MAP – CS445 - Project

The project will contribute 15 points to your final grade.

You will build a single page application for showing ASCII animations. The application consists of one HTML file + one JS files included with it. No updates should be made to index.html. All your code must be written to script.js. Generally, JS has all templates and logic to run a full application, JS will control the DOM. The DOM must not refresh at all time. However, only updated and the URL will change according to the selected route.

It's very important that you work on this project individually, copying code will be considered cheating and you won't receive any credit for your work.

Fork your initial repository along with the file to test the Rest API.

A daily push is required to track your code and performance. If you miss a push that will affect your final grade.

The project deadline will be on Thursday December 17th at 8 pm. No further pushes should be made to your repository. Perform one Pull Request to the original repository when you are done.

Project technical details

Your single index.html HTML file should only have one "outlet" element where all the JS code runs, you control the application from JavaScript. We need to create two routes:

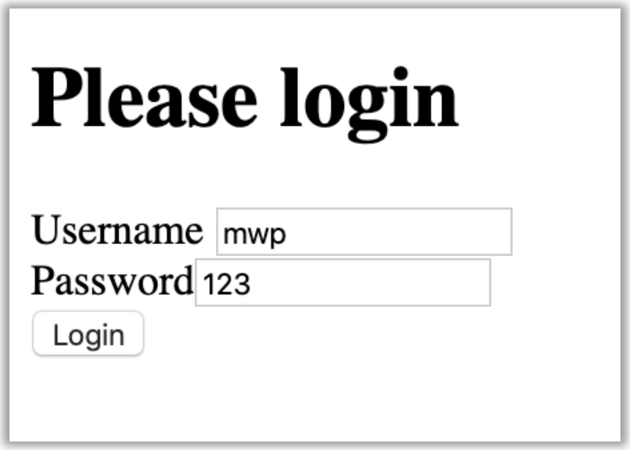
- Login page (default) "/login"
- Animation page "/animation"

When a route is displayed you need to:

- Load the template and insert it in the DOM (to the outlet element)
- Push a new state into the history API
- Add/Remove all event listeners to the new DOM elements accordingly
- Call any special functionality needed for the page (fetch animation, fetch location)

The login page

Displays two input controls and a button.

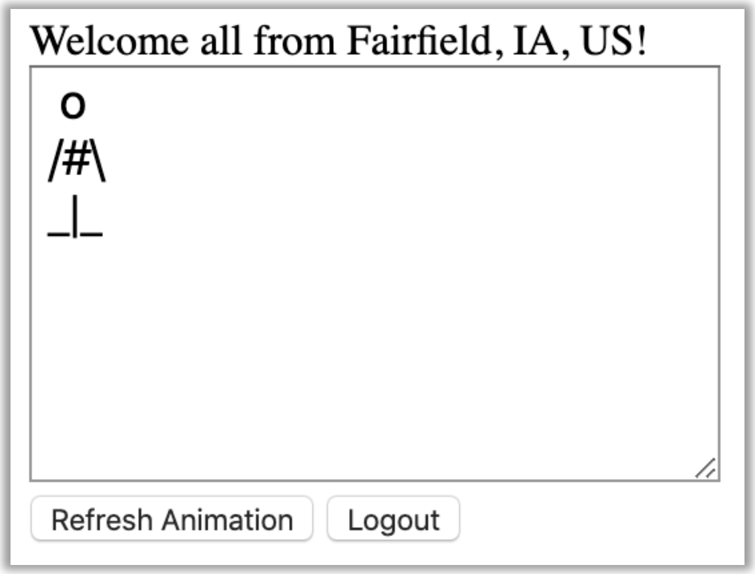


Please login

Username mwp
Password 123

The Animation page

Displays users' location (Fetched from an online service), a textarea where the animation is playing, a refresh button to load a new animation, and a logout button.



Welcome all from Fairfield, IA, US!

```
 o
/#\
_|_
```

Coding Instructions:

Start by wrapping all your code with a function, so all code is encapsulated and leave the global object clean.
The application should have references to the following:

- Token state: whether users are logged in or not.
- Animation interval ID: the current ID of your animation interval.
- Geolocation API Key: Register for free account at mapquestapi.com

Define two templates constants:

- loginTemplate
- animationTemplate

These constants should be loaded in the "outlet" every time the user asks for a new route.
By default, login template is loaded into the DOM when the application starts and the URL should be set to "/login".

After you load a view:

- You need to push a new state to history API.
- You need to attach/detach listeners to the DOM elements.

Listeners are:

- Login page: You need one listener on the "Login" button, when clicked, send an Ajax call to the server to authenticate the user. The server should respond back with a token. Save this token into your application global state.
- Animation page: has two listeners, one for "Load animation" so every time is clicked you will fetch a new animation frames from the server. The other listener is for "Logout button".

Listeners flow:

- When users click on "Load animation" you need to clear the previous animation, send an Ajax call to fetch a new animation frames, start a new interval.
- When users click on "Logout" button, you will need to clear the token, clear the animation, and load the login page again into the DOM.

Along with the animation page, you will need to fetch the current user location, and send their longitude and latitude to **mapquestapi** to find out their city, state, country. Display an appropriate welcome message on top of the animation page.

Animation string consists of ASCII chars, frames are separated with "=====\n", you will need to break the frames and load one frame at a time in the textarea every 200ms.

Finally, you will need to handle history changes, if the users used the "back" and "forward" buttons, read the state and render the correct view into the DOM.

The Server API:

URL: https://cs445-project.herokuapp.com/api/login
HTTP verb: POST
Request body: JSON format
{“username”: “mwp”, “password”: “123”
The server will send a JSON response with the following format:
{token: string, status: true}

URL: https://cs445-project.herokuapp.com/api/animation
HTTP verb: GET
Request header must include the following:
Authorization: Bearer token
Replace the token from the response received after you log in.
The server will send back a string response contains the full ASCII animation frames

URL: http://open.mapquestapi.com/geocoding/v1/reverse
HTTP verb: GET
You will need to pass an API key and the user current coordinates as **query parameters**. Read the docs for more details.
This request should return back a JSON object with full details about the location.

Good luck and happy coding!