Welcome to our **Software Engineer** coding challenge!

In this challenge, you will create a web application that utilises NASA's Open APIs (https://api.nasa.gov/) to showcase space-related data. Your task is to build a **React frontend** that communicates with a **Node.js backend** running Express. The application should allow users to explore and interact with NASA's vast array of space data in a creative and engaging way.

Challenge Overview:

Your task is divided into three main parts:

1. **React Frontend**: Create a user interface that allows users to interact with NASA data. The design and functionality are up to you – be creative! Your frontend should make requests to your backend server to fetch data from the NASA API.
2. **Node.js Backend**: Develop a server using Node.js and Express that acts as an intermediary between your frontend and the NASA API. Your server should handle requests from the frontend, make appropriate calls to the NASA API, and return the data to your frontend.
3. **Data Visualization**: Process the information received from your backend and present it to the user in an intuitive and visually appealing manner. Consider using charts, graphs, images, or any other means of data visualisation that enhances the user experience.

NASA API:

You will be working with NASA's Open APIs, available at https://api.nasa.gov/. NASA offers a wide range of endpoints, including but not limited to:

- Astronomy Picture of the Day (APOD)
- Mars Rover Photos
- Earth Polychromatic Imaging Camera (EPIC)
- Near Earth Object Web Service (NeoWs)
- NASA Image and Video Library

You have the freedom to choose which endpoint(s) you want to work with. You can focus on a single endpoint or integrate multiple endpoints for a more comprehensive application. Be creative and build something that showcases your skills and interests!

We're looking for code that is well-structured, readable, and efficient. You should use best practices for both frontend and backend development, and make use of any relevant libraries and tools that you feel will make your work easier. Good luck!

Mandatory Technologies:

- React
- NodeJS
- Express

Evaluation Criteria:

- Frontend design & UI/UX
- Creativity and uniqueness of the concept
- Quality of data visualisation and presentation
- Backend architecture and API integration
- Error handling and edge cases
- Loading state management
- Code structure, quality, and best practices
- File/repository organisation
- README.md clarity and completeness
- Deployment of the application (e.g., Vercel, Render, Heroku)

Bonus Points:

- Implementing user interactivity (e.g., filters, search functionality)
- Responsive design for various screen sizes
- Performance optimization
- AI features
- Testing (e.g., Jest, React Testing Library for frontend; Jest for backend)
- Additional features that enhance the user experience

Submission Guidelines:

Reply to the email in which you received this coding challenge with the following:

1. A link to the deployed application where it's running live.
2. A link to the GitHub repository containing your source code, which must include:
    - A comprehensive README.md describing how to set up, start, and use the application
    - Example file structure:
        - ├── frontend/
        - ├── backend/
        - └── README.md

Deadline:

Please submit your completed challenge no longer than **two weeks after receipt of this email**.

Good luck, and we look forward to seeing your creative NASA data explorer!