AI-Powered SQL Query Generator

1. Introduction

Organizations often need to interact with databases to retrieve valuable insights. However, not all users are familiar with SQL. This project introduces an AI-powered agent that translates natural language queries into SQL, executes them on a PostgreSQL database, and returns the results in a structured format.

2. Features Implemented

2.1 Natural Language to SQL Conversion

Users input queries in natural language (e.g., "Show me all employees who joined after 2020.")

The system employs Google Gemini AI to generate corresponding SQL queries.

It handles synonyms and variations in phrasing to enhance flexibility.

2.2 Query Execution & Result Processing

The generated SQL query is executed on a PostgreSQL database. (Database contain ID, Name, Department, joining date and salary)

The results are displayed in a structured table.

2.3 Error Handling & Query Optimization

Detects and corrects invalid or ambiguous queries.

Optimizes SQL queries to enhance performance.

2.4 Database Support

Uses PostgreSQL as the database management system.

Demonstrates functionality with an employee records dataset.

2.5 User Interface

A web interface built using Flask facilitates interaction with the AI agent.

Results are displayed in an easily interpretable format.

3. Implementation Details

3.1 AI Model for NL-to-SQL Conversion

The model leverages Google Gemini AI to process natural language queries. A prompt is used to guide the AI in generating SQL statements, ensuring that it accurately interprets user input and converts it into valid PostgreSQL syntax.

3.2 Database Connection & Query Execution

The application connects to a PostgreSQL database using SQLAlchemy. Queries are executed securely, and results are formatted before being returned to the user.

3.3 Flask Web Interface

A Flask-based web application handles user interactions. The user enters a query, which is processed by the AI, converted into SQL, executed, and then displayed in a structured table format.

3.4 Frontend Web UI

A simple HTML interface allows users to enter queries, view generated SQL statements, and examine results.

3.5 Authentication & Security (Bonus):

For security purpose and not deployed on cloud I just replace my real gemini API key and Postgre credentials with default one with the massage of
#replace with your credentials
while deploying on the cloud we have alternated methods

4. Deployment & Usage

4.1 Setting Up the Environment

Install dependencies: pip install flask sqlalchemy psycopg2 google-generativeai
Start the Flask application: python app.py
Access the interface at http://127.0.0.1:5000/

5. Deliverables

Code Repository: Hosted on GitHub.
README File: Includes setup instructions, dependencies, and usage guidelines.

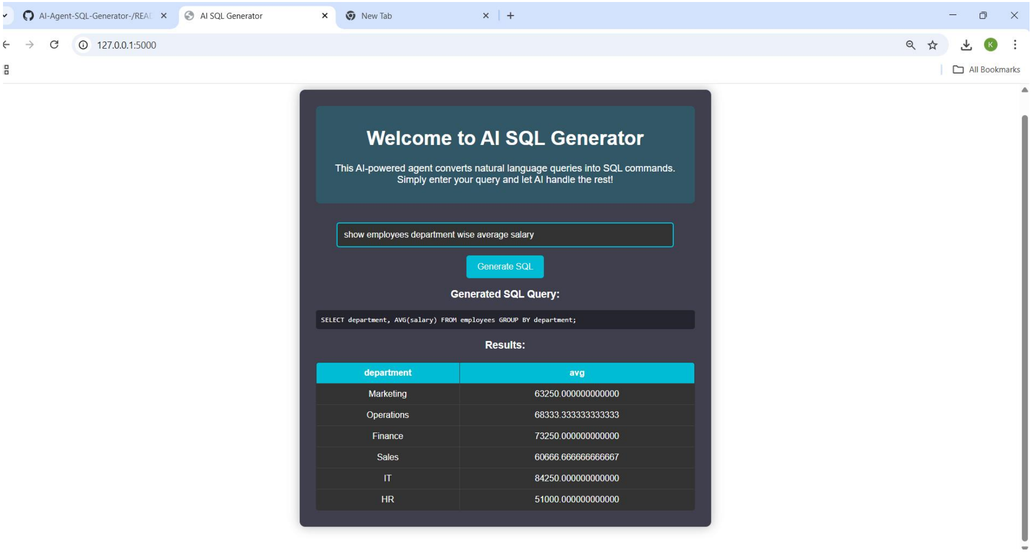
6. Conclusion

This AI-powered SQL generator simplifies database interactions by allowing users to query in natural language. It enhances accessibility, security, and usability, making data retrieval more efficient and user-friendly.

7. Web Interface Snippet

Here is snippet of my web interface which I locally hosted and get output

Q. 1) Show employees department wise average salary



Welcome to AI SQL Generator

This AI-powered agent converts natural language queries into SQL commands. Simply enter your query and let AI handle the rest!

show employees department wise average salary

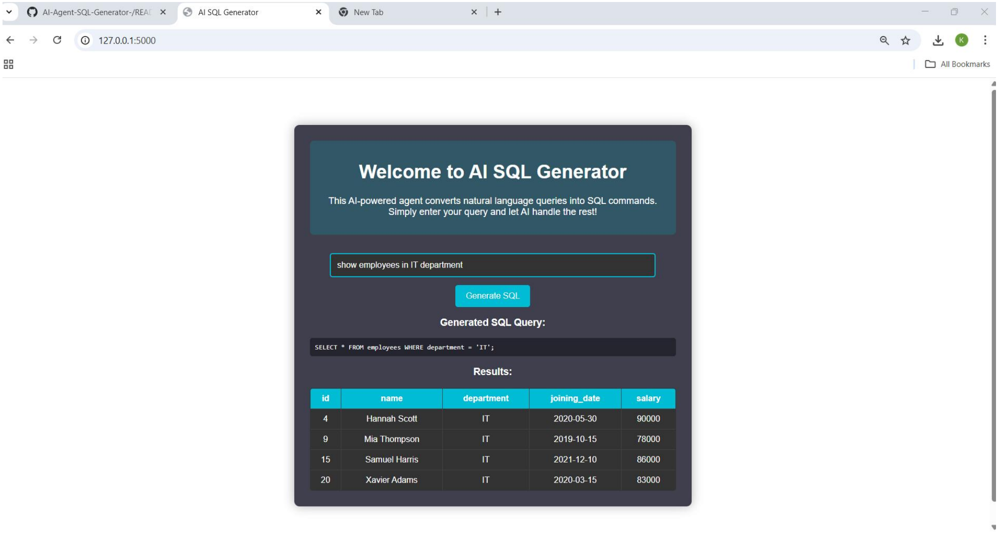
Generated SQL Query:

```
SELECT department, AVG(salary) FROM employees GROUP BY department;
```

Results:

department	avg
Marketing	63250.000000000000
Operations	68333.333333333333
Finance	73250.000000000000
Sales	60666.666666666667
IT	84250.000000000000
HR	51000.000000000000

Q.2) show employees in IT department



Welcome to AI SQL Generator

This AI-powered agent converts natural language queries into SQL commands. Simply enter your query and let AI handle the rest!

show employees in IT department

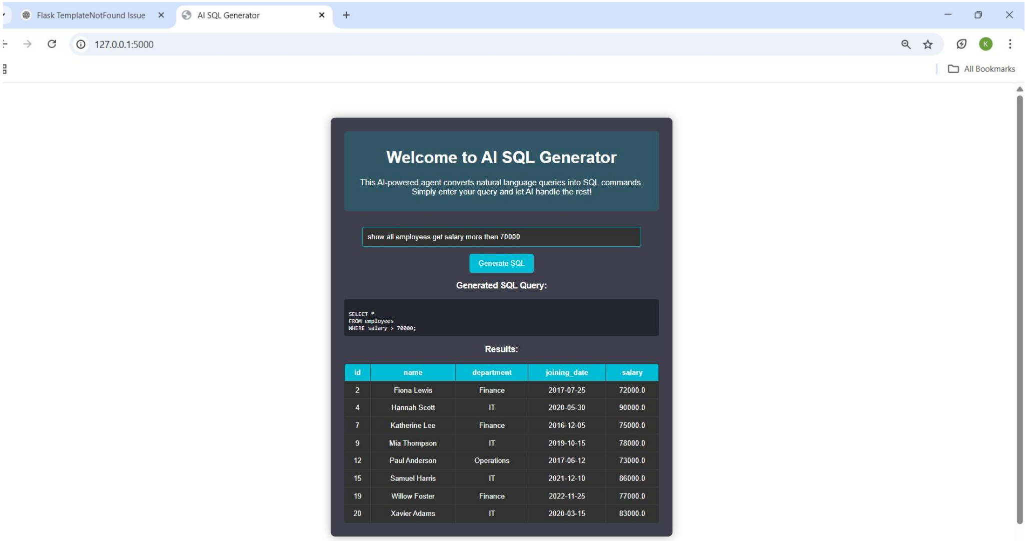
Generated SQL Query:

```
SELECT * FROM employees WHERE department = 'IT';
```

Results:

id	name	department	joining_date	salary
4	Hannah Scott	IT	2020-05-30	90000
9	Mia Thompson	IT	2019-10-15	78000
15	Samuel Harris	IT	2021-12-10	86000
20	Xavier Adams	IT	2020-03-15	83000

Q.3) show all employees get salary more then 70000



Welcome to AI SQL Generator

This AI-powered agent converts natural language queries into SQL commands. Simply enter your query and let AI handle the rest!

show all employees get salary more then 70000

Generated SQL Query:

```
SELECT *
FROM employees
WHERE salary > 70000;
```

Results:

id	name	department	joining_date	salary
2	Fiona Lewis	Finance	2017-07-25	72000.0
4	Hannah Scott	IT	2020-05-30	90000.0
7	Katherine Lee	Finance	2016-12-05	75000.0
9	Mia Thompson	IT	2019-10-15	78000.0
12	Paul Anderson	Operations	2017-06-12	73000.0
15	Samuel Harris	IT	2021-12-10	86000.0
19	Willow Foster	Finance	2022-11-25	77000.0
20	Xavier Adams	IT	2020-03-15	83000.0