**Keyboard Navigation**

Many users will navigate your site with only the keyboard. It is critical that your web apps accommodate this style of navigation. A few things to consider while building your web app is to always enable interactive items to show focus. Focus is the primary visual que to display where a user is interacting with your web app. Another great feature is to include skip links. Skip links enable the user to bypass the content of the introduction of the site and get to the main content.

Below is a simple example of how a user might navigate a site with a keyboard. Use TAB and SHIFT + TAB to move right and left respectfully.

1. Try to navigate to the input field using only the keyboard
2. Start the timer
3. Spell your name using the virtual keyboard. You must time the selection of rows and columns using the enter key to submit.
4. Spell your first name using the virtual keyboard and tweet me @councit and tell me your time!

**Keyboard Navigation**

Many users will navigate your site with only the keyboard. It is critical that your web apps accommodate this style of navigation. A few things to consider while building your web app is to always enable interactive items to show focus. Focus is the primary visual que to display where a user is interacting with your web app. Another great feature is to include skip links. Skip links enable the user to bypass the content of the introduction of the site and get to the main content.

Below is a simple example of how a user might navigate a site with a keyboard. Use TAB and SHIFT + TAB to move right and left respectfully.

1. Try to navigate to the start button using only the keyboard
2. Navigate to the key buttons and spell your name using the virtual keyboard. Use TAB to move right and SHIFT+TAB to move left. Press enter to submit to the entry field.
3. After you are done. Navigate to the stop button and press enter.
4. Send me your best time [**@Councit**](https://twitter.com/councit), and let me know how you did!

**Keyboard Navigation**

Many users will navigate your site with only the keyboard. It is critical that your web apps accommodate this style of navigation. A few things to consider while building your web app is to always enable interactive items to show focus. Focus is the primary visual que to display where a user is interacting with your web app. Another great feature is to include skip links. Skip links enable the user to bypass the content of the introduction of the site and get to the main content.

Below is a simple example of how a user might navigate a site with a keyboard. Use TAB and SHIFT + TAB to move right and left respectfully.

1. Try to navigate to the start button using only the keyboard
2. Navigate to the key buttons and spell your name using the virtual keyboard. Use TAB to move right and SHIFT+TAB to move left. Press enter to submit to the entry field.
3. After you are done. Navigate to the stop button and press enter.
4. Send me your best time [**@Councit**](https://twitter.com/councit), and let me know how you did!

Start

Focus Control:

Focus control is useful when you want to enable a user to tab appropriately through a component. Here is an example of a modal that when tabbed through returns to the first modal focus item.

Tools:

Listing several extensions and web applications, Jon goes through tools that help in creating accessible web projects.

https://chrome.google.com/webstore/detail/accessibility-developer-t/fpkknkljclfencbdbgkenhalefipecmb?hl=en

http://tenon.io

https://addons.mozilla.org/en-US/firefox/addon/accessibility-evaluation-toolb/

http://wave.webaim.org/extension/

https://khan.github.io/tota11y/

https://addyosmani.com/a11y/

<https://github.com/reactjs/react-a11y>

|  |
| --- |
| **Lead Generator** - [github.com/councit/lead-manager-React-Django](https://github.com/councit/lead-manager-React-Django) |
| • Provides safe and secure customer repository with scalability in mind  • Robust product using technologies such as React, Django, and Redux  • Full stack application with developer friendly Rest API’s as data endpoints |
|  |
| **Digi-Check** - [github.com/councit/digi-check](https://github.com/councit/digi-check) |
| • Application that utilized Python AI to estimate hand-written digits  • User friendly GUI, created with Tkinter with the ability to style based on customer demand  • Learning models trained with Keras’s deep learning API |
|  |
| **Keto Council** - [Keto-Council.com](http://www.Keto-Council.com) |
| • Keto-inspired blog site, connecting a community of almost 400 health conscious users with great recipes  • Reliable backend created with Flask, front end utilizing JavaScript, HTML, and CSS  • Instagram bot created with Python to generate leads and followers to site |
|  |
| **Crwn-Clothing** - [crwn-council.herokuapp.com/](https://crwn-council.herokuapp.com/) |
| • Scalable e-commerce site to showcase high-end boutique clothing  • Clean reusable components and state management created with React and Redux  • Customer and site data validated and stored using Google Firebase |