



Computer Science Apprenticeship by the Faculty of Engineering

## ***ChatGISBot – Project Report***

**Supervised by : Dr.Eehab Hamzi Hijazi**

**Prepared by:**

- **Ne'meh Abu Issa 12114863**
- **Mayar Basheer 12112956**

## 1. Introduction

This project presents ChatGISBot, a QGIS plugin that enables users to perform spatial analysis on PostGIS databases using natural language queries.

The main goal of the project is to bridge the gap between GIS users and complex SQL/PostGIS syntax by allowing users to interact with spatial data through simple textual commands.

## 2. System Architecture

The system is composed of three main components:

QGIS Plugin (Frontend)

Provides a user-friendly interface inside QGIS.

Accepts natural language queries from the user.

Displays generated spatial results directly as map layers.

FastAPI Server (Middleware)

Receives user queries from the QGIS plugin.

Sends the queries to a local Large Language Model (LLM).

Returns valid PostGIS SQL queries to the plugin.

Local LLM via Ollama (Backend)

Processes natural language input.

Generates valid PostGIS SQL queries following predefined constraints.

Runs locally to ensure privacy and offline usability.

## 3. Workflow and Implementation Steps

### Step 1: Data Preparation

Spatial datasets were stored in a PostGIS database.

The project uses four spatial layers:

counties (polygons)

cities (points)

interstates (lines)

recareas (polygons)

All layers were connected directly to QGIS using the PostGIS provider.

### **Step 2:** Plugin Interaction

The user opens the ChatGISBot plugin from within QGIS.

A text input field allows the user to write spatial questions in natural language.

No SQL knowledge is required from the user.

### **Step 3:** Natural Language Processing

The entered query is sent to the FastAPI server.

The server forwards the query to the local LLM through Ollama.

The LLM generates a single valid PostGIS SQL statement that:

Creates a result table (analysis\_result)

Includes a unique identifier column

Returns exactly one geometry column

### **Step 4:** Query Execution

The generated SQL query is executed directly on the PostGIS database.

A new result layer is automatically added to the QGIS map.

Users can visualize, style, and explore the results immediately.

## **4. Demonstrated Spatial Queries**

During the demonstration, the following spatial analyses were successfully performed using natural language:

### **County and City Relationship**

Show counties that have cities within 10 km

Identifies counties that contain nearby cities based on distance.

### **City Proximity to Roads**

Show cities within 5 km of interstates

Finds cities located close to major road networks.

### **County and Interstate Intersection**

Show counties that intersect with interstates  
Detects counties crossed by interstate highways.

Recreation Areas Containment

Show recreation areas that are inside counties  
Identifies recreational areas located within county boundaries.

These examples demonstrate support for distance-based, intersection, and containment spatial operations.

## **5. Results and Visualization**

Each query produces a new spatial layer in QGIS.

Results are visualized immediately on the map.

Users can toggle layers, apply styles, and perform further GIS analysis.

## **6. Conclusion**

ChatGISBot demonstrates how natural language interfaces can simplify spatial analysis workflows.

By integrating QGIS, PostGIS, FastAPI, and a local LLM, the system allows users to perform advanced spatial queries without writing SQL, improving accessibility and usability for GIS practitioners.

## **7. Future Enhancements**

Automatic schema discovery

Support for more complex multi-layer queries

Query history and reuse

Performance optimizations for large datasets