Full Stack Assignment-01: Building a Geo-Data App

**Introduction to our company:** You can check out our page: <https://skyserve.ai/> and <https://www.linkedin.com/company/skyserve-ai/>

**Assignment Overview:** You are tasked with building a fullstack web application that allows users to manage and visualize geospatial data. The application consists of a backend written in Go and a frontend built with React/Next.js. Users should be able to create an account, upload GeoJSON or KML files, render these files on a map using React Leaflet, draw custom shapes on the map, save those shapes, and edit them.

**Technical Requirements:**

1. **Frontend:** 
   1. The frontend should be developed using React or Next.js.
   2. Implement user authentication and user account management.
   3. Create a user-friendly interface for uploading and managing GeoJSON and KML files.
   4. Integrate the React Leaflet library to render maps and uploaded GeoJSON/KML files.
   5. Implement a map view where users can draw custom shapes on the map using a tool like Leaflet.
   6. Allow users to save the drawn shapes to their account.
   7. Provide functionality to edit existing shapes, including modifying their geometry and attributes.
2. Backend:
   1. The backend should be developed using the Go programming language.
   2. Implement RESTful APIs for user authentication, account creation, and management.
   3. Create endpoints for uploading and managing GeoJSON and KML files.
   4. Implement a database (e.g., PostgreSQL or SQLite) to store user data and geospatial information.
   5. Develop API endpoints for creating, updating, and retrieving geospatial data.
   6. Ensure user data security and access control.

**Additional Requirements:**

1. Use Git for version control and maintain a repository on a platform like GitHub.
2. Write clear and well-documented code with comments.
3. Implement error handling and validation to provide a smooth user experience.
4. Use appropriate libraries and tools for geospatial data processing and visualization.
5. Ensure responsiveness and user-friendly design.
6. Write a brief README that includes instructions on how to set up and run the project.

**Evaluation Criteria:**

1. Functionality: Does the application meet the specified requirements, allowing users to manage geospatial data effectively?
2. Code Quality: Is the code well-structured, maintainable, and well-documented?
3. Security: Are user accounts and geospatial data adequately protected?
4. Creativity: Did the developer go beyond the basic requirements to add extra features or improvements?

**Submission:** Share a link to your Git repository containing the source code and any additional documentation or instructions needed to evaluate the project. The repository should be public so that we can evaluate.

**Time to deliver the assignment:** 48 hours after receiving the email.