# Restaurant menu documentation

When the application starts, it loads menu items from the text file. After that it asks user to enter their choice if they want to go to cashier interface or admin interface.

For the admin interface, an admin can add new menu items, delete menu items, modify details of an existing item and display all menu items.

Cashier takes orders from customers. System allows him to add items to an order. When he finishes, system displays order total.

## MENU.H Header File

This class manages restaurant menu items.

### Struct item.

This is a structure that saves item details. Id, price, name.

The next pointer points to the next item in the linked list.

### menuItems class.

It implements a linked list. There is a head pointer that points to first item of the list and the tail item points to last item of the list.

#### void addMenuItem(char itemName[50], float price)

This functions takes 2 parameters as input. Name and price. And adds a new restaurant menu item. It then appends the new menu item to the text file named menu.txt.

#### Void displayMenuItems()

This function displays all menu items with their prices, available at the restaurant.

#### bool deleteMenuItem(int id)

Takes item id as input parameter. The function first searches for item if it exists. If item is found, it deletes the item from the menu and updates the menu.

#### bool editMenuItem(int id,char itemName[50],float price)

this function takes 3 parameters as input, item id, item name and price. It searches for the item using item id if it exists or not. If it exists, user is able to modify item name and price. The details are updated and the updated menu items are written to the text file menu.txt.

#### char\* getItemName(int id)

this function takes item id as input parameter, searches for item. If item exists, return item name.

#### float getItemPrice(int id)

this function takes item id as input parameter, searches for item. If item exists, return item price.

#### void writeDataToTextFile()

this function reads all items from the linked list one by one and writes them to the data file menu.txt.

#### void loadDataFromTextFile()

this functions reads the text file line by line. Tokenizes each line on the basis of “/”. And extracts item id, name and price. And add these items to the linked list.

#### bool searchMenuItem(int id)

this function takes item id as input parameter. Searches for items in the list. If item exists, returns true. False, otherwise.

## ORDERS.H

This function manages details for orders.

#### struct orderItem

#### this structure stores details for a line item. It saves item name, price and quantity.

#### void addItem(int id, char itemName[50], float price, int qty)

This function adds new item in an order. The function takes 3 input parameters which are id, item name, and price. The items are then stored onto the linked list.

#### void displayMenuItems()

this function display order details. It shows item name, price, quantity, and line total. At the end, this functions displays total amount due.