

CHAMELI DEVI GROUP OF INSTITUTIONS INDORE (M.P.)



Rental Zone Mini Project Report

CS604- Project Management

Guided By:

Ms. Madhu Sharma
Asst. Professor, CSE Dept.
CDGI Indore

Submitted By:

Mr. Hatim Saif Press Wala
0832CS191077
Mr. Karun Mourya 0832CS191088
Mr Khushhal Gupta 0832CS191089
Mr. Murtaza Barwahawala 0832CS191110

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHAMELI DEVI GROUP OF INSTITUTIONS INDORE (M.P.)



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that Mr. Hatim Saifee Press wala , Mr. Karun Mourya , Mr. Khushhal Gupta, Mr. Murtaza Barahawala with RGTU Enrollment No. 0832CS191074, 0832CS191088, 0832CS191089, 0832CS191110 have satisfactorily completed the Mini Project on Rental Zone in “**CS604-Project Management**”, for **B. Tech, VI Semester** of the **Computer Science & Engineering** during year **2021 – 22**.

Prof. Shailendra Kumar Mishra
Head of the Department

Signature of
Faculty In-charge

TABLE OF CONTENTS

CONTENTS	PAGE NO.
Introduction of the Project	1
Description of the Project	2
Technical Details of the Project	3-4
System Design	5-8
Screenshot of the Code (Min. 3)	8-10
Screenshot of working Project (Min. 5)	10-12
References	13

Introduction of the Project

RENTISH is a website that will provide rental service for real estate property, vehicles, household appliances, furniture, electrical products etc.

On this website people who want to rent or want to explore the possibility of renting will visit. When they come they will land on the home page. where they will get to explore all the items they can rent. There will be an option to search where the user will get to search items according to their need. There will be a login or register page where new users will register themselves. Users will have to register themselves by filling in all their detail. From the home page, users can go to different sections of the fields which they want to explore. Their user can see all the details about the product with images. When the user finds the product they will be able to click on the rent button. From there they will be redirected to the rent page of that product which will show all product details with product ID and payment summary. On that page, users will have to mention the duration for which they want to rent the item. According to which their payable amount will be shown with deposit.

From there users can go back to exploring more suitable items or can click on confirm button. which will take them to a payment page where they will be able to select the payment method they prefer. After payment confirmation, they will land on confirm page.

After renting the item and using it the person will be able to go to the order menu and will be able to return it by clicking on the return button. If the rent duration ends and the user does not return the item he will be charged per day for it.

Users will be able to give feedback on the order by going on the feedback option on the item they have already rented. They will find the feedback option when he opens the product he already rented in my orders.

Description of the Project

Renting items have become an important factor in modern society hence the need to have a website for it. In India, everyone tries to find a way to save money By taking the best he can at least cost. Every Indian goes for the most efficient choice. And renting is most efficient for short period.

Study of already available open-source websites developed in metro cities and foreign countries. There are many other renting websites which give cars, bikes and other gives furniture and few gives appliances. we will study all these websites and give an advance combination in rentish. So we will Analyse the foreign website and modify it according to our local needs.

We will try to develop an efficient website that will be helpful to users and will bring out the best in the technical renting world.

In our project, we will be using Agile methodology. In agile methodology, we do development and testing simultaneously. Agile methodologies attempt to produce the proper product through small cross-functional self-organizing teams that produce small pieces of functionality regularly, allowing for frequent customer input and course correction as needed.

Technical Details of the Project

Hardware Interface:

Development End

Hardware Specifications (Recommended):

Processor : i3 Processor

Storage : 10 GB

RAM : 4 GB

Software Specifications (Recommended):

Operating System : Windows 10

Browser : Chrome Browser Version 96.0

Designing Tool : VS Code

Front End : React Version 17.0.2

Back End : Express.js Version 4.17.1, Node.js Version 16.13.0

Database : MongoDB Version 4.4

Deployment End

Server Side

Hardware Specifications(Recommended):

Processor : Quad Core Processor

Storage : 50 GB

RAM : 4 GB

Software Specifications(Recommended):

Operating System : Windows 10

Browser : Chrome Version 96.0

DBMS : MongoDB Version 4.4

Client Side

Hardware Specifications(Recommended):

Processor : Dual Core Processor

Storage : 10 GB

RAM : 512 MB

Software Specifications(Recommended):

Operating System : Windows 10

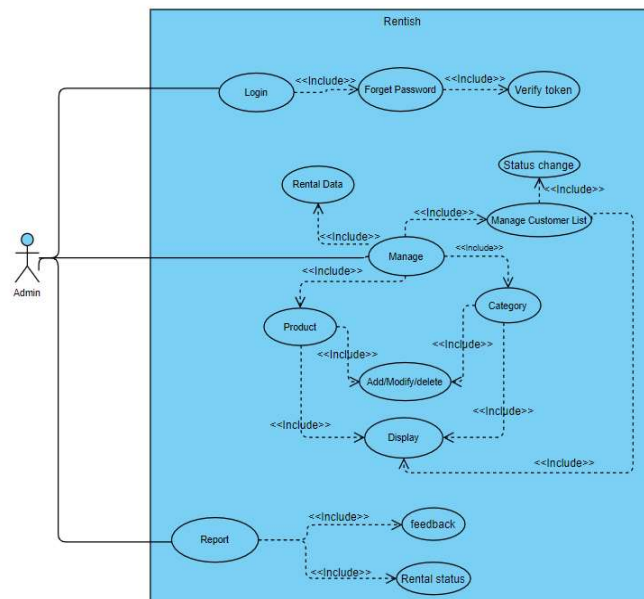
Browser : Chrome Browser 96.0

System Design

1 Use Case Diagram

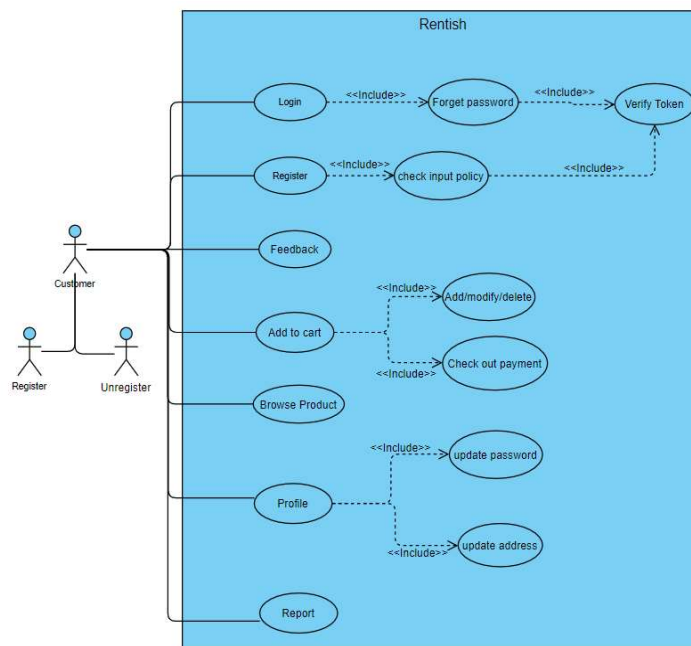
a) Admin

Fig 1.1 : Use case (Admin)



b) Customer

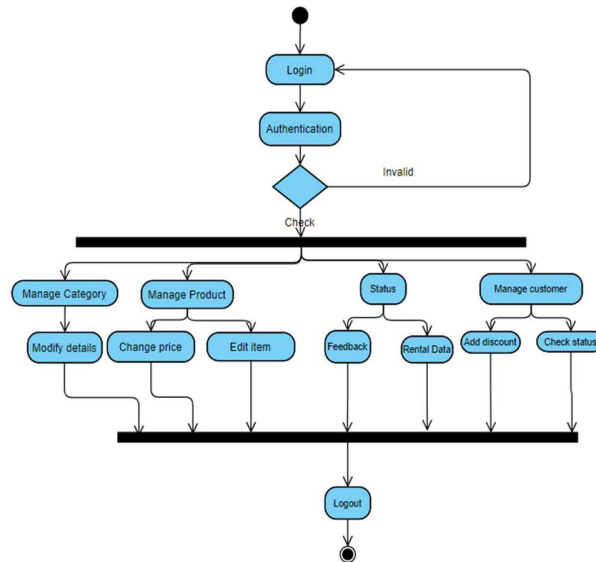
Fig 1.2 : Use case (Customer)



2 Activity Diagram

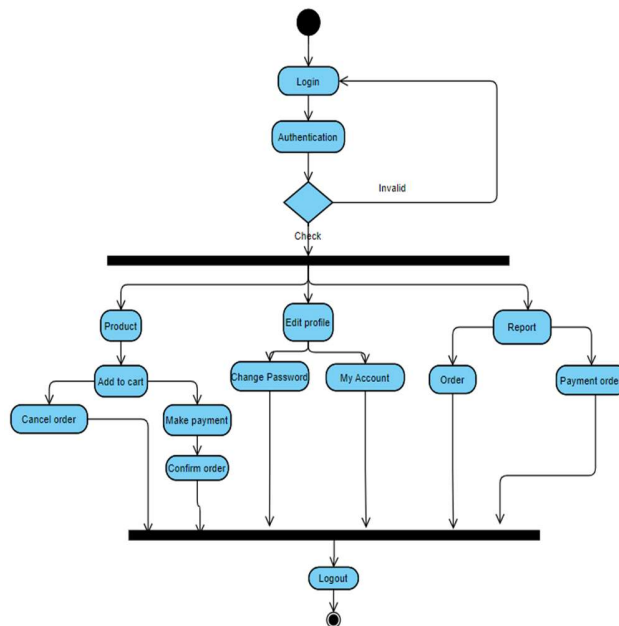
a) Admin

Fig 2.1 : Activity (Admin)



b) Customer

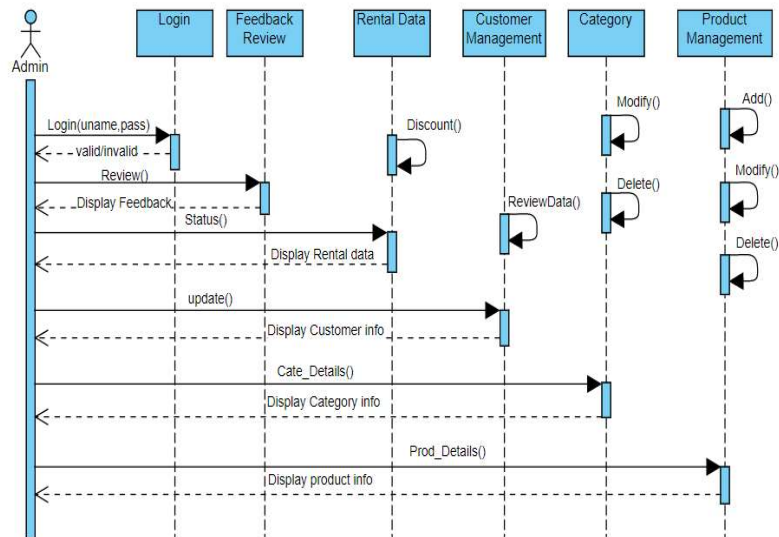
Fig 2.2 : Activity (Customer)



3 Sequence Diagram

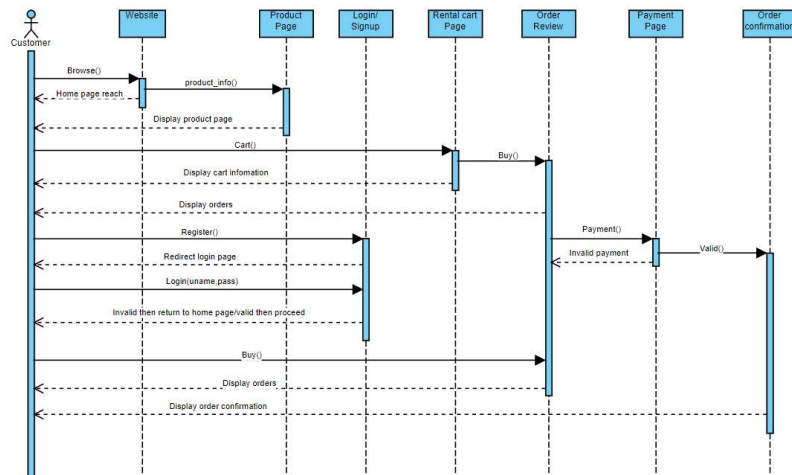
a) Admin

Fig 3.1 : Sequence Diagram (Admin)



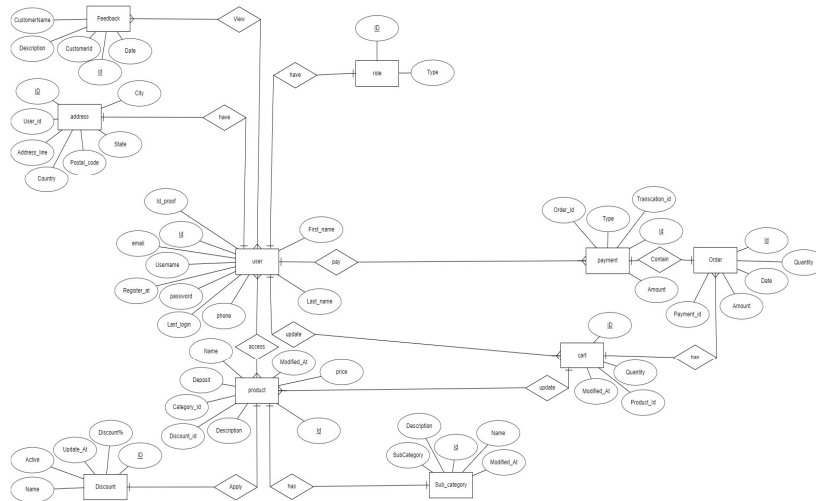
b) Customer

Fig 3.2 : Sequence Diagram (Customer)



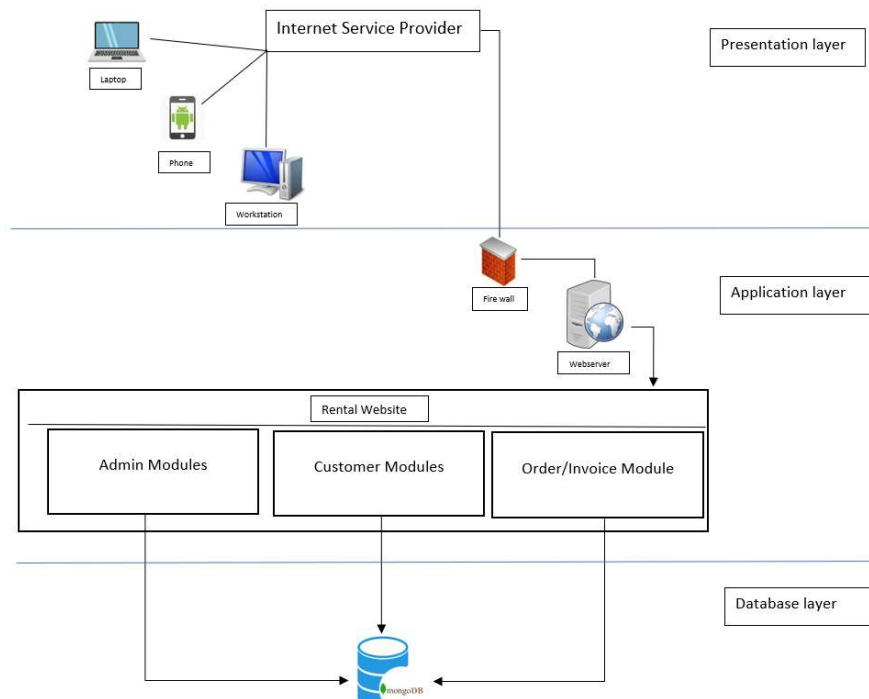
4 ER Diagram

Fig 4.1 : ER Diagram



5 System Architecture

Fig 5.1 : System Architecture Diagram



Screenshot of the Code

The screenshot shows the Visual Studio Code editor with the 'App.js' file open. The Explorer panel on the left shows the project structure for 'RENTISH_FRONTEND', including 'node_modules', 'public', 'src', 'App', 'components', 'layout', 'pages', 'redux', 'routes', and 'validation'. The 'App.js' file is selected in the Explorer. The main editor area displays the code for 'App.js', which imports 'BasicRoutes' from '../routes/basicRoutes', 'BrowserRouter' from 'react-router-dom', 'LoginLayout' from '../layout/LoginLayout', 'Provider' from 'react-redux', and 'store' from '../redux/store'. The code defines a functional component 'App' that returns a JSX element with a 'div' containing a 'Provider' wrapping a 'BrowserRouter' which contains a 'LoginLayout' and 'BasicRoutes'. The 'App' component is exported as the default export.

```
1 import BasicRoutes from "../routes/basicRoutes";
2 import { BrowserRouter } from "react-router-dom";
3 import LoginLayout from "../layout/LoginLayout";
4 import { Provider } from "react-redux";
5 import store from "../redux/store";
6
7
8 const App = () => {
9   return (
10     <div className="App">
11       <Provider store={store}>
12         <BrowserRouter>
13           <LoginLayout>
14             <BasicRoutes />
15           </LoginLayout>
16         </BrowserRouter>
17       </Provider>
18     </div>
19   );
20 };
21
22 export default App;
```

The screenshot shows the Visual Studio Code editor with the 'BasicLayout.js' file open. The Explorer panel on the left shows the project structure for 'RENTISH_FRONTEND', including 'node_modules', 'public', 'src', 'App', 'components', 'layout', 'pages', 'redux', 'routes', and 'validation'. The 'BasicLayout.js' file is selected in the Explorer. The main editor area displays the code for 'BasicLayout.js', which imports 'React', 'useState', 'useEffect' from 'react', 'Header' from '../components/Header/Header', and 'Footer' from '../components/Footer/Footer'. The code defines a functional component 'BasicLayout' that takes 'props' as an argument. It uses 'useState' to manage 'user' and 'setUser' to update 'user' from 'localStorage.getItem("user")'. It uses 'useEffect' to set 'user' when the component mounts. The code returns a JSX element with a 'div' containing a 'Header', a 'div' with a 'div' containing 'props.children' and a 'Footer', and a 'Footer'. The 'BasicLayout' component is exported as the default export.

```
1 import React, { useState, useEffect } from "react";
2 import Header from "../components/Header/Header";
3 import Footer from "../components/Footer/Footer";
4
5 const BasicLayout = (props) => {
6   const [user, setUser] = useState("");
7
8   useEffect(() => {
9     | setUser(localStorage.getItem("user"));
10   }, [user]);
11
12   return (
13     <div className="container-scroller">
14       <Header />
15       <div className="page-body-wrapper">
16         <div className="main-panel">
17           <div className="content-wrapper">{props.children}</div>
18         </div>
19       </div>
20       <Footer />
21     </div>
22   );
23 };
24
25 export default BasicLayout;
```

```

1  import React, { useEffect, useState } from "react";
2  import { Carousel } from "react-bootstrap";
3  import Category from "../Category";
4  import "../Dashboard.css";
5  import SlideComponent from "../SlideComponent";
6  import BedIcon from "@mui/icons-material/Bed";
7  import EngineeringIcon from "@mui/icons-material/Engineering";
8  import CancelScheduleSendOutlinedIcon from "@mui/icons-material/CancelScheduleSendOutlined";
9  import { Link } from "react-router-dom";
10 import SwiperComponent from "../SwiperComponent";
11 import axios from "axios";
12 import { setUser } from "../../redux/actions/userActions";
13 import { useDispatch, useSelector } from "react-redux";
14
15 function Dashboard() {
16   const [transition1, setTransition1] = useState(0);
17   const [op1, setOp1] = useState(0.5);
18   const [transition2, setTransition2] = useState(0);
19   const [op2, setOp2] = useState(0.5);
20   const [prod, setProd] = useState([]);
21   const [user, setUser] = useState();
22   const [feedback, setFeedback] = useState([]);
23   const [trigger, setTrigger] = useState(true);
24   const dispatch = useDispatch();
25
26   async function fetchData() {
27     try {
28       const res = await axios.post("/user/updateuser");
29       setUser(res.data.data);
30       const pro = await axios.post("/product");
31       setProd(pro.data.data);
32       const feed = await axios.get("/admin/feedback");
33       setFeedback(feed.data.data);

```

```

31   setProd(pro.data.data);
32   const feed = await axios.get("/admin/feedback");
33   setFeedback(feed.data.data);
34   dispatch(setUser(res.data.data));
35 } catch (error) {
36   const refreshToken = localStorage.getItem("refreshToken");
37   if (error.message === "Request failed with status code 401") {
38     axios.defaults.headers.common["Authorization"] = refreshToken;
39     const result = await axios.post("/revoketoken");
40     localStorage.setItem("accessToken", result.data.data.accessToken);
41     localStorage.setItem("refreshToken", result.data.data.refreshToken);
42   }
43   console.error(error);
44 }
45 }
46
47 useEffect(() => {
48   fetchData();
49 }, [trigger]);
50
51 return (
52   <div className="dashboard">
53     <div className="d-flex align-items-center justify-content-center">
54       <Carousel className="carousel" pause={false} interval={5000}>
55         <Carousel.Item>
56           <img
57             className="d-block w-100 h-25"
58             src={process.env.PUBLIC_URL + "/Images/sample1.jpg"}
59             alt="First slide"
60           />
61         </Carousel.Item>
62       </Carousel>
63     </div>
64   </div>

```


File Edit Selection View Go Run Terminal Help Dashboard.js - Rentish_Frontend - Visual Studio Code

EXPLORER

- RENTISH_FRONTEND
 - node_modules
 - public
 - src
 - App
 - App.js
 - components
 - Alert
 - Auth
 - Footer
 - Header
 - layout
 - pages
 - About
 - Cart
 - Checkout
 - Contact
 - Dashboard
 - Category.js
 - Dashboard.css
 - Dashboard.js
 - SlideComponent.js
 - SwiperComponent.js
 - Forget_Password
 - otp_verification
 - Product
 - Profile
 - Success
 - redux
 - routes

src > pages > Dashboard > JS Dashboard.js > Dashboard

```

93 <div className="slide_nav">
94   <div className="main_head">
95     You'll love to
96     <span>take these home</span>
97   </div>
98   <div className="slide_nav">
99     <div
100       className="swipe_left"
101       role="button"
102       tabIndex="0"
103       aria-label="Previous Slide"
104       onClick={() => {
105         if (transition1 !== 0) {
106           setTransition1(transition1 + 15);
107         }
108         if (transition1 === -15) {
109           setOp1(0.5);
110         }
111       }}
112       style={{ opacity: op1 }}
113     >
114     <i class="fas fa-chevron-right"></i>
115   </div>
116   <div
117     className="swipe_right"
118     role="button"
119     tabIndex="0"
120     aria-label="Next Slide"
121     onClick={() => {
122       setTransition1(transition1 - 15);
123       setOp1(1);
124     }}
125   >
126   <i class="fas fa-chevron-left"></i>

```

File Edit Selection View Go Run Terminal Help Checkout.js - Rentish_Frontend - Visual Studio Code

EXPLORER

- RENTISH_FRONTEND
 - node_modules
 - public
 - src
 - App
 - App.js
 - components
 - Alert
 - Auth
 - Footer
 - Header
 - layout
 - pages
 - About
 - Cart
 - Checkout
 - CheckoutSuccess
 - FormFields
 - FormModel
 - Forms
 - ReviewOrder

src > pages > Checkout > JS Checkout.js > _renderStepContent

```

1 import React, { useState } from 'react';
2 import {
3   Stepper,
4   Step,
5   StepLabel,
6   Button,
7   Typography,
8   CircularProgress
9 } from '@material-ui/core';
10 import { Formik, Form } from 'formik';
11 import MaterialLayout from './MaterialLayout';
12 import AddressForm from './Forms/AddressForm';
13 import PaymentForm from './Forms/PaymentForm';
14 import ReviewOrder from './ReviewOrder/ReviewOrder';
15 import CheckoutSuccess from './CheckoutSuccess/CheckoutSuccess';
16
17 import validationSchema from './FormModel/validationSchema';
18 import checkoutFormModel from './FormModel/checkoutFormModel';
19 import formInitialValues from './FormModel/formInitialValues';
20
21 import useStyles from './Cstyles';
22
23 const steps = ['Shipping address', 'Payment details', 'Review your order'];
24 const { formId, formField } = checkoutFormModel;
25
26 function _renderStepContent(step) {
27   switch (step) {
28     case 0:
29       return <AddressForm formField={formField} />;
30     case 1:
31       return <PaymentForm formField={formField} />;
32     case 2:
33       return <ReviewOrder />;

```

Screenshot of working Project

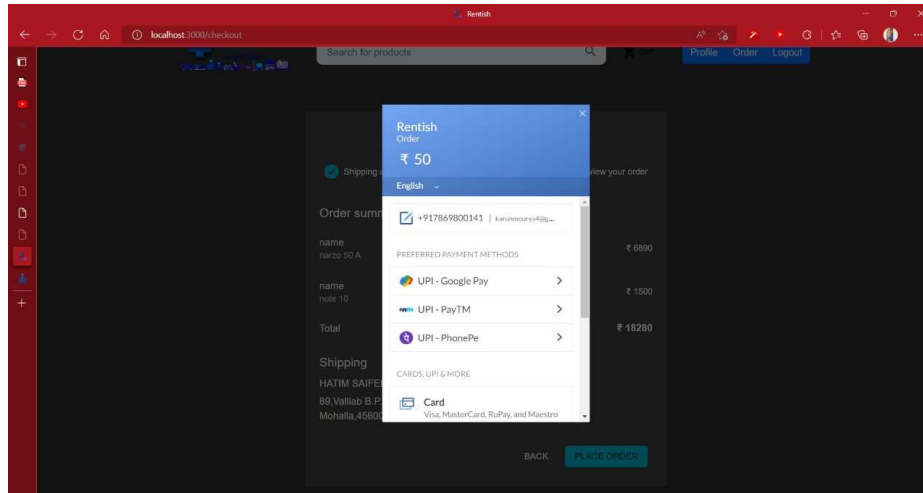


Fig : 4.1.1 Razorpay payment gateway

A screenshot of the 'Checkout' page in the 'Rentish' application. The page is titled 'Checkout' and has two steps: 'Shipping address' (active) and 'Review your order'. The 'Shipping address' section contains several form fields: 'First name*', 'Last name*', 'Address Line 1*', 'Address Line 2', 'City*', 'State/Province/Region', 'Zipcode*', and 'Country*'. There is also a checkbox labeled 'Use this address for payment details'. The page has a dark theme with a red sidebar on the left. The browser's address bar shows 'localhost:3000/checkout'.

Fig : 4.1.2 Checkout Address Form

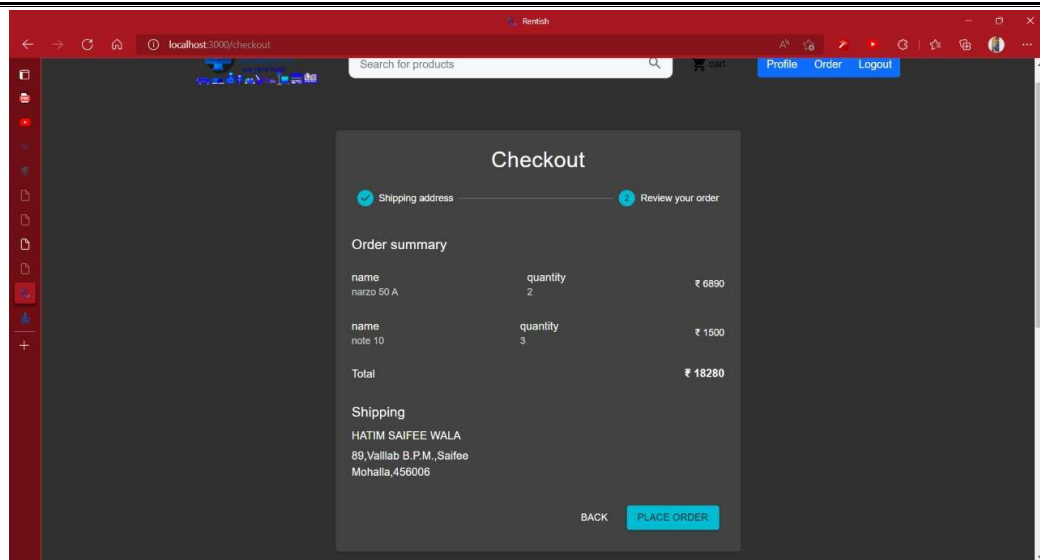


Fig : 4.1.3 Checkuot Order summary page

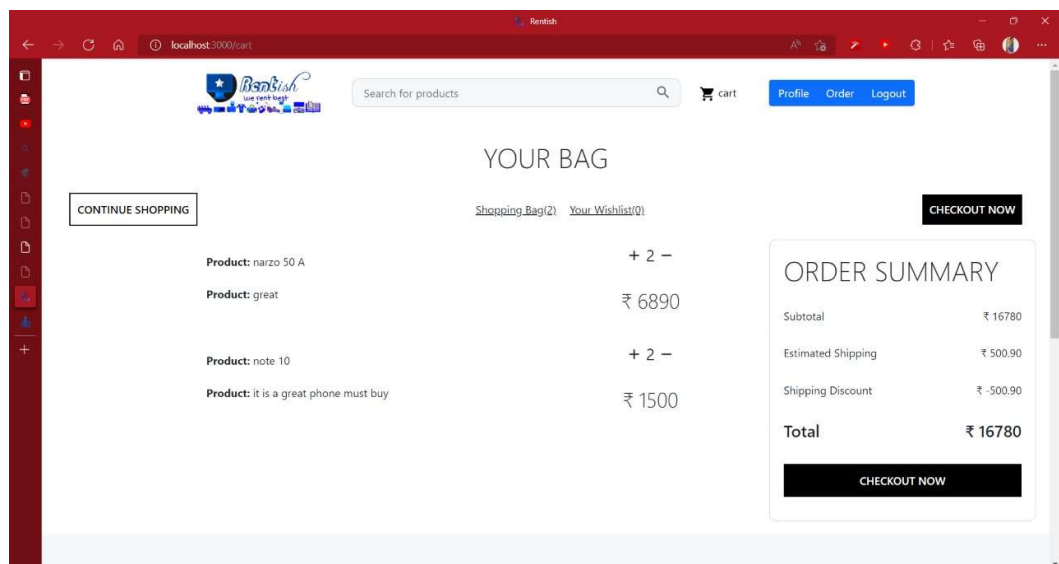


Fig : 4.1.4 Cart Page

localhost:3000/profile

Search for products

Profile Order Logout

Your profile

Personal Details

Name

Email

Phone Number

Address

[Update Details](#)

Change Password

New Password

Confirm New Password

[Update Password](#)

Email Verification

Fig : 4.1.5 Profile Page

localhost:3000/Login

Create Account

[f](#) [G+](#)

or use your email for registration

Full Name

Email

Password

Phone Number

Address

[SIGN UP](#)

Welcome Back!

To keep connected with us please login with your personal info

[Sign In](#)

Fig : 4.1.6 Signup page

References

Book:

- [1] Henry Chan (Author), Raymond Lee (Author), Tharam Dillon (Author) and Elizabeth Chang (Author). “E-Commerce: Fundamentals and Applications”.

Research sites:

- [2] Kieraya Furnishing Solutions Pvt. Ltd. "<https://www.furlenco.com>".
- [3] NoBroker.in "<https://furniture.nobroker.in/>".
- [4] RentMacha "<https://www.rentmacha.com>".

Reference Youtube channel:

- [5] Lama Dev (Youtube) “React Node.js E-Commerce App Full Tutorial (REDUX - Stripe - JWT) - MERN Stack Shopping App”.
- [6] Thapa Technical (Youtube) “MERN Stack Tutorial 2021”.