# Implement Movie Data Transformation Function

## Description

Create a file transformers.ts into src/utils folder. Create a function named formatMovie within that file. This function will handle the transformation of API movie data to the business model Movie.

By creating a dedicated file for transformers and the formatMovie function, this task ensures a modular and organized approach to data transformation within the project.

## Acceptance Criteria:

1. File and Folder Structure:
   1. Ensure the existence of a utils folder within the src directory.
   2. Create a file named transformers.ts within the utils folder.
2. formatMovie Function:
   1. Implement a function named formatMovie within the transformers.ts file.
   2. The function should take an API movie data object as a parameter and return a business model object Movie.
   3. Map relevant fields from the API data to the corresponding fields in the business model Movie.
   4. Handle any necessary data conversions or transformations to ensure consistency.
3. Unit Testing:
   1. Write unit tests for the formatMovie function in a \_\_tests\_\_ folder within the utils directory.
   2. Test the function with various types of API movie data to ensure correct transformation.
   3. Include tests for edge cases and potential errors.

## Definition of Done:

1. The utils folder is created within the src directory.
2. The transformers.ts file is created within the utils folder.
3. The formatMovie function is implemented and successfully transforms API movie data to the business model Movie.
4. Unit tests for the formatMovie function pass successfully, covering various scenarios.
5. The code adheres to coding standards, is well-documented, and includes comments where necessary.