**Home page**

* **TC001:** Verify the GitHub homepage URL loads successfully. **Expected Result:** GitHub homepage should load without errors**.**
* **TC002:** Verify the page is responsive across different devices (desktop, tablet, and mobile). **Expected Result:** The layout adjusts appropriately to the screen size.
* **TC003:** Verify the GitHub logo is displayed at the top-left corner.  
  **Expected Result:** Logo is visible and redirects to the homepage on click.
* **TC004:** Verify the primary navigation bar is displayed. **Expected Result:** Navigation items are visible, clickable, and redirect to the correct pages.
* **TC005:** Verify the homepage hero section contains a search bar and promotional content. **Expected Result:** Search bar and text are aligned and functional.
* **TC006:** Verify the search bar accepts input. **Expected Result:** Search bar accepts text input without errors.
* **TC007:** Verify that clicking the search icon or pressing Enter shows search results. **Expected Result:** Search results page is displayed with related repositories.
* **TC008:** Verify the "Sign In" button is visible and redirects to the login page. **Expected Result:** Login page loads on click.
* **TC009:** Verify the "Sign Up" button is visible and redirects to the registration page. **Expected Result:** Registration page loads on click.
* **TC010:** Verify the footer contains quick links such as Terms, Privacy, Security, Product etc. **Expected Result:** Links are visible and clickable, redirecting to correct pages.
* **TC011:** Verify the links redirect to appropriate webpages. **Expected Result:** Links are visible and clickable, redirecting to correct pages.
* **TC012:** Verify the site is keyboard-navigable. **Expected Result:** All interactive elements are accessible via keyboard.
* **TC013:** Verify the homepage renders correctly on popular browsers (Chrome, Firefox, Safari, Edge).  
  **Expected Result:** The layout and functionality are consistent across browsers.

**Sign in Page**

* **TC001**: Verify that the GitHub logo is displayed at the top of the page.  
  **Expected Result**: GitHub logo is visible and clickable, redirecting to the homepage.
* **TC002**: Verify the presence of the following input fields and buttons:
  + Username or email address input field.
  + Password input field.
  + "Forgot password?" link.
  + "Sign in" button.
  + "Sign in with a passkey" link.
  + "Create an account" link.  
    **Expected Result**: All elements are visible, properly aligned, and clickable.
* **TC003**: Verify the "Username or email address" field accepts valid input.  
  **Steps**: Enter a valid username or email address.  
  **Expected Result**: Field accepts the input without errors.
* **TC004**: Verify the "Password" field accepts valid input.  
  **Steps**: Enter a valid password.  
  **Expected Result**: Field accepts input and masks it as dots or asterisks.
* **TC005**: Verify the "Password" field rejects empty input.  
  **Steps**: Leave the password field blank and click "Sign in".  
  **Expected Result**: Error message displayed, prompting the user to enter a password.
* **TC006**: Verify successful login with valid credentials.  
  **Steps**: Enter a valid username/email and password, then click "Sign in".  
  **Expected Result**: User is redirected to their GitHub dashboard.
* **TC007**: Verify login attempt with invalid credentials.  
  **Steps**: Enter incorrect username/email or password and click "Sign in".  
  **Expected Result**: An error message is displayed indicating invalid credentials.
* **TC008**: Verify the "Forgot password?" link redirects to the password recovery page.  
  **Expected Result**: Password recovery page is loaded when the link is clicked.
* **TC009**: Verify the "Sign in with a passkey" link redirects to the passkey authentication flow.  
  **Expected Result**: Passkey authentication page is loaded when the link is clicked.
* **TC010**: Verify the "Create an account" link redirects to the account creation page.  
  **Expected Result**: Account creation page is loaded when the link is clicked.
* **TC011**: Verify footer links (*Terms*, *Privacy*, *Docs*, etc.) redirect to the appropriate pages.  
  **Expected Result**: Each link in the footer opens the respective page.
* **TC012**: Verify behavior when "Sign in" is clicked without entering any credentials.  
  **Expected Result**: Error messages is displayed saying “Please fill out this field”
* **TC013**: Verify behavior when the email field contains an invalid email format whil signing up  
  **Expected Result**: Error message is displayed as “Email is invalid or already taken”.
* **TC014**: Verify behavior when the password field contains less than the minimum allowed characters.  
  **Expected Result**: Error message is displayed as “Password is too short”.
* **TC015**: Verify that the Sign In page functions correctly across multiple browsers (Chrome, Firefox, Safari, Edge).  
  **Expected Result**: The page loads and behaves consistently across browsers.
* **TC016**: Verify the password field does not display typed characters in plain text.  
  **Expected Result**: Input in the password field is masked.
* **TC017**: Verify that pasting passwords into the password field is allowed (if expected behavior).  
  **Expected Result**: Pasted passwords should be accepted.  
    
  Home Page
* **TC001**: Verify the URL of the home page after login.  
  **Expected Result**: The URL should be https://github.com/ or specific to the dashboard.
* **TC002**: Verify the GitHub logo and dashboard menu are visible at the top-left.  
  **Expected Result**: GitHub logo is present and redirects to the dashboard/home page on click.
* **TC003**: Verify the presence of the search bar at the top-right.  
  **Expected Result**: The search bar is functional and accepts input.
* **TC004**: Verify the "Create your first project" section is visible on the leftside bar.  
  **Expected Result**: "Create repository" and "Import repository" buttons are displayed.
* **TC005**: Verify that clicking "Create repository" redirects to the repository creation page.  
  **Expected Result**: Repository creation page is loaded successfully.
* **TC006**: Verify that clicking "Import repository" redirects to the import repository page.  
  **Expected Result**: Import repository page is loaded successfully.
* **TC007**: Verify the "Learn with a tutorial project" section is displayed.  
  **Expected Result**: Tutorial cards (*Introduction to GitHub*, *GitHub Pages*, *Code with Copilot*, *Hello GitHub Actions*) are visible and clickable.
* **TC008**: Verify each tutorial card redirects to the correct tutorial/project page.  
  **Expected Result**: Clicking a card navigates to the respective tutorial page.
* **TC009**: Verify the "See more tutorial projects" link is functional.  
  **Expected Result**: Redirects to a page showing additional tutorials.
* **TC010**: Verify the "Start a new repository" section is displayed.  
  **Expected Result**: The section includes fields for "Repository name" and visibility options (Public/Private).
* **TC011**: Verify the "Repository name" field accepts valid input.  
  **Expected Result**: The field accepts the input without errors.
* **TC012**: Verify the visibility options (Public/Private) are selectable.  
  **Expected Result**: User can switch between the two visibility options.
* **TC013**: Verify clicking the "Create a new repository" button creates the repository.  
  **Expected Result**: A new repository is created, and the user is redirected to its page.
* **TC016**: Verify the "Give feedback" link is functional.  
  **Expected Result**: Clicking the link redirects to the feedback submission page or modal.
* **TC017**: Verify the "Filter" button is functional.  
  **Expected Result**: Clicking the button shows filter options such as Announcements, releases, repositories etc.
* **TC018**: Verify the footer contains links to *Terms*, *Privacy*, *Support*, etc.  
  **Expected Result**: All links are visible and functional, redirecting to the correct pages.
* **TC019**: Verify behavior when trying to create a repository with a blank name.  
  **Expected Result**: An error message is displayed as “Repository creation failed” on the top of the page.
* **TC021**: Verify the dashboard works correctly on various browsers (Chrome, Firefox, Safari, Edge).  
  **Expected Result**: Layout and functionality remain consistent across browsers.

**Create Repository Page**

* **TC001**: Verify the URL of the "Create Repository" page.  
  **Steps**: Navigate to the page URL directly.  
  **Expected Result**: The URL should be https://github.com/new.
* **TC002**: Verify the page title and heading.  
  **Expected Result**: The title should display "Create a New Repository" and the heading should clearly state "Create a new repository".
* **TC003**: Verify the "Owner" dropdown is visible and populated.  
  **Expected Result**: The dropdown should list available owners (personal account or organizations).
* **TC004**: Verify changing the owner updates the repository's path.  
  **Expected Result**: The repository path updates accordingly.
* **TC005**: Verify the "Repository name" field is visible and editable.  
  **Expected Result**: The field accepts the input without errors.
* **TC006**: Verify the repository name field enforces validation rules.  
  **Expected Result**: An error message is displayed, and invalid names are not accepted.
* **TC007**: Verify the repository name suggestion link works.  
  **Expected Result**: The suggested name is filled into the repository name field.
* **TC008**: Verify the "Description" field is optional and editable.  
  **Expected Result**: Users can enter a description, but leaving it blank does not result in errors.
* **TC009**: Verify the "Public" and "Private" visibility options are displayed.  
  **Expected Result**: Both radio buttons are visible and selectable.
* **TC010**: Verify selecting each visibility option updates its description.  
  **Expected Result**: Descriptions under each option are updated to reflect the selected visibility.
* **TC011**: Verify the "Initialize this repository with" options are displayed.  
  **Expected Result**: Options to add a README file, a .gitignore file, or a license are visible.
* **TC012**: Verify the "Add a README file" checkbox is functional.  
  **Expected Result**: The checkbox can be toggled without errors.
* **TC013**: Verify the "Add .gitignore" dropdown displays templates.  
  **Expected Result**: The dropdown lists available templates and retains the selected template.
* **TC014**: Verify the "Choose a license" dropdown displays options.  
  **Expected Result**: The dropdown lists available licenses and retains the selected license.
* **TC015**: Verify the "Create repository" button when required fields are empty.  
  **Expected Result**: Error message - **New repository name must not be blank should be displayed.**
* **TC016**: Verify clicking "Create repository" creates a new repository when inputs are valid.  
  **Expected Result**: The repository is created successfully, and the user is redirected to the repository page.
* **TC017**: Verify error handling when attempting to create a repository with invalid input.  
  **Expected Result**: An appropriate error message is displayed, and the repository is not created.
* **TC018**: Verify the "Import a repository" link is functional.  
  **Expected Result**: Clicking the link redirects to the import repository page.
* **TC019**: Verify the sidebar and top navigation menu are functional.  
  **Expected Result**: Links in the sidebar and top navigation menu are clickable and functional.
* **TC020**: Verify behavior when creating a repository with a name that already exists.  
  **Expected Result**: An error message is displayed, and the repository is not created.
* **TC021**: Verify behavior when the network connection is interrupted during repository creation.  
  **Expected Result**: A proper error message or retry option is displayed.

**Issue Creation Page**

**TC001 :** Verify the 'Title' Field

* **Expected Result:** The "Title" field should accept and display the entered text.

**TC002:** Verify the 'Description' Field

* **Expected Result:** The "Description" field should accept and display the entered text.

**TC003:** Check 'Attach Files' Feature, Verify that files can be uploaded.

* **Expected Result:** The file should be uploaded and displayed as an attachment or embedded link.

**TC004:** Verify Label Assignment

* **Expected Result:** The selected label should be applied and visible under "Labels."

**TC005:** Verify Assignee Assignment

* **Expected Result:** The selected assignee(s) should appear in the "Assignees" section.

**TC006:** Check the 'Submit New Issue' Button

* **Expected Result:** A new issue should be created and visible in the repository's issue list.

**TC007:** Verify 'Milestones' Selection, Check if milestones can be assigned.

* **Expected Result:** The selected milestone should be assigned and displayed under "Milestones."

**TC008:** Verify 'Projects' Assignment

* **Expected Result:** The project should be linked and displayed under "Projects."

**TC009:** Verify Input Validation for title field, Ensure the "Submit new issue" button is disabled if required fields are empty.

* **Expected Result:** The button should remain disabled until the "Title" field is filled.

**TC010:** Check Accessibility

* **Expected Result:** The page should be navigable via keyboard and provide clear screen reader feedback.

**New Pull Request  
TC001:** Verify Creation of a New Pull Request

* **Expected Result:** A new pull request should be created and listed under "Open Pull Requests."

**TC002:** Verify Pull Request Diff View

* **Expected Result:** A detailed diff view of the changes should be displayed, highlighting additions and deletions.

**TC003:** Verify Adding Review Comments, Check if comments can be added to specific lines in the pull request.

* **Expected Result:** The comment should be visible next to the selected line.

**TC004:** Verify Pull Request Merging, Ensure a pull request can be merged.

* **Expected Result:** The pull request should be successfully merged, and the changes should reflect in the destination branch.

**TC005:** Verify Pull Request Labels, Check if labels can be added to a pull request.

* **Expected Result:** The selected label(s) should be visible on the pull request.

**TC006:** Verify Assignee Assignment, Ensure assignees can be added to a pull request.

* **Expected Result:** The selected assignee(s) should be visible under "Assignees."

**TC007:** Verify Status Checks

* **Expected Result:** The status of checks (passed, failed, pending) should be visible.

**TC008:** Verify Requesting Reviewers

* **Expected Result:** The selected reviewers should be notified and listed.

**TC009:** Verify Commenting on a Pull Request

* **Expected Result:** The comment should be visible under the "Conversation" tab.

**TC010:** Verify Closing a Pull Request, Ensure an open pull request can be closed without merging.

* **Expected Result:** The pull request should be closed and moved to the "Closed Pull Requests" section.

**TC0011:** Verify Branch Restrictions, Check if branch protection rules are respected during a pull request.

* **Expected Result:** The merge should be blocked until the branch protection rules are satisfied.

**TC012:** Verify Editing a Pull Request**,** Ensure the title and description of an open pull request can be edited.

* **Expected Result:** The updated title or description should be displayed.

**TC013:** Verify Notification System, Ensure notifications are sent to participants when actions are performed.

* **Expected Result:** Notifications should be sent for all actions performed on the pull request.