Movie Rating System

Objective:
Create a system to rate movies, built using JavaScript ES6 classes, encapsulating related data and behavior, demonstrating inheritance, and showcasing polymorphism.

Exercise 1: The Media Superclass

1.1 Define a `Media` class. The constructor should accept two parameters, `title` and `duration`.

1.2 Inside the constructor, initialize private instance variables for `title`, `duration`, and `ratings`. `ratings` should be an empty array for collecting multiple ratings for a media object. Use the `#` prefix to denote private properties.

1.3 Implement a getter for `title` and `duration` to allow read access to these properties.

1.4 Add a method named `addRating` which accepts a parameter `rating`. This method should add the received `rating` to the private `ratings` array.

1.5 Implement a method to calculate the average rating of the media. If there are no ratings, it should return 'No ratings yet'. This method should not be directly accessible from outside the class (make it a private method and create a getter to expose the calculated average rating).

1.6 Create a public method `displayDetails` that returns a string containing the media's title, duration, and average rating. Use getters within this method to access private properties.

Exercise 2: The Movie and Series Subclasses

2.1 Create a `Movie` subclass that extends `Media`. Its constructor should accept additional parameters for `director` and `genre`. Initialize these as private properties.

2.2 Implement getters for the `director` and `genre` properties.

2.3 Override the `displayDetails` method to include the movie's `director` and `genre` along with the details provided by the superclass. Ensure you're using getters to access private properties.

2.4 Similarly, define a `Series` subclass with an additional parameter in its constructor for `seasons`. This should also be a private property.

2.5 Provide a getter for the `seasons` property.

2.6 Override `displayDetails` in `Series` to include the number of `seasons` along with the superclass details, using getters for private property access.

Exercise 3: The User Class

3.1 Define a `User` class. Its constructor should accept a `username` and initialize it as a private property.

3.2 Implement a getter for the `username`.

3.3 Initialize a private property `watchedMedia` as an empty array.

3.4 Implement an `addMedia` method that accepts a `media` object and adds it to the `watchedMedia` array.

3.5 Add a `rateMedia` method that takes a `media` object and a `rating`. It should verify if the `media` is in `watchedMedia`. If so, call the `addRating` method on the `media` object with the provided `rating`. Otherwise, return a message indicating the user can only rate watched media.

3.6 Implement a method to display all watched media, including their details. For each media in `watchedMedia`, call its `displayDetails` method and join the results into a single string.