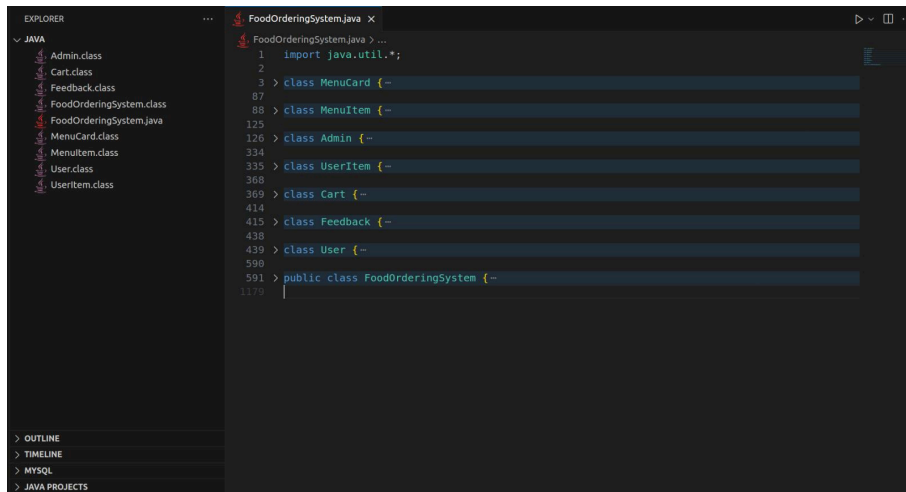


DO's and DON'Ts of CODING STYLE :

DO's :

1. Usage of User-Defined Datatypes :

Throughout the coding phase around 9 classes(including main) and various user defined functions were used.



2. Hide Data Structures behind access functions :

The data of each class has been hidden using the private access modifier. Since the member functions in the class can only access them . Access functions were defined in each class.

```
class MenuItem {
    private String name;
    private double price;
    private int quantity;
    private String adminName;

    MenuItem(String name, double price, int quantity, String adminName) {
        this.name = name;
    }
}
```

```
        this.price = price;
        this.quantity = quantity;
        this.adminName = adminName;
    }

    public String getName() {
        return this.name;
    }

    public double getPrice() {
        return this.price;
    }

    public int getQuantity() {
        return this.quantity;
    }

    public String getAdminName() {
        return this.adminName;
    }
}
```

3. User Manual and Software Documentation has been included with the zip file
4. Proper use of indentation has been deployed in the interface design of the Java code.

```

}
System.out.println();
System.out.println();
System.out.println("-----");
System.out.println("Menu: by " + name);
System.out.println("-----");
System.out.println("S/No\tName\tQuantity\tPrice\n");
int cnt = 1;
for (MenuItem menuItem : menuCard) {
    System.out.println(
        cnt + "\t" + menuItem.getName() + "\t\t" + menuItem.getQuantity() + "\t\t₹" +
        menuItem.getPrice());
    cnt++;
}
System.out.println("-----");
System.out.println("Total Items: " + menuCard.size());
System.out.println("-----");
System.out.println();
}

```

```

-----
User Choices:
-----
1. To view the menu
2. To order an item
3. To delete an item from cart
4. To update quantity
5. To view cart
6. To place an order
7. To provide feedback
8. To view order history
-----

Enter your choice:
1

-----
Menu: by admin1
-----
S/No    Name        Quantity    Price
-----
1.      bread        20          ₹15.0
-----
Total Items: 1
-----

```

DON'Ts:

1. There was not a single usage of the null Then and Then_IF statements.
2. Nesting was limited to a considerable extent in the code.

```

for (Cart order : orders) {
    if (order.getUsername().equals(name)) {
        orders.remove(order);
        System.out.println("Order removed successfully!");
        for (User user : users) {
            if (user.getUsername().equals(name)) {
                users.remove(user);
                break;
            }
        }
        return;
    }
}
System.out.println("ERROR : Order not found!");

```

3. An identifier was not used for multiple purposes.