Design Inspection

Product: Frindr
Date: 9/19/2019
Author: Development Team

Defect #	Description	Severity	How Corrected
1	Login only worked with username	Workaround	Added code to allow login via username or email.
2	Reset password link was easily able to be brute forced via URL	Workaround	Make the URL completely unique and long so it can't be brute forced.
3	No already registered link on the register page	Workaround	Added link for users to go back to the login page if they're already registered
4	Register/forgot password / login page still accessible even if the user logged in	Important	Added middleware for each of these pages to ensure the user is not authenticated (no session exists). If the session exists, they'll be redirected to the dashboard
5	Expecting location API to only return back latitude and longitude but instead returns back JSON with multiple extra fields	Workaround	Parse through JSON and extract latitude and longitude fields
6	Location is not as accurate as expected, it is within the mile	Workaround	When designing the matching algorithm we will take into account the variability of accuracy of location.
7	Location is requested from user on profile page, but needs to be from either login page or registration page	Important	Change location of where API is called from profile page to both registration and login page

Code Inspection

Product: Frindr

Date: 9/19/2019

Author: Development Team

Defect #	Description	Severity	How Corrected
1	Incorrect values for a user were being placed in the wrong columns for the users table	Important	Update the SQL query to properly insert the values from the registration form into the users table so they match
2	Text content were not being escaped when passed into the SQL queries (login/register/forgot-pass)	Important	Called the escape function from mysql library to escape all variables passed into queries. Prevents SQLi.
3	SQL query connections were not being killed	Important	Everytime a query would be end, the connection to the DB is killed. Prevents overflowing/hanging connections flooding the DB.
4	After resetting password, user was not being redirected to the login page	Workaround	User is redirected to the login page where they will then be given a notification that their password has been reset and may login.
5	Error messages were displaying as success messages	Workaround	Updated the express flash to flash for error instead of success all the time.
6	After user logs in for second time database does not update current location properly since old location field is already populated in db	Currently Working	First check if db location fields are populated, if so then update db field for location (latitude, longitude)

7	Expected latitude and longitude values from API call to be of type int but were decimal values	Workaround	Changed database to hold float values instead of int for latitude and longitude
8	Location API returns fields including accuracy, speed, and altitude and currently these are stored in database but not used	Workaround	Remove fields from database, and no longer save these fields
9	Environment variable setup was not operating as expected	Currently working	Environment variables were removed for the time being until a suitable workaround is found, data is instead currently hardcoded
10	Awaiting the asynchronous database requests were throwing errors	Workaround	Raw promises were instead implemented to avoid the other implementation.
11	Currently there is no authentication for the profile rest requests.	Important	Authentication will have to be added, and logic behind the logged in state required will have to be deciphered.
12	Userend file bloated	Currently Working	Profile has already been spread out, but most functions take place in this file, and it needs some serious work done to be navigable

Unit Testing

Module: Login

Product: Frindr

Date: 9/19/19

Author: Mayur

Defect #	Description	Input	Output	Severity	How Corrected
1	Empty username textbox when logging in	Empty username	Empty username is passed in post request to the servers	Important	Use express validation to confirm the username is not empty
2	Empty password textbox when logging in	Empty password field	Empty password is passed in the post request to the servers	Important	Use express validation to confirm the password is not empty
3	Length of username/password exceeds 255 characters	The textbox for username/password on the login page	What happens is the field isn't limited. Expected output is to stop typing even if the user types (hits limit of 255)	Workaround	Add limits to the input tags for both fields. This limits the characters to a minimum/maximum

4	Failure message after logging in.	Failure message after incorrect login (wrong password / non-existent user)	Actual output: Incorrect password. / User does not exist Expected output: User or Password is incorrect	Important	Instead of stating whether a user exists or whether the password is incorrect, state either user or password is incorrect.

Module: Register

Product: Frindr

Date: 9/19/2019

Author: Mayur

Defect #	Description	Input	Output	Severity	How Corrected
1	Empty text-fields on the register page	All text-fields on the register page	Actual output: Success Expected output: Failure - ensure all fields are filled etc	Important	Use express validation on the route that serves the register page to check each field of the body. Confirm they’re filled and not empty etc and if empty, return error messages accordingly.
2	Mismatch between the password and confirmation password fields	Password and confirm password text fields	Actual output: Success registration. Expected output: Confirmation password does	Important	Double check if confirmation password equals password fields and if so, continue. Else return error message to the user

			not match		
3	Length of username/password exceeds 255 characters	The textbox for username/password on the register page	What happens is the field isn’t limited. Expected output is to stop populating the box even if the user types (hits limit of 255)	Important	Add limits to the input tags for both fields. This limits the characters to a minimum/maximum

Module: Forgot Password

Product: Frindr

Date: 9/19/2019

Author: Mayur

Defect #	Description	Input	Output	Severity	How Corrected
1	If the user does not exist, do not state the user doesn’t exist.	Form submission for resetting forgotten password	Actual output is “This user does not exist” “Password has been reset” Expected should be “If this email exists in our	Workaround	Display one message for all possible results/inputs to be more secure

			system, you will receive a link to reset your password”		
2	Password link URL cannot be guessed.	Password link sent via email	Actual Link ends with /1 /2 /3 etc. Expected Shoud be random so it can’t be bruteforced.	Important	Made the link have a completely random number of digits/characters so it cannot be guessed by anyone trying to bruteforce etc.

3	Ensure password is changed	New password should work and old shouldn’t work	Actual: Password wasn’t being set Expected: New password should work and be the only working password	Important	Added query to update the user’s password and then removed the forgot password toggle preventing the link from being accessed.

Module: Profile

Product: Frindr

Date: 9/19/2019

Author: Joe & CJ

Defect #	Description	Input	Output	Severity	How Corrected
1	Adding multiple collaborators simultaneously, took a lot of time to verify that collaborators were users of the app and the app seemed to stop working	Click on the ‘+’ icon next to collaborators while viewing idea details. Add multiple emails in the dialog	Actual: After confirming, app stopped responding for few seconds due to checks in the backend Expected: Emails corresponding to existing users are verified and users are	Important	Due to time consuming database checks, modified adding collaborators to only add one collaborator at a time

		box and confirm	added as collaborators to the idea (UI and database)		
2	Test to show that entering no info when creating a profile will fail	Just a request to the post profile api	400 BAD REQUEST and the output info from the database	Important	Handling error messages to avoid crashing
3	Test for only email sent when creating profile	A body with only the email is sent on the post profile request	201 Created is returned and the row is added to the database, temp values are added for all other columns	Important	Default values were set for all the columns outside email
4.	Test for all profile fields being used	A body with email, bio, name, interests, and characteristics is sent	201 Created and the row contains all values	Important	Values are sent as part of the sql query instead of being defaulted if seen
5	Test for creating profile with same email	Body with same email as another row is sent	400 bad request and message	Important	Send request to check for the email first and only post if it comes up empty

6	Test for getting profile without email	Body without email is sent	404 not found	Important	When not found, send back special error message and read that
7	Test for getting profile without email	Send email with no profile	404 not found	Important	When not found, send back special error message and read that
8	Test for getting profile with email	Send email in body	200 OK and all information with that profile	Important	When found, a message is sent containing all of the rows with that email, should be 1
9	All updates tested without email (saving time)	Send body with update but no email	400 BAD REQUEST and error message with requirements	Important	Throw error whenever email is not found in body, gives message
10	Name updated without name included	Send body with update but no name, also send GET on profile to ensure not sent	400 BAD REQUEST and error message with requirements	Important	Throw error whenever name is not found in body, gives message
11	Update name with requirement s	Send body with name and email, also check also send GET on	200 OK and output from database	Important	Just ensure that it is changed by sending a 2nd test to show that the updated name matches

		profile to ensure sent			
12	Bio updated without bio included	Send body with update but no bio, also send GET on profile to ensure not sent	400 BAD REQUEST and error message with requirements	Important	Throw error whenever bio is not found in body, gives message
13	Update bio with requirements	Send body with bio and email, also check also send GET on profile to ensure sent	200 OK and output from database	Important	Just ensure that it is changed by sending a 2nd test to show that the updated bio matches
14	Interests updated without interests included	Send body with update but no interests, also send GET on profile to ensure not sent	400 BAD REQUEST and error message with requirements	Important	Throw error whenever interests are not found in body, gives message
15	Update interests with requirements	Send body with interests and email, also check also send	200 OK and output from database	Important	Just ensure that it is changed by sending a 2nd test to show that the updated interests match

		GET on profile to ensure sent			
17	Characteristics updated without characteristics included	Send body with update but no characteristics , also send GET on profile to ensure not sent	400 BAD REQUEST and error message with requirements	Important	Throw error whenever characteristics are not found in body, gives message
18	Update characteristics with requirements	Send body with characteristics and email, also check also send GET on profile to ensure sent	200 OK and output from database	Important	Just ensure that it is changed by sending a 2nd test to show that the updated characteristics match
19	Delete profile from database, no email	Body does not contain email	400 bad request and output from database	important	Ensure that email is being sent, or throw error
20	Delete profile from database, include email	Send body with email for deletion	200 OK	Important	Send request to delete, ensure on database that it is now deleted

21	Delete non-existent profile	Send body with email when that row dosen’t exist	400 BAD REQUEST	Important	Checks to see how many rows in the database were affected by the deletion, if it is 0, that means there were no email matches, so error is thrown

Module: Location

Product: Frindr

Date: 9/19/2019

Author: Aeson & Haris

Defect #	Input	Output	Description	Severity	How Corrected
1	Button that will trigger “get location” function is pressed	Actual: Error due to permission to get location did not work Expected: location information was added to the database	Once get location has been triggered, location should be added to the database	Critical	User is redirected back to the login page and an error is shown to enable location services in the browser

2	Return location information that has an error set, that will prevent location from being set in the database	Actual: Did not detect the error, and unwanted values set in the database Expected: website would redirect to log in page, put the server throws an error	Send bad information back to the website and make sure it is handled gracefully	critical	User is redirected to log in page
3	Sample geolocation API input (JSON object)	Actual: Values (latitude, longitude)returned did not have decimal precision Expected: Values passed in are equal to expected values from API input	Handling of API input does not support decimal values.	Critical	Change variable type that handles input values to float