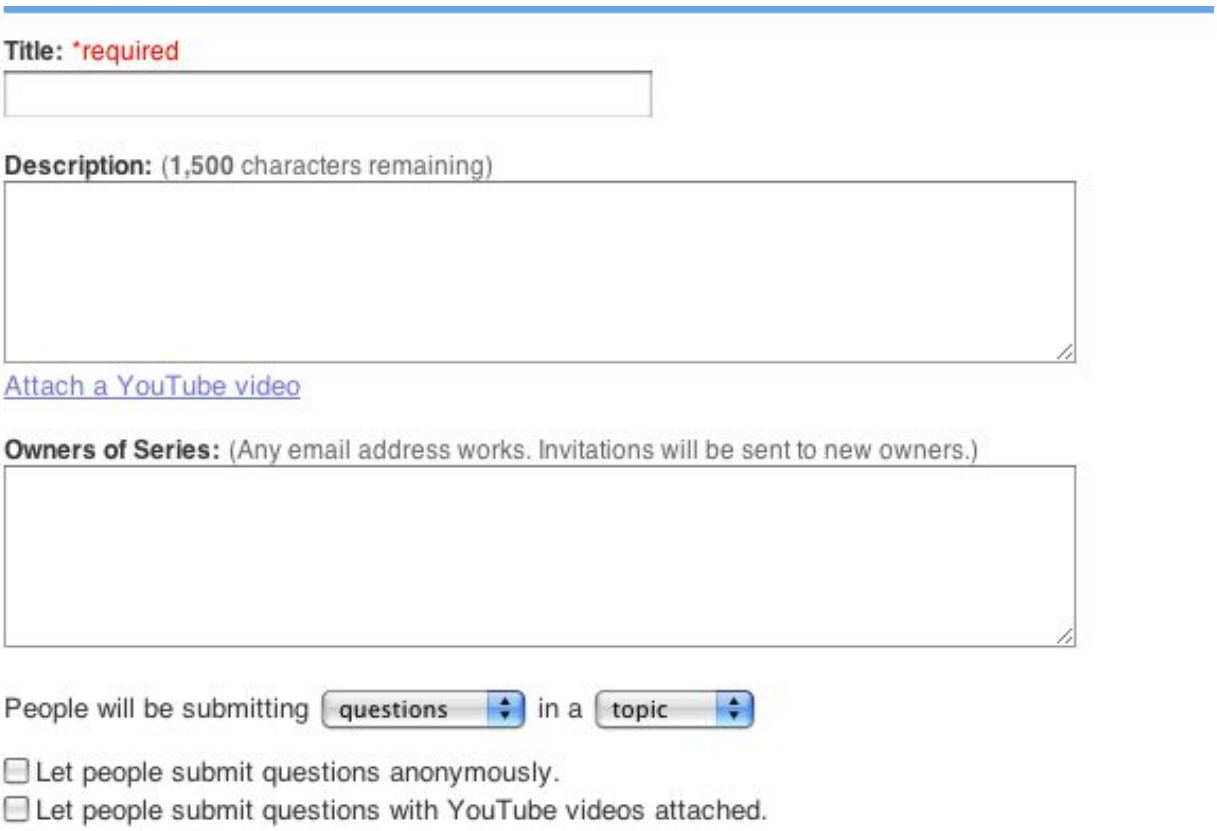

Google Moderator Clone

08.09.17

JP Miller
Udacity Connect | Infosys UI Intensive
Purdue University
West Lafayette, IN

Overview

In groups of four or five, your team will build a clone of Google's Moderator. This project will incorporate skills learned throughout the Front-end Developer Nanodegree as well as new skills not yet learned. This is an ambitious project that will span the remaining breakout sessions available to our intensive.

Goals

1. Build a clone of Google's Moderator using Vue.js.
2. Incorporate skills learned throughout the Nanodegree (and some not yet learned).
    a. The growth mindset.
    b. Create a mobile responsive web app using a design framework (consider Vuetify.js).
    c. Develop a dynamic web app using Vue.js.
    d. Incorporate test driven development with JasmineJS.
    e. Manage tasks with a task runner such as Grunt.
    f. Utilize collaborative version control with git and GitHub.
3. Optionally, create a more real-world development environment:
    a. Write code in ES6 syntax
    b. Use a CSS preprocessor such as SasS or LesS
    c. Packaging development code with a transpiler such as Babel or Vue Loader
    d. Design mockups and wireframes of the UI
    e. Create abstractions of the ViewModel
    f. Push your site live using Firebase, Heroku or Google Cloud Platform

Project Specification

You will work collaboratively with a team of four or five members to conceive, design and develop a Google Moderator Clone using tools sourced from the Nanodegree program and internet. The project is open in scope and method of implementation, but must meet the following basic requirements:

Vue.js Moderator

Interface Design

Criteria	Meets Specifications	Tips
Responsiveness	All application components render on-screen in a responsive manner.	1. All application components render on screen. 2. Application components must include a series creation screen for admins. This view will include: a. A title field b. A Description field c. A field for owners d. A cancel and submission button 3. Application components must include a moderation view for guests. This view will include: a. The title and description information from the admin panel. b. A field for questions/comments. c. A method to upvote and downvote submitted questions/comments. d. A submit and cancel button. e. Featured Questions

		(the most upvoted question) must highlight with a border box. 4. A registration view is required that will sign users via Google sign-in and Facebook Login. 5. Any additional components must also show on screen.
Usability	All application components are usable across modern desktop, tablet, and phone browsers.	This project isn't graded on the UX, but all components should be usable on desktops and mobile devices. Test for iPad and Nexus 5X

App Functionality

Criteria	Meets Specifications	Tips
Registration	Users can register with Google Sign-in or Facebook Login.	Use the Google Sign-in and Facebook Login documentation to help with implementation.
Landing Page	User has the option of creating a series or entering into an existing series.	
Series Creation	Anyone can create a series. Series creators become the admins of their series. Series views must have fields for a topic title, description and submit and cancel buttons.	
Comments, Questions and Voting	Users can see all created series, enter them and ask questions, pose comments and upvote and downvote	

	existing questions/comments	

App Architecture

Criteria	Meets Specifications	Tips
Proper use of Vue.js 2.x	Vue.js is used as the app's framework. Vue.js's MVC pattern should be used to control the model and view	Use Vue documentation here: https://vuejs.org/v2/guide/ A TodoMVC example exists here: https://vuejs.org/v2/examples/todomvc.html
Responsive CSS	Use a responsive design framework such as Bootstrap or Foundation.	My suggestion is to try and use Vuetify.js https://vuetifyjs.com/?ref=madewithvuejs
Test Driven Development	Incorporate JasmineJS to ensure your app passes the test	

Error Handling

Criteria	Meets Specifications	Tips
Defensive Programming	Error messages are provided to the user when APIs can't connect or fields are entered incorrectly.	Test error messages by intentionally breaking the code.

Git & GitHub

Criteria	Meets Specifications	Tips
Use git and post your project on GitHub	Work collaboratively within your team using git and Github	

Task Runner

Criteria	Meets Specifications	Tips

Uses a Task Runner	Incorporate the use of a JavaScript Task Runner such as Grunt, Gulp or Broccoli	Feel free to stick with Grunt, but if you're ambitious or curious, there are other task runners available for your use.

Extras

Criteria	Meets Specifications	Tips
Use ES6 aka ES2015	Optional: Incorporate the use of ES6 functions	This is optional. ES6 includes things like arrow functions.
Use a Transpiler	Optional: Use a transpiler to convert ES6/ES2015 to traditional JavaScript	If you opt to use ES6 you'll need to use a transpiler such as Babel. If you opt for this, then check out Vue Loader and the vue-cli
Use SasS	Use a preprocessor to manage your CSS. SasS and LesS are the most popular preprocessors.	Many of you will use SasS in your work with clients. Strongly consider this option.
Make your site live	Optional: Push your site to the live internet using a back-end service such as Firebase, Heroku or Google Cloud Platform	This is completely optional. AWS is another popular choice, but the other options are FREE for small projects such as this.

Documentation

Criteria	Meets Specifications	Tips
README	A README file is included detailing all steps required to successfully run the application.	Need a README refresher? Refer to https://www.udacity.com/course/writing-readmes--ud777
Comments	Comments are present and effectively explain longer code procedures.	
Code Quality	Code is formatted with	

	consistent, logical, and easy-to-read formatting as described in the Udacity JavaScript Style Guide.	

Resources

I. Vue.js 2.x

Vue.js: https://vuejs.org/

Vuetify.js: https://vuetifyjs.com/?ref=madewithvuejs

II. Google Moderator

The Moderator Toolkit site:
https://sites.google.com/site/moderatorhelpcenter/getting-started

III. Transpilers

JavaScript Transpilers:
https://scotch.io/tutorials/javascript-transpilers-what-they-are-why-we-need-them