



Initial challenge

Next.js Phrase Generator Challenge

Objective

Create a Next.js application that generates phrases by combining subjects and predicates while applying conditional logic. The application must include a user interface to display the results dynamically.

Instructions

1. **Set up your project:**
 - a. Create a new Next.js project using create-next-app.
 - b. Configure TailwindCSS for styling.
2. **Define the data:**
 - a. In the backend, define two arrays:
 - i. Subjects: ["The dog", "The turtle", "My friend", "Sebastian"]
 - ii. Predicates: ["runs fast", "is very wise", "loves coding", "sings poorly"]
3. **Implement conditional logic:**
 - a. If the subject has more than 8 characters, append an adjective such as "intelligent".
 - b. If the predicate includes the word "coding", add an exclamation mark (!) at the end.
 - c. If none of the conditions are met, simply concatenate the subject and predicate.
4. **Break down the logic:**
 - a. Create a method to randomly select a subject.
 - b. Create another method to randomly select a predicate.
 - c. Combine and apply the rules to generate a phrase.
5. **Design the frontend:**
 - a. Build a user interface where users can:
 - i. Click a button to generate a new phrase.
 - ii. See the generated phrase displayed in real-time.
6. **Backend connection:**



- a. Use a Next.js API route to handle the logic and return the generated phrase to the frontend.

Deliverables

1. **Project Repository:**
 - a. Push your completed project to a public GitHub repository.
 - b. Include a clear README with setup instructions.
2. **Demonstration:**
 - a. Ensure the application runs without errors and generates phrases dynamically.
 - b. Include screenshots or a short video of the application in action (optional).

Expected Features

- **Dynamic phrase generation:**
 - The application must apply the conditional logic described above.
- **Modern UI:**
 - Use TailwindCSS to create a clean and visually appealing interface.
- **Separation of concerns:**
 - Logic should be implemented in the backend API route.
 - The frontend should focus on displaying the result and user interaction.

Additional Challenges (Optional)

- Add more subjects and predicates to the arrays.
- Allow users to input custom subjects or predicates via the interface.
- Enhance the UI with animations or advanced styles using TailwindCSS.

Good luck, and have fun creating your Next.js phrase generator!