**Tech Stack Overview**

[ForestGPT](https://forest-chat-gpt-jvalenzano.replit.app/auth)

The technical stack in your project is set up primarily for a React-based web application with a back-end service. Here’s a breakdown of the components:

**Frontend**

1. **React**: The application uses React for building the user interface. Components like ChartContainer, ChartStyle, and ChatInterface.tsx are examples where React's functional components and hooks (like useId) are utilized.
2. **Recharts**: The ChartContainer integrates the Recharts library for data visualization, which is evident from its use of ResponsiveContainer and other chart-related components.
3. **CSS Tailwind**: CSS-in-JS styling is applied using Tailwind CSS, which is apparent from class names like bg-emerald-200 and rounded-2xl.
4. **TypeScript**: The .tsx file extensions indicate TypeScript usage, allowing for type-checking capabilities in the codebase.

**Backend**

1. **Node.js**: The project is configured to use Node.js as indicated by the nodejs-20 in the .replit file. The back-end service likely handles data processing and business logic.
2. **API Request Handling**: The apiRequest function in the queryClient.ts file suggests a RESTful API approach for communication between the frontend and backend.
3. **Web Scraping**: The scraper service in server/services/scraper.ts indicates functionality for web scraping, fetching content relevant to the US Forest Service.

**Replit Configuration**

* **Run Command**: The project is set to run with npm run dev, which typically starts a development server. This ensures that both the frontend and backend services can be executed in parallel during development.
* **Port Forwarding**: The app is configured to use port 5000 for development, which is mapped to the external ports 80 and 443 in deployment, allowing for HTTP and HTTPS connections.

**Project Structure:**

| **client/  ├── src/  │ ├── components/  │ │ ├── ChatInterface.tsx  │ │ └── ui/  │ │ ├── chart.tsx  │ │ ├── context-menu.tsx  │ │ └── menubar.tsx  │ └── lib/  │ └── queryClient.ts  ├── index.css  └── index.html server/  └── services/  └── scraper.ts .replit** |
| --- |

This stack effectively combines modern front-end development with a robust back-end service, all orchestrated within the Replit environment for ease of development and deployment.