B2V Technology

Web Development Projects

Connecting
Business
To
Technology

www.b2vtech.com

Table of Contents

Basic_1 Projects List

Responsive Navbar

Create a responsive navbar with a sidebar for a landing page.

Brief

Creating a responsive navbar with a sidebar often involves combining elements of a traditional top navbar with a sidebar that can be hidden or shown based on the screen size. Here's a step-by-step description of how you can create a responsive navbar with a sidebar

To Do

- ✓ Create a navbar
- ✓ Create a sidebar
- ✓ Create a homepage

Level 1

- Plan your navigation items and decide what should be in the top navbar and what should be in the sidebar.
- Consider using a recognizable icon (like a hamburger icon) for the sidebar toggle on smaller screens.
- Start designing your navbar for mobile devices first. Think about the layout, font sizes, and spacing suitable for smaller screens.

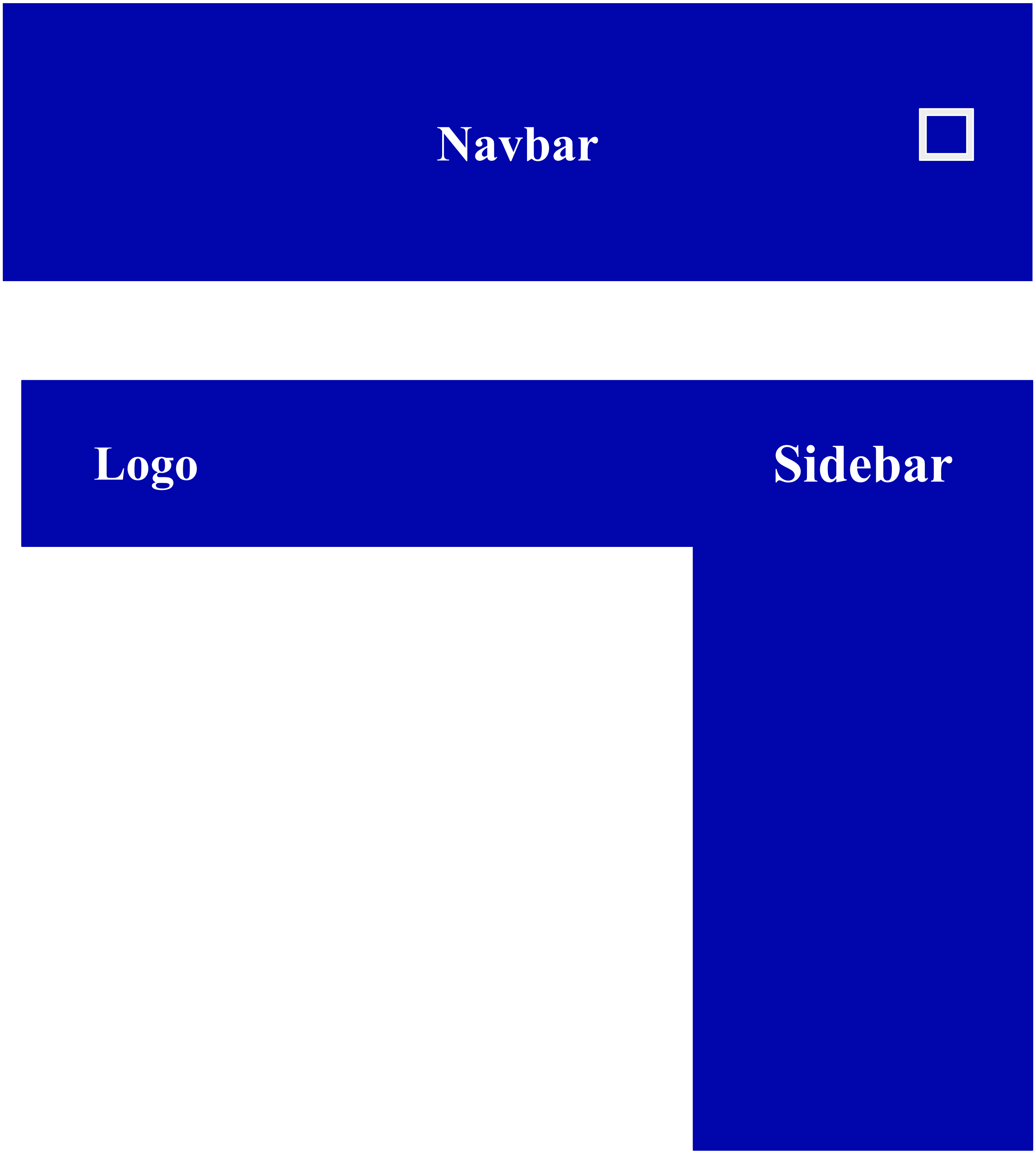
Level 2

- Include a simple top navbar with your logo and maybe a menu icon for the sidebar.
- Ensure that the top navbar elements are responsive and easy to interact with on touch devices.

Level 3

- Design a sidebar with links to various sections of your website. Make it visually appealing and easy to navigate.
- Initially, consider hiding the sidebar on larger screens and showing it only on smaller screens.
- Enhance your design for larger screens by potentially displaying both the top navbar and the sidebar simultaneously.

Layout Design



Static Route Navbar

Create a static route navbar in a React application involves setting up a navigation bar with links to different pages (routes) within your application.

Brief

Create separate React components for each page or view you want to display in your application. These components will represent the content of the different routes.

Level 1

- Build a navigation bar (navbar) with Link components that point to the different routes in your application. These links will enable users to navigate between pages.
- Test your application to ensure that clicking on the navigation links updates the content of the page based on the selected route.

Level 2

- Test your application to ensure that clicking on the navigation links updates the content of the page based on the selected route.

Level 3

- Implement responsive design principles for your navbar to ensure that it looks good and functions well on different screen sizes.
- Pay attention to the user experience. Ensure that users can easily understand the purpose of each route and navigate through your application seamlessly.

To Do

- ✓ Create a navbar
- ✓ Create a static route
- ✓ Create a homepage

Layout Design

Logo **Home** **About** **Course** **Contactus**

Home

Footer

Design Card

Designing a responsive card involves creating a visually appealing and user-friendly.

Brief

Designing a responsive card involves creating a visually appealing and user-friendly component that adapts to different screen sizes.

Level 1

- Start by defining the content you want to display in the card. This may include an image, title, description, and additional information.
- Decide on the layout of your card. Consider whether you want a simple rectangular card or if you prefer a card with rounded corners or a different shape.

Level 2

- If your card includes images, make them responsive. Consider using CSS to ensure that images scale appropriately on different screen sizes. This prevents images from being too large on smaller screens or too small on larger screens.

Level 3

- Test your card design on mobile devices to ensure a pleasant user experience. Consider stacking elements vertically or making adjustments to the layout to fit smaller screens.

To Do

- [✓] Create a card design
- [✓] Add button
- [✓] Add content
- [✓] Add images
- [✓] Make responsive card

Layout Design

Card_1 Card_2 Card_3

Card_4 Card_5 Card_6

Card_7 Card_8 Card_9

Passing Data

Passing data from child components to parent components in React involves a unidirectional flow of information.

Brief

First, ensure that you have a clear parent-child relationship between your components. In React, data generally flows from parent to child, but you need to pass data from child to parent.

Level 1

- In your parent component, define a function that will act as a callback. This function should receive data from the child component as an argument.
- Pass the callback function as a prop to the child component. This allows the child component to invoke the callback and pass data back to the parent.

Level 2

- Inside the callback function in the parent component, update the state or perform any necessary actions with the received data. This allows the parent component to react to changes initiated by the child.

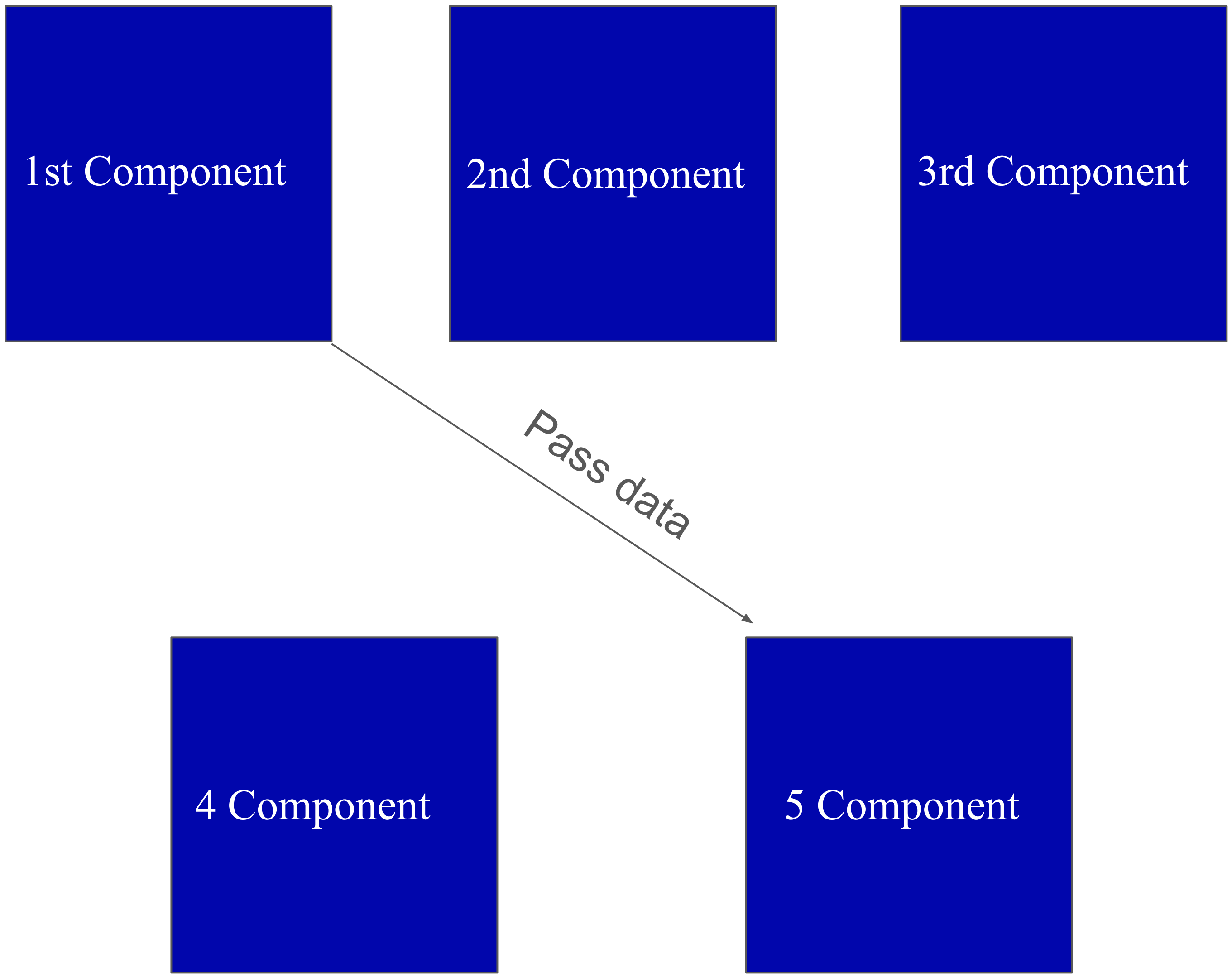
Level 3

- Be mindful of props drilling if your component tree is deep. If the child component is nested several levels deep, you might need to pass the callback function through intermediary components.

To Do

- ✓ Create a components
- ✓ Atleast 5 components
- ✓ Pass a data
- ✓ First to Last component

Layout Design



Resuable Components

Building reusable components in React involves adopting certain principles and patterns to create modular and adaptable pieces of UI.

Brief

Break down complex UI elements into smaller, independent components. These components should encapsulate a specific piece of functionality or represent a distinct visual element.

Level 1

- Utilize props to make components configurable. Design components in a way that allows users to customize their appearance and behavior by passing different props.
- Design components to be parameterized, allowing users to pass data, styles, or callbacks dynamically.

Level 2

- Adopt the container and presentational component pattern. Container components handle data logic and state, while presentational components focus on rendering UI based on props.

Level 3

- Design components to be semantic and accessible. This ensures that your components can be used inclusively and meet web accessibility standards.
- Document your components thoroughly. Include information about props, use cases, and any considerations for usage.

To Do

- ✓ Create a components
- ✓ Make it resuable
- ✓ Reuse components

Layout Design

Make Reusable components

Use reusable components

Use reusable components

Components_1

Components_2

Use reusable components

Use reusable components

Components_3

Components_4