const express = require("express");

require("dotenv").config();

const path = require("path");

const hbs = require("hbs");

const cookieParser = require("cookie-parser");

const cors = require("cors");

const { authenticateToken } = require("./middleware/authMiddleware");

// Import routes

const authRoutes = require("./routes/auth");

const teamRoutes = require("./routes/teamRoutes");

const bookingRoutes = require("./routes/bookingRoutes");

const dashboardRoutes = require("./routes/dashboardRoutes");

// Database connection and models

require("./db/conn");

const Signup = require("./models/signup");

const app = express();

const port = process.env.PORT || 3000;

console.log("🚀 Starting server...");

console.log(`🔑 JWT\_SECRET: ${process.env.JWT\_SECRET}`);

// Define paths

const static\_path = path.join(\_\_dirname, "../login");

const templates\_path = path.join(\_\_dirname, "../templates/views");

const partials\_path = path.join(\_\_dirname, "../templates/partials");

console.log("📂 Paths defined:");

console.log(`🔹 Static Path: ${static\_path}`);

console.log(`🔹 Templates Path: ${templates\_path}`);

console.log(`🔹 Partials Path: ${partials\_path}`);

// Middleware

console.log("🛠️ Setting up middleware...");

app.use(express.static(static\_path));

app.use(express.static(path.join(\_\_dirname, "public")));

app.set("view engine", "hbs");

app.set("views", templates\_path);

app.use("/uploads", express.static("uploads"));

hbs.registerPartials(partials\_path);

app.use(cors({ origin: "http://localhost:3000", credentials: true })); // Use cors once

app.use(cookieParser());

app.use(express.json());

app.use(express.urlencoded({ extended: true }));

console.log("✅ Middleware setup complete.");

// Register routes

console.log("🛣️ Registering routes...");

app.use("/api/auth", authRoutes);

app.use("/api/teams", teamRoutes);

app.use("/api/bookings", bookingRoutes);

app.use("/api/dashboard", dashboardRoutes);

console.log("✅ Routes registered.");

// Serve CSS file

app.use("/styles.css", (req, res) => {

    console.log("🎨 Serving CSS file...");

    res.type("text/css");

    res.sendFile(path.join(\_\_dirname, "public", "styles.css"));

});

// Routes

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

    console.log("🔗 Rendering login page...");

    res.render("login");

});

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

    console.log("🔗 Rendering signup page...");

    res.render("signup");

});

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

    console.log("🔗 Rendering home page...");

    res.render("home");

});

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

    console.log("🔗 Rendering book-now page...");

    res.render("book-now");

});

// Single /teams route

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

    console.log("🔗 Fetching teams data...");

    try {

        const teams = await require("./models/team").find();

        console.log("✅ Teams data fetched successfully.");

        res.render("teams", { teams });

    } catch (error) {

        console.error("❌ Error fetching teams:", error);

        res.status(500).send("Internal Server Error");

    }

});

app.get("/ground-owner-dashboard", (req, res) => {

    console.log("🔗 Rendering ground owner dashboard...");

    res.render("groundOwnerDashboard");

});

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

    console.log("🔗 Rendering team dashboard...");

    res.render("teamDashboard");

});

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

    console.log("🔗 Rendering dashboard...");

    console.log("👤 User data:", req.user);

    res.render("dashboard", { user: req.user });

});

// Signup Route

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

    console.log("🔗 Handling signup request...");

    try {

        const { role, yourname, email, password, confirmpassword, phonenumber, groundname, groundlocation, teamname } = req.body;

        console.log("📝 Signup data received:");

        console.log(`🔹 Role: ${role}`);

        console.log(`🔹 Name: ${yourname}`);

        console.log(`🔹 Email: ${email}`);

        console.log(`🔹 Phone: ${phonenumber}`);

        if (password !== confirmpassword) {

            console.log("❌ Passwords do not match.");

            return res.status(400).send("Passwords do not match");

        }

        let newUser;

        if (role === "groundOwner") {

            newUser = new Signup({

                role: "Ground Owner",

                yourname,

                email,

                password,

                phonenumber,

                groundname,

                groundlocation

            });

        } else if (role === "teamOwner") {

            newUser = new Signup({

                role: "Team Owner",

                yourname,

                email,

                password,

                phonenumber,

                teamname

            });

        } else {

            console.log("❌ Invalid signup role.");

            return res.status(400).send("Invalid signup role");

        }

        await newUser.save();

        console.log("✅ User saved:", newUser);

        res.status(201).redirect("/login");

    } catch (error) {

        console.error("❌ Signup Error:", error);

        res.status(500).send("Error in signup: " + error.message);

    }

});

// Start Server

app.listen(port, () => {

    console.log(`🚀 Server is running at port ${port}`);

});