Roku Flashback Streaming App

Authoring 4 - Team Build

Assignment Description:

You will be building an audio / video app in the Authoring class, using queries and database assets that you'll be compiling (or have compiled) in the Advanced Web class. Create assets based on the Roku Flashback brief (to be supplied) to create the layout, views and components you will be working with.

This project requires two roles - designer and developer. Because there are usually more of one than the other, you'll have to adapt to the demands of the project throughout its lifecycle. In other words, approach this as a true team - the designer might have to help with some code challenges, and vice versa.

Use the LAMP stack (Linux, Apache, MySql and PHP) for the server side and Vue.js as your front-end framework. You can re-use most of your work from the fall term - your movie API that you did in the Web class. You've got a couple of endpoints set up, queries written etc; you'll need to expand on those and adapt it to work with AJAX / JavaScript.

A user should be able to log in, filter and select a movie, tv show, or audio based on user profile (adult or kids). This means that you'll have to restrict access to some content based on age or ratings, and your users will need rights and priveleges as well. The UX / UI should update based on the user (children / adults) - create 2 themes, and render the appropriate theme on login.

Part of the API that you're building should allow a user to comment on the selection, rate it, and share via social media.

Requirements Per Role

Designer

Design the look and feel of a mobile-first web application based on the supplied brief (Roku Flashback). Creatively this project is wide open - push yourself to explore the latest trends in mobile-first design. Consider the end user; focus on engaging UX and UI.

Use the prototyping tool of your choice to provide direction for your developer. Think about interactive elements, including what might be required in terms of Single Page Applications (SPA), hybrid multipage applications, and data transactions (preloaders, microtransactions, flash messaging etc).

Include a Design Reference document (this should be a Google Doc) that includes as much detail as appropriate - branding concerns, creative direction, style guide etc.

Provide the required SASS elements to your developer - remember, these are your design choices, not theirs. You'll need to provide colour values, CSS transition specs, etc via Git and Github to your developer.

Submit a link to a live working prototype - please **DO NOT submit XD files or other working files**.

Developer

Create the proper project stucture including a Github repo / readme. You can use the database assets and API from the previous Web class as your starting point, and test / implement the API endpoints to retrieve and populate content in your application.

Refer to the design prototype(s) to determine the Vue components and additional functionality you'll need to create. Use the Vue router to control what appears on the page and when.

All data will be loaded via AJAX on initial load and as required during the app's runtime; create the required mechanisms to accomplish this (the Fetch API / Vue.js components and client-side routing).

You should research and implement how to include social media APIs (Facebook, Twitter, etc) to like, share, and comment / rate on each of the individual multimedia selections.

Implement your functionality on individual branches (ex. login should be on its own branch, social media should be on its own branch, etc) and merge to the master branch as you go. Think (and develop) in sprints - one piece at a time.

CHECKPOINTS

There will be a checkpoint at around midterm time (week 7 / 8) to guage progess. Break your project down into deliverables / sprints so that you're ready with something to show - **you should do this weekly** and have something to contribute **each and every week** until you're done.

Your design work should be mostly finished; your prototyping should be well underway; you should have endpoints tested and working, content and controls functional (or at least tested) and a strategy for completing the project in place.

Submission

Homework must be submitted by midnight of Saturday of **Week 13**. Include the following:

Github repo: master branch with any other dev branches you feel you need

Name the dev branches appropriately per feature IE des.tvr.icons, des.tvr.player, dev.tvr.login, dev.tvr.commenting

Design document (Google Drive doc)

Dev notes (Google Drive doc)

Submit the repo link via FOL dropbox

Please follow correct folder and file structure as outlined in class

Additional Information:

Missed tests/exams will not be rescheduled without some valid evidence of some important event over which the student has no control (e.g., Court appearance, death in the family). Missed tests or exams, therefore, can receive a zero. The students are advised to notify the professor prior to missing the test.

Students are expected to hand in all assignments to the course instructor on the due date, and all assignments must be submitted in the format specified by the instructor (e.g., on FOL, in printed form, on a specific lab computer, etc.); assignments will not be accepted in any format other than that specified.

Late assignments will not be accepted, nor will make up test or assignments be permitted, without some valid evidence of some important event over which the student has no control (e.g., documented illness, death in the family). Missed tests or assignments, therefore, will receive a mark of zero. Late assignments and make-up tests will only be permitted following the submission of adequate documentation acceptable to the instructor (e.g., a doctor's note). Students are advised to notify the instructor prior to missing an assignment due date or a scheduled test.

Immediately upon return from an illness/absence in which a test or assignment has been missed, the student is responsible for contacting the course instructor to discuss the problem. The instructor will make arrangements for any student deemed eligible. The alternative test/assignment will be of equal value to the one missed with no grade penalty. The timeline and due dates will be determined by the course instructor.

At mid-term, any unsatisfactory results will be reported to the student.

This course may be revised by the professor with suitable notification to the students. Students are responsible for making arrangements to pick up missed handouts, assignments and course announcements from classmates.

Plagiarism (e.g., failure to acknowledge sources used, submitting another student's work under your name, or producing work for another student to submit) is a serious academic offense that shall result in appropriate penalties, to be determined at the discretion of the course professor in consultation with the chairperson of the Communication Arts division. The penalties shall range from failure of an assignment to possible failure of the course. Students shall not make the assumption that any provision will be made by the professor to permit the student to rewrite or redo failed assignments.