

Unit Tests

Front-end Components:

Dashboard (Brian & Colin & Hang & Franz):

- Check to see that it displays user statistics fetched from the back-end.
- NG charts modules are connected to the user data
 - Should display at the bottom of the user profile
 - Should display the user in the charts
- Display account details
 - User ID, Email, Email Verification
- Display User Stats
 - Average WPM
 - Average Accuracy
 - Top 5 frequent mistakes the user makes
- Modified the CSS and tested if the correct modules displayed correctly.

Display-textbox (Brian & Colin & Hang & Franz):

- Check to see that text appears in the text box.
- Obtain random text from the back end depending on the user
- Send display string to other components
- Wrap letters with span
 - If letters from the display textbox is the same as the letter in the typing textbox add it to the class 'correct'
 - If letters from the display textbox is not the same as the letter in the typing textbox add it to the class 'incorrect'
 - Otherwise, the letters in the display textbox should be in the span class of 'default'
- Type into the textbox to make sure that right/wrong letters are displayed accordingly.
- Modified the CSS and tested if the correct modules displayed correctly.

Forgot-passwords (Colin & Franz):

- Modified the CSS and tested if the correct modules displayed correctly.

Front-page (Colin & Hang):

- Check to see that every component is displayed.
- Use console.log to make sure test done/timer done signals are received and passed to child components.

Header (Franz):

- Modified the CSS and tested if the correct modules displayed correctly.
- Login button should redirect the user to the login page

Lin-front-page (Brian & Colin & Hang & Franz):

- Check to see that every component is displayed.
- Use console.log to make sure test done/timer done signals are received and passed to child components.
- Modified the CSS and tested if the correct modules displayed correctly.

Lin-header (Colin & Franz)

- Click on profile icon to view dropdown with “My Profile” and “Log out” listed
 - “My Profile” should route to the dashboard
 - “Log Out” should log the user out and route them to Front-page
- Modified the CSS and tested if the correct modules displayed correctly.

Login-button (Colin & Franz):

- When pressed, should redirect to the login page
- Modified the CSS and tested if the correct modules displayed correctly.

Lout-display-textbox (Brian & Colin & Hang & Franz):

- Check to see that text appears in the text box.
- Obtain random text from the back end
- Send display string to other components
- Wrap letters with span
 - If letters from the display textbox is the same as the letter in the typing textbox add it to the class ‘correct’
 - If letters from the display textbox is not the same as the letter in the typing textbox add it to the class ‘incorrect’
 - Otherwise, the letters in the display textbox should be in the span class of ‘default’
- Type into the textbox to make sure that right/wrong letters are displayed accordingly.
- Modified the CSS and tested if the correct modules displayed correctly.

Lout-result (Brian & Colin & Hang & Franz)

- Check to see that the WPM and Accuracy are displayed without the improvements
- Use console.log to make sure necessary variables are received from the typing-textbox and timer component.
- We break the string into words, compare the words to the words from the display textbox, then compare each letter if they are not the same word. We tested this by using console.log.
- Modified the CSS and tested if the correct modules displayed correctly.

Refresh (Franz)

- Refresh button in the main page refreshes the main page
- Refresh button is displayed in the main page below the typing textbox
- Modified the CSS and tested if the correct modules displayed correctly.

Sign-in (Colin & Franz):

- Checked each buttons functionality
 - Login Button should let the user login if the username and password fields are correct
 - If not correct then it should not let them log in
 - Log in with google should allow the user to see a google login popup and log them in once they have selected a google user
 - Forgot passwords, sign up, and back to the typing test button should redirect to the correct page
- Typing fields should allow the user to type
 - Obtain data and get it from the back end
- Modified the CSS and tested if the correct modules displayed correctly.

Sign-up (Colin & Franz):

- Checked each buttons functionality
 - Sign Up Button should let the user sign up if the username and password fields have not been already signed up
 - If signed up, display an error message
 - Continue in with google should allow the user to see a google login popup and sign them up once they have selected a google user
 - Already have an account? Log In, and back to the typing test button should redirect to the correct page
- Typing fields should allow the user to type
 - Obtain data and send it to the back end
- Modified the CSS and tested if the correct modules displayed correctly.
- Tested to see if verification email is sent and received by the user.

Timer (Franz & Hang)

- Check to see that the timer updates correctly.
- Make sure timer stops when test is done.
- Use console.log to make sure the current time is saved and emitted to the parent component.

Typing-textbox (Hang):

- Use console.log to make sure textbox is disabled when test is done and that the user input gets emitted to parent component.

User-data (Colin):

- Called component in lin-front-page to verify that it is displaying user's data correctly

Verify-email (Colin & Franz):

- Modified the CSS and tested if the correct modules displayed correctly.

Result (Brian & Colin & Hang & Franz):

- If logged in wpm, accuracy, and improvements should be displayed

- Use console.log to make sure necessary variables are received from the typing-textbox and timer component.
- We break the string into words, compare the words to the words from the display textbox, then compare each letter if they are not the same word. We tested this by using console.log.
- Verify that the wpm, accuracy, and improvement are matching with the expected value calculated using a calculator.
- Modified the CSS and tested if the correct modules displayed correctly.
- Verify that results are still displayed even if user doesn't finish the test.

Back-end:

api_controller

- Make every HTTP request listed in back-end/routes.js using Postman
 - Verify data received from the response is correct
 - Verify data received from the response is formatted correctly
 - Verify that *getrandomadpttext*, *store_usr_data*, and *get_usr_data* create firestore document for specified user ID if one does not already exist.
 - Verify that *getrandomadpttext* algorithm works correctly using console.log.