Project Documentation: Interactive Cold War Map

Essential Questions

1. What is the problem or question to solve?

The project aims to create an **Interactive Cold War Map** that combines historical storytelling with modern web technology. The goal is to provide an immersive user experience featuring Three.js animations, Tailwind CSS styling, and interactive sections that showcase the journey of building the map and the inspirations behind it.

2. Why is this a relevant problem or question?

The Cold War is a crucial historical period that shaped modern geopolitics, yet it's often overlooked by younger generations. By creating an engaging and interactive experience, the project helps educate users in an accessible and visually appealing way.

3. How are you going to solve this?

The solution involves:

- Developing the project using **React** for modularity and efficiency.
- Incorporating **Three.js** for animated starry backgrounds.
- Using **Tailwind CSS** for responsive and modern styling.
- Structuring the project into reusable components for maintainability.
- Leveraging AI tools to assist in debugging, generating content, and improving workflows.

Journey of the Project

The development of this project was both challenging and enlightening. Here's how it unfolded step by step:

Initial Setup

The first step was to set up the foundational environment for the project. This involved creating a React application and installing necessary dependencies:

```
npx create-react-app cold-war-map
cd cold-war-map
npm install three tailwindcss postcss autoprefixer lucide-react react react-dom
npx tailwindcss init -p
```

Configurations

Configured `tailwind.config.js` to include all component paths:

```
 module.exports = {
  content: ["./src/**/*.{js,ts,jsx,tsx}"],
  theme: {
    extend: {},
  },
  plugins: [],
};
```

-

Imported Tailwind directives in `src/styles/index.css`:

```
 @tailwind base;
@tailwind components;
@tailwind utilities;
```

-

Struggle: Initially, the `@tailwind` directives were not recognized. **Solution**: With the help of AI tools, I diagnosed the issue and confirmed that the PostCSS configuration was missing. Adding this resolved the problem:

```
module.exports = {
  plugins: {
    tailwindcss: {},
    autoprefixer: {},
  },
};
```

Component Development

StarCanvas Component

This component was central to the project, creating an animated starry background using **Three.js**. I leveraged AI to refine the Three.js implementation and debug canvas rendering issues.

Implementation Highlights:

- Created a starry effect using Three.js geometry and particle systems.

Ensured the canvas covered the viewport using CSS:

```
 canvas {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  z-index: -1;
}
```

-

Struggle: A persistent white space overlayed the canvas. **Solution**: AI-assisted debugging helped identify CSS conflicts, and I corrected z-index settings.

NavigationBar Component

The navigation bar included a dark mode toggle, using **lucide-react** icons for visual feedback.

Dark mode toggle:

```
 <button onClick={toggleDarkMode}>
  {darkMode ? <SunIcon /> : <MoonIcon />}
</button>
```

-

Struggle: Implementing dynamic dark mode classes. **Solution**: AI suggested integrating Tailwind's class manipulation and React state effectively.

HeroSection Component

This section introduced the project, combining Tailwind for styling and React for structure:

```
<section id="hero">
  <h1 className="text-4xl font-bold text-cyan-400">A Journey Through Time</h1>
  <p className="text-lg">Welcome to my creative journey...</p>
</section>
```

Struggle: Centering elements dynamically. **Solution**: AI tools recommended using Flexbox utilities such as `flex`, `justify-center`, and `items-center`.

Phases and Inspirations

JourneySection Component

This section showcased the project's development phases:

- Used props to dynamically render `phases` data.

Example data:

```
 const phases = [
  {
    title: 'The Spark',
    description: 'It all began with a passion...',
    image: '/assets/images/spark.jpg',
    date: 'September 2023',
  },
];
```

-

Struggle: Properly aligning phase elements. **Solution**: AI debugging recommended using Tailwind utilities and responsive design practices.

InspirationSection Component

Highlighted inspirations with placeholder images and PDF links:

```
const inspirations = [
  {
    title: 'NATO Interactive Map',
    description: 'Studying NATO’s approach...',
    document: '/assets/pdfs/nato-analysis.pdf',
  },
];
```

Struggle: Linking PDF files dynamically. **Solution**: AI-assisted documentation guided me in organizing assets in the `public/` directory.

AI's Role in the Project

AI was instrumental in:

- Debugging Tailwind and Three.js configuration issues.
- Suggesting responsive design solutions for complex components.
- Refining the directory structure to enhance maintainability.
- Offering insights on resolving dependency issues.

Final Structure

```
project/
├── public/
│   ├── assets/
│       ├── images/
│       ├── pdfs/
├── src/
│   ├── components/
│       ├── StarCanvas.jsx
│       ├── NavigationBar.jsx
│       ├── HeroSection.jsx
│       ├── JourneySection.jsx
│       ├── InspirationSection.jsx
│       ├── App.jsx
│   ├── styles/
│       ├── index.css
│   ├── index.js
├── package.json
├── tailwind.config.js
└── postcss.config.js
```

Results

- **Working Features**:
    - Fully functional starry background using Three.js.
    - Interactive dark/light mode toggle.
    - Display of development phases and inspirations.
- **Challenges Overcome**:
    - Resolved Tailwind CSS setup and z-index issues with AI guidance.
    - Debugged Three.js rendering problems.

- Addressed missing dependencies effectively.

Next Steps

1. **Add Real Assets**:
    - Replace placeholder images and paths with actual project assets.
2. **Optimize Responsiveness**:
    - Test and improve layout for different screen sizes.
3. **Deploy Project**:
    - Use Vercel, Netlify, or GitHub Pages for deployment.
4. **Iterate Based on Feedback**:
    - Gather feedback and refine the design and functionality.

Current Status

Despite overcoming numerous challenges and implementing key features, the project is **not yet fully functional** due to unresolved alignment issues and incomplete asset integration. The current setup serves as a **default structure for all project pages** in my portfolio, providing a foundational template for future enhancements and projects.
