International Research Journal of Education and Technology

Peer Reviewed Journal, **ISSN: 2581-7795**

Stellar Space Weather Monitoring and Interpretation Platform for Earthlings

Faizal S A, Vishal P S, Kishore K

[1]Student (Mechanical Engineering) at Bannari Amman Institute of Technology
[2]Student (Mechanical Engineering) at Bannari Amman Institute of Technology
[3]Student (Mechanical Engineering) at Bannari Amman Institute of Technology
Dept. of Mechanical Engineering, Bannari Amman Institute of Technology, India

---***---

Abstract

Space weather events such as solar flares, geomagnetic storms, and coronal mass ejections significantly influence satellite communication, navigation systems, and Earth's magnetic environment. However, the complex scientific representation of these phenomena often makes them difficult for students and general audiences to understand. This paper presents "Stellar Stories", an interactive space-themed storytelling platform designed to transform real-time space weather data into engaging educational narratives. The system integrates external scientific data sources with modern web technologies to create a visually rich and informative learning environment.

The platform is developed as a full-stack web application using Vue for the frontend interface and Tailwind CSS for responsive styling. The backend is implemented using the Django REST framework, which manages business logic, API routing, and data processing. A PostgreSQL database is used for efficient storage and retrieval of user profiles, bookmarked stories, and cached event data. Real-time space weather information is retrieved from NASA Open APIs, including the Astronomy Picture of the Day (APOD) API and the DONKI API, which provides solar flare and geomagnetic storm data.

To enhance visualization and user engagement, the system incorporates CesiumJS to render a dynamic 3D Earth model where users can observe the impact of space weather events. Character-based storytelling logic translates scientific events into narrative explanations, enabling improved understanding of complex phenomena. The system architecture supports scalable deployment in a production server environment, ensuring reliable performance and accessibility. The proposed platform demonstrates how modern web technologies and open scientific data can be combined to create an effective educational tool for space science learning.

Key Words: Space Weather, Web Application, Django REST API, 3D Earth Visualization, NASA API, Educational Platform, Storytelling System

1.INTRODUCTION

Space weather refers to environmental conditions in space that arise due to solar activities such as solar flares, coronal mass ejections, and geomagnetic storms. These phenomena can significantly influence satellite communication systems, navigation technologies, and electrical power grids on Earth. Understanding and monitoring space weather events is therefore essential for both scientific research and technological infrastructure. However, the complex representation of space weather data often makes it difficult for students and general audiences to interpret and understand.

Recent advancements in web technologies and data-driven platforms have enabled the development of interactive systems that simplify complex scientific information. Modern web applications use scalable frameworks and API-based architectures to deliver real-time data visualization and user interaction. The Django web framework is widely used for building secure and scalable backend systems due to its modular architecture and strong security features [1][6][7]. Additionally, the Django REST framework enables developers to create efficient RESTful APIs that support communication between frontend interfaces and backend services [2][3].

Frontend technologies also play a critical role in creating interactive educational platforms. Modern responsive web development frameworks enable the creation of dynamic and user-friendly interfaces that improve accessibility and user engagement [9][10]. Integration of frontend frameworks with backend APIs allows developers to build full-stack applications capable of processing real-time data and delivering it through visually rich interfaces.

Several studies have highlighted the effectiveness of combining Django REST frameworks with modern frontend technologies such as Vue.js for developing scalable web applications that support agile development methodologies and efficient data communication [8]. These technologies enable developers to design modular applications that can easily integrate external data services and provide responsive visualization capabilities.

Motivated by these developments, the Stellar Stories platform aims to transform complex space weather data into an interactive storytelling experience. The system integrates real-time data from NASA Open APIs and converts scientific information into narrative-based explanations supported by 3D visualization. By combining web technologies, API integration, and interactive storytelling, the platform seeks to improve the accessibility and educational value of space science for students and general learners.

1.1 SYSTEM OVERVIEW

The Stellar Stories platform is designed as a full-stack web application consisting of frontend,backend, database, and external API integration layers. The frontend interface allows users to explore space weather stories, visualize Earth in 3D, and interact with categorized content. The backend manages API requests, authentication, and storytelling logic. External APIs from NASA provide real-time astronomical and space weather data, which are processed and converted into narrative-based explanations. The integration of CesiumJS enables 3D Earth visualization where users can observe spatial representations of solar and geomagnetic events.

1.2 OBJECTIVES OF THE SYSTEM

The main objectives of the Stellar Stories platform are:

- To transform scientific space weather data into interactive storytelling content.
- To integrate real-time data from NASA Open APIs.
- To provide an engaging 3D visualization of Earth using modern web technologies.
- To develop a scalable full-stack web architecture for educational applications.
- To improve accessibility and understanding of space science for students and learners.

2.SYSTEM ARCHITECTURE

The architecture of the Stellar Stories platform consists of four main layers: frontend, backend, database, and external data integration.

Frontend Layer: The user interface is built using Vue, which enables dynamic content rendering and
component-based development. Tailwind CSS is used for responsive styling and modern UI design. Backend Layer: The backend is implemented using Django REST Framework, which manages business logic, user authentication, API routing, and communication between frontend and external services. Database Layer: PostgreSQL is used as the primary database system for storing user profiles,
bookmarked stories, cached API data, and application metadata. Visualization Layer: CesiumJS is integrated into the frontend to render a real-time 3D Earth model.
This allows users to visualize the geographic and spatial context of space weather events. External APIs: The system retrieves real-time information from NASA APIs, including the Astronomy Picture of the Day (APOD) API and the DONKI API, which provides solar flare and geomagnetic storm data.

3. IMPLEMENTATION

The development process involved integrating multiple technologies to ensure seamless data flow and interactive visualization. The Vue frontend communicates with the Django backend using REST API endpoints. These endpoints process requests and retrieve relevant data from the PostgreSQL database or external NASA APIs. When a user explores a story category, the backend retrieves corresponding space weather data from the DONKI API. The system processes this information and converts it into simplified narrative descriptions using predefined storytelling templates. The CesiumJS library renders a 3D Earth model that allows users to visually explore the spatial relationship between Earth and solar activity. This enhances the learning experience by combining scientific data with interactive visual elements.

4.RESULT AND DISCUSSION

The developed platform successfully integrates real-time space weather data with interactive storytelling and visualization features. Users can explore categorized stories related to solar flares, geomagnetic storms, and other cosmic phenomena. The 3D Earth visualization significantly improves user engagement by providing a spatial perspective of events. The use of REST APIs ensures efficient communication between system components, while PostgreSQL provides reliable data storage. The system demonstrates how educational technology can combine scientific datasets with modern web development frameworks to create an accessible and engaging learning platform.

5. CONCLUSIONS

The Stellar Stories platform demonstrates an innovative approach to presenting complex scientific information through storytelling and interactive visualization. By integrating NASA space weather data with modern web technologies such as Vue, Django REST Framework, PostgreSQL, and CesiumJS, the system creates an engaging educational environment for exploring space science. The platform successfully converts raw scientific data into understandable narratives while providing visual representation through a 3D Earth model. Future improvements may include advanced data analytics, AI-driven storytelling, and extended visualization features to further enhance the learning experience.

ACKNOWLEDGEMENT

The authors would like to acknowledge the Smart Education Initiative and the open data resources provided by NASA APIs, which made the development of this project possible.

REFERENCES

[1] Chen, S., Ahmmed, S., Lal, K., & Deming, C. (2020). Django web development framework: Powering the modern web. American Journal of Trade and Policy, 7(3), 99-106.

[2] Hillar, G. C. (2018). Django RESTful Web Services: The Easiest Way to Build Python RESTful APIs and Web Services with Django. Packt Publishing Ltd.

[3] Vincent, W. S. (2022). Django for APIs: Build web APIs with Python and Django. WelcomeToCode.

[4] Ravindran, A. (2015). Django Design Patterns and Best Practices. Packt Publishing Ltd.

[5] Ram, J. R., Begum, M. A. F., Sindhu, P., Chandana, M., & Hebsheeba, M. (2023). A new Django web framework software metrics measurement using radon and pylint. Turkish Journal of Computer and Mathematics Education, 14(3), 772-779.

[6] Chandiramani, A., & Singh, P. (2021). Management of Django web development in Python. Journal of Management and Service Science (JMSS), 1(2), 1-17.

[7] Duisebekova, K., Khabirov, R., & Zholzhan, A. (2021). Django as secure web-framework in practice. Вестник Казахской академии транспорта и коммуникаций им. М. Тынышпаева, (1), 275-281.

[8] Arévalo, J. G., Viecco, L., & Arévalo, L. (2020, May). Methodology to define an integration process between frameworks SCRUM, Django REST framework y Vue. js, implemented for software development, from quality management approach and agility. In IOP Conference Series: Materials Science and Engineering (Vol. 844, No. 1, p. 012022). IOP Publishing.

[9] 9.Mohane, A., & Ghate, K. (2021). Understanding Web Front-end Development Technology based upon Responsiveness, Optimization and Interactive Nature.

[10] 10. Vayadande, K., Purohit, S., Rathod, C., Rathod, M., Rathi, P., & Rathi, P. (2024). Web Development and Performance Comparison of Web Development Frameworks: A Review Paper. Grenze International Journal of Engineering & Technology (GIJET), 10.