



**INSTITUTE FOR ADVANCED  
COMPUTING AND SOFTWARE  
DEVELOPMENT (IACSD), AKURDI  
PUNE**

Documentation on

**The Tasty Order Hub**

PG-DAC September 2023

**Submitted By:**

Group No. : 80

<b>Roll No</b>	<b>Name</b>
<b>239144</b>	<b>Shital Dineshrao Bhendekar</b>
<b>239142</b>	<b>Apoorva Deepak Bamgude</b>

**Mrs. Monika Sindhikar**

**Project Guide**

**Mr. Rohit Puranik**

**Centre Co-ordinator**

## **ABSTRACT**

An Online Food Ordering System is proposed here which simplifies the food ordering process. The proposed system shows an user interface and update the menu with all available options so that it eases the customer work. Customer can choose more than one item to make an order and can view order details before logging off. The order confirmation is sent to the customer. The order is placed in the queue and updated in the database and returned in real time. This system assists the staff to go through the orders in real time and process it efficiently with minimal errors.

## **ACKNOWLEDGEMENT**

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavour to a successful culmination. I extend my sincere and heartfelt thanks to our esteemed guide, Mrs. Monika Sindhikar for providing me with the right guidance and advice at the crucial juncture and for showing me the right way. I extend my sincere thanks to our respected Centre Co-ordinator Mr. Rohit Puranik, for allowing us to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement they have given me during the course of our work.

Apoorva Deepak Bamgude (230941220037)

Shital Dineshrao Bhendekar (230941220039)

## **INDEX**

<b>Sr. No.</b>	<b>Title</b>	<b>Page no.</b>
1.	Introduction	1
1.2	Purpose	1
1.3	Scope	1
2.	Requirement	2
2.1	Functional Requirement	2
2.2	Non-Functional Requirement	3
2.3	Other Requirements	5
3.	Database Design	6
4.	Appendix B: Analysis Model	11
5.	Use Case diagrams	13
6.	Data Flow Diagram	16
7.	ER Diagram	19
8.	Class Diagram	20
9.	Sequence Diagram	21
10.	Appendix C: Screen Shots	22
11.	Conclusion	25
12.	References	26

## **1. Introduction**

The Online Food Ordering and Delivery System is a web-based application that enables users to browse a variety of restaurants, select dishes, place orders, and have the food delivered to their location. This document outlines the requirements for the development of the application using Java, Spring Boot for the backend, and React, HTML, and CSS for the frontend.

### **Purpose**

The purpose of this document is to define the functional and non-functional requirements of the Online Food Ordering and Delivery System. It provides a clear understanding of the application's features, its interactions with users and restaurants, and the technical constraints under which it operates.

### **Scope**

The Online Food Ordering and Delivery System application will allow users to:

- Browse a list of restaurants.
- View menus, dishes, and prices.
- Add dishes to the cart.
- Customize orders (e.g., special instructions, quantity).
- Place orders and make payments.
- Track the status of orders.
- Provide feedback and ratings for restaurants.

Objective Of The Tasty Order Hub

### **Functionalities provided by The Tasty Order Hub**

“The Tasty Order Hub” provides a range of functionalities aimed at simplifying and enhancing the food ordering and delivery for users. Some of the key functionalities offered by the project include:

## **2. Requirements**

### **2.1 Functional Requirements**

#### **FR 1. System Features**

- User authentication: Ensure that the user's credentials are valid and match those in the app's database.
- Session management: Keep track of the user's session and log them out after a certain amount of time or when they log out manually.
- Data encryption: Ensure that the user's password is encrypted before it is stored in the database.

#### **FR 2. Customer Dashboard**

- Users should be able to browse a list of restaurants available on the platform. Each restaurant should have a menu that users can view.
- Users should be able to select items from the restaurant's menu and place an order. Users should be able to specify delivery details.
- Users should be able to view their cart, modify quantities, and remove items. The total order amount should be visible in the cart.
- Users should be able to leave reviews and ratings for restaurants. Restaurants should be able to view and respond to customer reviews.

#### **FR 3. Restaurant Dashboard**

- Restaurant owners should have access to a dashboard to manage their restaurant's information and menu.
- Restaurant owners should be able to update menu.
- Restaurant owners should be able to view and fulfil orders.

#### **FR 4. Admin Role**

- Admin should be able to log in using unique credentials
- Admin should be able to manage customer and restaurant accounts.
- Admin should be able to approve Restaurant Registration.
- Admin should be able to block and unblock restaurants.

## 2.2 Non-Functional Requirements

- Registered Customer will allowed to place an order.
- Each stakeholder will be to access system through authentication process. Who are you?
- System will provide access to the content, operations using Role based security (Authorization) (Permissions based on Role)
- Using SSL in all transactions which will be performed stakeholder. It would protect confidential information Shared by stakeholder to system and vice versa.
- System will automatically log out all stakeholder after some time due to inactiveness. System will block operations for inactive stakeholder and would redirect for authentication.
- System will internally maintain secure communication channel between Servers (Web Servers, App Servers, database Server)
- Sensitive data will be always encrypted across communication.
- User proper firewall to protect servers from outside fishing, vulnerable attacks.

### NFR 1.Reliability

- The system will backup business data on regular basis and recover in short time duration to keep system operational.
- Continuous updates are maintained, continuous Administration is done to keep system operational.
- During peak hours system will maintain same user experience by using load balancing.

### NFR 2.Maintainability

- A Commercial database software will be used to maintain System data Persistence.
- A readymade Web Server will be installed to host Lunchbox (Web Site) to manage server capabilities.
- Staff will easily monitor and configure System using administrative tools provided by Servers.
- Separate environment will be maintained for system for isolation in production, testing, and development.

### NFR 3.Portability

- System will provide portable User Interface (HTML, CSS, JS) through which users will be able to access The Tasty Order Hub.
- System can be deployed to single server, multi-server, to any OS, Cloud (Azure or AWS or GCP).

**NFR 4.Availability**

- uptime: 24\* 7 available 99.999%

**NFR 5.Accessibility**

- Only registered customer will be able to place an order after authentication.
- Vendors will be able to maintain a menu and can reject or approve orders.
- Staff will be able to view daily, weekly, monthly, annual business Growth through customized dashboard.

**NFR 6.Durability**

- System will retain customer order cart for 15 minutes even though customer loose internet connection and join again.
- Customer will be able to add or remove tiffin's from order cart whenever needed.
- System will implement backup and recovery for retaining stakeholders' data, business operation data and business data over time.

**NFR 7.Efficiency**

- On Festival season, maximum number of users will place order, view available tiffin services with same response time.
- System will be able to manage all transactions with isolation.

**NFR 8.Modularity**

- System will designed and developed using reusable, independent or dependent business scenarios in the form of modules.
- These modules will be loosely coupled and highly cohesive.
- System will contain login, registration, menu, order cart, order processing, paymentprocessing, membership and Roles management modules.

**NFR 9.Scalability**

- System will be able to provide consistent user experience to stakeholder as well asvisitors irrespective of load.

**NFR 10.Safety**

- Its functionalities are protected from outside with proper firewall configuration. It will be always kept updated with latest anti-virus software.

**NFR 11.Performance**

- The application should load restaurant menus and process orders within 5 seconds.



## **NFR 12.User Interface**

- The frontend should provide an intuitive and responsive user interface for both desktop and mobile devices.

## **2.3 Other Requirements:**

- Front End: ReactJS
- Back End: Java Spring Boot REST API
- Database: MYSQL

### 3. DATABASE DESIGN

#### Database Design

The following table structures depict the database design.

#### Tables

```
mysql> use lunchbox
Database changed
mysql> show tables
-> ;
```

Tables_in_lunchbox
address
customer_plans
customers
hibernate_sequence
logins
orders
otps
payments
plan_orders
subscription_plans
tiffin
vendors

```
12 rows in set (0.15 sec)
```

#### Table 1:

```
mysql> desc address
-> ;
```

Field	Type	Null	Key	Default	Extra
address_id	bigint	NO	PRI	NULL	
address	varchar(100)	YES		NULL	
city	varchar(20)	YES		NULL	
pincode	int	YES		NULL	
state	varchar(20)	YES		NULL	
street	varchar(100)	YES		NULL	

```
6 rows in set (0.01 sec)
```

#### Table 2:

```
mysql> desc customer_plans
-> ;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
end_date	date	YES		NULL	
start_date	date	YES		NULL	
customer_id	bigint	YES	MUL	NULL	
subscription_plan_id	bigint	YES	MUL	NULL	

5 rows in set (0.00 sec)

Table 3:

```
mysql> desc customers
-> ;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
email	varchar(50)	YES	UNI	NULL	
first_name	varchar(50)	YES		NULL	
last_name	varchar(50)	YES		NULL	
mobile	varchar(10)	YES	UNI	NULL	
profile_image	varchar(255)	YES		NULL	
register_date	datetime(6)	YES		NULL	
address_id	bigint	YES	MUL	NULL	

8 rows in set (0.00 sec)

Table 4:

```
mysql> desc hibernate_sequence
-> ;
```

Field	Type	Null	Key	Default	Extra
next_val	bigint	YES		NULL	

1 row in set (0.30 sec)

Table 5:

```
mysql> desc logins
-> ;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
email	varchar(35)	YES	UNI	NULL	
password	varchar(350)	YES		NULL	
user_role	varchar(255)	YES		NULL	

```
4 rows in set (0.00 sec)
```

Table 6:

```
mysql> desc orders
-> ;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
date_time	datetime(6)	YES		NULL	
quantity	int	NO		NULL	
customer_id	bigint	YES	MUL	NULL	

```
4 rows in set (0.00 sec)
```

Table 7:

```
mysql> desc payments
-> ;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
amount	double	YES		NULL	
payment_time	datetime(6)	YES		NULL	
payment_type	varchar(255)	YES		NULL	
razorpay_payment_id	varchar(255)	YES		NULL	
status	varchar(255)	YES		NULL	
order_id	bigint	YES	MUL	NULL	

```
7 rows in set (0.00 sec)
```

Table 8:

```
mysql> desc plan_orders
-> ;
+-----+-----+-----+-----+-----+-----+
| Field          | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| subscription_id | bigint | NO   | PRI | NULL    |       |
| order_id        | bigint | NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Table 9:

```
mysql> desc subscription_plans
-> ;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id             | bigint        | NO   | PRI | NULL    | auto_increment |
| description     | varchar(255)  | YES  |     | NULL    |               |
| image          | varchar(255)  | YES  |     | NULL    |               |
| is_avaliable   | bit(1)        | NO   |     | NULL    |               |
| name           | varchar(255)  | YES  |     | NULL    |               |
| plan_type      | varchar(255)  | YES  |     | NULL    |               |
| price          | double        | NO   |     | NULL    |               |
| vendor_id      | bigint        | YES  | MUL | NULL    |               |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Table 10:

```
MySQL 8.0 Command Line Client
| vendor_id | bigint | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> desc tiffin
-> ;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id             | bigint        | NO   | PRI | NULL    | auto_increment |
| day            | varchar(255)  | YES  |     | NULL    |               |
| description     | varchar(255)  | YES  |     | NULL    |               |
| food_type      | varchar(255)  | YES  |     | NULL    |               |
| image_path     | varchar(255)  | YES  |     | NULL    |               |
| name           | varchar(255)  | YES  |     | NULL    |               |
| price          | double        | NO   |     | NULL    |               |
| subscription_plan_id | bigint        | YES  | MUL | NULL    |               |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Table 11:

```
mysql> desc otps
-> ;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
date_created	datetime(6)	YES		NULL	
email	varchar(35)	YES		NULL	
otp	int	NO		NULL	

```
4 rows in set (0.00 sec)
```

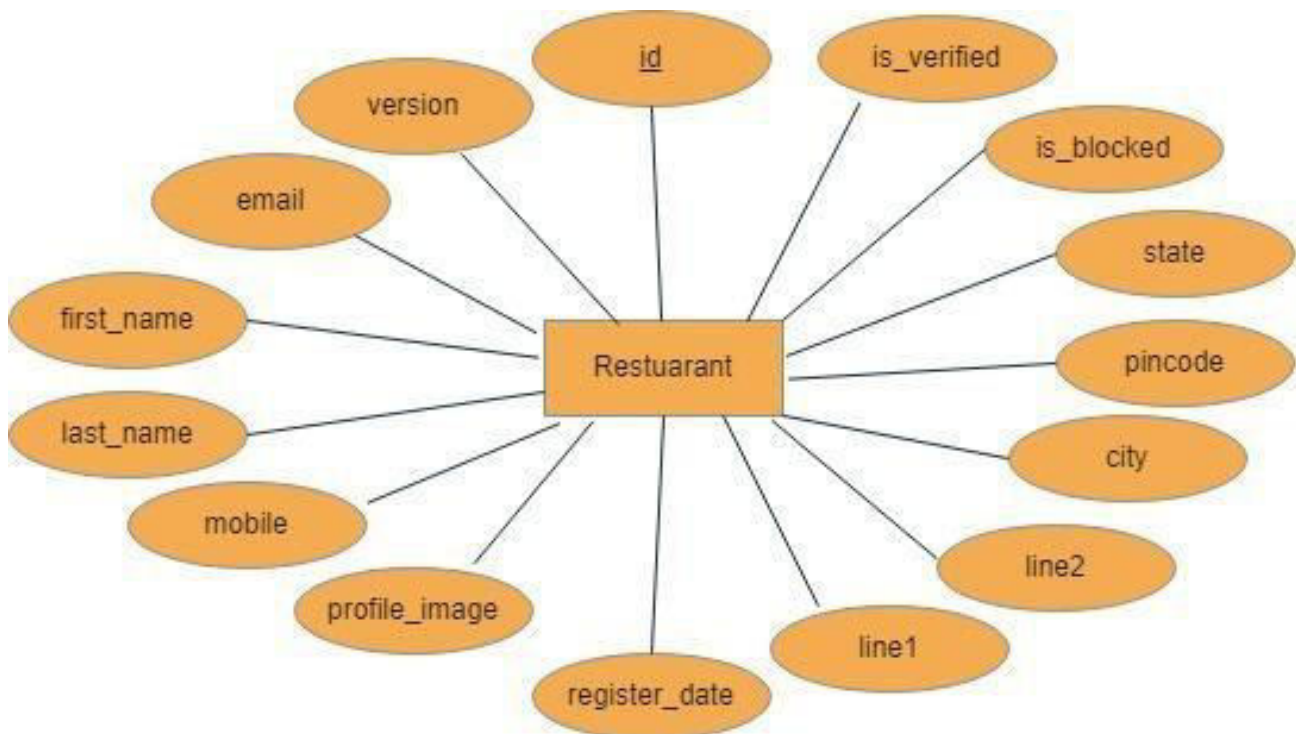
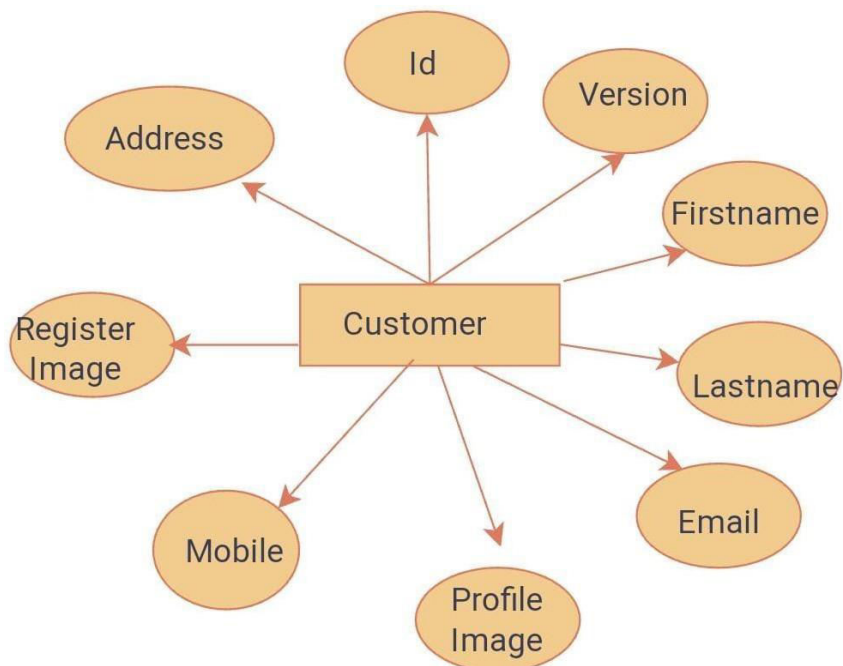
Table 12:

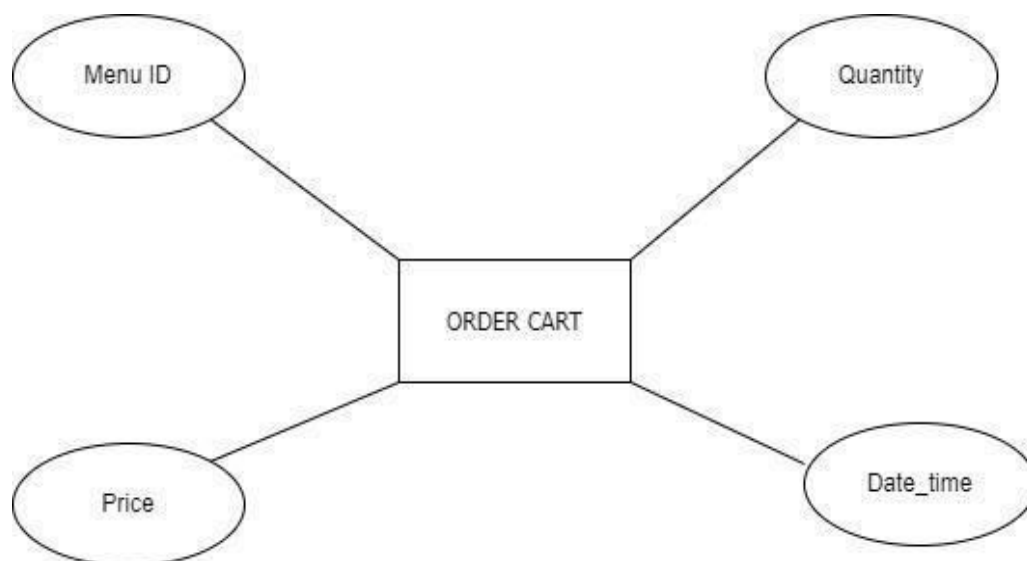
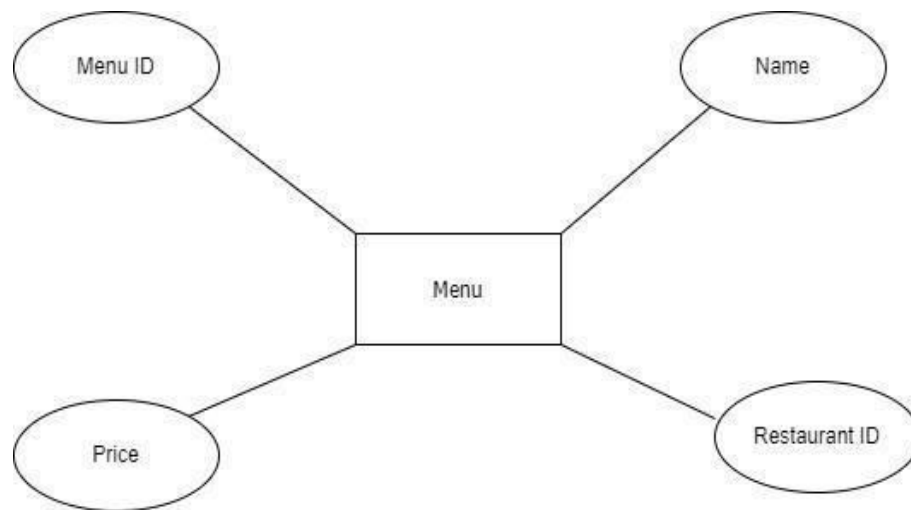
```
mysql> desc vendors
-> ;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
email	varchar(50)	YES	UNI	NULL	
first_name	varchar(50)	YES		NULL	
last_name	varchar(50)	YES		NULL	
mobile	varchar(10)	YES	UNI	NULL	
profile_image	varchar(255)	YES		NULL	
register_date	datetime(6)	YES		NULL	
is_blocked	bit(1)	NO		NULL	
is_verified	bit(1)	NO		NULL	
address_id	bigint	YES	MUL	NULL	

```
10 rows in set (0.00 sec)
```

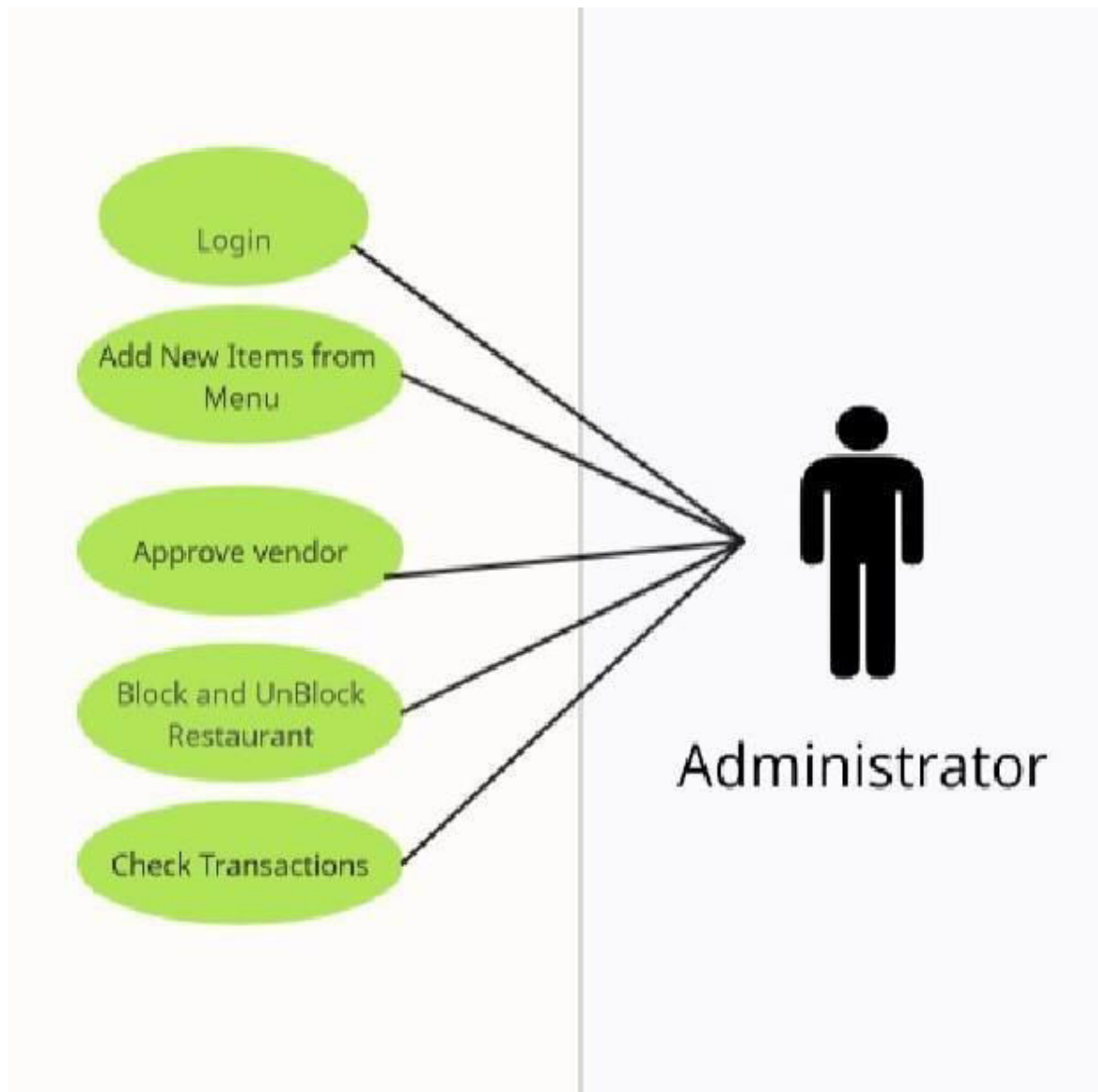
## APPENDIX B: ANALYSIS MODEL



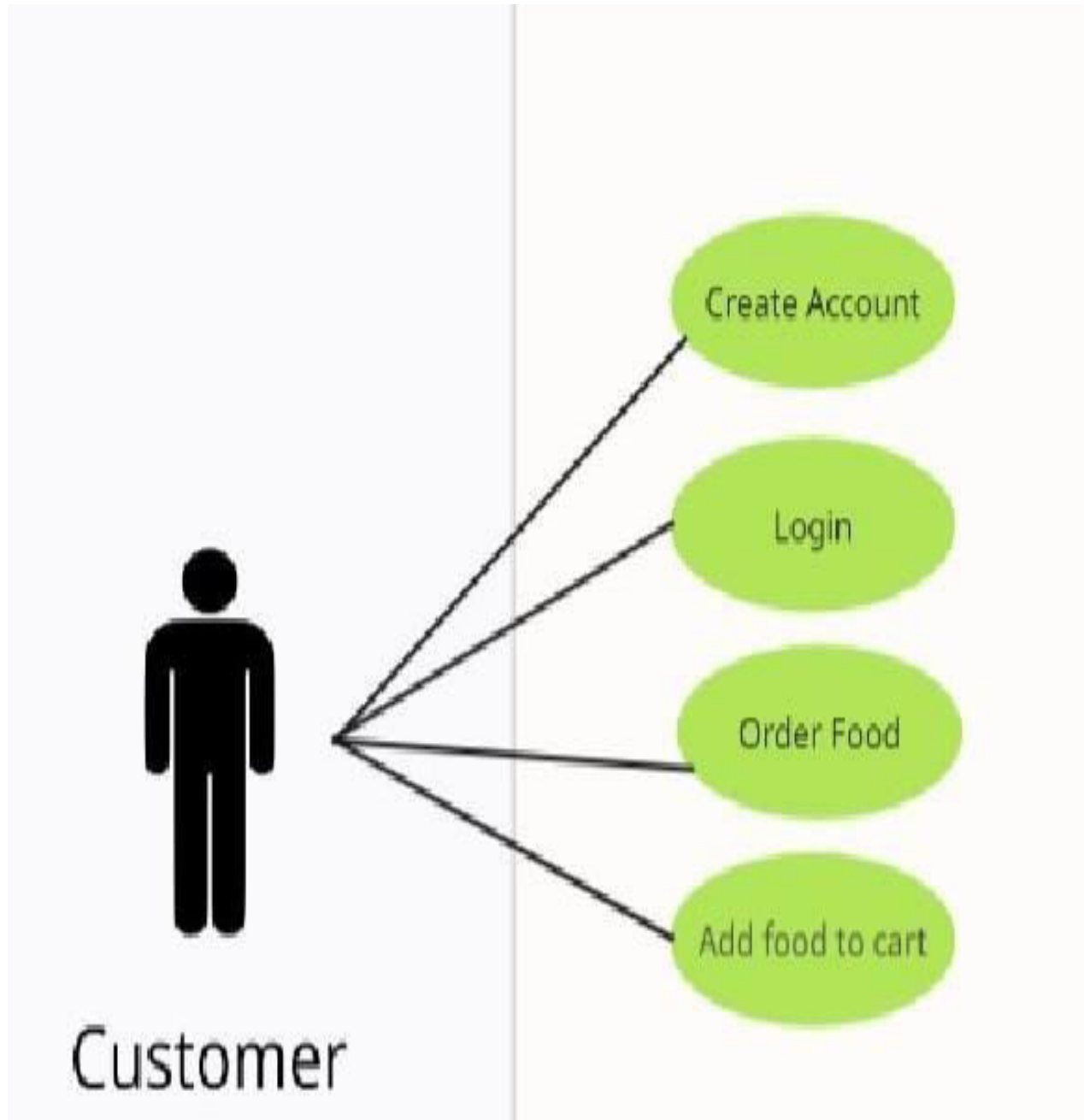




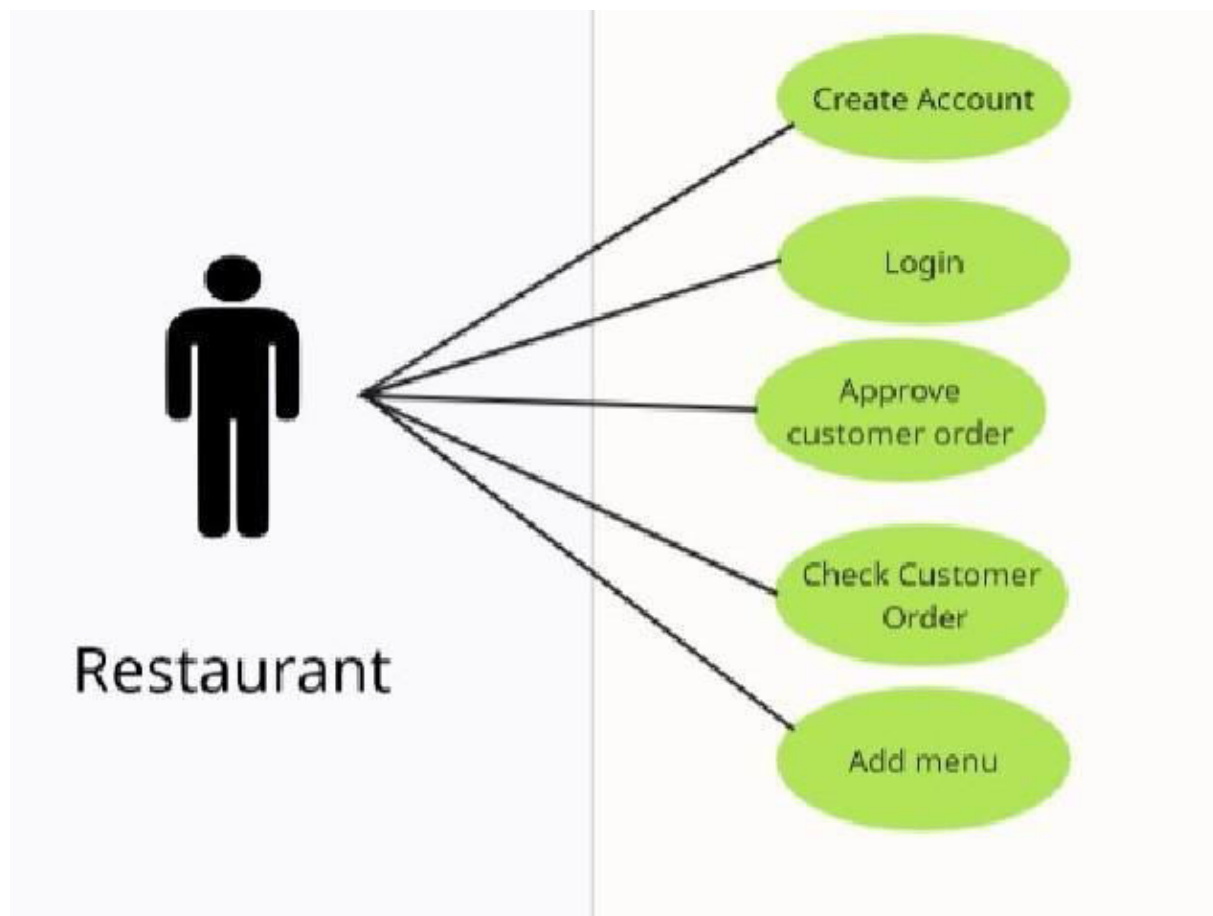
## Use case diagram for Admin



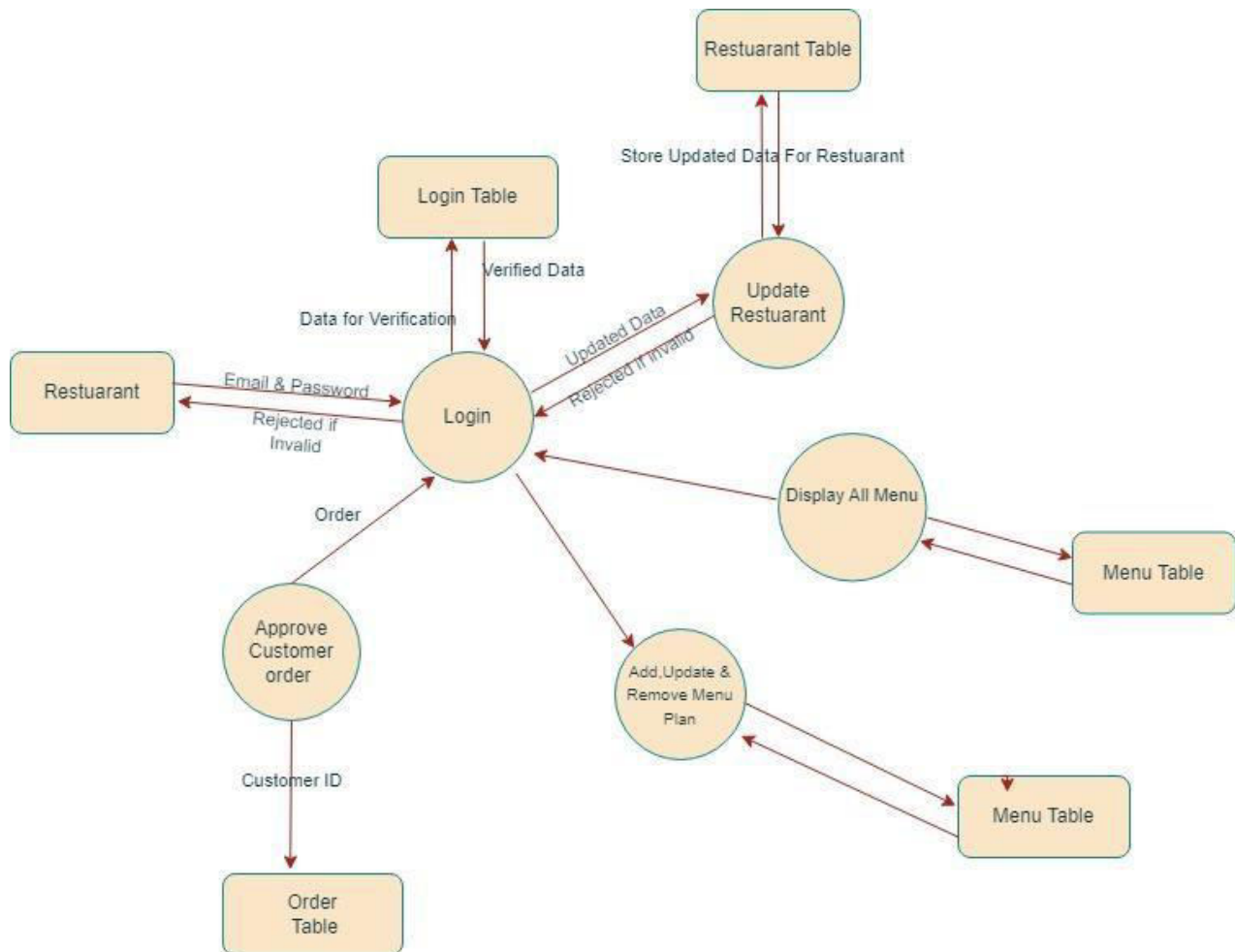
## Use Case diagram for Customer

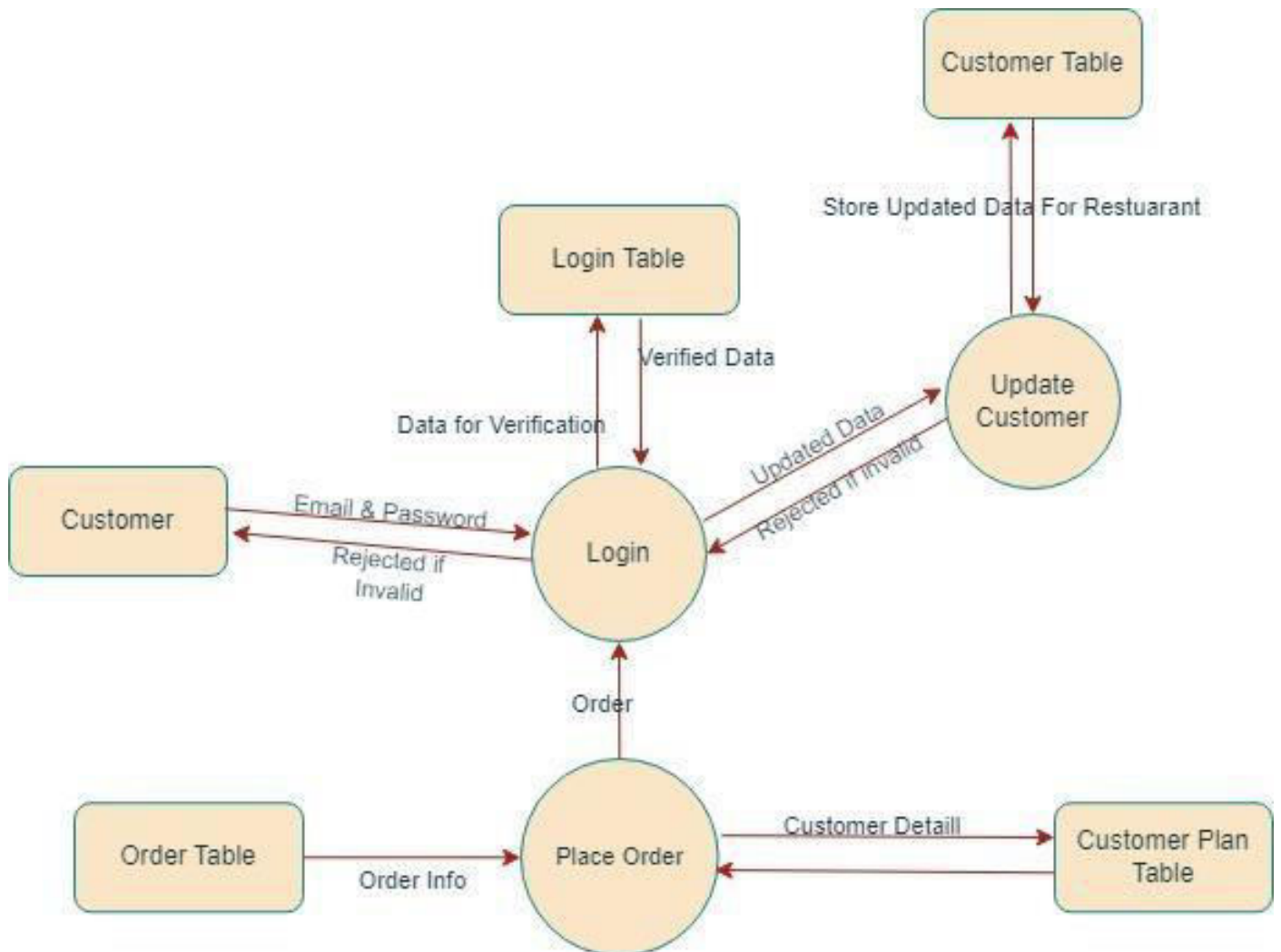


## Use case diagram for Restaurant

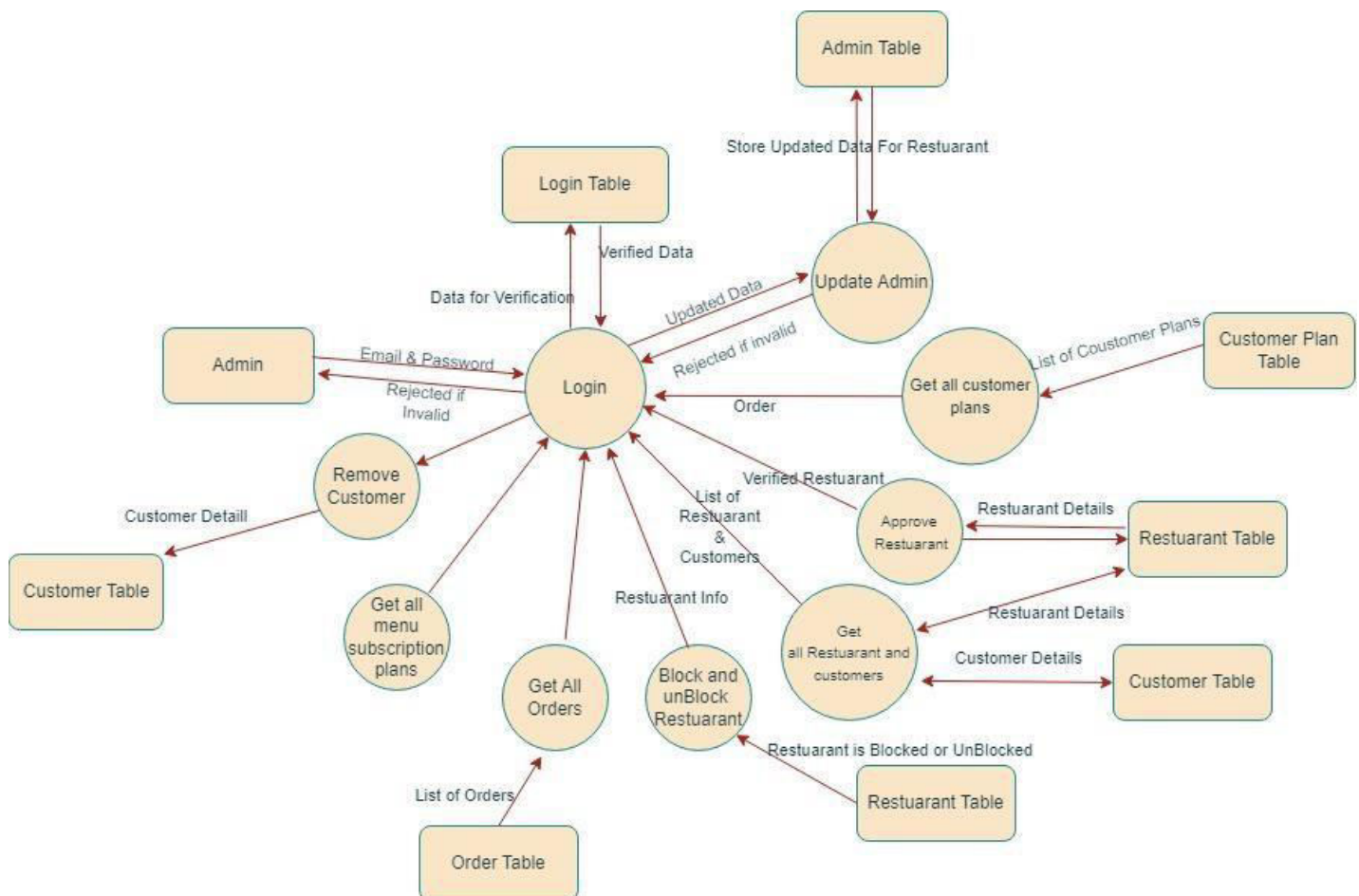


## DATA FLOW DIAGRAM FOR VENDOR / RESTAURANT

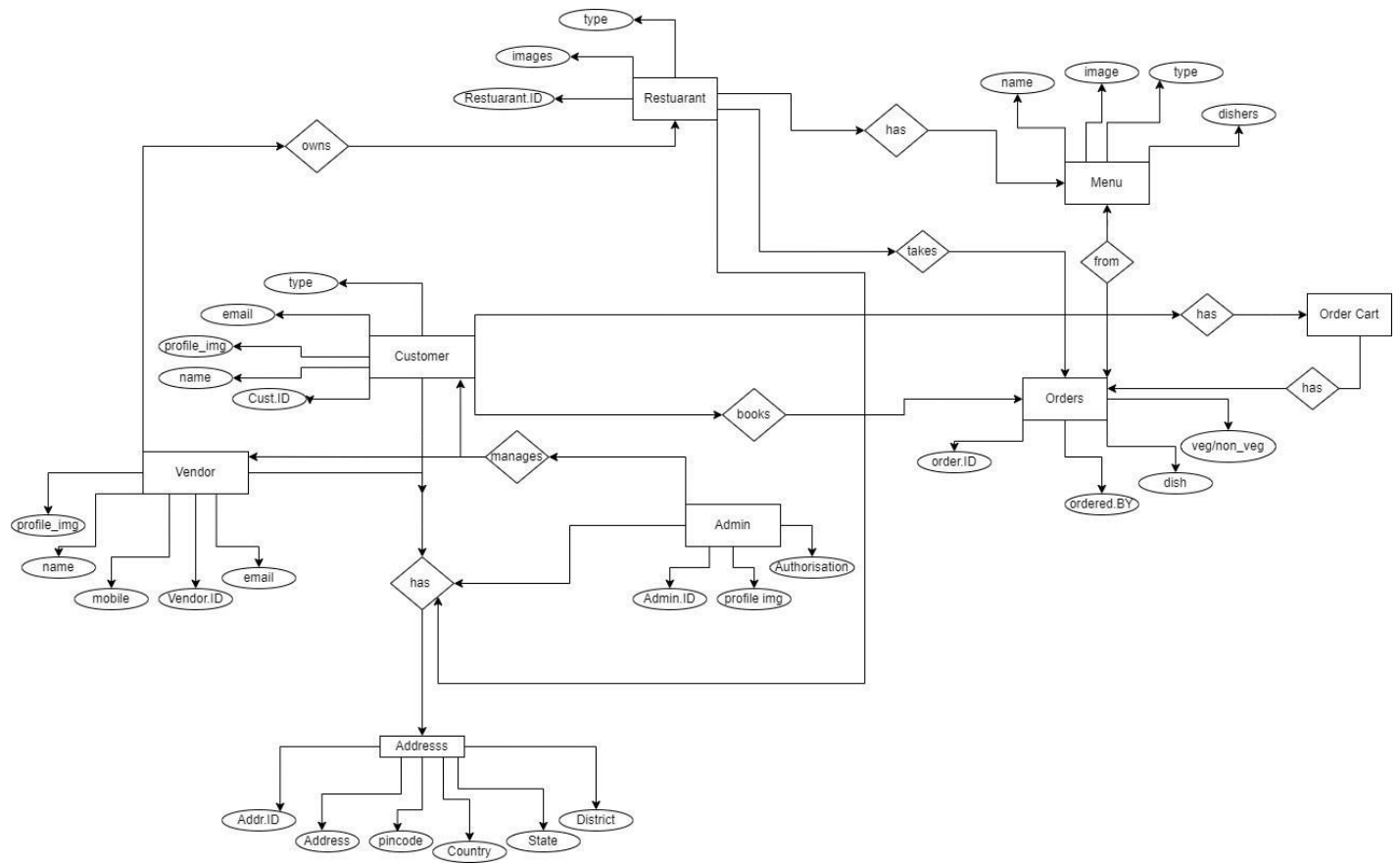


**DATA FLOW DIAGRAM FOR CUSTOMER**

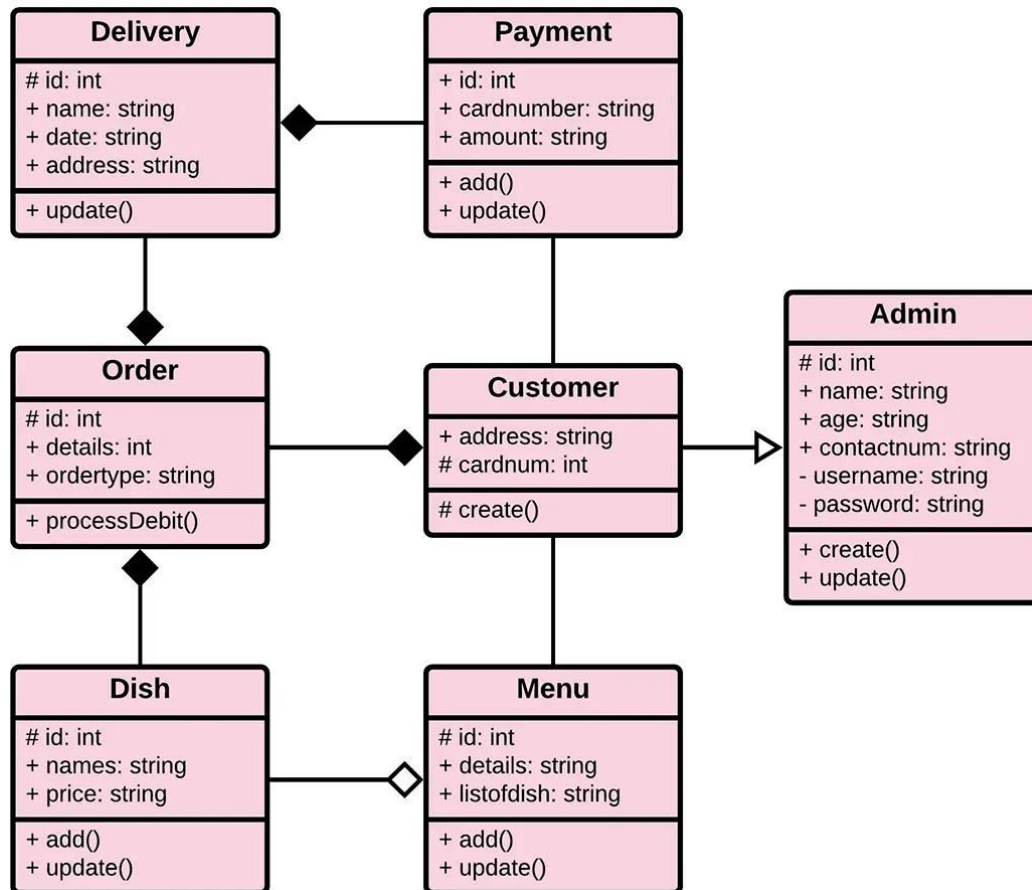
## DATA FLOW DIAGRAM FOR ADMIN



## Entity Relationship Diagram

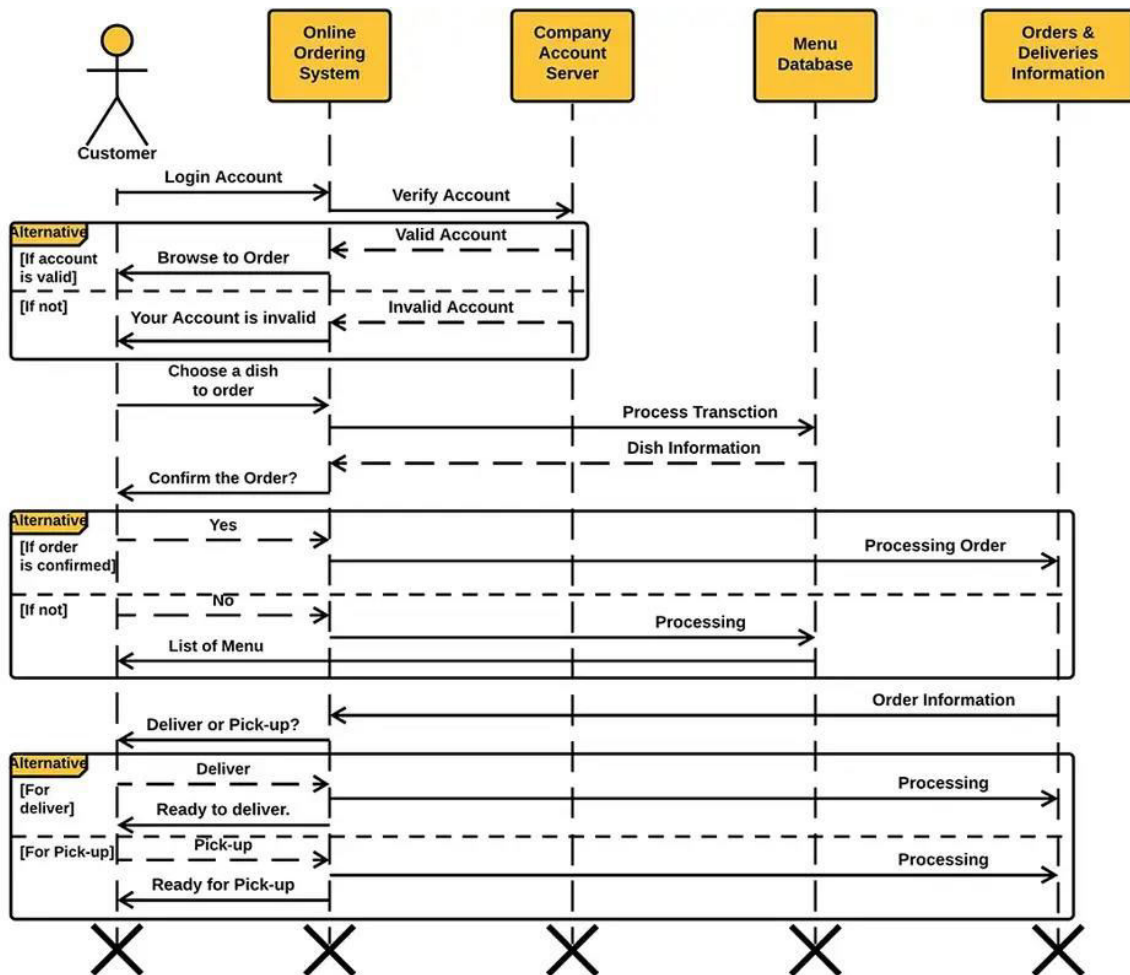


## CLASS DIAGRAM



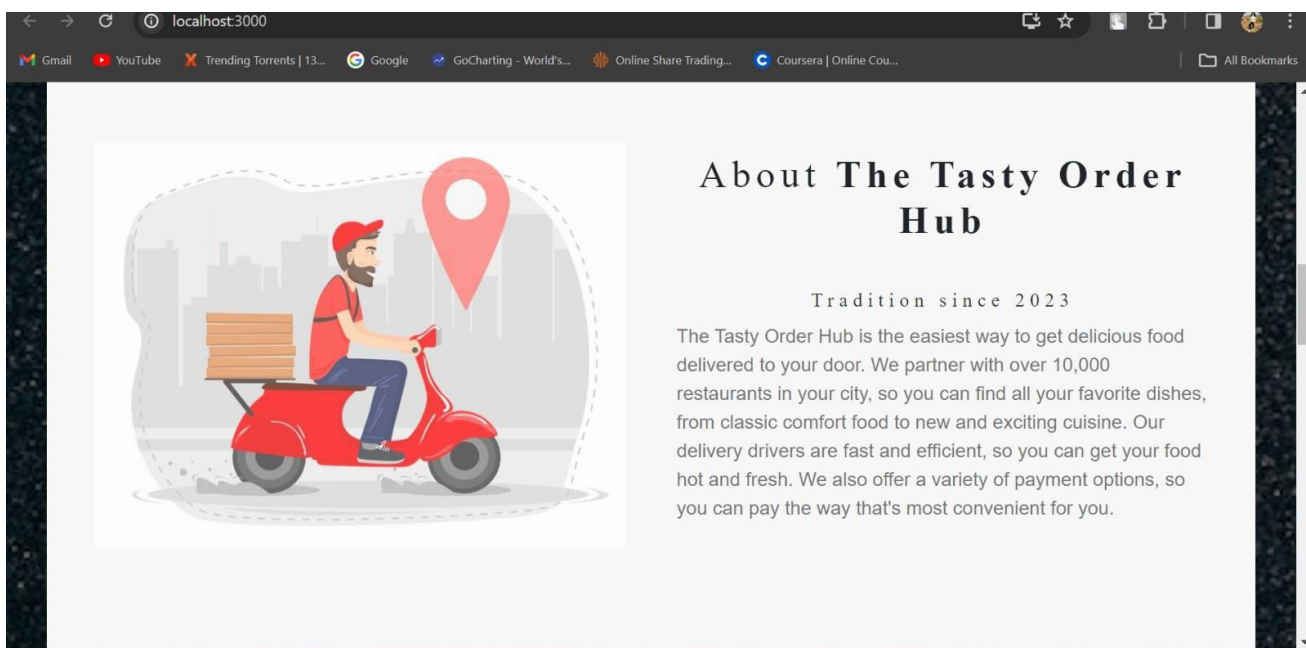
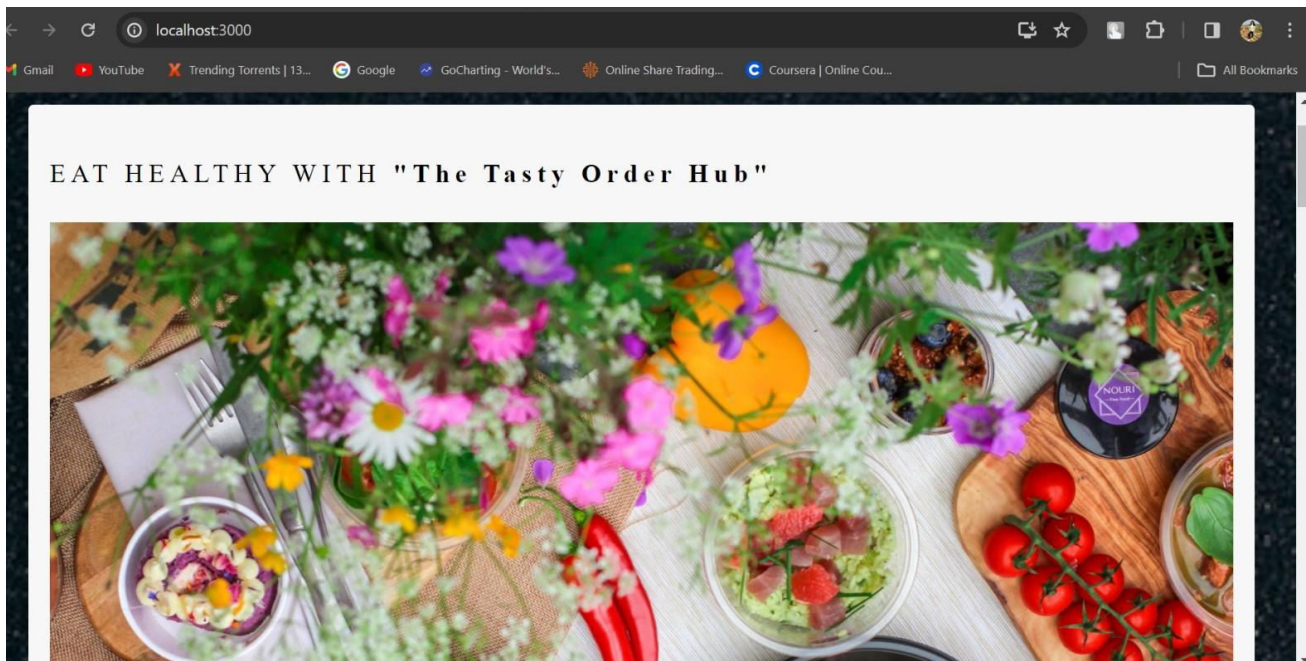


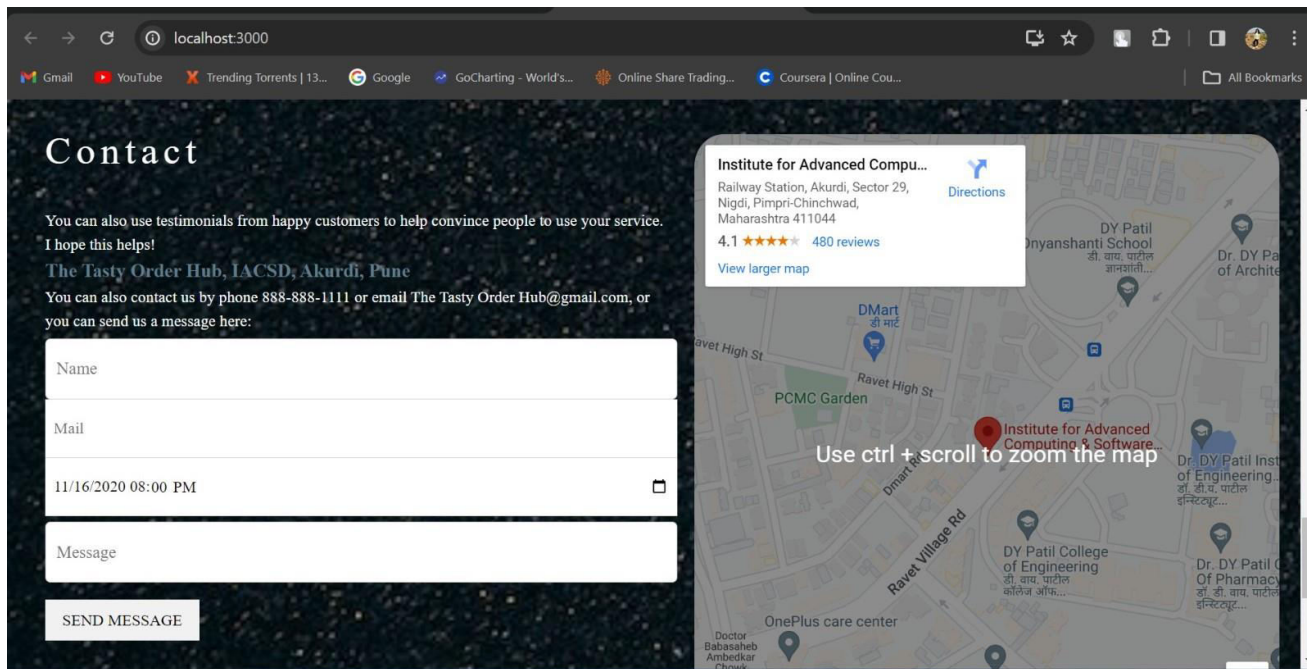
# SEQUENCE DIAGRAM



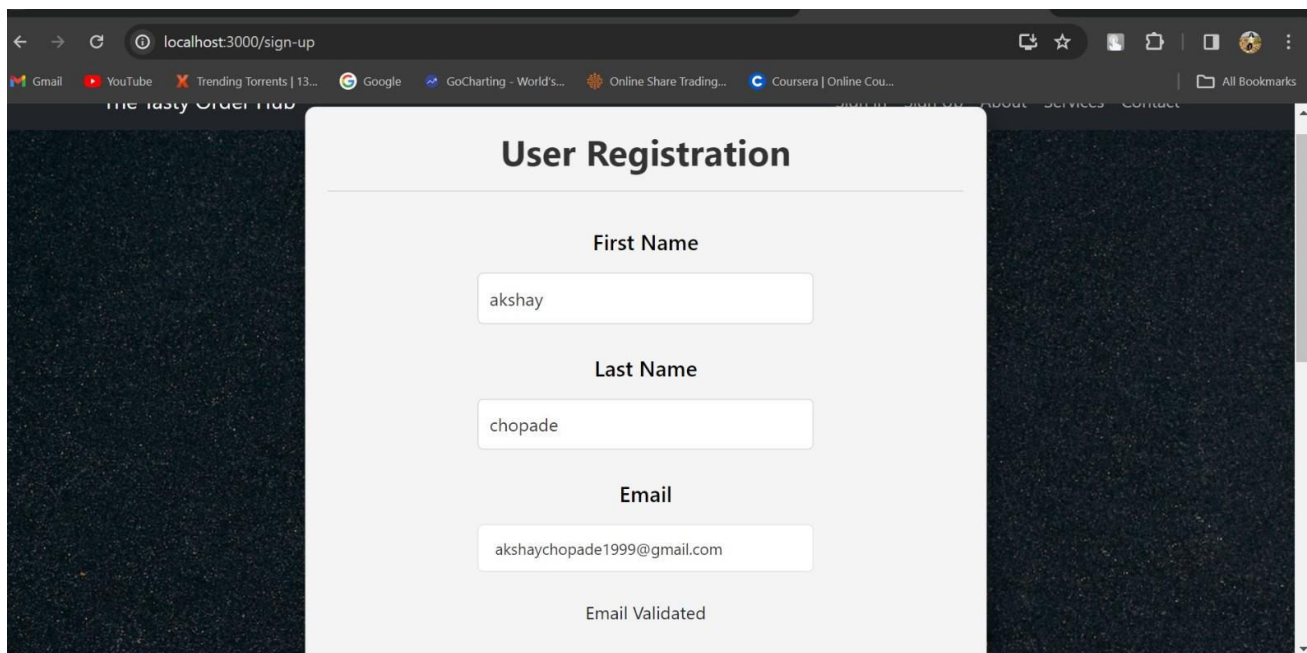
## APPENDIX C: Screen Shots

### Homepage:



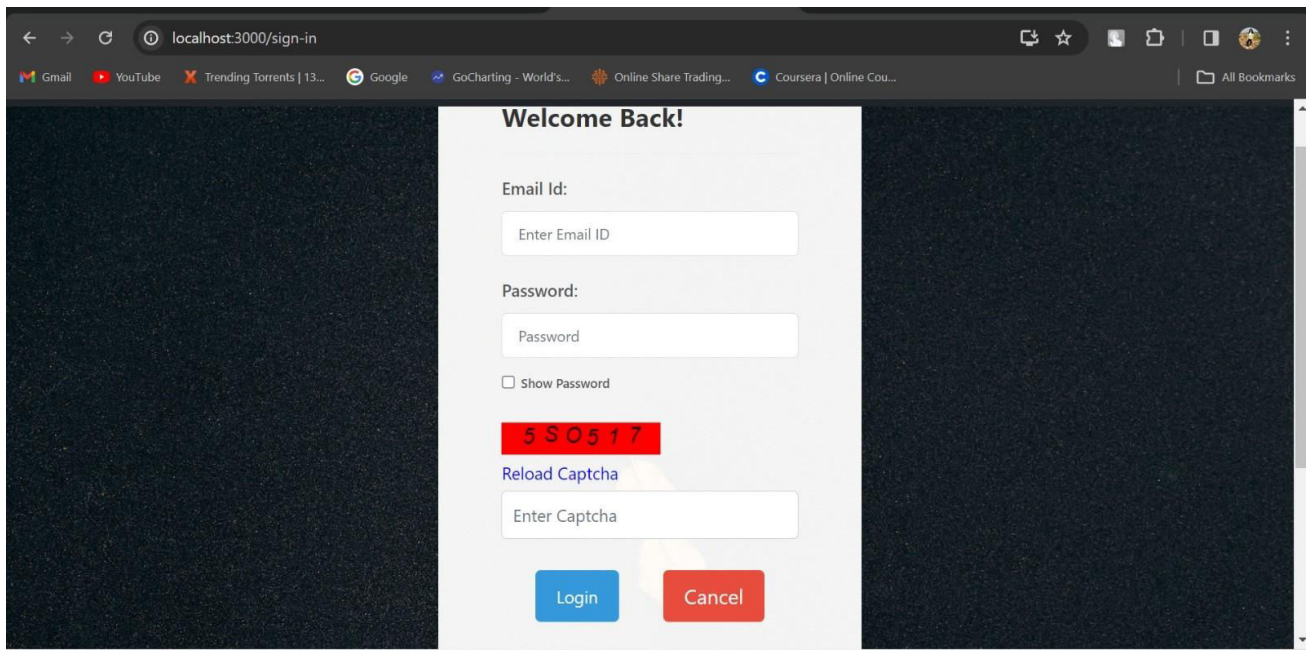


## Register page:





## Login page:



localhost:3000/sign-in

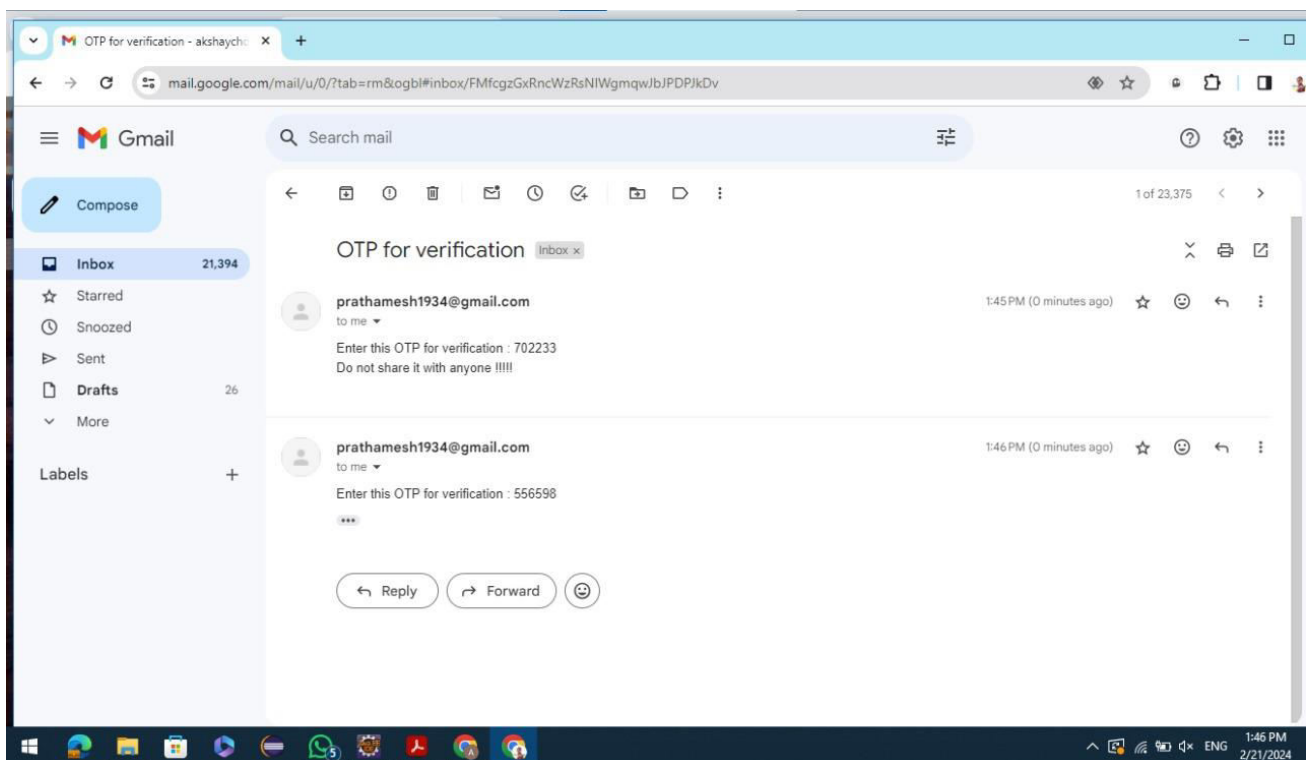
**Welcome Back!**

Email Id:

Password:  
  
☐ Show Password

**5S0517**  
[Reload Captcha](#)

## OTP:



## **CONCLUSION**

- The Tasty Order Hub is an online food ordering system where customers can view menu, select restaurant of their choice and place order.
- Restaurants/Vendors can be able to register their restaurant, update menu, etc.
- Admin manages restaurants, customers, orders, etc.
- The Tasty Order Hub gives an outstanding experience not only to customers but also to vendors.

## REFERENCES

The following references were used in the creation of this document:

- <https://www.javascript.com/>
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
- Spring Boot + React: JWT Authentication with Spring Security – BezKoder.