Full-Stack Engineer Technical Assignment

Your task is to build a Movies Management Web application using React, with a back-end powered by a small NodeJS API.

Project Requirements

Movie Management Interface:

- Display a table with the following movie details:
  - **Title**
  - **Year**
  - **Genre**
  - **Image**
- Include the ability to add, remove, and edit movies.
- Use dialogs (preferably modals) for add/edit actions.
- Implement infinite scrolling: Load the next “page” of movies automatically when scrolling down, with each page having a maximum of 15 movies.

Backend API for Movie Management:

- Create a backend API to handle the following actions:
  - Fetch movies
  - Add a movie
  - Edit a movie
  - Delete a movie
- The API should read from and write to a CSV file (attached).
- The API should support CRUD operations:
  - `GET /movies`: Fetch the list of movies (paginated).
  - `POST /movies`: Add a new movie.
  - `PUT /movies/{id}`: Edit an existing movie.
  - `DELETE /movies/{id}`: Delete a movie.
- Handle pagination on the server side, with each page having a maximum of 15 movies.

Front-End Requirements:

- The React app should interact with the backend API for all data-related operations.

- Ensure all CRUD operations are handled via API calls (using `fetchMovies()`, `addMovie()`, `deleteMovie()`, etc.).
- Use `async/await` for API interactions.
- Display loading indicators while fetching data from the API.

Bonus:

- Implement auth with JWT

What to Include:

- **README.md**: Instructions on how to run both the front-end and the backend API.
- **API Documentation**: Brief explanation of the backend API endpoints and how to interact with them.