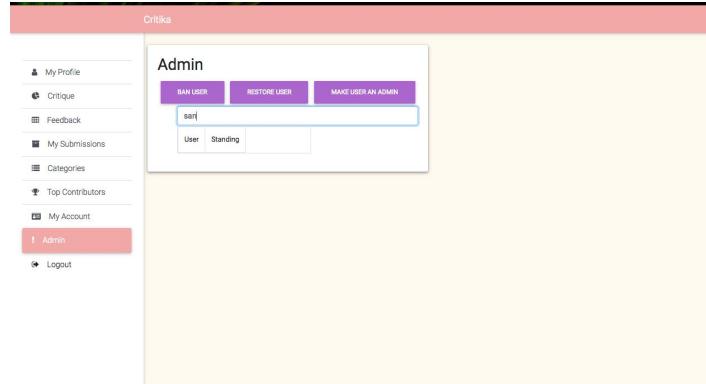


## Sprint 3 Manual Testing Document

**Testing User Story 1:** As an admin, I would like to be able to see all registered users.

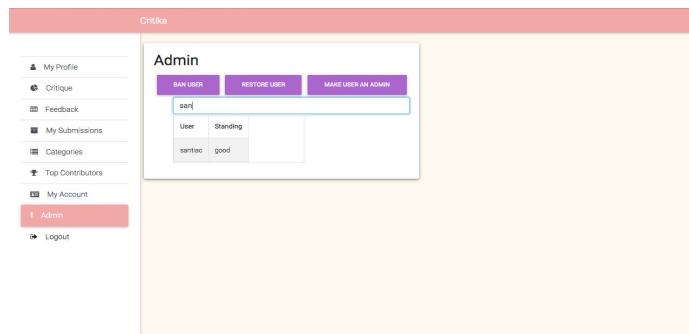
First test: when user is not an admin, the user won't be able to search for users. Request does not trigger backend route, and search filter does not get triggered either.

Here, the user is not an admin, and when searching, no users are displayed under the search bar (there is a user called santiac).



Second test: When a user is an admin, the user will be able to search for other users on the admin page, and see users matching the search input in real time.

In this picture, the user (santiac) shows up. The user was set to be an admin, and therefore can search the website and database for users.



This was a simple one to test, as there is a backend field called status which either has values 'admin' or 'reg-user'.

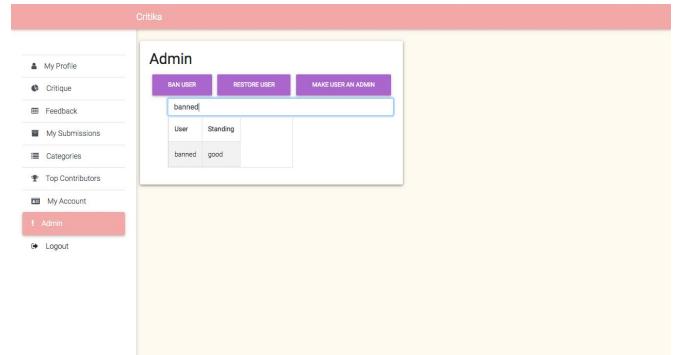
A user is set to 'reg-user' after registration by default, so this permission will be locked to most. The route in the backend that handles this is get(/users/all-users), which is tested by the backend script, and only returns a success if the user is an admin.

**Testing User Story 2:** As an Admin I would like to be able to ban, restore, and make users admins.

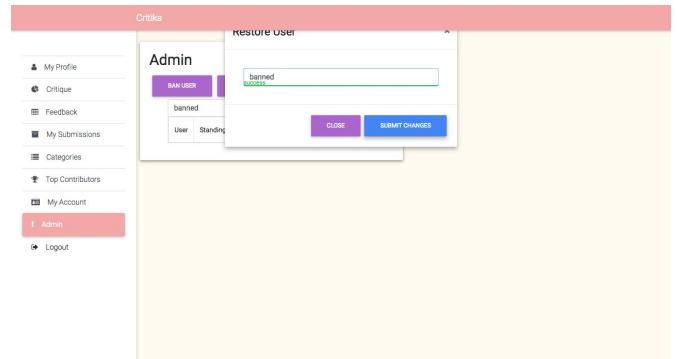
First test: When user is not an admin, they will not be able to ban, restore, or make other users admins.

Second test: When user is an admin, they can ban and restore users.

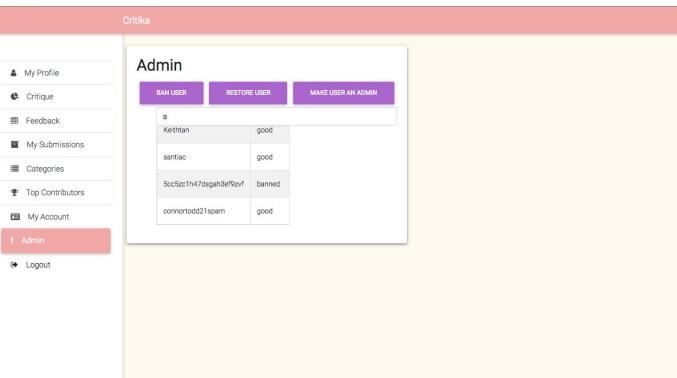
In this first image, the user (banned) has a ‘good’ standing. Standing is a binary, and a user is either in good standing or is banned from the website.



Since the user is an admin, when they try to ban another user (in this case the user named ‘banned’), they are shown success. When the user bans another user, the now-banned user’s standing changes from ‘good’ to ‘banned’, which is reflected in the database.



The way the ban user route works in the backend is that it essentially hashes the username, so the user will never be able to log in again. As you can see, the old user’s name was changed to a random string and their status was updated to banned.



Since banning is an admin only function, it will mostly be done on a per-user base. Admins will be able to search the database for the user to restore, and find their new hashed username to un-ban.

User	Status
Kethitan	good
santiac	good
Scc5zc1h47dsqah3ef9zvf	banned

After un-banning the user, their username is restored and their new status is reflected in the database.

User	Status
Kethitan	good
santiac	good
banned	good

Admins can also make other users admins as well, and the change will be reflected in the database. The updated user will also be able to use admin-only routes.

User	Status
banned	banned
Stand	stand
good	good

**Testing User Story 3:** As a user I would like to view all the categories that I can add my submission to.

First test: When a user types characters into the categories search bar, a list of categories will be displayed.

Since basketball and alphabet both contain the letter t, they are displayed underneath as clickable html objects. The search bar is in the add submission form, because every submission needs to be submitted to a category, and being able to search here is friendly to the user.

The screenshot shows a user interface for adding a submission. On the left is a sidebar with links: My Profile, Critique, Feedback, My Submissions (which is highlighted in red), Categories, Top Contributors, My Account, Admin, and Logout. The main area has a header 'Add Submission'. Below the header is a search bar containing 'Basketball' and 'Alphabet'. Underneath the search bar are input fields for 'Submission Name' (with validation 'Submission Name is Required'), 'Description' (with validation 'Description is Required'), and 'What feedback are you looking for?'. At the bottom of the form are buttons for 'Feedback is Required' and 'Skill Level' (set to 'Beginner').

When the category is clicked, the value of the category now occupied the input field for the form.

This screenshot shows the same 'Add Submission' form as above, but with a noticeable difference: the 'Category' input field now contains the value 'Basketball'. The rest of the form, including the search bar, validation messages, and skill level selection, remains the same.

**Test User Story 4:** As a user, I would like to be able to view and edit my profile.

**First Test:** When a user clicks the 'my profile', they can view their profile and see all of their profile information.

When the user clicks the 'my profile' tab, they are directed to the my profile page, where data is pulled from the database, and they can see all of their important user information that is stored in the backend.

The screenshot shows the 'My Profile' page of the Critika application. The left sidebar has a 'My Profile' section selected. The main content area displays the following information:

- About Me:** Hey! I'm new here!
- Homepage:** critika.com
- Occupation:** Critika User
- Location:** Crittstown
- Submissions:** 2
- Total Score:** 99996
- Critka Rating:** Not yet rated
- Member Since:** 2018-11-30

At the bottom right is a blue 'Edit Profile' button.

**Second Test:** When a user clicks 'edit profile', they are prompted to a new page where they can edit their profile information.

When the user clicks 'edit profile' they are redirected to a page and shown a form where they can edit various user profile attributes.

The screenshot shows the 'Edit Profile' page of the Critika application. The left sidebar has a 'My Profile' section selected. The main content area contains a form with the following fields:

- About Me:** (empty input field)
- Homepage:** homepage
- Location:** (empty input field)
- Occupation:** (empty input field)

At the bottom right is a red 'SUBMIT CHANGES' button with the word 'DANIEL' typed into it.

**Third Test:**

The screenshot shows the 'My Profile' page of the Critika application. The left sidebar has a 'My Profile' section selected. The main content area displays the following updated information:

- About Me:** Hey guys!
- Homepage:** www.espn.com
- Occupation:** CS 307 Student!
- Location:** West Lafayette
- Submissions:** 4
- Total Score:** 100036
- Critka Rating:** 4.666666666666667/5
- Member Since:** 2018-11-30

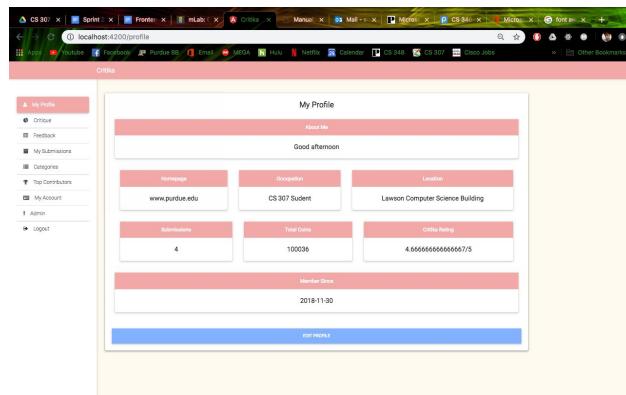
At the bottom right is a blue 'Edit Profile' button.

The screenshot shows the 'Edit Profile' page of the Critika application. The left sidebar has a 'My Profile' section selected. The main content area contains a form with the following updated fields:

- About Me:** Good afternoon
- Homepage:** www.purdue.edu
- Location:** Lawson Computer Science Building
- Occupation:** CS 307 Student!

At the bottom right is a red 'SUBMIT CHANGES' button with the word 'DANIEL' typed into it.

As you can see, when a user edits their info, the database and website is updated with the new information that they wanted to change. The changes are reflected on the profile page.



### Test User Story 5: As a user, I would like to view/edit my account

First Test: When a user clicks the 'my account' tab, they are directed to the my account page

When the user clicks the my account tab, they are redirected to the my account page, where they can see all of their important account info

A screenshot of a web browser showing the "Edit Account Information" page. The URL is localhost:4200/profile. The page has a sidebar with a "My Account" tab selected. The main form is titled "Edit Account Information - Username: santiac". It contains three sections: "Email" (xrbubic@gmail.com), "Password" (Change My Password, New Password, Confirm Password), and "Security Question and Answer" (What is your mother's maiden name?, New Security Question Answer, Security Question Answer). Below the form are three buttons: "CHANGE EMAIL", "CHANGE PASSWORD", and "CHANGE SECURITY QUESTION".

Second Test: When a user clicks ‘change email’, they are able to change/update their email.

The image consists of three side-by-side screenshots of a web application's account edit page. Each screenshot shows a different step in the process of changing the email address.

- Screenshot 1:** Shows the initial form with the current email as "x@x.x" and a placeholder "Please Enter Email". Below the form are three buttons: "CHANGE EMAIL", "CHANGE PASSWORD", and "CHANGE SECURITY QUESTION".
- Screenshot 2:** Shows the same form after entering a new email "testEmail@gmail.com" into the "New Email" field. The "CHANGE EMAIL" button is now highlighted in blue, indicating it is the active button.
- Screenshot 3:** Shows the final state where the email has been successfully changed. A green success message box at the bottom left of the form area says "Email Successfully Changed".

This sequence shows the progression of how a user updates their email. They have to type in a valid email into the textbox above the change email button (there is form validation as seen on the last picture), and when they submit a valid email, they will see success upon a success. After the success, the email is updated in the database and reflected as seen in picture 2. Picture 3 shows an example of the form validation and the updated email from picture 2.

Third Test: When a user clicks ‘change password’, they are able to change/update their password.

The image consists of two side-by-side screenshots of a web application's account edit page. Each screenshot shows a different step in the process of changing the password.

- Screenshot 1:** Shows the initial form with the current password as "test123" and a placeholder "Please Enter New Password". Below the form are three buttons: "CHANGE EMAIL", "CHANGE PASSWORD", and "CHANGE SECURITY QUESTION". A red error message box at the bottom left says "Passwords Do Not Match".
- Screenshot 2:** Shows the same form after entering a new password "test123456" into the "New Password" field and confirming it in the "Confirm Password" field. The "CHANGE PASSWORD" button is now highlighted in blue, indicating it is the active button.
- Screenshot 3:** Shows the final state where the password has been successfully changed. A green success message box at the bottom left of the form area says "Password Successfully Changed".

Similar to the second test, when a user wants to change their password, if their passwords match, they are able to change passwords, and this change is reflected in the database. There

is form validation to make sure the passwords match, and if they don't, the user won't be able to submit.

Fourth Test: When a user clicks 'change security question', they are able to change/update their security question and answer.

The screenshots show the 'Edit Account Information' page for a user named 'santac'. The first screenshot shows the initial state with a red box around the 'Change Security Question' button. The second screenshot shows the dropdown menu open, listing various security questions. The third screenshot shows the 'Security Question and Answer' section with a green success message at the bottom stating 'Security Question and Answer Successfully Changed'.

Similar to above, a user can change their security question and answer. If the user does not fill in the security answer box, they will not be able to change their security question and answer. The drop down menu for security question contains a list of security questions, and when the user submits a valid answer, the website shows them a success and the database is updated accordingly.

**Test User Story 6:** As a user, I would like to be able to view all the feedbacks I have received

First Test: When a user clicks 'feedback', they are redirected to the feedback page where they can see all of their feedbacks

When the user clicks the feedback tab, they are directed to the feedbacks page as seen on the right. All of the feedbacks that their submissions have received are shown, along with the feedback specific information.

The screenshot shows the 'My Feedbacks' page with a table displaying two rows of feedback information. The columns are: Submission ID, User, What went well, What went wrong, What could be improved, and Rate & Feedback. The first row has submission ID 'testing123', user 'bsung1234', 'good form', 'testing testing', 'frequency', and a 'RATE' button. The second row has submission ID 'testing123', user 'Keltisan', 'Pretty good test!', 'Video quality suffers past 3 minutes', 'Must be my computer', and a rating of '5/5'.

**Second Test:** When a user clicks rate feedback, they can give the feedback they've received a rating from 1-5.

The three screenshots illustrate the user flow for rating feedback:

- Screenshot 1:** Shows the "My Feedbacks" page with a list of feedback items. One item is selected, showing its details: "User: beng123, What went well: good form, What went wrong: being wrong, What could be improved: less feedback". Below the details are two buttons: "RATE" and "DELETE".
- Screenshot 2:** Shows the same "My Feedbacks" page after the "RATE" button has been clicked. A dropdown menu is open, allowing the user to select a rating from 1 to 5. The "RATE" button is now grayed out.
- Screenshot 3:** Shows the "My Feedbacks" page again, but the feedback item now includes a new column "Rate": "User: beng123, What went well: good form, What went wrong: being wrong, What could be improved: less feedback, Rate: 45".

The way to rate feedback is shown above. First the user must be able to click the rate button (as seen in the first test), and on click they must be shows a way to rate from 1-5. The drop down menu makes the user choose a number 1-5, and when the user clicks the rate button again, the feedback receives a rating, which is updated in the database as seen in the third picture.

**Test User Story 7:** As a user, I would like to be able to view, edit, and add submissions

**First Test:** When a user clicks the my submissions tab, they are directed to a page where they can view all of their submissions

When the my submissions tab is clicked, the user is redirected to the my submissions page, and the proper data is pulled from the backend and displayed for the user to see. In this case there is only one submission.

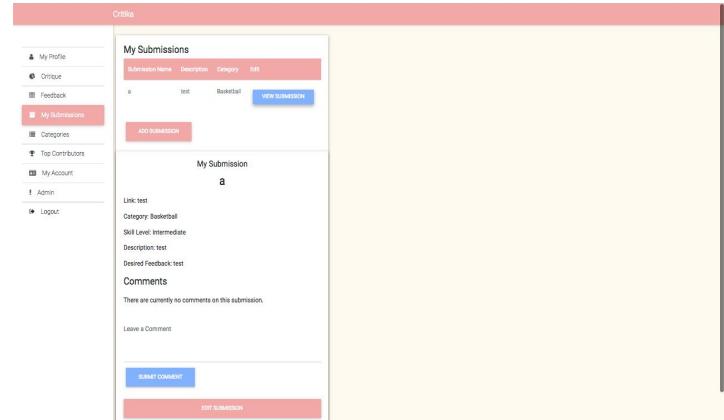
The screenshot shows the "My Submissions" page with a single submission listed:

Submission Name	Description	Category	Action
test		Basketball	<a href="#">VIEW SUBMISSION</a>

Below the table is a red "ADD SUBMISSION" button.

Second Test: When a user clicks the view submissions tab, they are able to view the submission more easily and see more information

When the user clicks view submission, they can see more submission information, including comments left by other users, and an option to add a comment (to respond to others), and edit the submission.



Third Test: When a user clicks edit submission, they are redirected to a form to edit their submission, and when they edit the submission, the database is updated accordingly.

My Submission  
a

Link: test  
Category: Basketball  
Skill Level: Intermediate  
Description: test  
Desired Feedback: test  
Comments  
There are currently no comments on this submission.  
Leave a Comment:  
  
**SUBMIT COMMENT**

Link: testing the link  
Category: Basketball  
Skill Level: Intermediate  
Description: im testing the description  
Desired Feedback: im testing the feedback  
Comments  
There are currently no comments on this submission.  
Leave a Comment:  
  
**SUBMIT**

My Submissions  
a

Link: im testing the link  
Category: Basketball  
Skill Level: Intermediate  
Description: im testing the description  
Desired Feedback: im testing the feedback  
Comments  
There are currently no comments on this submission.  
Leave a Comment:  
  
**SUBMIT COMMENT**

Link: im testing the link  
Category: Basketball  
Skill Level: Intermediate  
Description: im testing the description  
Desired Feedback: im testing the feedback  
Comments  
There are currently no comments on this submission.  
Leave a Comment:  
  
**SUBMIT**

CLOSE

When a user clicks edit submission, they are given an option to edit certain submission information. When they hit submit, the page redirects them back to the my feedback tab, and the database and website are updated accordingly. The submission info on the right and the form fields on the left match.

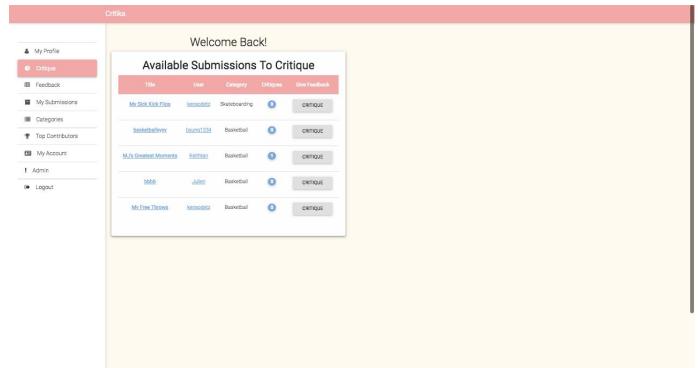
Fourth Test: When a user clicks add submission, they can add a submission and it will be reflected on the website.

In the first image, the add submission form pops up when you click the add submission button. Similar to earlier, there is a search function for categories, as seen with the blue text boxes. When the user fills out the correct information (picture 2) and hits add submission, the submission is added to the database with the correct information, as seen in the third and fourth images. There is form validation so null or blank submissions cannot be added, and a category is a required option.

**Test User Story 8:** As a user, I would like to have a dashboard that shows submissions I can critique.

First Test: When a user clicks the critique page, they are taken to the dashboard that displays available submissions in the queue that the user can critique (give feedback to).

The picture on the right shows the result of the user clicking the critique tab. They are redirected to a page where they can see available submissions to critique, along with links that redirect to the user and the submission. The 5 submissions displayed are generated from the queue, specifically for this user.



Second Test: When a user clicks critique, they will be able to provide feedback to the submission, and the submission will be removed from their queue and replaced by another one if there are enough available.

This screenshot illustrates the workflow for providing feedback. On the left, a modal window titled 'Title: My Sick Kick Flips' is open, prompting the user to enter feedback. The input field contains the text: 'Feedback on form', 'Nice form on the jump, really great balance and concentration.', 'Work on hand and neck stability.', and 'Confidence, you need more practice!'. At the bottom of the modal are 'DONE' and 'CANCEL' buttons. On the right, the 'Available Submissions To Critique' dashboard is shown, identical to the one in the previous screenshot, displaying the same five submissions.

The picture on the left shows where you are redirected when you click the critique button on a submission. Once you fill out the form and submit valid information, you are redirected back to the critique tab, where the previous submission was removed (one of our rules is that you can only give one feedback per submission, and if you want to continue communication you can comment instead). When there are more than 5 available submissions, the available submissions will show 5 submissions. Since our database is not too populated, there are only four available.

## Test User Story 9: As a user, I would like to be able to login, logout, and register

First Test: When a user logs in, they must supply the correct info, and are taken to the website upon a successful login.

The first screenshot shows the login page with 'incorrect' entered into both the username and password fields. A red error message at the bottom states 'Username or Password is incorrect'. The second screenshot shows the same login page but with 'testing123' in the username field and 'incorrect' in the password field. The third screenshot shows the critique page with a list of submissions, indicating a successful login.

When a user supplies invalid info (left) they cannot login and are told the error. Upon successful login, the user gets assigned an auth token and is taken to the critique page. Their auth token is updated, added to local storage, and is counting down to expire too.

Second Test: When a user wants to register, they are taken to the register page, and can register an account.

The left screenshot shows the registration form with all fields filled: 'testing123' for Username, 'incorrect' for Password, 'incorrect' for Confirm Password, 'testing@gmail.com' for Email, and 'orange' for Security Question Answer. The right screenshot shows the same form after submission, with a green success message 'Account Successfully Created' displayed below the registration button.

The top left picture shows what the register page looks like on immediate redirect from the login page when the user clicks register. The second image (top right) shows a successful registration, and the user is added to the database and after verifying their email, they can login. The picture to the right shows the form validator for registration, to make sure all fields are input.

This screenshot shows the registration form with validation errors: 'Email is required' for the Email field and 'Security Question Answer is required' for the Security Question Answer field. The other fields are correctly filled: 'testing123' for Username, 'incorrect' for Password, 'incorrect' for Confirm Password, and 'orange' for Security Question.

### Third Test: When a user logs out, they are taken back to the login page

The top left image is the website before the user logs out. Their auth token is still valid, so they can access the pages and routes. After clicking the logout button (notice the cursor is on top of it), the user is redirected back to the login page, and the auth token is removed from local storage.

The top left image is the website before the user logs out. Their auth token is still valid, so they can access the pages and routes. After clicking the logout button (notice the cursor is on top of it), the user is redirected back to the login page, and the auth token is removed from local storage.

### **Test User Story 10:** As a user, I want to see a leaderboard ranking of users on the website

First Test: When a user clicks the top contributors tab, they are redirected to the top contributors page and can see the top contributors sorted.

When a user clicks the top contributors tab, they are redirected to the top contributors page, which shows the top 10 Critika users sorted by their rating. Since there are currently only 9 users, only 9 display. But if there were more than 10, only the 10 with the highest ratings would be displayed.

User	No. Submissions	No. Critiques	Rating
santosh	4	12	5/5
testtess	2	2	5/5
testtest1234	9	2	4/5
team003	0	0	0/5
team001	0	0	0/5
team002	4	5	0/5
testtest123	0	0	0/5
Julien	2	1	0/5
connected421user	5	1	0/5

Second Test: When a user clicks the rating, it will sort by top and bottom contributors

When the icon is clicked, the users with the lowest scores are displayed. Since there are only 9 users, this just displays the rating in reverse from above, showing the correct functionality.

User	No. Submissions	No. Critiques	Rating
santosh	4	12	5/5
testtess	2	2	5/5
testtest1234	9	2	4/5
team003	0	0	0/5
team001	0	0	0/5
team002	4	5	0/5
testtest123	0	0	0/5
Julien	2	1	0/5
connected421user	5	1	0/5

**Test User Story 11:** As a user, I would like to be able to view another user's profile

**First Test:** When a user clicks another user's username, they are directed to that user's profile

The left screenshot shows the homepage of the Critika website. It features a navigation bar with links for My Profile, Critique, Feedback, My Submissions, Categories, Top Contributors, My Account, Admin, and Logout. Below the navigation bar is a section titled "Available Submissions To Critique" with a table. The table has columns for Title, User, Category, Critiques, and Give Feedback. The first row shows a submission by user "8" titled "Skateboarding". The second row shows a submission by user "basketballyyy" titled "Basketball". The third row shows a submission by user "MJ's Greatest Moments" titled "Basketball". The fourth row shows a submission by user "bbbb" titled "Basketball". The fifth row shows a submission by user "My Free Throw" titled "Basketball". The right screenshot shows the profile page for user "connortodd21spam". The profile page includes sections for "About Me" (with the message "Hey! I'm new here!"), "Homepage" (critika.com), "Occupation" (Critika User), "Location" (Crittown), "Submissions" (5), "Critiques" (1), and "Critika Rating" (Not yet rated). At the bottom of the profile page is a "Member Since" section showing the date 2018-11-30 and a "RETURN" button.

When a user clicks another user's username (connortodd21 in this case), they are redirected to that user's profile (top right) and are able to see some user's profile. This can be done from any place on the website where another username is visible.

**Test User Story 12:** As a user, when I spend Critika coins on a submission, it should be placed into an availability queue

This is done in the backend, you can't check this on the front end because you aren't allowed to critique your own submissions.

**Test User Story 13:** As a user, I would like to be able to view other user's submission information

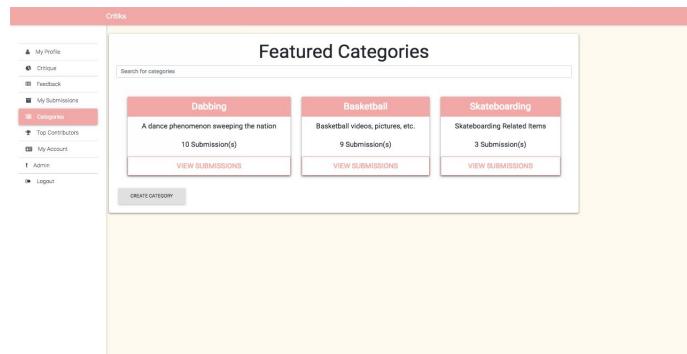
The left screenshot shows the homepage of the Critika website, identical to the one in the previous test. The right screenshot shows the detailed submission page for "MJ's Greatest Moments". The page includes a summary of the submission: Created By: Keithtan, MJ's Greatest Moments, Link: https://www.youtube.com/watch?v=vp0gZOLfzDM, Category: Basketball, Skill Level: Intermediate, Description: A video I made with Michael Jordan's highlights, and Desired Feedback: I'm looking for feedback on how I did syncing with the clips and the music. Below this summary is a comment section with two comments from user "Keithtan": "poop" and "Hello". There is also a "Leave a Comment" input field and a "SUBMIT COMMENT" button.

When a user clicks the submission name (in this case MJ's Greatest Moments), it goes to a page that shows the more detailed submission information, including comments. This works from anywhere where a user can see the submission title.

**Test User Story 14:** As a user, I would like to be able to view Critika categories

First Test: When a user goes to the categories page, the top feature categories are displayed

When the user goes to the categories page, the website displays the categories with the highest number of submissions, ordered. These are the top 3 categories, and a user can go to the categories page and view the submissions in the category.



Second Test: When a user types in the search bar, categories are displayed, and when a user clicks on the category, they can see the submissions in the category

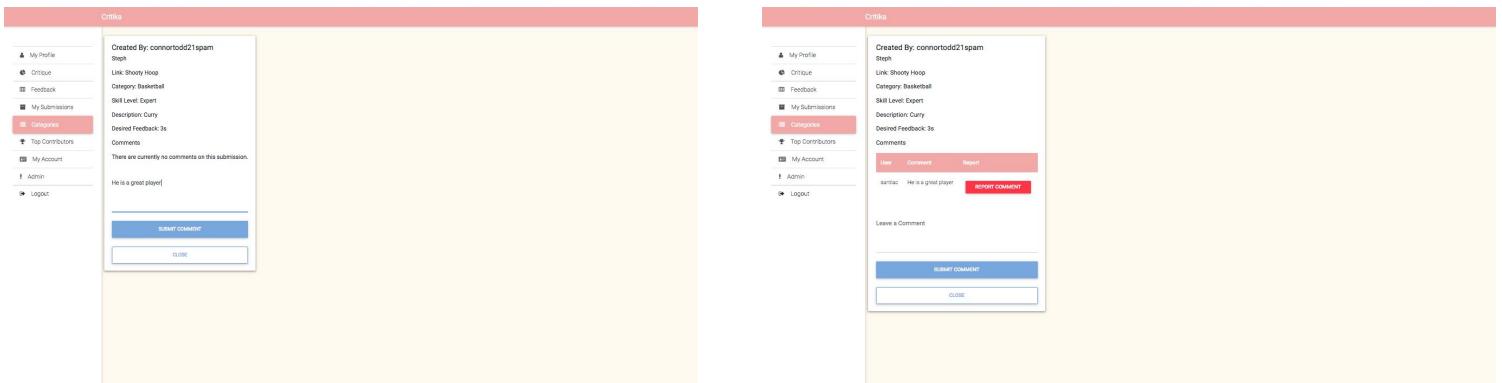
Two screenshots of the Critika website. The left screenshot shows the "Featured Categories" page with a search bar containing the letter "s". It lists categories: Basketball, Skateboarding, and Mixed Martial Arts. The right screenshot shows a close-up of the "Skateboarding" category card. The card has a title "Skateboarding", a description "A dance phenomenon sweeping the nation", "10 Submission(s)", and a "VIEW SUBMISSIONS" button. A tooltip for "Skateboarding" shows additional information: "Best place to do tricks near WAC Location", "a", "a", and "My Sick Kick Flips Feedback on form". The sidebar on the left is identical to the first screenshot.

On the left, when a user types in a letter, corresponding categories matching the typed in string are displayed. The picture on the right represents when you click on one of the categories from the search bar. The website displays the submissions from that category in a list for the user to see.

Third Test: When a user creates a category, it is added to the database and accessible to the user.

When the user clicks the create category button, a form pops up for them to create a category (they have to submit a name and description). When they add the category, the popup goes away, and the category is added to the database. This is shown in the picture to the right, where right after I add the submission, as it appears in the search result

## Test User Story 15: As a user, I would like to be able to comment on a submission



When you click on submit comment on a submission page, the comment gets added to the submission object and gets displayed properly when you navigate back to that submission object. We had an issue where the flag that indicates whether the comment was reported was not returning from the database. We fixed this by doing the computation in the front-end, trying to access parameters within an object within an object was proving to be too difficult and time consuming in the backend because of Mongo calling conventions. So we essentially removed the /get-all-reported-comments route and did the computation in the front end on the entire array of submissions. We sacrificed efficiency for functionality.