Personalized web app for Movies

For this project you are obligated to use the following REST API: https://developers.themoviedb.org/3/getting-started/introduction

1. User:
   1. Registration with verification of the mail. Required fields are mail, nickname (both will be able to be used as the username), name, surname, address, city, country
   2. Upon login the user will see in the top toolbar his achievement badges and a drop-down menu with the two options to choose from: profile administration, movie history
   3. Profile administration: edit name, birthday, address, city, country
   4. Movie history with pagination: list all movies the user has interacted with:
      1. Already watched movies will be highlighted yellow
      2. Liked movies will be highlighted green
      3. Disliked movies will be highlighted red
   5. Verify mail action
2. Administration:
   1. Reset user’s password by invoking sending an email to the verified user`s mail.
   2. Creating badges for certain achievements: upload image and create condition for leveling up (e.g. watch 10 movies, watch 20 movies from top 100, watch 10 horror movies…). This will require some predefined drop down lists that will generate a rule in the DB.
3. Movie preferences:
   1. Listing of randomly suggested movies:
      1. Upon first login the user will be able to search for movies that he/she has watched and select them for the algorithm to suggest 10 movies similar to their taste (**suggestion service**). The requirement is to select 20 movies exactly. This will be shown only once at first login.
      2. After the bucket list gets empty the list will be populated with movies like those the user has watched and rated positively.
      3. As long as the user has movies in his bucket list, every hit of this page will lead to the bucket list
4. Bucket list: will show 10 >= movies => 1. There are several cases and actions:
   * 1. Initialization of the list is based on the selection made upon first login will lead to 10 movies in the list **(backend service / library is required)**
     2. Bucket list get shorter with each watched movie **(ajax call upon click to the backend)**
     3. When the last movie from the bucket list is marked as watched, the process of reinitialization is restarted based on the movies that the user has watched and rated positively. **(redirect to the movie preferences page)**
     4. For each movie in the bucket list the user will have three options to choose from: already watched, liked, disliked. After selection one of the options for a certain movie, the movie will disappear from the bucket list and all actions on movies will be suspended for the next 2 hours.
5. Movie page will show the details of a single movie. The user will be able to land here from 3 different places:
   1. Movie history
   2. Preferences list
   3. Bucket list
6. **Suggestion service** will have methods for checking the already selected/seen movies of the user and others for requesting/providing movies based on the list.
7. API service for interacting with the movie database
   1. Fetch N random movies
   2. Fetch N random movies based on a criteria (movies already likes/seen will be passed to this function)
   3. Fetch a single movie data
   4. Fetch a single movie image