Snowsylvania: A Modern Platform For the Sharing of Creative Work

Eldon Hoolsema and Dr. Jonathan R. Engelsma

April 2019
Abstract

Snowsylvania is a website that aims to provide a modern platform on which users can upload, share, and save their creative endeavors. Users can browse for content in various ways, comment on posts, subscribe to other users, get tailored recommendations, and more - all in a responsive, state driven single page application. The motivation behind this project was the observation that similar sites such as DeviantArt are slow and clunky. They use aged technology and lack features that people are increasingly coming to expect. This project takes the basic idea of DeviantArt and extends it, incorporating modern frameworks and Software-as-a-Service (SaaS) solutions to implement features that are rarely, if ever, seen in websites catering to a creative audience.

Introduction and Motivation

Anybody who creates something that they are proud of will undoubtedly wish to share that with others, regardless of what the something is. On the Internet today, there are various different websites where artists can post, share, and save their work, ranging in size from small niche sites to mainstream cornerstones like DeviantArt.

Originally launched in August of 2000, DeviantArt started as a place to connect together people who took computer applications and modified them to fit their own needs or desires - hackers, of a sort. It was a forum that let users share their “deviations”. In 2006, the site added the ability to upload most any creative content under the Creative Commons License, opening the door for photography, digital paintings, writings, and much more. Since then it has grown to be the largest online artwork community with over 250 million submissions. While it remains one of, if not the most popular online artistic communities, it has begun to show it’s age.

DeviantArt has a somewhat clunky user interface and uses just a bare minimum of client side rendering. There are many parts of the site that could be called “cluttered”. Content tags are seemingly an afterthought - DeviantArt provides no indication of which tags are in use and how popular they are. When uploading a post, a user literally just types a string of words, spelling errors and all. Finally, DeviantArt has no recommendations of any kind. A feature that is used to great effect on other sites like Facebook and Twitter would seemingly be a natural fit for a website geared towards an artistic community, but it is completely absent.
Thus, the goal for Snowsylvania from the start was to become something akin to DeviantArt, but better. A better user interface, faster web page rendering, easy to use content tags, personalized content recommendations, and even some niche features like banners to advertise when a user is streaming (many artists like to stream as they work - it is enjoyable to watch very skilled artists).

Technologies

Snowsylvania makes use of several popular technology stacks and frameworks. It also leverages several Software-as-a-Solution providers to handle the more tedious or difficult aspects of the application.

Frameworks

1. **ReactJS** - a Javascript framework developed by Facebook. Focuses on reusable components, encourages stateful interface design. Manages many of the difficult and error prone aspects of developing and controlling a User Interface.

2. **Spring Boot MVC** - an open source, Java based MVC web server framework managed by Pivotal Labs. Utilizes inversion of control and dependency injection to enhance organization and code reuse. Includes many powerful features right out of the box, including HTTP request serialization, database connection management, session management, scheduled jobs, and much more.

3. **MySQL** - open source relational database project managed by Oracle. Extremely popular, very reliable with respectable performance. The primary data store for Snowsylvania.

4. **Solr** - an open source indexing engine managed by the Apache Software Foundation. Built on Apache Lucene, Solr indexes documents according to a preconfigured schema. These indices then provide lightning fast search capabilities. Also includes fuzzy matches, wildcard searching, value ranges, and other querying capabilities.

SaaS Solutions

1. **Cloudinary** - a cloud based image storage and delivery platform. A spinoff from Ebay, Cloudinary provides storage and organization of various media types, including images, video, and audio. Can perform numerous on-the-fly image transformations, including
cropping, rotating, and scaling. All of the actual media files for Snowsylvania are stored in and served by Cloudinary.

2. **Recombee** - a cloud based recommendation provider developed in collaboration with the Czech Technical Institute in Prague. Provides real time recommendations based upon the data you feed it. Uses deep learning algorithms to look for relations in the data, which it uses to generate personalized recommendations.

**Features and Design**

Most of the features of Snowsylvania are exactly what you expect from a site focused on user generated content and social interaction. Many of the basic features are common to vast number of websites, including user accounts, comments, content filtering, different browse / search criteria, and user to user messaging. Effort was made to try and make these common features as good as they can be, however, much more effort was spent focusing on the three main aspects that made Snowsylvania different than its competitors.

**Single Page**

By combining ReactJS with additional Javascript libraries like Redux and React Router, the entire application technically lives in a single web page, dynamically swapping content based upon what the URL shows. Pages are extremely responsive and network errors - should they occur - can be contained better without displaying broken web pages. Single page applications (SPA) are difficult to develop but provide a superior end user experience. Not even such juggernauts as Reddit utilize the SPA architecture, much less a smaller community like DeviantArt.

**Tags**

Every post has the option for content tags. Each tag needs to be at least four characters long - to avoid spam and lessen the load on the Solr index - and each post may have up to 128 tags. These tags are used to power the search feature. To help with selecting the "right" tag for a post during the upload process, tags will be suggested based upon what you had typed - this is to help avoid spelling errors and to increase the uniformity of the tags, enhancing a post's discoverability. The number of posts that include a given tag is also known and shown to the
user, which is helpful in determine which tags to choose. Figure 1 shows the tag selector from the Upload page. Suggested tags appear in the top box, chosen tags in the bottom.

**Tags**

Used for searching and generating recommendations. It's helpful to tag your post by the things it visually contains, not that which you "know" about it.

- dogbert
  - dogbert (1)
  - (no tags chosen)

Figure 1. The tag selector from the Upload Page.

**Recommendations**

This is a feature not seen in any other online art community. Every time a user clicks on a post thumbnail, favorites an post, or adds a comment, a data point is sent to Recombee. Upon request, Recombee can utilize the relations it finds amongst the provided data to generate a batch of recommended posts unique to each user. The recommendations have been set to rotate slowly rotate in an attempt to avoid frequently repeated suggestions.

**Reflection**

The development of this application was less of a drive to learn something new but to take existing knowledge and enhance it, sharpen it. Building an entire website was a tremendous challenge but I was able to leverage ideas about how to design and organize web pages that I had learned from both work and class. Along with the features of React, I was able to make swift progress in the assembly of this project. It was, however, not without some problems.

Spring Boot is very powerful, but if you let it, it can become very complicated. I would not call it an exaggeration to say that the underlying Spring MVC architecture of Spring Boot has around 10,000 different knobs and dials to turn, and early in the development of Snowsylvania I
got overwhelmed. I had just reached the point where I had just enough User Interface to start wiring up the API to begin sending and retrieving data. Unfortunately, in my attempts to get Spring Boot to serve my Javascript files, something in my configurations for the project snapped and the folder containing my static files was emptied. Frustrating for sure, but thankfully I did not lose too much work - it was early on. Still do not know what happened.

Another hurdle was conflicts within Java libraries. Throughout the project, Lombok has been used to make the creation of getters, setters, constructors, and other boilerplate code as simple as adding an annotation to the class declaration. A tremendous time saver. However, problems occurred when I attempted to include AspectJ to the project. My plan had been to use AspectJ - a Java library that implements Aspect Oriented Programming - to automatically add input and output sanitizers to all my model classes. Turns out, Lombok and AspectJ do not like each other one bit. I never even got the two to compile, so I dumped AspectJ and got a bit creative, instead hooking my sanitizers into the HTTP JSON serialization / deserialization.

The development of Snowsylvania was not, thankfully, all problems. There were many moments of great learning and success. The simple object relational mapper I created to perform simple queries and convert database records into Java models and vice versa worked marvelously. Hooking an input sanitizer into the underlying HTTP request deserialization was quite the challenge but it worked better than I had hoped for - better than what using AspectJ might have been able to do! Spring Security - an add on module to Spring Boot that supplies all the security tools one might ever need - is a complex beast but I was able to get it sufficiently configured in just a few hours. I learned a tremendous amount about how to programmatically send emails, a feature I used to send account verification links and password resets!

**Conclusion and Future Work**

Snowsylvania has been a very fulfilling project to work on. I personally have a passion for creativity and being able to build what I consider to be a piece of the next generation of DeviantArt-like sites has been very rewarding. It features many of the common features one would expect for a user content drive site, yet includes some comparatively fresh and interesting ideas. It brings together a whole host of technologies into a unique platform with a great deal of room to grow.

As such, Snowsylvania is not - at time of writing - fully ready to be deployed. There are still some bugs to be fixed, some inputs to be polished, and a few features to be implemented.
While there is nothing to my knowledge that is considered to be broken, things do need a bit of polish. Just a few such things include:

1. Some of the modals do not scale well to smaller screens.
2. It would be nice to add a site roadmap and way to display announcements
3. Moderators need some tools to moderate content that does not require them to access the database directly.

All in due time.