SRINIVAS UNIVERSITY

TITLE: GROUND CONTROL STATION FOR CANSATS

PROJECT SYNOPSIS

STUDENT NAME	: Hemanth .M
REGISTRATION NO.	: 3SU22BC003
BATCH	: 2022 - 2025
COURSE & SEM	: BCA VI SEMESTER
GUIDE NAME	: Prof. Manjula C M Prasad
DESIGNATION	: Professor & Head Academics
EXTERNAL GUIDE	: Ashwin Reddy
COMPANY	: ITCA, INDIRANAGAR
GUIDE SIGN	:
STUDENT SIGN	: Hemanth .M

SYNOPSIS

Title: Ground Control Station for cansats

ABOUT THE PROJECT

The Ground Control Station (GCS) for CanSats is a software-based system designed to receive, process, and visualize telemetry data from a CanSat—a small satellite used for educational and research purposes. Since real CanSat hardware is not available, this project will use simulated telemetry data to develop and test the GCS. The system will generate artificial sensor data such as temperature, pressure, altitude, and GPS location, simulating the actual behaviour of a CanSat in flight. This data will be processed in real time and displayed on an interactive dashboard, allowing users to monitor and analyze telemetry trends effectively.

The project consists of a web-based application built using React.js for the frontend, Node.js with Express.js for the backend, and PostgreSQL for data storage. A Python script will simulate CanSat telemetry data and send it via WebSockets or a serial port (UART) to the backend, which will process and store the data in a database. The real-time dashboard will present the telemetry information using graphs, heatmaps, and 3D models, enabling users to track the CanSat's flight status as if it were operating in a real-world environment. The system will also include authentication, historical data storage, and export features to enhance usability and security.

This project is entirely software-based and does not require real hardware, making it an ideal solution for educational institutions, research teams, and engineers looking to study CanSat telemetry systems. While the current implementation uses simulated data, the system is designed to be hardware-ready, meaning it can seamlessly integrate with real CanSat telemetry in the future. By providing an interactive and scalable platform, this project serves as an essential tool for developing and testing CanSat missions without the need for expensive satellite hardware.

OBJECTIVES:

The primary objective of this project is to develop a **Ground Control Station (GCS) for CanSats** that can efficiently receive, process, and visualize telemetry data. Since hardware is not available, the system will use **simulated telemetry data** to test its functionality. The project aims to build an **interactive web-based dashboard** for real-time monitoring, data analysis, and historical telemetry storage. By integrating modern web technologies and data visualization tools, the system will provide users with an intuitive and scalable platform for CanSat operations. The following objectives outline the key functionalities of the project:

- **Develop a Ground Control Station (GCS) for CanSats –** Create a software-based system to receive, process, and visualize telemetry data from a CanSat.
- **Simulate CanSat Telemetry Data –** Generate artificial sensor data (temperature, pressure, altitude, GPS location) to test and validate the GCS system without actual hardware.
- **Design a Web-Based Interactive Dashboard –** Build a real-time telemetry dashboard using React.js for the front end and Node.js with Express.js for the back end.
- **Implement Real-Time Data Streaming –** Use WebSockets or UART (serial communication) to transmit and display real-time telemetry data dynamically.
- **Store and Manage Telemetry Data Efficiently –** Utilize PostgreSQL for structured storage and retrieval of historical CanSat telemetry records.
- **Create Advanced Data Visualization Features –** Integrate Chart.js, D3.js, and Three.js to display telemetry data using interactive graphs, heatmaps, and 3D models.
- **Ensure Security and Scalability –** Implement **JWT-based authentication**, caching, error handling, and a modular architecture for future hardware integration.

PROJECT CATEGORY & TYPE

Category	**:** Ground Control Station & Telemetry Monitoring
Type	**:** Software Development (Full-Stack Web Application)

EXISTING SYSTEM

In the current scenario, most CanSat ground control stations rely on manual data collection or basic telemetry logs, making real-time data analysis and interpretation difficult. Existing systems often present raw telemetry in static tables or reports, lacking interactive elements for effective trend visualization and anomaly detection. Additionally, many traditional telemetry platforms require expensive hardware and specialized expertise, making them unsuitable for lightweight CanSat missions. These limitations highlight the need for a modern, scalable, and user-friendly solution that enables real-time monitoring, interactive visualization, and intuitive access to telemetry data for students, researchers, and enthusiasts.

Limitations of the Existing System

- **Lack of Interactive Visualization –** Most existing systems display telemetry data in raw logs or static tables, making it difficult to analyze trends and detect anomalies.
- **No Real-Time Data Streaming –** Traditional ground control systems often require manual data refreshes, lacking dynamic real-time telemetry updates.
- **Complex and Non-User-Friendly Interfaces –** Many current solutions require technical expertise, making them inaccessible to students and beginners.
- **Limited Data Storage and Analysis** – Inefficient database structures lead to slow data retrieval, and users have limited options for filtering and analyzing telemetry data.
- **Scalability and Integration Challenges –** Many platforms are designed for large satellites, making them unsuitable for CanSat missions, and they struggle to handle growing telemetry data efficiently.

PROPOSED SYSTEM

The **GROUND Control Station for CanSats** aims to provide a real-time, interactive, and user-friendly platform for monitoring and analyzing CanSat telemetry data. This web-based system will track key parameters such as altitude, temperature, pressure, GPS location, and system status through dynamic visualizations. Built using **React.js (front end), Node.js (back end), and PostgreSQL (database),** it will ensure seamless telemetry processing and efficient data management. Real-time data streaming will eliminate manual refreshes, while **Chart.js, D3.js, and Leaflet.js** will enable live graphs, heatmaps, and real-time maps for advanced visualization. Users will have access to filtering, comparison tools, secure authentication, and data export features for comprehensive analysis. Designed for scalability and ease of use, this system will provide students, researchers, and space enthusiasts with an intuitive and high-performance solution for both live and historical CanSat telemetry monitoring.

Advantages of the Proposed System

- **Real-Time Data Updates –** The dashboard will support live telemetry streaming, allowing users to access the latest CanSat data without manual refreshes.

- **Interactive and User-Friendly Visualization** – The platform will use Chart.js, D3.js, and Leaflet.js to create dynamic graphs, heatmaps, and real-time maps for easy data interpretation.
- **Efficient Data Storage and Processing –** PostgreSQL will be used for structured data management, ensuring quick and efficient handling of telemetry datasets.
- **Advanced Filtering and Analysis –** Users can filter, compare, and analyze telemetry data over different time frames, making it easier to identify trends and anomalies.
- **Secure and Scalable System** – The platform will include authentication, role-based access control, and encryption for data security, while its modular architecture ensures scalability for future enhancements.

ANALYSIS (DFDs, ER Diagrams, Database Design):

- **DFD (Data Flow Diagram):**

  Level 1: User logs in → System authenticates user → User selects CanSat data source → System retrieves and processes CanSat telemetry data → Visualization module displays interactive charts/maps.

  Level 2: Breaks down each step, including User Authentication, Data Retrieval, Data Processing, Data Visualization, User Actions.
- **ER Diagram:** Defines relationships between entities (Users, CanSats, Telemetry Data, Visualizations, Reports).
- **Database Design:** Users, CanSats, Telemetry Data, Visualizations, Reports, Logs.

PROJECT MODULES

1. **User Authentication & Role Management**
2. **CanSat Data Acquisition Module**
3. **Data Processing & Storage Module**
4. **Real-Time Data Visualization & UI/UX Module**
5. **Data Filtering & Analysis Module**

1. User Authentication & Role Management

Purpose: Ensures secure access and role-based permissions, allowing authorized users to manage and monitor CanSat telemetry data.

- Secure login and registration using encrypted passwords.
- Role-based access control (RBAC) for different user types (Admin, viewer).
- Token-based authentication using **JWT (JSON Web Tokens)**.
- Account recovery options (forgot password, email verification).

2. CanSat Data Acquisition Module

Purpose: Collects real-time telemetry data from the CanSat or simulated sources, ensuring accurate data reception and validation.

- Receives **real-time telemetry data** from the CanSat through wireless transmission or a simulated source.
- Supports multiple data sources, including **pre-recorded datasets for testing**.
- Parses raw telemetry packets (e.g., altitude, temperature, pressure, GPS coordinates).
- Ensures **error handling** and validation to filter out corrupted or missing data.

3. Data Processing & Storage Module

Purpose: Processes raw telemetry data, removes errors, and stores it in a structured database for easy retrieval and analysis.

- Processes **raw telemetry data** and converts it into structured formats.
- Stores telemetry readings in a **PostgreSQL database** for future analysis.
- Implements **data validation** (removing duplicates, handling outliers, and ensuring integrity).
- Prepares structured datasets for efficient visualization and retrieval.

4. Real-Time Data Visualization & UI/UX Module

Purpose: Provides an interactive and user-friendly interface with dynamic charts, 3D visualizations, and real-time data updates.

- Displays telemetry data using **interactive graphs, charts, and maps** in real time.
- Uses **Chart.js, D3.js, and Leaflet.js** to create dynamic and engaging visualizations.
- Integrates a **3D visualization** feature (using **Three.js**) for tracking CanSat movement.
- Ensures a **user-friendly web interface** built with **React.js**, providing smooth navigation and real-time updates.
- Supports **customizable dashboards**, allowing users to personalize their visualization preferences.

5. Data Filtering & Analysis Module

Purpose: Enables users to filter, compare, and analyze telemetry data to identify trends, anomalies, and performance insights.

- Enables users to filter telemetry data based on **time range, sensor type, and thresholds**.
- Supports **trend analysis, anomaly detection, and comparative analysis** for insights.
- Provides **real-time and historical data comparison** to study CanSat performance.
- Enhances data-driven decision-making through **custom reports and detailed analytics**.

HARDWARE AND SOFTWARE REQUIREMENTS

LIBRARIES:

- **React.js** → For building the front-end user interface, ensuring a dynamic and interactive Ground Control Station.
- **Node.js & Express.js** → For backend development, API handling, and server-side processing of telemetry data.
- **PostgreSQL & Sequelize** → For database management, ensuring efficient storage and retrieval of telemetry data.
- **D3.js / Chart.js** → For real-time data visualization, including telemetry charts and graphical representations of sensor data.
- **Three.js** → For 3D visualization of the CanSat trajectory, altitude variations, and environmental simulations.
- **Leaflet.js / Google Maps API** → For geospatial mapping and real-time tracking of CanSat location via GPS telemetry.
- **Socket.io** → For real-time data updates using WebSocket communication between CanSat, server, and dashboard.
- **MQTT.js** → For handling telemetry data transmission using MQTT protocol, ensuring reliable data reception.
- **Moment.js / Date-fns** → For handling timestamps, time-based calculations, and historical data analysis.
- **Multer** → For handling file uploads, such as CanSat logs or telemetry data reports.
- **Winston / Morgan** → For logging and monitoring backend processes for debugging and security.
- **Helmet.js** → For securing HTTP headers, protecting against common security vulnerabilities.

HARDWARE REQUIREMENTS:

Processor	**:** Intel i5 (10th Gen) / AMD Ryzen 5 or higher
RAM	**:** 8 GB DDR4
Storage	**:** 256GB SSD or 500GB HDD
Graphics	**:** Integrated Graphics (Intel UHD / AMD Vega)
Display	**:** 1366x768 resolution or higher
Internet Connection	**:** Required for real-time telemetry simulation and cloud integration

SOFTWARE REQUIREMENTS:

Operating System	: Windows 10 or Higher / Linux (Ubuntu 20.04+)
Programming Language	**:** JavaScript (ES6+), Node.js (LTS version)
Front-End Framework	: React.js (Latest version)
Back-End Framework	**:** Node.js with Express.js
IDE	: Visual Studio Code (VS Code)
Visualization Libraries	:D3.js / Chart.js / Three.js (for interactive data visualization)
Version Control	: Git & GitHub
Mapping Tools	**:**Leaflet.js / Google Maps API (for geospatial visualization)
Package Manager	**:**npm / yarn (for dependency management)
API & Real-Time Communication	**:**RESTful API with Express.js, WebSockets (Socket.io)
Database	: PostgreSQL with Sequelize ORM (for telemetry data storage and management)

LIMITATIONS OF THE PROJECT

- **Dependency on Physical Hardware** – The system requires a functional CanSat and a properly connected ground station. Any hardware failure can disrupt data collection.
- **Serial Communication Limitations** – The data transmission speed and stability depend on the serial port's baud rate and potential buffer overflows, which may cause data loss.
- **Limited Mobility** – Since the data is received via a wired connection, the ground station setup must remain in a fixed location, reducing flexibility in field operations.
- **Hardware Compatibility Issues** – Different CanSat models and sensor configurations may require additional integration work to ensure seamless data reception and processing.
- **Data Processing Overhead** – High-frequency telemetry updates may create processing delays, especially when handling large amounts of real-time data.

- **Power & Connection Reliability** – Any interruptions in power supply or loose serial connections can cause data transmission failures, affecting real-time monitoring.

FUTURE SCOPE & ENHANCEMENTS

- **Integration with Real CanSat Hardware –** The system can be upgraded to receive live telemetry from actual CanSat missions instead of simulated data.
- **Advanced Data Analytics & AI Integration** – Machine learning models can be incorporated for anomaly detection, trend prediction, and automated insights.
- **Mobile Application Development –** A dedicated Android/iOS app can be developed for real-time monitoring and alerts on the go.
- **Cloud-Based Deployment –** Migrating to cloud platforms like AWS or Google Cloud can enhance scalability, storage, and real-time data processing.
- **Multi-Satellite Support –** The system can be expanded to track multiple CanSats or even larger satellites simultaneously.
- **Offline Data Storage & Analysis –** Features can be added to allow users to download telemetry data and analyze it offline.

CONCLUSION

The **Ground Control Station for CanSats** is a comprehensive and interactive platform designed to monitor and visualize CanSat telemetry data efficiently. By leveraging modern web technologies such as **React.js, Node.js, PostgreSQL, and WebSockets**, the system provides a seamless, real-time, and user-friendly experience for satellite data monitoring. Through dynamic visualizations, geospatial mapping, and real-time updates, users can analyze critical telemetry parameters such as temperature, altitude, pressure, and location with ease.

One of the key advantages of this system is its ability to function without physical hardware by utilizing simulated telemetry data. This allows researchers, students, and space enthusiasts to test and analyze satellite data in a controlled environment before deploying actual hardware. The dashboard's interactive features, including data filtering, historical analysis, and customizable reports, enhance its usability for both educational and research purposes. Additionally, the modular design ensures scalability, allowing future enhancements such as **AI-based data analysis, mobile app integration, and real CanSat hardware support**.

Despite its limitations, such as the dependency on simulated data and network connectivity, the system lays a strong foundation for future development in satellite telemetry monitoring. With continued improvements, including cloud deployment and multi-satellite support, this project has the potential to serve as a powerful tool for aerospace research, CubeSat missions, and educational programs. By bridging the gap between raw telemetry and meaningful insights, the **Ground Control Station for CanSats** provides a **scalable, efficient, and accessible solution** for satellite data visualization and analysis.