Team Details

a. Team name: DISPICABLES
b. Team leader name: KASHISH SHRIVASTAV
c. Problem Statement: imagine our connected earth.

Brief about the idea:

My project is an educational platform designed to teach students about climate change and environmental science in an engaging and interactive way.

It features:

- **Podcast Integration:** Students can listen to podcasts on climate-related topics.
- **MCQ Generator:** An AI-powered MCQ generator that creates quizzes based on given text, using the Gemini API key.
- **Climate Data Visualization:** A visually appealing interface that displays climate data in tables, 2D, and 3D graphs.
- **Interactive Earth Model:** An interactive 3D Earth model with changing textures to visualize climate patterns and effects.
- **Educational Games:** Engaging games that help students learn about water pollution and other environmental issues.
- **User-Friendly Interface:** A well-designed UI and UX for a seamless learning experience.

Opportunities

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

Comprehensive Approach: Your platform combines multiple learning modalities, including podcasts, MCQs, data visualization, and interactive games, providing a well-rounded and engaging learning experience.

AI-Powered MCQ Generator: The use of an AI-powered MCQ generator adds a unique and personalized element to your platform, tailoring quizzes to the student's learning style and progress.

Interactive Earth Model: The 3D Earth model with changing textures offers a visually appealing and immersive way to explore climate data and its impacts.

Real-World Data Integration: The integration of NASA climate data provides students with authentic and up-to-date information.

Educational Games: The inclusion of educational games makes learning about climate science fun and interactive, appealing to a wider range of students.

Holistic Approach: Your platform addresses multiple aspects of climate education, from theoretical knowledge to practical application and engagement.

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

Providing an Interactive Platform: The 3D Earth model and user-friendly interface allow users to explore and visualize Earth's systems in a dynamic and engaging way.
Utilizing NASA Data: The integration of NASA Earth observing satellite data provides accurate and up-to-date information about Earth's systems.
Highlighting Interconnections: The visualization and data analysis features of your app can showcase how changes in one system (e.g., climate change) impact others (e.g., wildfires, ecosystems).
Engaging the Public: The interactive nature of your app and the use of multimedia elements (e.g., audio, visuals) can capture the attention of a wider audience and foster a deeper understanding of Earth's interconnectedness.

USP of the proposed solution

Interactive and Engaging Experience: Your application offers a visually appealing and interactive way to explore Earth's systems, making it more engaging than traditional educational materials.
Real-Time Data: By utilizing NASA satellite data, your application provides up-to-date information on Earth's systems, allowing users to see the latest developments.
Visual Storytelling: The 3D Earth model and visualizations effectively communicate complex concepts and relationships between Earth's systems.
Educational Focus: Your application is designed to educate users about climate change and environmental science, making it a valuable tool for students and educators.
Accessibility: By providing a user-friendly interface and interactive elements, your application is accessible to a wide range of users, including those with limited technical knowledge.
Customization: The ability to customize the visualization and focus on specific regions or data points allows users to tailor the experience to their interests.

List of features offered by the solution

Interactive 3D Earth Model: A visually appealing and immersive representation of Earth.
NASA Data Integration: Access to real-time or near-real-time data from NASA satellites.
Data Visualization: Visualization of various Earth systems (e.g., climate, land use, ocean currents) using different data layers and visualization techniques.
User Interaction: Features like zooming, panning, and rotating the Earth model for exploration.
Information Panels: Display of relevant information, statistics, and explanations related to the visualized data.
ADDITIONAL FEATURES GOING TO INCLUDE:

- **Time Slider:** Ability to explore data over time to observe changes and trends.
- **Search Functionality:** Search for specific locations or data points.
- **Customization Options:** Allow users to customize the visualization (e.g., color schemes, markers).
- **Educational Content:** Integrate educational resources like articles, videos, or quizzes.
- **Social Sharing:** Enable users to share visualizations or findings on social media.
- **User Profiles:** Create user profiles for personalization and tracking progress.

Initialize Django Project

Set up the Django project structure

Register necessary apps (e.g., grid_model)

Retrieve grid data from NASA or other sources

Clean, preprocess, and format the data for visualization

Use a 3D visualization library (e.g., Plotly) to create the Earth model

Apply textures, lighting, and other visual effects

Map grid data points onto the 3D model

Visualize data using color-coding, markers, or other techniques

Design and develop the user interface (UI) using HTML, CSS, and JavaScript

Include elements like navigation, controls, and information panels

Enable user interactions like zooming, panning, and filtering data

Handle user input and update the visualization accordingly

Integrate the 3D model and UI with your Django application

Create views and URL patterns to handle user requests

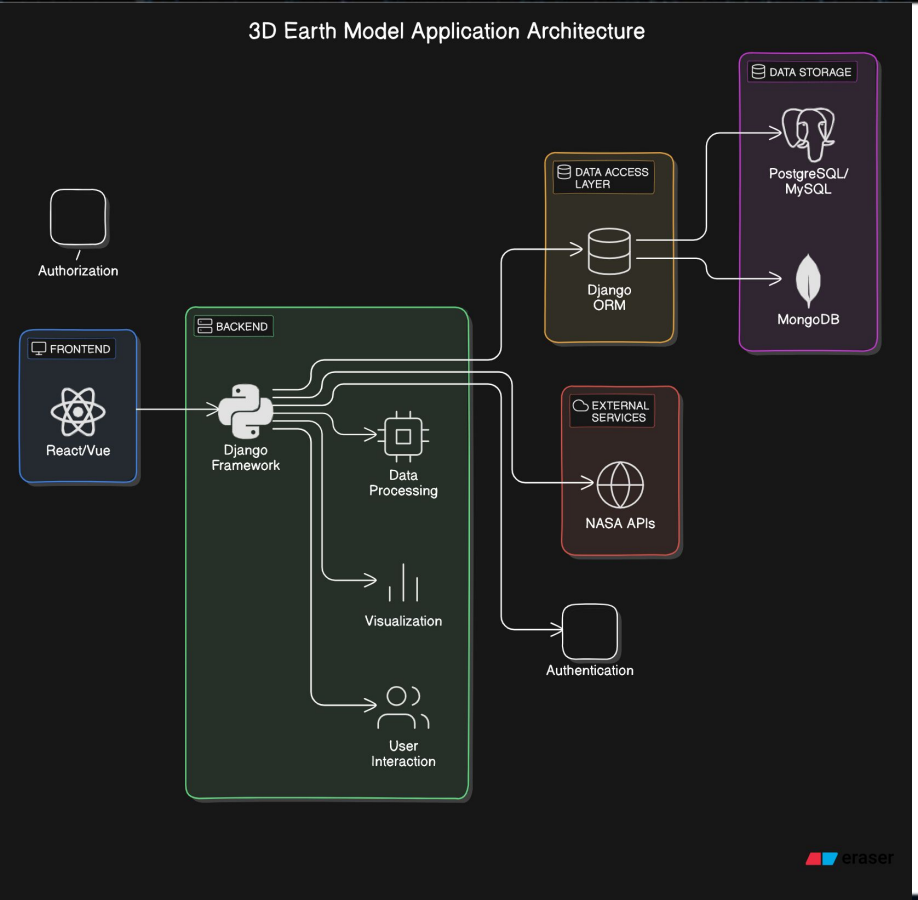
Start the Django development server to test and view the application

Adapt the flow diagram based on specific project requirements

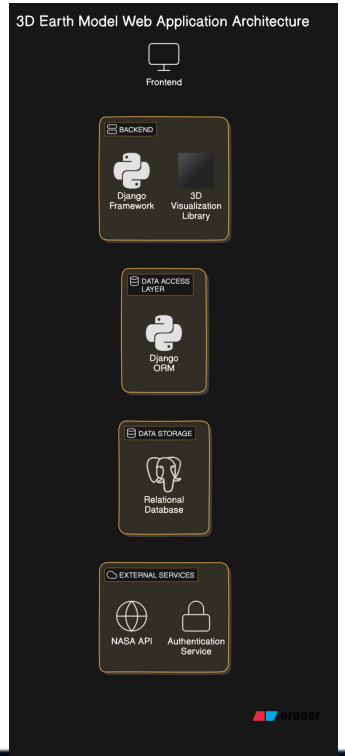
Consider adding steps for data validation, error handling, and performance optimization

Add steps for real-time data updates or machine learning integration if needed

FLOWCHART



Architecture diagram of the proposed solution



Technologies to be used in the solution

Backend:

- **Django:** A popular Python web framework for building web applications.
- **Django REST framework:** For creating RESTful APIs to interact with the frontend.
- **Database:** A relational database like PostgreSQL or MySQL to store data.
- **Data Access Layer:** Django ORM or other tools for interacting with the database.

Frontend:

- **HTML, CSS, JavaScript:** For creating the user interface and handling interactions.
- **Frontend Framework:** A framework like React, Vue, or Angular can be used for complex UI components and state management.

3D Visualization:

- **Plotly:** A Python library for creating interactive visualizations, including 3D plots.
- **Three.js:** A JavaScript library for creating and displaying 3D graphics in a web browser.
- **Other 3D Libraries:** Explore other libraries like WebGL or Cesium for specific requirements.

Data Integration:

- **NASA APIs:** Use NASA's APIs to access and retrieve grid data.
- **Data Processing Libraries:** Libraries like Pandas or NumPy for data cleaning and manipulation.

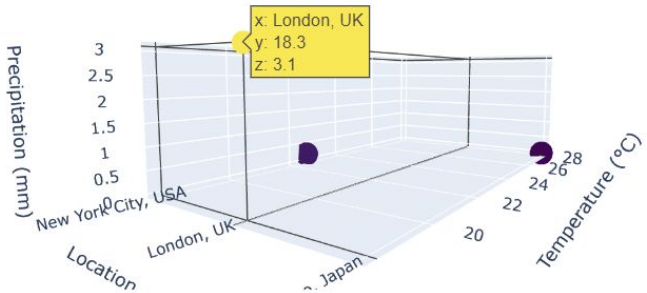
Snapshots of the prototype

CLIMATE DATA VISUALIZATION

Location	Date	Temperature (°C)	Precipitation (mm)
New York City, USA	2023-09-10	22.5	0.2
London, UK	2023-09-10	18.3	3.1
Tokyo, Japan	2023-09-10	27.8	0.0

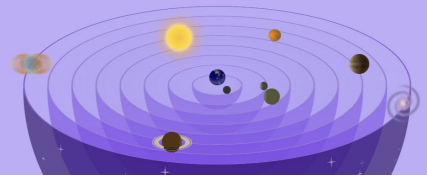
2D Climate Data Visualization

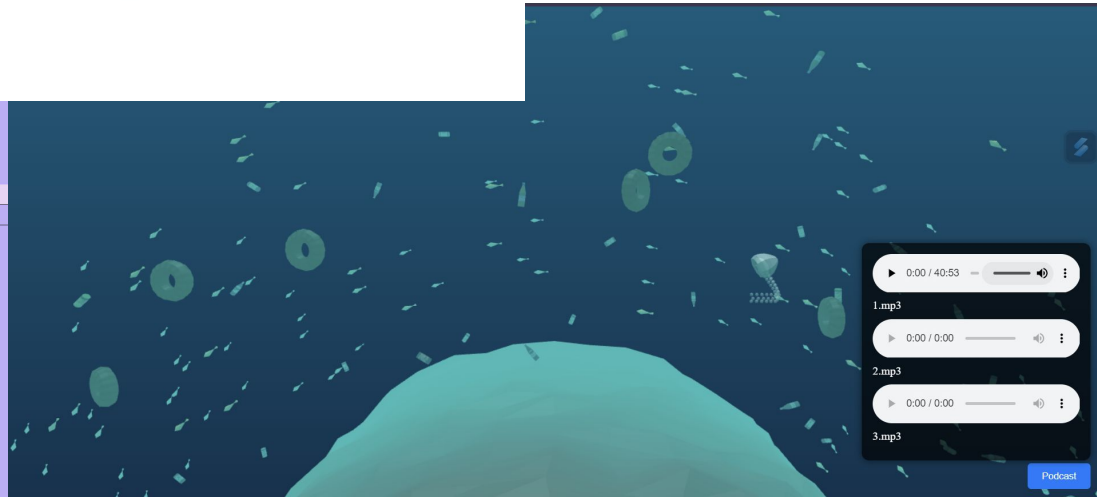
3D Climate Data Visualization

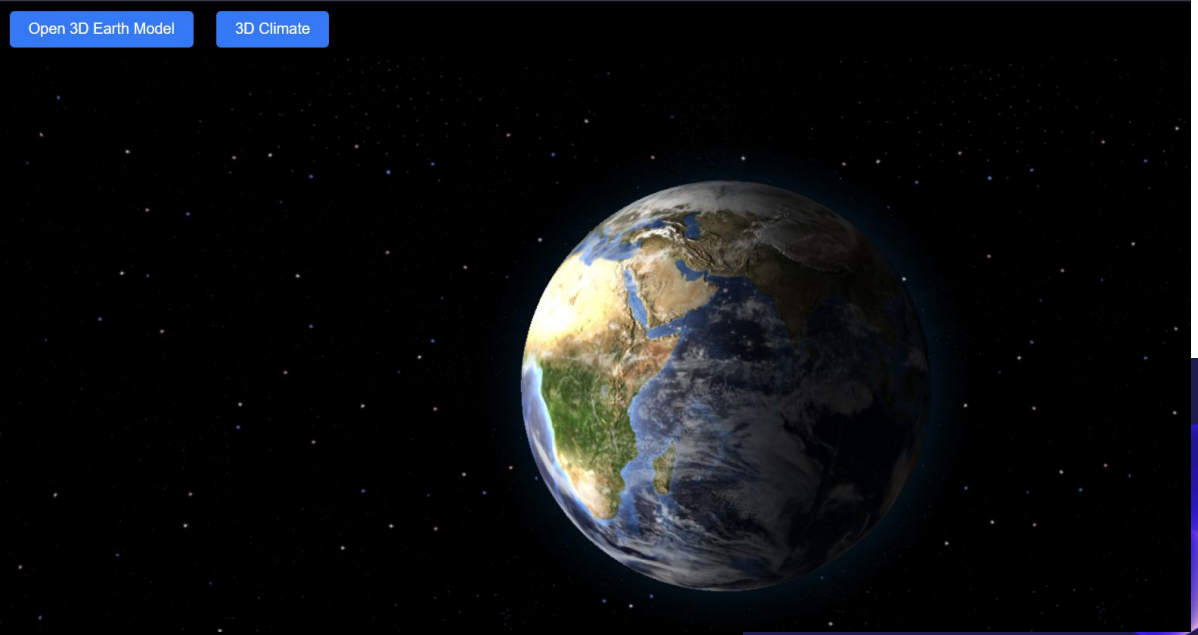


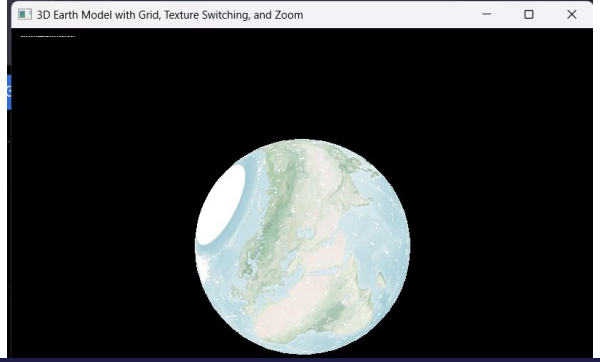
CLIMATE DATA^_^

Location	Date	Temperature (°C)	Precipitation (mm)
New York City, USA	2023-09-10	22.5	0.2

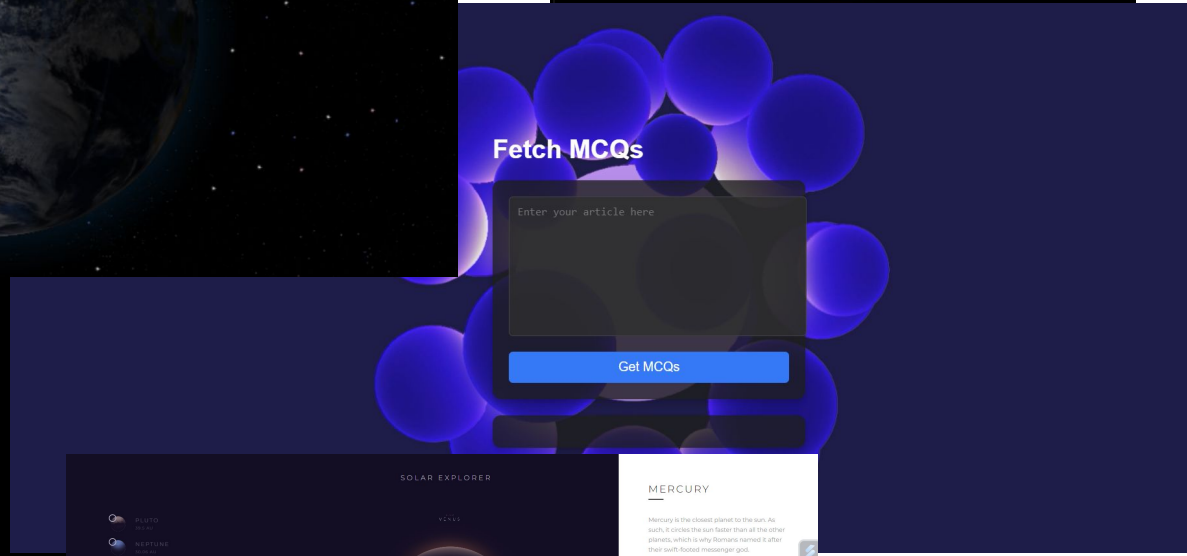





Open 3D Earth Model
3D Climate


3D Earth Model with Grid, Texture Switching, and Zoom


SOLAR EXPLORER
PLUTO
NEPTUNE
URANUS
SATURN
JUPITER
MARS
EARTH
VENUS
MERCURY
PLANET
MERCURY
THE CLOSEST PLANET TO THE SUN. IT CIRCLES THE SUN FASTER THAN ALL THE OTHER PLANETS, WHICH IS WHY ROMANS NAMED IT AFTER THEIR SWIFT-FOOTED MESSENGER GOD
READ MORE


Fetch MCQs
Enter your article here
Get MCQs


SOLAR EXPLORER
Back
PLANET
MERCURY

MERCURY
Mercury is the closest planet to the sun. As such, it circles the sun faster than all the other planets, which is why Romans named it after their swift-footed messenger god.
The Sumerians also knew of Mercury since at least 5,000 years ago. It was often associated with Nabu, the god of writing. Mercury was also given separate names for its appearance as both a morning star and as an evening star. Greek astronomers knew, however, that the two names referred to the same body, and Heraclitus, around 500 B.C., correctly thought that both Mercury and Venus orbited the sun, not Earth.

Prototype Performance report/Benchmarking

It is under building phase so we have not develop its ui and ux that much good but we will develop it in shortly

Additional Details/Future Development (if any)

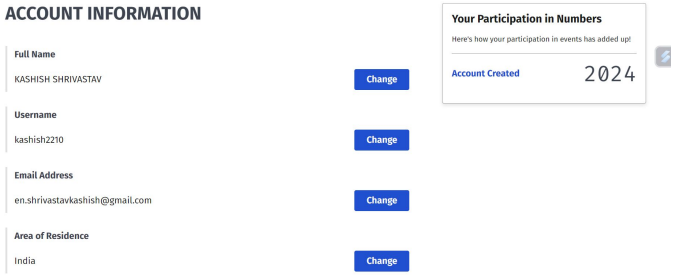
We include this features in future:

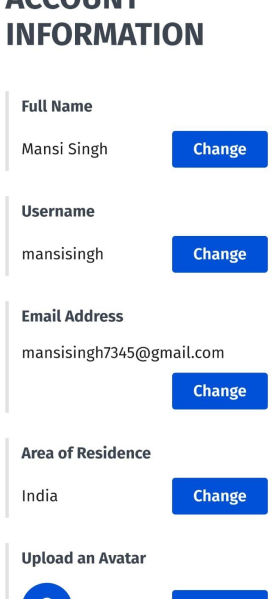
- **Machine Learning Integration:** Incorporate machine learning algorithms to analyze data and make predictions.
- **Real-time Data Updates:** Connect to live data feeds to display the most recent information.
- **User-Generated Content:** Allow users to upload their own data or create custom visualizations.
- **Virtual Reality (VR) Compatibility:** Enable users to experience the 3D Earth model in a VR environment.
- **Augmented Reality (AR) Integration:** Combine the 3D model with real-world surroundings using AR technology.

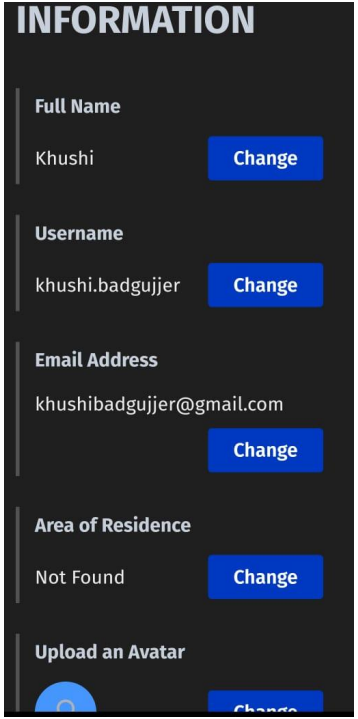
Provide links to your:

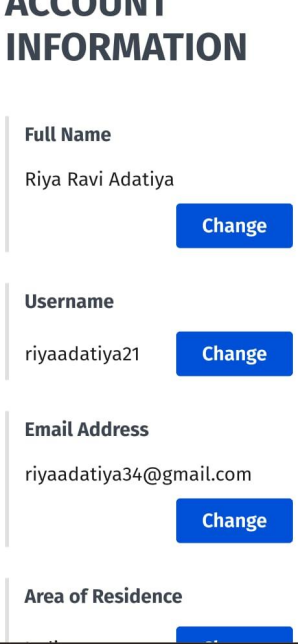
1. GitHub Public Repository
   clickHere
2. Demo Video Link (use headphones for clear voice^_^)
   clickHere
3. Final Product Link
   “It is not deployed yet it is running on localhost”

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