JavaScript 2 Course Assignment

2023/4 Study Plan

Introduction

In this course assignment you will be building a client-side social media application. This application will allow users to perform CRUD operations (Create, Read, Update and Destroy/Delete) on their own posts as well as allow for additional features, such as following/unfollowing users, commenting on posts and reacting to a post with an emoji.

Unlike previous projects, you will be working on the app logic first and styling the application later.

# Brief

Using the provided API and API documentation, create a functioning user interface that allows for viewing, posting, editing and deleting social media content.

Social API routes require authorisation via JWT (JSON Web Tokens). You will need to register an account and login to access your token.

To complete the required features, you will also need to make use of GET, POST, PUT, and DELETE HTTP methods.

Using localStorage is highly recommended, especially for storing JWT tokens.

A finished project fulfils the requirements below with an easy to use and error-free user interface.

As we are testing your JavaScript abilities only, there is no requirement to style or design this application beyond a basic wireframe. You will have the chance to style and improve on this application's design later in the program. No grades will be awarded for styling work at this stage.

## API Functionality

The API supports the following actions, required actions are marked with a \*:

* **Register new user\***
* **Login user\***
* Follow / Unfollow user
* **Create post\***
* **Get single post\***
* **Get many posts\***
* Get posts of a user
* Get posts from followed users
* Search posts
* **Edit post\***
* **Delete post\***
* Comment on post
* Reply to a comment
* React to a post

The API has been designed to be flexible enough to emulate any one of the popular social media applications. For example, the API may be used to create an image driven application, or a text message driven application depending on your preference.

As a front-end developer, you will need to work within the limits of the API, understand the documentation and handle errors gracefully for your user. Your code should be neat, well formatted, structured and named so that another developer can easily find key functions.

You are not required to complete all of the listed functions above, however if you want to push yourself with an extra challenge, we have included Level 2 goals in the requirements below.

# Project Template

This project has been set up with a Vite template using Vanilla JavaScript settings, using MPA (Multi-page application) mode. Additional HTML pages not originally included in the project template must be listed in the vite.config.js file.

The template contains JavaScript files that must be finished to complete this assignment.

# API Stories

To complete this assignment, the following API features must be present in the repository and working without runtime errors:

* Register function that allows for the creation of new users
* Login function that allows for existing users to login with a token
* Create post function that allows a logged in user to make a new post entry
* Edit post function that allows a logged in user to edit an existing post
* Delete post function that allows a logged in user to remove an existing post
* Get post function that allows a logged in user to view a post

# UI Stories

To complete this assignment, the following UI features must be present in the repository and working without runtime errors:

* Register form allows a user to create a new account
* Login form allows a user to access an existing account
* Logout button that clears the token from the browser
* Post form that allows a user to create or edit a post
* Delete button that allows a user to remove a post
* Listing page showing 12 recent posts
* Listing page for a single specific post by ID

# Restrictions

To complete this assignment, please observe the following technical restrictions:

* Use of a JavaScript client-side framework or UI library is not permitted. This includes React, VueJS, Svelte, Angular or similar libraries.
* Please check with your teacher about group working

# Process

1. Accept the GitHub Classroom invitation [here](https://classroom.github.com/a/G_Y0UKBm).
2. Clone the repository to your work computer.
3. Create or update the HTML pages required.
4. Commit regularly and push your work using [meaningful commit messages](https://www.conventionalcommits.org/en/v1.0.0/).
5. Using GitHub projects, list the functions your project requires.
6. Create or update JavaScript files and functions to meet the stories.
7. Update your plan with each accomplishment.
8. Test your work, track and process bugs in the Issues tab.
9. Refactor your code, looking for areas for improvement.
10. Document all required functions using JS Docs.

# Delivery

* A link to your Public GitHub repo.

## **Optional Deliverables**

* A link to your deployment production website (Netlify/ Vercel/ GitHub Pages).
* A link to a gantt chart or Kanban project board.
* A video or written summary explaining your work
* A link to Figma design assets (wireframe or prototype)

# Level 2

In addition to the required features detailed above, the following **optional** features add deeper functionality to the application for those who want an extra challenge:

* Emoji menu allows a user to react to a post
* Comment form allows a user to comment on a post
* Reply comment form allows a user to reply to a comment on a post
* Follow button allows a user to follow another user
* Unfollow button allows a user to unfollow another user
* Pagination system allows for any number of results pages
* Search bar allows a user to find posts based on a keyword
* Use TypeScript instead of JavaScript
* Create unit tests for your key functions

# Resources

* Noroff API Documentation:
  + <https://docs.noroff.dev/docs/v2/social/posts>
* Noroff API Swagger:
  + <https://v2.api.noroff.dev/docs/static/index.html#/social-profiles>
* GitHub Classroom:
  + <https://classroom.github.com/a/G_Y0UKBm>