Components in-Depth for My Project

* **Component Registration**-Components can be registered or created either globally or locally. Globally registered components can be included in any new Vue instance while locally registered components are only available when they are called upon in a code block. Locally registered components avoid the unnecessary JavaScript occupation when a component is not being used.
* **Props**-Props or properties are used to pass data from a parent component to its child components where components are arranged in a tree hierarchical manner.
* **Custom Events**-Vue can listen to events on elements and trigger certain functions to run when the events are called. Events are used to emit some events between parent and child components. They are case-sensitive so exact syntax must be always maintained.
* **Slots**-Slots can pass data outside of the regular parent-child interactions. Any type of content like HTML can be placed in new places.
* **Dynamic and Async Components**-Dynamic components can be used to cache user action or data so that when the user changes the state of a component, he or she does not lose their previous activity. Asynchronous components allow a browser to only load a component from the server when it’s necessary in order to display the webpage as quickly as possible.
* **Handling Edge Cases**-These are unique Vue instances where regular Vue rules do not apply, and out-of-the-box approaches can be implemented.

I will choose component registration and props for my projects because I want to break apart my website into reusable components so that I do not have to duplicate unnecessary code. When I create components, I will need props to pass data values between root Vue instances and the components.