|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Requirement | Precondition(s) | Steps | Test Data | Expected result | Actual Result | Pass/Fail | Tester |
| 1 | User registration | A unique email and password meeting requirements | User clicks register, then enters info | <valid email>  <valid password> | Account info is registered and saved | Same as expected | **Pass** | Travis Lamb |
| 2 | User login | A valid account | On login page, user enters email and password | <valid email>  <valid password> | User is logged in | Same as expected | **Pass** | Travis Lamb |
| 3 | User logout | A logged in account | After logout, pages should be inaccessible | Cookies, LocalStorage, and inability to access other pages | User is logged out | Same as expected | **Pass** | Travis Lamb |
| 4 | Change Password | A valid account | On forgot password page, user enters new password and emailed confirmation code | <username>  <confirmation code><new password> | User’s password is changed | Users password is changed | **Pass** | Travis Lamb |
| 5 | Verification Email | An account registration | User registration | <registration email> | User receives registration email | Email sent to user | **Pass** | Travis Lamb |
| 6 | Review Assignment Notification | A valid account and  a review | User is assigned to a review | <review notification> | User receives a notification on review assignment | None | **Fail** | Christina Nguyen |
| 7 | Review approval notification | A valid account and review | User’s review is approved | <review notification> | User receives a notification on review approval | None | **Fail** | Christina Nguyen |
| 8 | Review rejection notification | A valid account and review | User’s review is rejected | <review notification> | User receives a notification on review rejection | None | **Fail** | Christina Nguyen |
| 9 | The user can assign a code review | Two valid accounts and a review | A user can assign another to a review | <review>  <user email> | User is assigned to review | None | **Fail** | Michael Bloomquist |
| 10 | The user can view a diff between file versions | A valid account, file with version history | Provide filename for file on server | <filename>  <diff> | User is shown a diff for the given file name | Same as expected | **Pass** | Michael Bloomquist |
| 11 | The user can view diffs between all the file versions in one folder | A valid account and .git folder in a valid directory | Provide the .git folder directory | <.git folder>  <diff> | User is shown all current diffs of files | Same as expected | **Pass** | Sarah Ramazani |
| 11 | The user can leave inline comments on a review | A valid account assigned to a review | Select line to comment and press add inline comment button | <user email>  <review>  <inline comment> | An inline comment is left and visible on the review | None | **Fail** | Michael Bloomquist |
| 12 | The user can leave comments on a review (not inline) | A valid account assigned to a review | Press add comment button | <user email>  <review>  <comment> | A comment is left and visible on the review | None | **Fail** | Michael Bloomquist |
| 13 | Users assigned to a review can approve changes | A valid account assigned to a review | Press approve review button | <user email>  <review> | An approval is left and visible on the review | None | **Fail** | Michael Bloomquist |
| 14 | Users assigned to a review can reject changes | A valid account assigned to a review | Press reject review button | <user email>  <review> | A rejection is left and visible on the review | None | **Fail** | Michael Bloomquist |
| 15 | ReviewCreator retrieves diff from flask API | A valid filename | N/A | <filename>  <diff text> | The diff text is stored in the review creator | Same as expected | **Pass** | Michael Bloomquist |
| 16 | The ReviewCreator can create several different diffs | A valid filename | N/A | <filename>  <diff text> | Multiple DiffDisplay components can be rendered | Same as expected | **Pass** | Michael Bloomquist |
| 17 | DiffDisplay is collapsible | A rendered DiffDisplay | User presses +/- at topleft of diff | <diff text> | DiffDisplay component can be opened and closed | Same as expected | **Pass** | Michael Bloomquist |
| 18 | Diff formatting is consistent | A rendered DiffDisplay | N/A | <diff text> | Diff formatting is consistent from API to component | Line indices affect indentation | **Fail** | Michael Bloomquist |
| 19 | Cognito registration also pushes to DB | A valid user account defined by Cognito | Register new user as usual | <New user data> | DB retrieves new user | As expected | **Pass** | Travis Lamb |
| 20 | Users can “create review” by selecting file, data is pushed to DB | File to be pushed to DB | Go to projects page, select file, click create review | <file to be parsed for DB insertion> | DB retrieves new file data | Database receives blob, react.js processes blob | Pass | Travis Lamb |
| 21 | Users can “create review” by selecting file, data is pushed to DB | File to be pushed to DB | Go to projects page, select file, click create review | <file to be parsed for DB insertion> | DB retrieves new file data | Database receives all values correctly | Pass | Travis Lamb |
| 22 | The user can leave inline comments | A valid account and project file | Go to projects page and click on the line of code that the user wants to comment on | <Username> <comment message>  <user email> | A comment is on the projects page and in view | A hardcoded box is shown for now in place of the actual comment box | **Fail** | Christina Nguyen |
| 23 | Hardcoded text is split and displayed on the page line by line | A valid account and a “project file” | Go to projects page and view project file on the page | <Hardcoded Text File> | Text is displayed on the page line by line according to the correct index | Text is displayed on the page line by line according to the correct index | **Pass** | Christina Nguyen |
| 24 | Database API receives new review and displays all user reviews | User exists in the database | User selects create review, selects file, confirms review | <File> | Review is received by database, database updates works\_on\_reviews table | Yeah it works.. after 1200 test cases where it didn’t lmao | **Pass** | Travis Lamb |
| 25 | EC2 Instance listens for http requests | EC2 instance is running | Send get request to apache2 routed api | <apache logs> | Get requests function properly | Same as expected | **Pass** | Michael Bloomquist |
| 26 | EC2 instance listens for https requests | EC2 instance is running, has valid SSL cert | Send post request to apache2 routed api | <apache logs> | Post requests function properly | Same as expected | **Pass** | Michael Bloomquist |
| 27 | EC2 instance’s apache2 redirects nodejs requests through proxy | EC2 instance is running, has valid SSL cert, nodejs app is running | Send get or post request to apache2 routed nodejs api | <apache logs> | Nodejs receives proxy api requests without breaking flask app | Nodejs receives proxy api requests, breaks flask app | **Fail** | Michael Bloomquist |
| 28 | Users can be invited to projects | All live sites are operational, user is inviting valid user, user has a project | User clicks invite to collaborate, enters valid username | <valid username><valid project> | User is invited to collaborate on project | User is invited to collaborate on project | **PASS** | Travis Lamb |
| 28 | Users can be invited to review a file | All live sites are operational, user is inviting valid user, user has a project, file exists in project | User clicks invite to review on project collaborator | <valid project>  <valid collaborator> | User is invited to review file, redirected on accept | As expected | **Pass** | Travis Lamb |
|  |  |  |  |  |  |  |  |  |