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:
2. It has two links Login and Register which will route to the Login and Signup page.
3. Register Page:
4. It is the common page for all the users(Customer, NGO, Restaurants)
5. It shows a form with fields and options for roles to be filled
6. It has the button Signup which signup the user and put it in the database
7. If the email or username is already signed up, then it will show the error message
8. If everything is fine user can sign up through the sign up button.
9. Login Page:
10. It has two fields
11. Username
12. Password
13. 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
14. Example:
15. Satisfied Credential

`1. Input

1. Username: User\_ngo
2. Password: Type1

2. Output: Alert(“Logged in”)

ii. Unsatisfied Credential

`1. Input

1. Username: @User\_ngo
2. Password: Type1

2. Output: Alert(“invalid username/password”)

iii. Unsatisfied Credential

`1. Input

1. Username: User\_ngo
2. Password: @Type1

2. Output: Alert(“invalid username/password”)

1. Logout
2. After logging in user can see the logout link on the top of navbar
3. Navbar
4. If the user is not logged in navbar will have the link of signup and login
5. If the user is logged in user can check profiles, Restaurant, Order history, donation and cart.
6. Profiles
   1. It will show Name, Location and Role of the User with **Profile Image if provided during registration.**
   2. It will also show a map with the location of the user marked on it.
   3. Example:
      1. 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:
   1. It shows all the available restaurant under the distance 10km
   2. View food button will be on the every card of restaurant which will take u too the menu of the restaurant.
8. Menu:
   1. In menu, Restaurant owner can see all the food registered by him
   2. There is the button for adding food
   3. For every food u can update the price or amount of the food
   4. Delete button will remove the food from the menu.
9. Add Food
   1. It renders a form with some required fields.
   2. If the required fields(like the columns in the database shown above) are not filled then it will show an appropriate message.
   3. If all required fields are filled then it will show positive feedback.
   4. If Input for Price or Number of plates are not Numbers then it will show an appropriate message.
   5. If Input for Price or Number of Plates are zero or negative then it will show an appropriate message.
      1. If the owner tries to add an item whose quantity is -4, then he gets an error.
   6. It has a button to Submit the details of Food item.
   7. After Submitting, It will redirect to the menu of the restaurant with the message “Food item is Added to the Menu”.
   8. Response can be verified in the Menu of the Restaurant
10. Edit food
    1. Edit food button is available in the menu section for the restaurant
    2. Restaurant user can update the food price, no. of available plate or description of the food.
    3. If Input for Price or Number of plates are not numbers then it will show an appropriate message
    4. Edit food button will update all the fields for the food.
11. Remove Food
    1. Remove food button is available in the menu for the restaurant
    2. It removes the food from the menu
    3. Only restaurant owner can remove their food.
12. Add to Cart
    1. Add to Cart button is available in the view food section for the customer and NGO
    2. User can add food in their cart using this button
    3. If the no. of plates is 0 the food will not be added in the cart.
13. Cart
    1. Cart link is present in navbar for Customer/NGO
    2. It will redirect to the cart page
    3. The food added by user from the menu will be viewed here
    4. User can remove or update the no. of plates of the food here and save the changes using save changes button.
    5. **There is also a pay offline button which will**
14. Remove Cart food
    1. Remove button is available in the cart
    2. It removes the added food from the cart
    3. Result of removed food can be seen as a change in total amount.
15. Save Changes
    1. It updates the no. of plates in the cart
    2. User can change the required plate no. of plates and update using save change button.
16. Admin
    1. Admin can check all the registered user(customer, NGO and Restaurant)
    2. 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
2. Input for register will be given through user client(frontend)
3. ACTION: Adds a new user to the database who can logged in using login
4. OUTPUT: Add a new user
5. 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 |

1. Restaurant
   1. PRE-CONDITION:- Let the initial Database is given as per the above.
   2. INPUT: Click On the Button Restaurants
   3. ACTION: GET All Restaurants as per the User Location
   4. OUTPUT: Send The Data of All Restaurants
2. Menu
   1. PRE-CONDITION:- Let the initial Database is given as per the above.
   2. INPUT: Click On the Button View Menu of particular Restaurants
   3. ACTION: GET All Foods in the Menu of the Restaurants
   4. OUTPUT: Send The Data of All Foods in the Menu
3. Add Food
   1. PRE-CONDITION:- Let the initial Database is given as per above
   2. 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:

* + 1. Name: Dora Cake
    2. Price: 100/-
    3. No. of plates: 15
  1. ACTION: Add food in the menu of restaurant
  2. OUTPUT: Add food to the database of the menu and directs to the menu.
  3. DATABASE:

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

1. Edit Food
   1. PRE-CONDITION: The good going to be edited should be in database.
   2. INPUT: Edit food button is available in the my menu section for restaurant.
   3. EVENT: After giving an event, in response a form will appear and after satisfying all fields food will be updated.

Example:

1. Name: Rassgulla
2. Price: 25/-
3. Count:35
   1. ACTION: The food will be updated with new values.
   2. OUTPUT: Update the food in the database.
   3. Database:

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

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

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

1. ADD TO CART
   1. PRE-CONDITION:- Let the initial Database is given as per the above.
   2. INPUT:Number Of Plates
   3. ACTION: Add Food to the Cart of the User
   4. OUTPUT: Add Food to the Database and Cart, and redirects to Menu.
2. CART
   1. PRE-CONDITION:- Let the initial Database is given as per the above.
   2. INPUT: Click On the Button Cart
   3. ACTION: GET All Foods in the Cart
   4. OUTPUT: Send The Data of All Food in the Cart
3. EDIT CART FOOD
   1. PRE-CONDITION:- Let the initial Database is given as per the above.
   2. INPUT: Number Of Plates
   3. ACTION: Edit Food to the Cart of the User
   4. OUTPUT: Edit Food to the Database and Cart, and redirects to cart.
4. REMOVE CART FOOD
   1. PRE-CONDITION:- Let the initial Database is given as per the above.
   2. INPUT: On Clicking Delete Button
   3. ACTION: DELETE Food in the Cartof the USer
   4. OUTPUT: DELETE Food to the Database and Cart, and redirects to Cart.