Anthony Petrosino

Collaboration log and work summary

Nav Menu and Homepage
(NavMenu.razor, NavMenu.razor.css, Home.razor, Home.razor.css)

I made the homepage to be a hub with info on all SpaceTracker features. It has interactive cards linking to dashboards, astronauts, spacecraft, launches, space news and programs.

The main navigation bar is visible across all pages, displaying a dropdown menu system with four categories (Dashboards, Explore, Events, Site Info) using Blazor's NavLink component for client side routing. I used AI to help me make custom CSS for dropdown styling and a more responsive design to enable smooth navigation without page reloads.

Space News
(spaceNewsModel.cs, spaceNewsService.cs, SpaceNews.razor, SpaceNews.razor.css)

The space news webpage displays paginated space news articles from multiple sources using a three-layer architecture that is used throughout our project: spaceNewsModel.cs defines the data structure for articles, spaceNewsService.cs handles API calls to fetch and process the data, and SpaceNews.razor renders the webpage with a scrolling news source banner and the option to load additional articles. This pattern helped me understand the difference between data models, logic services, and presentation components in Blazor. I consulted documentation from SNAPI (Space News API) to better understand how to interact with its endpoints.

*** 2d Dashboard ***
(Dasboard2D.razor, Dasboard2D.razor.css)

The 2D Dashboard shows upcoming launch locations on an interactive world map with live weather data. The goal was to show the next 5 launches as clickable markers on a world map image using their coordinates. Users are able to select different launches to view detailed mission information and current weather conditions at each launch site. It integrates SpaceDevsService for launch data and weatherService for weather info, providing a lightweight alternative to the 3D dashboard. I also did this page before the 3D dashboard and it helped me get an idea as to how to go about implementing it.

*** 3d Dashboard ***
(Dashboard3D.razor, Dashboard3D.razor.css, cesiumService.cs, cesium-interop.js)

The 3D Dashboard is an interactive Cesium globe displaying live ISS tracking and upcoming launch sites. It features an interactive 3D Earth where users can click on launch markers to view detailed mission information and you can watch the ISS move across the globe every 5 seconds. The implementation required creating a C# service wrapper (cesiumService.cs) to handle JavaScript interop with the Cesium library, along with a custom JavaScript module (cesium-interop.js) for direct globe manipulation. This was the more technically challenging than the 2D dashboard as it involved coordinating between C#, JavaScript, and external mapping APIs.

The vast majority of my css came from my built-in AI editor in my IDE, Amazon Q, which greatly helped me focus on the C# aspects of my project rather than styling the frontend. I also used Amazon Q to aid in small technical problems and to provide proof of concept ideas as to how to go about the project.