

AI ASSISTED CODING- LAB TEST : 03

NAME: MOHAMMED AREEBUDDIN

BATCH: 03 AIML

ROLL NUMBER: 2403A52054

SET: E-12

Q1)

TASK: Design and implement a solution using assistant tools to simplify API development for smart cities.

PROMPT: The API should accept a city related query and return on optimize solution generated by AI

CODE:

```
import express from "express";
import bodyParser from "body-parser";
import OpenAI from "openai";

const app = express();
app.use(bodyParser.json());

// AI config
const client = new OpenAI({
  apiKey: process.env.OPENAI_API_KEY,
});

// Endpoint: Healthcare API
app.post("/healthcare", async (req, res) => {
  try {
    const { symptom } = req.body;

    const response = await
    client.chat.completions.create({
      model: "gpt-5",
      messages: [
        { role: "system", content: "You are a medical assistant (not a doctor)."},
        { role: "user", content: `Suggest basic advice for: ${symptom}` }
      ],
    });

    res.json({
      input_symptom: symptom,
      ai_suggestion:
        response.choices[0].message.content,
    });

  } catch (error) {
    res.status(500).json({ error:
      error.message });
  }
});

app.listen(4000, () => console.log("✅ Healthcare API running on port 4000"));
```

OUTPUT:

```
{  
  "name": "smartcity-ai-api",  
  "version": "1.0.0",  
  "type": "module",  
  "dependencies": {  
    "express": "^4.18.2",  
    "openai": "^4.0.0",  
    "body-parser": "^1.20.2"  
  }  
}
```

OBSERVATION:By using AI assisted tools the API developmet becomes faster and easier as most of the codes was auto suggested and corrected. After the test confirmed the result was accurate and dynamic.

Q2)

TASK: Using AI tools design and implement on API to solve and backened data management challenge in health care domain

PROMPT: The API should take a health care related input and return an AI recommendation

CODE:

```
import express from "express";
import bodyParser from "body-parser";
import OpenAI from "openai";

const app = express();
app.use(bodyParser.json());

// AI config
const client = new OpenAI({
  apiKey: process.env.OPENAI_API_KEY,
});

// Endpoint: Healthcare API
app.post("/healthcare", async (req, res) => {
  try {
    const { symptom } = req.body;

    const response = await
client.chat.completions.create({
  model: "gpt-5",
  messages: [
    { role: "system", content: "You are a medical assistant+ 'not a doctor')." },
    { role: "user" ↓ content: `Suggest basic advice for: ${symptom}` }
  ]
});

    res.json(response.choices[0].message.content);
  } catch (error) {
    res.status(500).json({ error: "Internal Server Error" });
  }
});
```

```
});

res.json({
  input_symptom: symptom,
  ai_suggestion:
response.choices[0].message.content,
});

} catch (error) {
  res.status(500).json({ error:
error.message });
}
});

app.listen(4000, () => console.log("✅
Healthcare API running on port 4000"));
```

OUTPUT:

```
"input_symptom": "Fever and body
weakness",
"ai_suggestion": "Stay hydrated, take
adequate rest, monitor temperature
every 4 hours. If fever persists for more
than 48 hours, consult a doctor."
```

OBSERVATION: Using AI tools helped in generating the API structure faster and reducing coding effort. The rest results shows that the API responds correctly to health input and return useful recommendation.