**Category**

When we talk about food, food is always divided into multiple categories.

Ex:- Breakfast, lunch, dinner snacks, brunch etc.

So this feature will allow users to add a category for the restaurant.

**Create category**

1. User/Restaurant should be able to create a category from the menu page.
2. This will be an independent DB table with below given column

**Column name**

id

name

createdOn

lastUpdatedOn

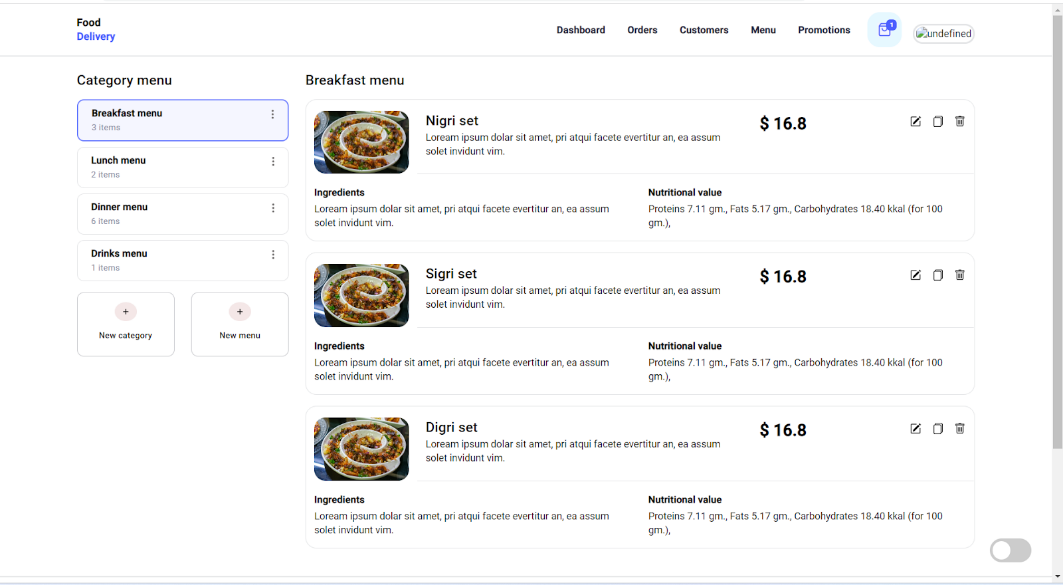
**Api details**

Path:

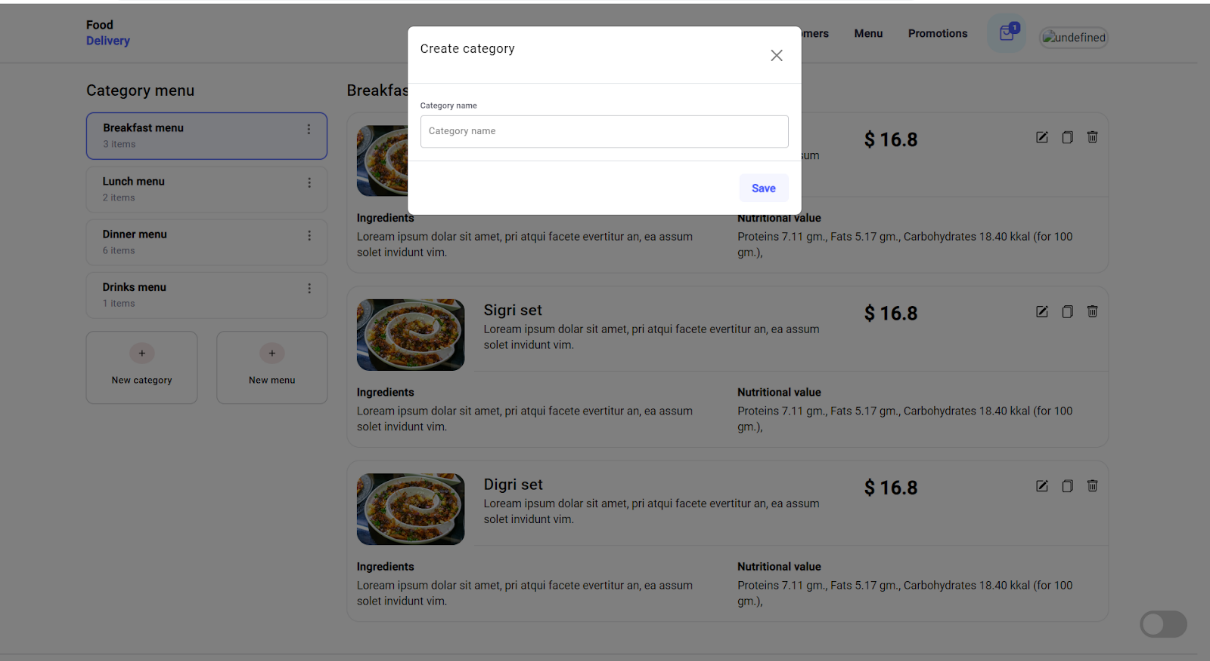
Method: post

Request Body: { name: string }

1. This option should be available in the menu section of the restaurant panel.
2. FE path will be **https://<domain name>/restaurant/menu**
3. Below is the design of the menu page of the restaurant panel.

****

1. When the user clicks on the **New category** card in the above design it should open a modal with a single input field **Name.**



1. Name should be required input field without any other validation.
2. There should be a button with a save label.
3. Button should be disabled until the name is given.
4. Clicking on the save button should hit the save api.
5. There should be a close **(X)** icon in the top right corner of the modal. Clicking on the **(X)** icon should close the modal.
6. When the user clicks outside the modal then also the modal should be closed.

**View category**

1. User should be able to view all the categories that he had created.
2. Users will get the list of categories.

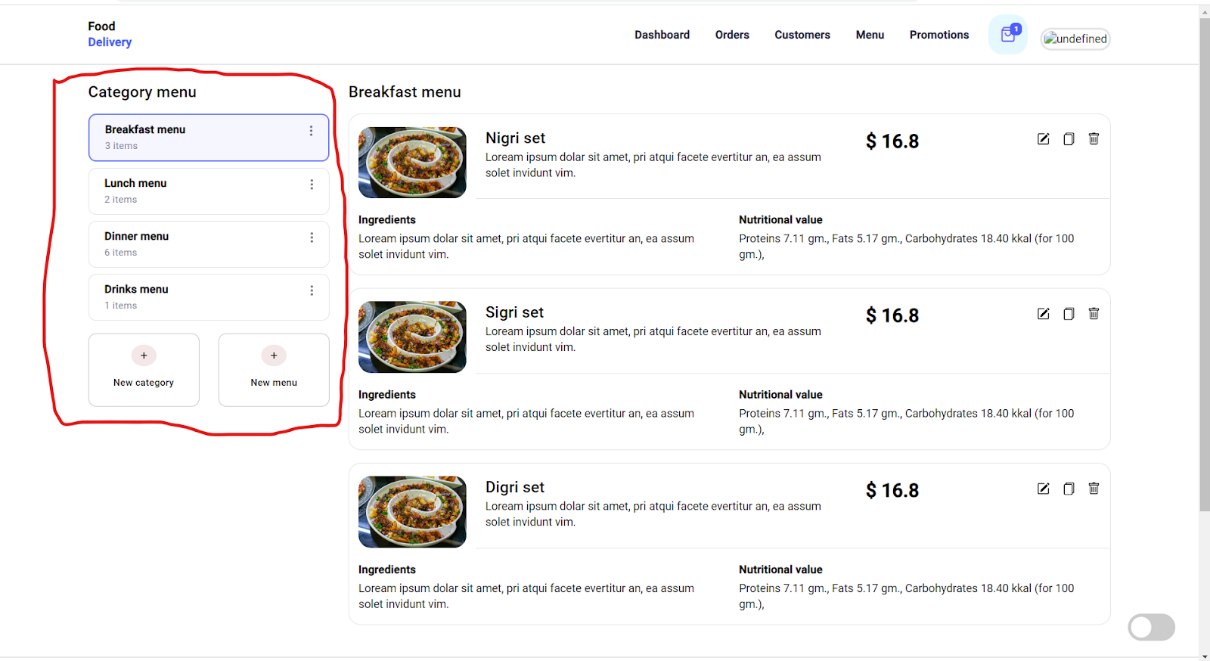
**Api details**

Path:

Method: get

Response Body: { data: [ { name: string } ], status: boolean }

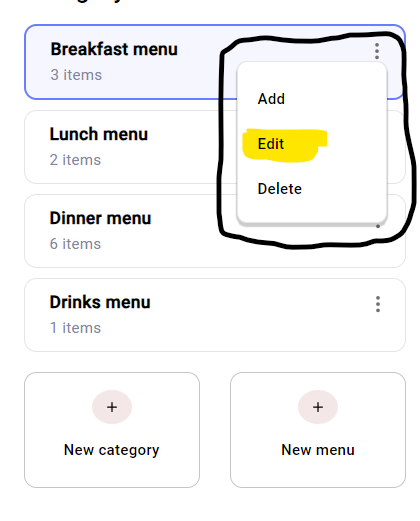
1. This list will be displayed in the menu section of the restaurant panel.
2. FE path will be **https://<domain name>/restaurant/menu**
3. Design to display categories in the menu section.



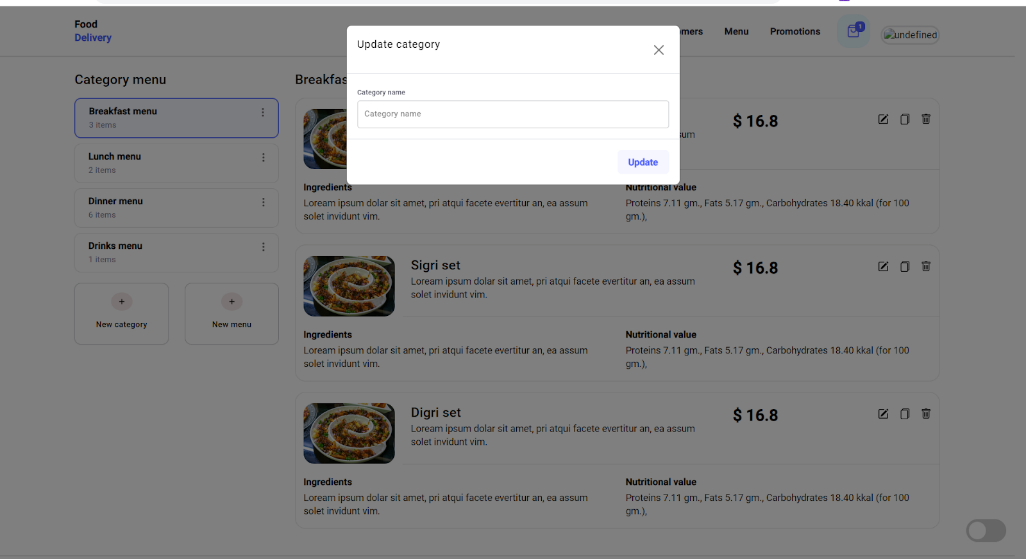
1. Below details will be displayed in the card
   1. Name - Name of the category.
   2. Items - Total menu items added in the category.
2. These category cards are selectable.
3. when a user clicks on the card
   1. Background colour and the border of the card will be changed.
   2. In the right section of the screen, the user will be able to see the menu items of the selected category. This feature will be implemented in the **menu** feature.
4. There will be a kebab/menu icon in the top right corner of the card.
5. Clicking on the kebab/menu icon will open a dropdown with below options.
   1. Edit ( this feature will be covered in **Update** category feature )
   2. Delete ( this feature will be covered in **Delete** category feature )

**Updated category**

1. Users should be able to update the name of the category.
2. Users will be able to update categories on the menu section of the restaurant panel.
3. FE path will be **https://<domain name>/restaurant/menu**
4. Design



1. When the user clicks on the **Edit**, it should open a modal with the prefilled current name of the category.
2. Design of the modal given below.



1. Update button should be disabled until the value is changed from the previous value.
2. Clicking on the Update button should hit the updated api and it should update the category name. The modal should be closed once the user clicks on the **Update** button.

**Api details**

Path:

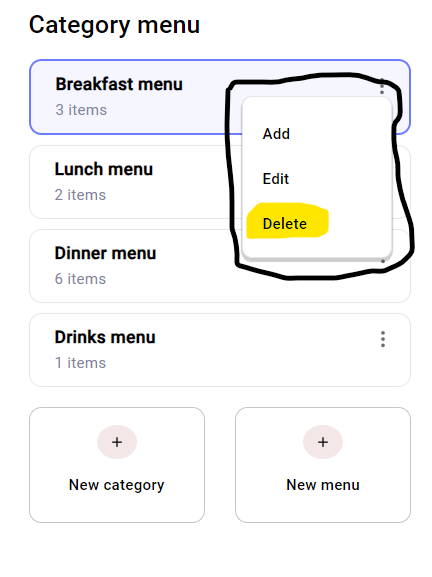
Method: put

Request Body: { name: string }

1. Clicking on **(X)** icon should close the modal.
2. When the user clicks outside the modal, the modal should be closed.
3. If the category name is updated, then it should update the categories in real time.
4. If the update api fails, an error message should be displayed in the snackbar on the top right corner.

**Delete category**

1. Users should be able to delete categories.
2. Users will be able to delete categories on the menu section of the restaurant panel.
3. FE path will be **https://<domain name>/restaurant/menu**
4. Design



1. When the user clicks on **Delete.**
   1. If there is no menu item assigned to the category, the modal should be open with the message **“Are you sure you want to delete <Category name>?”**

**Action**

1. Yes - Clicking on yes should hit delete api and delete the category.

**Api details**

Path:

Method: delete

1. No - Clicking on no should close the modal.
2. If the menu is assigned to the category, the modal should be open with the message **“Menu items are assigned to <Category name>. Deleting the category will automatically unassign all the menu items. This action can not be reversed. Are you sure you want to delete it ?”**

**Action**

1. Yes - Clicking on yes should hit delete api. It should delete the category and unassign all the menu items assigned to the category.

**Api details**

Path:

Method: delete

1. No - Clicking on no should close the modal.
2. If the **delete** api fails, the user should see the error message in the top right corner in the snackbar.

**Menu**

**Create menu**

1. User/Restaurant should be able to create a menu item from the menu page.
2. This will be an independent DB table with below given column

**Column name**

id

img

name

price

description

ingredients

nutritionalValue

createdOn

lastUpdatedOn

**Api details**

**Path**:

**Method**: post

**Request Body**: {

    img: string,

    name: string,

    price: string,

    description: string,

    ingredients: string,

    nutritionalValue: string,

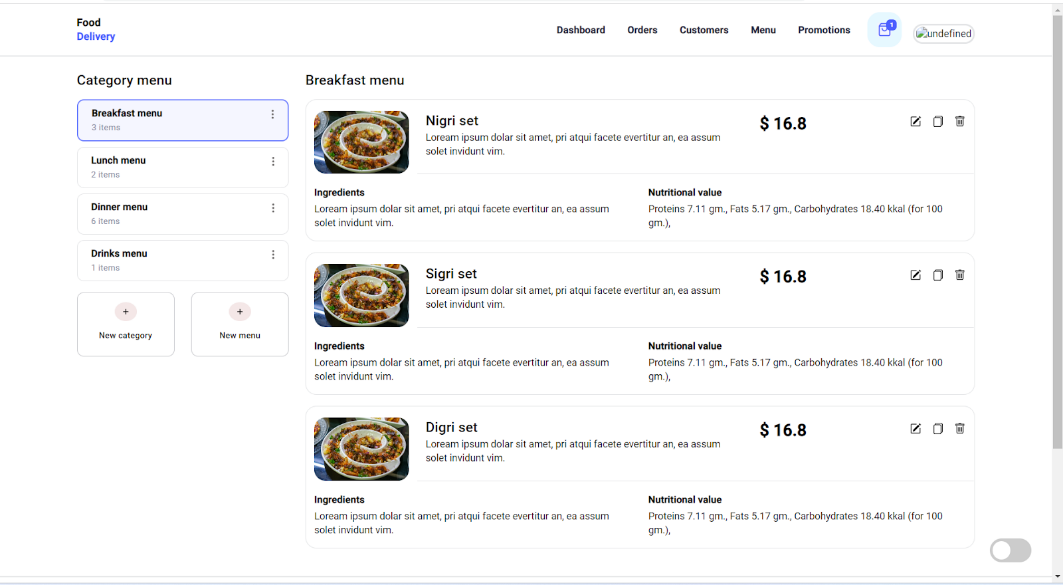
    taggedCategories: [number, number],

    createdOn: string,

    lastUpdatedOn: string

}

1. This option should be available in the menu section of the restaurant panel.
2. FE path will be **https://<domain name>/restaurant/menu**
3. Below is the design of the menu page of the restaurant panel.

****

1. When the user clicks on the **New menu** card in the above design it should open a modal with the following input fields

**img**

**name**

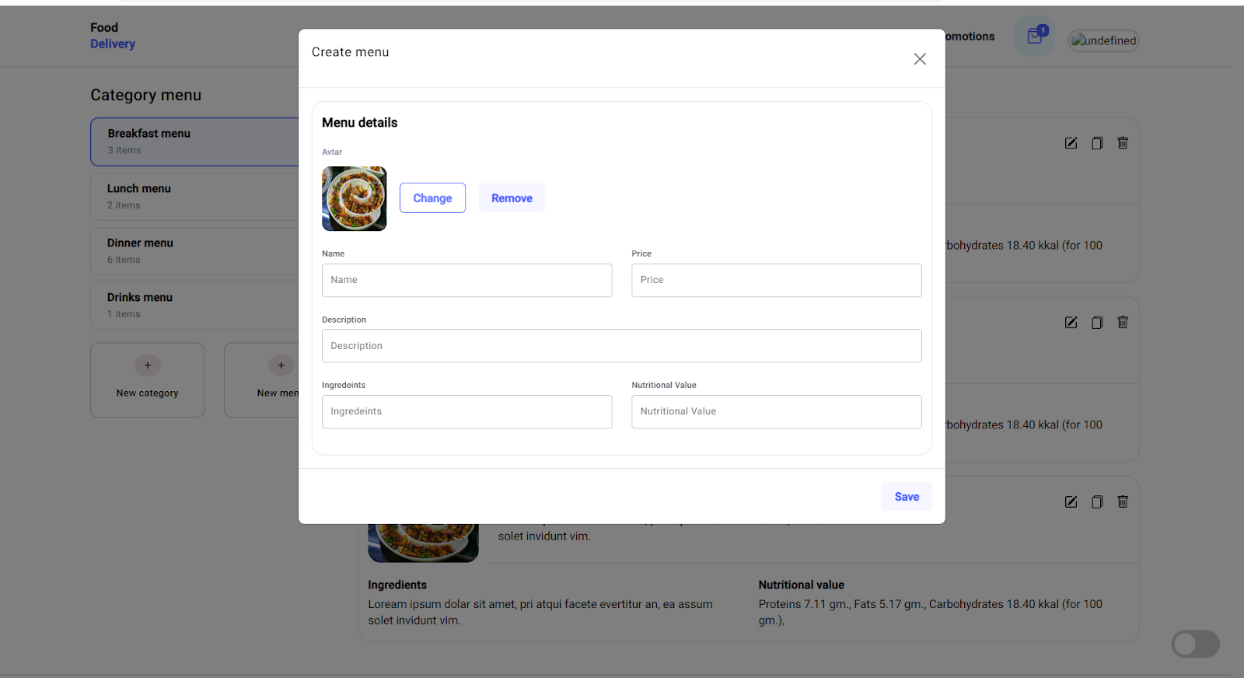
**price**

**description**

**ingredients**

**nutritionalValue**

**tagged**



1. Img should be a required input field without any other validation. If the user has not selected any image then there should be a default avatar image.
2. There are two buttons for image, named as **Change** and **Remove.**
   1. **Change** - Clicking on changes should allow the user to select a new image from the system.
   2. **Remove** - Clicking on remove will set the image as default avatar image.
3. Name, price, description, ingredients and nutritional value should be required in the input field without any other validation.
4. There should be a button with a save label.
5. Button should be disabled until all the validation passes.
6. Clicking on the save button should hit the save api.
7. There should be a close **(X)** icon in the top right corner of the modal. Clicking on the **(X)** icon should close the modal.
8. When the user clicks outside the modal then also the modal should be closed.

**View menu**

1. User should be able to view all the categories that he had created.
2. Users will get the list of categories.

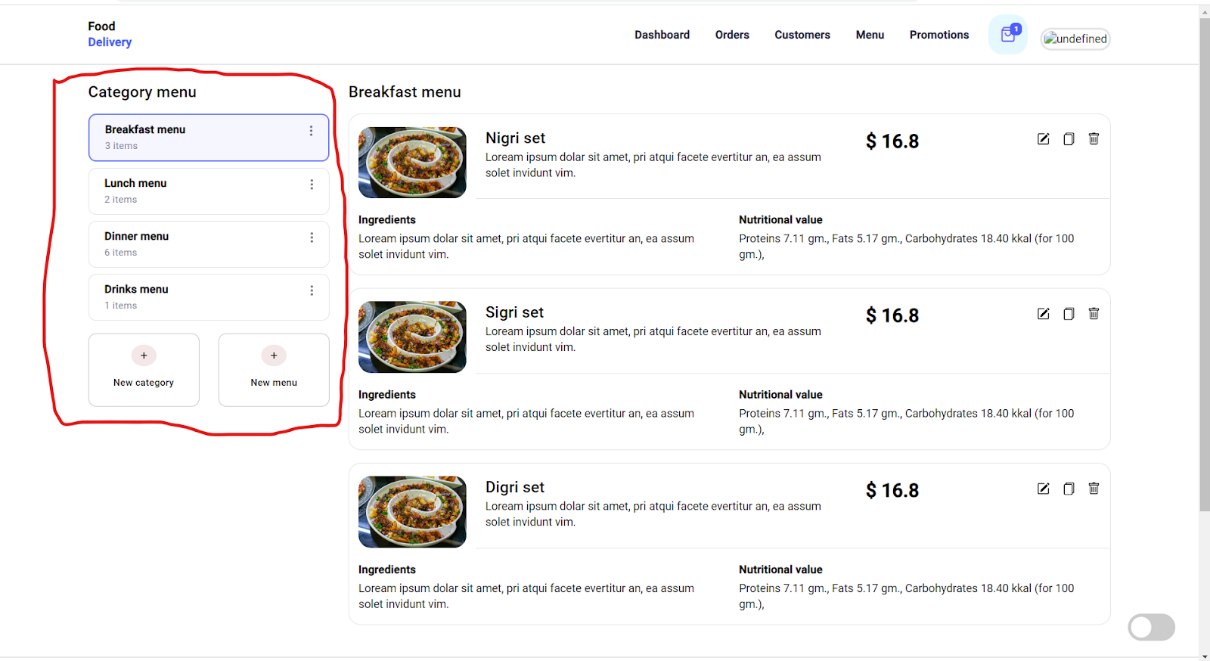
**Api details**

Path:

Method: get

Response Body: { data: [ { name: string } ], status: boolean }

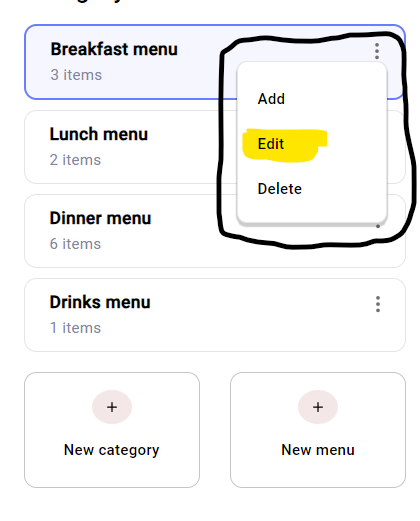
1. This list will be displayed in the menu section of the restaurant panel.
2. FE path will be **https://<domain name>/restaurant/menu**
3. Design to display categories in the menu section.



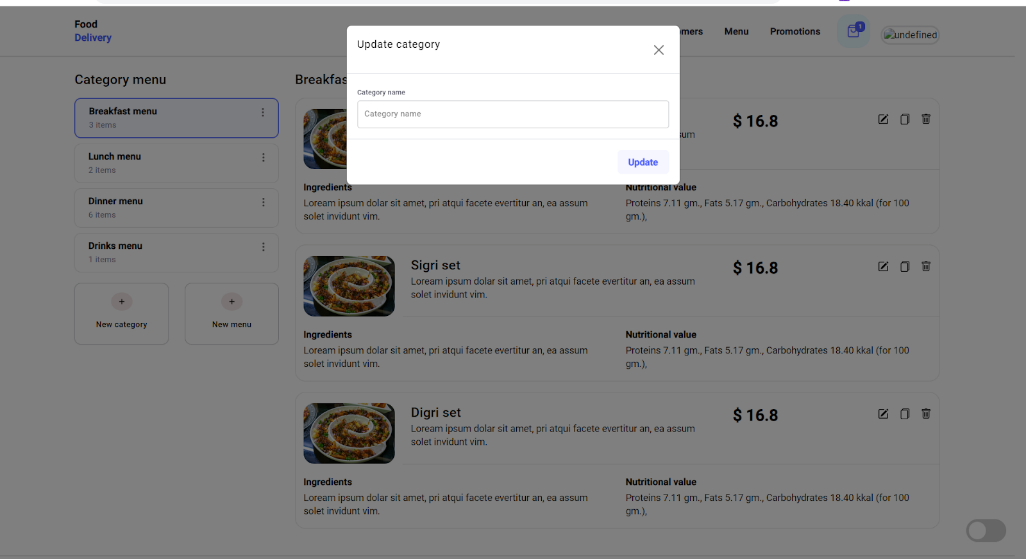
1. Below details will be displayed in the card
   1. Name - Name of the category.
   2. Items - Total menu items added in the category.
2. These category cards are selectable.
3. when a user clicks on the card
   1. Background colour and the border of the card will be changed.
   2. In the right section of the screen, the user will be able to see the menu items of the selected category. This feature will be implemented in the **menu** feature.
4. There will be a kebab/menu icon in the top right corner of the card.
5. Clicking on the kebab/menu icon will open a dropdown with below options.
   1. Edit ( this feature will be covered in **Update** category feature )
   2. Delete ( this feature will be covered in **Delete** category feature )

**Updated menu**

1. Users should be able to update the name of the category.
2. Users will be able to update categories on the menu section of the restaurant panel.
3. FE path will be **https://<domain name>/restaurant/menu**
4. Design



1. When the user clicks on the **Edit**, it should open a modal with the prefilled current name of the category.
2. Design of the modal given below.



1. Update button should be disabled until the value is changed from the previous value.
2. Clicking on the Update button should hit the updated api and it should update the category name. The modal should be closed once the user clicks on the **Update** button.

**Api details**

Path:

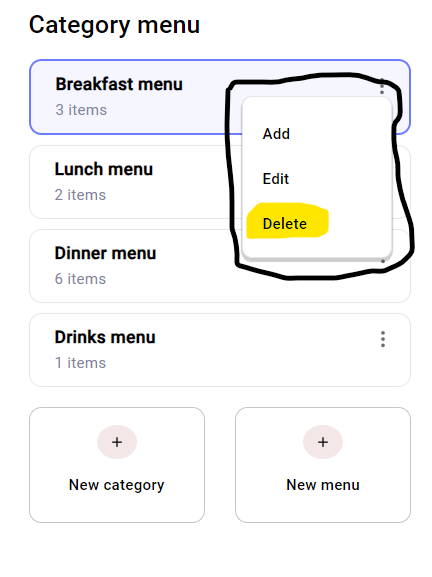
Method: put

Request Body: { name: string }

1. Clicking on **(X)** icon should close the modal.
2. When the user clicks outside the modal, the modal should be closed.
3. If the category name is updated, then it should update the categories in real time.
4. If the update api fails, an error message should be displayed in the snackbar on the top right corner.

**Delete menu**

1. Users should be able to delete categories.
2. Users will be able to delete categories on the menu section of the restaurant panel.
3. FE path will be **https://<domain name>/restaurant/menu**
4. Design



1. When the user clicks on **Delete.**
   1. If there is no menu item assigned to the category, the modal should be open with the message **“Are you sure you want to delete <Category name>?”**

**Action**

1. Yes - Clicking on yes should hit delete api and delete the category.

**Api details**

Path:

Method: delete

1. No - Clicking on no should close the modal.
2. If the menu is assigned to the category, the modal should be open with the message **“Menu items are assigned to <Category name>. Deleting the category will automatically unassign all the menu items. This action can not be reversed. Are you sure you want to delete it ?”**

**Action**

1. Yes - Clicking on yes should hit delete api. It should delete the category and unassign all the menu items assigned to the category.

**Api details**

Path:

Method: delete

1. No - Clicking on no should close the modal.
2. If the **delete** api fails, the user should see the error message in the top right corner in the snackbar.