To effectively test the demoblaze.com website, it's essential to cover various aspects such as **Functional Tests (Both Manual and Automated Tests)** and **Non-Functional Tests** including **Performance, Security, Usability and Accessibility**. Below is a list of tasks along with estimated durations and prioritization based on importance:

**1. Functional Testing:**

* Verify navigation: Ensure all links, buttons, and menus navigate users correctly. (2 days)
* Test search functionality: Confirm that the search feature returns relevant results. (1 day)
* Validate product browsing: Ensure users can browse products by category and view details correctly. (2 days)
* Test cart functionality: Verify adding/removing items to/from the cart and updating quantities. (2 days)
* Checkout process testing: Ensure users can proceed through the checkout process smoothly. (3 days)
* Verify user registration/login: Test user registration, login, and logout functionalities. (2 days)

**2. Performance Testing:**

* Create a Performance test Script for major calls flows and APIs including :
  1. Login and Logout
  2. New user Sign up
  3. Load Products per category
     + One API call for each
     + Phones
     + Laptops
     + Monitors
  4. View products page
  5. Add product to Cart
  6. Check Out and Purchase request
  7. Home Page Load for https://www.demoblaze.com/index.html
* Types of Tests to execute :
  1. Load Test :
     + Output :
       - 1) Get the anticipated traffic on Demo blaze website for every day in hits per second from the Product Owner/ Business stakeholder
       - 2) Perform the Load test to check if the application can handle the expected traffic
  2. Stress Test : To check for 20% more than anticipated traffic on the demo blaze website.
     + Output :
       - 1) To find the breakpoint of the application
       - 2) To check if application under stress can be scaled either horizontally or vertically
  3. Spike Testing : This is to simulate a sudden spike or peak load on the application for APIs like add to cart and checkout. This will be userful when we have any mega sale event like 10.10 or 11.11 sale when the users will suddenly rush to the website to buy at discounted price
     + Output :
       - 1) To check that the application can handle a sudden burst in traffic
       - 2) To check if we need to scale up the infra and deployment before the sale
       - 3) To check that the response time is not impacted
       - 4) To check the failure rate is <2% and user experience is not dropped
  4. Soak Testing : This to run a performance testing on demo blaze for >24 hours to 48 hours.
     + Output:
       - 1) To Determine if the application have any memory leaks or dead logs over long run
       - 2) To check the response time for critical APIs remain same over period of longer time
       - 3) To check there is no sudden crash or application failures
  5. Performance Test result analysis :
     + 1) To Monitor and analyze the test results on
       - Jmeter :
         * 1) Check hits per seconds is consistent
         * 2) Response codes per second is consistent and 200 OK
         * 3) Check response time over time is consistent
       - Server side :
         * 1) Check CPU and memory utilization is <80%
         * 2) No exceptions, errors or warnings In server logs
         * 3) No crash and consistent healthy running application
         * 4) CPU and memory utilization should go down to base values after performance test is stopped
     + Create and document the details in a report in confluence along with release version of each micro service components.

**3. Security Testing:**

* Create and document Security test cases to include:
  + **Input Validation Testing**: Test input fields (such as forms, search bars, and URL parameters) to ensure they properly sanitize user input. This helps prevent common attacks like SQL injection and cross-site scripting (XSS).
  + **Authentication Testing**: Verify that the authentication mechanism securely handles user credentials (e.g., passwords) and prevents common vulnerabilities like brute force attacks, session fixation, and account enumeration.
  + **Authorization Testing**: Test whether users are only able to access resources and perform actions that they are authorized to do. Ensure that privilege escalation and insecure direct object references are prevented.
  + **Session Management Testing**: Check how session tokens are generated, transmitted, and stored. Test for session fixation, session hijacking, and session timeout vulnerabilities.
  + **Sensitive Data Exposure Testing**: Ensure that sensitive data such as credit card information, passwords, and personal details are encrypted both in transit (using HTTPS) and at rest (in databases).
  + **Payment Gateway Security Testing**: Verify the security of payment processing systems and gateways to prevent attacks like credit card skimming, man-in-the-middle (MITM) attacks, and payment fraud.
  + **Cross-Site Request Forgery (CSRF) Testing**: Test for CSRF vulnerabilities by crafting malicious requests that execute unauthorized actions on behalf of authenticated users. Verify that proper anti-CSRF tokens are implemented.
  + **Security Headers Testing**: Check if security headers such as Content Security Policy (CSP), Strict Transport Security (HSTS), and X-Content-Type-Options are correctly configured to mitigate various web-based attacks.
  + **Third-Party Integration Testing**: Assess the security of third-party libraries, plugins, and APIs used by the ecommerce platform. Verify that they are up-to-date, properly configured, and do not introduce security vulnerabilities.
  + **Security misconfiguration** : Check if any API keys are exposed over network response for configs.
  + **Denial of service attacks** : Make sure that APIs have rate limiting implemented

**4. Usability Testing:**

* UI/UX consistency: Ensure a consistent and intuitive user experience across different devices and browsers. (2 days)
* Mobile responsiveness: Test the website's responsiveness on various mobile devices. (2 days)

**5. Accessibility Testing:**

1. **Create and document Accessibility Tests which includes :**
   1. **Keyboard Navigation Testing**: Verify that users can navigate through the website using only the keyboard, without relying on a mouse. Test tab order, focus indicators, and keyboard shortcuts.
   2. **Screen Reader Compatibility Testing**: Test the website with popular screen reader software (such as JAWS, NVDA, or VoiceOver) to ensure that users with visual impairments can access and understand the content.
   3. **Color Contrast Testing**: Check the color contrast ratios between text and background colors to ensure readability, particularly for users with low vision or color blindness. Tools like WebAIM's Color Contrast Checker can help with this.
   4. **Semantic HTML Testing**: Ensure that HTML elements are used semantically and appropriately (e.g., using headings, lists, labels) to provide context and structure for screen readers and other assistive technologies.
   5. **Alternative Text (Alt Text) Testing**: Verify that all images and non-text content have descriptive and meaningful alternative text (alt text) to provide information to users who cannot see the images.
   6. **Form Field Labels and Errors Testing**: Check that form fields have associated labels and error messages are descriptive and clearly associated with the corresponding form fields. This helps users understand input requirements and errors.
   7. **Focus Management Testing**: Test focus management to ensure that keyboard users can easily identify and navigate through interactive elements. Ensure that focus styles are visible and consistent.
   8. **Resizing and Zooming Testing**: Verify that the website's layout remains usable and content remains readable when the user zooms in or changes text size. Test responsiveness to different viewport sizes.
   9. **Skip Navigation Link Testing**: Ensure that a "skip to main content" link is available and functional, allowing users to bypass repetitive navigation elements and access the main content directly.
   10. **Accessible Rich Internet Applications (ARIA) Testing**: Verify that ARIA attributes are correctly implemented to enhance accessibility for interactive elements such as sliders, tabs, and menus.

**Prioritization:**

1. Functional Testing: Critical for ensuring the core functionality of the website works as expected.
2. Security Testing: Protecting user data and ensuring a secure environment is crucial.
3. Usability Testing: A user-friendly interface enhances user satisfaction and retention.
4. Performance Testing: Ensuring optimal performance prevents user frustration and improves conversion rates.

Keep in mind that these estimates are approximate and can vary depending on the complexity of the website and the testing environment. Additionally, it's essential to iterate on testing tasks continuously and adjust priorities based on feedback and emerging issues.