COMP 3133 – Lab Test – 2 (06%)

Submission Date and Time: 02nd April 2025, 08:30 PM (Week – 13)

Objective: Create an Angular application to implement the following features.

Instructions:

1. Create the Angular Application: (10 points)

- Create an Angular app named `studentid-lab-test2-comp3133`. ***e.g. 1012355-lab-test2-comp3133***
- Initialize a GitHub repository and commit all your code to it.

2. Host the Application: (10 points)

- Deploy your Angular application on a hosting platform (such as Vercel, Railway, Render, Docker, etc.).

3. Create `MissionList` Component: (20 points)

- Create a component named `missionlist` to display a list of all SpaceX launches using the provided REST API endpoint.
  Endpoint: SpaceX Launches API
- Display the following fields for each mission:
  - `flight_number`
  - `mission_name`
  - `launch_year`
  - `details`
  - `mission_patch_small`
  - `rocket`
    1. `rocket_name`
    2. `rocket_type`
  - `links`
    1. `mission_patch_small`
    2. `article_link`
    3. `Wikipedia`
    4. `video_link`

4. Implement Search or Filter by Launch Year: (10 points)

- Create a `missionfilter` component that allows users to search/filter missions by their year of launch.
  Endpoint: SpaceX Launch Filter API

5. Create `MissionDetails` Component: (20 points)

- Create a component named `missiondetails` to display detailed information about a selected mission from the `missionlist` component.
  Endpoint: SpaceX Mission Details API
- Hint: Pass data to the `missiondetails` component using `@Input()` or use Route parameter.

6. Create Service to Fetch Data: (10 points)

- Create a service to fetch data from the provided SpaceX REST API.

7. **Create Interface/Class for Data Structure: (10 points)**
   - Create an interface or class to define the structure of the data fetched from the SpaceX API.

8. **Use Angular Material for Design: (10 points)**
   - Use Angular Material components to enhance the design and user interface of the application.

Additional Optional Instructions (for using GraphQL API):

- **Optional**: If you prefer to use a GraphQL API instead of the REST API, you can integrate the SpaceX GraphQL API into your Angular application.
  - SpaceX GraphQL API: SpaceX GraphQL

  In this case, you will need to:

  - Set up a GraphQL client (e.g., Apollo Client) in your Angular project.
  - Modify the components and services to query the SpaceX GraphQL API instead of the REST API.
  - Update the data fetching and filtering logic to match the structure of the GraphQL API responses.

This option is **not mandatory**, but it may help you learn about integrating GraphQL with Angular.

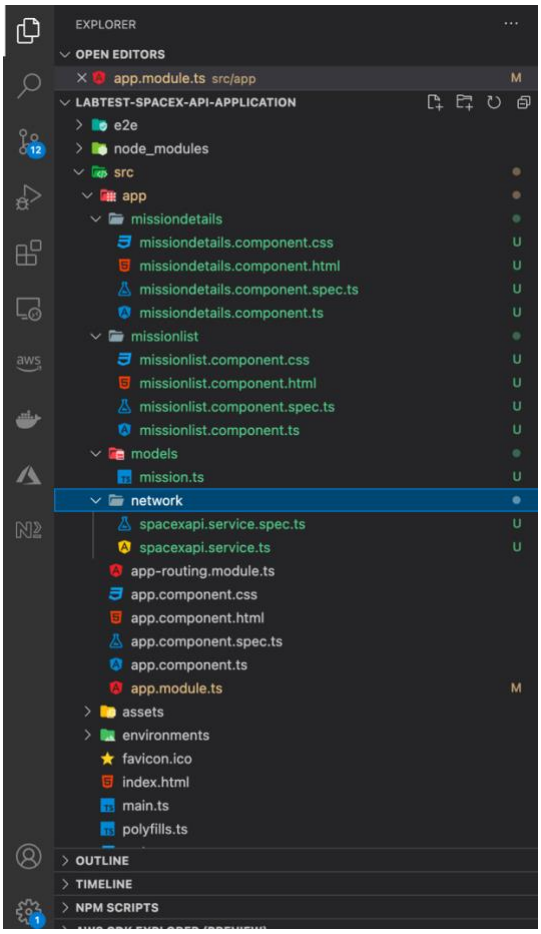
Submission Requirements:

1. **Upload the Source Code**:
   - Upload a ZIP file containing your source code to Blackboard before the deadline.
2. **Provide GitHub Repository Link**:
   - In the comments section of your submission, include the link to your GitHub repository.
3. **Screenshots**:
   - Take screenshots of both pages of your application and upload them to D2L to show evidence of your work outside ZIP file.
4. **Hosting Link**:
   - Provide the link to your hosted application (on Render, Vercel, Railway, docker or another platform).

References for SpaceX API:

- SpaceX REST API Documentation
- **Optional**: SpaceX GraphQL API Documentation

Folder Structure of application



Sample Screen

Figure 1 Mission List

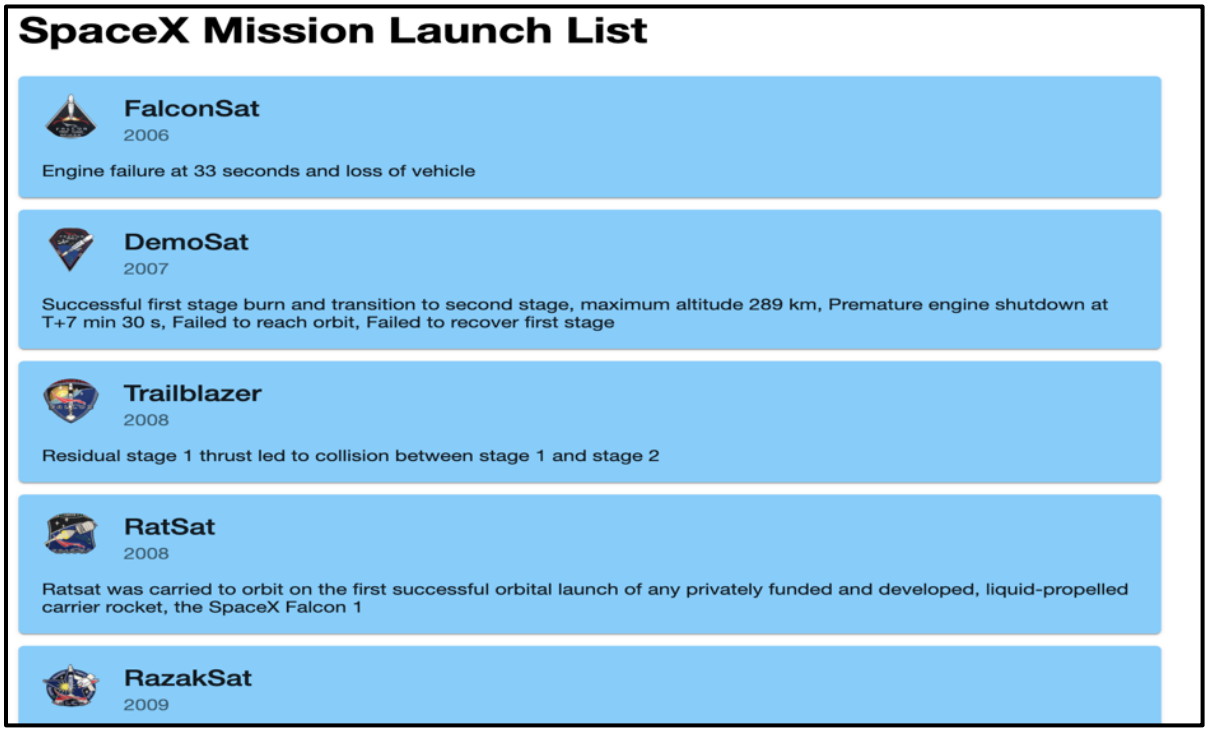


Figure 2 Mission filter

Figure 3 Mission Details