

# Test Suites

Social Servicing Food Delivery System (SSFDS) Group Gautam Kumar(21CS30020), Ajay Kumar Dhakar(21CS3002), Sakshi Agrawal(21CS30045)

Initial Database:

Type	Email - Id	Password	Username	Image	Location
NGO	type1_NGO@gmail.com	Type1	User_ngo	--	Kharagpur, West Bengal, India
Customer	Type2_customer@gmail.com	Type2	User_customer	--	Kharagpur, West Bengal, India
Restaurant	Type3_restaurant@gmail.com	Type3	User_Restaurant	--	Kharagpur, West Bengal, India

Food item added by Restaurant user:

Name	Image	Price	No. of available Plates
Chole Kulche	--	120/-	20
Rassgulla	--	20/-	30

## Front-End Testing

### 1. Home Page:

- It has two links Login and Register which will route to the Login and Signup page.

### 2. Register Page:

- It is the common page for all the users(Customer, NGO, Restaurants)
- It shows a form with fields and options for roles to be filled
- It has the button Signup which signup the user and put it in the database
- If the email or username is already signed up, then it will show the error message
- If everything is fine user can sign up through the sign up button.

### 3. Login Page:

- a. It has two fields
  - i. Username
  - ii. Password
- b. If the user enter the correct username and password ,he/she will be logged in but if didn't enters both the fields correct an alert of "invalid username/password" will popup
- c. Example:
  - i. Satisfied Credential
    - `1. Input
      - a. Username: User\_ngo
      - b. Password: Type1
    - 2. Output: Alert("Logged in")
  - ii. Unsatisfied Credential
    - `1. Input
      - c. Username: @User\_ngo
      - d. Password: Type1
    - 2. Output: Alert("invalid username/password")
  - iii. Unsatisfied Credential
    - `1. Input
      - a. Username: User\_ngo
      - b. Password: @Type1
    - 2. Output: Alert("invalid username/password")

#### 4. Logout

- a. After logging in user can see the logout link on the top of navbar

#### 5. Navbar

- a. If the user is not logged in navbar will have the link of signup and login
- b. If the user is logged in user can check profiles, Restaurant, Order history, donation and cart.

## 6. Profiles

- a. It will show Name, Location and Role of the User with **Profile Image** if provided during registration.
- b. It will also show a map with the location of the user marked on it.
- c. Example:
  - i. Database:-
    - 1. Username: User\_Restaurant
    - 2. Location: Kharagpur
    - 3. Coordinates: Decoded with the help of API Mapbox
    - 4. Role: Restaurant
    - 5. **Image: Link**

## 7. Restaurant:

- a. It shows all the available restaurant under the distance 10km
- b. View food button will be on the every card of restaurant which will take u too the menu of the restaurant.

## 8. Menu

- a. In menu, Restaurant owner can see all the food registered by him
- b. There is the button for adding food
- c. For every food u can update the price or amount of the food
- d. Delete button will remove the food from the menu.

## 9. Add Food

- a. It renders a form with some required fields.
- b. If the required fields(like the columns in the database shown above) are not filled then it will show an appropriate message.
- c. If all required fields are filled then it will show positive feedback.
- d. If Input for Price or Number of plates are not Numbers then it will show an appropriate message.

- e. If Input for Price or Number of Plates are zero or negative then it will show an appropriate message.
  - i. If the owner tries to add an item whose quantity is -4, then he gets an error.
- f. It has a button to Submit the details of Food item.
- g. After Submitting, It will redirect to the menu of the restaurant with the message "Food item is Added to the Menu".
- h. Response can be verified in the Menu of the Restaurant

## **10. Edit food**

- a. Edit food button is available in the menu section for the restaurant
- b. Restaurant user can update the food price, no. of available plate or description of the food.
- c. If Input for Price or Number of plates are not numbers then it will show an appropriate message
- d. Edit food button will update all the fields for the food.

## **11. Remove Food**

- a. Remove food button is available in the menu for the restaurant
- b. It removes the food from the menu
- c. Only restaurant owner can remove their food.

## **12. Add to Cart**

- a. Add to Cart button is available in the view food section for the customer and NGO
- b. User can add food in their cart using this button
- c. If the no. of plates is 0 the food will not be added in the cart.

## **13. Cart**

- a. Cart link is present in navbar for Customer/NGO
- b. It will redirect to the cart page
- c. The food added by user from the menu will be viewed here
- d. User can remove or update the no. of plates of the food here and save the changes using save changes button.

- e. There is also a pay offline button which will

#### 14. Remove Cart food

- a. Remove button is available in the cart
- b. It removes the added food from the cart
- c. Result of removed food can be seen as a change in total amount.

#### 15. Save Changes

- a. It updates the no. of plates in the cart
- b. User can change the required plate no. of plates and update using save change button.

#### 16. Admin

- a. Admin can check all the registered user(customer, NGO and Restaurant)
- b. Admin can change the Time window of the restaurant so that after that time left over food can be donated to poor and needful person.

## BACKEND SERVER TESTING

### 1. REGISTER

- a. Input for register will be given through user client(frontend)
- b. ACTION: Adds a new user to the database who can logged in using login
- c. OUTPUT: Add a new user
- d. Database:

Type	Email - Id	Password	Username	Image	Location
NGO	type1_NGO@gmail.com	Type1	User_ngo	--	Kharagpur, West Bengal, India
Customer	Type2_customer@gmail.com	Type2	User_customer	--	Kharagpur, West Bengal, India
Restaurant	Type3_restaurant@gmail.com	Type3	User_Restaurant	--	Kharagpur, West Bengal, India

Admin	admin@gmail.com	Type4	User_Admin	--	Kharagpur, West Bengal, India
-------	-----------------	-------	------------	----	-------------------------------------

## 2 Restaurant

- PRE-CONDITION:- Let the initial Database is given as per the above.
- INPUT: Click On the Button Restaurants
- ACTION: GET All Restaurants as per the User Location
- OUTPUT: Send The Data of All Restaurants

## 3. Menu

- PRE-CONDITION:- Let the initial Database is given as per the above.
- INPUT: Click On the Button View Menu of particular Restaurants
- ACTION: GET All Foods in the Menu of the Restaurants
- OUTPUT: Send The Data of All Foods in the Menu

## 4. Add Food

- PRE-CONDITION:- Let the initial Database is given as per above
- INPUT: Add food button is available for restaurant by doing this restaurant user can fill the form to add the food in the menu.

Example:

- Name: Dora Cake
- Price: 100/-
- No. of plates: 15

- ACTION: Add food in the menu of restaurant
- OUTPUT: Add food to the database of the menu and directs to the menu.
- DATABASE:

Name	Image	Price	No. of available Plates
Chole Kulche	--	120/-	20
Rassgulla	--	20/-	30
Dora Cake	--	100/-	15

## 5. Edit Food

- PRE-CONDITION: The good going to be edited should be in database.
- INPUT: Edit food button is available in the my menu section for restaurant.

- c. EVENT: After giving an event, in response a form will appear and after satisfying all fields food will be updated.

Example:

- i. Name: Rassgulla
  - j. Price: 25/-
  - k. Count:35
- d. ACTION: The food will be updated with new values.
- e. OUTPUT: Update the food in the database.
- f. Database:

Name	Image	Price	No. of available Plates
Chole Kulche	--	120/-	20
Rassgulla	--	25/-	35
Dora Cake	--	100/-	15

## 6. Remove food

- a. PRE-CONDITION:- Let the initial Database is given as per above
- b. INPUT: Remove button is available for restaurant by doing this restaurant user can remove the food in the menu.
- c. ACTION: Delete the food from the database
- d. OUTPUT: Menu will be update after removing that food.
- e. Database:

Name	Image	Price	No. of available Plates
Chole Kulche	--	120/-	20
Dora Cake	--	100/-	15

## 7. ADD TO CART

- a. PRE-CONDITION:- Let the initial Database is given as per the above.
- b. INPUT: Number Of Plates
- c. ACTION: Add Food to the Cart of the User
- d. OUTPUT: Add Food to the Database and Cart, and redirects to Menu.

## **8. CART**

- a. PRE-CONDITION:- Let the initial Database is given as per the above.
- b. INPUT: Click On the Button Cart
- c. ACTION: GET All Foods in the Cart
- d. OUTPUT: Send The Data of All Food in the Cart

## **9. EDIT CART FOOD**

- a. PRE-CONDITION:- Let the initial Database is given as per the above.
- b. INPUT: Number Of Plates
- c. ACTION: Edit Food to the Cart of the User
- d. OUTPUT: Edit Food to the Database and Cart, and redirects to cart.

## **10. REMOVE CART FOOD**

- a. PRE-CONDITION:- Let the initial Database is given as per the above.
- b. INPUT: On Clicking Delete Button
- c. ACTION: DELETE Food in the Cart of the User
- d. OUTPUT: DELETE Food to the Database and Cart, and redirects to Cart.