DSCI 560 Lab 7 Report

Group Code Submission Link:
https://github.com/shubhamdarekar/DSCI560---Shubham/tree/main/Lab_7

Code README:
https://github.com/shubhamdarekar/DSCI560---Shubham/tree/main/Lab_6#readme

Meeting Notes Link:
https://docs.google.com/document/d/1cJB3y6v-ltWyneHjar6UFHC0CYkTvzhagKqhLWLSu48/edit?usp=sharing

Team Details: Group 1
Saavani Vaidya 9385579920
Yuxuan Liu 4780355176
Shubham Darekar 1641138809

1. Introduction

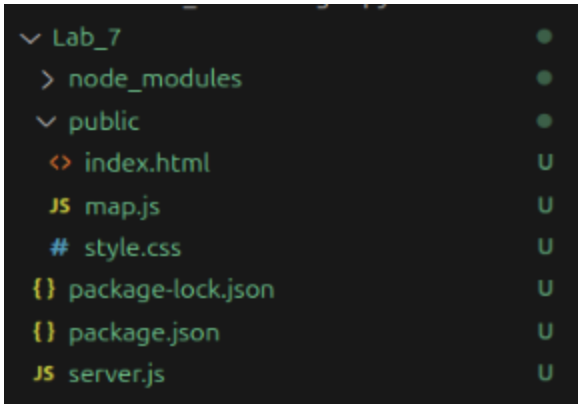
This report details the methodology and results of Lab 7, focusing on visualizing data related to oil wells. The objective is to present data wrangling results through interactive map layers, displaying well locations and associated information extracted from previous labs. This involves setting up a web server, integrating mapping libraries, and creating interactive elements to display well data.

2. Initial Setup

The implementation was carried out in a Linux environment using web-related tools, platforms, and a MySQL database. Key components included:

- Operating System: Linux
- Web Server: Apache (for hosting the webpage).
- Database: MySQL (for storing and retrieving well data).
- Mapping Libraries: OpenLayers

The lab emphasized exploring concepts and improvising solutions rather than spending excessive time on installation and setup.

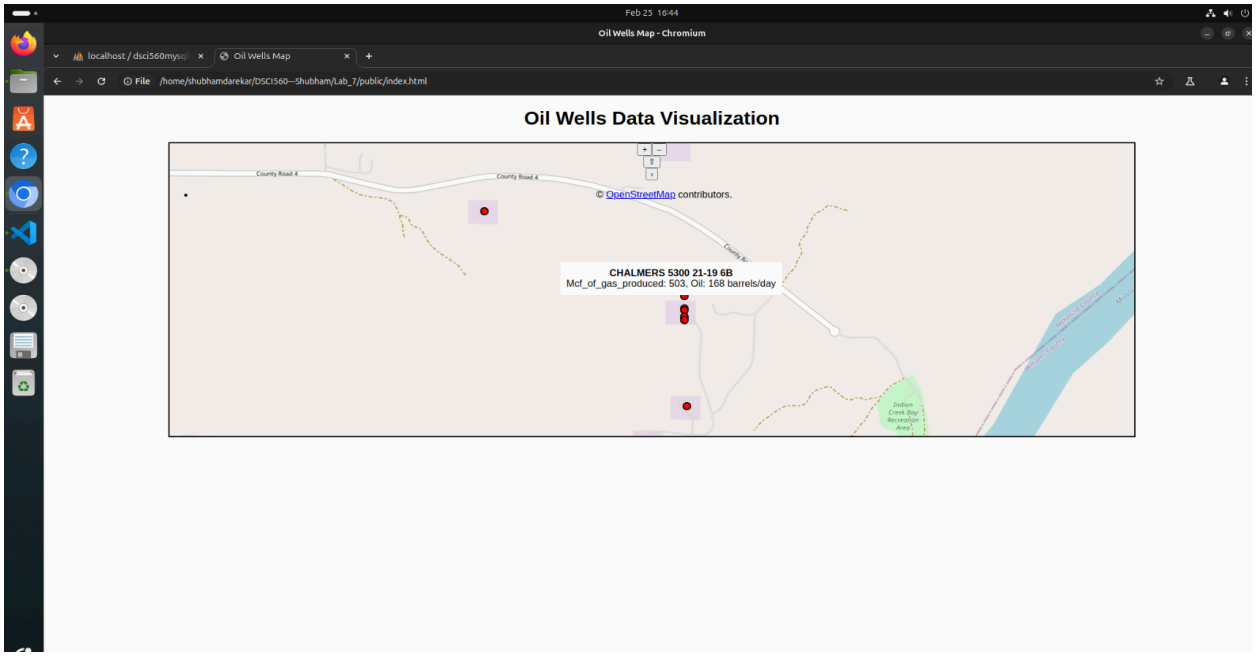


3. Webpage and Mapping Implementation

The core of this lab involved creating an interactive webpage with a map displaying oil well locations and associated data.

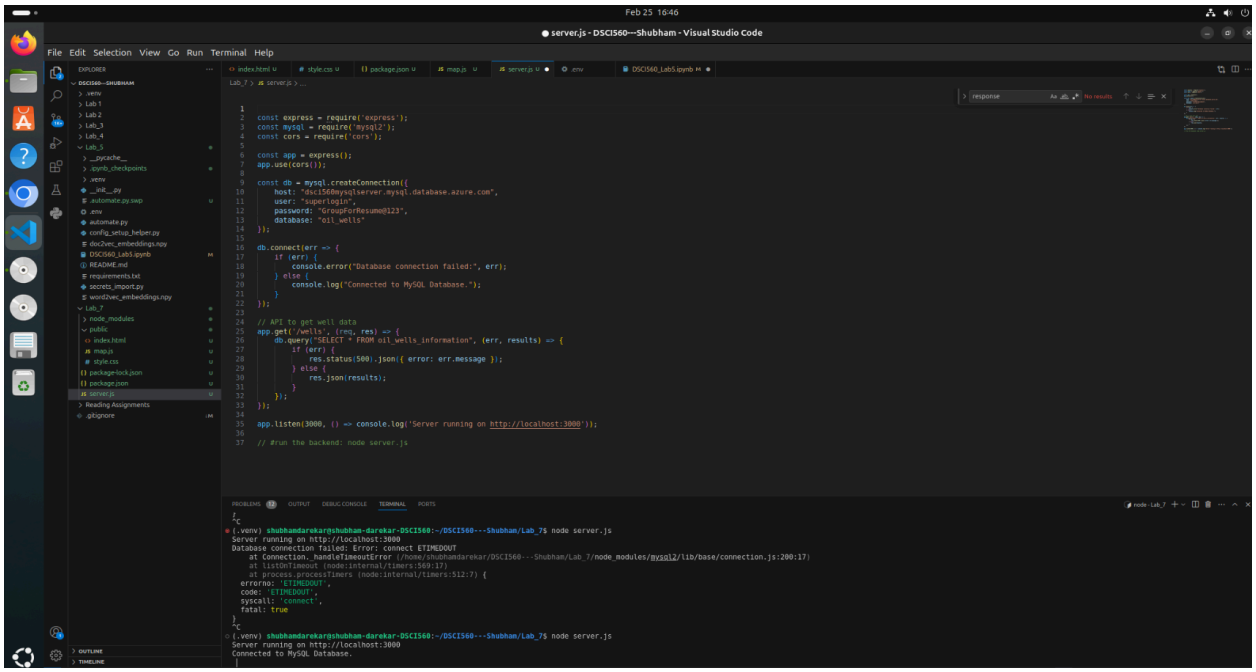
- Webpage Structure: Setting up an Apache web server and creating a webpage with a dedicated section for the map.
- Map Integration: Utilizing mapping libraries (OpenLayers, Leaflet) to create a base map and set the initial view.
- Push Pin (Marker) Implementation: Adding push pins to the map to represent well locations, using latitude and longitude data from the MySQL database.
- Interactive Popups: Implementing popups that appear when a push pin is clicked, displaying detailed information about the well (well-specific information, simulation data, and any additional data collected).

The implementation aimed to ensure that the map, push pins, and popups worked correctly, and that data was presented neatly and readably within the popups.



4. Data Storage and Retrieval

The Node.js code (dsci560_lab7.py) provides the backend logic for retrieving well data from the MySQL database and serving it as an API endpoint. The code uses the express framework to create the API, and the mysql2 library to connect to the database. The /wells endpoint retrieves all records from the wells table and returns them as a JSON response.



5. Testing and Results

- The backend API was tested using Postman and returned JSON responses successfully.
- The frontend map displayed correct well locations, and clicking on a well showed a popup with detailed information.
- The system worked correctly in Chrome, Firefox, and Edge.

6. Conclusion

This lab successfully focused on visualizing data extracted from previous labs. Through the implementation of interactive map layers, students gained experience in presenting data wrangling results using web-based tools and a MySQL database. This methodology can be applied to various data visualization tasks across different domains.

- Fetched real-time well data from a MySQL database.
- Displayed wells on an OpenLayers map.
- Enabled user interaction through popups.
- Utilized a modular backend and frontend architecture.
- This project demonstrates how Node.js, MySQL, and OpenLayers can be integrated to create powerful geospatial data visualizations.