

MWA mini project  
Yuvraj Vijay Agarkar  
CS3172, RBTL22CS144  
TY A

Github link [https://github.com/Yuvraj102/MWA\\_mini-project-](https://github.com/Yuvraj102/MWA_mini-project-)  
(it also has a mp4 video, describing the project, instruction to run the code are written in repo readme file)

Source code frontend

App.js

```
import './App.css';
import { MyContext } from './MyContext';
import { useEffect, useState } from 'react';
import { makePayment } from './stripe/stripedata';
import { BrowserRouter as Router, Route, Switch }
from 'react-router-dom';
import Register from './components/Register'
import Login from './components/Login'
import Home from './components/Home';
import Sell from './components/Sell';
import Donation from './components/Donation';

function App() {
  const [loggedInUser, setloggedInUser] =
  useState(null);
  useEffect(() => {
```

```

(async function() {
    // await makePayment([
    //     name : "sample",
    //     images :
    ['https://www.google.com/url?sa=i&url=https%3A%2F%2Fp
    Fpixabay.com%2Fimages%2Fsearch%2Fnature%2F&psig=AOv
    Vaw2aK0DMdxTRW8nebSSottLs&ust=1711964109888000&sour
    ce=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCJj0
    07eZnoUDFQAAAAAdAAAAABAE'],
    //     price : 20,
    //     quantity: 1
    // ]])
  })()
}, [])

return (
  <MyContext.Provider value={{ loggedInUser,
setloggedInUser }}>
    /* Components start here */
    <Router>
      <Switch>

        <Route path="/register" component={Register}
/>

        <Route path="/login" component={Login} />

```

```

    <Route path="/sell" component={Sell} />
    <Route path="/donate" component={Donation} />
    <Route path="/" component={Home} />
    { /* <Route path="/home" component={Home} />
    <Route path="/" component={Login} /> */ }

  </Switch>
  </Router>
</MyContext.Provider>
);
}

export default App;

```

Mycontext.js

```

import { createContext } from 'react';

export const MyContext = createContext("");

```

firebase.js

```

// Import the functions you need from the SDKs you
need

```

```

import { initializeApp } from "firebase/app";
import { getFirestore } from "@firebase/firestore";
// TODO: Add SDKs for Firebase products that you
want to use
//
https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyA5rwodpSFeckEoBM3nQNkkvQd_WrsF-so",
  authDomain: "waste-mang.firebaseio.com",
  projectId: "waste-mang",
  storageBucket: "waste-mang.appspot.com",
  messagingSenderId: "810149910204",
  appId: "1:810149910204:web:051f272dc353690edde2b9"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
export const db = getFirestore(app);

```

App.css

```

.App {
  text-align: center;
}

```

```
.App-logo {
  height: 40vmin;
  pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

.App-link {
  color: #61dafb;
}
```

```
}

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}
```

ViewItem.css

```
button.buy-button {
  width: 100%;
  background-color: #c3ff9e;
  color: #000; /* Text color */
  border: none;
  margin: 8px 8px;
  padding: 10px 20px; /* Adjust padding as needed
*/
  font-size: 16px; /* Adjust font size as needed
*/
  border-radius: 8px;
  cursor: pointer;
  outline: none; /* Remove default focus styles */
}
```

```
    transition: background-color 0.3s ease; /*
Smooth transition for color change */
}

    button.buy-button:hover {
        background-color: #a0f46e; /* Change color on
hover */
    }

    button.buy-button:active {
        background-color: #7ddb3f; /* Change color on
click */
    }

.view-main{
    display: flex;
    height: fit-content;
    border: 1px solid black;
    margin: 64px auto;
    background-color: #f7f7f7;
    width: 90%;
    border-radius: 8px;
}

body{
    background-color: #f7f8e2;
}
```

```
.image-view{
  margin: 64px 8px;
  width: 60%;
  height: auto;
}

.detail-view
{
  padding: 32px 32px;
  text-align: left;
  /* border: 1px solid black; */
  /* display: inline; */
  width: 39%;
  margin: 8px 8px;
}

img
{
  object-fit: fill;
}

.border
{
  border-top: 1px solid black;
  /* border-style:groove */
}

.border:last-of-type{
```



```
border-bottom: 1px solid black;
}

.h2{
color: #a9a9a9;
}
```

OrderComplete.css

```
.order-card {
background-color: #fff;
box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
border-radius: 8px;
padding: 16px;
margin: 16px;
align-items: center; /* Center items vertically
*/
}

.checkmark-image {
width: 240px; /* Adjust width as needed */
height: 240px; /* Adjust height as needed */
margin-right: 12px; /* Add some margin between
the image and text */
}

.order-title {
font-size: 1.5rem;
```

```
    font-weight: 800;
    color: #000;
}

.order-message {
    font-size: 1rem;
    color: #333;
}

.order-points {
    font-size: 1rem;
    color: #2196f3;
}

.continue-button {
    background-color: #2196f3; /* Use Material
Design primary color for button */
    color: #fff;
    border: none;
    border-radius: 4px;
    padding: 12px 24px;
    font-size: 1rem;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

.continue-button:hover {
    background-color: #1976d2; /* Darken button
color on hover */
```

```
}
```

Nav.css

```
.nav {  
    height: max(64px, 10%);  
    width: 100%;  
    background-color: #2196f3; /* Change to Material  
Design color */  
    color: #fff; /* Change text color to white */  
    font-family: "Roboto", sans-serif; /* Use Roboto  
font for Material Design */  
    font-size: 16px; /* Base font size */  
}  
  
.nav-appname {  
    font-weight: 500; /* Reduce font weight for  
better readability */  
    font-size: 1.5rem; /* Increase font size for  
prominence */  
    margin: 8px; /* Add some margin for better  
spacing */  
}  
  
.nav-upper {  
    display: flex;
```

```
    align-items: center; /* Align items vertically
*/
    justify-content: space-between;
    padding: 8px; /* Add padding for better spacing
*/
}

.nav-lower {
    width: 80%;
    margin: 0 auto;
    display: flex;
    align-items: center;
    justify-content: space-evenly;
    padding: 8px;
    background-color: #42a5f5;
    opacity: .9;
}

button {
    background-color: #64b5f6;
    border: none;
    border-radius: 4px;
    padding: 8px 16px;
    margin: 0 8px;
    color: #fff;
```

```
    cursor: pointer;
    font-size: 1rem;
    transition: background-color 0.3s;
}

button:hover {
    background-color: #42a5f5;
}

a{
    color: #fff;
}
```

ItemCard.css

```
.item-card {
    border: 1px solid #bdbdbd; /* Use Material
Design gray for border */
    width: fit-content;
    text-align: left;
    margin: 8px 8px;
    border-radius: 8px;
    background-color: #fff; /* Use white background
*/
```

```
        box-shadow: 0px 2px 4px rgba(0, 0, 0, 0.1); /*
Add shadow for elevation */
    }

    .text-items {
        padding: 16px; /* Increase padding for better
spacing */
    }

    .price {
        font-size: 1.25rem; /* Increase font size */
        font-weight: 500; /* Use medium font weight */
        color: #2196f3; /* Use Material Design primary
color for price */
    }

    .name {
        font-weight: 500; /* Use medium font weight */
        color: #000; /* Use black for name */
    }

    .username {
        font-style: italic;
        color: #616161; /* Use Material Design gray for
username */
    }
}
```

```

}

a {
    text-decoration: none;
    color: #000; /* Use black for link */
}

a:hover {
    text-decoration: underline; /* Underline link on
hover */
}

.card-img{
    width: 252px;
}

```

BuyItemList.css

```

.buy-items-list {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-around;
    padding: 16px; /* Add some padding */
}

.item-card {
    border: 1px solid #bdbdbd; /* Use Material
Design gray for border */
}

```

```
    width: 250px; /* Set a fixed width for each item
card */
    margin: 16px; /* Add some margin */
    border-radius: 8px; /* Add border radius for
rounded corners */
    background-color: #fff; /* Use white background
*/
    box-shadow: 0px 2px 4px rgba(0, 0, 0, 0.1); /*
Add shadow for elevation */
}

.text-items {
    padding: 16px; /* Increase padding for better
spacing */
}

.price {
    font-size: 1.25rem; /* Increase font size */
    font-weight: 500; /* Use medium font weight */
    color: #2196f3; /* Use Material Design primary
color for price */
}

.name {
    font-weight: 500; /* Use medium font weight */
    color: #000; /* Use black for name */
}

.username {
```



```

    font-style: italic;
    color: #616161; /* Use Material Design gray for
username */
}
a {
    text-decoration: none;
    color: #000; /* Use black for link */
}
a:hover {
    text-decoration: underline; /* Underline link on
hover */
}

```

#### PaymentForm.js

```

import React, { useState } from 'react';
import { Elements } from '@stripe/react-stripe-js';
import { loadStripe } from '@stripe/stripe-js';
import { CardElement, useElements } from
 '@stripe/react-stripe-js';

const YOUR_PUBLISHABLE_KEY =
'pk_test_51K3hkxSI1I85igSuNW4xGWxwV7nDsPILOIT2EUTqv
UDMiP9yM2Jqm3GfBRnyuzSG50MWxyV7tXHdvGUw9pxL4iY000Wd
qdEQWV'; // Replace with your actual key
const PaymentForm = () => {

```

```
    const [cardElement, setCardElement] =
useState(null);

    const [paymentError, setPaymentError] =
useState(null);

    const elements = useElements();

    const stripePromise =
loadStripe(YOUR_PUBLISHABLE_KEY);

    const handleSubmit = async (event) => {
        event.preventDefault();

        if (!cardElement) {
            return;
        }

        const { paymentIntent, error } = await
stripePromise.confirmCardPayment(

'http://localhost:5000/your-backend-confirmation-ro
ute', // Replace with your actual backend route
        {
            payment_method: {
                card: cardElement,
            },
        }
    );

    if (error) {
        setPaymentError(error.message);
    }
}
```

```

        return;
    }

    console.log('Payment successful!',
paymentIntent); // Handle on frontend (replace with
logic)
    };

    return (
        <Elements stripe={stripePromise}>
            <form onSubmit={handleSubmit}>
                <div role="alert">{paymentError} &&
<p>{paymentError}</p></div>
                <CardElement options={{ style: { base: {
fontSize: '16px' } } }} onReady={(element) =>
setCardElement(element)} />
                <button type="submit"
disabled={!cardElement}>
                    Pay
                </button>
            </form>
        </Elements>
    );
};

export default PaymentForm;

```

StripeData.js

```
import { useStripe } from
"@stripe/react-stripe-js";
import { loadStripe } from "@stripe/stripe-js";

export async function makePayment(products) {
  const stripe = await
loadStripe('pk_test_51K3hkxSI1I85igSuNW4xGWxwV7nDsP
ILOIT2EUTqvUDMiP9yM2Jqm3GfBRnyuzSG50MWxyV7tXHdvgUw9
pxL4iY000WdqdEQWV')
  const body = {
    products
  }
  const headers = {
    "content-type": "application/json"
  }
  console.log('making payment')
  const response = await
fetch('http://localhost:5000',{
    method: 'POST',
    headers,
    body : JSON.stringify(body)
  })
  const session = await response.json()
  stripe.redirectToCheckout({
    // successUrl: 'http://localhost:3000',
```

```
        sessionId: session.id
    })
}
```

database.js

```
import {setDoc,query, collection,where, updateDoc,
getDocs, addDoc, deleteDoc, doc} from
'firebase/firestore'
import { db } from '../firebase.js'

export async function registerUser(user){
    try {

        return await setDoc(doc(db, "users",
user.email), user, { merge: true });
    } catch (err) {
        return new Error("not able to add user to
firestore: ", err);
    }
}
```

```
export async function addEntity(collectionName,
data){
    return
addDoc(collection(db,collectionName),data)
}

export async function
getAllEntities(collectionName){

    const snapshots = await
getDocs(query(collection(db,collectionName)));

    const documents = []
    snapshots.forEach(snapshot => {
        const data = { id: snapshot.id ,
...snapshot.data() }
        documents.push(data)
    });
    return documents
}

// updateCriteria = {key == value}, data is {} data
to be updated
export async function updateEntity(collectionName,
data, updateCriteria){
```

```
const key = Object.keys(updateCriteria)[0]
const value = updateCriteria[key]
const q = query(collection(db, collectionName),
where(key, "==", value));
const querySnapshot = await getDocs(q);
  querySnapshot.docs.map(async (doc) => {

    await updateDoc(doc.ref, data)

  });
}
```

```
export async function deleteEntity(collectionName,
deleteCriteria){
  const key = Object.keys(deleteCriteria)[0]
  const value = deleteCriteria[key]
  const q = query(collection(db, collectionName),
where(key, "==", value));
  const querySnapshot = await getDocs(q);
    querySnapshot.docs.map(async (doc) => {

      await deleteDoc(doc.ref)

    });
}
```

## Donation.css

```
/* Donation.css */

.organization-view {
  display: flex;
  flex-direction: column;
  align-items: center;
  width: 100%;
  max-width: 500px; /* Adjust max-width as needed */
  margin: 20px;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 8px;
}

img {
  width: 150px; /* Set image width */
  height: 150px; /* Set image height */
  object-fit: cover;
  border-radius: 8px;
  margin-bottom: 10px; /* Spacing between image
and info */
}
```



```
.info {  
    flex: 1;  
    text-align: center;  
}  
  
h3 {  
    margin-bottom: 10px;  
}  
  
button {  
    padding: 8px 16px;  
    background-color: #007bff;  
    color: #fff;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
}  
  
button:hover {  
    background-color: #0056b3;  
}
```

Login.css

```
.login-container {  
    display: flex;
```

```
justify-content: center;
align-items: center;
min-height: 100vh;
}

.login-form {
width: 300px;
padding: 20px;
border: 1px solid #ccc;
border-radius: 5px;
background-color: #f9f9f9;
}

.form-group {
margin-bottom: 15px;
}

label {
display: block;
margin-bottom: 5px;
}

input[type="email"],
input[type="password"] {
width: 100%;
```

```
padding: 8px;
border-radius: 3px;
border: 1px solid #ccc;
}

button {
  width: 100%;
  padding: 10px;
  border: none;
  border-radius: 3px;
  background-color: #E3E7A0;
  color: black;
  cursor: pointer;
}

button:hover {
  background-color: #0056b3;
}
```

Sell.css

```
.product-form-container {
  width: 400px;
  margin: 20px auto;
  background-color: #FEFAE0; /* Grey background */
  padding: 20px;
```

```
border-radius: 8px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.product-form {
  display: grid;
  gap: 10px;
}

.form-group {
  display: grid;
  gap: 5px;
}

label {
  font-weight: bold;
}

input[type="text"],
input[type="number"],
textarea,
select {
  width: 100%;
  padding: 8px;
  border: 1px solid #ccc;
```

```
    border-radius: 4px;
    box-sizing: border-box;
}

button {
    width: 100%;
    padding: 10px;
    background-color: #CCD5AE;
    color: black;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

button:hover {
    background-color: #E3E7A0;
}
```

Donation.jsx

```
// OrganizationView.js

import React from 'react';
import '../css/Donations.css'; // Import CSS file
for styling
```

```

const Donation= ({ imageUrl, itemName,
onDonateClick }) => {
    return (
        <div className="organization-view">
            <img
src={"https://letsenhance.io/static/8f5e523ee6b2479
e26ecc91b9c25261e/1015f/MainAfter.jpg"}
alt="Organization" />
            <div className="info">
                <h3>{itemName}</h3>
                <div><button
onClick={onDonateClick}>Donate</button></div>
            </div>
        </div>
    );
}

export default Donation;

```

ButItemList.jsx

```

import React, { useState } from "react";
import "../styles/BuyItemsList.css";
import ItemCard from "../ItemCard";
import { makePayment } from "../stripe/stripedata";

```

```

const BuyItemsList = () => {
  const [price, setPrice] = useState(200)
  const handlePay = async (event) => {
    await makePayment([
      {
        name : "sample",
        images :
['https://www.google.com/url?sa=i&url=https%3A%2F%2Fp
ixabay.com%2Fimages%2Fsearch%2Fnature%2F&psig=AOv
Vaw2aK0DMdxTRW8nebSSottLs&ust=1711964109888000&sour
ce=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCJj0
07eZnoUDFQAAAAAdAAAAABAE'],
        price : price,
        quantity: 1
      }
    ])
  }
  return (
    <>
    <div onClick={handlePay}>
      <ItemCard
        name="Ceramic White Vase"
        userName="vasefuluser"
        imageURI={
"https://plus.unsplash.com/premium_photo-1681382743

```

```

881-6afac0a7a556?q=80&w=2500&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D"
    }
    price={price}
  />
</div>
</>
);
};

export default BuyItemsList;

```

## DonationSuccess

```

import React from "react";

const DonationSuccessPage = ({earnedPoints}) => {
  return (
    <div className="order-card">
      <h1 className="order-title">Donation
Success</h1>
      <img
src="https://cdn-icons-png.flaticon.com/512/14900/14900771.png"

```



```

        alt="Checkmark"
        className="checkmark-image"
    />
    <p className="order-message">
        Your item will be delivered to your address.
    </p>
    <p className="order-points">You earned
{earnedPoints} points.</p>
    <button
className="continue-button">Continue</button>
</div>
);
};

export default DonationSuccessPage;

```

Home.jsx

```

import Nav from './Nav'
import BuyItemsList from './BuyItemsList'
import { useContext } from 'react';
import { MyContext } from '../MyContext';
export default function Home () {
    const {loggedInUser, setloggedInUser} =
useContext(MyContext);

```

```

return (
  <>
    <Nav />
    <BuyItemsList />
  </>
)
}

```

ItemCard.jsx

```

import React from "react";
import "../styles/ItemCard.css";
const ItemCard = ({ name, userName, imageURI, price
}) => {
  return (
    <div className="item-card">
      { /* <a href=""> */ }
      <div>
        <img className="card-img" src={imageURI}
alt={name} />
      </div>
      <div className="text-items">
        <p className="price">${price}</p>
        <div className="name">{name}</div>
        <div
className="username">@{userName}</div>

```

```

        </div>
        { /* </a> */ }
    </div>
);
};

export default ItemCard;

```

Login.jsx

```

import React from 'react';
import '../css/Login.css';

const Login = () => {
    function handleSubmit(event) {
        event.preventDefault()
        // alert(event.target.email.value)
        const email = event.target.email.value
        const password = event.target.password.value
        window.location.href = '/home';
    }

    return (
        <div className="login-container">
            <form className="login-form"
onSubmit={handleSubmit}>
                <h2>Login</h2>

```

```

        <div className="form-group">
            <label
htmlFor="email">Email</label>
                <input type="email" id="email"
name="email" />
            </div>
            <div className="form-group">
                <label
htmlFor="password">Password</label>
                <input type="password"
id="password" name="password" />
            </div>
            <button type="submit">Login</button>
            <div className="register-link">
                <p>Don't have an account? <a
href="#">Register here</a></p>
            </div>
        </form>
    </div>

    );
}

export default Login;

```

Nav.jsx

```
import React from "react";
import "../styles/Nav.css";
import {Link} from 'react-router-dom';
const Nav = () => {
  return (
    <div className="nav">
      <div className="nav-upper">
        <div className="nav-appname
pad">Marketplace</div>
        <div>
          <div className="pad">
            <input className="pad"
placeholder="Pune"></input>
            <input placeholder="Search for
something..."></input>
          </div>
        </div>
        <div className="pad">
          <button>Profile</button>
          <button> <a href="/sell"></a>
Sell</button>
        </div>
      </div>
      <div className="nav-lower flex">
```

```

        <a>Electronics</a>
        <a>Decor</a>
        <a>Study</a>
        <a>Utiltiy</a>
    </div>
</div>
);
};

export default Nav;

```

OrderComplete.jsx

```

import React from "react";
import "../styles/OrderComplete.css";
const OrderCompletePage = ({ earnedPoints }) => {
    return (
        <div className="order-card">
            <h1 className="order-title">Order
Successful</h1>
            
    <p className="order-message">
      Your item will be delivered to your address.
    </p>
    <p className="order-points">You earned
{earnedPoints} points.</p>
    <button
className="continue-button">Continue</button>
  </div>
);
};

export default OrderCompletePage;

```

#### Register.jsx

```

import React, { useState } from 'react';
import { useHistory } from 'react-router-dom';
import '../css/Register.css';
import { addEntity, registerUser } from
'../datastore/databse';
import { uuidv4 } from '@firebase/util';
import { Redirect } from 'react-router-dom'
const Register = () => {
  const history = useHistory();
  const [redirect, setRedirect] = useState(false);

```

```
const handleSubmit = async (event) => {
  event.preventDefault();
  const email = event.target.email.value;
  const password =
event.target.password.value;
  const role = event.target.role.value;
  const result = await registerUser({
    email,
    password,
    role,
    points: 0,
    created_at: new Date(),
    id: uuidv4()
  });
  // alert(result);
  setRedirect(true);

}

if (redirect) {

  window.location.href = '/home';

}
```



```
    return (
      <div className="login-container">
        <form className="login-form"
onSubmit={handleSubmit}>
          <h2>Register</h2>
          <div className="form-group">
            <label
htmlFor="email">Email</label>
              <input type="email" id="email"
name="email" />
            </div>
            <div className="form-group">
              <label
htmlFor="password">Password</label>
                <input type="password"
id="password" name="password" />
              </div>
              <div className="form-group">
                <label
htmlFor="text">Role</label>
                  <input type="text" id="role"
name="role" placeholder='ngo, user, company' />
                </div>

```

```

        <button
type="submit">Register</button>
        <div className="login-link">
            <p>Already have an account? <a
href="/login">Sign in here</a></p>
        </div>
    </form>
</div>
);
}

export default Register;

```

Sell.jsx

```

import React, { useState } from 'react';
import '../css/Sell.css';

const Sell = () => {
    const [formData, setFormData] = useState({
        name: '',
        category: '',
        sellOrDonate: '',
        price: '',
        imageDescription: '',
        imageFile: null,
    });

```

```
});

const handleChange = (e) => {
  const { name, value } = e.target;
  setFormData({
    ...formData,
    [name]: value,
  });
};

const handleFileChange = (e) => {
  setFormData({
    ...formData, //creates a shallow copy
    imageFile: e.target.files[0],
  });
};

const handleSubmit = (e) => {
  e.preventDefault();
  console.log(formData);
};

return (
  <div className="product-form-container">
    <h2>Product Description Form</h2>
```

```
    <form className="product-form"
onSubmit={handleSubmit}>
    <div className="form-group">
        <label>Name:</label>
        <input type="text" name="name"
value={formData.name} onChange={handleChange} />
    </div>
    <div className="form-group">
        <label>Category:</label>
        <input type="text"
name="category" value={formData.category}
onChange={handleChange} />
    </div>
    <div className="form-group">
        <label>Sell or Donate:</label>
        <select name="sellOrDonate"
value={formData.sellOrDonate}
onChange={handleChange}>
            <option
value="">Select</option>
            <option
value="sell">Sell</option>
            <option
value="donate">Donate</option>
        </select>
    </div>
</form>
```

```
        </div>
        {formData.sellOrDonate === 'sell' &&
(
            <div className="form-group">
                <label>Price:</label>
                <input type="number"
name="price" value={formData.price}
onChange={handleChange} />
            </div>
        ) }
        <div className="form-group">
            <label>Description of
Image:</label>
            <textarea
name="imageDescription"
value={formData.imageDescription}
onChange={handleChange}></textarea>
        </div>
        <div className="form-group">
            <label>Upload Image:</label>
            <input type="file"
accept="image/*" onChange={handleFileChange} />
        </div>
        <button type="submit">Post</button>
    </form>
```

```

        </div>

    );
}
export default Sell;

```

ViewItem.jsx

```

import React from "react";
import
"react-responsive-carousel/lib/styles/carousel.min.
css"; // requires a loader
import { Carousel } from
"react-responsive-carousel";
import "../styles/ViewItem.css";
const ViewItem = ({ name, price, userName,
description, imageUrl }) => {
    return (
        <>
            <h2>View Item</h2>
            <div className="view-main">
                <div className="image-view">
                    <Carousel showThumbs={false}>
                        {imageUrl.map((imageUrl, index) => (
                            <div key={index}>
                                <img src={imageUrl} alt={`Image
${index + 1}`} />

```

```

        { /* <p className="legend">{ `Image
${index + 1}` }</p> */}
    </div>
    )}
</Carousel>
</div>
<div className="detail-view">
    <div className="item-details border">
        <h2>{name}</h2>
        <h6>{userName}</h6>

        <p>{description}</p>
    </div>
    <div className="border">
        <p className="price">${price}</p>
        <button
className="buy-button">Buy</button>
    </div>
</div>
</div>
</>
);
};

export default ViewItem;

```

## BACKEND

index.js

```
const express = require('express')
const {json,urlencoded} = require('express')
const cors = require('cors')
const stripe =
require('stripe')('sk_test_51K3hkxSI1I85igSuXmBcQBw
hB2W1PkaoultEhi6HMT1E5oT6tylGqj47NWxT4zjJOLBruOMQG1
kZmeoz5savWH2c00q2w2RrZb'); // Replace with your
Secret key

const app = express();
app.use(cors())
app.use(json());
app.use(urlencoded({ extended: true }));

app.post('/', async (req, res) => {
  try {
const {products} = req.body ;

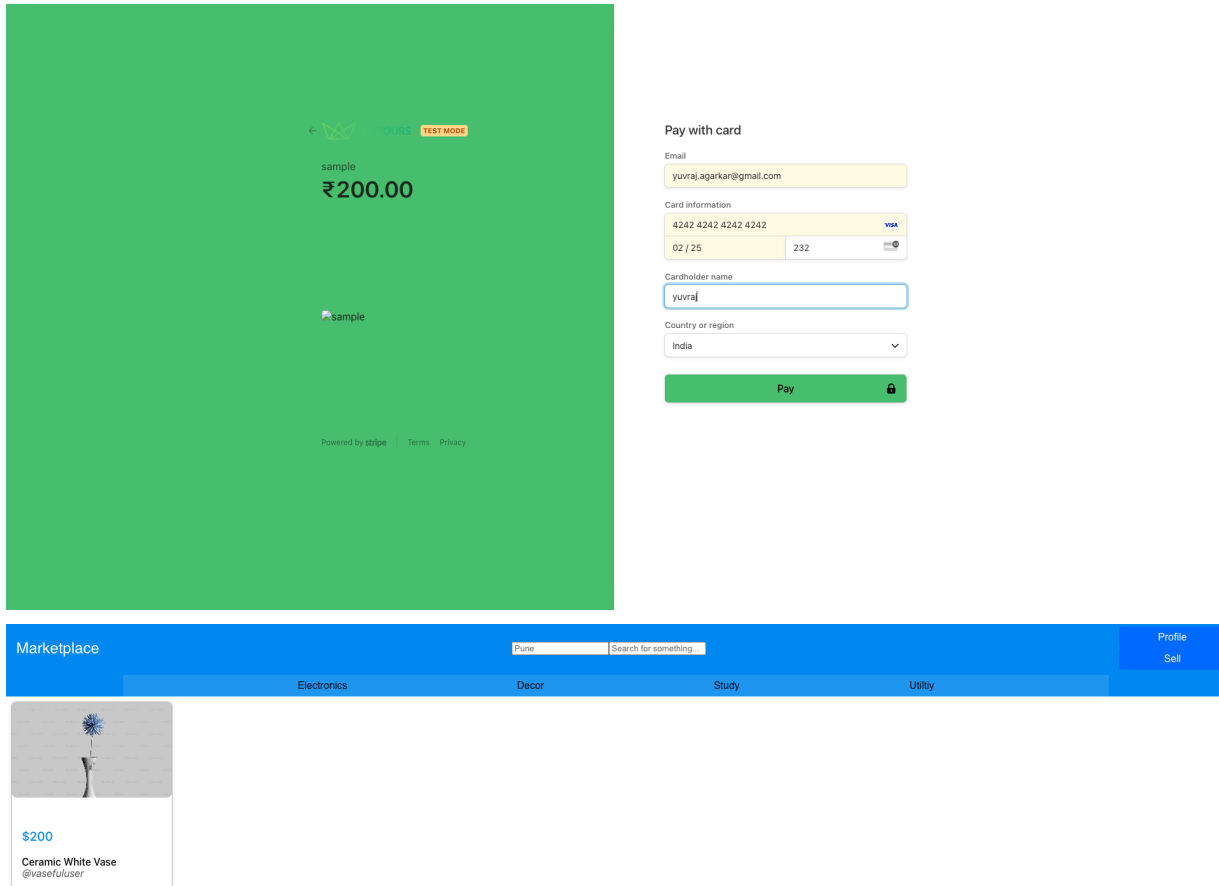
    const lineItems = products.map((product) => ({
      price_data : {
        currency: "inr",
```

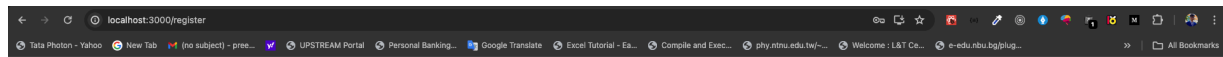


```
        product_data : {
            name : product.name ,
            images : [...product.images]
        },
        unit_amount: Math.round(product.price*
100 )
    } ,
    quantity: product.quantity
}))
console.log('line items>>>', lineItems)
const session = await
stripe.checkout.sessions.create({
    payment_method_types : ["card"],
    line_items : lineItems ,
    mode: "payment",
    success_url: "http://localhost:3000/home",
    cancel_url : "http://localhost:3000"
})
res.json ({id : session.id})
} catch (error) {
    console.error(error);
    res.status(500).json({ message: 'Payment
failed!' });
}
});
```

```
app.listen(5000)
```

## OUTPUT





### Register

Email

admin

Password

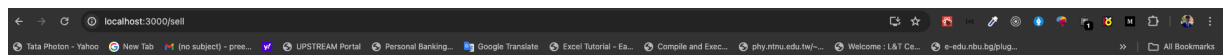
\*\*\*\*

Role

ngo, user, company

Register

Already have an account? Sign in here



### Product Description Form

Name:

Category:

Sell or Donate:

Select

Description of Image:

Upload Image:

Choose file | No file chosen

Post