**Steps to Install the application: -**

Step 1:

Download and Extract the File from the following locations:

Or clone the repository from GitHub: https://github.com/kushgadhvi/sailing-schedule

Step 2:

Open the project path in Terminal / CMD and run the command **npm install / i**.

Make sure the latest node/npm version is installed locally.

Step 3:

After node modules are successfully installed run the command **npm run start** to deploying the application locally.

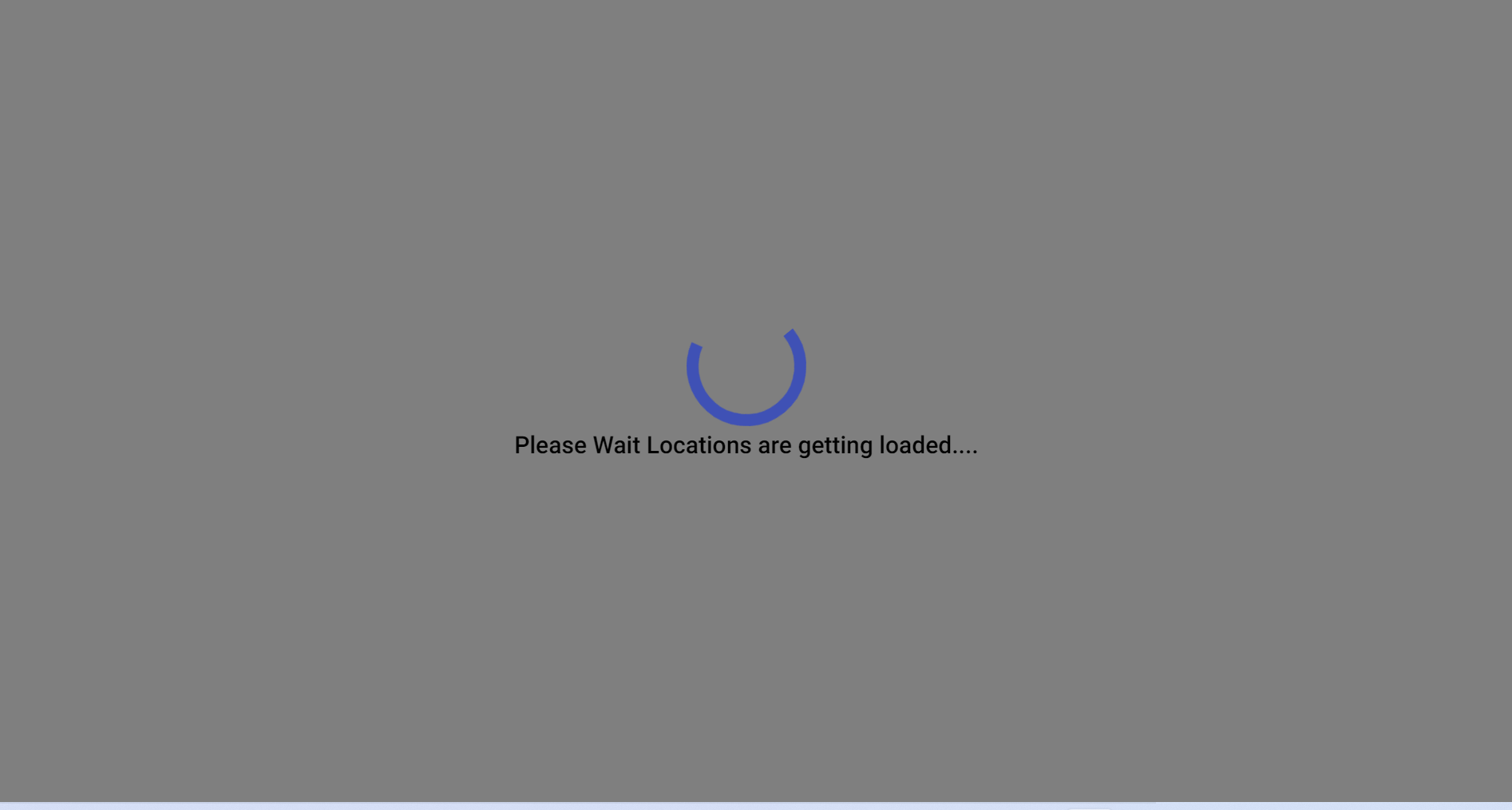
Step 4:

To check a use application paste the following link in the browser <http://localhost:4200/>

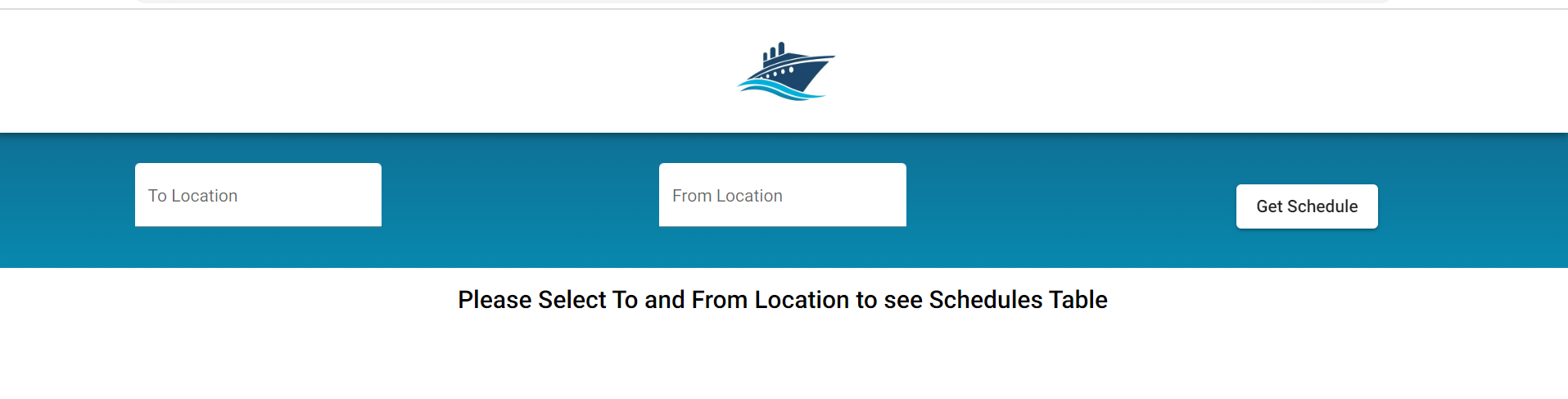
Application is also hosted on git pages and can be accessed with the following link <https://kushgadhvi.github.io/sailing-schedule/>

**Application Workflow: -**

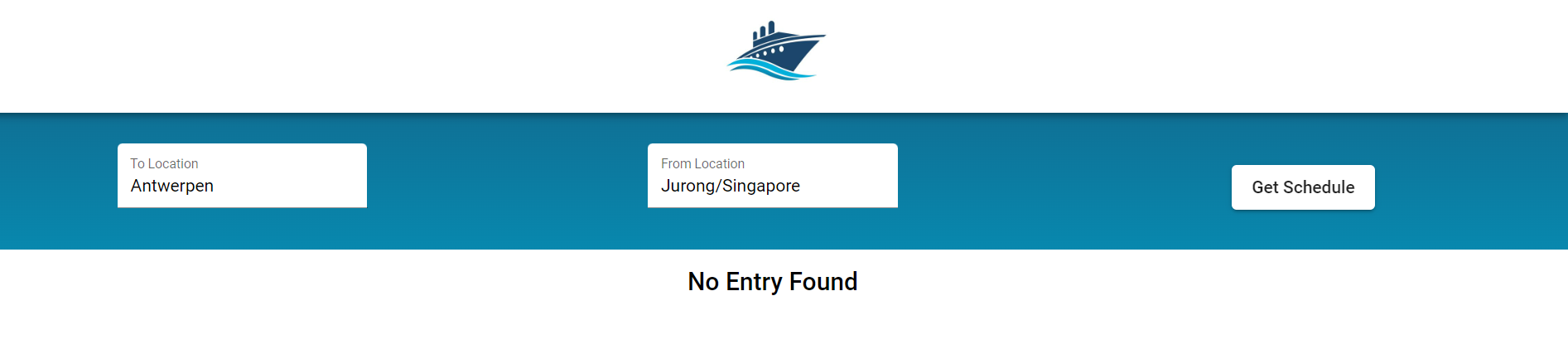
1. After launching the application we will see the loading page it will take a few seconds and since we are fetching the list of locations from the API which is taking time to retrieve all the locations.

****

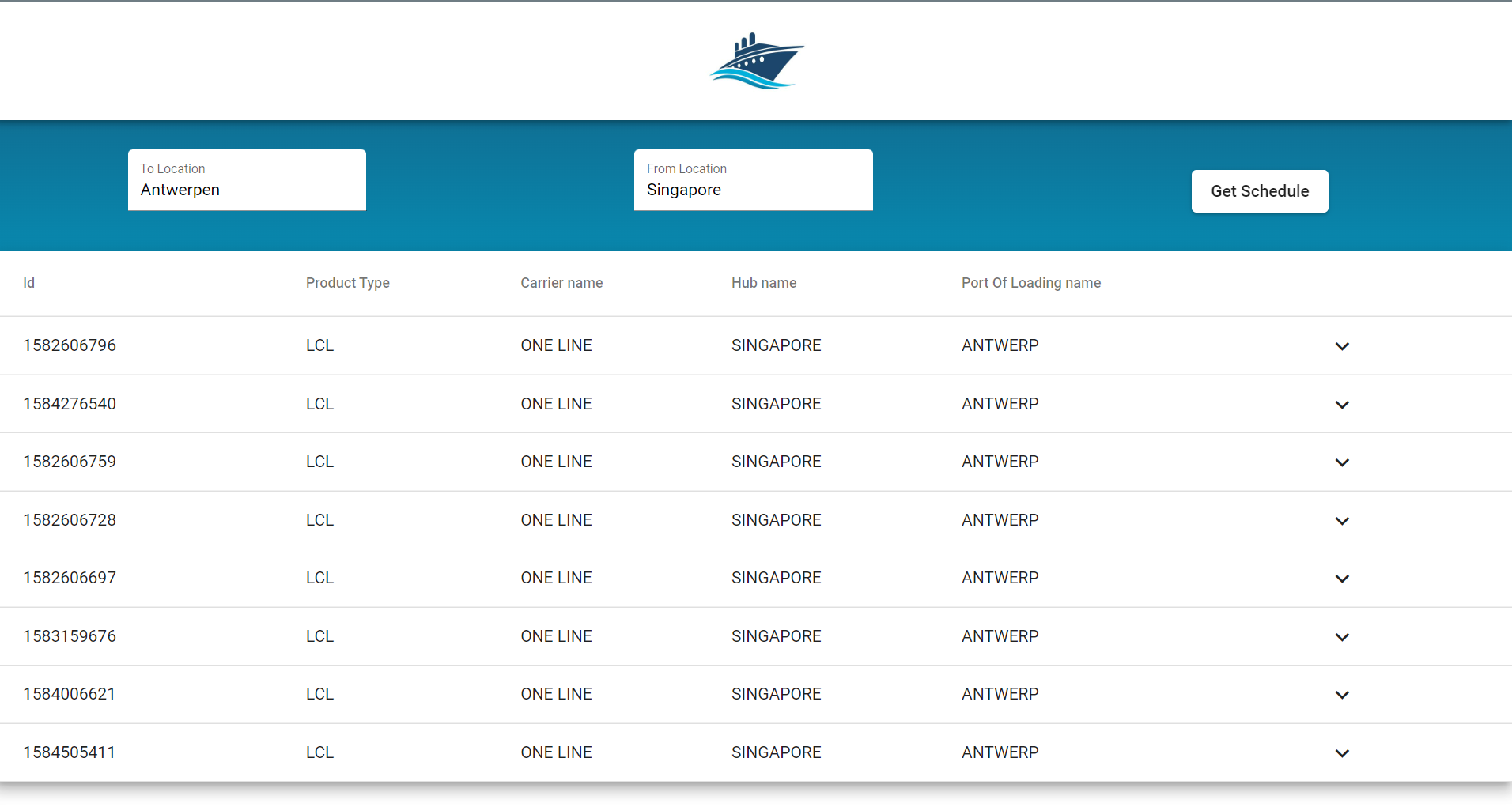
1. After locations list is fetch form the API we can see the main page which has here we can select to and from location. After typing you can see the list of locations select both locations and then click on Get Schedule.



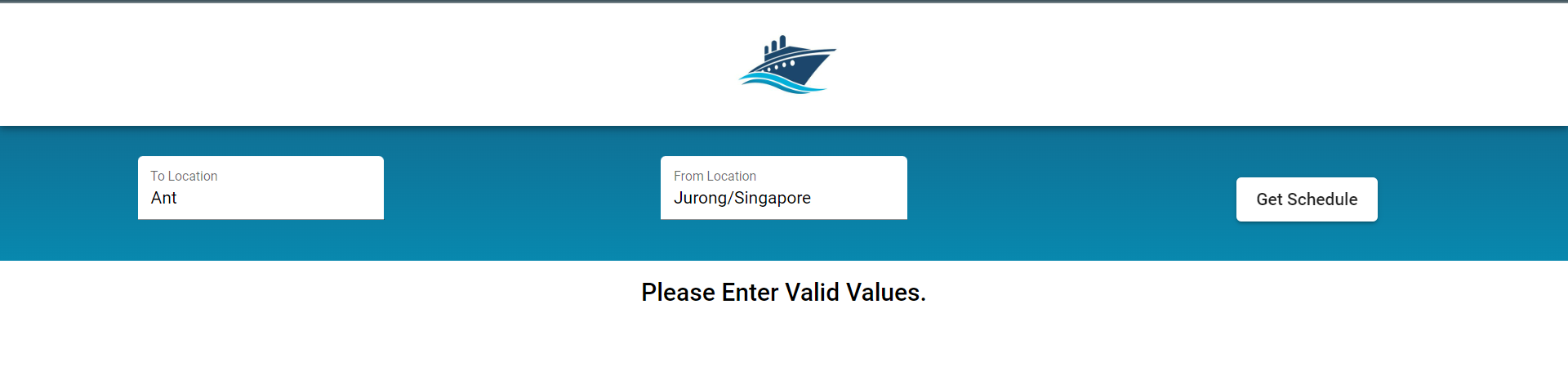
1. If no shipment is found for the given data then we can see the following message.



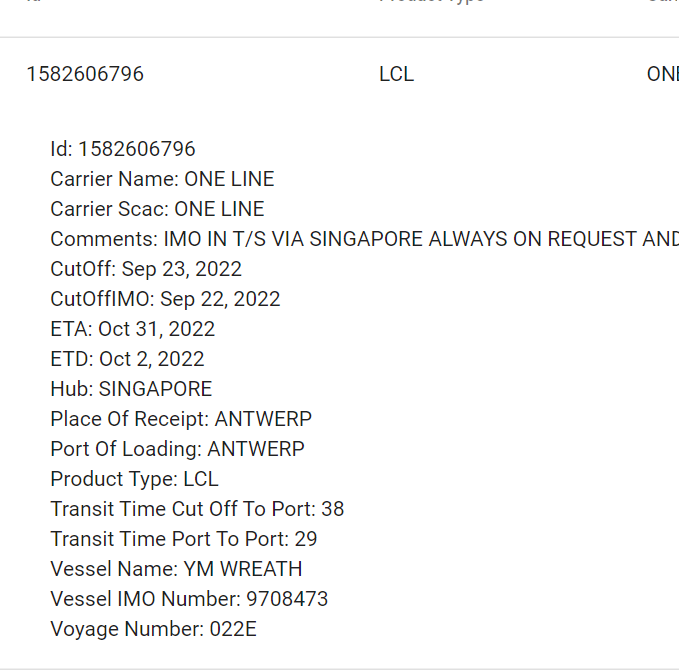
1. After getting the schedule from API you can list in the table :



1. If invalid value is enter in text box we can see the following error:

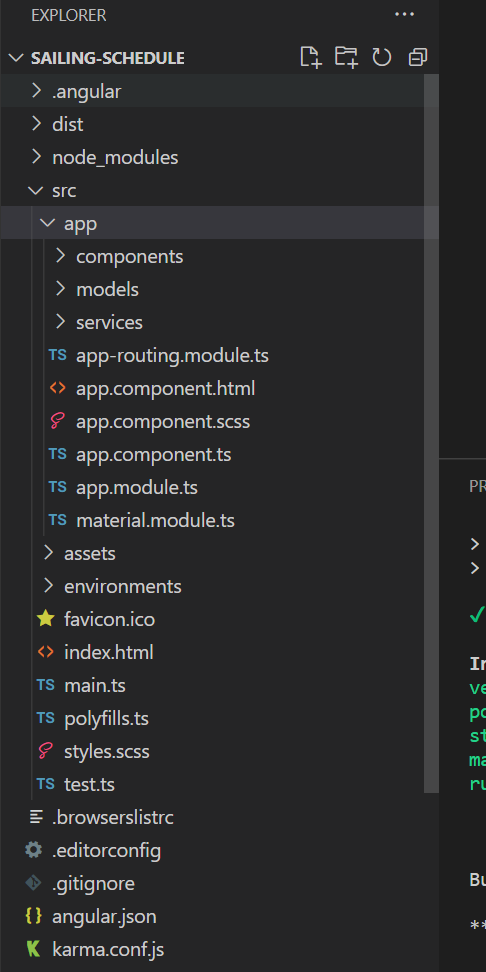


1. If you want more details on shipments then you can expand the row and see following details.



**Application Design : -**

This is a SPA application here we are showing the list of different shipments between two locations and we are using various API for those value.



We can we the project structure in above image.

Let’s talk about each folder and file in detail:

**Components folder**

This folder contains the list of all the angular components:

**Home Component**: This is the home comments of the Sailing Schedule App and contains all the other components.

**SelectLocationComponent**: This is the reusable component that has an auto-complete feature when someone types a location value it also holds a list of all the locations available

**SchedulesTableComponent:** Component to Display the List of Schedules for to and from the location.

**SpinnerComponent:** Component to display loading when API call is in progress.

**Service Folder**

This folder Contains all the services used to share common logic and communication for the applications.

**SailingSchedulesService**: This service is responsible to make API call to the server and fetch all the information related to locations and schedules.

**DialogService**: This service is used to load overlay on the application whenever needed for e.g. if an API call is going it is used to show the loading spinner

**Model Folder**

This folder contains the model for location and schedules models which we get from API