**web app accessible**

**Introductions’s**

* Making a website or web app accessible improves the user experience for people with disabilities and for all users as well.
* The benefits of making your website accessible to people with disabilities will extend to all users including non-disabled persons.
* I can't emphasize the importance of accessibility enough. If you don't pay attention to it right from the beginning of your project, **you risk turning accessibility into a burden and an expensive one in the future** if you start retrofitting
* if our website is accesiable for disabled persons.
* /PWDs people also then it be **good sign for SEO and google ranking** becoz we get more traffic and all the people whether disabled persons

/PWDS or ordinary people everyone is now able to access our website

* Besides, it won't be good for you or your business if you are accused of discrimination because your site is inaccessible to PWDs**. It will damage your brand and reputation,**
* Some coutries have strick rule that every website should implement the Accesibilities features in their website otherwise website will be band. Some countries have legislation that requires websites to be accessible to people with disabilities.

**Accessibility standards and guidelines**

* There are several different accessibility standards and guidelines. The most notable and widely recognized standards were developed by the World Wide Web Consortium (W3C), through its Web Accessibility Initiative (WAI)
* Web Content Accessibility Guidelines (WCAG) 2.1 WCAG is one of the internationally recognized standards for web content accessibility. It was developed by W3C through a participatory process with input from a number of individual and institutional stakeholders from around the world.
* Authoring Tools Accessibility Guidelines (ATAG) 2.0 is a set of accessibility guidelines that you can use for designing tools for authoring web content. This guideline helps you make sure that you produce authoring tools that are accessible to people with disabilities. The tools should, in turn, help authors create accessible web content,

**Accessibility Tools Built for React**

* **eslint-plugin-jsx-a11y**.  
  You can use this tool for linting accessibility issues on JSX elements in your React projects. You can use it in conjunction with tools such as eslint for linting your project for accessibility standards right in your text editor. Since it is distributed via npm, you can install it by running the command below in your project:
* Any React project which you've created using create-react-app comes with this tool already configured - but it has only a subset of the configurable accessibility rules enabled by default. You can enable additional rules by creating an .eslintre configuration file in your project and adding the following code to it. The code below activates the recommended rules:



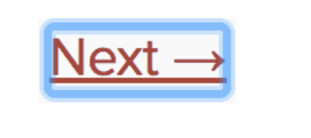
* **axe accessibility linter**   
  is a Visual Studio Code extension that you can use for linting React, HTML, Vue, and Markdown for some common accessibility defects. It checks for accessibility issues in .js, .jsx, .ts, .tsx, .vue, .html, .ht m, .md and markdown files. You don't need configuration to start using axe accessibility linter after installation. You install it from VS code marketplace and it automatically starts linting compatible files for accessibility defects out of the box without the need for additional configuration. For a complete list of rules used by axe accessibility linter, check the extension page on VS Code marketplace.
* Using tools such as eslint-plugin-jsx-a11y, axe accessibility linter, and axecore-react in your project will go a long way in helping you develop more accessible and inclusive products using React. Though they come in handy, the tools mentioned here will only flag a certain percentage of accessibility defects - mainly those that are possible to detect programmatically. So it's really important to integrate automated testing, manual testing, and actual user testing in your development because automated testing alone may not be able to detect even 50 percent of accessibility issues in your project.

**Steps to Integrating into our projects**

* We must need to add **sufficient contrast ratio** in our project, adding a sufficient contrast ratio is not only helpful for people

with low vision, color blindness, or cognitive impairment but it's also helpful to people working in different lighting conditions.

* Similarly adding an **alt attribute** with appropriate text will help people using screen readers as well as those with slow internet connections when the image fails to load or takes too long to load.
* WAI-ARIA   
  The Web Accessibility Initiative - Accessible Rich Internet Applications document contains techniques for building fully accessible JavaScript widgets.
* 
* Semantic HTML   
  if ur reading these doc I think u know the meaning of Semantic HTML,   
  Semantic HTML is the foundation of accessibility in a web application. Using the various HTML elements to reinforce the meaning of information in our websites will often give us accessibility for free
* 
* Keyboard focus and focus outline Keyboard focus refers to the current element in the DOM that is selected to accept input from the keyboard. We see it everywhere as a focus outline similar to that shown in the following image:



* Also use landmark elements and roles, such <main> tag and ,<aside> tag to demarcate page regions as assistive technology allow the user to quickly navigate to these sections.
* Mouse and pointer events Ensure that all functionality exposed through a mouse or pointer event can also be accessed using the keyboard alone. Depending only on the pointer device will lead to many cases where keyboard users cannot use your application.
* Other Points for Consideration Setting the language Indicate the human language of page texts as screen reader software uses this to select the correct voice settings:
* Always try to set the page title it will be so good for disabilities peoples
* chromevox integrated screen reader on Chromebooks and is available as an extension for Google Chrome.
* Also note the added aria-\* props to support screen-reader users.
* useRef() and useEffect() are somewhat advanced features, and you should be proud of yourself for using them! . Remember: our app would have been inaccessible to keyboard users without them!
* Labels are critical for blind users, user with low vision, users with mobility disabilities and users with memory loss. Missing labels will make a form inaccessible for many users

# 

**(WCAG) and (ADA) accessibility guidelines**

* Web Content Accessibility Guidelines (WCAG) are a series of accessibility guidelines from the World Wide Web Consortium (W3C) that explain how to make web content accessible to people with disabilities.
* Web Content Accessibility Guidelines (WCAG) 2 is developed through the [W3C process](https://www.w3.org/WAI/standards-guidelines/w3c-process/) in cooperation with individuals and organizations around the world, with a goal of providing a single shared standard for web content accessibility that meets the needs of individuals, organizations, and governments internationally, content accessible to people with disabilities.
* Web Content Accessibility Guidelines 2.0 and ADA are organized into three levels  
  **1) Level A** — deals with the most basic web accessibility features  
  **2) Level AA** — signifies the biggest and most common barriers for disabled users  
  **3) L evel AAA** — addresses the highest (and most complex) level of web accessibility  
    
  or in another words
* **Level A:** Your website is only accessible by some users
* **Level AA:**Your website is accessible by almost all users
* **Level AAA:**Your website is accessible by all users
* The web content accessibility guidelines are categorized under these four principles and they are, Perceivable, Operable, Understandable and Robust.
* **Perceivable** Content should be perceivable through sight, sound (hearing) and touch  
  Content should be transformable from one format into another easily eg : text must transfer to audio ( text-to-speech screen reader\_)
* **Operable**  
  User interface components and navigation must be operable.  
  Make all functionality available from a keyboard.
* **Understandable**Websites Information  should in a way that users understand   
  Make text content readable and understandable.  
  Ensure web pages appear and operate in predictable ways.
* **Robust**Maximize compatibility with current and future user agents, including assistive technologie, it means our website need to be render in all device whether its used by disabilities person (note disability person may use some special devices ) so our website need to render in all devices
* ADA compliance is short for the Americans with Disabilities Act Standards for Accessible Design. What that means is that all electronic information and technology—i.e, your website—must be accessible to those with disabilities. ADA compliance is often confused with 508 compliance.
* If we don’t follow the above rule then  you could still end up paying thousands of dollars in lawsuits if you're website isn't accessible to everyone.
* And if we don’t follow their rule we may definitely lose the no no of customer and traffic to our website becoz huge no of people in the world are disabilities person who are using internet