Visualize of space data by Rostislav Kirsanov, Dzhamilia Fatkullina, Sofia Pushkareva

Project Goal and Vision:

This project will create an interactive web application that allows users to explore data about planets, stars, galaxies and space travel. The application may include maps, projections, graphs and other visualizations. Users will be presented with basic characteristics of space objects, spaceflight data, and scales of space. The first visualization panel in the project will be as interactive 3D models of the solar system and star systems. The second panel will contain detailed information about a selected object with its characteristics. The project is aimed at astronomy and space enthusiasts.

Dataset Description:

- **Data Source and Collection Plan:**
  - External Dataset: Our team is planning to use NASA API. We will use APIs to fetch real-time data. We'll identify relevant pages and use XPath selectors to extract table data. The solution to the problem of changes on the site is solved by using JSON files. If there is a rate limiting problem, in this case we will limit the frequency of requests so that it satisfies the possible source rate.

- **Data Content:** The collected data will include:
  - Planets: Name, radius, mass, distance from the Sun, temperature, and atmospheric composition.
  - Stars: Name, brightness, distance from Earth, temperature, and spectral class.
  - Galaxies: Name, type, distance from Earth, diameter, and number of stars.
  - Space Missions: Name, target, launch date, status, and key results.
  - Current data from 2023 and up to the present day will be used, if sources allow.

Visualization App Architecture:

The application will follow a standard data visualization pipeline:

1. **Data Collection:** APIs will gather cosmic data and output JSON.
2. **Data Cleaning and Preprocessing:** Pandas will clean and organize the collected data, handling inconsistencies, missing values, and standardizing data formats.
3. **Data Exploration and Processing:** IPython and Matplotlib/Seaborn will be used for exploratory data analysis (EDA) to identify trends, outliers, and prepare data for visualization. Network analysis might be used to explore relationships between celestial objects.
4. **Data Delivery:** A Flask RESTful API will serve processed cosmic data in JSON, with endpoints for data by category (planets, stars, galaxies, missions) and specific attributes.
5. **Data Visualization:** D3.js and Three.js will be the main visualization libraries. We plan to create:
    - Interactive 3D models of the solar system and star systems.
    - Network graphs showing relationships between celestial objects.
    - Time-series charts of space mission launches and discoveries.

Proposed Visualizations and Features:

- **Panel 1 (Three.js) - Interactive 3D Star Map:** It will be used to realize a 3D model of the solar system, 3D graphics of the size distribution of the planets, as well as a 3D map of the starry sky. The first visualization panel in the project will be made using Three.js. Interactive functions of this panel will be Scaling and rotation, Planet selection, Animation, Animation of orbits, Switching between modes ("Solar system", "Starry sky", "Galaxies").
- **Panel 2 (D3.js) - Celestial Object Information Panel:** A panel displaying detailed information about a selected celestial object, including its characteristics, history, and related missions. The second visualization panel in the project will be implemented using D3.js. It will be used to realize graphs such as: Bar chart, Line chart, Point chart. Interactive functions of this panel will be data grouping, scaling, linked charts.

Project Timeline and Milestones:

- **Week 1:** Project setup, website analysis (for scraping), Scrapy setup or dataset download, initial data exploration.

- **Week 2:** Data cleaning and preprocessing with Pandas, exploring data and designing visualizations.
- **Week 3:** Develop backend and API, customize Flask to create RESTful API.
- **Week 4:** Test APIs.
- **Week 5-6:** Frontend development (Three.js and D3.js), create interactive visualizations.
- **Week 7:** Integration and testing, integrate the API with the frontend.
- **Week 8:** Final refinement, add the finishing touches.