General

The site can be used for two things: search info about a movie and chat about a movie. The user can search for a movie by typing the title in the search bar. This then queries themoviedb.org for information about the movie. From here on a chat is opened. This chat can then be used to chat (in real time) with everyone who has the site open. You can also see what movies everyone else is watching. The exact purpose is still to be discovered. But this can be useful if you want to watch a movie together over the internet with a group of people but can't decide on which movie to pick.

Hosting

The website is hosted at my own server. An nginx proxy is used to proxy request from api.search-for-a-website and search-for-a-website on to localhost:8090 and localhost:8091. This way the API and the front-end are running separately. The app is served using node and express. Together with socket.io to manage the chat function. The backend REST API is served using Django with the rest framework.

Design

The website itself is almost the same as the first assignment. At first the site was created in Django. To transition to angular was a lot harder than first expected. Eventually I did manage to get it to work with a angular framework. I also used "ui-router" and "ui-select" to render the search bar and manage the links. The layout is created with the default bootstrap theme and some minor modifications. This way it is very responsive.

Databases

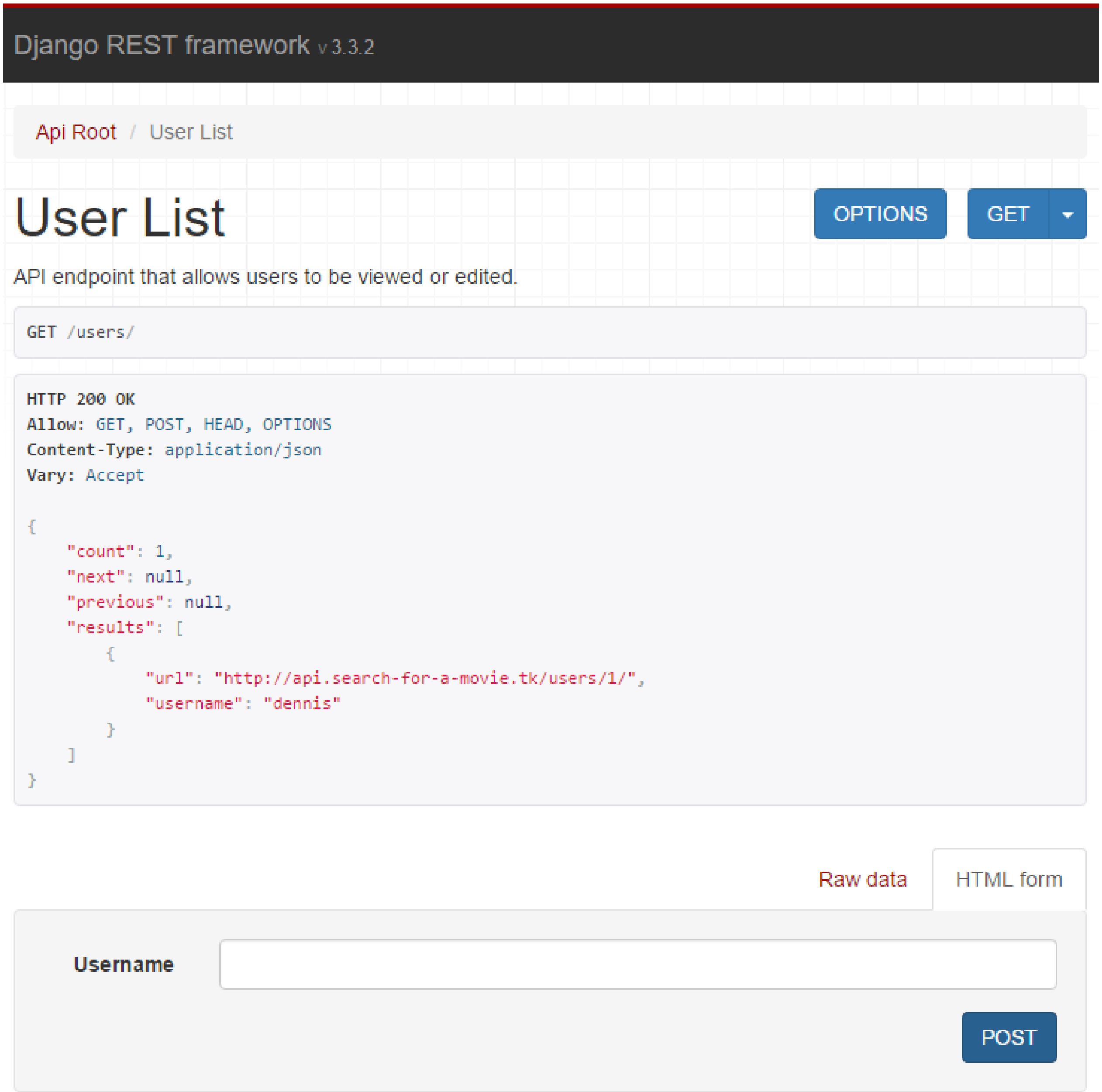
Django uses a SQLite database. The connection is configured in the settings.py and python manages the database process. Here is an overview of the models:

```
User(username)
MovieGenre(id, name)
Movie(id, name, genres)
Chat(creator, movie, name, timestamp)
ChatMessage(chat, user, text, timestamp)
```

Movie.genres is a Many-to-many relation an thus contains another listing table in the database.

Rest API

The complete (interactive) documentation of the rest API can be found at http://api.search-for-a-movie.tk/. You can click the url's to browse through the API. Also the POST forms below can be used to insert data in the database



User List

API endpoint that allows users to be viewed or edited.

```
GET /users/
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://api.search-for-a-movie.tk/users/1/",
            "username": "dennis"
        }
    ]
}
```