Project Report - EcoVista: Interactive Environmental Insights Map
Team members: ZiHan Li, YiXuan Li, YaTing Pai, Xuanming Zhang

1. **Please list out changes in the directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).**
   The purpose of the project remained consistent with the original proposal, which was to develop an environmental platform for users to access environmental data and compare specific environmental indicators for different states. While some functions were adjusted to improve performance and user experience, these changes did not alter the project's overall goals. The final project aligns closely with the original vision.

2. **Discuss what you think your application achieved or failed to achieve regarding its usefulness.**
   The application achieved its primary goal of providing users with a platform to visualize and analyze environmental data through an interactive map interface. By integrating diverse datasets such as air quality, drought levels, and weather data, the platform made complex information accessible to researchers, policymakers, and the general public. Its query and ranking features enhanced user engagement and supported informed decision-making. However, the application's utility could be further improved by implementing features for real-time data updates, which were beyond the project's initial scope. Despite this, EcoVista fulfilled its purpose of simplifying the understanding of environmental conditions.

3. **Discuss if you change the schema or source of the data for your application**
   The data sources were not changed. The data schema for the EcoVista application remained largely consistent with the original plan. However, minor adjustments were made to the schema to support application features. For example, additional fields were added to UserProfile to enable Last Login feature. Additionally, a new table, LoginHistory, was introduced to implement this feature.

4. **Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?**

We added a timestamp field in the user_profile table and a new UserLoginHistory table to store a historical record of users' login times. These changes allow us to track user engagement metrics and analyze user behavior effectively.

The difference is that: In the original design, we lacked a mechanism to capture user activity, making it difficult to analyze engagement trends. The final design addressed this by adding fields and tables to store login data, which improves the system's capability to monitor user metrics. This addition also enables the implementation of triggers, satisfying the project's technical requirements while enhancing the application's functionality.

The final design is more suitable because it not only supports the implementation of triggers but also adds practical value by enabling user engagement analysis. This aligns with the needs of commercial applications, where tracking and improving user metrics is critical for success. By integrating these features, the design achieves both technical and business objectives.

5. **Discuss what functionalities you added or removed. Why?**

   1. Functionalities Added.

   - **Worsened States Search**:

     A new feature was implemented to identify states with worsening weather scores compared to the previous month. This feature is critical for users who need to monitor environmental trends and assess the impact of changing weather conditions.

   - **User Notifications**:

     To improve user engagement, we introduced a notification system. This feature alerts users if the climate score in their region falls below a predefined threshold, helping them stay informed about adverse environmental conditions.

   2. Functionalities Removed

   - **Color-Coded Representation of Weather Conditions**:

Originally, we planned to display color-coded weather conditions for different regions based on their scores. Due to time constraints and the prioritization of higher-impact features, this functionality was removed.

6. **Explain how you think your advanced database programs complement your application.**

In our database, we manage four primary weather data tables along with a user_profile table for user management, a UserLoginHistory for tracking user activity, and a location table that maps county codes to state and county names. When retrieving specific weather data for a particular location, advanced queries are frequently used to efficiently index and retrieve data.

One of the key features of our application is the state ranking, which relies on aggregate functions to calculate and rank weather-related scores. To ensure data consistency during read and write operations, we leverage transactions to protect the integrity of the database.

For updating user login times, we implemented a trigger to maintain synchronization between the user_profile table and the UserLoginHistory. Whenever a user's login time is updated in the user_profile, the corresponding entry in the UserLoginHistory is automatically updated to reflect the change.

Additionally, for the worsened state tracking feature, we utilize a stored procedure to streamline the process. The procedure simplifies the comparison of weather scores across months and efficiently identifies states with deteriorating conditions.

7. **Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.**

   - Yixuan Li: Dependency Problem in the integration of a front-end framework.

One challenge we faced was integrating a visually appealing but older open-source React front-end into our project. The outdated dependencies caused persistent version conflicts in Node.js packages, disrupting development. To resolve this, we aligned all package versions and ultimately ported a working node_modules folder from a teammate's environment. This highlighted the importance of dependency management and version consistency in development.

- Ya-Ting Pai: Data Tabs and User Interface Components

One challenge for us was designing a user-friendly interface that could dynamically update based on user inputs while maintaining consistency across different components. To address this, we utilized a React template, which provided a structured framework for implementing interactive features efficiently and ensuring a smooth user experience.

- Zihan Li: Environmental Scoring and Trend Analysis

To calculate fair and representative environmental scores, I normalized all metrics to a common scale to address range discrepancies and applied weighted scoring based on their importance. Missing data was handled through interpolation, exclusion, or substitution with regional or historical averages. Trends were analyzed using linear regression to determine improvement or deterioration. This approach ensured balanced scores, minimized bias from missing data, and provided reliable insights for subsequent operations.

- Xuanming Zhang: Cookie management and user session control

One of the challenges that we met during developing our web app is implementing user authentication for user profiles. Cookies are created by the backend and sent to the frontend. There was a conflict with frontend which backend cannot delete the cookie successfully when user logout. To solve this problem, we decided to delete cookies in frontend which will have little effect on security. This method ensures the accuracy of user authentication and protects user information.

8. **Are there other things that changed comparing the final application with the original proposal?**
    - **Data Visualization**: The final application differs significantly from the original proposal in its approach to data visualization and interactivity. Instead of implementing a dynamic, color-coded map that overlays real-time environmental data (e.g., air pollution and drought severity) and visually distinguishes conditions through shading, the final application focuses on displaying quarterly summaries of four datasets (AQI, drought, NO2, CO) when a user clicks on a state. This change reflects a move

away from real-time, visually dynamic interactivity toward a more static, aggregated presentation of data for improved simplicity and feasibility.

- **Query Features:** The query functionality in the final application has been enhanced compared to the original proposal. In addition to the planned functionality, we have added the ability to search and retrieve specific datasets (e.g., AQI, drought, NO2, or CO) by county name. These enhancements make the query functionality more versatile and user-friendly, allowing users to access targeted information efficiently.

9. **Describe future work that you think, other than the interface, that the application can improve on**
   - **Integration of Real-Time Data Sources**
     Incorporate real-time environmental data feeds to provide up-to-date insights for users. This could enhance the application's relevance and usability for decision-making.
   - **Advanced Data Analytics and Predictions**
     Implement machine learning models to predict future environmental conditions based on historical trends, offering proactive insights for users.
   - **Customized Alerts and Notifications**
     Allow users to set up personalized alerts for significant changes in specific metrics (e.g., a sudden drop in the AQI or worsening drought conditions). And send alerts via email based on user preferences

10. **Describe the final division of labor and how well you managed teamwork.**

- **Division of labor**：
  Yating Pai:
  - Implemented the interactive map's frontend, ensuring it was functional and visually aligned with the rest of the application.
  - Developed backend functionalities to process user queries and return results.
  - Created and tested a trigger to log user login times in the database.

  ZiHan Li:
  - data preprocessing, including cleaning and structuring the datasets.
  - Focused on backend response logic, ensuring accurate and efficient query results were returned to the frontend.

- Designed and implemented the transaction mechanism for managing notifications.

YiXuan Li:

- Collaborated on developing frontend pages and user interactions using React.
- Worked on integrating the interactive map's backend functionality, ensuring smooth communication with the server.
- Implemented a stored procedure for identifying worsening environmental conditions.

Xuanming Zhang

- Collaborated on developing frontend pages and user interactions using React.
- Integrated frontend features with backend API calls to display query results.
- Contributed to implementing a stored procedure for advanced database operations.

- **Teamwork:**

  Xuanming Zhang and YiXuan Li worked closely on frontend development and database stored procedures, while Yating Pai and ZiHan Li handled backend response logic and database functionalities like triggers and transactions. The work on the interactive map was shared, with Yating Pai developing the frontend and YiXuan Li managing the backend.

  Regular team meetings and task updates ensured smooth collaboration, allowing frontend and backend components to integrate effectively. We used GitHub extensively to manage version control, track individual contributions, and handle integration conflicts. Each member maintained dedicated branches for their tasks, which were merged after reviews and conflict resolution.