

User		
Test Case Number	Test Case 1	
Test Case Name	Create User BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are creating a new user and verifying it by logging in.	
Specifications		
Input	Expected Output	Notes
name: "Jane Doe", email: "email@mun.ca", username: "JD2", password: "password", student_id: "1", gender: "Female", campus: "stjohns"	After completing the registration process we log in as the new user.	
Procedural Steps		
Step 1	Login as Username: JD2, Password: password	
Step 2	Load webpage localhost:3000	
Step 3	Select "Register" from the navigation menu	
Step 4	Enter the input variables and click "Submit"	
Step 5	Login using the username and password	
Step 6		
Step 7		
Step 8		
Step 9		
Step 10		
Step 11		
Actual Output		Pass/Fail
The user is created and the dashboard is displayed after successfully logging in.		Pass

User		
Test Case Number	Test Case 2	
Test Case Name	Create User WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are creating a user object in the UI and verifying its existence by attempting to access that user’s profile page and attempting to log in as that user.	
Specifications		
Input	Expected Output	Notes
name: “John Doe”, email: “email@mun.ca”, username: “JD”, password: “password”, student_id: “1”, gender: “Male”, campus: “stjohns”	A user object in the database with all the respective values for each corresponding variable.	Input restrictions apply.
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Register user with all necessary values	
Step 4	Login as Username: JD1, Password: password	
Step 5	Click “My Profile”	
Step 6	Refresh MongoDB Compass	
Step 7	Verify values are correct in database	
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
A user object in the database with all the respective values for each corresponding variable, with the additional ‘__v = 0’ field that mongodb generates		Pass

User		
Test Case Number	Test Case 3	
Test Case Name	Log Out	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are logging in as John Doe and then logging out.	
Specifications		
Input	Expected Output	Notes
N/A	After logging out, a confirmation message is displayed and the log in screen is displayed	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click “Logout” from the navigation menu	
Step 5	Verify that the login screen is now displayed and no options can be selected	
Step 6		
Step 7		
Step 8		
Step 9		
Step 10		
Step 11		
Actual Output		Pass/Fail
After clicking “Logout” the login screen is displayed and with a confirmation message displayed. The navigation menu now has a different set of options that are available while logged out.		Pass

User		
Test Case Number	Test Case 4	
Test Case Name	View Another Profile	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are accessing the profile of John Doe while logged out, and then again while logged in as Jane Doe.	
Specifications		
Input	Expected Output	Notes
Load webpage localhost:3000/profile/{id of John Doe}	The profile page for John Doe, reading “John Doe’s Profile” and “John Doe’s Friends” with “No friends to show.”	When logged in as Jane Doe, “+Add Friend” button should also be present.
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000/profile/{id of John Doe}	
Step 3	Click “Login”	
Step 4	Login as Username: JD2, Password: password	
Step 5	Load webpage localhost:3000/profile/{id of John Doe}	
Step 6	Verify changes to page in both circumstances (logged in as different user, logged out	
Step 7		
Step 8		
Step 9		
Actual Output		Pass/Fail
The profile page for John Doe, reading “John Doe’s Profile” and “John Doe’s Friends” with “No friends to show.”		Pass

User		
Test Case Number	Test Case 5	
Test Case Name	Request Friend/Add Friend BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are logging in as Jane Doe and attempting to add John Doe as a friend. We are verifying the other user’s name is displayed on the users profile.	
Specifications		
Input	Expected Output	Notes
Click “+Add Friend”, Logout, Login as John Doe, click “My Profile”, accept friend request	After requesting, both users should now have an attribute “friends”, with the name of the other user.	After requesting, status should be “requested” for John Doe in Jane Doe’s friends and “pending” for vice versa. After accepting, it should be “accepted” for both.
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD2, Password: password	
Step 4	Load webpage localhost:3000/profile/{id of John Doe}	
Step 5	Click “+Add Friend”	
Step 6	Click “My Profile”	
Step 7	Logout	
Step 8	Login as Username: JD1, Password: password	
Step 9	Click “My Profile”	
Step 10	Click “+Add Friend” under “My Friend Requests”	
Step 11	Verify the other user’s name is displayed	
Actual Output		Pass/Fail
Both users now have an attribute “friends”, with the name of the other user.		Pass

User		
Test Case Number	Test Case 6	
Test Case Name	Request Friend/Add Friend WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are logging in as Jane Doe and attempting to add John Doe as a friend. Before accepting the friend request while logged in as John Doe, we are verifying the objects in the database. Finally, we will accept the friend request as John Doe and again verify the object in the database.	
Specifications		
Input	Expected Output	Notes
Click "+Add Friend", Logout, Login as John Doe, click "My Profile", accept friend request	After requesting, both users should now have an attribute "friends", with one object, whose id is identical to the id of the user they are meant to be friends with.	After requesting, status should be "requested" for John Doe in Jane Doe's friends and "pending" for vice versa. After accepting, it should be "accepted" for both.
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD2, Password: password	
Step 4	Load webpage localhost:3000/profile/{id of John Doe}	
Step 5	Click "+Add Friend"	
Step 6	Click "My Profile"	
Step 7	Logout	
Step 8	Login as Username: JD1, Password: password	
Step 9	Click "My Profile"	
Step 10	Click "+Add Friend" under "My Friend Requests"	
Step 11	Verify the other user's ID is added to the user's friends list in the database	
Actual Output		Pass/Fail
The user objective in the database now has the other user's ID in its friends array.		Pass

Post		
Test Case Number	Test Case 7	
Test Case Name	Create Post BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are creating a post object in the UI, while logged in as John Doe, and verifying its existence by viewing it on the dashboard.	
Specifications		
Input	Expected Output	Notes
text: "This is a post."	The new post is displayed on the user's dashboard.	All other inputs from Test Case 5 are automatic. userId will be John Doe's user id, the date will be the current date, etc.
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Create Post"	
Step 5	Enter text and click "Post"	
Step 6	Load the posts from the dashboard and confirm the text is correct	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
A post is displayed on the dashboard with the text "This is a post."		Pass

Post		
Test Case Number	Test Case 8	
Test Case Name	Create Post WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are creating a post object in the UI, while logged in as John Doe, and verifying its existence by viewing the database.	
Specifications		
Input	Expected Output	Notes
text: "This is a post2."	The new post is in the database with the correct variables.	All other inputs from Test Case 5 are automatic. userId will be John Doe's user id, the date will be the current date, etc.
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Create Post"	
Step 5	Enter text and click "Post"	
Step 6	View the post in the database.	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The new post is in the database with the correct variables.		Pass

Post		
Test Case Number	Test Case 9	
Test Case Name	Edit Post BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are editing the text of a post we already created as John Doe and verifying it in the list of posts.	
Specifications		
Input	Expected Output	Notes
text: "This is an edited post."	The post will be changed and the new text will be displayed.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Edit Post" on a post previously created	
Step 5	Enter text and click "Post"	
Step 6	Verify the text has changed on the dashboard	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The post is updated with the new text on the dashboard.		Pass

Post		
Test Case Number	Test Case 10	
Test Case Name	Edit Post WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are editing the text of a post we already created as John Doe and verifying it has changed in the database.	
Specifications		
Input	Expected Output	Notes
text: "This is an edited post."	The post will be changed and the new text will be displayed.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Edit Post" on a post previously created	
Step 5	Enter text and click "Post"	
Step 6	Verify the text has changed in the database	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The post is updated with the new text in the database.		Pass

Comment		
Test Case Number	Test Case 11	
Test Case Name	Create Comment BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are creating a comment while logged in as John Doe on a previously created post and verifying its existence by viewing it on the dashboard.	
Specifications		
Input	Expected Output	Notes
text: "This is another comment."	A comment is displayed under the post on the dashboard.	This comment should also be editable by John Doe after it has been posted. Again, unlike Test Case 8, all inputs other than text are automatic.
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Add a Comment"	
Step 5	Enter text and click "Post"	
Step 6	Verify the comment is displayed under the post on the dashboard	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The comment is displayed under the post on the dashboard.		Pass

Comment		
Test Case Number	Test Case 12	
Test Case Name	Create Comment WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are creating a comment while logged in as John Doe on a previously created post and verifying its existence in the database.	
Specifications		
Input	Expected Output	Notes
text: "This is another comment2."	A comment is added to the database with the postID of the associated post.	This comment should also be editable by John Doe after it has been posted. Again, unlike Test Case 8, all inputs other than text are automatic.
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Add a Comment"	
Step 5	Enter text and click "Post"	
Step 6	Verify the comment is in the database with the correct values	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The comment is in the database with the correct values.		Pass

Comment		
Test Case Number	Test Case 13	
Test Case Name	Edit Comment BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are editing the text of a comment we already created as John Doe and verifying it by viewing it on the dashboard.	
Specifications		
Input	Expected Output	Notes
text: "This is an edited comment"	The comment will be changed and displayed with the new text on the dashboard.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Edit Comment" on comment from Test Case 9	
Step 5	Enter text and click "Post"	
Step 6	Verify the comment has been edited on the dashboard	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The comment has been edited on the dashboard.		Pass

Comment		
Test Case Number	Test Case 14	
Test Case Name	Edit Comment WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are editing the text of a comment we already created as John Doe and verifying it by checking the database.	
Specifications		
Input	Expected Output	Notes
text: "This is an edited comment"	The comment will be changed in the database to have the new text.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Edit Comment" on comment from Test Case 9	
Step 5	Enter text and click "Post"	
Step 6	Verify the comment has been edited in the database	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The comment has been edited in the database.		Pass

Comment		
Test Case Number	Test Case 15	
Test Case Name	Delete Comment BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are deleting a previously created comment while logged in as John Doe and verifying it has been removed by viewing the dashboard.	
Specifications		
Input	Expected Output	Notes
N/A	The comment is removed from the associated post and is no longer displayed on the dashboard.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click “Delete Comment” on the comment to be removed	
Step 5	Verify the comment is no longer displayed on the dashboard	
Step 6		
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The comment is no longer displayed on the dashboard.		Pass.

Comment		
Test Case Number	Test Case 16	
Test Case Name	Delete Comment WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are deleting a previously created comment while logged in as John Doe and verifying it has been removed by checking the database.	
Specifications		
Input	Expected Output	Notes
N/A	The comment is removed from the database.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click “Delete Comment” on the comment to be removed	
Step 5	Verify the comment is no longer in the database	
Step 6		
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The comment is no longer displayed in the database.		Pass.

Schedule		
Test Case Number	Test Case 17	
Test Case Name	Create Course BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are creating a new course while logged in as John Doe and verifying it by viewing the groups page.	
Specifications		
Input	Expected Output	Notes
Course name: "Math 1000", Monday box checked, Monday time: "2-3pm", Course slot: "2"	A new course is displayed on the user's schedule page with all the respective values for each corresponding variable.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Schedule" from the navigation menu	
Step 5	Enter the variables and click "Submit"	
Step 6	Load course and verify the new course is being displayed	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The schedule is created and added to the user's schedule page.		Pass.

Schedule		
Test Case Number	Test Case 18	
Test Case Name	Create Course WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are creating a new course while logged in as John Doe and verifying it by checking the database.	
Specifications		
Input	Expected Output	Notes
Course name: “Math 1000”, Monday box checked, Monday time: “2-3pm”, Course slot: “2”	A schedule object is created in the database with all the respective values for each corresponding variable.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click “Schedule” from the navigation menu	
Step 5	Enter the variables and click “Submit”	
Step 6	Verify values are correct in database	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The schedule is created and added to the database.		Pass.

Schedule		
Test Case Number	Test Case 19	
Test Case Name	Delete Course BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are deleting a previously created course while logged in as John Doe and verifying it by viewing the schedule page.	
Specifications		
Input	Expected Output	Notes
N/A	The course is removed from the user's schedule page.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Schedule" from the navigation menu	
Step 5	Click "Delete" on the course to delete.	
Step 6	Load courses and verify the course has been removed	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The course is no long displayed on the user's schedule page.		Pass.

Schedule		
Test Case Number	Test Case 20	
Test Case Name	Delete Course WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are deleting a previously created course while logged in as John Doe and verifying it by checking the database.	
Specifications		
Input	Expected Output	Notes
N/A	The schedule object is deleted from the database.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click “Schedule” from the navigation menu	
Step 5	Click “Delete” on the course to delete.	
Step 6	Verify the schedule object has been removed from the database.	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The schedule is deleted and removed from the database.		Pass.

Poll		
Test Case Number	Test Case 21	
Test Case Name	Create Poll BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are creating a new poll while logged in as John Doe and verifying it by viewing the poll page.	
Specifications		
Input	Expected Output	Notes
Course Name: “example”	A new poll is displayed on the polls page.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click “Poll” from the navigation menu	
Step 5	Enter the course name and click “Submit“	
Step 6	Verify the poll is being displayed on the poll page with the enter named	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
A is displayed on the polls page.		Pass.

Poll		
Test Case Number	Test Case 22	
Test Case Name	Create Poll WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are creating a new poll while logged in as John Doe and verifying it by checking the database.	
Specifications		
Input	Expected Output	Notes
Course Name: “example”	A poll object is added to the database.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click “Poll” from the navigation menu	
Step 5	Enter the course name and click “Submit“	
Step 6	Verify the poll object has been added to the database with the entered name	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
A poll object with the enter course name is created and added to the database with the correct name.		Pass.

Poll		
Test Case Number	Test Case 23	
Test Case Name	Vote on a Poll BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are voting on a previously created poll while logged in as John Doe and verifying it by viewing the poll page.	
Specifications		
Input	Expected Output	Notes
"3 Stars" radio button is selected.	The new vote is displayed on the associated poll on the polls page.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Poll" from the navigation menu	
Step 5	Select the "3 Stars" button and click "Vote"	
Step 6	Load the polls on the polls page and verify the new vote is displayed	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The new vote is displayed on the associated poll on the polls page.		Pass.

Poll		
Test Case Number	Test Case 24	
Test Case Name	Vote on a Poll WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are voting on a previously created poll while logged in as John Doe and verifying it by checking the database.	
Specifications		
Input	Expected Output	Notes
“3 Stars” radio button is selected.	A vote is added to the array of the poll object in the database.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click “Poll” from the navigation menu	
Step 5	Select the “3 Stars” button and click “Vote“	
Step 6	Verify the poll object has been added to the database	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The poll object’s vote array is updated in the database to include the new vote.		Pass.

Group		
Test Case Number	Test Case 26	
Test Case Name	Create a Group BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are creating a new group while logged in as John Doe and verifying it by viewing the groups page.	
Specifications		
Input	Expected Output	Notes
Group name: "Test Group", Group Description: "This is some text."	A group is created and displayed on the groups page with the correct values.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Groups" from the navigation menu	
Step 5	Click "Create a Group"	
Step 6	Enter the input and click "Submit"	
Step 7	Verify the group is displayed on the groups page	
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
A group object is created in the database with the correct variables.		Pass.

Group		
Test Case Number	Test Case 26	
Test Case Name	Create a Group WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are creating a new group while logged in as John Doe and verifying it by checking the database.	
Specifications		
Input	Expected Output	Notes
Group name: "Test Group2", Group Description: "This is some text."	A group object is created in the database with the correct values.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Groups" from the navigation menu	
Step 5	Click "Create a Group"	
Step 6	Enter the input and click "Submit"	
Step 7	Verify the group object has been added to the database	
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
A group object is created in the database with the correct variables.		Pass.

Group		
Test Case Number	Test Case 27	
Test Case Name	Join a Group BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are joining a previously created group while logged in as John Doe and verifying it by viewing the groups page.	
Specifications		
Input	Expected Output	Notes
N/A	The group we joined no long says “You are not part of this group” on the groups page.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click “Groups” from the navigation menu	
Step 5	Click “Join a Group” on the group “Test Group”	
Step 6	Verify the group no long says “You are not part of this group” on groups page	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The group joined no longer displays “You are not part of this group”.		Pass.

Group		
Test Case Number	Test Case 28	
Test Case Name	Join a Group WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are joining a previously created poll while logged in as John Doe and verifying it by checking the database.	
Specifications		
Input	Expected Output	Notes
N/A	The user's ID is added to the members array of "Test Group2".	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click "Groups" from the navigation menu	
Step 5	Click "Join a Group" on the group "Test Group2"	
Step 6	Verify the user ID was added to the groups members array in the database	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The user's ID is now in the groups members array in the database		Pass.

Group		
Test Case Number	Test Case 29	
Test Case Name	Delete a Group BB	
Test Type (White Box or Black Box)	Black Box	
Test Description	We are deleting a previously joined group while logged in as John Doe and verifying it by viewing the groups page.	
Specifications		
Input	Expected Output	Notes
N/A	The group is no longer displayed on the groups page.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click “Groups” from the navigation menu	
Step 5	Click “Delete” on the group “Test Group”	
Step 6	Verify the group has been removed from the groups page	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The group is no longer displayed on the groups page.		Pass.

Group		
Test Case Number	Test Case 30	
Test Case Name	Delete a Group WB	
Test Type (White Box or Black Box)	White Box	
Test Description	We are deleting a previously joined group while logged in as John Doe and verifying it by checking the database.	
Specifications		
Input	Expected Output	Notes
N/A	The group is no longer in the database.	
Procedural Steps		
Step 1	Console node app.js	
Step 2	Load webpage localhost:3000	
Step 3	Login as Username: JD1, Password: password	
Step 4	Click “Groups” from the navigation menu	
Step 5	Click “Delete” on the group “Test Group2”	
Step 6	Verify the group has been removed from the database	
Step 7		
Step 8		
Step 9		
Step 10		
Actual Output		Pass/Fail
The group is no longer in the database.		Pass.