

# Starlink Tracker

Daniel Yun, Mahin Miah, Humaid Mustajab

Git : <https://gitlab.cci.drexel.edu/mm4845/starlink-tracker>

---

## Project Description

A project designed to collect and display data [i.e. latitude, longitude, elevation, and speed] of individual satellites from the Starlink satellite constellation. The satellites are grouped by their launch dates. Users have the option of visualizing this data in a detailed table format or on a globe map. In the map visualization, users can select a single satellite to see its XYZ coordinates as well as its corresponding XYZ velocities. The map also includes a feature to track that satellite's path.

## Technologies

Our website/code structure has 2 main functions that are accessed through the 2 buttons displayed on initial page load: **'Visualize Starlink in Space'** and **'Get Satellite Orbital Data'** [\[Graphic 1\]](#).

**Visualize Starlink in Space:** Onclick, this button brings up an expanded side-bar with a clickable list of Starlink launch dates [\[Graphic 2\]](#). It also adds a toggle button to the left of the 'Visualize Starlink in Space' button which can be clicked to toggle the state of the sidebar. An iframe is also added to the page view and renders a 3d model of Earth using the ArcGIS API and custom parameters. Clicking on any of the launch dates listed in the sidebar gets satellite tracking data from the n2yo.com API in TLE format which is parsed on our server. The map parsed data is then sent to the mapping API which renders out the satellites on the globe [\[Graphic 3\]](#). The user can then click each satellite shown on the globe bringing up additional information about the satellite including its Name, NORAD ID, xyz position vector and the xyz velocity vector.

---

---

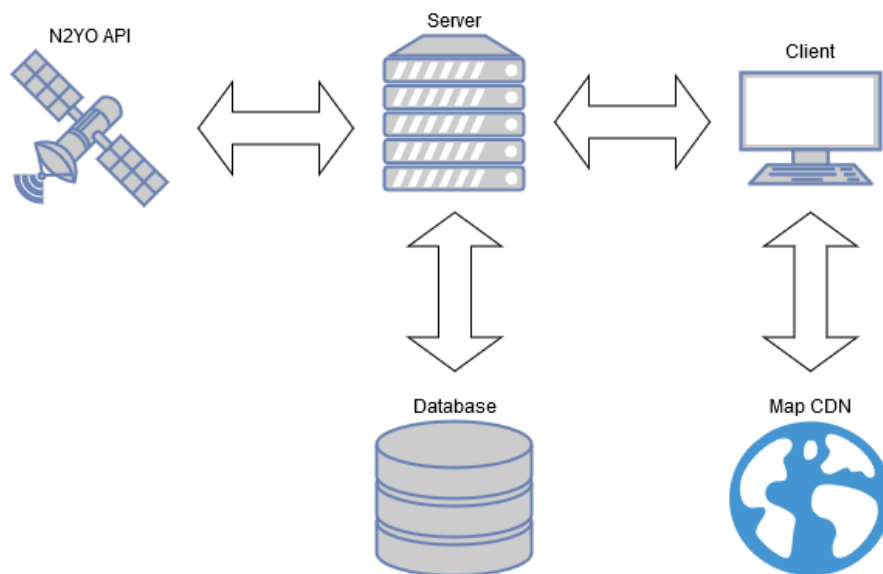
The popup menu also has 2 clickable targets **'Track Satellite Path'** and **'Zoom to'**. On-click, the **'Track Satellite Path'** button renders a line on the globe that shows the calculated/expected path of the satellite around the globe [\[Graphic 4\]](#). On-click, the **'Zoom to'** button pulls the selected satellite into focus.

**Get Satellite Orbital Data:** Onclick, this button clears up the webpage and brings up an interface built in [React.js and Material-UI](#) [\[Graphic 6\]](#). Clicking on the dropdown labelled 'Launch Group' pops open a menu with the Starlink launch date groups [\[Graphic 7\]](#). Clicking on one of the launch groups refreshes the table with data from the n2yo API [\[Graphic 8\]](#). The data includes the Satellite Name, Satellite NORAD ID, XYZ position, and XYZ velocity of the satellite. The table is paginated with custom row options and a sticky header. If the user clicks on the down-arrow icon in each row to the left, it opens up a collapsed table with tabled data on the future track of the individual satellite [\[Graphic 10\]](#). The **'Visualize Satellite Positions'** button also gets enabled when the user selects an item in the dropdown and can be clicked to visualise the launch group on a map. The map includes all of the same functionality as introduced earlier [\[Graphic 9\]](#).

---

## Application Architecture

The application starts by serving a static html page; the application's main page. The server can receive a request for data on a group of satellites from the client, and sends that request to the N2YO API. The API then responds with data on that group and, after the server parses through it, the client receives it. The client can then visualize the data either in map form, through the ArcGIS CDN, or table form, through React. The user may also create an account at any time through the main page. This information is stored in the PostgreSQL database connected to the server.



**Graphic 5: Diagram of the Application's Architecture**

---

## Reflection

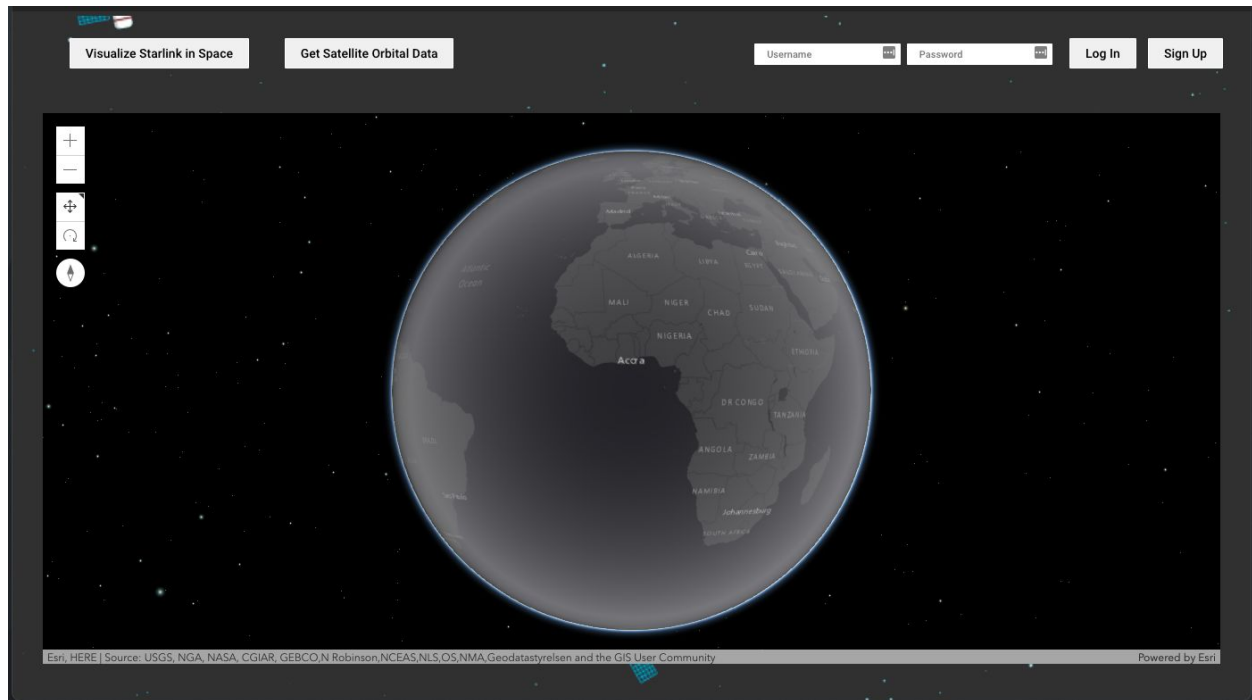
### Features

Our original plan for this project was to visual the satellite data in a detailed table format. We planned to pull the data from the API and display it with features that would help sort the information. However, we actually ended up implementing both a table and map option to show the information we pulled from the API. Later on, we also made the decision to show the Starlink satellites by their launch dates. The late addition of a map and React introduced problems that required more time than we had previously planned for.

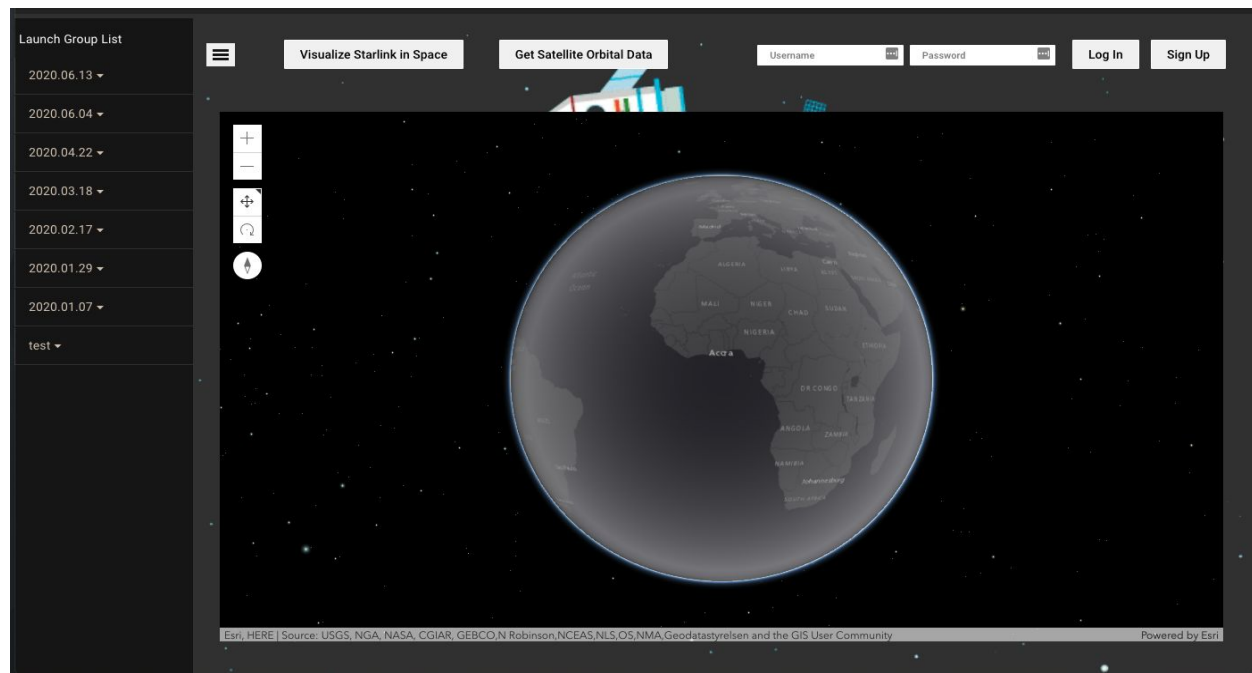
### Grievances

Our biggest obstacle was managing how our data was flowing throughout the project. Since we were working with multiple libraries, an API, and a CDN, it became harder and harder to manage where items were being held or initialized. It took the team additional time to figure out where and how the data was being managed. There were also plans of integrating and overlapping certain functions; i.e, when a satellite in the sidebar's dropdown was clicked, that satellite's path would be shown on the map. If we were to do this project again, we would begin with a stricter plan and layout within our code to help keep track of the API information throughout the project.

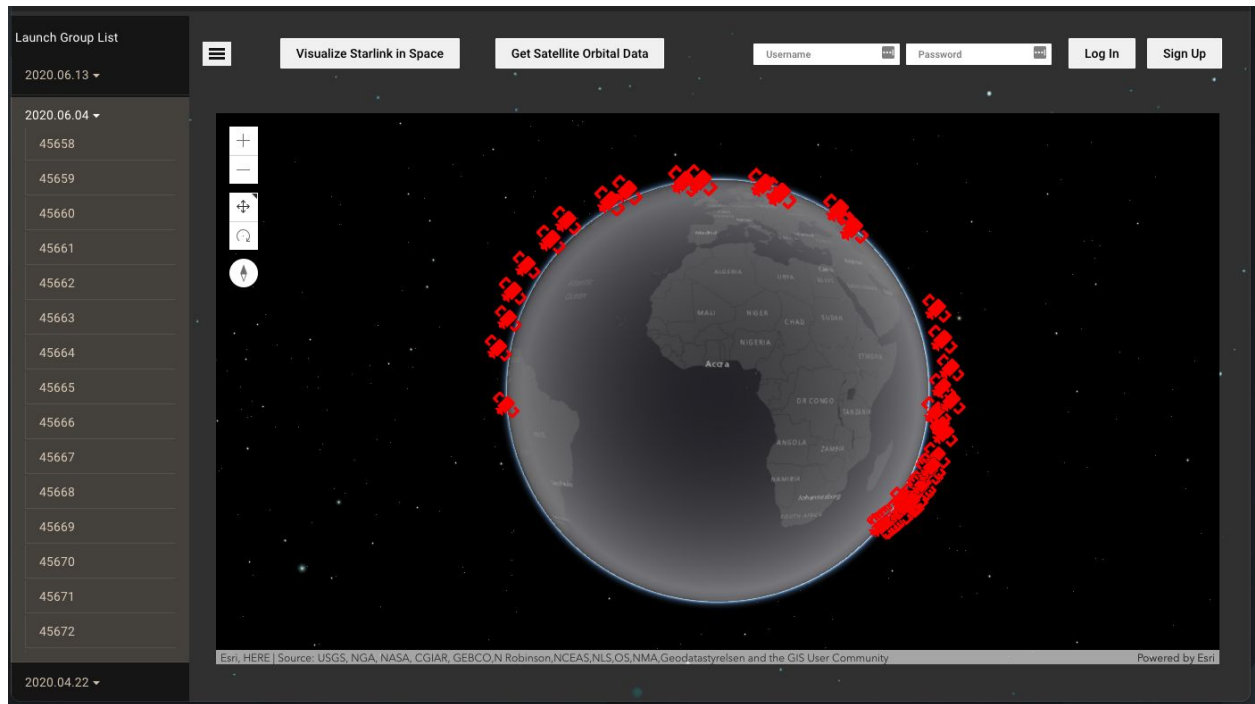
## Addendum



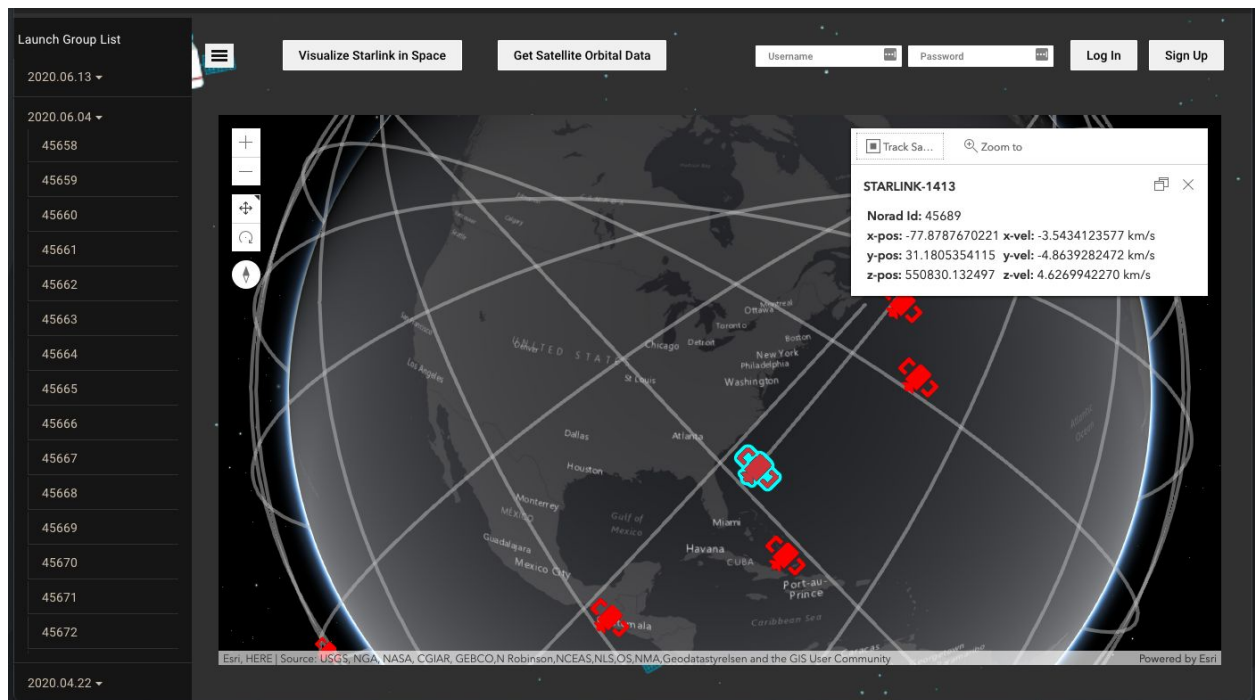
Graphic 1: Initial page load



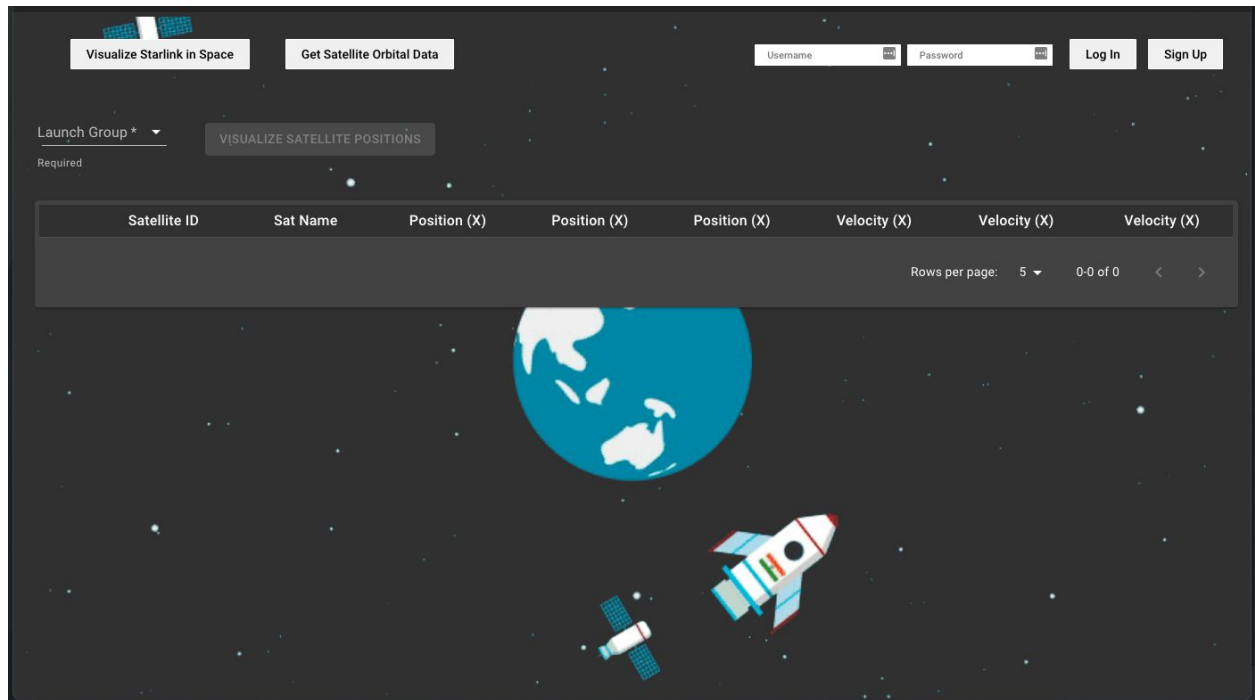
Graphic 2: Website view after clicking 'Track satellite in Space' button



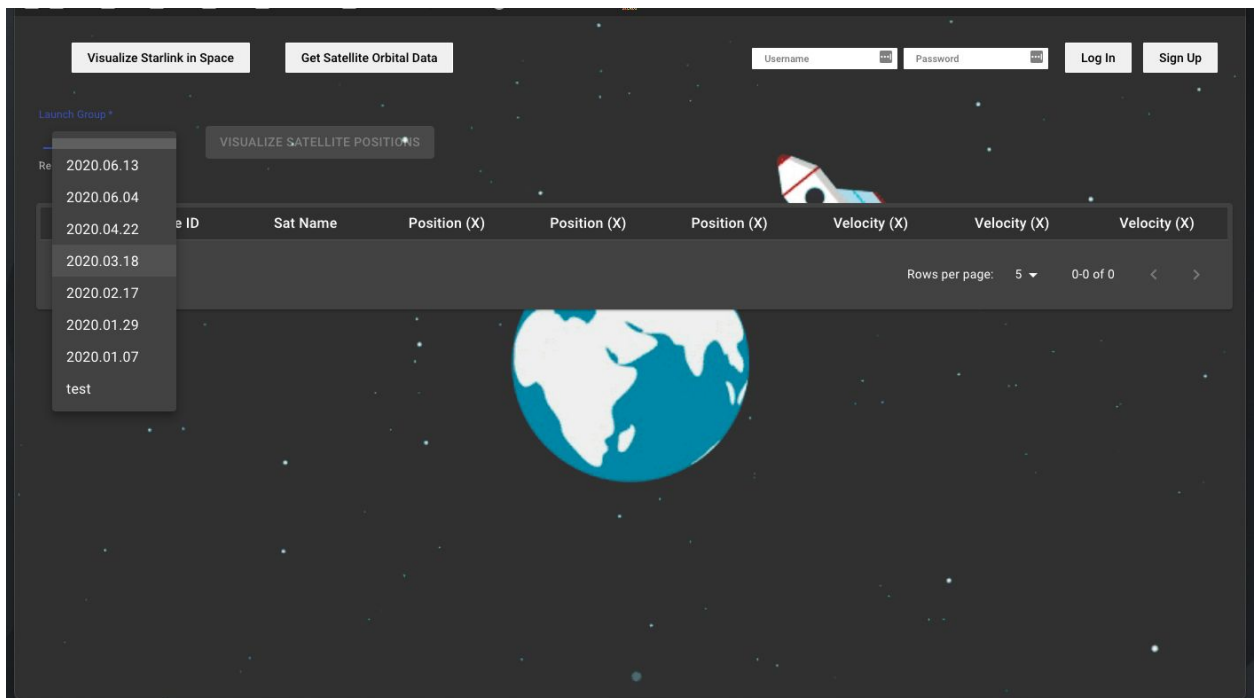
**Graphic 3: Website view after tracking launch group '2020.06.13'**



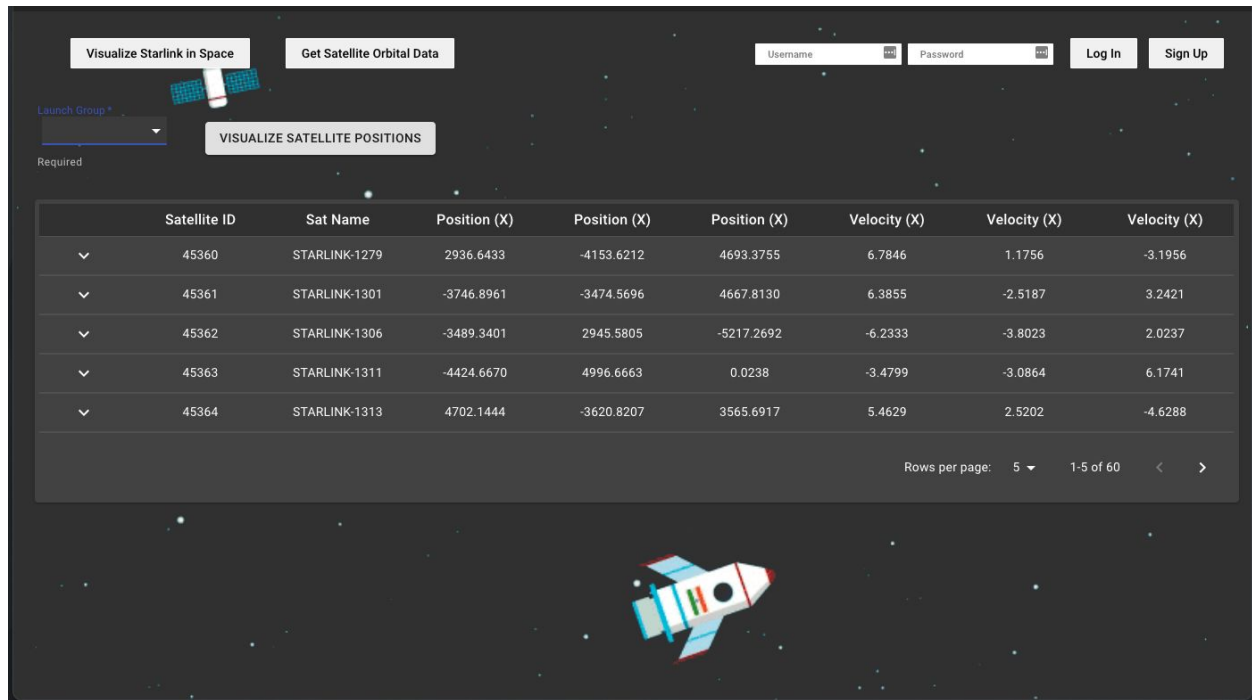
**Graphic 4: Website view after clicking 'Track Satellite Path' button**



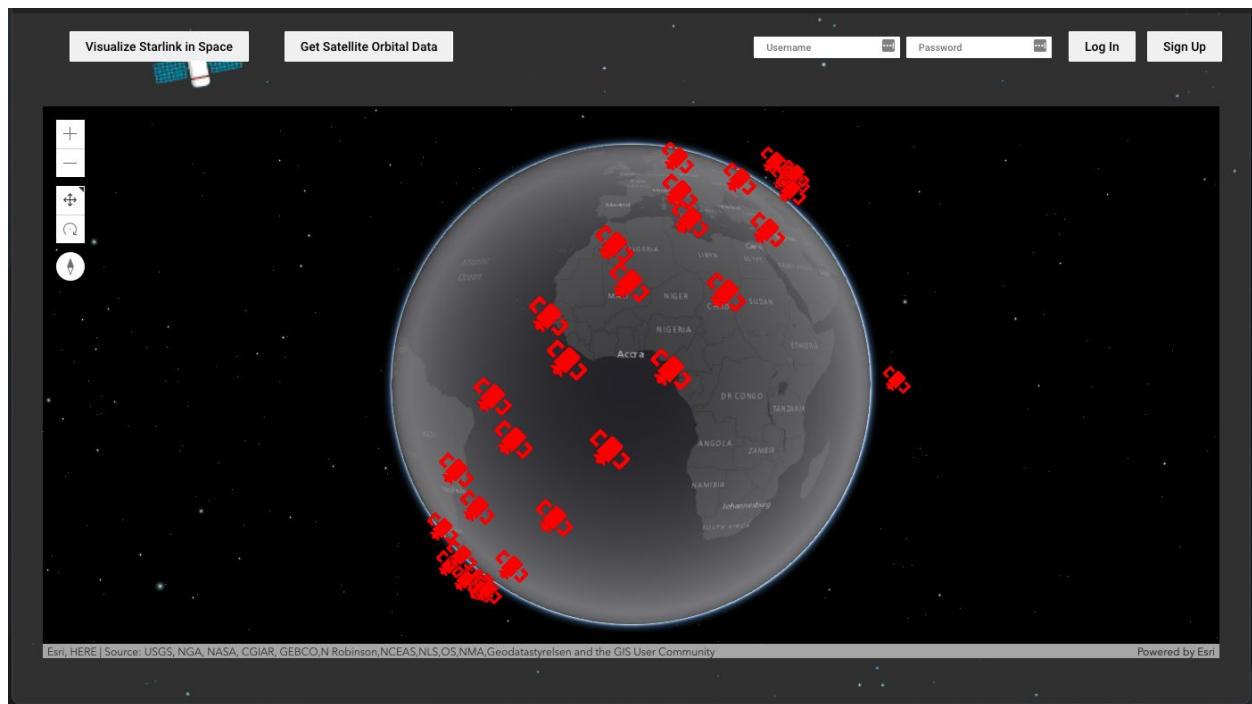
**Graphic 6: Website view after clicking 'Get Satellite Orbital Data' button**



**Graphic 7: Website view on clicking the 'Launch Group' dropdown**



**Graphic 8: Website view on selecting an item in 'Launch Group' dropdown**





**Graphic 9: Website view on clicking the 'Visualise Satellite Positions' button**



Launch Group \*  
Required

VISUALIZE SATELLITE POSITIONS

	Satellite ID	Sat Name	Position (X)	Position (X)	Position (X)	Velocity (X)	Velocity (X)	Velocity (X)
^	45360	STARLINK-1279	2936.6433	-4153.6212	4693.3755	6.7846	1.1756	-3.1956
Orbit Tracking								
	Time Stamp	Latitude	Longitude		Altitude		Elevation	
	1599258065	-22.90907803	46.51011183		556.81		-20.28	
	1599258066	-22.95676	46.55032368		556.83		-20.31	
	1599258067	-23.00442911	46.59056645		556.84		-20.34	
	1599258068	-23.05208535	46.63084021		556.86		-20.37	
	1599258069	-23.09973058	46.67114668		556.88		-20.41	
	1599258070	-23.14735901	46.71148108		556.9		-20.44	
	1599258071	-23.19497829	46.75184998		556.91		-20.47	
	1599258072	-23.24258261	46.7922486		556.93		-20.5	
	1599258073	-23.29017385	46.83267864		556.95		-20.53	
	1599258074	-23.33775199	46.8731402		556.97		-20.56	
v	45361	STARLINK-1301	-3746.8961	-3474.5696	4667.8130	6.3855	-2.5187	3.2421
v	45362	STARLINK-1306	-3489.3401	2945.5805	-5217.2692	-6.2333	-3.8023	2.0237
v	45363	STARLINK-1311	-4424.6670	4996.6663	0.0238	-3.4799	-3.0864	6.1741

Graphic 10: Website view on clicking the down-arrow button in the first table column