LATIN AMERICAN CUBESAT WORKSHOP
SALVADOR, BR
OCTOBER 2024

Development of an automated application to collect and filter CubeSat's data

Caique Sales Miranda Gomes*[1], Gabriel Mariano Marcelino[2], and João Cláudio Elsen Barcellos[2]

[1]Department of Mechanical Engineering, Federal University of Santa Catarina
[2]Department of Electrical and Electronics engineering, Federal University of Santa Catarina

Abstract

In recent years, the utilization of CubeSats has proliferated across diverse institutions worldwide, marking a transformative shift in space exploration. However, despite their widespread adoption, in low-cost missions there remains a conspicuous lack of comprehensive information and tracking accessibility for the broader community beyond traditional channels like Ham Radio "amateur radio". The NanoSatTrack project serves as a platform using MERN stack (MongoDB, Express, React, and Node.js) with the primary goal of making data generated by nanosatellites accessible to the general public in real time. It achieves this by offering a website equipped with interactive dashboards. These dashboards play a vital role in presenting and visualizing the key telemetry and information derived from the satellite's mission interface. In essence, NanoSatTrack enables anyone to conveniently access and understand the valuable data produced by nanosatellites. It is responsible for creating a dashboard showing the communication between the Earth (a ground station) and a satellite, that are divided in multiple telecommunication links. This project allows for future CubeSat missions to better show their project's achievements and help them ensure an easier approach to its collected data.

1 Introduction

The satellite industry has been working towards collecting and filtering data for a long time now, some of the biggest challenges with satellite data are volume, velocity, and variety [1] . In the big industry, there are a few programs that work to make up-to-date data accessible to users around the world as free,

*Email: kiqsmg@gmail.com

some examples are Copernicus Open Access Hub, NASA Earthdata, and Google Earth Engine. Although this practice has been fairly common in the satellite industry, little has been shown of how this concept can be implemented for CubeSats, in the cases that has been implemented, those data platforms have helped not only the developer team but also ham radios and students of many majors to have a easier access to satellite data [2].

The constant advance of small satellites has opened up space to a wider range of applications and a broader audience, especially academic institutions, and universities, enabling them to conduct space research with reduced financial investment [3].

Several technical obstacles still need to be resolved when developing these missions. One of the main difficulties is creating data volume [1], for some CubeSat missions, this is mostly done by gathering data from multiple ground stations around the globe. The velocity at which the data is been collected and stored is also a key component to have a real time mission platform. Collecting this high volume of data and keeping the database updated at a fast pace is challenge on its own ,to do in smaller missions with less budget is even harder due to less ground station segment investments. Another significant challenge with the data stored from CubeSat missions is the lack of a platform to make the data obtained easily accessible, with the data already filtered to be in the best way possible with graphs and grouping of relevant information, although the difficulty, the Alfa Crux mission has created a platform to help their team analyse and keep track of their CubeSat while orbiting the Earth [2] . This is crucial for mission operators, the ham radio community, and the general public who can later on use the data obtained during the satellite's mission.

As it is extremely complicated to gather enough data to build a consolidated database to show enough info from one single orbit cycle, it is also hard to connect all the data sent directly from the satellite to Earth, this makes it difficult to create an updated database without having a connection between ground stations. One solution found by many institutions is to gather data not only from the mission's main ground station but also from ham radio operators around the world who have linked with the CubeSat while its orbit was above their station [4] . This partnership is very beneficial to the mission operators but not very reliable since each ground station only has a short period to connect with the satellite. Additionally, the ham radio collaborators around the world can take some time to send back the data, and this system is not very organized [4].

The FloripaSat-1 and FloripaSat-2 platforms, which were and will be used in nanosatellite missions such as FloripaSat-1, GOLDS-UFSC, Catarina-A1, and Catarina-A3, among others, currently lack a platform to display the data collected during these missions. To address this problem, the NanoSatTracker project was created.

This project aims to filter the data collected from CubeSats and present it in a clear and accessible way, making it easier to monitor the satellite's health status throughout each orbit cycle, as the mission control group obtains data from the missions main ground station, and ham radio partners around the world. Additionally, it seeks to simplify the work of engineers by organizing the information into subsystems, enabling a more efficient analysis of component performance. This approach also helps anticipate and address potential challenges in future missions.

The network of ham radios that help collecting and sending the CubeSat data is voluntary, so there is no guarantee of data besides the missions main ground station, this challenge is faced by others satellite missions around the world [4] . Although we can't fully solve this problem, NanoSatTracker has a side project which sole purpose is to automate the process of sending the data obtained by the link from a ground station with a CubeSat. By installing code in the ground station's computer, when it receives a CubeSat beacon data, it will redirect the data to a database in the cloud, feeding the NanoSatTracker almost automatically.

Therefore, the main proposal is to develop a data tracking platform for small satellite missions. This platform will allow mission operators to have a better overview of how the system is working in orbit without having to wait for days or even weeks to collect the data needed to run tests and perform basic check-ups on the nanosatellite system. This paper is organized as follows: the methodology applied by other authors will be adapted and elaborated upon in the Methodology section, where the key components will be presented in detail. Further discussions on the project's development are provided in the Discussion section, followed by the Conclusion.

2 Methodology

The methodology adopted for the NanoSatTracker platform development is structured into 7 stages: 1) a comprehensive literature review, 2) analysis of the FloripaSat-1 mission data, 3) selection of the most valuable data for the developers and community, 4) Client side and server side diagram, 5) Server side development, 6) Client side development, 7) Verification and validation of the platform. The project is in-between stages 6 and 7, the development stages can be seen in Fig.1.

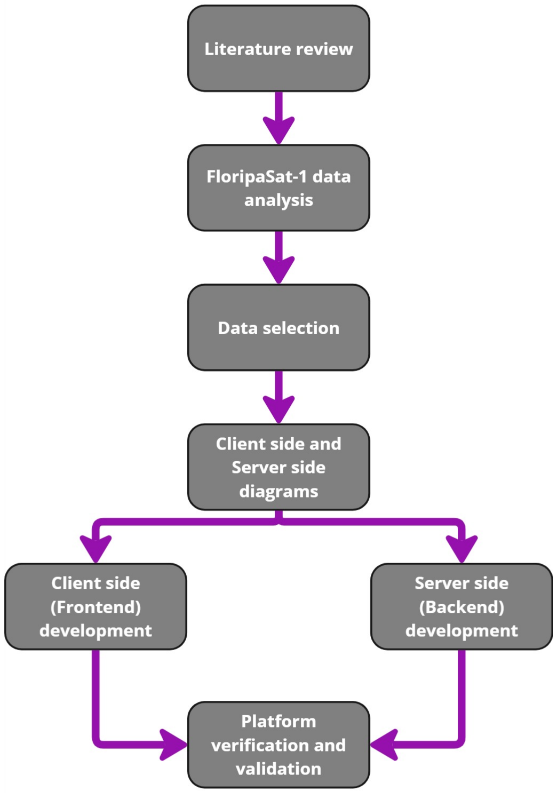

Figure 1: FloripaSat-1 platform

2.1 Comprehensive literature review

An extensive literature review was first conducted, focusing on existing projects from other institutions to identify best practices, innovative approaches, and the challenges encountered in similar developments. This research is divided into two sections: traditional large satellite missions and CubeSat missions. This division is necessary because these two distinct project models face different challenges and typically require different approaches to problem-solving. Traditional large satellites have a variety of ways to collect, filter, and display their data to users, as can be seen in projects like NASA's Earth Science Data System (ESDS) Program, Earth Observing System (EOS) Program [5], and projects like The Copernicus App Lab project [1], where they have an abundance of data due to many connections with the satellite and a more robust data system. Unlike traditional large satellite missions, CubeSat projects typically operate with smaller teams, making it difficult to allocate developers specifically to the creation of a platform for displaying their data, the FloripaSat-1 is an example of this [6]. By searching for more information we could understand better the 5 main challenges of collecting data for satellite missions (volume, velocity, variety, veracity, and value) [1], and then focus on solving them for our model.

2.2 Analysis of the FloripaSat-1 data

In this section, we will go through the basic subsystems of the FloripaSat-1 platform that are important for the CubeSat mission, so we can have a better understanding of where the mission's data comes from. The OBDH (On-Board Data Handling Module), the EPS (Electrical power system module), and TT&C (Telemetry, Tracking & Command). For its simplicity, data available, and user-friendly display of information, this work will be centered around the FloripaSat-1 platform (1U CubeSat) designed and built by the Space Systems Laboratory (Spacelab) at the Federal University of Santa Catarina. The FloripaSat-1 platform is shown in in Fig.2.

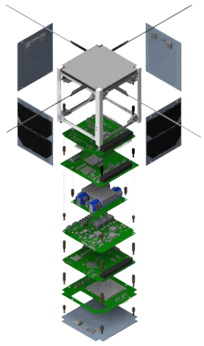

Figure 2: FloripaSat-1 platform

The On-Board Data Handling Module (OBDH) main goal is to manage all the CubeSat information. The OBDH2 is responsible for synchronizing actions and the data flow between other modules (ie. EPS, Payloads) and the Earth segment. Some of the most important satellite data comes from de OBDH, such as the date, hour, temperature sensors, and much more that tell about how the system is handling the orbit cycle. The OBDH data is one of the most important data collected since it describes the health status of the CubeSat [7]. The Electrical and Power System (EPS) collects data from different sensors around the EPS PCB and also the battery PCB. This module not only is one of the most complexes in the satellite but also is one with a large number of sensors regarding temperature and energy, it counts with electrical current and voltage measurements in the module and also in all solar panels, temperature sensors in the module and some more in the batteries PCB that is attached to the main PCB. Although all data is important, the most significant data for the battery's health and duty cycle are the OVERALL temperatures, the voltage and current in the solar panels, and in the EPS [7]. The Telemetry, Tracking & Telecommand (TT&C) is responsible for making the communication between the earth (a ground station) and a satellite and is divided into two sub-modules: Beacon and downlink/uplink., which is responsible for capturing data from Earth (ground stations) and transmitting it to the TT&C. Since the most common communications are the three mentioned above, as the NanoSatTracker is in it's Beta version, the most simplified data set was chosen to be displayed in the platform, the Beacon, as it can be seen in Table 1.

Column	Content	Column	Content
0	Year (UTC)	21	RESERVED
1	Month (UTC)	22	RESERVED
2	Day (UTC)	23	RESERVED
3	Hour (UTC)	24	RESERVED
4	Minute (UTC)	25	RESERVED
5	Second (UTC)	26	RESERVED
6	Battery cell 1 voltage	27	RESERVED
7	Battery cell 2 voltage	28	RESERVED
8	Battery monitor temp.[°C]	29	RESERVED
9	Battery current in A	30	RESERVED
10	Battery charge in Ah	31	RESERVED
11	Solar Panel -Y current in [A]	32	RESERVED
12	Solar Panel +X current in [A]	33	RESERVED
13	Solar Panel -X current in [A]	34	RESERVED
14	Solar Panel +Z current in [A]	35	RESERVED
15	Solar Panel -Z current in [A]	36	EPS temp.[°C]
16	Solar Panel +Y current in [A]	37	SatNOGS observation
17	Solar Panel -Y +X voltage	38	Callsign
18	Solar Panel -X +Z voltage	39	Grid locator
19	Solar Panel -Z +Y voltage		
20	Energy Level [1ex]		

Table 1: Beacon Data table.

2.3 Selection of the most valuable data for the developers and community

Based on previous Beacon information we received from the FloripaSat-1 mission regarding the main subsystems we gathered a total of the 20 content information shown in Table 1, and another 19 different content information from the OBDH and mission payload. The organization of the database (CSV files) is described in Fig.3, as it can be seen most of the information regards the EPS subsystem, although there is still space RESERVED to upcoming modules to send other data.

Column	Content
0	Year (UTC)
1	Month (UTC)
2	Day (UTC)
3	Hour (UTC)
4	Minute (UTC)
5	Second (UTC)
6	Battery cell 1 voltage
7	Battery cell 2 voltage
8	Battery monitor temperature in Celsius
9	Battery current in A
10	Battery charge in Ah
11	Solar panel -Y current in A
12	Solar panel +X current in A
13	Solar panel -X current in A
14	Solar panel +Z current in A
15	Solar panel -Z current in A
16	Solar panel +Y current in A
17	Solar panel -Y +X voltage
18	Solar panel -X +Z voltage
19	Solar panel -Z +Y voltage
20	Energy level
21	RESERVED
22	RESERVED
23	RESERVED
24	RESERVED
25	RESERVED
26	RESERVED
27	RESERVED
28	RESERVED
29	RESERVED
30	RESERVED
31	RESERVED
32	RESERVED
33	RESERVED
34	RESERVED
35	RESERVED
36	EPS microcontroller temperature in Celsius
37	SatNOGS observation
38	Callsign
39	Grid locator

Figure 3: Beacon data CSV file

2.3.1 Database Design

To decide which database model would be best for this project, we focused on the Beacon data package. There were a couple of options to choose for the NanosatTracker database, the first decision we made was to go with a NoSQL database for a couple of reasons.

- Flexibility NoSQL databases allow for schema-less data storage, making it easier to remove fields without the need for a major schema migration, since the structure of the database is changing along with the evolution of the CubeSat project and its needs to collect different information that might change over time.
- Scalability The NoSQL databases are easy to work with large volumes of data, making it a good choice to store hundreds of thousands of Beacon data packages throughout the months of the CubeSat mission.
- Simplicity SQL databases are great for working with complex data structure schemas, but for the NanosatTracker we use only one simple schema that is fetched when needed, so it makes sense to go with a simpler and more practical NoSQL database in this case.

2.4 Client side and server side diagram

To design the client-side and server-side diagrams for the NanoSatTracker application, we structured the platform around the data itself. The server-side architecture was specifically developed based on the data structure provided by the Beacon file, ensuring an intuitive and data-driven design for both sides of the application. Given the high volume of data and the lack of need for a structured SQL database, the most intuitive approach was to use the MERN stack to develop both the client-side and server-side of the application. The client side diagram can be seen in Fig.6, and the server side diagram can be seen in Fig.5.

2.4.1 MERN Stack

Since the database solution was chosen to be a NoSQL, it was natural to decide to use a Stack that works well with this database, the chosen stack was the MERN (MongoDB, Express.js, React, Node.js).

- MongoDB MongoDB is a NoSQL database that stores data in JSON-like documents. The database will store all the CubeSat's Beacon data related to the mission, such as date and time information, temperature sensors, logs, metrics, and other RESERVED content. Its flexibility and simplicity allow for easy changes to the data structure as the project evolves.
- Express.JS Express is a fast and minimalist web framework for Node.js that supports web and mobile applications, APIs, and middleware. The Express framework will be used to create the backend logic and structure for the project. It will handle the server-side logic, process requests from the user side (frontend), interact with the MongoDB database, and send the appropriate responses to the fetched data.
- React React is the library for web and native user interfaces. React will be used

to build the user side (frontend) of the NanosatTracker project. It will enable to create a dynamic and interactive user interface where not only the operators of the project will have access to the data collected but also any person who has the web page link will be able to manage data, view dashboards, and interact with various components. Reacts User Interface will facilitate to create a responsive screen size for many devices, like cell phones and tablets.

- Node.JS Node.js is an open-source server environment that runs on various platforms and uses JavaScript on the server. Node.js will serve as the platform to run the server-side part of the Nanosat-Tracker project. It will execute the JavaScript code written in Express.js, manage server operations, and handle asynchronous events. This will ensure that the server can handle multiple user requests efficiently, making the application simple and scalable.

3 Implementation

Through all the studies previously carried out, the MERN stack was chosen. To execute the projects plan we started first building the database, then the server side of the application (backend), and lastly the client side (frontend). Throughout the project evolution some changes were made, not only in the database format but also in the way the backend worked fetching data from the MongoDB and the frontend displayed the data, always trying to make the logic simpler and more user-friendly to the future users of the platform.

3.1 Database

The database structure was chosen based on simplicity and flexibility, so the beta model result included all the content the beacon was sending to the ground station and 4 more information, the ID of the element, the name of the CubeSat mission, and two date/time information about the creation and update of the element. The beta model data structure can be seen in Fig.4.

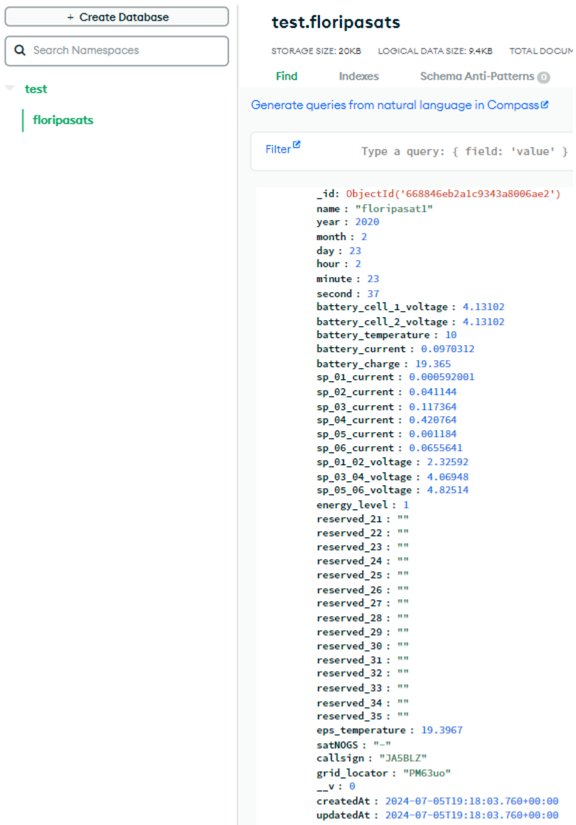

Figure 4: MongoDB data structure.

3.2 Server side (backend)

The server side of the web application was structured using Express and Node.JS. The structure is divided into 4 main sub-areas, Controllers, Models, Routes, and Data(only used in Beta testing model). The server division can be seen in Fig.5.

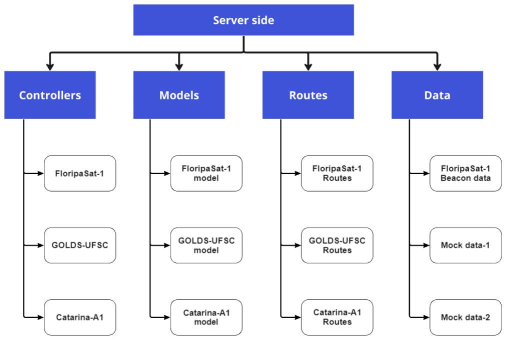

Figure 5: Server side structure.

3.2.1 Controllers

The controller's code will define the logic that handles incoming HTTP requests. Controllers usually act as an intermediary between the data models (data layer) and the views (UI/Client layer). They get the incoming requests and then process them, perform the necessary operations such as querying the database or processing data, and then send the corresponding responses back to the client.

3.2.2 Models

The model's code is used to define the structure of the data that will be stored in the MongoDB database. The model is created using Mongoose, which is an Object Data Modeling (ODM) library for MongoDB and Node.js. The schema was defined using the already existing beacon data from the FloripaSat-1 and then analyzed the best solution to model the application data in a way that facilitates query data fetch from the web application.

3.2.3 Routes

The Route's code on the server creates the endpoints that the server will respond to. It specifies the URLs that the server will handle and also maps these URLs to the corresponding controller functions, which contain the logic to process each HTTP request from the client side.

3.2.4 Data

The Data's code is basically mock data used in the Beta testing model for the first interactions while the ground stations do not yet feed the database with the CubeSat beacon data.

3.3 Client side (frontend)

The client side of the web application was structured using the React framework for developing of the user interface(UI). The client side of the application is responsible for not only displaying the (UI), but also handling the user's interaction, inputs, and events. The frontend structure is divided as seen in Fig.6.

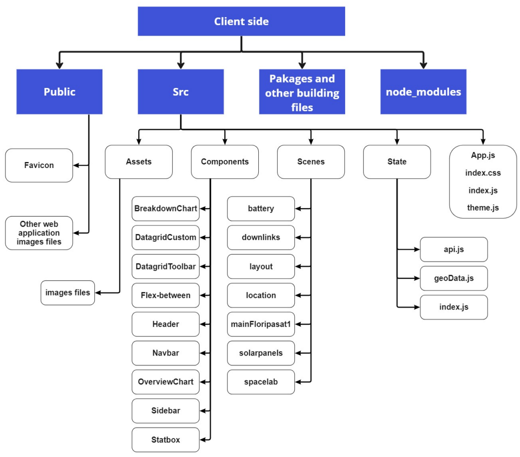

Figure 6: Client side structure.

4 Result

The result of this project was a Beta model of the NanosatTracker web application, it consists of a web platform that is divided into 6 pages dedicated to different aspects of the satellite, it was decided to use the FloripaSat-1 CubeSat as an example for the Beta model as for it's easy access to the nanosatellite data and information throughout the mission, all the data and information is presented in GitHub(https://github.com/floripasat). The reason we divided the satellite beacon data into 6 different pages was that they would be better arranged and it would be better for analysis during and after the mission to see the data separated in this way rather than mixed together, thus ensuring that the data that is presented together has high relevance to each other, and can be compared. The 6 pages created are Spacelab, FloripaSat-1, Downlink, Battery, Solarpanel, and Location. Each of the 6 pages has a main focus as we will see later.

4.1 Overview of the NanosatTracker platform

NanoSatTrack serves as a platform with the primary goal of making data generated by nanosatellites accessible to the general public. It achieves this by offering a website equipped with interactive dashboards. These dashboards play a vital role in presenting and visualizing the key telemetry and information derived from the satellite's mission interface. In essence, NanoSatTrack enables anyone to conveniently access and understand the valuable data produced by nanosatellites. It is responsible for creating a dashboard showing the communication between the earth (a ground station) and a satellite, that are divided in two sub-modules: Beacon and downlink/uplink.

4.2 Spacelab page

The Spacelab page is the main page of the web application, it is also the simplest out of the 6 pages. Its only goal is to give the user an overview of the Spacelab laboratory projects and what will be shown on the next pages. An image of the Beta version of the Spacelab page can be seen in Fig.7.

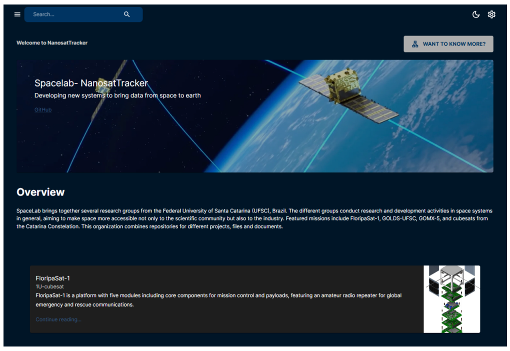

Figure 7: Spacelab page.

4.3 FloripaSat-1 Main page

The FloripaSat-1 Main page is a page dedicated to the FloripaSat-1 mission, it will give a brief explanation of the payloads and subsystems inside the CubeSat. An image of the Beta version of the FloripaSat-1 Main page can be seen in Fig.8.

4.4 Downlink page

The Downlink page has the goal of displaying all the collected data from the FloripaSat1 beacon in a simple and organized table, arranged per date of received data. It enables

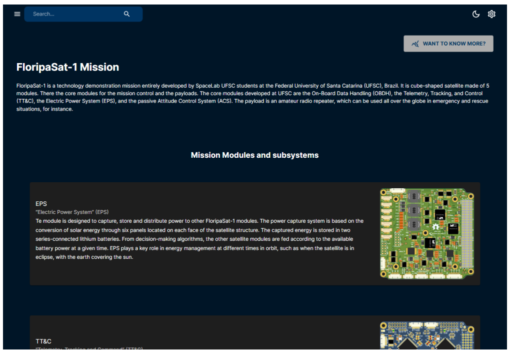

Figure 8: FloripaSat-1 Main page.

the team behind the CubeSat mission to see the data collected in a simpler and more organized way. An image of the Beta version of the Downlink page can be seen in Fig.9.

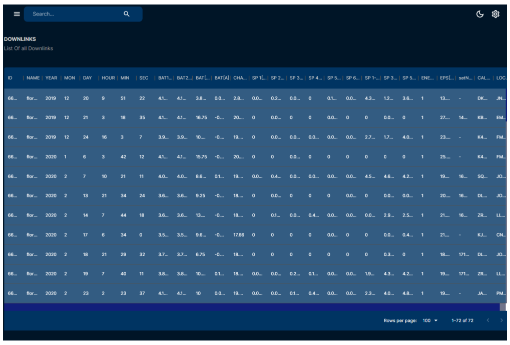

Figure 9: Downlink page.

4.5 Battery page

The Battery page is where we can find the data related to the Energy Power System. The three most important data we collected from the EPS were the temperature around different elements in this system, the voltage, and the current. Since the FloripaSat-1 Cubesat had only two battery cells, we decided to fetch the voltage data from each cell and plot a graph to compare those two. Fetching the temperatures of the system we also created another graph to display the temperatures of the EPS's PCB and the batteries' PCB. Lastly, we created a graph for the battery current and another for the battery charge levels. We decided to structure the page in a way that when the graphs are displayed on the screen they will have the same size and collected data pattern so the user can analyze the graphs simultaneously. An image of the Beta version of the Battery page can be seen in Fig.10.

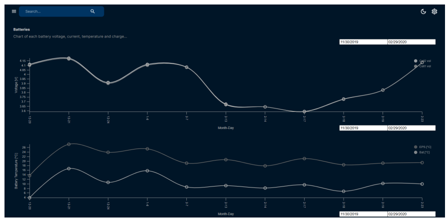

Figure 10: Battery page.

4.6 Solar Panels page

The Solar Panels page is the page dedicated to the data collected from the solar panels. Here, the data fetched from the project database is displayed in two different graphs, one of the graphs focuses on the current of all 6 solar panels, making it possible to see the different solar emissions each panel receives at a given time. The other graph is dedicated to each solar panel section's voltage, the 6 solar panels system is divided into 3 sets of 2 solar panels, making three different voltage values for each set. An image of the Beta version of the Solar Panels page can be seen in Fig.11.

4.7 Locations page

The Locations page fetches the data responsible for the ground station location information of each downlink connection made with

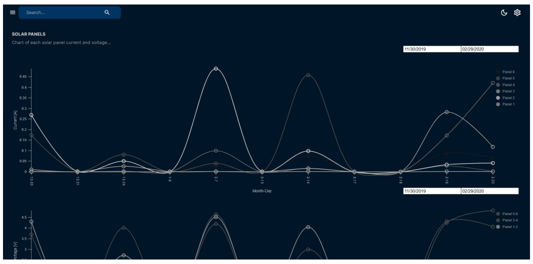

Figure 11: Solar Panels page.

the CubeSat around the world. After gathering the data we process it to plot a map that allows us to see all the countries that made contact with the Cubesat while in orbit. We only obtained this much data thanks to several Ram radio operators and partner institutions around the world. An image of the Beta version of the Locations page can be seen in Fig.12.

Figure 12: Locations page.

5 Discussion: trends and directions

After completing the Beta version of the project, we presented it for analysis to technical members of the FloripaSat-1 and FloripaSat-2 engineering and development teams. We received many positive feedbacks regarding the display of information that was previously unavailable to the group. The data presented in NanoSatTracker was considered crucial for analyzing not only past missions but also for predicting future situations regarding how the CubeSat might operate in orbit, thereby aiding the evolution of future projects. The possibility of simultaneously updating the page was also considered and will be the next step to make the web application even more useful to CubeSat teams. To achieve a platform with a fast data update rate we will need a strong collaboration of different ham radio operators, and laboratories around the world to help collect the CubeSat data and feed our database to have a more reliable and efficient data system to rely on. The ground station code segment of the NanoSatTracker is the solution to this current problem, so that's where our energy will be directed for the next phase of this project.

6 Conclusion

In conclusion, the NanoSatTracker has achieved success in solving the critical gap in the accessibility and visibility of CubeSat telemetry data. Using the MERN stack, the project provides a user-friendly platform that democratizes access to real-time satellite data, enabling not only the satellite mission team but also the broader community to engage with and understand CubeSat missions more effectively. The positive feedback received from the Spacelab team (FloripaSat-1, and FloripaSat-2) underscores the value of this platform, particularly in enhancing the analysis of past missions and aiding in the prediction of future operational scenarios. Now, the focus will shift to enhancing the platform's real-time data update capabilities, which will require close collaboration with ham radio operators and laboratories worldwide. By refining the ground station code and bolstering our data collection infrastructure, NanoSatTracker will become an even more indispensable tool for CubeSat teams, driving the evolution of future space missions.

Acknowledgements

The author wants to thank not only the Spacelab laboratory team but also Professor Eduardo Bezerra for all the encouragement given throughout the project. A huge thanks to Laura Beatris, your support and mentoring were essential for the creation of NanoSatTracker.

The authors want to to acknowledge that this work was supported by CNPq/Brazil (National Council for Scientific and Technological Development - http://www.cnpq.br), and by AEB (Brazilian Space Agency - http://www.aeb.gov.br).

References

[1] K. Bereta, H. Caumont, U. Daniels, E. Goor, M. Koubarakis, D.-A. Pantazi, G. Stamoulis, S. Ubels, V. Venus, and F. Wahyudi, "The copernicus app lab project: Easy access to copernicus data," 2019.

[2] R. A. Borges, A. C. dos Santos, W. R. Silva, L. Aguayo, G. A. Borges, M. M. Karam, R. B. de Sousa, B. F.-A. García, V. M. d. S. Botelho, J. M. Fernández-Carrillo, J. M. Lago Agra, F. A. Agelet, J. V. Q. S. Borges, A. C. A. de Oliveira, B. T. de Mello, Y. d. C. F. Avelino, V. F. Modesto, and E. C. Brenag, "The alfacrux cubesat mission description and early results," *Applied Sciences (Basel, Switzerland)*, vol. 12, no. 19, p. 9764, 2022.

[3] N. Saeed, A. Elzanaty, H. Almorad, H. Dahrouj, T. Y. Al-Naffouri, and M.-S. Alouini, "Cubesat communications: Recent advances and future challenges," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 1839–1862, 2020.

[4] J. P. Mason, T. N. Woods, P. C. Chamberlin, A. Jones, R. Kohnert, B. Schwab, R. Sewell, A. Caspi, C. S. Moore, S. Palo, S. C. Solomon, and H. Warren, "Minxss-2 cubesat mission overview: Improvements from the successful minxss-1 mission," *Advances in Space Research: The Official Journal of the Committee on Space Research (COSPAR)*, vol. 66, no. 1, pp. 3–9, 2020.

[5] H. Ramapriyan, "The role and evolution of nasa's earth science data systems," no. GSFC-E-DAA-TN24713, 2015.

[6] G. M. Marcelino, S. Vega-Martinez, L. O. Seman, L. Kessler Slongo, and E. A. Bezerra, "A critical embedded system challenge: The floripasat-1 mission," *IEEE Latin America Transactions*, vol. 18, no. 02, pp. 249–256, 2020.

[7] G. M. Marcelino, A. M. P. de Mattos, J. C. E. Barcellos, B. F. Ribeiro, L. O. Seman, E. M. Filho, and E. A. Bezerra, "Floripasat-2: An open-source platform for cubesats," *IEEE Embedded Systems Letters*, vol. 16, no. 1, pp. 77–80, 2024.

References

[1] K. Bereta, H. Caumont, U. Daniels, E. Goor, M. Koubarakis, D.-A. Pantazi, G. Stamoulis, S. Ubels, V. Venus, and F. Wahyudi, "The copernicus app lab project: Easy access to copernicus data," 2019.

[2] R. A. Borges, A. C. dos Santos, W. R. Silva, L. Aguayo, G. A. Borges, M. M. Karam, R. B. de Sousa, B. F.-A. García, V. M. d. S. Botelho, J. M. Fernández-Carrillo, J. M. Lago Agra, F. A. Agelet, J. V. Q. S. Borges, A. C. A. de Oliveira, B. T. de Mello, Y. d. C. F. Avelino, V. F. Modesto, and E. C. Brenag, "The alfacrux cubesat mission description and early results," *Applied Sciences (Basel, Switzerland)*, vol. 12, no. 19, p. 9764, 2022.

[3] N. Saeed, A. Elzanaty, H. Almorad, H. Dahrouj, T. Y. Al-Naffouri, and M.-S. Alouini, "Cubesat communications: Recent advances and future challenges," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 1839–1862, 2020.

[4] J. P. Mason, T. N. Woods, P. C. Chamberlin, A. Jones, R. Kohnert, B. Schwab, R. Sewell, A. Caspi, C. S. Moore, S. Palo, S. C. Solomon, and H. Warren, "Minxss-2 cubesat mission overview: Improvements from the successful minxss-1 mission," *Advances in Space Research: The Official Journal of the Committee on Space Research (COSPAR)*, vol. 66, no. 1, pp. 3–9, 2020.

[5] H. Ramapriyan, "The role and evolution of nasa's earth science data systems," no. GSFC-E-DAA-TN24713, 2015.

[6] G. M. Marcelino, S. Vega-Martinez, L. O. Seman, L. Kessler Slongo, and E. A. Bezerra, "A critical embedded system challenge: The floripasat-1 mission," *IEEE Latin America Transactions*, vol. 18, no. 02, pp. 249–256, 2020.

[7] G. M. Marcelino, A. M. P. de Mattos, J. C. E. Barcellos, B. F. Ribeiro, L. O. Seman, E. M. Filho, and E. A. Bezerra, "Floripasat-2: An open-source platform for cubesats," *IEEE Embedded Systems Letters*, vol. 16, no. 1, pp. 77–80, 2024.