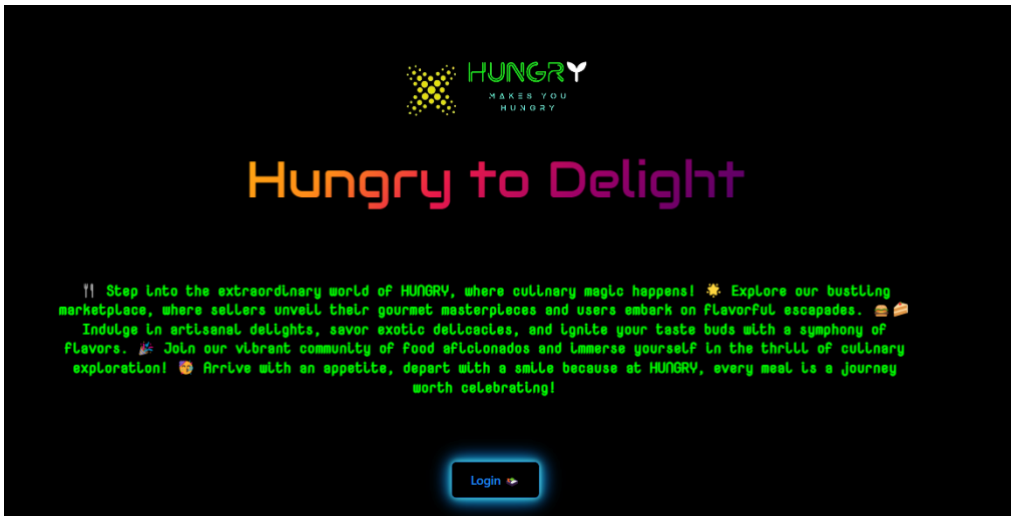
3 GOLDEN RULES

Rule 1. Place the User in Control

- Hide technical internals from the casual user

```
MYSQL_PASSWORD="enter your MySQL password here"
MYSQL_USERNAME="enter your MySQL username here"
STRIPE_PUBLIC_KEY='enter your stripe public key here'
STRIPE_PRIVATE_KEY='enter your stripe private key here'
SECRET="enter your secret for JWT token here"
```

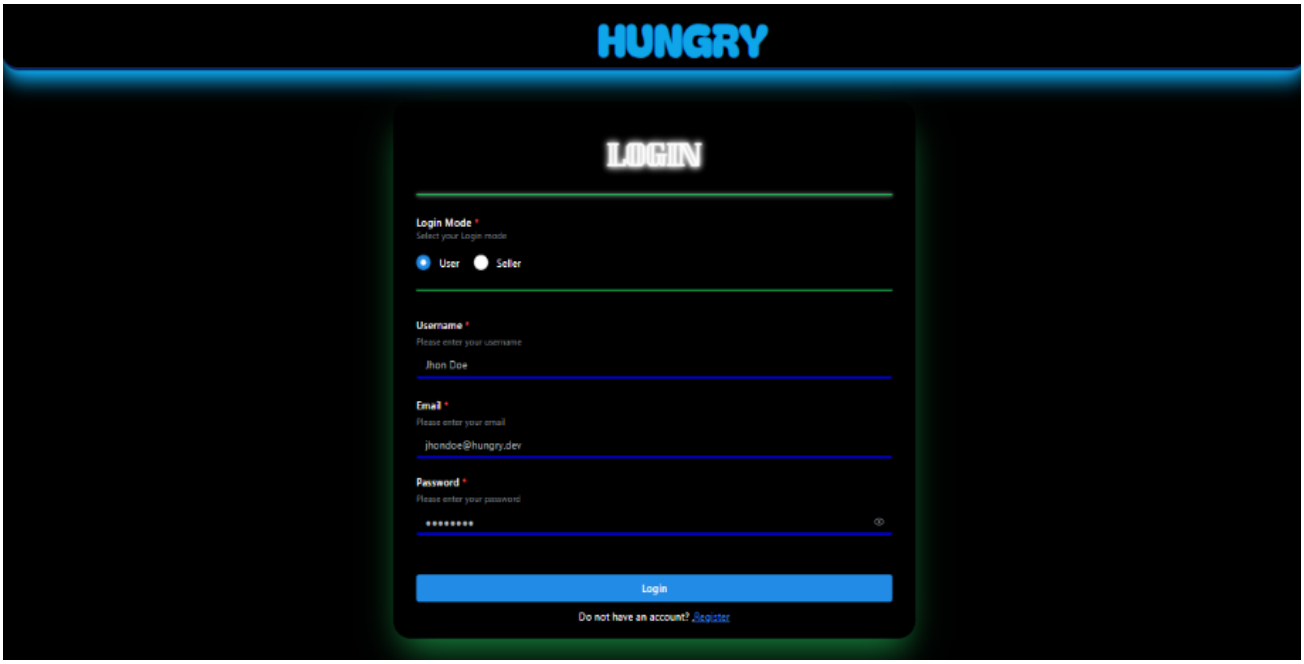
- Design for direct interaction with objects that appear on the screen



Rule 2. Reduce the User's Memory Load

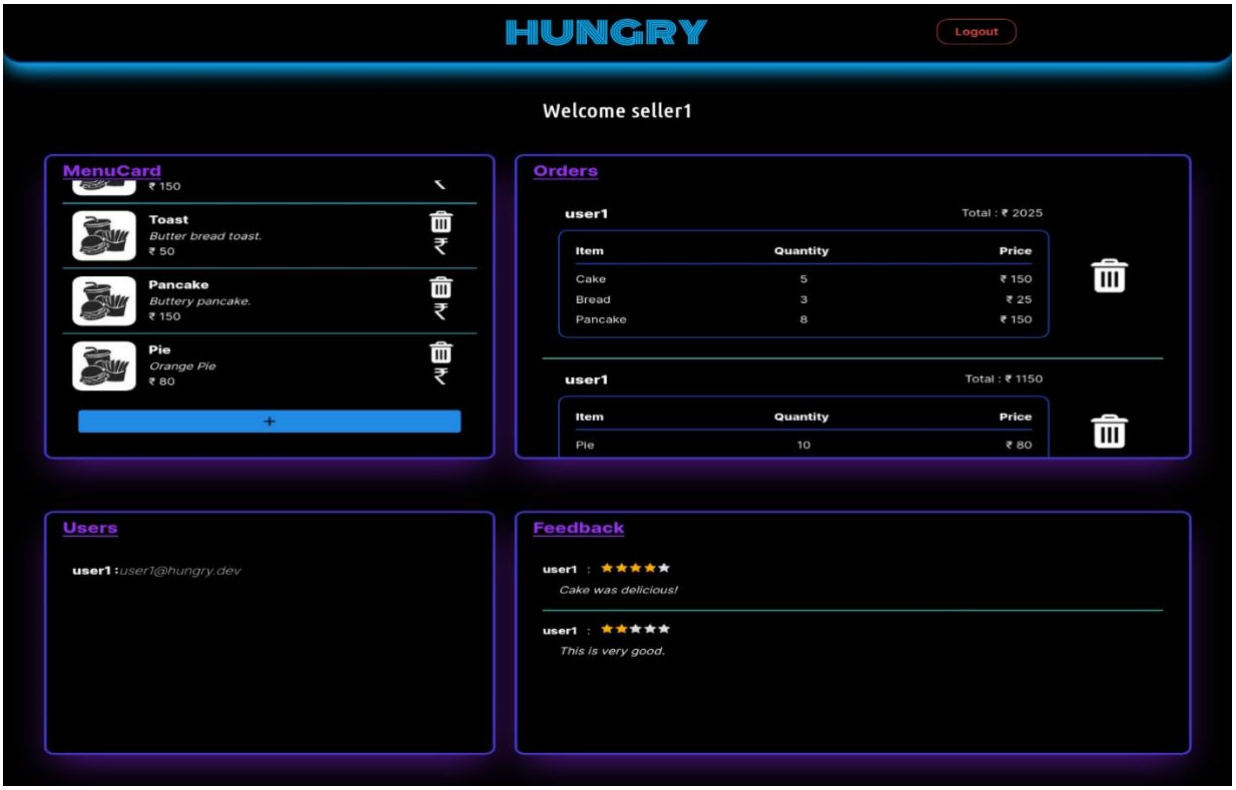
- Establish meaningful defaults

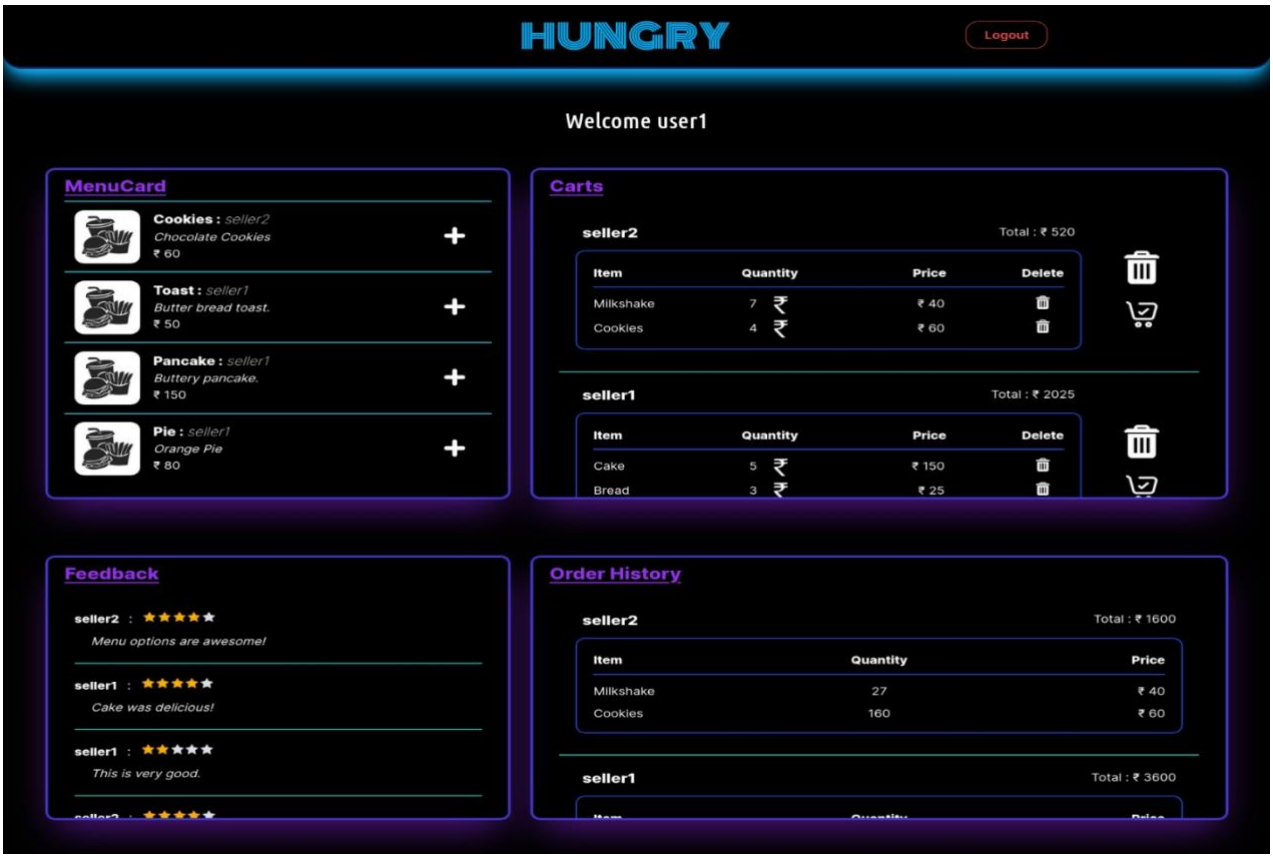
- The visual layout of the interface should be based on a real world metaphor



Rule 3. Make the Interface Consistent

- Maintain consistency across a family of applications




HUNGRY
Logout
Welcome user1
MenuCard
Cookies : seller2
Chocolate Cookies
₹ 60
Toast : seller1
Butter bread toast.
₹ 50
Pancake : seller1
Buttery pancake.
₹ 150
Pie : seller1
Orange Pie
₹ 80
Carts
seller2
Total : ₹ 520
Item
Quantity
Price
Delete
Milkshake
7
₹ 40
Cookies
4
₹ 60
seller1
Total : ₹ 2025
Item
Quantity
Price
Delete
Cake
5
₹ 150
Bread
3
₹ 25
Feedback
seller2 :
Menu options are awesome!
seller1 :
Cake was delicious!
seller1 :
This is very good.
Order History
seller2
Total : ₹ 1600
Item
Quantity
Price
Milkshake
27
₹ 40
Cookies
160
₹ 60
seller1
Total : ₹ 3600

BASIC DESIGN PRINCIPLE

- **EXIBIT UNIFORMITY AND INTEGRATION:**
    - The code possess uniformity throughout the process without any changes.

```
class Feedback{
    private String name;
    private String adminName;
    private String feedback;

    Feedback(String name,String aname, String feedback){
        this.name = name;
        this.adminName = aname;
        this.feedback = feedback;
    }

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

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

    public String getFeedback(){
        return feedback;
    }
}
```

- **ACCOMMODATE CHANGES:**
    - The code can accommodate the changes that required to done due to it's uniformity that has been done throughout the process.
    - We can incorporate a new function to that class if required and that can be replicated and the code could adopt changes.

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

    UserItem(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 String getAdminName(){
        return this.adminName;
    }

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

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

    public void setQuantity(int x){
        this.quantity = x;
    }
}
```

- **DEGRADE GENTLY:**
    - The system as the capability to handle the error's in all possible situation.

```
--------------------------------------------------
Welcome to HUNGRY!
--------------------------------------------------


Do you want to enter as admin or as user?
admin


If you are registered as admin enter signin else enter signup
sigin1
Invalid choice! Please enter again.

If you are registered as admin enter signin else enter signup
signup

--------------------------------------------------
SignUp:
--------------------------------------------------

Enter your username:
kishore

Enter your password:
12q1

--------------------------------------------------
```

```
Admin registered successfully!

Welcome kishore!


--------------------------------------------------
Admin Choices:
--------------------------------------------------
1. To view menu
2. To add a menu item
3. To delete a menu item
4. To update price
5. To update quantity
6. To view all users
7. To view all orders
8. To view all feedbacks
9. To remove the order
--------------------------------------------------

Enter your choice:
3

Enter the name of the menu item
pongal
ERROR : Menu card not created yet!

Do you want to continue as admin? (Y/N)
```

- **ASSESSED OR QUALITY:**
    - The design quality as given throughout the process for entire food app system with good quality.