Team Details

a. **Team name:** Quasar Tag
b. **Team leader name:** Aditya Kumar
c. **Problem Statement:** Create an Orrery Web App that Displays Near-Earth Objects

Brief about the idea:

- An interactive map of the Solar System created with Three.js and jQuery and display live data of NEOs, helping users understand their orbits, sizes, and potential risks. The 3D interactive model allows users to explore the solar system dynamically, providing a visually engaging and educational tool for space enthusiasts, researchers, and students.

Opportunities

a. How different is it from any of the other existing ideas?

The 3D orrery web app we're creating stands out by integrating modern web technologies like Three.js for 3D rendering, jQuery for streamlined DOM manipulation, jQuery UI for user interactions, and tween.js for smooth animations. While there are other orreries and space-related simulations online, many are either static or heavily platform-dependent. Our app is web-based, making it accessible across devices, lightweight, and interactive, differentiating it from more traditional desktop-based models or less interactive web orreries.

b. How will it be able to solve the problem?

This 3D orrery web app can solve the problem of providing an engaging, interactive, and accessible educational tool for visualizing and understanding the movement of celestial bodies. By simulating near-Earth objects (NEOs) in real time with smooth animations, users can grasp the dynamic nature of space. Its web-based nature ensures that no installation is required, and it's easily accessible by anyone with an internet connection, making space education more inclusive and user-friendly.

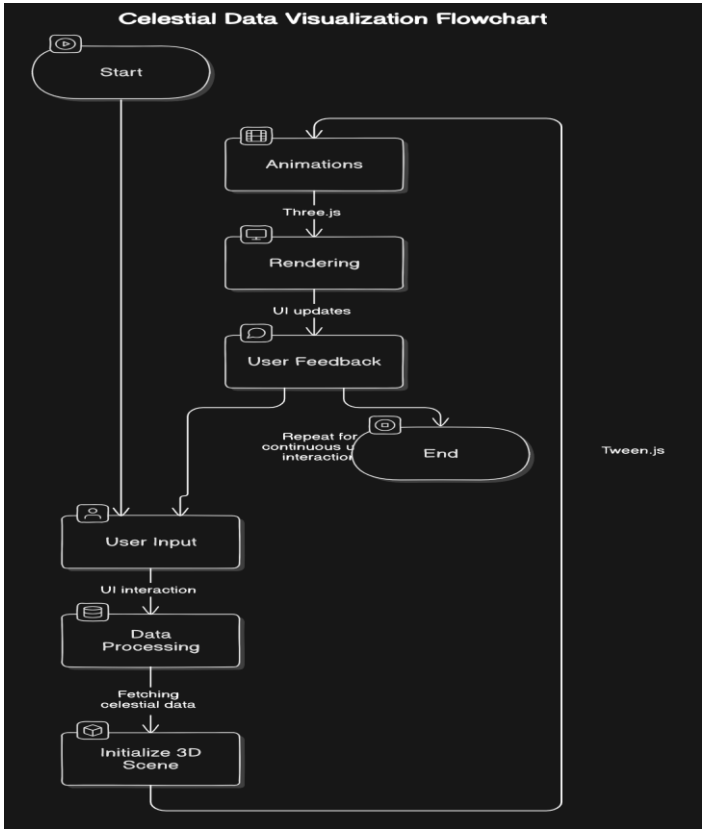
c. USP of the proposed solution

The unique selling point (USP) of our proposed solution lies in its combination of real-time 3D visualizations of space objects with web accessibility, allowing users to interact with an accurate orrery without needing specialized software. Additionally, its use of tween.js for animation and jQuery UI for interaction makes it an intuitive, engaging educational experience, appealing to both casual users and more advanced space enthusiasts. Its adaptability for NASA Space Apps adds a layer of credibility and relevance to the project.

List of features offered by the solution:

1. Ephemerides for astronomical bodies in the Solar System, searchable for all above 1km in radius.
2. Exaggerated orrery view at the solar system scale, zoomable to true scale for planetary systems.
3. Live data readout for the body's physical and orbital characteristics, right ascension/declination and altitude/azimuth coordinates relative to current location, rise and set times, and magnitude adjusted for atmospheric dispersion.
4. Links to articles and images from Wikipedia.
5. Background stars include all above 7th magnitude.
6. HTTPvars for latitude/longitude, start time, and reducing particle count.

Process flow diagram or Use-case diagram



Technologies to be used in the solution:

Front-end Development:

- HTML
- CSS
- JavaScript

Frameworks and Libraries:

- Three.js
- jQuery
- jQuery Ui
- tween.js

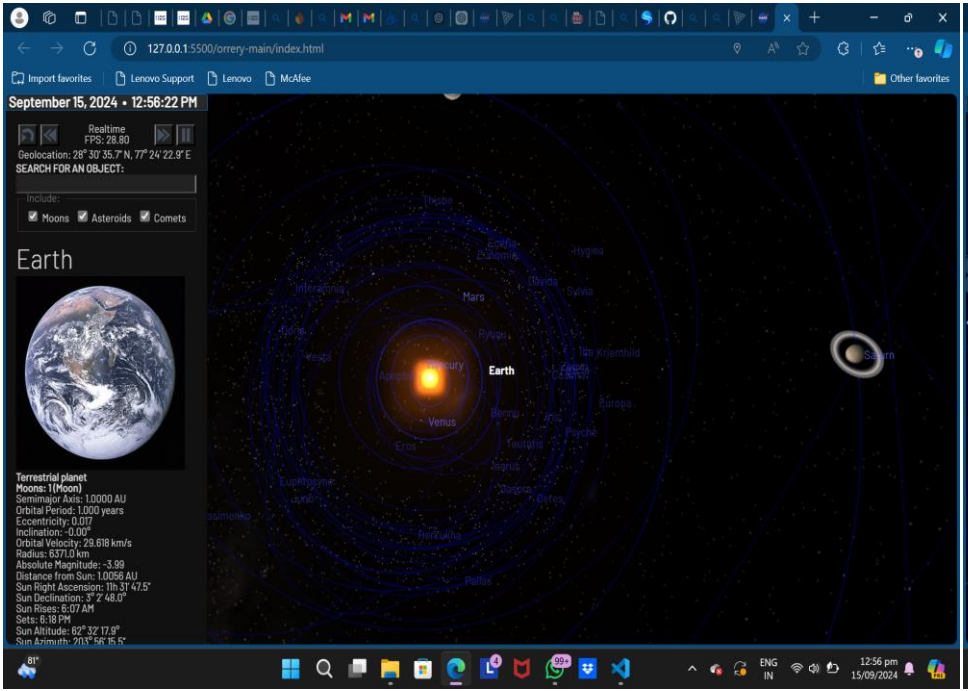
Back-end Development:

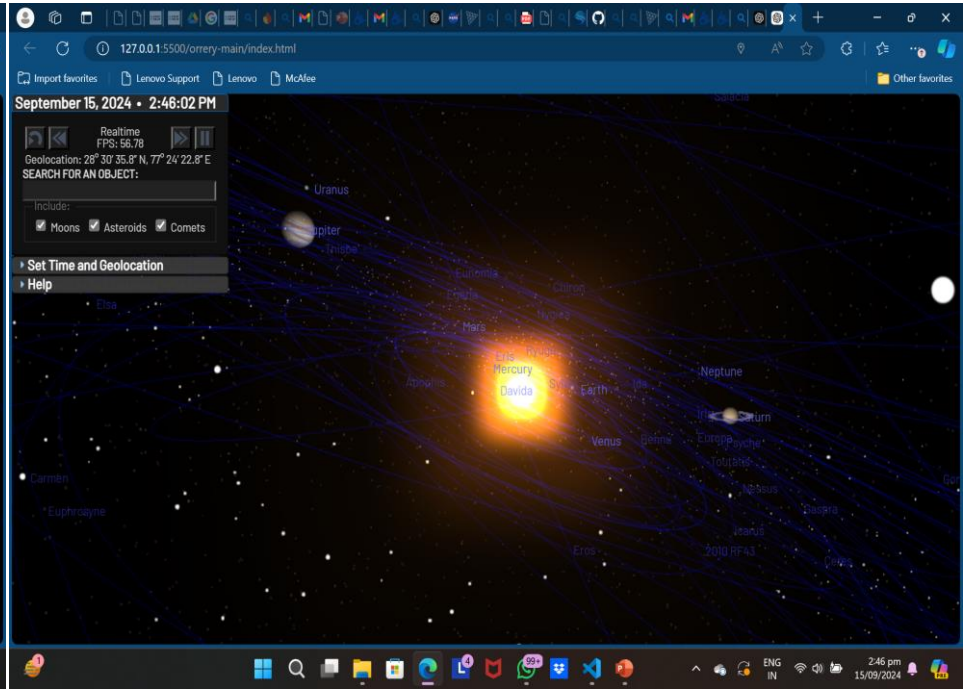
- Node.js

Databases:

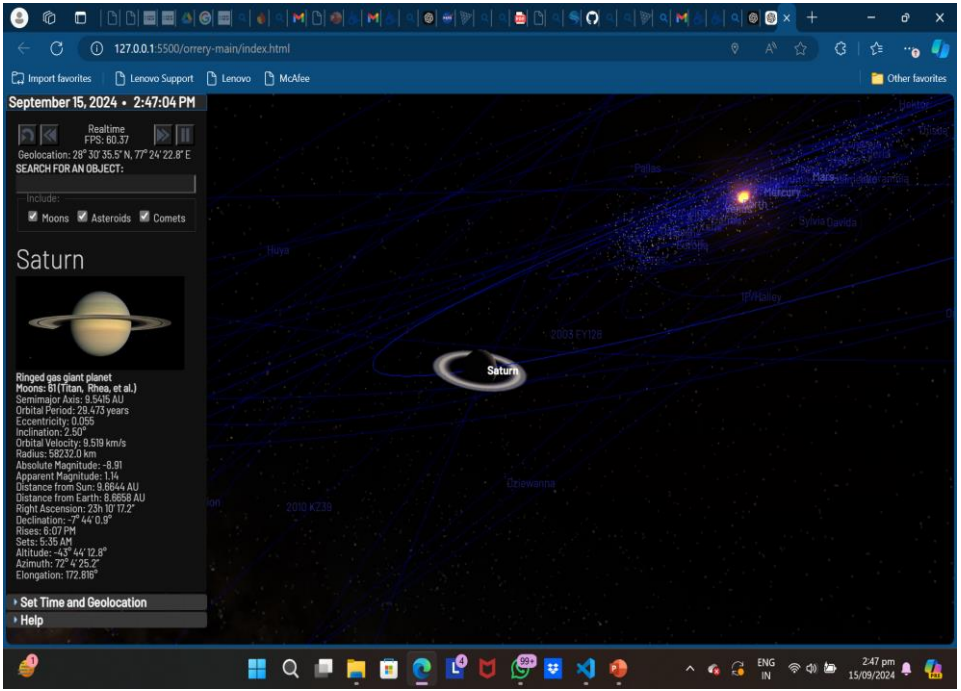
- MySQL

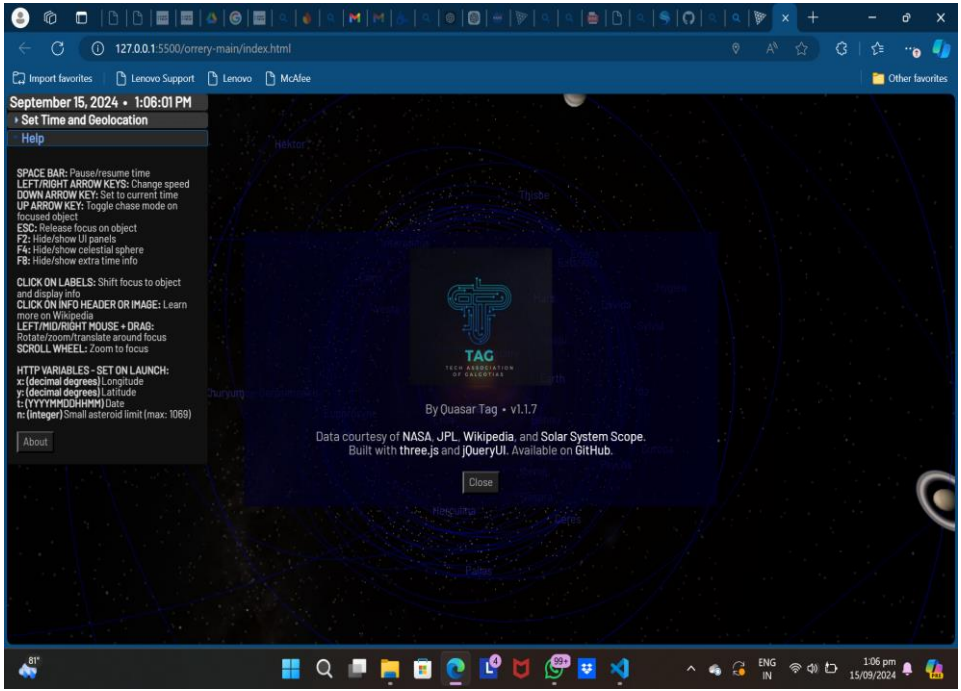
Snapshots of the prototype





Snapshots of the prototype





Prototype Performance report/Benchmarking:

1. Overview of the Prototype

•**Objective**: To create an interactive 3D orrery web app that visualizes celestial bodies, specifically Near-Earth Objects (NEOs), using modern web technologies like Three.js, jQuery, jQuery UI, and tween.js.

•**Scope**: The prototype aims to demonstrate real-time simulation of celestial movements with smooth animations and user interaction, making it accessible across devices via a web browser.

2. Performance Metrics

•**Load Time**:
- Benchmark the time it takes for the 3D orrery to fully load in different environments (desktop, mobile, various browsers).
- Target: Ideally, it should load within 2-5 seconds on average broadband.

•**Frame Rate (FPS)**:
- Measure the frame rate across different devices. The benchmark is to maintain 60 FPS for smooth animations on most modern devices.
- Tools like Chrome DevTools, FPS meters, or WebGL stats can help measure this.

•**Memory Usage**:
- Track memory consumption when rendering the orrery. High memory usage may indicate a need for optimization.
- Target: Less than 200MB for standard usage (depending on the complexity of objects and scene).

3. User Interaction Benchmark

•**Responsiveness**:
- Measure how quickly the app responds to user input (e.g., rotating the scene, zooming in/out).
- Target: Less than 100ms latency between interaction and response.

•**UI Controls**:
- Evaluate the ease of use of interactive elements like sliders and buttons (using jQuery UI). Ensure smooth operation across devices.

4. Cross-Browser Compatibility

•Test the performance and compatibility of the app across different browsers (Chrome, Firefox, Safari, Edge).

•Ensure consistent rendering of 3D objects and animations in all major browsers.

•**Benchmark**: Pass compatibility on all major browsers, including mobile versions.

Prototype Performance report/Benchmarking:

5. Optimization Testing

- **3D Object Rendering**: Test the complexity and number of objects the app can handle without significant performance degradation.
  - **Benchmark**: Render at least 10-15 complex objects (planets, NEOs) with textures and animations while maintaining stable performance.
- **Animation Smoothness**:
  - Use tween.js for smooth transitions. Test how the animations feel on different devices and adjust if needed.
  - **Benchmark**: No stuttering or frame drops during animations, especially during user interactions like zoom or rotate.

6. Security & Accessibility

- **Security**:
  - Ensure no vulnerabilities, such as cross-site scripting (XSS), are present in the app.
- **Accessibility**:
  - Check for accessibility compliance (WCAG standards), ensuring that the app can be navigated by people using assistive technologies.
  - **Benchmark**: Meet at least WCAG 2.1 Level AA standards.

Additional Details/Future Development:

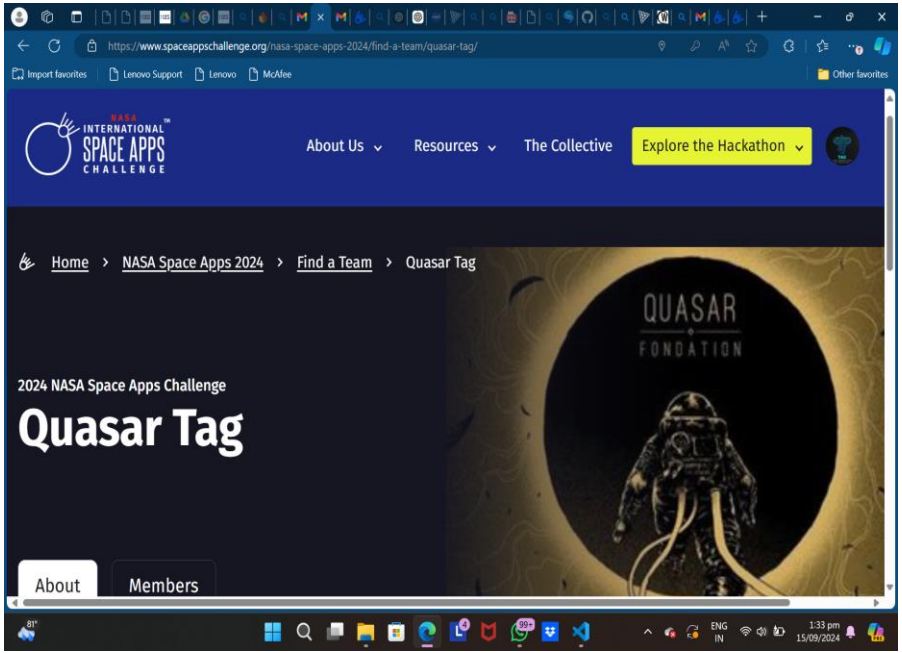
Potential Roadmap:

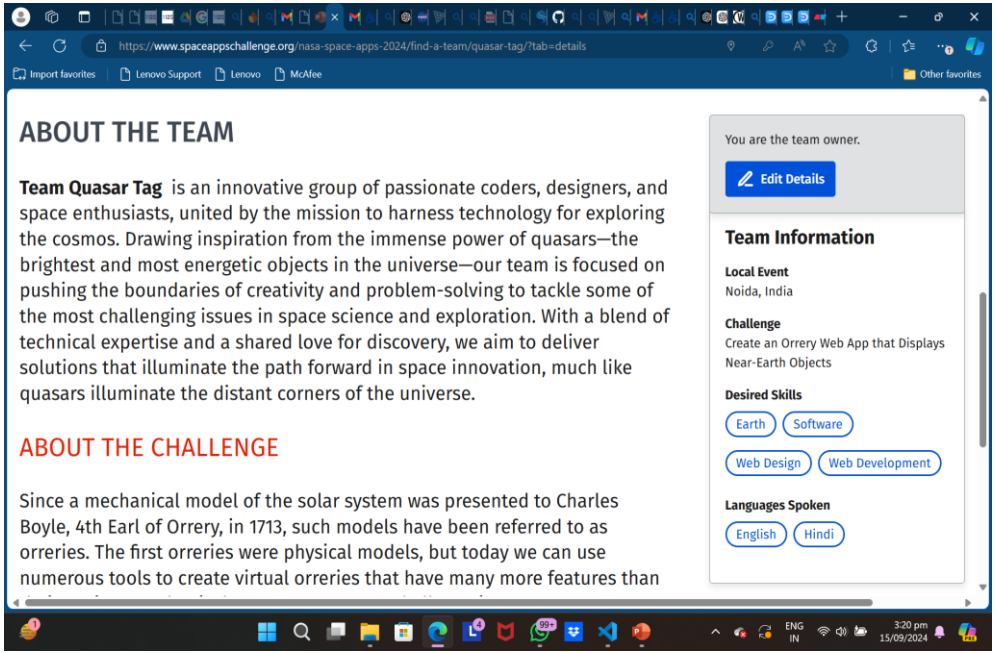
- UI scaling for mobile
- Object count throttling from FPS observation
- Non-periodic comets/hyperbolic orbits (`Omuamua, etc.)
- Extended data/add ins for smaller objects
- Planetarium view with sky shader
- Asteroid categorization from orbital elements
- Asteroid 3D shapes
- Ring shadows
- Spacecraft/Two Line Element reader
- NASA SPICE kernels
- Better image compression (Basis)
- THREE.CSS2D renderer or Canvas for tags (if faster)
- Procedural textures
- Exoplanetary systems
- Lagrange points/Hill spheres for planets
- XR
- Hohmann transfer orbits, launch window solver, delta-v requirements
- Gravitational simulations
- Compute shaders

Provide links:

1. *GitHub URL:* https://github.com/eak300515/Nasa_Spaceapps2024-3D_orrery.git

Proof of Registration on https://www.spaceappschallenge.org/nasa-space-apps-2024/2024-local-events/noida





NASA SPACE APPS NOIDA

Innovation partner H2S HACK2SKILL

NASA Space Apps Noida 2024

World's Largest Space & Science Hackathon

Thank You