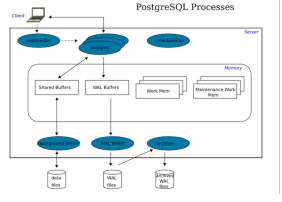
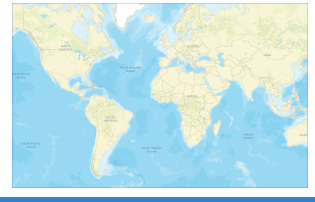
Geospatial Data Visualization and Query Interface

- SRUJAN KUMAR KONDI, DR. CHETAN TIWARI

INTRODUCTION

OpenLayers (OL) is employed to create interactive maps on the web. In our project, a map instance is created using OpenLayers with a default view centered at coordinates [0, 0] and zoom level 3. This map is then populated with two layers: a base layer from OpenStreetMap which is initially invisible, and an aerial imagery layer from Bing Maps which is visible. User interactions with the map, such as clicks, are handled to fetch geographic coordinates, which are then rounded off and used to query your PostgreSQL database via your backend server. Then using PostgreSQL, a robust open-source relational database, to manage geographic data related to storm events. This database is accessed through an Express.js server. The database connection parameters (host, database, user, and password) are defined in your server configuration. SQL queries are executed to fetch data based on latitude and longitude, which are passed from the frontend via an HTTP GET request.



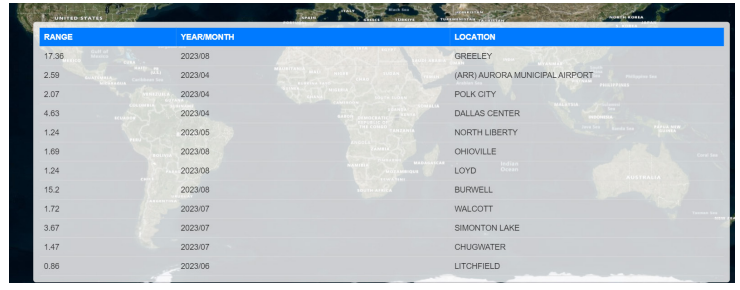


FEATURES AND CAPABILITIES

- **Interactive Map Navigation:** Utilizes OpenLayers for an immersive mapping experience, enabling users to easily navigate, zoom, and click on the map to query geospatial event data in real time.
- **Dynamic Visualization and Data Layers:** Offers dynamic data visualization on the map, including customizable layers and markers, to represent various geospatial events, powered by Bing Maps for detailed imagery.
- **Efficient Data Querying**: Features a responsive backend built with Express.js and PostgreSQL, designed for rapid querying and retrieval of geospatial event data based on user-selected locations.
- **User-Friendly and Secure:** Delivers a seamless and intuitive user experience across different devices, with a focus on data security and privacy, ensuring safe and reliable access to geospatial information
- **System Scalability and Reliability :** The system architecture supports scaling to handle increased load and concurrent users without performance degradation

RESULTS

The application displays query results in an intuitive format directly on the map interface, providing users with immediate access to the geospatial event data they've requested. This efficient presentation of information enhances user understanding and facilitates in-depth analysis of the data.



RANGE	YEAR/MONTH	LOCATION
17.36	2023/08	GREELEY
2.59	2023/04	(ARR) AURORA MUNICIPAL AIRPORT
2.07	2023/04	POLK CITY
4.63	2023/04	DALLAS CENTER
1.24	2023/05	NORTH LIBERTY
1.69	2023/08	OHIOVILLE
1.24	2023/08	LOYD
15.2	2023/08	BURWELL
1.72	2023/07	WALCOTT
3.67	2023/07	SIMONTON LAKE
1.47	2023/07	CHUGWATER
0.86	2023/06	LITCHFIELD

METHODOLOGY

1. Project Initiation and Requirements Gathering:
- Identified user needs for geospatial event visualization and interactivity.

2. System Design:
- Created a responsive web interface design using OpenLayers for map interactions and architected a RESTful API on the Express.js server for asynchronous data handling.

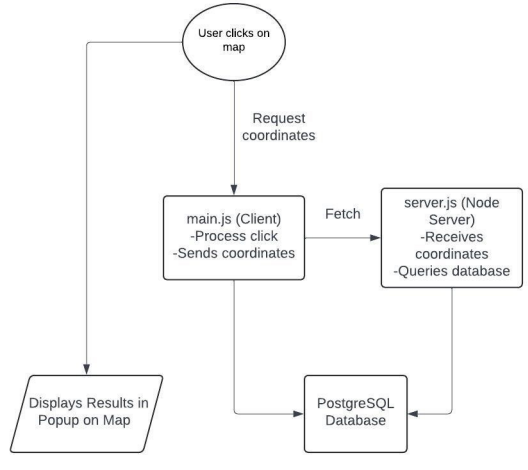
3. Development:
- Coded the frontend for user interactions, including map clicks that trigger data requests.
- Developed backend logic to receive coordinates, query the PostgreSQL database, and return results.

4.Integration:
- Integrated the interactive map with real-time data retrieval from the backend.

5. Testing:
- Tested frontend interactivity and backend response under various scenarios.
- Validated the accuracy and performance of geospatial queries and data display.

6. Deployment:
- Deployed the application on a server with considerations for scalability and security.

FLOW CHART



FUTURE ENHANCEMENTS

- **Error Handling:** Improve client-side error handling to more gracefully handle issues like network errors or invalid responses.
- **UI Enhancements**: Consider adding more interactive elements to the user interface, such as filters for the type of events displayed or a search box to jump to specific regions.
- **Performance**: For larger datasets, consider implementing server-side pagination or clustering on the map to improve load times and usability.

CONCLUSION

Our interactive mapping solution exemplifies a seamless blend of user-friendly design and technical robustness. It sets a foundation for sophisticated geospatial data interaction and paves the way for future enhancements that promise to deliver richer insights and a more personalized user experience, solidifying our commitment to innovation in geospatial data analysis.