

Food Delivery Web Application - Full Code

```
// Import necessary modules
const express = require("express");
const mongoose = require("mongoose");
const cors = require("cors");
const dotenv = require("dotenv");
const path = require("path");

// Load environment variables
dotenv.config();

// Initialize Express app
const app = express();
app.use(express.json());
app.use(cors());

// Connect to MongoDB
mongoose.connect(process.env.MONGO_URI, { useNewUrlParser: true, useUnifiedTopology: true })
  .then(() => console.log("MongoDB Connected"))
  .catch((err) => console.log(err));

// Define User Schema & Model
const userSchema = new mongoose.Schema({ name: String, email: String, password: String });
const User = mongoose.model("User", userSchema);

// Define Restaurant Schema & Model
const restaurantSchema = new mongoose.Schema({ name: String, address: String, menu: Array });
const Restaurant = mongoose.model("Restaurant", restaurantSchema);

// Define Order Schema & Model
const orderSchema = new mongoose.Schema({ userId: mongoose.Schema.Types.ObjectId, restaurantId: mongoose.Schema.Types.ObjectId, items: Array });
const Order = mongoose.model("Order", orderSchema);

// User Routes
app.post("/api/users/register", async (req, res) => {
  const { name, email, password } = req.body;
  const user = new User({ name, email, password });
  await user.save();
  res.json({ message: "User registered successfully" });
});

app.post("/api/users/login", async (req, res) => {
  const { email, password } = req.body;
  const user = await User.findOne({ email, password });
  if (user) {
    res.json({ message: "Login successful", user });
  } else {
    res.status(401).json({ message: "Invalid credentials" });
  }
});
```

```

// Restaurant Routes
app.get("/api/restaurants", async (req, res) => {
  const restaurants = await Restaurant.find();
  res.json(restaurants);
});

// Order Routes
app.post("/api/orders", async (req, res) => {
  const { userId, restaurantId, items } = req.body;
  const order = new Order({ userId, restaurantId, items, status: "Pending" });
  await order.save();
  res.json({ message: "Order placed successfully" });
});

// Serve React Frontend (Optional for Deployment)
app.use(express.static(path.join(__dirname, "frontend/build")));
app.get("*", (req, res) => {
  res.sendFile(path.join(__dirname, "frontend/build", "index.html"));
});

// Start the server
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));

/* ----- Frontend Code (React.js) ----- */

// src/App.js
const React = require("react");
const { BrowserRouter, Routes, Route } = require("react-router-dom");

function Home() { return React.createElement("h1", null, "Welcome to Food Delivery App"); }
function Login() { return React.createElement("h1", null, "Login Page"); }
function Register() { return React.createElement("h1", null, "Register Page"); }
function Navbar() { return React.createElement("nav", null, "Navbar"); }

function App() {
  return React.createElement(
    BrowserRouter,
    null,
    React.createElement(Navbar, null),
    React.createElement(
      Routes,
      null,
      React.createElement(Route, { path: "/", element: React.createElement(Home, null) }),
      React.createElement(Route, { path: "/login", element: React.createElement(Login, null) }),
      React.createElement(Route, { path: "/register", element: React.createElement(Register, null) })
    )
  );
}

module.exports = App;

```