A central media hub for all the things *you* love

CS 40700 - Senior Design Project

SPRINT 1 - RETROSPECTIVE

Team 6

Utkarsh Agarwal *uagarwal@purdue.edu*
Shivangi Chand *chands@purdue.edu*
Amol Moses Jha *jha8@purdue.edu*
Pooja Tewari *tewarip@purdue.edu*

WHAT WENT WELL?

We were able to meet all our expectations for the first sprint. We successfully completed all 12 user stories that we had planned for this sprint. We planned to deliver industry-standard login and user registration with options to use Facebook and Google for both. We were able to complete this deliverable to our satisfaction. Security was a big requirement for this sprint for both frontend and backend implementations. We were able to accomplish this requirement using industry leading technology - JWT (JSON Web Tokens).

One of our major concerns for the first sprint was if everyone would be able to acclimate to each other's working styles. We discovered that each of us complements the others very well in terms of work ethics. All of us were committed to uphold software development best practices in our daily work. Our pull requests were peer-reviewed in two steps in order to ensure that the code is well-tested and doesn't break existing code.

Our team worked well in terms of communication which was another major concern of ours going into the sprint. We held daily stand-ups with regular updates to the team on where we were for all the user stories and if there were any blockers present for any team-member. Our weekly meetings were really beneficial for integration purposes and removing the blockers discussed during stand-ups.

User Story #1

As a user, I should be able to view the application's main landing page before logging in.

Completed: This user story was primarily frontend work. The implementation did take about a week because the landing page was responsible for deciding the look and feel for much of the app. The frontend team had meetings to discuss and decide what components to add to the landing page and of course to decide what colors and background were to be used for the CSS. The frontend test plan was used to manually test the frontend of this user story.

User Story #2

As a user, I should be able to create an account.

Completed: This user story was definitely time consuming as it took time to set up the backend for this. The frontend for this user story was done rather quickly as the page followed the UI feel from the landing page. The integration was the challenging part, but was done successfully. The page was also well tested both on the backend as well as manually on the frontend.

User Story #3

As a user, I should be able to delete my account.

Completed: Again this user story took more time on the backend than on the frontend, but was successfully completed. The backend was able to successfully connect the backend code to the database as well. On the frontend the UI just had a button on the profile page, which was then integrated with the backend. Upon integration, the user story was tested both on the backend and the frontend.

User Story #4

As a user, I should be able to verify my email when I create an account.

Completed: This user story was slightly difficult as even though we generally knew what we were trying to implement, we were not sure how to specifically display it on the frontend. That took us some time to figure out. We went with a simple and to the point implementation though and were successfully able to complete the story. The user story was successfully tested on both the backend and the frontend as well.

User Story #5

As a user, I should be able to login into my account.

Competed: This story was not difficult to implement on the backend once we had the ‘create an account’ user story successfully implemented. The frontend followed the same look and feel for the UI as the ‘Signup’ page. Integration also was also successfully done and the user story was also successfully tested on both the backend and the frontend.

User Story #6

As a user, I should be able to logout of my account.

Completed: The user story was quick to implement. On the backend it was to the point. On the frontend the ‘Logout” button was implemented on the ‘Accounts’ page. The integration was also quick. This feature was successfully tested on the frontend and backend as well.

User Story #7

As a user, I should have an option to login using my Google account.

Completed: This user story took some time to figure out on the backend due to the differences in how this operated versus the general ‘Create an account.’ However, after putting in time and debugging through the issues, the backend team was successfully able to complete this story. The feature was quick to be implemented on the frontend as well with a button being added to both the ‘Signup’ and ‘Login’ page that allows for Google login. Testing was successfully done on this feature as well.

User Story #8

As a user, I should have an option to login using my Facebook account.

Completed: Once user story 7 was implemented, any challenges related to this story on the backend were quick to be solved. The frontend implementation was also quick as it was similar in nature to user story 7. Testing was successfully performed for this feature as well.

User Story #9

As a user, I should be able to view my profile page.

Completed: This user story definitely was very time consuming in nature and a significant one to implement for the frontend. The team spent time trying to figure out what the UI should look like and how to make it as simple and sleek as possible. The backend worked smoothly and the feature was successfully working upon integration. The user story was successfully tested on the backend and the frontend.

User Story #10

As a user, I should be able to change aspects about my profile - including my name, password, email, category preferences and number of previous searches - when on my profile page.

Completed: The user story took some time to figure out on how to implement the category preferences feature. There were multiple ways to do it, however, choosing one simple way took a lot of discussion. Upon deciding on one final way, we had a lot of debugging to do on the frontend while implementing this. The user story was completed successfully upon integration with the backend, which was working smoothly. This user story was successfully tested as well.

User Story #11

As a user, I should be able to reset my password in case I cannot remember my password.

Completed: This user story took a little bit of time to implement on the as two separate pages had to be created, one where the user enters their email and then an email is sent to them with a link to another page where the password is reset. The format of how we will implement this also took discussion to finalize. At the end, the user story was successfully implemented and tested on the fronten

d and backend.

User Story #12

As a user, after logging in, I should be able to view the front page - which would serve as the central page for displaying the search bar, the search results and my various preferred categories for displaying the search results in.

Completed: This user story took a while to complete as well and a lot of discussion with the frontend team on how the look and feel should be. The goal was to keep a similar look for all the three 'Search', 'Trending', and 'Dashboard' pages. After discussion, the UI was implemented successfully and tested well on the frontend.

WHAT DID NOT GO WELL?

Some of the technology we worked on during this sprint were new to us. While we understood that there would be a learning curve, we didn't estimate the amount of time it would take us to get accustomed with these technologies. This led to delay in development and production. We also planned to have our application deployed completely during this sprint. Due to delayed development, we weren't able to accomplish this.

Another issue that we faced was related to our branching strategies with our version control, namely Git, working via Github. Even though we initially planned for branching strategies that would allow all of us to work separately, we still ran into some overlapping development - which slowed our progress down, be it from fixing merge issues, to using a suboptimal branching strategy at times. This wasn't in alignment with software development best practices that our team aims to uphold.

The last concern we had for this sprint was regarding writing more unit tests. Although we implemented unit tests that get us a good code coverage, we would like to see our current numbers increase.

HOW TO IMPROVE?

One of the first order of business for us would be to separate out more code into smaller, modular pieces with a single responsibility. This would allow us to efficiently “quantize” our code - allowing us to cover a greater surface area with unit testing. We will also push for writing more code with the best coding practices in mind which would allow us to have a pristine and industry-standard codebase, allowing easy extension and maintenance.

Another important area where we want to make strides improving is our integration strategy - we aim to integrate more early and more often than before. This would ensure that we will avoid any undue stress at the end of a sprint because of integration because of mismatched interfaces on the frontend and the backend, and will allow us to see our product come together throughout the sprint - further improving team morale.

Furthermore, we will also take a deeper look and formalizing our branching strategy - and making it ironclad so that we can achieve the greatest developer productivity, with minimal interruptions due to merging issues. This will ensure that we can produce more code at an unprecedented pace, allowing us to be successful in this sprint.

Finally, an ever present area we can improve on as a team is identifying how to spread out our responsibilities so that we can achieve better time management for all team members, while working with a comfortable buffer for dealing with any unforeseen complications that may arise due to any number of unknown factors which are common whilst developing software - some of the common ones we faced being problematic third party integrations and deprecated framework features.