**Assessment**

**Developing an application using Angular 6**

**Instructions to use the project file provided:**

* You need to code the assessment in the folder containing partially coded project
* Read the problem statement, examples and the other details provided carefully and implement the solution
* Download the project **RestaurantBooking** in to your system and unzip it
* **RestaurantWS** folder contains the **server side code**
* **Restaurant\_FrontEnd** folder must contain the **client side code**
* Execute the **table script** given in **RestaurantWS** **folder** in your MongoDB terminal
* **DO NOT** alter the function name or the argument list of the function that is provided to you
* **DO NOT** add any new functions apart from the one given in the file where you write the solution
* Use the **bootstrap classes** wherever necessary

**Note:**

* The express application runs in port 3000
* Type npm install in both the UI and Express folders

**Problem Description:**

Infy Restaurant wants to automate the process of booking a buffet to their customers. The following functionalities are required to be implemented.

* Book a buffet
* View Booking

**In Restaurant\_FrontEnd folder:**

|  |  |
| --- | --- |
| **File name** | **Description** |
| app/app.module.ts | **Implemented** |
| app/book/booking.ts |
| app/book/mobile.ts |
| app/app-routing.module.ts | **To be Implemented** |
| app/book/book.service.ts |
| app/book/book.component.ts |
| app/book/book.component.html |
| app/view/view.service.ts |
| app/my\_date.pipe.ts |
| app/view/view.component.ts |
| app/view/view.component.html |
| app/app.component.html | **To be modified** |

**app/buffet-booking/buffet-booking.service.ts: (To be implemented)**

* The method **bookBuffet(data)** consumes the web service exposed at the URI.
  + It sends a POST request to URI “**http://localhost:3000/bookBuffet**” by passing form data
  + After sending the request, the response must be converted to an observable object of type **BuffetBooking**
  + Return the response back to the **buffet-booking.component.ts**

**app/buffet-booking/buffet-booking.component.ts: (To be implemented)**

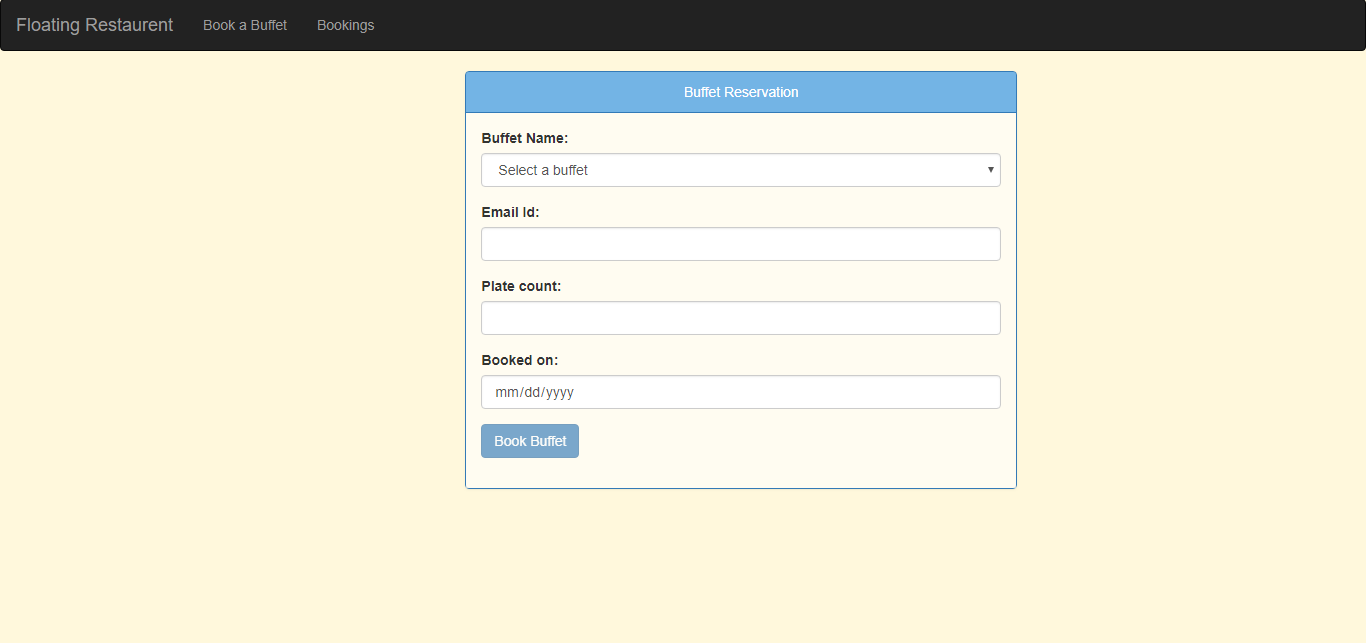
* Create the following attributes for the **BuffetBooking** class:
  + errorMessage : string
  + successMessage : string
* Inject the instances of FormBuilder and **BuffetBookingService** into the constructor
* Create a FormGroup object named **buffetBookingForm** using FormBuilder instance, having buffetName, emailId, plateCount & bookedOn as form controls
* Add the validators for each form controls as specified below:

|  |  |
| --- | --- |
| **Field name** | **Validations Required** |
| buffetName | Required validation |
| emailId | Required validation |
| plateCount | Required validation |
| bookedOn | Required validation |

* Create a method **bookBuffet ()** :
  + It should reset the errorMessage and successMessage
  + It should invoke bookBuffet() of **BuffetBookingService** by passing the value of **buffetBookingForm** object as a parameter, which in turn returns a observable
  + The Observable object returned should be subscribed by passing two callback functions i.e – success, error
  + The success callback should populate the **successMessage** with the message in response
  + The error callback should populate the **errorMessage** with the message in response

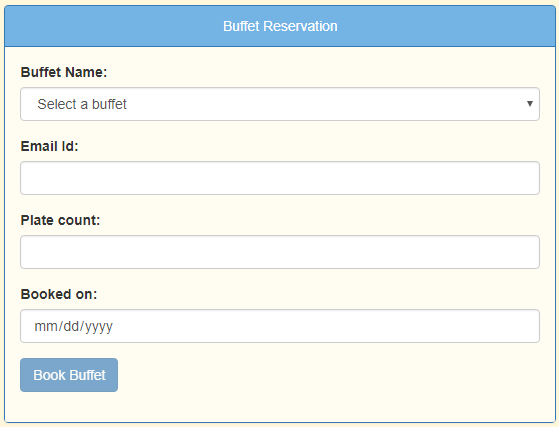
**app/app.component.html (To be modified)**

* Add the tag required to load the view of routed component
* When **“http://localhost:4200”** is entered in the browser, the **BuffetBookingComponent** should be displayed as shown below:



**app/buffet-booking/buffet-booking.component.html : (To be implemented)**

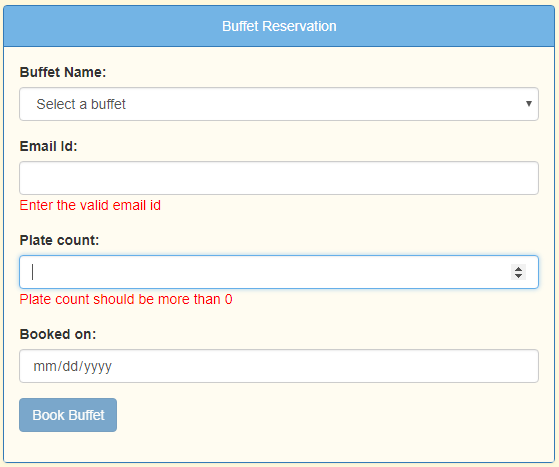
When **Book a buffet** is clicked from **app-component.html**, then initially **buffet-booking.component.html** view should be displayed as shown below:



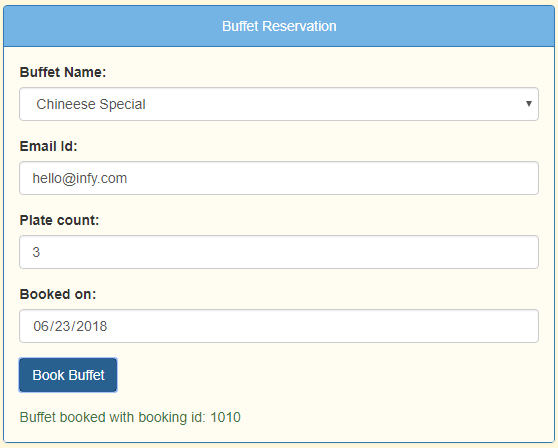
* Populate the Buffet Name field with the following static values

|  |  |  |
| --- | --- | --- |
| **Option** | **Value** | **Disabled?** |
| Select a buffet | <Empty string i.e ‘’> | Yes |
| South Indian Festival Special | SouthIndianFestivalSpecial | No |
| North Indian Festival Special | NorthIndianFestivalSpecial | No |
| Chineese Special | ChineeseSpecial | No |

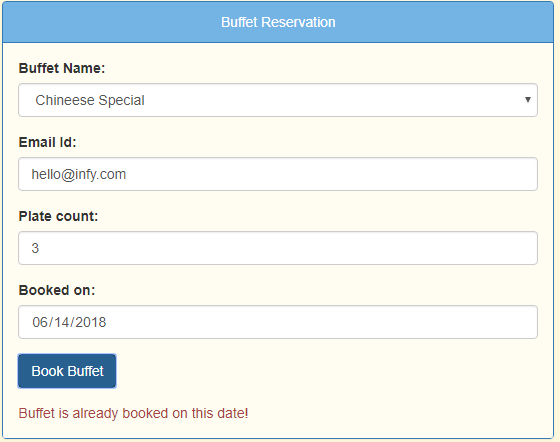
* Form should be bound to **buffetBookingForm** object
* The form fields should be bound to the form builder metadata values given in **BuffetBookingComponent class.**
* Validation messages should be rendered only when the field becomes dirty and validation fails for a form control as shown below:



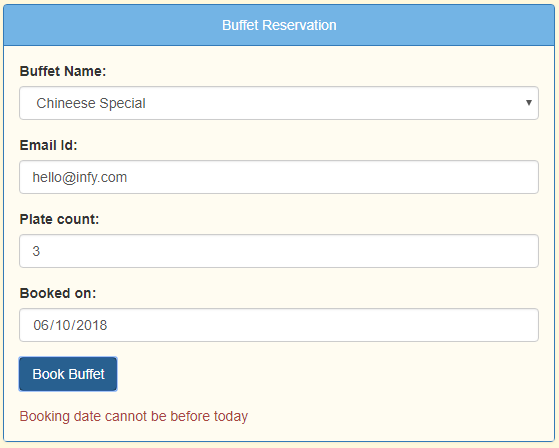
* Button should be disabled; until the form becomes valid
* When the form is submitted **bookBuffet()** of **BuffetBookingComponent** should be invoked
* On successful execution, the success message should be displayed as shown below:



* In case if you are trying to book the buffet for which it is already booked, the error message should be displayed as shown below:



* In case of user entering the date which is past date, the error message should be displayed as shown below:



**get-booking/get-booking.service.ts (To be implemented)**

* Methods  **getBooking(id)** will use **get** request at URI: “[**http://localhost:3000/fetchBooking/:id**](http://localhost:3000/fetchBooking/:id)**”** and return a Observable of **BuffetBooking.**

For example, **http://localhost:3000/fetchBooking/1010**

**get-booking/get-booking.component.ts (To be implemented)**

* Create following attribute for **GetBookingComponent** class:

errorMessage: string;

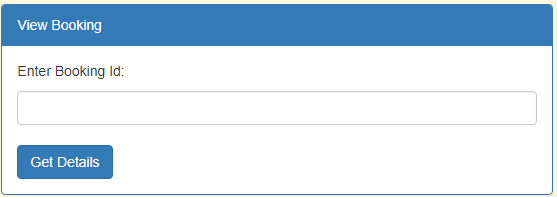
bookingId: string;

selectedBooking: BuffetBooking;

* Inject the instance of **GetBookingService** into the contructor
* Create a method called as getBooking().
* Invoke the **getBooking(bookingId)** method of **getBookingService** which in turn returns an observable.
* The Observable object returned should be subscribed by passing two callback functions i.e – **data**, **err**
  + The data callback should populate the **selectedBooking** with the data in response
  + The err callback should populate the **errorMessage** with the message in response

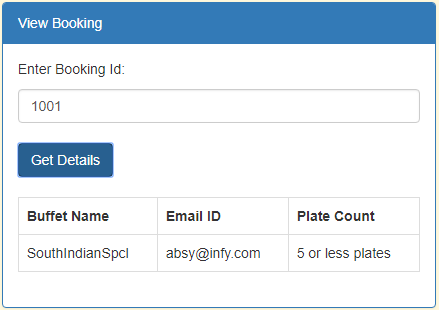
**get-booking/get-booking.component.html (To be implemented)**

When **Bookings** is clicked from **app-component.html**, then initially **get -booking.component.html** view should be displayed as shown below:

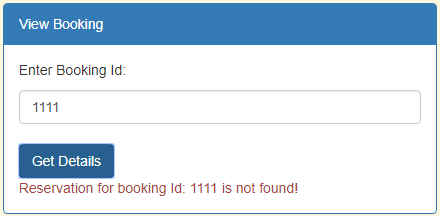


**Note: DO NOT USE a form**

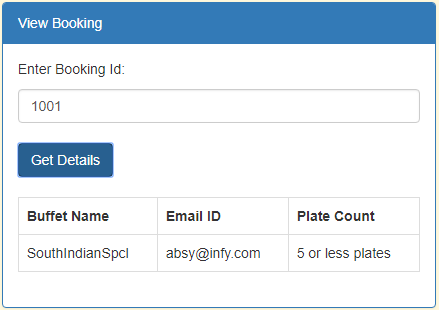
* When user enters booking id & click on get details, details of that particular booking id should be shown as below:



* If entered booking Id doesn’t exist, it should display error as per given screenshot



* If the correct booking Id is mentioned and if the “Get Details” button is clicked, then the error message ( if any ) must disappear and the appropriate details must be shown as below:



**app/plate-count.pipe.ts (To be implemented)**

* Transform the **no of plates** using this pipe
* If the no.of.plates is greater than 5, it should be transformed as “more than 5 plates”
* Else it should be transformed as “5 or less plates”

**app/app-routing.module.ts (To be implemented)**

* Configure the routes for the below information:

|  |  |
| --- | --- |
| **Path** | **Component** |
| bookBuffet | BuffetBookingComponent |
| getbooking | GetBookingComponent |

* In case an **Invalid URL** is provided, the user must be redirected to the **bookBuffet page**.

~~~~~~~~~~~~~~ All the Best ~~~~~~~~~~~~~~