Boston University
Electrical & Computer Engineering
EC463 Capstone Senior Design Project

First Prototype Testing Plan

Crowdsourcing platform for rating and improving research code: Reproducibility, Reusability, Readability (RE3)

Submitted to

Ana Trisovic
anatrisovic@g.harvard.edu

by

Team #5
RE3

Team Members

Andreas Francisco De Melo Oliveira andoliv@bu.edu
Ethan Hung ehung@bu.edu
Jyotsna Penumaka jyotsnap@bu.edu
Layan Bahaidarah layanb@bu.edu
Lukas Rosario lukasr@bu.edu

Required Materials

Hardware:

- Computer/Laptop - *Thats is it!*

Software:

- Front End
    - React JS web application
    - Tailwind CSS
    - Firebase JavaScript SDK
- Backend:
    - Google Firebase
        - Firestore
        - Storage
        - Authentication
    - Python Scripts
        - Upload all snippets to google storage
        - Read in user snippet ratings into json

Set Up:

Since our project is just software based all of the set up required is cloning the git repository with the following command:

'git clone https://github.com/BostonUniversitySeniorDesign/21-05-Re3.git'

Following that command simply go to the re-3-client folder inside the cloned repository, and run 'npm install', followed by 'npm start' in the terminal. The browser should open the website for our application on localhost:3000 and we can start testing it. Also, a python script is run beforehand to populate the storage with code snippets that will be displayed on the web page for users to evaluate (in terms of readability).

In bullet points:

1. Clone the main branch from *https://github.com/BostonUniversitySeniorDesign/21-05-Re3.git*
2. To open the project on a web browser, follow the following steps:
    2.1. cd 21-05-Re3/re3-client
    2.2. npm install
    2.3. npm start
3. Run the python script, *upload_snippet.py*
    3.1. Check google storage to find snippet files.

Testing Procedure:

1. Register user by clicking on *Login* button
2. Complete onboarding survey after successfully completing login
3. Click on the *Download* button on the top right hand corner to download the currently displayed file.
4. Click on any of the numbers below the text box to rate the snippet
5. Continue rating snippets until all 100 have been rated.
6. Run python script *upload_and_fetch.py* to extract user ratings for all snippets into a json file.

Measurable Criteria:

The criteria for successful running and output is as follows:

1. User should be directed to *Home Page* when the project is opened in a web browser
2. *Login Button* should direct a user to the login page
    a. If the user is registering, he should be redirected to the *Onboarding Page.*
    b. On clicking the *continue button* on the *Onboarding Page* the following information should be stored in the firestore under:

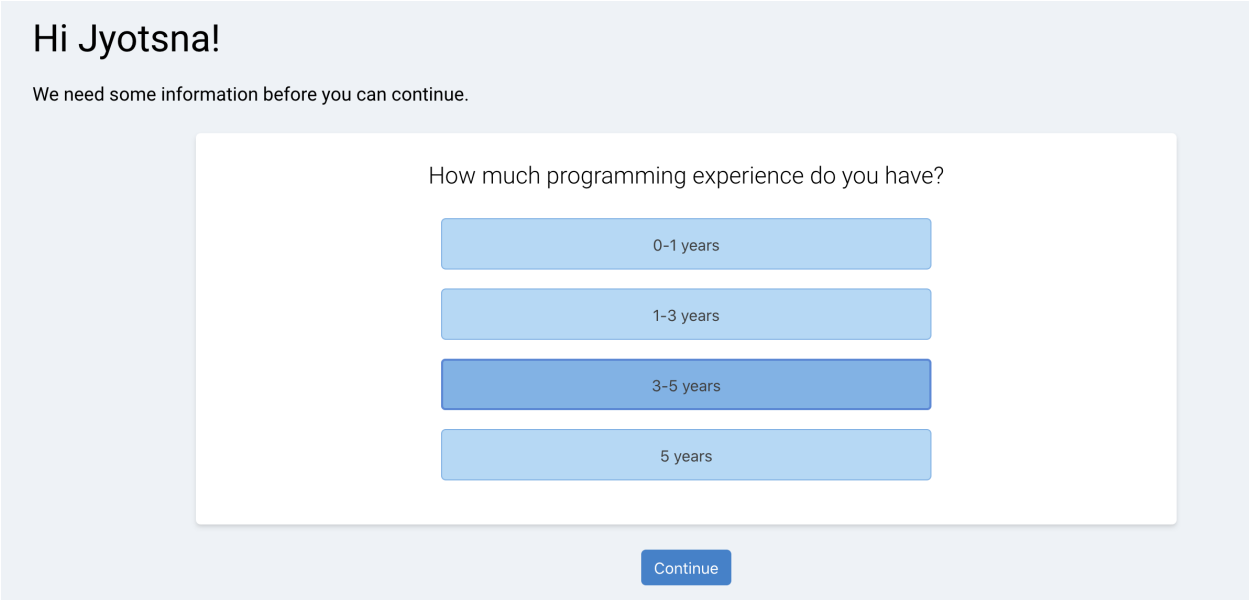
        users → uid → { currentSnippet, email, isOnboarded, experience, name }

    c. If the user logins again after signing out, he should be redirected to the *Rating Page* instead of the *Onboarding Page*
3. User should be registered in google firebase authentication (uid, name and email)
4. The text box displaying the code snippet should transition to the next snippet when the user clicks on a rating
5. Each rating should be updated on firestore under:

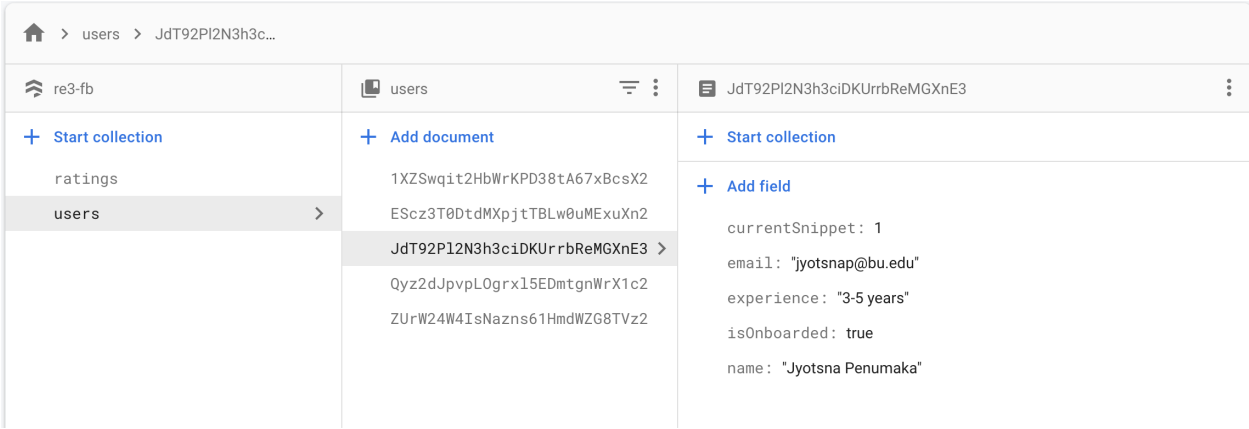
        ratings → snippet# → user_uid : rating

6. The *Download File button* should download the current snippet on the user's computer.
7. When the user clicks on the *Sign out button* they should be redirected to the *Login Page*
8. If the user enters an invalid url, they should be directed to the *NotFound Page*.

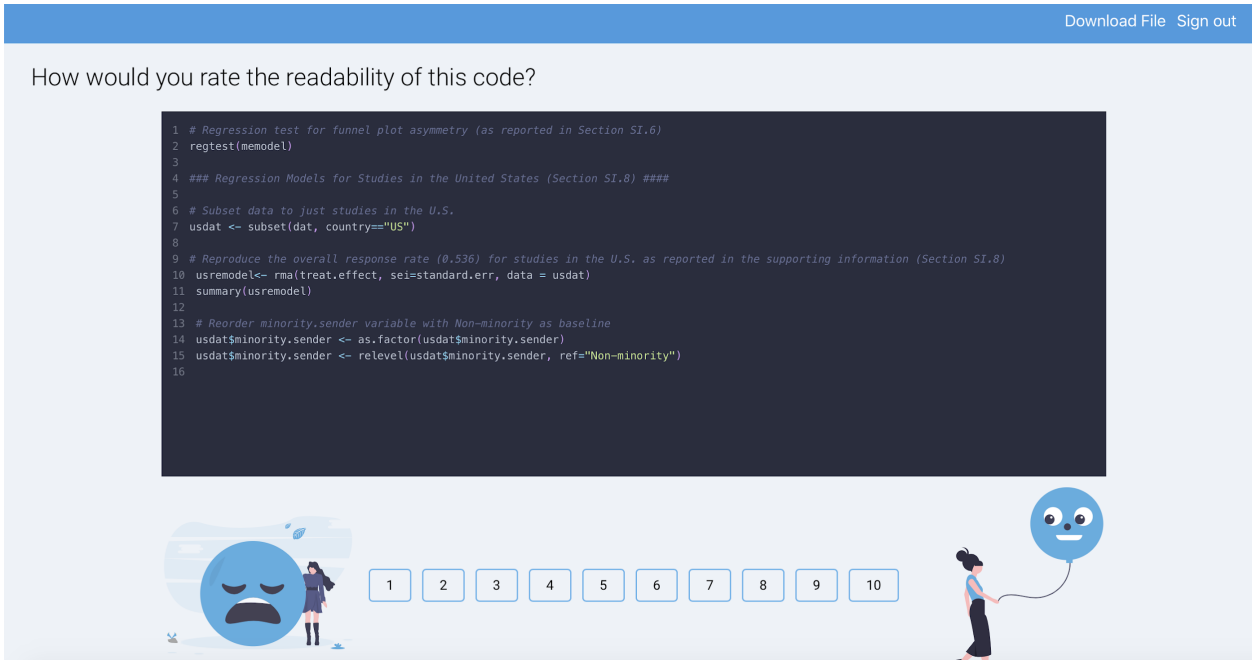
What should you expect?



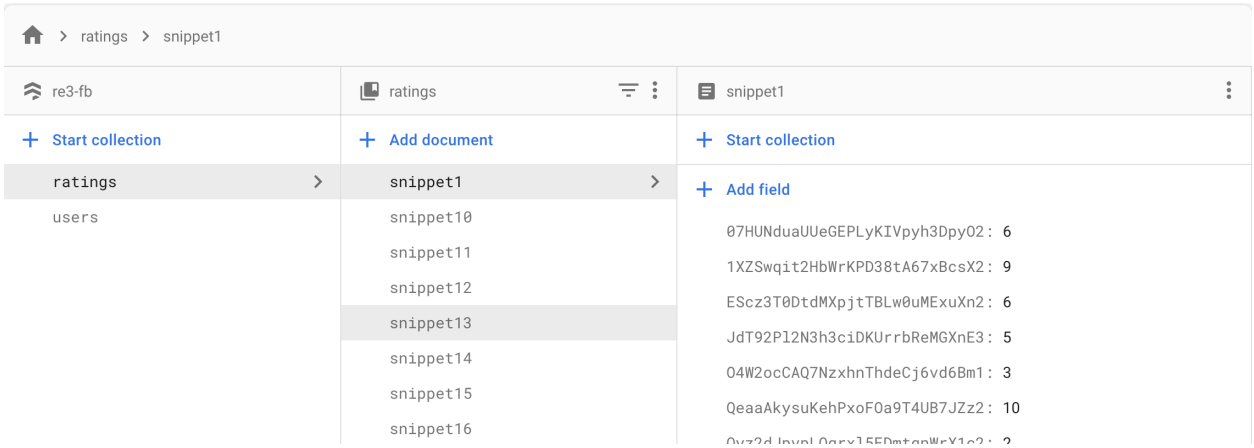
Onboarding Page



User Info is stored on Firestore after Onboarding



Dashboard (Rating snippets page)



Rating is stored on Firestore after user finishes rating a snippet