EXOPLANETARIUM By Team code4change
NASA SPACE APPS NOIDA | Innovation partner H2S

Exoplanetarium Execution Plan

Welcome to *Exoplanetarium*, an interactive platform for exploring distant exoplanets! Follow this guide to understand the application structure, explore its features, and run the project locally.

1. Repository Overview

- **Frontend Folder**:
  - Contains all user-facing components, built using **React.js** and **Three.js** for interactive 3D visualizations.
  - Key Files: index.html, App.js, components/ (for individual tools like Discovery Analyzer, Habitability Estimator).
- **Backend Folder**:
  - Powered by **Flask** and **Node.js** to handle API calls, database queries, and machine learning models.
  - Key Files: app.py, api/ (for endpoints related to exoplanet data processing and habitability scoring).
- **Data Folder**:
  - Includes the required datasets for exoplanets (NASA and other sources), including processed CSV files for habitability analysis and planetary characteristics.
  - Key Files: exoplanets.csv, processed_data/.

2. Installation & Setup

1. **Clone the Repository**:
2. bash
3. Copy code
4. git clone https://github.com/yourusername/exoplanetarium.git
5. cd exoplanetarium
6. **Backend Setup**:
   - Install the required Python packages:
   - bash
   - Copy code
   - pip install -r requirements.txt
   - Start the Flask server:
   - bash
   - Copy code
   - python app.py
7. **Frontend Setup**:

- Install the necessary dependencies:
- bash
- Copy code
- npm install
- Run the React frontend:
- bash
- Copy code
- npm start

8. **Data Loading**:
   - Ensure the datasets (e.g., exoplanets.csv) are correctly placed in the data/ folder. The backend will load and process them on startup.

3. Application Features

Explore the following features in the application:

- **Discovery Method Analyzer**:
  - Simulate exoplanet discovery methods (e.g., radial velocity, transit) with **3D visualizations**.
  - View the *Exoplanet Discovery Timeline*, showing how discoveries have progressed over the years with interactive graphs.
- **Habitability Estimator**:
  - Estimate the habitability of exoplanets based on factors like mass, orbit, and stellar temperature.
  - Visualize **scatter plots**, **heatmaps**, and **K-Means clustering** to analyze planetary characteristics and relationships.
- **Exo Comparator**:
  - Compare distant exoplanets by examining properties like **mass**, **orbital period**, and **atmospheric composition**.
  - Generate transmission spectra and see how similar other worlds might be to Earth.

4. Running the Application

Once the server is running and the frontend is started, access the application at http://localhost:3000:

- **Homepage**: Select an exoplanet to visualize its 3D model, textures, and basic properties.
- **Discovery Simulator**: Navigate to explore how different methods detect planets.
- **Habitability Estimator**: Input exoplanet properties and generate habitability scores, using the **ML models** running on the backend.
- **Exo Comparator**: Compare Exoplanets with each other, exploring similarities and potential for life.

5. Code Structure & Technologies

- **Frontend**: React.js, Three.js for dynamic and interactive exoplanet visualizations.
- **Backend**: Flask and Node.js to manage API calls and perform machine learning analysis.

- **Machine Learning**: Models for habitability estimation and clustering are built using **scikit-learn** and **TensorFlow**.

6. Future Enhancements (Coming Soon)

- **Mobile Compatibility**: A mobile-friendly interface for exploring exoplanets on the go.
- **Virtual Reality Integration**: Step into space with a VR experience to fully immerse yourself in the exploration.
- **Expanded Datasets**: Ongoing updates as new exoplanet discoveries are made.