

Prepared and Completed by Vaibhav Kansal

Below is a detailed documentation for the provided Vue component. This documentation explains the structure, purpose, and functionality of the component, along with how it uses template markup, data properties, methods, and scoped styling.

Component Documentation: SyllabusTable

Overview

The `SyllabusTable` component is designed to render a sidebar that lists several menu items. Each menu item can be expanded or collapsed to reveal a sub-menu. The component utilizes Vue's data binding and event handling to toggle the visibility of sub-menus, while also using scoped CSS to style the sidebar and its elements.

File Structure

The component is structured into three main sections:

1. **Template:** Defines the HTML structure.
 2. **Script:** Contains the component's logic, including data properties and methods.
 3. **Style:** Provides scoped CSS styles specific to the component.
-

Template Section

The template defines the layout of the sidebar and its elements.

```
<template>
  <div class="sidebar-container">
    <aside class="sidebar">
      <h2 class="explore">Explore All Previous Year Papers</h2>
      <nav>
        <ul class="space-y-2">
          <!-- Loop through each menu item -->
          <li v-for="item in menuItems" :key="item.text" class="dropdown
menu-item">
            <a
              href="#"
```

```

        class="block px-4 py-2 text-white rounded-md hover:bg-gray-700
flex items-center no-underline"
        @click="toggleDropdown(item)"
    >
    <!-- Dropdown arrow that rotates when the sub-menu is open -->
    <span
        class="dropdown-arrow"
        :class="{ 'rotate-90': item.isOpen }"
    >
        &#8250; <!-- Right arrow symbol ">" -->
    </span>
    <!-- Menu item text -->
    <span class="menu-item-text">{{ item.text }}</span>
</a>
<!-- Sub-menu: Only rendered when item.isOpen is true -->
<ul v-if="item.isOpen" class="dropdown-menu mt-2 space-y-2">
    <li v-for="subItem in item.subMenu" :key="subItem.text">
        <a
            :href="subItem.link"
            class="block px-4 py-2 text-white rounded-md hover:bg-gray-
700 no-underline"
        >
            {{ subItem.text }}
        </a>
    </li>
</ul>
</li>
</ul>
</nav>
</aside>
</div>
</template>

```

Key Points

- **v-for Loop:** The outer `` loops through `menuItems` defined in the data object. Each item represents a menu entry.
- **Key Binding:** Uses `:key="item.text"` to ensure each list item has a unique key.
- **Event Handling:** The `@click` event on the `<a>` tag calls the `toggleDropdown(item)` method to toggle the sub-menu.
- **Conditional Rendering:** The sub-menu `` is rendered only if `item.isOpen` is true, thanks to the `v-if` directive.
- **Dynamic Class Binding:** The arrow span uses `:class="{ 'rotate-90': item.isOpen }"` to apply a CSS rotation when the sub-menu is open.

Script Section

The script section defines the component's logic including data properties and methods.

```

<script>
export default {
  name: 'SyllabusTable',
  data() {
    return {
      // List of menu items with corresponding sub-menu entries
      menuItems: [
        {
          text: "All subject Syllabus",
          link: "#",
          isOpen: false,
          subMenu: [
            { text: "Sub Item 1", link: "#" },
            { text: "Sub Item 2", link: "#" },
          ]
        },
        {
          text: "Bihar Computer Teacher",
          link: "#",
          isOpen: false,
          subMenu: [
            { text: "Sub Item 1", link: "#" },
            { text: "Sub Item 2", link: "#" },
          ]
        },
        {
          text: "Intelligence Bureau (IB)",
          link: "#",
          isOpen: false,
          subMenu: [
            { text: "Sub Item 1", link: "#" },
            { text: "Sub Item 2", link: "#" },
          ]
        },
        {
          text: "CSIR SP ASO",
          link: "#",
          isOpen: false,
          subMenu: [
            { text: "Sub Item 1", link: "#" },
            { text: "Sub Item 2", link: "#" },
          ]
        },
        {
          text: "NAICL Assistant",
          link: "#",
          isOpen: false,
          subMenu: [
            { text: "Sub Item 1", link: "#" },
            { text: "Sub Item 2", link: "#" },
          ]
        },
        {
          text: "SSC CGL",
          link: "#",
          isOpen: false,
          subMenu: [

```

```

        { text: "Sub Item 1", link: "#" },
        { text: "Sub Item 2", link: "#" },
    ]
},
{
    text: "Bank SO",
    link: "#",
    isOpen: false,
    subMenu: [
        { text: "Sub Item 1", link: "#" },
        { text: "Sub Item 2", link: "#" },
    ]
},
],
};
},
methods: {
    /**
     * Toggles the sub-menu open/close state for a given menu item.
     * @param {Object} item - The menu item object.
     */
    toggleDropdown(item) {
        item.isOpen = !item.isOpen;
    },
},
};
</script>

```

Key Points

- **Data Property (`menuItems`):**
 - An array of objects where each object represents a menu item.
 - Each menu item includes:
 - `text`: The display text for the menu.
 - `link`: The hyperlink associated with the item (currently set to #).
 - `isOpen`: A boolean indicating whether the sub-menu is visible.
 - `subMenu`: An array of sub-menu items, each with its own `text` and `link`.
- **Method (`toggleDropdown`):**
 - A method that toggles the `isOpen` property for the clicked menu item, thus controlling whether the sub-menu is displayed.
 - It takes the menu item as a parameter and simply inverts its `isOpen` value.

Style Section

Scoped CSS styles are applied only to this component, ensuring that they do not interfere with other parts of the application.

```

<style scoped>
/* Container styling for the sidebar */

```

```

.sidebar-container {
  width: 250px;
  height: 50vh;
  background-color: whitesmoke;
  padding: 20px;
  border: 1px solid gray;
  overflow-y: auto; /* Enable vertical scrolling if content overflows */
}

/* Styling for the heading */
.explore {
  color: black;
  text-align: left;
  font-size: medium;
}

/* Remove default list styling */
ul {
  list-style-type: none;
  padding-left: 0;
}

/* Each menu item margin */
.menu-item {
  margin-bottom: 10px;
}

/* Sub-menu styling */
.dropdown-menu {
  list-style-type: none;
  padding-left: 0;
}

/* Dropdown arrow styling and transition for rotation */
.dropdown-arrow {
  margin-right: 10px;
  font-size: 18px;
  transition: transform 0.3s ease;
  text-decoration: none;
  color: black; /* Arrow color */
}

/* Rotate the arrow when the sub-menu is open */
.rotate-90 {
  transform: rotate(90deg);
}

/* Spacing for menu item text */
.menu-item-text {
  margin-left: 10px;
}

/* Base styling for dropdown links */
.dropdown a {
  display: flex;
  align-items: center;
  color: black;

```

```

    transition: color 0.3s ease;
    text-decoration: none;
}

/* Hover effect for dropdown links */
.dropdown a:hover {
    color: orange;
}

/* Styling for links in the sub-menu */
.dropdown-menu a {
    padding-left: 20px;
    color: black;
    text-decoration: none;
}

/* Hover effect for sub-menu links */
.dropdown-menu a:hover {
    color: orange;
}
</style>

```

Key Points

- **Container Styles:**
 - `.sidebar-container` sets the width, height, background color, padding, and border for the sidebar.
 - `overflow-y: auto` allows scrolling when the content exceeds the container's height.
- **Text and List Styles:**
 - The heading (`.explore`) and list (`ul`) styles ensure proper spacing and removal of default list styling.
- **Menu and Sub-Menu Styling:**
 - Each menu item (`.menu-item`) is given a margin for spacing.
 - The sub-menu (`.dropdown-menu`) inherits similar list styling properties.
- **Dropdown Arrow and Hover Effects:**
 - The `.dropdown-arrow` class includes a smooth rotation transition.
 - When a menu item is active (`item.isOpen`), the `rotate-90` class rotates the arrow.
 - Hover effects on both menu and sub-menu links change the text color to orange.

Usage

To use this component in your Vue application:

1. Import and Register the Component

In your parent component or main application file, import and register the `SyllabusTable` component.

```
2. import SyllabusTable from './path/to/SyllabusTable.vue';
3.
4. export default {
5.   components: {
6.     SyllabusTable,
7.   },
8.   // ...
9. };
```

10. Include the Component in Your Template

Use the component's tag in your template where you want the sidebar to appear.

```
11. <template>
12.   <div>
13.     <SyllabusTable />
14.   </div>
15. </template>
```

Summary

The `SyllabusTable` Vue component provides a functional sidebar with expandable menu items. Each menu item toggles a sub-menu when clicked. The component makes use of Vue's reactive data properties, event handling, conditional rendering, and scoped CSS to ensure that styles and functionality are encapsulated and maintainable.

This documentation should provide a clear understanding of the component's design and usage, enabling further customization or integration within your project.