## File Structure

The file structure of the inclusive shopping website project is organized into directories that segregate the various aspects of the site for easier management and development. Here's a high-level overview:



## Discussion of all pages:

Footer Component

The footer component serves as the bottom section across all pages of the shopping website, offering users additional navigation options and customer support information.

Key Features:

- Brand Statement: A welcoming message that emphasizes the website's commitment to a seamless shopping experience.

- Quick Links: Direct access to essential pages such as Home, Shop, Deals, and Contact for easy navigation.

- Customer Support: Provides an email link for customer queries, facilitating easy communication.

- Social Media Icons: Links to various social media platforms, promoting community engagement.

- Back to Top Button: A convenience feature allowing users to return to the top of the page without manual scrolling.

### Nav-Bar Page

The Nav-Bar page serves as a reusable component that provides a consistent navigation experience across the shopping website. It offers users a familiar interface to return to the homepage, receive notifications, adjust settings, and access their shopping cart, all from any page on the site.

Key Features:

- Responsive Design: Adapts to various screen sizes and devices, ensuring accessibility and usability.

- Dynamic Shopping Cart Indicator: Shows a real-time update of the number of items in the user's shopping basket.

- Search Functionality: Allows users to search for products directly from the navigation bar.

- Accessibility Features: Includes a 'Read Screen' option for screen readers and users with visual impairments.

Navigation:

Users can navigate to the Nav-Bar from any page on the website as it is included in the header across all pages. From the Nav-Bar, users can go to:

- Homepage: By clicking on the 'Home' icon or the 'Shopping' brand name.

- Notifications Page: By selecting the 'Notifications' bell icon.

- Settings Page: Through the 'Settings' gear icon.

- Basket Page: By interacting with the cart icon, which displays the current count of items.

### Basket.html

This page serves as the final checkpoint for users to review their selected items before proceeding to payment. It is a crucial step in the purchasing process where users confirm their order and enter billing and shipping details.

Key Features

- A dynamic progress bar that indicates the user's current stage in the shopping process.

- A detailed summary of items in the shopping cart, including product names, descriptions, and prices.

- A section for applying promotional codes for discounts.

- Form fields for entering billing and shipping information.

- Selection of payment methods, including credit card, debit card, and PayPal options.

- Payment information fields, including card details and billing address.

- Modal popup confirmation upon completing the purchase.

### Contact Us Page

The Contact Us page serves as a dedicated space for visitors to reach out to the website's support team. It provides an avenue for users to ask questions, seek assistance, provide feedback, or get more information about products and services.

- Key Features:

- Contact Form: Enables users to submit their queries directly through the website.

- Contact Information: Displays the email and phone number for users preferring to contact via traditional methods.

- Social Media Links: Quick access to the website's social media platforms for broader engagement.

- Customer Testimonials: Showcases customer feedback and experiences to build trust and credibility.

- Frequently Asked Questions (FAQs): Answers to common queries to help users find information quickly without needing to reach out.

- Explore Our Collection Provides a sneak peek into various products, inviting users to explore more items from the shopping collection.

### Deals Page

The Deals page is designed to showcase exclusive deals and offers across various categories such as electronics, home decor, fitness gear, and tech accessories. It aims to attract users with time-sensitive discounts, seasonal clearances, limited-time offers, and exclusive online deals, encouraging them to make the most of these opportunities before they expire.

Key Features:

- Exclusive Deals Section: Highlights current sales on electronics, home decor, and more, with details on the discount percentage and offer expiry.

- Limited Time Offers and Seasonal Clearance: Special sections for flash sales and end-of-season clearances, providing additional savings on select items.

- New Arrivals and Online Offers: Introduces users to the latest products and online-only discounts, enhancing the shopping experience.

-Upcoming Deals and Customer Favorites: A sneak peek at future sales and a showcase of popular products favored by customers.

- Newsletter Subscription: Allows users to subscribe to a newsletter for updates on new deals and special offers.

### Home Page

The Home Page serves as the entry point to the website, welcoming users and providing an overview of what the site offers, including popular items, exclusive deals, and more. It's designed to engage users with a visually appealing layout and intuitive navigation, guiding them towards featured products, deals, and important site sections.

Key Features:

- Interactive Tour: Utilizes intro.js for a step-by-step guide through the site's features and navigation, enhancing user orientation.

- Carousel of Popular Items: Displays a rotating selection of popular items, each with a brief description and pricing, to highlight trending products or deals.

- Accessibility Features: Demonstrates the website's commitment to accessibility with sections on accessibility tools and design principles, ensuring the site is usable for people with diverse abilities.

- Quick Links and Contact Information: Offers easy navigation to other parts of the site and provides contact information for customer support and social media links.

- Navigation:

- Directly accessible upon entering the website, with links to other major sections like Notifications, Settings, and the Shopping Basket prominently displayed in the navigation bar.

- Users can move from the Home Page to specific product categories, deals, contact information, and more, with intuitive navigation paths provided through the menu and various site sections.

### Items Page

The Items page serves as a comprehensive catalog for all products available on the website. It offers users a broad view of different categories and selections, ranging from special offers to popular picks, facilitating easy browsing and shopping experience.

- Key Features:

- Categories Section: Allows users to filter products based on different categories, enhancing the shopping experience by making it easier to find products of interest.

- Product Listings: Each product is displayed with an image, name, and price, with some items highlighted as "Sale" or "Popular" to guide users towards deals or trending products.

- Sale Badges: Highlight special offers and discounts, drawing attention to bargains and deals available on the site.

- Interactive Product Cards: Each card includes an option to view more details about the product, encouraging further interaction.

- Navigation:

- The Items page can be accessed from the navigation bar at the top of every page, under the "Shop" or "Items" link.

- From the Items page, users can navigate to detailed product pages via the "View options" or "Have a look" buttons on each product card.

- The navigation bar also provides links back to the Homepage, Notifications, and Settings pages, along with a search function and login/signup options.

### Settings Page

The Settings page is dedicated to allowing users to customize their shopping experience on the website. Users can tailor their preferences for receiving notifications, deals, and updates across different categories such as electronics, clothing, and accessories. Furthermore, this page provides accessibility settings to ensure a comfortable browsing experience for all users.

- Key Features:

- Category-Specific Notifications: Users can opt-in to receive notifications for deals in specific categories like electronics, clothing, and accessories.

- Preference Customization: For categories like electronics and clothing, users can specify preferences such as preferred categories or clothing sizes.

- Accessibility Options: Features various accessibility settings, including text-to-speech, high contrast mode, larger text, dark mode, voice control, and reduced motion options to improve site navigation for users with specific needs.

- Navigation:

- The Settings page can be accessed directly from the navigation bar present on all pages of the website.

- Navigation within the Settings page is facilitated through a form that organizes settings into distinct sections, making it easy for users to locate and adjust their preferences.

- Users can save their changes with a "Save Changes" button, ensuring their preferences are updated immediately.

### Single-item Page

The Single-item page is dedicated to showcasing detailed information about a specific product selected by the user. It provides comprehensive details about the product, including images, descriptions, pricing, and available offers. This page aims to give users all the information they need to make an informed purchase decision.

- Key Features:

- Detailed Product Information: Includes high-quality images of the product, a detailed description, the item number, and the price, including any discounts.

- Add to Cart: Features an "Add to Cart" button that allows users to select the quantity of the product they wish to purchase and add it directly to their shopping cart.

- Progress Bar: A progress bar indicates the step the user is currently on in the shopping process, enhancing the user experience by providing clear navigation cues.

- Related Products Section: Showcases products related to the one being viewed, encouraging users to explore similar items and potentially increase their purchase.

- Navigation:

- Users can navigate to the Single-item page by selecting a product from anywhere on the site, such as the homepage, search results, or category pages.

- Navigation to other parts of the site is facilitated through the navigation bar, including links to the homepage, notifications, settings, and the shopping cart.

- Users can return to previous pages or proceed to the checkout from this page.

# Discussing the JavaScript functionality:

### Main.js

enhance the accessibility and usability of a web application by providing features such as high contrast mode, larger text, dark mode, and speech synthesis for reading screen content. It also includes interactive elements like a portfolio filter and an auto-scrolling carousel. Let's break down the key parts and explain any complex concepts:

#### Window Load Event

JavaScript

window.onload = function () {

};

Executes code when the webpage fully loads. It checks if certain user preferences (like highContrast, largerText, and darkMode) are set in the browser's localStorage and applies those settings by calling respective functions. It also calls toggleFloatingMicrophone() to display or hide a floating microphone based on user settings.

- localStorage: A web storage object that stores data with no expiration date. This script uses localStorage to remember user preferences across sessions.

- toggle functions: These functions (toggleHighContrast(), toggleLargerText(), toggleDarkMode()) are called to apply the respective modes by adding or removing CSS classes to various elements.

Toggle Feature Functions

- High Contrast and Dark Mode: Both functions manipulate the DOM by toggling CSS classes (high-contrast or dark-mode) on various elements to switch to high contrast or dark mode. They target the body, sidebar, main content, headings (h2), and primary buttons. This approach ensures the visual aspects are immediately and uniformly applied or removed across multiple parts of the page.

- Larger Text: Simplifies text scaling by toggling a larger-text class on the body element, likely affecting font sizes site-wide.

Speech Synthesis for Screen Reading

This section includes functions to toggle reading the screen's content aloud using the Web Speech API's speech synthesis capabilities:

- toggleReadScreen(): Switches between reading and stopping.

- readScreenContent() and stopReading

(): Manage the initiation and cessation of speech synthesis. When reading starts, it grabs the entire text content of the body element, creates a SpeechSynthesisUtterance object with this content, and uses the speechSynthesis.speak() method to start speaking. Stopping the speech is done via speechSynthesis.cancel(). This functionality enhances accessibility by assisting users who benefit from auditory content consumption.

Portfolio Filter Section

This script adds interactive filtering to a portfolio section. It listens for clicks on filter options and then shows or hides portfolio items based on the selected filter:

- Event Listener: Added to each filter button to detect clicks.

- Filter Application: When a filter is clicked, the script removes the filter-active class from all filters, applies it to the clicked filter, and then displays only the portfolio items that match the selected filter. The items are also shuffled using Array.sort() with a random comparator function, introducing a dynamic, randomized presentation of items.

Carousel Functionality

This part initializes a carousel component (presumably using Bootstrap or a similar library) with an auto-scroll feature, controlled by the interval option. This is a common feature in web applications for cycling through a series of content (like images) automatically.

Floating Microphone Toggle

This function shows or hides a floating microphone icon based on a user setting stored in localStorage. This feature is likely part of a voice control or speech recognition system, indicating whether the user has enabled voice commands.

#### Progressbar.js

The code includes event listeners for document load and various user actions to update the progress bar's visual representation and status based on the shopping process's current step. Let's break down and explain the components and complexities involved:

Progress Bar Basic Update Mechanism

1. Initialization: Sets the initial step (currentStep) at which the user starts, catering to the user journey within the shopping cart process.

2. Updating the Progress Bar: Implements the updateProgressBar function, which calculates the progress percentage based on the current step and updates the progress bar's width, aria-valuenow attribute, and text content to reflect the current step's description. This provides visual and accessible feedback on the shopping process's progress.

Event Listeners and Dynamic Update Logic

- Event Listeners: The code listens for specific user actions (like clicking the "items correct" button, filling in billing information, selecting a payment method, or inputting payment info) to advance the progress bar.

- Validation Checks: For steps involving form inputs (billing and payment information), the code checks if all required fields are filled before advancing, ensuring users complete necessary information before proceeding.

ProgressBar Color Update

- The enhanced version of the updateProgressBar function includes logic to dynamically update the progress bar's color based on the current step, using CSS class manipulation. This visual cue further enhances user experience by providing immediate feedback on the phase of the shopping process.

Session Storage for Progress Tracking

- Uses sessionStorage to remember the current step in the shopping process if the user navigates away from the page, allowing for a persistent user experience across different pages of the application.

Progress Bar for Specific Actions

- Specific Actions: Includes logic for updating the progress bar and step description when adding an item to the cart and on page load for specific scenarios (e.g., navigating directly to the basket page). This tailored approach allows the application to adapt the progress bar based on user navigation patterns and actions.

Complexities and Interactions

- Interactions Between Components: The code demonstrates how to effectively manage interactions between HTML elements (progress bar, forms) and user actions, ensuring a dynamic and responsive UI.

- Accessibility Considerations: By updating the aria-valuenow attribute and providing textual content changes, the code enhances the accessibility of the progress bar, making the shopping process more inclusive for users with assistive technologies.

- CSS Class Management for Visual Feedback: Dynamically changing the progress bar's color based on the current step illustrates an effective use of CSS class manipulation to provide visual feedback to the user, enhancing the UX/UI of the web application.

Overall, these snippets demonstrate a comprehensive approach to managing a progress bar in a web-based shopping cart process, emphasizing dynamic updates based on user actions, accessibility, and visual feedback.

### Settings.js:

accessibility settings (e.g., high contrast, larger text, and dark mode), a screen reader, draggable elements, and dynamic user notifications. It also demonstrates the use of the Web Speech API, local storage for persistence of user settings, and DOM manipulation for interactive elements. Let's explore some of the key parts and elaborate on complex concepts:

Key Press and Accessibility Settings

- setupKeyPressEvent(): Adds an event listener for the "keydown" event to the entire document. It checks if the Enter key is pressed and triggers a function to highlight parts of the page, enhancing keyboard navigation.

- checkAndApplySetting(): Checks local storage for user-defined accessibility settings (high contrast, larger text, dark mode) and applies them by toggling respective CSS classes on elements. This function illustrates the application's effort to remember user preferences across sessions.

Toggle Functions for Accessibility Features

- toggleClassAndSaveSetting(): A versatile function that toggles a specified CSS class on an element and updates the corresponding setting in local storage. This demonstrates a pattern of changing the application's state and ensuring persistence.

Screen Reader Functionality

- Utilizes the Web Speech API to provide text-to-speech functionality. The toggleReadScreen, startReading, and stopReading functions control the initiation and termination of speech synthesis, providing an accessibility feature that reads out the content of the webpage.

Voice Control and Draggable Elements

- setupVoiceControl() and setupDraggableMicrophone(): These functions demonstrate interactive and responsive design features, such as toggling the visibility and functionality of a floating microphone icon based on user preferences. The draggable microphone feature is particularly notable for its implementation of drag-and-drop interaction without relying on external libraries.

Dynamic Notifications and Highlighting

- showNotification(): Shows dynamic notifications based on user interactions, such as enabling or disabling accessibility features. This function creates a new notification element, styles it, and appends it to a container, showcasing real-time feedback to user actions.

- highlightPartsOfPage(): Cycles through elements (like headings and buttons) to highlight them one at a time, based on key presses. This feature is aimed at improving keyboard navigation and accessibility.

Complex Concepts and Techniques

- Event Delegation and Manipulation: The script employs event listeners for various user actions (e.g., key presses, changes in settings) and manipulates the DOM in response, demonstrating a dynamic and interactive web application design.

- Local Storage for Persistence: Uses local storage to persist user preferences across sessions, showcasing a method for creating a more personalized and user-friendly web experience.

- Web Speech API: Illustrates the use of modern web APIs to enhance accessibility, providing an example of how web applications can be made more inclusive for users with different needs.

- CSS Class Toggling for Style Changes: Shows how JavaScript can be used to dynamically alter the appearance of a webpage by adding or removing CSS classes based on user interactions or preferences.