GetSetUp Accessibility Conformance Report

WCAG Edition

(Based on VPAT® Version 2.5Rev)

Name of Product/Version: GetSetUp Web Application (unified-app) — getsetup.io

Report Date: March 2026

Product Description: GetSetUp is a web-based platform that provides live and on-demand educational classes for older adults. The application includes class discovery and browsing, live interactive sessions with video and real-time chat, video-on-demand playback with subtitle support, scheduling, bookings, user accounts, an AI assistant (Helen), partner landing pages, and search/filter functionality. The platform is available in 18 languages: English (US), Spanish (US), Hindi (India), Chinese (Singapore), Malay (Singapore), Tamil (Singapore), Japanese, Arabic (Saudi Arabia), Bengali (Bangladesh), Chinese (Hong Kong), French, German, Italian, Korean, Polish, Russian, Tagalog (Philippines), and Ukrainian.

Contact Information: accessibility@getsetup.io

Notes:

- This report covers the consumer-facing website at getsetup.io, including the main site experience (class browsing, search, scheduling, accounts) and the live session/VOD experience (video player, chat, session overlays).
- This report does **not** cover embedded (iframe) partner experiences, admin interfaces, or third-party integrations (Zoom meetings, external email links).
- The product is a web application built with Next.js 16, React 19, and Tailwind CSS. It is designed for use with modern web browsers (Chrome, Firefox, Safari, Edge) on desktop and mobile devices.

Evaluation Methods Used:

- Static code analysis with `eslint-plugin-jsx-a11y` (28 rules enforced as errors) integrated into the development workflow and CI pipeline.
- Automated accessibility testing using `axe-core` (v4.11.1) via `vitest-axe` for component-level testing and `@axe-core/playwright` (v4.11.0) for end-to-end testing.
- Manual code review of ARIA attributes, semantic HTML structure, keyboard event handling, focus management, and color usage across all route groups and shared components.
- Functional testing of keyboard navigation (Tab, Shift+Tab, Arrow keys, Escape, Enter, Home, End) across interactive components including menus, modals, forms, and chat.
- Review of responsive behavior across viewport sizes from 320px to 1536px+.
- Verification of `prefers-reduced-motion` media query support.

Applicable Standards/Guidelines

This report covers the degree of conformance for the following accessibility standard/guidelines:

Standard/Guideline	Included In Report		
Web Content Accessibility Guidelines 2.0	Level A: **Yes**	Level AA: **Yes**	Level AAA: **No**
Web Content Accessibility Guidelines 2.1	Level A: **Yes**	Level AA: **Yes**	Level AAA: **No**

Terms

The terms used in the Conformance Level information are defined as follows:

- **Supports:** The functionality of the product has at least one method that meets the criterion without known defects or meets with equivalent facilitation.
- **Partially Supports:** Some functionality of the product does not meet the criterion.
- **Does Not Support:** The majority of product functionality does not meet the criterion.
- **Not Applicable:** The criterion is not relevant to the product.

WCAG 2.1 Report

Note: When reporting on conformance with the WCAG 2.x Success Criteria, they are scoped for full pages, complete processes, and accessibility-supported ways of using technology as documented in the WCAG 2.1 Conformance Requirements.

Table 1: Success Criteria, Level A

Criteria	Conformance Level	Remarks and Explanations
1.1.1 Non-text Content (Level A)	Partially Supports	Images throughout the application include alt attributes. Decorative icons (Lucide icon library) are marked with `aria-hidden="true"`. The GetSetUp logo has meaningful alt text. Content images provided by the CMS (class thumbnails, guide photos) include alt text when supplied by content creators; some CMS-provided images may have generic or missing alt text depending on content authoring practices.
1.2.1 Audio-only and Video-only (Prerecorded) (Level A)	Partially Supports	Prerecorded video content includes video playback with subtitle/caption track support (WebVTT format). Written class descriptions accompany all video content. Full text transcripts are not currently published alongside all

Criteria	Conformance Level	Remarks and Explanations
		prerecorded video content.
1.2.2 Captions (Prerecorded) (Level A)	Supports	All prerecorded class content includes English captions (WebVTT format). The VOD player supports multi-language subtitle tracks with automatic locale matching and English fallback. Locale-specific captions are produced as the platform rolls out to additional languages.
1.2.3 Audio Description or Media Alternative (Prerecorded) (Level A)	Partially Supports	Written class descriptions, class notes, and guide information accompany prerecorded video content, providing a media alternative. Formal audio description tracks are not provided for prerecorded video.
1.3.1 Info and Relationships (Level A)	Supports	Semantic HTML5 elements are used throughout: `<header>`, `<nav>`, `<main>`, `<footer>`, `<section>`, `<article>`. Heading hierarchy is maintained (h1 through h6 without skipping levels). Form inputs are associated with labels via `htmlFor`/`id`. Lists use `<ul>`/`<ol>` markup. ARIA roles applied to custom widgets: `role="menu"`, `role="tab"`, `role="tabpanel"`, `role="dialog"`, `role="log"`, `role="alert"`, `role="status"`. Data relationships conveyed through table markup and list structures.
1.3.2 Meaningful Sequence (Level A)	Supports	Content follows a logical DOM order matching the visual presentation. CSS flexbox and grid are used for layout without altering the meaningful reading sequence. Mobile-responsive layouts maintain logical content order across breakpoints.
1.3.3 Sensory Characteristics (Level A)	Supports	Instructions and UI cues do not rely solely on sensory characteristics (shape, color, size, visual location, orientation, or sound). Chat connection status indicators use both color and text labels (“Connected” / “Connecting”). Form errors are identified by descriptive text, not color alone.
1.4.1 Use of Color (Level A)	Supports	Color is not the sole means of conveying information. Chat connection status uses a colored dot paired with text label (explicitly implemented per WCAG 1.4.1). Form validation uses text error messages alongside color indicators. Interactive elements are identifiable by text labels, not color alone.
1.4.2 Audio Control (Level A)	Supports	Video players provide mute and volume controls. No audio auto-plays on page load. Live session and VOD playback require explicit user activation.
2.1.1 Keyboard	Supports	All interactive elements are operable via keyboard. Header

Criteria	Conformance Level	Remarks and Explanations
(Level A)		dropdown menus support Arrow Up/Down, Home, End, and Escape keys per WAI-ARIA Authoring Practices. Modal dialogs and drawers implement focus traps with Tab/Shift+Tab cycling. Chat input supports Enter to send messages. Skip-to-main-content link is the first focusable element on every page. Video player controls are keyboard accessible.
2.1.2 No Keyboard Trap (Level A)	Supports	Focus traps in modals and drawers allow the user to exit via the Escape key. Tab and Shift+Tab cycle within trapped containers without escaping to underlying content. Focus is returned to the triggering element when a modal or drawer is closed. No keyboard traps exist in standard navigation flows.
2.1.4 Character Key Shortcuts (Level A 2.1)	Not Applicable	The application does not implement single character key shortcuts.
2.2.1 Timing Adjustable (Level A)	Supports	No time limits are imposed on user interactions. Session schedules are informational and do not restrict content access. Authentication tokens have extended expiry periods. There are no auto-logout timers or timed form submissions.
2.2.2 Pause, Stop, Hide (Level A)	Supports	Animated content (loading skeletons, transitions) respects the `prefers-reduced-motion` media query, which effectively disables animations. No auto-scrolling or auto-updating content exists that cannot be controlled by the user. Chat messages persist in the log and do not auto-hide.
2.3.1 Three Flashes or Below Threshold (Level A)	Supports	No content flashes more than three times per second. UI animations consist of smooth transitions (fade, slide) and pulse-style loading indicators, all below the flash threshold. Video content is user-controlled and not produced by the application itself.
2.4.1 Bypass Blocks (Level A)	Supports	A “Skip to main content” link is the first focusable element on every page, visually hidden until focused. It targets `#main-content` on the `<main>` element. The skip link text is translated in all 18 supported locales. Semantic landmark regions (`<header>`, `<nav>`, `<main>`, `<footer>`) enable screen reader landmark navigation.
2.4.2 Page Titled (Level A)	Supports	Each page has a descriptive `<title>` set via Next.js metadata. Titles include page-specific content (e.g., class name, category name, “Schedule”, “Search Results”). Titles are provided in the active locale language.

Criteria	Conformance Level	Remarks and Explanations
2.4.3 Focus Order (Level A)	Supports	Focus order follows the logical reading sequence of the page. A `FocusManager` component resets focus to the main content area on route changes, preventing users from losing their place after navigation. Modal focus traps maintain logical tab order within the dialog. No positive `tabindex` values are used (enforced by ESLint rule `jsx-a11y/tabindex-no-positive`).
2.4.4 Link Purpose (In Context) (Level A)	Supports	Navigation links use descriptive text (e.g., “Schedule”, “Categories”, “Search”). Class cards link to detail pages with the class title as link text. Icon-only buttons include `aria-label` attributes. External links are identified by visual indicators and appropriate `target`/`rel` attributes.
2.5.1 Pointer Gestures (Level A 2.1)	Supports	No functionality requires multipoint or path-based gestures. All interactions can be accomplished via single-point activation (click or tap).
2.5.2 Pointer Cancellation (Level A 2.1)	Supports	Interactive elements use standard click events (activated on pointer up, the default browser behavior). No actions are triggered on pointer down. No drag-and-drop functionality is required.
2.5.3 Label in Name (Level A 2.1)	Supports	Visible button labels match their accessible names. Form inputs have matching visible labels and programmatic names via `htmlFor`/`id` associations. Where visible labels are not present (icon-only buttons), `aria-label` provides the accessible name and the label is consistent with the icon’s visual meaning.
2.5.4 Motion Actuation (Level A 2.1)	Not Applicable	No functionality is operated by device motion or user gestures detected by device sensors.
3.1.1 Language of Page (Level A)	Supports	The `<html>` element’s `lang` attribute is dynamically set to match the active locale across all 18 supported languages (e.g., `en-US`, `es-US`, `zh-SG`, `ar-SA`, `ja-JP`). An `HtmlLangSetter` component updates the attribute on every route change. RTL layout is applied for Arabic (ar-SA).
3.2.1 On Focus (Level A)	Supports	No context changes occur when any UI component receives focus. Dropdown menus require explicit activation via click or Enter/Space key. Search suggestions appear in a dropdown without navigating away from the page.
3.2.2 On Input (Level A)	Supports	Changing the setting of any UI component does not automatically cause a context change unless the user is advised beforehand. Form submissions require explicit

Criteria	Conformance Level	Remarks and Explanations
		button activation. The language switcher navigates to the selected locale page (an expected and conventional behavior).
3.3.1 Error Identification (Level A)	Supports	Form validation errors are displayed with descriptive text messages. Errors are announced to assistive technology via `aria-live="assertive"` with `role="alert"`. Invalid fields are marked with `aria-invalid="true"`. Error messages are programmatically linked to their associated input fields via `aria-describedby`.
3.3.2 Labels or Instructions (Level A)	Supports	All form inputs have associated `<label>` elements linked via `htmlFor`. Required fields are marked with `aria-required="true"`. Input purpose is clear from label text. Placeholder text supplements labels but does not replace them. Enforced by ESLint rule `jsx-a11y/label-has-associated-control`.
4.1.1 Parsing (Level A)	Supports	Per the WCAG 2.1 Editorial Errata (September 2023), this criterion is always satisfied.
4.1.2 Name, Role, Value (Level A)	Supports	Custom components use appropriate ARIA roles: `role="menu"` for dropdown menus, `role="tab"`/`role="tabpanel"` for tabbed interfaces, `role="dialog"` with `aria-modal="true"` for modals (including Helen AI chat), `role="log"` with `aria-live="polite"` for chat message feeds, `role="toolbar"` with `aria-label` for filter pill groups, `role="status"` for loading indicators. Component states are communicated via `aria-expanded`, `aria-selected`, `aria-pressed` (filter pills), and `aria-checked`. Radix UI primitives (Dialog, Tabs) provide built-in ARIA support. All interactive elements have programmatically determinable names.

Table 2: Success Criteria, Level AA

Criteria	Conformance Level	Remarks and Explanations
1.2.4 Captions (Live) (Level AA)	Does Not Support	Live class sessions streamed via Amazon IVS do not currently provide real-time captions. This is a known gap; integration of a live captioning service is planned for a future release.
1.2.5 Audio Description (Prerecorded) (Level AA)	Partially Supports	Written class descriptions, class notes, and guide information provide supplemental context for video content. Dedicated audio description tracks synchronized to prerecorded video are not currently provided.
1.3.4 Orientation (Level AA 2.1)	Supports	Content displays and functions in both portrait and landscape orientations. No CSS or JavaScript orientation lock is applied. Responsive layouts adapt to orientation changes.
1.3.5 Identify Input Purpose (Level AA 2.1)	Supports	Login and signup forms use appropriate HTML input types (`email`, `password`, `text`). Autocomplete attributes are applied to relevant fields (name, email). The programmatic purpose of form fields is determinable from input type and label context.
1.4.3 Contrast (Minimum) (Level AA)	Supports	The primary color palette is designed for accessible contrast ratios. Primary text (navy #181b56 on white backgrounds) exceeds the 4.5:1 minimum for normal text. Inverted text (white on navy) meets the same ratio. Dark mode palette maintains equivalent contrast. Contrast is enforced through design system tokens (CSS custom properties) and verifiable via axe-core testing.
1.4.4 Resize text (Level AA)	Supports	Text is rendered using relative units (rem). Browser text zoom up to 200% does not cause loss of content or functionality. Responsive CSS adapts layout to accommodate enlarged text. No content is clipped or overlapped at 200% zoom.
1.4.5 Images of Text (Level AA)	Supports	Text is rendered as styled HTML text throughout the application. The only images containing text are the GetSetUp logo (used for brand identification, exempt under WCAG) and user-uploaded content images. No informational content relies on images of text.
1.4.10 Reflow (Level AA 2.1)	Supports	Content reflows at 320px CSS width (equivalent to 400% zoom on a 1280px viewport) without requiring horizontal scrolling. Mobile-first responsive design uses Tailwind CSS breakpoints (sm: 640px, md: 768px, lg: 1024px, xl: 1280px). Single-column layouts are used at narrow viewports.

Criteria	Conformance Level	Remarks and Explanations
1.4.11 Non-text Contrast (Level AA 2.1)	Partially Supports	UI components (buttons, form inputs, icons) generally maintain a 3:1 contrast ratio against adjacent backgrounds. Focus indicators use a 2px solid outline. Some interactive elements in the live session chat overlay, which floats over video content, may have reduced contrast depending on the underlying video frame. The chat bubble redesign (white bubbles over transparent overlay) improves readability but overlay control elements may vary.
1.4.12 Text Spacing (Level AA 2.1)	Supports	Content adapts to user-defined text spacing overrides (line height, letter spacing, word spacing, paragraph spacing). No content or functionality is lost when spacing is increased per WCAG thresholds. Tailwind CSS utility classes use flexible box models that accommodate spacing changes.
1.4.13 Content on Hover or Focus (Level AA 2.1)	Supports	Tooltip and popover content triggered on hover or focus is dismissible via the Escape key. Hover content remains visible while the pointer is over it. Users can move the pointer to the additional content without it disappearing. Dropdown menus remain open until explicitly dismissed.
2.4.5 Multiple Ways (Level AA)	Supports	Users can locate content via: primary navigation menu, search functionality with real-time suggestions, category browsing, schedule/calendar view, direct URL entry, and sitemap. Multiple navigation paths are available to reach all content areas.
2.4.6 Headings and Labels (Level AA)	Supports	Pages use descriptive headings in a proper hierarchy (single h1 per page, sequential h2-h6 without skipping). Form labels clearly describe their associated input's purpose. Section headings accurately describe the content that follows. Heading content is enforced by ESLint rule `jsx-a11y/heading-has-content`.
2.4.7 Focus Visible (Level AA)	Supports	A global `:focus-visible` style is applied via CSS, providing a 2px solid outline on all focusable elements. The focus indicator has `scroll-margin-top: 5rem` to prevent the fixed header from obscuring focused elements. The `:focus:not(:focus-visible)` rule preserves clean appearance for mouse users while maintaining visible indicators for keyboard users. Custom focus styles are applied to interactive components (buttons, inputs, tabs) using Tailwind's `focus-visible:ring-*` utilities.
3.1.2 Language of Parts (Level AA)	Partially Supports	The page-level language is correctly set via the `<html lang>` attribute for all 18 supported locales. Content within the page is primarily in a single language matching the page locale. Some content returned from the CMS or GraphQL API (class titles, descriptions) may be in a different

Criteria	Conformance Level	Remarks and Explanations
		language than the page locale and may not have individual `lang` attributes applied to those elements.
3.2.3 Consistent Navigation (Level AA)	Supports	The header navigation appears consistently on every page within the main site, with the same items in the same order. The footer is consistent across all pages. Navigation patterns remain identical across all locale variants. Mobile and desktop navigation contain the same items, adapted for their respective layouts.
3.2.4 Consistent Identification (Level AA)	Supports	Components with the same function are identified consistently throughout the application. Search functionality uses the same icon and placement. Close buttons use the same X icon and aria-label. Navigation items maintain consistent labeling. The design system (shadcn/ui components) ensures visual and semantic consistency.
3.3.3 Error Suggestion (Level AA)	Supports	Form validation provides specific, actionable error messages (e.g., “Please enter a valid email address”, “Password is required”). Login failures indicate whether the issue is with the email or password. Password reset flow provides guidance on next steps. Error messages suggest how to correct the input.
3.3.4 Error Prevention (Legal, Financial, Data) (Level AA)	Supports	Booking actions require explicit user confirmation. Cancel booking presents a confirmation modal before processing. Account changes (password reset) require verification. No irreversible actions are performed without a user confirmation step. The application does not process legal or financial transactions directly.
4.1.3 Status Messages (Level AA 2.1)	Supports	Status messages are communicated to assistive technologies without receiving focus. Chat connection status uses `aria-live="polite"` with `role="status"`. The Helen AI chat modal announces new messages via `role="log"` with `aria-live="polite"`, connection errors via `role="alert"`, and loading states via `role="status"`. Form errors are announced via `aria-live="assertive"` with `role="alert"`. Search and filter results update via `aria-live="polite"` regions with result count announcements. Typing indicators in chat are announced via `role="status"` with screen-reader-only text.

Legal Disclaimer

This Accessibility Conformance Report is based on evaluation of the GetSetUp web application (unified-app) as of March 2026. Conformance levels reported reflect the state of the product at

the time of evaluation. Changes to the product, content, or underlying technologies may affect accessibility. GetSetUp is committed to ongoing accessibility improvement and welcomes feedback at accessibility@getsetup.io.