2. Methodology:

3.1 Process:

The process starts when the user clicks the *"Add Movie"* button. Then the user is taken to the *"add"* page, where he finds an input element where he enters the movie's name. This click triggers an API call and renders the options similar to the keyword. All the options are generated with the release date after the title. (format: "*<MOVIE NAME> - <RELEASE DATE>*".)

The user selects his desired title and gets redirected to the *"edit"* page, where he adds his review and rating. And then, he saves the rating and review, and the movie gets added to the home page.

The simulation mentioned above flow is from the user's point of view. the technical aspect of what happens is given below:

First, the index page gets rendered by querying all the movies from the database. Then, if a user wants to add a new film, he clicks on the *"Add movie"* button. Next, the app redirects him to the *"/add"* path, where he finds a form of an input field and a submit button.

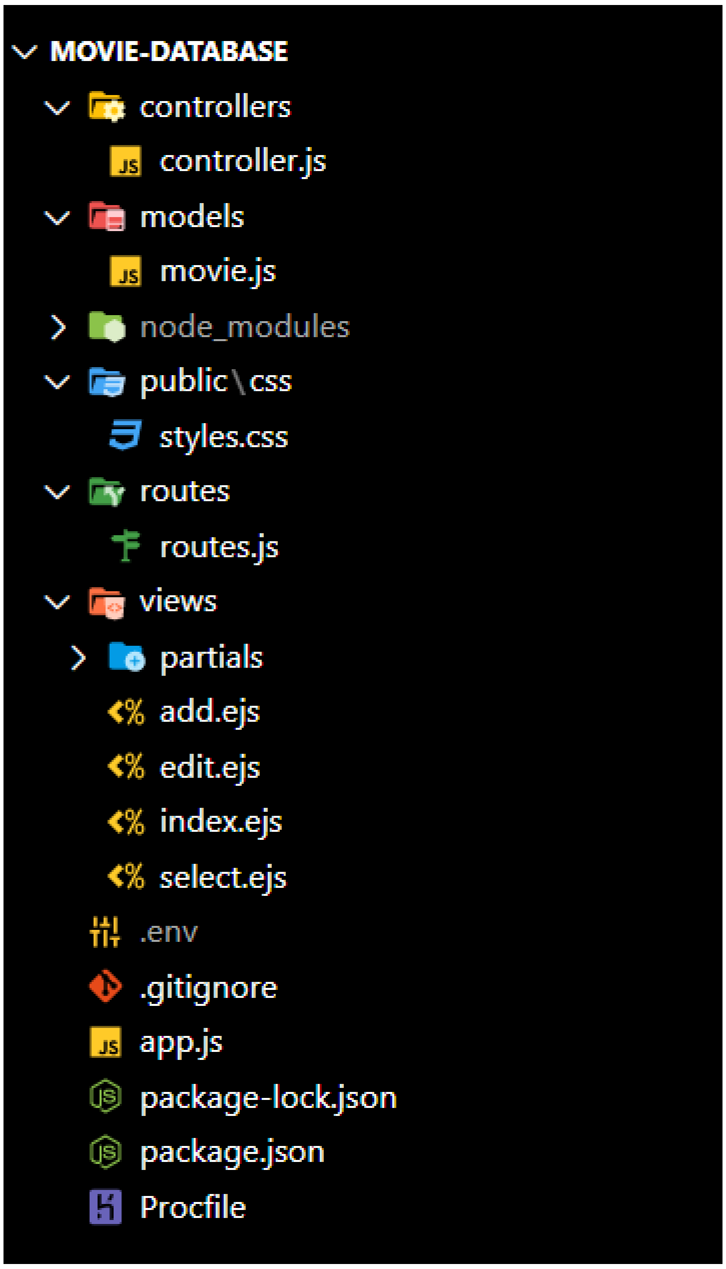
Next, the user enters a movie name and hits *"Add movie."* It triggers an API call *(GET)* which fetches all the possible matches and renders the movie titles and their release date in the format of *"<MOVIE TITLE> - <RELEASE DATE>,"* which are hyperlinks that point to the URL which leads to the edit page which makes another API call *(GET)* with the movie ID and gets the data of the movie.

Finally, the user adds the rating and review of the movie he just added, which triggers a database insertion of the film, and the movie gets stored in the database.

3.2 Technologies used:

- ExpressJS (a node framework)
- MongoDB
- EJS (templating Engine)
- Bootstrap

3.3 File Structure:



```
MOVIE-DATABASE
  controllers
    controller.js
  models
    movie.js
  node_modules
  public\css
    styles.css
  routes
    routes.js
  views
    partials
    add.ejs
    edit.ejs
    index.ejs
    select.ejs
  .env
  .gitignore
  app.js
  package-lock.json
  package.json
  Procfile
```

Fig 3(a). The file structure of the app

3.4 ER Diagram:

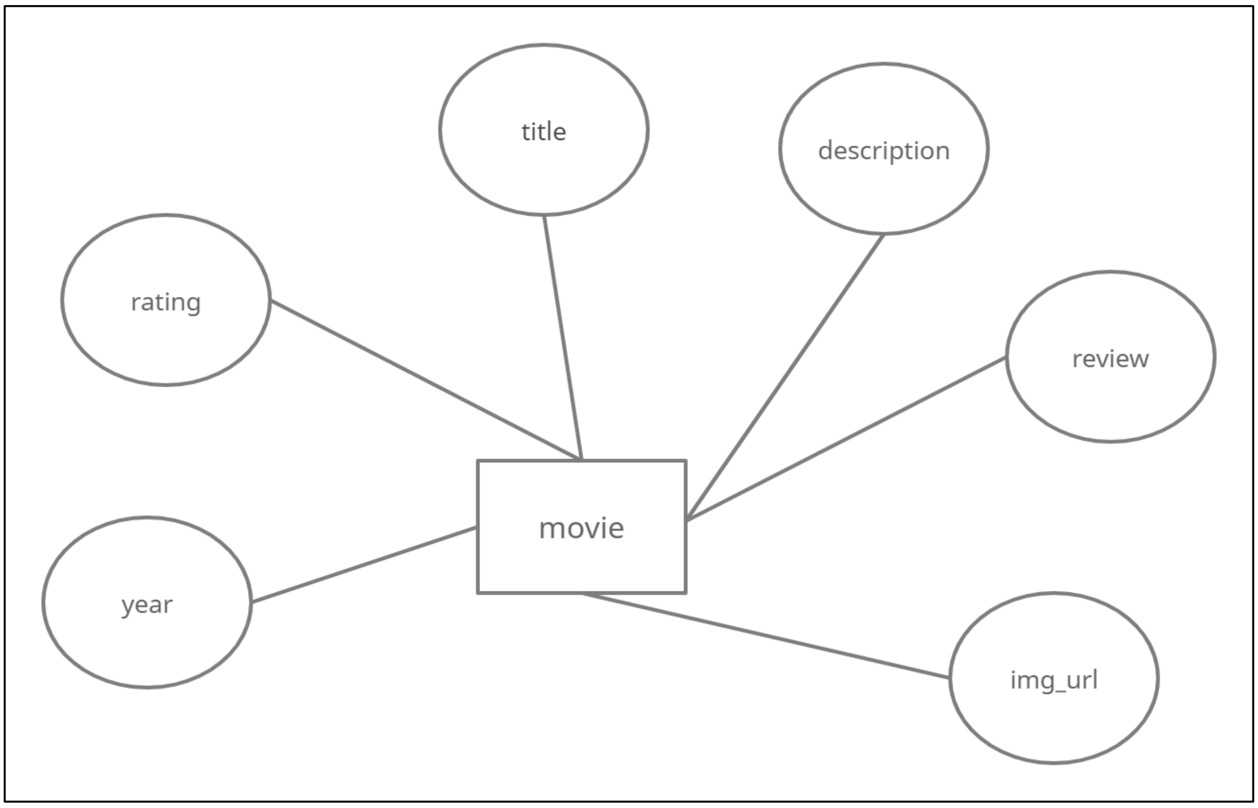


Fig 3(b). The ER Diagram of the Movie rating and review app

3.5 Use Case Diagram:

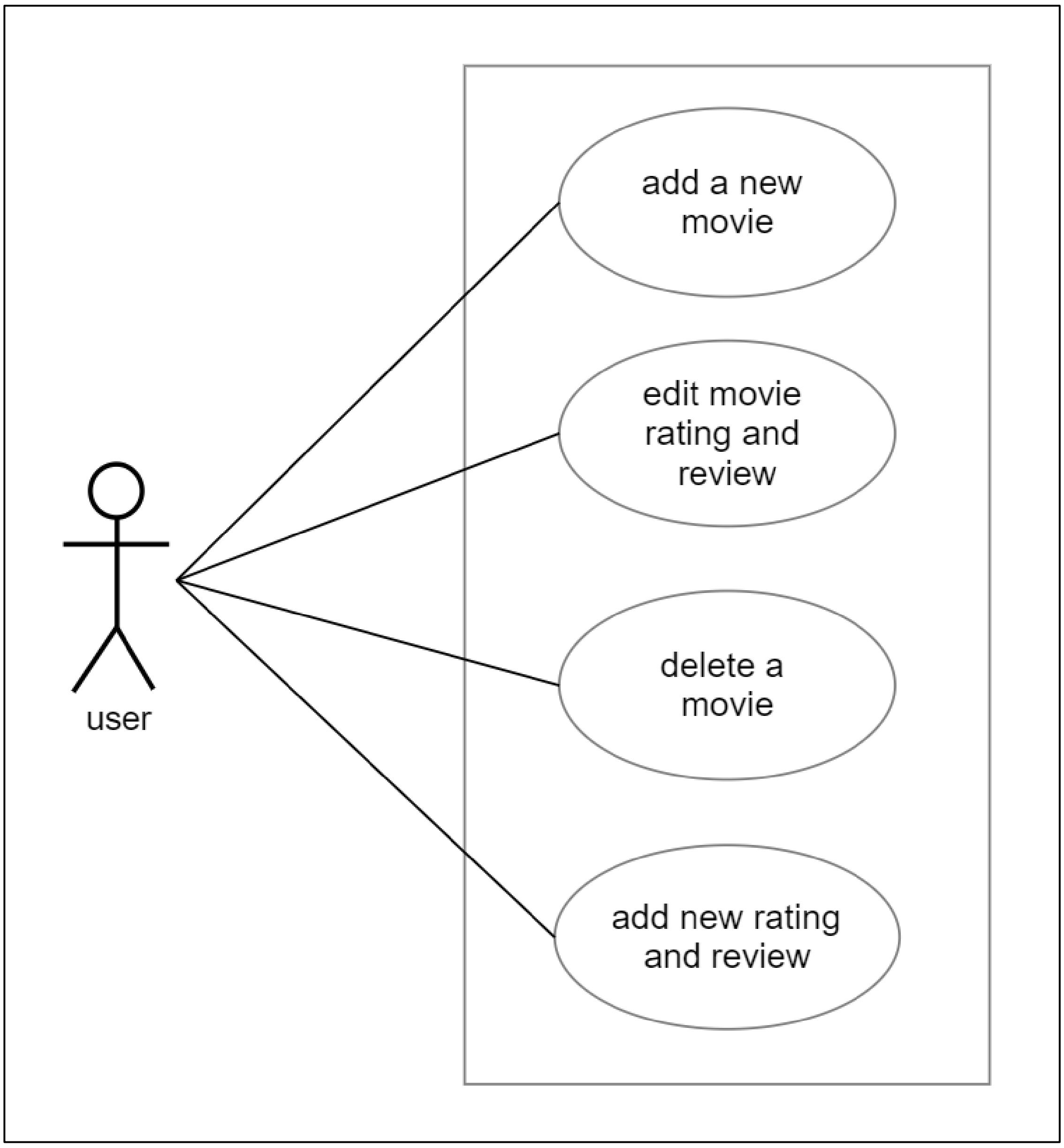


Fig 3(c). The Use Case Diagram of the Movie rating and review app