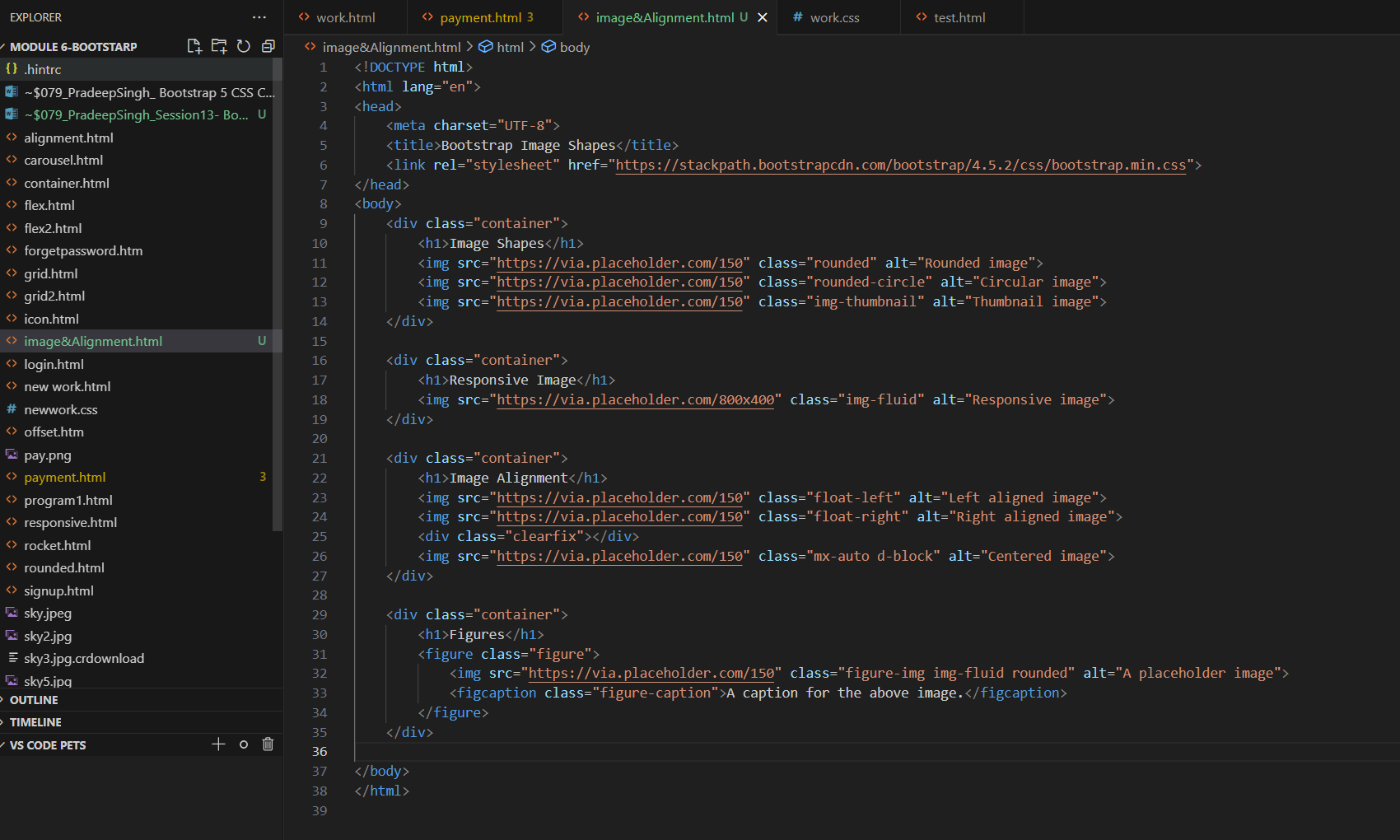
**1. Images - Shapes and Alignment**

**Shapes**

* **Rounded Images**: Use the .rounded class to make the edges of the image round.
* **Circular Images**: Apply the .rounded-circle class to give images a circular shape.
* **Thumbnail Images**: Use the .img-thumbnail class to add a border and padding around the image, making it look like a thumbnail.

**Size and Responsiveness**

* **Responsive Images**: Use the .img-fluid class to ensure images scale nicely to the parent element, making them responsive.
* **Sizing Classes**: Control the size of images with classes like .w-25, .w-50, .w-75, and .w-100 for width percentages.

  
  
  
  
Output for above code will be:





**Alignment**

* **Center Alignment**: Apply the .mx-auto class to center an image horizontally within a block-level element.
* **Float**: Use .float-start to float an image to the left and .float-end to float it to the right.
* **Text Alignment**: Use .text-start, .text-center, and .text-end to align text around images accordingly.

**2. Figures**

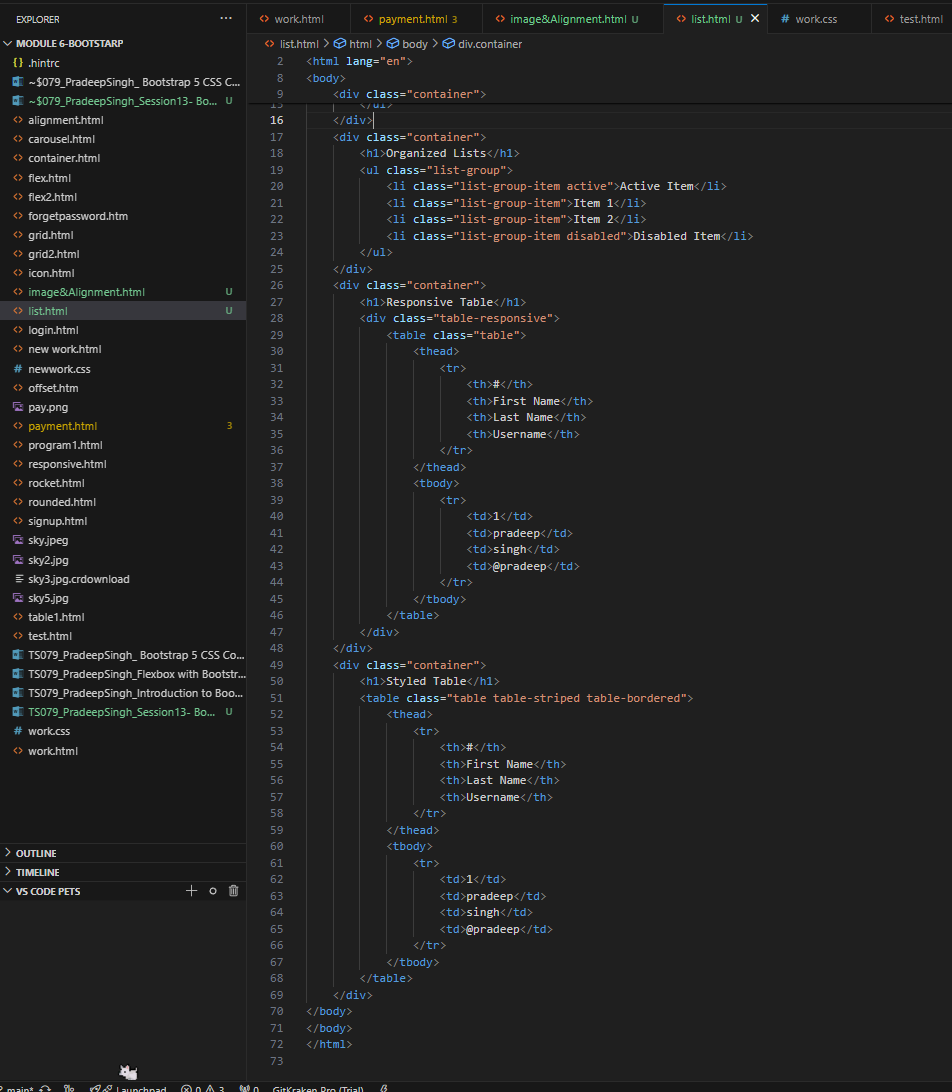
* **Figure Container**: Wrap images in a <figure> element to semantically group media and its caption.
* **Figure Image**: Use the .figure-img class to style the image inside the <figure>.
* **Figure Caption**: Add a caption to images using the <figcaption> element and style it with the .figure-caption class.

**3. Lists**

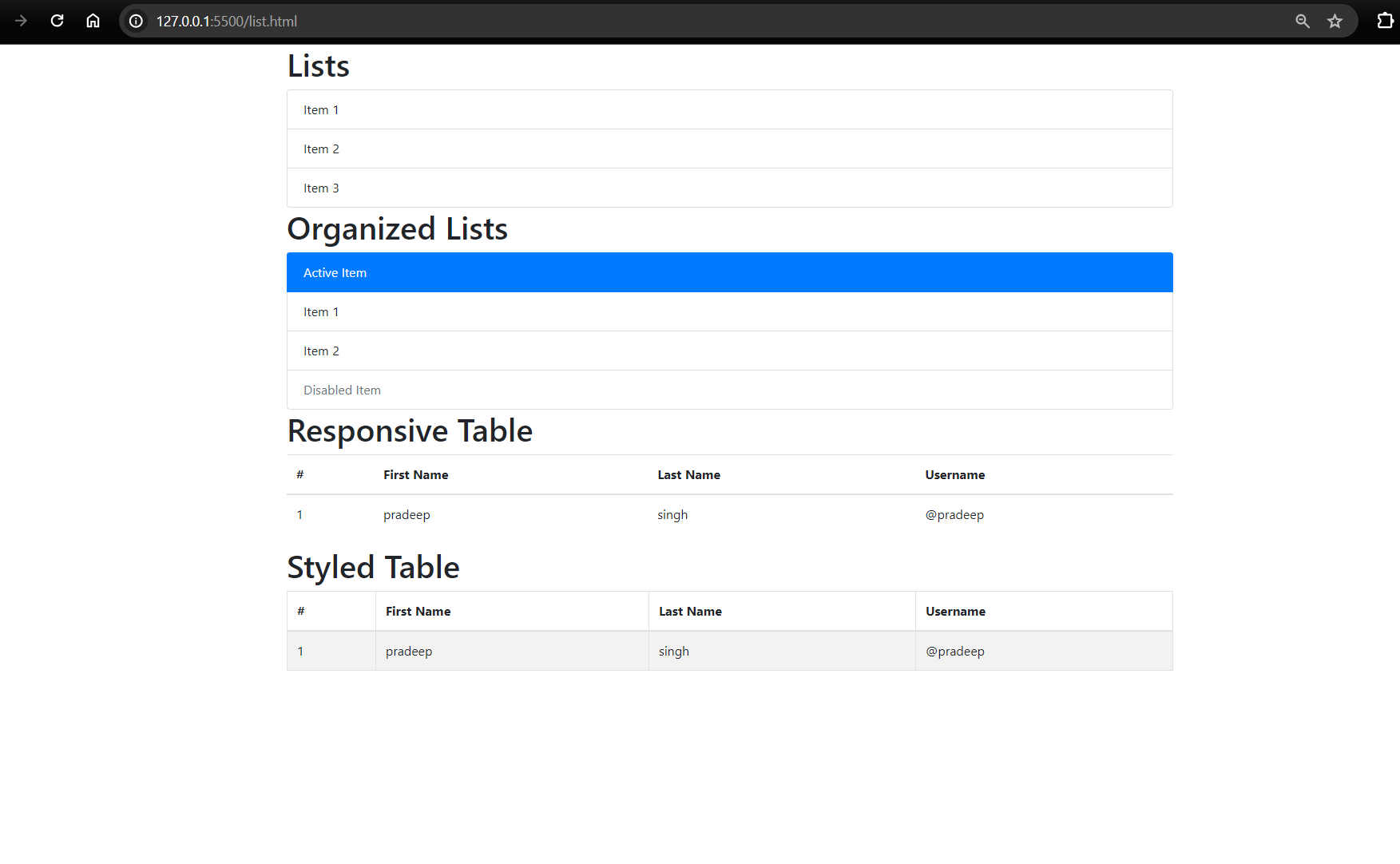
* **Unordered Lists**: Style bullet lists with the .list-unstyled class to remove default bullets or use .list-inline to display list items horizontally.
* **Ordered Lists**: Create numbered lists that can be styled similarly to unordered lists.
* **Inline Lists**: Use the .list-inline-item class for list items to display them inline, making them appear side by side.
* **Custom Icons**: Add icons to list items using libraries like Font Awesome for more visual appeal.

**4. Tables**

* **Basic Tables**: Use the .table class to apply basic Bootstrap styling.
* **Striped Rows**: Add .table-striped to alternate row colors.
* **Bordered Tables**: Use .table-bordered to add borders to all table cells.
* **Hover Rows**: Apply .table-hover to highlight rows on hover.
* **Responsive Tables**: Wrap tables in .table-responsive to make them scroll horizontally on small screens.



Output for above code will look like as below:



**5. Buttons and Button Groups**

**Buttons**

* **Basic Styles**: Use classes like .btn-primary, .btn-secondary, .btn-success, etc., to style buttons.
* **Sizes**: Adjust button sizes with .btn-lg for large buttons, .btn-sm for small buttons, and .btn-block for full-width buttons.
* **Outline Buttons**: Use .btn-outline-\* classes (e.g., .btn-outline-primary) for buttons with colored borders and transparent backgrounds.

**Button Groups**

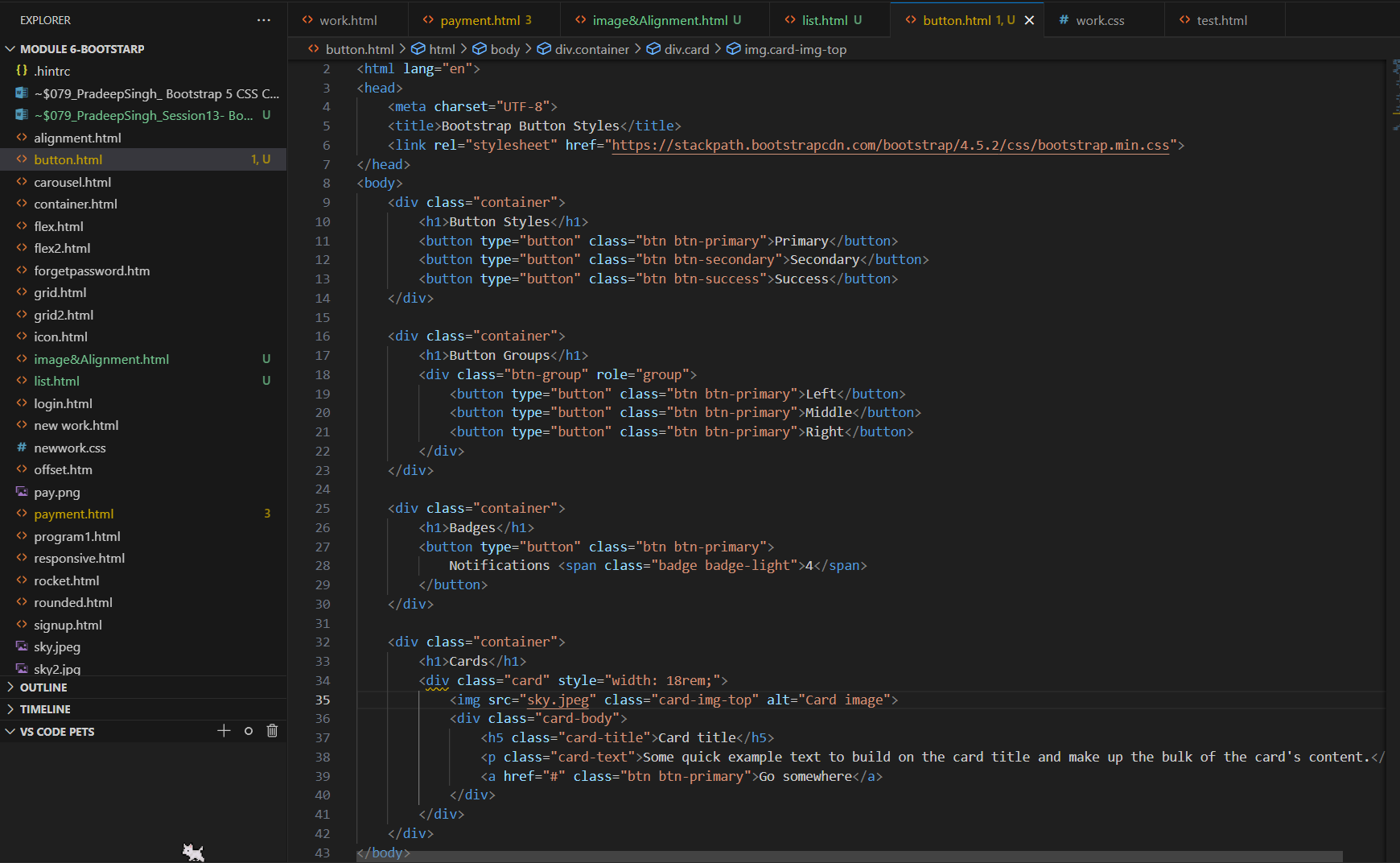
* **Grouping**: Use the .btn-group class to group a series of buttons together on a single line.
* **Vertical Alignment**: Use .btn-group-vertical to stack buttons vertically.
* **Toolbar**: Combine multiple button groups using .btn-toolbar for complex toolbar-like layouts.

**6. Badges**

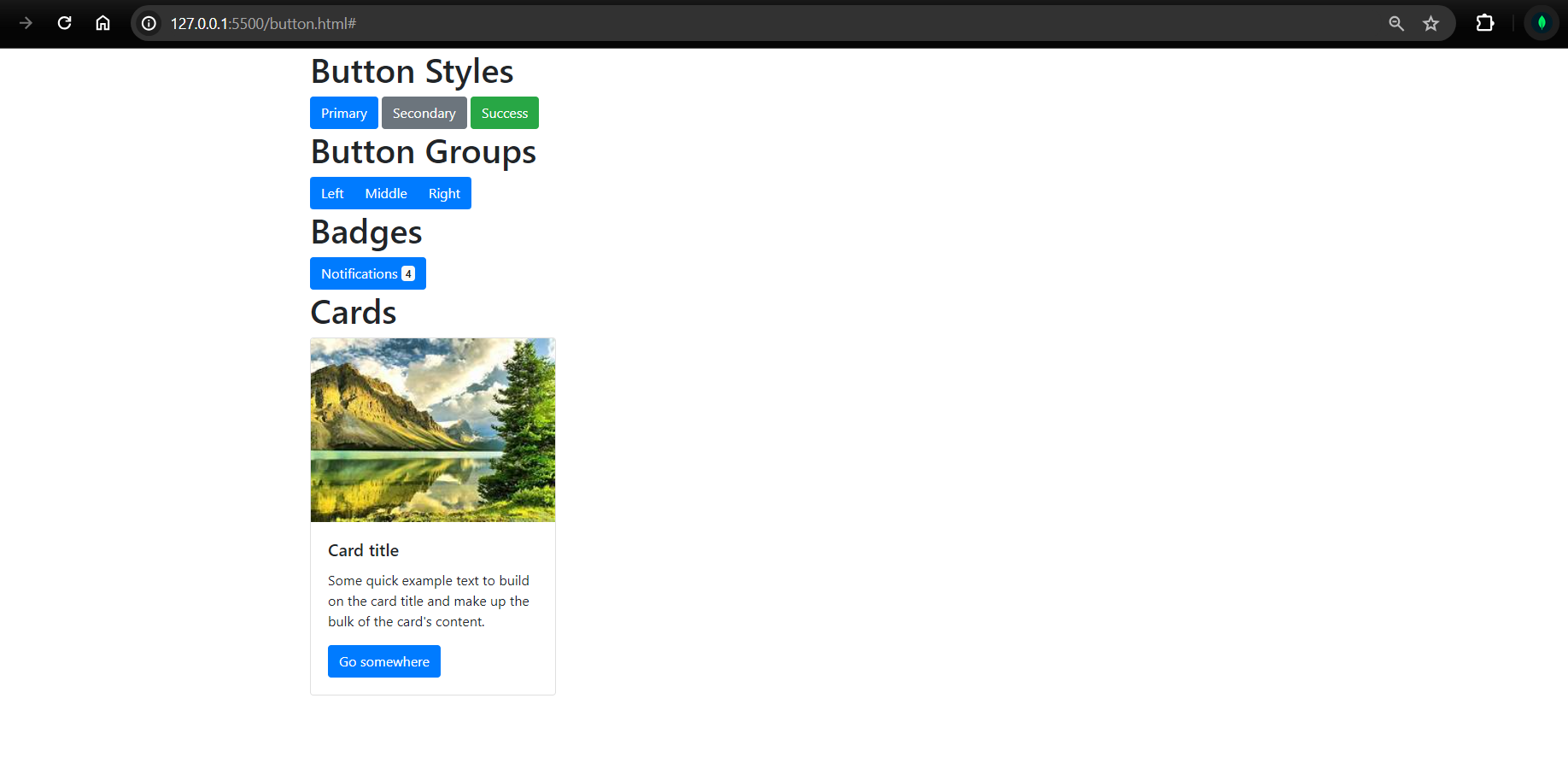
* **Basic Badges**: Add small counts or labels with the .badge class.
* **Colors**: Style badges using contextual classes like .badge-primary, .badge-secondary, .badge-success, etc.
* **Pill Badges**: Use .badge-pill to make badges more rounded and pill-shaped.

**7. Cards**

* **Basic Card**: Use the .card class to create a container with a shadow and padding.
* **Card Header and Footer**: Add .card-header and .card-footer for header and footer sections.
* **Card Body**: Use the .card-body class to wrap the main content of the card.
* **Card Image**: Use .card-img-top or .card-img-bottom to place images at the top or bottom of the card.
* **Card Groups**: Use .card-group to group multiple cards together.



Out put for above code will look like as below :

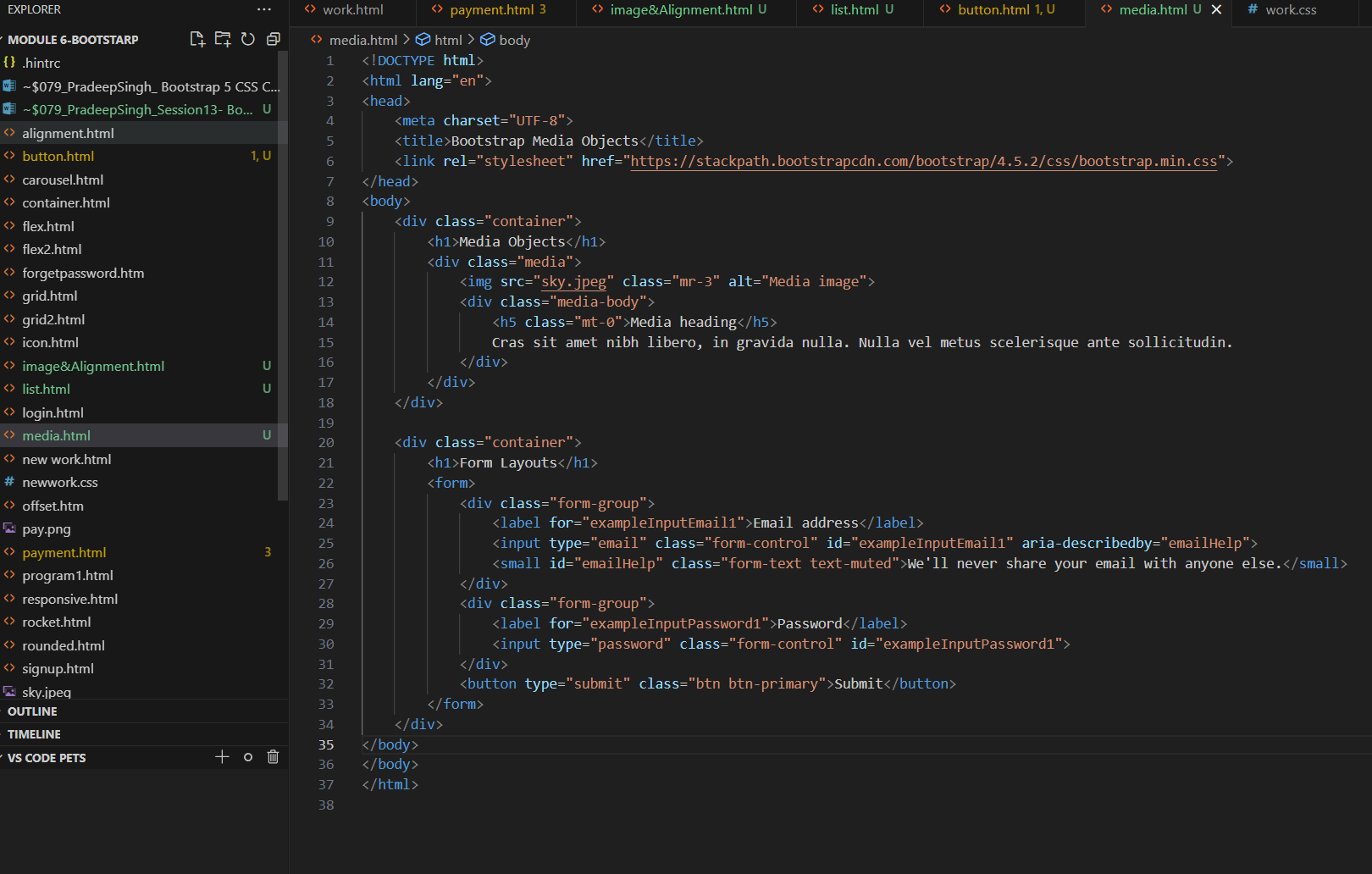


**8. Media Objects**

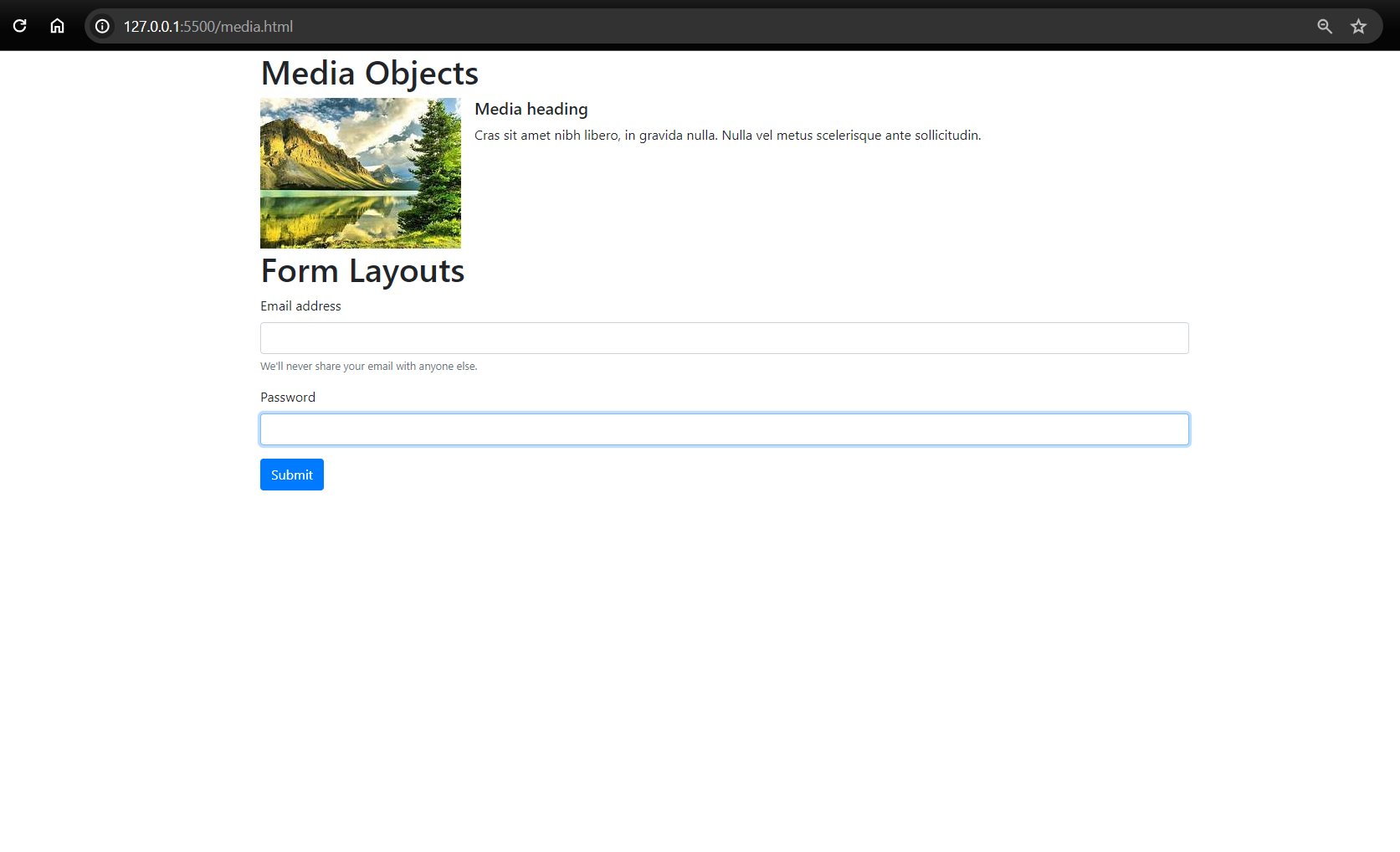
* **Media Container**: Use the .media class to create a container for media content.
* **Media Body**: Use .media-body to contain the text associated with the media.
* **Media Alignment**: Align media elements to the top, middle, or bottom using .align-self-start, .align-self-center, and .align-self-end.

**9. Forms**

* **Basic Form Layout**: Use the .form-group class to group labels and inputs.
* **Inline Forms**: Use the .form-inline class to display form elements inline.
* **Form Controls**: Style input elements with classes like .form-control, .form-check, .form-select, etc.
* **Validation**: Add validation feedback with .is-valid and .is-invalid classes.



Output for above code will look like as below :

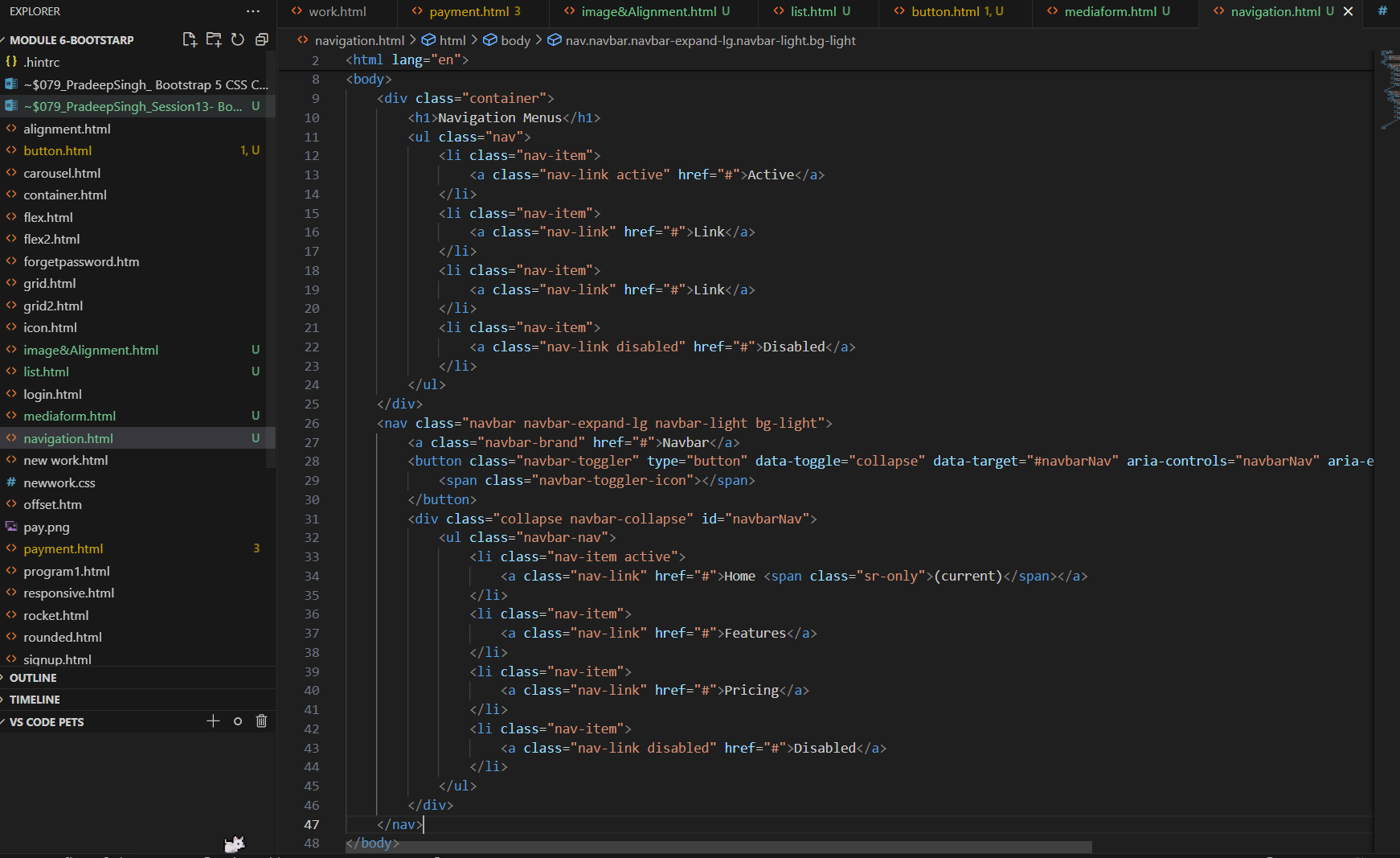


**10. Nav Menus**

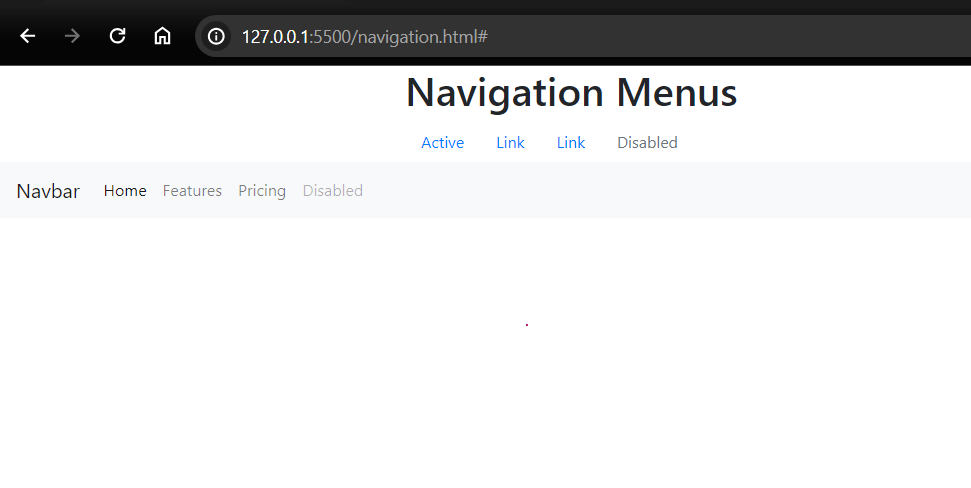
* **Basic Nav**: Use the .nav class to create a navigation menu.
* **Horizontal Alignment**: Use .nav-item and .nav-link classes to align items horizontally.
* **Vertical Alignment**: Use .flex-column to stack items vertically.
* **Pills and Tabs**: Use .nav-pills and .nav-tabs for different styles of navigation.

**11. Navbar**

* **Basic Navbar**: Use the .navbar class to create a responsive navigation bar.
* **Color Schemes**: Use .navbar-light or .navbar-dark with background color classes like .bg-light or .bg-dark.
* **Collapsible Content**: Use .navbar-toggler and .navbar-collapse to make the navbar responsive.
* **Brand and Links**: Use .navbar-brand for the brand/logo and .navbar-nav with .nav-item and .nav-link for links.



Output for above code will look like as shown below:



**12. Breadcrumb**

* **Breadcrumb Container**: Use the .breadcrumb class to create a breadcrumb trail.
* **Breadcrumb Items**: Add .breadcrumb-item to each list item in the breadcrumb trail.
* **Active Item**: Use .active to indicate the current page or section.

**13. Pagination**

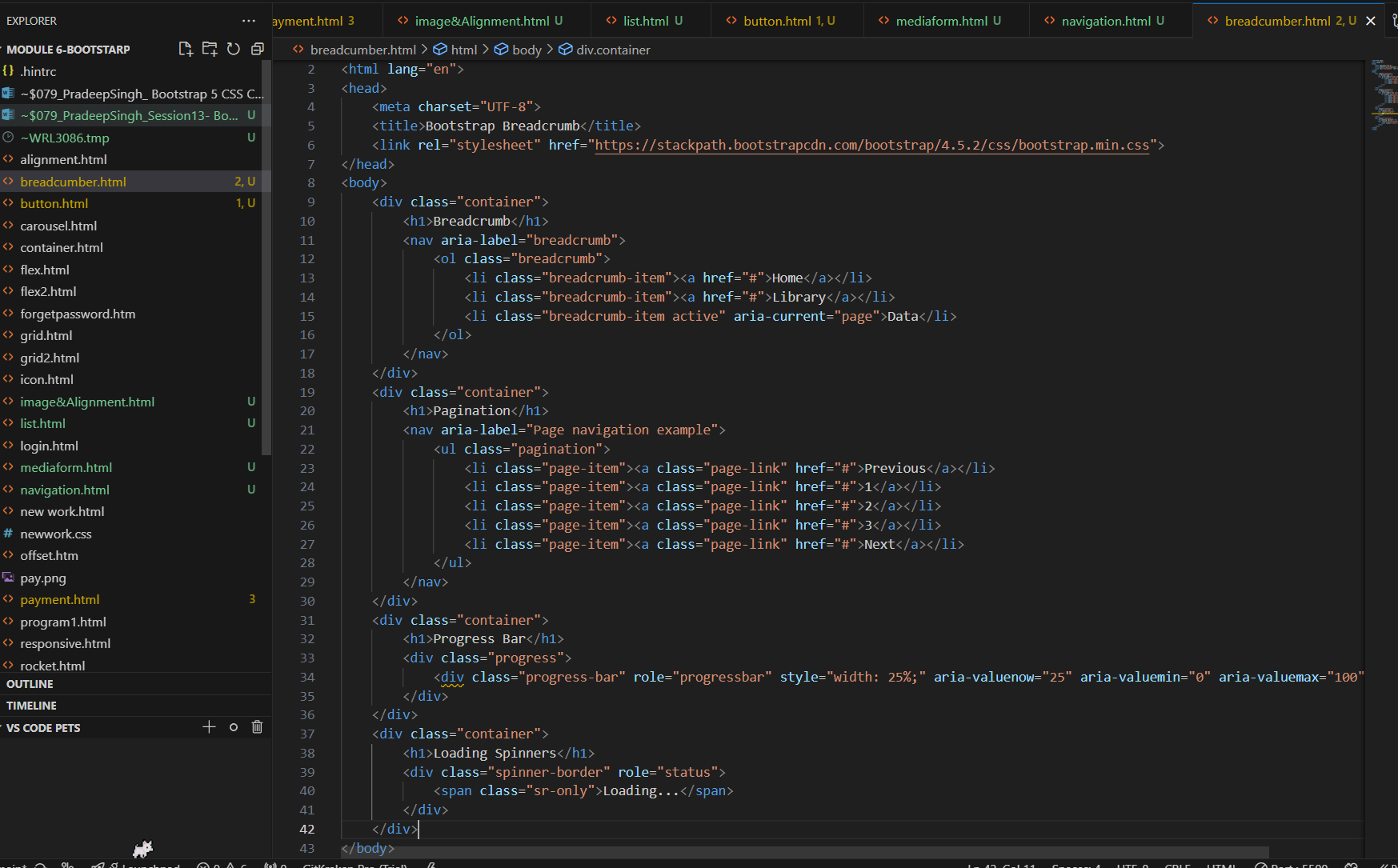
* **Pagination Container**: Use the .pagination class to create pagination controls.
* **Pagination Items**: Use .page-item and .page-link to create individual pagination items.
* **Size Options**: Adjust the size of pagination controls with .pagination-lg or .pagination-sm.

**14. Progress Bar**

* **Basic Progress Bar**: Use the .progress class to create a container and .progress-bar for the actual progress.
* **Labels and Stripes**: Add labels inside the progress bar and stripes using .progress-bar-striped.
* **Animated Stripes**: Use .progress-bar-animated to animate the stripes.

**15. Spinners**

* **Basic Spinner**: Use the .spinner-border class for a border spinner and .spinner-grow for a growing spinner.
* **Colors**: Apply contextual classes like .text-primary, .text-secondary, etc., for spinner colors.
* **Sizes**: Adjust spinner sizes with .spinner-border-sm or .spinner-grow-sm.



For above topics code is attchaed in above picture and output is shown in below picture:

