

Web Application Development Front End Project

Name: Watch This

By: Daemon Macklin

Student Number: 20075689

Watch This Web Application Functionality and DX

Navigation

When using Watch This users are able to navigate through the pages using the nav-bar on the left side of the page. The bar on the right side of the screen is where users can log in. Between the two bars will be where the pages will be displayed

Home Page

The home page is the first page that a user will be shown when accessing the application. The home page serves as a place where users can view statistics about the site and get a random media selection.

There are a couple of statics that are shown on the front page. On the top in the “What’s good?” section we can see the “Best media” and “Best review”. The best media is the media with the highest rating. If there are a couple of media with the same rating the best media will be the one with the most reviews. The best review is the review with the most up-votes. Further down the page we have a “Statistics” section. This displays a number of statistics about the application.

At the bottom of the page we have a random media displayed. This is a random media picker that users can use to randomly select a movie for them to watch or game for them to play. There is a new random movie each time the user goes to the home page. If the user wants the user can get a new random media by pressing the button. The user can also select which type of media they want. Either a movie or a game.

All Media, Movies and Games

These three pages is where users can view basic information and search though all the different media in the database. All Media displays all of the media in the database, Movies displays all of the movies in the database and Games displays all of the games in the database. Each of the pages has a table from the ‘vue-tables-2’ API, which displays all the media for that page. The table allows us to filter and search though the different media based on the fields on display.

For each media in the table there is a view column. This has a icon which when pressed will allow you to go to the specific media’s own page. This page contains all of the information on a review. In the about section there are two cards. On the left is the data about the movie, Type, Genre, Rating and the User who submitted it. If there is no User stored with the movie it will default to “Anonymous”. On the right is the video, which is an embedded YouTube video using the ‘vue-youtube’ API, based on the YouTube link stored with the media. If there is no YouTube link stored with the media the video will default to “Rick Astley – Never Gonna Give You Up”.

In the “Submit a Review” section users will be able to submit a review for the movie. All they have to do is input their review which is less than 60 characters and provide a score between zero and five and select submit. The review will be submitted and viewable in the “Reviews” section below.

The “Reviews” section contains all of the review for this media. Users are able to see the score given, what the review says and the number of up-votes. Users are able to up-vote reviews by pressing the check mark. Users do not have to be signed in to submit or up-vote reviews.

Add Media

This page contains a simple form that allows users to submit a new media. The user will supply a Title and a YouTube link, they will then select a Genre and a Type. Press submit and if the fields pass their validation checks the new media will be added to the database. Users do not have to be signed in to submit new media.

User

This page is for the user. When not signed in going to this page will provide the user with a form to sign-up. However if the user is signed in and has a valid token the user will be shown a user dashboard. At the top in the User Info section we see the users user-name. There is a log out button and an update button. The log out button will log out the user. The update button will show the user a form where they can update their user-name.

In the next two sections we can view all of the media and then reviews the user has submitted. It displays them in a ‘vue-tables-2’ table. In each of these tables there is a view icon which allows users to go and view the media or the media the review is associated with and a remove icon that allows users to remove the media or review from the database. Users must be signed in remove items and will have their tokens validated when performing these actions.

Users can sign-in by inputting their email and password into the sign-in form on left bar. If the credentials are successfully validated the user will be signed in and the user will be supplied a token. This token along with the user’s id and user-name will be stored locally in the browsers cookies. This is done using the ‘vue-cookies’ API. When Signed in any media or review submitted by the users will have their id submitted with it. The sign-in form will be replaced with a welcome message. Users can log-out by going to the users page and selecting log out. This will remove all of the cookies set and send the user to the home page.

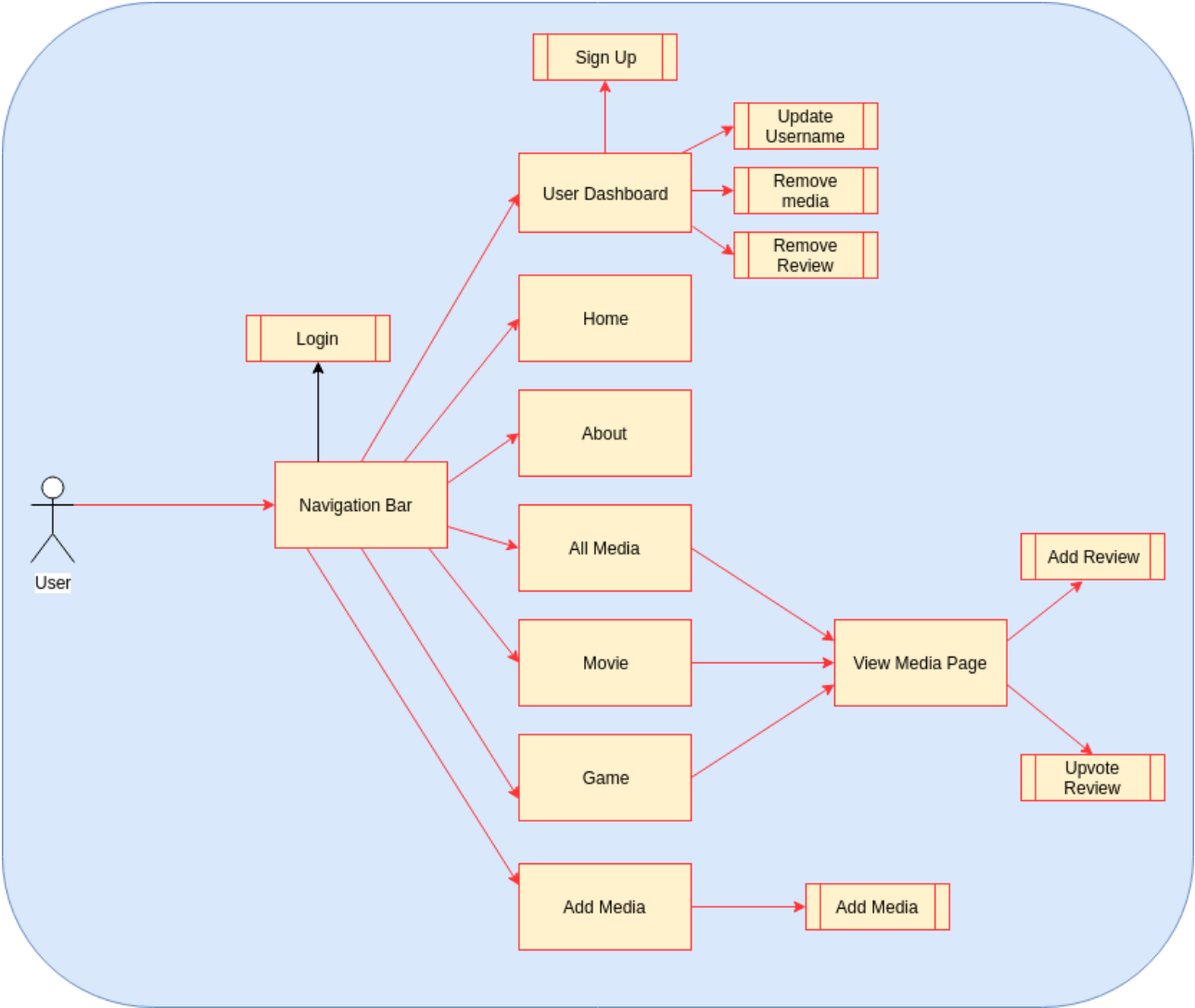
About

The about page is a simple page which contains a short message about the project and links to the project github.

Server

Information about the server can be viewed here:
<https://github.com/Daemon-Macklin/Watch-ThisV1>

Use Cases



Database Schemas

Media

```
1  {
2    "_id": "Unique id for media",
3    "type": "Type of media, Movie or Games",
4    "title": "Title of the media",
5    "genre": "What genre the media belongs to",
6    "userId": "Id of the user who submitted the media",
7    "youtubeLink": "Youtube link to media trailer",
8    "rating": "Average of all review ratings",
9    "reviews": [{
10      "review": "Text of the review",
11      "userId": "Id of the user who submitted the review",
12      "score": "The score given by the user",
13      "upvotes": "The amount of upvotes the review has"
14    }]
15 }
```

User

```
1  {
2    "_id": "Unique id for media",
3    "userName": "User name for user",
4    "email": "Email for user",
5    "hash": "Encrypted password",
6    "salt": "Randomly generated salt"
7  }
```

References

- Used David Drohan's labs for learning how to use vue <https://ddrohan.github.io/wit-wad-2-2018/>
- Used code from this project for v-for methods <https://github.com/ndabAP/vue-sails-example>

Frameworks and APIs

- SemanticUI <https://semantic-ui.com>
Used extensively for style and layout.
- Vue <https://www.npmjs.com/package/vue>
Used as main framework.
- VueYouTube <https://www.npmjs.com/package/vue-youtube>
Used to embed YouTube Videos.
- VueCookies <https://www.npmjs.com/package/vue-cookies>
Used extensively to handle cookies.
- VueTables <https://www.npmjs.com/package/vue-tables-2>
Used to display data.
- VueForm <https://www.npmjs.com/package/vue-form>
Used to provided forms.
- VueLidate <https://www.npmjs.com/package/vuelidate>
Used for validations on forms.
- Font Awesome <https://fontawesome.com>
Used for Icons.
- Firebase <https://firebase.google.com>
Used to host web application