Sprint 1: Manual Testing Documentation

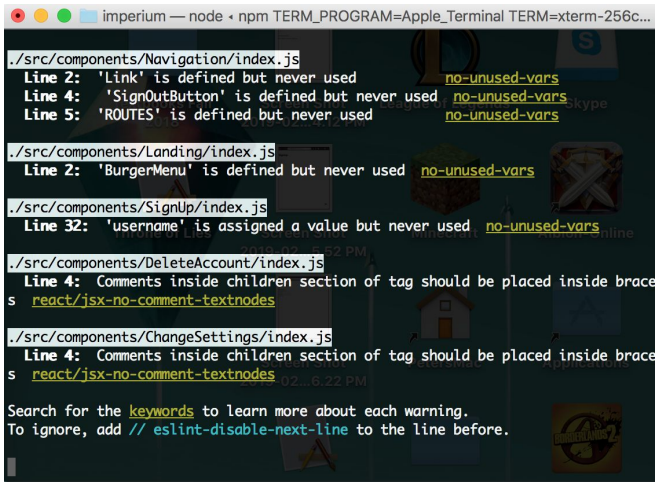
This document will describe our manual testing process, which was done according to our first sprint planning document. Each user story will include a procedure to test, inputs used and both a description and reasoning for edge cases used. If the expected and actual outputs differ, there will be a brief description about how the bug was fixed.

Testing Set #1: Create an account for Imperium.

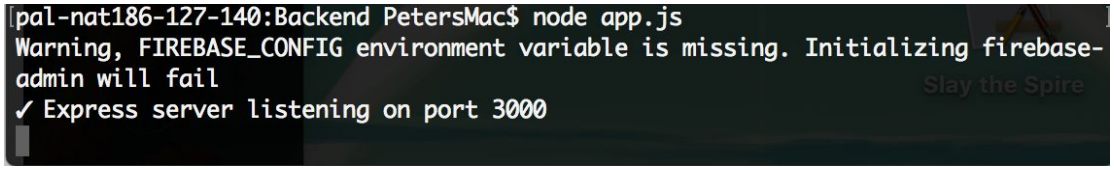
Testing Procedure:

- Compile and run Frontend
- Compile and run Backend
- Navigate to Imperium landing page
- Click hamburger button
- Click sign in button
- Click sign up button
- Input data into fields shown
- Click sign up button
- Check Firestore in Firebase, see if data from fields is saved

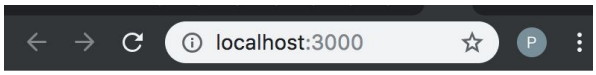
1. Compile and run frontend: Success. No errors, but a couple of warnings. Picture below:



2. Compile and run backend: Success. Picture below:

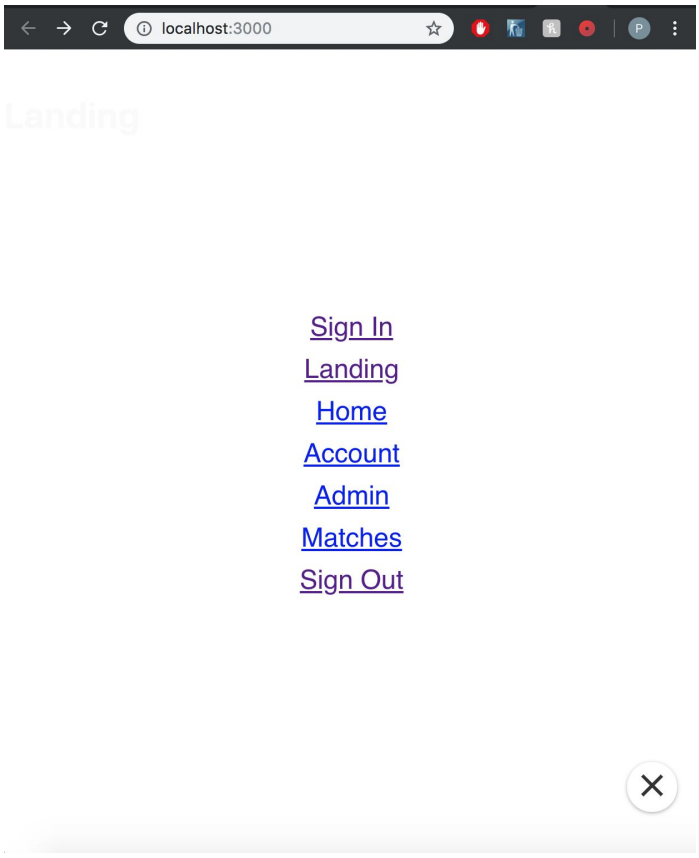


3. Navigate to Imperium landing page: Success. Running npm start in the Frontend directory successfully loads the app and puts you on the landing page. Below is a picture of the landing page:

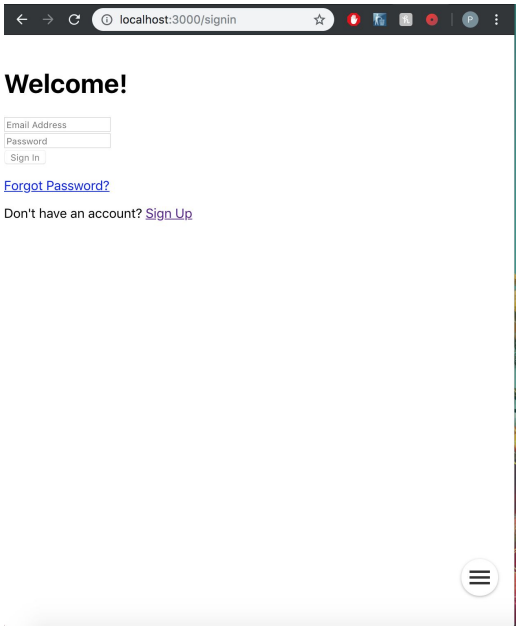


Landing

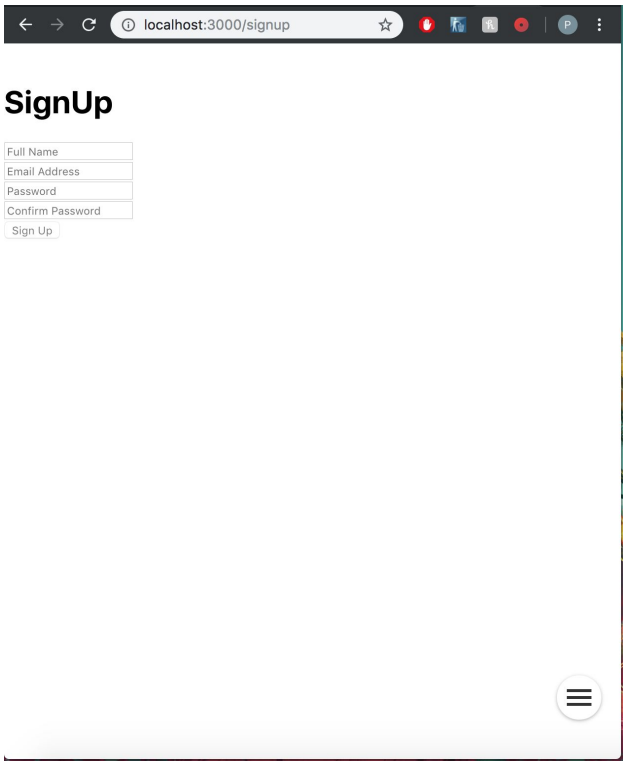
4. Click hamburger button: Success. Clicking button opens up a menu overlay with the proper links. Picture below:



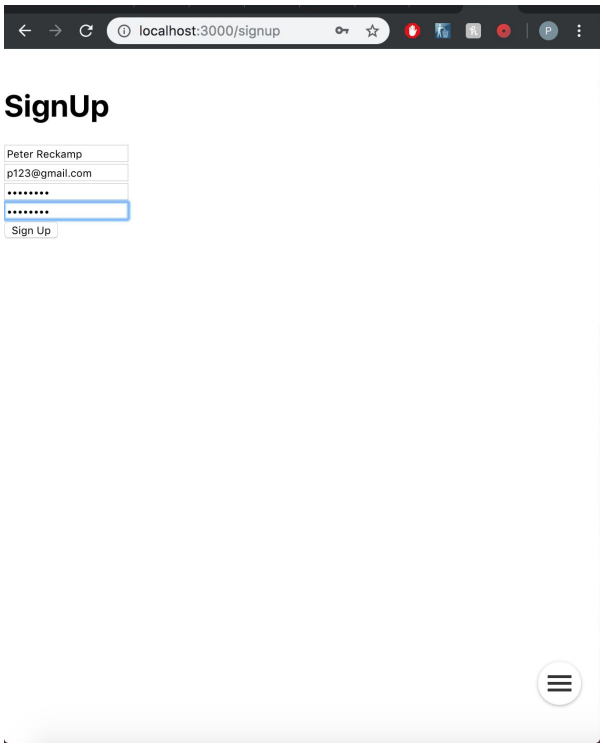
5. Click sign in button: Success. Clicking sign up will direct you to the correct page. Picture below:



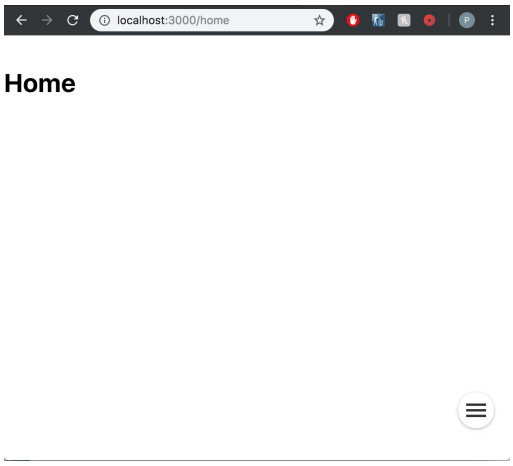
6. Click sign up button: Success. Clicking sign up will direct you to the correct page. Picture below:



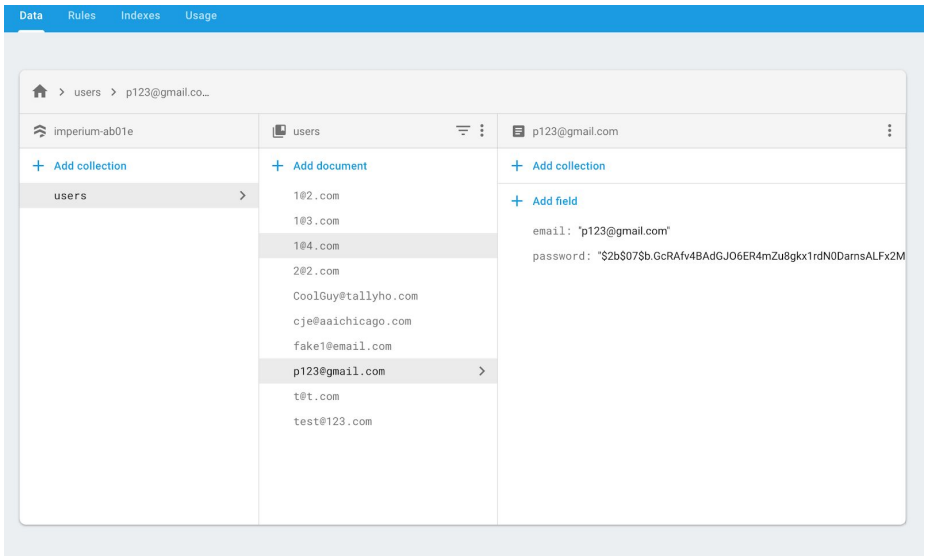
7. Input data into fields shown: Success. All four fields allow for text entry, display which text should be entered where, and the password and confirm password fields censor the password. Picture below:

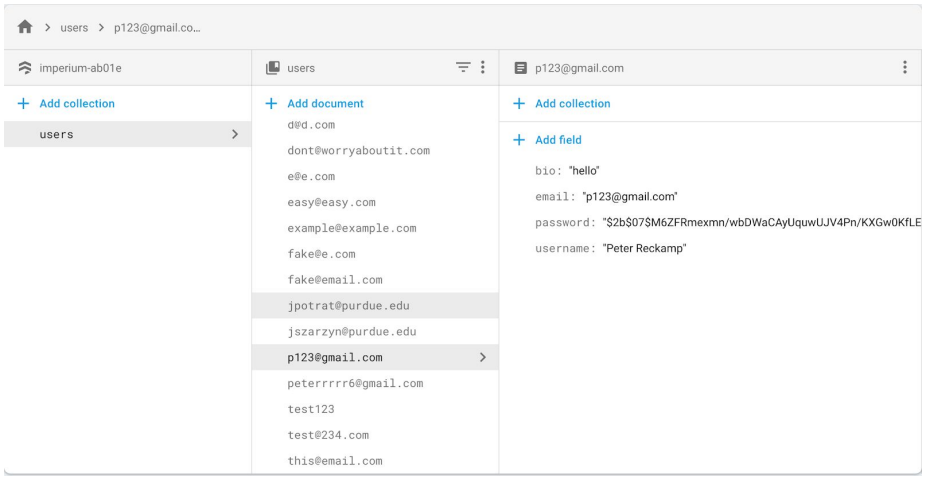


8. Click sign up button: Success. Redirects to home page after button is clicked. Picture below:



9. Check Firestore in Firebase, see if data inputted is stored: Partial success. The email and password are stored correctly (password is hashed), however the name of the person is not stored correctly. We added another collection in Firestore to correct this. The first picture below is the first time this was tested, and the second picture is the second time this was tested with the name now correctly store in Firebase:





Extra test cases:

1. Name: Peter Reckamp
   Email: 1234
   Password: hello123
   Confirm Password: hello123
   Why? See if the error handling is correct for an email that is too short (less than five characters)
   Correct output? (Y/N) Y
2. Name: Peter Reckamp
   Email: hello@gmail.com
   Password: hello123
   Confirm Password: 123

Why? Check if both passwords need to be the same for the account to be created
Correct output? (Y/N) Y

3. Name:
   Email:
   Password:
   Confirm Password:
   Why? See if error handling works for no data
   Correct output? (Y/N) Y
4. Name: Anton Popilov
   Email: hello@gmail.com
   Password: hello123
   Confirm Password: hello123
   (Repeat this sign up)
   Why? Make sure an email can only be used to create one account
   Correct output? (Y/N) Y
5. Name: sample1
   Email: sample1@gmail.com
   Password: sample1
   Confirm Password: sample1
   Why? Make sure there are no errors if someone makes multiple fields the same string
   Correct output? (Y/N) Y
6. Name: Peter Reckamp
   Email: 123456@gmail.com
   Password: hello123
   Confirm Password: hello123
   Why? See if two people can have the same name (should be allowed)
   Correct output? (Y/N) Y
7. Name: Peter Reckamp
   Email: abcdefghijklmnopqrstuvwxyz@antidisestablishmentarianism.superlong
   Password: hello123
   Confirm Password: hello123
   Why? See if very long emails work
   Correct output? (Y/N) Y

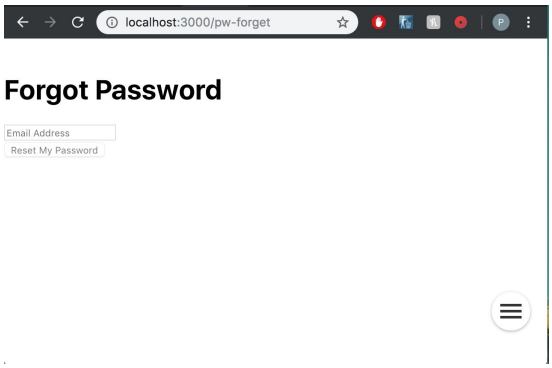
Testing Set #2: Reset password by having an email sent to the specified account email with a new random password.

Testing Procedure:

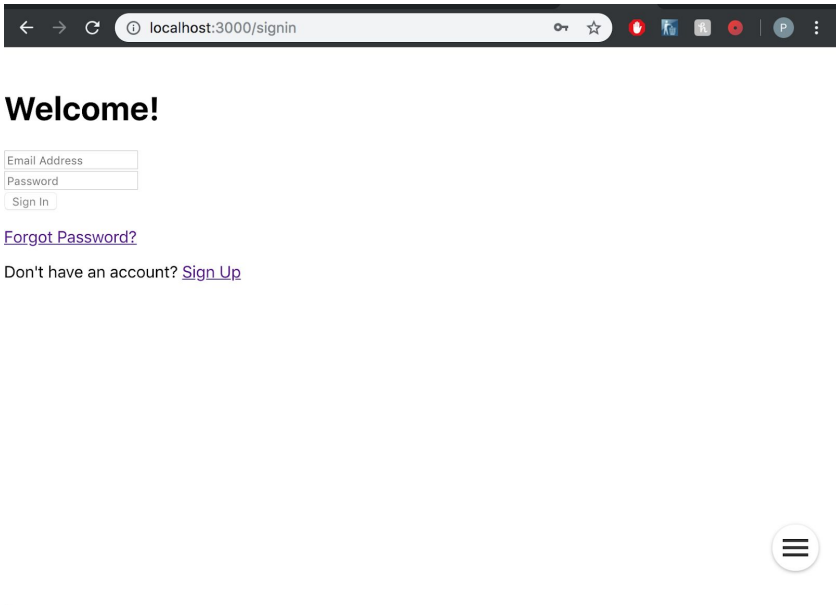
- Compile and run Frontend
- Compile and run Backend
- Navigate to Imperium landing page
- Click hamburger button
- Click sign in button
- Click “Forgot Password?”
- Input email address and click reset password button
- Check corresponding email address for email
- Login with updated password

1-5 tested in set 1.

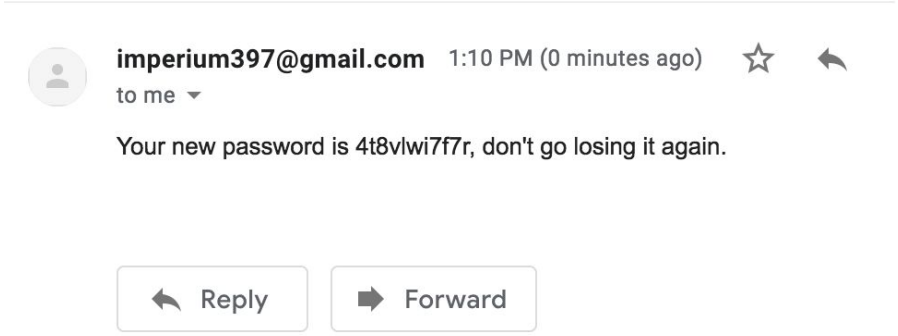
6. Click “Forgot Password?”: Success. Properly redirects to correct page. Picture below:



7. Input email address and click reset password button: Success. Properly takes text and redirects to sign in page. Picture below:



8. Check corresponding email address for email: Success. Email is sent to the proper account with a random password included. Picture below:



9. Log in with updated password: Success. Logging in using the same account and the new password properly logs the user in.

Extra Tests:

1. User tries to log into their account with old password: Success, login does not work with old password.
2. User resets password a second time: Success, new password works and old random password does not.
3. User does not input an email before clicking the forgot email button: Success, cannot click button.
4. User tries to reset email on email that does not exist in database: Success, unauthorized message is displayed.
5. User tries to reset email with an email length > 32: Success. Long emails work correctly.
6. User tries to reset password after logging in: Success. User is unable to navigate to forgot password page if logged in.
7. User resets their password more than 5 times: Success. No limit on how many password resets are allowed per user.
8. User inputs wrong password after resetting: Success. Unable to login.

Test Set #3: Login and manage Imperium account

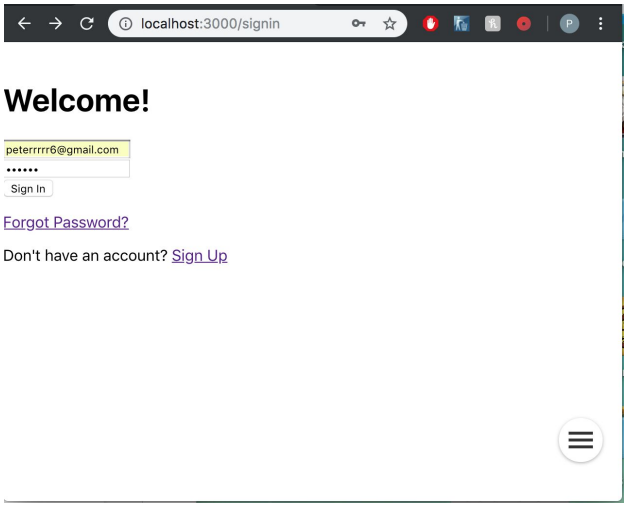
Testing Procedure:

- Compile and run Frontend

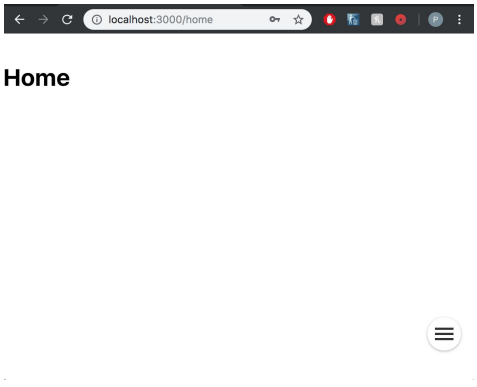
- Compile and run Backend
- Navigate to Imperium landing page
- Click hamburger button
- Click sign in button
- Input existing email address and password
- Click sign in button
- Click hamburger button and click account
- Fill in password fields
- Click save preferences
- Check Firestore for updated password

1-5 tested in set 1.

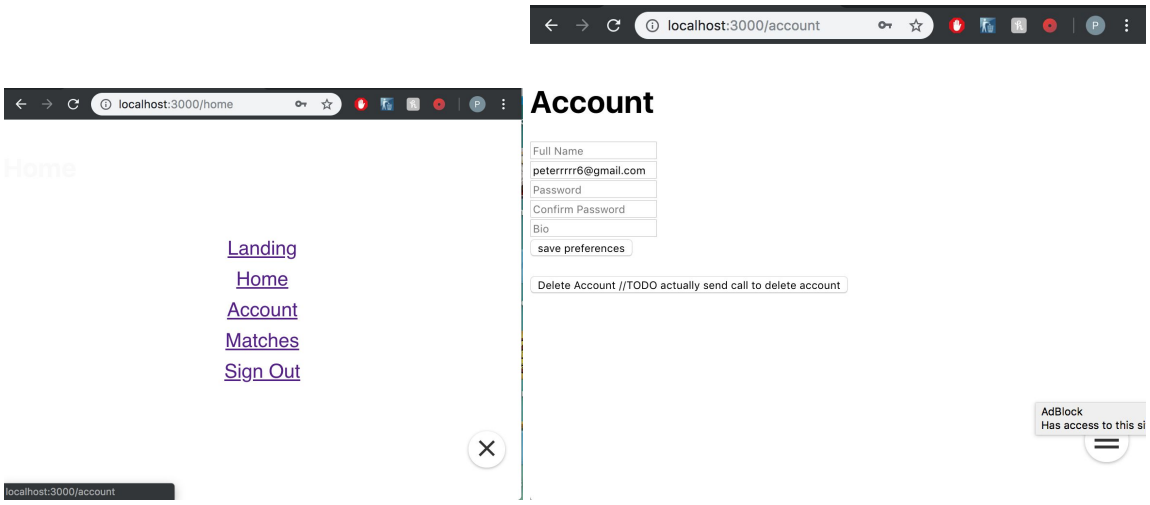
6. Input existing email and password: Success. Both fields take text and the password field is censored. Picture below:



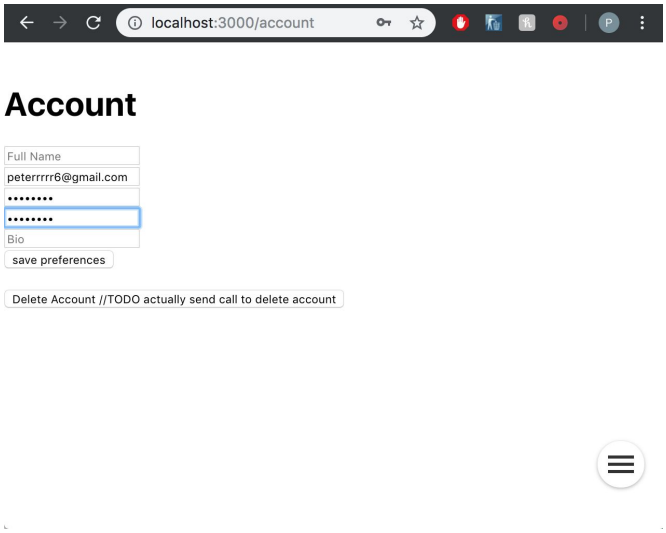
7. Click sign in button: Success. Redirects to home page if the account exists and the information is correct. Picture below:



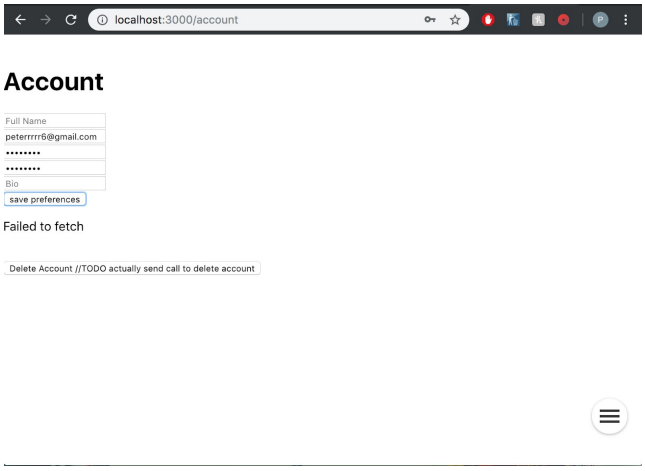
8. Click hamburger button and click account: Success. Navigates to the correct page with proper buttons shown. Pictures below:



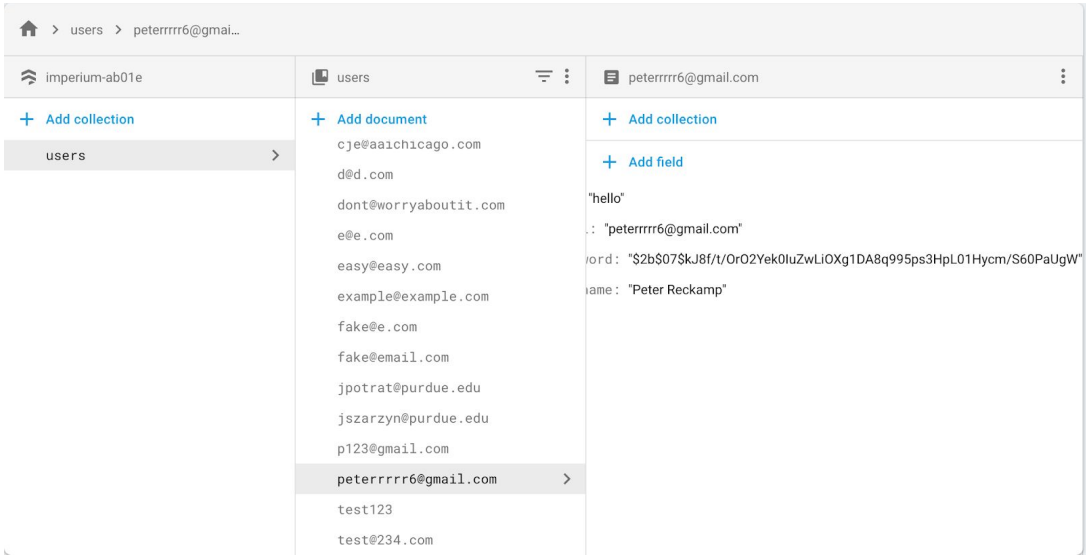
9. Fill in password fields: Success. Data fields properly take text and censor. Picture below:



10. Click save preferences: Failure. Failure to fetch error is given and page is not updated. Picture below:



11. Check Firestore for updated password: Success. Password is changed in the database after preferences are saved on the frontend. Picture below:



Extra tests:

1. Put different strings in the password and confirm password fields: Success. Outputs error message and does not update.
2. No text is put into the password fields: Success. Password is not updated.
3. Current password for the account is put into the password fields: Success. Password is not changed.
4. Reset password using the forgot password link, login with new password, and then update the password: Success. Password updated correctly.
5. Try to log into account with old password after it is changed in account settings: Success. Cannot login with old password.
6. Login with incorrect password: Success. Error code given and not logged in.
7. Login with incorrect email: Success. Error code given and not logged in.

Testing Set #4: Delete an account.

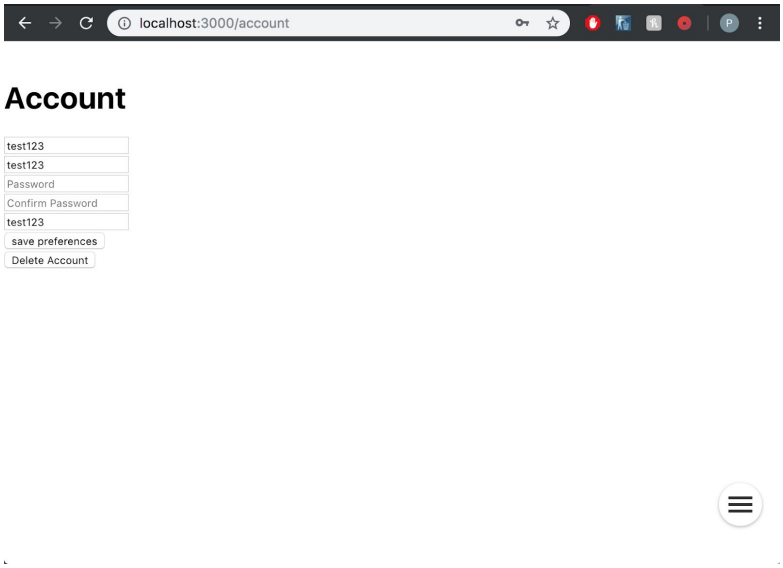
Testing Procedure:

- Compile and run Frontend
- Compile and run Backend
- Navigate to Imperium landing page

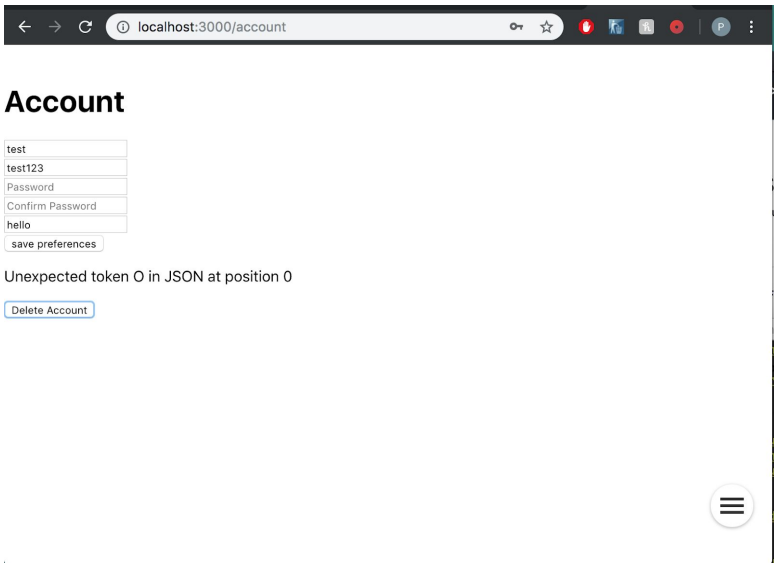
- Click hamburger button
- Click sign in button
- Navigate to hamburger bar and click account
- Input email address and click delete account
- Try to log in with deleted account
- Check Firestore and see if account still exists in database

1-5 tested in set 1.

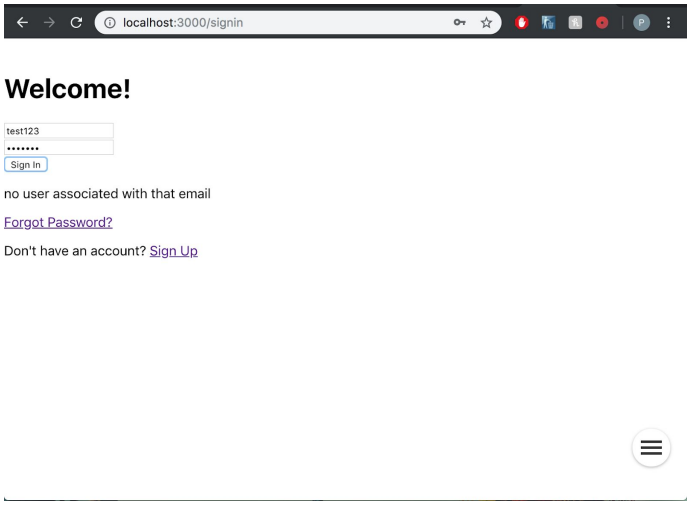
6. Navigate to hamburger bar and click account: Success. Delete account button appears and is formatted correctly. Picture below:



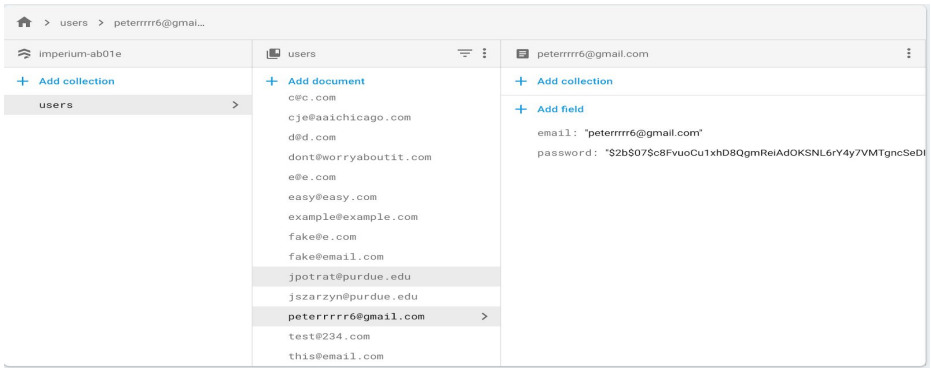
7. Input email address and click delete account: Partial Success. Does not redirect and gives an error, however the account is actually deleted upon sign up. Bug identified and fixed, now properly signs out user and drops them at home page. Picture of error below:



8. Try to log in with deleted account: Success. User is not allowed to login again after deleting their account. Picture below:



9. Check Firestore and see if account still exists in database: Success. Account information has been deleted and no longer appears anywhere in the database. Picture below:



Extra Tests:

1. Delete account and then make a new account with same email: Success. Allows for account to be created as long as it doesn't exist in the database.
2. Delete account that doesn't exist: Success. Unable to navigate to the page that contains delete account page unless already logged in.
3. Input different account to delete on account page: Success. Changing the email on the account page does not change which account is deleted.
4. Change password and then delete account after logging in: Success. Account deletes as it normally would.

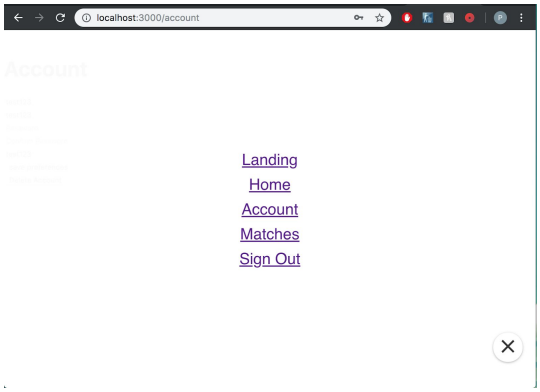
Testing Set #5: Logout of an account.

Testing Procedure:

- Compile and run Frontend
- Compile and run Backend
- Navigate to Imperium landing page
- Click hamburger button
- Click sign in button
- Navigate to hamburger bar
- Click sign out
- Navigate to hamburger bar

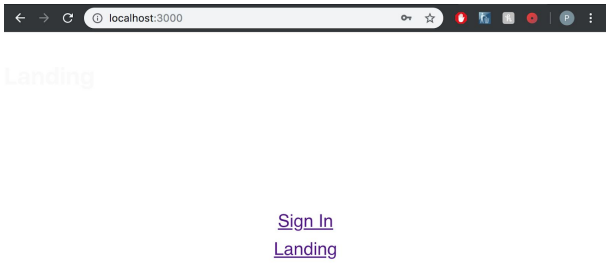
1-5 tested in set 1.

6. Navigate to hamburger bar: Success. Hamburger button is correctly updated and contains a sign out option. Picture below:



7. Click sign out: Success. Redirects to home page and signs the user out correctly. Picture omitted due to redundancy.

8. Navigate to hamburger bar: Success. Logged out user has their options in the hamburger bar updated, only allows access to landing page and sign in page. Picture below:



Extra Tests:

1. Sign in, log out, sign in with different user, log out: Success. Allows for users to log in and log out on same device correctly. Hamburger bar has correct options both times.
2. Log out twice: Success. Logging out multiple times does not work, as the redirect prevents this.
3. Log in, change password, log in, logout: Success. Changing passwords does not affect users ability to log out.
4. Delete account, create account with same email, logout: Success. Logging out is not dependent upon how recently the account was created or whether it has been deleted before.
5. Use forgot password, log in with new password, logout: Success. Forgot password function works correctly with log out.

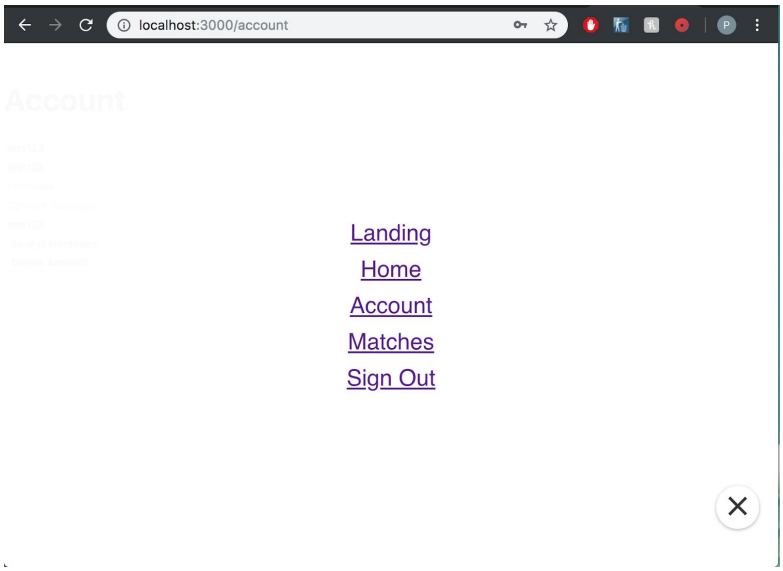
Testing Set #6: Successful implementation of the hamburger bar to navigate the website.

Testing Procedure:

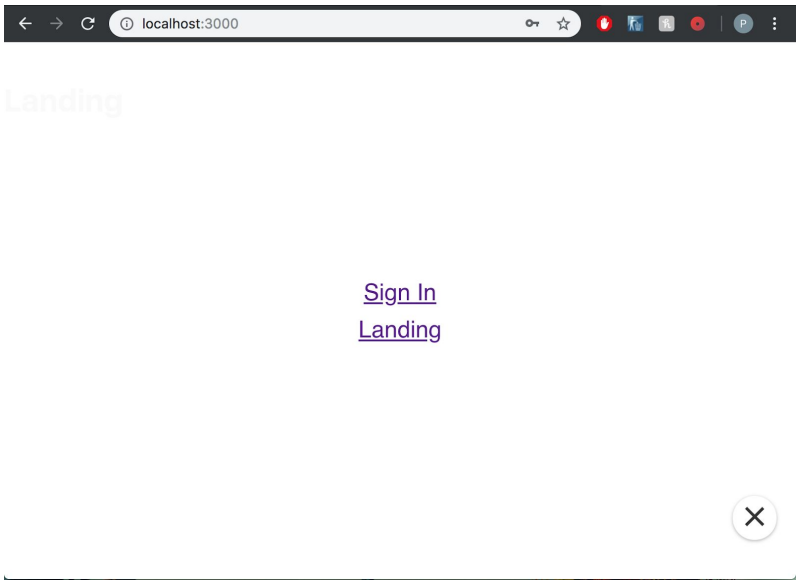
- Compile and run Frontend
- Compile and run Backend
- Navigate to Imperium landing page
- Click hamburger button
- Click Sign In button then hamburger button
- Click Sign Out button then hamburger button
- Click Sign In button then hamburger button
- Click Account then hamburger button
- Click Landing then hamburger button
- Click Landing then hamburger button
- Click Home then hamburger button
- Click Matches then hamburger button
- Click Sign Out then hamburger button
- Click Sign Up then hamburger button
- Click Sign Up -> Forgot Password? then hamburger button

1-4 have previously been tested

5. Selecting the hamburger bar after hitting the Sign In button still works successfully, displays proper pages. This is pictured below



6. Selecting the hamburger bar after signing out still works successfully, displays correct pages to a user that is not logged in. This is pictured below



7-14 are successful and not pictured below due to redundancy. The hamburger button can be successfully accessed from any screen.

Testing Set #7: Successfully retrieving a forgotten password.

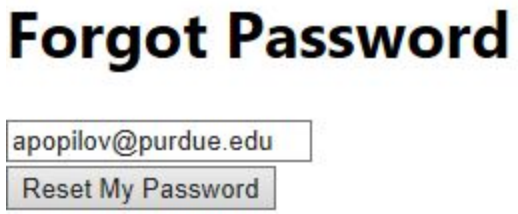
Testing Procedure:

- Compile and run Frontend
- Compile and run Backend
- Navigate to Imperium landing page
- Click hamburger button
- Click sign in button
- Click “Sign Up”
- Fill out necessary information
- Click “Sign Up”
- Sign Out using the hamburger button
- Navigate to sign in page and hit “Forgot Password?”
- Type in email and hit “Reset My Password” button
- Check email to see if updated password has been sent
- Login with the updated password

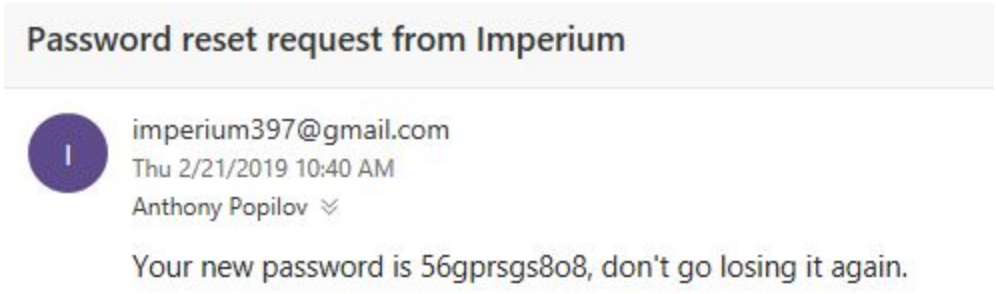
1-9 have previously been tested

10. The sign in page is pictured below where the “Forgot Password?” link is highlighted for reference. Hitting “Forgot Password?” successfully takes you to the following image pictured in #11.

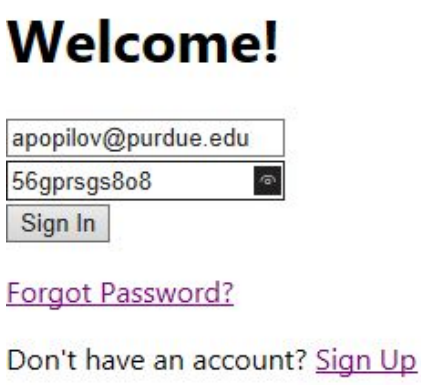
11. Once at the Forgot Password page, the user must type in their email and hit Reset My Password in order for Imperium to email them a new password which should be successful upon login.



12. Following this, the user must check their email to see their newly updated password sent from Imperium. This is successful and is pictured below.



13. Subsequently, the user should now be able to login with their newly reset password. This is successful and is seen by the picture below.



Extra Tests:

1. Try resetting the password of a user that does not exist. This should not do anything, and Imperium cannot send the password to an email that it is not storing, therefore this is successful.
2. Try resetting a password multiple times for the same email. This is successful; Imperium will continue sending you updated passwords if you keep requesting new passwords.

3. Try resetting a password without filling in an email address first. This is successfully tested against, as the user cannot hit the “Reset My Password” button without actually typing in the textbox first.
4. Try signing in with a previous password that has already been reset. The old password no longer works, and only the most recently updated password via email is successful. This is optimal.
5. Try resetting password from the Account menu. By saving and confirming a new password, the old password should no longer be acceptable. This is successful.

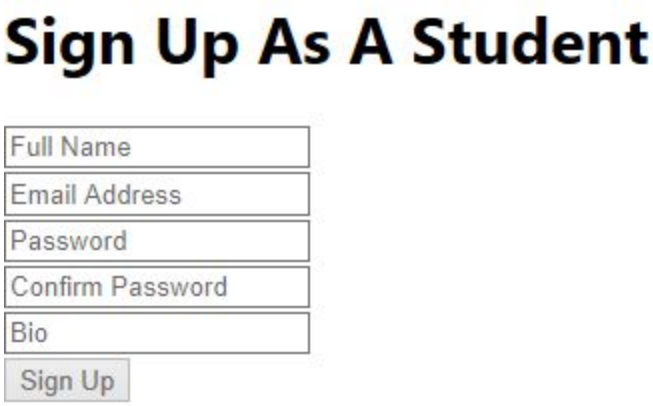
Testing Set #8: Creating and updating the bio.

Testing Procedure:

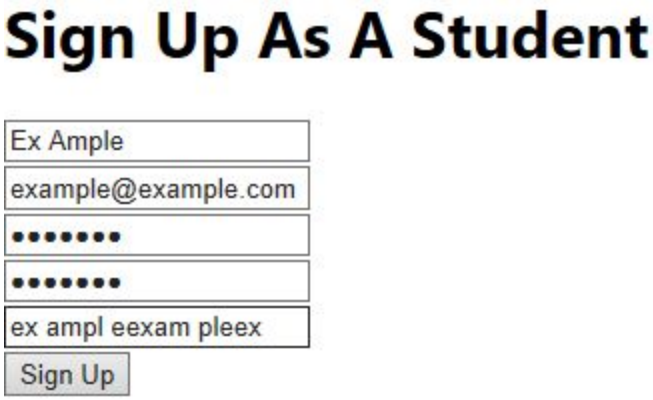
- Compile and run Frontend
- Compile and run Backend
- Navigate to Imperium landing page
- Click hamburger button
- Click sign in button
- Click “Sign Up”
- Fill out necessary information as well as the “bio” field
- Click “Sign Up”
- Check Firebase for properly stored info
- Navigate to Account page and fill out the updated Bio
- Check Firebase for proper storage

1-5 tested in set 1.

6. Click “Sign Up”: Success. Properly redirects to correct page where the bio input is properly displayed below the rest of the necessary information. Picture below:

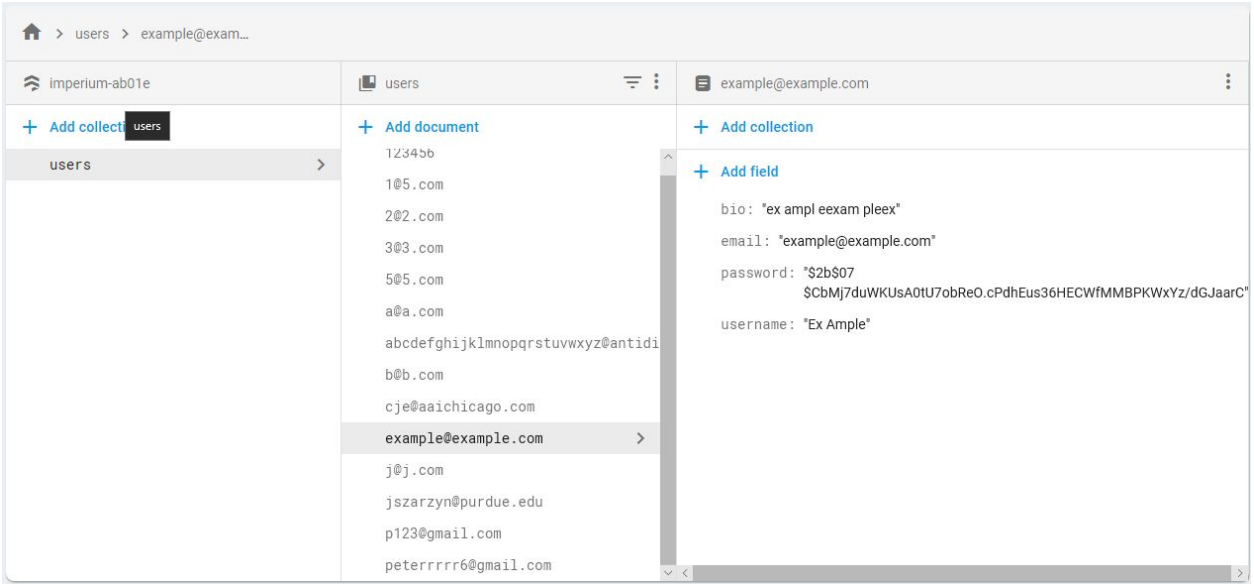


7. Fill out the necessary information in order to be able to sign up. The “Sign Up” button should only light up once the “Bio” has been completed Picture below

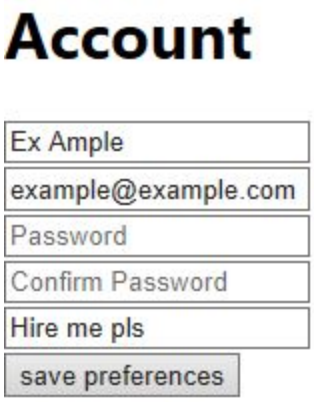


8. Hit the “Sign Up” button. This should redirect to the Home page. Picture below.

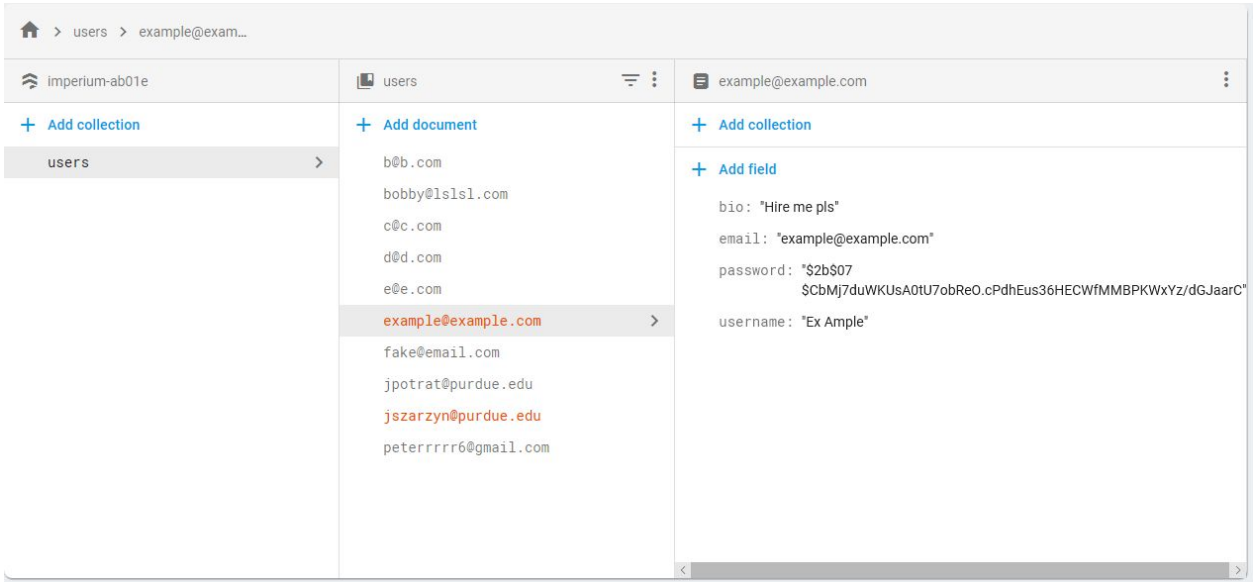
9. To ensure the “Bio” is functioning properly, we must check Firebase to see if the field has been updated. It should say “Ex ampl eexam pleex” for user example@example.com. This is successful and can be seen from the picture below.



10. Navigate to the “Account” page, fill out the updated Bio field, and click “save preferences” in order to update the Bio. Picture is below.



11. Check Firebase to ensure the Bio has been properly updated. Success. Pictured Below.



Extra Tests:

1. User tries to update the Bio multiple times. The Bio field in the database contains the most recent version. Success. The user can navigate to the account page in order to update various preferences.
2. User updates the bio with an empty string: Success. Bio remains unchanged.