**TechWorld online shopping website**

Course Name: e-**DAC**

|  |  |
| --- | --- |
| **NAME** | **PRN NUMBER** |
| Quelin Geeson C | **200950181078** |
| Diksha Tiwari | **200950181032** |
| Sanjay Prasad | **200950181090** |
| Nitish Kumar | **200950181062** |

# WHAT OUR PROJECT DOES?

* Online shopping portal to sell tech products like mobile phones, headphones etc.
* Users can register and login to buy the products.
* There is an option for searching which are available in stock.
* Admin can add and remove products and also update the status of the products.
* Users can give reviews and rating about the products (one rating per user).
* User can add all the products he/she wants to buy to his/her cart.
* We have used PayPal API for accepting payments from user.
* All the data is saved in database (Mongo DB).
* There is a profile page where user can see all his orders.
* We will be MERN stack for creating this application.

# WORK FLOW

**Workflow for homepage:**

**Presention:**

HomeScreen.js:

1.ProductList:Get the products from state

2.using useEffect() :we dispatch:action in redux to get list of all products and pass listProducts() as parameter.and render the products.

**Service:**

**Frontend:**

productActions.js:

Using dispatch(type:PRODUCT\_LIST\_REQUEST) we calls the function of productListReducer() which returns{loading:true,products:[]}

**Web service**: we use axios to contact the backend server and fetch the data from server.

**Request:**const {data}=await axios.get(

`/api/products?keyword=${keyword}&pageNumber=${pageNumber}`

**Backend:**According to the axios call and route ,the server handles the request.

**Server.js:**

app.use('/api/products',productRoutes)

**productRouts.js:**

router.route('/').get(getProducts)

**productControllers.js:**

getProducts()

**response:** res.json({products ,page ,pages: Math.ceil(count / pageSize)})

**Frontend:** the response from server is saved in data and is passed as payload to reducer, which sets the state.

**productActions:**

dispatch(type:PRODUCT\_LIST\_SUCCESS,payload:data): calls the function of productListReducer()

and the reducer returns{ loading: false,

products: action.payload.products,

pages: action.payload.pages,

page:action.payload.page

}

dispatch(type:PRODUCT\_LIST\_FAIL): calls the function of productListReducer() and

returns: {loading:false,error:action.payload}

**DAO:**fetch the details from the database

productController: const products = await Product.find({ ...keyword}).limit(pageSize).skip

(pageSize \* (page -1)

**SOME SCENARIOS WHERE WE GOT STUCK AND HOW DID WE OVERCOME**

* + **IN FROTEND:**
    - **Products were not loading on page:** The products were not loading as the state of products was empty. It was taking few seconds to load the data.

**Solution**: We used a loader component to handle the response delay.

* + - **Normal users were able to use admin features:** Features such as mark as delivered and update and delete products.
    - **Solution**: We used multiple conditional statements to check if user is logged in or admin is logged in by checking the state of user.
  + **IN BACKEND:**

If one user logs out and other user logs in, the second user were able to see the details of the previously logged in user.

**Solution** :we used a function to reset the state of user details once a user logs out

of his account using dispatch(ORDER\_LIST\_MY\_RESET)

Integration of PayPal API payment method.

**Solution** :we used PayPal sandbox SDK to create business and personal dummy accounts , and used the PayPal button API for processing the transaction.

MongoDB cluster connection issue

**Solution:** we had to use version 1.1 and earlier to connect to mongo compass and we had to set access list entry to: 0.0.0.0/0

**WHAT WERE YOUR LEARNINGS DURING THIS PROJECT?**

* We learnt to develop Full Stack application and learnt how Presentation Layer interacts with the Service Layer.
* We have learnt how to use component talking, routing between two components.
* How to Authenticate User Credentials when the user logins to respective account using JWT Token .
* How to create RESTAPIs .
* How to create databases and do CRUD operations in MongoDB.
* Basic architecture of MERN stack.
* Doing a small POC first before applying the logic in the project code.
* Always first check the value returned by a function by writing console.log() and looking at Console in the Developer Tools of the browser window.
* Using Redux developer tools to debug errors.
* Basic React fundamental like Components, Props, Hooks, State.
* Scheduling and prioritizing tasks using Trello SCRUM Board by creating sprints and following Agile Methodology.
* Collaboration using git and GitHub.
* How to use postman for checking REST API’s.