CodeSPrint UX – GIft Card

Angular and Bootstrap

Please read the document carefully!

Follow the guidelines including project naming conventions!

Contents

[Technology to use: 2](#_Toc33793219)

[Template code URL: 2](#_Toc33793220)

[Use-case: 2](#_Toc33793221)

[Landing page: 3](#_Toc33793222)

[Guidelines: 3](#_Toc33793223)

[Place gift card order component: 4](#_Toc33793224)

[Guidelines: 4](#_Toc33793225)

[View gift card order component: 5](#_Toc33793226)

[Guidelines: 5](#_Toc33793227)

[Contact Us component: 5](#_Toc33793228)

[Key Notes: 5](#_Toc33793229)

[Team work: 5](#_Toc33793230)

[Repository Naming Convention (Strict): 6](#_Toc33793231)

[Example: 6](#_Toc33793232)

[Deployment/Demonstration: 6](#_Toc33793233)

[Proposed team-task distribution: 7](#_Toc33793234)

# Technology to use:

Angular, Bootstrap

# Template code URL:

<https://github.com/FSDIIHT/codesprint-ux-angularSPA>

1. Download angular template project from above github repository.
2. Open the command prompt for the folder in which downloaded and un-compressed project is present.
3. Open command promt an run following command:
   1. npm install
   2. ng serve
4. Above commands will install the dependencies and deploy the project at local host at port 4200.
5. Open another command prompt and run following command:
   1. npm run json-server
6. Above command will deploy following fake rest api’s at:
   1. <http://localhost:4444/allfriends>
   2. <http://localhost:4444/allfriends/1>
   3. <http://localhost:4444/allfriends/2>
   4. Etc
7. You may send POST, PUT, DELETE REQUEST as well. Pl. refer to below link:
   1. <https://medium.com/codingthesmartway-com-blog/create-a-rest-api-with-json-server-36da8680136d>
   2. <https://java2blog.com/json-server/>
8. Modify friends.json or rename it as per the need of requirement document

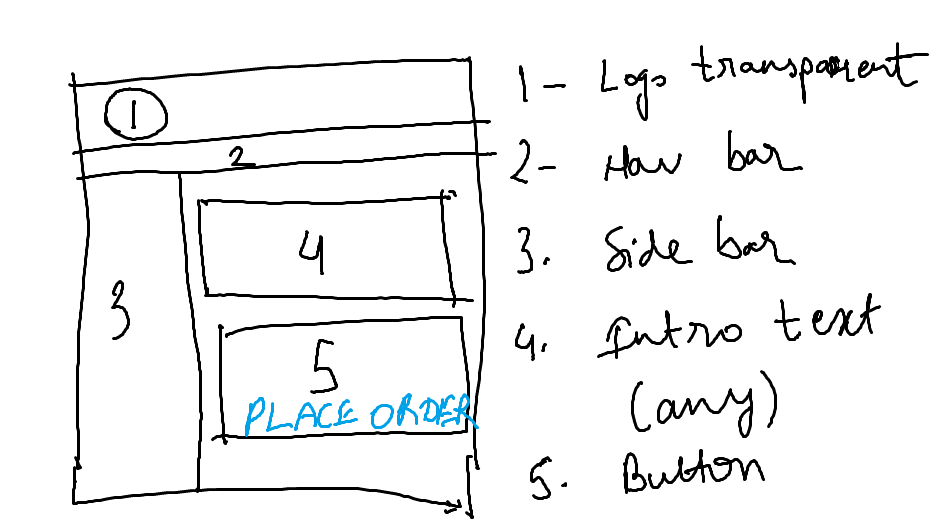
# Use-case:

Create a responsive single page app (SPA) with following components:

1. Landing page component
2. Place Gift card order page component
3. View Gift Card orders component
4. Contact Us component

**Use fake rest api created via JSON-SERVER for placing gift-card order.**

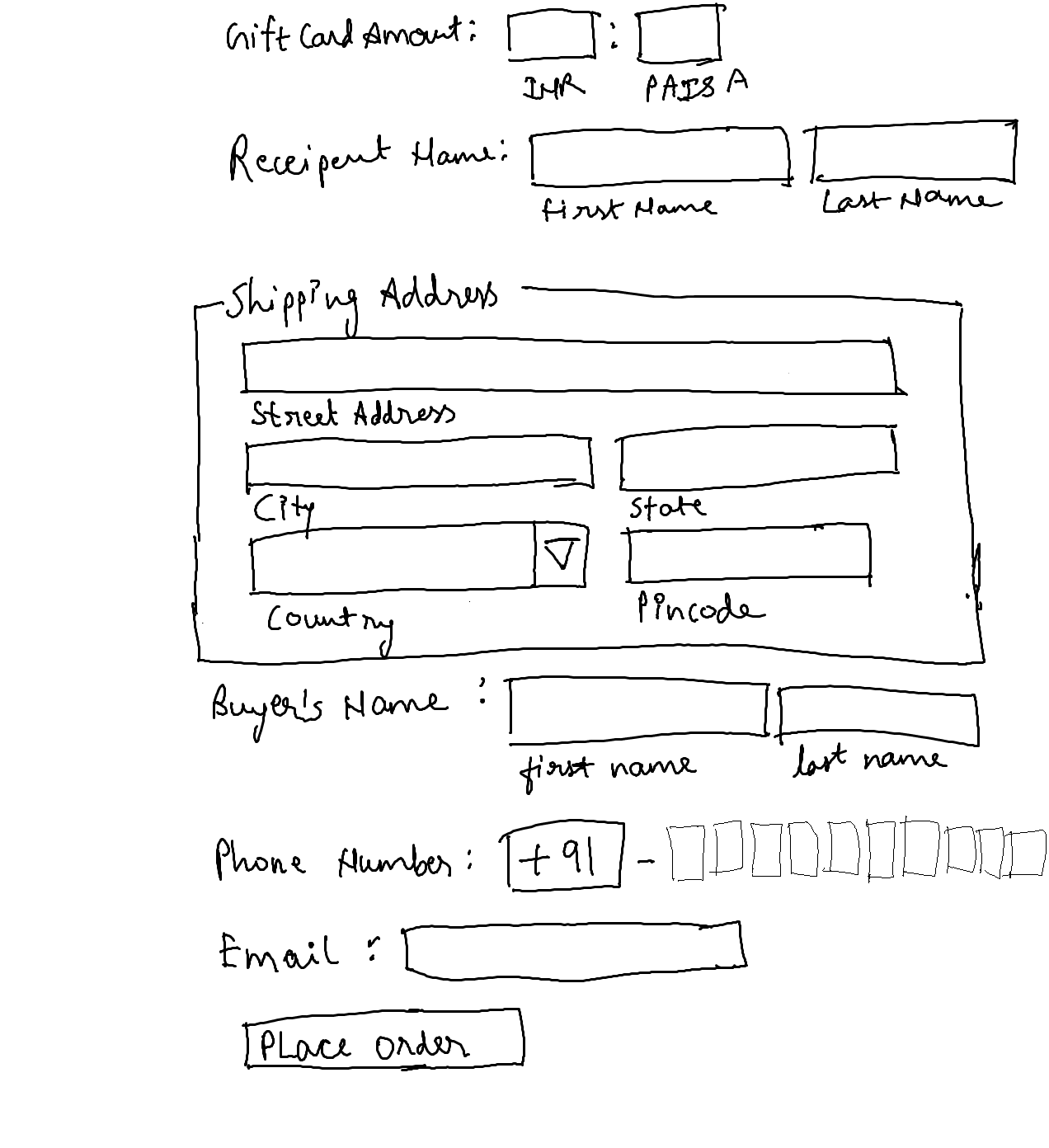
# Landing page component:



## Guidelines:

1. “Place Order” i.e. 5 is a big full page button.
2. Page is responsive.
3. Use bootstrap navbar for 2
4. Logo image must be rounded using bootstrap
5. Nav-bar component and Side-bar component will contain menu items OR routes:
   1. Home
   2. Place Gift card order
   3. View Gift Card orders
   4. Contact Us
6. Navbar component, sidebar component and logo component (all 3 are sub-components of header component) must be repeated on all pages.

# Place gift card order component:

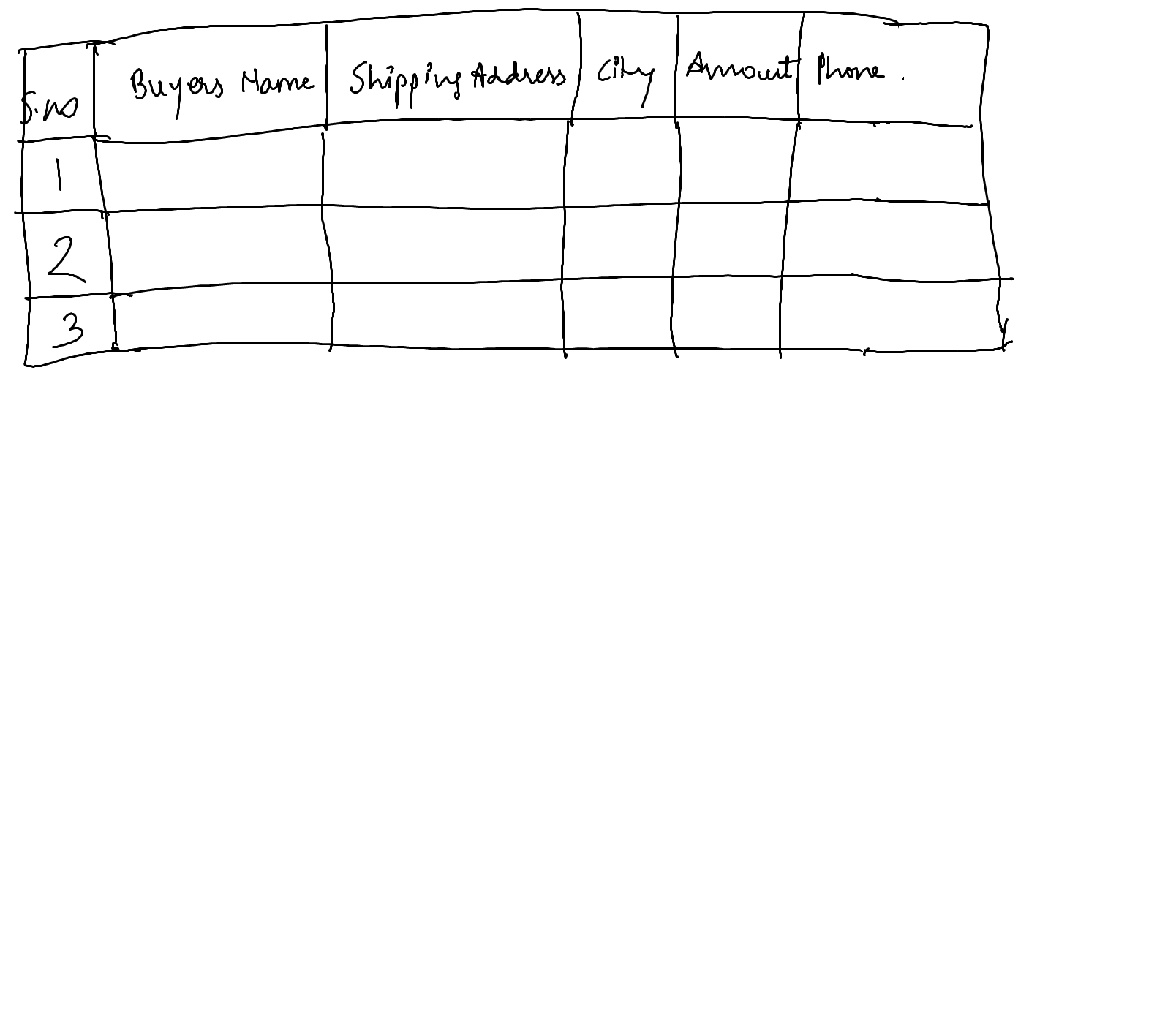


## Guidelines:

1. Implement form using reactive forms module.
2. On the page load, input focus must come to gift card amount input field.
3. You should be able to press “TAB” key and “SHIFT + TAB” to navigate from top field to bottom field and vide-versa.
4. On click of “Place order” button, gift card order details must be saved via fake REST api created with json-server.
5. Gift card amount must allow numbers only
6. Minimum value for paise must be 0.
7. Maximum value for paise must be 100.
8. Firstname and Lastname must not allow numbers.
9. Use fieldset for shipping address.
10. In shipping address, input field labels must be below input fields as shown in image.
11. Pincode must allow only number. The length of number must be 6. Length less than 6 or more than 6 is not allowed.
12. Buyers name must not allow numbers.
13. You have to implement 10 text input fields for mobile number. Each text field will allow input of 1 number between 0 to 9. First text filed must not allow 0. The input focus should move automatically from one text field to another after input of number.
14. Use email validation for email field.
15. Place order button on “place gift card order” page must be disabled till all form validations are passed. You may check the valid or invalid status of each form or form field and conditionally set the button to be enabled or disabled. Ref link is shared below.
16. <https://www.youtube.com/watch?v=y8ah0Xnr1UQ&list=PLz5rnvLVJX5VQZdTz4MTE52aKto-MoPwB&index=13&t=0s>

# View gift card order component:

It must display all gift card orders in responsive table:



## Guidelines:

1. Use ngFor directive to create table.
2. Component should inject the service, which in turn will use httpClient to consume rest api created with JSON-SERVER.
3. Use in-built pipe to display amount with INR icon.
4. Table must be responsive.

# Contact Us component:

1. Use forms module to create template form.
2. Implement generic contact us form with basic fields and basic validation.
3. It is to your discretion:
   1. To choose the layout of contact form.
   2. To choose alignment of form on the page.
   3. To implement/non-implement the validation for each field in form.
   4. To choose number of fields in contact us form.

# Key Notes:

## Team work:

1. You have to develop this UX as one of the team member.
2. Mentor will be assigning you part/module/feature of UX, to be designed by you.
3. Your team will have to integrate all components in any one machine before final presentation/demo.

## Repository Naming Convention (Strict):

1. Create two repositories in your github account.
   1. **123456-codesprint-ux-mywork**
   2. **123456-codesprint-ux-teamwork**
2. Strictly, follow the github repository naming convention, else we will not be able to track and review your submission. It may be as good as not participating in codesprint.
   1. **associateid-codesprint-ux-mywork:**
      1. **it will contain the module/feature developed by you.**
      2. **only associateid will change as per each associate. Rest of the repository name will remain same.**
      3. **associateid-codesprint-ui-mywork must contain mywork.txt file.**
      4. **you must document your work/role, describe the module developed, any challenges faced in the file mywork.txt**
   2. associateid-codesprint-ux-teamwork repository will contain complete integrated UX solution where, each component is developed by different associates.

## Example:

1. If my associate Id is 123456 and my team includes members with associate id 123457, 123458, 123459, 123460 and 123461. In above team, 123460 is team lead. Each of the team member will work on respective module/feature assigned to them. Mentor will moderate the same.
   * 1. In given scenario, my github account will contain 2 repository:
        1. 123456-codesprint-ux-mywork
        2. 123460-codesprint-ux-teamwork
        3. 123456 will commit the code in his/her repo on github so that same can be used for integration.
        4. Integrated solution will be present in 123460-codesprint-ux-teamwork
2. 123456-codesprint-ux-mywork will contain mywork.txt,
3. Mywork.txt will describe my work/role in team. The sample is given in github template.

# Deployment/Demonstration:

1. Once solution is integrated, you should create production version by using command:
   1. ng build --prod
   2. Above command will create compressed distributable. It is being done by webpack which is pre-configured by ng.
   3. Use below tools or the approach you have used in codesprint-ui to deploy the project:
      1. <https://ngrok.com/> (It may require installation)
      2. OR
      3. <https://www.npmjs.com/package/live-server>

# Proposed team-task distribution:

1. Mentor may distribute the task for the team as per his/her discretion as per local circumstances and team skills.
2. Proposed modules:
   1. Developer 1:
      1. Homepage + Contact Us Page
   2. Developer 2:
      1. Form development 1 - All fields and validations excluding shipping address
   3. Developer 3:
      1. Form development 2 – shipping address part of form
   4. Developer 4:
      1. Saving to localstorage
      2. Reading from localstorage
   5. Developer 5:
      1. View gift card orders page
   6. …
   7. …
   8. …
   9. …
   10. Developer n
       1. Some task…