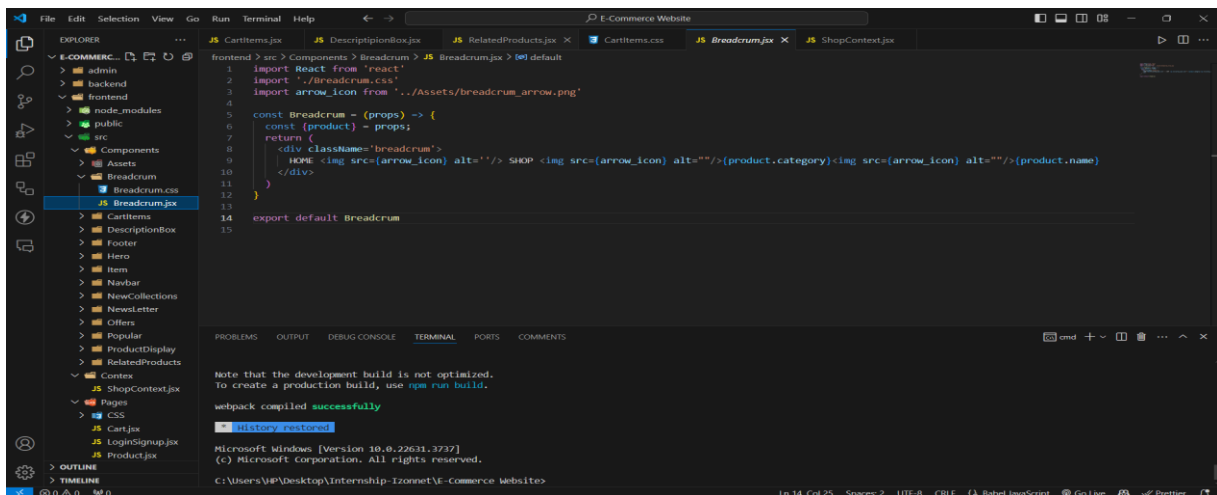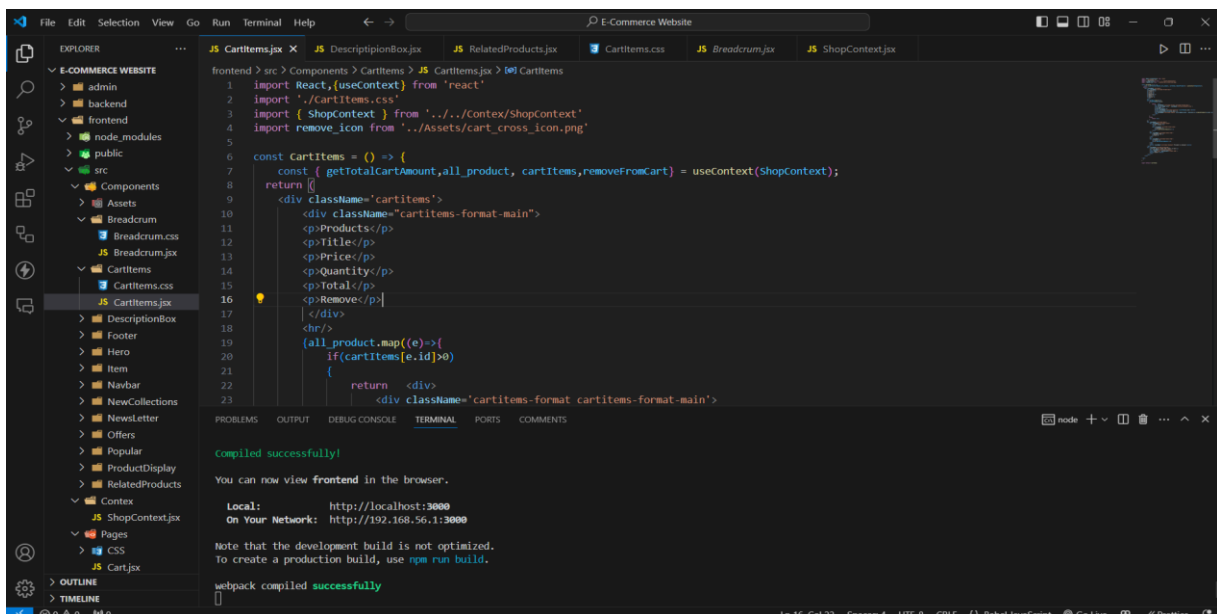# WEEKLY REPORT

Work done in last week (Attach supporting Documents):

1. 10/06/2024 Monday
   - 9.30am to 11.30 am: Create Breadcrum and add CSS styling in it.

   - 12.00pm to 2.30 pm: Create cart item For the Website

- 3.00 pm to 5.00 pm: Create Shop Context for the website

```jsx
import React,{createContext} from 'react'
import all_product from '../Components/Assets/all_product'

export const ShopContext = createContext(null);
const getDefaultCart =()=>{
    let cart={};
    for(let index= 0; index<all_product.length+1; index++){
        cart[index]=0;
    }
    return cart;
}

const ShopContextProvider = (props)=>{
const [cartItems,setCartItems]=React.useState(getDefaultCart());

const addToCart = (itemId)=>{
    setCartItems((prev)=>{
        return {...prev,[itemId]:prev[itemId]+1}
    })
}
const removeFromCart = (itemId)=>{
    setCartItems((prev)=>{
```

2. 11/06/2024 Tuesday
    - 9.30am to 11.30 am: Create Product Display page for Website and add CSS styling.

```jsx
import React, { useContext } from 'react'
import './ProductDisplay.css'
import star_icon from '../Assets/star_icon.png';
import star_dull_icon from '../Assets/star_dull_icon.png';
import { ShopContext } from '../../Contex/ShopContext';

const ProductDisplay = (props) => {
    const {product} = props;
    const {addToCart} = useContext(ShopContext);
    return (
        <div className='productdisplay'>
            <div className='productdisplay-left'>
                <div className='productdisplay-img-list'>
                    <img src={product.image} alt="" />
                    <img src={product.image} alt="" />
                    <img src={product.image} alt="" />
                    <img src={product.image} alt="" />
                </div>
                <div className='productdisplay.img'>
                    <img className='productdisplay-main-img' src={product.image} alt=''/>
                </div>
            </div>
            <div className='productdisplay-right'>
```

```css
.productdisplay
{
    display: flex;
    margin: 0px 170px ;
}
.productdisplay-left
{
    display: flex;
    flex-direction: column;
    gap:16px;
}
.productdisplay-img-list img{
    height:163px;
}
.productdisplay-main-img{
    width: 586px;
    height:700px;
}
.productdisplay-right{
    margin: 0px 70px;
    display: flex;
    flex-direction: column;
```

- 12.00pm to 2.30 pm: Create Related Product components and add CSS styling

```
JS CartItems.jsx        JS DescriptipionBox.jsx        JS RelatedProducts.jsx ×        3 CartItems.css        JS ShopContext.jsx        3 RelatedProducts.css

frontend > src > Components > RelatedProducts > JS RelatedProducts.jsx > [∅] RelatedProducts
   1   import React from 'react'
   2   import './RelatedProducts.css'
   3   import data_product from '../Assets/data'
   4   import Item from '../Item/Item';
   5   const RelatedProducts = () => {
   6     return (
   7       <div className=' relatedproducts'>
   8       <h1>Related Products</h1>
   9       <hr/>
  10       <div className="relatedproducts-item">
  11       {data_product.map((item,i)=>{
  12       return < Item key={i} id={item.id} name={item.name} image ={item.image} new_price={item.new_price} old_price = {item.old_price} />
  13       })}
  14       </div>
  15       </div>
  16     )
  17   }
  18
  19   export default RelatedProducts
  20
```

```
JS CartItems.jsx        JS DescriptipionBox.jsx        JS RelatedProducts.jsx        3 CartItems.css        JS ShopContext.jsx        3 RelatedProducts.css ×

frontend > src > Components > RelatedProducts > 3 RelatedProducts.css > ₺ .relatedproducts hr
   1   .relatedproducts{
   2       display: flex;
   3       flex-direction: column;
   4       align-items: center;
   5       gap: 10px;
   6       height: 90vh;
   7   }
   8   .relatedproducts h1{
   9       color: ☐#171717;
  10       font-size: 50px;
  11       font-weight: 600;
  12   }
  13   .relatedproducts hr{
  14       width: 200px;
  15       height: 6px;
  16       border-radius: 10px;
  17       background :☐#252525;
  18   }
  19   .relatedproducts-item{
  20       margin-top: 50px;
  21       display: flex;
  22       gap: 30px;
  23
```

- 3.00 pm to 5.00 pm: Create a Description box for the website and add CSS styling

```
JS CartItems.jsx        JS DescriptipionBox.jsx ×        JS RelatedProducts.jsx        3 CartItems.css        JS ShopContext.jsx        JS LoginSignup.jsx

frontend > src > Components > DescriptionBox > JS DescriptipionBox.jsx > [∅] DescriptipionBox
   1   import React from 'react'
   2   import './DescriptipionBox.css'
   3
   4   const DescriptipionBox = () => {
   5     return (
   6       <div className='descriptionbox'>
   7       <div className='descriptionbox-navigator'>
   8           <div className='descriptionbox-nav-box'>Description</div>
   9           <div className="descriptionbox-nav-box fade">Reviews (122)</div>
  10       </div>
  11
  12       <div className='descriptionbox-description'>
  13       <p>An E-commerce website is an online platform that facilitate the
  14   buying and selling of products or services over the internet.It
  15   serves as a virtual marketplace where businesses and individuals can
  16   showcase their products, interact with customers, and conduct
  17   transactions without the need for a physical presence. E-commerce
  18   websites have gained immense popularity due to their convenience,
  19   accessibility, and the global reach they offer.</p>
  20           <p>E-commerce websites typically display products or services along with detailed descriptions, images, prices, and any available variati
  21       </div>
  22
  23       </div>
```

3. 12/06/2024 Wednesday
   - 9.30 am to 11.30 am: Create Admin for the website and add Product component

```jsx
import React from 'react'
import './AddProduct.css'
import upload_area from '../../assets/upload_area.svg'

const AddProduct = () => {

    const [image,setImage] = useState(false);
    const [productDetails, setProductDetails] = useState({
        name:"",
        image:"",
        category: "women",
        new_price:"",
        old_price:""
    })

    const imageHandler = (e) =>{
        setImage(e.target.files[0]);
    }

    const changeHandler = (e) =>{
        setProductDetails({...productDetails,[e.target.name]:e.target.value})
    }
```

- 12.00pm to 2.30 pm: Create a Navigation bar and sidebar for the admin.

```jsx
import React from 'react'
import './Sidebar.css'
import { Link } from 'react-router-dom'
import add_product_icon from '../../assets/Product_Cart.svg'
import list_product_icon from '../../assets/Product_list_icon.svg'

const Sidebar = () => {
  return (
    <div className='sidebar'>
      <Link to={'/addproduct'} style={{textDecoration:"none"}}>
        <div className="Sidebar-item">
          <img src="{add_product_icon}" alt="" />
          <p>Add Product</p>
        </div>
      </Link>

      <Link to={'/listproduct'} style={{textDecoration:"none"}}>
        <div className="Sidebar-item">
          <img src="{list_product_icon}" alt="" />
          <p>Product List</p>
        </div>
      </Link>
```

- 3.00 pm to 5.00 pm: Create Product List for the Admin

```jsx
import React from 'react'
import './ListProduct.css'
import cross_icon from '../../assets/cross_icon.png'

const ListProduct = () => {

  const [allproducts, setAllProducts] = useState([]);

  const fetchInfo = async ()=>{
    await fetch('http://localhost:4000/allproducts')
    .then((res)=>res.json())
    .then((data)=>{setAllProducts(data)});
  }

  useEffect(()=>{
    fetchInfo();
  },[])

  const remove_product = async (id)=>{
    await fetch('http://localhost:4000/remove_product',{
      method: 'POST',
      headers:{
        Accept:'aoolication/json',
```

4. 13/06/2024 Thursday
   - 9.30 am to 11.30 am: Connect Backend with MongoDB

- 12.00pm to 2.30 pm: Create Image store engine and uploading endpoint for images

```js
// Image Storage Engine

const storage = multer.diskStorage({

    destination: './upload/images',
    filename:(req,file,cb)=>{
        return cb(null,`${file.fieldname}_${Date.now()}${path.extname(file.originamname)}`)
    }
})

    const upload = multer({storage:storage})

    // Creating Upload Endpoint for images

    app.use('/images',express.static('upload/images'))

    app.post("/upload",upload.single('product'),(req,res)=>{
        res.json({
            success: 1,
            image_url :`http://localhost:${port}/images/{req.file.filename}`
        })
    })
```

- 3.00pm to 5.00 pm: Create Schema for Product and user model.

```js
})

// Schema for creating Product

const Product = mongoose.model( "Product" , {
    id:{
        type: Number,
        required : true,
    },
    name:{
        type: String,
        required : true,
    },
    image:{
        type: String,
        required : true,
    },
    category:{
        type:String,
        required : true,
    },
    new_price:{
        type:Number,
```

5. 14/06/2024 Friday
    - 9.30 am to 11.30 am: Create a API for add product

```js
app.post('/addproduct' , async (req,res)=>{
    let products = await Product.find({});
    let id;
    if(product.length>0)
    {
        let last_product_array = products.slice(-1);
        let last_product = last_products_array[0];
        id = last_product.id+1;
    }
    else
    {
        id=1;
    }

    const product = new Product({
        id:id,
        name : req.body.name,
        image : req.body.image,
        category : req . body. category,
        new_price: req.body.new_price,
        old_price : req .body.old_price,
    });
```

- 12.00pm to 2.30 pm: Create a backend part for login, section fetch data etc.

```js
// creating endpoint for user login

app.post('/login', async (req, res)=>{
    let user = await Users.findOne({email:req.body.email});
    if (user) {
        const passCompare = req.body.password === user.password;
        if (passCompare) {
            const data = {
                user:{
                    id:user.id
                }
            }
            const token = jwt.sign(data,'secret_ecom');
            res.json({success:true,token});
        }
        else{
            res.json({success:false,errors:"Wrong Password"});
        }
    }
    else{
        res.json({success:false,errors:"Wrong Email id"})
    }
})
```

- 3.00pm to 5.00 pm: Create a backend for Cart.

```js
app.post('/addtocart',fetchUser,async(req,res)=>{
    userData.cartData[req.body.itemId] +=1;
    await Users.findOneAndUpdate({_id:req.user.id},{cartData:userData.cartData});
    res.send("Added")
})

// creating endpoint to remove product from cartdata

app.post('/removefromcart',fetchUser,async(req,res)=>{
    console.log("removed",req.body.itemId);
    let userData = await Users.findOne({_id:req.user.id});
    if(userData.cartData[req.body.itemId]>0)
        userData.cartData[req.body.itemId] -= 1;
    await Users.findOneAndUpdate({_id:req.user.id},{cartData:userData.cartData});
    res.send("Removed")
})

// creating endpoint to get cartdata

app.post('/getcart', fetchUser, async (req, res)=>{
    console.log("GetCart");
    let userData = await Users.findOne({_id:req.user.id});
    res.json(userData.cartData);
```

References:

1. YouTube Channel: Thapa Technical
2. React Documentation
3. Express Documentation
4. Node Documentation
5. MongoDB Documentation

Signature of External Guide   **Signature of Internal Guide**

Student Id: 22IT116
Student Name: TRUSHANGKUMAR SUMANBHAI PATEL