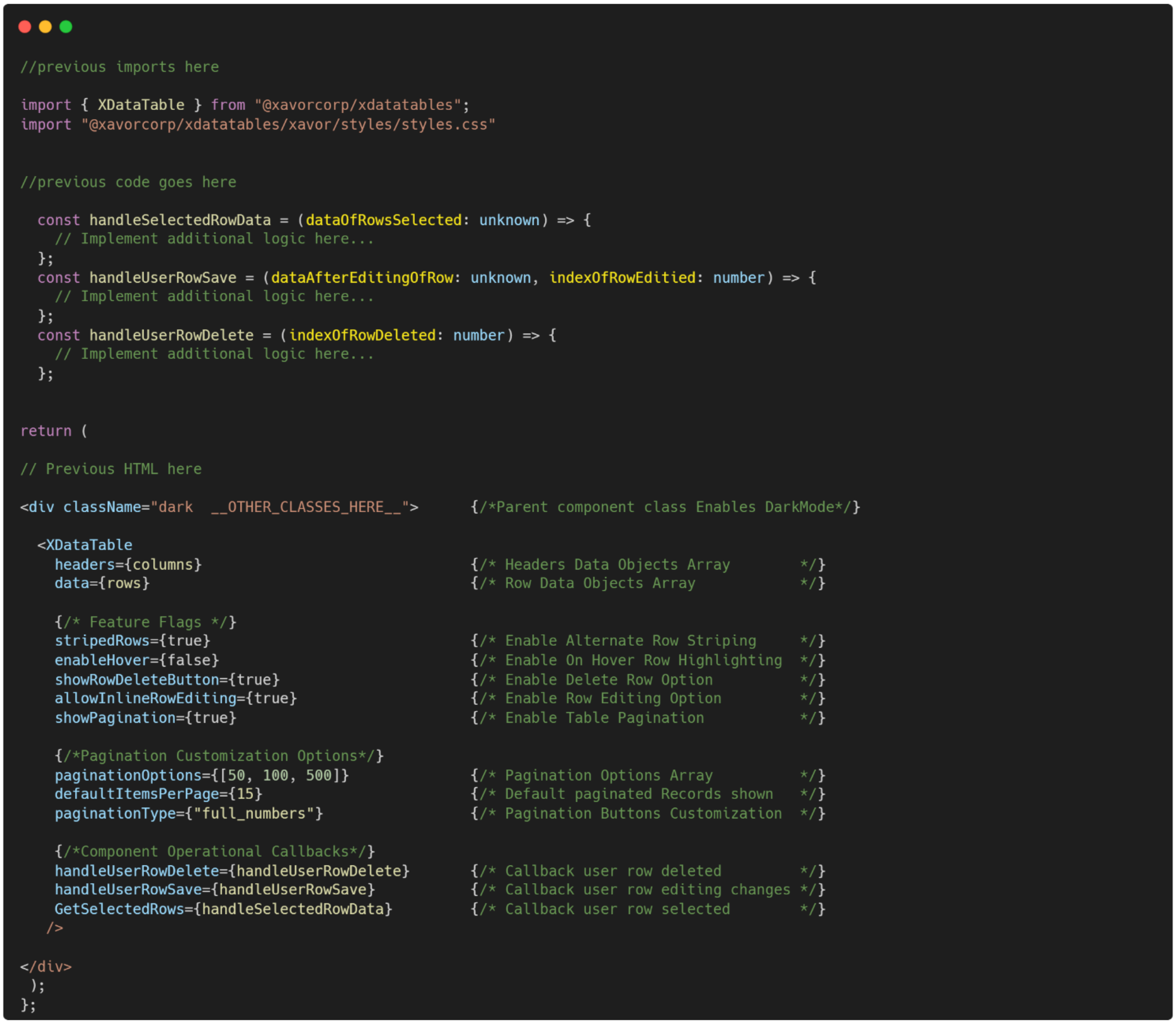
XDataTable Documentation v1.0

The XDataTable plugin is a powerful and versatile table plugin designed for seamless development. It offers a wide range of features to enhance data display and manipulation while cutting down development time by more than 35%.

Quick Start:

To utilize the XDataTable:

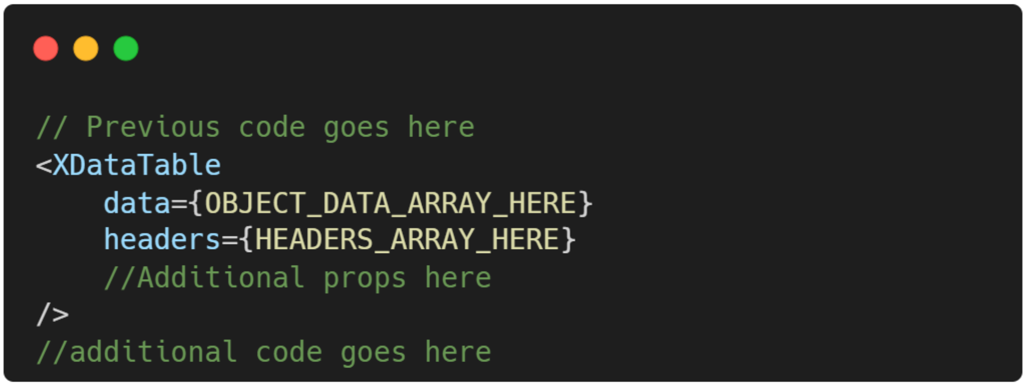
1. Follow the instructions for installing the package provided by the admin document.
2. Import the component in your application.
    a. Example Import:



```
//previous imports here

import { XDataTable } from "@xavorcorp/xdatatables";
import "@xavorcorp/xdatatables/xavor/styles/styles.css"


//previous code goes here

  const handleSelectedRowData = (dataOfRowsSelected: unknown) => {
    // Implement additional logic here...
  };
  const handleUserRowSave = (dataAfterEditingOfRow: unknown, indexOfRowEditied: number) => {
    // Implement additional logic here...
  };
  const handleUserRowDelete = (indexOfRowDeleted: number) => {
    // Implement additional logic here...
  };


return (

// Previous HTML here

<div className="dark  __OTHER_CLASSES_HERE__">      {/*Parent component class Enables DarkMode*/}

  <XDataTable
    headers={columns}                               {/* Headers Data Objects Array        */}
    data={rows}                                     {/* Row Data Objects Array            */}

    {/* Feature Flags */}
    stripedRows={true}                              {/* Enable Alternate Row Striping     */}
    enableHover={false}                             {/* Enable On Hover Row Highlighting  */}
    showRowDeleteButton={true}                      {/* Enable Delete Row Option          */}
    allowInlineRowEditing={true}                    {/* Enable Row Editing Option         */}
    showPagination={true}                           {/* Enable Table Pagination           */}

    {/*Pagination Customization Options*/}
    paginationOptions={[50, 100, 500]}              {/* Pagination Options Array          */}
    defaultItemsPerPage={15}                        {/* Default paginated Records shown   */}
    paginationType={"full_numbers"}                 {/* Pagination Buttons Customization  */}

    {/*Component Operational Callbacks*/}
    handleUserRowDelete={handleUserRowDelete}       {/* Callback user row deleted         */}
    handleUserRowSave={handleUserRowSave}           {/* Callback user row editing changes */}
    GetSelectedRows={handleSelectedRowData}         {/* Callback user row selected        */}
   />

</div>
 );
};
```

Feature Configuration:

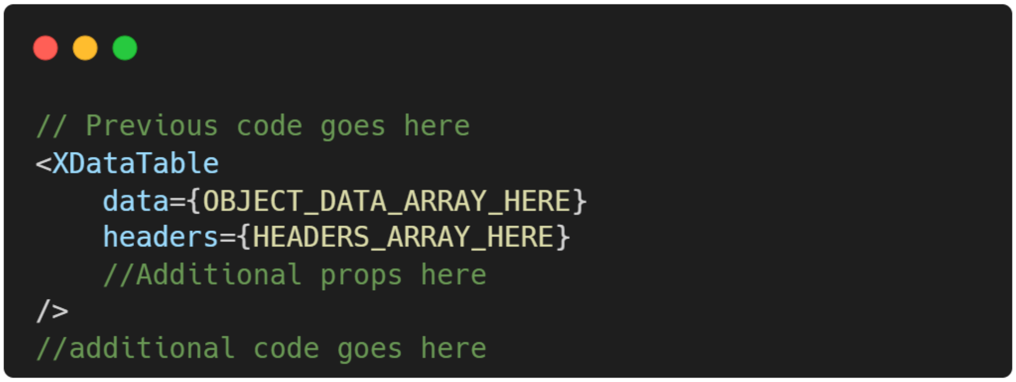
- **Searching**
    - Allows search through data globally within the table
    - Searching is enabled by default within the plugin and is available globally.



```
// Previous code goes here
<XDataTable
    data={OBJECT_DATA_ARRAY_HERE}
    headers={HEADERS_ARRAY_HERE}
    //Additional props here
/>
//additional code goes here
```

- **Column Sorting**

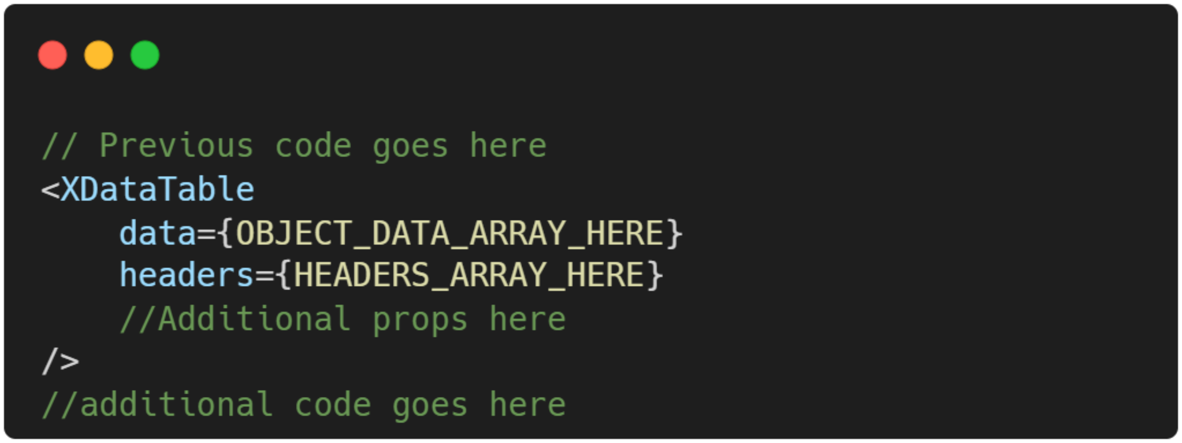
- Allows user to sort data within the table at individual columns level
- Columns Sorting is enabled by default within the plugin and is available globally.



```
// Previous code goes here
<XDataTable
    data={OBJECT_DATA_ARRAY_HERE}
    headers={HEADERS_ARRAY_HERE}
    //Additional props here
/>
//additional code goes here
```

- **Export to CSV,PDF**

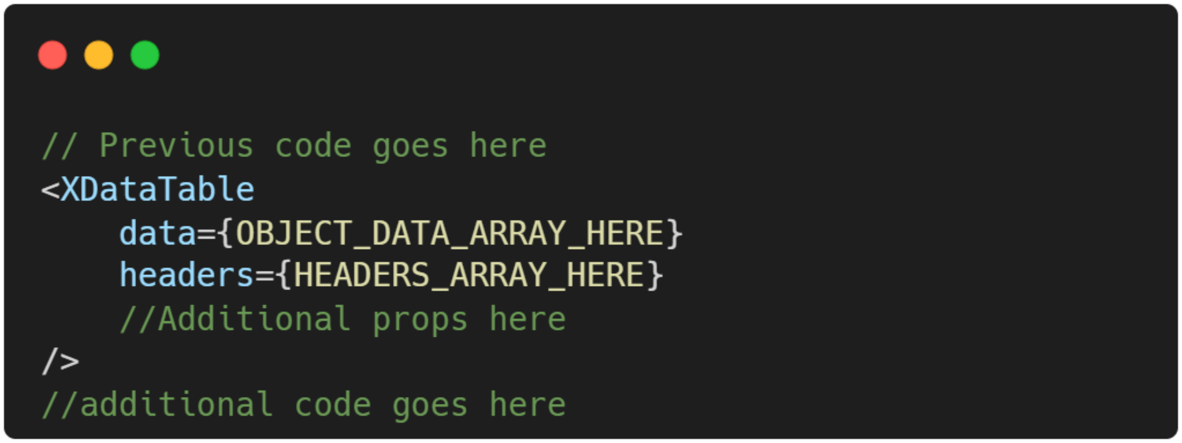
  - Export Table data directly as a PDF or CSV File
  - Exporting as CSV and PDF is enabled by default within the plugin and is available globally.



```
// Previous code goes here
<XDataTable
    data={OBJECT_DATA_ARRAY_HERE}
    headers={HEADERS_ARRAY_HERE}
    //Additional props here
/>
//additional code goes here
```

- **Import Excel File**

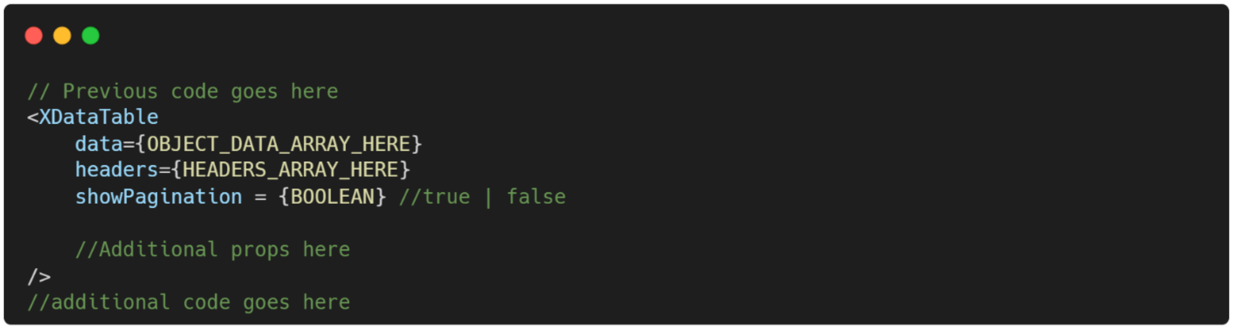
  - Import an Excel sheet as Table data directly.
  - Import Excel sheet is enabled by default within the plugin and is available globally.



```
// Previous code goes here
<XDataTable
    data={OBJECT_DATA_ARRAY_HERE}
    headers={HEADERS_ARRAY_HERE}
    //Additional props here
/>
//additional code goes here
```

- **Pagination**

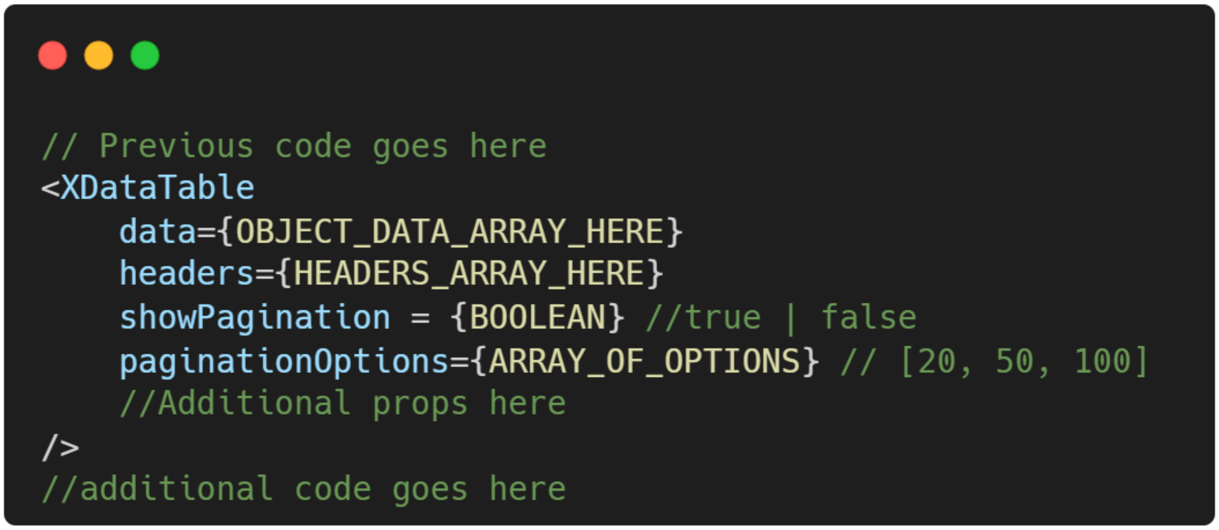
  - XDataTable provides data pagination options out of the box.
  - Pagination can be enabled or disabled by passing props to the component.
    - **ShowPagination** can receive True or False as Boolean parameter value.



```
// Previous code goes here
<XDataTable
    data={OBJECT_DATA_ARRAY_HERE}
    headers={HEADERS_ARRAY_HERE}
    showPagination = {BOOLEAN} //true | false

    //Additional props here
/>
//additional code goes here
```

- **Pagination Options for records on page**

  - Custom pagination options are supported out of the box and can be configured by passing an array of positive integer values starting from value 1…N, these values will then be visible in the dropdown list of selections, where on-page records will then be set equal to selected option.
  - **paginationOptions** receive interger value array for example ` [20,50,100] `.

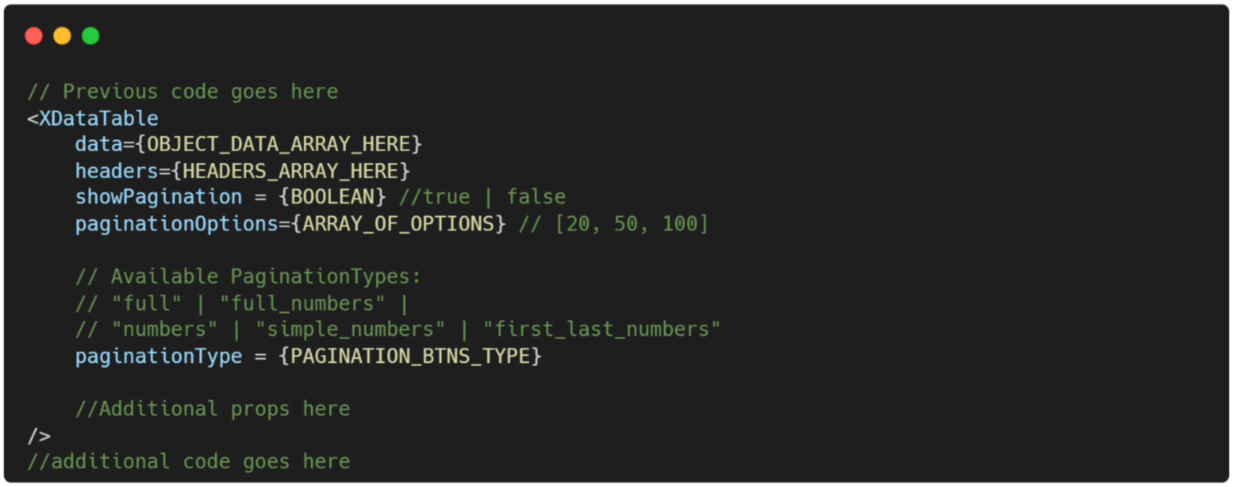


```
// Previous code goes here
<XDataTable
    data={OBJECT_DATA_ARRAY_HERE}
    headers={HEADERS_ARRAY_HERE}
    showPagination = {BOOLEAN} //true | false
    paginationOptions={ARRAY_OF_OPTIONS} // [20, 50, 100]
    //Additional props here
/>
//additional code goes here
```

- **Pagination buttons Options**

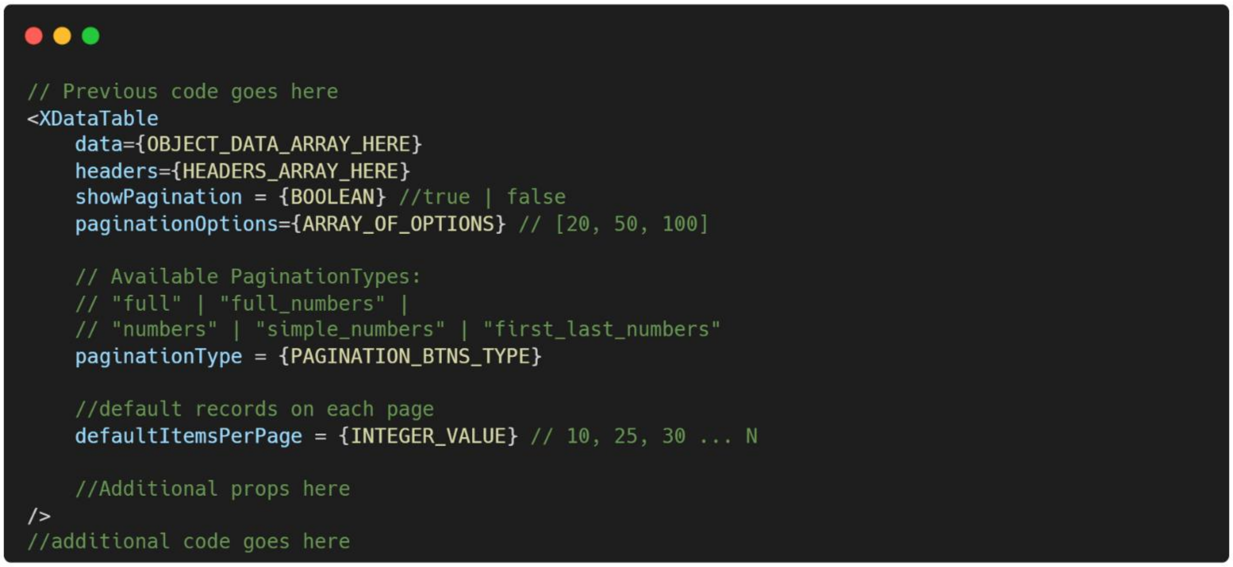
  - XDataTable Also provides data pagination button Types options out of the box as well.
  - More than five Pagination types are available at your disposal out of the box
  - **paginationType** expects a string value that must be one of the following:
    - full
    - full_numbers
    - numbers
    - simple_numbers
    - first_last_numbers

Note: that these are case sensitive values



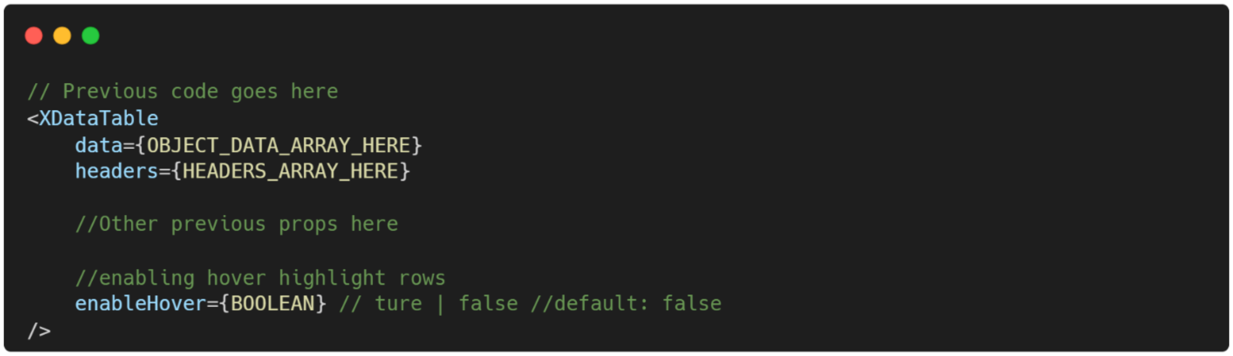
```
// Previous code goes here
<XDataTable
    data={OBJECT_DATA_ARRAY_HERE}
    headers={HEADERS_ARRAY_HERE}
    showPagination = {BOOLEAN} //true | false
    paginationOptions={ARRAY_OF_OPTIONS} // [20, 50, 100]

    // Available PaginationTypes:
    // "full" | "full_numbers" |
    // "numbers" | "simple_numbers" | "first_last_numbers"
    paginationType = {PAGINATION_BTNS_TYPE}

    //Additional props here
/>
//additional code goes here
```

- **Customizable Default Pagination Records Shown Option**
  - Pagination supports customization of default records being shown on each page when component loads, by default it shows 10 records if no pagination options are passed.



```
// Previous code goes here
<XDataTable
    data={OBJECT_DATA_ARRAY_HERE}
    headers={HEADERS_ARRAY_HERE}
    showPagination = {BOOLEAN} //true | false
    paginationOptions={ARRAY_OF_OPTIONS} // [20, 50, 100]

    // Available PaginationTypes:
    // "full" | "full_numbers" |
    // "numbers" | "simple_numbers" | "first_last_numbers"
    paginationType = {PAGINATION_BTNS_TYPE}

    //default records on each page
    defaultItemsPerPage = {INTEGER_VALUE} // 10, 25, 30 ... N

    //Additional props here
/>
//additional code goes here
```

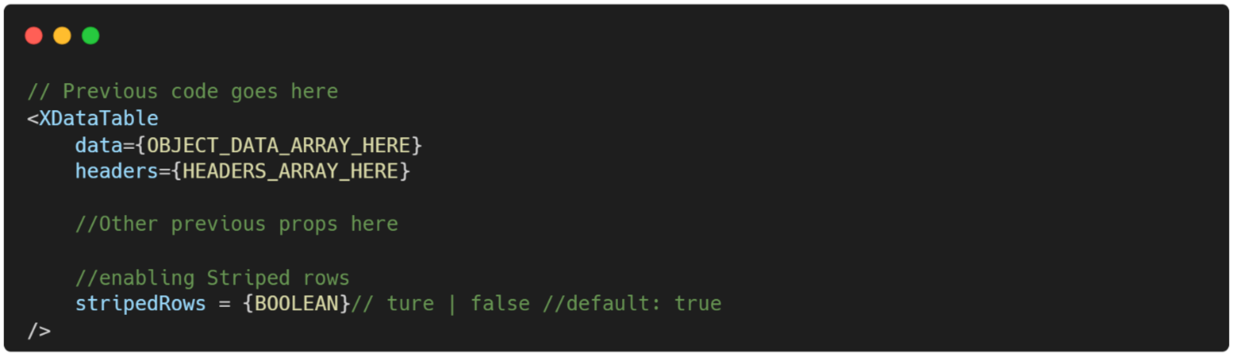
- **Enable/disable Hover On Rows options**
  - XDataTable supports highlighted rows on hovering by default it is disabled and can be enabled by passing a Boolean value
  - **enableHover** can receive True or False as Boolean parameter value



```
// Previous code goes here
<XDataTable
    data={OBJECT_DATA_ARRAY_HERE}
    headers={HEADERS_ARRAY_HERE}

    //Other previous props here

    //enabling hover highlight rows
    enableHover={BOOLEAN} // ture | false //default: false
/>
```

- **Enable/disable Alternate Rows Striped**

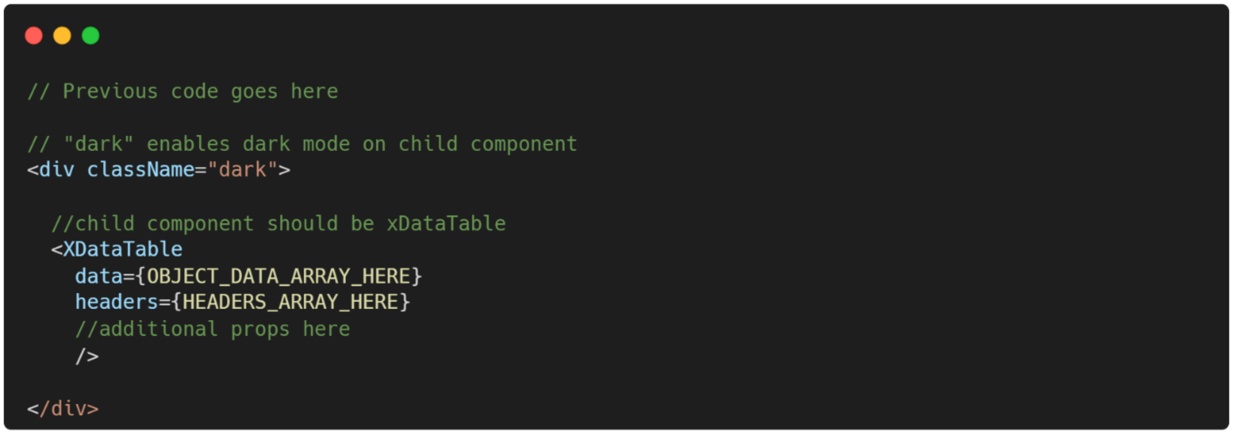
  - XDataTable supports striped alternate rows on and can be enabled by passing a Boolean value to the attribute.
  - **stripedRows** can receive True or False as Boolean parameter value



```
// Previous code goes here
<XDataTable
    data={OBJECT_DATA_ARRAY_HERE}
    headers={HEADERS_ARRAY_HERE}

    //Other previous props here

    //enabling Striped rows
    stripedRows = {BOOLEAN}// ture | false //default: true
/>
```

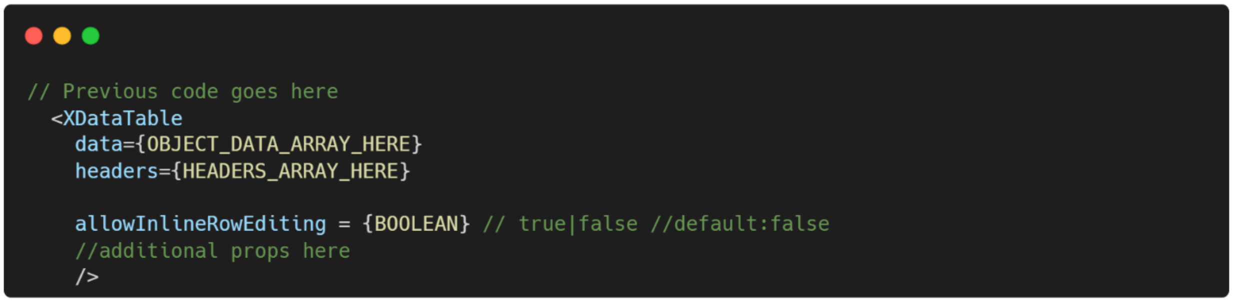
- **Explicit Theme Handling Options for Changing Dark/Light Mode of plugin component:**

  - XDataTable supports both dark and light theme options out of the box and can be easily configured. No need to implement it from scratch.
  - To enable/disable dark mode of XDataTable, add ‘dark’ keyword as a class value in parent component of XDataTable.



```
// Previous code goes here

// "dark" enables dark mode on child component
<div className="dark">

  //child component should be xDataTable
  <XDataTable
    data={OBJECT_DATA_ARRAY_HERE}
    headers={HEADERS_ARRAY_HERE}
    //additional props here
    />

</div>
```

- **Inline Row Actions:**
    - XDataTable supports both display and manipulation options out of the box and can be easily enabled/disabled.
    - Each row can be edited or deleted.
    - **allowInlineRowEditing** can receive True or False as Boolean parameter value



```
// Previous code goes here
  <XDataTable
    data={OBJECT_DATA_ARRAY_HERE}
    headers={HEADERS_ARRAY_HERE}

    allowInlineRowEditing = {BOOLEAN} // true|false //default:false
    //additional props here
    />
```

- **Confirmation of Deletion**
    - When a row is selected for deletion, a pop up will show confirmation of task that you wish to terminate.

Support

The XDataTable plugin is currently supported by React-ts and Next-ts frameworks for react v17.