**Futsal Field Reservation System**

**Full Code**

1. **Backend**

* **Constant**

**#constant.js:**

export const CONNECTION\_URL = "mongodb+srv://hammadahsan07:qhNkH6SGY7F65mWU@futsal-cluster.y8gj5r6.mongodb.net/?retryWrites=true&w=majority&appName=Futsal-Cluster";

export const PORT = process.env.PORT || 5000;

* **Models**

**#academy.js:**

import mongoose from 'mongoose';

const academySchema =  mongoose.Schema({

    studentname: String,

    fathername: String,

    email: String,

    age: Number,

    contact: String,

    address: String,

    myacademy: String

})

export default new mongoose.model("Academy", academySchema)

**#ground.js**

import mongoose from "mongoose";

import bcrypt from "bcryptjs";

const groundSchema = mongoose.Schema({

  name: String,

  ownername: String,

  email: String,

  password: String,

  phone: String,

  address: String,

  imgURL: String,

  slot: [

    {

      date: String,

      time: [

        {

          bookedby: { type: String, default: "" },

          date: { type: String, default: "" },

          in: { type: String, default: "" },

        },

      ],

    },

  ],

});

groundSchema.pre("save", async function (next) {

  if (this.isModified("password")) {

    this.password = await bcrypt.hash(this.password, 12);

  }

  next();

});

export default new mongoose.model("Ground", groundSchema);

**#teams.js**

import mongoose from "mongoose";

import bcrypt from "bcryptjs";

const teamSchema = mongoose.Schema({

  name: String,

  captainName: String,

  email: String,

  password: String,

  phone: String,

  address: String,

  // aboutTeam: {

  //     type: String,

  //     default: []

  // },

  statusArray: {

    type: Array,

    default: [],

  },

  imgURL: String,

});

teamSchema.pre("save", async function (next) {

  if (this.isModified("password")) {

    this.password = await bcrypt.hash(this.password, 12);

  }

  next();

});

export default new mongoose.model("Team", teamSchema);

* **Routes**

**#academy.js:**

import express from "express";

const router = express.Router();

//Model

import Academy from "../models/academy.js";

router.get("/users/:name", async (req, res) => {

  const name = req.params.name;

  const academy = await Academy.find({ myacademy: name });

  if (!academy) {

    res.send({});

  } else {

    res.send(academy);

  }

});

// /academy/test

router.get("/test3", async (req, res, next) => {

  const academies = await Academy.find();

  res.send(academies);

});

router.post("/academyformregister", async (req, res) => {

  //console.log(req.body)

  const { studentname, fathername, email, age, contact, address, myacademy } =

    req.body;

  Academy.findOne({ email: email }, (err, mongoacademy) => {

    if (mongoacademy) {

      res.send({

        message:

          "Student already Registered with this Email. Try something new. ",

      });

    }

    //Creating. Enter data in database

    else {

      const mongoacademy = new Academy({

        studentname,

        fathername,

        email,

        age,

        contact,

        address,

        myacademy,

      });

      mongoacademy.save((err) => {

        if (err) {

          res.send(err);

        } else {

          res.send({ message: "Student Successfull Registered to Academy" });

        }

      });

    }

  });

});

export default router;

**#ground.js:**

import express from "express";

import bcrypt from "bcryptjs";

const router = express.Router();

//Model

import Ground from "../models/ground.js";

// /ground/test

router.get("/test", async (req, res, next) => {

  const grounds = await Ground.find();

  res.send(grounds);

});

//GET PROFILE DETAILS

router.get("/profile/:\_id", async (req, res) => {

  const \_id = req.params.\_id;

  const ground = await Ground.findOne({ \_id });

  if (!ground) {

    res.send({ message: "Not found" });

  } else {

    res.send(ground);

  }

});

router.post("/dp", async (req, res) => {

  try {

    const { \_id, imgURL } = req.body;

    const ground = await Ground.findOneAndUpdate(

      { \_id },

      {

        $set: {

          imgURL,

        },

      }

    );

    res.send({ message: "Updated" });

  } catch (error) {}

});

router.post("/login2", async (req, res, next) => {

  try {

    const { email, password } = req.body;

    if (!email || !password) {

      return res.send({ message: "Fill all the Fields" });

    }

    const ground = await Ground.findOne({ email: email });

    if (ground) {

      const isMatch = await bcrypt.compare(password, ground.password);

      if (!isMatch) {

        res.send({ message: "Password didn't match" });

      } else {

        res.send({ message: "Login Successsfully", ground });

      }

    } else {

      return res.send({

        message: "Ground not Registered, Please make an account first",

      });

    }

  } catch (err) {

    console.log(err);

    res.send(err);

  }

});

router.post("/register", async (req, res, next) => {

  try {

    const { name, ownername, email, password, phone, address } = req.body;

    const ground = await Ground.findOne({ email });

    if (ground) {

      res.send({ message: "Ground already Registered" });

    }

    //Creating. Enter data in database

    //teamname:teamname;

    else {

      const newGround = new Ground({

        name,

        ownername,

        email,

        password,

        phone,

        address,

      });

      await newGround.save();

      res.send({ message: "Successfully Registered, Please Login Now" });

    }

  } catch (err) {

    console.log(err);

    res.send(err);

  }

});

router.post("/getSlot", async (req, res) => {

  try {

    const { \_id, date } = req.body;

    const ground = await Ground.findOne({ \_id });

    console.log('ground',ground)

    if (!ground) {

      return res.status(404).json({ error: "Ground not found" }); // Use return

    }

    const slots = ground.slot;

    const matchedSlot = slots.find(item => item.date === date);

    if (matchedSlot) {

      return res.json(matchedSlot.time); // Return time if date matches

    } else {

      return res.json([]); // Return empty array if no match

    }

  } catch (error) {

    console.error('error',error);

    return res.status(500).json({ error: "Server error" });

  }

});

router.post("/bookSlot", async (req, res) => {

  try {

    const { \_id, date, name, index } = req.body;

    console.log(\_id, date, name, index);

    const ground = await Ground.findOne({ \_id });

    const slot = ground.slot;

    let times = [];

    let flag = false;

    const data = {

      bookedby: name,

      date: date,

      in: index,

    };

    slot.map((item, index) => {

      if (item.date == date) {

        times = item.time;

        flag = true;

      }

    });

    if (flag == true) {

      // if (times[index] != null || times[index] != undefined) {

      //   return res.send({ message: "Already Booked this slot" });

      // }

      times[index] = data;

      await Ground.findOneAndUpdate(

        { \_id, "slot.date": date },

        {

          $push: {

            "slot.$[date].time": {

              $each: [data],

              $position: index,

            },

          },

        },

        {

          arrayFilters: [{ "date.date": date }],

        }

      );

      res.send({ message: "Successfully Booked " });

    } else {

      times.push(data);

      let dat = { date, time: times };

      await Ground.findOneAndUpdate(

        { \_id },

        {

          $push: {

            slot: dat,

          },

        }

      );

      res.send({ message: "Successfully Booked " });

    }

  } catch (error) {

    res.send({ error: error });

  }

});

router.post("/undoSlot", async (req, res) => {

  try {

    const { index, \_id, date } = req.body;

    const ground = await Ground.findOne({ \_id });

    const slot = ground.slot;

    await Ground.findOneAndUpdate(

      { \_id },

      {

        $pull: {

          "slot.$[date].time": { in: index },

        },

      },

      {

        arrayFilters: [{ "date.date": date }],

      }

    );

    res.send({ message: "Sucessfully Removed " });

  } catch (error) {

    res.send({ error: error });

  }

});

export default router;

**#team.js:**

import express from "express";

import bcrypt from "bcryptjs";

const router = express.Router();

//Model

import Team from "../models/team.js";

import Ground from "../models/ground.js";

//ALL TEAMS AND GROUND

router.get("/all/:\_id", async (req, res) => {

  const \_id = req.params.\_id;

  let k = "new ObjectId(";

  const team = await Team.find({});

  const ground = await Ground.find({});

  const all = team.concat(ground);

  let complete = [];

  all.map((item, index) => {

    if (item.\_id.toString() != \_id) {

      complete.push(item);

    }

  });

  // console.log(complete);

  // console.log(\_id, `new ObjectId(\"${\_id}\")`);

  res.send(complete);

});

//GET PROFILE DETAILS

router.get("/profile/:\_id", async (req, res) => {

  const \_id = req.params.\_id;

  const team = await Team.findOne({ \_id });

  if (!team) {

    res.send({ message: "Not found" });

  } else {

    res.send(team);

  }

});

// /team/test

router.get("/test", async (req, res, next) => {

  const teams = await Team.find({});

  res.send(teams);

});

router.post("/dp", async (req, res) => {

  try {

    const { \_id, imgURL } = req.body;

    const team = await Team.findOneAndUpdate(

      { \_id },

      {

        $set: {

          imgURL,

        },

      }

    );

    res.send({ message: "Updated" });

  } catch (error) {}

});

router.post("/login", async (req, res, next) => {

  try {

    const { email, password } = req.body;

    if (!email || !password) {

      return res.send({ message: "Fill all the Fields" });

    }

    const team = await Team.findOne({ email: email });

    console.log("team", team);

    if (team) {

      const isMatch = await bcrypt.compare(password, team.password);

      if (!isMatch) {

        res.send({ message: "Password didn't match" });

      } else {

        res.send({ message: "Login Successsfully", team });

      }

    } else {

      return res.send({

        message: "Team not Registered, Please make an account first",

      });

    }

  } catch (err) {

    console.log('error',err);

    res.send(err);

  }

});

router.post("/register", async (req, res, next) => {

  try {

    const { name, captainName, email, password, phone, address } = req.body;

    const team = await Team.findOne({ email });

    if (team) {

      res.send({ message: "Team already Registered" });

    }

    //Creating. Enter data in database

    //teamname:teamname;

    else {

      const newTeam = new Team({

        name,

        captainName,

        email,

        password,

        phone,

        address,

      });

      await newTeam.save();

      res.send({ message: "Successfully Registered, Please Login Now" });

    }

  } catch (err) {

    console.log(err);

    res.send(err);

  }

});

router.post("/setStatus/:teamId", async (req, res, next) => {

  console.log("ROUTE ACTIVATED");

  const { status } = req.body;

  try {

    const { teamId } = req.params;

    const team = await Team.findById(teamId);

    const statusArray = team.statusArray;

    console.log(statusArray);

    statusArray.push(status);

    team.statusArray = statusArray;

    await team.save();

    res.send({ message: "Success", statusArray });

  } catch (err) {

    res.send(err);

  }

});

router.get("/getStatus/:teamId", async (req, res, next) => {

  try {

    const { teamId } = req.params;

    const team = await Team.findById(teamId);

    const statusArray = team.statusArray;

    res.send({ message: "Success", statusArray });

    console.log(statusArray);

  } catch (err) {

    res.send(err);

  }

});

// router.post("/setAbout/:teamId", async (req, res, next) => {

//     console.log("ROUTE ACTIVATED")

//     const {about} = req.body;

//     try{

//         const {teamId} = req.params;

//         const team = await Team.findById(teamId);

//         const aboutTeam = team.aboutTeam;

//         console.log(about)

//         aboutTeam.push(about);

//         team.aboutTeam = aboutTeam;

//         await team.save();

//         res.send({message: "Success", aboutTeam})

//     }

//     catch(err){

//         res.send(err);

//     }

// });

// router.get("/getAbout/:teamId", async (req, res, next) => {

//     try{

//         const {teamId} = req.params;

//         const team = await Team.findById(teamId);

//         const aboutTeam = team.aboutTeam;

//         res.send({message: "Success", aboutTeam})

//     }

//     catch(err){

//         res.send(err);

//     }

// })

export default router;

* **#Academybackend.js**

import express from "express";

import cors from "cors";

import mongoose from "mongoose";

import \* as CONSTANT from "./Constants/constants.js";

import bcrypt from "bcryptjs";

// Configuration

const app = express();

app.use(express.json());

app.use(express.urlencoded());

app.use(cors());

mongoose.connect(

  CONSTANT.CONNECTION\_URL,

  {

    useNewUrlParser: true,

    useUnifiedTopology: true,

  },

  () => {

    console.log("Database Connected");

  }

);

const academySchema = new mongoose.Schema({

  studentname: String,

  fathername: String,

  email: String,

  age: Number,

  contact: Number,

  address: String,

  myacademy: String,

});

const Academy = new mongoose.model("Academy", academySchema);

app.post("/academyformregister", async (req, res) => {

  //console.log(req.body)

  const { studentname, fathername, email, age, contact, address, myacademy } =

    req.body;

  Academy.findOne({ email: email }, (err, mongoacademy) => {

    if (mongoacademy) {

      res.send({

        message:

          "Student already Registered with this Email. Try something new. ",

      });

    }

    //Creating. Enter data in database

    else {

      const mongoacademy = new Academy({

        studentname,

        fathername,

        email,

        age,

        contact,

        address,

        myacademy,

      });

      mongoacademy.save((err) => {

        if (err) {

          res.send(err);

        } else {

          res.send({ message: "Student Successfull Registered to Academy" });

        }

      });

    }

  });

});

MongoClient.connect(url, (err, db) => {

  if (err) throw error;

  const dbo = db.db("bookmyslot");

  dbo

    .collection("teams")

    .aggregate([

      {

        $lookup: {

          from: "academies",

          localField: "myacademy",

          foreignField: "\_id",

          as: "new\_data",

        },

      },

    ])

    .toArray((err, res) => {

      if (err) throw err;

      console.log(JSON.stringify(res));

      db.close();

    });

});

app.listen(CONSTANT.PORT, () => {

  console.log("Academyformbackend started at port 9006");

});

* **#App.js**

import express from "express";

import mongoose from "mongoose";

import cors from "cors";

//Routes

import teamRoute from "./routes/team.js";

import groundRoute from "./routes/ground.js";

import academyRoute from "./routes/academy.js";

import \* as CONSTANT from "./Constants/constants.js";

const app = express();

// CORS Configuration (Yeh pehle hi add karein)

app.use(cors({

  origin: ['http://localhost:3000', 'http://localhost:3001'], // Specific allowed origins

  methods: ['GET', 'POST', 'PUT', 'DELETE'], // Allowed methods

  credentials: true // If using cookies/auth

}));

// Body Parsers

app.use(express.json());

app.use(express.urlencoded({ extended: true })); // <-- extended option add karein

// Home Route (Add this before other routes)

app.get("/", (req, res) => {

  res.send("Futsal API Working! Try /team, /ground or /academy");

});

// Routes

app.use("/team", teamRoute);

app.use("/ground", groundRoute);

app.use("/academy", academyRoute);

// 404 Handler

app.use("\*", (req, res) => {

  res.status(404).send({ message: "Wrong Endpoint" });

});

// DB Connection

mongoose.connect(CONSTANT.CONNECTION\_URL, {

  useNewUrlParser: true,

  useUnifiedTopology: true

})

.then(() => {

  console.log("Database Connected");

  app.listen(CONSTANT.PORT, () =>

    console.log(`Server running on port ${CONSTANT.PORT}`)

  );

})

.catch(err => console.log("Connection Failed:", err.message));

* **Package-lock.json**
* **Package.json**

1. **Frontend**

* **Public**

1. **Images folder**
2. **#index.html:**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="theme-color" content="#000000" />

    <meta

      name="description"

      content="Web site created using create-react-app"

    />

    <link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

    <link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

    <link

      href="https://fonts.googleapis.com/css2?family=Kalam&family=Roboto:wght@100&family=Sriracha&display=swap"

      rel="stylesheet"

    />

    <link

      href="https://fonts.googleapis.com/css2?family=Sarala&display=swap"

      rel="stylesheet"

    />

    <link

      href="https://fonts.googleapis.com/css2?family=Londrina+Shadow&family=Sarala&display=swap"

      rel="stylesheet"

    />

    <link

      href="https://fonts.googleapis.com/css2?family=Big+Shoulders+Inline+Display:wght@700&family=Londrina+Shadow&family=Poppins:wght@200&family=Sarala&display=swap"

      rel="stylesheet"

    />

    <link

      href="https://fonts.googleapis.com/css2?family=Bakbak+One&family=Big+Shoulders+Inline+Display:wght@700&family=Londrina+Shadow&family=Poppins:wght@700&family=Sarala&display=swap"

      rel="stylesheet"

    />

    <link

      href="https://fonts.googleapis.com/css2?family=Poppins:wght@600&display=swap"

      rel="stylesheet"

    />

    <link

      href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;600&display=swap"

      rel="stylesheet"

    />

    <link

      href="https://fonts.googleapis.com/css2?family=Anton&display=swap"

      rel="stylesheet"

    />

    <link

      href="https://fonts.googleapis.com/css2?family=Oswald:wght@200&display=swap"

      rel="stylesheet"

    />

    <link

      href="https://fonts.googleapis.com/css2?family=Lora&display=swap"

      rel="stylesheet"

    />

    <link

      href="https://fonts.googleapis.com/css2?family=Roboto:wght@100&display=swap"

      rel="stylesheet"

    />

    <!-- Bootstrap 5 CSS -->

    <link

      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"

      rel="stylesheet"

    />

    <!-- Bootstrap 5 JS Bundle (Optional for dropdowns/modals/etc.) -->

    <script

      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"

      defer

    ></script>

    <title>React App</title>

  </head>

  <body>

    <div id="root"></div>

  </body>

</html>

* **SRC**

1. **Assets**

**Bookingpic.png and groundpic.png**

1. **Constant**

**#constant.js:**

import axios from "axios";

export const API = axios.create({

  baseURL: 'http://localhost:5000',

  // baseURL: "https://football-backend-asad.herokuapp.com",

});

1. **Images**
2. **Pages**

**#Academydisplaypage.jsx:**

import React from "react";

import ArrowDropDownIcon from '@mui/icons-material/ArrowDropDown';

import GroupsIcon from '@mui/icons-material/Groups';

function Academydisplaypage ()

{

    return (

        <>

            <img id="academyfrontpagemainpic" src='./images/19.jpg' alt="futsal" />

            <div id = "academyfrontpagebox">

            <input type="text" placeholder="Search Academy by Name" id= "academysearch" />

                    <li id="academypageli">

                        <ul id="academypageul"><h1 id="groupsportspic"><GroupsIcon/></h1><span id="academypageulname">Kings Football Academy </span><h1 id="dropicon"><ArrowDropDownIcon/></h1></ul>

                        <ul id="academypageul"><h1 id="groupsportspic"><GroupsIcon/></h1><span id="academypageulname">Midfield Football Academy</span><h1 id="dropicon"><ArrowDropDownIcon/></h1></ul>

                        <ul id="academypageul"><h1 id="groupsportspic"><GroupsIcon/></h1><span id="academypageulname">Greenfield Football Academy</span><h1 id="dropicon"><ArrowDropDownIcon/></h1></ul>

                        <ul id="academypageul"><h1 id="groupsportspic"><GroupsIcon/></h1><span id="academypageulname">Strikers Football Academy</span><h1 id="dropicon"><ArrowDropDownIcon/></h1></ul>

                        <ul id="academypageul"><h1 id="groupsportspic"><GroupsIcon/></h1><span id="academypageulname">Vibrant Football Academy</span><h1 id="dropicon"><ArrowDropDownIcon/></h1></ul>

                        <ul id="academypageul"><h1 id="groupsportspic"><GroupsIcon/></h1><span id="academypageulname">Zaraj Football Academy</span><h1 id="dropicon"><ArrowDropDownIcon/></h1></ul>

                        <ul id="academypageul"><h1 id="groupsportspic"><GroupsIcon/></h1><span id="academypageulname">Raad Football Academy</span><h1 id="dropicon"><ArrowDropDownIcon/></h1></ul>

                        <ul id="academypageul"><h1 id="groupsportspic"><GroupsIcon/></h1><span id="academypageulname">Raad Football Academy</span><h1 id="dropicon"><ArrowDropDownIcon/></h1></ul>

                    </li>

            </div>

        </>

    );

}

export default Academydisplaypage;

**#Academyform.jsx:**

import React, { useState, useEffect } from "react";

import \* as CONSTANT from "../Constant/constant.js";

const Academyform = () => {

  const [ground, setGround] = useState([]);

  const [academy, setAcademy] = useState({

    studentname: "",

    fathername: "",

    email: "",

    age: "",

    contact: "",

    address: "",

    myacademy: "",

  });

  useEffect(() => {

    getGrounds();

  });

  const getGrounds = async () => {

    await CONSTANT.API.get("/ground/test").then((res) => {

      if (res.data.message) {

        alert("Something Went Wrong");

      } else {

        setGround(res.data);

      }

    });

  };

  const onSubmits = (event) => {

    event.preventDefault();

  };

  const inputEvent = (event) => {

    const name = event.target.name;

    const value = event.target.value;

    console.log(name, value);

    setAcademy({ ...academy, [name]: value });

  };

  const academyformregister = () => {

    const { studentname, fathername, email, age, contact, address, myacademy } =

      academy;

    if (

      studentname &&

      fathername &&

      email &&

      age &&

      contact &&

      address &&

      myacademy

    ) {

      CONSTANT.API.post("/academy/academyformregister", academy).then((res) => {

        alert(res.data.message);

      });

    } else {

      alert("Not posted");

    }

  };

  return (

    <>

      <div

        className="container-fluid d-flex justify-content-center align-items-center min-vh-100 "

        style={{ background: "#1e3046" }}

      >

        <div

          className=" bg-gradient shadow-lg "

          style={{ width: "100%", maxWidth: "500px", background: "#1e3046", color:"white", padding:"30px" }}

        >

          <div className="card-body">

            <h1 className="text-center mb-4 text-primary border-bottom pb-3">

              REGISTRATION FORM

            </h1>

            <form onSubmit={onSubmits} className="needs-validation" noValidate style={{color:"white"}}>

              {/\* Student Name \*/}

              <div className="mb-3">

                <label htmlFor="studentname" className="form-label">

                  Student Name

                </label>

                <input

                  type="text"

                  className="form-control p-2"

                  id="studentname"

                  name="studentname"

                  onChange={inputEvent}

                  value={academy.studentname}

                  required

                />

                <div className="invalid-feedback">

                  Please enter student name

                </div>

              </div>

              {/\* Father Name \*/}

              <div className="mb-3">

                <label htmlFor="fathername" className="form-label">

                  Father Name

                </label>

                <input

                  type="text"

                  className="form-control p-2"

                  id="fathername"

                  name="fathername"

                  onChange={inputEvent}

                  value={academy.fathername}

                  required

                />

                <div className="invalid-feedback">Please enter father name</div>

              </div>

              {/\* Email \*/}

              <div className="mb-3">

                <label htmlFor="email" className="form-label">

                  Email

                </label>

                <input

                  type="email"

                  className="form-control p-2"

                  id="email"

                  name="email"

                  onChange={inputEvent}

                  value={academy.email}

                  required

                />

                <div className="invalid-feedback">

                  Please enter a valid email

                </div>

              </div>

              {/\* Age \*/}

              <div className="mb-3">

                <label htmlFor="age" className="form-label">

                  Age

                </label>

                <input

                  type="number"

                  className="form-control p-2"

                  id="age"

                  name="age"

                  onChange={inputEvent}

                  value={academy.age}

                  min="1"

                  required

                />

                <div className="invalid-feedback">Please enter valid age</div>

              </div>

              {/\* Contact \*/}

              <div className="mb-3">

                <label htmlFor="contact" className="form-label">

                  Contact Number

                </label>

                <input

                  type="tel"

                  className="form-control p-2"

                  id="contact"

                  name="contact"

                  onChange={inputEvent}

                  value={academy.contact}

                  required

                />

                <div className="invalid-feedback">

                  Please enter contact number

                </div>

              </div>

              {/\* Address \*/}

              <div className="mb-3">

                <label htmlFor="address" className="form-label">

                  Address

                </label>

                <input

                  type="text"

                  className="form-control p-2"

                  id="address"

                  name="address"

                  onChange={inputEvent}

                  value={academy.address}

                  required

                />

                <div className="invalid-feedback">Please enter address</div>

              </div>

              {/\* Academy Select \*/}

              <div className="mb-4">

                <label htmlFor="myacademy" className="form-label">

                  Select Academy

                </label>

                <select

                  className="form-select p-2"

                  id="myacademy"

                  name="myacademy"

                  onChange={inputEvent}

                  value={academy.myacademy}

                  required

                >

                  <option value="">--Choose--</option>

                  {ground.map((item, index) => (

                    <option key={index} value={item.name}>

                      {item.name}

                    </option>

                  ))}

                </select>

                <div className="invalid-feedback">Please select an academy</div>

              </div>

              {/\* Submit Button \*/}

              <div className="d-grid">

                <button

                  type="submit"

                  className="btn btn-primary btn-lg"

                  onClick={academyformregister}

                >

                  Submit

                </button>

              </div>

            </form>

          </div>

        </div>

      </div>

    </>

  );

};

export default Academyform;

**#Academyfrontpage.jsx:**

import React from "react";

// import { useNavigate } from "react-router-dom";

function Academyfrontpage() {

  //   const navigate = useNavigate();

  const handleRegisterClick = () => {

    window.location.replace("/Academyform");

    // navigate("/Academyform");

  };

  return (

    <div

    style={{

      backgroundImage: "url('./Images/academy.png')",

      backgroundSize: "contain",

      backgroundRepeat: "no-repeat",

      backgroundPosition: "right",

      height: "100vh",

      width: "100vw",

      display: "flex",

      alignItems: "center",

      justifyContent: "flex-start",

      paddingLeft: "5%",

      color: "white",

    }}

  >

   <div

        style={{

          backgroundColor: "rgba(0, 0, 0, 0.6)",

          padding: "40px",

          borderRadius: "12px",

          maxWidth: "600px",

        }}

      >

        <h1

          style={{ fontSize: "48px", fontWeight: "bold", marginBottom: "20px" }}

        >

          Join The Academy

        </h1>

        <p

          style={{ fontSize: "18px", lineHeight: "1.8", marginBottom: "30px" }}

        >

          Today every ground has its academy with professional training staff.

          <br />

          Train with the best coaches to improve your skills & play on a higher

          level.

          <br />

          Get registered with any of the academies in twin cities through our

          website.

        </p>

        <button

          onClick={handleRegisterClick}

          style={{

            padding: "12px 30px",

            fontSize: "16px",

            fontWeight: "bold",

            border: "none",

            borderRadius: "8px",

            backgroundColor: "#007bff",

            color: "#fff",

            cursor: "pointer",

          }}

        >

          Register Now

        </button>

      </div>

  </div>

  );

}

export default Academyfrontpage;

**#academyusers.jsx:**

import React, { useState, useEffect } from "react";

import "bootstrap/dist/css/bootstrap.min.css";

import "../academyy.css";

import \* as CONSTANT from "../Constant/constant";

import { useLocation } from "react-router-dom/cjs/react-router-dom";

const StudentDataTable = () => {

  const [searchTerm, setSearchTerm] = useState("");

  const [sortConfig, setSortConfig] = useState({

    key: "studentname",

    direction: "ascending",

  });

  const [users, setUsers] = useState([]);

  const location = useLocation();

  const groundName = location.state?.groundName;

  useEffect(() => {

    getUsers();

  }, []);

  const getUsers = async () => {

    await CONSTANT.API.get(`/academy/users/${groundName}`).then((res) => {

      setUsers(res.data);

      console.log(res.data);

    });

  };

  // Sort function

  const sortedUsers = [...users].sort((a, b) => {

    if (!a[sortConfig.key] || !b[sortConfig.key]) return 0;

    if (a[sortConfig.key] < b[sortConfig.key]) {

      return sortConfig.direction === "ascending" ? -1 : 1;

    }

    if (a[sortConfig.key] > b[sortConfig.key]) {

      return sortConfig.direction === "ascending" ? 1 : -1;

    }

    return 0;

  });

  // Request a sort

  const requestSort = (key) => {

    let direction = "ascending";

    if (sortConfig.key === key && sortConfig.direction === "ascending") {

      direction = "descending";

    }

    setSortConfig({ key, direction });

  };

  // Filter users based on search term

  const filteredUsers = sortedUsers.filter((user) => {

    return (

      user.studentname.toLowerCase().includes(searchTerm.toLowerCase()) ||

      user.email.toLowerCase().includes(searchTerm.toLowerCase()) ||

      (user.fathername && user.fathername.toLowerCase().includes(searchTerm.toLowerCase())) ||

      (user.myacademy && user.myacademy.toLowerCase().includes(searchTerm.toLowerCase()))

    );

  });

  // Get sort direction indicator

  const getSortDirectionIcon = (name) => {

    if (sortConfig.key === name) {

      return sortConfig.direction === "ascending" ? "↑" : "↓";

    }

    return "";

  };

  return (

    <section className="student-table-section">

      <div className="container">

        <div className="row mb-4">

          <div className="col-12">

            <div className="student-table-header">

              <h2 className="student-table-title">

                <i className="bi bi-mortarboard-fill me-2"></i>

               Academy Student Directory

              </h2>

              <div className="student-search-container">

                <div className="input-group">

                  {/\* <span className="input-group-text bg-primary text-white">

                    <i className="bi bi-search"></i>

                  </span> \*/}

                  <input

                    type="text"

                    className="form-control student-search-input"

                    placeholder="Search students..."

                    value={searchTerm}

                    onChange={(e) => setSearchTerm(e.target.value)}

                  />

                </div>

              </div>

            </div>

          </div>

        </div>

        {filteredUsers.length === 0 ? (

          <div className="student-empty-state text-center py-5">

            <i className="bi bi-emoji-frown student-empty-icon"></i>

            <h3>No Students Found</h3>

            <p>Try adjusting your search criteria</p>

          </div>

        ) : (

          <>

            <div className="student-table-stats mb-3">

              <p className="mb-0">

                Showing <span className="text-primary fw-bold">{filteredUsers.length}</span> of{" "}

                <span className="fw-bold">{users.length}</span> students

              </p>

            </div>

            <div className="table-responsive student-table-container">

              <table className="table student-data-table">

                <thead>

                  <tr>

                    <th onClick={() => requestSort("studentname")} className="student-sortable-header">

                      <div className="d-flex align-items-center">

                        Student Name {getSortDirectionIcon("studentname")}

                        <i className="bi bi-arrow-down-up text-muted ms-1 sort-icon"></i>

                      </div>

                    </th>

                    <th onClick={() => requestSort("age")} className="student-sortable-header">

                      <div className="d-flex align-items-center">

                        Age {getSortDirectionIcon("age")}

                        <i className="bi bi-arrow-down-up text-muted ms-1 sort-icon"></i>

                      </div>

                    </th>

                    <th onClick={() => requestSort("contact")} className="student-sortable-header">

                      <div className="d-flex align-items-center">

                        Contact {getSortDirectionIcon("contact")}

                        <i className="bi bi-arrow-down-up text-muted ms-1 sort-icon"></i>

                      </div>

                    </th>

                    <th onClick={() => requestSort("email")} className="student-sortable-header">

                      <div className="d-flex align-items-center">

                        Email {getSortDirectionIcon("email")}

                        <i className="bi bi-arrow-down-up text-muted ms-1 sort-icon"></i>

                      </div>

                    </th>

                    <th onClick={() => requestSort("fathername")} className="student-sortable-header">

                      <div className="d-flex align-items-center">

                        Father's Name {getSortDirectionIcon("fathername")}

                        <i className="bi bi-arrow-down-up text-muted ms-1 sort-icon"></i>

                      </div>

                    </th>

                    <th onClick={() => requestSort("myacademy")} className="student-sortable-header">

                      <div className="d-flex align-items-center">

                        Academy {getSortDirectionIcon("myacademy")}

                        <i className="bi bi-arrow-down-up text-muted ms-1 sort-icon"></i>

                      </div>

                    </th>

                    <th>Actions</th>

                  </tr>

                </thead>

                <tbody>

                  {filteredUsers.map((student, index) => (

                    <tr key={index} className="student-table-row">

                      <td className="student-name-cell">

                        <div className="d-flex align-items-center">

                          <div className="student-avatar">

                            {student.studentname.charAt(0).toUpperCase()}

                          </div>

                          <div className="ms-2">

                            <div className="fw-bold">{student.studentname}</div>

                            <div className="student-id text-muted small">ID: {index + 1000}</div>

                          </div>

                        </div>

                      </td>

                      <td>{student.age || "-"}</td>

                      <td>

                        {student.contact ? (

                          <span>

                            <i className="bi bi-telephone-fill text-success me-1"></i>

                            {student.contact}

                          </span>

                        ) : (

                          "-"

                        )}

                      </td>

                      <td>

                        {student.email ? (

                          <a href={`mailto:${student.email}`} className="student-email">

                            <i className="bi bi-envelope-fill me-1"></i>

                            {student.email}

                          </a>

                        ) : (

                          "-"

                        )}

                      </td>

                      <td>{student.fathername || "-"}</td>

                      <td>

                        {student.myacademy && (

                          <span className="student-academy-badge">

                            {student.myacademy}

                          </span>

                        )}

                        {!student.myacademy && "-"}

                      </td>

                      <td>

                        <div className="student-actions">

                          <button className="btn btn-sm btn-primary me-1">

                            <i className="bi bi-pencil-square"></i>

                          </button>

                          <button className="btn btn-sm btn-danger">

                            <i className="bi bi-trash"></i>

                          </button>

                        </div>

                      </td>

                    </tr>

                  ))}

                </tbody>

              </table>

            </div>

          </>

        )}

      </div>

    </section>

  );

};

export default StudentDataTable;

**#Booking.jsx:**

import React, { useState, useEffect } from "react";

import abc from "../images/abc.jpg";

import \* as CONSTANT from "../Constant/constant";

const Booking = () => {

  const [Grounds, setGrounds] = useState([]);

  useEffect(() => {

    window.localStorage.removeItem("groundID");

    window.localStorage.removeItem("groundName");

    getAllGround();

  });

  const getAllGround = async () => {

    await CONSTANT.API.get("/ground/test").then((res) => {

      setGrounds(res.data);

    });

  };

  const Bookingg2 = (e) => {

    // e.preventDefault();

    window.location.replace("/Booking2");

  };

  const [filter, setFilter] = useState("");

  const searchText = (event) => {

    setFilter(event.target.value);

  };

  return (

    <section className="search-grounds-section">

    <div className="grounds-container">

      <div className="search-box-wrapper">

        <h3 className="search-heading">Find Ground for Booking</h3>

        <div className="search-input-wrapper">

          <input

            placeholder="Search a Ground for Booking"

            type="search"

            id=""

            onChange={searchText.bind(this)}

            className="ground-search-input"

          />

          <i className="search-icon fa fa-search"></i>

        </div>

      </div>

      <div className="grounds-grid-container">

        {Grounds.map((item, index) => {

          return (

            <div

              className="ground-item-wrapper"

              key={index}

              onClick={() => {

                window.localStorage.setItem("groundID", item.\_id);

                window.localStorage.setItem("groundName", item.name);

              }}

            >

              <div className="ground-card-item">

                <div className="ground-image-wrapper">

                  <img

                    src={item.imgURL ? item.imgURL : abc}

                    alt={item.name}

                    className="ground-thumbnail"

                  />

                </div>

                <div className="ground-details-container">

                  <h4 className="ground-title">{item.name}</h4>

                  <h4 className="location-label">Location</h4>

                  <p className="address-text">{item.address}</p>

                  <button className="book-ground-btn" onClick={Bookingg2}>

                    Book Now

                  </button>

                </div>

              </div>

            </div>

          );

        })}

      </div>

    </div>

  </section>

  );

};

export default Booking;

**#Booking2.jsx:**

import React, { useEffect, useState } from "react";

import abc from "../images/abc.jpg";

import abc3 from "../images/abc3.jpg";

import DatePicker from "react-datepicker";

import "react-datepicker/dist/react-datepicker.css";

import \* as CONSTANT from "../Constant/constant";

import "../booking.css";

import zIndex from "@mui/material/styles/zIndex";

const Booking2 = () => {

  const [Slots, setSlots] = useState([]);

  const [\_id, set\_ID] = useState(null);

  const [name, setName] = useState(null);

  const [teamName, setTeamName] = useState(null);

  const [inde, setinde] = useState([]);

  const [bookedBy, setBookedBy] = useState([]);

  const times = [

    "7-8:30 am",

    "9-9:30 am",

    "10-11:30 am",

    "12-1:30 pm",

    "2-2:30 pm",

    "3-3:30 pm",

    "4-4:30 pm",

    "5-5:30 pm",

    "6-6:30 pm",

    "7-8:30 pm",

    "9-10:30 pm",

    "11-12:30 pm",

  ];

  const [startDate, setStartDate] = useState(new Date());

  useEffect(() => {

    set\_ID(window.localStorage.getItem("groundID"));

    setName(window.localStorage.getItem("groundName"));

    setTeamName(window.localStorage.getItem("teamname"));

  });

  useEffect(() => {

    set\_ID(window.localStorage.getItem("groundID"));

    setTimeout(() => {

      getSlots();

    }, 1000);

  }, []);

  const getSlots = async () => {

    let today = new Date();

    let curent\_date =

      today.getFullYear() +

      "-" +

      (today.getMonth() + 1) +

      "-" +

      today.getDate();

    let id;

    let k;

    if (\_id == null) {

      id = window.localStorage.getItem("groundID");

    }

    if (\_id == null) {

      k = id;

    } else {

      k = \_id;

    }

    await CONSTANT.API.post(`/ground/getSlot`, {

      \_id: k,

      date: curent\_date,

    }).then((res) => {

      if (res.data.error) {

        console.log(res.data.error);

      } else {

        setSlots(res.data);

        console.log(res.data);

        setData(res.data);

      }

    });

  };

  const getSlotsFromDateChange = async (year, month, date) => {

    let curent\_date = year + "-" + month + "-" + date;

    await CONSTANT.API.post(`/ground/getSlot`, { \_id, date: curent\_date }).then(

      (res) => {

        if (res.data.error) {

          console.log(res.data.error);

        } else {

          setSlots(res.data);

          setData(res.data);

        }

      }

    );

  };

  const bookSlot = async (index) => {

    let curent\_date =

      startDate.getFullYear() +

      "-" +

      (startDate.getMonth() + 1) +

      "-" +

      startDate.getDate();

    await CONSTANT.API.post("/ground/bookSlot", {

      index,

      \_id,

      date: curent\_date,

      name: teamName,

    }).then((res) => {

      alert(res.data.message);

      getSlotsFromDateChange(

        startDate.getFullYear(),

        startDate.getMonth() + 1,

        startDate.getDate()

      );

    });

  };

  const undoSlot = async (index) => {

    let curent\_date =

      startDate.getFullYear() +

      "-" +

      (startDate.getMonth() + 1) +

      "-" +

      startDate.getDate();

    await CONSTANT.API.post("/ground/undoSlot", {

      index,

      \_id,

      date: curent\_date,

    }).then((res) => {

      alert(res.data.message);

      getSlotsFromDateChange(

        startDate.getFullYear(),

        startDate.getMonth() + 1,

        startDate.getDate()

      );

    });

  };

  const setData = (time) => {

    if (time.length == 0) {

      setinde([]);

    }

    let ij = [];

    let k = [];

    time.map((item, index) => {

      ij[item.in] = item.in;

      setinde([...ij]);

      k[item.in] = item.bookedby;

      setBookedBy([...k]);

    });

  };

  return (

    <>

      <div className="booking-section">

        <h1 className="groundname" style={{ marginTop: "-80px" }}>

          {name != null ? name : "hello"}

        </h1>

        <h2 className="dateheading">Select Date You Want To Select</h2>

        <form className="formdate d-flex justify-content-center mt-3">

          <div>

            <DatePicker

              id="datestart"

              minDate={new Date()}

              selected={startDate}

              onChange={(date) => {

                setStartDate(date);

                setinde([]);

                getSlotsFromDateChange(

                  date.getFullYear(),

                  date.getMonth() + 1,

                  date.getDate()

                );

              }}

              popperPlacement="bottom"

              popperModifiers={[

                {

                  name: "offset",

                  options: {

                    offset: [0, 10],

                  },

                },

                {

                  name: "preventOverflow",

                  options: {

                    boundary: "viewport",

                  },

                },

              ]}

              className="form-control"

            />

          </div>

        </form>

        {/\*

      <button className="allslots"> All Slots</button>

      <button className="avaslots"> Available Slots</button> \*/}

        <div className="slots-row" style={{ marginTop: "-40px" }}>

          {times.map((item, index) => (

            <div className="center3" key={index}>

              <div className="bookingcard">

                <div className="booking-image-container">

                  <img src={abc3} alt="Time slot" className="slot-image" />

                </div>

                <div className="container1">

                  <h4 className="time-heading">Time: {item}</h4>

                  {inde.length > 0 && inde[index] == index ? (

                    <div className="booked-info">

                      <h4

                        className="booked-by"

                        onClick={() => {

                          if (teamName == bookedBy[index]) {

                            undoSlot(index);

                          }

                        }}

                      >

                        Booked By: <b>{bookedBy[index]}</b>

                      </h4>

                    </div>

                  ) : (

                    <button

                      className="btnn2"

                      onClick={() => {

                        bookSlot(index);

                      }}

                    >

                      Book now

                    </button>

                  )}

                </div>

              </div>

            </div>

          ))}

        </div>

      </div>

    </>

  );

};

export default Booking2;

**#data.js:**

import img1 from '../images/foot.jpg'

import img2 from '../images/foot 3.jpg'

import img3 from '../images/foot2.jpg'

import img4 from '../images/foot 7.jpg'

import img5 from '../images/foot 6.jpg'

import img6 from '../images/foot 5.jpg'

import img7 from '../images/ground.jpg'

import img8 from '../images/foot 8.jpg'

import img9 from '../images/ground2.jpg'

import img10 from '../images/goal.jpg'

import img11 from '../images/players.jpg'

import img12 from '../images/d4.jfif'

const data ={

    cardData:

    [

        {

            id:1,

            img:img1,

            title:'Municipal Football Stadium',

            address:' located at Asghar Mall Scheme, Rawalpindi.',

            time: '7 - 8.30am',

        }

        ,

        {

            id:2,

            img:img2,

            title:'Greenfield Football Ground',

            address:'Ayub national park gate #4, Rawalpindi, 46000',

            time: '9 - 9.30am',

        },

        {

            id:3,

            img:img3,

            title:'The King Football Club',

            address:' H37J+QVP, Ayub National Park, Rawalpindi, Punjab',

            time: '10 - 11.30am',

        },

        {

            id:4,

            img:img4,

            title:'RMC Football ground',

            address:'J38H+4M3, Chaklala Cantt., Rawalpindi',

            time: '12 - 1.30pm',

        },

        {

            id:5,

            img:img5,

            title:'Midfield Football Club and Academy',

            address:' Grand Trunk Rd, near Ayub Park, Rawalpindi',

            time: '2 - 2.30pm',

        }

        ,

        {

            id:6,

            img:img6,

            title:'Smurfs football club',

            address:' Wallayat Complex, Phase 7 Bahria Town, Rawalpindi',

            time: '3 - 3.30pm',

        },

        {

            id:7,

            img:img7,

            title:'Zaraj Football Club',

            address:' Zaraj Housing Society, Zaraj Housing Society Sector B Islamabad, Rawalpindi',

            time: '4 - 4.30pm',

        },

        {

            id:8,

            img:img8,

            title:'Garrison Football Ground',

            address:'H3WV+G7V, Chaklala Cantt., Rawalpindi',

            time: '5 - 5.30pm',

        },

        {

            id:9,

            img:img9,

            title:'Raad Football Ground',

            address:'H3WV+G7V, Chaklala Cantt., Rawalpindi',

            time: '6 - 6.30pm',

        },

        {

            id:10,

            img:img10,

            title:'Velocity Football Ground',

            address:'H3WV+G7V, Chaklala Cantt., Rawalpindi',

            time: '7 - 8.30pm',

        },

        {

            id:11,

            img:img11,

            title:'F-6 Multipurpose Football Ground',

            address:'H3WV+G7V, Chaklala Cantt., Rawalpindi',

            time: '9 - 10.30pm',

        },

        {

            id:12,

            img:img12,

            title:'Jinnah Garden Football Ground',

            address:'H3WV+G7V, Chaklala Cantt., Rawalpindi',

            time: '11 - 12.30am',

        },

    ]

}

export default data;

**#donation.jsx:**

import React from "react";

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

const Donation = () => {

  const handleClick = async () => {

    try {

      const stripe = await loadStripe(

        "pk\_test\_51RL24xR06YjoyyuWJ0OleiToZ9U9sEA5HFO7WcXl4TMOsefr8mg2595zkQXK7xvHAY8XqD1TUYmxMBNTYexWRP5T00M0j1kMlW"

      );

      const { error } = await stripe.redirectToCheckout({

        lineItems: [

          {

            price: "price\_1RL947R06YjoyyuWVDhQGZku",

            quantity: 1,

          },

        ],

        mode: "payment",

        successUrl: "http://localhost:3000/donation",

        cancelUrl: "http://localhost:3000/login",

      });

      if (error) {

        console.error("Stripe Checkout error:", error);

        alert("Payment failed to initiate. Please try again.");

      }

    } catch (err) {

      console.error("Error loading Stripe:", err);

      alert("An error occurred. Please refresh the page.");

    }

  };

  return (

    <>

      <div>

        <img

          src="./images/iii.jpeg"

          className="bggimg"

          alt="Donation background"

        />

        <h1 className="centerr">D O N A T I O N</h1>

        <p className="centerr1">

          Sport has the power to change the world. It has the power to inspire.

          It has the power to unite people in a way that little else does. It

          speaks to youth in a language they understand. Sport can create hope

          where once there was only despair.

        </p>

      </div>

      <div>

        <button

          id="buttonn1"

          style={{ fontSize: "14px" }}

          onClick={handleClick}

        >

          Donate 500 PKR Now

        </button>

        <button id="buttonn2" style={{ fontSize: "14px" }}>

          Discover

        </button>

      </div>

    </>

  );

};

export default Donation;

**#Donationform.jsx:**

import React from 'react';

const Donationform = () => {

  return <>

    <img src='./images/iii.jpeg' className='bgimg1'/>

    <div className="container">

    <form>

        <label className='label' for="fname">Full Name</label>

            <input type="text" id="fname" name="fullname" placeholder="Your fullname.."/><br/>

        <label className='label'for="city">City</label>

        <select id="city" name="city">

            <option value="Islamabad">Islamabad</option>

            <option value="Rawalpindi">Rawalpindi</option>

        </select>

        <br/>

        <label className='label' for="Email">Email</label>

            <input type="text" id="email" name="Email" placeholder="Your email.."/> <br/>

        <label className='label' for="Address">Address</label>

            <input type="text" id="address" name="Address" placeholder="Your address.."/><br/>

        <label className='label' for="Phone">Phone No</label>

            <input type="number" id="phone" name="phone" placeholder="Your phone.."/><br/>

        <label className='label' for="Amount">Amount</label>

            <select id="Amount" name="Amount">

                <option value="100">RS 100</option>

                <option value="500">RS 500</option>

                <option value="1000">RS 1000</option>

            </select>

            <input type="submit" value="Submit"/>

    </form>

</div>

 </>

};

export default Donationform;

**#firebase.js:**

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

import { initializeApp } from "firebase/app";

import { getAnalytics } from "firebase/analytics";

import { getStorage } from "firebase/storage";

// TODO: Add SDKs for Firebase products that you want to use

// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration

// For Firebase JS SDK v7.20.0 and later, measurementId is optional

const firebaseConfig = {

  apiKey: "AIzaSyBaDj7MJ974TPeah62mw85GE6-zaKxP4\_c",

  authDomain: "futsal-field-reservation.firebaseapp.com",

  projectId: "futsal-field-reservation",

  storageBucket: "futsal-field-reservation.appspot.com",

  messagingSenderId: "1020710132153",

  appId: "1:1020710132153:web:c20c02c666245db8273e68",

  measurementId: "G-1R2R4L9YY9",

};

// Initialize Firebase

const app = initializeApp(firebaseConfig);

const analytics = getAnalytics(app);

export const storage = getStorage(app);

**#Footer.jsx:**

import React from "react";

function Footer ()

{

    return (

        <>

            <div id="footer">

        hello

            </div>

        </>

    );

};

export default Footer;

**#Groundform.jsx:**

import React, { useEffect, useState } from "react";

import axios from "axios";

import \* as CONTANT from "../Constant/constant.js";

import { useHistory } from "react-router-dom";

function Groundform() {

  const history = useHistory();

  useEffect(() => {

    window.localStorage.removeItem("dpUrl");

    window.localStorage.removeItem("coverUrl");

  }, []);

  const [ground, setGround] = useState({

    groundname: "",

    ownername: "",

    email: "",

    password: "",

    phone: "",

    address: "",

  });

  // // const [teamname, setteamname] = useState("")

  // // const [captainname, setcaptainname] = useState("")

  // // const [youremail, setyouremail] = useState("")

  // // const [yourpassword, setyourpassword] = useState("")

  // // const [mobileno, setmobileno] = useState("")

  // // const [address, setaddress] = useState("")

  const onSubmits = (event) => {

    event.preventDefault();

  };

  const inputEvent = (event) => {

    const name = event.target.name;

    const value = event.target.value;

    console.log(name, value);

    setGround({ ...ground, [name]: value });

  };

  const register = () => {

    const data = {

      name: ground.groundname,

      ownername: ground.ownername,

      email: ground.email,

      password: ground.password,

      phone: ground.phone,

      address: ground.address,

    };

    if (

      data.name &&

      data.ownername &&

      data.email &&

      data.password &&

      data.phone &&

      data.address

    ) {

      CONTANT.API.post("/ground/register", data).then((res) => {

        if (res.data.message == "Successfully Registered, Please Login Now") {

          alert(res.data.message);

          history.push("/login2");

        } else {

          alert(res.data.message);

        }

      });

    } else {

      alert("Not posted");

    }

  };

  return (

    <>

    <div

      className="d-flex justify-content-center align-items-center"

      style={{

        backgroundImage: "url('./images/groundspic.png')",

        backgroundSize: "cover",

        backgroundPosition: "center",

        minHeight: "100vh",

        position: "absolute",

        width: "100%",

        marginTop: "70px",

        paddingTop: "40px",

        paddingBottom: "40px",

        zIndex: "-100",

      }}

    >

      <div

        className="bg-white bg-opacity-25 p-5 rounded shadow"

        style={{

          backgroundColor: "rgba(255, 255, 255, 0.7)", // Semi-transparent background for contrast

          padding: "30px",

          borderRadius: "8px",

          boxShadow: "0px 4px 12px rgba(0, 0, 0, 0.1)", // Soft shadow around the form

          width: "80%", // Width of the form, adjust as necessary

          maxWidth: "420px", // Max width for responsiveness

          marginLeft: "430px",

        }}

      >

        <form id="teamform" onSubmit={onSubmits}>

          <h3 className="text-secondary fw-bold mb-4 text-center">

            SIGN UP AS GROUND'S OWNER

          </h3>

          <input

            type="text"

            name="groundname"

            autoComplete="off"

            placeholder="Enter Ground's Name"

            onChange={inputEvent}

            value={ground.groundname}

            className="form-control mb-3 p-3"

          />

          <input

            type="text"

            name="ownername"

            autoComplete="off"

            placeholder="Enter Ground's Owner Name"

            onChange={inputEvent}

            value={ground.ownername}

            className="form-control mb-3 p-3"

          />

          <input

            type="email"

            name="email"

            autoComplete="off"

            placeholder="Enter Your Email"

            onChange={inputEvent}

            value={ground.email}

            className="form-control mb-3 p-3"

          />

          <input

            type="password"

            name="password"

            autoComplete="off"

            placeholder="Enter Your Password"

            onChange={inputEvent}

            value={ground.password}

            className="form-control mb-3 p-3"

          />

          <input

            type="number"

            name="phone"

            autoComplete="off"

            placeholder="Enter Your Mobile Number"

            onChange={inputEvent}

            value={ground.phone}

            className="form-control mb-3 p-3"

          />

          <input

            type="text"

            name="address"

            autoComplete="off"

            placeholder="Enter Ground's Address"

            onChange={inputEvent}

            value={ground.address}

            className="form-control mb-4 p-3"

          />

          <div className="text-center">

            <button

              type="submit"

              className="btn btn-primary w-100 p-3"

              onClick={register}

            >

              Sign Up

            </button>

          </div>

        </form>

      </div>

    </div>

  </>

  );

}

export default Groundform;

**#Groundprofile.jsx:**

import React from "react";

import { useState, useEffect } from "react";

import Button from "@material-ui/core/Button";

import { Grid } from "@material-ui/core";

import { useSelector, useDispatch } from "react-redux";

import { storage } from ".//firebase";

import \* as CONSTANT from "../Constant/constant";

import { ref, uploadBytes, getDownloadURL, listAll } from "firebase/storage";

import { v4 } from "uuid";

import {

  setGroundId,

  setAddress,

  setOwnerName,

  setEmail,

  setPhone,

  setGroundName,

} from "../redux/slices/groundSlice";

import { getStorage, deleteObject } from "firebase/storage";

import { Link } from "react-router-dom/cjs/react-router-dom";

import { useHistory } from "react-router-dom/cjs/react-router-dom.min";

function Groundprofile() {

  const history = useHistory();

  const dispatch = useDispatch();

  const { localStorage } = window;

  const groundId = useSelector((state) => state?.groundSlice?.groundId);

  const groundName = useSelector((state) => state?.groundSlice?.groundname);

  //Dp

  const [dp\_Url, setDpUrl] = useState(localStorage.getItem("dpUrl"));

  const [dpUpload, setDpUpload] = useState(null);

  useEffect(() => {

    setTimeout(() => {

      setDpUrl(localStorage.getItem("dpUrl"));

    }, 1000);

  }, [dp\_Url]);

  const uploadDp = () => {

    if (dpUpload == null) return;

    // const dpRef = ref(storage, `teamdp/${dpUpload.name + v4()}`)

    const dpRef = ref(storage, `grounddp/${groundId}`);

    // const dpRef = ref(storage, `teamdp/${}`)

    uploadBytes(dpRef, dpUpload)

      .then(() => {

        getDownloadURL(dpRef)

          .then((url3) => {

            setDpUrl(url3);

            CONSTANT.API.post("/ground/dp", { \_id: groundId, imgURL: url3 });

            window.localStorage.setItem("dpUrl", url3);

          })

          .catch((error) => {

            console.log(error.message, "error getting while uploading");

          });

        setDpUpload(null);

      })

      .catch((error) => {

        console.log(error.message);

      });

  };

  //Cover

  const [cover\_Url, setCoverUrl] = useState(localStorage.getItem("coverUrl"));

  const [coverUpload, setCoverUpload] = useState(null);

  useEffect(() => {

    setTimeout(() => {

      setCoverUrl(localStorage.getItem("coverUrl"));

    }, 1000);

  }, [cover\_Url]);

  const uploadCover = () => {

    if (coverUpload == null) return;

    const coverRef = ref(storage, `groundcover/${groundId}`);

    uploadBytes(coverRef, coverUpload)

      .then(() => {

        getDownloadURL(coverRef)

          .then((url2) => {

            setCoverUrl(url2);

            window.localStorage.setItem("coverUrl", url2);

          })

          .catch((error) => {

            console.log(error.message, "error getting while uploading");

          });

        setCoverUpload(null);

      })

      .catch((error) => {

        console.log(error.message);

      });

  };

  //Timeline

  const [imageUpload, setImageUpload] = useState(null);

  const [imageList, setImageList] = useState([]);

  const [imagePath, setImagePath] = useState([]);

  const imageListRef = ref(storage, `groundtimeline/${groundId}`);

  const uploadImage = () => {

    if (imageUpload == null) return;

    const imageRef = ref(

      storage,

      `groundtimeline/${groundId}/${imageUpload.name + v4()}`

    );

    uploadBytes(imageRef, imageUpload)

      .then((snapshot) => {

        getDownloadURL(snapshot.ref)

          .then((url) => {

            const list = [...imageList];

            list.unshift(url);

            setImageList(list);

            localStorage.setItem("imageList", list);

          })

          .catch((error) => console.log("Error: ", error));

      })

      .catch((error) => console.log("Error: ", error));

  };

  useEffect(() => {

    setImageList([]);

    listAll(imageListRef).then((response) => {

      response.items.forEach((item) => {

        let path = imagePath;

        path.push(item.fullPath);

        setImagePath([...path]);

        getDownloadURL(item).then((url) => {

          let AllURL = imageList;

          AllURL.push(url);

          setImageList([...AllURL]);

        });

      });

    });

  }, []);

  const deletePicture = (e, index) => {

    e.preventDefault();

    const list = [...imageList];

    const paths = [...imagePath];

    const deletePath = paths[index];

    paths.splice(index, 1);

    setImagePath(paths);

    list.splice(index, 1);

    setImageList(list);

    const storage = getStorage();

    // Create a reference to the file to delete

    const desertRef = ref(storage, deletePath);

    deleteObject(desertRef)

      .then(() => {

        // File deleted successfully

        console.log("Deleted");

      })

      .catch((error) => {

        // Uh-oh, an error occurred!

      });

    window.localStorage.setItem("imageList", list);

  };

  return (

    <>

      <div className="ground-page">

        {/\* Hero Section with Cover Photo \*/}

        <div className="ground-hero-section">

          <div className="ground-cover-container">

            {cover\_Url ? (

              <a href={cover\_Url} target="\_blank" rel="noopener noreferrer">

                <img

                  src={cover\_Url}

                  className="ground-cover-image"

                  alt="Ground Cover"

                />

              </a>

            ) : (

              <div className="ground-cover-placeholder">

                <span>Add a cover photo for your ground</span>

              </div>

            )}

            <div className="ground-cover-controls">

              <input

                type="file"

                id="select-image2"

                className="ground-file-input"

                onChange={(event) => {

                  setCoverUpload(event.target.files[0]);

                }}

              />

              <label htmlFor="select-image2" className="me-2">

                <span className="btn btn-outline-light ground-select-btn">

                  <i className="bi bi-file-earmark me-1"></i> Select Cover

                </span>

              </label>

              <button

                className="btn btn-success ground-upload-btn"

                onClick={uploadCover}

              >

                <i className="bi bi-cloud-arrow-up-fill me-1"></i> Upload Cover

              </button>

            </div>

          </div>

        </div>

        <div className="container ground-content-container">

          <div className="row">

            {/\* Left Column - Ground Profile Info \*/}

            <div className="col-md-4">

              <div className="ground-profile-card">

                <div className="ground-dp-section">

                  <div className="ground-dp-container">

                    {dp\_Url ? (

                      <a

                        href={dp\_Url}

                        target="\_blank"

                        rel="noopener noreferrer"

                      >

                        <img

                          src={dp\_Url}

                          className="ground-dp-image"

                          alt="Ground Logo"

                        />

                      </a>

                    ) : (

                      <div className="ground-dp-placeholder">

                        <i className="bi bi-image"></i>

                      </div>

                    )}

                  </div>

                  <div className="ground-dp-controls">

                    <input

                      type="file"

                      id="select-image"

                      className="ground-file-input"

                      onChange={(event) => {

                        setDpUpload(event.target.files[0]);

                      }}

                    />

                    <label htmlFor="select-image" className="me-2">

                      <span className="btn btn-outline-dark btn-sm ground-select-btn">

                        <i className="bi bi-file-earmark me-1"></i> Select Image

                      </span>

                    </label>

                    <button

                      className="btn btn-success btn-sm ground-upload-btn"

                      onClick={uploadDp}

                    >

                      <i className="bi bi-cloud-arrow-up-fill me-1"></i> Upload

                      Logo

                    </button>

                  </div>

                </div>

                <div className="ground-info-section">

                  <h1 className="ground-name">{groundName}</h1>

                  <p className="ground-description">

                    Best Ground in Twin Cities

                    <br />

                    Here to achieve your dreams

                  </p>

                  <div className="ground-action-buttons">

                    <button

                      className="btn btn-primary ground-msg-btn"

                      onClick={() => {

                        window.open("http://localhost:3001/", "\_blank");

                      }}

                    >

                      <i className="bi bi-chat-dots-fill me-2"></i>Message

                    </button>

                    <button className="btn btn-info ground-booking-btn">

                      <Link to="/booking">

                        <i className="bi bi-calendar-check me-2"></i>Bookings{" "}

                      </Link>

                    </button>

                    <button

                      className="btn btn-secondary ground-academy-btn"

                      onClick={() =>

                        history.push("/academyuser", { groundName })

                      }

                    >

                      <i className="bi bi-trophy me-2"></i>Academy

                    </button>

                    {/\* <button className="btn btn-success ground-location-btn">

                    <i className="bi bi-geo-alt-fill me-2"></i>Location

                  </button> \*/}

                  </div>

                </div>

              </div>

            </div>

            {/\* Right Column - Timeline Gallery \*/}

            <div className="col-md-8">

              <div className="ground-timeline-section">

                <div className="ground-timeline-header">

                  <h2 className="ground-timeline-title">

                    <i className="bi bi-images me-2"></i>Ground Timeline

                  </h2>

                  <div className="ground-timeline-upload">

                    <input

                      type="file"

                      id="timeline-image-upload"

                      className="ground-file-input"

                      onChange={(event) => {

                        setImageUpload(event.target.files[0]);

                      }}

                    />

                    <label htmlFor="timeline-image-upload" className="me-2">

                      <span className="btn btn-outline-dark ground-select-btn">

                        <i className="bi bi-file-earmark-image me-1"></i> Select

                        Photo

                      </span>

                    </label>

                    <button

                      className="btn btn-success ground-upload-btn"

                      onClick={uploadImage}

                    >

                      <i className="bi bi-cloud-arrow-up-fill me-1"></i> Add to

                      Timeline

                    </button>

                  </div>

                </div>

                <div className="ground-timeline-gallery">

                  <div className="row g-4">

                    {imageList.length > 0 &&

                      imageList.map((url, index) => (

                        <div key={index} className="col-lg-4 col-md-6">

                          <div className="ground-gallery-item">

                            <div className="ground-gallery-image-container">

                              <a

                                href={url}

                                target="\_blank"

                                rel="noopener noreferrer"

                              >

                                <img

                                  src={url}

                                  className="ground-gallery-image"

                                  alt="Timeline"

                                />

                              </a>

                            </div>

                            <div className="ground-gallery-actions">

                              <button

                                className="btn btn-danger ground-delete-btn"

                                onClick={(e) => deletePicture(e, index)}

                              >

                                <i className="bi bi-trash me-1"></i> Remove

                              </button>

                            </div>

                          </div>

                        </div>

                      ))}

                    {imageList.length === 0 && (

                      <div className="ground-empty-gallery">

                        <div className="text-center p-5">

                          <i className="bi bi-camera-fill ground-empty-icon"></i>

                          <h4>No Images in Timeline</h4>

                          <p>

                            Upload your first image to start building your

                            ground's timeline.

                          </p>

                        </div>

                      </div>

                    )}

                  </div>

                </div>

              </div>

            </div>

          </div>

        </div>

      </div>

    </>

  );

}

export default Groundprofile;

**#groundsfrontpage.jsx:**

import React, { useState, useEffect } from "react";

import SportsIcon from "@mui/icons-material/Sports";

import \* as CONSTANT from "../Constant/constant.js";

import { useHistory } from "react-router-dom";

import GroupsIcon from "@mui/icons-material/Groups";

import ArrowDropDownIcon from "@mui/icons-material/ArrowDropDown";

function Groundsfrontpage() {

  const [ground, setGround] = useState([]);

  const [OnlineUser, setOnlineUser] = useState(null);

  const [from, setFrom] = useState(null);

  const history = useHistory();

  useEffect(() => {

    window.localStorage.removeItem("visitorID");

    window.localStorage.removeItem("visitordpUrl");

    window.localStorage.removeItem("visitorcoverUrl");

    window.localStorage.removeItem("visitiorName");

    setOnlineUser(window.localStorage.getItem("email"));

    setFrom(window.localStorage.getItem("from"));

    getAllGrounds();

  });

  const getAllGrounds = () => {

    CONSTANT.API.get("/ground/test").then((res) => {

      setGround(res.data);

    });

  };

  const ProfileView = (email, id, groundName) => {

    if (OnlineUser == email && from == "ground") {

      window.localStorage.setItem("visitiorName", groundName);

      history.push("/Groundprofile");

    } else {

      window.localStorage.setItem("visitorID", id);

      window.localStorage.setItem("visitiorName", groundName);

      history.push("/visitorgroundprofile");

    }

  };

  return (

    <>

      <div

        className="d-flex justify-content-center align-items-center"

        style={{

          backgroundImage: "url('./images/groundpicfrontmain.jpg')",

          backgroundSize: "cover",

          backgroundPosition: "center",

          minHeight: "100vh",

          width: "100%",

          position: "absolute",

          top: 0,

          left: 0,

          padding: "40px 0",

          overflow: "auto",

          marginTop: "70px",

        }}

      >

        <div

          className="bg-white bg-opacity-50 p-4 rounded shadow"

          style={{

            backgroundColor: "rgba(255, 255, 255, 0.7)",

            width: "90%",

            maxWidth: "550px",

            maxHeight: "80vh",

            overflowY: "auto",

            marginTop: "-400px",

          }}

        >

          <input

            type="text"

            placeholder="Search Team by Name"

            className="form-control mb-4 p-3"

            style={{

              padding: "40px",

              borderRadius: "8px",

              fontWeight: "bold",

            }}

            id="teamgroundsearch"

          />

          <ul className="list-group">

            {ground &&

              ground.map((item, index) => (

                <li

                  key={index}

                  className="list-group-item d-flex justify-content-between align-items-center mb-2"

                  style={{

                    cursor: "pointer",

                    // borderRadius: "10px",

                    // backgroundColor: "rgba(255, 255, 255, 0.9)",

                    boxShadow: "0 2px 6px rgba(0,0,0,0.1)",

                  }}

                  id="teamgroundpageul"

                  onClick={() => ProfileView(item.email, item.\_id, item.name)}

                >

                  <span>

                    <GroupsIcon style={{ marginRight: "10px" }} />

                    <strong>{item.name}</strong>

                  </span>

                  <ArrowDropDownIcon />

                </li>

              ))}

          </ul>

        </div>

      </div>

    </>

  );

}

export default Groundsfrontpage;

**#Home.jsx:**

import React from "react";

function Home() {

  const Bookingg = (e) => {

    e.preventDefault();

    window.location.replace("/Booking");

  };

  return (

    <>

       <section className="hero\_section">

        <div className="background-img">

          <img src="./images/home2.jpg" className="bg" />

          {/\* Buttons over image \*/}

          <div

            className="position-absolute top-50 start-50 translate-middle d-flex gap-4" style={{marginTop:"300px", marginLeft:"80px"}}

          >

            <a href="#" onClick={Bookingg}>

              <button className="btn btn-success btn-lg px-5 py-3 shadow">

                Book Now

              </button>

            </a>

            <a

              href="http://localhost:3001"

              target="\_blank"

              rel="noopener noreferrer"

            >

              <button className="btn btn-primary btn-lg px-5 py-3 shadow">

                Message

              </button>

            </a>

          </div>

        </div>

      </section>

    </>

  );

}

export default Home;

**#imagefirebase.js:**

import React, { useState } from "react";

import {storage} from './/firebase';

import { useEffect } from "react";

import {ref, uploadBytes, getDownloadURL, listAll, list} from "firebase/storage"

import { v4 } from  "uuid";

function Imagefirebase ()

{

    const [imageUpload, setImageUpload ] = useState(null);

    const [coverUpload, setCoverUpload ] = useState(null);

    const [url2, setUrl2] =useState (null);

    const [dpUpload, setDpUpload ] = useState(null);

    const [url3, setUrl3] =useState (null);

    const [imageList, setImageList] = useState([]);

    const imageListRef = ref(storage, "gallery/");

    const uploadImage = () =>

    {

        if (imageUpload == null) return;

        const imageRef = ref(storage, `gallery/${imageUpload.name + v4() }`)

        uploadBytes(imageRef, imageUpload).then ((snapshot)=> {

            getDownloadURL(snapshot.ref).then((url) => {

                setImageList((prev) => [...prev, url]);

              });

            });

          };

        useEffect(() => {

            listAll(imageListRef).then((response) => {

              response.items.forEach((item) => {

                getDownloadURL(item).then((url) => {

                  setImageList((prev) => [...prev, url]);

                });

              });

            });

          }, []);

          //For Cover

          const uploadCover = () =>

          {

            if (coverUpload == null) return;

            const coverRef = ref(storage, `cover/${coverUpload.name + v4() }`)

            uploadBytes(coverRef, coverUpload).then (()=> {

                getDownloadURL(coverRef).then((url2) => {

                    setUrl2(url2);

                  })

                  .catch((error) => {

                      console.log(error.message, "error getting while uploading")

                  });

                  setCoverUpload(null);

                })

                .catch((error) => {

                    console.log(error.message);

                });

              };

          //For Dp

          const uploadDp = () =>

          {

            if (dpUpload == null) return;

            const dpRef = ref(storage, `dp/${dpUpload.name + v4() }`)

            uploadBytes(dpRef, dpUpload).then (()=> {

                getDownloadURL(dpRef).then((url3) => {

                    setUrl3(url3);

                  })

                  .catch((error) => {

                      console.log(error.message, "error getting while uploading")

                  });

                  setDpUpload(null);

                })

                .catch((error) => {

                    console.log(error.message);

                });

          }

    return (

        <>

            <div>

                <input

                    type="file"

                    onChange={(event) => {

                                            setImageUpload(event.target.files[0])

                                        }}

                />

                <button onClick={uploadImage}> Upload Gallery </button>

                {imageList.map((url) => {

                    return  <img src={url} />

                })}

                        <br/>

                <input

                    type="file"

                    onChange={(event) => {

                                            setCoverUpload(event.target.files[0])

                                        }}

                />

                <button onClick={uploadCover}> Upload Cover </button>

                <img src ={url2} />

                                        <br/>

                <input

                    type="file"

                    onChange={(event) => {

                                            setDpUpload(event.target.files[0])

                                        }}

                />

                <button onClick={uploadDp}> Upload Dp </button>

                <img src ={url3} />

            </div>

        </>

    );

};

export default Imagefirebase;

**#login.jsx:**

import React, { useState, useEffect } from "react";

import Teamform from "./Teamform";

import { Link } from "react-router-dom";

import axios from "axios";

import \* as CONSTANT from "../Constant/constant.js";

import { useHistory } from "react-router-dom";

import { useDispatch, useSelector } from "react-redux";

import {

  setTeamId,

  setAddress,

  setCaptainName,

  setEmail,

  setPhone,

  setTeamName,

} from "../redux/slices/teamSlice";

import { getStorage, ref, getDownloadURL } from "firebase/storage";

import MuiAlert from "@material-ui/lab/Alert";

import Snackbar from "@material-ui/core/Snackbar";

import { Typography } from "@material-ui/core";

import { ToastContainer, toast } from "react-toastify";

import "react-toastify/dist/ReactToastify.css";

const Login = ({ setTeamLogin }) => {

  const teamId = useSelector((state) => state?.teamSlice.teamId);

  const history = useHistory();

  const dispatch = useDispatch();

  useEffect(() => {

    window.localStorage.removeItem("dpUrl");

    window.localStorage.removeItem("coverUrl");

  }, []);

  const [user, setUser] = useState({

    email: "",

    password: "",

  });

  //Snackbar

  const [snackBarstate, setSnackBar] = useState({

    open: false,

    vertical: "top",

    horizontal: "right",

    type: "success",

    message: "",

  });

  const { vertical, horizontal, open } = snackBarstate;

  const handleCloseSnackBar = () => {

    setSnackBar({ ...snackBarstate, open: false });

  };

  const inputEvent2 = (event) => {

    const name = event.target.name;

    const value = event.target.value;

    setUser({ ...user, [name]: value });

  };

  const onSubmits = (event) => {

    event.preventDefault();

  };

  const { localStorage } = window;

  const setTeamData = (res) => {

    localStorage.setItem("teamId", res.data.team.\_id);

    localStorage.setItem("teamname", res.data.team.name);

    localStorage.setItem("captainname", res.data.team.captainName);

    localStorage.setItem("email", res.data.team.email);

    localStorage.setItem("phone", res.data.team.phone);

    localStorage.setItem("address", res.data.team.address);

    localStorage.setItem("from", "team");

    localStorage.setItem("login", true);

    setDpUrl(res.data.team.\_id);

    setCoverUrl(res.data.team.\_id);

    dispatch(setTeamId(res.data.team.\_id));

    dispatch(setTeamName(res.data.team.name));

    dispatch(setCaptainName(res.data.team.captainName));

    dispatch(setEmail(res.data.team.email));

    dispatch(setPhone(res.data.team.phone));

    dispatch(setAddress(res.data.team.address));

  };

  const setDpUrl = (teamId) => {

    const storage = getStorage();

    getDownloadURL(ref(storage, `teamdp/${teamId}`))

      .then((url) => {

        console.log("DP URL: ", url);

        localStorage.setItem("dpUrl", url);

      })

      .catch((err) => console.log(err));

  };

  const setCoverUrl = (teamId) => {

    const storage = getStorage();

    getDownloadURL(ref(storage, `teamcover/${teamId}`))

      .then((url) => {

        console.log("Cover URL: ", url);

        localStorage.setItem("coverUrl", url);

      })

      .catch((err) => console.log(err));

  };

  const login = () => {

    CONSTANT.API.post("/team/login", user).then((res) => {

      //localStorage.setItem("id",res.data.team.\_id)

      if (res.data.message == "Login Successsfully") {

        console.log(res.data);

        setTeamData(res);

        setTeamLogin(res.data.team);

        setSnackBar({

          ...snackBarstate,

          type: "success",

          message: "Success: Login Successfully ",

          open: true,

        });

        toast.success("Login Successfully");

        history.push("/profile");

      } else {

        alert(res.data.message);

      }

    });

  };

  return (

    <>

      <div

        className="d-flex justify-content-center align-items-center"

        style={{

          backgroundImage: "url('./images/loginpic.png')",

          backgroundSize: "cover",

          backgroundPosition: "center",

          minHeight: "100vh",

          position: "absolute",

          width: "100%",

          marginTop: "70px",

          paddingTop: "40px",

          paddingBottom: "40px",

          zIndex: "-100",

        }}

      >

        {/\* Centered Form Box \*/}

        <div

          className="bg-white bg-opacity-25 p-5 rounded shadow"

          style={{ width: "100%", maxWidth: "420px", marginLeft: "430px" }}

        >

          <form id="teamform" onSubmit={onSubmits}>

            <h3 className="text-secondary fw-bold mb-4 text-center">

              Sign In as Team

            </h3>

            <div className="mb-3" style={{marginTop:"30px"}}>

              <input

                type="text"

                name="email"

                placeholder="Enter Your Email"

                onChange={inputEvent2}

                value={user.email}

                className="form-control form-control-lg p-3 shadow-sm"

              />

            </div>

            <div className="mb-3">

              <input

                type="password"

                name="password"

                placeholder="Enter Your Password"

                onChange={inputEvent2}

                value={user.password}

                className="form-control form-control-lg p-3 shadow-sm"

              />

            </div>

            <p className="text-end mb-2">

              <a href="#" className="text-decoration-none text-primary">

                Forgot Password?

              </a>

            </p>

            <button

              type="submit"

              className="btn btn-primary btn-lg w-100 mb-3"

              onClick={login}

            >

              Log in

            </button>

            <p className="text-center">

              No account?{" "}

              <Link

                to="/teamform"

                className="text-decoration-none text-success fw-bold"

              >

                Create an Account

              </Link>

            </p>

          </form>

        </div>

      </div>

      {/\* Snackbar (Toast Alert) \*/}

      <Snackbar

        open={open}

        anchorOrigin={{ vertical, horizontal }}

        autoHideDuration={5000}

        onClose={handleCloseSnackBar}

        key={vertical + horizontal}

      >

        <MuiAlert

          elevation={6}

          variant="filled"

          onClose={handleCloseSnackBar}

          severity={snackBarstate.type}

        >

          <Typography style={{ color: "#fff" }}>

            {snackBarstate.message}

          </Typography>

        </MuiAlert>

      </Snackbar>

    </>

  );

};

export default Login;

**#login2.jsx:**

import React, { useState, useEffect } from "react";

import { Link } from "react-router-dom";

import axios from "axios";

import { useHistory } from "react-router-dom";

import \* as CONTANT from "../Constant/constant.js";

import { useDispatch, useSelector } from "react-redux";

import {

  setGroundId,

  setAddress,

  setGroundName,

  setEmail,

  setPhone,

  setOwnerName,

} from "../redux/slices/groundSlice";

import { getStorage, ref, getDownloadURL } from "firebase/storage";

const Login2 = ({ setGroundLogin }) => {

  const groundId = useSelector((state) => state?.groundSlice.groundId);

  const history = useHistory();

  const dispatch = useDispatch();

  useEffect(() => {

    window.localStorage.removeItem("dpUrl");

    window.localStorage.removeItem("coverUrl");

  }, []);

  const [user2, setUser2] = useState({

    email: "",

    password: "",

  });

  const inputEvent2 = (event) => {

    const name = event.target.name;

    const value = event.target.value;

    setUser2({ ...user2, [name]: value });

  };

  const onSubmits = (event) => {

    event.preventDefault();

  };

  const { localStorage } = window;

  const setGroundData = (res) => {

    localStorage.setItem("groundId", res.data.ground.\_id);

    localStorage.setItem("groundname", res.data.ground.name);

    localStorage.setItem("ownername", res.data.ground.ownername);

    localStorage.setItem("email", res.data.ground.email);

    localStorage.setItem("phone", res.data.ground.phone);

    localStorage.setItem("address", res.data.ground.address);

    localStorage.setItem("from", "ground");

  localStorage.setItem("login", true);

    setDpUrl(res.data.ground.\_id);

    setCoverUrl(res.data.ground.\_id);

    dispatch(setGroundId(res.data.ground.\_id));

    dispatch(setGroundName(res.data.ground.name));

    dispatch(setOwnerName(res.data.ground.ownername));

    dispatch(setEmail(res.data.ground.email));

    dispatch(setPhone(res.data.ground.phone));

    dispatch(setAddress(res.data.ground.address));

  };

  const setDpUrl = (groundId) => {

    const storage = getStorage();

    getDownloadURL(ref(storage, `grounddp/${groundId}`))

      .then((url) => {

        console.log("DP URL: ", url);

        localStorage.setItem("dpUrl", url);

      })

      .catch((err) => console.log(err));

  };

  const setCoverUrl = (groundId) => {

    const storage = getStorage();

    getDownloadURL(ref(storage, `groundcover/${groundId}`))

      .then((url) => {

        console.log("Cover URL: ", url);

        localStorage.setItem("coverUrl", url);

      })

      .catch((err) => console.log(err));

  };

  const login2 = () => {

    CONTANT.API.post("/ground/login2", user2).then((res) => {

      if (res.data.message == "Login Successsfully") {

        console.log(res.data.ground);

        setGroundData(res);

        alert(res.data.message);

        setGroundLogin(res.data.ground);

        history.push("/groundprofile");

      } else {

        alert(res.data.message);

      }

    });

  };

  return (

      <>

          <div

            className="d-flex justify-content-center align-items-center"

            style={{

              backgroundImage: "url('./images/Loginpagepic.png')",

              backgroundSize: "cover",

              backgroundPosition: "center",

              minHeight: "100vh",

              position: "absolute",

              width: "100%",

              marginTop: "70px",

              paddingTop: "40px",

              paddingBottom: "40px",

              zIndex: "-100",

            }}

          >

            {/\* Centered Form Box \*/}

            <div

              className="bg-white bg-opacity-25 p-5 rounded shadow"

              style={{ width: "100%", maxWidth: "420px", marginLeft: "430px" }}

            >

              <form id="teamform" onSubmit={onSubmits}>

                <h3 className="text-secondary fw-bold mb-4 text-center">

                  Sign In as Ground Owner

                </h3>

                <div className="mb-3" style={{marginTop:"30px"}}>

                  <input

                    type="text"

                    name="email"

                    placeholder="Enter Your Email"

                    onChange={inputEvent2}

                    value={user2.email}

                    className="form-control form-control-lg p-3 shadow-sm"

                  />

                </div>

                <div className="mb-3">

                  <input

                    type="password"

                    name="password"

                    placeholder="Enter Your Password"

                    onChange={inputEvent2}

                    value={user2.password}

                    className="form-control form-control-lg p-3 shadow-sm"

                  />

                </div>

                <p className="text-end mb-2">

                  <a href="#" className="text-decoration-none text-primary">

                    Forgot Password?

                  </a>

                </p>

                <button

                  type="submit"

                  className="btn btn-primary btn-lg w-100 mb-3"

                  onClick={login2}

                >

                  Log in

                </button>

                <p className="text-center">

                  No account?{" "}

                  <Link

                    to="/groundform"

                    className="text-decoration-none text-success fw-bold"

                  >

                    Create an Account

                  </Link>

                </p>

              </form>

            </div>

          </div>

        </>

  );

};

export default Login2;

**#menu.jsx:**

import React from "react";

import { useDispatch, useSelector } from "react-redux";

import { Link } from "react-router-dom";

import {

  setTeamId,

  setAddress,

  setCaptainName,

  setEmail,

  setPhone,

  setTeamName,

} from ".././redux/slices/teamSlice";

import {

  setGroundId,

  setOwnerName,

  setGroundName,

} from ".././redux/slices/groundSlice";

function Menu() {

  const login = localStorage.getItem("login");

  const dispatch = useDispatch();

  const Logout = () => {

    const { localStorage } = window;

    localStorage.clear();

    window.localStorage.clear();

    dispatch(setTeamId(""));

    dispatch(setTeamName(""));

    dispatch(setCaptainName(""));

    dispatch(setEmail(""));

    dispatch(setPhone(""));

    dispatch(setAddress(""));

    dispatch(setGroundId(""));

    dispatch(setGroundName(""));

    dispatch(setOwnerName(""));

    window.location.replace("/");

  };

  const Login = (e) => {

    window.location.replace("/Mylogin");

  };

  return (

    <>

      <nav className="main-nav">

        <div className="logo">

          <Link to="/">

            <img src="./images/bms.png" alt="Logo" />

          </Link>

        </div>

        <div className="menu-link">

          <ul>

            <li>

              <Link to="/"> Home </Link>{" "}

            </li>

            <li>

              <Link to="/teams"> TEAMS </Link>{" "}

            </li>

            <li>

              <Link to="/grounds"> GROUNDS </Link>

            </li>

            <li>

              <Link to="/academies"> ACADEMY </Link>

            </li>

            <li>

              <Link to="/donation"> DONATION </Link>

            </li>

            <li>

              <Link to="/booking"> BOOKINGS </Link>

            </li>

          </ul>

        </div>

        {/\*login button\*/}

        <div className="login">

          <a className="cta">

            <button

              className={

                login

                  ? "btn btn-danger btn-lg px-5 py-3"

                  : "btn btn-success btn-lg px-5 py-3"

              }

              onClick={login ? Logout : Login}

            >

              {login ? "Logout" : "Login"}

            </button>

          </a>

        </div>

      </nav>

    </>

  );

}

export default Menu;

**#Mylogin.jsx:**

import React from "react";

function Mylogin() {

  const login1 = (e) => {

    e.preventDefault();

    window.location.replace("/Login");

  };

  const login2 = (e) => {

    e.preventDefault();

    window.location.replace("/Login2");

  };

  return (

    <>

      {/\* Full Background Image \*/}

      <div

        className="vh-100 vw-100 d-flex justify-content-center align-items-center"

        style={{

          backgroundImage: "url('./images/grass3.jpg')",

          backgroundSize: "cover",

          backgroundPosition: "center",

        }}

      >

        {/\* Centered Login Buttons \*/}

        <div className="text-center bg-white bg-opacity-50 rounded shadow">

          <div style={{ padding: "8rem" }}>

            <span

              className="mb-5"

              style={{ fontWeight: "bold", fontSize: "3rem" }}

            >

              Login As

            </span>

            <button

              className="btn btn-success w-100 py-3 mb-4"

              style={{ fontSize: "1.3rem", marginTop: "30px" }}

              onClick={login1}

            >

              Login As Team

            </button>

            <button

              className="btn btn-primary w-100 py-3"

              style={{ fontSize: "1.3rem" }}

              onClick={login2}

            >

              Login As Ground

            </button>

          </div>

        </div>

      </div>

    </>

  );

}

export default Mylogin;

**#profile.jsx:**

import React from "react";

import AddIcon from "@mui/icons-material/Add";

import { useState, useEffect } from "react";

import Button from "@material-ui/core/Button";

import { Grid } from "@material-ui/core";

import axios from "axios";

import \* as CONTANT from "../Constant/constant.js";

import { useHistory } from "react-router-dom";

import { fontSize } from "@mui/system";

import { useSelector } from "react-redux";

import { ToastContainer, toast } from "react-toastify";

import "react-toastify/dist/ReactToastify.css";

import { storage } from ".//firebase";

import {

  ref,

  uploadBytes,

  getDownloadURL,

  listAll,

  list,

} from "firebase/storage";

import { v4 } from "uuid";

import { useDispatch } from "react-redux";

import {

  setTeamId,

  setAddress,

  setCaptainName,

  setEmail,

  setPhone,

  setTeamName,

} from "../redux/slices/teamSlice";

import { getStorage, deleteObject } from "firebase/storage";

import "../timeline.css";

import { Link } from "react-router-dom/cjs/react-router-dom.js";

// import firebase from "firebase/compat/app";

// import "firebase/compat/auth";

// import "firebase/compat/firestore";

function Profile() {

  const storagee = getStorage();

  const dispatch = useDispatch();

  const { localStorage } = window;

  const userEmail = useSelector((state) => state?.utilitySlice?.userEmail);

  const storedTeamId = localStorage.getItem("teamId");

  const reduxTeamId = useSelector((state) => state?.teamSlice?.teamId);

  const teamId = storedTeamId || reduxTeamId;

  console.log("teamId", teamId);

  const storedTeamName = localStorage.getItem("teamname");

  const reduxTeamName = useSelector((state) => state?.teamSlice?.teamname);

  const team = storedTeamName || reduxTeamName;

  console.log("team", team);

  //Dp

  const [dp\_Url, setDpUrl] = useState(localStorage.getItem("dpUrl"));

  const [dpUpload, setDpUpload] = useState(null);

  const [url3, setUrl3] = useState(null);

  useEffect(() => {

    setTimeout(() => {

      setDpUrl(localStorage.getItem("dpUrl"));

      getAllStatus();

    }, 1000);

  }, [dp\_Url]);

  const uploadDp = () => {

    if (dpUpload == null) return;

    // const dpRef = ref(storage, `teamdp/${dpUpload.name + v4()}`)

    const dpRef = ref(storage, `teamdp/${teamId}`);

    // const dpRef = ref(storage, `teamdp/${}`)

    uploadBytes(dpRef, dpUpload)

      .then(() => {

        getDownloadURL(dpRef)

          .then((url3) => {

            setDpUrl(url3);

            window.localStorage.setItem("dpUrl", url3);

            CONTANT.API.post("/team/dp", { \_id: teamId, imgURL: url3 });

          })

          .catch((error) => {

            console.log(error.message, "error getting while uploading");

          });

        setDpUpload(null);

      })

      .catch((error) => {

        console.log(error.message);

      });

  };

  //Cover

  const [cover\_Url, setCoverUrl] = useState(localStorage.getItem("coverUrl"));

  const [coverUpload, setCoverUpload] = useState(null);

  const [url2, setUrl2] = useState(null);

  useEffect(() => {

    getAllStatus();

  });

  useEffect(() => {

    setTimeout(() => {

      setCoverUrl(localStorage.getItem("coverUrl"));

      getAllStatus();

    }, 1000);

  }, [cover\_Url]);

  const uploadCover = () => {

    if (coverUpload == null) return;

    const coverRef = ref(storage, `teamcover/${teamId}`);

    uploadBytes(coverRef, coverUpload)

      .then(() => {

        getDownloadURL(coverRef)

          .then((url2) => {

            setCoverUrl(url2);

            window.localStorage.setItem("coverUrl", url2);

          })

          .catch((error) => {

            console.log(error.message, "error getting while uploading");

          });

        setCoverUpload(null);

      })

      .catch((error) => {

        console.log(error.message);

      });

  };

  //Delete Cover

  const Deletecover = () => {

    const storage = getStorage();

    // Create a reference to the file to delete

    const desertRef = ref(storage, `teamcover/${teamId}`);

    // Delete the file

    deleteObject(desertRef)

      .then(() => {

        window.localStorage.removeItem("coverUrl");

        setCoverUrl();

        // File deleted successfully

      })

      .catch((error) => {

        // Uh-oh, an error occurred!

      });

  };

  //Delete Dp

  const Deletedp = () => {

    const storage = getStorage();

    // Create a reference to the file to delete

    const desertRef = ref(storage, `teamdp/${teamId}`);

    // Delete the file

    deleteObject(desertRef)

      .then(() => {

        // File deleted successfully

        window.localStorage.removeItem("dpUrl");

        setDpUrl();

      })

      .catch((error) => {

        // Uh-oh, an error occurred!

      });

    // window.location.reload(false);

  };

  //Status

  const [status, setStatus] = useState("");

  const [statusArray, setStatusArray] = useState([]);

  const getAllStatus = async () => {

    const response = await CONTANT.API.get(`/team/getStatus/${teamId}`);

    if (response?.data) {

      setStatusArray(response.data.statusArray.reverse());

    }

  };

  //Timeline

  const [imageUpload, setImageUpload] = useState(null);

  const [imageList, setImageList] = useState([]);

  const [imagePath, setImagePath] = useState([]);

  const imageListRef = ref(storage, `teamtimeline/${teamId}`);

  const uploadImage = () => {

    if (imageUpload == null) return;

    const imageRef = ref(

      storage,

      `teamtimeline/${teamId}/${imageUpload.name + v4()}`

    );

    uploadBytes(imageRef, imageUpload).then((snapshot) => {

      getDownloadURL(snapshot.ref).then((url) => {

        const list = [...imageList];

        list.unshift(url);

        setImageList(list);

        window.localStorage.setItem("imageList", list);

      });

    });

  };

  const deletePicture = async (e, index) => {

    e.preventDefault();

    const list = [...imageList];

    const paths = [...imagePath];

    const deletePath = paths[index];

    paths.splice(index, 1);

    setImagePath(paths);

    list.splice(index, 1);

    setImageList(list);

    const storage = getStorage();

    // Create a reference to the file to delete

    const desertRef = ref(storage, deletePath);

    deleteObject(desertRef)

      .then(() => {

        // File deleted successfully

        console.log("Deleted");

      })

      .catch((error) => {

        // Uh-oh, an error occurred!

      });

    window.localStorage.setItem("imageList", list);

  };

  const deleteStatus = (e, index) => {

    e.preventDefault();

    const list = [...statusArray];

    list.splice(index, 1);

    setStatusArray(list);

    const url = list[index];

    console.log(url);

    // let imageRef = storagee.refFromURL(url);

    // imageRef.delete();

    window.localStorage.setItem("statusArray", list);

  };

  useEffect(() => {

    listAll(imageListRef).then((response) => {

      response.items.forEach((item) => {

        let path = imagePath;

        path.push(item.fullPath);

        setImagePath([...path]);

        getDownloadURL(item).then((url) => {

          let AllURL = imageList;

          AllURL.push(url);

          setImageList([...AllURL]);

        });

      });

    });

  }, []);

  const logout = () => {

    const { localStorage } = window;

    localStorage.clear();

    dispatch(setTeamId(""));

    dispatch(setTeamName(""));

    dispatch(setCaptainName(""));

    dispatch(setEmail(""));

    dispatch(setPhone(""));

    dispatch(setAddress(""));

  };

  const addStatus = async () => {

    const list = [...statusArray];

    list.unshift(status);

    setStatusArray(list);

    window.localStorage.setItem("statusArray", list);

    const response = await CONTANT.API.post(`/team/setStatus/${teamId}`, {

      status,

    });

    setStatus("");

    console.log(response);

  };

  return (

    <>

      <div className="profile-page bg-light">

        <ToastContainer />

        {/\* Header with navigation \*/}

        {/\* <nav className="navbar navbar-expand-lg navbar-dark bg-primary sticky-top">

          <div className="container">

            <a className="navbar-brand fw-bold" href="#">

              TeamConnect

            </a>

            <button

              className="btn btn-outline-light ms-auto"

              onClick={() => logout()}

            >

              Logout

            </button>

          </div>

        </nav> \*/}

        <div className="container mt-3">

          {/\* Cover Photo Section \*/}

          <div className="position-relative mb-5">

            <div className="cover-photo-container rounded shadow">

              {cover\_Url ? (

                <a href={cover\_Url} target="\_blank" rel="noopener noreferrer">

                  <img

                    src={cover\_Url}

                    className="cover-photo w-100"

                    alt="Cover"

                  />

                </a>

              ) : (

                <div className="cover-placeholder"></div>

              )}

              <div

                className="cover-photo-controls position-absolute bottom-0 end-0 m-3"

                style={{ zIndex: "99" }}

              >

                <input

                  type="file"

                  id="select-image2"

                  className="d-none"

                  onChange={(event) => {

                    setCoverUpload(event.target.files[0]);

                  }}

                />

                <label htmlFor="select-image2" className="me-2">

                  <span className="btn btn-light btn-sm">

                    <i className="bi bi-file-earmark me-1"></i> Select

                  </span>

                </label>

                <button

                  className="btn btn-primary btn-sm me-2"

                  onClick={uploadCover}

                >

                  <i className="bi bi-cloud-arrow-up-fill me-1"></i>

                </button>

                <button

                  className="btn btn-outline-danger"

                  title="Remove Cover"

                  onClick={Deletecover}

                >

                  <i className="bi bi-trash-fill me-1"></i>

                </button>

              </div>

            </div>

            {/\* Profile Picture & Team Info \*/}

            <div className="profile-header">

              <div className="d-flex align-items-end">

                <div className="profile-picture-container position-relative">

                  {dp\_Url ? (

                    <a href={dp\_Url} target="\_blank" rel="noopener noreferrer">

                      <img

                        src={dp\_Url}

                        className="profile-picture rounded-circle border border-4 border-white"

                        alt="Profile"

                        style={{ cursor: "pointer" }}

                      />

                    </a>

                  ) : (

                    <div className="profile-picture-placeholder rounded-circle border border-4 border-white"></div>

                  )}

                  <div className="profile-picture-edit">

                    <input

                      type="file"

                      id="select-image"

                      className="d-none"

                      onChange={(event) => {

                        setDpUpload(event.target.files[0]);

                      }}

                    />

                    <div className="mt-2 d-flex">

                      <label htmlFor="select-image" className="me-2">

                        <span className="btn btn-sm btn-light">

                          <i className="bi bi-file-earmark me-1"></i> Select

                        </span>

                      </label>

                      <button

                        className="btn btn-sm btn-primary"

                        onClick={uploadDp}

                      >

                        <i className="bi bi-cloud-arrow-up-fill me-1"></i>{" "}

                        {/\* Upload DP \*/}

                      </button>

                      <button

                        className="btn btn-outline-danger ms-2"

                        onClick={Deletedp}

                      >

                        <i className="bi bi-trash-fill"></i>

                      </button>

                    </div>

                  </div>

                </div>

                <div className="ms-3 mb-2">

                  <h1 className="fw-bold team-name">{team}</h1>

                  <div className="mt-2">

                    <button

                      className="btn btn-primary me-2"

                      onClick={() => {

                        window.open("http://localhost:3000", "\_blank");

                      }}

                    >

                      <i className="bi bi-chat-dots-fill me-1"></i> Message

                    </button>

                    <button className="btn btn-success">

                      <Link to="/booking">

                        <i className="bi bi-people-fill me-1"></i> Match Up

                      </Link>

                    </button>

                  </div>

                </div>

              </div>

            </div>

          </div>

          <div className="row">

            {/\* Left Column - Status Section \*/}

            <div className="col-md-5 mb-4">

              <div className="card shadow-sm">

                <div className="card-header bg-white">

                  <h5 className="mb-0">Share an Update</h5>

                </div>

                <div className="card-body">

                  <textarea

                    className="form-control mb-3"

                    rows="3"

                    placeholder="What's on your mind?"

                    onChange={(event) => setStatus(event.target.value)}

                    value={status}

                  ></textarea>

                  <button

                    className="btn btn-primary float-end"

                    onClick={addStatus}

                    type="button"

                  >

                    <i className="bi bi-send-fill me-1"></i> Share

                  </button>

                </div>

              </div>

              {/\* Status Posts \*/}

              {statusArray &&

                statusArray.map((statusText, index) => (

                  <div key={index} className="card shadow-sm my-3 status-card">

                    <div className="card-header bg-white d-flex justify-content-between align-items-center">

                      <div className="d-flex align-items-center">

                        {dp\_Url && (

                          <img

                            src={dp\_Url}

                            className="status-profile-pic rounded-circle me-2"

                            alt=""

                          />

                        )}

                        <h4 className="mb-0">{team}</h4>

                      </div>

                      {/\* <button

                        className="btn btn-sm btn-outline-danger"

                        onClick={(e) => deleteStatus(e, index)}

                      >

                        <i className="bi bi-trash"></i>

                      </button> \*/}

                    </div>

                    <div className="card-body">

                      <h5 className="card-text">{statusText}</h5>

                    </div>

                    <div className="card-footer bg-white text-muted small">

                      {/\* <i className="bi bi-clock me-1"></i> Just now \*/}

                    </div>

                  </div>

                ))}

            </div>

            {/\* Right Column - Timeline Gallery \*/}

            <div className="col-md-7">

              <div className="card shadow-sm mb-4">

                <div className="card-header bg-white">

                  <h4 className="mb-0">Team Timeline</h4>

                </div>

                <div className="card-body">

                  <div className="mb-3">

                    <label htmlFor="formFile" className="form-label">

                      Upload a photo to your timeline

                    </label>

                    <div className="input-group">

                      <input

                        className="form-control"

                        type="file"

                        id="formFile"

                        onChange={(event) => {

                          setImageUpload(event.target.files[0]);

                        }}

                      />

                      <button className="btn btn-primary" onClick={uploadImage}>

                        <i className="bi bi-cloud-arrow-up-fill me-1"></i>{" "}

                        Upload

                      </button>

                    </div>

                  </div>

                </div>

              </div>

              {/\* Timeline Images \*/}

              <div className="timeline-gallery">

                <div className="row g-3">

                  {imageList &&

                    imageList.map((url, index) => (

                      <div key={index} className="col-md-6 col-lg-4">

                        <div className="card timeline-image-card h-100">

                          <a

                            href={url}

                            target="\_blank"

                            rel="noopener noreferrer"

                          >

                            <img

                              src={url}

                              className="card-img-top timeline-image"

                              alt="Timeline"

                            />

                          </a>

                          <div className="card-body p-2">

                            <button

                              className="btn btn-sm btn-outline-danger w-100"

                              onClick={(e) => deletePicture(e, index)}

                            >

                              <i className="bi bi-trash-fill me-1"></i>

                            </button>

                          </div>

                        </div>

                      </div>

                    ))}

                </div>

              </div>

            </div>

          </div>

        </div>

      </div>

    </>

  );

}

export default Profile;

**#selectlogin.jsx:**

import React from "react";

function Selectlogin ()

{

    return (

        <>

            <div id="selectlogin">

                <p id="selectlogintext">Login as Team</p>

                <p id="selectlogintext2">Login as Ground</p>

                <hr id="hrr"></hr>

            </div>

        </>

    );

}

export default Selectlogin;

**#Teamform.jsx:**

import React, { useState } from "react";

import axios from "axios";

import \* as CONTANT from "../Constant/constant.js";

import { useHistory } from "react-router-dom";

import Teamfrontpages from "./Teamfrontpages";

import { useDispatch } from "react-redux";

import { teamAction } from "../redux/actions/loadingAction";

import { setTeamData } from "../redux/slices/utilitySlice";

import MuiAlert from "@material-ui/lab/Alert";

import Snackbar from "@material-ui/core/Snackbar";

import { Typography } from "@material-ui/core";

function Teamform() {

  const dispatch = useDispatch();

  const history = useHistory();

  //Snackbar

  const [snackBarstate, setSnackBar] = useState({

    open: false,

    vertical: "top",

    horizontal: "right",

    type: "success",

    message: "",

  });

  const { vertical, horizontal, open } = snackBarstate;

  const handleCloseSnackBar = () => {

    setSnackBar({ ...snackBarstate, open: false });

  };

  //For Registraition List of Teams

  const [items, setItems] = useState([]);

  const [team, setTeam] = useState({

    teamName: "",

    captainName: "",

    email: "",

    password: "",

    phone: "",

    address: "",

  });

  const onSubmits = (event) => {

    event.preventDefault();

  };

  const inputEvent = (event) => {

    console.log(event);

    const name = event.target.name;

    const value = event.target.value;

    console.log(name, value);

    setTeam({

      ...team,

      [name]: value,

    });

  };

  const register = () => {

    const data = {

      name: team.teamName,

      captainName: team.captainName,

      email: team.email,

      password: team.password,

      phone: team.phone,

      address: team.address,

    };

    if (

      data.name &&

      data.captainName &&

      data.email &&

      data.password &&

      data.phone &&

      data.address

    ) {

      CONTANT.API.post("/team/register", data).then((res) => {

        if (res.data.message == "Successfully Registered, Please Login Now") {

          alert(res.data.message);

          setSnackBar({

            ...snackBarstate,

            type: "success",

            message: "Success: Team Added Successfully ",

            open: true,

          });

          history.push("/login");

        } else {

          console.log(res.data);

          alert(res.data.message);

        }

      });

    } else {

      alert("Not posted");

    }

  };

  const addItem = () => {

    // value.push(team);

    // console.log(value);

    // const temp = Object.freeze([]);

    // temp.push(team);

    let newArr = [...items];

    newArr.push(team);

    setItems(newArr);

    console.log(newArr);

    dispatch(setTeamData(newArr));

    //localStorage.setItem("teams", newArr);

    // history.push("/teams");

    // if(!team.teamname)

    // {

    // }

    // else

    // {

    //     setItems([...items, team.teamname]);

    // }

  };

  return (

    <>

      <div

        className="d-flex justify-content-center align-items-center"

        style={{

          backgroundImage: "url('./images/teamspic.png')",

          backgroundSize: "cover",

          backgroundPosition: "center",

          minHeight: "100vh",

          position: "absolute",

          width: "100%",

          marginTop: "70px",

          paddingTop: "40px",

          paddingBottom: "40px",

          zIndex: "-100",

        }}

      >

        <div

          className="bg-white bg-opacity-25 p-5 rounded shadow"

          style={{

            backgroundColor: "rgba(255, 255, 255, 0.7)", // Semi-transparent background for contrast

            padding: "30px",

            borderRadius: "8px",

            boxShadow: "0px 4px 12px rgba(0, 0, 0, 0.1)", // Soft shadow around the form

            width: "80%", // Width of the form, adjust as necessary

            maxWidth: "420px", // Max width for responsiveness

            marginLeft: "430px",

          }}

        >

          <form id="teamform" onSubmit={onSubmits}>

            <h3 className="text-secondary fw-bold mb-4 text-center">

              SIGN UP AS TEAM

            </h3>

            <input

              style={{ marginTop: "30px" }}

              type="text"

              name="teamName"

              autoComplete="off"

              placeholder="Enter Your Team's Name"

              onChange={(e) => setTeam({ ...team, teamName: e.target.value })}

              value={team.teamName}

              id=""

              className="form-control mb-3 p-3"

            />

            <input

              type="text"

              name="captainName"

              autoComplete="off"

              placeholder="Enter Captain's Name"

              onChange={(e) =>

                setTeam({ ...team, captainName: e.target.value })

              }

              value={team.captainName}

              id=""

              className="form-control mb-3 p-3"

            />

            <input

              type="email"

              name="email"

              autoComplete="off"

              placeholder="Enter Your Email"

              onChange={(e) => setTeam({ ...team, email: e.target.value })}

              value={team.email}

              id=""

              className="form-control mb-3 p-3"

            />

            <input

              type="password"

              name="password"

              autoComplete="off"

              placeholder="Enter Your Password"

              onChange={(e) => setTeam({ ...team, password: e.target.value })}

              value={team.password}

              id=""

              className="form-control mb-3 p-3"

            />

            <input

              type="number"

              name="phone"

              autoComplete="off"

              placeholder="Enter Your Mobile Number"

              onChange={(e) => setTeam({ ...team, phone: e.target.value })}

              value={team.phone}

              id=""

              className="form-control mb-3 p-3"

            />

            <input

              type="text"

              name="address"

              autoComplete="off"

              placeholder="Enter Your Address"

              onChange={(e) => setTeam({ ...team, address: e.target.value })}

              value={team.address}

              id=""

              className="form-control mb-4 p-3"

            />

            <div className="text-center">

              <button

                type="submit"

                id=""

                className="btn btn-primary w-100 p-3"

                onClick={register}

              >

                Sign Up

              </button>

            </div>

          </form>

        </div>

      </div>

      <div>

        <Snackbar

          open={open}

          anchorOrigin={{ vertical, horizontal }}

          autoHideDuration={5000}

          onClose={handleCloseSnackBar}

          key={vertical + horizontal}

        >

          <MuiAlert

            elevation={6}

            variant="filled"

            onClose={handleCloseSnackBar}

            severity={snackBarstate.type}

          >

            <Typography style={{ color: "#fff" }}>

              {snackBarstate.message}

            </Typography>

          </MuiAlert>

        </Snackbar>

      </div>

    </>

  );

}

// function Teamfrontpage ()

// {

//     return (

//         <>

//         </>

//     );

// }

export default Teamform;

**#Teamfrontpages.js:**

import React, { useState, useEffect } from "react";

import GroupsIcon from "@mui/icons-material/Groups";

import ArrowDropDownIcon from "@mui/icons-material/ArrowDropDown";

import Teamform from "./Teamform";

import { useSelector } from "react-redux";

import { useHistory } from "react-router-dom";

import { getStorage, ref, getDownloadURL, listAll } from "firebase/storage";

import { storage } from ".//firebase";

import \* as CONSTANT from "../Constant/constant";

function Teamfrontpages() {

  //getting teams from redux

  //const team = useSelector((state) => state?.utilitySlice?.teamData);

  const history = useHistory();

  const [team, setTeams] = useState([]);

  const [OnlineUser, setOnlineUser] = useState(null);

  const [from, setFrom] = useState(null);

  useEffect(() => {

    window.localStorage.removeItem("visitorID");

    window.localStorage.removeItem("visitordpUrl");

    window.localStorage.removeItem("visitorcoverUrl");

    window.localStorage.removeItem("visitiorName");

    setOnlineUser(window.localStorage.getItem("email"));

    setFrom(window.localStorage.getItem("from"));

    getAllTeams();

  });

  const getAllTeams = () => {

    CONSTANT.API.get("/team/test").then((res) => {

      setTeams(res.data);

    });

  };

  const ProfileView = (email, id, teamName) => {

    if (OnlineUser == email && from == "team") {

      history.push("/profile");

    } else {

      window.localStorage.setItem("visitorID", id);

      window.localStorage.setItem("visitiorName", teamName);

      history.push("/visitorprofile");

    }

  };

  return (

    <>

      <div

        className="d-flex justify-content-center align-items-center"

        style={{

          backgroundImage: "url('./images/teampicfrontmain.jpg')",

          backgroundSize: "cover",

          backgroundPosition: "center",

          minHeight: "100vh",

          width: "100%",

          position: "absolute",

          top: 0,

          left: 0,

          padding: "40px 0",

          overflow: "auto",

          marginTop: "70px",

        }}

      >

        <div

          className="bg-white bg-opacity-25 p-4 rounded shadow"

          style={{

            backgroundColor: "rgba(255, 255, 255, 0.7)",

            width: "90%",

            maxWidth: "550px",

            maxHeight: "80vh",

            overflowY: "auto",

            marginTop: "-400px",

          }}

        >

          <input

            type="text"

            placeholder="Search Team by Name"

            className="form-control mb-4 p-3"

            style={{

              padding: "40px",

              borderRadius: "8px",

              fontWeight: "bold",

            }}

            id="teamgroundsearch"

          />

          <ul className="list-group">

            {team &&

              team.map((item, index) => (

                <li

                  key={index}

                  className="list-group-item d-flex justify-content-between align-items-center mb-2"

                  style={{

                    cursor: "pointer",

                    // borderRadius: "10px",

                    // backgroundColor: "rgba(255, 255, 255, 0.9)",

                    boxShadow: "0 2px 6px rgba(0,0,0,0.1)",

                  }}

                  id="teamgroundpageul"

                  onClick={() => ProfileView(item.email, item.\_id, item.name)}

                >

                  <span>

                    <GroupsIcon style={{ marginRight: "10px" }} />

                    <strong>{item.name}</strong>

                  </span>

                  <ArrowDropDownIcon />

                </li>

              ))}

          </ul>

        </div>

      </div>

    </>

  );

}

export default Teamfrontpages;

**#visitorgroundprofile.jsx:**

import React from "react";

import { useState, useEffect } from "react";

import Button from "@material-ui/core/Button";

import { Grid } from "@material-ui/core";

import { useDispatch } from "react-redux";

import { useSelector } from "react-redux";

import { storage } from ".//firebase";

import { getStorage, deleteObject } from "firebase/storage";

import {

  ref,

  uploadBytes,

  getDownloadURL,

  listAll,

  list,

} from "firebase/storage";

import { v4 } from "uuid";

import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";

import {

  setGroundId,

  setAddress,

  setOwnerName,

  setEmail,

  setPhone,

  setGroundName,

} from "../redux/slices/groundSlice";

import { Link, useHistory } from "react-router-dom/cjs/react-router-dom";

function VisitorGroundprofile() {

  const history = useHistory();

  const { localStorage } = window;

  const [id, setID] = useState(localStorage.getItem("visitorID"));

  const [groundName, setGroundName] = useState(

    window.localStorage.getItem("visitiorName")

  );

  useEffect(() => {

    setDp();

    setCover();

  });

  //Dp

  const [dp\_Url, setDpUrl] = useState(localStorage.getItem("visitordpUrl"));

  useEffect(() => {

    setTimeout(() => {

      setDpUrl(localStorage.getItem("visitordpUrl"));

    }, 1000);

  }, [dp\_Url]);

  //Cover

  const [cover\_Url, setCoverUrl] = useState(

    localStorage.getItem("visitorcoverUrl")

  );

  useEffect(() => {

    setTimeout(() => {

      setCoverUrl(localStorage.getItem("visitorcoverUrl"));

    }, 1000);

  }, [cover\_Url]);

  //Timeline

  const [imageList, setImageList] = useState([]);

  const imageListRef = ref(storage, `groundtimeline/${id}`);

  useEffect(() => {

    listAll(imageListRef).then((response) => {

      response.items.forEach((item) => {

        getDownloadURL(item).then((url) => {

          let AllURL = imageList;

          AllURL.push(url);

          setImageList([...AllURL]);

        });

      });

    });

  }, []);

  const setDp = () => {

    const storage = getStorage();

    getDownloadURL(ref(storage, `grounddp/${id}`))

      .then((url) => {

        localStorage.setItem("visitordpUrl", url);

      })

      .catch((err) => console.log(err));

  };

  const setCover = (groundId) => {

    const storage = getStorage();

    getDownloadURL(ref(storage, `groundcover/${id}`))

      .then((url) => {

        console.log("Cover URL: ", url);

        localStorage.setItem("visitorcoverUrl", url);

      })

      .catch((err) => console.log(err));

  };

  return (

    <div className="ground-page">

      {/\* Hero Section with Cover Photo \*/}

      <div className="ground-hero-section">

        <div className="ground-cover-container">

          {cover\_Url ? (

            <a href={cover\_Url} target="\_blank" rel="noopener noreferrer">

              <img

                src={cover\_Url}

                className="ground-cover-image"

                alt="Ground Cover"

              />

            </a>

          ) : (

            <div className="ground-cover-placeholder">

              {/\* <span>Add a cover photo for your ground</span> \*/}

            </div>

          )}

          {/\* <div className="ground-cover-controls">

              <input

                type="file"

                id="select-image2"

                className="ground-file-input"

                onChange={(event) => {

                  setCoverUpload(event.target.files[0]);

                }}

              />

              <label htmlFor="select-image2" className="me-2">

                <span className="btn btn-outline-light ground-select-btn">

                  <i className="bi bi-file-earmark me-1"></i> Select Cover

                </span>

              </label>

              <button

                className="btn btn-success ground-upload-btn"

                onClick={uploadCover}

              >

                <i className="bi bi-cloud-arrow-up-fill me-1"></i> Upload Cover

              </button>

            </div> \*/}

        </div>

      </div>

      <div className="container ground-content-container">

        <div className="row">

          {/\* Left Column - Ground Profile Info \*/}

          <div className="col-md-4">

            <div className="ground-profile-card">

              <div className="ground-dp-section">

                <div className="ground-dp-container">

                  {dp\_Url ? (

                    <a href={dp\_Url} target="\_blank" rel="noopener noreferrer">

                      <img

                        src={dp\_Url}

                        className="ground-dp-image"

                        alt="Ground Logo"

                      />

                    </a>

                  ) : (

                    <div className="ground-dp-placeholder">

                      <i className="bi bi-image"></i>

                    </div>

                  )}

                </div>

                {/\* <div className="ground-dp-controls">

                    <input

                      type="file"

                      id="select-image"

                      className="ground-file-input"

                      onChange={(event) => {

                        setDpUpload(event.target.files[0]);

                      }}

                    />

                    <label htmlFor="select-image" className="me-2">

                      <span className="btn btn-outline-dark btn-sm ground-select-btn">

                        <i className="bi bi-file-earmark me-1"></i> Select Image

                      </span>

                    </label>

                    <button

                      className="btn btn-success btn-sm ground-upload-btn"

                      onClick={uploadDp}

                    >

                      <i className="bi bi-cloud-arrow-up-fill me-1"></i> Upload

                      Logo

                    </button>

                  </div> \*/}

              </div>

              <div className="ground-info-section">

                <h1 className="ground-name">{groundName}</h1>

                <p className="ground-description">

                  Best Ground in Twin Cities

                  <br />

                  Here to achieve your dreams

                </p>

                <div className="ground-action-buttons">

                  <button

                    className="btn btn-primary ground-msg-btn"

                    onClick={() => {

                      window.open("http://localhost:3001/", "\_blank");

                    }}

                  >

                    <i className="bi bi-chat-dots-fill me-2"></i>Message

                  </button>

                  <button className="btn btn-info ground-booking-btn">

                    <Link to="/booking">

                      <i className="bi bi-calendar-check me-2"></i>Bookings{" "}

                    </Link>

                  </button>

                  <button

                    className="btn btn-secondary ground-academy-btn"

                    onClick={() => history.push("/academyuser", { groundName })}

                  >

                    <i className="bi bi-trophy me-2"></i>Academy

                  </button>

                  {/\* <button className="btn btn-success ground-location-btn">

                    <i className="bi bi-geo-alt-fill me-2"></i>Location

                  </button> \*/}

                </div>

              </div>

            </div>

          </div>

          {/\* Right Column - Timeline Gallery \*/}

          <div className="col-md-8">

            <div className="ground-timeline-section">

              <div className="ground-timeline-header">

                <h2 className="ground-timeline-title">

                  <i className="bi bi-images me-2"></i>Ground Timeline

                </h2>

                {/\* <div className="ground-timeline-upload">

                    <input

                      type="file"

                      id="timeline-image-upload"

                      className="ground-file-input"

                      onChange={(event) => {

                        setImageUpload(event.target.files[0]);

                      }}

                    />

                    <label htmlFor="timeline-image-upload" className="me-2">

                      <span className="btn btn-outline-dark ground-select-btn">

                        <i className="bi bi-file-earmark-image me-1"></i> Select

                        Photo

                      </span>

                    </label>

                    <button

                      className="btn btn-success ground-upload-btn"

                      onClick={uploadImage}

                    >

                      <i className="bi bi-cloud-arrow-up-fill me-1"></i> Add to

                      Timeline

                    </button>

                  </div> \*/}

              </div>

              <div className="ground-timeline-gallery">

                <div className="row g-4">

                  {imageList.length > 0 &&

                    imageList.map((url, index) => (

                      <div key={index} className="col-lg-4 col-md-6">

                        <div className="ground-gallery-item">

                          <div className="ground-gallery-image-container">

                            <a

                              href={url}

                              target="\_blank"

                              rel="noopener noreferrer"

                            >

                              <img

                                src={url}

                                className="ground-gallery-image"

                                alt="Timeline"

                              />

                            </a>

                          </div>

                          {/\* <div className="ground-gallery-actions">

                              <button

                                className="btn btn-danger ground-delete-btn"

                                onClick={(e) => deletePicture(e, index)}

                              >

                                <i className="bi bi-trash me-1"></i> Remove

                              </button>

                            </div> \*/}

                        </div>

                      </div>

                    ))}

                  {imageList.length === 0 && (

                    <div className="ground-empty-gallery">

                      <div className="text-center p-5">

                        <i className="bi bi-camera-fill ground-empty-icon"></i>

                        <h4>No Images in Timeline</h4>

                        <p>

                          Upload your first image to start building your

                          ground's timeline.

                        </p>

                      </div>

                    </div>

                  )}

                </div>

              </div>

            </div>

          </div>

        </div>

      </div>

    </div>

  );

}

export default VisitorGroundprofile;

**#visitorprofile.jsx:**

import React from "react";

import AddIcon from "@mui/icons-material/Add";

import { useState, useEffect } from "react";

import Button from "@material-ui/core/Button";

import { Grid } from "@material-ui/core";

import \* as CONTANT from "../Constant/constant.js";

import { useHistory } from "react-router-dom";

import { fontSize } from "@mui/system";

import { useSelector } from "react-redux";

import { ToastContainer, toast } from "react-toastify";

import "react-toastify/dist/ReactToastify.css";

import { storage } from ".//firebase";

import {

  ref,

  uploadBytes,

  getDownloadURL,

  listAll,

  list,

} from "firebase/storage";

import { useDispatch } from "react-redux";

import { getStorage, deleteObject } from "firebase/storage";

import { Link } from "react-router-dom/cjs/react-router-dom.js";

function Profile() {

  const { localStorage } = window;

  const userEmail = useSelector((state) => state?.utilitySlice?.userEmail);

  const teamId = useSelector((state) => state?.teamSlice?.teamId);

  const [id, setID] = useState(localStorage.getItem("visitorID"));

  const [team, setTeam] = useState(window.localStorage.getItem("visitiorName"));

  const imageListRef = ref(storage, `teamtimeline/${id}`);

  const [imageList, setImageList] = useState([]);

  useEffect(() => {

    setCover();

    setDp();

  });

  useEffect(() => {

    setImageList([]);

    listAll(imageListRef).then((response) => {

      response.items.forEach((item) => {

        getDownloadURL(item).then((url) => {

          let AllURL = imageList;

          AllURL.push(url);

          setImageList([...AllURL]);

        });

      });

    });

  }, []);

  //Dp

  const [dp\_Url, setDpUrl] = useState(localStorage.getItem("visitordpUrl"));

  useEffect(() => {

    setTimeout(() => {

      setDpUrl(localStorage.getItem("visitordpUrl"));

      getAllStatus();

    }, 1000);

  }, [dp\_Url]);

  //Cover

  const [cover\_Url, setCoverUrl] = useState(

    localStorage.getItem("visitorcoverUrl")

  );

  useEffect(() => {

    setTimeout(() => {

      setCoverUrl(localStorage.getItem("visitorcoverUrl"));

      getAllStatus();

    }, 1000);

  }, [cover\_Url]);

  //Status

  const [statusArray, setStatusArray] = useState([]);

  const getAllStatus = async () => {

    const response = await CONTANT.API.get(`/team/getStatus/${id}`);

    if (response?.data) {

      setStatusArray(response.data.statusArray);

    }

  };

  const setDp = () => {

    const storage = getStorage();

    getDownloadURL(ref(storage, `teamdp/${id}`))

      .then((url) => {

        // setDpUrl(url);

        localStorage.setItem("visitordpUrl", url);

      })

      .catch((err) => console.log(err));

  };

  const setCover = () => {

    const storage = getStorage();

    getDownloadURL(ref(storage, `teamcover/${id}`))

      .then((url) => {

        // setCoverUrl(url);

        localStorage.setItem("visitorcoverUrl", url);

      })

      .catch((err) => console.log(err));

  };

  return (

    <>

      <div className="profile-page bg-light">

        <ToastContainer />

        {/\* Header with navigation \*/}

        {/\* <nav className="navbar navbar-expand-lg navbar-dark bg-primary sticky-top">

        <div className="container">

          <a className="navbar-brand fw-bold" href="#">

            TeamConnect

          </a>

          <button

            className="btn btn-outline-light ms-auto"

            onClick={() => logout()}

          >

            Logout

          </button>

        </div>

      </nav> \*/}

        <div className="container mt-3">

          {/\* Cover Photo Section \*/}

          <div className="position-relative mb-5">

            <div className="cover-photo-container rounded shadow">

              {cover\_Url ? (

                <a href={cover\_Url} target="\_blank" rel="noopener noreferrer">

                  <img

                    src={cover\_Url}

                    className="cover-photo w-100"

                    alt="Cover"

                  />

                </a>

              ) : (

                <div className="cover-placeholder"></div>

              )}

              {/\* <div

              className="cover-photo-controls position-absolute bottom-0 end-0 m-3"

              style={{ zIndex: "99" }}

            >

              <input

                type="file"

                id="select-image2"

                className="d-none"

                onChange={(event) => {

                  setCoverUpload(event.target.files[0]);

                }}

              />

              <label htmlFor="select-image2" className="me-2">

                <span className="btn btn-light btn-sm">

                  <i className="bi bi-file-earmark me-1"></i> Select

                </span>

              </label>

              <button

                className="btn btn-primary btn-sm me-2"

                onClick={uploadCover}

              >

                <i className="bi bi-cloud-arrow-up-fill me-1"></i>

              </button>

              <button

                className="btn btn-outline-danger"

                title="Remove Cover"

                onClick={Deletecover}

              >

                <i className="bi bi-trash-fill me-1"></i>

              </button>

            </div> \*/}

            </div>

            {/\* Profile Picture & Team Info \*/}

            <div className="profile-header">

              <div className="d-flex align-items-end">

                <div className="profile-picture-container position-relative">

                  {dp\_Url ? (

                    <a href={dp\_Url} target="\_blank" rel="noopener noreferrer">

                      <img

                        src={dp\_Url}

                        className="profile-picture rounded-circle border border-4 border-white"

                        alt="Profile"

                        style={{ cursor: "pointer" }}

                      />

                    </a>

                  ) : (

                    <div className="profile-picture-placeholder rounded-circle border border-4 border-white"></div>

                  )}

                  {/\* <div className="profile-picture-edit">

                  <input

                    type="file"

                    id="select-image"

                    className="d-none"

                    onChange={(event) => {

                      setDpUpload(event.target.files[0]);

                    }}

                  />

                  <div className="mt-2 d-flex">

                    <label htmlFor="select-image" className="me-2">

                      <span className="btn btn-sm btn-light">

                        <i className="bi bi-file-earmark me-1"></i> Select

                      </span>

                    </label>

                    <button

                      className="btn btn-sm btn-primary"

                      onClick={uploadDp}

                    >

                      <i className="bi bi-cloud-arrow-up-fill me-1"></i>{" "}

                    </button>

                    <button

                      className="btn btn-outline-danger ms-2"

                      onClick={Deletedp}

                    >

                      <i className="bi bi-trash-fill"></i>

                    </button>

                  </div>

                </div> \*/}

                </div>

                <div className="ms-3 mb-2">

                  <h1 className="fw-bold team-name">{team}</h1>

                  <div className="mt-2">

                    <button

                      className="btn btn-primary me-2"

                      onClick={() => {

                        window.open("http://localhost:3000", "\_blank");

                      }}

                    >

                      <i className="bi bi-chat-dots-fill me-1"></i> Message

                    </button>

                    <button className="btn btn-success">

                      <Link to="/booking">

                        <i className="bi bi-people-fill me-1"></i> Match Up

                      </Link>

                    </button>

                  </div>

                </div>

              </div>

            </div>

          </div>

          <div className="row">

            {/\* Left Column - Status Section \*/}

            <div className="col-md-5 mb-4">

              {/\* <div className="card shadow-sm">

              <div className="card-header bg-white">

                <h5 className="mb-0">Share an Update</h5>

              </div>

              <div className="card-body">

                <textarea

                  className="form-control mb-3"

                  rows="3"

                  placeholder="What's on your mind?"

                  onChange={(event) => setStatus(event.target.value)}

                  value={status}

                ></textarea>

                <button

                  className="btn btn-primary float-end"

                  onClick={addStatus}

                  type="button"

                >

                  <i className="bi bi-send-fill me-1"></i> Share

                </button>

              </div>

            </div> \*/}

              {/\* Status Posts \*/}

              {statusArray &&

                statusArray.map((statusText, index) => (

                  <div key={index} className="card shadow-sm my-3 status-card">

                    <div className="card-header bg-white d-flex justify-content-between align-items-center">

                      <div className="d-flex align-items-center">

                        {dp\_Url && (

                          <img

                            src={dp\_Url}

                            className="status-profile-pic rounded-circle me-2"

                            alt=""

                          />

                        )}

                        <h4 className="mb-0">{team}</h4>

                      </div>

                      {/\* <button

                      className="btn btn-sm btn-outline-danger"

                      onClick={(e) => deleteStatus(e, index)}

                    >

                      <i className="bi bi-trash"></i>

                    </button> \*/}

                    </div>

                    <div className="card-body">

                      <h5 className="card-text">{statusText}</h5>

                    </div>

                    <div className="card-footer bg-white text-muted small">

                      {/\* <i className="bi bi-clock me-1"></i> Just now \*/}

                    </div>

                  </div>

                ))}

            </div>

            {/\* Right Column - Timeline Gallery \*/}

            <div className="col-md-7">

              <div className="card shadow-sm mb-4">

                <div className="card-header bg-white">

                  <h4 className="mb-0">Team Timeline</h4>

                </div>

                {/\* <div className="card-body">

                <div className="mb-3">

                  <label htmlFor="formFile" className="form-label">

                    Upload a photo to your timeline

                  </label>

                  <div className="input-group">

                    <input

                      className="form-control"

                      type="file"

                      id="formFile"

                      onChange={(event) => {

                        setImageUpload(event.target.files[0]);

                      }}

                    />

                    <button className="btn btn-primary" onClick={uploadImage}>

                      <i className="bi bi-cloud-arrow-up-fill me-1"></i>{" "}

                      Upload

                    </button>

                  </div>

                </div>

              </div> \*/}

              </div>

              {/\* Timeline Images \*/}

              <div className="timeline-gallery">

                <div className="row g-3">

                  {imageList &&

                    imageList.map((url, index) => (

                      <div key={index} className="col-md-6 col-lg-4">

                        <div className="card timeline-image-card h-100">

                          <a

                            href={url}

                            target="\_blank"

                            rel="noopener noreferrer"

                          >

                            <img

                              src={url}

                              className="card-img-top timeline-image"

                              alt="Timeline"

                            />

                          </a>

                          {/\* <div className="card-body p-2">

                          <button

                            className="btn btn-sm btn-outline-danger w-100"

                            onClick={(e) => deletePicture(e, index)}

                          >

                            <i className="bi bi-trash-fill me-1"></i>

                          </button>

                        </div> \*/}

                        </div>

                      </div>

                    ))}

                </div>

              </div>

            </div>

          </div>

        </div>

      </div>

    </>

  );

}

export default Profile;

1. **Routes**

**#Routes.js:**

import React, { useEffect, useState } from "react";

import { Route, Switch, BrowserRouter as Router } from "react-router-dom";

import Menu from "../pages/Menu";

import Home from "../pages/Home";

import Groundsfrontpage from "../pages/Groundsfrontpage";

import Teamfrontpages from "../pages/Teamfrontpages";

import Academydisplaypage from "../pages/Academydisplaypage";

import Login from "../pages/Login";

import Login2 from "../pages/Login2";

import Mylogin from "../pages/Mylogin";

import Footer from "../pages/Footer";

import Teamform from "../pages/Teamform";

import Groundform from "../pages/Groundform";

import Groundprofile from "../pages/Groundprofile";

import Academyform from "../pages/Academyform";

import Academyfrontpage from "../pages/Academyfrontpage";

import Donation from "../pages/Donation";

import Donationform from "../pages/Donationform";

import Profile from "../pages/Profile";

import Booking from "../pages/Booking";

import Booking2 from "../pages/Booking2";

import VisitorProfile from "../pages/VisitorProfile.jsx";

import VisitorGroundProfile from "../pages/VisitorGroundProfile";

import AcademyUsers from "../pages/AcademyUsers";

import { useSelector } from "react-redux";

import { css } from "@emotion/react";

import ClipLoader from "react-spinners/ClipLoader";

function Routes() {

  const storedTeamId = localStorage.getItem("teamId");

  const reduxTeamId = useSelector((state) => state?.teamSlice?.teamId);

  const teamId = storedTeamId || reduxTeamId;

  console.log("teamId", teamId);

  const storedGroundId = localStorage.getItem("groundId");

  const reduxGroundId = useSelector((state) => state?.groundSlice?.groundId);

  const groundId = storedGroundId || reduxGroundId;

  console.log("groundId", groundId);

  const [teamlogin, setTeamLogin] = useState({});

  const [groundlogin, setGroundLogin] = useState({});

  let [loading, setLoading] = useState(true);

  let [color, setColor] = useState("#ffffff");

  const override = css`

    display: block;

    margin: 0 auto;

    border-color: red;

  `;

  return (

    <Router>

      <Menu />

      <Switch>

        <Route exact path="/">

          <Home />

        </Route>

        <Route path="/donation">

          <Donation />

        </Route>

        <Route path="/donationform">

          <Donationform />

        </Route>

        <Route path="/academyuser">

          <AcademyUsers />

        </Route>

        <Route path="/visitorprofile">

          <VisitorProfile />

        </Route>

        <Route path="/visitorgroundprofile">

          <VisitorGroundProfile />

        </Route>

        <Route path="/academies">

          <Academyfrontpage />

        </Route>

        <Route path="/academydisplaypage">

          <Academydisplaypage />

        </Route>

        <Route path="/grounds">

          <Groundsfrontpage />

        </Route>

        <Route eaxct path="/teams">

          <Teamfrontpages />

        </Route>

        <Route path="/booking">

          {teamId || groundId ? <Booking /> : <Login setTeamLogin={setTeamLogin} />}

        </Route>

        {/\* <Route eaxct path="/booking">

          <Booking />

        </Route> \*/}

        <Route path="/booking2">

          <Booking2 />

        </Route>

        <Route path="/teamform">

          <Teamform />

        </Route>

        <Route path="/groundform">

          <Groundform />

        </Route>

        <Route path="/academyform">

          <Academyform />

        </Route>

        <Route path="/mylogin">

          <Mylogin />

        </Route>

        <Route path="/groundprofile">

          {groundId ? (

            <Groundprofile setGroundLogin={setGroundLogin} />

          ) : (

            <Login2 setGroundLogin={setGroundLogin} />

          )}

        </Route>

        <Route path="/login2">

          <Login2 setGroundLogin={setGroundLogin} />

        </Route>

        <Route path="/profile">

          {teamId ? (

            <Profile setTeamLogin={setTeamLogin} />

          ) : (

            <Login setTeamLogin={setTeamLogin} />

          )}

        </Route>

        <Route path="/login">

          <Login setTeamLogin={setTeamLogin} />

        </Route>

      </Switch>

    </Router>

  );

}

export default Routes;

1. **Redux**
2. **Actions**

**#authAction.js:**

import { login } from "../../service/Auth";

import { LOGIN\_SUCCESS, LOGIN\_ERROR } from "../slices/authenticationSlice";

export const loginAction = (authData) => async (dispatch) => {

    const res = await login(authData);

    if (res?.data?.login === true) {

        localStorage.setItem("login", true);

        localStorage.setItem("token", res?.data?.token);

        localStorage.setItem("nav", JSON.stringify(res?.data?.menu));

        localStorage.setItem("userName", res?.data?.tblUser?.userName);

        localStorage.setItem("userEmail", res?.data?.tblUser?.email);

        localStorage.setItem("isActive", res?.data?.tblUser?.isActive);

        localStorage.setItem("userRole", res?.data?.tblUser?.roleId);

        localStorage.setItem("userId", res?.data?.tblUser?.userId);

        dispatch(LOGIN\_SUCCESS(res?.data));

        return res;

    } else {

        dispatch(LOGIN\_ERROR("Error"));

        return res;

    }

};

**#loadingaction.js:**

import { setTeamData } from "../slices/utilitySlice";

export const teamAction = (data) => async (dispatch) => {

    dispatch(setTeamData(data));

};

1. **Slices**

**#groundSlice.js:**

import { createSlice } from "@reduxjs/toolkit";

export const slice = createSlice({

    name: "groundSlice",

    initialState: {

        groundId: "",

        groundname: "",

        ownername: "",

        email: "",

        phone: "",

        address: ""

    },

    reducers: {

        setGroundId(state, action) {

            state.groundId = action.payload

        },

        setGroundName(state, action) {

            state.groundname = action.payload

        },

        setOwnerName(state, action) {

            state.ownername = action.payload

        },

        setEmail(state, action) {

            state.email = action.payload

        },

        setPhone(state, action) {

            state.phone = action.payload

        },

        setAddress(state, action) {

            state.address = action.payload

        },

    }

})

export const {

    setAddress, setOwnerName, setEmail, setPhone, setGroundId, setGroundName

} = slice.actions;

export default slice.reducer;

**#teamSlice.js:**

import { createSlice } from "@reduxjs/toolkit";

export const slice = createSlice({

  name: "teamSlice",

  initialState: {

    teamId: "",

    teamname: "",

    captainname: "",

    email: "",

    phone: "",

    address: "",

  },

  reducers: {

    setTeamId(state, action) {

      state.teamId = action.payload;

    },

    setTeamName(state, action) {

      state.teamname = action.payload;

    },

    setCaptainName(state, action) {

      state.captainname = action.payload;

    },

    setEmail(state, action) {

      state.email = action.payload;

    },

    setPhone(state, action) {

      state.phone = action.payload;

    },

    setAddress(state, action) {

      state.address = action.payload;

    },

  },

});

export const {

  setAddress,

  setCaptainName,

  setEmail,

  setPhone,

  setTeamId,

  setTeamName,

} = slice.actions;

export default slice.reducer;

**#utilitySlice.js:**

import { createSlice } from "@reduxjs/toolkit";

export const slice = createSlice({

  name: "utilitySlice",

  initialState: {

    teamData: [],

    userEmail: "",

    Grounds: "",

  },

  reducers: {

    setTeamData: (state, action) => ({

      ...state,

      teamData: action.payload,

    }),

    setEmail:(state, action) => ({

      ...state,

      userEmail: action.payload

    }),

    setGrounds:(state, action) => ({

      ...state,

      Grounds: action.payload

    })

  },

});

export const { setTeamData, setEmail, setGrounds } = slice.actions;

export default slice.reducer;

**Dist:**

**#utilitySlice.dev.js:**

"use strict";

Object.defineProperty(exports, "\_\_esModule", {

  value: true

});

exports["default"] = exports.setTeamData = exports.slice = void 0;

var \_toolkit = require("@reduxjs/toolkit");

function \_toConsumableArray(arr) { return \_arrayWithoutHoles(arr) || \_iterableToArray(arr) || \_nonIterableSpread(); }

function \_nonIterableSpread() { throw new TypeError("Invalid attempt to spread non-iterable instance"); }

function \_iterableToArray(iter) { if (Symbol.iterator in Object(iter) || Object.prototype.toString.call(iter) === "[object Arguments]") return Array.from(iter); }

function \_arrayWithoutHoles(arr) { if (Array.isArray(arr)) { for (var i = 0, arr2 = new Array(arr.length); i < arr.length; i++) { arr2[i] = arr[i]; } return arr2; } }

function ownKeys(object, enumerableOnly) { var keys = Object.keys(object); if (Object.getOwnPropertySymbols) { var symbols = Object.getOwnPropertySymbols(object); if (enumerableOnly) symbols = symbols.filter(function (sym) { return Object.getOwnPropertyDescriptor(object, sym).enumerable; }); keys.push.apply(keys, symbols); } return keys; }

function \_objectSpread(target) { for (var i = 1; i < arguments.length; i++) { var source = arguments[i] != null ? arguments[i] : {}; if (i % 2) { ownKeys(source, true).forEach(function (key) { \_defineProperty(target, key, source[key]); }); } else if (Object.getOwnPropertyDescriptors) { Object.defineProperties(target, Object.getOwnPropertyDescriptors(source)); } else { ownKeys(source).forEach(function (key) { Object.defineProperty(target, key, Object.getOwnPropertyDescriptor(source, key)); }); } } return target; }

function \_defineProperty(obj, key, value) { if (key in obj) { Object.defineProperty(obj, key, { value: value, enumerable: true, configurable: true, writable: true }); } else { obj[key] = value; } return obj; }

var slice = (0, \_toolkit.createSlice)({

  name: "utilitySlice",

  initialState: {

    teamData: []

  },

  reducers: {

    setTeamData: function setTeamData(state, action) {

      return \_objectSpread({}, state, {

        teamData: [].concat(\_toConsumableArray(state.teamData), [action.payload])

      });

    }

  }

});

exports.slice = slice;

var setTeamData = slice.actions.setTeamData;

exports.setTeamData = setTeamData;

var \_default = slice.reducer;

exports["default"] = \_default;

1. **#groundReducer.js:**

import {combineReducer} from "redux";

import groundSlice from "./slices/groundSlice";

export default groundReducers({

    groundSlice

})

1. **#rootReducer.js:**

import { combineReducers } from "redux";

import utilitySlice from './slices/utilitySlice'

import teamSlice from "./slices/teamSlice"

import groundSlice from "./slices/groundSlice";

export default combineReducers({

    utilitySlice,

    teamSlice,

    groundSlice

});

1. **#Store.js:**

import { configureStore } from "@reduxjs/toolkit";

import rootReducer from "./rootReducer";

export default configureStore({

    reducer: rootReducer,

});

1. **#teamReducer.js:**

import {combineReducer} from "redux";

import teamSlice from "./slices/teamSlice";

export default teamReducers({

    teamSlice

})

**#academycss:**

/\* Import Bootstrap icons \*/

@import url("https://cdn.jsdelivr.net/npm/bootstrap-icons@1.8.1/font/bootstrap-icons.css");

.student-table-section {

  margin-top: 70px;

  position: absolute;

  width: 100%;

  padding: 2rem 0;

  background-color: #f8f9fa;

  min-height: 100vh;

}

/\* Header Styling \*/

.student-table-header {

  background-color: #ffffff;

  padding: 1.5rem;

  border-radius: 10px;

  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.05);

  margin-bottom: 1.5rem;

  display: flex;

  justify-content: space-between;

  align-items: center;

  flex-wrap: wrap;

  gap: 1rem;

}

.student-table-title {

  font-size: 1.5rem;

  color: #212529;

  margin: 0;

  font-weight: 600;

  display: flex;

  align-items: center;

}

.student-search-container {

  width: 300px;

}

.student-search-input {

  border-left: none;

  padding-left: 0;

}

.student-search-input:focus {

  box-shadow: none;

  border-color: #ced4da;

}

/\* Table Styling \*/

.student-table-container {

  background-color: #ffffff;

  border-radius: 10px;

  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.05);

  overflow: hidden;

}

.student-data-table {

  margin-bottom: 0;

  border-collapse: separate;

  border-spacing: 0;

}

.student-data-table thead th {

  background-color: #f8f9fa;

  font-weight: 600;

  color: #495057;

  padding: 1rem 0.75rem;

  border-bottom: 2px solid #e9ecef;

}

.student-sortable-header {

  cursor: pointer;

  user-select: none;

}

.student-sortable-header:hover {

  background-color: #e9ecef;

}

.sort-icon {

  font-size: 0.7rem;

  opacity: 0.5;

}

.student-table-row {

  transition: background-color 0.2s;

}

.student-table-row:hover {

  background-color: #f8f9fa;

}

.student-table-row td {

  padding: 1rem 0.75rem;

  vertical-align: middle;

}

/\* Avatar Styling \*/

.student-avatar {

  width: 36px;

  height: 36px;

  border-radius: 50%;

  background-color: #007bff;

  color: white;

  display: flex;

  align-items: center;

  justify-content: center;

  font-weight: bold;

  font-size: 1rem;

}

/\* Name Cell Styling \*/

.student-name-cell {

  min-width: 200px;

}

/\* Email Styling \*/

.student-email {

  color: #007bff;

  text-decoration: none;

}

.student-email:hover {

  text-decoration: underline;

}

/\* Academy Badge \*/

.student-academy-badge {

  background-color: #e3f2fd;

  color: #0d6efd;

  border-radius: 20px;

  padding: 0.2rem 0.75rem;

  font-size: 0.85rem;

  display: inline-flex;

  align-items: center;

}

/\* Action Buttons \*/

.student-actions {

  display: flex;

  gap: 0.3rem;

}

/\* Empty State \*/

.student-empty-state {

  background-color: #ffffff;

  border-radius: 10px;

  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.05);

}

.student-empty-icon {

  font-size: 3rem;

  color: #adb5bd;

  margin-bottom: 1rem;

  display: block;

}

/\* Stats Bar \*/

.student-table-stats {

  background-color: #ffffff;

  border-radius: 10px;

  padding: 0.75rem 1rem;

  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.05);

}

/\* Pagination \*/

.student-table-pagination .page-link {

  color: #007bff;

  border-radius: 0.

}

**#app.jsx:**

import React, { useState } from "react";

import "../node\_modules/bootstrap/dist/css/bootstrap.min.css";

import "./index.css";

import { BrowserRouter as Router, Route, Switch } from "react-router-dom";

import Routes from "./routes/Routes";

import { useDispatch } from "react-redux";

import {

  setTeamId,

  setAddress,

  setCaptainName,

  setEmail,

  setPhone,

  setTeamName,

} from "./redux/slices/teamSlice";

import { useEffect } from "react";

import { css } from "@emotion/react";

import ClipLoader from "react-spinners/ClipLoader";

function App() {

  const dispatch = useDispatch();

  const { sessionStorage } = window;

  let [loading, setLoading] = useState(true);

  let [color, setColor] = useState("#ffffff");

  const override = css`

    display: block;

    margin: 0 auto;

    border-color: red;

  `;

  useEffect(() => {

    const teamId = sessionStorage.getItem("teamId");

    if (teamId) {

      dispatch(setTeamId(sessionStorage.getItem("teamId")));

      dispatch(setTeamName(sessionStorage.getItem("teamname")));

      dispatch(setCaptainName(sessionStorage.getItem("captainname")));

      dispatch(setEmail(sessionStorage.getItem("email")));

      dispatch(setPhone(sessionStorage.getItem("phone")));

      dispatch(setAddress(sessionStorage.getItem("address")));

    } else {

      sessionStorage.clear();

      dispatch(setTeamId(""));

      dispatch(setTeamName(""));

      dispatch(setCaptainName(""));

      dispatch(setEmail(""));

      dispatch(setPhone(""));

      dispatch(setAddress(""));

    }

  }, [window.location]);

  return (

    <div>

      <Routes />

    </div>

  );

}

export default App;

**#booking.css:**

/\* Booking Section Specific Styling \*/

.booking-section {

  background: linear-gradient(135deg, #0a3b0a 0%, #052505 50%, #031803 100%);

  color: #ffffff;

  padding: 3rem 2rem;

  box-shadow: 0 10px 30px rgba(0, 0, 0, 0.5);

  position: absolute;

  overflow: hidden;

  margin-top: 70px;

  width: 100%;

}

/\* Football field pattern overlay \*/

.booking-section::before {

  content: "";

  position: absolute;

  top: 0;

  left: 0;

  width: 100%;

  height: 100%;

  background-image: repeating-linear-gradient(

    0deg,

    transparent,

    transparent 30px,

    rgba(255, 255, 255, 0.05) 30px,

    rgba(255, 255, 255, 0.05) 60px

  );

  pointer-events: none;

  opacity: 0.5;

  z-index: 1;

}

/\* Football field center circle \*/

.booking-section::after {

  content: "";

  position: absolute;

  top: 50%;

  left: 50%;

  transform: translate(-50%, -50%);

  width: 200px;

  height: 200px;

  border: 2px solid rgba(255, 255, 255, 0.1);

  border-radius: 50%;

  pointer-events: none;

  z-index: 1;

}

/\* Ground name heading \*/

.groundname {

  font-size: 2.5rem;

  text-align: center;

  margin-bottom: 2rem;

  font-weight: 700;

  color: #ffffff;

  text-shadow: 0 4px 8px rgba(0, 0, 0, 0.6);

  position: relative;

  z-index: 2;

  letter-spacing: 1px;

}

/\* Date selection heading \*/

.dateheading {

  font-size: 1.6rem;

  text-align: center;

  margin: 2rem 0 1rem;

  font-weight: 600;

  color: #4ade80;

  text-shadow: 0 2px 4px rgba(0, 0, 0, 0.4);

  position: relative;

  z-index: 2;

}

/\* Date picker form container \*/

.formdate {

  /\* margin: 1.5rem 0 3rem; \*/

  /\* position: relative; \*/

  z-index: 2;

  display: flex;

  margin-right: 210px;

  justify-content: center;

}

/\* Date picker input \*/

.form-control {

  background-color: rgba(255, 255, 255, 0.15);

  border: 1px solid rgba(255, 255, 255, 0.3);

  color: black;

  padding: 0.75rem 1.25rem;

  border-radius: 50px;

  font-size: 1.2rem;

  min-width: 230px;

  text-align: center;

  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.3);

  transition: all 0.3s ease;

}

.form-control:focus {

  background-color: rgba(255, 255, 255, 0.25);

  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.4);

  border-color: #4ade80;

  outline: none;

}

/\* Filter buttons \*/

.allslots,

.avaslots {

  background-color: rgba(255, 255, 255, 0.15);

  color: white;

  border: none;

  padding: 0.6rem 1.2rem;

  border-radius: 50px;

  margin: 0 0.5rem 2rem;

  font-weight: 500;

  transition: all 0.3s ease;

  cursor: pointer;

  position: relative;

  z-index: 2;

}

.allslots:hover,

.avaslots:hover {

  background-color: rgba(255, 255, 255, 0.25);

  transform: translateY(-2px);

  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);

}

/\* Slots grid \*/

.slots-row {

  display: grid;

  grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));

  gap: 2rem;

  /\* position: relative; \*/

  z-index: 1;

  margin-right: 50px;

}

/\* Center slot card \*/

.center3 {

  display: flex;

  justify-content: center;

}

/\* Booking card styling \*/

.bookingcard {

  background: rgba(255, 255, 255, 0.1);

  border-radius: 16px;

  overflow: hidden;

  width: 100%;

  max-width: 280px;

  height: 250px;

  transition: transform 0.3s ease, box-shadow 0.3s ease;

  box-shadow: 0 4px 20px rgba(0, 0, 0, 0.3);

  backdrop-filter: blur(5px);

  border: 1px solid rgba(255, 255, 255, 0.1);

}

.bookingcard:hover {

  transform: translateY(-5px);

  box-shadow: 0 8px 24px rgba(0, 0, 0, 0.5);

}

/\* Booking image container \*/

.booking-image-container {

  width: 100%;

  display: flex;

  justify-content: center;

  padding: 10px 0;

  background: rgba(0, 0, 0, 0.2);

}

/\* Slot image \*/

.slot-image {

  width: 230px;

  height: 150px;

  object-fit: cover;

  border-radius: 8px;

  transition: transform 0.5s ease;

  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.3);

}

.bookingcard:hover .slot-image {

  transform: scale(1.05);

}

/\* Booking info container \*/

.container1 {

  padding: 1rem;

}

/\* Time heading \*/

.time-heading {

  font-size: 1.1rem;

  margin-bottom: 1rem;

  color: #ffffff;

  padding-left: 3px;

  display: flex;

  align-items: center;

}

/\* Add clock icon before time \*/

.time-heading::before {

  content: "⏰";

  margin-right: 8px;

  font-size: 1.2rem;

}

/\* Booked info container \*/

.booked-info {

  /\* background-color: rgba(74, 222, 128, 0.15);

  border-radius: 8px;

  padding: 0.2rem; \*/

  transition: all 0.3s ease;

}

/\* Booked by text \*/

.booked-by {

  padding-left: 3px;

  margin-bottom: 20px;

  color: rgba(255, 255, 255, 0.9);

  cursor: pointer;

  transition: all 0.3s ease;

}

.booked-by:hover {

  color: #ffffff;

}

/\* Emphasize booked team name \*/

.booked-by b {

  color: #4ade80;

  font-weight: 700;

}

/\* Book now button \*/

.btnn2 {

  width: 100%;

  padding: 0.75rem 0;

  background: linear-gradient(to right, #4ade80, #22c55e);

  color: #ffffff;

  border: none;

  border-radius: 50px;

  font-size: 1rem;

  font-weight: 600;

  cursor: pointer;

  transition: all 0.3s ease;

  box-shadow: 0 4px 12px rgba(34, 197, 94, 0.4);

  text-align: center;

  padding-left: 3px;

}

.btnn2:hover {

  background: linear-gradient(to right, #22c55e, #16a34a);

  box-shadow: 0 6px 16px rgba(34, 197, 94, 0.6);

  transform: translateY(-2px);

}

/\* React DatePicker customization \*/

.react-datepicker {

  background-color: #052505 !important;

  border: 1px solid rgba(255, 255, 255, 0.2) !important;

  border-radius: 12px !important;

  box-shadow: 0 10px 25px rgba(0, 0, 0, 0.5) !important;

}

.react-datepicker\_\_header {

  background-color: #031803 !important;

  border-bottom: 1px solid rgba(255, 255, 255, 0.1) !important;

  padding-top: 12px !important;

  border-top-left-radius: 12px !important;

  border-top-right-radius: 12px !important;

}

.react-datepicker\_\_current-month,

.react-datepicker\_\_day-name {

  color: #ffffff !important;

}

.react-datepicker\_\_day {

  color: rgba(255, 255, 255, 0.8) !important;

}

.react-datepicker\_\_day:hover {

  background-color: rgba(74, 222, 128, 0.3) !important;

  border-radius: 50% !important;

}

.react-datepicker\_\_day--selected,

.react-datepicker\_\_day--keyboard-selected {

  background-color: #4ade80 !important;

  border-radius: 50% !important;

  color: #031803 !important;

  font-weight: bold !important;

}

.react-datepicker\_\_day--disabled {

  color: rgba(255, 255, 255, 0.3) !important;

}

/\* Responsive adjustments \*/

@media (max-width: 768px) {

  .booking-section {

    padding: 2rem 1rem;

  }

  .groundname {

    font-size: 2rem;

  }

  .dateheading {

    font-size: 1.4rem;

  }

  .slots-row {

    grid-template-columns: repeat(auto-fill, minmax(220px, 1fr));

    gap: 1.5rem;

  }

}

@media (max-width: 480px) {

  .booking-section {

    padding: 1.5rem 0.75rem;

  }

  .groundname {

    font-size: 1.6rem;

  }

  .dateheading {

    font-size: 1.2rem;

  }

  .slots-row {

    grid-template-columns: 1fr;

    gap: 1rem;

  }

  .bookingcard {

    max-width: 100%;

  }

}

/\* Search Grounds Section - Dark Green Theme \*/

.search-grounds-section {

  background: linear-gradient(125deg, #0d4e0d 0%, #063906 40%, #042004 100%);

  color: #ffffff;

  padding: 3rem 2rem;

  box-shadow: 0 12px 30px rgba(0, 0, 0, 0.4);

  position: absolute;

  overflow: hidden;

  width: 100%;

  height: 94vh;

  margin-top: 70px;

}

/\* Field pattern overlay \*/

.search-grounds-section::before {

  content: "";

  position: absolute;

  top: 0;

  left: 0;

  width: 100%;

  height: 100%;

  background-image: radial-gradient(

      circle at 50% 50%,

      rgba(255, 255, 255, 0.06) 0%,

      transparent 80%

    ),

    linear-gradient(

      45deg,

      transparent 48%,

      rgba(255, 255, 255, 0.03) 50%,

      transparent 52%

    ),

    linear-gradient(

      -45deg,

      transparent 48%,

      rgba(255, 255, 255, 0.03) 50%,

      transparent 52%

    );

  background-size: 100% 100%, 30px 30px, 30px 30px;

  pointer-events: none;

  opacity: 0.8;

  z-index: 1;

}

.grounds-container {

  position: relative;

  z-index: 2;

  max-width: 1200px;

  margin: 0 auto;

}

/\* Search Box Styling \*/

.search-box-wrapper {

  margin-bottom: 3.5rem;

  text-align: center;

}

.search-heading {

  margin-top: 30px;

  font-size: 2.25rem;

  font-weight: 700;

  margin-bottom: 1.5rem;

  color: #ffffff;

  text-shadow: 0 3px 6px rgba(0, 0, 0, 0.3);

  letter-spacing: 1px;

}

.search-input-wrapper {

  text-align: center;

  justify-content: center;

  align-items: center;

  position: relative;

  max-width: 550px;

  margin: 0 auto;

  margin-top: 30px;

}

.ground-search-input {

  width: 100%;

  padding: 1rem 1.5rem;

  padding-right: 3rem;

  background-color: rgba(255, 255, 255, 0.12);

  border: 1px solid rgba(255, 255, 255, 0.2);

  border-radius: 50px;

  color: #ffffff;

  font-size: 1rem;

  backdrop-filter: blur(4px);

  transition: all 0.3s ease;

  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.2);

}

.ground-search-input::placeholder {

  color: rgba(255, 255, 255, 0.6);

}

.ground-search-input:focus {

  outline: none;

  background-color: rgba(255, 255, 255, 0.18);

  border-color: rgba(255, 255, 255, 0.4);

  box-shadow: 0 8px 20px rgba(0, 0, 0, 0.25);

}

.search-icon {

  position: absolute;

  right: 1.5rem;

  top: 50%;

  transform: translateY(-50%);

  color: rgba(255, 255, 255, 0.7);

  font-size: 1.2rem;

}

/\* Grounds Grid Layout \*/

.grounds-grid-container {

  margin-top: 60px;

  display: grid;

  grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));

  gap: 2rem;

}

/\* Ground Item Styling \*/

.ground-item-wrapper {

  display: flex;

  justify-content: center;

}

.ground-card-item {

  width: 100%;

  max-width: 300px;

  background: rgba(255, 255, 255, 0.08);

  border-radius: 14px;

  overflow: hidden;

  transition: all 0.3s ease;

  cursor: pointer;

  box-shadow: 0 8px 20px rgba(0, 0, 0, 0.25);

  border: 1px solid rgba(255, 255, 255, 0.1);

  backdrop-filter: blur(4px);

}

.ground-card-item:hover {

  transform: translateY(-8px);

  box-shadow: 0 14px 28px rgba(0, 0, 0, 0.3);

  border-color: rgba(255, 255, 255, 0.2);

}

/\* Ground Image \*/

.ground-image-wrapper {

  position: relative;

  overflow: hidden;

  height: 190px;

}

.ground-thumbnail {

  width: 100%;

  height: 100%;

  object-fit: cover;

  transition: transform 0.5s ease;

}

.ground-card-item:hover .ground-thumbnail {

  transform: scale(1.08);

}

/\* Ground Details \*/

.ground-details-container {

  padding: 1.5rem;

}

.ground-title {

  font-size: 1.35rem;

  font-weight: 600;

  margin-bottom: 0.75rem;

  color: #ffffff;

  letter-spacing: 0.3px;

}

.location-label {

  font-size: 1.1rem;

  font-weight: 500;

  color: #3eca5d;

  margin-bottom: 0.5rem;

  display: flex;

  align-items: center;

}

.location-label::before {

  content: "📍";

  margin-right: 8px;

  font-size: 1.1rem;

}

.address-text {

  color: rgba(255, 255, 255, 0.8);

  font-size: 0.95rem;

  margin-bottom: 1.5rem;

  line-height: 1.5;

}

/\* Book Now Button \*/

.book-ground-btn {

  width: 100%;

  padding: 0.8rem 0;

  background: linear-gradient(to right, #3eca5d, #28a745);

  color: #ffffff;

  border: none;

  border-radius: 50px;

  font-size: 1rem;

  font-weight: 600;

  cursor: pointer;

  transition: all 0.3s ease;

  box-shadow: 0 4px 12px rgba(40, 167, 69, 0.4);

}

.book-ground-btn:hover {

  background: linear-gradient(to right, #28a745, #218838);

  box-shadow: 0 6px 16px rgba(40, 167, 69, 0.5);

  transform: translateY(-2px);

}

/\* Responsive Adjustments \*/

@media (max-width: 992px) {

  .grounds-grid-container {

    grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));

    gap: 1.5rem;

  }

}

@media (max-width: 768px) {

  .search-grounds-section {

    padding: 2rem 1.5rem;

  }

  .search-heading {

    font-size: 1.8rem;

  }

  .grounds-grid-container {

    grid-template-columns: repeat(auto-fill, minmax(220px, 1fr));

    gap: 1.2rem;

  }

}

@media (max-width: 480px) {

  .search-grounds-section {

    padding: 1.5rem 1rem;

  }

  .search-heading {

    font-size: 1.5rem;

  }

  .grounds-grid-container {

    grid-template-columns: 1fr;

    gap: 1rem;

  }

  .ground-card-item {

    max-width: 100%;

  }

}

**#index.css:**

\* {

  box-sizing: border-box;

  margin: 0px;

  padding: 0 px;

}

body {

 background: linear-gradient(#141e30, #243b55);

}

li {

  list-style: none;

}

a {

  text-decoration: none;

}

html {

  font-size: 63%;

}

button {

  /\* padding: 9px 25px;

  background-color: #ffffff;

  color: #142458;

  border: none;

  border-radius: 50px;

  cursor: pointer;

  font-size: 16px;

  font-family: "poppins",

    sans-serif;

  text-transform: uppercase;

  transition: all 0.3s ease 0s; \*/

}

.main-nav {

  width: 100%;

  height: 7rem;

  display: grid;

  margin-left: 0px;

  align-items: center;

  background-color: #142458;

  grid-template-columns: 3rem 1fr 2fr 1fr 3rem;

  position: fixed;

}

.logo {

  grid-column: 2/3;

  justify-content: start;

  align-items: center;

}

.menu-link {

  grid-column: 3/4;

  margin-right: 0px;

}

.menu-link ul {

  height: 7rem;

  display: flex;

  justify-content: space-around;

  align-items: center;

}

.menu-link ul li {

  font-size: 15px;

}

.menu-link ul li a {

  text-transform: uppercase;

  font-family: "Poppins",

    sans-serif;

  color: #ffffff;

}

.login {

  height: 7rem;

  grid-column: 4/5;

  display: flex;

  justify-content: end;

  align-items: center;

}

.menu-link ul li a:hover {

  transition: all 0.3s ease 0s;

  color: #0088a9;

  transform: scale(1.3);

  font-weight: bold;

}

button:hover {

  /\* background-color: #31c2c7;

  color: #ffffff;

  font-weight: 700;

  transform: scale(1.1); \*/

}

.logo:hover {

  height: 40px;

  width: 120px;

  transition: 0.2s;

  cursor: pointer;

}

.bg {

  width: 100%;

}

.outer {

  padding: 0;

  /\* margin: -30px; \*/

  display: inline-block;

  align-items: center;

  position: absolute;

  /\* top: -150px; \*/

  /\* left: 50%; \*/

}

.matchup {

  margin-left: 15px;

}

.background-img {

  padding-top: 40px;

  background-size: cover;

  width: 100%;

  display: -moz-flex;

  display: -ms-flex;

  display: -o-flex;

  margin: 0;

  -webkit-background-size: cover;

  background-position: center center;

  background-attachment: fixed;

  background-repeat: no-repeat;

}

.hero\_section {

  animation: fadeIn 2s;

  -webkit-animation: fadeIn 1s;

  -moz-animation: fadeIn 2s;

  -o-animation: fadeIn 2s;

  -ms-animation: fadeIn 2s;

}

@keyframes fadeIn {

  0% {

    opacity: 0;

  }

  100% {

    opacity: 1;

  }

}

@-moz-keyframes fadeIn {

  0% {

    opacity: 0;

  }

  100% {

    opacity: 1;

  }

}

@-webkit-keyframes fadeIn {

  0% {

    opacity: 0;

  }

  100% {

    opacity: 1;

  }

}

@-o-keyframes fadeIn {

  0% {

    opacity: 0;

  }

  100% {

    opacity: 1;

  }

}

@-ms-keyframes fadeIn {

  0% {

    opacity: 0;

  }

  100% {

    opacity: 1;

  }

}

/\* Footer \*/

#footer {

  width: 100%;

  height: 50px;

  background-color: darkgrey;

}

/\*Teamform\*/

#teamformbg {

  margin-top: 30px;

  height: 100vh;

  width: 100%;

}

#teamform {

  /\* position: absolute;

  margin-top: -600px;

  margin-left: 840px; \*/

}

#teamformsignup {

  position: absolute;

  margin-left: 855px;

  letter-spacing: 1px;

  margin-top: -590px;

  font-family: "Lora",

    serif;

  font-size: 20px;

  color: rgb(107,

      105,

      107);

  text-shadow: 1px 1px grey;

}

#groundformsignup {

  position: absolute;

  margin-left: 808px;

  letter-spacing: 1px;

  margin-top: -590px;

  font-family: "Lora",

    serif;

  font-size: 18px;

  color: rgb(107,

      105,

      107);

  text-shadow: 1px 1px grey;

}

#teamforminput {

  font-family: "Lora",

    serif;

  background-color: rgb(155,

      62,

      241);

  color: white;

  width: 200px;

  height: 28px;

  border-radius: 20px 20px 20px 20px;

  margin: 7px;

  padding: 5px 20px;

  font-weight: bold;

  border-style: none;

  padding: 3px;

  border-color: rgb(146,

      44,

      241);

  box-shadow: 2px 2px 2px 1px rgb(196,

      157,

      231);

}

::-webkit-input-placeholder {

  font-family: "Lora",

    serif;

  color: white;

  font-weight: 500;

  font-size: 12px;

  opacity: 70%;

  margin: 10px;

  letter-spacing: 1px;

}

#teamformsubmit {

  color: white;

  width: 90px;

  height: 35px;

  margin-top: 15px;

  margin-left: 56px;

  border-radius: 30px 30px 30px 30px;

  background-color: green;

}

#teamformsubmit:hover {

  background-color: rgb(2,

      173,

      2);

  font-weight: bold;

}

#formsubmit {

  letter-spacing: 0.5px;

  margin-left: -8px;

  color: whitesmoke;

  margin-top: -1px;

  font-size: 14px;

  text-align: center;

  font-family: "Lora",

    serif;

}

#createaccount {

  margin-left: 763px;

  font-size: 12px;

  font-weight: bold;

  font-family: "Poppins",

    sans-serif;

}

#genderid {

  text-align: center;

  display: inline;

  font-family: "Lora",

    serif;

  background-color: rgb(155,

      62,

      241);

  border-radius: 20px 20px 20px 20px;

  font-weight: bold;

  border-style: none;

  padding: 8px;

  border-color: rgb(146,

      44,

      241);

  box-shadow: 2px 2px 2px 1px rgb(196,

      157,

      231);

  color: rgba(255,

      255,

      255,

      0.685);

  font-weight: bold;

  font-size: 11px;

  letter-spacing: 1px;

}

#genderp {

  display: inline;

  font-family: "Lora",

    serif;

  background-color: rgb(155,

      62,

      241);

  border-radius: 20px 20px 20px 20px;

  font-weight: bold;

  border-style: none;

  padding: 8px;

  border-color: rgb(146,

      44,

      241);

  box-shadow: 2px 2px 2px 1px rgb(196,

      157,

      231);

  color: rgba(255,

      255,

      255,

      0.685);

  font-weight: bold;

  font-size: 12px;

  letter-spacing: 1px;

}

#academyform {

  margin-top: -530px;

  margin-left: 820px;

}

#academyname {

  display: inline;

  font-family: "Lora",

    serif;

  background-color: rgb(155,

      62,

      241);

  border-radius: 20px 20px 20px 20px;

  font-weight: bold;

  border-style: none;

  padding: 8px;

  border-color: rgb(146,

      44,

      241);

  box-shadow: 2px 2px 2px 1px rgb(196,

      157,

      231);

  color: rgba(255,

      255,

      255,

      0.685);

  font-weight: bold;

  font-size: 12px;

  letter-spacing: 1px;

}

#academyforminput {

  font-family: "Lora",

    serif;

  background-color: rgb(155,

      62,

      241);

  color: rgba(255,

      255,

      255,

      0.685);

  width: 200px;

  height: 24px;

  border-radius: 20px 20px 20px 20px;

  margin: 4px;

  font-weight: bold;

  border-style: none;

  padding: 3px;

  border-color: rgb(146,

      44,

      241);

  box-shadow: 2px 2px 2px 1px rgb(196,

      157,

      231);

}

#dateaddressinput {

  font-family: "Lora",

    serif;

  background-color: rgb(155,

      62,

      241);

  color: rgba(255,

      255,

      255,

      0.685);

  width: 200px;

  height: 24px;

  border-radius: 20px 20px 20px 20px;

  margin: 4px;

  font-weight: bold;

  border-style: none;

  padding: 3px;

  border-color: rgb(146,

      44,

      241);

  box-shadow: 2px 2px 2px 1px rgb(196,

      157,

      231);

  margin-top: 10px;

  position: absolute;

}

#academyselect {

  display: inline;

  font-family: "Lora",

    serif;

  background-color: rgb(155,

      62,

      241);

  border-radius: 20px 20px 20px 20px;

  font-weight: bold;

  border-style: none;

  padding: 8px;

  border-color: rgb(146,

      44,

      241);

  box-shadow: 2px 2px 2px 1px rgb(196,

      157,

      231);

  color: rgba(255,

      255,

      255,

      0.685);

  font-weight: bold;

  font-size: 12px;

  letter-spacing: 1px;

  margin-left: 8px;

}

#academyformsignup {

  position: absolute;

  margin-left: 878px;

  margin-top: -50px;

  font-family: "Sarala",

    sans-serif;

  font-size: 24px;

  color: rgb(131,

      16,

      238);

}

#academysubmit {

  color: white;

  width: 70px;

  margin-left: 72px;

  height: 25px;

  border-radius: 10px 10px 10px 10px;

  background-color: green;

  margin-top: 15px;

}

/\*Login\*/

#forgotpass {

  font-family: "Poppins",

    sans-serif;

  font-weight: bolder;

  font-size: 12px;

  margin-top: -8px;

  color: rgb(104,

      101,

      101);

}

#loginid {

  position: absolute;

  margin-left: 910px;

  letter-spacing: 1px;

  margin-top: -590px;

  font-family: "Poppins",

    sans-serif;

  font-size: 20px;

  font-weight: 600;

  color: rgb(107,

      105,

      107);

  text-shadow: 1px 1px grey;

}

#loginbutton {

  margin-top: 10px;

  width: 90px;

  height: 40px;

  font-size: 12px;

  font-family: "Poppins",

    sans-serif;

}

#createaccount {

  position: absolute;

  margin-left: 846px;

  margin-top: -354px;

  color: rgb(61,

      61,

      61);

}

/\*Grounds\*/

#groundsimg1 {

  position: relative;

  width: 100%;

  height: 400px;

}

#groundsimg2 {

  position: relative;

  width: 100%;

  height: 400px;

}

/\* TeamGroundAcademyfrontpage \*/

#teamgroundfrontpagemainpic,

#academyfrontpagemainpic {

  background-repeat: no-repeat;

  height: 100vh;

  width: 100%;

  background-position: center center;

  opacity: 100%;

  position: absolute;

  margin-top: 70px;

}

#teamgroundfrontpagebox,

#academyfrontpagebox {

  width: 36%;

  height: 600px;

  background-color: #ece8e8a4;

  align-items: center;

  margin-left: 34%;

  margin-top: 140px;

  position: absolute;

  overflow: auto;

  border-radius: 5%;

  box-shadow: 1px 3px 8px 3px grey;

}

#teamgroundpageli,

#academypageli {

  margin-bottom: 20px;

  list-style: none;

  color: white;

  background-size: 3px 10px;

  position: absolute;

  margin-top: 20px;

  justify-content: center;

  align-items: center;

}

#teamgroundpageul,

#academypageul {

  /\* box-shadow: 1px 5px 14px 5px rgb(189,

      188,

      188); \*/

  /\* position: relative; \*/

  background-color: rgba(32,

      31,

      31,

      0.897);

  border-radius: 7px 7px 7px 7px;

  color: white;

  padding: 12px;

  font-size: 16px;

  /\* width: 360px; \*/

  /\* margin-top: 20px;

  height: 50px;

  font-family: "Poppins",

    sans-serif;

  letter-spacing: 5px;

  margin-left: 27%;

  font-size: 16px;

  text-align: center;

  line-height: 50px; \*/

  /\* justify-content: center;

  align-items: center;

  text-align: center; \*/

}

#teamgroundpageulname {

  position: relative;

  margin-right: 30px;

  color: white;

  cursor: pointer;

}

#academypageulname {

  position: relative;

  margin-right: 30px;

  color: white;

  cursor: pointer;

  font-size: 14px;

  letter-spacing: 3px;

}

#teamgroundfrontpagebox::-webkit-scrollbar,

#academyfrontpagebox::-webkit-scrollbar {

  width: 15px;

}

#teamgroundfrontpagebox::-webkit-scrollbar-track,

#academyfrontpagebox::-webkit-scrollbar-track {

  border-radius: 10px;

  background: #f5f5f5;

  box-shadow: inset 0 0 6px rgba(0,

      0,

      0,

      0.5);

}

#teamgroundfrontpagebox::-webkit-scrollbar-thumb,

#academyfrontpagebox::-webkit-scrollbar-thumb {

  width: 20px;

  background: rgb(153,

      152,

      152);

  box-shadow: inset 0 0 6px rgba(0,

      0,

      0,

      0.5);

}

#teamgroundpageulname:hover,

#academypageulname:hover {

  color: green;

  font-weight: bold;

}

#dropicon:hover {

  color: green;

  font-weight: bold;

  transform: scale(1.8);

  cursor: pointer;

}

#groupsportspic {

  position: absolute;

  margin-left: -10px;

  margin-top: -8px;

  line-height: 60px;

  transform: scale(1.5);

  color: gray;

}

#dropicon {

  position: absolute;

  margin-left: 290px;

  margin-top: -60px;

  line-height: 60px;

  transform: scale(1.5);

  color: gray;

}

#teamgroundsearch,

#academysearch {

  width: 300px;

  height: 30px;

  line-height: 30px;

  margin-left: 130px;

  margin-top: 30px;

  border-radius: 20px 20px 20px 20px;

  color: black;

  font-family: "Poppins",

    sans-serif;

  border: none;

  box-shadow: 2px 2px 2px 1px grey;

}

input[id^="teamgroundsearch"]::-webkit-input-placeholder,

input[id^="academysearch"]::-webkit-input-placeholder {

  color: black;

  font-weight: 500;

  margin-left: 50px;

  text-transform: uppercase;

}

#practicee {

  background-color: white;

  border: 2px solid black;

  width: 250px;

  height: 250px;

  margin: 50px 20px;

  overflow: auto;

}

#practicee::-webkit-scrollbar {

  width: 20px;

}

#practicee::-webkit-scrollbar-track {

  border-radius: 10px;

  background: red;

}

#practicee::-webkit-scrollbar-thumb {

  width: 20px;

  background: green;

}

/\* Profile \*/

.profilecontainer {

  padding: 20px 15%;

}

.coverimg {

  width: 143%;

  height: 400px;

  align-items: center;

  margin-left: -230px;

  margin-top: 50px;

  border-radius: 6px;

  margin-bottom: 15px;

  border: 5px solid #142458;

}

.coverinputt

{

  display:inline-flex;

  color:"primary";

  margin-top: 500px;

  margin-left: -200px;

  position: absolute;

}

.coverimginput {

  margin-top: 00px;

}

.profileimg {

  display: flex;

  align-items: center;

  margin-left: 50px;

  justify-content: center;

}

.dpinputt

{

  display:inline-flex;

  color:"primary";

  margin-top: 560px;

  margin-left: -360px;

  position: absolute;

}

.profilephoto {

  width: 160px;

  height: 160px;

  margin: 10px;

  border-radius: 100px;

  border: 5px solid #142458;

  position: relative;

  top: -110px;

  display: inline-block;

}

.profilephotoinput {

  width: 160px;

  height: 160px;

  margin: 10px;

  border-radius: 100px;

  border: 5px solid #142458;

  position: absolute;

  margin-top: 230px;

  margin-right: -180px;

  display: inline-block;

}

.profilebuttons {

  display: flex;

  align-items: right;

  justify-content: right;

  padding-right: 5px;

  position: relative;

  top: -190px;

}

.matchup2 {

  background-color: rgb(37,

      30,

      131);

  color: #ffffff;

  margin-top: -40px;

  height: 45px;

}

.message {

  background-color: rgb(37,

      30,

      131);

  color: #ffffff;

  margin-top: -40px;

  height: 45px;

}

.logout {

  background-color: rgb(37,

      30,

      131);

  color: #ffffff;

  margin-top: 70px;

  margin-left: 1130px;

  position: absolute;

}

.logout:hover {

  background-color: rgb(218,

      82,

      82);

  color: #ffffff;

  margin-top: 70px;

  margin-left: 1130px;

  position: absolute;

  transform: scale(1.1);

}

.details {

  display: flex;

  align-items: center;

  justify-content: center;

}

.Name {

  font-family: "Poppins", sans-serif;

  font-size: 30px;

  position: relative;

  margin-top: 60px;

}

.about {

  background-color: rgb(255,

      230,

      230);

  border-radius: 10px;

  height: auto;

  width: auto;

  text-align: center;

  position: relative;

  top: -90px;

  font-family: "poppins",

    sans-serif;

  padding: 5px;

  font-size: 17px;

  text-align: center;

}

.teamgalleryvisitor {

  background-color: rgb(255,

      230,

      230);

  border-radius: 10px;

  height: auto;

  width: auto;

  text-align: center;

  margin-top: 20%;

  font-family: "poppins",

    sans-serif;

  padding: 5px;

  font-size: 17px;

}

.teamgallery {

  background-color: rgb(255,

      230,

      230);

  border-radius: 10px;

  height: auto;

  width: auto;

  text-align: center;

  margin-top: 46px;

  font-family: "poppins",

    sans-serif;

  padding: 5px;

  font-size: 17px;

}

.timelinepics

{

  border-radius: 40px;

  margin: 40px;

  width:600px;

  height: 600px;

}

.teamgallerypics {

  width: 150px;

  height: 150px;

  margin: 10px;

  padding: 5px;

  align-items: left;

  justify-content: left;

}

.abouttext {

  width: 1000px;

  margin-top: 10px;

  height: auto;

  text-justify: auto;

  text-align: justify;

  text-align: left;

  font-size: 16px;

  background-color: transparent;

  background: transparent;

  border: none;

  padding: 0px;

  box-shadow: 5px 2px 5px 1px gray;

  height: 150px;

  opacity: 60% rgb(83,

      52,

      52);

}

.innerdetails {

  align-items: center;

  color: rgb(49,

      46,

      46);

  font-size: 12px;

  justify-content: center;

  font-weight: 50;

}

.profilepage {

  background: rgb(238,

      174,

      202);

  background: radial-gradient(circle,

      rgba(238,

        174,

        202,

        1) 0%,

      rgba(148,

        187,

        233,

        1) 100%);

  height: auto;

}

/\*MyLogin\*/

.buttonloginasteam {

  position: absolute;

  background: indigo;

  border-radius: 10px;

  padding-left: 60px;

  padding-right: 60px;

  margin-top: 100px;

  color: white;

  margin-left: -110px;

}

.myloginbutton {padding: 100px;

  background-color: #ece8e865;

  width: 400px;

  height: 340px;

  display: inline-block;

  align-content: center;

  border-radius: 35px;

  box-shadow: 1px 3px 10px 3px grey;

}

.buttonloginasground {

  margin-left: -131px;

  position: absolute;

  background: indigo;

  border-radius: 10px;

  padding-left: 60px;

  padding-right: 60px;

  margin-top: 165px;

  color: white;

  width: 270px;

}

#myloginbox {

  background-color: #ece8e865;

  width: 500px;

  height: 340px;

  position: absolute;

  margin-top: -538px;

  margin-left: 34%;

  border-radius: 35px;

  box-shadow: 1px 3px 10px 3px grey;

}

#myloginheading {

  position: absolute;

  text-align: center;

  margin-top: 100px;

  margin-left: 150px;

  font-family: "Lora",

    serif;

  font-size: 26px;

  font-weight: bold;

  letter-spacing: 2px;

  color: white;

  cursor: pointer;

}

#myloginheading2 {

  position: absolute;

  text-align: center;

  margin-left: 135px;

  margin-top: 170px;

  font-family: "Lora",

    serif;

  font-size: 26px;

  letter-spacing: 2px;

  font-weight: bold;

  color: white;

  cursor: pointer;

}

#myloginheading:hover,

#myloginheading2:hover {

  color: rgb(4,

      87,

      4);

  transform: scale(1.2);

  font-weight: 900;

  font-weight: bolder;

}

#selectlogin {

  position: absolute;

  width: 172px;

  height: 100px;

  background-color: whitesmoke;

  margin-left: 1354px;

  margin-top: 60px;

  border-radius: 20px;

}

#selectlogintext {

  position: absolute;

  text-align: center;

  margin-top: 26px;

  margin-left: 28px;

  font-family: "Lora",

    serif;

  font-size: 12px;

  text-transform: uppercase;

  letter-spacing: 2px;

  color: black;

  cursor: pointer;

}

#selectlogintext2 {

  position: absolute;

  text-transform: uppercase;

  text-align: center;

  margin-top: 51px;

  margin-left: 18px;

  font-family: "Lora",

    serif;

  font-size: 12px;

  letter-spacing: 2px;

  color: black;

  cursor: pointer;

}

#selectlogintext:hover,

#selectlogintext2:hover {

  cursor: pointer;

  transform: scale(1.1);

  color: #142458;

  font-weight: bold;

}

#hrr {

  margin-top: 80px;

  width: 100px;

  font-weight: bold;

  text-align: center;

  margin-left: 40px;

  color: gray;

}

/\* Donation Page \*/

.bggimg {

  height: 100vh;

  width: 100vw;

  margin-top: 72px;

  background-position: center;

  background-repeat: no-repeat;

  background-size: cover;

  filter: brightness(30%);

}

.centerr {

  position: absolute;

  top: 37%;

  left: 52%;

  transform: translate(-50%,

      -50%);

  color: azure;

  font-family: Impact,

    Haettenschweiler,

    "Arial Narrow Bold",

    sans-serif;

  font-style: bold;

  font-size: 50px;

}

.centerr1 {

  position: absolute;

  top: 49%;

  left: 52%;

  transform: translate(-50%,

      -50%);

  color: azure;

  font-family: Poppins;

  text-align: center;

  font-family: "Crimson Pro",

    serif;

  font-size: 20px;

}

#buttonn1 {

  position: absolute;

  top: 64%;

  left: 45%;

  border-radius: 0px;

  transform: translate(-50%,

      -50%);

  background-color: firebrick;

  color: white;

  border: 2px solid firebrick;

  width: 12%;

  height: 6%;

  font-family: "Crimson Pro",

    serif;

}

#buttonn1:hover {

  background-color: white;

  color: firebrick;

}

#buttonn2 {

  border-radius: 0px;

  position: absolute;

  top: 64%;

  left: 60%;

  transform: translate(-50%,

      -50%);

  background-color: transparent;

  color: white;

  border: 2px solid firebrick;

  width: 12%;

  height: 6%;

  font-family: "Crimson Pro",

    serif;

}

#buttonn2:hover {

  background-color: firebrick;

  color: white;

}

/\* Academy Front Page \*/

.colorbg{

  position: static;

  background-color: #142458;

  width: 1536px;

  height: 682px;

  margin-top: 70px;

  margin-left: 0px;

  position: absolute;

}

.img{

  max-width:100%;

  max-height: auto;

  text-align: right;

  padding-left: 500px;

  padding-bottom:0;

  padding-top: 0;

}

.headertext

{

  position: absolute;

  top: 150px;

  left: 50px;

  color: aliceblue;

  font-family: 'poppins',san-serif;

  font-size: 8vh;

  text-transform: uppercase;

}

.paragraphtext

{

  position: absolute;

  top: 250px;

  font-weight: 300;

  left: 50px;

  color:  #F0FFFF;

  font-family: 'poppins',san-serif;

  text-align: justify;

  font-size: 3vh;

  line-height: 1.6;

}

.solid

{

  position: absolute;

  top: 270px;

  font-weight: 300;

  left: 50px;

}

.register

{

  position: absolute;

  top: 450px;

  font-weight: 400;

  left: 50px;

  background-color: #848b25;

  font-family: 'poppins',san-serif;

  text-align: justify;

  font-size: 3vh;

}

/\* Academyform CSS \*/

.academyformbody {

  margin:0;

  padding:0;

  font-family: sans-serif;

  background: linear-gradient(#141e30, #243b55);

}

.login-box {

font-family: sans-serif;

position: absolute;

top: 72%;

left: 51%;

height: 730px;

width: 500px;

padding: 40px;

transform: translate(-50%, -50%);

background: rgba(0,0,0,.5);

box-sizing: border-box;

box-shadow: 0 15px 25px rgba(0,0,0,.6);

border-radius: 10px;

}

.login-box h1 {

margin: 0 0 30px;

padding: 0;

color: #04c2cb;

margin-bottom: 50px;

text-align: center;

}

.login-box .user-box {

position: relative;

}

.login-box .user-box input {

width: 100%;

padding: 10px 0;

font-size: 16px;

color: #fff;

margin-bottom: 30px;

border: none;

border-bottom: 1px solid #fff;

outline: none;

background: transparent;

}

.login-box .user-box select

{

width: 300px;

height: 40px;

margin-left: 50px;

margin-top: 28px;

color: black;

background-color: rgb(194, 196, 197);

}

.login-box .user-box label {

position: absolute;

top:0;

left: 0;

padding: 10px 0;

font-size: 16px;

color: #fff;

pointer-events: none;

transition: .5s;

}

.login-box .user-box input:focus ~ label,

.login-box .user-box input:valid ~ label {

top: -20px;

left: 0;

color: #03e9f4;

font-size: 12px;

}

.login-box .user-box select:focus ~ label,

.login-box .user-box select:valid ~ label {

top: -20px;

left: 0;

color: #03e9f4;

font-size: 12px;

}

.login-box form a {

position: relative;

display: inline-block;

padding: 0px 0px;

color: #03e9f4;

font-size: 16px;

text-decoration: none;

text-transform: uppercase;

overflow: hidden;

transition: .5s;

margin-top: 15px;

letter-spacing: 4px

}

.login-box a:hover {

background: #03e9f4;

color: #fff;

border-radius: 5px;

box-shadow: 0 0 5px #03e9f4,

            0 0 25px #03e9f4,

            0 0 50px #03e9f4,

            0 0 100px #03e9f4;

}

.login-box a span {

position: absolute;

display: block;

}

.login-box a span:nth-child(1) {

top: 0;

left: -100%;

width: 2px;

height: 2px;

background: linear-gradient(90deg, transparent, #03e9f4);

animation: btn-anim1 1s linear infinite;

}

@keyframes btn-anim1 {

0% {

  left: -100%;

}

50%,100% {

  left: 100%;

}

}

.login-box a span:nth-child(2) {

top: -100%;

right: 0;

width: 2px;

height: 2px;

background: linear-gradient(180deg, transparent, #03e9f4);

animation: btn-anim2 1s linear infinite;

animation-delay: .25s

}

@keyframes btn-anim2 {

0% {

  top: -100%;

}

50%,100% {

  top: 100%;

}

}

.login-box a span:nth-child(3) {

bottom: 0;

right: -100%;

width: 2px;

height: 2px;

background: linear-gradient(270deg, transparent, #03e9f4);

animation: btn-anim3 1s linear infinite;

animation-delay: .5s

}

@keyframes btn-anim3 {

0% {

  right: -100%;

}

50%,100% {

  right: 100%;

}

}

.login-box a span:nth-child(4) {

bottom: -100%;

left: 0;

width: 2px;

height: 2px;

background: linear-gradient(360deg, transparent, #03e9f4);

animation: btn-anim4 1s linear infinite;

animation-delay: .75s

}

@keyframes btn-anim4 {

0% {

  bottom: -100%;

}

50%,100% {

  bottom: 100%;

}

}

#sub

{

background-color: rgba(0, 0, 0, 0);

color: white;

height: 10px;

width: -30px;

margin-left: 0px;

}

/\*Profile Page\*/

#coverbutton

{

    margin-top: 500px;

    position: absolute;

    margin-left: -200px;

    width:95px;

}

.profilecontainer{

    padding: 20px  15%;

}

.coverimg {

    position: absolute;

    width:1738px;

    height: 390px;

    align-items: center;

    margin-left: -230px;

    margin-top: 0px;

}

.coverborder

{

    position: absolute;

    border: 5px solid #142458;

    width: 94%;

    height: 400px;

    margin-left: -228px;

    margin-top: 52px;

}

#coverimginput

{

  margin-top: 468px;

  margin-left: -200px;

  position: absolute;

}

#dpborder

{

    width: 180px;

    height: 180px;

    margin-top: 400px;

    position: absolute;

    margin-left: -36px;

    border-radius:100px;

    border: 5px solid #142458;

}

.profileimg

{

    display: flex;

    align-items: center;

    margin-left: 50px;

    justify-content: center;

}

.profileimgvisitor

{

    display: flex;

    align-items: center;

    margin-left: 50px;

    margin-top:25%;

    justify-content: center;

}

#profilephotoinputt

{

    height: 26px;

    margin-top: 430px;

    margin-left: -250px;

}

.dpinputtt

{

  position: relative;

  margin-left: -250px;

  margin-top: 500px;

}

.profilephoto{

    width: 172px;

    height: 170px;

    margin-top: 110px;

    border-radius:100px;

    margin-left: -1px;

    display:inline-block;

}

#profilephotoinput

{

    height: 26px;

    position: relative;

    margin-top: 412px;

    margin-left: -216px;

}

#dpconfirm

{

    height: 26px;

    margin: 20px;

    margin-top: 496px;

    position: relative;

    margin-left: -76px;

}

.profilebuttons{

  display: flex;

  align-items: right;

  justify-content: right;

  padding-right: 5px;

  position: relative;

  top: -190px;

}

.matchup2{

  background-color: rgb(37, 30, 131);

  color: #ffffff;

  margin-top: -100px;

  height: 45px;

  margin-left: 980px;

  position: absolute;

}

.message{

  background-color: rgb(37, 30, 131);

  color: #ffffff;

  margin-top: -100px;

  margin-left: 860px;

  position: absolute;

  height: 45px;

}

.logoutt{

  background-color: rgb(37, 30, 131);

  color: #ffffff;

  margin-top: 500px;

  margin-left: 1170px;

  position: absolute;

}

.logoutt:hover{

  background-color: red;

  color: #ffffff;

  transform: scale(1.1);

}

.details{

  display: flex;

  align-items: center;

  justify-content: center;

}

.detailsvisitor{

  display: flex;

  align-items: center;

  justify-content: center;

  margin-top:25%

}

.Name{

  font-family: 'Poppins', sans-serif;

  font-size: 30px;

  position: relative;

  top: -40px;

  margin-left: 20px;

}

.about{

  background: radial-gradient(circle, rgb(248, 187, 187) 5%, rgb(255, 230, 230) 95%);

  border-radius: 18px;

  height: auto;

  width: auto;

  text-align: center;

  position: relative;

  top:-280px;

  margin-top: 300px;

  font-family:Arial, Helvetica, sans-serif;

  padding-top:-15px;

  font-size: 17px;

}

#aboutbutton

{

  margin-bottom: 10px;

  background-color: rgb(96, 96, 247);

  line-height: 10px;

  height: 30px;

  width:80px;

  border: 5px solid rgb(96, 96, 247);

  color: aliceblue;

  border-radius: 10%;

  margin-left: 900px;

}

#status

{

  background-color: rgb(253, 233, 233);

  width:500px;

  height: 120px;

  color: black;

  font-size: 16px;

  font-family:Arial, Helvetica, sans-serif;

  position: relative;

  padding: 10px;

   border-radius: 10px;

   box-shadow: 0 15px 25px rgba(75, 75, 75, 0.6);

}

textarea[id^="status"]::-webkit-input-placeholder

{

  color: black;

  border: none;

  font-size: 18px;

  position: relative;

}

#statusbutton

{

  margin-bottom: 6px;

  background-color: rgb(96, 96, 247);

  line-height: 10px;

  height: 30px;

  width:80px;

  border: 5px solid rgb(96, 96, 247);

  color: aliceblue;

  border-radius: 10%;

  margin-left: 900px;

}

.statusresult{

  background: radial-gradient(circle, rgb(248, 187, 187) 5%, rgb(255, 230, 230) 95%);

  height: auto;

  width: auto;

  text-align: center;

  top:-180px;

  font-family: 'poppins',sans-serif;

  padding:40px;

  font-size: 17px;

}

.statusposted

{

  background-color: rgb(239, 236, 236);

  width:580px;

  height: 140px;

  color: black;

  font-size: 16px;

  font-family:Arial, Helvetica, sans-serif;

  border-width: 5px;

  border: 2px solid rgb(239, 236, 236);

  border-bottom-style: 2px solid rgb(244, 147, 147);

  padding: 10px;

  text-align: center;

  display: flex;

  align-items: center;

  justify-content: center;

  margin-left: 22%;

  box-shadow: 0 15px 25px rgba(75, 75, 75, 0.6);

  border-radius: 10px;

}

.teamgallery{

  background: radial-gradient(circle, rgb(248, 187, 187) 20%, rgb(255, 230, 230) 80%);

  border-radius: 18px;

  height: auto;

  width: auto;

  text-align: center;

  position: relative;

  top:-240px;

  font-family: 'poppins',sans-serif;

  padding:5px;

  font-size: 17px;

}

.teamgallerypics

{

  width:150px;

  height: 150px;

  margin: 10px;

  padding: 5px;

  align-items: left;

  justify-content: left;

}

.abouttext

{

  width: 70%;

  margin-top: 10px;

  height: auto;

  text-justify: auto;

  text-align: justify;

  text-align: left;

  font-size: 16px;

   background:  rgb(255, 230, 230) ;

  border: none;

  padding: 0px;

  box-shadow: 5px 2px 5px 1px gray;

  height: 150px;

  opacity:60% rgb(83, 52, 52);

  border-radius: 10px;

}

.innerdetails{

  align-items: center;

  color: rgb(49, 46, 46);

  font-size: 12px;

  justify-content: center;

  font-weight: 50;

}

.profilepage

{

  background: radial-gradient(circle, rgba(238,174,202,1) 0%, rgba(148,187,233,1) 100%);

  height: auto;

}

/\* Ground Profile \*/

.body{

  font-family: Montserrat, sans-serif;

  background: #28223f;

  height: auto;

}

.containerr1{

  position: absolute;

  user-select: none;

  margin: 100px auto;

  background: #231e39;

  color: #b3b8cd;

  border-radius: 5px;

  width: 1080px;

  text-align: left;

  box-shadow: 0 10px 20px -10px rgba(0,0,0,.75);

}

.cover-photo{

  /\* background: url(https://images.unsplash.com/photo-1540228232483-1b64a7024923?ixlib=rb-1.2.1&auto=format&fit=crop&w=967&q=80); \*/

  height: 400px;

  position: absolute;

  width: 1518px;

  border-radius: 5px 5px 0 0;

  margin-top: 72px;

  border: 5px solid black;

}

.coverpicc

{

  height: 390px;

  width: 1508px;

}

.dpinputt

{

  position: absolute;

  margin-left: 10px;

}

.dpborderr

{

    width: 180px;

    height: 180px;

    margin-top: 370px;

    position: absolute;

    margin-left: 100px;

    border-radius:100px;

    border: 8px solid #142458;

}

.dpphoto

{

  position: absolute;

  width: 170px;

  height: 170px;

  margin-top: -2px;

  margin-left: -3px;

  border-radius: 100%;

  display:inline-block;

}

#dpbuttonn

{

    height: 26px;

    position: absolute;

    margin-top: 512px;

    margin-left: 16px;

}

.profilename{

  font-size: 25px;

  font-weight: bold;

  color: rgb(198, 191, 191);

  margin: 27px 0 0 200px;

  margin-top: 520px;

  margin-left: 270px;

  position: absolute;

}

.about2{

  margin-top: 555px;

  margin-left: 270px;

  line-height: 21px;

  font-size: 15px;

  color: white;

  position: absolute;

}

#covergroundbtn

{

  position: absolute;

  margin-top: 520px;

  width: 80px;

  height: 26px;

  margin-left: 1344px;

  font-size: 10px;

  text-align: center;

}

.covergroundinput

{

  position: absolute;

  margin-top: 490px;

  margin-left: 1340px;

}

.groundgallery{

  background: radial-gradient(circle, rgb(249, 210, 227) 0%, #49406e 100%);

  border-radius: 30px;

  height: auto;

  width: 1200px;

  margin-top: 200px;

  margin-left: 160px;

  text-align: center;

  position: relative;

  top:-180px;

  font-family: 'poppins',sans-serif;

  padding:35px;

  font-size: 17px;

}

.groundtimelinepics

{

  width:600px;

  height: 600px;

  margin: 16px;

  padding: 5px;

  align-items: left;

  border-radius: 6%;

  justify-content: left;

}

.msg-btn, .follow-btn{

  margin: 10px 0 40px 200px;

  background: #03bfbc;

  border: 1px solid #03bfbc;

  padding: 10px 25px;

  color: #231e39;

  border-radius: 3px;

  font-family: Montserrat, sans-serif;

  cursor: pointer;

  margin-top: 40px;

  margin-left: 280px;

}

.follow-btn{

  position: relative;

  margin-left: 20px;

  margin-top: 620px;

  background: transparent;

  color: #02899c;

}

.follow-btn:hover{

  color: #231e39;

  background: #03bfbc;

  transition: .5s;

}

.container i{

  padding-left: 20px;

  font-size: 20px;

  margin-bottom: 20px;

  cursor: pointer;

  transition: .5s;

}

.container i:hover{

  color: #03bfbc;

}

.icons {

  margin-left: 180px;

}

.location{

  margin-left: 200px;

  margin-top:-10px;

  padding-bottom: 20px;

  color: whitesmoke;

  font-size: 17px;

}

/\* Booking \*/

.centerr3

  {

     position: relative;

     width: 260px;

     height: 300px;

     margin-top: 20px;

     margin-left: 50px;

     margin-bottom: 70px;

     left: 230px;

    }

.btnn{

    position: relative;

    background-color: #04AA6D;

    border-radius: 0px;

    font-size: 12px;

    color: #FFFFFF;

    padding: 10px;

    width: 90px;

    height: 35px;

    text-align: center;

    cursor: pointer;

}

.btnn:hover

{

    background-color: #058757;

}

.containerr{

    margin-top: 6px;

    margin-left: 6px;

    height: 140px;

    line-height: 12px;

}

#searchh{

    margin-top: 70px;

    display: block;

    width: 400px;

    height: 40px;

    margin-left: 720px;

    font-size: 14px;

}

.innerdata

{

  font-size: 12px;

}

input[id^="searchh"]::-webkit-input-placeholder {

    color: black;

    font-weight: 500;

    margin-left: 50px;

    font-size: 14px;

    text-transform: uppercase;

}

.hhh{

  text-align: center;

}

/\* Booking2 \*/

.bgimg2 {

  height: 40%;

  width: 100%;

  }

  .bookingcard

  {

    box-shadow: rgba(0, 0, 0, 0.25) 0px 54px 55px, rgba(0, 0, 0, 0.12) 0px -12px 30px, rgba(0, 0, 0, 0.12) 0px 4px 6px, rgba(0, 0, 0, 0.17) 0px 12px 13px, rgba(0, 0, 0, 0.09) 0px -3px 5px;

    text-align: center;

    background-color: #ddd;

    padding-bottom: 10px;

    margin-top: 90px;

    border-radius: 10px 10px 10px 10px;

  }

  .container1

{

  top:55%;

  right:20%;

  text-align: left;

  margin-left: 6px;

}

/\*1st Card \*/

.center3

  {

    position: relative;

    width: 260px;

    height: 300px;

    margin-top: 20px;

    margin-left: 50px;

    margin-bottom: 70px;

  }

/\* 2 card\*/

.center4

{

  position: absolute;

  top: 41%;

  left: 40%;

  transform: translate(-50%, -50%);

  font-size: 15px;

}

/\* 3 card\*/

.center5

{

  position: absolute;

  top: 41%;

  left: 62%;

  transform: translate(-50%, -50%);

  font-size: 15px;

}

/\* 4 card\*/

.center6

{

  position: absolute;

  top: 41%;

  left: 85%;

  transform: translate(-50%, -50%);

  font-size: 15px;

}

.center7

  {

    position: absolute;

    top: 82%;

    left: 16%;

    transform: translate(-50%, -50%);

    font-size: 15px;

  }

  .center8

  {

    position: absolute;

    top: 82%;

    left: 40%;

    transform: translate(-50%, -50%);

    font-size: 15px;

  }

  .center9

  {

    position: absolute;

    top: 82%;

    left: 62%;

    transform: translate(-50%, -50%);

    font-size: 15px;

  }

  .center10

  {

    position: absolute;

    top: 82%;

    left: 85%;

    transform: translate(-50%, -50%);

    font-size: 15px;

  }

  .center11

  {

    position: absolute;

    top: 122%;

    left: 16%;

    transform: translate(-50%, -50%);

    font-size: 15px;

  }

  .center12

  {

    position: absolute;

    top: 122%;

    left: 40%;

    transform: translate(-50%, -50%);

    font-size: 15px;

  }

  .center13

  {

    position: absolute;

    top: 122%;

    left: 62%;

    transform: translate(-50%, -50%);

    font-size: 15px;

  }

  .center14

  {

    position: absolute;

    top: 122%;

    left: 85%;

    transform: translate(-50%, -50%);

    font-size: 15px;

  }

  .btnn2{

    background-color: #04AA6D;

    border-radius: 0px;

    font-size: 12px;

    color: #FFFFFF;

    padding: 10px;

    width: 90px;

    height: 35px;

    text-align: center;

    cursor: pointer;

}

  .btnn2:hover

{

    background-color: #058757;

}

.groundname

{

  color: white;

 position: relative;

 text-align:center;

padding-top: 5%;

}

.dateheading

{

  color: white;

  position: relative;

  text-align:center;

 padding-top: 2%;

}

.formdate

{

  text-align:center;

  flex-direction: row;

}

#datestart

{

  position: absolute;

  height: 24px;

  font-size: 14px;

  width: 100px;

}

.allslots

{

  position: absolute;

  margin-top: -1140px;

  margin-left: 10px;

  width: 110px;

  font-size: 12px;

  height: 35px;

  border-radius: 0%;

  color: white;

  font-weight: bold;

  background-color: #04AA6D;

  cursor: pointer;

}

.avaslots

{

  position: absolute;

  cursor: pointer;

  margin-top: -1100px;

  margin-left: 10px;

  width: 160px;

  font-size: 12px;

  height: 35px;

  border-radius: 0%;

  color: white;

  font-weight: bold;

  background-color: #04AA6D;

}

**#index.js:**

import React from "react";

import ReactDOM from "react-dom";

import App from "./App.jsx";

import { BrowserRouter } from "react-router-dom";

import { Provider } from "react-redux";

import store from "./redux/store";

import "bootstrap/dist/css/bootstrap.min.css";

import 'bootstrap/dist/js/bootstrap.bundle.min.js';

ReactDOM.render(

  <Provider store={store}>

    <App></App>

  </Provider>,

  document.getElementById("root")

);

**#profile.jsx:**

import React from "react";

import AddIcon from "@mui/icons-material/Add";

import { useState, useEffect } from "react";

import Button from "@material-ui/core/Button";

import { Grid } from "@material-ui/core";

import axios from "axios";

import \* as CONTANT from "../Constant/constant.js";

import { useHistory } from "react-router-dom";

import { fontSize } from "@mui/system";

import { useSelector } from "react-redux";

import { ToastContainer, toast } from "react-toastify";

import "react-toastify/dist/ReactToastify.css";

import { storage } from ".//firebase";

import {

  ref,

  uploadBytes,

  getDownloadURL,

  listAll,

  list,

} from "firebase/storage";

import { v4 } from "uuid";

import { useDispatch } from "react-redux";

import {

  setTeamId,

  setAddress,

  setCaptainName,

  setEmail,

  setPhone,

  setTeamName,

} from "../redux/slices/teamSlice";

import { getStorage, deleteObject } from "firebase/storage";

// import firebase from "firebase/compat/app";

// import "firebase/compat/auth";

// import "firebase/compat/firestore";

function Profile() {

  const storagee = getStorage();

  const dispatch = useDispatch();

  const { localStorage } = window;

  const userEmail = useSelector((state) => state?.utilitySlice?.userEmail);

    const storedTeamId = localStorage.getItem("teamId");

    const reduxTeamId = useSelector((state) => state?.teamSlice?.teamId);

    const teamId = storedTeamId || reduxTeamId;

    console.log("teamId", teamId);

    const storedTeamName = localStorage.getItem("teamname");

    const reduxTeamName = useSelector((state) => state?.teamSlice?.teamname);

    const team = storedTeamName || reduxTeamName;

    console.log("team", team);

  //Dp

  const [dp\_Url, setDpUrl] = useState(localStorage.getItem("dpUrl"));

  const [dpUpload, setDpUpload] = useState(null);

  const [url3, setUrl3] = useState(null);

  useEffect(() => {

    setTimeout(() => {

      setDpUrl(localStorage.getItem("dpUrl"));

      getAllStatus();

    }, 1000);

  }, [dp\_Url]);

  const uploadDp = () => {

    if (dpUpload == null) return;

    // const dpRef = ref(storage, `teamdp/${dpUpload.name + v4()}`)

    const dpRef = ref(storage, `teamdp/${teamId}`);

    // const dpRef = ref(storage, `teamdp/${}`)

    uploadBytes(dpRef, dpUpload)

      .then(() => {

        getDownloadURL(dpRef)

          .then((url3) => {

            setDpUrl(url3);

            window.localStorage.setItem("dpUrl", url3);

            CONTANT.API.post("/team/dp", { \_id: teamId, imgURL: url3 });

          })

          .catch((error) => {

            console.log(error.message, "error getting while uploading");

          });

        setDpUpload(null);

      })

      .catch((error) => {

        console.log(error.message);

      });

  };

  //Cover

  const [cover\_Url, setCoverUrl] = useState(localStorage.getItem("coverUrl"));

  const [coverUpload, setCoverUpload] = useState(null);

  const [url2, setUrl2] = useState(null);

  useEffect(() => {

    getAllStatus();

  });

  useEffect(() => {

    setTimeout(() => {

      setCoverUrl(localStorage.getItem("coverUrl"));

      getAllStatus();

    }, 1000);

  }, [cover\_Url]);

  const uploadCover = () => {

    if (coverUpload == null) return;

    const coverRef = ref(storage, `teamcover/${teamId}`);

    uploadBytes(coverRef, coverUpload)

      .then(() => {

        getDownloadURL(coverRef)

          .then((url2) => {

            setCoverUrl(url2);

            window.localStorage.setItem("coverUrl", url2);

          })

          .catch((error) => {

            console.log(error.message, "error getting while uploading");

          });

        setCoverUpload(null);

      })

      .catch((error) => {

        console.log(error.message);

      });

  };

  //Delete Cover

  const Deletecover = () => {

    const storage = getStorage();

    // Create a reference to the file to delete

    const desertRef = ref(storage, `teamcover/${teamId}`);

    // Delete the file

    deleteObject(desertRef)

      .then(() => {

        window.localStorage.removeItem("coverUrl");

        setCoverUrl();

        // File deleted successfully

      })

      .catch((error) => {

        // Uh-oh, an error occurred!

      });

  };

  //Delete Dp

  const Deletedp = () => {

    const storage = getStorage();

    // Create a reference to the file to delete

    const desertRef = ref(storage, `teamdp/${teamId}`);

    // Delete the file

    deleteObject(desertRef)

      .then(() => {

        // File deleted successfully

        window.localStorage.removeItem("dpUrl");

        setDpUrl();

      })

      .catch((error) => {

        // Uh-oh, an error occurred!

      });

    // window.location.reload(false);

  };

  //Status

  const [status, setStatus] = useState("");

  const [statusArray, setStatusArray] = useState([]);

  const getAllStatus = async () => {

    const response = await CONTANT.API.get(`/team/getStatus/${teamId}`);

    if (response?.data) {

      setStatusArray(response.data.statusArray.reverse());

    }

  };

  //Timeline

  const [imageUpload, setImageUpload] = useState(null);

  const [imageList, setImageList] = useState([]);

  const [imagePath, setImagePath] = useState([]);

  const imageListRef = ref(storage, `teamtimeline/${teamId}`);

  const uploadImage = () => {

    if (imageUpload == null) return;

    const imageRef = ref(

      storage,

      `teamtimeline/${teamId}/${imageUpload.name + v4()}`

    );

    uploadBytes(imageRef, imageUpload).then((snapshot) => {

      getDownloadURL(snapshot.ref).then((url) => {

        const list = [...imageList];

        list.unshift(url);

        setImageList(list);

        window.localStorage.setItem("imageList", list);

      });

    });

  };

  const deletePicture = async (e, index) => {

    e.preventDefault();

    const list = [...imageList];

    const paths = [...imagePath];

    const deletePath = paths[index];

    paths.splice(index, 1);

    setImagePath(paths);

    list.splice(index, 1);

    setImageList(list);

    const storage = getStorage();

    // Create a reference to the file to delete

    const desertRef = ref(storage, deletePath);

    deleteObject(desertRef)

      .then(() => {

        // File deleted successfully

        console.log("Deleted");

      })

      .catch((error) => {

        // Uh-oh, an error occurred!

      });

    window.localStorage.setItem("imageList", list);

  };

  const deleteStatus = (e, index) => {

    e.preventDefault();

    const list = [...statusArray];

    list.splice(index, 1);

    setStatusArray(list);

    const url = list[index];

    console.log(url);

    // let imageRef = storagee.refFromURL(url);

    // imageRef.delete();

    window.localStorage.setItem("statusArray", list);

  };

  useEffect(() => {

    listAll(imageListRef).then((response) => {

      response.items.forEach((item) => {

        let path = imagePath;

        path.push(item.fullPath);

        setImagePath([...path]);

        getDownloadURL(item).then((url) => {

          let AllURL = imageList;

          AllURL.push(url);

          setImageList([...AllURL]);

        });

      });

    });

  }, []);

  const logout = () => {

    const { localStorage } = window;

    localStorage.clear();

    dispatch(setTeamId(""));

    dispatch(setTeamName(""));

    dispatch(setCaptainName(""));

    dispatch(setEmail(""));

    dispatch(setPhone(""));

    dispatch(setAddress(""));

  };

  const addStatus = async () => {

    const list = [...statusArray];

    list.unshift(status);

    setStatusArray(list);

    window.localStorage.setItem("statusArray", list);

    const response = await CONTANT.API.post(`/team/setStatus/${teamId}`, {

      status,

    });

    setStatus("");

    console.log(response);

  };

  return (

    <>

      <div className="profilepage">

        <ToastContainer />

        <div className="profilecontainer">

          <button className="logoutt" onClick={() => logout()}>

            Logout

          </button>

          {/\* Cover \*/}

          <input

            type="file"

            className="coverinputt"

            id="select-image2"

            onChange={(event) => {

              setCoverUpload(event.target.files[0]);

            }}

          />

          <label htmlFor="select-image2">

            <Button

              onClick={uploadCover}

              variant="contained"

              color="primary"

              id="coverimginput"

            >

              Upload Cover

            </Button>

          </label>

          <div className="coverborder">

            {cover\_Url && <img src={cover\_Url} className="coverimg" alt="" />}

          </div>

          {/\* Dp \*/}

          <div className="profiledetails">

            <div className="profileimg">

              <input

                type="file"

                className="dpinputtt"

                id="select-image"

                onChange={(event) => {

                  setDpUpload(event.target.files[0]);

                }}

              />

              <label htmlFor="select-image">

                <Button

                  onClick={uploadDp}

                  variant="contained"

                  color="primary"

                  id="profilephotoinputt"

                >

                  Upload Dp

                </Button>

              </label>

              <div id="dpborder">

                {dp\_Url && <img src={dp\_Url} className="profilephoto" alt="" />}

              </div>

            </div>

            <button

              className="message"

              onClick={() => {

                window.open(

                  "http://localhost:3001",

                  "\_blank"

                );

              }}

            >

              Message

            </button>

            <button className="matchup2">Match Up</button>

          </div>

          <Button onClick={Deletecover}> Delete Cover </Button>

          <Button onClick={Deletedp}> Delete Dp </Button>

          <br></br>

          <br></br>

          <div className="details">

            <h1 className="Name">{team}</h1>

          </div>

          {/\* About \*/}

          <form method="POST">

            {/\* Timeline \*/}

            <div className="teamgallery">

              <h2>Timeline</h2>

              <input

                type="file"

                onChange={(event) => {

                  setImageUpload(event.target.files[0]);

                }}

              />

              <br></br>

              <br></br>

              <div></div>

              <Button onClick={uploadImage} variant="contained" color="primary">

                Upload Image

              </Button>

              <br></br>

              <br></br>

              {imageList &&

                imageList?.map((url, index) => {

                  return (

                    <Grid key={index} container>

                      <Grid item xs={12}>

                        <img src={url} className="timelinepics" alt="" />

                      </Grid>

                      <Grid item xs={12}>

                        <Button

                          style={{}}

                          onClick={(e) => deletePicture(e, index)}

                          variant="contained"

                          color="secondary"

                        >

                          Delete Picture

                        </Button>

                      </Grid>

                    </Grid>

                  );

                })}

            </div>

            {/\*Status\*/}

            <div className="teamgallery">

              <h2>Status</h2>

              <div>

                <textarea

                  type="text"

                  name="status"

                  autoComplete="off"

                  placeholder="What's on Your Mind ......"

                  onChange={(event) => setStatus(event.target.value)}

                  value={status}

                  id="status"

                ></textarea>

              </div>

              <button id="statusbutton" onClick={addStatus} type="button">

                <p

                  style={{

                    marginLeft: "-11px",

                    marginTop: "-3px",

                    fontSize: "14px",

                  }}

                >

                  {" "}

                  Share{" "}

                </p>

              </button>

            </div>

          </form>

          <br></br>

          <br></br>

          {statusArray &&

            statusArray?.map((status, index) => {

              return (

                <div key={index} className="statusresult">

                  <div style={{ marginTop: "2vh", float: "right" }}>

                    <Button

                      style={{}}

                      onClick={(e) => deleteStatus(e, index)}

                      variant="contained"

                      color="secondary"

                    >

                      Delete Status

                    </Button>

                  </div>

                  <div className="statusposted">

                    <h1>{status}</h1>

                  </div>

                </div>

              );

            })}

        </div>

      </div>

    </>

  );

}

export default Profile;

**#teamprofile.css:**

/\*Profile Page\*/

#coverbutton

{

    margin-top: 100px;

    position: absolute;

    margin-left: -200px;

    width:95px;

}

.profilecontainer{

    padding: 20px  15%;

}

.coverimg {

    position: absolute;

    width:1738px;

    height: 390px;

    align-items: center;

    margin-left: -230px;

    margin-top: 0px;

}

.coverborder

{

    position: absolute;

    border: 5px solid #142458;

    width: 1518px;

    height: 400px;

    margin-left: -228px;

    margin-top: 52px;

}

#coverimginput

{

  margin-top: 468px;

  margin-left: -200px;

  position: absolute;

}

#dpborder

{

    width: 180px;

    height: 180px;

    margin-top: 400px;

    position: absolute;

    margin-left: -36px;

    border-radius:100px;

    border: 5px solid #142458;

}

.profileimg

{

    display: flex;

    align-items: center;

    margin-left: 50px;

    justify-content: center;

}

.profilephoto{

    width: 172px;

    height: 169px;

    margin-top: 1px;

    border-radius:100px;

    margin-left: -1px;

    display:inline-block;

}

#profilephotoinput

{

    height: 26px;

    position: relative;

    margin-top: 412px;

    margin-left: -216px;

}

#dpconfirm

{

    height: 26px;

    margin: 20px;

    margin-top: 496px;

    position: relative;

    margin-left: -76px;

}

.profilebuttons{

  display: flex;

  align-items: right;

  justify-content: right;

  padding-right: 5px;

  position: relative;

  top: -190px;

}

.matchup2{

  background-color: rgb(37, 30, 131);

  color: #ffffff;

  margin-top: -105px;

  height: 45px;

  margin-left: 980px;

  position: absolute;

}

.message{

  background-color: rgb(37, 30, 131);

  color: #ffffff;

  margin-top: -105px;

  margin-left: 860px;

  position: absolute;

  height: 45px;

}

.logout{

  background-color: rgb(37, 30, 131);

  color: #ffffff;

  margin-top: 500px;

  margin-left: 1170px;

  position: absolute;

}

.logout:hover{

  background-color: red;

  color: #ffffff;

  transform: scale(1.1);

}

.details{

  display: flex;

  align-items: center;

  justify-content: center;

}

.Name{

  font-family: 'Poppins', sans-serif;

  font-size: 30px;

  position: relative;

  top: -10px;

  margin-left: 20px;

}

.about{

  background-color: rgb(255, 230, 230);

  border-radius: 10px;

  height: auto;

  width: auto;

  text-align: center;

  position: relative;

  top:-250px;

  margin-top: 300px;

  font-family:Arial, Helvetica, sans-serif;

  padding-top:-15px;

  font-size: 17px;

}

#aboutbutton

{

  margin-bottom: 10px;

  background-color: rgb(96, 96, 247);

  line-height: 10px;

  height: 30px;

  width:80px;

  border: 5px solid rgb(96, 96, 247);

  color: aliceblue;

  border-radius: 10%;

  margin-left: 900px;

}

#status

{

  background-color: rgb(253, 233, 233);

  width:500px;

  height: 120px;

  color: black;

  font-size: 16px;

  font-family:Arial, Helvetica, sans-serif;

  position: relative;

  padding: 10px;

  box-shadow:  1px 5px 12px 4px rgb(255, 230, 230);

}

textarea[id^="status"]::-webkit-input-placeholder

{

  color: black;

  border: none;

  font-size: 18px;

  position: relative;

}

#statusbutton

{

  margin-bottom: 6px;

  background-color: rgb(96, 96, 247);

  line-height: 10px;

  height: 30px;

  width:80px;

  border: 5px solid rgb(96, 96, 247);

  color: aliceblue;

  border-radius: 10%;

  margin-left: 900px;

}

.statusresult{

  background-color: rgb(255, 230, 230);

  border-radius: 10px;

  height: auto;

  width: auto;

  text-align: center;

  position: relative;

  top:-180px;

  font-family: 'poppins',sans-serif;

  padding:40px;

  font-size: 17px;

}

.statusposted

{

  background-color: rgb(239, 236, 236);

  width:580px;

  height: 140px;

  color: black;

  font-size: 16px;

  font-family:Arial, Helvetica, sans-serif;

  border-width: 5px;

  border: 2px solid rgb(239, 236, 236);

  border-bottom-style: 2px solid rgb(244, 147, 147);

  position: relative;

  padding: 10px;

  text-align: center;

  display: flex;

  align-items: center;

  justify-content: center;

  margin-left: 22%;

  box-shadow: 0 15px 25px rgba(75, 75, 75, 0.6);

}

.teamgallery{

  background-color: rgb(255, 230, 230);

  border-radius: 10px;

  height: auto;

  width: auto;

  text-align: center;

  position: relative;

  top:-180px;

  font-family: 'poppins',sans-serif;

  padding:5px;

  font-size: 17px;

}

.teamgallerypics

{

  width:150px;

  height: 150px;

  margin: 10px;

  padding: 5px;

  align-items: left;

  justify-content: left;

}

.abouttext

{

  width: 1000px;

  margin-top: 10px;

  height: auto;

  text-justify: auto;

  text-align: justify;

  text-align: left;

  font-size: 16px;

  background-color: transparent;

  background: transparent;

  border: none;

  padding: 0px;

  box-shadow: 5px 2px 5px 1px gray;

  height: 150px;

  opacity:60% rgb(83, 52, 52);

}

.innerdetails{

  align-items: center;

  color: rgb(49, 46, 46);

  font-size: 12px;

  justify-content: center;

  font-weight: 50;

}

.profilepage

{

  background: rgb(238,174,202);

  background: radial-gradient(circle, rgba(238,174,202,1) 0%, rgba(148,187,233,1) 100%);

  height: auto;

}

/\*Profile Page\*/

#coverbutton

{

    margin-top: 100px;

    position: absolute;

    margin-left: -200px;

    width:95px;

}

.profilecontainer{

    padding: 20px  15%;

}

.coverimg {

    position: absolute;

    width:1738px;

    height: 390px;

    align-items: center;

    margin-left: -230px;

    margin-top: 0px;

}

.coverborder

{

    position: absolute;

    border: 5px solid #142458;

    width: 1518px;

    height: 400px;

    margin-left: -228px;

    margin-top: 52px;

}

#coverimginput

{

  margin-top: 468px;

  margin-left: -200px;

  position: absolute;

}

#dpborder

{

    width: 180px;

    height: 180px;

    margin-top: 400px;

    position: absolute;

    margin-left: -36px;

    border-radius:100px;

    border: 5px solid #142458;

}

.profileimg

{

    display: flex;

    align-items: center;

    margin-left: 50px;

    justify-content: center;

}

.profilephoto{

    width: 172px;

    height: 169px;

    margin-top: 1px;

    border-radius:100px;

    margin-left: -1px;

    display:inline-block;

}

#profilephotoinput

{

    height: 26px;

    position: relative;

    margin-top: -392px;

    margin-left: -216px;

    position: absolute;

}

#dpconfirm

{

    height: 26px;

    margin: 20px;

    margin-top: 496px;

    position: relative;

    margin-left: -76px;

}

.profilebuttons{

  display: flex;

  align-items: right;

  justify-content: right;

  padding-right: 5px;

  position: relative;

  top: -190px;

}

.matchup2{

  background-color: rgb(37, 30, 131);

  color: #ffffff;

  margin-top: -105px;

  height: 45px;

  margin-left: 980px;

  position: absolute;

}

.message{

  background-color: rgb(37, 30, 131);

  color: #ffffff;

  margin-top: -105px;

  margin-left: 860px;

  position: absolute;

  height: 45px;

}

.logout{

  background-color: rgb(37, 30, 131);

  color: #ffffff;

  margin-top: 500px;

  margin-left: 1170px;

  position: absolute;

}

.logout:hover{

  background-color: red;

  color: #ffffff;

  transform: scale(1.1);

}

.details{

  display: flex;

  align-items: center;

  justify-content: center;

}

.Name{

  font-family: 'Poppins', sans-serif;

  font-size: 30px;

  position: relative;

  top: -10px;

  margin-left: 20px;

}

.about{

  background-color: rgb(255, 230, 230);

  border-radius: 10px;

  height: auto;

  width: auto;

  text-align: center;

  position: relative;

  top:-250px;

  margin-top: 300px;

  font-family:Arial, Helvetica, sans-serif;

  padding-top:-15px;

  font-size: 17px;

}

#aboutbutton

{

  margin-bottom: 10px;

  background-color: rgb(96, 96, 247);

  line-height: 10px;

  height: 30px;

  width:80px;

  border: 5px solid rgb(96, 96, 247);

  color: aliceblue;

  border-radius: 10%;

  margin-left: 900px;

}

#status

{

  background-color: rgb(253, 233, 233);

  width:500px;

  height: 120px;

  color: black;

  font-size: 16px;

  font-family:Arial, Helvetica, sans-serif;

  position: relative;

  padding: 10px;

  box-shadow:  1px 5px 12px 4px rgb(255, 230, 230);

}

textarea[id^="status"]::-webkit-input-placeholder

{

  color: black;

  border: none;

  font-size: 18px;

  position: relative;

}

#statusbutton

{

  margin-bottom: 6px;

  background-color: rgb(96, 96, 247);

  line-height: 10px;

  height: 30px;

  width:80px;

  border: 5px solid rgb(96, 96, 247);

  color: aliceblue;

  border-radius: 10%;

  margin-left: 900px;

}

.statusresult{

  background-color: rgb(255, 230, 230);

  border-radius: 10px;

  height: auto;

  width: auto;

  text-align: center;

  position: relative;

  top:-180px;

  font-family: 'poppins',sans-serif;

  padding:40px;

  font-size: 17px;

}

.statusposted

{

  background-color: rgb(239, 236, 236);

  width:580px;

  height: 140px;

  color: black;

  font-size: 16px;

  font-family:Arial, Helvetica, sans-serif;

  border-width: 5px;

  border: 2px solid rgb(239, 236, 236);

  border-bottom-style: 2px solid rgb(244, 147, 147);

  position: relative;

  padding: 10px;

  text-align: center;

  display: flex;

  align-items: center;

  justify-content: center;

  margin-left: 22%;

  box-shadow: 0 15px 25px rgba(75, 75, 75, 0.6);

}

.teamgallery{

  background-color: rgb(255, 230, 230);

  border-radius: 10px;

  height: auto;

  width: auto;

  text-align: center;

  position: relative;

  top:-180px;

  font-family: 'poppins',sans-serif;

  padding:5px;

  font-size: 17px;

}

.teamgallerypics

{

  width:150px;

  height: 150px;

  margin: 10px;

  padding: 5px;

  align-items: left;

  justify-content: left;

}

.abouttext

{

  width: 1000px;

  margin-top: 10px;

  height: auto;

  text-justify: auto;

  text-align: justify;

  text-align: left;

  font-size: 16px;

  background-color: transparent;

  background: transparent;

  border: none;

  padding: 0px;

  box-shadow: 5px 2px 5px 1px gray;

  height: 150px;

  opacity:60% rgb(83, 52, 52);

}

.innerdetails{

  align-items: center;

  color: rgb(49, 46, 46);

  font-size: 12px;

  justify-content: center;

  font-weight: 50;

}

.profilepage

{

  background: rgb(238,174,202);

  background: radial-gradient(circle, rgba(238,174,202,1) 0%, rgba(148,187,233,1) 100%);

  height: auto;

}

**#teamline.css:**

/\* Custom CSS for ProfilePage component \*/

/\* General Page Styling \*/

.profile-page {

  min-height: 100vh;

  background-color: #f0f2f5;

  padding-bottom: 3rem;

  margin-top: 70px;

  position: absolute;

  width: 100%;

}

/\* Cover Photo \*/

.cover-photo-container {

  height: 400px;

  overflow: hidden;

  position: relative;

  background-color: #e9ecef;

}

.cover-photo {

  width: 100%;

  height: 100%;

  object-fit: cover;

}

.cover-placeholder {

  width: 100%;

  height: 100%;

  background-color: #6c757d;

  background-image: linear-gradient(to bottom right, #064a93, #000000);

}

/\* Profile Header \*/

.profile-header {

  position: relative;

  margin-top: -80px;

  padding-left: 1rem;

  padding-right: 1rem;

  z-index: 1;

}

.profile-picture-container {

  margin-right: 1rem;

}

.profile-picture, .profile-picture-placeholder {

  width: 160px;

  height: 160px;

  object-fit: cover;

  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);

}

.profile-picture-placeholder {

  background-color: #e9ecef;

  display: flex;

  align-items: center;

  justify-content: center;

}

.profile-picture-placeholder::after {

  content: 'Team';

  font-size: 20px;

  color: #6c757d;

}

.team-name {

  margin-bottom: 30px;

  font-size: 2.5rem;

  color: #212529;

}

/\* Status Cards \*/

.status-card {

  transition: transform 0.2s;

}

.status-card:hover {

  transform: translateY(-3px);

}

.status-profile-pic {

  width: 30px;

  height: 30px;

  object-fit: cover;

}

/\* Timeline Gallery \*/

.timeline-gallery {

  margin-top: 1rem;

}

.timeline-image-card {

  transition: all 0.3s ease;

  border: none;

  overflow: hidden;

}

.timeline-image-card:hover {

  box-shadow: 0 10px 15px rgba(0, 0, 0, 0.1);

}

.timeline-image {

  height: 200px;

  object-fit: cover;

}

/\* Custom Buttons \*/

.btn {

  border-radius: 0.5rem;

  padding: 0.5rem 1rem;

}

.btn-sm {

  padding: 0.25rem 0.5rem;

  font-size: 0.875rem;

}

/\* Responsive Adjustments \*/

@media (max-width: 767px) {

  .cover-photo-container {

    height: 200px;

  }

  .profile-header {

    margin-top: -40px;

  }

  .profile-picture, .profile-picture-placeholder {

    width: 100px;

    height: 100px;

  }

  .team-name {

    font-size: 1.5rem;

  }

}

/\* Bootstrap Icon Integration \*/

@import url("https://cdn.jsdelivr.net/npm/bootstrap-icons@1.8.1/font/bootstrap-icons.css");

/\* Ground Timeline \*/

/\* Custom CSS for Ground Profile Page \*/

/\* Import Bootstrap icons \*/

@import url("https://cdn.jsdelivr.net/npm/bootstrap-icons@1.8.1/font/bootstrap-icons.css");

/\* General Page Styling \*/

.ground-page {

  min-height: 100vh;

  background-color: #f8f9fa;

  padding-bottom: 3rem;

  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

  margin-top: 70px;

  position: absolute;

  width: 100%;

}

/\* Hero Section with Cover Photo \*/

.ground-hero-section {

  position: relative;

  margin-bottom: 2rem;

}

.ground-cover-container {

  position: relative;

  height: 380px;

  overflow: hidden;

  background-color: #343a40;

}

.ground-cover-image {

  width: 100%;

  height: 100%;

  object-fit: cover;

  filter: brightness(0.85);

}

.ground-cover-placeholder {

  width: 100%;

  height: 100%;

  display: flex;

  align-items: center;

  justify-content: center;

  background: linear-gradient(135deg, #2c3e50, #4ca1af);

  color: white;

  font-size: 1.5rem;

  text-align: center;

  padding: 2rem;

}

.ground-cover-controls {

  position: absolute;

  bottom: 20px;

  right: 20px;

  display: flex;

  z-index: 2;

}

.ground-file-input {

  display: none;

}

/\* Content Container \*/

.ground-content-container {

  margin-top: -80px;

  position: relative;

  z-index: 3;

}

/\* Ground Profile Card \*/

.ground-profile-card {

  background-color: white;

  border-radius: 10px;

  box-shadow: 0 8px 20px rgba(0, 0, 0, 0.1);

  padding: 1.5rem;

  margin-bottom: 2rem;

}

/\* DP (Logo) Section \*/

.ground-dp-section {

  display: flex;

  flex-direction: column;

  align-items: center;

  margin-bottom: 1.5rem;

}

.ground-dp-container {

  width: 180px;

  height: 180px;

  border-radius: 50%;

  overflow: hidden;

  margin-bottom: 1rem;

  border: 5px solid white;

  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);

  background-color: #f8f9fa;

}

.ground-dp-image {

  width: 100%;

  height: 100%;

  object-fit: cover;

}

.ground-dp-placeholder {

  width: 100%;

  height: 100%;

  display: flex;

  align-items: center;

  justify-content: center;

  font-size: 2.5rem;

  color: #adb5bd;

  background-color: #e9ecef;

}

.ground-dp-controls {

  display: flex;

  gap: 0.5rem;

}

/\* Ground Info Section \*/

.ground-info-section {

  text-align: center;

}

.ground-name {

  font-size: 1.8rem;

  font-weight: 700;

  margin-bottom: 0.5rem;

  color: #343a40;

}

.ground-description {

  color: #6c757d;

  margin-bottom: 1.5rem;

  line-height: 1.5;

}

/\* Action Buttons \*/

.ground-action-buttons {

  display: grid;

  grid-template-columns: 1fr 1fr;

  gap: 0.8rem;

}

.ground-msg-btn,

.ground-booking-btn,

.ground-academy-btn,

.ground-location-btn {

  display: flex;

  align-items: center;

  justify-content: center;

  padding: 0.6rem;

  border-radius: 8px;

  font-weight: 500;

  transition: all 0.2s ease;

}

.ground-msg-btn {

  background-color: #0d6efd;

}

.ground-booking-btn {

  background-color: #17a2b8;

}

.ground-academy-btn {

  background-color: #6c757d;

}

.ground-location-btn {

  background-color: #28a745;

}

/\* Timeline Section \*/

.ground-timeline-section {

  background-color: white;

  border-radius: 10px;

  box-shadow: 0 8px 20px rgba(0, 0, 0, 0.1);

  padding: 1.5rem;

}

.ground-timeline-header {

  display: flex;

  flex-direction: column;

  margin-bottom: 1.5rem;

  border-bottom: 1px solid #e9ecef;

  padding-bottom: 1.5rem;

}

.ground-timeline-title {

  font-size: 1.5rem;

  font-weight: 600;

  margin-bottom: 1rem;

  color: #343a40;

  display: flex;

  align-items: center;

}

.ground-timeline-upload {

  display: flex;

  gap: 0.5rem;

  flex-wrap: wrap;

}

/\* Gallery Styling \*/

.ground-timeline-gallery {

  padding-top: 1rem;

}

.ground-gallery-item {

  position: relative;

  border-radius: 8px;

  overflow: hidden;

  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);

  margin-bottom: 1rem;

  transition: transform 0.3s ease;

}

.ground-gallery-item:hover {

  transform: translateY(-5px);

}

.ground-gallery-image-container {

  height: 220px;

  overflow: hidden;

}

.ground-gallery-image {

  width: 100%;

  height: 100%;

  object-fit: cover;

  transition: transform 0.3s ease;

}

.ground-gallery-item:hover .ground-gallery-image {

  transform: scale(1.05);

}

.ground-gallery-actions {

  padding: 0.8rem;

  background-color: #f8f9fa;

  border-top: 1px solid #e9ecef;

}

.ground-delete-btn {

  width: 100%;

  border-radius: 6px;

}

/\* Empty Gallery State \*/

.ground-empty-gallery {

  border: 2px dashed #dee2e6;

  border-radius: 8px;

  padding: 2rem;

  color: #6c757d;

}

.ground-empty-icon {

  font-size: 3rem;

  margin-bottom: 1rem;

  color: #adb5bd;

}

/\* Button Styling \*/

.ground-select-btn,

.ground-upload-btn {

  display: flex;

  align-items: center;

  padding: 0.5rem 1rem;

  border-radius: 6px;

}

.ground-select-btn {

  background-color: transparent;

}

.ground-upload-btn {

  background-color: #28a745;

  border-color: #28a745;

}

/\* Responsive Styles \*/

@media (max-width: 767px) {

  .ground-cover-container {

    height: 250px;

  }

  .ground-content-container {

    margin-top: -40px;

  }

  .ground-dp-container {

    width: 120px;

    height: 120px;

  }

  .ground-timeline-header {

    flex-direction: column;

  }

  .ground-timeline-upload {

    margin-top: 1rem;

  }

  .ground-cover-controls {

    flex-direction: column;

    gap: 0.5rem;

    align-items: flex-end;

  }

}

**#user.js:**

import { useState, useEffect } from 'react';

import Button from '@material-ui/core/Button';

import Box from '@material-ui/core/Box';

const User = () => {

  const [selectedImage, setSelectedImage] = useState(null);

  const [imageUrl, setImageUrl] = useState(null);

  useEffect(() => {

    if (selectedImage) {

      setImageUrl(URL.createObjectURL(selectedImage));

    }

  }, [selectedImage]);

  return (

    <>

      <input

        accept="image/\*"

        type="file"

        id="select-image"

        style={{ display: 'none' }}

        onChange={e => setSelectedImage(e.target.files[0])}

      />

      <label htmlFor="select-image">

        <Button variant="contained" color="primary" component="span">

          Upload photo

        </Button>

      </label>

      {imageUrl && selectedImage && (

        <Box mt={2} textAlign="center">

          <img src={imageUrl} alt={selectedImage.name} height="200px" />

        </Box>

      )}

    </>

  );

};

export default User;

* **Package-lock.json**
* **Package.json**

**New working :**

**Backend:  
Routes:**

**#chatbot.js:**import express from "express";

import { GoogleGenerativeAI } from "@google/generative-ai";

import Ground from "../models/ground.js";

import Academy from "../models/academy.js";

import Team from "../models/team.js";

const router = express.Router();

// POST /api/chatbot

router.post("/", async (req, res) => {

  try {

    // --- FIX: Initialize Gemini AI client here, inside the handler ---

    // This ensures that process.env has been loaded by dotenv in App.js first.

    const genAI = new GoogleGenerativeAI(process.env.GEMINI\_API\_KEY);

    const model = genAI.getGenerativeModel({ model: "gemini-1.5-flash" });

    // --- END OF FIX ---

    const { prompt } = req.body;

    console.log("Received prompt from frontend:", prompt);

    if (!prompt) {

      return res.status(400).send({ error: "Prompt is required." });

    }

    // Fetch all necessary data from your database

    console.log("Fetching data from database...");

    const grounds = await Ground.find({});

    const academies = await Academy.find({});

    const teams = await Team.find({});

    console.log(`Data fetched: ${grounds.length} grounds, ${academies.length} academies, ${teams.length} teams.`);

    // Create a detailed context for the Gemini API

    const context = `

      You are a helpful and friendly chatbot for a Futsal Field Reservation System.

      Your main goal is to provide information about our services based on the data provided below.

      Please be conversational and answer questions in a clear and concise manner.

      Here is the data you have access to:

      \*\*Grounds Information:\*\*

      ${JSON.stringify(grounds, null, 2)}

      \*\*Academies Information:\*\*

      ${JSON.stringify(academies, null, 2)}

      \*\*Teams Information:\*\*

      ${JSON.stringify(teams, null, 2)}

      \*\*Donation Information:\*\*

      We welcome donations to support our community and local youth sports programs. Users can make a donation by navigating to the "Donation" section of our website.

      Based on this information, please answer the following user query:

    `;

    console.log("Sending request to Gemini API...");

    // Generate content with Gemini

    const result = await model.generateContent([context, prompt]);

    const response = await result.response;

    const text = await response.text();

    console.log("Received response from Gemini:", text);

    // Send the response back to the frontend

    res.send({ response: text });

  } catch (error) {

    // Log the full error object for detailed diagnostics

    console.error("--- FULL ERROR in /api/chatbot route ---");

    console.error(error);

    console.error("--- END OF ERROR ---");

    res.status(500).send({ error: "An unexpected error occurred. Please try again later." });

  }

});

export default router;

**Frontend:**

**SRC/Pages:**

**#chatbot.jsx:**

import React, { useState, useEffect, useRef } from "react";

import "../chatbot.css"; // We'll create this file next

import \* as CONSTANT from "../Constant/constant";

const Chatbot = () => {

  const [isOpen, setIsOpen] = useState(false);

  const [messages, setMessages] = useState([

    { text: "Hello! How can I help you with the Futsal Reservation System today?", sender: "bot" }

  ]);

  const [inputValue, setInputValue] = useState("");

  const [isLoading, setIsLoading] = useState(false);

  const messagesEndRef = useRef(null);

  const scrollToBottom = () => {

    messagesEndRef.current?.scrollIntoView({ behavior: "smooth" });

  };

  useEffect(scrollToBottom, [messages]);

  const toggleChatbot = () => {

    setIsOpen(!isOpen);

  };

  const handleInputChange = (e) => {

    setInputValue(e.target.value);

  };

  const handleSendMessage = async () => {

    if (inputValue.trim() === "" || isLoading) return;

    const userMessage = { text: inputValue, sender: "user" };

    setMessages(prevMessages => [...prevMessages, userMessage]);

    setInputValue("");

    setIsLoading(true);

    try {

      const res = await CONSTANT.API.post("/api/chatbot", { prompt: inputValue });

      const botResponse = { text: res.data.response, sender: "bot" };

      setMessages(prevMessages => [...prevMessages, botResponse]);

    } catch (error) {

      console.error("Error fetching chatbot response:", error);

      const errorResponse = { text: "Sorry, I'm having trouble connecting. Please try again later.", sender: "bot" };

      setMessages(prevMessages => [...prevMessages, errorResponse]);

    } finally {

      setIsLoading(false);

    }

  };

  return (

    <div className={`chatbot-container ${isOpen ? "open" : ""}`}>

       <div className="chatbot-header" onClick={toggleChatbot}>

        <h2>Futsal Assistant</h2>

        <span className="chatbot-toggle-icon">{isOpen ? "▼" : "▲"}</span>

      </div>

      {isOpen && (

        <div className="chatbot-body">

          <div className="chatbot-messages">

            {messages.map((message, index) => (

              <div key={index} className={`message-bubble ${message.sender}`}>

                {message.text}

              </div>

            ))}

            {isLoading && (

              <div className="message-bubble bot">

                <div className="typing-indicator">

                  <span></span>

                  <span></span>

                  <span></span>

                </div>

              </div>

            )}

            <div ref={messagesEndRef} />

          </div>

          <div className="chatbot-input">

            <input

              type="text"

              value={inputValue}

              onChange={handleInputChange}

              onKeyPress={(e) => e.key === "Enter" && handleSendMessage()}

              placeholder="Ask me anything..."

              disabled={isLoading}

            />

            <button onClick={handleSendMessage} disabled={isLoading}>

              Send

            </button>

          </div>

        </div>

      )}

    </div>

  );

};

export default Chatbot;