Betterflye CMS

Closure

Requirements matching

Initial Features

- If inappropriate content is drafted, the content filter prevents it from being posted.
- If a user's post is blocked, they can make an appeal to an admin
- If inappropriate content bypasses this filter, then users can manually flag the content.
- Admins shall receive alerts when content is blocked
- If content is flagged by users, the content is temporarily removed from the platform until admin review
- The system shall accept user input as text or photo file
- The system shall evaluate the submitted content before it is posted
- The admin shall be able to approve or reject content
- The user shall receive an alert if their upload is blocked by the filter

Final Features

- If inappropriate content is drafted, the content filter prevents it from being posted.
- If a user's post is blocked, they can make an appeal to an admin
- If inappropriate content bypasses this filter, then users can manually flag the content.
- Admins shall receive alerts when content is blocked
- If content is flagged by users, the content is temporarily removed from the platform until admin review
- The system shall accept user input as text or photo file
- The system shall evaluate the submitted content before it is posted
- The admin shall be able to approve or reject content
- The user shall receive an alert if their upload is blocked by the filter
- Image filtering via predictive model
- Super admin blocked post tools
- Super admin whitelist/blacklist management
- Organization admin user banning
- Organization admin post approval / flag view / blacklist / post deletion

Requirements changes

Added requirements

- Image filtering via predictive model
- Super admin blocked post tools
- Super admin whitelist/blacklist management
- Organization admin user banning
- Organization admin post approval / flag view / blacklist / post deletion

Changes came from client and mentor feedback, mostly based on their interactions with clients and what they wanted most

Final Meeting

Will take place on Wednesday, April 28 from 1-2 PM

Transfer

TBD

Unsuccessful Steps

- Blacklist was originally a text file in the project
    - Since the text file was in the project each of us had to make sure the file path was set correctly
- Absolute Paths for most ‘include’ statements earlier on
    - These had to be changed for each team members testing environment and led to time wasting bugs if left unset

Successful Steps

- Creating separate project
  - our project was called within a single location, making it easy to troubleshoot
- Integration with existing project
  - visually matched existing Betterflye website
  - client and mentor said our contributions fit in well and did not disrupt the flow of the website
- UI and backend testing
- Clean code
  - followed existing naming conventions
  - separate classes for each feature
- Teamwork and communication
  - all client’s requests were met

Actions/Steps We Missed

- Refactoring library to be more modular
- More UI testing earlier on

Final Thoughts - Daniel

- I’m very happy that we accomplished everything we wanted to and that the client is happy, as well.
- The process was long, but it felt good to be able to contribute to a real world project and make great progress along the way.

Final Thoughts - Dan

- Overall I am happy with the project and I feel we accomplished a lot of the things we wanted

Final Thoughts - Jake

- Very pleased with the final state of the project, created many features and is satisfying that these are all things that will be very useful to the client

Final Thoughts - Sara

- Pleased with how project turned out
- Working with existing code base is more likely when first getting hired than creating a new project, glad for the experience