



CSE370 : Database Systems Project Report

Project Title : Next Generation Bus Ticketing System

Group No : 3 , CSE370 Lab Section : 02 , Fall 2023		
ID	Name	Contribution
22101157	Tanveer Shahriar Arnob	Admin Panel
22101334	Md.Rumman Shahriar	Authentication & ticket options
23341032	A K M Nihalul Kabir	Home Page and Bus Search (Same Page)
22101728	Rabaya Farzana Sikder	Bus driver upcoming bus ride and statistic page
22101273	Md. Salman Pasha	Ticket refund and bus/driver rating

Table of Contents

Section No	Content	Page No
1	Introduction	3
2	Project Features	3
3	ER/EER Diagram	4
4	Schema Diagram	5
5	Frontend Development	6
6	Backend Development	23
7	Conclusion	39
8	References	39

Introduction

Introducing the Next Generation Bus Ticketing System: Your One-Stop Shop for Effortless Travel

Get ready to experience the future of travel with our revolutionary next-generation ticketing system website. Our platform offers a comprehensive suite of features designed to simplify the entire travel booking process for both passengers and transportation providers.

Project Features

Here's a glimpse into the project's features:

For Passengers:

- Seamless Booking: Search and book your bus tickets with ease from the comfort of your home.
- Easy Authentication: Securely register and verify your identity for a smooth travel experience.
- Ticket Confirmation: Receive instant confirmation of your booking with all relevant details.
- Ticket Refund: Request and manage ticket refunds effortlessly through the website.
- Bus Search: Find the perfect bus ride and route with our advanced search filters.
- Driver and Passenger Authentication: Ensure a secure travel environment with verified identities.

For Bus Drivers:

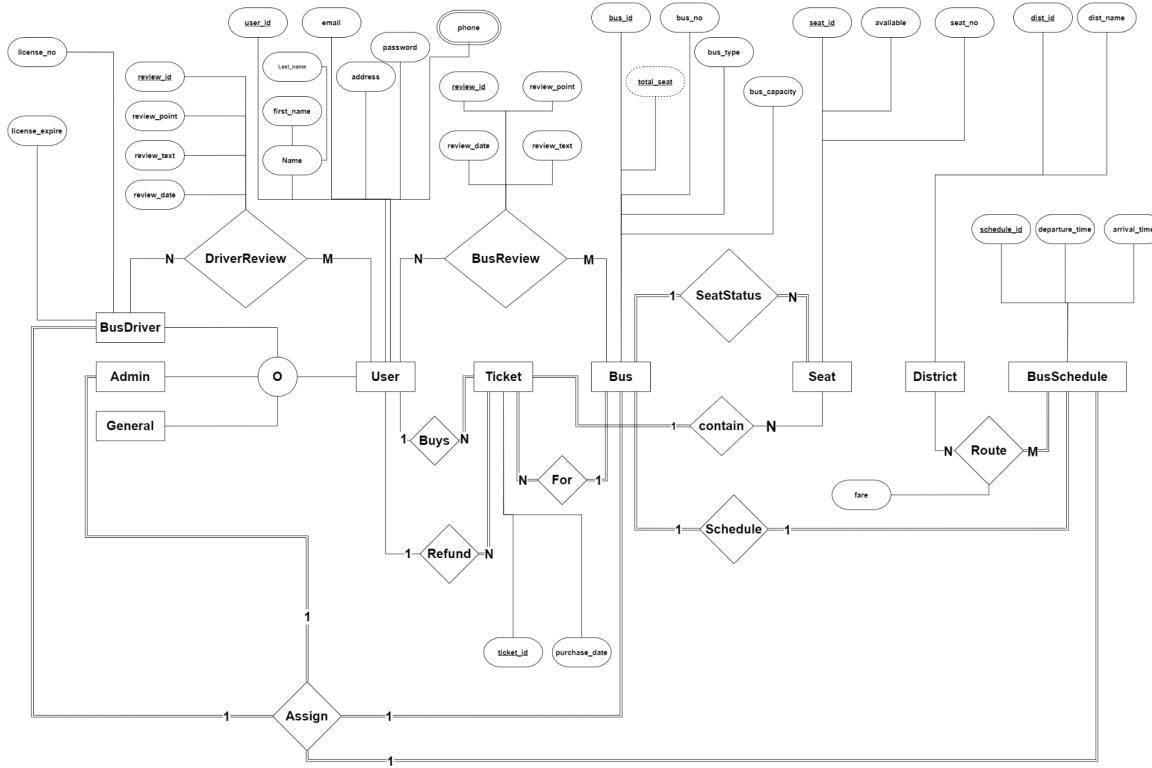
- Upcoming Bus Rides: View your assigned bus rides with detailed information about schedules and passengers.

For Admins:

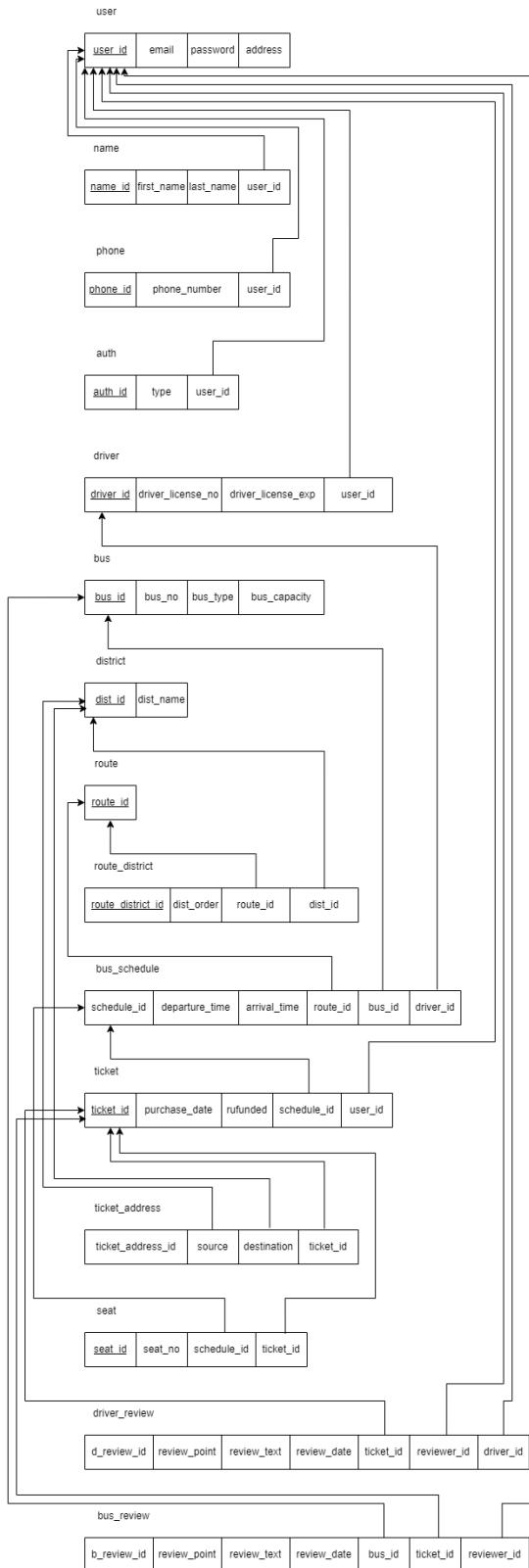
- Comprehensive Admin Panel: Control every aspect of the platform, manage users, fares, and schedules.
- Detailed Statistics: Gain valuable insights into user behavior and platform performance to select the route.

Our Next Generation Ticketing System is more than just a platform; it's a complete service.

ER/EER Diagram



Schema Diagram



Frontend Development

Contribution of ID : 22101157, Name : Tanveer Shahriar Arnob

1. Admin Panel

```
 1 import React, { useContext, useEffect } from "react";
 2 import { UserToken } from "../../Token/UserToken";
 3 import { useNavigate } from "react-router-dom";
 4 import "./adminPanel.css";
 5
 6 function AdminPanel() {
 7   const { authorised, userId, userType } = useContext(UserToken);
 8   let navigate = useNavigate();
 9
10   useEffect(() => {
11     if (authorised === "false") {
12       navigate("/login");
13     }
14     if (userType !== "admin") {
15       navigate("/home");
16     }
17   }, [authorised, userType]);
18
19   const handleAddNewBus = () => {
20     navigate("/addBus");
21   };
22   const handleAddNewRoute = () => {
23     navigate("/addRoute");
24   };
25
26   const handleAddNewSchedule = () => {
27     navigate("/addSchedule");
28   };
29
30   const handleAddNewDriver = () => {
31     navigate("/addDriver");
32   };
33
34   return (
35     <div className="flex justify-center items-center h-screen background_image_admin">
36       <div className="bg-white bg-opacity-25 rounded-lg p-6">
37         <button
38           className="w-full block bg-black hover:bg-gray-900 text-white border-2 border-white text-center py-4 px-20 rounded-lg font-bold"
39           onClick={handleAddNewDriver}
40         >
41           Add Driver
42         </button>
43
44         <button
45           className="mt-2 w-full block bg-black hover:bg-gray-900 text-white border-2 border-white text-center py-4 px-20 rounded-lg font-bold"
46           onClick={handleAddNewBus}
47         >
48           Add new Bus
49         </button>
50
51         <button
52           className="mt-2 w-full block bg-black hover:bg-gray-900 text-white border-2 border-white text-center py-4 px-20 rounded-lg font-bold"
53           onClick={handleAddNewRoute}
54         >
55           Add new Route
56         </button>
57
58         <button
59           className="mt-2 w-full block bg-black hover:bg-gray-900 text-white border-2 border-white text-center py-4 px-20 rounded-lg font-bold"
60           onClick={handleAddNewSchedule}
61         >
62           Add new Schedule
63         </button>
64       </div>
65     </div>
66   );
67 }
68
69 export default AdminPanel;
```

2. Add Bus



```

1 import React, { useContext, useState } from "react";
2 import { UserToken } from "../../tokens/UserToken";
3 import { Link, useNavigate } from "react-router-dom";
4 import "./adminPanel.css";
5 import BusService from "../../Service/BusService";
6
7 function AddBus() {
8   const [authorized, userRole, userType] = useContext(UserContext);
9   const [busNumber, setBusNumber] = useState("");
10  const [busType, setBusType] = useState("24");
11  const [totalSeats, setTotalSeats] = useState("24");
12  const [error, setError] = useState("");
13  const [busNumberError, setBusNumberError] = useState("");
14  const [busTypeError, setBusTypeError] = useState("");
15  const [busError, setBusError] = useState("");
16  const [successMsg, setSuccessMsg] = useState("");
17
18  let navigate = useNavigate();
19
20  useEffect(() => {
21    if (authorized === "raise") {
22      navigate("/login");
23    }
24    if (userRole === "admin") {
25      navigate("/home");
26    }
27  }, [authorized, userRole]);
28
29  const handleBusNumberChange = (e: React.ChangeEvent<HTMLInputElement>) => {
30    setBusNumber(e.target.value);
31    setError("");
32    if (e.target.value === "") {
33      setBusNumberError("Bus Number is required");
34    } else {
35      setBusNumberError("");
36    }
37  };
38
39  const handleBusTypeChange = (e: React.ChangeEvent<HTMLSelectElement>) => {
40    setBusType(e.target.value);
41    if (e.target.value === "") {
42      setBusTypeError("Bus Type is required");
43    } else {
44      setBusTypeError("");
45    }
46  };
47
48  const handleTotalSeatsChange = (e: React.ChangeEvent<HTMLInputElement>) => {
49    setTotalSeats(e.target.value);
50    if (seats < 5 || seats > 24) {
51      setTotalSeatsError("Total Seats must be between 24 and 52");
52    } else {
53      setTotalSeatsError("");
54    }
55    setTotalSeats(e.target.value);
56  };
57  if (e.target.value === "") {
58    setTotalSeatsError("Seat is required");
59  } else {
60    setTotalSeatsError("");
61  }
62  setTotalSeats("");
63  };
64
65  const handleAddBus = () => {
66    BusService.createBus({ busNumber, busType, totalSeats }).then((res) => {
67      if (res.data.message === "Bus length 0") {
68        setBusError("Bus Already exists!");
69      } else {
70        const bus = {
71          busNumber: busNumber,
72          busType: busType,
73          busCapacity: totalSeats,
74        };
75        BusService.insert(bus).then((res) => {
76          setBusNumber("");
77          setBusType("");
78          setTotalSeats("24");
79          setSuccessMsg("Bus Added Successfully!");
80        });
81      }
82    });
83  };
84
85  function isButtonDisabled() {
86    return !(error
87      || busNumber === ""
88      || busType === ""
89      || seats === ""
90      || busNumber === "" ||
91      busNumber === "" ||
92      busType === "" ||
93      totalSeats === "");
94  }
95
96  return (
97    <div className="flex justify-center items-center h-screen background_image_admin">
98      <div className="bg-black bg-opacity-75 p-6 rounded">
99        <div className="w-96">
100          <input type="text"
101            placeholder="Bus Number"
102            value={busNumber}
103            onChange={handleBusNumberChange}
104            className="w-full mb-4 p-2 border border-gray-300 rounded"
105            />
106            {busNumberError} &>
107            <select name="Bus Type"
108              value={busType}
109              onChange={handleBusTypeChange}
110              className="w-full mb-4 p-2 border border-gray-300 rounded"
111              >
112                <option value="24">24</option>
113                <option value="48">48</option>
114                <option value="Non-AC">Non-AC</option>
115            </select>
116            {busTypeError} &>
117            <input type="number"
118              placeholder="Total Seats"
119              value={totalSeats}
120              onChange={handleTotalSeatsChange}
121              className="w-full mb-4 p-2 border border-gray-300 rounded"
122              />
123            {seatsError} &>
124            <button onClick={handleAddBus}>
125              Add Bus
126            </button>
127            {"/"}/<Link to="/adminPanel" style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
128              <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
129                <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
130                  <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
131                    <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
132                      <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
133                        <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
134                          <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
135                            <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
136                              <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
137                                <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
138                                  <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
139                                    <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
140                                      <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
141                                        <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
142                                          <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
143                                            <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
144                                              <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
145                                                <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
146                                                  <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
147                                                    <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
148                                                      <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
149                                                        <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
150                                                          <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
151                                                            <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
152                                                              <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
153                                                                <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
154                                                                  <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
155                                                                    <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
156                                                                      <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
157                                                                        <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
158                                                                          <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
159                                                                            <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
160                                                                              <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
161                                                                                <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
162                                                                                  <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
163                                                                                    <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
164                                                                                      <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
165                                                                                        <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
166                                                                                            <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
167                                                                                                <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
168                                                                                                  <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
169                                                                                                    <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
170                                                                                                      <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
171                                                                                                        <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
172                                                                                                          <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
173                                                                                                            <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
174                                                                                                              <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
175                                                                                                                <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
176                                                                                                                  <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
177                                                                                                                    <span style={{color: "#0000ff", textDecoration: "underline", cursor: "pointer", margin: "0 10px", border: "1px solid #0000ff", padding: "0 5px", display: "block", width: "100%"}>
178................................................................

```

3. Add Driver

```

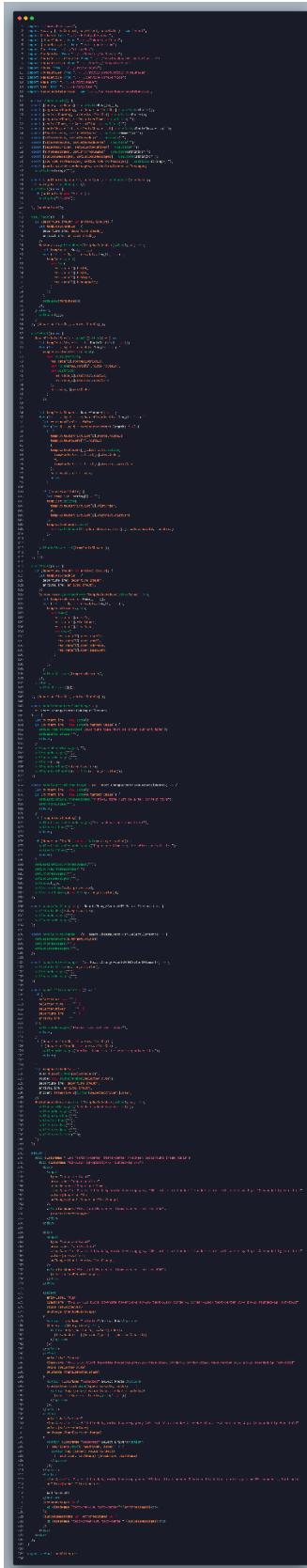
1 import React, { useState, useEffect, useContext } from 'react';
2 import { Link, useNavigate } from 'react-router-dom';
3 import { useAuthContext } from './AuthContext';
4 import './AddDriver.css';
5 import { useState, useEffect, useContext } from 'react';
6 import UserService from './Services/UserService';
7 import AuthService from './Services/AuthService';
8 import DriverService from './Services/DriverService';
9
10 function AddDriver() {
11   const [userList, setUserList] = useState([]);
12   const [errors, setErrors] = useState({});
13   const [selectedUser, setSelectedUser] = useState(null);
14   const [licenseNumber, setLicenseNumber] = useState("");
15   const [expirationDate, setExpirationDate] = useState("");
16   const [formErrors, setFormErrors] = useState({});
17   const [isFormValid, setIsFormValid] = useState(true);
18   const [licenseNumberError, setLicenseNumberError] = useState("");
19   const [expirationDateError, setExpirationDateError] = useState("");
20   const [userError, setUserError] = useState("");
21   const [isSubmitting, setIsSubmitting] = useState(false);
22   const [isSubmitted, setIsSubmitted] = useState(false);
23   const [isSuccess, setIsSuccess] = useState(false);
24   const [isError, setIsError] = useState(false);
25
26   useEffect(() => {
27     if (isAuthenticated === "false") {
28       navigate("login");
29     }
30     if (isUserType === "admin") {
31       navigate("home");
32     }
33   }, [isAuthenticated, isUserType, navigate]);
34
35   const handleFormSubmit = () => {
36     setIsSubmitting(true);
37     const validation = () => {
38       const errors = {};
39       const selectedUserIndex = document.querySelector('#userSelect');
40       const licenseNumberInput = document.querySelector('#licenseNumber');
41       const expirationDateInput = document.querySelector('#expirationDate');
42
43       if (!selectedUserIndex) {
44         errors.userError = "User is required";
45       }
46       if (!licenseNumberInput) {
47         errors.licenseNumberError = "License number is required";
48       }
49       if (!expirationDateInput) {
50         errors.expirationDateError = "Expiration date is required";
51       }
52
53       return errors;
54     };
55
56     const formData = () => {
57       if (isFormValid) {
58         const user = userList.find(user => {
59           return user.id === selectedUser;
60         });
61         let userObj = {
62           id: user.id,
63           licenseNumber: licenseNumber,
64           expirationDate: expirationDate,
65           userRole: "driver"
66         };
67
68         return userObj;
69       }
70     };
71
72     const postDriver = () => {
73       const requestOptions = {
74         method: "POST",
75         headers: {
76           "Content-Type": "application/json"
77         },
78         body: JSON.stringify(formData())
79       };
80
81       fetch(`${process.env.REACT_APP_API_URL}/api/drivers`, requestOptions)
82         .then(res => res.json())
83         .then(data => {
84           if (data.error) {
85             setErrors(data.error);
86           } else {
87             setIsSubmitted(true);
88             setIsSuccess(true);
89             setFormErrors({});
90             setIsSubmitting(false);
91             setExpirationDate("");
92             setLicenseNumber("");
93             setSelectedUser(null);
94             setErrors({});
95           }
96         })
97         .catch(error => {
98           console.error(error);
99         });
100   };
101
102   const handleUserSelectChange = (e) => {
103     const user = userList.find(user => {
104       return user.id === e.target.value;
105     });
106     setSelectedUser(user);
107   };
108
109   const handleLicenseNumberChange = (e) => {
110     setLicenseNumber(e.target.value);
111   };
112
113   const handleExpirationDateChange = (e) => {
114     setExpirationDate(e.target.value);
115   };
116
117   const handleFormReset = () => {
118     setIsSubmitted(false);
119     setIsSuccess(false);
120     setIsError(false);
121     setFormErrors({});
122     setExpirationDate("");
123     setLicenseNumber("");
124     setSelectedUser(null);
125   };
126
127   const handleFormError = (error) => {
128     setFormErrors(error);
129   };
130
131   const handleFormSubmitError = (error) => {
132     setIsError(true);
133     setFormErrors(error);
134   };
135
136   const handleFormSuccess = (success) => {
137     setIsSuccess(true);
138     setFormErrors(success);
139   };
140
141   const handleFormError = (error) => {
142     setFormErrors(error);
143   };
144
145   const handleFormSuccess = (success) => {
146     setFormErrors(success);
147   };
148
149   const handleFormError = (error) => {
150     setFormErrors(error);
151   };
152
153   const handleFormSuccess = (success) => {
154     setFormErrors(success);
155   };
156
157   const handleFormError = (error) => {
158     setFormErrors(error);
159   };
160
161   const handleFormSuccess = (success) => {
162     setFormErrors(success);
163   };
164
165   const handleFormError = (error) => {
166     setFormErrors(error);
167   };
168
169   const handleFormSuccess = (success) => {
170     setFormErrors(success);
171   };
172
173   const handleFormError = (error) => {
174     setFormErrors(error);
175   };
176
177   const handleFormSuccess = (success) => {
178     setFormErrors(success);
179   };
180
181   const handleFormError = (error) => {
182     setFormErrors(error);
183   };
184
185   const handleFormSuccess = (success) => {
186     setFormErrors(success);
187   };
188
189   const handleFormError = (error) => {
190     setFormErrors(error);
191   };
192
193   const handleFormSuccess = (success) => {
194     setFormErrors(success);
195   };
196
197   const handleFormError = (error) => {
198     setFormErrors(error);
199   };
200
201   const handleFormSuccess = (success) => {
202     setFormErrors(success);
203   };
204
205   const handleFormError = (error) => {
206     setFormErrors(error);
207   };
208
209   const handleFormSuccess = (success) => {
210     setFormErrors(success);
211   };
212
213   const handleFormError = (error) => {
214     setFormErrors(error);
215   };
216
217   const handleFormSuccess = (success) => {
218     setFormErrors(success);
219   };
220
221   const handleFormError = (error) => {
222     setFormErrors(error);
223   };
224
225   const handleFormSuccess = (success) => {
226     setFormErrors(success);
227   };
228
229   const handleFormError = (error) => {
230     setFormErrors(error);
231   };
232
233   const handleFormSuccess = (success) => {
234     setFormErrors(success);
235   };
236
237   const handleFormError = (error) => {
238     setFormErrors(error);
239   };
240
241   const handleFormSuccess = (success) => {
242     setFormErrors(success);
243   };
244
245   const handleFormError = (error) => {
246     setFormErrors(error);
247   };
248
249   const handleFormSuccess = (success) => {
250     setFormErrors(success);
251   };
252
253   const handleFormError = (error) => {
254     setFormErrors(error);
255   };
256
257   const handleFormSuccess = (success) => {
258     setFormErrors(success);
259   };
260
261   const handleFormError = (error) => {
262     setFormErrors(error);
263   };
264
265   const handleFormSuccess = (success) => {
266     setFormErrors(success);
267   };
268
269   const handleFormError = (error) => {
270     setFormErrors(error);
271   };
272
273   const handleFormSuccess = (success) => {
274     setFormErrors(success);
275   };
276
277   const handleFormError = (error) => {
278     setFormErrors(error);
279   };
280
281   const handleFormSuccess = (success) => {
282     setFormErrors(success);
283   };
284
285   const handleFormError = (error) => {
286     setFormErrors(error);
287   };
288
289   const handleFormSuccess = (success) => {
290     setFormErrors(success);
291   };
292
293   const handleFormError = (error) => {
294     setFormErrors(error);
295   };
296
297   const handleFormSuccess = (success) => {
298     setFormErrors(success);
299   };
299
300   const handleFormError = (error) => {
301     setFormErrors(error);
302   };
303
304   const handleFormSuccess = (success) => {
305     setFormErrors(success);
306   };
307
308   const handleFormError = (error) => {
309     setFormErrors(error);
310   };
311
312   const handleFormSuccess = (success) => {
313     setFormErrors(success);
314   };
315
316   const handleFormError = (error) => {
317     setFormErrors(error);
318   };
319
320   const handleFormSuccess = (success) => {
321     setFormErrors(success);
322   };
323
324   const handleFormError = (error) => {
325     setFormErrors(error);
326   };
327
328   const handleFormSuccess = (success) => {
329     setFormErrors(success);
330   };
331
332   const handleFormError = (error) => {
333     setFormErrors(error);
334   };
335
336   const handleFormSuccess = (success) => {
337     setFormErrors(success);
338   };
339
340   const handleFormError = (error) => {
341     setFormErrors(error);
342   };
343
344   const handleFormSuccess = (success) => {
345     setFormErrors(success);
346   };
347
348   const handleFormError = (error) => {
349     setFormErrors(error);
350   };
351
352   const handleFormSuccess = (success) => {
353     setFormErrors(success);
354   };
355
356   const handleFormError = (error) => {
357     setFormErrors(error);
358   };
359
360   const handleFormSuccess = (success) => {
361     setFormErrors(success);
362   };
363
364   const handleFormError = (error) => {
365     setFormErrors(error);
366   };
367
368   const handleFormSuccess = (success) => {
369     setFormErrors(success);
370   };
371
372   const handleFormError = (error) => {
373     setFormErrors(error);
374   };
375
376   const handleFormSuccess = (success) => {
377     setFormErrors(success);
378   };
379
380   const handleFormError = (error) => {
381     setFormErrors(error);
382   };
383
384   const handleFormSuccess = (success) => {
385     setFormErrors(success);
386   };
387
388   const handleFormError = (error) => {
389     setFormErrors(error);
390   };
391
392   const handleFormSuccess = (success) => {
393     setFormErrors(success);
394   };
395
396   const handleFormError = (error) => {
397     setFormErrors(error);
398   };
399
400   const handleFormSuccess = (success) => {
401     setFormErrors(success);
402   };
403
404   const handleFormError = (error) => {
405     setFormErrors(error);
406   };
407
408   const handleFormSuccess = (success) => {
409     setFormErrors(success);
410   };
411
412   const handleFormError = (error) => {
413     setFormErrors(error);
414   };
415
416   const handleFormSuccess = (success) => {
417     setFormErrors(success);
418   };
419
420   const handleFormError = (error) => {
421     setFormErrors(error);
422   };
423
424   const handleFormSuccess = (success) => {
425     setFormErrors(success);
426   };
427
428   const handleFormError = (error) => {
429     setFormErrors(error);
430   };
431
432   const handleFormSuccess = (success) => {
433     setFormErrors(success);
434   };
435
436   const handleFormError = (error) => {
437     setFormErrors(error);
438   };
439
440   const handleFormSuccess = (success) => {
441     setFormErrors(success);
442   };
443
444   const handleFormError = (error) => {
445     setFormErrors(error);
446   };
447
448   const handleFormSuccess = (success) => {
449     setFormErrors(success);
450   };
451
452   const handleFormError = (error) => {
453     setFormErrors(error);
454   };
455
456   const handleFormSuccess = (success) => {
457     setFormErrors(success);
458   };
459
460   const handleFormError = (error) => {
461     setFormErrors(error);
462   };
463
464   const handleFormSuccess = (success) => {
465     setFormErrors(success);
466   };
467
468   const handleFormError = (error) => {
469     setFormErrors(error);
470   };
471
472   const handleFormSuccess = (success) => {
473     setFormErrors(success);
474   };
475
476   const handleFormError = (error) => {
477     setFormErrors(error);
478   };
479
480   const handleFormSuccess = (success) => {
481     setFormErrors(success);
482   };
483
484   const handleFormError = (error) => {
485     setFormErrors(error);
486   };
487
488   const handleFormSuccess = (success) => {
489     setFormErrors(success);
490   };
491
492   const handleFormError = (error) => {
493     setFormErrors(error);
494   };
495
496   const handleFormSuccess = (success) => {
497     setFormErrors(success);
498   };
499
500   const handleFormError = (error) => {
501     setFormErrors(error);
502   };
503
504   const handleFormSuccess = (success) => {
505     setFormErrors(success);
506   };
507
508   const handleFormError = (error) => {
509     setFormErrors(error);
510   };
511
512   const handleFormSuccess = (success) => {
513     setFormErrors(success);
514   };
515
516   const handleFormError = (error) => {
517     setFormErrors(error);
518   };
519
520   const handleFormSuccess = (success) => {
521     setFormErrors(success);
522   };
523
524   const handleFormError = (error) => {
525     setFormErrors(error);
526   };
527
528   const handleFormSuccess = (success) => {
529     setFormErrors(success);
530   };
531
532   const handleFormError = (error) => {
533     setFormErrors(error);
534   };
535
536   const handleFormSuccess = (success) => {
537     setFormErrors(success);
538   };
539
540   const handleFormError = (error) => {
541     setFormErrors(error);
542   };
543
544   const handleFormSuccess = (success) => {
545     setFormErrors(success);
546   };
547
548   const handleFormError = (error) => {
549     setFormErrors(error);
550   };
551
552   const handleFormSuccess = (success) => {
553     setFormErrors(success);
554   };
555
556   const handleFormError = (error) => {
557     setFormErrors(error);
558   };
559
560   const handleFormSuccess = (success) => {
561     setFormErrors(success);
562   };
563
564   const handleFormError = (error) => {
565     setFormErrors(error);
566   };
567
568   const handleFormSuccess = (success) => {
569     setFormErrors(success);
570   };
571
572   const handleFormError = (error) => {
573     setFormErrors(error);
574   };
575
576   const handleFormSuccess = (success) => {
577     setFormErrors(success);
578   };
579
580   const handleFormError = (error) => {
581     setFormErrors(error);
582   };
583
584   const handleFormSuccess = (success) => {
585     setFormErrors(success);
586   };
587
588   const handleFormError = (error) => {
589     setFormErrors(error);
590   };
591
592   const handleFormSuccess = (success) => {
593     setFormErrors(success);
594   };
595
596   const handleFormError = (error) => {
597     setFormErrors(error);
598   };
599
600   const handleFormSuccess = (success) => {
601     setFormErrors(success);
602   };
603
604   const handleFormError = (error) => {
605     setFormErrors(error);
606   };
607
608   const handleFormSuccess = (success) => {
609     setFormErrors(success);
610   };
611
612   const handleFormError = (error) => {
613     setFormErrors(error);
614   };
615
616   const handleFormSuccess = (success) => {
617     setFormErrors(success);
618   };
619
620   const handleFormError = (error) => {
621     setFormErrors(error);
622   };
623
624   const handleFormSuccess = (success) => {
625     setFormErrors(success);
626   };
627
628   const handleFormError = (error) => {
629     setFormErrors(error);
630   };
631
632   const handleFormSuccess = (success) => {
633     setFormErrors(success);
634   };
635
636   const handleFormError = (error) => {
637     setFormErrors(error);
638   };
639
640   const handleFormSuccess = (success) => {
641     setFormErrors(success);
642   };
643
644   const handleFormError = (error) => {
645     setFormErrors(error);
646   };
647
648   const handleFormSuccess = (success) => {
649     setFormErrors(success);
650   };
651
652   const handleFormError = (error) => {
653     setFormErrors(error);
654   };
655
656   const handleFormSuccess = (success) => {
657     setFormErrors(success);
658   };
659
660   const handleFormError = (error) => {
661     setFormErrors(error);
662   };
663
664   const handleFormSuccess = (success) => {
665     setFormErrors(success);
666   };
667
668   const handleFormError = (error) => {
669     setFormErrors(error);
670   };
671
672   const handleFormSuccess = (success) => {
673     setFormErrors(success);
674   };
675
676   const handleFormError = (error) => {
677     setFormErrors(error);
678   };
679
680   const handleFormSuccess = (success) => {
681     setFormErrors(success);
682   };
683
684   const handleFormError = (error) => {
685     setFormErrors(error);
686   };
687
688   const handleFormSuccess = (success) => {
689     setFormErrors(success);
690   };
691
692   const handleFormError = (error) => {
693     setFormErrors(error);
694   };
695
696   const handleFormSuccess = (success) => {
697     setFormErrors(success);
698   };
699
700   const handleFormError = (error) => {
701     setFormErrors(error);
702   };
703
704   const handleFormSuccess = (success) => {
705     setFormErrors(success);
706   };
707
708   const handleFormError = (error) => {
709     setFormErrors(error);
710   };
711
712   const handleFormSuccess = (success) => {
713     setFormErrors(success);
714   };
715
716   const handleFormError = (error) => {
717     setFormErrors(error);
718   };
719
720   const handleFormSuccess = (success) => {
721     setFormErrors(success);
722   };
723
724   const handleFormError = (error) => {
725     setFormErrors(error);
726   };
727
728   const handleFormSuccess = (success) => {
729     setFormErrors(success);
730   };
731
732   const handleFormError = (error) => {
733     setFormErrors(error);
734   };
735
736   const handleFormSuccess = (success) => {
737     setFormErrors(success);
738   };
739
740   const handleFormError = (error) => {
741     setFormErrors(error);
742   };
743
744   const handleFormSuccess = (success) => {
745     setFormErrors(success);
746   };
747
748   const handleFormError = (error) => {
749     setFormErrors(error);
750   };
751
752   const handleFormSuccess = (success) => {
753     setFormErrors(success);
754   };
755
756   const handleFormError = (error) => {
757     setFormErrors(error);
758   };
759
760   const handleFormSuccess = (success) => {
761     setFormErrors(success);
762   };
763
764   const handleFormError = (error) => {
765     setFormErrors(error);
766   };
767
768   const handleFormSuccess = (success) => {
769     setFormErrors(success);
770   };
771
772   const handleFormError = (error) => {
773     setFormErrors(error);
774   };
775
776   const handleFormSuccess = (success) => {
777     setFormErrors(success);
778   };
779
780   const handleFormError = (error) => {
781     setFormErrors(error);
782   };
783
784   const handleFormSuccess = (success) => {
785     setFormErrors(success);
786   };
787
788   const handleFormError = (error) => {
789     setFormErrors(error);
790   };
791
792   const handleFormSuccess = (success) => {
793     setFormErrors(success);
794   };
795
796   const handleFormError = (error) => {
797     setFormErrors(error);
798   };
799
800   const handleFormSuccess = (success) => {
801     setFormErrors(success);
802   };
803
804   const handleFormError = (error) => {
805     setFormErrors(error);
806   };
807
808   const handleFormSuccess = (success) => {
809     setFormErrors(success);
810   };
811
812   const handleFormError = (error) => {
813     setFormErrors(error);
814   };
815
816   const handleFormSuccess = (success) => {
817     setFormErrors(success);
818   };
819
820   const handleFormError = (error) => {
821     setFormErrors(error);
822   };
823
824   const handleFormSuccess = (success) => {
825     setFormErrors(success);
826   };
827
828   const handleFormError = (error) => {
829     setFormErrors(error);
830   };
831
832   const handleFormSuccess = (success) => {
833     setFormErrors(success);
834   };
835
836   const handleFormError = (error) => {
837     setFormErrors(error);
838   };
839
840   const handleFormSuccess = (success) => {
841     setFormErrors(success);
842   };
843
844   const handleFormError = (error) => {
845     setFormErrors(error);
846   };
847
848   const handleFormSuccess = (success) => {
849     setFormErrors(success);
850   };
851
852   const handleFormError = (error) => {
853     setFormErrors(error);
854   };
855
856   const handleFormSuccess = (success) => {
857     setFormErrors(success);
858   };
859
860   const handleFormError = (error) => {
861     setFormErrors(error);
862   };
863
864   const handleFormSuccess = (success) => {
865     setFormErrors(success);
866   };
867
868   const handleFormError = (error) => {
869     setFormErrors(error);
870   };
871
872   const handleFormSuccess = (success) => {
873     setFormErrors(success);
874   };
875
876   const handleFormError = (error) => {
877     setFormErrors(error);
878   };
879
880   const handleFormSuccess = (success) => {
881     setFormErrors(success);
882   };
883
884   const handleFormError = (error) => {
885     setFormErrors(error);
886   };
887
888   const handleFormSuccess = (success) => {
889     setFormErrors(success);
890   };
891
892   const handleFormError = (error) => {
893     setFormErrors(error);
894   };
895
896   const handleFormSuccess = (success) => {
897     setFormErrors(success);
898   };
899
900   const handleFormError = (error) => {
901     setFormErrors(error);
902   };
903
904   const handleFormSuccess = (success) => {
905     setFormErrors(success);
906   };
907
908   const handleFormError = (error) => {
909     setFormErrors(error);
910   };
911
912   const handleFormSuccess = (success) => {
913     setFormErrors(success);
914   };
915
916   const handleFormError = (error) => {
917     setFormErrors(error);
918   };
919
920   const handleFormSuccess = (success) => {
921     setFormErrors(success);
922   };
923
924   const handleFormError = (error) => {
925     setFormErrors(error);
926   };
927
928   const handleFormSuccess = (success) => {
929     setFormErrors(success);
930   };
931
932   const handleFormError = (error) => {
933     setFormErrors(error);
934   };
935
936   const handleFormSuccess = (success) => {
937     setFormErrors(success);
938   };
939
940   const handleFormError = (error) => {
941     setFormErrors(error);
942   };
943
944   const handleFormSuccess = (success) => {
945     setFormErrors(success);
946   };
947
948   const handleFormError = (error) => {
949     setFormErrors(error);
950   };
951
952   const handleFormSuccess = (success) => {
953     setFormErrors(success);
954   };
955
956   const handleFormError = (error) => {
957     setFormErrors(error);
958   };
959
960   const handleFormSuccess = (success) => {
961     setFormErrors(success);
962   };
963
964   const handleFormError = (error) => {
965     setFormErrors(error);
966   };
967
968   const handleFormSuccess = (success) => {
969     setFormErrors(success);
970   };
971
972   const handleFormError = (error) => {
973     setFormErrors(error);
974   };
975
976   const handleFormSuccess = (success) => {
977     setFormErrors(success);
978   };
979
980   const handleFormError = (error) => {
981     setFormErrors(error);
982   };
983
984   const handleFormSuccess = (success) => {
985     setFormErrors(success);
986   };
987
988   const handleFormError = (error) => {
989     setFormErrors(error);
990   };
991
992   const handleFormSuccess = (success) => {
993     setFormErrors(success);
994   };
995
996   const handleFormError = (error) => {
997     setFormErrors(error);
998   };
999
1000   const handleFormSuccess = (success) => {
1001     setFormErrors(success);
1002   };
1003
1004   const handleFormError = (error) => {
1005     setFormErrors(error);
1006   };
1007
1008   const handleFormSuccess = (success) => {
1009     setFormErrors(success);
1010   };
1011
1012   const handleFormError = (error) => {
1013     setFormErrors(error);
1014   };
1015
1016   const handleFormSuccess = (success) => {
1017     setFormErrors(success);
1018   };
1019
1020   const handleFormError = (error) => {
1021     setFormErrors(error);
1022   };
1023
1024   const handleFormSuccess = (success) => {
1025     setFormErrors(success);
1026   };
1027
1028   const handleFormError = (error) => {
1029     setFormErrors(error);
1030   };
1031
1032   const handleFormSuccess = (success) => {
1033     setFormErrors(success);
1034   };
1035
1036   const handleFormError = (error) => {
1037     setFormErrors(error);
1038   };
1039
1040   const handleFormSuccess = (success) => {
1041     setFormErrors(success);
1042   };
1043
1044   const handleFormError = (error) => {
1045     setFormErrors(error);
1046   };
1047
1048   const handleFormSuccess = (success) => {
1049     setFormErrors(success);
1050   };
1051
1052   const handleFormError = (error) => {
1053     setFormErrors(error);
1054   };
1055
1056   const handleFormSuccess = (success) => {
1057     setFormErrors(success);
1058   };
1059
1060   const handleFormError = (error) => {
1061     setFormErrors(error);
1062   };
1063
1064   const handleFormSuccess = (success) => {
1065     setFormErrors(success);
1066   };
1067
1068   const handleFormError = (error) => {
1069     setFormErrors(error);
1070   };
1071
1072   const handleFormSuccess = (success) => {
1073     setFormErrors(success);
1074   };
1075
1076   const handleFormError = (error) => {
1077     setFormErrors(error);
1078   };
1079
1080   const handleFormSuccess = (success) => {
1081     setFormErrors(success);
1082   };
1083
1084   const handleFormError = (error) => {
1085     setFormErrors(error);
1086   };
1087
1088   const handleFormSuccess = (success) => {
1089     setFormErrors(success);
1090   };
1091
1092   const handleFormError = (error) => {
1093     setFormErrors(error);
1094   };
1095
1096   const handleFormSuccess = (success) => {
1097     setFormErrors(success);
1098   };
1099
1100   const handleFormError = (error) => {
1101     setFormErrors(error);
1102   };
1103
1104   const handleFormSuccess = (success) => {
1105     setFormErrors(success);
1106   };
1107
1108   const handleFormError = (error) => {
1109     setFormErrors(error);
1110   };
1111
1112   const handleFormSuccess = (success) => {
1113     setFormErrors(success);
1114   };
1115
1116   const handleFormError = (error) => {
1117     setFormErrors(error);
1118   };
1119
1120   const handleFormSuccess = (success) => {
1121     setFormErrors(success);
1122   };
1123
1124   const handleFormError = (error) => {
1125     setFormErrors(error);
1126   };
1127
1128   const handleFormSuccess = (success) => {
1129     setFormErrors(success);
1130   };
1131
1132   const handleFormError = (error) => {
1133     setFormErrors(error);
1134   };
1135
1136   const handleFormSuccess = (success) => {
1137     setFormErrors(success);
1138   };
1139
1140   const handleFormError = (error) => {
1141     setFormErrors(error);
1142   };
1143
1144   const handleFormSuccess = (success) => {
1145     setFormErrors(success);
1146   };
1147
1148   const handleFormError = (error) => {
1149     setFormErrors(error);
1150   };
1151
1152   const handleFormSuccess = (success) => {
1153     setFormErrors(success);
1154   };
1155
1156   const handleFormError = (error) => {
1157     setFormErrors(error);
1158   };
1159
1160   const handleFormSuccess = (success) => {
1161     setFormErrors(success);
1162   };
1163
1164   const handleFormError = (error) => {
1165     setFormErrors(error);
1166   };
1167
1168   const handleFormSuccess = (success) => {
1169     setFormErrors(success);
1170   };
1171
1172   const handleFormError = (error) => {
1173     setFormErrors(error);
1174   };
1175
1176   const handleFormSuccess = (success) => {
1177     setFormErrors(success);
1178   };
1179
1180   const handleFormError = (error) => {
1181     setFormErrors(error);
1182   };
1183
1184   const handleFormSuccess = (success) => {
1185     setFormErrors(success);
1186   };
1187
1188   const handleFormError = (error) => {
1189     setFormErrors(error);
1190   };
1191
1192   const handleFormSuccess = (success) => {
1193     setFormErrors(success);
1194   };
1195
1196   const handleFormError = (error) => {
1197     setFormErrors(error);
1198   };
1199
1200   const handleFormSuccess = (success) => {
1201     setFormErrors(success);
1202   };
1203
1204   const handleFormError = (error) => {
1205     setFormErrors(error);
1206   };
1207
1208   const handleFormSuccess = (success) => {
1209     setFormErrors(success);
1210   };
1211
1212   const handleFormError = (error) => {
1213     setFormErrors(error);
1214   };
1215
1216   const handleFormSuccess = (success) => {
1217     setFormErrors(success);
1218   };
1219
1220   const handleFormError = (error) => {
1221     setFormErrors(error);
1222   };
1223
1224   const handleFormSuccess = (success) => {
1225     setFormErrors(success);
1226   };
1227
1228   const handleFormError = (error) => {
1229     setFormErrors(error);
1230   };
1231
1232   const handleFormSuccess = (success) => {
1233     setFormErrors(success);
1234   };
1235
1236   const handleFormError = (error) => {
1237     setFormErrors(error);
1238   };
1239
1240   const handleFormSuccess = (success) => {
1241     setFormErrors(success);
1242   };
1243
1244   const handleFormError = (error) => {
1245     setFormErrors(error);
1246   };
1247
1248   const handleFormSuccess = (success) => {
1249     setFormErrors(success);
1250   };
1251
1252   const handleFormError = (error) => {
1253     setFormErrors(error);
1254   };
1255
1256   const handleFormSuccess = (success) => {
1257     setFormErrors(success);
1258   };
1259
1260   const handleFormError = (error) => {
1261     setFormErrors(error);
1262   };
126
```

4. Add Route

```

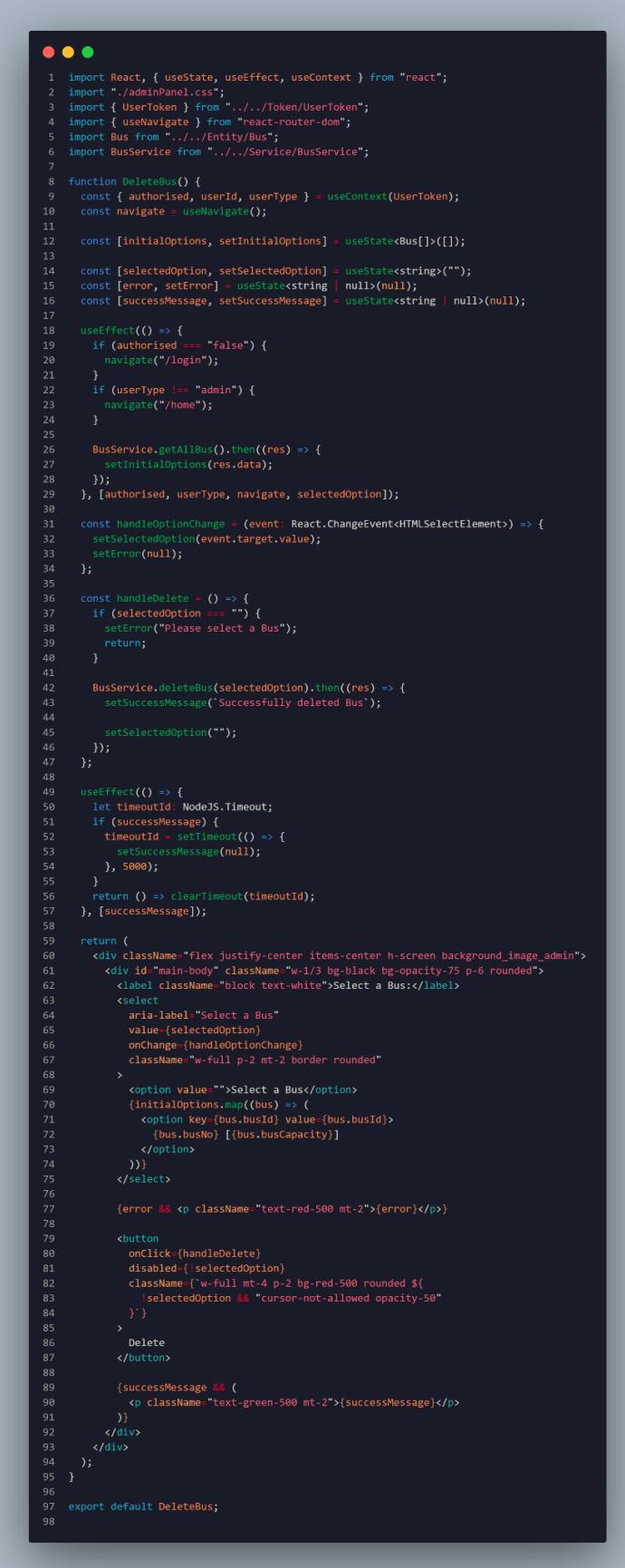
1 import React, { useContext, useState, useEffect } from "react";
2 import { UserToken } from "../../Tokens/UserToken";
3 import { useNavigate } from "react-router-dom";
4 import "./AddRoute.css";
5 import { useGetDistricts } from "../../Services/DistrictService";
6 import District from "../../Entity/District";
7 import RouteService from "../../Services/RouteService";
8 import route from "../../Entity/Route";
9 import RouteDistrictService from "../../Services/RouteDistrictService";
10 import RouteDistrict from "../../Entity/RouteDistrict";
11
12 function AddRoute() {
13   const [districts, setDistricts] = useState(Districts[]);
14   const [source, setSource] = useState("Select Source");
15   const [sourceDistrict, setSourceDistrict] = useState(Districts[0]);
16   const [newDistrict, setNewDistrict] = useState("");
17
18   const { authorized, userId, userType } = useContext(UserToken);
19   let navigate = useNavigate();
20
21   useEffect(() => {
22     const [currentOrder, setCurOrder] = useState(0);
23     const [currentRoute, setCurrentRoute] = useState(route());
24     const [currentDistrict, setCurrentDistrict] = useState(routes[0]);
25     const [routeList, setRouteList] = useState(routes[0].id);
26
27     const { authorized, userId, userType } = useContext(UserToken);
28     let navigate = useNavigate();
29
30     useEffect(() => {
31       if (authorized === "false") {
32         navigate("/login");
33       }
34       if (userType === "admin") {
35         navigate("/");
36       }
37     }, [authorized, userType]);
38
39     useEffect(() => {
40       DistrictService.getDistricts().then((res) => {
41         let districts = res.data.embedded.districts;
42         let tempDistricts: District[] = [];
43         for (let i = 0; i < districts.length; i++) {
44           tempDistricts.push(new District(i + 1, districts[i].distName));
45         }
46         setDistricts(tempDistricts);
47       });
48     }, []);
49
50     function isButtonDisabled() {
51       return source === "Select Source" || sourceError === "";
52     }
53
54     const sourceChangeHandler = (e: React.ChangeEvent) => {
55       const newSource = e.target.value;
56       setSource(newSource);
57       if (newSource === "Select Source") {
58         setSourceError("Please select a source");
59       } else {
60         setSourceError("");
61       }
62     }
63
64     const routeDistrictHandler = () => {
65       if (currentOrder === 1) {
66         RouteService.insert(route).then((res) => {
67           let tempRoute = new route(res.data.routeId);
68           setCurrentRoute(tempRoute);
69           let tempDistrict = {
70             route: tempRoute,
71             source: sourceDistrict,
72             distOrder: currentOrder,
73           };
74           if (currentOrder === 1) {
75             RouteDistrictService.insert(tempDistrict).then((res) => {
76               let tempRouteDistrict = new RouteDistrict();
77               res.data.routeDistrictId = tempRouteDistrict.id;
78               res.data.routeDistrict = tempRouteDistrict;
79               res.data.distOrder = currentOrder;
80               res.data.distId = tempDistrict.id;
81               setRouteList([tempRouteDistrict]);
82             });
83           }
84         });
85       }
86     }
87
88     const routeStopHandler = (currentOrder: number) => {
89       let tempRoute = new route(currentRoute.routeId);
90       let tempRouteDistrict = {
91         route: tempRoute,
92         distOrder: currentOrder,
93         distId: sourceDistrict.id,
94         distName: sourceDistrict.distName,
95         distOrder: currentOrder,
96       };
97       tempRouteDistrict.id = tempRoute.id;
98       tempRouteDistrict.routeId = tempRoute.id;
99       tempRouteDistrict.distOrder = currentOrder;
100      tempRouteDistrict.distId = sourceDistrict.id;
101      tempRouteDistrict.distName = sourceDistrict.distName;
102      tempRouteDistrict.distOrder = currentOrder;
103      tempRouteDistrict.routeId = tempRoute.id;
104      tempRouteDistrict.routeOrder = currentOrder;
105      tempRouteDistrict.routeOrder = currentOrder;
106      tempRouteDistrict.routeOrder = currentOrder;
107      tempRouteDistrict.routeOrder = currentOrder;
108      tempRouteDistrict.routeOrder = currentOrder;
109      tempRouteDistrict.routeOrder = currentOrder;
110      tempRouteDistrict.routeOrder = currentOrder;
111      tempRouteDistrict.routeOrder = currentOrder;
112      tempRouteDistrict.routeOrder = currentOrder;
113      tempRouteDistrict.routeOrder = currentOrder;
114      tempRouteDistrict.routeOrder = currentOrder;
115      tempRouteDistrict.routeOrder = currentOrder;
116      tempRouteDistrict.routeOrder = currentOrder;
117      tempRouteDistrict.routeOrder = currentOrder;
118      tempRouteDistrict.routeOrder = currentOrder;
119      tempRouteDistrict.routeOrder = currentOrder;
120      tempRouteDistrict.routeOrder = currentOrder;
121      tempRouteDistrict.routeOrder = currentOrder;
122      tempRouteDistrict.routeOrder = currentOrder;
123      tempRouteDistrict.routeOrder = currentOrder;
124      tempRouteDistrict.routeOrder = currentOrder;
125      tempRouteDistrict.routeOrder = currentOrder;
126      tempRouteDistrict.routeOrder = currentOrder;
127      tempRouteDistrict.routeOrder = currentOrder;
128      tempRouteDistrict.routeOrder = currentOrder;
129      tempRouteDistrict.routeOrder = currentOrder;
130      tempRouteDistrict.routeOrder = currentOrder;
131      tempRouteDistrict.routeOrder = currentOrder;
132      tempRouteDistrict.routeOrder = currentOrder;
133      tempRouteDistrict.routeOrder = currentOrder;
134      tempRouteDistrict.routeOrder = currentOrder;
135      tempRouteDistrict.routeOrder = currentOrder;
136      tempRouteDistrict.routeOrder = currentOrder;
137      tempRouteDistrict.routeOrder = currentOrder;
138      tempRouteDistrict.routeOrder = currentOrder;
139      tempRouteDistrict.routeOrder = currentOrder;
140      tempRouteDistrict.routeOrder = currentOrder;
141      tempRouteDistrict.routeOrder = currentOrder;
142      tempRouteDistrict.routeOrder = currentOrder;
143      tempRouteDistrict.routeOrder = currentOrder;
144      tempRouteDistrict.routeOrder = currentOrder;
145      tempRouteDistrict.routeOrder = currentOrder;
146      tempRouteDistrict.routeOrder = currentOrder;
147    }
148
149    export default AddRoute;
150  
```

5. Add Schedule



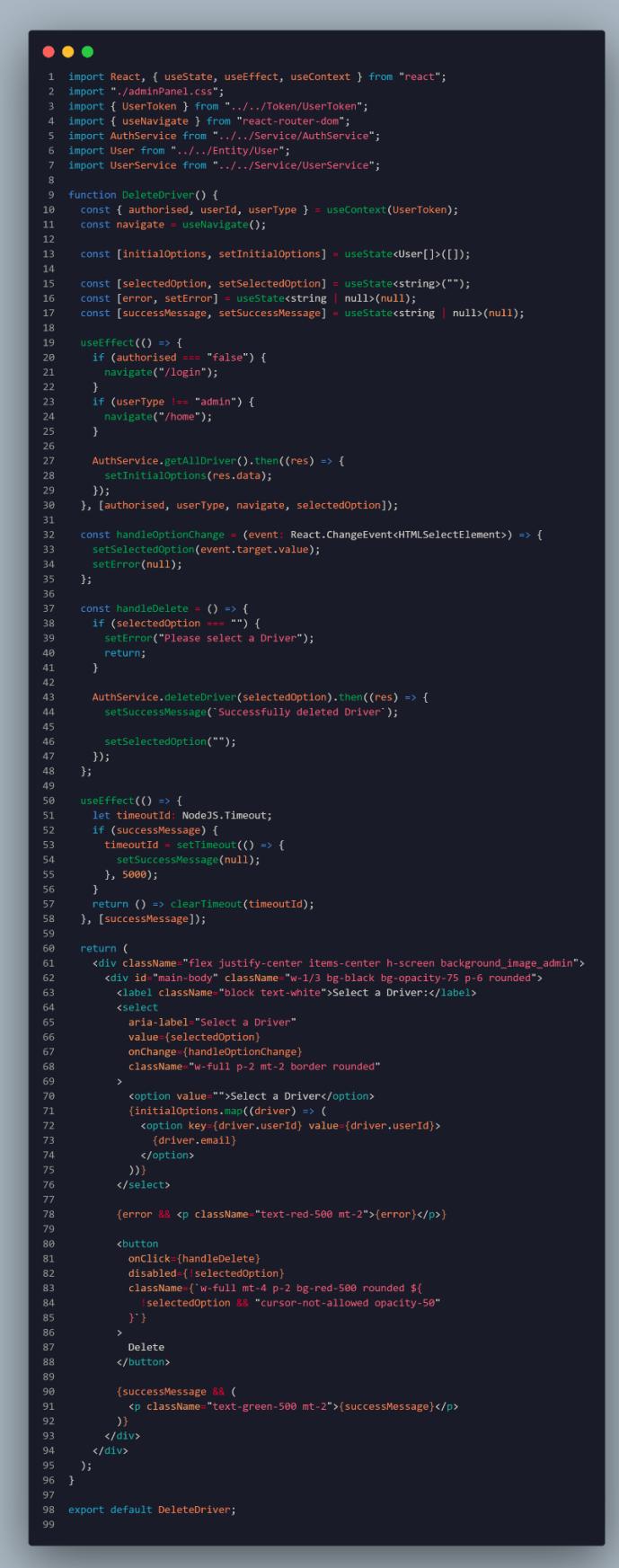
The image shows a terminal window with a dark background and light-colored text. The text is a long, continuous stream of code or log output, likely from a script or a command like 'git log' or 'grep'. The code is written in a programming language, possibly Python, and includes various imports, functions, and data structures. The terminal window has a standard OS X-style title bar at the top.

6. Delete Bus



```
1 import React, { useState, useEffect, useContext } from "react";
2 import "./adminPanel.css";
3 import { UserToken } from "../../Token/UserToken";
4 import { useNavigate } from "react-router-dom";
5 import Bus from "../../Entity/Bus";
6 import BusService from "../../Service/BusService";
7
8 function DeleteBus() {
9   const { authorised, userId, userType } = useContext(UserToken);
10  const navigate = useNavigate();
11
12  const [initialOptions, setInitialOptions] = useState<Bus[]>([]);
13
14  const [selectedOption, setSelectedOption] = useState<string>("");
15  const [error, setError] = useState<string | null>(null);
16  const [successMessage, setSuccessMessage] = useState<string | null>(null);
17
18  useEffect(() => {
19    if (authorised === "false") {
20      navigate("/login");
21    }
22    if (userType === "admin") {
23      navigate("/home");
24    }
25
26    BusService.getAllBus().then((res) => {
27      setInitialOptions(res.data);
28    });
29  }, [authorised, userType, navigate, selectedOption]);
30
31  const handleOptionChange = (event: React.ChangeEvent<HTMLSelectElement>) => {
32    setSelectedOption(event.target.value);
33    setError(null);
34  };
35
36  const handleDelete = () => {
37    if (selectedOption === "") {
38      setError("Please select a Bus");
39      return;
40    }
41
42    BusService.deleteBus(selectedOption).then((res) => {
43      setSuccessMessage("Successfully deleted Bus");
44
45      setSelectedOption("");
46    });
47  };
48
49  useEffect(() => {
50    let timeoutId: NodeJS.Timeout;
51    if (successMessage) {
52      timeoutId = setTimeout(() => {
53        setSuccessMessage(null);
54      }, 5000);
55    }
56    return () => clearTimeout(timeoutId);
57  }, [successMessage]);
58
59  return (
60    <div className="flex justify-center items-center h-screen background_image_admin">
61      <div id="main-body" className="w-1/3 bg-black bg-opacity-75 p-6 rounded">
62        <label className="block text-white">Select a Bus:</label>
63        <select
64          aria-label="Select a Bus"
65          value={selectedOption}
66          onChange={handleOptionChange}
67          className="w-full p-2 mt-2 border rounded"
68        >
69          <option value="">Select a Bus</option>
70          {initialOptions.map((bus) => (
71            <option key={bus.busId} value={bus.busId}>
72              {bus.busNo} [{bus.busCapacity}]
73            </option>
74          ))}
75        </select>
76
77        {error && <p className="text-red-500 mt-2">{error}</p>}
78
79        <button
80          onClick={handleDelete}
81          disabled={!selectedOption}
82          className="w-full mt-4 p-2 bg-red-500 rounded ${!selectedOption && "cursor-not-allowed opacity-50"}"
83        >
84        </button>
85        Delete
86      </button>
87
88        {successMessage && (
89          <p className="text-green-500 mt-2">{successMessage}</p>
90        )}
91      </div>
92    </div>
93  );
94}
95
96 export default DeleteBus;
```

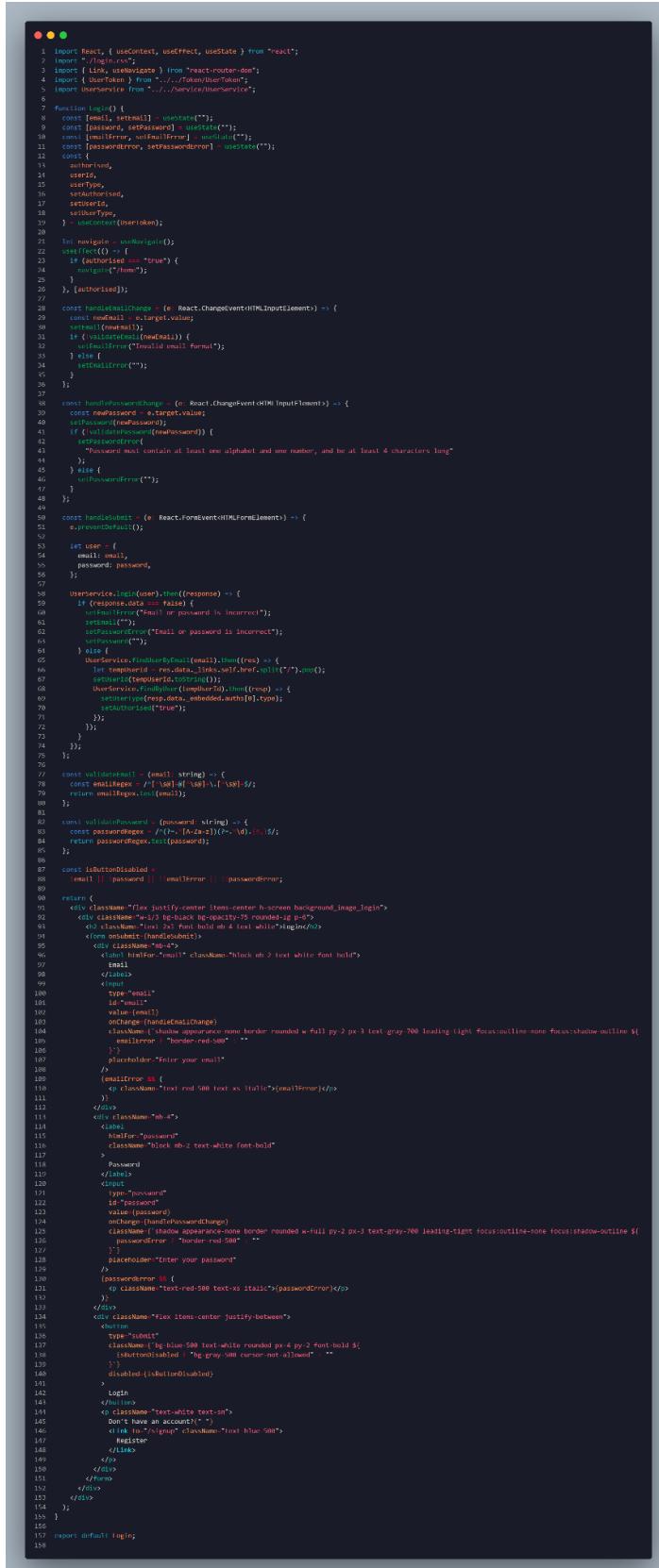
7. Delete Driver



```
1 import React, { useState, useEffect, useContext } from "react";
2 import "./adminPanel.css";
3 import { UserToken } from "../../Token/UserToken";
4 import { useNavigate } from "react-router-dom";
5 import AuthService from "../../Service/AuthService";
6 import User from "../../Entity/User";
7 import UserService from "../../Service/UserService";
8
9 function DeleteDriver() {
10   const [authorised, userId, userType] = useContext(UserToken);
11   const navigate = useNavigate();
12
13   const [initialOptions, setInitialOptions] = useState<User[]>([]);
14
15   const [selectedOption, setSelectedOption] = useState<string>("");
16   const [error, setError] = useState<string | null>(null);
17   const [successMessage, setSuccessMessage] = useState<string | null>(null);
18
19   useEffect(() => {
20     if (authorised === "false") {
21       navigate("/login");
22     }
23     if (userType === "admin") {
24       navigate("/home");
25     }
26
27     AuthService.getAllDriver().then((res) => {
28       setInitialOptions(res.data);
29     });
30   }, [authorised, userType, navigate, selectedOption]);
31
32   const handleOptionChange = (event: React.ChangeEvent<HTMLSelectElement>) => {
33     setSelectedOption(event.target.value);
34     setError(null);
35   };
36
37   const handleDelete = () => {
38     if (selectedOption === "") {
39       setError("Please select a Driver");
40       return;
41     }
42
43     AuthService.deleteDriver(selectedOption).then((res) => {
44       setSuccessMessage("Successfully deleted Driver");
45       setSelectedOption("");
46     });
47   };
48
49   useEffect(() => {
50     let timeoutId: NodeJS.Timeout;
51     if (successMessage) {
52       timeoutId = setTimeout(() => {
53         setSuccessMessage(null);
54       }, 5000);
55     }
56     return () => clearTimeout(timeoutId);
57   }, [successMessage]);
58
59   return (
60     <div className="flex justify-center items-center h-screen background_image_admin">
61       <div id="main-body" className="w-1/3 bg-black bg-opacity-75 p-6 rounded">
62         <label className="block text-white">Select a Driver:</label>
63         <select
64           aria-label="Select a Driver"
65           value={selectedOption}
66           onChange={handleOptionChange}
67           className="w-full p-2 mt-2 border rounded"
68         >
69           <option value="">Select a Driver</option>
70           {initialOptions.map((driver) => (
71             <option key={driver.userId} value={driver.userId}>
72               {driver.email}
73             </option>
74           ))}
75         </select>
76
77         {error && <p className="text-red-500 mt-2">{error}</p>}
78
79         <button
80           onClick={handleDelete}
81           disabled={!selectedOption}
82           className="w-full mt-4 p-2 bg-red-500 rounded ${!selectedOption ? "cursor-not-allowed opacity-50" : ""}"
83         >
84           Delete
85         </button>
86
87         {successMessage && (
88           <p className="text-green-500 mt-2">{successMessage}</p>
89         )}
90       </div>
91     </div>
92   );
93 }
94
95 export default DeleteDriver;
```

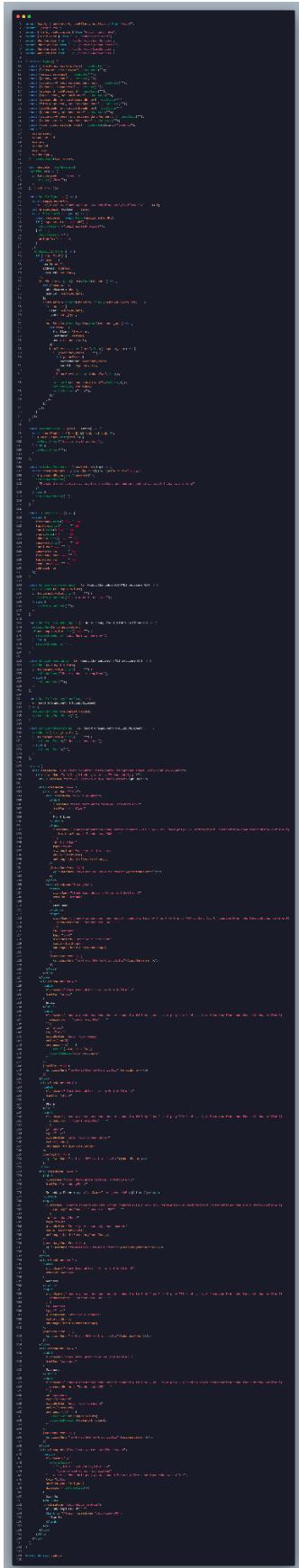
Contribution of ID : 22101334, Name : Md.Rumman Shahriar

1. Login

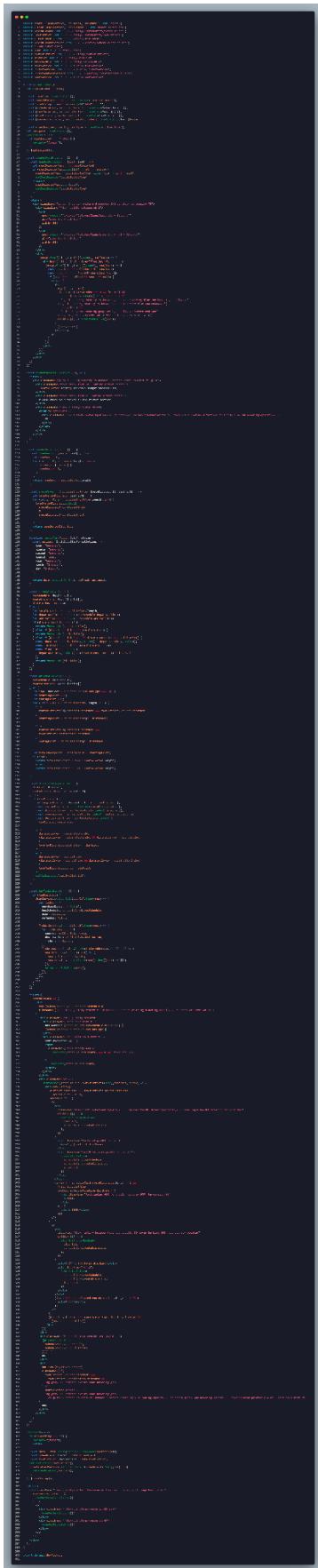


```
1 import React, { useContext, useState } from "react";
2 import "./Login.css";
3 import { Link, useNavigate } from "react-router-dom";
4 import { useState } from "react";
5 import userService from "../../service/UserService";
6
7 function LoginPage() {
8   const [email, setEmail] = useState("");
9   const [password, setPassword] = useState("");
10  const [mailError, validateEmail] = useState("");
11  const [passwordError, validatePassword] = useState("");
12  const [
13    authorized,
14    setAuthorized,
15    userType,
16    setAuthorized,
17    userId,
18    userType,
19    setUserId,
20    userType,
21  ] = useState([{}]);
22
23  let navigate = useNavigate();
24  useEffect(() => {
25    if (authorized === "true") {
26      navigate("/home");
27    }
28  }, [authorized]);
29
30  const handleEmailChange = (e: React.ChangeEvent<HTMLInputElement>) => {
31    const newEmail = e.target.value;
32    setEmail(newEmail);
33    if (!validateEmail(newEmail)) {
34      setMailError("Invalid email format");
35    } else {
36      setMailError("");
37    }
38  };
39
40  const handlePasswordChange = (e: React.ChangeEvent<HTMLInputElement>) => {
41    const newPassword = e.target.value;
42    setPassword(newPassword);
43    if (!validatePassword(newPassword)) {
44      setPasswordError(
45        "Password must contain at least one alphabet and one number, and be at least 4 characters long"
46      );
47    } else {
48      setPasswordError("");
49    }
50  };
51
52  const handleSubmit = (e: React.FormEvent<HTMLFormElement>) => {
53    e.preventDefault();
54
55    const user = {
56      email: email,
57      password: password,
58    };
59
60    userService.login(user).then((response) => {
61      if (response.data === false) {
62        setMailError("Email or password is incorrect");
63        setPasswordError("Email or password is incorrect");
64        setAuthorized("");
65      } else {
66        userService.fetchUserByEmail(email).then((res) => {
67          let tempUserId = res.data._links.self.href.split("/")[-1];
68          getProfile(tempUserId).then((profile) => {
69            setProfile(profile);
70            setAuthorized("true");
71          });
72        });
73      }
74    });
75  };
76
77  const validateEmail = (email: string) => {
78    const emailRegex = /^[^@]+@[^\.\n]+\.[^\.\n]+$/;
79    return emailRegex.test(email);
80  };
81
82  const validatePassword = (password: string) => {
83    const passwordRegex = /^(?=.*[a-zA-Z])(?=.*\d)(?=.*[!@#$%^&*])[a-zA-Z\d!@#$%^&*]{8,}$/;
84    return passwordRegex.test(password);
85  };
86
87  const isButtonDisabled =
88    !(email || password || mailError || passwordError);
89
90  return (
91    <div className="flex justify-center items-center h-screen background_image_login">
92      <div className="w-1/2" style={{ position: "relative", padding: "20px" }}>
93        <div className="text-2xl font-bold mb-4 text-white font-bold">
94          <form onSubmit={handleSubmit}>
95            <div className="mb-4">
96              <label>
97                <input type="email" id="email" value="rumman" onChange={handleEmailChange} />
98                <small>#shadow appearance none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline $1</small>
99                <small>border-red-500 :</small>
100               <small>placeholder="Enter your email" />
101            </label>
102            <small>(email)</small>
103            <small><small>#border-red-500</small></small>
104          </div>
105          <div className="mb-4">
106            <label>
107              <input type="password" id="password" value="rumman" onChange={handlePasswordChange} />
108              <small>#shadow appearance none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline $1</small>
109              <small>border-red-500 :</small>
110              <small>placeholder="Enter your password" />
111            </label>
112            <small>(password)</small>
113            <small><small>#border-red-500 text-xs static">(passwordError)</small></small>
114          </div>
115          <div className="flex items-center justify-between">
116            <small><small>#border-red-500</small></small>
117            <small><small>#border-red-500</small></small>
118            <small><small>#border-red-500</small></small>
119            <small><small>#border-red-500</small></small>
120            <small><small>#border-red-500 cursor-not-allowed</small></small>
121            <small><small>#border-red-500 cursor-not-allowed</small></small>
122          >
123            <small><small>#border-red-500 cursor-not-allowed</small></small>
124            <small><small>#border-red-500 cursor-not-allowed</small></small>
125            <small><small>#border-red-500 cursor-not-allowed</small></small>
126            <small><small>#border-red-500 cursor-not-allowed</small></small>
127            <small><small>#border-red-500 cursor-not-allowed</small></small>
128            <small><small>#border-red-500 cursor-not-allowed</small></small>
129            <small><small>#border-red-500 cursor-not-allowed</small></small>
130            <small><small>#border-red-500 cursor-not-allowed</small></small>
131            <small><small>#border-red-500 cursor-not-allowed</small></small>
132            <small><small>#border-red-500 cursor-not-allowed</small></small>
133            <small><small>#border-red-500 cursor-not-allowed</small></small>
134            <small><small>#border-red-500 cursor-not-allowed</small></small>
135            <small><small>#border-red-500 cursor-not-allowed</small></small>
136            <small><small>#border-red-500 cursor-not-allowed</small></small>
137            <small><small>#border-red-500 cursor-not-allowed</small></small>
138            <small><small>#border-red-500 cursor-not-allowed</small></small>
139            <small><small>#border-red-500 cursor-not-allowed</small></small>
140            <small><small>#border-red-500 cursor-not-allowed</small></small>
141            <small><small>#border-red-500 cursor-not-allowed</small></small>
142            <small><small>#border-red-500 cursor-not-allowed</small></small>
143            <small><small>#border-red-500 cursor-not-allowed</small></small>
144            <small><small>#border-red-500 cursor-not-allowed</small></small>
145            <small><small>#border-red-500 cursor-not-allowed</small></small>
146            <small><small>#border-red-500 cursor-not-allowed</small></small>
147            <small><small>#border-red-500 cursor-not-allowed</small></small>
148            <small><small>#border-red-500 cursor-not-allowed</small></small>
149            <small><small>#border-red-500 cursor-not-allowed</small></small>
150            <small><small>#border-red-500 cursor-not-allowed</small></small>
151            <small><small>#border-red-500 cursor-not-allowed</small></small>
152            <small><small>#border-red-500 cursor-not-allowed</small></small>
153            <small><small>#border-red-500 cursor-not-allowed</small></small>
154          </small>
155        </small>
156      </small>
157    </small>
158  );
159}
160
161 export default LoginPage;
```

2. Signup

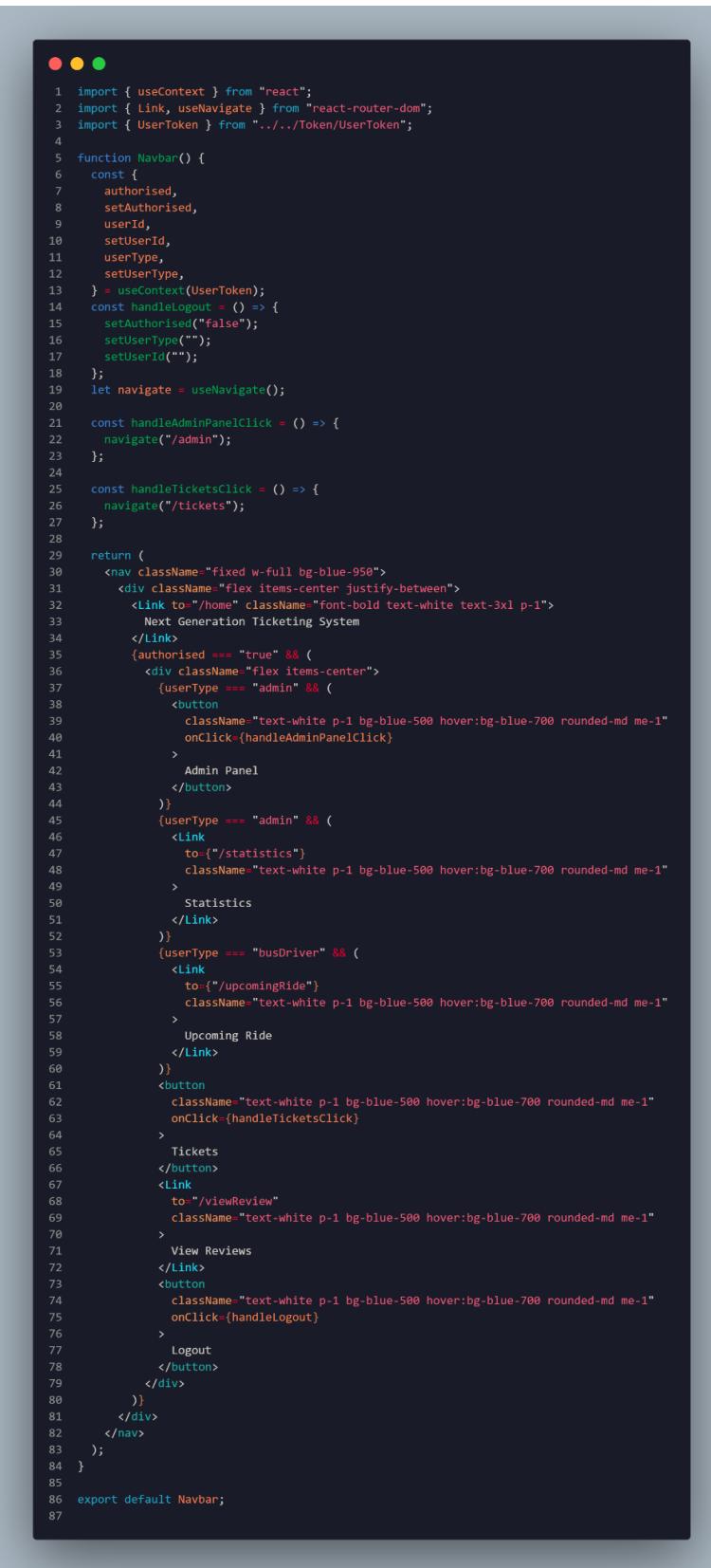


3. Buy Ticket



Contribution of ID :23341032, Name : A K M Nihalul Kabir

1. Navbar



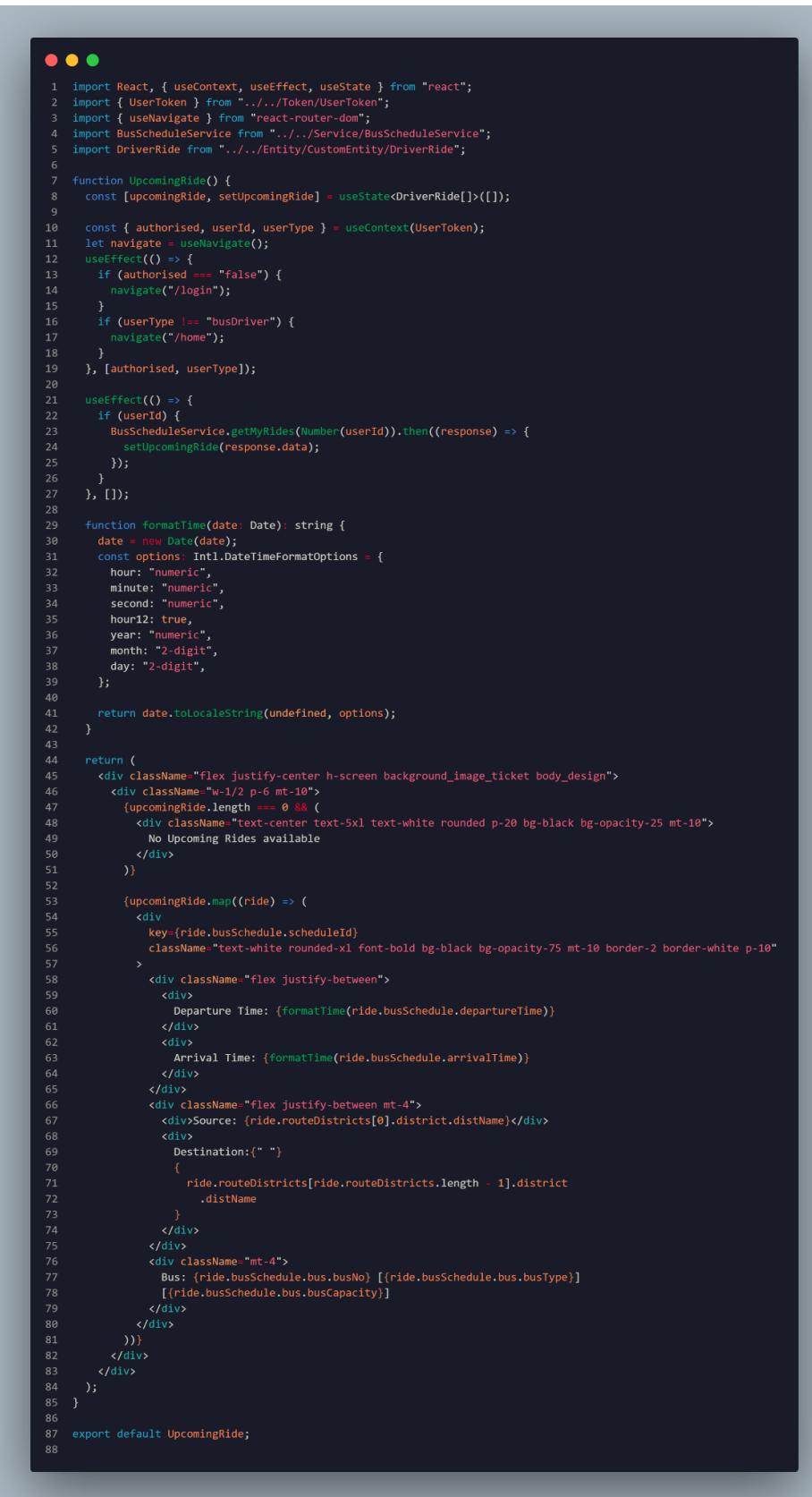
```
1 import { useContext } from "react";
2 import { Link, useNavigate } from "react-router-dom";
3 import { UserToken } from "../../Token/UserToken";
4
5 function Navbar() {
6   const {
7     authorised,
8     setAuthorised,
9     userId,
10    setUserId,
11    userType,
12    setUsertype,
13  } = useContext(UserToken);
14  const handleLogout = () => {
15    setAuthorised("false");
16    setUsertype("");
17    setUserId("");
18  };
19  let navigate = useNavigate();
20
21  const handleAdminPanelClick = () => {
22    navigate("/admin");
23  };
24
25  const handleTicketsClick = () => {
26    navigate("/tickets");
27  };
28
29  return (
30    <nav className="fixed w-full bg-blue-950">
31      <div className="flex items-center justify-between">
32        <Link to="/home" className="font-bold text-white text-3xl p-1">
33          Next Generation Ticketing System
34        </Link>
35        {authorised === "true" && (
36          <div className="flex items-center">
37            {userType === "admin" && (
38              <button
39                className="text-white p-1 bg-blue-500 hover:bg-blue-700 rounded-md me-1"
40                onClick={handleAdminPanelClick}
41              >
42                Admin Panel
43              </button>
44            )}
45            {userType === "admin" && (
46              <Link
47                to="/statistics"
48                className="text-white p-1 bg-blue-500 hover:bg-blue-700 rounded-md me-1"
49              >
50                Statistics
51              </Link>
52            )}
53            {userType === "busDriver" && (
54              <Link
55                to="/upcomingRide"
56                className="text-white p-1 bg-blue-500 hover:bg-blue-700 rounded-md me-1"
57              >
58                Upcoming Ride
59              </Link>
60            )}
61            <button
62              className="text-white p-1 bg-blue-500 hover:bg-blue-700 rounded-md me-1"
63              onClick={handleTicketsClick}
64            >
65              Tickets
66            </button>
67            <Link
68              to="/viewReview"
69              className="text-white p-1 bg-blue-500 hover:bg-blue-700 rounded-md me-1"
70            >
71              View Reviews
72            </Link>
73            <button
74              className="text-white p-1 bg-blue-500 hover:bg-blue-700 rounded-md me-1"
75              onClick={handleLogout}
76            >
77              Logout
78            </button>
79          </div>
80        )}
81      </div>
82    </nav>
83  );
84}
85
86 export default Navbar;
87
```

2. Home Page with Bus searching



Contribution of ID : 22101728 , Name : Rabaya Farzana Sikder

1. Bus Driver profile and upcoming bus ride



The screenshot shows a mobile application interface. At the top, there are three circular icons: red, yellow, and green. Below them is a navigation bar with icons for home, search, and other functions. The main content area displays a bus driver's profile. At the top of the profile section, there is a placeholder for a profile picture with the text "Profile Picture". Below this, the driver's name "Rabaya Farzana Sikder" is displayed, along with their role "Bus Driver" and a "Follow" button. A "Edit Profile" button is also present. The "Upcoming Rides" section shows a list of rides. The first ride listed is for route "10" from "Dhaka" to "Chittagong" on "2023-09-15". The second ride listed is for route "10" from "Chittagong" to "Dhaka" on "2023-09-16". The third ride listed is for route "10" from "Dhaka" to "Chittagong" on "2023-09-17". Each ride entry includes the departure and arrival times, source and destination locations, and the bus number.

```
1 import React, { useContext, useEffect, useState } from "react";
2 import { UserToken } from "../../Token/UserToken";
3 import { useNavigate } from "react-router-dom";
4 import BusScheduleService from "../../Service/BusScheduleService";
5 import DriverRide from "../../Entity/CustomEntity/DriverRide";
6
7 function UpcomingRide() {
8   const [upcomingRide, setUpcomingRide] = useState<DriverRide[]>([]);
9
10  const { authorised, userId, userType } = useContext(UserToken);
11  let navigate = useNavigate();
12  useEffect(() => {
13    if (!authorised === "false") {
14      navigate("/login");
15    }
16    if (userType !== "busDriver") {
17      navigate("/home");
18    }
19  }, [authorised, userType]);
20
21  useEffect(() => {
22    if (userId) {
23      BusScheduleService.getMyRides(Number(userId)).then((response) => {
24        setUpcomingRide(response.data);
25      });
26    }
27  }, []);
28
29  function formatTime(date: Date): string {
30    date = new Date(date);
31    const options: Intl.DateTimeFormatOptions = {
32      hour: "numeric",
33      minute: "numeric",
34      second: "numeric",
35      hour12: true,
36      year: "numeric",
37      month: "2-digit",
38      day: "2-digit",
39    };
40
41    return date.toLocaleString(undefined, options);
42  }
43
44  return (
45    <div className="flex justify-center h-screen background_image_ticket_body_design">
46      <div className="w-1/2 p-6 mt-10">
47        {upcomingRide.length === 0 ? (
48          <div className="text-center text-5xl text-white rounded p-20 bg-black bg-opacity-25 mt-10">
49            No Upcoming Rides available
50          </div>
51        ) : (
52          {upcomingRide.map((ride) => (
53            <div
54              key={ride.busSchedule.scheduleId}
55              className="text-white rounded-xl font-bold bg-black bg-opacity-75 mt-10 border-2 border-white p-10"
56            >
57              <div className="flex justify-between">
58                <div>
59                  Departure Time: {formatTime(ride.busSchedule.departureTime)}
60                </div>
61                <div>
62                  Arrival Time: {formatTime(ride.busSchedule.arrivalTime)}
63                </div>
64              </div>
65              <div className="flex justify-between mt-4">
66                <div>Source: {ride.routeDistricts[0].district.distName}</div>
67                <div>
68                  Destination:{" "}
69                  {
70                    ride.routeDistricts[ride.routeDistricts.length - 1].district
71                    .distName
72                  }
73                </div>
74              </div>
75              <div className="mt-4">
76                Bus: {ride.busSchedule.bus.busNo} [{ride.busSchedule.bus.busType}]
77                [{ride.busSchedule.bus.busCapacity}]
78              </div>
79            </div>
80          </div>
81        ))}
82      </div>
83    </div>
84  );
85}
86
87 export default UpcomingRide;
```

2. Statistics for admin and company use

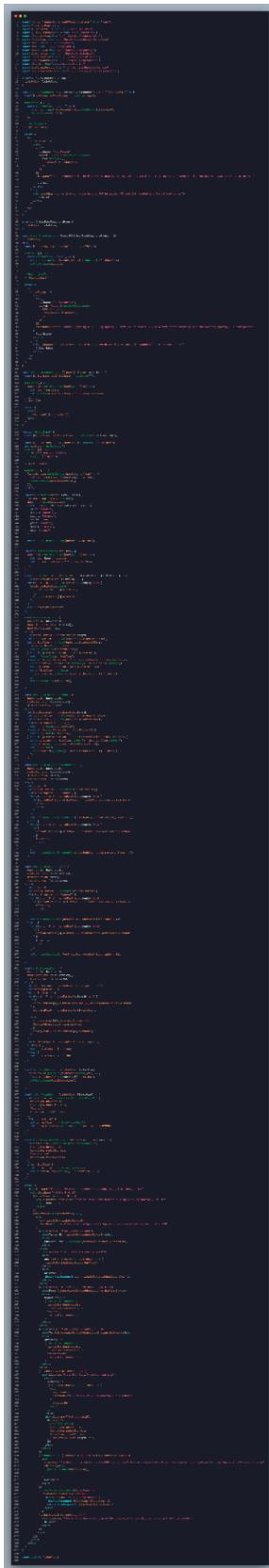
```

1 import { useNavigate } from "react-router-dom";
2 import { useState, useEffect, useContext } from "react";
3 import { UserToken } from "./UserToken";
4 import { UserServices } from "./Service/UserServices";
5 import { UserService } from "./Service/UserService";
6 import { DriverService } from "./Service/DriverService";
7 import { PassengerService } from "./Service/PassengerService";
8 import { HouseService } from "./Service/HouseService";
9 import { RiderService } from "./Service/RiderService";
10 import { BuscheduleService } from "./Service/BuscheduleService";
11 import { TotalRideService } from "./Service/TotalRideService";
12 import { SeatService } from "./Service/SeatService";
13
14 function Statistics() {
15   const [totalUsers, setTotalUser] = useState(0);
16   const [totalDrivers, setTotalDriver] = useState(0);
17   const [totalPassenger, setTotalPassenger] = useState(0);
18   const [totalHouse, setTotalHouse] = useState(0);
19   const [totalIncomingRide, setTotalIncomingRide] = useState(0);
20   const [totalCompletedRide, setTotalCompletedRide] = useState(0);
21   const [averageSeatOccupancy, setAverageSeatOccupancy] = useState(0);
22   const [averageSeatOccupancyX, setAverageSeatOccupancyX] = useState(0);
23
24   const [authorized, userType] = useContext(UserToken);
25   let navigate = useNavigate();
26
27   useEffect(() => {
28     if (authorized === "user") {
29       window.location.href = "/login";
30     }
31     if (userType === "admin") {
32       navigate("/home");
33     }
34   }, [authorized, userType]);
35
36 useEffect(() => {
37   userService.totalUser().then((res) => {
38     setTotalUser(res.data);
39   });
40
41   driverService.totalDriver().then((res) => {
42     setTotalDriver(res.data);
43   });
44
45   authservice.totalPassenger().then((res) => {
46     setTotalPassenger(res.data);
47   });
48
49   houseService.gralHouse().then((res) => {
50     setTotalHouse(res.data.length);
51   });
52
53   routeService.totalRoute().then((res) => {
54     setTotalRoute(res.data);
55   });
56
57   buscheduleService.totalBuschedule().then((res) => {
58     setTotalBuschedule(res.data);
59   });
60
61   ticketService.totalTicket().then((res) => {
62     setTotalTicket(res.data);
63   });
64
65   seatService.averageSeatOccupancy().then((res) => {
66     setAverageSeatOccupancy(res.data);
67   });
68
69   seatService.averageSeatOccupancyX().then((res) => {
70     setAverageSeatOccupancyX(res.data);
71   });
72 }, [authorized, userType]);
73
74
75 return (
76   <div>
77     <h1> Statistic Home </h1>
78     <div>
79       <div>
80         <div> Total User </div>
81         <div> {totalUsers} </div>
82       </div>
83
84       <div> Total Passenger </div>
85       <div> {totalPassenger} </div>
86
87       <div> Total Driver </div>
88       <div> {totalDrivers} </div>
89
90       <div> Total House </div>
91       <div> {totalHouse} </div>
92
93       <div> Total Incoming Ride </div>
94       <div> {totalIncomingRide} </div>
95
96       <div> Total Completed Ride </div>
97       <div> {totalCompletedRide} </div>
98
99       <div> Average Seat Occupancy </div>
100      <div> {averageSeatOccupancy} </div>
101
102      <div> Average Seat Occupancy X </div>
103      <div> {averageSeatOccupancyX} </div>
104    </div>
105  </div>
106)
107
108 export default Statistics;
109
110

```

Contribution of ID : 22101273, Name : Md. Salman Pasha

1. User's tickets and Refund system



2. Review for Bus

3. Review for drivers

Backend Development

Contribution of ID : 22101157, Name : Tanveer Shahriar Arnob

1. Bus Schedule

```
● ● ●

1  @CrossOrigin(origins = "http://localhost:3000")
2  @RestController
3  @RequestMapping("/api/")
4  public class BusScheduleService {
5
6      @Autowired
7      private BusScheduleRepo busScheduleRepo;
8
9      @Autowired
10     private SeatRepo seatRepo;
11
12     @Autowired
13     private UserRepo userRepo;
14
15     @Autowired
16     private RouteDistrictRepo routeDistrictRepo;
17
18     @PostMapping("/busSchedules/insert")
19     public BusSchedule insert(@RequestBody BusSchedule busSchedule) {
20         BusSchedule busSchedule1 = busScheduleRepo.save(busSchedule);
21         char ch = '@';
22         for (int i = 0; i < busSchedule1.getBus().getBusCapacity(); i++) {
23             if (i % 4 == 0) {
24                 ch++;
25             }
26             Seat seat = new Seat();
27             seat.setSeatNo(ch + String.valueOf((i % 4) + 1));
28             seat.setBusSchedule(busSchedule1);
29             seatRepo.save(seat);
30         }
31         return busSchedule1;
32     }
33
34
35     @GetMapping("/busSchedules/{userId}")
36     public List<DriverRide> getMyRides(@PathVariable Long userId) {
37         User user = userRepo.findById(userId);
38         List<BusSchedule> busSchedules = busScheduleRepo.findBusSchedulesByDriverOrderByDepartureTime(user);
39         Timestamp currentTime = new Timestamp(System.currentTimeMillis());
40         List<BusSchedule> newBusSchedules = busSchedules.stream()
41             .filter(schedule -> schedule.getDepartureTime().after(currentTime)).collect(Collectors.toList());
42         List<DriverRide> driverRides = new ArrayList<>();
43         for (BusSchedule busSchedule : newBusSchedules) {
44             List<RouteDistrict> routeDistricts = routeDistrictRepo.findAllByRoute(busSchedule.getRoute());
45             Comparator<RouteDistrict> byDistOrder = Comparator.comparing(RouteDistrict::getDistOrder);
46             Collections.sort(routeDistricts, byDistOrder);
47             DriverRide driverRide = new DriverRide(busSchedule, routeDistricts);
48             driverRides.add(driverRide);
49         }
50     }
51 }
```

2. Add Bus

```
● ● ●

1  @CrossOrigin(origins = "http://localhost:3000")
2  @RestController
3  @RequestMapping("/api/")
4  public class BusService {
5
6      @Autowired
7      private BusRepo busRepo;
8
9      @Autowired
10     private BusScheduleRepo busScheduleRepo;
11
12     @PostMapping("/buses/insert")
13     public Bus insert(@RequestBody Bus bus) {
14         return busRepo.save(bus);
15     }
16
17     @PostMapping("/buses/getFreeBuses")
18     public List<Bus> getFreeBuses(@RequestBody BusSchedule busSchedule) {
19         List<BusSchedule> busSchedules = busScheduleRepo.findAll();
20         List<Bus> allBuses = busRepo.findAll();
21         List<Bus> freeBuses = new ArrayList<Bus>();
22         ;
23         List<Bus> busyBuses = new ArrayList<Bus>();
24         for (BusSchedule busS : busSchedules) {
25             if (!((busS.getDepartureTime().before(busSchedule.getDepartureTime())
26                   && busS.getArrivalTime().before(busSchedule.getArrivalTime())))
27                 || (busS.getDepartureTime().after(busSchedule.getDepartureTime())
28                     && busS.getArrivalTime().after(busSchedule.getArrivalTime())))) {
29                 busyBuses.add(busS.getBus());
30             }
31         }
32         for (Bus bus : allBuses) {
33             if (!busyBuses.contains(bus)) {
34                 freeBuses.add(bus);
35             }
36         }
37         return freeBuses;
38     }
39
40     @GetMapping("/buses/getAllBus")
41     public List<Bus> getAllBus() {
42         return busRepo.findAll();
43     }
44
45     @DeleteMapping("/buses/{busId}")
46     public ResponseEntity<String> deleteUser(@PathVariable Long busId) {
47         Optional<Bus> optionalBus = busRepo.findById(busId);
48
49         if (optionalBus.isPresent()) {
50             busRepo.deleteById(busId);
51             return ResponseEntity.ok("Bus deleted successfully");
52         } else {
53             return ResponseEntity.status(HttpStatus.NOT_FOUND).body("Bus not found");
54         }
55     }
56 }
```

3. Add driver

```
1  @GetMapping("/names/getAllDriver")
2  public List<Name> getAllDriver() {
3      List<User> allUsers = userRepo.findAll();
4      List<Name> allDriver = new ArrayList<Name>();
5
6      for (User user : allUsers) {
7          List<Auth> auths = authRepo.findByUser(user);
8          boolean isDriver = false;
9          for (Auth auth : auths) {
10              if (auth.getType().equals("busDriver")) {
11                  isDriver = true;
12                  break;
13              }
14          }
15          if (isDriver) {
16              allDriver.add(nameRepo.findByUser(user));
17          }
18      }
19      return allDriver;
20  }
21
22  @GetMapping("/names/getAllUser")
23  public List<User> getAllUser() {
24      List<User> allUser = userRepo.findAll();
25      List<User> notDriver = new ArrayList<User>();
26      for (User user : allUser) {
27          List<Auth> auths = authRepo.findByUser(user);
28          boolean isDriver = false;
29          for (Auth auth : auths) {
30              if (auth.getType().equals("busDriver") || auth.getType().equals("admin")) {
31                  isDriver = true;
32                  break;
33              }
34          }
35          if (!isDriver) {
36              notDriver.add(user);
37          }
38      }
39      return notDriver;
40  }
41
42  @PostMapping("/names/getFreeDriver")
43  public List<Name> getFreeDriver(@RequestBody BusSchedule busScheduleProp) {
44      List<User> allUsers = userRepo.findAll();
45      List<Name> freeDriver = new ArrayList<Name>();
46
47      for (User user : allUsers) {
48          List<Auth> auths = authRepo.findByUser(user);
49          boolean isDriver = false;
50          for (Auth auth : auths) {
51              if (auth.getType().equals("busDriver")) {
52                  isDriver = true;
53                  break;
54              }
55          }
56          if (isDriver) {
57              Driver driver = driverRepo.findById(user);
58              if (driver.getDriverLicenseExp().before(busScheduleProp.getArrivalTime())) {
59                  isDriver = false;
60              }
61          }
62          if (isDriver) {
63              List<BusSchedule> busSchedules = busScheduleRepo.findBusSchedulesByDriver(user);
64              boolean hasScheduleClash = false;
65              for (BusSchedule busSchedule : busSchedules) {
66                  if (((busSchedule.getDepartureTime().before(busScheduleProp.getDepartureTime()))
67                      && busSchedule.getArrivalTime().before(busScheduleProp.getArrivalTime()))
68                      || (busSchedule.getDepartureTime().after(busScheduleProp.getDepartureTime())
69                          && busSchedule.getArrivalTime().after(busScheduleProp.getArrivalTime())))
70                  hasScheduleClash = true;
71                  break;
72              }
73          }
74          if (!hasScheduleClash) {
75              freeDriver.add(nameRepo.findByUser(user));
76          }
77      }
78  }
79  return freeDriver;
80 }
```

```
● ● ●  
1 @CrossOrigin(origins = "http://localhost:3000")  
2 @RestController  
3 @RequestMapping("/api/")  
4 public class DriverService {  
5  
6     @Autowired  
7     private DriverRepo driverRepo;  
8  
9     @PostMapping("/drivers/insert")  
10    public Driver insert(@RequestBody Driver driver) {  
11        return driverRepo.save(driver);  
12    }  
13  
14    @PostMapping("/drivers/updateExp")  
15    public Driver updateExp(@RequestBody Driver driver) {  
16        Driver driver1 = driverRepo.findById(driver.getUserId());  
17        driver1.setDriverLicenseExp(driver.getDriverLicenseExp());  
18        return driverRepo.save(driver1);  
19    }  
20}
```

4. Add Route

```
● ● ●  
1 @CrossOrigin(origins = "http://localhost:3000")  
2 @RestController  
3 @RequestMapping("/api/")  
4 public class RouteService {  
5  
6     @Autowired  
7     private RouteRepo routeRepo;  
8  
9     @PostMapping("/routes/insert")  
10    public Route insert(@RequestBody Route route) {  
11        return routeRepo.save(route);  
12    }  
13  
14    @GetMapping("/routes/getRoutes")  
15    public List<Route> getRoutes() {  
16        return routeRepo.findAll();  
17    }  
18  
19    @GetMapping("/routes/totalRoute")  
20    public Integer totalRoute() {  
21        return routeRepo.findAll().size();  
22    }  
23 }
```



```
1  @CrossOrigin(origins = "http://localhost:3000")
2  @RestController
3  @RequestMapping("/api/")
4  public class RouteDistrictService {
5
6      @Autowired
7      private RouteDistrictRepo routeDistrictRepo;
8
9      @PostMapping("/routeDistricts/insert")
10     public RouteDistrict insert(@RequestBody RouteDistrict routeDistrict) {
11         return routeDistrictRepo.save(routeDistrict);
12     }
13
14     @GetMapping("/routeDistricts/getAll")
15     public List<RouteDistrict> getAll() {
16         return routeDistrictRepo.findAll();
17     }
18 }
```

Contribution of ID : 22101334, Name : Md.Rumman Shahriar

1. Authorization

```
● ● ●
1  @CrossOrigin(origins = "http://localhost:3000")
2  @RestController
3  @RequestMapping("/api/")
4  public class AuthService {
5
6      @Autowired
7      private AuthRepo authRepo;
8
9      @Autowired
10     private UserRepo userRepo;
11
12     @PostMapping("/auths	insert")
13     public Auth insert(@RequestBody Auth auth) {
14         return authRepo.save(auth);
15     }
16
17     @PostMapping("/auths/update")
18     public Auth update(@RequestBody User user) {
19         List<Auth> auths = authRepo.findByUser(user);
20         Auth auth = auths.get(0);
21         auth.setType("busDriver");
22         return authRepo.save(auth);
23     }
24
25     @GetMapping("/auths/getAllDriver")
26     public List<User> getAllDriver() {
27         List<Auth> auths = authRepo.findAll();
28         List<User> drivers = new ArrayList<>();
29
30         for (Auth auth : auths) {
31             if (auth.getType().equals("busDriver")) {
32                 drivers.add(auth.getUser());
33             }
34         }
35         return drivers;
36     }
37
38     @DeleteMapping("/auths/{userId}")
39     public ResponseEntity<String> deleteDriver(@PathVariable Long userId) {
40         Optional<User> optionalUser = userRepo.findById(userId);
41
42         if (optionalUser.isPresent()) {
43             List<Auth> auths = authRepo.findByUser(optionalUser.get());
44             for (Auth auth : auths) {
45                 if (auth.getType().equals("busDriver")) {
46                     auth.setType("general");
47                     authRepo.save(auth);
48                 }
49             }
50             return ResponseEntity.ok("User deleted successfully");
51         } else {
52             return ResponseEntity.status(HttpStatus.NOT_FOUND).body("User not found");
53         }
54     }
```

2. Buy Ticket

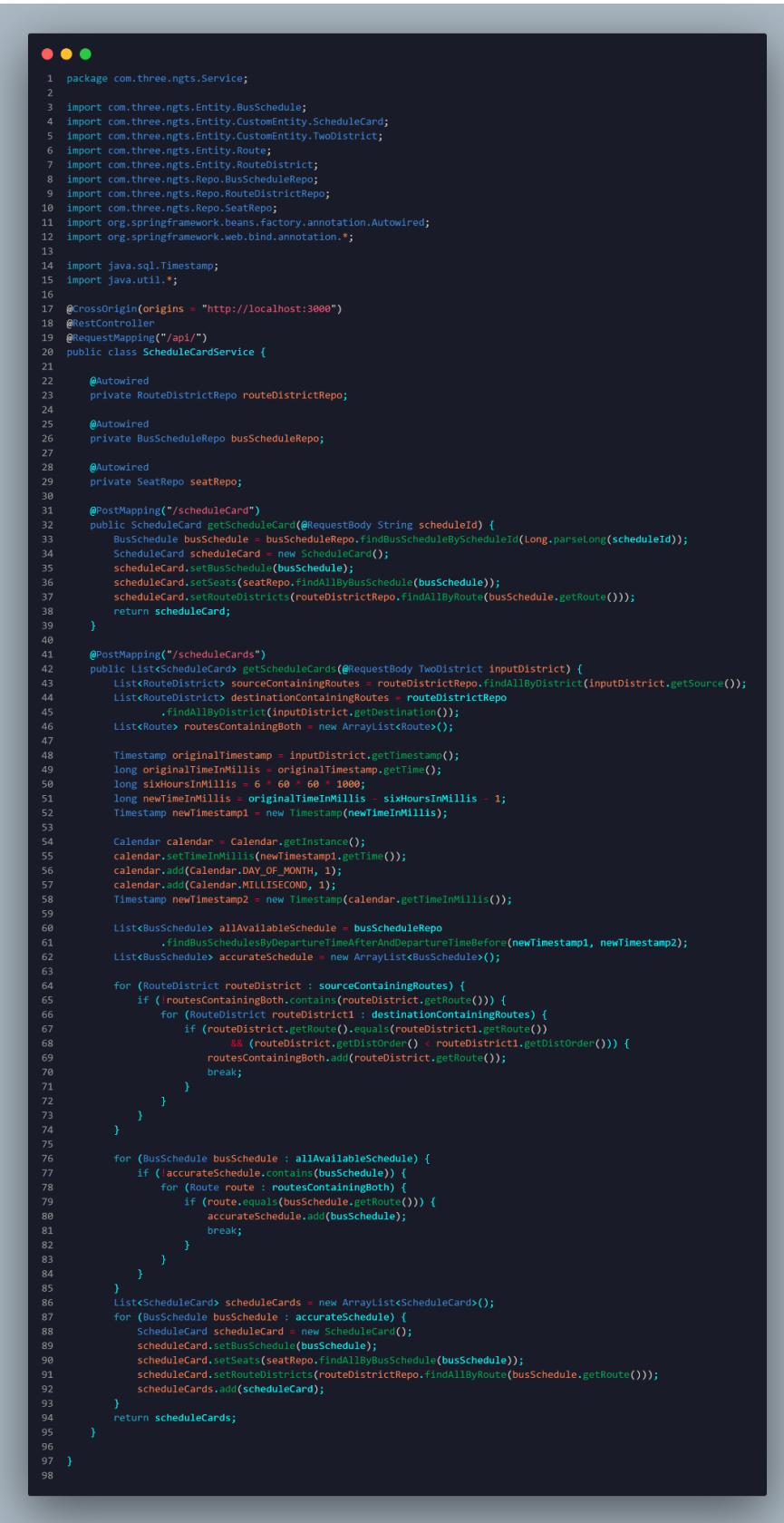
```
1  @CrossOrigin(origins = "http://localhost:3000")
2  @RestController
3  @RequestMapping("/api/")
4  public class TicketService {
5
6      @Autowired
7      private TicketRepo ticketRepo;
8
9      @Autowired
10     private UserRepo userRepo;
11
12     @Autowired
13     private SeatRepo seatRepo;
14
15     @Autowired
16     private TicketAddressRepo ticketAddressRepo;
17
18     @Autowired
19     private RouteDistrictRepo routeDistrictRepo;
20
21     @PostMapping("/tickets/insert")
22     public Ticket insert(@RequestBody Ticket ticket) {
23         return ticketRepo.save(ticket);
24     }
25
26     @PostMapping("/tickets/getMyTickets")
27     public List<TicketView> getMyTickets(@RequestBody String userId) {
28         List<TicketView> ticketViews = new ArrayList<TicketView>();
29         User user = userRepo.findById(Long.parseLong(userId));
30         List<Ticket> tickets = ticketRepo.findByUser(user);
31         Collections.sort(tickets, Comparator.comparing(Ticket::getPurchaseDate));
32         Collections.reverse(tickets);
33         for (Ticket ticket : tickets) {
34             BusSchedule busSchedule = ticket.getBusSchedule();
35             TicketAddress ticketAddress = ticketAddressRepo.findByTicket(ticket);
36             List<Seat> seatsTemp = seatRepo.findAllByBusSchedule(busSchedule);
37             List<Seat> seats = new ArrayList<Seat>();
38             for (Seat seat : seatsTemp) {
39                 if (seat.getTicket() != null && seat.getTicket() == ticket) {
40                     seats.add(seat);
41                 }
42             }
43             List<RouteDistrict> routeDistricts = routeDistrictRepo.findAllByRoute(busSchedule.getRoute());
44             TicketView ticketView = new TicketView(ticket, busSchedule, ticketAddress, seats, routeDistricts);
45             ticketViews.add(ticketView);
46         }
47     }
48 }
```

3. Buy Ticket

```
1  @CrossOrigin(origins = "http://localhost:3000")
2  @RestController
3  @RequestMapping("/api/")
4  public class UserService {
5
6      @Autowired
7      private UserRepo userRepo;
8
9      @PostMapping("/users/getUser")
10     public User getUser(@RequestBody String userId) {
11         return userRepo.findById(Long.parseLong(userId));
12     }
13
14     @PostMapping("/users/signup")
15     public User signUp(@RequestBody User user) {
16         return userRepo.save(user);
17     }
18
19     @PostMapping("/users/login")
20     public boolean login(@RequestBody User user) {
21         try {
22             User newUser = userRepo.findUserByEmail(user.getEmail());
23             return newUser.getEmail().equals(user.getEmail()) && newUser.getPassword().equals(user.getPassword());
24         } catch (Exception e) {
25             return false;
26         }
27     }
}
```

Contribution of ID : 23341032, Name : A K M Nihalul Kabir

1. Search Buses



```
1 package com.three.ngts.Service;
2
3 import com.three.ngts.Entity.BusSchedule;
4 import com.three.ngts.Entity.CustomEntity.ScheduleCard;
5 import com.three.ngts.Entity.CustomEntity.TwoDistrict;
6 import com.three.ngts.Entity.Route;
7 import com.three.ngts.Entity.RouteDistrict;
8 import com.three.ngts.Repo.BusScheduleRepo;
9 import com.three.ngts.Repo.RouteDistrictRepo;
10 import org.springframework.beans.factory.annotation.Autowired;
11 import org.springframework.web.bind.annotation.*;
12
13 import java.sql.Timestamp;
14 import java.util.*;
15
16 @CrossOrigin(origins = "http://localhost:3000")
17 @RestController
18 @RequestMapping("/api/")
19 public class ScheduleCardService {
20
21     @Autowired
22     private RouteDistrictRepo routeDistrictRepo;
23
24     @Autowired
25     private BusScheduleRepo busScheduleRepo;
26
27     @Autowired
28     private SeatRepo seatRepo;
29
30
31     @PostMapping("/scheduleCard")
32     public ScheduleCard getScheduleCard(@RequestBody String scheduleId) {
33         BusSchedule busSchedule = busScheduleRepo.findBusScheduleByScheduleId(Long.parseLong(scheduleId));
34         ScheduleCard scheduleCard = new ScheduleCard();
35         scheduleCard.setBusSchedule(busSchedule);
36         scheduleCard.setSeats(seatRepo.findAllByBusSchedule(busSchedule));
37         scheduleCard.setRouteDistricts(routeDistrictRepo.findAllByRoute(busSchedule.getRoute()));
38         return scheduleCard;
39     }
40
41     @PostMapping("/scheduleCards")
42     public List<ScheduleCard> getScheduleCards(@RequestBody TwoDistrict inputDistrict) {
43         List<RouteDistrict> sourceContainingRoutes = routeDistrictRepo.findAllByDistrict(inputDistrict.getSource());
44         List<RouteDistrict> destinationContainingRoutes = routeDistrictRepo
45             .findAllByDistrict(inputDistrict.getDestination());
46         List<Route> routesContainingBoth = new ArrayList<Route>();
47
48         Timestamp originalTimestamp = inputDistrict.getTimestamp();
49         long originalTimeInMillis = originalTimestamp.getTime();
50         long sixHoursInMillis = 6 * 60 * 60 * 1000;
51         long newTimeInMillis = originalTimeInMillis - sixHoursInMillis - 1;
52         Timestamp newTimestamp1 = new Timestamp(newTimeInMillis);
53
54         Calendar calendar = Calendar.getInstance();
55         calendar.setTimeInMillis(newTimestamp1.getTime());
56         calendar.add(Calendar.DAY_OF_MONTH, 1);
57         calendar.add(Calendar.MILLISECOND, 1);
58         Timestamp newTimestamp2 = new Timestamp(calendar.getTimeInMillis());
59
60         List<BusSchedule> allAvailableSchedule = busScheduleRepo
61             .findAllBusSchedulesByDepartureTimeAfterAndDepartureTimeBefore(newTimestamp1, newTimestamp2);
62         List<BusSchedule> accurateSchedule = new ArrayList<BusSchedule>();
63
64         for (RouteDistrict routeDistrict : sourceContainingRoutes) {
65             if (!routesContainingBoth.contains(routeDistrict.getRoute())) {
66                 for (RouteDistrict routeDistrict1 : destinationContainingRoutes) {
67                     if (routeDistrict.getRoute().equals(routeDistrict1.getRoute())
68                         && (routeDistrict.getDistOrder() < routeDistrict1.getDistOrder())) {
69                         routesContainingBoth.add(routeDistrict.getRoute());
70                         break;
71                     }
72                 }
73             }
74         }
75
76         for (BusSchedule busSchedule : allAvailableSchedule) {
77             if (!accurateSchedule.contains(busSchedule)) {
78                 for (Route route : routesContainingBoth) {
79                     if (route.equals(busSchedule.getRoute())) {
80                         accurateSchedule.add(busSchedule);
81                         break;
82                     }
83                 }
84             }
85         }
86         List<ScheduleCard> scheduleCards = new ArrayList<ScheduleCard>();
87         for (BusSchedule busSchedule : accurateSchedule) {
88             ScheduleCard scheduleCard = new ScheduleCard();
89             scheduleCard.setBusSchedule(busSchedule);
90             scheduleCard.setSeats(seatRepo.findAllByBusSchedule(busSchedule));
91             scheduleCard.setRouteDistricts(routeDistrictRepo.findAllByRoute(busSchedule.getRoute()));
92             scheduleCards.add(scheduleCard);
93         }
94     }
95 }
96
97 }
```

2. Handling seats of the Buses

```
● ● ●  
1 package com.three.ngts.Service;  
2  
3 import com.three.ngts.Entity.Seat;  
4 import com.three.ngts.Repo.SeatRepo;  
5 import org.springframework.beans.factory.annotation.Autowired;  
6 import org.springframework.web.bind.annotation.*;  
7  
8 import java.text.DecimalFormat;  
9 import java.util.List;  
10  
11 @CrossOrigin(origins = "http://localhost:3000")  
12 @RestController  
13 @RequestMapping("/api/")  
14 public class SeatService {  
15  
16     @Autowired  
17     private SeatRepo seatRepo;  
18  
19     @PutMapping("/seats/updateTicket")  
20     public Seat updateTicket(@RequestBody Seat seat) {  
21         Seat newSeat = seatRepo.findById(seat.getSeatId());  
22         newSeat.setTicket(seat.getTicket());  
23         return seatRepo.save(newSeat);  
24     }  
25  
26     @GetMapping("/seats/averageSeatOccupancy")  
27     public Double averageSeatOccupancy() {  
28         List<Seat> seats = seatRepo.findAll();  
29         int totalSeat = seats.size();  
30         int totalSold = 0;  
31         for (Seat seat : seats) {  
32             if (seat.getTicket() != null) {  
33                 totalSold++;  
34             }  
35         }  
36         Double seatOccupancy = ((double) totalSold / totalSeat) * 100;  
37         DecimalFormat decimalFormat = new DecimalFormat("#.##");  
38         String formattedValue = decimalFormat.format(seatOccupancy);  
39         return Double.parseDouble(formattedValue);  
40     }  
41 }  
42
```

Contribution of ID : 22101728 , Name : Rabaya Farzana Sikder

1. Upcoming rides

```
● ● ●  
1 @GetMapping("/busSchedules/totalUpcomingRide")  
2     public Integer totalUpcomingRide() {  
3         Timestamp currentTime = new Timestamp(System.currentTimeMillis());  
4         return busScheduleRepo.findAllByDepartureTimeAfter(currentTime).size();  
5     }  
6
```

2. Statistics

```
● ● ●  
1 @GetMapping("/auths/totalPassenger")  
2     public Integer totalPassenger() {  
3         List<Auth> auths = authRepo.findAll();  
4         Integer count = 0;  
5         for (Auth auth : auths) {  
6             if (auth.getType().equals("general")) {  
7                 count++;  
8             }  
9         }  
10        return count;  
11    }
```

```
● ● ●  
1 @GetMapping("/busSchedules/totalCompletedRide")  
2     public Integer totalCompletedRide() {  
3         Timestamp currentTime = new Timestamp(System.currentTimeMillis());  
4         return busScheduleRepo.findAllByDepartureTimeBefore(currentTime).size();  
5     }
```

```
● ● ●  
1 @GetMapping("/drivers/totalDriver")  
2     public Integer totalDriver() {  
3         return driverRepo.findAll().size();  
4     }
```

```
● ● ●  
1 @DeleteMapping("/users/{userId}")  
2     public ResponseEntity<String> deleteUser(@PathVariable Long userId) {  
3         Optional<User> optionalUser = userRepo.findById(userId);  
4  
5         if (optionalUser.isPresent()) {  
6             userRepo.deleteById(userId);  
7             return ResponseEntity.ok("User deleted successfully");  
8         } else {  
9             return ResponseEntity.status(HttpStatus.NOT_FOUND).body("User not found");  
10        }  
11    }  
12  
13    @GetMapping("/users/totalUser")  
14    public Integer totalUser() {  
15        return userRepo.findAll().size();  
16    }
```

3. Handle Multivalued Phone numbers

```
1 package com.three.ngts.Service;
2
3 import com.three.ngts.Entity.Phone;
4 import com.three.ngts.Repo.PhoneRepo;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.web.bind.annotation.*;
7
8 @CrossOrigin(origins = "http://localhost:3000")
9 @RestController
10 @RequestMapping("/api/")
11 public class PhoneService {
12
13     @Autowired
14     private PhoneRepo phoneRepo;
15
16     @PostMapping("/phones/insert")
17     public Phone insert(@RequestBody Phone phone) {
18         return phoneRepo.save(phone);
19     }
20 }
21
```

Contribution of ID : 22101273, Name : Md. Salman Pasha

1. Review for Bus

```
1 package com.three.ngts.Service;
2
3 import com.three.ngts.Entity.Bus;
4 import com.three.ngts.Entity.BusReview;
5 import com.three.ngts.Entity.CustomEntity.TicketView;
6 import com.three.ngts.Repo.BusReviewRepo;
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.web.bind.annotation.*;
9
10 import java.util.List;
11
12 @CrossOrigin(origins = "http://localhost:3000")
13 @RestController
14 @RequestMapping("/api/")
15 public class BusReviewService {
16
17     @Autowired
18     private BusReviewRepo busReviewRepo;
19
20     @PostMapping("/busReviews/insert")
21     public BusReview insert(@RequestBody BusReview busReview) {
22         return busReviewRepo.save(busReview);
23     }
24
25     @PostMapping("/busReviews/checkReview")
26     public boolean checkReview(@RequestBody TicketView ticketView) {
27         List<BusReview> busReviews = busReviewRepo.findAllByTicketAndReviewerAndBus(ticketView.getTicket(),
28             ticketView.getTicket().getUser(), ticketView.getBusSchedule().getBus());
29         return !busReviews.isEmpty();
30     }
31
32     @PostMapping("/busReviews/getBusReview")
33     public List<BusReview> getBusReview(@RequestBody Bus bus) {
34         return busReviewRepo.findAllByBus(bus);
35     }
36 }
37
```

2. Review for Driver

```
1 package com.three.ngts.Service;
2
3 import com.three.ngts.Entity.CustomEntity.TicketView;
4 import com.three.ngts.Entity.DriverReview;
5 import com.three.ngts.Entity.User;
6 import com.three.ngts.Repo.DriverReviewRepo;
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.web.bind.annotation.*;
9
10 import java.util.List;
11
12 @CrossOrigin(origins = "http://localhost:3000")
13 @RestController
14 @RequestMapping("/api/")
15 public class DriverReviewService {
16
17     @Autowired
18     private DriverReviewRepo driverReviewRepo;
19
20     @PostMapping("/driverReviews/insert")
21     public DriverReview insert(@RequestBody DriverReview driverReview) {
22         return driverReviewRepo.save(driverReview);
23     }
24
25     @PostMapping("/driverReviews/checkReview")
26     public boolean checkReview(@RequestBody TicketView ticketView) {
27         List<DriverReview> driverReviews = driverReviewRepo.findAllByTicketAndDriverAndReviewer(ticketView.getTicket(),
28             ticketView.getBusSchedule().getDriver(), ticketView.getTicket().getUser());
29         return !driverReviews.isEmpty();
30     }
31
32     @PostMapping("/driverReviews/getDriverReview")
33     public List<DriverReview> getDriverReview(@RequestBody User user) {
34         return driverReviewRepo.findAllByDriver(user);
35     }
36 }
37
```

3. Refund Ticket

```
● ● ●

1 package com.three.ngts.Service;
2
3 import com.three.ngts.Entity.*;
4 import com.three.ngts.Entity.CustomEntity.TicketView;
5 import com.three.ngts.Repo.*;
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.web.bind.annotation.*;
8
9 import java.util.ArrayList;
10 import java.util.Collections;
11 import java.util.Comparator;
12 import java.util.List;
13
14 @CrossOrigin(origins = "http://localhost:3000")
15 @RestController
16 @RequestMapping("/api/")
17 public class TicketService {
18
19     @Autowired
20     private TicketRepo ticketRepo;
21
22     @Autowired
23     private UserRepo userRepo;
24
25     @Autowired
26     private SeatRepo seatRepo;
27
28     @Autowired
29     private TicketAddressRepo ticketAddressRepo;
30
31     @Autowired
32     private RouteDistrictRepo routeDistrictRepo;
33
34     @PostMapping("/tickets/getRefund")
35     public List<TicketView> getRefund(@RequestBody TicketView ticketView) {
36         List<Seat> seats = ticketView.getSeats();
37         for (Seat seat : seats) {
38             seat.setTicket(null);
39             seatRepo.save(seat);
40         }
41         User user = ticketView.getTicket().getUser();
42         Ticket ticket = ticketView.getTicket();
43         ticket.setRefunded(true);
44         ticketRepo.save(ticket);
45         return getMyTickets(user.getUserId().toString());
46     }
}
```

Conclusion

In conclusion, our next-generation ticketing system website redefines the travel booking experience for both passengers and transportation providers. With its intuitive interface, powerful features, and comprehensive services, our platform sets a new standard for online travel booking.

Embrace the future of travel and join us on a journey of convenience, efficiency, and seamless travel.

References

1. login signup background :
<https://unsplash.com/photos/white-and-black-bus-running-near-the-mountain-6mcVaoGNz1w>
2. home page background :
https://unsplash.com/photos/passenger-bus-interior-LoQC6U6gOpQ?utm_content=creditShareLink&utm_medium=referral&utm_source=unsplash
3. admin page background :
https://unsplash.com/photos/people-riding-passenger-bus-during-daytime-6fF-Ojxov6o?utm_content=creditShareLink&utm_medium=referral&utm_source=unsplash
4. steering wheel icon:
https://www.flaticon.com/free-icon-font/steering-wheel_7433984
5. exit icon:
https://www.flaticon.com/free-icon-font/exit-alt_13088003?page=1&position=42&term=door&origin=search&related_id=13088003