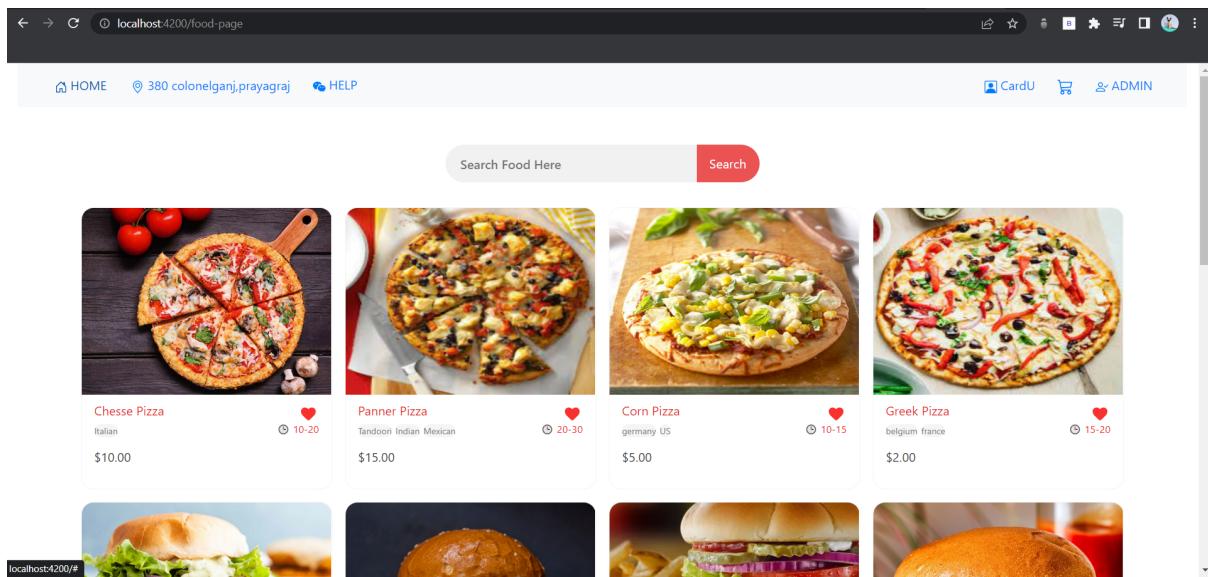


# FOODBOX



## DESCRIPTION

To Create a dynamic and responsive online food delivery web application for ordering food items of different cuisines from a restaurant.

## Background of the problem statement

Foodbox is a restaurant chain that delivers food items of different cuisines at affordable prices. It was established in 2014 in Bengaluru, India. It had been serving fine all these years, however, the business analysts noticed a decline in sales since 2016. They found out that the online ordering of food items with companies, such as Swiggy and Foodpanda were gaining more profit by eliminating middlemen from the equation. As a result, the team decided to hire a Full Stack developer to develop an online food delivery web application with a rich and user-friendly interface.

You are hired as the Full Stack Java developer and are asked to develop the web application. The management team has provided you with the requirements and their business model so that you can easily arrange different components of the application.

## User Portal:

It deals with the user activities. The end-user is be able to:

- Sign-in to the application to maintain a record of activities.
- Search for food items based on the search keyword.
- Add all the selected food items to a cart and customise the purchase at the end.
- Perform a seamless payment process.

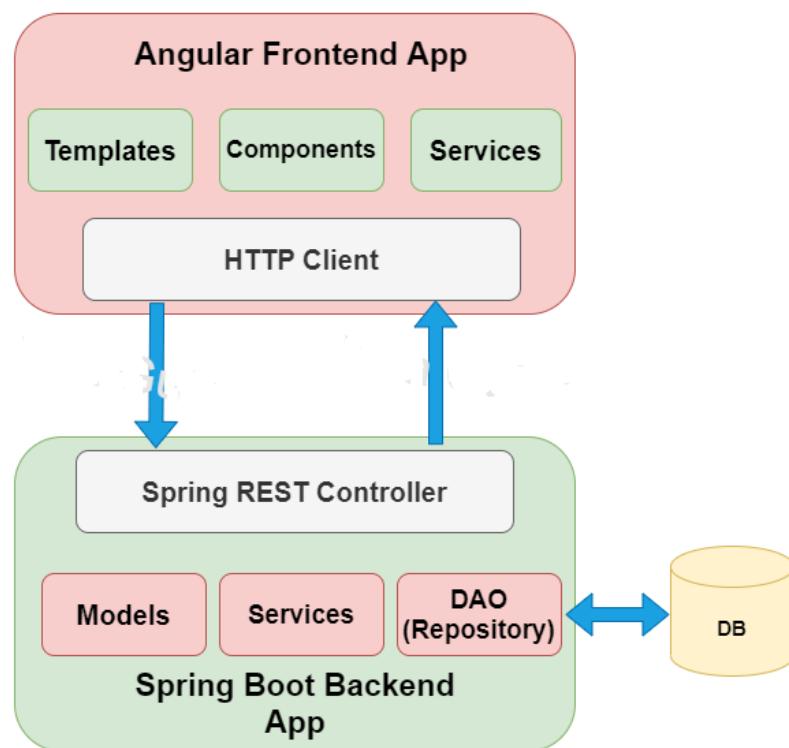
### **Admin Portal:**

The admin portal deals with all the backend data generation and the management of sensitive user detail information.

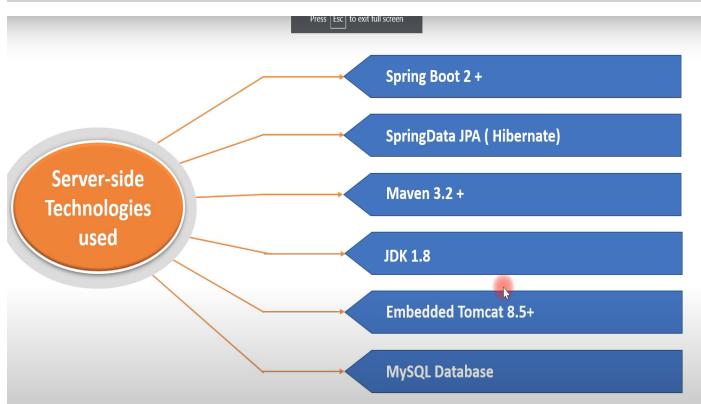
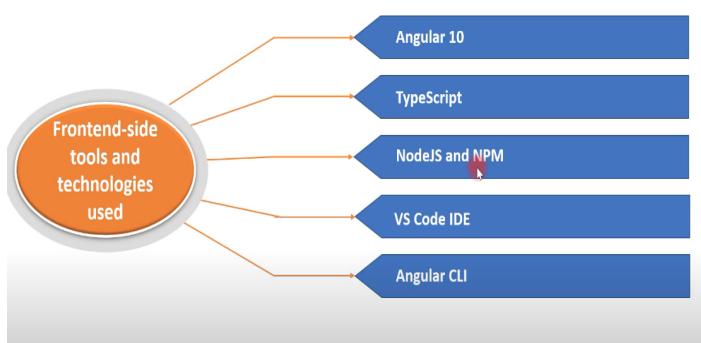
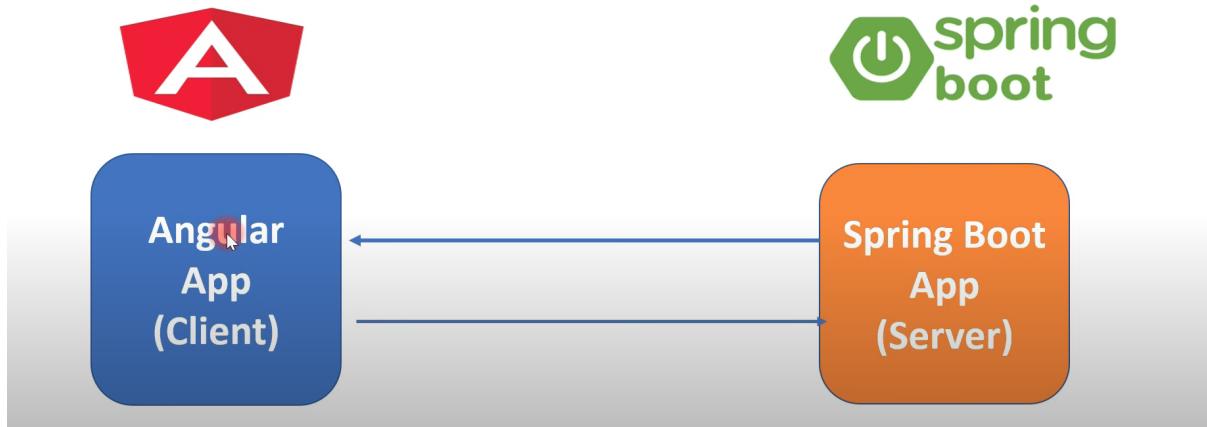
- Add or remove different user details to or from the application to keep sensitive user information private.
- Manage user details like name, email, phone number, address and other details.

---

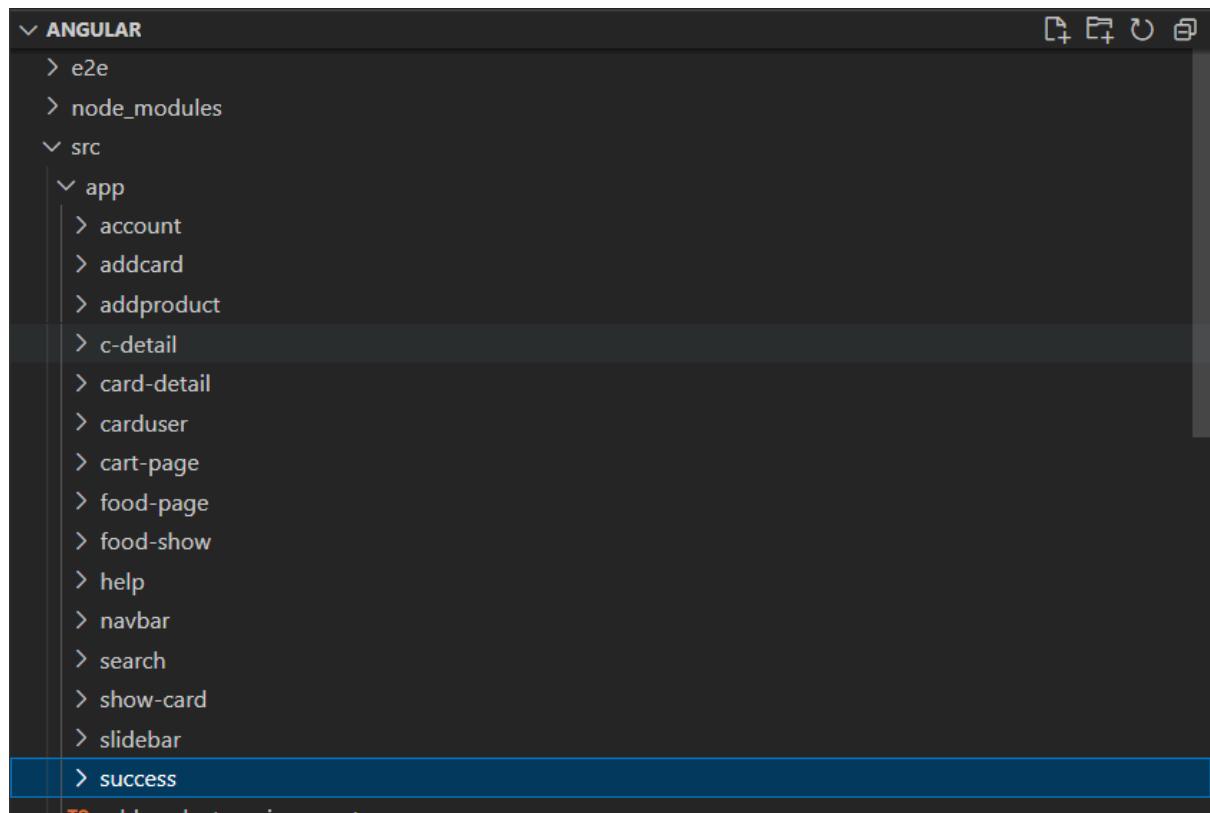
### **Workflow of data from frontend to backend and vice versa**



# Client server Architecture



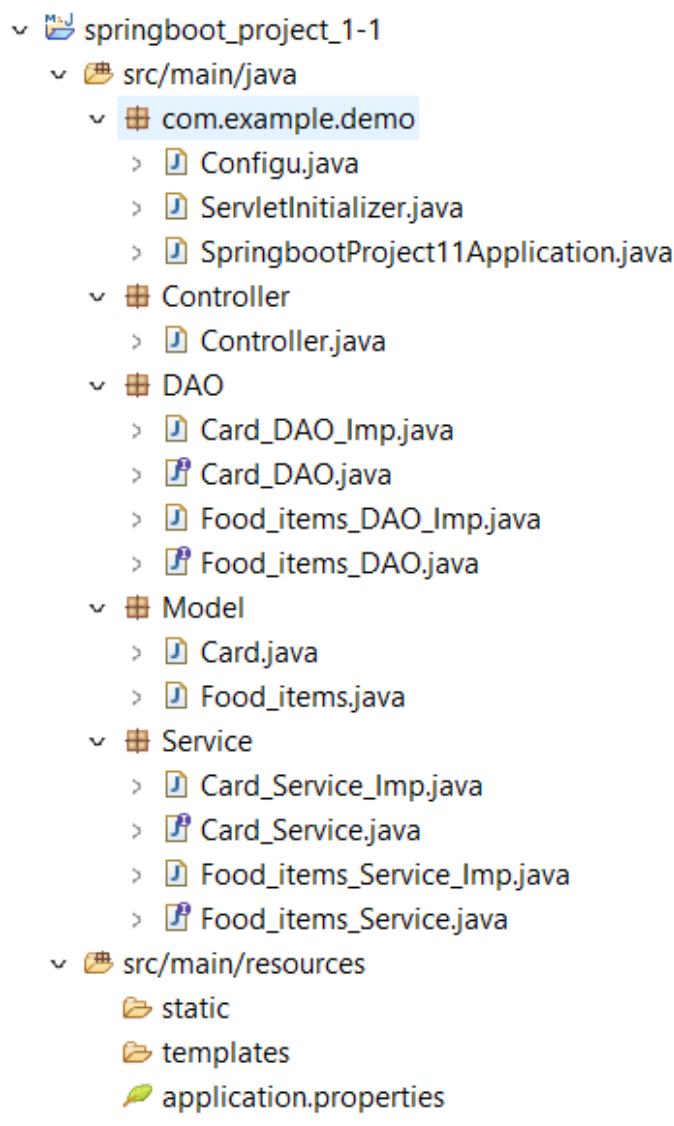
Folder view in Angular component.

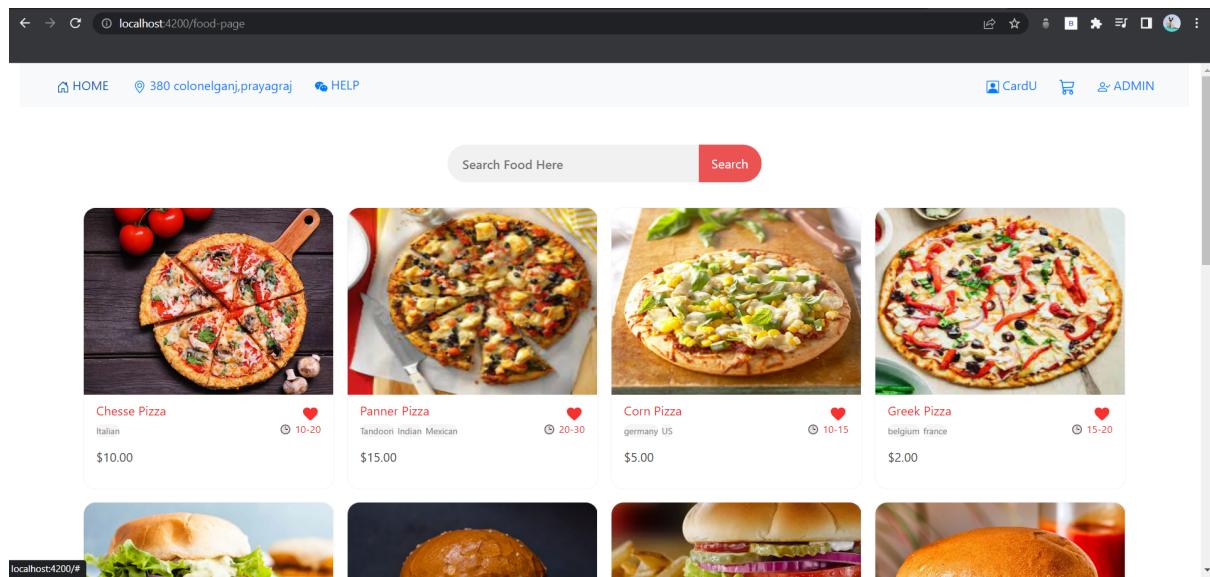


## service

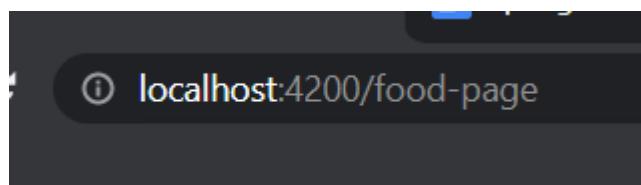
```
TS addproduct.service.spec.ts
TS addproduct.service.ts
TS addproduct.spec.ts
TS addproduct.ts
TS app-routing.module.ts
# app.component.css
▷ app.component.html
TS app.component.spec.ts
TS app.component.ts
TS app.module.ts
TS card.service.spec.ts
TS card.service.ts
TS card.spec.ts
TS card.ts
TS cart.service.spec.ts
TS cart.service.ts
TS Cart.ts
TS CartItem.ts
TS food.service.spec.ts
TS food.service.ts
TS Food.ts
TS searchfilter.pipe.spec.ts
TS searchfilter.pipe.ts
▽ assets
◆ .gitkeep
```

## Folder view in Spring Sts:



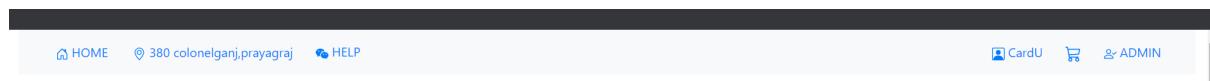


The center/ home component is the food-page component.



It contains the navigation bar on the top.

## NAVBAR WORKFLOW



Technologies used:

Bootstrap icons ,buttons and navbar templates.

Event binding from angular.

Routing to different pages by click of button.

Edit /addcard with /cart-page.

## FOODPAGE

### WORK FLOW OF SEARCH BAR.

Type “pizza” in the search box.

Search Food Here

Search

localhost:4200/search/pizza

HOME HELP CardU ADMIN

pizza

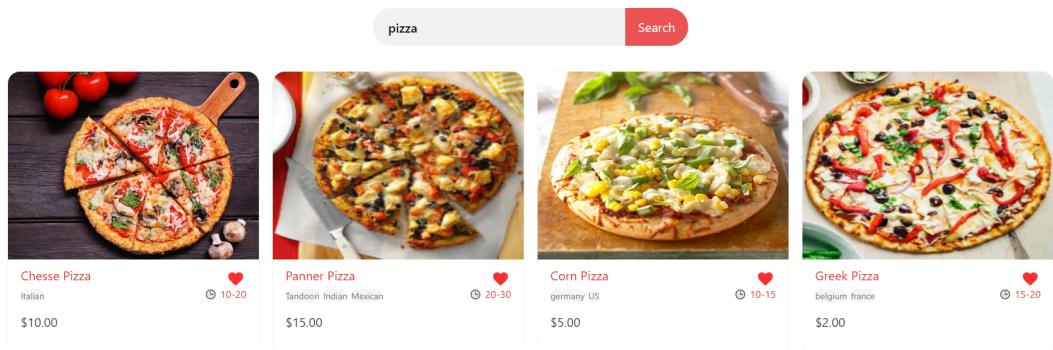
Chesse Pizza  
Italian \$10.00

Panner Pizza  
Tandoori Indian Mexican \$15.00

Corn Pizza  
germany US \$5.00

Greek Pizza  
belgium france \$2.00

Click the Search button:



## WORKFLOW OF ORDERING FOOD.

1. Click on the item to select.



[Chesse Pizza](#)

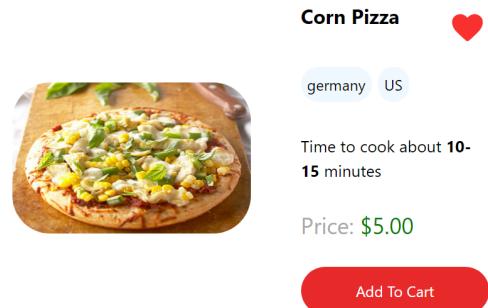
Italian



10-20

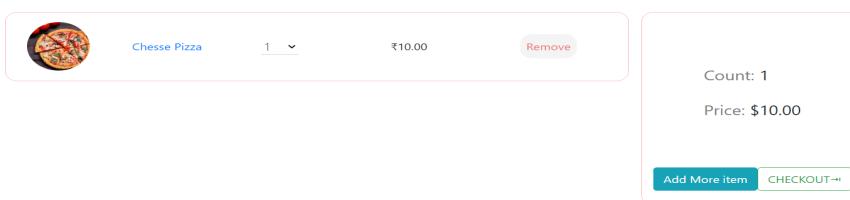
[\\$10.00](#)

You will be redirected to that item's page

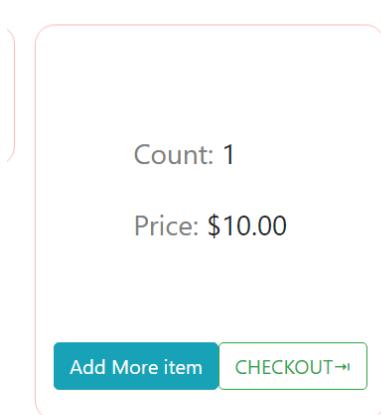


**Click on Add to Cart**

**You will be redirected to cart-page and the items will be added to cart.**



**Click on “Add More item” to add more items.**



**You will be redirected to the food-page where you can select more items.  
Click on the items you want to select.**

\$9.00



[Sponge Cake](#)

0

Japan

40-50

[\\$9.00](#)

You will be directed to the item's details page.

**Sponge Cake**



Japan

Time to cook about **40-50** minutes



Price: **\$9.00**

[Add To Cart](#)

Click on “Add to Cart” to add to the cart.

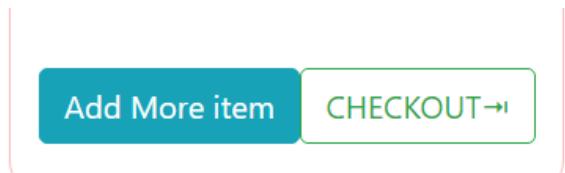
	Chesse Pizza	1 ▾	₹10.00	<button>Remove</button>
	Sponge Cake	1 ▾	₹9.00	<button>Remove</button>

Count: 2  
Price: \$19.00

[Add More item](#) [CHECKOUT→](#)

**The total quantity and total price is displayed before checkout.**

**Click on the checkout button to proceed for card payment.**



## PAYMENT

**Enter the user details and card details to pay:**

# ONLINE FOOD ORDER

## Payment

User Details:

User name



Hari

User Email



hari@gmail.com

User Phone



9113302599

User Address



Kormangala

User Date\_time

User Date\_time

13-09-2022 17:00

CARD Comapny

SBI

Add new card:

Cardholder's Name

Hari Sinha

Card Number

1234852878945612

Expire

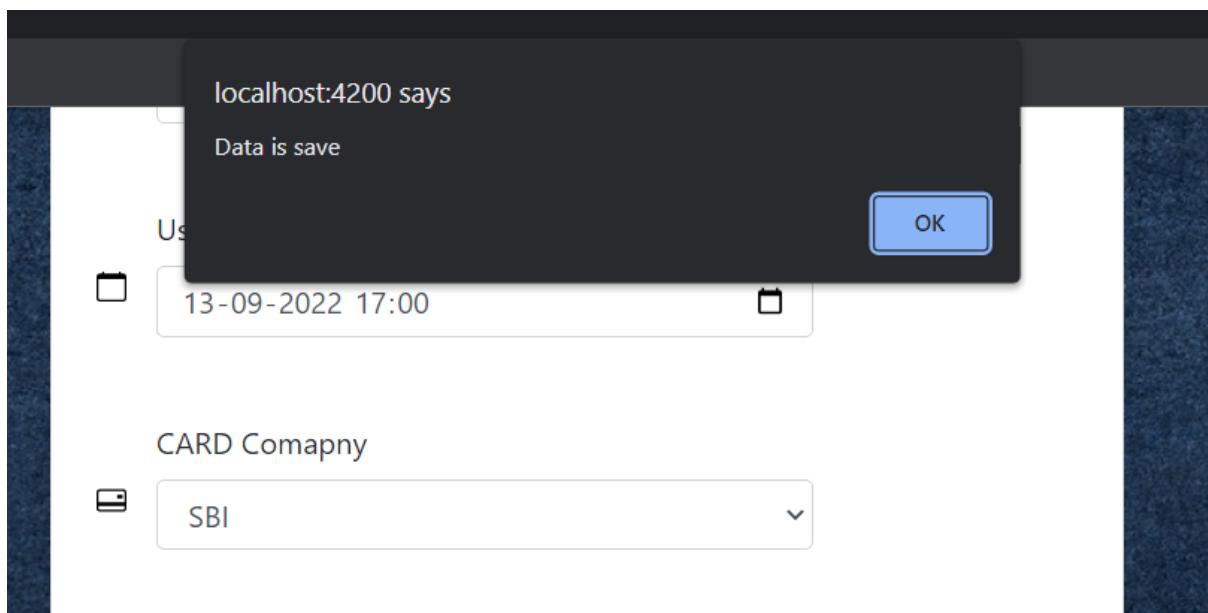
04/25

Cvv

...

**Add card**

**Click on “Add card” to save your card details and complete the order.**



**Data saved alert will be displayed and the details will be saved in the database .**

**Yay! Your order will be placed!**

**Yay! Your order has been placed!**

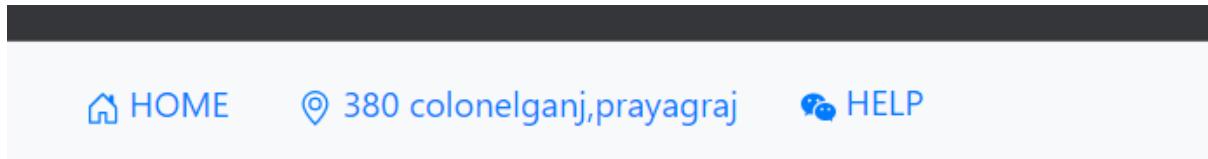
It will soon be out for delivery.



**HOME**

**Click on the HOME button to be redirected to the home page.**

**Click on the Help button on Navigation bar to contact the restaurant for any issues.**



## Having an issue with your order or payment?



Please enter your details below

Name:

Email:

Contact number:

I have an issue

---

**Submit**

**Users can access their card details using the following steps:**

**1.Click on the CardUser Icon.**

 CardU

**2.Enter the username and password as the phone number.**



Enter username  
User name

Enter password  
Phone number

Login



user1  
User name

.....  
Phone number

Login

localhost:4200 says  
congratulation , you can see your details now

OK



user1

User name

.....

Phone number

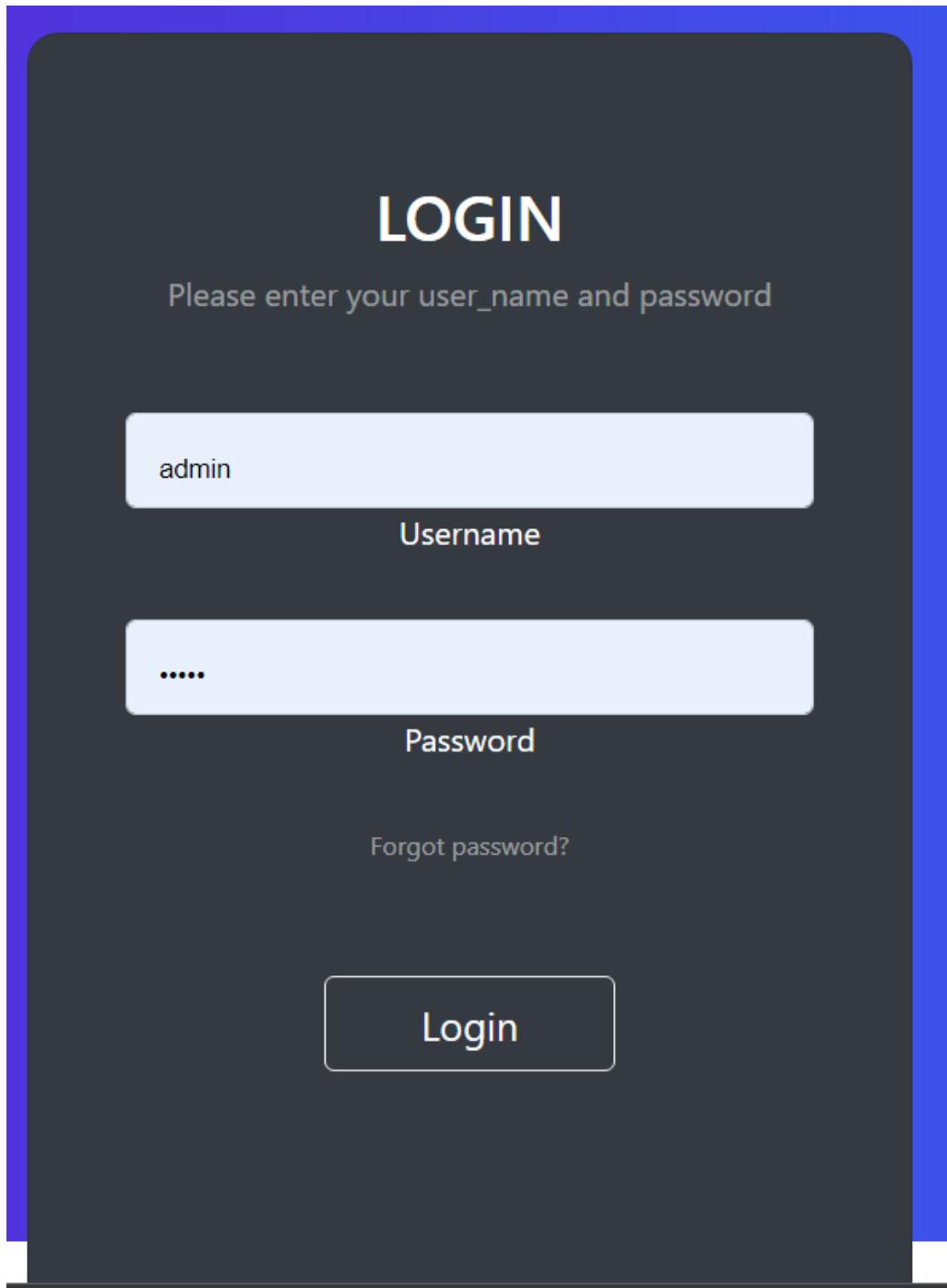
Login

Card user	Card Email	Card Phone	Card Address	Card Date_Time	Card Branch	Card User Name	Card new_Number	Card Expire	Card Cvv	Action
user1	user1@gmail.com	1123456789	user1home	2022-09-13T08:52	SBI	user1	1234894556476232	02/23	256	

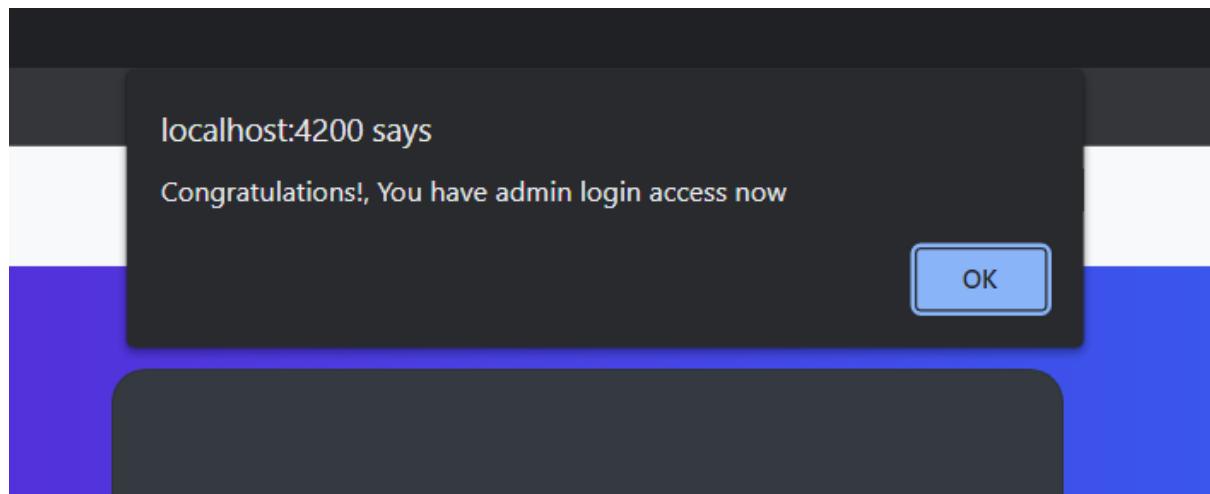
[Back Page](#)

**With this functionality , user doesn't have to enter the card details every time he orders.**

# **Administrator Access.**



**Only the database admin can access all the card details of users which gets saved in backend.**



## Admin can Delete and Update the user data.

Card User										
<input type="button" value="Show 6 entries"/> <input type="text" value="Search:"/>										
Card user	Card Email	Card Phone	Card Address	Card Date_Time	Card Branch	Card User Name	Card new_Number	Card Expaire	Card Cvv	Action
saket	saketis@gmail.com	7004551155	Kormangala	2022-09-09T21:23	SBI	Saket Narnolia	2541 8564 7894	01/23	256	<input type="button" value="Delete"/> <input type="button" value="Update"/>
narendra modi	mitron@gmail.com	1112454567	Rashtrapati bhawan	2022-09-12T18:07	SBI	narendra modi	987445644545	02/23	256	<input type="button" value="Delete"/> <input type="button" value="Update"/>
modi	modi@gmail.com	1234567891	dnsajd	2022-09-12T18:13	SBI	modi	1234123412341234	02/23	256	<input type="button" value="Delete"/> <input type="button" value="Update"/>
user1	user1@gmail.com	1123456789	user1home	2022-09-13T08:52	SBI	user1	1234894556476232	02/23	256	<input type="button" value="Delete"/> <input type="button" value="Update"/>
user2	useremail@gmail.com	1223456789	user2home					admin	admin	<input type="button" value="Delete"/>

user8home

userhome

userhome

userhome

user12hom

Kormangala

## Update Card

card User Name

Hari

card Email

hari@gmail.com

card User phone

9931727685

card User Address

Kormangala

card User Date\_Time

2022-09-13T17:00

Select Bank Branch

RBL

card User Name

**Click on the Update button to Save the Updated data.**

card User Expaire MM/YY

card User cvv

**Update** **Close**

**Update will be successfully saved.**

home

ome

ome

ome

### Update Card

Card Detail Updated!

**Close**

2022-09-13T13:55

Card user	Card Email	Card Phone	Card Address	Card Date_Time	Card Branch	Card User Name	Card new_Number	Card Expaire	Card Cvv	Action
Hari	hari@gmail.com	9931727685	Kormangala	2022-09-13T17:00		Hari Sinha	1234852878945612	04/25	214	<b>Delete</b> <b>Update</b>

## Developer's details.

Name: Rishu Goswami.

Designation: Full Stack Developer.

---