Data Representation and Querying Project

Authors: Karle Sleith,

Daniel Verdejo.

***Project Outline***

We are required to develop a single-page web application (SPA) using multiple programming languages such as: JS via Angular1.6 framework, HTML5 & CSS3 via the Bootstrap framework,& Python using the Flask framework . You must devise an idea for a web application, write the software, write documentation explaining how the application works, and write a short user guide for it.

***We will learn…***

* How to integrate concepts learned across a variety of subject areas.
* How to identify relevant material on a given topic from available information sources.
* How to succinctly present rational and reasoned arguments to a range of audiences.
* How to develop innovative solutions to pragmatic situations.
* How to recognise the suitability of a given solution to a problem.
* How to apply knowledge learned in new situations.

***Project Guidelines***

In this project we are required to use Angular JS, Python, HTML, & CSS to create a single page app.

The goal of the web application is to use multiple programming languages to create a fast & responsive web application. We will also try to link the web app to a small database that will store some information.

* Using python & flask a database will store the post & user info, data links into factory then using angular ng repeat contents from factory.
* Using controllers and factories written in JavaScript, we will parse information from our database and filter information requested by the user. The information will be filtered by tags.
* Using ng-route we will be able to route our web application in a quick responsive way allowing for a seamless user experience.
* Creating a basic enough blogging website.

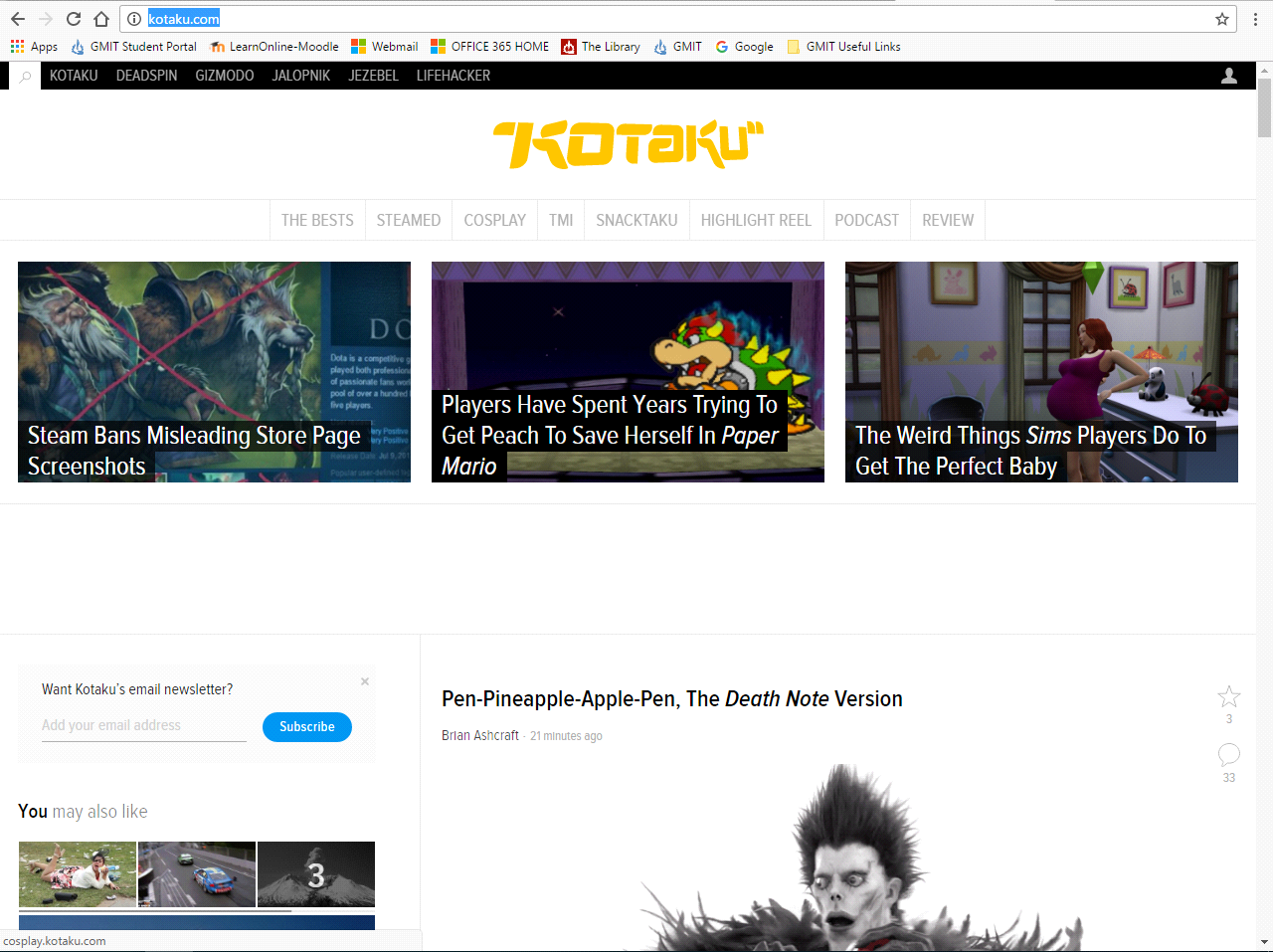
**The pages should consist of**

* A home page that shows different topic 2 the detailed list of posts under a topic - should have input box to allow users to add to the blog 3 create new topic found on the topics page.
* The banner navbar the list of topics each topic displayed in a card style view - including preview of first (original) post, author, date posted. footer bar.
* A second page that will hold the topics ,banner navbar list of posts under a topic - topics are in cards with title, author of post, body of post, date posted an input section where users can then add to a topic (comment, discuss) add button, cancel button footer.
* A third page that will allow the user to enter new topics ,banner navbar title of the topic author of topic body of topic add button, cancel button footer.

**Product Outline**

We will create a single page web application, based around the topic of “Entertainment Media”, which will include Video Games, Movies, Music and Television. Our primary inspiration for our Website is the site “Kotaku”, which hosts these topics.

<http://kotaku.com/>

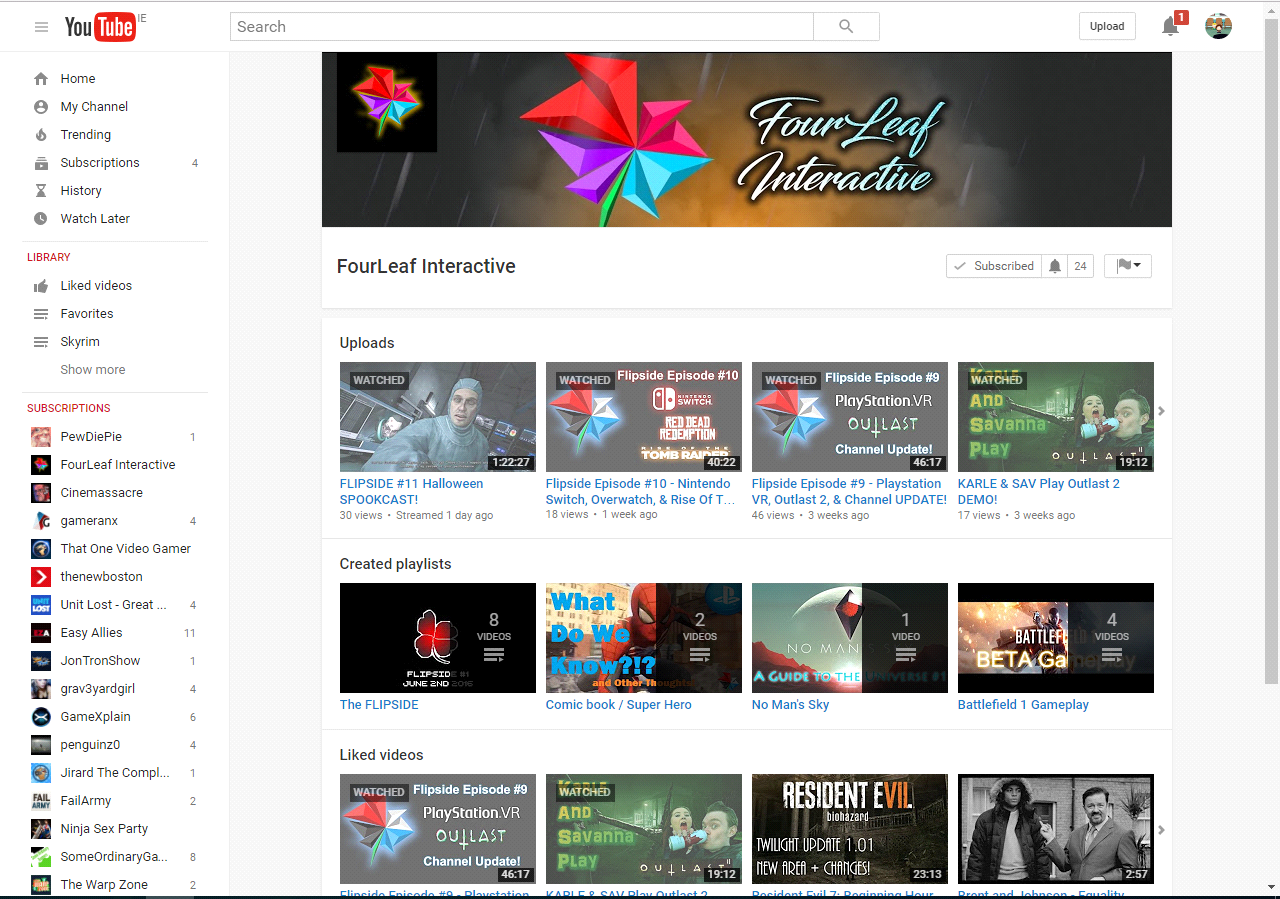


We want the user to add to the Website, adding their own article, and posting it to the relevant topic heading, allowing for easy search, and other users to be able to read other articles.

Also will be the option to embed videos from YouTube, such as links to movie reviews and gaming tutorials.

The site will be linked to the Youtube Channel “FourLeafInteractive” which is multimedia channel hosted by Karle Sleith and Daniel Verdejo, which features fortnightly Podcasts and Gaming and Movie Reviews.

<https://www.youtube.com/channel/UCD0i7gLN-kEg3RQRU3dop7w>



***What do we hope to achieve?***

* A single page web application, using AJAX calls to move between topics and articles.
* User Interaction, to create articles, Display the author, and the date the article was posted.
* Store data using MongoDB.
* Create a secure and professional system, passing information from the DataBase to the User.
* Create a professional User Interface, allowing easy navigation of the SPA.

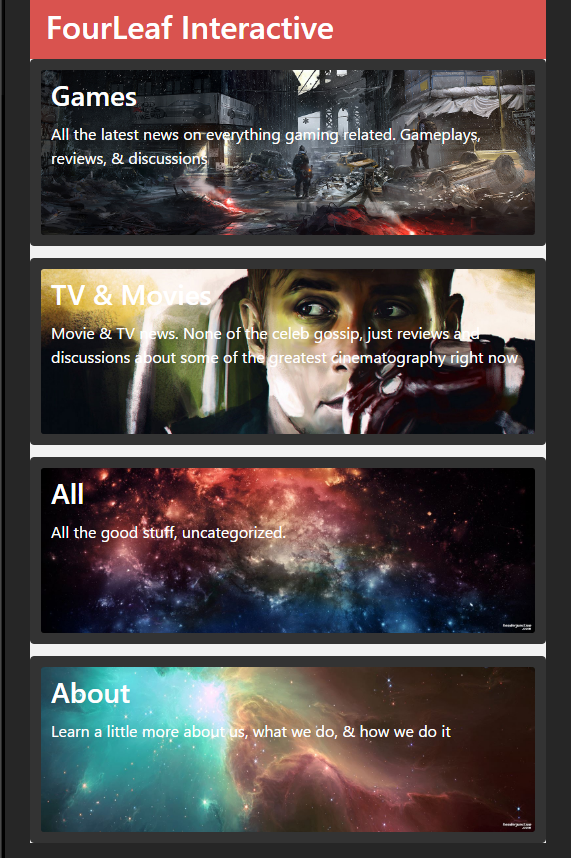
***Problems we expect to face?***

* We expect to be slowed, while learning new concepts, integrating a DataBase into a website, using AJAX calls.
* Learning new applications such as Python/Flask and Mongo.
* Disagreements on how to approach various tasks.

***How will we solve these problems?***

* We plan to have regular meet ups and work together to solve problems.
* In disagreements, we will try and respect the other person’s solution and come to a compromise, in a calm a professional manner.

**The Final Solution**

In this project, we look to utilize multiple technologies to build a single page web application. The first tasks of the project we’re choosing what we wanted our single page application to do, and then choosing the technologies we would use to accomplish this. The choice of application would be a blog style entertainment website.

The technologies to be used we’re to be: AngularJS, Bootstrap, Flask via Python, HTML5 & CSS3, & MongoDB.

To accomplish our goal we would need to research and expand upon our existing knowledge of: AngularJS,  Flask via Python, & MongoDB. To do this we would reference the related API’s, read articles, & communicate to workout problems together.

**Technical Information**

**Python**

Python is a high level, general purpose programing language. Is an easy-to-use language that makes it simple to get your program working. This makes Python ideal for prototype development and other ad-hoc programming tasks, without compromising maintainability.

**Mongo**

**MongoDB** is an open source, document-oriented database designed with both scalability and developer agility in mind. Instead of storing your data in tables and rows as you would with a relational database, in MongoDB you store JSON-like documents with dynamic schemas.

**AngularJS**

is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. Angular's data binding and dependency injection eliminate much of the code you would otherwise have to write.

**Bootstrap**

is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. Angular's data binding and dependency injection eliminate much of the code you would otherwise have to write.