|  |  |
| --- | --- |
| **Task** | **Description** |
| Markup Language Selection & Testing Approach | Markup language selection and testing involve choosing the right markup language for a project and rigorously testing it. First, analyse project requirements and research available markup languages. Criteria for selection include compatibility, features, ease of use, and standards compliance. Prototyping helps evaluate suitability. Testing includes syntax validation, functional, cross-browser/platform, accessibility, performance, and security testing. Integration testing ensures seamless interaction with other components. Documentation captures syntax, features, and best practices. The process is iterative, allowing for continuous improvement. By following this approach, you ensure the chosen markup language meets project needs, adheres to standards, and performs reliably across various platforms. |
| Markup Language Choice | HTML5 |
| Reasoning | HTML5 is compatible with various browsers and devices, supports multimedia elements, provides semantic structure for SEO, and integrates well with CSS and JavaScript. |
| Testing Approach |  |
| 1. Functionality Testing | - Ensure links, buttons, forms, and interactive elements work correctly.<br>- Validate HTML markup with W3C Markup Validation Service.<br>- Test multimedia elements for proper display and functionality across browsers. |
| 2. Usability Testing | - Evaluate UI for intuitiveness and navigation.<br>- Conduct user testing sessions for feedback.<br>- Ensure design consistency across pages. |
| 3. Compatibility Testing | - Test website across various browsers and devices for consistent rendering and functionality.<br>- Verify compatibility with different screen resolutions and older browser versions.<br> |
| 4. Security Testing | - Implement measures against common vulnerabilities.<br>- Use HTTPS for secure data transmission.<br>- Regularly update libraries for security patches. |
| 5. Performance Testing | - Evaluate loading speed and performance with tools like Google PageSpeed Insights.<br>- Optimize resources for faster load times.<br>- Test responsiveness, especially on mobile devices. |
| 6. Database Testing (Assuming database integration) | - Test database connectivity and data retrieval.<br>- Validate data integrity and accuracy.<br>- Perform stress testing for concurrent user requests. |
| Browser Testing Approach |  |
| Test Cases |  |
| 1. Homepage Display | - Verify correct loading and layout on different browsers.<br>- Ensure content elements render properly. |
| 2. Navigation Testing | - Test navigation links and menus for functionality.<br>- Verify correct page redirection. |
| 3. Form Functionality | - Test form submission and validation across browsers.<br>- Ensure proper error handling. |
| 4. Media Playback | - Test audio and video playback across browsers.<br>- Ensure compatibility and error-free loading. |
| 5. Responsive Design Testing | - Test responsiveness by resizing browser windows and using developer tools to simulate devices.<br>- Ensure adaptability to various screen sizes. |
| Testing Checklist |  |
| - Browser Compatibility | Test on latest versions of Chrome, Firefox, Safari, Edge, and Internet Explorer (if applicable). |
| - Mobile Compatibility | Verify compatibility with mobile browsers (Chrome Mobile, Safari on iOS, etc.). |
| - Operating System Compatibility | Cross-check appearance and functionality on different operating systems (Windows, macOS, iOS, Android). |
| - Issue Documentation | Document encountered issues and prioritize based on severity. |

|  |  |
| --- | --- |
| **Accessibility Testing** | **Description** |
| Keyboard Navigation Testing | Ensure all interactive elements can be accessed and used via keyboard navigation alone. |
|  | Test tab order and focus visibility. |
| Screen Reader Compatibility | Verify content and interactive elements are properly read by screen readers (e.g., VoiceOver, NVDA). |
|  | Ensure meaningful alternative text for images and other non-text content. |
|  | Check ARIA (Accessible Rich Internet Applications) roles and attributes for enhanced accessibility. |
| Color Contrast Testing | Evaluate color combinations for sufficient contrast to accommodate users with visual impairments. |
|  | Ensure text is readable against background colors. |
| Semantic HTML Structure | Confirm the use of semantic HTML elements (e.g., <nav>, <header>, <main>, <footer>) for better screen reader comprehension and navigation. |
| Form Accessibility | Test form fields with screen readers to ensure labels, instructions, and error messages are properly conveyed. |
|  | Verify input fields have associated labels or accessible names. |
| Focus Management | Ensure focus indicators are clearly visible and distinguishable, especially for users navigating with keyboard-only. |
| Text Resize Testing | Test content readability at various text sizes to ensure scalability without loss of functionality or readability. |
| Assistive Technology Interaction | Test interactive components (e.g., dropdown menus, accordions) with assistive technologies to ensure proper functionality and feedback. |