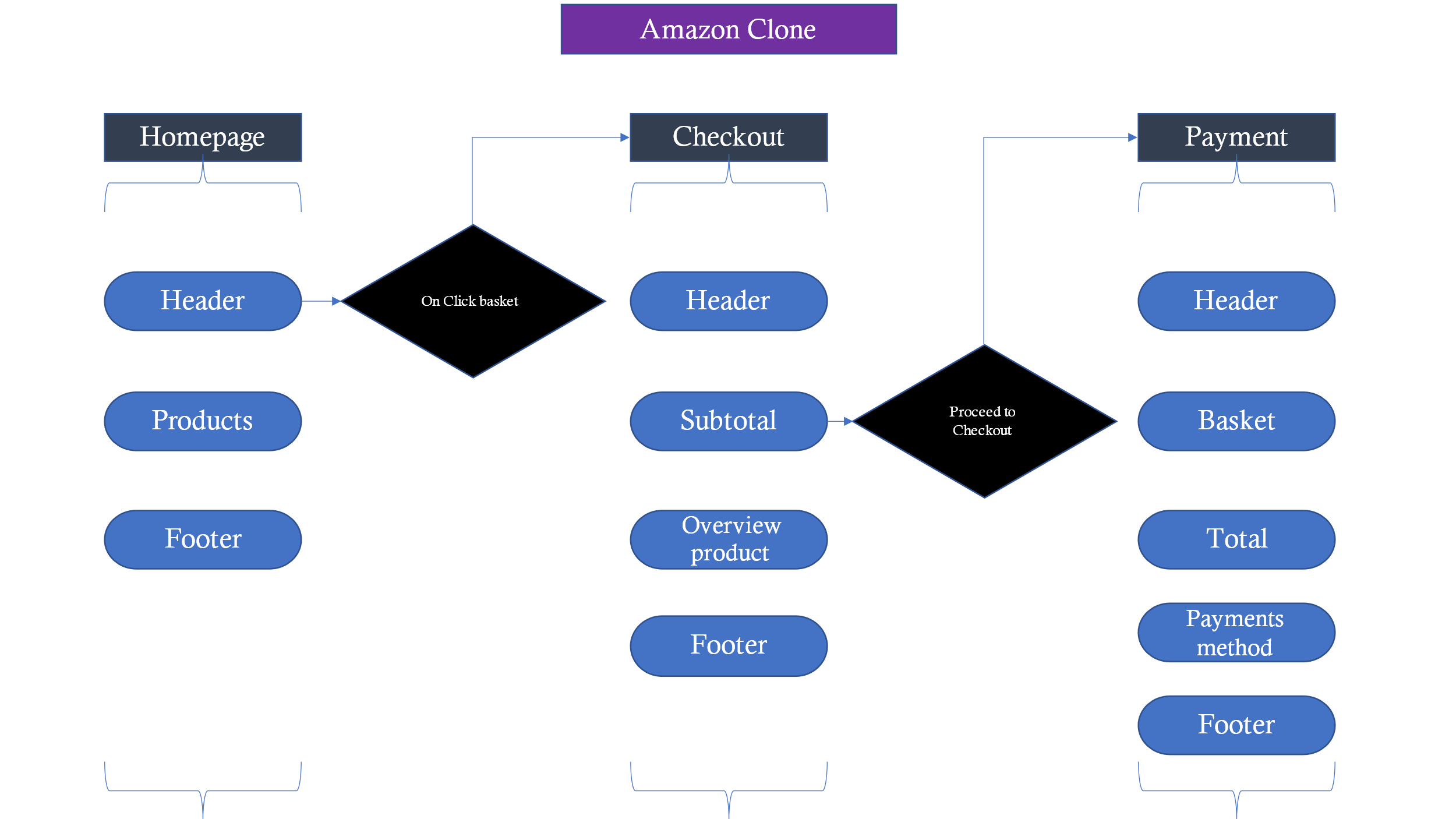
1. Deploy

To deploy we first initialize firebase inside our project folder by running firebase init and used the already created project in firebase and used it to deploy our react app. This are following step to intilize our firebase. Then the next step is run the deployed version of our react project. Using npm run deploy command



**Software implementation:**

* Flowchart



* App routes

<Router>

      <div className='apple'>

        <Switch>

          <Route path='/login'>

            <Login />

          </Route>

          <Route path='/checkout'>

            <Header />

            <Checkout />

            <Footer />

          </Route>

          <Route path='/payment'>

            <Header />

            <Elements stripe={promise}>

              <Payment />

            </Elements>

            <Footer />

          </Route>

          <Route path='/paymentViaStripe'>

            <StripeContainer />

          </Route>

          <Route path='/paymentSuccess'>

            <PaymentSuccess />

          </Route>

          <Route path='/description'>

            <Description />

          </Route>

          <Route path='/footer'>

            <Footer />

          </Route>

          <Route path='/'>

            <Header />

            <Home />

            <Footer />

          </Route>

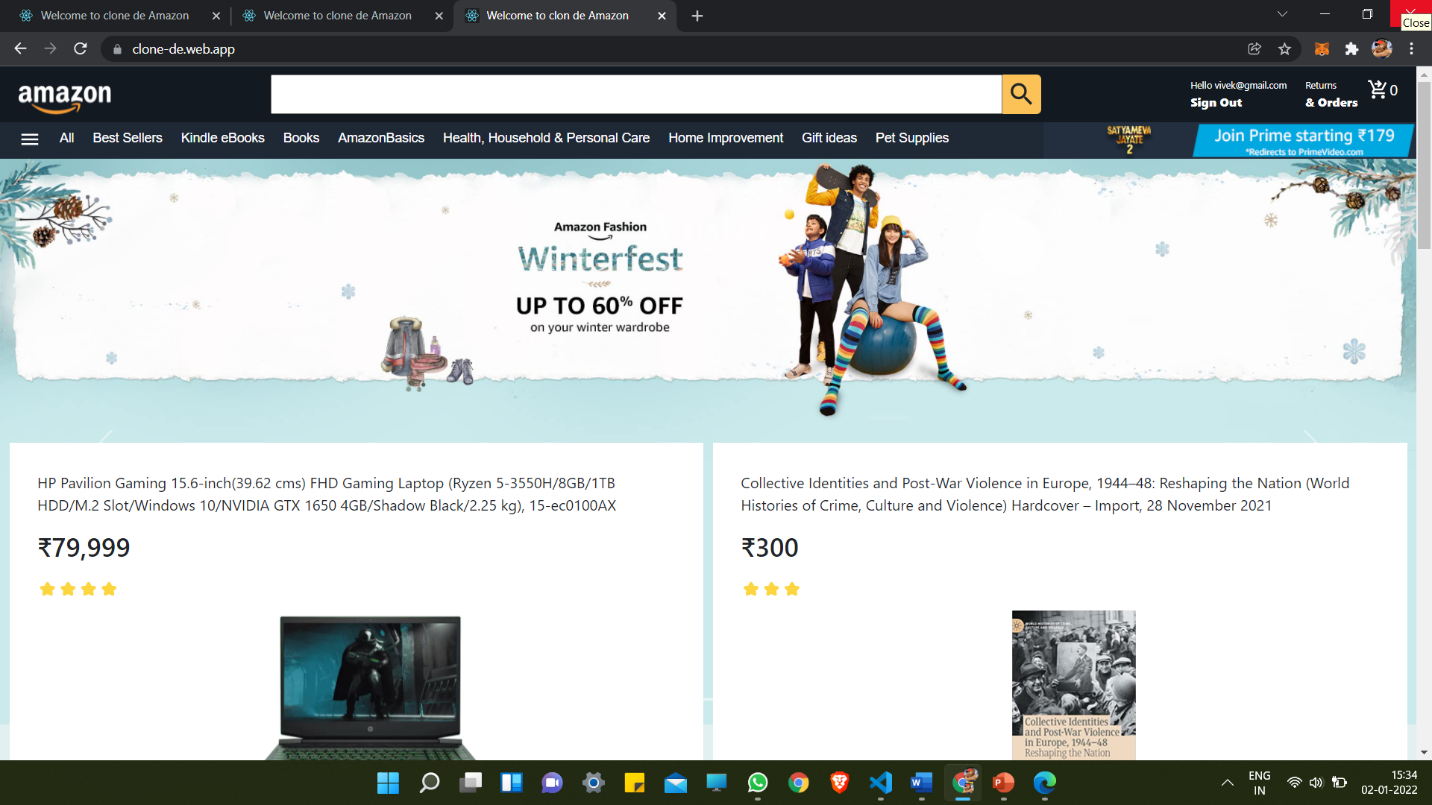
        </Switch>

      </div>

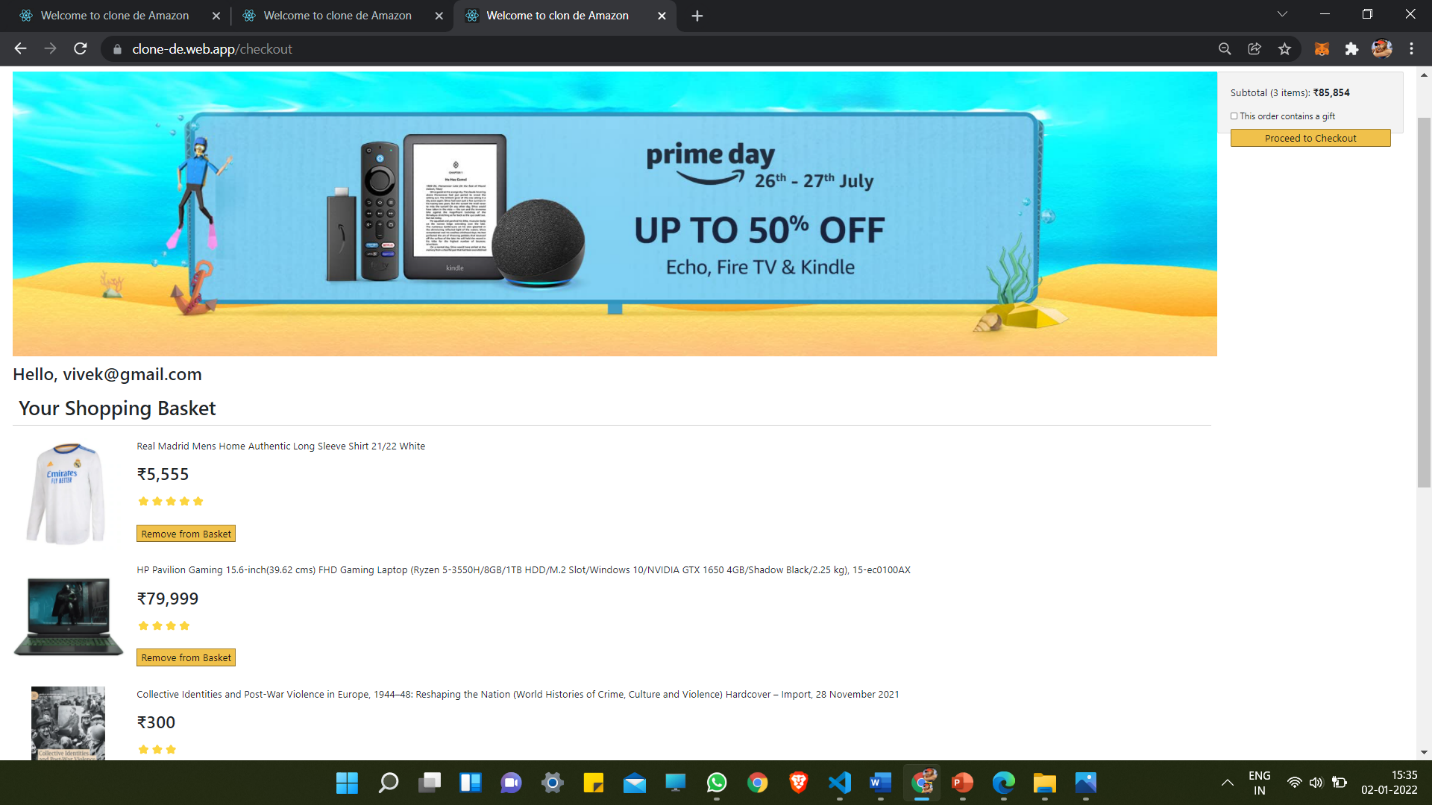
    </Router>

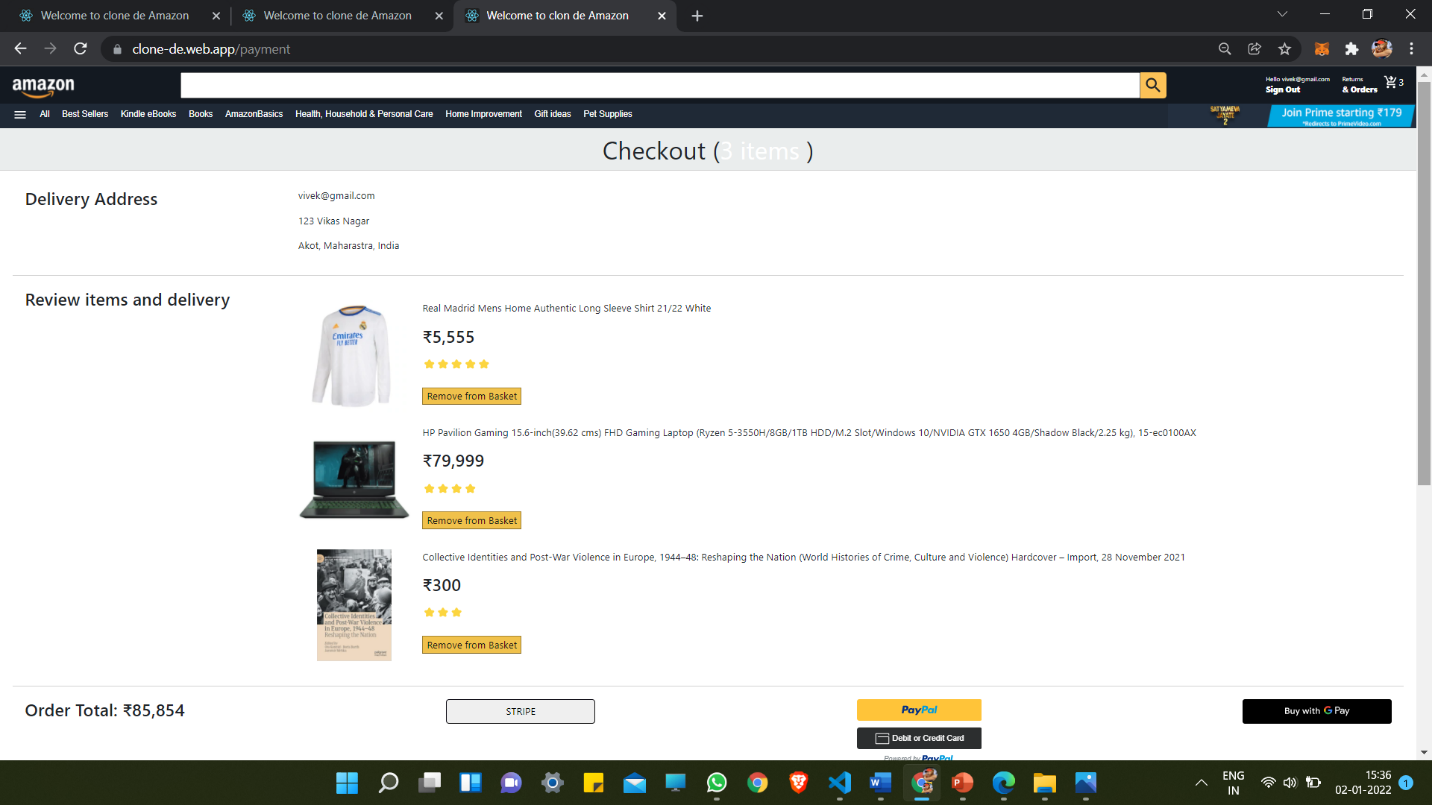
**Result:**

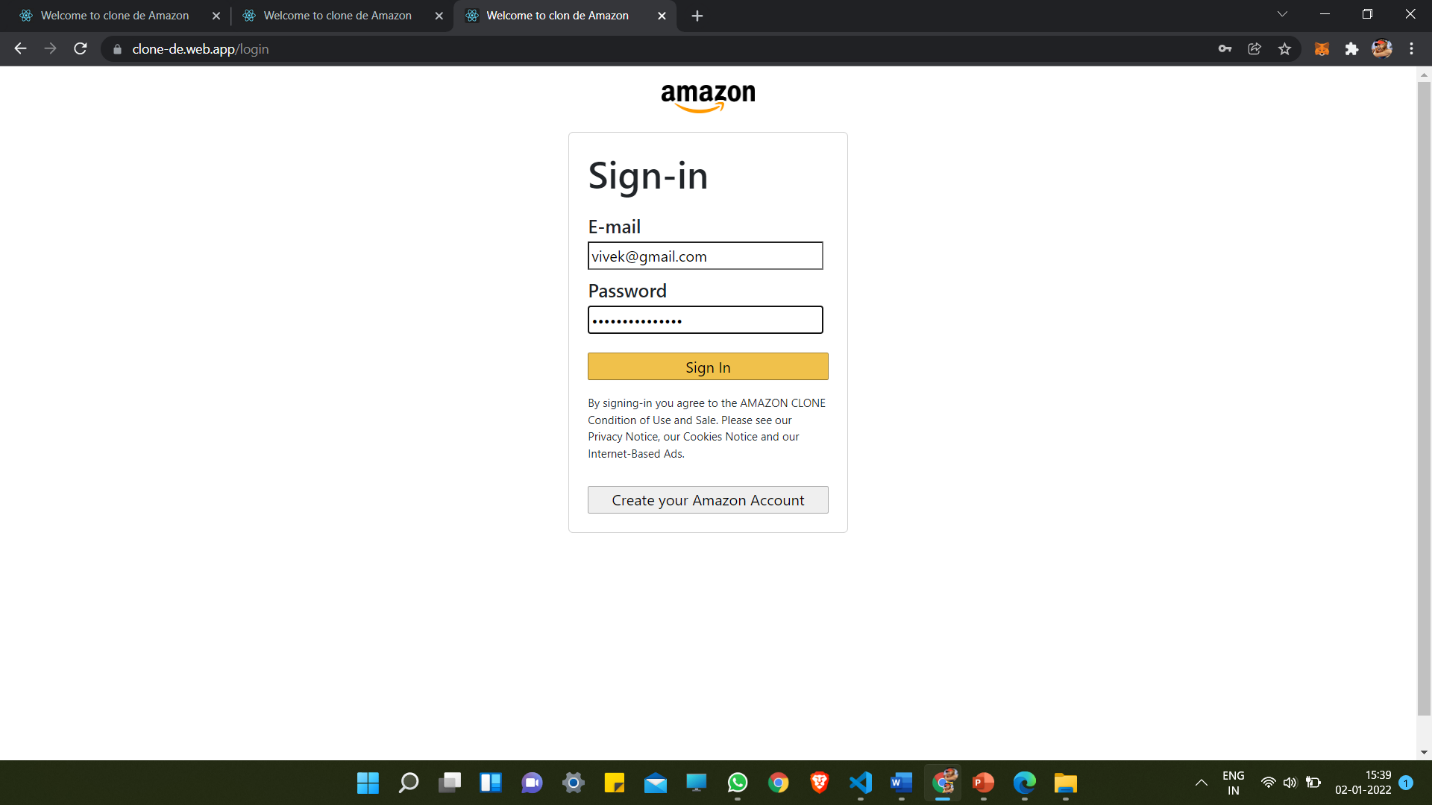
* Home



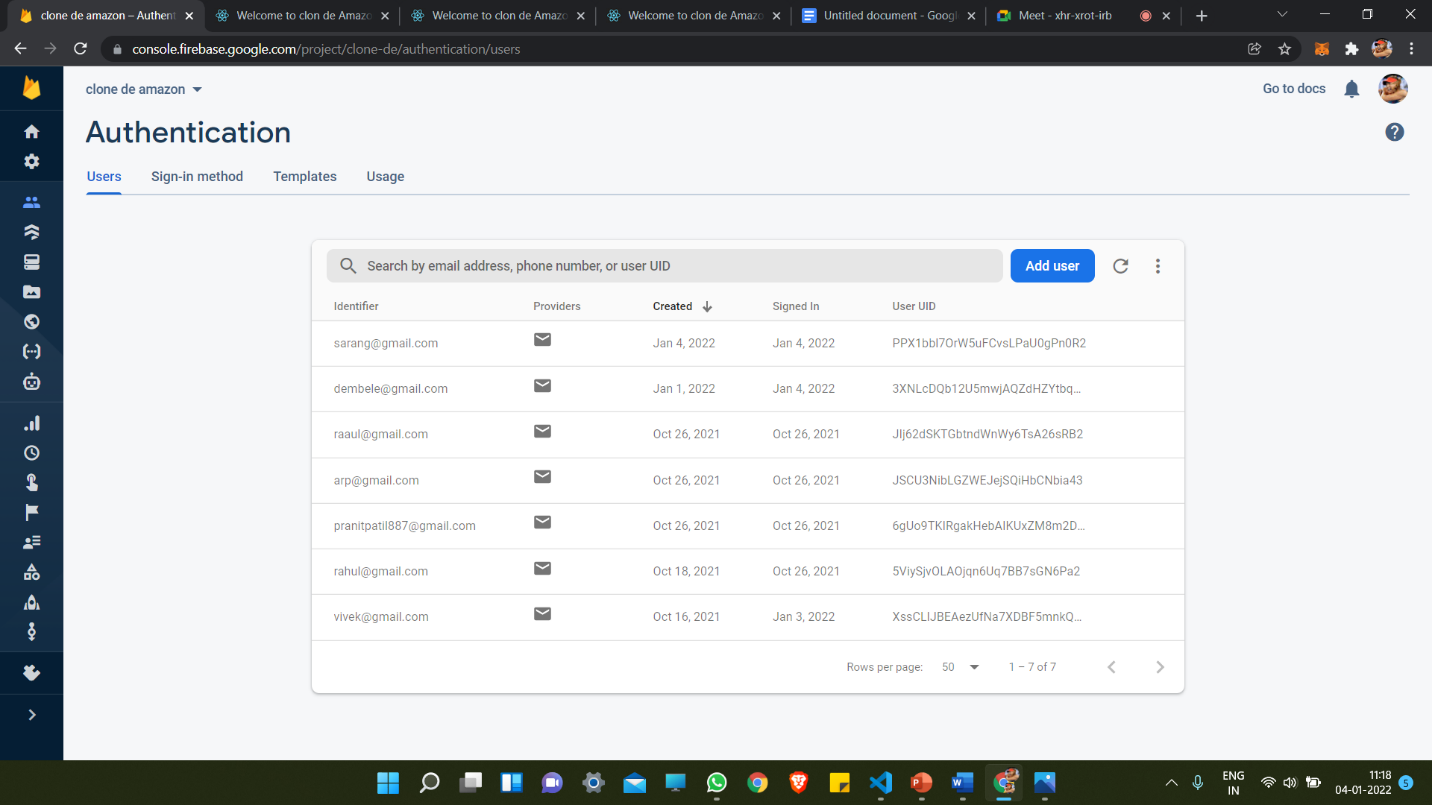
* Checkout



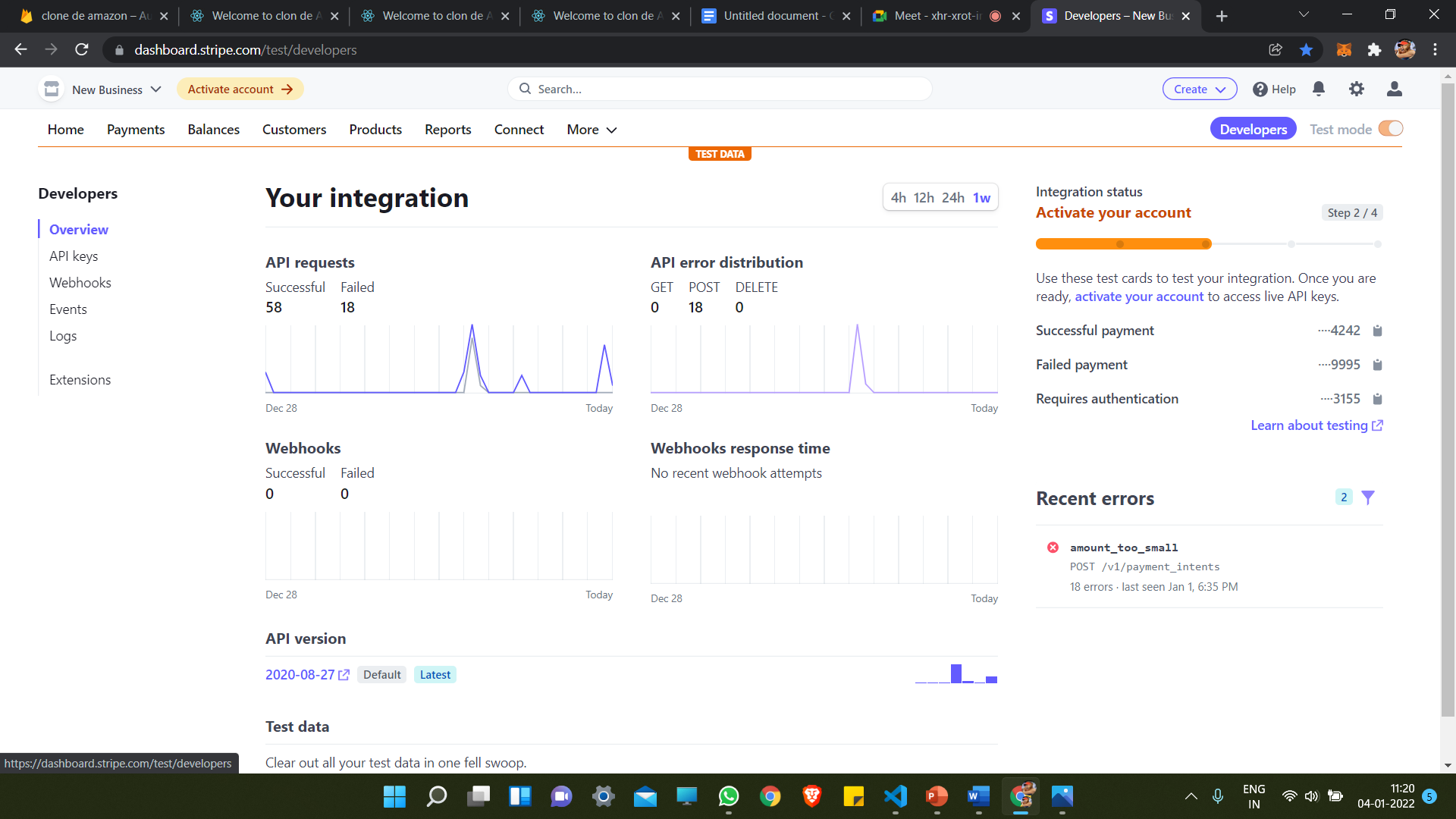
* Payment
* User Registration



* Authentication



* Stripe



**Conclusion:**

* All the functionalities are working properly. Deployed an Amazon Clone react JS app on firebase. We learned about different components of React and also learned about different CSS syntax.
* This project helped us in developing our skills in domain of web development. We got to know about various features of firebase.