**Our Diploma Project Back End**

**Server side:**

const express = require("express");

const mysql = require("mysql2");

const cors = require("cors");

require("dotenv").config();

const app = express();

app.use(cors());

app.use(express.json());

// connect to the database

const db = mysql.createConnection({

  host: process.env.DB\_HOST,

  user: process.env.DB\_USER,

  password: process.env.DB\_PASSWORD,

  database: process.env.DB\_NAME

});

db.connect(err => {

  if (err) throw err;

  console.log("✅ MySQL Connected");

});

// signup

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

  const { email, password } = req.body;

  db.query("SELECT \* FROM users WHERE email = ?", [email], (err, results) => {

    if (err) return res.status(500).json({ error: "Database error" });

    if (results.length > 0) {

      return res.status(409).json({ message: "User already exists" });

    }

    db.query(

      "INSERT INTO users (email, password) VALUES (?, ?)",

      [email, password],

      (err, result) => {

        if (err) return res.status(500).json({ error: "Failed to create user" });

        res.status(201).json({ message: "User created successfully" });

      }

    );

  });

});

// login

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

  const { email, password } = req.body;

  db.query(

    "SELECT \* FROM users WHERE email = ? AND password = ?",

    [email, password],

    (err, results) => {

      if (err) return res.status(500).json({ error: err.message });

      if (results.length === 0) return res.status(401).json({ message: "Invalid credentials" });

      res.json({ message: "Login successful" });

    }

  );

});

// take the data from manuall analysis table

app.post("/api/analyze", (req, res) => {

  const { city, businessType, budget, email } = req.body;

  // budget level

  let budgetLevel = "high";

  if (budget < 1000) {

    budgetLevel = "low";

  } else if (budget < 3000) {

    budgetLevel = "medium";

  }

  db.query(

    "SELECT \* FROM manual\_analysis WHERE city = ? AND business\_type = ? AND budget\_level = ?",

    [city, businessType, budgetLevel],

    (err, results) => {

      if (err) {

        console.error("❌ Error fetching manual analysis:", err);

        return res.status(500).json({ message: "Database query error" });

      }

      if (results.length === 0) {

        return res.status(404).json({ message: "No analysis found for the given inputs" });

      }

      const analysis = results[0];

      // save the analysis in history table

      db.query(

        "INSERT INTO analysis\_history (email, business\_type, city, budget, budget\_level, result) VALUES (?, ?, ?, ?, ?, ?)",

        [email, businessType, city, budget, budgetLevel, analysis.final\_report],

        (err, result) => {

          if (err) {

            console.error("Error saving analysis history:", err);

          } else {

            console.log("✅ Analysis history saved");

          }

        }

      );

      res.json({ analysis });

    }

  );

});

// return the history for the user

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

  const { email } = req.query;

  db.query(

    "SELECT \* FROM analysis\_history WHERE email = ? ORDER BY created\_at DESC",

    [email],

    (err, results) => {

      if (err) {

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

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

      }

      res.json({ history: results });

    }

  );

});

// delete from the history

app.delete("/api/history/:id", (req, res) => {

  const { id } = req.params;

  db.query("DELETE FROM analysis\_history WHERE id = ?", [id], (err, result) => {

    if (err) {

      console.error("❌ Error deleting analysis:", err);

      return res.status(500).json({ message: "Failed to delete analysis" });

    }

    res.json({ message: "Analysis deleted successfully" });

  });

});

// server

app.listen(5000, () => {

  console.log("✅ Server running on http://localhost:5000");

});

**Env File:**

DB\_HOST=localhost

DB\_USER=root

DB\_PASSWORD=ALazri@22f22901

DB\_NAME=market\_app

**Database side:**

CREATE DATABASE IF NOT EXISTS market\_app;

USE market\_app;

CREATE TABLE IF NOT EXISTS users (

id INT AUTO\_INCREMENT PRIMARY KEY,

email VARCHAR(255) NOT NULL,

password VARCHAR(255) NOT NULL

);

CREATE TABLE IF NOT EXISTS manual\_analysis (

id INT AUTO\_INCREMENT PRIMARY KEY,

business\_type VARCHAR(100),

city VARCHAR(100),

budget\_level ENUM('low', 'medium', 'high'),

best\_area VARCHAR(255),

competition\_level VARCHAR(100),

rent\_estimate VARCHAR(100),

suggestions TEXT,

google\_map\_url TEXT,

final\_report TEXT

);

CREATE TABLE analysis\_history (

id INT AUTO\_INCREMENT PRIMARY KEY,

email VARCHAR(255) NOT NULL,

business\_type VARCHAR(100) NOT NULL,

city VARCHAR(100) NOT NULL,

budget INT NOT NULL,

budget\_level ENUM('low', 'medium', 'high') NOT NULL,

result TEXT NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);