Your objective is to implement a **TABLE component** according to the specifications outlined below, using the listed technologies only – please **do not** use 3rd party table components such as those included with Bootstrap, Angular Material, etc

**High-level Specifications:**

* The component should display Sample Data in a table
* User should be able to select how many rows are displayed in the table
* Table should be paginated if not all rows are displayed on the screen based on the user’s selection
* Pagination options should be displayed in the table footer
* Column names should be displayed in the table header
* Entire table, table header and table footer should always be displayed on the screen while scrolling
* If number of rows exceeds the size of the table body, a vertical scrollbar should be displayed within the table body – only table body shall scroll vertically, table header and footer shall remain as is
* If number of columns exceed the size of the table body, a horizontal scrollbar should be displayed within the table body – only table body and table header shall scroll to reveal the additional columns, table footer shall remain as is
* Each row should contain a button which shall submit the row ID and row status to /api/submit as a POST request – You are not expected to create the POST endpoint, but you can mock one if you like

**Sample data:**

Attachment: sample\_data.json

**Technologies**:

* Angular 4+
* Angular CLI
* Karma / Jasmine
* Sass (optional)

**Submission:**

* Please commit your code to a repository on <https://bitbucket.org> and provide the clone URL with your submission
* Please ensure your package.json includes all dependencies required to build and run code
* You are encouraged to provide any details specific to your build in the README.MD file
* Submissions are expected to compile cleanly and pass all unit tests

**How** **your** **submission** **will** **be** **evaluated:**

* Evaluators will run ng lint to ensure linting standards are adhered to (standard tslint rules from Angular CLI)
* Evaluators will run ng test to ensure all unit tests pass
* Evaluators will run ng build to create the dist version of the app
* Evaluators will run dist version of the app and test functionality
* Evaluators may replace the sample data provided with a different data set

**You will be evaluated on:**

* Overall functionality of the component
  + How data is loaded,
  + How look and feel is implemented
  + How interactivity is implemented
* Code logic, bugginess
* Performance with large data sets
* Error / exception handling
* Code readability and commenting
* Code reuse
* Adherence to Angular style guide and best practices
* Unit tests

**Extra Credit:**

* Implement another table on a different route within the Angular application with:
  + No pagination
  + Infinite vertical scroll that loads and unloads rows into the DOM as required
  + Horizontal scroll based on number of columns