



ACCESSIBILITY TRAINING

Version 1.1

Updated By: Bradley Harrison
On the Date: 10th April 2023

Stackle

Table of Contents

Table of Contents	2
1. Introduction	9
2. Training Overview	10
2.1.1. Introduction to Web Accessibility	10
2.1.2. Inclusive Design Principles	10
2.1.3. Understanding Assistive Technologies	10
2.1.4. Common Accessibility Barriers	10
2.1.5. Accessible Coding Practices	11
2.1.6. Testing and Evaluation	11
2.1.7. Ongoing Learning and Resources	11
3. Introduction to Web Accessibility	11
3.1. Importance of Web Accessibility	11
3.2. Moral and Legal Obligations	12
3.3. Benefits of Inclusive Digital Products	12
3.4. Web Content Accessibility Guidelines (WCAG)	12
3.5. Conformance Levels (A, AA, and AAA)	12
4. Inclusive Design Principles	13
4.1. Diverse User Needs	13
4.1.1. Different Abilities	14
4.1.2. Age Groups	14
4.1.3. Cultural Backgrounds	14
4.1.4. Learning Styles	14
4.1.5. Cognitive Abilities and Literacy Levels	15
4.1.6. Device and Platform Compatibility	15
4.1.7. Customization Options	15
4.1.8. Section Summary	15
4.2. Multiple Input Methods	15
4.2.1. Keyboard Navigation	16
4.2.2. Touchscreen Compatibility	16
4.2.3. Voice Control	16
4.2.4. Assistive Technologies	16
4.2.5. Adaptive Input Methods	17
4.3. Clear and Consistent Navigation	17
4.3.1. Descriptive Headings	17
4.3.2. Logical Menu Structures	17
4.3.3. Consistent Layout Patterns	18
4.3.4. Breadcrumbs	18
4.3.5. Accessible Navigation Elements	18
4.3.6. Skip Navigation Links	18
4.3.7. Clear and Meaningful Link Text	19

4.4. Flexibility and Customization	19
4.4.1. Adjustable Text Size:	19
4.4.2. Alternative Color Schemes	19
4.4.3. Content Display Options	20
4.4.4. Keyboard Navigation and Shortcuts	20
4.4.5. Scalable Interface	20
4.4.6. Customizable Audio Controls	20
4.4.7. Personalization Features	21
4.5. Perceivable Information	21
4.5.1. Text Alternatives for Non-Text Content	21
4.5.2. Captions and Transcripts for Multimedia Content	21
4.5.3. Color and Contrast	22
4.5.4. Readability	22
4.5.5. Flexible Layouts and Scalable Content	22
4.5.6. Clear Audio Signals	22
4.6. Robust and Compatible Content	23
4.6.1. Semantic Markup	23
4.6.2. ARIA Attributes	23
4.6.3. Coding Best Practices	23
4.6.4. Responsive Design	24
4.6.5. Keyboard Accessibility	24
4.6.6. Cross-Browser Compatibility	24
4.6.7. Assistive Technology Compatibility	24
4.7. User Control and Error Prevention	25
4.7.1. Customization Options	25
4.7.2. Clear Instructions	25
4.7.3. Multiple Ways to Complete Tasks	25
4.7.4. Error Prevention Techniques	26
4.7.5. Error Recovery	26
4.7.6. Consistent Interface Design	26
4.7.7. User Testing and Feedback	26
4.8. Module Summary	27
4.9. Module Assessment	27
5. Understanding Assistive Technologies	27
5.1. Types of Assistive Technologies	27
5.1.1. Screen Readers	27
5.1.2. Screen Magnifiers	28
5.1.3. Voice Input Software	28
5.1.4. Alternative Input Devices	28
5.1.5. Text-to-Speech (TTS) Software	28
5.1.6. Captioning and Transcription Tools	29
5.1.7. Cognitive Support Technologies	29
5.2. Compatibility Principles	29
5.2.1. Semantic HTML	29

5.2.2. ARIA Attributes	30
5.2.3. Keyboard Navigation	30
5.2.4. Perceivable Content	30
5.2.5. Operable Content	31
5.2.6. Understandable Content	31
5.2.7. Robust Content	31
5.3. Assistive Technology-Friendly Design	32
5.3.1. Color Contrast	32
5.3.2. Font Size and Style	32
5.3.3. Layout	32
5.3.4. Navigational Elements	33
5.3.5. Accessible Multimedia Content	33
5.4. Development Best Practices	33
5.4.1. Proper Markup	33
5.4.2. CSS Techniques	34
5.4.3. Accessible JavaScript	34
5.4.4. Accessible Forms	34
5.4.5. Accessible Tables	35
5.4.6. Interactive Elements	35
5.5. Testing and Evaluation	35
5.5.1. Overview of Accessibility Testing	36
5.5.2. Automated Testing Tools	36
5.5.3. Manual Testing Methods	36
5.5.4. Inclusive User Testing	36
5.5.5. Identifying and Addressing Accessibility Issues	36
5.5.6. Testing and Evaluation Best Practices	37
5.6. Keeping up with Emerging Technologies:	37
5.6.1. Industry News and Publications	37
5.6.2. Conferences and Workshops	38
5.6.3. Webinars and Online Courses	38
5.6.4. Social Media and Online Communities	38
5.6.5. Collaboration with Assistive Technology Users	38
5.6.6. Industry Partnerships	38
5.6.7. Internal Knowledge Sharing	39
5.7. Module Summary	39
6. Common Accessibility Barriers	39
6.1. Proper Use of Headings and Landmarks	40
6.1.1. Heading Levels (H1-H6)	40
6.1.2. ARIA Landmarks:	40
6.2. Creating Meaningful Link Text	41
6.2.1. Be Descriptive and Specific	42
6.2.2. Provide Context	42
6.2.3. Keep It Concise	42
6.2.4. Use Actionable Language	42

6.2.5. Avoid Ambiguous Phrases	43
6.2.6. Distinguish Between Different Links	43
6.2.7. Consider the Link's Surrounding Text	43
6.3. Providing Alternative Text for Images	43
6.3.1. Importance of Alt Text	43
6.3.2. Context	44
6.3.3. Brevity	44
6.3.4. Relevance	44
6.3.5. Clarity	44
6.3.6. Functionality	45
6.3.7. Decorative Images	45
6.4. Ensuring Sufficient Color Contrast	45
6.4.1. Understanding Color Contrast	45
6.4.2. WCAG Guidelines for Minimum Contrast Ratios	46
6.4.3. Selecting Accessible Color Combinations:	46
6.4.4. Testing Color Contrast:	46
6.5. Designing Accessible Forms	47
6.5.1. Proper Use of Labels	47
6.5.2. Fieldsets and Legends	47
6.5.3. Clear Instructions and Error Messages	47
6.5.4. Keyboard Accessibility	48
6.5.5. Logical Tab Order	48
6.5.6. Input Assistance	48
6.5.7. Accessible Controls	48
6.5.8. Adequate Click Targets	48
6.5.9. Time Limits	49
6.6. Module Summary	49
7. Accessible Coding Practices	49
7.1. Semantic HTML	49
7.1.1. Headings	50
7.1.2. Lists	50
7.1.3. Tables	50
7.1.4. Forms	50
7.1.5. ARIA (Accessible Rich Internet Applications)	50
7.1.6. Navigation	51
7.1.7. Multimedia	51
7.2. ARIA Attributes	51
7.2.1. ARIA Roles	51
7.2.2. ARIA States and Properties	52
7.2.3. Enhancing Complex UI Components	52
7.2.4. Best Practices	52
7.3. Keyboard Accessibility	53
7.3.1. Interactive Element Identification	53
7.3.2. Focus Management	53

7.3.3. Keyboard Event Handling	54
7.3.4. Visual Focus Indicators	54
7.3.5. Skip Navigation Links	54
7.3.6. Accessible Keyboard Shortcuts	54
7.3.7. Testing and Validation	55
7.4. Accessible CSS	55
7.4.1. Proper Color Contrast	55
7.4.2. Font Sizing and Line Spacing	55
7.4.3. Responsive Design	56
7.4.4. Media Queries for User Preferences	56
7.4.5. Focus Indicators and Visual States	56
7.5. JavaScript and Dynamic Content	57
7.5.1. Announcing Content Updates	57
7.5.2. Managing Focus	57
7.5.3. Handling AJAX Requests	57
7.5.4. Accessible Form Submissions:	57
7.5.5. Keyboard Accessibility	58
7.5.6. Responsiveness and Adaptability	58
7.6. Accessible Forms	58
7.6.1. Proper Labeling of Form Controls	59
7.6.2. Grouping Related Controls	59
7.6.3. Providing Helpful Error Messages	59
7.6.4. Implementing Accessible Form Validation	59
7.6.5. Keyboard Accessibility	60
7.6.6. Accessible Instructions and Help Text	60
7.7. Accessible Multimedia	60
7.7.1. Alternative Text for Images	60
7.7.2. Transcripts for Audio Content	61
7.7.3. Captions for Video Content	61
7.7.4. Audio Descriptions for Video Content	61
7.8. Testing and Debugging	62
7.8.1. Introduction to Accessibility Testing	62
7.8.2. Automated Testing Tools	62
7.8.3. Manual Testing Techniques	62
7.8.4. Screen Reader Testing	63
7.8.5. Debugging Accessible Code	63
7.8.6. Integrating Testing into the Development Lifecycle	63
7.8.7. Documenting and Communicating Test Results	63
7.9. Module Summary	64
8. Testing and Evaluation:	64
8.1. Overview of Accessibility Testing	64
8.1.2. Types of Accessibility Testing	65
8.1.3. Automated Testing	65
8.1.4. Manual Testing	65

8.1.5. User Testing	66
8.1.6. Assistive Technology Testing	66
8.2. Automated Testing Tools	66
8.2.1. Lighthouse	66
8.2.2. WAVE (Web Accessibility Evaluation Tool)	67
8.2.3. Best Practices for Using Automated Testing Tools:	67
8.3. Manual Testing Techniques	68
8.3.1. Keyboard Testing	68
8.3.2. Focus Management	68
8.3.3. Screen Reader Testing	68
8.3.4. Colour Contrast and Visibility	69
8.3.5. Zoom and Magnification Testing	69
8.3.6. Assistive Technology Compatibility	69
8.4. Screen Reader Testing	69
8.4.1. Hands-on Training	70
8.4.2. Navigation Techniques	70
8.4.3. Interacting with Elements	70
8.4.4. Identifying Issues	70
8.4.5. Collaboration and Remediation	71
8.4.6. Continuous Improvement	71
8.5. Testing with Assistive Technologies	71
8.5.1. Screen Magnifiers	71
8.5.2. Screen Readers	72
8.5.3. Speech Recognition Software	72
8.5.4. Alternative Input Devices	72
8.5.5. Testing Scenarios and Techniques	72
8.5.6. Identifying Accessibility Barriers	73
8.5.7. Documentation and Reporting	73
8.6. User Testing	73
8.6.1. Participant Recruitment	73
8.6.2. Inclusive Test Environment	74
8.6.3. Test Planning	74
8.6.4. Conducting User Testing	74
8.6.5. Evaluation of Test Results	74
8.6.6. Implementing Feedback	75
8.6.7. Continuous Improvement	75
8.7. Reporting and Remediation	75
8.7.1. Identifying Accessibility Issues	75
8.7.2. Documenting Accessibility Issues	76
8.7.3. Reporting Accessibility Issues	76
8.7.4. Prioritizing Accessibility Issues	76
8.7.5. Developing Remediation Plans	76
8.7.6. Implementing Remediation Plans	76
8.7.7. Reviewing and Validating Remediation Efforts	77

8.8. Module Summary	77
9. Ongoing Learning and Resources	77
9.1. Online Courses and Webinars	78
9.2. Blogs and Newsletters	78
9.3. Conferences and Workshops	78
9.4. Accessibility Guidelines and Standards	78
9.5. Community Involvement	78
9.6. Internal Knowledge Sharing	79
9.7. Accessibility Tools and Assistive Technologies	79

1. Introduction

At Stackle, we understand the importance of equipping our development team members with the knowledge and skills necessary to create accessible digital products. To achieve this, we have designed a comprehensive accessibility training program that covers essential topics and best practices. The training is mandatory for all development team members and is regularly updated to stay current with industry standards and advancements.

2. Training Overview

The accessibility training program at Stackle is structured to cover various aspects of accessibility, including inclusive design, assistive technologies, and common accessibility barriers. The curriculum is divided into the following key modules:

2.1.1. Introduction to Web Accessibility

This module provides an overview of the importance of web accessibility, the moral and legal obligations, and the benefits of creating inclusive digital products. It introduces the Web Content Accessibility Guidelines (WCAG) and explains the different conformance levels (A, AA, and AAA).

2.1.2. Inclusive Design Principles

This module focuses on the principles of inclusive design, which helps ensure that our digital products cater to a diverse range of users, including those with disabilities. Topics covered include designing for various input methods, creating clear and consistent navigation, and designing with flexibility in mind to accommodate different user preferences and assistive technologies.

2.1.3. Understanding Assistive Technologies

In this module, participants learn about various assistive technologies used by people with disabilities to access digital content, such as screen readers, magnifiers, and voice input software. The training covers how to design and develop digital products that are compatible with these technologies, as well as best practices for testing compatibility.

2.1.4. Common Accessibility Barriers

This module explores common accessibility barriers that users may encounter and provides guidance on how to address them. Topics covered include proper use of headings and landmarks, creating meaningful link text, providing alternative text for images, ensuring sufficient color contrast, and designing accessible forms.

2.1.5. Accessible Coding Practices

In this module, participants learn how to write accessible HTML, CSS, and JavaScript code. Topics include the use of semantic HTML, proper ARIA attributes, keyboard accessibility, and techniques for creating accessible, dynamic content using JavaScript.

2.1.6. Testing and Evaluation

This module covers accessibility testing methods and tools, including both automated and manual testing. Participants learn how to use tools like Lighthouse and WAVE for automated testing, as well as techniques for manual testing, such as keyboard testing and using screen readers.

2.1.7. Ongoing Learning and Resources

The training concludes with a list of resources for further learning and staying up-to-date with accessibility advancements. Participants are encouraged to continuously expand their knowledge and skills in web accessibility.

By providing comprehensive accessibility training to our development team members, Stackle ensures that they have the necessary skills and knowledge to create digital products that cater to the needs of all users, including those with disabilities.

3. Introduction to Web Accessibility

3.1. Importance of Web Accessibility

Web accessibility is the practice of ensuring that websites, applications, and digital content are easily accessible and usable by all people, regardless of their abilities or disabilities. Accessible web design allows individuals with diverse abilities to perceive, understand, navigate, and interact with digital content, enabling equal access to information and services.

3.2. Moral and Legal Obligations

In addition to being a moral responsibility, web accessibility is often a legal requirement. Many countries have enacted laws and regulations mandating accessible websites and digital content, such as the Americans with Disabilities Act (ADA) in the United States and the Disability Discrimination Act (DDA) in Australia. These laws aim to protect the rights of people with disabilities and ensure equal access to information, products, and services.

3.3. Benefits of Inclusive Digital Products

Creating accessible digital products offers numerous benefits for businesses, organizations, and users alike. Accessible design leads to improved user experiences, increased audience reach, and enhanced search engine optimization (SEO). It also fosters a positive brand reputation, demonstrating a commitment to social responsibility and inclusivity.

3.4. Web Content Accessibility Guidelines (WCAG)

The Web Content Accessibility Guidelines (WCAG) are a set of internationally recognized standards developed by the World Wide Web Consortium (W3C) to provide guidance on creating accessible web content. WCAG is organized around four key principles known as POUR: Perceivable, Operable, Understandable, and Robust. These principles ensure that digital content can be accessed and used by people with various disabilities, including visual, auditory, cognitive, and motor impairments.

3.5. Conformance Levels (A, AA, and AAA)

WCAG defines three conformance levels (A, AA, and AAA) to help organizations meet different degrees of accessibility:

- Level A

The most basic level of accessibility. Compliance with Level A guidelines helps to address the most critical accessibility barriers,

but it may not be sufficient for providing full accessibility to users with disabilities.

- Level AA

A higher level of accessibility, addressing a broader range of accessibility issues. Compliance with Level AA guidelines is generally considered the standard for most organizations and is often the target level required by accessibility laws and regulations.

- Level AAA

The highest level of accessibility, which goes beyond the basic and intermediate requirements. Compliance with Level AAA guidelines can be challenging for some types of content and is not always required or achievable for all organizations.

In conclusion, the Introduction to Web Accessibility module helps participants understand the significance of web accessibility, the moral and legal obligations associated with it, and the benefits of creating inclusive digital products. It also provides an overview of the Web Content Accessibility Guidelines (WCAG) and explains the different conformance levels to help organizations prioritize their accessibility efforts.

4. Inclusive Design Principles

Inclusive design aims to create products and services that cater to the diverse needs of all users, including those with disabilities. By embracing these principles, Stackle ensures that our digital products offer an accessible and user-friendly experience for everyone, regardless of their abilities or preferences.

The key inclusive design principles covered in this module include:

4.1. Diverse User Needs

Designers at Stackle must consider the diverse needs and preferences of users to create products that are adaptable and accommodating for a wide range of users. By recognizing and addressing the unique

requirements of different individuals, we can build a more inclusive and accessible learning environment for everyone. The following aspects are taken into account to ensure our products cater to diverse user needs:

4.1.1. Different Abilities

Our designers consider users with various physical, cognitive, and sensory abilities, ensuring that our products are compatible with assistive technologies, such as screen readers, alternative input devices, and magnification tools. This includes providing keyboard accessibility, appropriate contrast ratios, and captions for multimedia content.

4.1.2. Age Groups

Our products are designed to be user-friendly for individuals across all age groups, from young children to older adults. We pay attention to factors such as readability, font size, and user interface elements to create a seamless experience that is easy to navigate and understand for all users.

4.1.3. Cultural Backgrounds

We strive to create culturally sensitive and inclusive products by considering the diverse backgrounds of our users. This includes offering language options, ensuring that images and content are appropriate and respectful, and accommodating for variations in cultural norms and practices.

4.1.4. Learning Styles

Our designers take into account the different learning styles of users, such as visual, auditory, and kinesthetic learners. We incorporate various content formats, such as text, images, videos, and interactive elements, to support multiple learning preferences and facilitate a more effective and engaging learning experience.

4.1.5. Cognitive Abilities and Literacy Levels

We design our products to be accessible to users with varying cognitive abilities and literacy levels. By using clear and concise language, providing alternative text for images, and offering adjustable reading levels, we can make our content more approachable and comprehensible for all users.

4.1.6. Device and Platform Compatibility

To cater to diverse user needs, our designers ensure that our products function effectively across various devices and platforms, such as desktop computers, laptops, tablets, and smartphones. This approach allows users to access our content and tools regardless of their preferred device or operating system.

4.1.7. Customization Options

We provide customization options that allow users to tailor their experience to their individual preferences and needs. These options may include adjustable font sizes, color schemes, and interface layouts, enabling users to create a comfortable and personalized learning environment.

4.1.8. Section Summary

By considering and addressing the diverse needs and preferences of users, Stackle's designers create products that are adaptable, accommodating, and accessible to a wide range of users, ensuring an inclusive and engaging learning experience for all.

4.2. Multiple Input Methods

Inclusive design aims to create digital products and services that can be used by people with diverse abilities, preferences, and circumstances. One key aspect of inclusive design is accommodating multiple input methods, which allows users to interact with the product in a way that best suits their needs. By considering various input methods, designers can create more accessible and user-friendly experiences.

4.2.1. Keyboard Navigation

Some users, including those with motor impairments or visual impairments, rely on keyboard navigation rather than a mouse or touchpad. Designers should ensure that all interactive elements, such as buttons, links, and form fields, are accessible using the keyboard alone. This includes providing visible focus indicators, logical tab order, and keyboard shortcuts where appropriate.

4.2.2. Touchscreen Compatibility

Touchscreen devices, such as smartphones and tablets, are increasingly common. Inclusive design should consider users who primarily interact with digital products through touch input. Designers should create touch-friendly interfaces with larger, easily tappable targets and sufficient spacing between elements to reduce the risk of accidental taps. Additionally, they should ensure that touch gestures, such as swiping and pinching, are supported and accessible.

4.2.3. Voice Control

Voice control is an essential input method for users with motor impairments or those who prefer hands-free interaction. Designers should ensure that their products are compatible with voice control technologies, such as speech recognition software and virtual assistants. This may include providing clear and concise labels for interactive elements, supporting voice commands, and offering appropriate feedback to confirm user actions.

4.2.4. Assistive Technologies

Users with disabilities may rely on various assistive technologies to interact with digital products. These include screen readers, screen magnifiers, eye-tracking systems, and sip-and-puff devices. Inclusive design should consider the compatibility of products with these technologies by following accessibility guidelines such as

WCAG and incorporating semantic markup, proper use of ARIA attributes, and accessible JavaScript techniques.

4.2.5. Adaptive Input Methods

Some users may rely on adaptive input methods, such as alternative keyboards or switches, to interact with digital products. Designers should ensure their products are compatible with these devices by providing flexible input options and accommodating different input speeds and response times.

By considering multiple input methods in the design process, designers can create more inclusive and accessible products that cater to a broader range of users. This approach ensures that people with diverse abilities and preferences can effectively interact with digital products and services, promoting a more equitable and user-friendly experience for all.

4.3. Clear and Consistent Navigation

Clear and consistent navigation is an essential aspect of inclusive design, ensuring that all users can efficiently and effectively navigate digital products. By providing a predictable and easily understandable structure, users can find their way around the product with greater ease, ultimately enhancing the user experience for everyone, including those with disabilities. The following are some key elements to consider when implementing clear and consistent navigation

4.3.1. Descriptive Headings

Headings should be descriptive, concise, and meaningful, providing an accurate summary of the content in the section they represent. Utilizing appropriate heading hierarchy (H1, H2, H3, etc.) helps users, including those using screen readers, to understand the structure of the content and navigate between sections more easily.

4.3.2. Logical Menu Structures

A well-organized menu structure with logically grouped items enables users to quickly locate the information or features they are seeking. Menus should be arranged in a hierarchical manner, with

clear labels and intuitive categorization. Submenus should be used judiciously to avoid excessive complexity and to maintain a straightforward navigation experience.

4.3.3. Consistent Layout Patterns

Consistent layout patterns across the digital product help users build familiarity with the interface and reduce cognitive load. This includes maintaining consistent placement of navigation elements (such as menus, search bars, and buttons), as well as applying uniform styling and formatting to ensure visual coherence.

4.3.4. Breadcrumbs

Breadcrumbs provide a visual representation of the user's location within the product's content hierarchy. They offer an additional means of navigation, allowing users to quickly return to previous pages or sections and understand their current position in the overall structure.

4.3.5. Accessible Navigation Elements

Navigation elements, such as menus, buttons, and links, should be designed to be accessible for all users, including those using keyboard navigation or assistive technologies. This includes providing visible focus indicators, proper use of ARIA attributes, and ensuring that navigation elements are easily discoverable and operable.

4.3.6. Skip Navigation Links

Skip navigation links allow users, particularly those using screen readers or keyboard navigation, to bypass repetitive content (e.g., headers and menus) and quickly access the main content of a page. This improves the efficiency of navigation and overall usability of the digital product.

4.3.7. Clear and Meaningful Link Text

Links should have clear and meaningful text that accurately describes the destination or action associated with the link. Ambiguous or generic link text (e.g., "click here" or "read more") should be avoided, as it hinders users' understanding of the link's purpose and can be particularly problematic for screen reader users.

By incorporating these elements into the design and development of digital products, Stackle ensures that clear and consistent navigation is provided, ultimately creating a more inclusive and accessible user experience.

4.4. Flexibility and Customization

Inclusive design is focused on creating user experiences that can be easily adapted to individual needs and preferences, ensuring that products and services cater to a wide range of users, including those with disabilities. Flexibility and customization are key components of inclusive design, as they enable users to tailor their experience to suit their unique requirements and interact with products using their preferred assistive technologies. Here, we explore various aspects of flexibility and customization in more detail.

4.4.1. Adjustable Text Size:

Allowing users to adjust text size is essential for individuals with visual impairments or reading difficulties. By providing an easy-to-use interface for changing the text size, users can customize the display to meet their specific needs, enhancing readability and comprehension.

4.4.2. Alternative Color Schemes

Offering alternative color schemes can greatly improve the user experience for individuals with color vision deficiencies or specific contrast requirements. Providing options for high contrast, dark mode, or user-defined color schemes allows users to select the most

comfortable visual setting, making it easier to navigate and engage with content.

4.4.3. Content Display Options

Different users may have different preferences for how content is displayed, depending on their needs and the assistive technologies they use. By providing options for displaying content in alternative formats (e.g., text-only versions, transcripts for audio content, or simplified layouts), you can ensure that users can access information in a format that works best for them.

4.4.4. Keyboard Navigation and Shortcuts

Some users rely on keyboard navigation or prefer to use keyboard shortcuts to interact with digital products. Ensuring that all features and functions are accessible via keyboard input is crucial for inclusive design. Providing customizable keyboard shortcuts can further enhance the user experience, allowing individuals to tailor their interactions to their specific preferences.

4.4.5. Scalable Interface

A scalable interface enables users to resize the entire layout of a website or application, not just the text. This can be particularly helpful for users with low vision or those using screen magnification software, as it allows them to adjust the display to a comfortable size without losing context or functionality.

4.4.6. Customizable Audio Controls

For users who rely on screen readers or other auditory assistive technologies, customizable audio controls are essential. Offering options to adjust the volume, speed, and pitch of audio content, as well as enabling users to select their preferred voice for text-to-speech, can greatly enhance the user experience.

4.4.7. Personalization Features

Allowing users to save their accessibility preferences can make it easier for them to interact with a digital product consistently. By providing user accounts or other personalization features, users can save their settings and have them automatically applied when they revisit the site or application.

By incorporating flexibility and customization options into the design and development of digital products, inclusive design ensures that all users can access, understand, and engage with content in a manner that best suits their individual needs and preferences. This, in turn, promotes a more inclusive and accessible user experience for everyone.

4.5. Perceivable Information

The principle of perceivable information is a crucial aspect of creating accessible and inclusive digital products. It ensures that all users, regardless of their sensory abilities, can perceive and understand the information presented. Designers should take the following measures to ensure that all information is perceivable by users with diverse abilities:

4.5.1. Text Alternatives for Non-Text Content

Designers should provide meaningful text alternatives for non-text content, such as images, graphics, and icons. These alternatives, often in the form of alt text, allow screen readers and other assistive technologies to convey the purpose or function of the non-text content to users who cannot see it.

4.5.2. Captions and Transcripts for Multimedia Content

For multimedia content, such as videos and audio recordings, designers should provide captions and transcripts. Captions are synchronized text descriptions of the audio content, including dialogue, sound effects, and other relevant auditory information. Transcripts are text-based versions of the multimedia content, including both spoken words and descriptions of important visual

elements. Providing captions and transcripts ensures that users who are deaf or hard of hearing can access and understand the content.

4.5.3. Color and Contrast

Designers should ensure that color is not the sole means of conveying information, distinguishing visual elements, or indicating an action. This helps users who are color-blind or have difficulty distinguishing certain color combinations. Additionally, designers should use sufficient color contrast between text and its background to ensure readability for users with low vision or those using high-contrast display settings.

4.5.4. Readability

To enhance readability, designers should use appropriate font sizes, line spacing, and paragraph formatting. Text should be resizable without loss of functionality or content. Providing users with options to customize text size and contrast can further improve the perceivability of the content.

4.5.5. Flexible Layouts and Scalable Content

Designers should create flexible layouts that can adapt to different screen sizes and orientations. This ensures that content remains perceivable and usable on various devices, including desktops, laptops, tablets, and smartphones. Additionally, content should be scalable, enabling users to zoom in or out without losing functionality or legibility.

4.5.6. Clear Audio Signals

For users with hearing impairments, designers should ensure that any audio signals, such as notifications or alerts, are clear and distinguishable. Where possible, provide alternative ways to access the information conveyed by the audio signal, such as visual cues or vibration feedback.

By incorporating these strategies, designers can create digital products that are more accessible and inclusive, ensuring that information is perceivable for all users, regardless of their sensory abilities. This approach is in line with the Web Content Accessibility Guidelines (WCAG) and helps create a more user-friendly experience for everyone.

4.6. Robust and Compatible Content

Creating content that is robust and compatible with a wide range of devices, platforms, and assistive technologies is essential for inclusive design. This ensures that our digital products are accessible to users with diverse abilities and preferences, regardless of the technology they rely on. To achieve robust and compatible content, Stackle follows several key practices:

4.6.1. Semantic Markup

Using semantic markup in HTML helps convey the structure and meaning of the content to assistive technologies, such as screen readers. By using appropriate HTML tags for headings, lists, tables, and other elements, we enable these technologies to accurately interpret and present the content to users with disabilities.

4.6.2. ARIA Attributes

Accessible Rich Internet Applications (ARIA) is a specification that provides additional information about the structure, functionality, and state of web content for assistive technologies. By using appropriate ARIA attributes, Stackle ensures that dynamic and interactive elements of our products are accessible to users who rely on assistive technologies.

4.6.3. Coding Best Practices

Following coding best practices is crucial for creating content that works well across different devices, platforms, and assistive technologies. Stackle adheres to well-established standards, such

as WCAG 2.1 Level AA and WAI-ARIA, and employs clean, structured, and valid code to ensure compatibility and ease of maintenance.

4.6.4. Responsive Design

To accommodate various screen sizes and resolutions, Stackle adopts responsive design principles in our digital products. This ensures that our content is visually accessible and functional on different devices, including desktop computers, laptops, tablets, and smartphones.

4.6.5. Keyboard Accessibility

Ensuring that our content is easily navigable and operable through keyboard-only input is vital for users who cannot use a mouse or touchscreen. Stackle ensures that all interactive elements, such as buttons, links, and form fields, are keyboard accessible and provide clear visual focus indicators.

4.6.6. Cross-Browser Compatibility

Stackle tests its digital products across multiple browsers, including Chrome, Firefox, Safari, and Edge, to ensure consistent performance and accessibility. This helps us identify and address any browser-specific issues that could hinder the user experience.

4.6.7. Assistive Technology Compatibility

To ensure compatibility with various assistive technologies, Stackle conducts thorough testing with tools such as screen readers (e.g., JAWS, NVDA, VoiceOver), screen magnifiers, and speech recognition software. This helps us identify and address any potential barriers for users who rely on these technologies.

By focusing on robust and compatible content, Stackle ensures that our digital products are accessible to a wide range of users, regardless of their abilities or the technologies they use. This commitment to inclusive design enables us to provide a seamless and enjoyable learning experience for all.

4.7. User Control and Error Prevention

Inclusive design focuses on providing users with control over their experience, ensuring that they can access and interact with digital products in a manner that suits their individual needs and preferences. By prioritizing user control and implementing error prevention techniques, we can create a more inclusive environment that accommodates a wide range of users, including those with disabilities.

4.7.1. Customization Options

Providing users with customization options, such as adjustable text size, color contrast, and alternative font choices, empowers them to tailor the interface to their specific needs. This can be particularly helpful for users with visual impairments, cognitive disabilities, or learning difficulties.

4.7.2. Clear Instructions

Inclusive design emphasizes the importance of clear, concise instructions for all user interactions. This includes using plain language, providing step-by-step guidance, and offering clear feedback on user actions. These practices can help users with cognitive disabilities or those who may struggle with complex tasks.

4.7.3. Multiple Ways to Complete Tasks

Offering multiple ways for users to complete tasks ensures that they can choose the method that best suits their needs and preferences. This may include providing keyboard navigation, voice commands, or gesture controls in addition to traditional point-and-click functionality. By offering these options, we cater to a wider range of users, including those with motor impairments or who rely on assistive technologies.

4.7.4. Error Prevention Techniques

Implementing error prevention techniques helps users avoid mistakes and reduces the likelihood of frustration or confusion. These techniques can include form validation, offering suggestions or auto-completion, and providing clear error messages that help users identify and resolve issues.

4.7.5. Error Recovery

In the event that a user does encounter an error, it is essential to provide mechanisms for error recovery. This can include allowing users to undo their actions, offering clear guidance on how to correct mistakes, and providing easily accessible help resources.

4.7.6. Consistent Interface Design

Maintaining a consistent interface design throughout the digital product helps users build familiarity and predictability, reducing the likelihood of errors. Consistent design includes using similar navigation elements, consistent button placement, and a cohesive visual style across all pages and components.

4.7.7. User Testing and Feedback

Engaging with users throughout the design and development process helps identify potential barriers and areas for improvement. By incorporating user feedback and conducting usability testing with individuals with diverse abilities, we can refine our designs and ensure that user control and error prevention are effectively implemented.

By prioritizing user control and error prevention, we create a more inclusive and accessible digital experience that caters to the diverse needs and preferences of all users. These design principles help minimize barriers and ensure that our products are usable and enjoyable for everyone, including individuals with disabilities.

4.8. Module Summary

By incorporating these inclusive design principles into our development process, Stackle can create digital products that are accessible, user-friendly, and accommodating for a diverse range of users. This approach not only helps us comply with accessibility guidelines but also ensures a positive user experience for everyone, regardless of their abilities or preferences.

4.9. Module Assessment

4.9.1.

5. Understanding Assistive Technologies

The Understanding Assistive Technologies module aims to educate participants on the wide range of assistive technologies employed by individuals with disabilities to access digital content. By gaining a thorough understanding of these technologies, developers and designers can create digital products that are compatible with and optimized for various accessibility needs.

5.1. Types of Assistive Technologies

Assistive technologies are tools and devices that help individuals with disabilities overcome barriers and facilitate access to digital content and services. They cater to a wide range of needs and support users with various types of disabilities, including visual, auditory, cognitive, and motor impairments. Understanding the different types of assistive technologies can help developers create more accessible and inclusive digital products.

5.1.1. Screen Readers

Screen readers are software applications that convert on-screen text and other visual elements into speech or Braille output. They enable visually impaired users, including those with blindness or low vision, to access digital content. Some popular screen readers include JAWS, NVDA, and VoiceOver.

5.1.2. Screen Magnifiers

Screen magnifiers are tools that enlarge portions of the screen to make content more readable for individuals with low vision. They can magnify text, images, and other screen elements, providing a clearer and more accessible view. Examples of screen magnifiers are ZoomText and the built-in magnifiers in Windows and macOS.

5.1.3. Voice Input Software

Voice input software, also known as speech recognition software, allows users to control their devices and input text using voice commands. This technology is particularly useful for individuals with motor impairments or limited dexterity, as it eliminates the need for a keyboard or mouse. Popular voice input software includes Dragon NaturallySpeaking and the built-in speech recognition tools in modern operating systems.

5.1.4. Alternative Input Devices

Alternative input devices are specialized hardware that enables users with motor impairments to interact with their devices more easily. Examples of these devices include adaptive keyboards, trackballs, sip-and-puff systems, and eye-tracking technology. These tools can help users with limited mobility, dexterity, or muscle control to access digital content without relying on traditional input methods.

5.1.5. Text-to-Speech (TTS) Software

Text-to-speech software converts written text into spoken words, assisting users with cognitive impairments or reading difficulties such as dyslexia. TTS software can read aloud web content, emails, documents, and other text-based materials, making digital content more accessible for users with diverse needs. Examples include Read&Write and NaturalReader.

5.1.6. Captioning and Transcription Tools

Captioning and transcription tools provide a text-based representation of audio content, making multimedia more accessible for deaf or hard-of-hearing users. Closed captions display spoken dialogue and sound effects as on-screen text, while transcription services convert speech into a written format. These tools ensure that auditory content is accessible to users with hearing impairments.

5.1.7. Cognitive Support Technologies

Cognitive support technologies are tools designed to help users with cognitive impairments, such as attention deficit disorders, learning disabilities, or memory problems. Examples include text simplification tools, word prediction software, and visual aids like mind mapping applications. These technologies can assist users in understanding and processing digital content more effectively.

By understanding and considering the various types of assistive technologies, developers can create digital products that are more accessible and inclusive for all users, regardless of their abilities or disabilities.

5.2. Compatibility Principles

Compatibility with assistive technologies is a crucial aspect of creating accessible digital products. This module aims to provide participants with an in-depth understanding of the key principles and best practices for designing and developing digital products that are compatible with various assistive technologies, ensuring an inclusive and accessible experience for all users.

5.2.1. Semantic HTML

Semantic HTML is the use of appropriate HTML tags and elements to convey the structure and meaning of content. Participants will learn about the importance of using semantic markup to improve the

compatibility of digital products with assistive technologies, such as screen readers. Examples of semantic elements include headings (`<h1>` to `<h6>`), lists (`<ul>`, `<ol>`, `<li>`), and navigation (`<nav>`).

5.2.2. ARIA Attributes

ARIA (Accessible Rich Internet Applications) is a set of attributes that can be added to HTML elements to improve the accessibility of dynamic content and user interface components. Participants will learn about the proper use of ARIA attributes, including roles, states, and properties, to enhance the accessibility of complex web applications and provide additional information to assistive technologies.

5.2.3. Keyboard Navigation

Keyboard navigation is essential for users who rely on keyboard input rather than a mouse or touchscreen. Participants will learn how to design and develop digital products that are fully operable using keyboard input alone, ensuring that all interactive elements are reachable and usable through keyboard commands. This includes techniques for managing focus, implementing keyboard shortcuts, and creating custom controls that are accessible via the keyboard.

5.2.4. Perceivable Content

To create perceivable content, participants will learn how to use appropriate text alternatives for non-text content (e.g., images, video, and audio), ensuring that information is available in a format that can be processed by assistive technologies. This module also covers techniques for providing captions, transcripts, and audio descriptions for multimedia content, as well as best practices for creating content that can be easily resized, zoomed, and adapted to different viewing modes without loss of information or functionality.

5.2.5. Operable Content

Operable content is designed to be easily used and navigated by users with diverse abilities. Participants will learn about best practices for creating user interfaces that are operable through various input methods (e.g., keyboard, voice, touch) and designing content that avoids causing seizures or physical reactions.

Additionally, they will learn about techniques for providing clear navigation mechanisms, such as descriptive link text, logical page structure, and consistent navigation menus.

5.2.6. Understandable Content

Understandable content is easy to comprehend for users with diverse cognitive abilities. Participants will learn about techniques for writing clear, concise text and using consistent language, as well as best practices for presenting content in a logical and predictable order. They will also learn about methods for providing context-sensitive help, error identification, and error prevention in form inputs and interactive elements.

5.2.7. Robust Content

Robust content can be reliably interpreted by a wide variety of user agents, including assistive technologies. Participants will learn about techniques for ensuring that content can be parsed accurately by different browsers, devices, and assistive technologies, as well as methods for validating code and implementing progressive enhancement to ensure compatibility with older or less capable devices.

By gaining a deep understanding of these compatibility principles, participants will be better equipped to design and develop digital products that are accessible to all users, regardless of their abilities or the assistive technologies they use.

5.3. Assistive Technology-Friendly Design

This section delves into the various design elements that can significantly improve the user experience for individuals who rely on assistive technologies. By understanding and implementing these design principles, Stackle can create digital products that cater to the needs of all users, including those with disabilities.

5.3.1. Color Contrast

Adequate color contrast is essential for individuals with visual impairments. High contrast between text and background colors can improve readability and ensure that users can easily discern the content. Designers should follow WCAG guidelines for minimum contrast ratios, using tools like contrast checkers to verify compliance.

5.3.2. Font Size and Style

The choice of font size and style can significantly impact the accessibility of digital content. Designers should use easily readable fonts and provide options for users to adjust text size according to their preferences. Additionally, it's crucial to avoid using small fonts or styles that are difficult to read, such as decorative or highly stylized fonts.

5.3.3. Layout

A well-structured and organized layout is essential for users with cognitive disabilities and those using screen readers. Logical and consistent placement of elements, such as headings, links, and buttons, can improve navigation and make it easier for users to understand the content. Proper use of headings, landmarks, and other semantic elements can also help screen reader users navigate the content more efficiently.

5.3.4. Navigational Elements

Accessible navigation is a key aspect of assistive technology-friendly design. This includes providing skip links to bypass repetitive content, consistent and descriptive link text, and keyboard accessibility for all interactive elements. Additionally, designers should ensure that menus, buttons, and other controls are easy to understand and use.

5.3.5. Accessible Multimedia Content

Creating accessible multimedia content, such as videos and audio files, is crucial for users with hearing or visual impairments. This involves providing captions for videos, ensuring that they are accurately timed and convey all relevant information. Additionally, transcripts should be available for audio content, including podcasts and audio descriptions for video content, allowing users with visual impairments to access the information.

By incorporating these assistive technology-friendly design principles, Stackle can create digital products that are accessible and usable by a wide range of users, regardless of their abilities or disabilities. This not only promotes inclusivity but also helps ensure compliance with accessibility standards like WCAG 2.1 Level AA.

5.4. Development Best Practices

In this module, participants will explore coding best practices that promote compatibility with assistive technologies and ensure an accessible user experience. By following these best practices, developers can create digital products that are more inclusive and navigable for all users, including those with disabilities.

5.4.1. Proper Markup

Using semantic HTML is crucial for accessibility, as it provides a clear structure and context for assistive technologies. Developers should use appropriate HTML elements and attributes to define the purpose

of each part of the content, such as headings, lists, and navigation elements. This enables screen readers and other assistive tools to better understand the content and provide meaningful information to users.

5.4.2. CSS Techniques

CSS should be used to separate content from presentation, ensuring that the visual appearance does not interfere with the accessibility of the content. Developers should use flexible layouts, relative font sizes, and high-contrast colors to make content more legible and adaptable to various devices and user settings. Additionally, they should ensure that the content remains accessible even when CSS is disabled or not supported.

5.4.3. Accessible JavaScript

JavaScript should be used carefully, as it can introduce accessibility barriers if not implemented properly. Developers should follow progressive enhancement principles, ensuring that the core functionality of the digital product is available even without JavaScript. They should also use ARIA attributes and accessible event handling to create interactive elements that can be operated by all users, including those using keyboard-only navigation or assistive technologies.

5.4.4. Accessible Forms

Creating accessible forms is crucial for user interaction. Developers should use proper form markup, including fieldsets, legends, and labels to provide context and ensure that assistive technologies can understand the purpose of each form element. They should also implement error handling and validation messages that are clear, concise, and accessible, helping users identify and correct any input errors.

5.4.5. Accessible Tables

Tables should be used primarily for presenting tabular data and should include appropriate markup to provide context and assist navigation. Developers should use table headers (th and scope attributes) to define rows and columns, and add captions and summaries to describe the purpose and structure of the table. They should also ensure that tables can be navigated easily using keyboard shortcuts and that they scale well on different devices.

5.4.6. Interactive Elements

Interactive elements, such as buttons, menus, and sliders, should be designed with accessibility in mind. Developers should use ARIA roles, states, and properties to convey the purpose and status of interactive elements to assistive technologies. They should also ensure that these elements can be operated using keyboard-only navigation and provide clear visual focus indicators to help users understand their current position within the interface.

By following these development best practices, developers at Stackle can create digital products that are compatible with assistive technologies and accessible to a wide range of users, ensuring an inclusive and enjoyable learning experience for all.

5.5. Testing and Evaluation

The Testing and Evaluation module in Stackle's accessibility training program focuses on teaching participants how to effectively test digital products for compatibility with assistive technologies. This module covers various testing tools, manual testing methods, and strategies for involving users with disabilities in the testing process. Additionally, participants will learn how to identify and address accessibility issues during development to ensure the creation of inclusive and accessible learning experiences.

5.5.1. Overview of Accessibility Testing

The module begins with an introduction to the importance of accessibility testing and an overview of the different types of testing methods used to evaluate digital products for compatibility with assistive technologies.

5.5.2. Automated Testing Tools

Participants will learn about various automated testing tools, such as Lighthouse, axe, and WAVE, that can help identify potential accessibility issues in digital products. The module will demonstrate how to integrate these tools into the development process and interpret the results to make informed decisions about accessibility improvements.

5.5.3. Manual Testing Methods

The module covers manual testing methods, such as keyboard-only navigation, screen reader testing, and using browser-based accessibility inspection tools. Participants will learn how to perform these tests and understand their significance in identifying accessibility issues that automated tools might miss.

5.5.4. Inclusive User Testing

In this section, participants will learn the value of involving users with disabilities in the testing process. The module will provide guidance on how to recruit, engage, and gather feedback from users with diverse abilities to gain valuable insights into the real-world accessibility of digital products.

5.5.5. Identifying and Addressing Accessibility Issues

Participants will learn how to identify common accessibility issues during development, such as improper use of semantic HTML, missing alternative text for images, and inadequate color contrast.

The module will provide strategies for addressing these issues and improving overall accessibility.

5.5.6. Testing and Evaluation Best Practices

The module concludes with a summary of best practices for accessibility testing and evaluation, emphasizing the importance of combining automated and manual testing methods, involving users with disabilities in the testing process, and continuously monitoring and improving accessibility throughout the development lifecycle.

By completing the Testing and Evaluation module, participants will gain a comprehensive understanding of the various testing methods and tools required to evaluate digital products for compatibility with assistive technologies. They will be equipped with the knowledge and skills to identify and address accessibility issues during development, ensuring the creation of inclusive and accessible learning experiences for all users.

5.6. Keeping up with Emerging Technologies:

The assistive technology landscape is constantly evolving, presenting new opportunities to enhance the accessibility and usability of digital products. Staying informed about emerging technologies is crucial for developers and designers, as it enables them to create digital products that cater to the diverse needs of users with disabilities. Here are some resources and strategies for keeping up-to-date with the latest advancements in assistive technologies:

5.6.1. Industry News and Publications

Regularly follow industry news and publications focused on accessibility and assistive technologies. These sources can provide insights into the latest trends, innovations, and best practices. Examples of such publications include A List Apart, WebAIM Blog, The Paciello Group Blog, and The Accessible Technology Coalition Blog.

5.6.2. Conferences and Workshops

Attend accessibility-focused conferences, workshops, and seminars, which can provide valuable opportunities for learning and networking. These events typically feature presentations from experts, case studies, and hands-on training sessions. Some notable conferences include CSUN Assistive Technology Conference, Accessing Higher Ground, and Inclusive Design 24.

5.6.3. Webinars and Online Courses

Participate in webinars and online courses that cover assistive technology topics. These resources offer convenient and cost-effective ways to stay informed and expand your knowledge. Platforms such as Coursera, edX, LinkedIn Learning, and WebAIM provide a variety of courses and training materials related to accessibility.

5.6.4. Social Media and Online Communities

Follow accessibility experts, organizations, and advocates on social media platforms like Twitter and LinkedIn. Joining online communities, forums, and mailing lists related to accessibility can also help you stay informed about emerging technologies and trends.

5.6.5. Collaboration with Assistive Technology Users

Engage with users of assistive technologies to gain firsthand insights into their experiences and needs. This can help you identify potential areas for improvement in your digital products and better understand the impact of emerging technologies on user experiences.

5.6.6. Industry Partnerships

Partner with accessibility organizations, consultants, and other experts to stay informed about the latest advancements in assistive

technologies. These partnerships can provide valuable insights, resources, and support in your ongoing efforts to enhance the accessibility of your digital products.

5.6.7. Internal Knowledge Sharing

Establish internal channels for knowledge sharing among team members, such as regular meetings, presentations, or internal newsletters focused on accessibility and assistive technology advancements. This encourages a culture of continuous learning and ensures that the entire team stays informed about the latest developments in the field.

By actively staying informed about emerging technologies, developers and designers can ensure that their digital products continue to meet the diverse needs of users with disabilities. This proactive approach helps maintain a high level of accessibility compliance and fosters an inclusive user experience.

5.7. Module Summary

By completing the Understanding Assistive Technologies module, participants will gain the knowledge and skills necessary to design and develop digital products that are accessible to users with various disabilities. This understanding will contribute to creating an inclusive and accessible learning experience for all users.

6. Common Accessibility Barriers

This module provides an in-depth exploration of common accessibility barriers that users may encounter in digital products, as well as guidance on how to address and prevent them. The goal is to equip designers and developers with the knowledge and tools necessary to create inclusive and accessible experiences for all users.

6.1. Proper Use of Headings and Landmarks

Headings and landmarks play a crucial role in providing structure and organization in digital content, making it more accessible for all users, especially those who rely on screen readers. By using headings and landmarks effectively, content creators can ensure that users can easily navigate and understand the content hierarchy. This section delves into the appropriate use of heading levels (H1-H6) and ARIA landmarks to create a coherent and accessible content structure.

6.1.1. Heading Levels (H1-H6)

Heading levels, ranging from H1 (highest level) to H6 (lowest level), are used to define the structure and hierarchy of content on a webpage. Proper use of headings is essential for creating a clear content outline that can be easily understood by all users, including those using screen readers.

- H1: Use H1 for the main heading of the page, typically the page title. There should be only one H1 per page.
- H2-H6: Use H2-H6 headings to create a logical hierarchy for subheadings within the content. The hierarchy should flow from H2 as the first level of subheading down to H6 as the last level, without skipping levels.

6.1.2. ARIA Landmarks:

ARIA (Accessible Rich Internet Applications) landmarks provide a way to identify specific areas of a webpage for users of assistive technologies, such as screen readers. Landmarks help users quickly navigate to different sections of the page and better understand its structure. Some common ARIA landmarks include:

- banner: Represents the header or branding area of the page.
- navigation: Indicates a navigation menu or section containing links to other pages or sections within the page.
- main: Represents the main content area of the page, excluding headers, footers, and navigation elements.

- **complementary**: Identifies content that complements the main content, such as a sidebar, but can be independently consumed.
- **contentinfo**: Represents the footer or meta-information section of the page, such as copyright and contact information.

To implement ARIA landmarks, use the 'role' attribute within the corresponding HTML elements. For example, **<header role="banner">** or **<nav role="navigation">**.

Best Practices for Using Headings and Landmarks:

- **Ensure a logical heading structure**: The heading hierarchy should flow logically, without skipping levels or using headings for purely visual purposes.
- **Use concise, descriptive headings**: Keep headings brief and informative, so users can quickly understand the content they represent.
- **Use landmarks to identify key page areas**: Implement ARIA landmarks to help users navigate and understand the structure of the page.
- **Avoid overuse of landmarks**: Too many landmarks can make navigation more complex, so use them judiciously to highlight essential page sections.
- **Test with screen readers**: Regularly test your content with screen readers to ensure that the heading and landmark structure is clear and navigable.

By following these guidelines for proper use of headings and landmarks, you can create a coherent and accessible content structure that caters to the diverse needs of all users, including those who rely on assistive technologies.

6.2. Creating Meaningful Link Text

Meaningful link text is essential for ensuring a positive user experience, particularly for those who rely on screen readers or other assistive technologies. By crafting descriptive and informative link

text, developers and content creators can make it easier for users to understand the purpose of a link without needing additional context. This section provides an in-depth look at the best practices for creating meaningful link text that clearly conveys the link's destination or function while avoiding ambiguous phrases like "click here" or "read more."

6.2.1. Be Descriptive and Specific

When creating link text, it's important to be as descriptive and specific as possible about the destination or function of the link. This helps users understand what they can expect when clicking on the link. For example, instead of using "click here" or "learn more," use a more specific phrase like "explore our course catalog" or "view our tutorial on accessible web design."

6.2.2. Provide Context

Ensure that the link text provides enough context for users to understand its purpose, even when viewed in isolation. Users who rely on screen readers often navigate through a page by scanning the list of links, so it's crucial that each link is understandable on its own.

6.2.3. Keep It Concise

While it's important to be descriptive, it's also essential to keep link text concise. Aim for a balance between providing enough information and maintaining brevity. Shorter link text is easier to scan visually and makes it simpler for screen reader users to process.

6.2.4. Use Actionable Language

Using actionable language in link text helps convey the purpose of the link more effectively. For example, use verbs like "download," "register," or "subscribe" to indicate the action users can expect to take when clicking on the link.

6.2.5. Avoid Ambiguous Phrases

Steer clear of generic and ambiguous phrases like "click here," "read more," or "learn more." These phrases do not provide any information about the link's destination or purpose, making it difficult for users to determine if the link is relevant to their needs.

6.2.6. Distinguish Between Different Links

If a page contains multiple links leading to different destinations, ensure that the link text is unique for each one. This helps users differentiate between various links and navigate more efficiently.

6.2.7. Consider the Link's Surrounding Text

When crafting link text, take into account the surrounding text to ensure that the link's purpose is clear. Screen reader users often rely on the surrounding text for context, so it's important that the link text and the content around it work together to convey a clear message.

By following these best practices for creating meaningful link text, developers and content creators can improve the overall accessibility and user experience of their digital products. This, in turn, ensures that users with diverse abilities can easily navigate and understand the content on the website.

6.3. Providing Alternative Text for Images

Alternative text (alt text) is an essential aspect of accessible images, as it offers a textual description of the image content for users who cannot see it. Users relying on screen readers or other assistive technologies benefit significantly from well-crafted alt text. This guide will discuss the importance of alt text and offer best practices for creating effective and meaningful descriptions that enhance the user experience.

6.3.1. Importance of Alt Text

Alt text plays a vital role in web accessibility by allowing users with visual impairments or those using assistive technologies, such as

screen readers, to understand the content and context of images. By providing accurate and descriptive alt text, you ensure that users can access the information conveyed by an image even if they cannot see it.

6.3.2. Context

When creating alt text, consider the context in which the image appears. The alt text should provide relevant information that contributes to the overall understanding of the content. Consider the purpose of the image and its relationship to the surrounding text, as well as what information the user needs to gain from the image.

6.3.3. Brevity

Alt text should be concise yet informative. Aim for a description that is typically between 5–15 words long, as overly lengthy alt text can be cumbersome for users relying on screen readers. While brevity is important, ensure that the description is still meaningful and accurately conveys the image's purpose and content.

6.3.4. Relevance

When writing alt text, focus on the essential elements of the image that are relevant to the user. Descriptions should be specific and informative, without including unnecessary details or visual attributes. Avoid using phrases such as "image of" or "picture of," as these are redundant and do not add value to the description.

6.3.5. Clarity

Ensure that your alt text is clear, descriptive, and easy to understand. Use plain language and avoid jargon or abbreviations that may be unfamiliar to users. Write in complete sentences, using proper grammar and punctuation, to provide a coherent description.

6.3.6. Functionality

If an image serves a functional purpose, such as a button or a link, the alt text should reflect this function. For example, instead of describing the visual appearance of a button, the alt text should indicate the action that will occur when the button is clicked, such as "Submit Form" or "Download PDF."

6.3.7. Decorative Images

For decorative images that do not convey any meaningful information, use empty or null alt text (`alt=""`) to indicate that the image can be ignored by screen readers. This approach ensures that users with assistive technologies are not presented with unnecessary or confusing information.

By following these best practices for providing alternative text for images, you can create a more inclusive and accessible experience for all users. Accurate and meaningful alt text ensures that individuals with visual impairments or those using assistive technologies can fully engage with and understand your digital content.

6.4. Ensuring Sufficient Color Contrast

Color contrast is a crucial aspect of visual accessibility, as it ensures that text and other visual elements are distinguishable for users with low vision or color blindness. Adequate color contrast allows users to perceive content clearly and interact with the interface effectively. This section covers the principles of color contrast, including WCAG guidelines for minimum contrast ratios, and provides guidance on selecting accessible color combinations and using online tools to test contrast levels.

6.4.1. Understanding Color Contrast

Color contrast refers to the difference in luminance or perceived brightness between two colors, typically used for text and its background. High contrast ratios make content more legible, while

low contrast ratios may result in difficulty distinguishing text or other interface elements.

6.4.2. WCAG Guidelines for Minimum Contrast Ratios

The Web Content Accessibility Guidelines (WCAG) 2.1 specify minimum contrast ratios for text and images of text to ensure adequate readability for users with low vision or color blindness.

According to WCAG 2.1:

- 6.4.2.1. Level AA: A minimum contrast ratio of 4.5:1 is required for normal text (below 18-point or 14-point bold), and 3:1 for large text (18-point or 14-point bold and above).
- 6.4.2.2. Level AAA: A minimum contrast ratio of 7:1 is required for normal text, and 4.5:1 for large text.

6.4.3. Selecting Accessible Color Combinations:

To create accessible color combinations, consider the following guidelines:

- Choose colors with high contrast ratios that meet or exceed WCAG requirements.
- Avoid using color combinations that are known to be problematic for users with color blindness, such as red/green or blue/yellow.
- Ensure that interactive elements, such as buttons and links, have sufficient contrast both in their default state and when in focus or hover states.

6.4.4. Testing Color Contrast:

Several online tools can help you test color contrast levels to ensure compliance with WCAG guidelines. These tools analyze the contrast ratio between foreground and background colors and indicate whether the combination meets the required standards. Some popular color contrast checkers include:

- WebAIM Color Contrast Checker (<https://webaim.org/resources/contrastchecker/>)
- Contrast Ratio by Lea Verou (<https://contrast-ratio.com/>)

- Colorable by Brent Jackson (<https://colorable.jxnblk.com/>)

By understanding and applying the principles of color contrast and adhering to WCAG guidelines, you can create accessible digital products that cater to the diverse needs of all users, including those with low vision or color blindness.

6.5. Designing Accessible Forms

Accessible forms are crucial for creating an inclusive user experience, as they enable users with disabilities to interact with and submit information easily. To design accessible forms, it is essential to follow best practices that address the diverse needs of users, including those with visual, auditory, cognitive, and motor impairments. Below are key considerations and best practices for designing accessible forms.

6.5.1. Proper Use of Labels

Labels are essential for providing context and instructions to users, as well as assisting screen reader users in understanding the purpose of each form element. Always associate labels with their corresponding form elements using the 'for' attribute and the form element's 'id' attribute. Ensure that labels are clear, concise, and positioned consistently in relation to form elements.

6.5.2. Fieldsets and Legends

Use the 'fieldset' element to group related form elements and the 'legend' element to provide a descriptive title for the group. This helps users, particularly those using screen readers, to understand the context and relationship between form elements.

6.5.3. Clear Instructions and Error Messages

Provide clear instructions for completing the form, including any required formats or input constraints. When form validation fails, display informative error messages that explain the issue and guide users on how to resolve it. Ensure error messages are easily

perceivable and associated with the relevant form element, using techniques like 'aria-describedby' for screen reader users.

6.5.4. Keyboard Accessibility

Ensure that all form elements and interactive components can be accessed and operated using a keyboard, as some users with motor impairments or those using screen readers may not use a mouse. Provide visual focus indicators for form elements to help keyboard users track their progress through the form.

6.5.5. Logical Tab Order

Arrange form elements in a logical tab order that follows the visual flow of the form. This allows keyboard users to navigate the form efficiently and helps screen reader users understand the expected sequence of interaction.

6.5.6. Input Assistance

Offer input assistance where possible, such as auto-completion, input masks, or pre-populated fields, to help users complete the form more easily. However, ensure that these features can be disabled or bypassed by users who do not wish to use them.

6.5.7. Accessible Controls

Use accessible form controls, such as checkboxes, radio buttons, and dropdown lists, that are compatible with assistive technologies. Avoid using custom controls that may not be accessible or require additional development effort to make them accessible.

6.5.8. Adequate Click Targets

Ensure that form elements and interactive components have adequately sized click targets to accommodate users with motor impairments or those using touch devices.

6.5.9. Time Limits

If a form includes a time limit for completion, provide users with the option to extend the time limit or disable the timer. This helps accommodate users with cognitive impairments or those who require additional time to complete the form.

By following these best practices for designing accessible forms, you can create a more inclusive user experience that caters to the diverse needs of all users, including those with disabilities.

6.6. Module Summary

By understanding and addressing these common accessibility barriers, designers and developers can create digital products that cater to the diverse needs of all users, ensuring a more inclusive and accessible experience.

7. Accessible Coding Practices

The Accessible Coding Practices module is designed to educate developers on how to write accessible HTML, CSS, and JavaScript code. By following the principles and techniques covered in this module, developers can create digital products that are compatible with assistive technologies and easily navigated by all users.

7.1. Semantic HTML

Semantic HTML is a fundamental aspect of creating accessible web content. By using HTML elements that convey the structure and meaning of the content, developers can ensure that assistive technologies can correctly interpret and present the content to users. This approach not only benefits users with disabilities but also enhances the overall user experience and improves the search engine optimization (SEO) of the website.

Participants in our accessibility training program learn the importance of semantic HTML and its role in creating accessible content. Key topics covered in this section include:

7.1.1. Headings

Developers learn the importance of using appropriate heading levels (h1 to h6) to indicate the hierarchy and structure of the content. Proper use of headings makes it easier for screen reader users to understand the content's organization and navigate between sections.

7.1.2. Lists

The training emphasizes the use of ordered (ol) and unordered (ul) lists for presenting items in a logical sequence. Lists help screen reader users understand the relationships between the items and their organization.

7.1.3. Tables

Participants learn how to create accessible data tables using the appropriate table elements, such as `thead`, `tbody`, `th`, and `scope`. These elements provide context and improve the readability of the table for users of assistive technologies.

7.1.4. Forms

The training covers best practices for creating accessible forms, such as using the label element for form controls, grouping related form elements with `fieldset` and `legend`, and providing accessible error messages and validation feedback.

7.1.5. ARIA (Accessible Rich Internet Applications)

ARIA attributes are introduced to enhance the accessibility of dynamic content and complex UI components. Developers learn how to use ARIA landmarks, roles, states, and properties to provide

additional context and improve the usability of interactive elements for assistive technology users.

7.1.6. Navigation

The training highlights the importance of using semantic elements, such as `nav` and `main`, to indicate the primary navigation and main content areas of a webpage. This helps users of assistive technologies to easily navigate and understand the page's structure.

7.1.7. Multimedia

Developers learn best practices for providing accessible multimedia content, such as using the `figure` and `figcaption` elements for images and providing transcripts, captions, and audio descriptions for audio and video content.

By understanding and implementing semantic HTML, developers at Stackle create more accessible and user-friendly digital products. This approach enables users with disabilities to interact more effectively with our content, fostering an inclusive and accessible learning environment.

7.2. ARIA Attributes

Accessible Rich Internet Applications (ARIA) is a technical specification developed by the World Wide Web Consortium (W3C) to improve the accessibility of web content and applications. ARIA attributes provide additional information about elements and their roles, states, and properties, making it easier for assistive technologies like screen readers to understand and interact with complex UI components.

7.2.1. ARIA Roles

ARIA roles are used to define the semantic meaning of an element, helping assistive technologies understand the purpose of each component within a web page. By assigning appropriate roles to elements, developers can convey the intended functionality to users who rely on assistive technologies. Examples of common ARIA roles include "button", "checkbox", "tabpanel", and "navigation".

7.2.2. ARIA States and Properties

ARIA states and properties are used to provide additional information about the current state of an element or its characteristics. These attributes help convey dynamic changes in an element's state or the relationships between different components to assistive technologies. Examples of ARIA states and properties include "aria-expanded", "aria-checked", "aria-labelledby", and "aria-describedby".

7.2.3. Enhancing Complex UI Components

Using ARIA attributes effectively can significantly enhance the accessibility of complex UI components, such as accordions, tabs, and modals. For instance:

- Accordions: By using ARIA roles like "tab" and "tabpanel" and attributes such as "aria-expanded" and "aria-controls", developers can create a more accessible accordion component that communicates its state and functionality to assistive technologies.
- Tabs: Assigning ARIA roles like "tablist", "tab", and "tabpanel" and using attributes like "aria-selected" and "aria-controls" can make it easier for screen reader users to understand and navigate between different sections of a tabbed interface.
- Modals: Implementing ARIA roles like "dialog" and "alertdialog" and attributes such as "aria-labelledby", "aria-describedby", and "aria-modal" can help convey the purpose and structure of a modal dialog, making it more accessible to users who rely on assistive technologies.

7.2.4. Best Practices

While using ARIA attributes can enhance accessibility, it's essential to follow best practices to ensure their effectiveness:

- Use native HTML elements whenever possible, as they often provide built-in accessibility features. Only use ARIA attributes when native elements do not meet your needs.

- Avoid overusing ARIA attributes, as this can create unnecessary complexity and confusion for assistive technology users.
- Test your implementation with various assistive technologies to ensure that the ARIA attributes are correctly interpreted and provide the intended user experience.

In summary, ARIA attributes play a crucial role in enhancing the accessibility of web content and applications. By understanding and implementing ARIA roles, states, and properties effectively, developers can create more accessible and inclusive user interfaces that cater to the needs of individuals who rely on assistive technologies.

7.3. Keyboard Accessibility

Keyboard accessibility is a crucial aspect of creating inclusive and accessible digital experiences. This ensures that users who rely on keyboard navigation, such as individuals with motor impairments or those who prefer not to use a mouse, can interact with the content effectively. The following elements are essential for ensuring keyboard accessibility:

7.3.1. Interactive Element Identification

To ensure keyboard accessibility, it's vital to identify all interactive elements on a web page, such as buttons, links, form fields, and menus. These elements should be accessible through keyboard navigation, allowing users to interact with them using only the keyboard.

7.3.2. Focus Management

Proper focus management is essential for keyboard accessibility. This means that the focus should move logically and predictably through the interactive elements, following a natural and intuitive order. Developers should ensure that the tab order matches the visual order of elements on the page, making it easier for keyboard users to navigate and understand the content.

7.3.3. Keyboard Event Handling

Interactive elements should respond appropriately to keyboard events, such as pressing the Enter or Space key to activate a button or using arrow keys to navigate through menus. Developers need to ensure that all interactive elements have appropriate event handlers for keyboard events, allowing users to perform actions using the keyboard alone.

7.3.4. Visual Focus Indicators

When a user navigates through a web page using the keyboard, it's essential to provide clear visual focus indicators. These indicators highlight the currently focused element, making it easy for users to track their position on the page. Developers should ensure that focus indicators are highly visible and contrast well with the background to accommodate users with low vision.

7.3.5. Skip Navigation Links

Providing skip navigation links at the beginning of a web page allows keyboard users to bypass repetitive content, such as site headers and navigation menus, and jump directly to the main content. This feature improves the efficiency of keyboard navigation and enhances the overall user experience.

7.3.6. Accessible Keyboard Shortcuts

When implementing keyboard shortcuts, developers should ensure that they are accessible and do not conflict with existing keyboard commands used by assistive technologies or the operating system. Providing a list of available keyboard shortcuts and an option to disable or customize them can also enhance accessibility for users who rely on keyboard navigation.

7.3.7. Testing and Validation

To ensure keyboard accessibility, thorough testing should be conducted using only the keyboard. This includes navigating through all interactive elements, performing actions, and verifying that the focus is managed correctly and visual focus indicators are visible and clear.

By implementing these keyboard accessibility principles, developers can create more inclusive and accessible digital experiences, allowing all users, including those with motor impairments or alternative input preferences, to effectively interact with the content.

7.4. Accessible CSS

The Accessible CSS Module is designed to teach developers how to create accessible visual designs using CSS (Cascading Style Sheets). This module focuses on understanding the importance of accessible design and implementing CSS best practices to ensure that digital content is usable and visually appealing for all users, including those with disabilities or diverse needs.

Key Topics:

7.4.1. Proper Color Contrast

This topic covers the importance of using appropriate color contrast ratios for text and background colors to ensure that content is easily readable by users with visual impairments, such as color blindness or low vision. The module teaches developers how to choose accessible color combinations and verify contrast ratios using various tools and guidelines, such as WCAG's color contrast requirements.

7.4.2. Font Sizing and Line Spacing

The module addresses the significance of using accessible font sizes and line spacing to enhance readability and comprehension for users with varying levels of visual ability. Developers will learn how to

use relative units, such as 'em' or 'rem', for font sizing and line-height to ensure that text is easily resizable and adaptable to different devices and user preferences.

7.4.3. Responsive Design

Developers will learn the principles of responsive design, which ensures that digital content adapts seamlessly to different screen sizes and devices, including desktops, tablets, and mobile phones. This topic covers techniques for creating fluid layouts, flexible images, and using CSS media queries to adjust styles based on the characteristics of the user's device.

7.4.4. Media Queries for User Preferences

This topic dives into the use of CSS media queries to detect and accommodate user preferences, such as reduced motion or increased text spacing. Developers will learn how to create accessible animations, transitions, and transformations that respect user preferences, as well as how to implement custom styles that adapt to user settings for text spacing, contrast, and other visual aspects.

7.4.5. Focus Indicators and Visual States

The module highlights the importance of designing clear and visible focus indicators for interactive elements, such as buttons, links, and form controls. Developers will learn how to style various visual states, including focus, hover, and active states, to ensure that users can easily navigate and interact with content using keyboards or other assistive technologies.

By completing the Accessible CSS Module, developers will gain a strong understanding of how to create visually appealing and accessible digital content. They will be equipped with the knowledge and skills to implement CSS best practices that cater to the diverse needs of all users, ensuring an inclusive and user-friendly experience.

7.5. JavaScript and Dynamic Content

Creating accessible dynamic content with JavaScript is crucial for providing an inclusive and user-friendly experience for all users, including those using assistive technologies. To ensure that our digital products meet accessibility requirements, Stackle developers follow best practices and techniques for managing dynamic content in JavaScript.

Key areas of focus include:

7.5.1. Announcing Content Updates

When content on the page is updated dynamically, it's essential to ensure that these changes are announced to assistive technologies like screen readers. Developers can use ARIA Live Regions (`aria-live`) to inform screen readers about updated content, specifying the level of importance (`polite`, `assertive`) to manage updates appropriately.

7.5.2. Managing Focus

As new content is added or removed from a page, developers must manage focus to ensure that users can navigate the changes effectively. This includes setting focus on new elements when they are inserted, maintaining focus on the current element when content is removed, and using appropriate focus management techniques (e.g., `tabindex`) to enable smooth keyboard navigation.

7.5.3. Handling AJAX Requests

When loading content asynchronously using AJAX, it's crucial to ensure that the new content is accessible. Developers should use ARIA attributes (e.g., `aria-busy`, `aria-controls`) to communicate the loading status and relationship between elements, as well as manage focus and announce content updates as described earlier.

7.5.4. Accessible Form Submissions:

Handling form submissions in an accessible manner involves validating user input, providing clear error messages, and ensuring

that users can easily navigate between form fields. Developers should use appropriate ARIA roles and attributes (e.g., `aria-required`, `aria-invalid`, `aria-describedby`) to convey the state and purpose of form elements, and manage focus and error announcements to facilitate smooth user interactions.

7.5.5. Keyboard Accessibility

As dynamic content is added, updated, or removed, developers must ensure that it remains keyboard accessible. This includes providing keyboard equivalents for all interactive elements (e.g., buttons, links) and ensuring that the tab order is logical and consistent with the visual layout.

7.5.6. Responsiveness and Adaptability

Dynamic content should be designed to adapt to different devices, screen sizes, and orientations to maintain accessibility across various contexts. Developers should use responsive design techniques and ensure that interactive elements are touch-friendly and easily operable on mobile devices.

By following these best practices for managing JavaScript and dynamic content, Stackle developers create accessible digital products that cater to the diverse needs of all users. This ensures that our learning experiences remain inclusive and user-friendly, regardless of the user's abilities or the assistive technologies they rely on.

7.6. Accessible Forms

The Accessible Forms module focuses on the essential techniques and best practices required to create accessible forms that cater to a wide range of users, including those with disabilities. By ensuring that forms are accessible, we can provide a more inclusive and user-friendly experience for all learners interacting with our digital products. This module covers the following key aspects:

7.6.1. Proper Labeling of Form Controls

Clear and descriptive labels should be provided for all form controls, such as text inputs, checkboxes, radio buttons, and select menus. Labels should be explicitly associated with their corresponding form controls using the 'for' attribute with the 'id' attribute of the control. This makes it easier for users, especially those using assistive technologies like screen readers, to understand the purpose of each form control and complete the form successfully.

7.6.2. Grouping Related Controls

When a form contains multiple related controls, such as a set of radio buttons or checkboxes, these controls should be grouped using appropriate markup like the 'fieldset' and 'legend' elements. Grouping related controls improves the organization of the form and helps users understand the relationship between the controls, making it easier for them to navigate and complete the form.

7.6.3. Providing Helpful Error Messages

Forms should be designed to provide clear and helpful error messages when users enter incorrect or incomplete information. Error messages should be concise, descriptive, and placed close to the associated form control, ensuring that users can easily identify and rectify any mistakes. Additionally, error messages should be programmatically associated with the relevant form controls using techniques like 'aria-describedby' to ensure they are accessible to assistive technologies.

7.6.4. Implementing Accessible Form Validation

Form validation should be implemented in a way that is accessible to all users. This includes providing both client-side and server-side validation to accommodate users with various types of browsers and assistive technologies. Client-side validation should be implemented using JavaScript techniques that do not rely on mouse interactions, such as the 'onsubmit' event, and should provide visual

and programmatic feedback (using ARIA roles and properties) to inform users of any errors or incomplete fields.

7.6.5. Keyboard Accessibility

All form controls and interactive elements should be easily navigable and operable using the keyboard alone. This includes ensuring that form controls can be accessed using the 'Tab' key and activated using the 'Enter' key or other appropriate keys. Providing full keyboard accessibility ensures that users who cannot use a mouse or other pointing devices can still interact with and complete the form.

7.6.6. Accessible Instructions and Help Text

Any instructions or help text provided for form controls should be clear, concise, and easily accessible to all users. This includes ensuring that the text is large enough to read, has sufficient color contrast, and is programmatically associated with the relevant form controls using the 'aria-describedby' attribute. Providing accessible instructions and help text supports users in understanding the form's requirements and completing it successfully.

By incorporating these principles and techniques into the design and development of forms, Stackle ensures that our digital products provide an accessible, inclusive, and user-friendly experience for all learners.

7.7. Accessible Multimedia

The Accessible Multimedia section focuses on ensuring that multimedia content, including images, audio, and video, is accessible to all users, regardless of their abilities or disabilities. This involves providing alternative ways to access the information conveyed through multimedia content, enabling users with visual, auditory, or other impairments to understand and engage with the content effectively.

7.7.1. Alternative Text for Images

Alternative text, or alt text, is a brief description of an image's content or purpose. It is used by screen readers and other assistive

technologies to convey the meaning of the image to users who are visually impaired or cannot see the image for any reason.

Developers and content creators should provide meaningful and descriptive alternative text for all images, ensuring that users who rely on screen readers can understand the context and information conveyed by the image.

7.7.2. Transcripts for Audio Content

Transcripts are text versions of spoken content in audio files or audio tracks within multimedia content. Providing transcripts allows users who are deaf or hard of hearing, or those who cannot access audio content for any reason, to read the spoken content. Transcripts should be accurate, comprehensive, and synchronized with the audio content to ensure an equivalent experience for users who rely on them.

7.7.3. Captions for Video Content

Captions are text representations of the spoken content and important non-verbal sounds in a video. They are synchronized with the video and displayed within the video player or on a separate captioning device. Captions enable users who are deaf or hard of hearing to access the audio information in a video. When creating captions, it is essential to ensure they are accurate, complete, and well-timed to provide an equivalent experience for users relying on them.

7.7.4. Audio Descriptions for Video Content

Audio descriptions are narrations that describe essential visual elements in a video, such as actions, scene changes, and on-screen text. These descriptions are added to the video's audio track and can be accessed by users who are visually impaired or have difficulty understanding the visual content. Audio descriptions should be clear, concise, and synchronized with the video content to provide an equivalent experience for users who rely on them.

By incorporating accessible multimedia practices into the development and content creation process, Stackle ensures that all users can access and engage with multimedia content effectively. This commitment to accessible multimedia helps create an inclusive learning environment that caters to the diverse needs of all users, aligning with our overall accessibility goals and guidelines.

7.8. Testing and Debugging

The Testing and Debugging module is an essential part of Stackle's accessibility training program. It focuses on equipping participants with the skills and knowledge needed to effectively test and debug accessible code. The module covers various testing methodologies, tools, and techniques to ensure that our digital products meet WCAG 2.1 Level AA guidelines and provide an inclusive learning experience for all users.

7.8.1. Introduction to Accessibility Testing

This section introduces participants to the importance of accessibility testing and the various types of testing methods, including automated and manual testing, as well as user testing with individuals with diverse abilities. Participants will learn about the benefits and limitations of each method and how to combine them effectively for thorough accessibility testing.

7.8.2. Automated Testing Tools

Participants will explore various automated accessibility testing tools, such as browser extensions (e.g., axe, WAVE), and testing suites (e.g., Lighthouse, pa11y). This section covers how to use these tools to identify potential accessibility issues in code quickly, as well as understanding the limitations of automated testing and when manual testing is necessary.

7.8.3. Manual Testing Techniques

This section delves into manual testing techniques, which play a crucial role in ensuring accessibility compliance. Participants will

learn how to navigate a web page using only a keyboard and how to identify potential barriers for keyboard-only users. They will also be introduced to techniques for visually inspecting a web page for proper color contrast, text resizing, and focus indicators.

7.8.4. Screen Reader Testing

Participants will gain hands-on experience with popular screen readers, such as NVDA and VoiceOver, to understand how individuals with visual impairments interact with digital content. This section covers the basics of using screen readers for testing purposes, common issues that may arise, and best practices for coding to ensure compatibility with screen readers.

7.8.5. Debugging Accessible Code

This section focuses on identifying and resolving accessibility issues discovered during testing. Participants will learn how to interpret test results, locate the source of issues in code, and apply appropriate fixes to ensure WCAG compliance. They will also learn about common accessibility pitfalls and how to avoid them during the development process.

7.8.6. Integrating Testing into the Development Lifecycle

In this section, participants will learn how to integrate accessibility testing into their development workflow effectively. They will explore strategies for incorporating automated testing into the build pipeline, as well as the importance of allocating time for manual testing and user feedback throughout the development process.

7.8.7. Documenting and Communicating Test Results

The module concludes with an emphasis on documenting and communicating accessibility test results to stakeholders. Participants will learn how to create clear, concise reports that detail accessibility issues, proposed solutions, and timelines for

remediation. They will also learn how to communicate the importance of accessibility testing to project stakeholders, fostering a culture of accessibility awareness within the organization.

By completing the Testing and Debugging module, participants will be equipped with the necessary skills and knowledge to effectively test and debug accessible code, ensuring that Stackle's digital products meet accessibility standards and provide an inclusive learning experience for all users.

7.9. Module Summary

By completing the Accessible Coding Practices module, developers gain a thorough understanding of how to create accessible digital products that adhere to the Web Content Accessibility Guidelines (WCAG) 2.1 Level AA and cater to the diverse needs of all users.

8. Testing and Evaluation:

The Testing and Evaluation module is an essential part of Stackle's accessibility training program. It provides participants with a comprehensive understanding of various accessibility testing methods and tools, including both automated and manual testing approaches. This knowledge is crucial in identifying and addressing potential accessibility issues in our digital products.

Key components of the Testing and Evaluation module include:

8.1. Overview of Accessibility Testing

Accessibility testing plays a crucial role in ensuring an inclusive and accessible user experience for all, including individuals with diverse abilities and disabilities. It involves evaluating digital products, such as websites, applications, and e-learning resources, to identify and address accessibility barriers that may prevent users from fully interacting with and benefiting from the content.

8.1.1. The Importance of Accessibility Testing:

Accessibility testing is essential to:

1. Ensure compliance with accessibility guidelines and legislation, such as the Web Content Accessibility Guidelines (WCAG).
2. Create inclusive digital products that cater to users with diverse abilities, including those with visual, auditory, cognitive, and motor impairments.
3. Enhance usability, as accessible design principles often lead to an improved user experience for all users, not just those with disabilities.
4. Protect against potential legal risks, as non-compliant websites and applications may be subject to lawsuits and fines.

8.1.2. Types of Accessibility Testing

Accessibility testing can be conducted using various methods to ensure a comprehensive evaluation of the digital product. These methods include:

8.1.3. Automated Testing

Automated accessibility testing tools, such as Lighthouse, axe, or WAVE, help identify potential issues by scanning the code and content of a digital product. These tools can quickly detect common accessibility problems, such as missing alternative text for images, improper heading structure, or incorrect ARIA attributes. However, automated testing tools may not capture all accessibility issues, so it's essential to combine them with other testing methods.

8.1.4. Manual Testing

Manual accessibility testing involves a human tester evaluating the digital product using a set of predetermined guidelines and criteria, such as WCAG. Manual testing may include assessing keyboard navigation, screen reader compatibility, and proper usage of semantic markup. This type of testing can identify issues that automated tools might miss, providing a more in-depth evaluation of the product's accessibility.

8.1.5. User Testing

User testing involves individuals with diverse abilities interacting with the digital product to provide real-world feedback on its accessibility and usability. This type of testing is invaluable, as it offers insights into how users with disabilities experience the product and highlights areas for improvement that may not be apparent through automated or manual testing alone.

8.1.6. Assistive Technology Testing

Testing with assistive technologies, such as screen readers, magnifiers, and speech recognition software, is essential to ensure compatibility and a seamless user experience for individuals who rely on these tools to access digital content.

Combining these testing methods provides a thorough evaluation of the digital product's accessibility and helps identify areas for improvement. By integrating accessibility testing into the development lifecycle, Stackle can create digital products that cater to the diverse needs of all users and foster an inclusive and accessible learning environment.

8.2. Automated Testing Tools

As part of our commitment to accessibility, Stackle provides training on various automated testing tools that help assess accessibility compliance. These tools are crucial in identifying potential issues early in the development process and ensuring that our digital products adhere to the Web Content Accessibility Guidelines (WCAG) 2.1 Level AA.

8.2.1. Lighthouse

Lighthouse is an open-source tool developed by Google that provides automated auditing for various aspects of web pages, including performance, SEO, and accessibility. Participants learn the following about Lighthouse:

1. How to install and use Lighthouse as a browser extension or a command-line tool.

2. How to generate and interpret accessibility audit reports, including understanding the scoring system and recommendations.
3. The benefits and limitations of using Lighthouse, such as its ability to identify many accessibility issues automatically but not being able to cover all WCAG criteria due to the limitations of automated testing.

8.2.2. WAVE (Web Accessibility Evaluation Tool)

WAVE, developed by WebAIM, is another popular automated accessibility testing tool. It is available as a browser extension and online service. Training covers the following aspects of WAVE:

1. How to install and use WAVE as a browser extension or via the online service.
2. How to generate and interpret accessibility reports, which provide an overview of issues, alerts, and features related to accessibility.
3. The benefits and limitations of using WAVE, such as its ease of use and helpful visual representations of issues, but also its inability to cover all WCAG criteria.

8.2.3. Best Practices for Using Automated Testing Tools:

Participants are also taught best practices for using these tools effectively:

1. Understanding that automated testing tools are only part of the accessibility evaluation process and cannot replace manual testing or user feedback.
2. Combining the use of multiple tools to get a more comprehensive view of potential accessibility issues.
3. Regularly using these tools throughout the development process to identify and address issues early, reducing the need for extensive remediation later.

By providing training on automated testing tools like Lighthouse and WAVE, Stackle equips its development team with the knowledge and skills needed to effectively assess and improve the accessibility of our digital products.

This ensures that we maintain a strong focus on creating inclusive and accessible learning experiences for all users.

8.3. Manual Testing Techniques

Manual testing techniques play a crucial role in ensuring the accessibility of digital products, as they help identify issues that automated testing tools might overlook. These techniques involve human testers interacting with the product in various ways to simulate the experiences of users with diverse abilities. Here are some key manual testing techniques:

8.3.1. Keyboard Testing

Keyboard testing involves navigating the product using only a keyboard, without relying on a mouse or other pointing devices. Testers use the 'Tab', 'Shift+Tab', arrow keys, 'Enter', and 'Space' keys to interact with interactive elements, such as links, buttons, and form controls. This process helps identify any elements that are not keyboard-accessible, ensuring that users who rely on keyboards or keyboard-like devices can effectively navigate and interact with the content.

8.3.2. Focus Management

Proper focus management is essential for users who rely on keyboard navigation or assistive technologies like screen readers. During manual testing, testers ensure that the focus order is logical and follows a consistent pattern, enabling users to understand the structure and flow of the content. Testers also check that focus indicators, such as visible outlines or highlights, are present and clearly visible, helping users know which element currently has focus.

8.3.3. Screen Reader Testing

Screen reader testing involves using popular screen reader software like JAWS, NVDA, or VoiceOver to navigate and interact with the content. Testers with expertise in screen reader usage assess how effectively the software conveys information and whether users can

understand and interact with the content as intended. This process helps identify issues related to semantic markup, ARIA attributes, and other factors that impact the experience for screen reader users.

8.3.4. Colour Contrast and Visibility

Testers evaluate the colour contrast between text and background elements to ensure compliance with WCAG guidelines, which require a minimum contrast ratio for better readability. They also assess the visibility and legibility of text, icons, and other visual elements to ensure that users with low vision or colour vision deficiencies can perceive and understand the content.

8.3.5. Zoom and Magnification Testing

Testers use browser zoom and magnification tools to simulate the experience of users who require enlarged content. They evaluate whether the content scales properly, remains legible, and maintains a functional layout when zoomed or magnified.

8.3.6. Assistive Technology Compatibility

In addition to screen readers, testers assess the compatibility of the product with other assistive technologies, such as speech recognition software, alternative input devices, and screen magnifiers. This process helps ensure that the product works well for users who rely on various assistive tools.

By incorporating manual testing techniques into the accessibility testing process, Stackle can identify and address a wider range of accessibility issues. This comprehensive approach ensures that our digital products cater to the diverse needs of all users and provide an inclusive and accessible experience.

8.4. Screen Reader Testing

Screen reader testing is a crucial aspect of ensuring accessibility compliance and providing an inclusive user experience for individuals who rely on assistive technologies. As part of our commitment to accessibility,

Stackle conducts thorough screen reader testing to evaluate the compatibility of our digital content with popular screen readers such as NVDA, JAWS, and VoiceOver. This helps us identify and address potential issues that may impact screen reader users.

8.4.1. Hands-on Training

Participants in our accessibility training program receive hands-on instruction in using screen readers. This training aims to familiarize team members with the functionality, navigation, and interaction techniques of various screen readers, enabling them to better understand the needs and experiences of screen reader users.

8.4.2. Navigation Techniques

During training, team members learn different navigation techniques used by screen reader users, such as keyboard shortcuts, heading navigation, and landmark navigation. This helps them assess how easy it is for screen reader users to move through our digital content and interact with different elements.

8.4.3. Interacting with Elements

Our training also covers how screen reader users interact with various web elements, such as links, buttons, forms, and multimedia content. Team members learn the importance of providing proper labels, alternative text, and accessible controls to ensure a smooth and frustration-free experience for screen reader users.

8.4.4. Identifying Issues

As part of the screen reader testing process, team members are trained to identify potential issues that may affect users of assistive technologies. This includes detecting missing or improper labels, inaccessible content, and any other barriers that could impede the overall user experience.

8.4.5. Collaboration and Remediation

After identifying issues during screen reader testing, team members collaborate with designers, developers, and other stakeholders to address and resolve these concerns. Our goal is to ensure that our digital content is fully accessible and provides a seamless experience for all users, including those who rely on screen readers.

8.4.6. Continuous Improvement

We recognize the importance of staying up-to-date with the latest developments in assistive technologies and screen reader software. To ensure our team members maintain their knowledge and expertise, we provide ongoing training, updates, and resources related to screen reader testing and accessibility best practices.

By incorporating screen reader testing into our accessibility evaluation process, Stackle demonstrates a commitment to creating accessible and inclusive digital content for all users. This thorough testing helps us identify potential barriers and work towards delivering a seamless and enjoyable learning experience for individuals who rely on screen readers.

8.5. Testing with Assistive Technologies

This module focuses on understanding and conducting testing with various assistive technologies, such as screen magnifiers, speech recognition software, and alternative input devices. By familiarizing themselves with these tools, development team members can identify potential accessibility barriers and ensure compatibility with a wide range of technologies, ultimately providing a more inclusive and accessible user experience.

8.5.1. Screen Magnifiers

Screen magnifiers are tools that help users with low vision enlarge and manipulate the content displayed on the screen. Participants in this module learn how to test with different magnification levels and

settings to ensure that the layout, design, and functionality of the product remain usable and accessible, even when zoomed in.

8.5.2. Screen Readers

Screen readers are software applications that convert on-screen text and interface elements into spoken words or Braille output for users with visual impairments. This module teaches participants how to test with popular screen readers like JAWS, NVDA, and VoiceOver to ensure that the product's content and interactions are accurately conveyed and accessible to users relying on these tools.

8.5.3. Speech Recognition Software

Speech recognition software, such as Dragon NaturallySpeaking, allows users to control their devices and input text using voice commands. Participants learn how to test the product's compatibility with speech recognition software to ensure that users can effectively navigate, interact with, and input information using their voice.

8.5.4. Alternative Input Devices

Alternative input devices, such as switch controls or eye-tracking systems, enable users with mobility impairments or other disabilities to interact with digital products. In this module, participants learn how to test compatibility with various alternative input devices, ensuring that the product's interface and interactions can accommodate diverse user needs.

8.5.5. Testing Scenarios and Techniques

This module covers different testing scenarios and techniques, including testing for keyboard accessibility, touch screen interactions, and compatibility with various operating systems and browsers. Participants learn how to perform comprehensive testing that considers diverse user needs and preferences.

8.5.6. Identifying Accessibility Barriers

Participants learn how to identify potential accessibility barriers using assistive technologies during testing, including issues related to color contrast, font sizes, focus management, and ARIA attributes. They also learn how to prioritize and address these issues based on their impact on user experience and compliance with accessibility standards.

8.5.7. Documentation and Reporting

This module teaches participants how to document and report any accessibility issues identified during testing with assistive technologies. Proper documentation helps track progress and ensures that accessibility concerns are adequately addressed throughout the development process.

By incorporating testing with assistive technologies into the development lifecycle, Stackle ensures that its digital products are compatible with a wide range of tools and accessible to all users, regardless of their abilities or the technologies they use.

8.6. User Testing

User testing plays a crucial role in ensuring that our products cater to the diverse needs of all users, including those with disabilities. By involving individuals with diverse abilities in the testing process, we gain valuable insights into potential accessibility barriers and areas for improvement. This section provides an overview of best practices for conducting user testing with individuals with diverse abilities.

8.6.1. Participant Recruitment

To ensure the effectiveness of user testing, it's essential to recruit a diverse group of participants with various abilities, including those who use assistive technologies. Participant recruitment can be achieved through community outreach, partnerships with disability advocacy organizations, and targeted advertisements. It's important

to consider factors such as age, gender, cultural background, and type of disability to ensure a representative sample of users.

8.6.2. Inclusive Test Environment

Creating an inclusive testing environment is vital for accommodating the needs of participants with diverse abilities. This may involve providing accessible facilities, offering alternative formats for instructions, and incorporating any necessary accommodations, such as sign language interpreters or additional time for completing tasks.

8.6.3. Test Planning

When planning user testing sessions, it's essential to develop test scenarios and tasks that cover key user journeys and interactions with the product. These tasks should be designed with accessibility in mind, ensuring that they can be completed by individuals with diverse abilities. Additionally, it's crucial to establish clear goals and objectives for each testing session, which will help guide the evaluation of test results.

8.6.4. Conducting User Testing

During user testing sessions, it's important to establish open communication with participants and encourage them to share their thoughts and experiences. Observing how participants interact with the product and asking probing questions can help uncover potential accessibility issues that may not be apparent through automated testing. It's also essential to record these sessions for later analysis.

8.6.5. Evaluation of Test Results

After the user testing sessions, the team should thoroughly analyze the results to identify trends, patterns, and potential accessibility issues. This process may involve reviewing session recordings, analyzing user feedback, and discussing observations among team

members. The insights gained from this analysis can then be used to inform product improvements and prioritize accessibility-related tasks.

8.6.6. Implementing Feedback

The feedback gathered from user testing should be incorporated into the development process to address any identified accessibility issues. This may involve updating design elements, modifying code, or refining the overall user experience. Regular communication with participants throughout this process can help ensure that their feedback is effectively addressed.

8.6.7. Continuous Improvement

User testing should be an ongoing process, with additional sessions conducted as the product evolves and new features are introduced. Regular testing with individuals with diverse abilities helps ensure that accessibility remains a priority and that any new issues are identified and addressed promptly.

By following these best practices for user testing involving individuals with diverse abilities, Stackle can effectively identify and address accessibility barriers, ensuring that our digital products cater to the needs of all users.

8.7. Reporting and Remediation

The Reporting and Remediation section in our accessibility training focuses on teaching participants how to effectively document, report, and prioritize accessibility issues discovered during testing. This training module ensures that our team members have a clear understanding of how to develop and implement remediation plans to address these concerns and enhance the overall accessibility of our digital products.

8.7.1. Identifying Accessibility Issues

Participants learn how to identify accessibility issues using various testing methods, such as automated tools, manual testing, and user

feedback. They become familiar with common accessibility barriers and the corresponding WCAG success criteria.

8.7.2. Documenting Accessibility Issues

The training emphasizes the importance of thoroughly documenting accessibility issues. Participants learn how to record essential information such as the nature of the issue, the affected product or feature, the WCAG success criterion it violates, and any additional relevant details.

8.7.3. Reporting Accessibility Issues

Our team members are trained on how to effectively report accessibility issues using an issue-tracking system. This ensures that all relevant stakeholders are aware of the issues and can collaborate on finding solutions.

8.7.4. Prioritizing Accessibility Issues

Participants learn how to prioritize accessibility issues based on factors such as their impact on users, the complexity of remediation, and alignment with organizational goals. This helps the team focus on addressing the most critical issues first and ensures efficient allocation of resources.

8.7.5. Developing Remediation Plans

The training guides participants in creating remediation plans that outline the necessary steps to resolve accessibility issues. These plans include assigning responsibility to specific team members, setting timelines for completion, and identifying any resources or support needed to address the issue.

8.7.6. Implementing Remediation Plans

Our team members learn how to collaborate with other stakeholders, such as designers, developers, and quality assurance

professionals, to implement remediation plans effectively. This includes coordinating efforts, monitoring progress, and providing regular updates on the status of issue resolution.

8.7.7. Reviewing and Validating Remediation Efforts

Finally, participants are taught how to review and validate the success of remediation efforts by retesting the affected product or feature. This ensures that the implemented solution effectively resolves the identified accessibility issue and complies with the relevant WCAG success criteria.

By providing comprehensive training on reporting and remediation, Stackle ensures that our team members are equipped with the knowledge and skills required to address accessibility issues effectively and contribute to creating a more inclusive and accessible learning environment for all users.

8.8. Module Summary

By completing the Testing and Evaluation module, participants gain a comprehensive understanding of various accessibility testing methods and tools. This knowledge enables them to effectively identify and address accessibility issues, ensuring that Stackle's digital products provide an inclusive and accessible experience for all users.

9. Ongoing Learning and Resources

Stackle recognizes the importance of continuous learning and staying current with accessibility advancements to create a truly inclusive and accessible learning environment. To support this, the accessibility training provided to our team members concludes with a comprehensive list of resources and tools for further learning. Participants are encouraged to actively engage with these resources and continuously expand their knowledge and skills in web accessibility.

9.1. Online Courses and Webinars

Our team members are provided with access to various online courses, webinars, and tutorials that cover a wide range of accessibility topics. These resources help them deepen their understanding of accessibility principles, assistive technologies, and techniques for creating accessible digital products.

9.2. Blogs and Newsletters

We encourage our team to subscribe to accessibility-focused blogs and newsletters, which provide insights into the latest trends, case studies, and best practices in the field. These resources help our team stay informed about industry updates and advancements in web accessibility.

9.3. Conferences and Workshops

Staff members are given opportunities to attend conferences, workshops, and meetups related to accessibility. These events facilitate networking with accessibility professionals, learning from experts in the field, and staying up-to-date with the latest research and technologies.

9.4. Accessibility Guidelines and Standards

Our team is encouraged to regularly review and stay informed about updates to accessibility guidelines and standards, such as the Web Content Accessibility Guidelines (WCAG) and the Authoring Tool Accessibility Guidelines (ATAG). This ensures that our digital products adhere to the most current best practices and requirements.

9.5. Community Involvement

Stackle promotes active involvement in accessibility communities and forums, where team members can collaborate, share knowledge, and learn from the experiences of others working in the field. Engaging with these communities helps our staff stay informed about emerging accessibility issues and solutions.

9.6. Internal Knowledge Sharing

We foster a culture of internal knowledge sharing, where team members can discuss their learnings, experiences, and insights related to accessibility. Regular team meetings, workshops, and presentations facilitate the exchange of ideas and help spread accessibility awareness throughout the organization.

9.7. Accessibility Tools and Assistive Technologies

Our team is provided with access to various accessibility testing tools, such as Lighthouse and axe, as well as assistive technologies like screen readers and voice recognition software. Familiarity with these tools and technologies helps our team better understand the needs of users with diverse abilities and design accessible digital products accordingly.

By promoting ongoing learning and providing access to valuable resources, Stackle ensures that our team members remain well-versed in web accessibility and are equipped to create inclusive and accessible digital products for all users.