

# Android Training Project

## PROBLEM STATEMENT:

Build a food ordering application with the following functionalities:

1. A **'Welcome page'** which displays the logo and/or name of the app. Choose any name of your choice.
2. A **'Login Page'** which asks for users' mobile number and password.
3. A **'Registration Page'** which enables users to sign up for the app.
4. A **'Forgot Password Page'** which enables users to reset their password.
5. A **'Navigation Drawer'** with the app logo and user name on top and menu options to open the following pages:
  - a. Home
  - b. User Profile
  - c. Favorite Restaurants
  - d. Order History
  - e. Frequently Asked Questions(FAQs)
  - f. Log out
6. The app should be able to fetch the list of all restaurant from the internet using the link provided below:  
**<http://13.235.250.119/v2/restaurants/fetch\_result/>**
7. A **'My Profile'** page (where the user's name, phone number, and address is displayed).
8. A **'Favorites'** page (where the list of all favourite restaurants is displayed).
9. An **'Order History'** page which lists the previously placed orders of the user.
10. An **'FAQ'** page which lists some frequently asked questions. You can create any random questions which you feel could be relevant for a food delivery application. (Min. 5 questions)
11. A **'Logout'** functionality which takes the user to the login page.
12. A **'Restaurant Details'** page which displays the menu items of that particular restaurant, each item's price and the option to add an item to cart.
13. A **'Cart'** which lists the items added to cart and the total amount to be paid.

We have created a sample app for you. Please refer to it for a demo. However, we encourage you to use your own custom icons, logos, names, and colour theme. You can also change the colour theme and design of the application.

[Click here to download demo app](#)

## SPECIFICATIONS:

### Welcome Page

This will be the first page which gets displayed when the user opens the app. This page will contain the app name and/or app logo. You can also use any custom background for this page. This page will be displayed for 1 second and then it moves on to the login page\* without any user interaction.

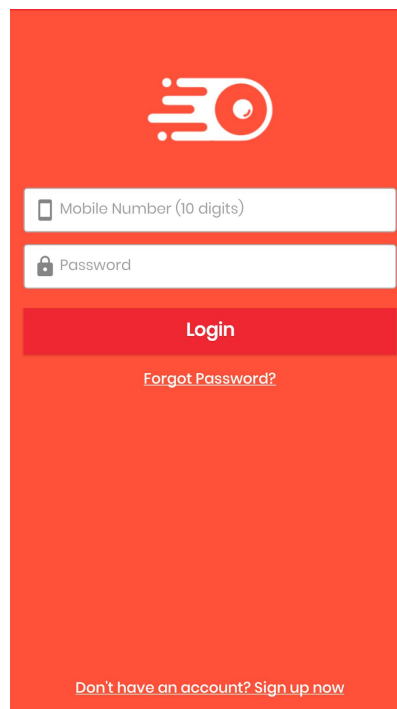
*(\*If the user has previously logged into the application, then the next page should be 'All Restaurants' page else the next page should be login page only).*

### Login Page

The login page is displayed after the welcome page. The user should be able to enter the mobile number and password to login into the app. It should perform validations which checks for the length of mobile number (should be 10), length of the password (should be more than 4) and also authenticates the login with the API given below:

**<http://13.235.250.119/v2/login/fetch\_result/>**

If the login is successful then the app is redirected to home page otherwise it should show the corresponding error. The login page should also have an option for users to register for the application if they are not already registered.

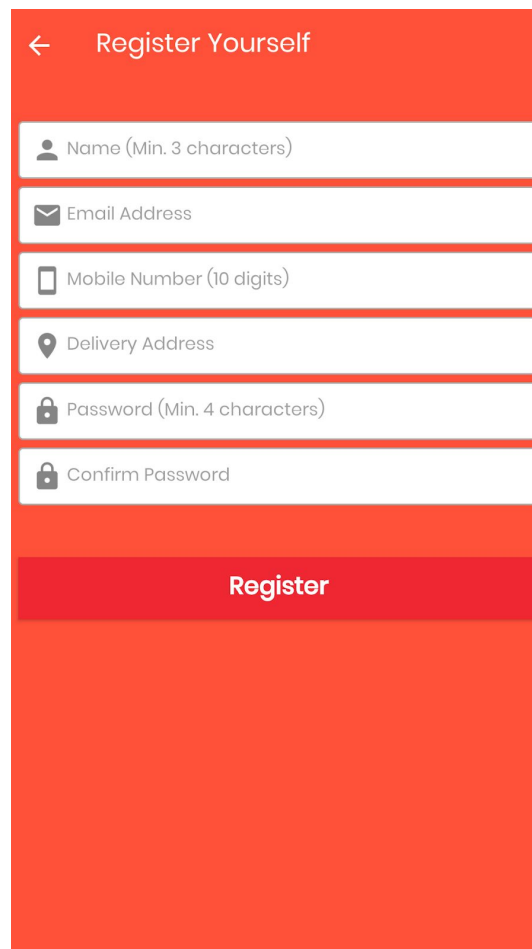
A mobile app login screen with a solid orange background. At the top center is a white logo consisting of a circle with a dot inside and three horizontal lines to its left. Below the logo are two white input fields with orange borders. The first field is labeled 'Mobile Number (10 digits)' and the second is labeled 'Password' with a small lock icon. Below these fields is a solid orange 'Login' button. Under the button is a link that says 'Forgot Password?'. At the bottom of the screen is a link that says 'Don't have an account? Sign up now'.

## Registration Page

The page should contain the following fields namely: **Name, Email Address, Mobile Number, Delivery Address, and Password** which users will input and register for the app. The registration process will send the details to the server using the link:

**<[http://13.235.250.119/v2/register/fetch\\_result](http://13.235.250.119/v2/register/fetch_result)>**

If the registration process is successful the app should take the user to the home page which is the page listing all the restaurants.



A mobile application registration form titled "Register Yourself" with a back arrow icon. The form contains six input fields, each with a placeholder icon and text: "Name (Min. 3 characters)" with a person icon, "Email Address" with an envelope icon, "Mobile Number (10 digits)" with a mobile phone icon, "Delivery Address" with a location pin icon, "Password (Min. 4 characters)" with a lock icon, and "Confirm Password" with a lock icon. Below the input fields is a red "Register" button. The entire form is set against a solid red background.

## Forgot Password Page

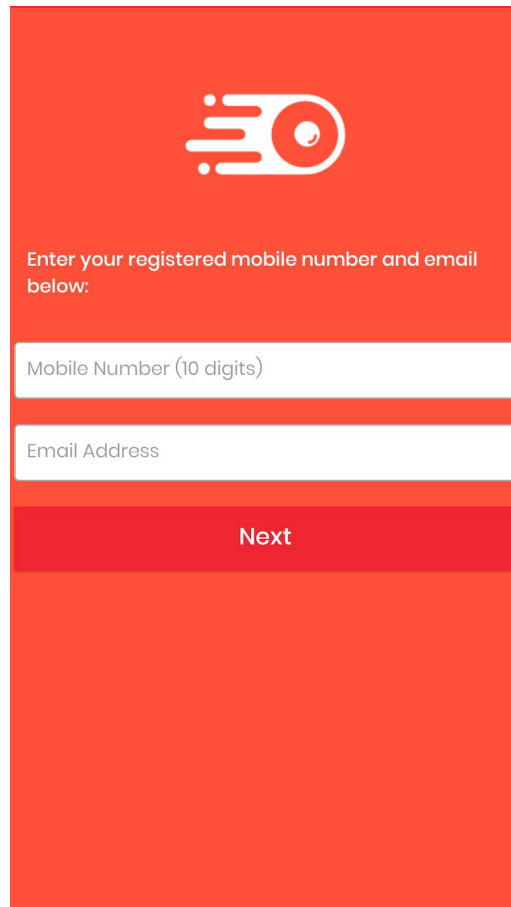
The page is used to reset the user password. This will be a two-step process.

### Step 1:

The information asked from the user will be the **Registered Mobile Number** which will be used to verify the user mobile number and the registered email address. Then you need to send the OTP via email using the below API:

<[http://13.235.250.119/v2/forgot\\_password/fetch\\_result](http://13.235.250.119/v2/forgot_password/fetch_result)>

Please note that you just need to send the mobile number to the server and the process of sending the email will be taken care of by the API itself.

The image shows a mobile application interface for a forgot password page. It has a solid orange background. At the top center is a white logo consisting of a stylized 'O' with three horizontal lines to its left. Below the logo, the text 'Enter your registered mobile number and email below:' is displayed in white. There are two white input fields stacked vertically. The first field is labeled 'Mobile Number (10 digits)' and the second is labeled 'Email Address'. Below these fields is a red button with the white text 'Next'. The bottom half of the screen is a solid orange rectangle, likely a placeholder for a footer or additional content.

Enter your registered mobile number and email below:

Mobile Number (10 digits)

Email Address

Next

**Step 2:**

Now once the user receives the OTP, on the next page we will ask him to enter the OTP and the new password, which will be sent to the server along with the registered mobile number (which is carried forward using the Bundle to this page) at the API given below which will reset the password to its new value. The user will now be redirected to the login page where he can use the new password to login into the application.

**<[http://13.235.250.119/v2/reset\\_password/fetch\\_result](http://13.235.250.119/v2/reset_password/fetch_result)>**

*(The OTP email would be sent only once in 24 hours. Hence, if you need to reset the password multiple times within 24 hours kindly use the same OTP)*

Enter the received OTP below

OTP (4 characters)

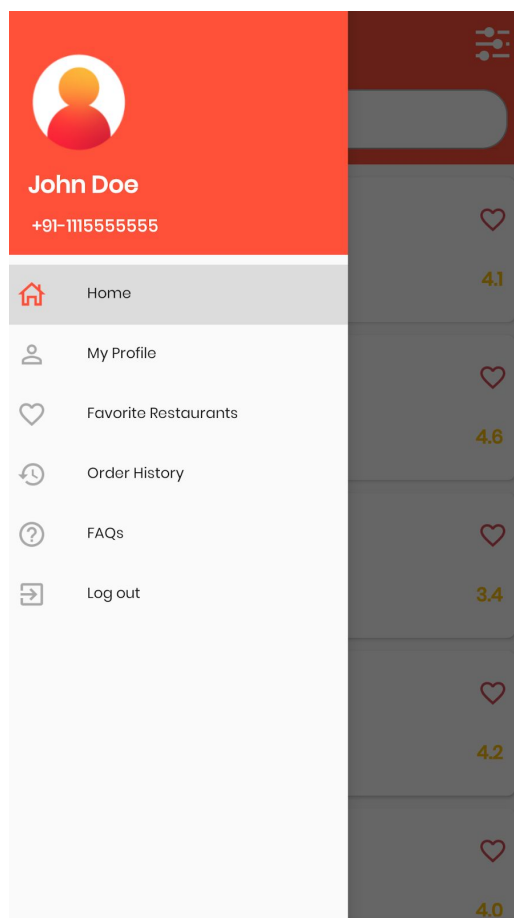
New Password (Min. 4 characters)

Confirm Password

Submit

## Navigation Drawer

The navigation drawer is needed so that the user can navigate to different pages inside the app. The user should be able to access the navigation drawer on all the app screens by clicking the hamburger button on the left side of the toolbar or by swiping right from the left edge. It will have a header section at the top which contains the app logo and below it should be the name of the user. Below the header section, there will be options for 'Home', 'My Profile', 'Favorite Restaurants', 'Order History', 'FAQ', and 'Logout' in a list style. Clicking on any option will open the corresponding screen or perform the corresponding action.



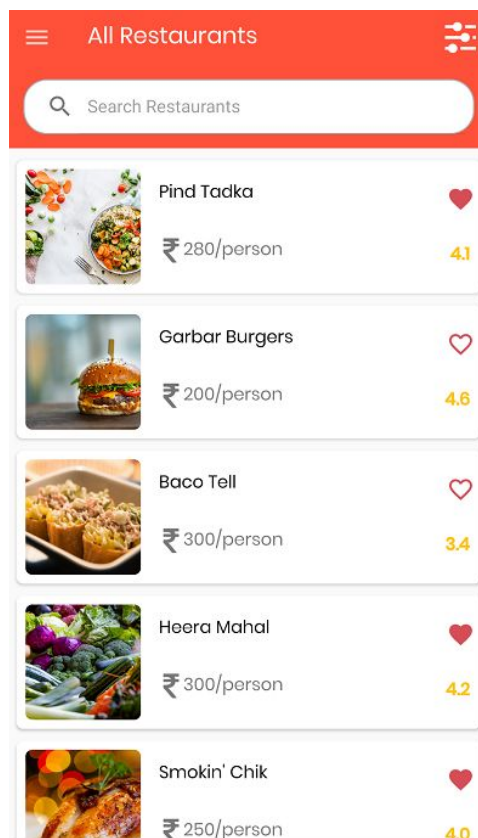
## All Restaurants Page

The 'All Restaurants' page will be the homepage of the app, meaning when the app is launched and the user has already once logged into the app then the user will see the 'All Restaurants' page screen after the splash screen. Once the app is launched, all the restaurants will be fetched from the server using the given link:

[<http://13.235.250.119/v2/restaurants/fetch\\_result/>](http://13.235.250.119/v2/restaurants/fetch_result/)

and will be displayed on the 'All Restaurants' page in the form of a list. For each restaurant on the list, the user should see the name of the restaurant, its rating, and the cost for one person. If there is an error then suitable error handling should be done. When the user clicks on any restaurant, the 'Restaurant details' page should open.

- On the homepage, the toolbar should have menu options using which the user should be able to sort the restaurant list according to cost (low to high), cost (high to low), rating, and cost for one.



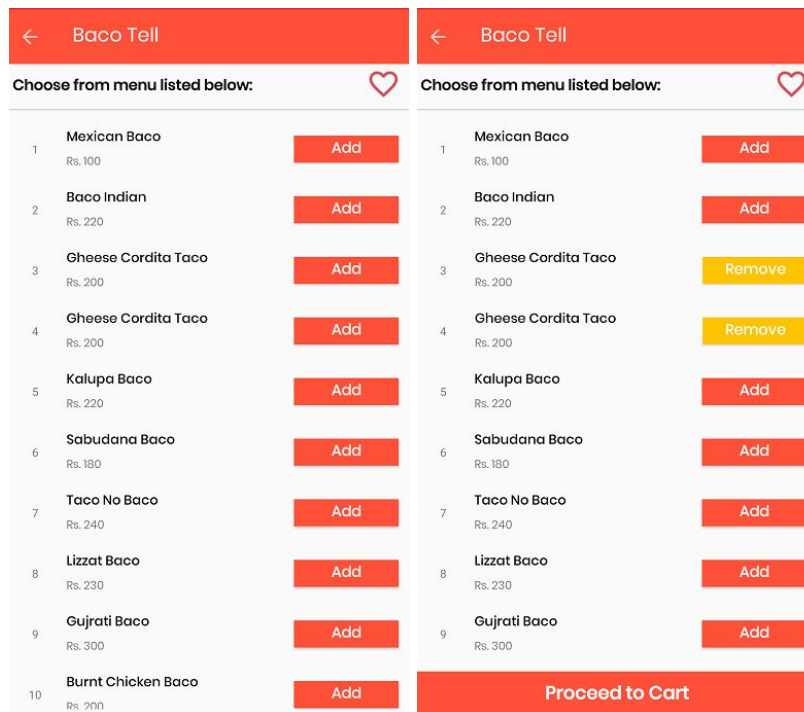
## Restaurant Details Page

The restaurant details are used to display the menu of the restaurant. When clicked on any restaurant we send a request to the server using the link:

[<http://13.235.250.119/v2/restaurants/fetch\\_result/id>](http://13.235.250.119/v2/restaurants/fetch_result/id).

We display the result of menu items in the form of a list. Each item will have a button alongside, which will be used to add or remove that particular item from the cart. Any item should be added to the cart only once. Here you need to finalise your order.

- Pressing back from this page should clear the items added in the cart.
- The proceed to cart button should be hidden when there are no items in the cart and should automatically appear when we add items to cart.
- Clicking the proceed to cart button should take the user to the cart page.





## Cart Page

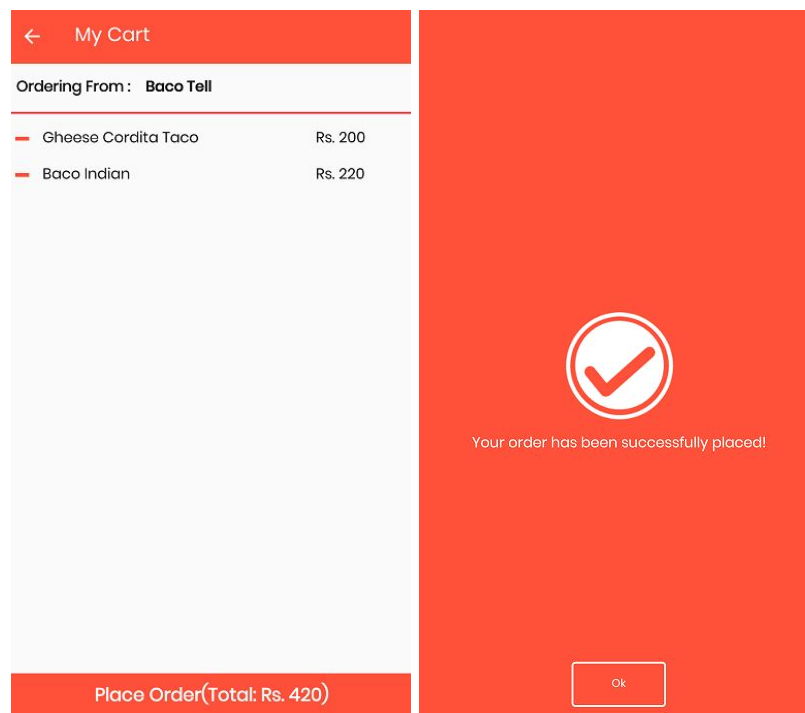
The cart page contains all the items added to the cart along with their individual prices. A total cost of all the items should be displayed on the page. These details should be fetched from the local database where you will save them while moving on to the cart page.

Here a button should also be present which can be used to place the order. Send the request to the server using this link :

**<[http://13.235.250.119/v2/place\\_order/fetch\\_result/](http://13.235.250.119/v2/place_order/fetch_result/)>.**

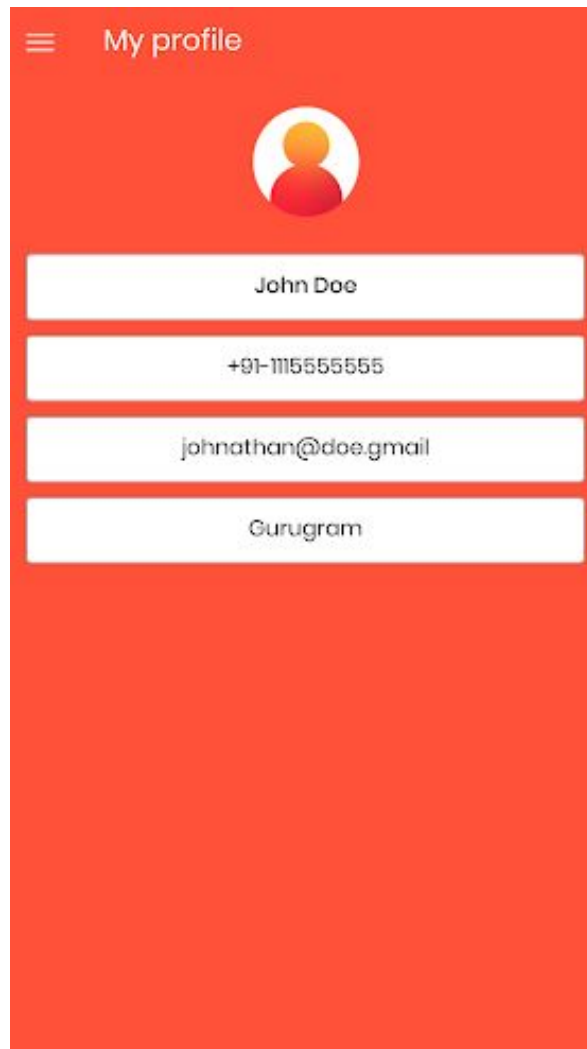
The parameters which you need to send to the server are the user\_id, res\_id and a JSON array of food\_id.

After receiving the confirmation from the server, send a notification to the user that their order has been placed and redirect them to the 'all restaurants' page which will be your homepage.



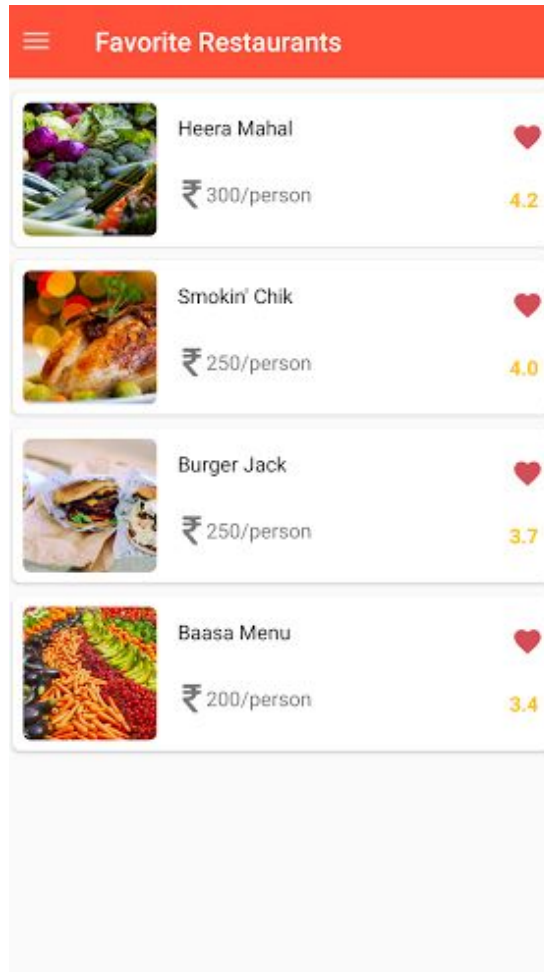
## My Profile

This page opens up from the navigation drawer. You need to put your details i.e. Name, Phone number, and Address. You can choose your own design for the screen. However, these details will not be editable.



## Favorites

This page will contain the list of the restaurants which were marked as a favourite by you. You need to fetch them from the local database. The functionality of this page will be the same as the all restaurants screen and the remaining process will remain the same.



## Order History

This page will contain the list of the food items you have previously ordered, arranged in reverse chronological order. The details to be provided on this page are:

1. Restaurant name
2. Food items list
3. Cost of each item

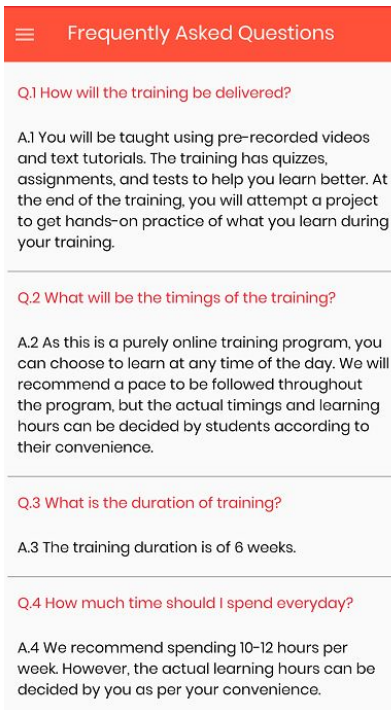
The previous order data needs to be fetched from the server using the link :

<[http://13.235.250.119/v2/orders/fetch\\_result/user\\_id](http://13.235.250.119/v2/orders/fetch_result/user_id)>

My Previous Orders	
Your previous orders are listed below :	
Pind Tadka	27/12/2018
— Kachaa Aloo Tadka	Rs. 60
— Bhajia Tadka	Rs. 60
— Mirchi Tadka	Rs. 50
— Daal No Tadka	Rs. 100
— Roti Tadka	Rs. 30
— Chole Tadka	Rs. 120
— Gobi Tadka	Rs. 80
— Murg Bhadka	Rs. 150
— Bhedu Bhadka	Rs. 180
— Chola Bhadka	Rs. 150
Heera Mahal	27/12/2018
— Hari Bhari Plate	Rs. 320
— Jogger Jagger Shake	Rs. 220
Garbar Burgers	27/12/2018

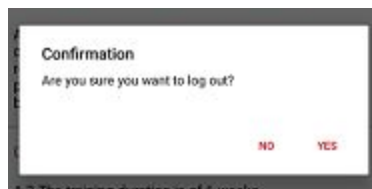
## Frequently Asked Question(FAQ) page

This page will contain a static list of questions. It is up to you as to what questions you want to put there. This page will contain static data i.e. some questions and answers. You can hardcode these questions and answers.



## Logout

This will be an option provided in the navigation drawer to log out of the application. This will be a simple log out which redirects the user to the login page while clearing all the preferences stored. Make sure that after logout, if the user presses the back button then the app should exit and not take the user back to the home page.



All the best!

## Request and Responses of the different APIs:

### 1. Register:

- a. Request Method - POST
- b. URL - [http://13.235.250.119/v2/register/fetch\\_result](http://13.235.250.119/v2/register/fetch_result)
- c. Params -

```
{  
    "name": "John Doe",  
    "mobile_number": "9998886666",  
    "password": "123456",  
    "address": "Gurugram",  
    "email": "john@doe.com"  
}
```

- d. Response -

```
{  
    "data": {  
        "success": true,  
        "data": {  
            "user_id": "30",  
            "name": "John Doe",  
            "email": "john@doe.com",  
            "mobile_number": "9998886666",  
            "address": "Gurugram"  
        }  
    }  
}
```

*\*\*Save the values in shared preferences so that in the profile fragment you will only need to extract the values from the shared preferences.*

## 2. Login:

- a. Request Method - POST
- b. URL - [http://13.235.250.119/v2/login/fetch\\_result](http://13.235.250.119/v2/login/fetch_result)
- c. Params -

```
{  
    "mobile_number": "9998886666",  
    "password": "123456"  
}
```

- d. Response -

```
{  
    "data": {  
        "success": true,  
        "data": {  
            "user_id": "30",  
            "name": "John Doe",  
            "email": "john@doe.com",  
            "mobile_number": "9998886666",  
            "address": "Gurugram"  
        }  
    }  
}
```

*\*\*Save the values in shared preferences so that in the profile fragment you will only need to extract the values from the shared preferences.*

### 3. Forgot Password:

- a. Request Method - POST
- b. URL - [http://13.235.250.119/v2/forgot\\_password/fetch\\_result](http://13.235.250.119/v2/forgot_password/fetch_result)
- c. Params -

```
{  
  "mobile_number": "9998886666",  
  "email": "john@doe.com"  
}
```

- d. Response -

- i. When the request is sent for the first time, we get the response given below along with an OTP sent to the registered email id.

```
{  
  "data": {  
    "success": true,  
    "first_try": true  
  }  
}
```

- ii. When the request is sent from the next time onwards, the response given below is obtained.

```
{  
  "data": {  
    "success": true,  
    "first_try": false  
  }  
}
```



#### 4. Reset Password:

- a. Request Method - POST
- b. URL - [http://13.235.250.119/v2/reset\\_password/fetch\\_result](http://13.235.250.119/v2/reset_password/fetch_result)
- c. Params -

```
{  
  "mobile_number": "9998886666",  
  "password": "654321",  
  "otp": "5989"  
}
```

*\*\*The OTP will be the one sent to your registered email ID.*

- d. Response -

```
{  
  "data": {  
    "success": true,  
    "successMessage": "Password has successfully changed."  
  }  
}
```

*\*\* Redirect the user to the login page after this and clear the shared preferences.*

## 5. All Restaurants:

- a. Request Method - GET
- b. URL - [http://13.235.250.119/v2/restaurants/fetch\\_result/](http://13.235.250.119/v2/restaurants/fetch_result/)
- c. Params - null
- d. Response -

```
{
  "data": {
    "success": true,
    "data": [
      {
        "id": "1",
        "name": "Pind Tadka",
        "rating": "4.1",
        "cost_for_one": "280",
        "image_url":
"https://images.pexels.com/photos/1640777/pexels-photo-1640777.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=650&w=940"
      },
      {
        "id": "2",
        "name": "Garbar Burgers",
        "rating": "4.6",
        "cost_for_one": "200",
        "image_url":
"https://images.pexels.com/photos/1639565/pexels-photo-1639565.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=650&w=940"
      },
      {
        "id": "3",
        "name": "Baco Tell",
        "rating": "3.4",
        "cost_for_one": "300",
        "image_url":
"https://images.pexels.com/photos/674578/pexels-photo-674578.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500"
      },
      {
        "id": "4",
        "name": "Heera Mahal",
        "rating": "4.2",
        "cost_for_one": "300",
```

```

        "image_url":
"https://images.pexels.com/photos/1300972/pexels-photo-1300972.jpeg?auto=
compress&cs=tinysrgb&dpr=1&w=500"
    },
    {
        "id": "5",
        "name": "Smokin' Chik",
        "rating": "4.0",
        "cost_for_one": "250",
        "image_url":
"https://images.pexels.com/photos/265393/pexels-photo-265393.jpeg?auto=co
mpress&cs=tinysrgb&dpr=1&w=500"
    },
    {
        "id": "6",
        "name": "Swirley's Shack",
        "rating": "3.8",
        "cost_for_one": "400",
        "image_url":
"https://images.pexels.com/photos/699544/pexels-photo-699544.jpeg?auto=co
mpress&cs=tinysrgb&dpr=1&w=500"
    },
    {
        "id": "7",
        "name": "Dominoe's bread",
        "rating": "3.6",
        "cost_for_one": "200",
        "image_url":
"https://images.pexels.com/photos/905847/pexels-photo-905847.jpeg?auto=co
mpress&cs=tinysrgb&dpr=1&w=500"
    },
    {
        "id": "8",
        "name": "Everything but Food",
        "rating": "3.2",
        "cost_for_one": "150",
        "image_url":
"https://images.pexels.com/photos/5938/food-salad-healthy-lunch.jpg?auto=
compress&cs=tinysrgb&dpr=2&h=650&w=940"
    },
    {
        "id": "9",
        "name": "LFC",
        "rating": "4.0",
        "cost_for_one": "200",

```

```

        "image_url":
"https://images.pexels.com/photos/60616/fried-chicken-chicken-fried-crunc
hy-60616.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500"
    },
    {
        "id": "10",
        "name": "Central Terk",
        "rating": "5.0",
        "cost_for_one": "300",
        "image_url":
"https://images.pexels.com/photos/434213/pexels-photo-434213.jpeg?auto=co
mpress&cs=tinysrgb&dpr=1&w=500"
    },
    {
        "id": "11",
        "name": "Mitti ke Sandwiches",
        "rating": "4.0",
        "cost_for_one": "250",
        "image_url":
"https://images.pexels.com/photos/1600711/pexels-photo-1600711.jpeg?auto=
compress&cs=tinysrgb&dpr=2&h=650&w=940"
    },
    {
        "id": "13",
        "name": "Pizza Put",
        "rating": "4.4",
        "cost_for_one": "350",
        "image_url":
"https://images.pexels.com/photos/724216/pexels-photo-724216.jpeg?auto=co
mpress&cs=tinysrgb&h=650&w=940"
    },
    {
        "id": "14",
        "name": "Burger Jack",
        "rating": "3.7",
        "cost_for_one": "250",
        "image_url":
"https://images.pexels.com/photos/983297/pexels-photo-983297.jpeg?auto=co
mpress&cs=tinysrgb&dpr=2&h=650&w=940"
    },
    {
        "id": "15",
        "name": "Rotten Tomatoes",
        "rating": "3.2",
        "cost_for_one": "100",

```

```

        "image_url":
"https://images.pexels.com/photos/428301/pexels-photo-428301.jpeg?auto=co
mpress&cs=tinysrgb&dpr=1&w=500"
    },
    {
        "id": "16",
        "name": "McDonald's",
        "rating": "3.6",
        "cost_for_one": "150",
        "image_url":
"https://images.pexels.com/photos/551991/pexels-photo-551991.jpeg?auto=co
mpress&cs=tinysrgb&dpr=2&h=650&w=940"
    },
    {
        "id": "17",
        "name": "Askin' Poppins",
        "rating": "4.1",
        "cost_for_one": "300",
        "image_url":
"https://images.pexels.com/photos/3631/summer-dessert-sweet-ice-cream.jpg
?auto=compress&cs=tinysrgb&dpr=1&w=500"
    },
    {
        "id": "18",
        "name": "Baasa Menu",
        "rating": "3.4",
        "cost_for_one": "200",
        "image_url":
"https://images.pexels.com/photos/264537/pexels-photo-264537.jpeg?auto=co
mpress&cs=tinysrgb&dpr=1&w=500"
    }
]
}
}

```

## 6. Restaurant Menu:

- a. Request Method - GET
- b. URL - [http://13.235.250.119/v2/restaurants/fetch\\_result/<restaurant\\_id>](http://13.235.250.119/v2/restaurants/fetch_result/<restaurant_id>)  
\*\*<restaurant\_id> - use the string template to add the restaurant ID at the end of the URL.
- c. Params - null
- d. Response - (for restaurant\_id = 5)

```
{
  "data": {
    "success": true,
    "data": [
      {
        "id": "50",
        "name": "Vegetarian Chicken",
        "cost_for_one": "340",
        "restaurant_id": "5"
      },
      {
        "id": "51",
        "name": "Lamb Chicken",
        "cost_for_one": "320",
        "restaurant_id": "5"
      },
      {
        "id": "52",
        "name": "Smoking Chicken",
        "cost_for_one": "200",
        "restaurant_id": "5"
      },
      {
        "id": "53",
        "name": "Chicken and Coke",
        "cost_for_one": "240",
        "restaurant_id": "5"
      },
      {
        "id": "54",
        "name": "Nanha Chicken",
        "cost_for_one": "180",
        "restaurant_id": "5"
      },
      {

```

```
        "id": "55",
        "name": "Chicken Khurana",
        "cost_for_one": "440",
        "restaurant_id": "5"
    },
    {
        "id": "56",
        "name": "Chicken ki khees",
        "cost_for_one": "240",
        "restaurant_id": "5"
    },
    {
        "id": "57",
        "name": "Chicken ke ladoo",
        "cost_for_one": "280",
        "restaurant_id": "5"
    }
]
}
```

## 7. Place Order:

- a. Request Method - POST
- b. URL - [http://13.235.250.119/v2/place\\_order/fetch\\_result/](http://13.235.250.119/v2/place_order/fetch_result/)
- c. Params -

```
{
  "user_id": "99",
  "restaurant_id": "5",
  "total_cost": "660",
  "food": [
    {
      "food_item_id": "50"
    },
    {
      "food_item_id": "51"
    }
  ]
}
```

- d. Response -

```
{
  "data": {
    "success": true
  }
}
```



## 8. Order History:

a. Request Method - GET

b. URL - [http://13.235.250.119/v2/orders/fetch\\_result/<user\\_id>](http://13.235.250.119/v2/orders/fetch_result/<user_id>)

**\*\*<user\_id>** - use the string template to add the user ID at the end of the URL.

c. Params - null

d. Response -

```
{
  "data": {
    "success": true,
    "data": [
      {
        "order_id": "9",
        "restaurant_name": "Smokin' Chik",
        "total_cost": "660",
        "order_placed_at": "17-12-19 11:37:53",
        "food_items": [
          {
            "food_item_id": "50",
            "name": "Vegetarian Chicken",
            "cost": "340"
          },
          {
            "food_item_id": "51",
            "name": "Lamb Chicken",
            "cost": "320"
          }
        ]
      },
      {
        "order_id": "8",
        "restaurant_name": "Pind Tadka",
        "total_cost": "340",
        "order_placed_at": "16-12-19 19:42:17",
        "food_items": [
          {
            "food_item_id": "2",
            "name": "Bhajia Tadka",
            "cost": "60"
          },
          {
            "food_item_id": "4",
            "name": "Daal No Tadka",

```

```
        "cost": "100"
    },
    {
        "food_item_id": "15",
        "name": "Bhеду Bhadka",
        "cost": "180"
    }
]
}
}
```

