

Technical Test: Framework Next.js

Repository: GeoData Management System (Next.js)

- **Client-Side Data Upload Component:**

- Task: Develop a client-side component using Next.js that allows users to upload geojson files.
- Details: Create a user-friendly interface for uploading files, with validation checks to ensure the selected files are in the correct format before submission.

- **Data Processing Visualization:**

- Task: Implement a data visualization component in the client-side using Next.js to display processed spatial data.
- Details: Design and develop a visualization interface to display the processed data on maps or other graphical representations.

- **Unit Testing with Jest:**

- Task: Write unit tests using Jest to verify the functionality of key endpoints and middleware within the application.
- Details: Write tests to cover various scenarios, including valid and invalid inputs, edge cases, and error handling. Test both happy paths and error conditions to ensure robustness.

- **Deployment and CI/CD Pipeline :**

- Task: Set up a deployment pipeline using platforms like Vercel or HEROKU - Details: Configure continuous integration and continuous deployment (CI/CD) pipelines to automatically deploy changes to production or staging environments

SERVER SIDE COMPONENT (BIG PLUS)

Task List:

- **Seeding and Migration:**

- Task: Implement seeding and migration functionality to initialize and update the database schema.
- Details: Write scripts or commands to seed initial data into the database and perform database migrations when schema changes occur. Ensure data integrity and consistency during migration processes.

- **Implement Data Reader Endpoint:**

- Task: Develop an API route using Next.js that allows users to upload geojson

files.

- Details: Ensure the API route can receive and validate the uploaded data. Implement validation checks to ensure the uploaded files are in the correct format and contain valid geojson data.

- **Authentication and Authorization:**

- Task: Implement authentication and authorization mechanisms using Next.js API routes middleware.
- Details: Secure API routes requiring authentication and authorization using middleware. Ensure only authorized users can access sensitive API routes or perform certain actions.

- **User Input Validation:**

- Task: Implement validation middleware in Next.js API routes to ensure that data uploaded by users is valid and conforms to the expected format.
- Details: Write custom validation logic to validate incoming requests containing data. Provide informative error responses if validation fails, including details on which fields are incorrect.

- **Development of Data Processing Endpoint:**

- Task: Create an API route in Next.js responsible for processing spatial data uploaded into the system.
- Details: Implement logic to efficiently process uploaded data and store it in the database. Ensure the API route can handle large datasets and provide responses indicating the success or failure of the processing operation.

- **Unit Testing with Jest:**

- Task: Write unit tests using Jest to verify the functionality of key API routes and middleware within the application.
- Details: Write tests to cover various scenarios, including valid and invalid inputs, edge cases, and error handling. Test both happy paths and error conditions to ensure robustness.

- **Design and Implementation of API Documentation:**

- Task: Document the API routes using tools like Swagger or Next.js API routes documentation to provide detailed documentation for developers.
- Details: Generate documentation for all API routes, including descriptions, request/response formats, and example usage. Ensure the documentation is up-to-date with the latest changes in the API.

Note: **The results of the technical test are submitted in the format of a github link.**