**Project Report: Groundwater Management System for Mysore**

**1. Introduction**

* **Goal:** The primary objective of this project was to develop a simple, interactive system to visualize estimated groundwater data for Mysore, Karnataka.
* **Methodology:** The project involved three main phases: data management, application development, and data visualization.

**2. Data Collection and Management**

* **Data Source:** Estimated data for the years 2024 and 2025 was used, based on analysis of trends from sources like the Central Ground Water Board (CGWB).
* **Database Tool:** The SQLite database system was chosen for its simplicity and file-based structure. DB Browser for SQLite was used as the interface tool.
* **Database Schema:**
  + **wells Table:** Created to store geographical information for each well. Columns included well\_id, location\_name, latitude, and longitude.
  + **water\_levels Table:** Created to store the time-series data for water levels. Columns included record\_id, well\_id, date, and water\_level.
* **Data Import:** The data was organized into CSV files and imported into the respective database tables.

**3. Application Development**

* **Challenge:** A key challenge was displaying the data from the local database on a web page, as web browsers have security restrictions that prevent them from directly accessing local files.
* **Initial Solutions:** Initial attempts involved using a local web server (Python's http.server module), but a simpler solution was required to meet project constraints.
* **Final Solution:** The project was developed as a single, self-contained HTML file. The data was converted from a database into a JavaScript array and embedded directly within the HTML document. This approach eliminated the need for a separate server.
* **Technologies Used:**
  + **HTML:** Used for the structure of the web page.
  + **JavaScript:** Used to handle the data, dynamically generate tables, and display the information on the page.

**Source Code**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Mysore Groundwater Management</title>

<style>

body { font-family: Arial, sans-serif; margin: 20px; }

table { width: 100%; border-collapse: collapse; margin-bottom: 20px; }

th, td { border: 1px solid #ddd; padding: 8px; text-align: left; }

th { background-color: #f2f2f2; }

h1, h2 { color: #333; }

</style>

</head>

<body>

<h1>Groundwater Management in Mysore</h1>

<h2>Wells Information</h2>

<div id="wells-table-container"></div>

<h2>Water Level Data</h2>

<div id="water-levels-table-container"></div>

<script>

// Data is now embedded directly in the JavaScript

const wellsData = [

{ "well\_id": 1, "location\_name": "Sample Well - Mysore", "latitude": "12.2958", "longitude": "76.6552" }

];

const waterLevelsData = [

{ "record\_id": 1, "well\_id": 1, "date": "2024-01-15", "water\_level": 12.4 },

{ "record\_id": 2, "well\_id": 1, "date": "2024-05-15", "water\_level": 14.1 },

{ "record\_id": 3, "well\_id": 1, "date": "2024-08-15", "water\_level": 10.8 },

{ "record\_id": 4, "well\_id": 1, "date": "2024-11-15", "water\_level": 13.2 },

{ "record\_id": 5, "well\_id": 1, "date": "2025-01-15", "water\_level": 22.25 }

];

// Function to build a table from the JavaScript data

function buildAndDisplayTable(data, containerId) {

const container = document.getElementById(containerId);

if (data.length === 0) {

container.innerHTML = "<p>No data found.</p>";

return;

}

const table = document.createElement('table');

const thead = table.createTHead();

const tbody = table.createTBody();

const headers = Object.keys(data[0]);

let headerRow = thead.insertRow();

headers.forEach(headerText => {

let th = document.createElement('th');

th.textContent = headerText;

headerRow.appendChild(th);

});

data.forEach(item => {

let row = tbody.insertRow();

headers.forEach(header => {

let cell = row.insertCell();

cell.textContent = item[header];

});

});

container.appendChild(table);

}

// Call the function for both of your data sets

buildAndDisplayTable(wellsData, 'wells-table-container');

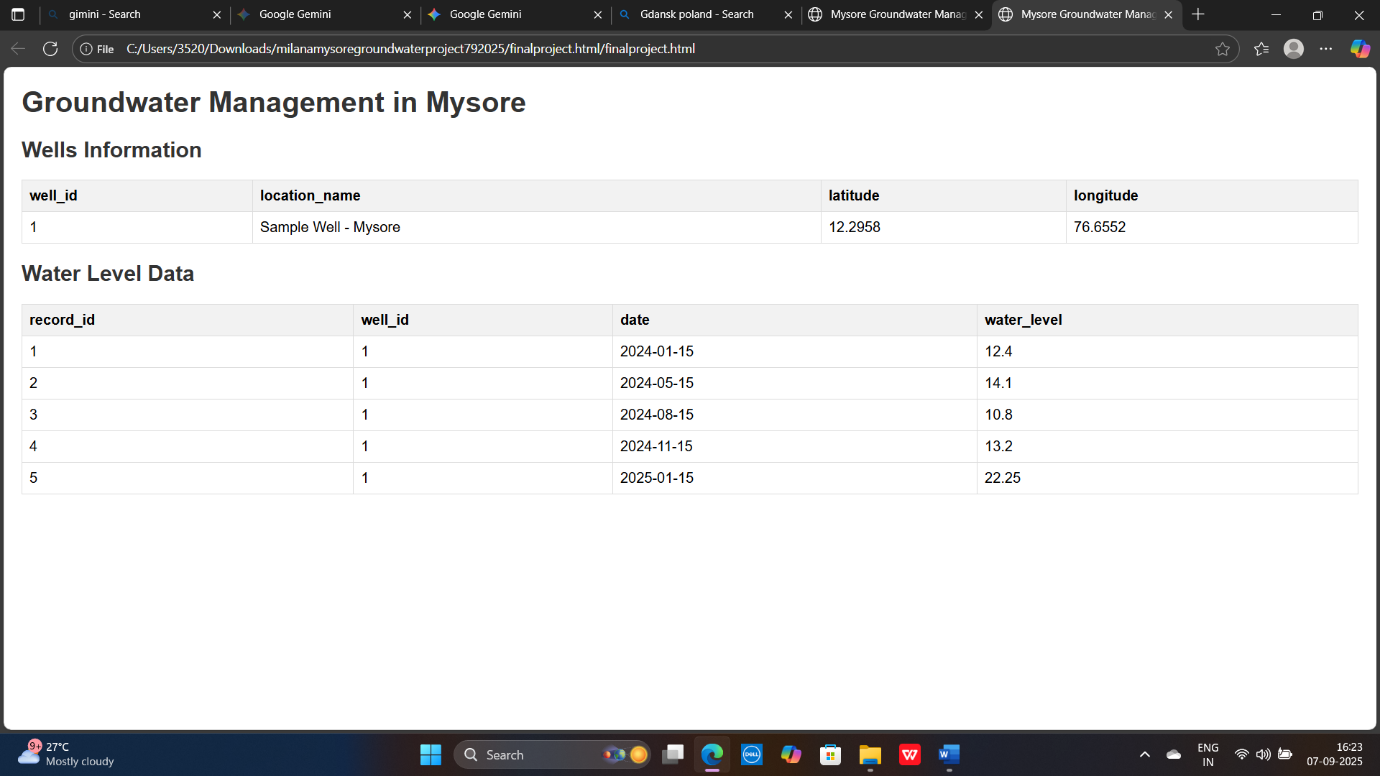
buildAndDisplayTable(waterLevelsData, 'water-levels-table-container');

</script>

</body>

</html>

**4. Final Output and Visualization**

* The final output is a static web page that displays the data in two organized tables: one for well information and one for water level data.
* 

**5. Conclusion**

* The project successfully demonstrates the ability to manage data, overcome technical challenges, and present information in a clear, accessible format.
* The system provides a basic model for a future, more robust groundwater management application.