## Test Management Process and Tool Recommendations

### Test Management Process

1. **Requirement Analysis**: Understand website features and create a traceability matrix to map test cases to requirements.
2. **Test Planning**: Define scope, objectives, and resources in a test plan.
3. **Test Case Design**: Write clear, concise test cases covering functional, usability, compatibility, and performance aspects.
4. **Test Execution**: Perform manual testing for UI/UX and exploratory testing; use automation for repetitive tasks (e.g., login, form submission).
5. **Defect Management**: Log defects in GitHub Issues, prioritize based on severity, and track resolution.
6. **Reporting**: Generate reports with pass/fail metrics, defect status, and recommendations.
7. **Review and Retest**: Validate bug fixes and update test cases as needed.

### Recommended Tools

Based on research and experience, the following tools are suitable for managing the testing process:

* **TestRail**: A comprehensive test management tool for creating test plans, cases, and reports. It integrates with GitHub for defect tracking and supports manual and automated testing. Ideal for structured testing processes.
* **TestQuality**: Extends GitHub workflows with test case management, offering live two-way integration with GitHub Issues. Suitable for teams already using GitHub for development.
* **Testmo**: Supports manual, automated, and exploratory testing with GitHub Actions integration. Good for visualizing test results and tracking metrics.
* **TestLink**: An open-source tool for managing test cases and plans. Suitable for small teams with budget constraints, though it lacks advanced integrations compared to TestRail.
* **Excel/Google Sheets**: For small projects, test cases and reports can be managed in spreadsheets. Simple but lacks automation and scalability.

### Why These Tools?

* **TestRail**: Intuitive UI, robust reporting, and integrations make it ideal for structured testing.
* **TestQuality/Testmo**: Seamless GitHub integration suits teams using GitHub for version control and issue tracking.
* **TestLink**: Free and open-source, good for beginners or small teams.
* **Excel**: Familiar and accessible, but manual effort is high for large projects.

### Recommended Approach

* Use **TestRail** for its comprehensive features and ease of use, especially for managing test cases and generating reports.
* Integrate with **GitHub Issues** for defect tracking to keep testing and development workflows aligned.
* Use **Selenium** for automating repetitive functional tests (e.g., login, registration).
* Leverage **Google Lighthouse** for basic performance testing.
* Store test artifacts (plan, cases, reports) in a GitHub repository for version control and collaboration.