NEXUL ACADEMY

‘Immersive Program 212’ Prep Work

The purpose of the prep work is to ensure a prospective student is ready for the intensity of the immersive program at Nexul Academy. The immersive program will cover multiple technologies and many simulations of work in a real software development environment.

Prep work ensures a foundation in software development basic skills and problem solving skills to apply them. Prospective students choose their own projects, with suggestions available.

Required Concepts

A prospective student must demonstrate understanding of core concepts for acceptance into the immersive program, including:

- Distributed source control (git, github.com)
- Html
- Css (custom and with frameworks such as bootstrap)
- Javascript (basic use of variables and functions, plus conditionals and/or loops)
    - Access an API, see the “Local weather’ project idea for an example.

At completion of your prep-work, please schedule an assessment with a mentor.

Moving Forward

Preparation 101 (Recommended)

Our preparation 101 mentoring program offers guidance through all of the required concepts to enter the immersive program. As a participant, you attend free weekly open mentoring workshops.

Schedule: https://www.meetup.com/St-Louis-Coding-Apprenticeship-Prep/

Free Code Sites

You are encouraged to use freely available code study sites for prep work practice. Recommended sites include KhanAcademy.com and FreeCodeCamp.com. Each site gives basic exercises and may include video explanations.

These are even better when paired with preparation 101 mentoring to accelerate your learning.

Project Suggestions

All projects must demonstrate care was taken in the design of the application in appearance and user experience. You can pick any project you desire. Some project ideas include:

1. Local Weather report
2. Favorite Book / Movie list
3. Custom ideas welcome, discuss them with your mentor.

You may refer to the layout of other application or website designs to generate ideas, but you should add a personal flare to each project. Existing algorithm concepts can be studied, but complete code samples should not be copied.

Local Weather

A weather app should attain real weather data for the location of the current user or a user selected location, and then display the weather. The page design should reflect the weather in the choice of colors, icons and other visual elements.

This project should use the location API of the browser for a default location, as well as offering a way for the user to change the location.

The weather for a given location should be queried from a real weather API over REST using an Http client (such as the built-in ‘fetch’ or jQuery Ajax.) Some API options include:
https://openweathermap.org/api
https://developer.yahoo.com/weather/?guccounter=1

Bonus for adding forecasts, or allowing the user to change the forecast API source, or a means to compare discrepancies from various sources.

Favorite Book List

A book favoriting site would allow the user to view a list of books and add selected books to a favorites list.

The list of books should be loaded from an API. Book data can be found from a google API. Documentation for use is described here:
https://developers.google.com/books

The books chosen as a favorite by the user could be stored in “localStorage”.
https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage

APIs can be called from JavaScript using fetch. See the google books api for example code, or check the MDN documentation site:
https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch