

SCREENSHOTS

App.js

```
motor_client > src > JS App.js > ...
1
2 import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
3 import Home from './components/Home';
4 import Registration from './components/Registration';
5 import Aboutus from './components/Aboutus';
6 import EmployeeRegistration from './components/EmployeeRegistration';
7 import Login from './pages/Login';
8 import Userhome from './user/Userhome';
9 import Adminhome from './admin/Adminhome';
10 import { CssBaseline, ThemeProvider } from "@mui/material";
11 import { createTheme } from "@mui/material/styles";
12 import { useMemo } from "react";
13 import { useSelector } from "react-redux";
14 import { themeSettings } from "./theme"
15 import Layout from './admin/layout';
16 import AdminViewusersPage from './admin/pages/admin_view_users_page';
17 import AdminViewUsers from './admin/components/admin_view_users';
18 import ProtectedRoutes from './pages/ProtectedRoutes';
19 import PrivateRoute from './pages/PrivateRoute';
20 import AdminViewEmployee from './admin/components/admin_view_employee';
21 import ForgotPassword from './pages/ForgotPassword';
22 //import ResetPassword from './pages/ResetPassword';
23 import OtpVerification from './pages/otpVerification';
24 import AdminAddCars from './admin/components/admin_add_cars';
25 import CarList from './pages/getCars';
26 import CarSelectorForm from './pages/prefference';
27 import MapComponent from './user/map';
28 import { Toaster } from 'react-hot-toast';
29 import CreateCategory from './admin/pages/admin_add_categories';
30 import CreateColor from './admin/pages/admin add color';
```

```

function App() {
  const mode = useSelector((state) => state.global.mode);
  const theme = useMemo(() => createTheme(themeSettings(mode)), [mode]);
  return (
    <div className="App">
      <Toaster/>
      <Router>
        <ThemeProvider theme={theme}>
          <CssBaseline />
          <Routes>
            { /*protected*/ }
            <Route element={ <ProtectedRoutes /> }>
              <Route path="/adminhome" element={ <Adminhome /> } />
            </Route>
            <Route element={ <ProtectedRoutes /> }>
              <Route path="/layout" element={ <Layout /> } />
            </Route>
            <Route element={ <ProtectedRoutes /> }>
              <Route path="/getalluser" element={ <AdminViewusersPage /> } />
            </Route>
            <Route element={ <ProtectedRoutes /> }>
              <Route path="/deleteuser/:id" element={ <AdminViewUsers /> } />
            </Route>
            <Route element={ <ProtectedRoutes /> }>
              <Route path="/viewemployee" element={ <AdminViewEmployee /> } />
            </Route>
            <Route element={ <ProtectedRoutes /> }>
              <Route path="/AddCars" element={ <AdminAddCars /> } />
            </Route>
          </Routes>
        </ThemeProvider>
      </Router>
    </div>
  );
}

```

Server.js

```

motor_server > JS server.js > ...
17
18
19 app.use(cors());
20 app.use(bodyParser.json());
21 app.use(bodyParser.urlencoded({ extended: true }));
22 app.use(express.static('public'));
23 app.use('/downloads', express.static(path.join(__dirname, 'public', 'uploads')));
24 app.use('/public', express.static('public'));
25 app.use(express.static('public'));
26
27
28
29
30 mongoose.connect('mongodb://127.0.0.1:27017/elmotors', { useNewUrlParser: true, useUnifiedTopology: true })
31   .then(() => {
32     console.log('Connected to MongoDB');
33   })
34   .catch((error) => {
35     console.error('Error connecting to MongoDB:', error);
36   });
37
38 const getalluser=require('./controllers/getallusers');
39 app.use('/api/getalluser',getalluser);
40
41 const joinus=require('./controllers/joinus');
42 app.use('/api/joinus',joinus);
43
44 const employees=require('./controllers/employees');
45 app.use('/api/employees',employees);
46

```

Model

```
motor_server > model > JS usermodel.js > userSchema > email
1  const mongoose = require('mongoose');
2
3  const userSchema = new mongoose.Schema({
4    username: { type: String, required: true },
5    email: { type: String, required: true, unique: true }, // Set as unique
6    phone: { type: String, required: true },
7    dob: { type: Date, required: true }, // Change this to Date if you want to store it as a Date type
8  });
9
10
11 const User = mongoose.model('User', userSchema);
12
13 module.exports = User;
14
15
```