# ***Business requirements document***

**Name:** Akeem Jokosenumi

**Module:** Software Testing

**Website Tested:** GitHub

## ***Introduction***

***1.1. Purpose***

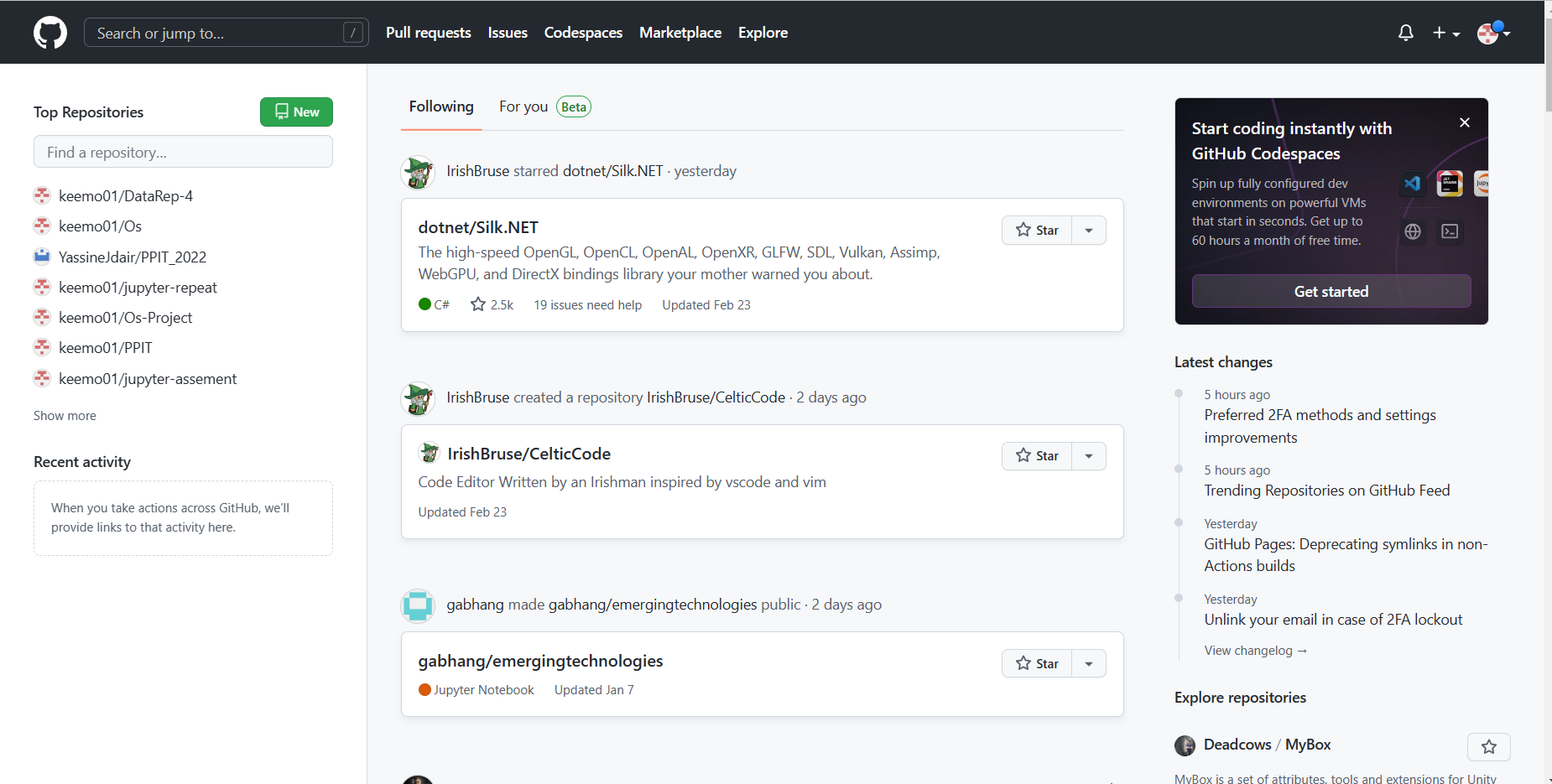
*The purpose of this software testing project is to ensure the quality of GitHub web application through a comprehensive testing approach. The testing plan will provide a test strategy based on the project's objectives, start and end dates, and assumptions. The execution strategy will describe how the tests will be performed, including the creation of test cases, entry/exit criteria, scheduling, and data strategy. The test management process will handle the logistics of the testing and all events that occur during execution, such as identifying and reporting defects and implementing fixes to the bugs found. Through this testing project, we aim to provide GitHub users with a reliable and high-quality platform for their software development needs.*

***Objective***

*My software testing project aims to thoroughly test the features and functionality of the well-known GitHub software development platform. Version control, collaboration, and project management are all common uses for GitHub among developers and software teams. This project's goal is to create and execute a set of test cases that cover various aspects of the platform, such as user registration, repository creation, branch management, and error handling. I hope to identify and report any defects or issues that may impact the platform's usability or functionality by conducting extensive testing. Furthermore, by providing valuable feedback to the development team, this project hopes to contribute to the overall improvement of the GitHub platform.*

*.*

***Test Case 1.0 Launch*** [***www.github.com***](http://www.github.com)



***Test case 2.0 Sign Up Feature***

*1. Access www.github.com website.*

*2. Navigate to the sign-up button located in the website's navigation panel.*

*3. Fill in the required fields with valid information, including a username, email, opting-out of updates, and a password.*

*4. Verify the anti-bot puzzle and select the free plan option.*

*5. Enter the digits provided for email confirmation.*

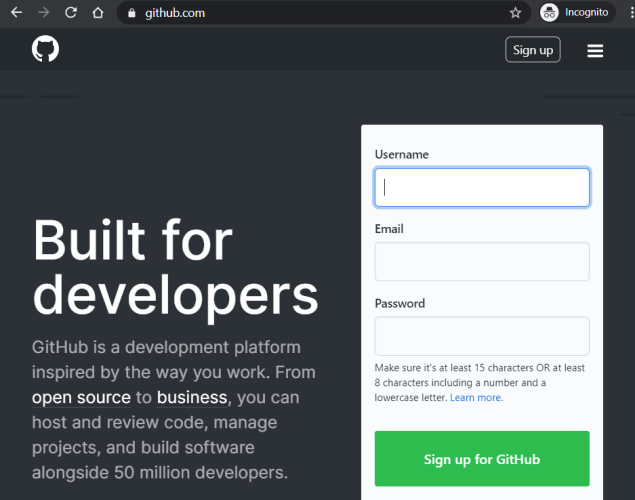
*6. Select the "just me" and "Student" options, then proceed by clicking the blue "continue" button.*

*7. On the tools page, select the blue "continue" button.*

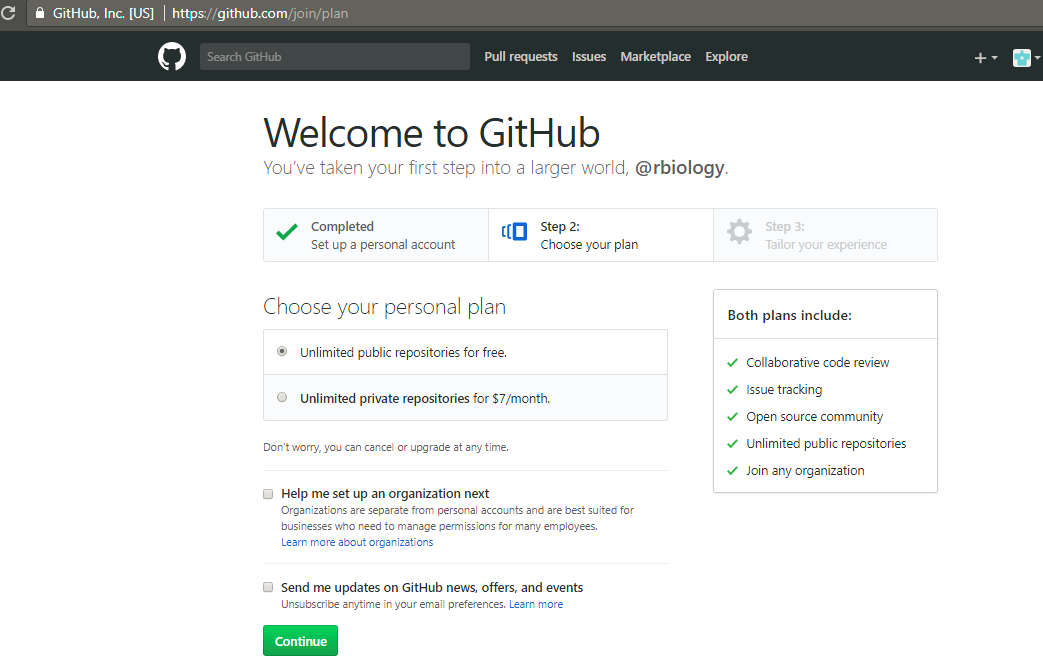
*8. Finally, choose the "continue as free" option on the payment page.*



*This is the Sign-up button up above.*



*Sign up page to create an account up above.*



*Choose your plan and then click continue.*

Graphical user interface, application

Description automatically generated

*This is the page your taken to when the account is created.*

***Test case 3.0 Existing Username***

*1. Attempt to sign up with an existing username.*

*2. Verify that an error message is displayed, informing the user that the selected username is already taken.*

*3. Verify that the user is prompted to select a different username.*

A screenshot of a computer

Description automatically generated with medium confidence

*Username already exists so account can’t be created.*

***Test case 4.0 Invalid credentials***

*1. Enter an invalid username and password combination.*

*2. Click on the "Sign in" button.*

*3. Verify that an error message is displayed, informing the user that the login details are incorrect.*

*4. Verify that the user is prompted to enter valid login credentials.*

A screenshot of a computer

Description automatically generated with medium confidence

***Test case 5.0 new repository***

*1. Click on the "+" icon in the top-right corner of the navigation panel.*

*2. Select "New repository" from the dropdown menu.*

*3. Enter a valid repository name and an optional description.*

*4. Select the appropriate repository visibility (public or private).*

*5. Choose the appropriate repository initialization option (initialize this repository with a README, add. gitignore or add a license).*

*6. Click the "Create repository" button.*

*7. Verify that the new repository is created and displayed on the user's repository page.*

Graphical user interface, text, application, chat or text message

Description automatically generated

*Click on New on the top left.*

Graphical user interface, text, application, email

Description automatically generated

*Fill in all the details.*

*Graphical user interface, text, application, email

Description automatically generatedRepository is successfully created.*

***Test case 6.0 Repository is visible on the user's profile.***

*1. Sign in with valid credentials.*

*2. Navigate to the user's repository page.*

*3. Verify that the newly created repository is displayed on the user's repository page.*

*4. Navigate to the user's profile page.*

*5. Verify that the newly created repository is visible on the user's profile page.*

Graphical user interface, text, application

Description automatically generated

*Click on profile on the top right then go to your repositories.*

Graphical user interface, application

Description automatically generated

Here you can see the repositories have been added and are visible on my repositories page.

***Test case 7.0: Verify error message for invalid repository details.***

*1. Click on the "Sign in" button and enter valid credentials.*

*2. Navigate to the "Create New Repository" page.*

*3. Enter invalid details (e.g. empty repository name, special characters in the repository name, etc.).*

*4. Click on the "Create repository" button.*

*5. Verify that an error message is displayed indicating that the details entered are invalid.* Graphical user interface, text, application, email

Description automatically generated

*Automatically changes the name so it can be created*

***Test case 8.0: Verify duplicate repository name.***

*1. Navigate to the user's repository page.*

*2. Note down the name of an existing repository.*

*3. Click on the "Create New Repository" button.*

*4. Enter the same name as the existing repository in the "Repository name" field.*

*5. Click on the "Create repository" button.*

*6. Verify that an error message is displayed indicating that a repository with the same name already exists.*

Graphical user interface, text, application, email

Description automatically generated

***Test case 9.0: Verify that a user can create a new branch in a repository.***

*1. Click on the repository where the new branch will be created.*

*2. Click on the "Branch" dropdown menu.*

*3. Enter a name for the new branch in the "Find or create a branch..." field.*

*4. Press "Enter" or click on the "Create branch: [branch-name]" button.*

*5. Verify that the new branch is created and listed under the "Branch" dropdown menu.*

Graphical user interface, text, application, email

Description automatically generated*This is to see the branches click on view all.*

Graphical user interface, text, application, chat or text message

Description automatically generated

*Option to create a new branch.*

***Test case 10.0: Verify that the new branch is visible in the repository's branch list.***

*1. Navigate to the user's repository page.*

*2. Click on the repository where the new branch will be created.*

*3. Click on the "Branch" dropdown menu.*

*4. Enter a name for the new branch in the "Find or create a branch..." field.*

*5. Press "Enter" or click on the "Create branch: [branch-name]" button.*

*6. Verify that the new branch is created and listed under the "Branch" dropdown menu.*

*7. Click on the "Branch" dropdown menu again.*

*8. Verify that the new branch is visible in the repository's branch list.*

Graphical user interface, text, application, email

Description automatically generated

*This is to show that the new branch is created and visible.*

***Test case 11.0: At branch formation, check that the redirection is accurate.***

*1. Navigate to the user's repository page.*

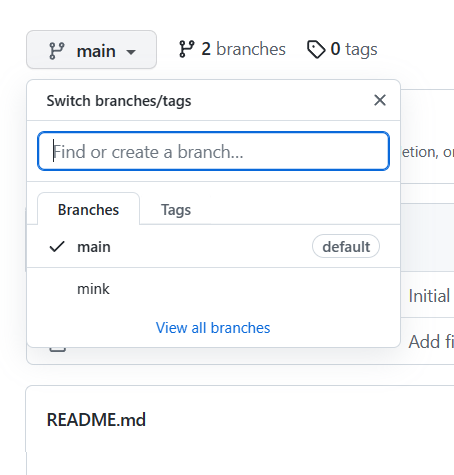
*2. Click on the repository where the new branch will be created.*

*3. Click on the "Branch" dropdown menu.*

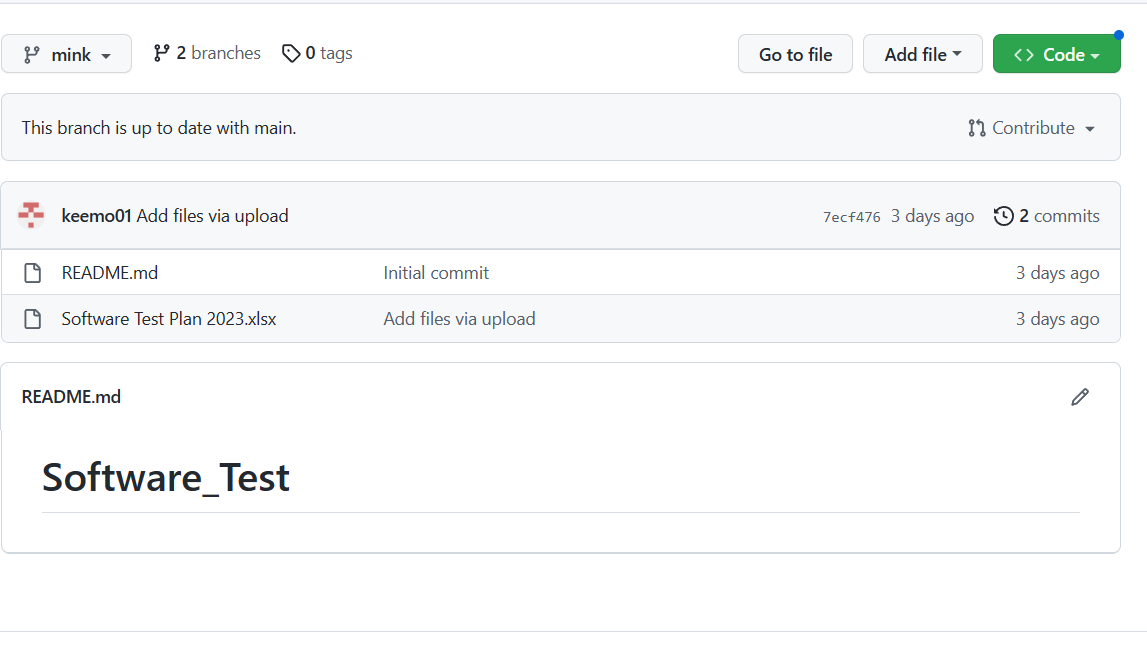
*4. Enter a name for the new branch in the "Find or create a branch..." field.*

*5. Press "Enter" or click on the "Create branch: [branch-name]" button.*

*6. Verify that the user is redirected to the correct page upon creating the new branch.*



*This is where you get option to switch branch*



*Now I can see branch page has been switched and is fine n up to date*

***Test case 12.0: Check for duplicate branch name when creating a new branch.***

*1. Navigate to the user's repository page.*

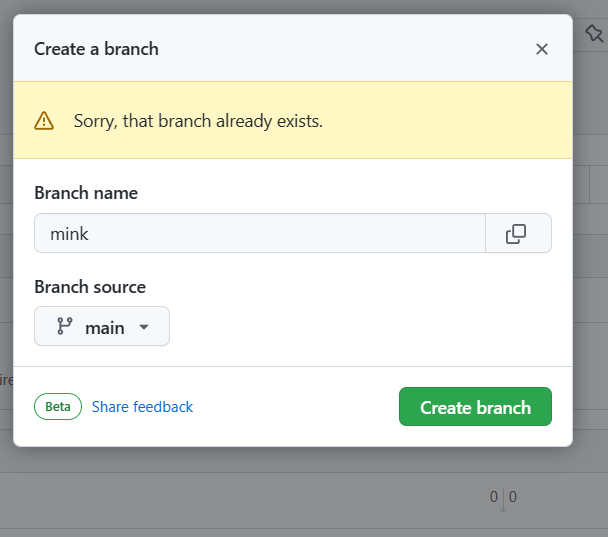
*2. Click on the repository where the new branch will be created.*

*3. Click on the "Branch" dropdown menu.*

*4. Enter the name of an existing branch in the "Find or create a branch..." field.*

*5. Press "Enter" or click on the "Create branch: [branch-name]" button.*

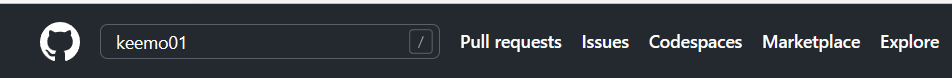
*6. Verify that an error message is displayed indicating that the branch name already exists.*



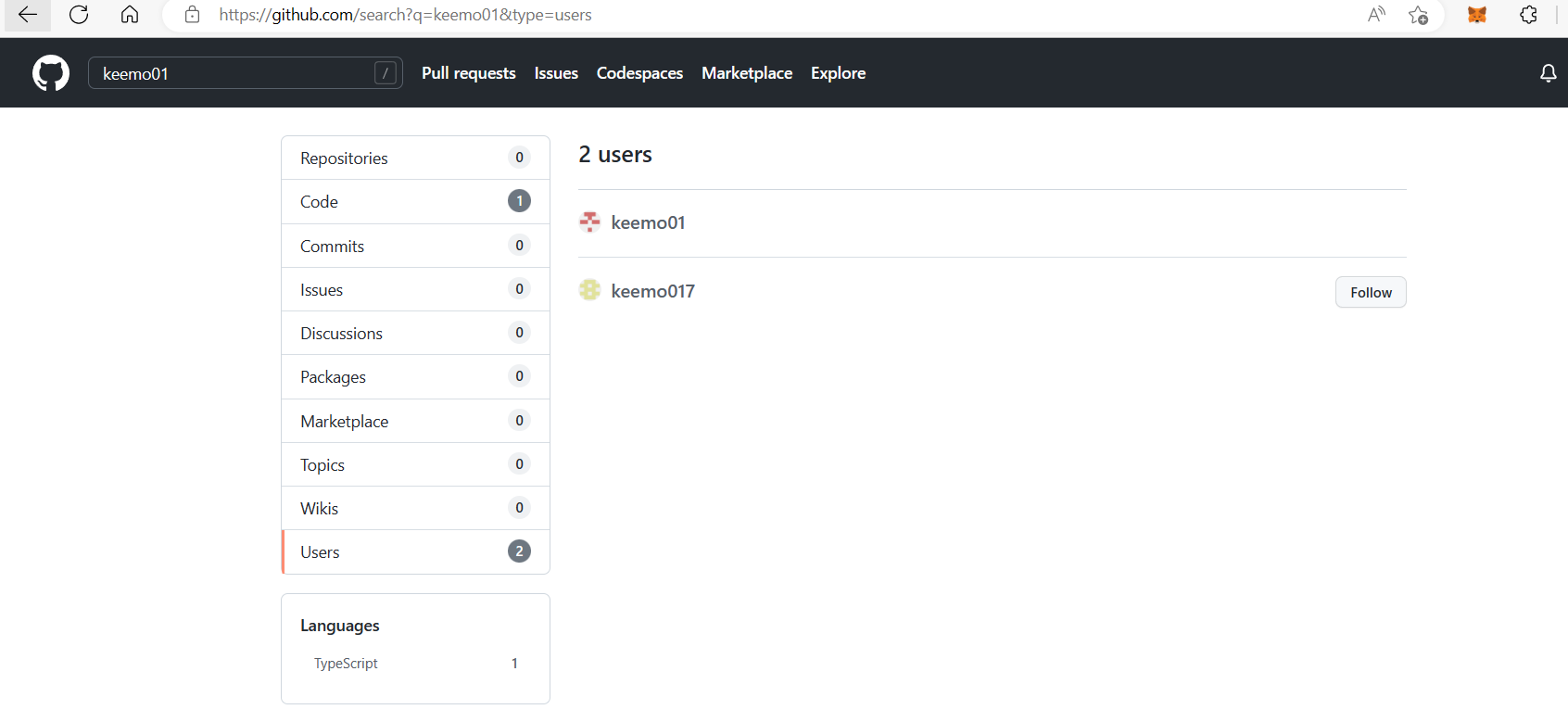
*This pops up when you try create duplicate branches*

***Test case 13.0:* *Search user from search engine.***

1. *Launch the web engine www.google.com and type the github 'Keemo01' and click enter.*
2. *Click first Result ' Keemo01' to see the page.*



*This is where you go input the user you wish to search*



*Here you click on the user you wish to view and will be redirected to their page*

***Test case 14.0:* *Unfollow a user.***

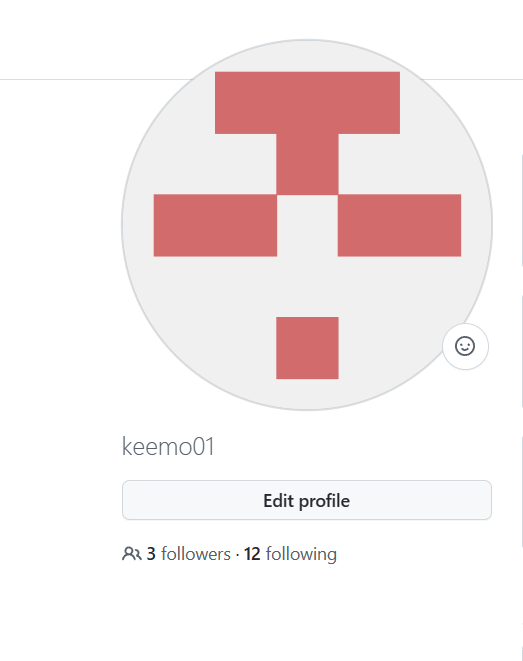
*1. Click on the "Following" tab.*

*2. Locate the user that the current user wishes to unfollow.*

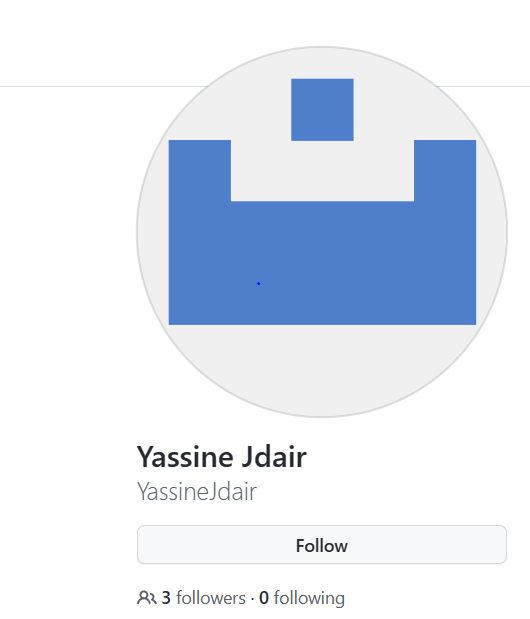
*3. Click on the "Unfollow" button next to the user's name.*

*4. Verify that the "Follow" button replaces the "Unfollow" button next to the user's name.*

*5. Verify that the user's updates are no longer displayed on the current user's newsfeed.*

*Click on following*

*Now you click the unfollow and it should change to follow which indicates the user has been unfollowed*



*Now you can see it’s been unfollowed successfully*

***Test case 15.0:*** ***Verify that a user is able to upload a file to a repository***

*1. Launch the GitHub website*

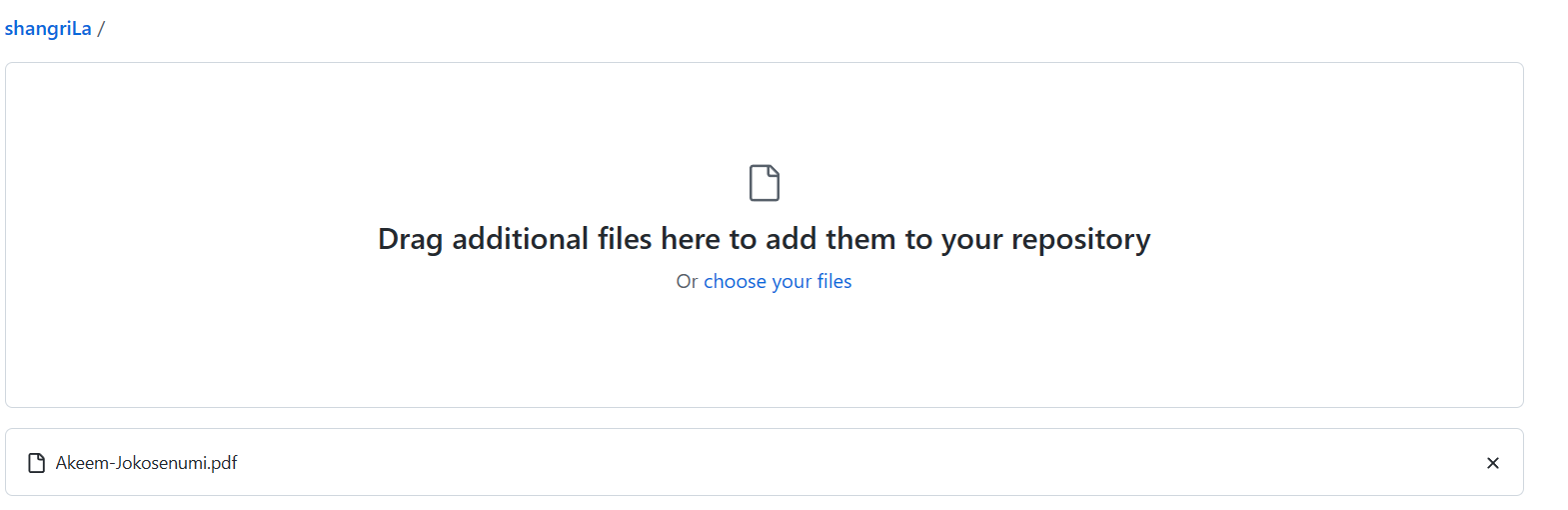
*2. Navigate to the repository where the file is to be uploaded*

*3. Click on the "Upload files" button*

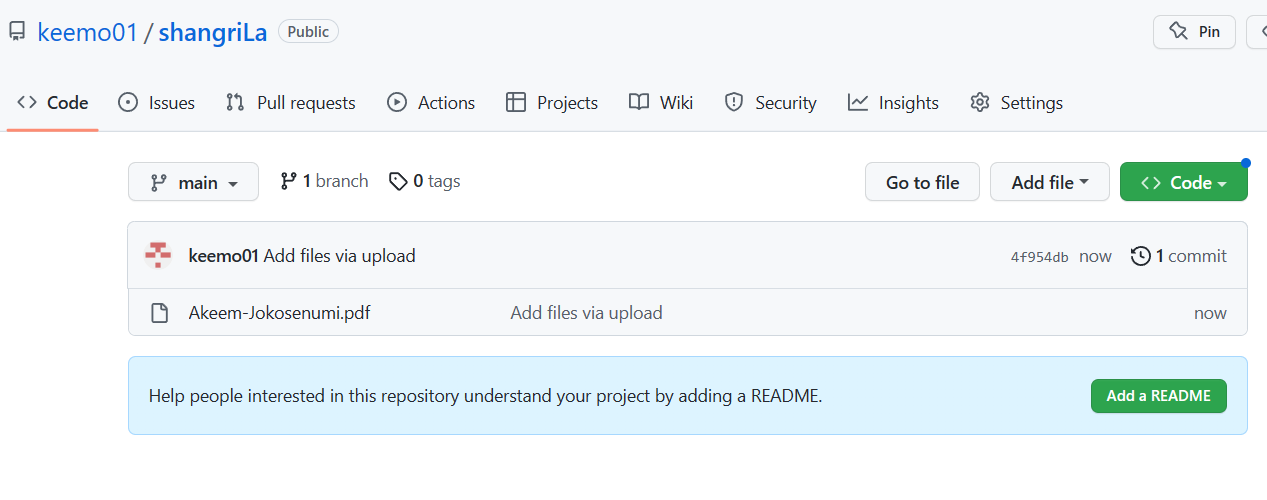
*4. Select a file to upload from the local machine*

*5. Click on the "Commit changes" button*

*6. Verify that the file has been uploaded successfully and is visible in the repository.*



*Click choose your files for the files you wish to upload*



*Here you can see the pdf I wished to upload is now visible.*

***Test case 16.0:*** ***Verify that a user is able to confirm their email address on GitHub***

*1. Click on the "Emails" tab*

*2. Click on the "Verify" button next to the email address to be confirmed*

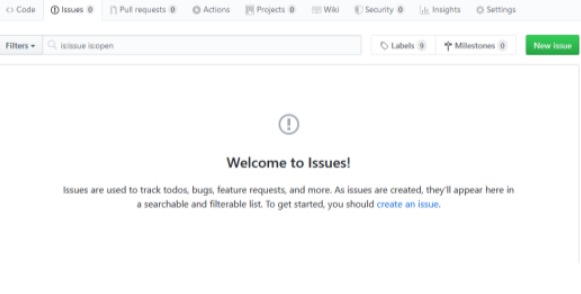
*3. Check the user's email inbox for the confirmation email*

*4. Follow the instructions in the confirmation email to confirm the email address*

*5. Return to the GitHub website and verify that the email address is now confirmed*

***Tools:***

*During my software testing project, I utilized various tools to create and execute test cases for GitHub. One of the primary tools that I used was Selenium WebDriver, which allowed me to automate web browsers and perform various actions such as clicking buttons, filling out forms, and verifying results. I also used GitHub's own testing tool, GitHub Actions, to automatically run tests on code changes and pull requests. This allowed me to catch bugs and errors early on in the development process. Additionally, I used JUnit and TestNG frameworks to write and organize test cases in a systematic manner. These frameworks also provided detailed reports and allowed me to easily identify and track any failures or issues that arose during testing. Overall, the combination of these tools enabled me to efficiently and effectively test the functionality of GitHub and ensure a high level of quality.*



***Conclusion:***

*I gained valuable experience in creating and managing test cases for software development while working on the GitHub testing project. This project has taught me the value of a well-defined testing strategy, which includes entry and exit criteria, test cycles, and test metrics. I've learned how to collaborate with a team of developers, project managers, and other testers to ensure that all testing efforts are in line with the project's goals and objectives.*

*I've learned how to write effective test cases, use tools like GitHub Issues to track defects and manage test cases, and work with automation scripts to streamline testing efforts. I've also learned the value of communicating with team members clearly and on a regular basis, including providing regular status updates, reporting defects, and identifying opportunities for process improvement.*

*One of the most important lessons learned from this project is the value of adaptability and flexibility in the testing process. As the project progresses and new features are added or requirements change, the ability to adjust testing efforts and maintain a focus on the most important testing objectives becomes critical. This necessitates a thorough understanding of the project's goals and priorities, as well as the ability to collaborate with team members to ensure that testing efforts are in line with these goals.*

*Overall, this GitHub testing project has been a valuable learning experience, providing me with hands-on experience in creating and managing software development test cases. I am excited to continue honing these skills and contributing to future software development testing projects.*