DIT5404 Individual Assignment – A simple movie rating web application

You are going to implement a simple movie rating system using Node.js, Express.js, NoSQL database (e.g., MongoDB), HTML5, CSS and frontend Javascript.
The submission date is **01 May 2020 (Thu) 23:55 on Moodle**.

Assignment Deliverables:
Zip all the files into one for submission:

- All the source codes of the system
- A simple technical documentation of the system using a Microsoft Word file within 2 pages (How to run the system? What are the public APIs? What is the file structure of the system?)

Instructions:
Please implement the movie rating system using the following functional requirements:

- The movie rating system allows users to rate a movie in a scale of 1 (the lowest score) to 5 (the highest score)
- There are three types of users, admin users (login required), normal users (login required) and anonymous users (without login)
- There must already be one admin user with username “admin” and password “pass”
- For anonymous users (without login), they can only view the (1) list of movies and (2) details (including rating) of a particular movie
- For normal users (login required), they can do all the things that an anonymous user can do, and also they can rate a particular movie
- For admin users, they can do all the things that a normal user can do, and also they can add a movie, remove a movie and update the details of a movie.
- Anonymous users can register to become a normal user
- A user can only rate a particular movie once, and cannot change the rate afterwards
- A search function should be provided to all types of users (by movie title, description, running time, release date, genre, etc)

Implementation details:

- A users collection in MongoDB or some NoSQL database, each user document must have at least the fields:
  - user_id
  - password
  - type
- A movies collection in MongoDB or some NoSQL database, each movie document must have at least the fields (you may add your own fields, if you do, please document it clearly in the documentation file.):
  - movie_id
  - title
  - description
  - running time
  - release date
  - genre
  - rating
    - user_id
    - rate

- After login, the user_id must be stored in cookie
- The following RESTful APIs should be provided
    - GET /api/search
        - Parameters
            - name=xxxx
            - description=yyyy
            - etc…
        - Response (JSON)
            - `[ {doc1}, {doc2}, ... ]`
        - Description: return the list of movies satisfies the search parameters
    - GET /api/rank
        - Response (JSON)
            - `[ {doc1}, {doc2}, ... ]`
        - Description: return the list of movies with descending order of rating
    - You may add your own RESTful APIs, if you do, please document it clearly in the documentation file.

Marking Criteria

Satisfactory of functional requirements (40%)
Add-on functionalities (e.g., RESTful APIs) (10%)
Passing the test cases (20%)
Clarity of code (e.g., comments) and file structure (10%)
Clarity of documentation (10%)
User interface (10%)

End of Assignment