Code a movie database site  
----  
   
Technologies used: ASP.NET Core, Vue (or Angular), Foundation 6 CSS Framework (or Bootstrap)  
   
Make it an SPA (Single Page Application). Data is fetched from server API in JSON format and bound to Vue templates (if Vue is used).  
Views are styled using Foundation6 - responsive CSS framework (if Foundation6 is used).  
   
Design the application incrementally. Start by creting read-only responsive views of movie list and details and  
data API for read-only access.  
   
   
Iteration1  
   
Views  
----  
   
1. List of movies:  
  - Display title, category, year, rating  
  - Add search to search movies by title. Do simple search first and then see if this could be done as autocomplete  
  - Add multi select control to filter movies by certain category. One should be able to filter movies by multiple categories at the same time.   
     
   
2. Movie details  
   - Display movie title, year, rating and description  
   
  
Data and business logic.  
---  
   
1. Create classes for Movie [id (int), title, year (int), description, rating (int), category id], Movies list and Category (id, name)   
2. Create movies Repository class:   
    - method to returned fixed (hard coded) list of movies  
    - method to return movie by id (from the hard coded list)  
3. Create movie service class:   
   - list of movies (return list of movies from Repository class method)  
   - details of selected movie (return from movie repository by id)  
5. Create MVC Controller for returning list of movies and movie details (using movie service class) in json format:  
   - API method to return list of movies  
   - API method to return movie details  
   
Tests  
---  
 Create tests for movie service methods: list of movies and movie details  
   
  
The reason for having repository and service classes is for easy replacement of repository implementation in further iterations  
and making it possible to keep all business logic in service layer independent of data.