

Critika

Team 8 - Sprint 1 Planning Document
Brandon Sung, Cyrus Santiago, Connor Todd
Keith Tan, Ken Sodetz, Julien San Diego

Sprint Overview:

The main goal of the first sprint is to set up the barebones platform for logging in, storing user data, pulling user data, creating a Front End UI, and deployment. The back end team will focus on setting up the database, login, and deployment. The front end team will focus on the logical layout and general design of the website.

Scrum Master: Keith Tan

Meeting Plan: Tuesdays/Thursdays (6:00PM-7:30PM)

Risks and Challenges:

While there are fairly experienced programmers, there are some members who have not worked on projects outside of class and are unfamiliar with certain technologies such as MongoDB, React, Node.js. As there's a gap in terms of technical skills, there will be a lot of work to help ramp up the technical skills of the less-experienced members so they will be able to contribute.

Current Sprint Details

User Story #1 - 28 hrs

As a user, I would like to be able to register and verify a Critika account.

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Set up Back End server	3	Back End	Keith
2	Set up Mongo database	2	Back End	Keith
3	Set up deployment server	2	Back End	Keith
4	Create UI form to enter information about user	2	Front End	Julien
5	Create button to send information to Back End to register	1	Front End	Julien
6	Create API route to register user and add them to database	6	Back End	Keith
7	Implement API route to generate a verification token	6	Back End	Ken
8	Implement mailer to send verification token	3	Back End	Ken
9	Create Unit Tests	3	Back End	Connor

Acceptance Criteria:

- ❖ Given that the Back End to create users is created successfully, when a request to create an account is made then the user's account will be added to the database given that the email has not been taken and the password is strong.
- ❖ Given that the Back End and database has been created successfully, when a request to create an account with an email address already existing in the database the user interface and Back End response will alert the user.
- ❖ Given that mailer is implemented correctly, when a request is made to create a user account, a email should be sent to the given email address to with the verification token to verify the account.

- ❖ Given that a user has successfully create an account, an authentication token will be generated

User Story #2 - 13 hrs

As a user, I would like to be able to login to my Critika account

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create UI form to for entering login information	2	Front End	Julien
2	Create button to initiate login	1	Front End	Julien
3	Create API route to login user and generate authentication token	5	Back End	Keith
4	Allow user to maintain a session on web client	2	Back End	Keith
5	Create Unit Tests	3	Back End	Connor

Acceptance Criteria:

- ❖ Given that a user already has an account, login fields and buttons should be accessible
- ❖ Given that the user has an account and the Back End was successfully implemented, then the user should have an access token for maintaining a web session while using the application
- ❖ Given that a user has successfully logins into an account, an authentication token will be generated

User Story #3 - 15 hrs

As a user, I would like to be able to reset my password if I forget it

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create form and screen to reset old password to new password	2	Front End	Julien
2	Create button on login page to reset password	1	Front End	Julien
3	Set up mailer to send emails to users	2	Back End	Ken
4	Send reset token to user's email	2	Back End	Ken
5	Create API route to change user's password in the database	5	Back End	Keith
6	Create Unit Tests	3	Back End	Connor

Acceptance Criteria:

- ❖ Given that a user is on the login page, a button to reset their password should be accessible
- ❖ Given that the user has accessed their reset token, changed their password, and the Back End routes were implemented successfully, then the user's password should be changed to the new one on the database
- ❖ Given that a user has successfully authenticated themselves and has requested a password reset, an e-mail should be sent to their stored e-mail address

User Story #4 - 10 hrs

As a user, I would like to be able to view to my Critika profile

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create profile screen where user-specific attributes are displayed	2	Front End	Julien
2	Create buttons or clickable elements that navigate to profile screen from dashboard	2	Front End	Julien
3	Create API route to retrieve profile information	3	Back End	Keith
4	Create Unit Tests	3	Back End	Connor

Acceptance Criteria:

- ❖ Given that a user has successfully logged into their account, a user should be able to view all their account details (name, summary, etc.)
- ❖ Given that a user wants to look at their account information, they need to pass in an authentication token to see it
- ❖ Given that a user has successfully logged into their account, the information given should not include any unnecessary information

User Story #5 - 10 hrs

As a user, I would like to be able to edit my Critika account

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create edit button on profile page	1	Front End	Julien
2	Create UI form where all user info is editable	3	Front End	Julien
3	Create submit button for when user is done editing	1	Front End	Julien
4	Create API route to update	2	Back End	Keith

	user info in the database			
5	Create Unit Tests	3	Back End	Connor

Acceptance Criteria:

- ❖ Given that the profile information has been implemented correctly, when a user goes to their profile to view their information, there should be an edit button
- ❖ Given that there is an edit button on the profile page, when clicked, the edit button should make the user's personal info editable, and a submit button must appear in order to correctly save changed
- ❖ Given that the edit profile interface was correctly implemented, when the user hits submit, the changed must be sent to and stored in the database, and the old information must either be deleted or overwritten and replaced with the new, edited version

User Story #6 - 11 hrs

As an admin, I would like to be able to view all registered users

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create UI for seeing all registered users	2	Front End	Cyrus
2	Create search bar and filter for finding specific users	1	Front End	Cyrus
3	Create UI for viewing a registered users reported submissions and comments	2	Front End	Cyrus
4	Create option menu for admins to manage accounts	1	Front End	Cyrus
5	Create route from database of all current registered users and their report info	2	Back End	Keith
6	Create Unit Tests	3	Back End	Connor

Acceptance Criteria:

- ❖ Given that the admin login works, admins should be able to login to an admin page to view all users
- ❖ Given that the backend is successfully implemented, only admins should be able to see all users
- ❖ Given the frontend was successfully created, there should be a list that contains all the users

User Story #7 - 9 hrs

As a user, I would like to be able to add other users so that I can collaborate with them further. (9 hours)

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a button to add another user as a friend	1	Front End	Cyrus
2	Create UI for viewing a list of all friends	2	Front End	Cyrus
3	Create a button to accept another user's friend request	1	Front End	Cyrus
4	Update database user object to store other user in array of friends	2	Back End	Connor
5	Create Unit Tests	3	Back End	Connor

Acceptance Criteria:

- ❖ Given that a user is viewing a profile that is not their own, an "add friend" button should be accessible
- ❖ Given that a user has received a friend request, a "accept friend request" button should be accessible
- ❖ Given that a user has friends, all friends should be viewable on the same screen

User Story #8 - 11 hrs

As a user, I would like to be able to submit a form for submission. (11 hours)

#	Task Description	Estimated Time	Team	Owner
---	------------------	----------------	------	-------

		(Hours)		
1	Create button to submit a new post to be critiqued	1	Front End	Cyrus
2	Create UI form where a user can upload their submission	3	Front End	Cyrus
3	Create submit button that will upload the user submission	1	Front End	Cyrus
4	Update database user object to include the submission	3	Back End	Keith
5	Create Unit Tests	3	Back End	Connor

Acceptance Criteria:

- ❖ Given that a user wants to submit a post to be critiqued, a new submissions button should be accessible.
- ❖ Given that a user has clicked the new submission button, they should be directed to a new submission page where the user can add a title and text, with a submit button that will post their submission
- ❖ Given that the backend routes are implemented, once they submit, the database should add the new user submission

User Story #9 - 14 hrs

As a user, I would like to be able to easily see submissions from other users I can give feedback to. (14 hours)

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create dashboard page button	1	Front End	Cyrus
2	Create dashboard page logo link	1	Front End	Cyrus
3	Create API query to receive	3	Back End	Keith

	other user submissions			
4	Parse information from database and send a response to the client	3	Back End	Keith
5	Create home page UI where other user's submissions are shown	3	Front End	Cyrus
6	Create Unit Tests	3	Back End	Connor

Acceptance Criteria:

- ❖ Given that a user has posted a submission, a queue of other user submissions should be displayed
- ❖ Given that the backend route has been implemented correctly, the user should be able to see all the available submissions
- ❖ Given that a user is on the explore page, filters should be available to specify what types of other user submissions to display

User Story #10 - 12 hrs

As a user, I would like to be able to easily access my submissions. (9 hours)

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create button to navigate to submissions from the profile screen	1	Front End	Brandon
2	Create submissions page where all past submissions can be viewed	3	Front End	Brandon
3	Create dropdown filters to specify which types of submissions to view	2	Front End	Brandon
4	Create backend route to retrieve submissions for the specific user	3	Back End	Keith
5	Create Unit Tests	3	Back End	Connor

Acceptance Criteria:

- ❖ A user should be able to access the submissions page from their profile page where they can view all of their submissions in one place
- ❖ Given that a user has multiple submissions, a dropdown filter should be accessible on the submissions page to specify what type of submissions to display
- ❖ Given that submissions exist for a user, the submissions that the user sees will match their submission that are in the database

User Story #11 - 11 hrs

As a user, I would like to be able to remove submissions as I see fit. (14 hours)

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create button to “remove” submission	1	Front End	Brandon
2	Create option window to confirm removal	2	Front End	Brandon
3	Remove submission from the database	3	Back End	Ken
4	Update user statistics to reflect submission removal	2	Back End	Ken
5	Create unit tests to confirm addition and removal is successful	3	Back End	Connor

Acceptance Criteria:

- ❖ Given that a user wants to remove a submission, they can specify which one through the submission name
- ❖ Give that a submission has been selected for removal, the submission will be removed from the database
- ❖ Given that a submission is removed, the user’s statistics (number of submissions, number of critiques) will be updated to reflect the submission’s removal

User Story #12 - 11 hrs

As a user, I would like to be able to edit submissions as I see fit. (7 hours)

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create button to “Edit”	1	Front End	Brandon
2	Create options window to edit submission attributes	3	Front End	Brandon
4	Update changes in the database	4	Back End	Keith
5	Create Unit tests	3	Back End	Connor

Acceptance Criteria:

- ❖ Given that the user has selected a submission to edit, the user will be able to edit the description, category, and skill level for that submission
- ❖ Given that a user wants to edit a submission, it must only allow them to edit their submissions
- ❖ Given that a submission has had edits made to it, the edits will successfully be made in the database

User Story #13 - 6 hrs

As a user, I would like to be able to specify my skill level for submissions. (6 hours)

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create slider for skill level	1	Front End	Brandon
2	Add skill level to submission object	2	Back End	Ken
3	Create Unit Tests	3	Back End	Connor

Acceptance Criteria:

- ❖ Given that the user interface was implemented correctly, all submissions will contain a field for skill levels (beginner, intermediate, expert) in order to categorize that particular submission for feedback

- ❖ Given that the skill level was added to the submission object the submission skill level will be accessible
- ❖ Given that the skill level has been added, the skill level will be stored with the submission in the database

User Story #14 - 16 hrs

As a user, I would like to be able to see my Critika rating on my Profile page. (11 hours)

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create a display for the user's Critika rating	2	Front End	Brandon
2	Create API route to create/update the user's rating in the database	4	Back End	Ken
3	Calculate rating based on number of ratings received and the sum of all ratings	2	Back End	Ken
4	Create Unit Tests	3	Back End	Connor

Acceptance Criteria:

- ❖ Given that the user interface is implemented correctly, the user will be able to see the Critika rating on the Profile Page
- ❖ Given that ratings exist, the score will be tallied in the database
- ❖ Given that new ratings are added to a user, they will be added to the existing rating

User Story #15

As a user, I would like to be able to easily view all categories for submissions. (5 hours)

#	Task Description	Estimated Time (Hours)	Team	Owner
1	Create user interface showing all possible categories	3	Front End	Cyrus
3	Create dropdown option on submission creation that shows all	2	Front End	Cyrus

	possible categories			
--	---------------------	--	--	--

Acceptance Criteria:

- ❖ Given that the user interface is implemented correctly, the user should be able to navigate to a portion of the application that shows all the categories for submissions
- ❖ Given that the dropdown is implemented correctly, the user should be able to easily view all the categories when submitting an application for feedback
- ❖ Given that the user interface is implemented correctly, all possible categories should match in every area that shows all categories