Backend API

If database in needed:

1. Create the database and add the desired starting data
2. Create a yml file that contains the credentials to log into the database

You will first need to create the Webservice. This begins with the creation of a new maven project.

1. Next, you will need to edit the pom.xml file. Add the needed dependencies and maven plugins. This includes the plugin that allows an application to run as a Spring Boot
2. Create a main method, then import the bootstrapping to allow you to call SpringApplication.run();
3. Add the application.yml file and fill it out to allow for connection to the database.
4. Create a SecurityConfiguration class.
5. Create the needed models.
6. Create the Spring Data repository that corresponds to the models
7. Create the DTOs (Data Transfer Objects) so that object data has a place to go to be sent to the database.
8. Create the Service Layer. These will contain all the methods for models and controllers to use.
9. Create corresponding service classes for the appropriate models.
10. Begin creation of the controllers for corresponding models and app services (such as a Session Controller for checking out items in a car)
11. Add CORS support to allow for flexible browser support for Angular 7 if needed. Crea a CORS Configuration class and place in the config directory.

Angular Side

1. Create the Angular project through the command window
2. Refactor the sample page as needed
3. In addition, refactor the back end to pass in the necessary values that Angular will be reading
4. Create the models needed for the class that correspond to the models that you created in the API
5. Create a repository typescript file. Make sure it can pull a JSON form the specified address.
6. Create a module file.
7. In the app module class, add the necessary components into the declarations as well as the imports.
8. Within the components and their html, display the data from the JSON. Decorate the page as needed.

RESTful API (Project)

1. Create the project
2. Create the main class that will run the application.
3. Establish dependencies within the pom.xml file
4. Create the database
5. Implement credentials and connection data in the application.yml file
6. Create entities for Games, their Developers, and their Ratings
7. Create repositories for corresponding entities.
8. Create the controllers for the corresponding entities
9. Create the appropriate mapping methods

Angular Client (Project)

1. Create the models that correspond to the ones in the backend API
2. Create the repository class.
3. In the repository class, set the constructor up so that it can connect to the address that is used by the backend API.
4. Create the components for different parts of the site (Navigation menu and game list)
5. Import the classes of the models into the components.
6. For the Nav menu, design the html template as shown the mockup
7. For the game list, do the same.
8. Implement the button clicks for the navigation buttons. After that, you should have a list that can be filtered by genre.