Angular Lab 2 (A whole new Angular Project that does everything)

This is an opportunity to build an entire Angular front-end project. If you like, take this time to begin building the front-end for your team's back-end API. Even if the back end is not finalized, you can build the front-end structure with components your team might need. You can also write Angular services to interact with your back-end API

Lab 2 Option A

If you'd like to build a complete Angular App that makes use of an existing (non-financial) API, try the following:

1. Generate a new Angular project called 'swapi-proj' with no routing
2. You can write all the content within app.component.html or you can generate components as you wish
3. Generate a service called 'swapi'
4. Inside the service, write a method to retrieve data from
   https://swapi.dev/cat/id
   where cat is one of 'people, planets, species' and id is a positive number
5. In your app provide a way for users to choose a category and an id (e.g. form fields)
6. When the user clicks a button, call your service method to retrieve the category/id data
7. Show the returned data nicely

Lab 2 Option B

If you want to practice with an Angular Project that is closer to your final project work, try the following:

1. Generate a new Angular project with a name you choose and no routing
2. Generate new components such as:
   a. add-new-data
   b. net-worth
   c. live-charting (NB on this course we do not cover live web charts)
3. If you like, copy in the 'head' and 'foot' components from previous work, but make sure they are properly imported and referenced in NgModule
4. Place instances of all these components inside app.component.html
   At some point, you will have to think about the layout of the components with CSS (e.g. Flexbox or Bootstrap)
5. Use the 'add-new-data' component to let users enter new data into form fields. Maybe use the sort of data you will need in your final project, or choose any data structure you like
6. New data should be appended to a data structure, i.e. a property of this component
7. Use the 'net-worth' component to show data from an existing data structure that you need to write in this components class
8. Just put a static image in the 'live-charting' component
9. If your back-end API is ready, generate an Angular Service then write methods to interact with your back-end API
10. Use your service in the other components you have created