

**Contents**

[1 Problem Statement 2](#_heading=h.30j0zll)

[2 Proposed Online Grocery App Wireframe 3](#_heading=h.1fob9te)

[2.1 Welcome page 3](#_heading=h.3znysh7)

[2.2 View Appointments 3](#_heading=h.2et92p0)

[2.3 Place Appointment 4](#_heading=h.tyjcwt)

[2.4 QUERY 4](#_heading=h.3dy6vkm)

[3 Business-Requirement: 6](#_heading=h.4d34og8)

# Problem Statement

1. Application will support user, vendor and admin role. Use Angular either for admin or front end.
2. Vendor will add grocery item under various category.
3. Admin is responsible for adding categories in application.
4. Admin will allow vendor to add products.
5. User can add items in cart and order products.
6. Admin will provide offers and discounts.
7. Every purchase will add points to the user wallet. Can be reimbursed during billing once reach to some point.
8. User will get offers information once logged in into this application based on their previous purchased items and amount spent.
9. User can also schedule particular items for delivery.
10. User will give feedback and ratings.
11. User can communicate with admin and vendor to resolve their query once logged in.

**The core modules of Online Grocery app are:**

1. Welcome Page
2. Gallery
3. Select Required Grocery
4. Billing Page
5. Admin

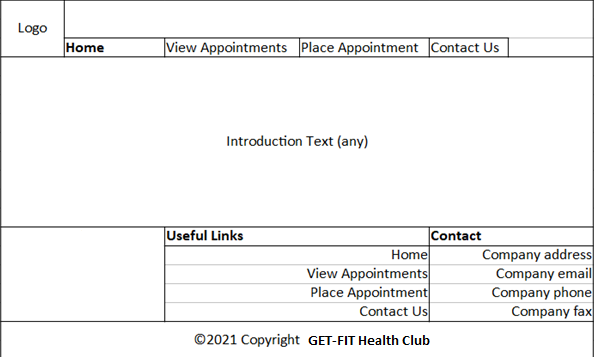
**Project Requirement**

1. **Micro Service**
2. **MySql**
3. **Rest Web Services API**

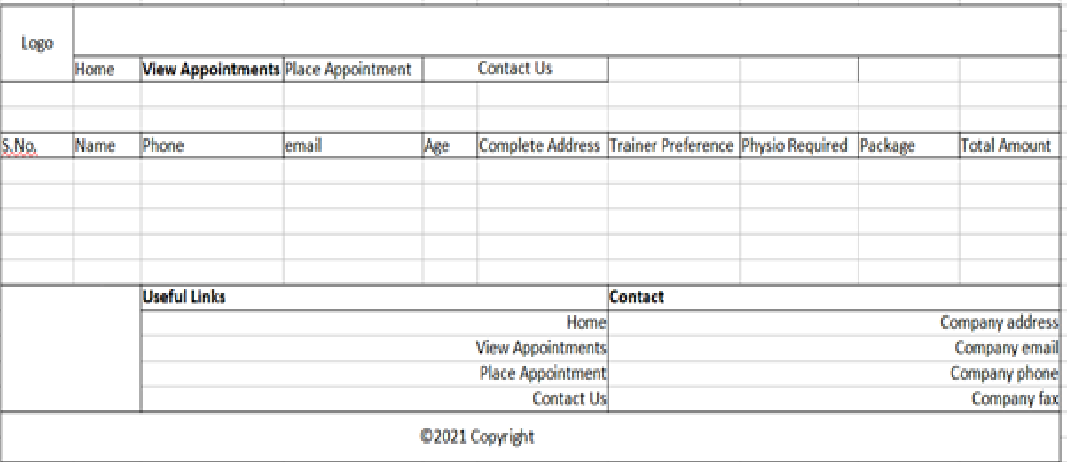
# Proposed Online Grocery app Wireframe

UI needs improvisation and modification as per given use case and to make test cases passed. **It is a sample Form.**

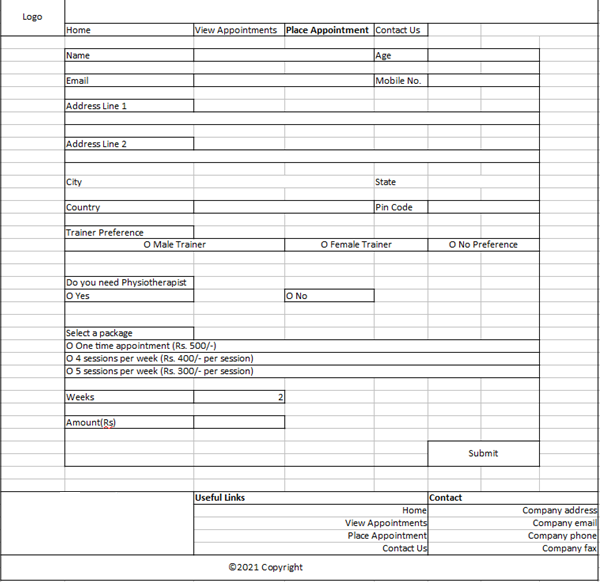
## Welcome page



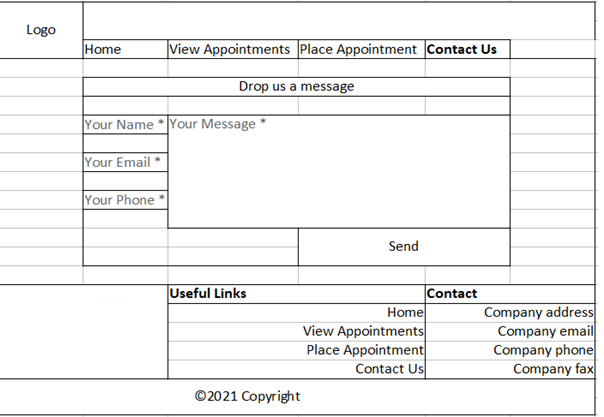
## View Appointments



## Place Appointment



## QUERY



# Business-Requirement:

As an application developer, develop the BookMyShow App (Single Page App) with below guidelines:

| User  Story # | User Story Name | User Story |
| --- | --- | --- |
| US\_01 | Welcome Page | As a user I should be able to visit the welcome page as default page.  Acceptance criteria:   1. User can click any button given in the menu bar. |
| US\_02 | Post Appointment | As a user I should be able to post an appointment  Acceptance criteria:   1. As a user I should be able to furnish following details at the time of placing an appointment    1. Name    2. Age    3. Email    4. Mobile No    5. Address Line 1    6. Address Line 2    7. City    8. State    9. Country 2. Weeks number type input box should be visible when 2nd or 3rd package option is selected. 3. If movie ticket is required add additional 200/- in final amount as Service TAX. 4. Amount should be disabled and should be calculated automatically based on the selected package. 5. All details fields must be mandatory. 6. Address line 2 may contain the same address as address line 1. 7. Email& Mobile must be unique. 8. If any constraint is not satisfied, a validation message must be shown. 9. A success or failure message should be visible after the submit button is clicked. |
| US\_03 | Manage Appointment | * 1. As a user I should be able to view all appointment requests, and after selecting any appointment   Acceptance criteria:   1. Message should be visible if no appointment is available to show. |
| US\_04 | Query | As a user I should be able to post a feedback/query/message  Acceptance criteria:   1. As a user I should be able to furnish following details at the time of filling contact us form    1. Name    2. Email    3. Phone    4. Message 2. Message should not go beyond 200 characters. 3. All four fields must be mandatory. 4. A success or failure message should be visible after the submit button is clicked. |

# Constraints

1. On the page load, input focus must come to the first name input field.
2. You should be able to press the “TAB” key and “SHIFT + TAB” to navigate from top field to bottom field and vice-versa.
3. On click of “Submit” button, appointment details must be saved via fake-rest API in grocery.json.
4. Fake rest api is implemented with json-server.

Example JSON for reference of fields to be used for placing appointment:

{

"firstname": "test",

"lastname": "test",

"age": 24,

"phonenumber": 9988776655,

"email": "test@test.com",

"streetaddress": "test",

"city": "test",

"state": "test",

"country": "india",

"pincode": 560058,

"trainerpreference": "Male Trainer",

"physiotherapist": "Yes",

"packages": "500",

"inr": 1000,

"paisa": 10,

"id": 1

}

# Mandatory Assessment Guidelines

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to**

**Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.**

1. **This editor Auto Saves the code.**
2. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
3. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
4. **This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.**
5. **You can follow series of command to setup React environment once you are in your project-name folder:**
   1. **npm install -> Will install all dependencies -> takes 10 to 15 min**
   2. **npm run start -> To compile and deploy the project in browser. You can press <Ctrl> key while clicking on localhost:8080/8081 to open project in browser -> takes 2 to 3 min**
   3. **npm run json-server -> to deploy fake rest api created with json-server -> takes 10 to 15 seconds**
   4. **npm run test -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min**
6. **You may also run “npm run jest” while developing the solution to re-factor the code to pass the test-cases.**
7. **Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on “Submit Assessment” after you are done with code.**
8. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**