School of Computer Science and Engineering (SCOPE)

VIEAT – Online Food Ordering and Delivery Website

Course Code: CSE2004

Course Title: Database Management Systems

Class Number: CH2020211700475

Semester: Winter 2020-21

Slot: D1

Team members:

Srinath NS, 20BCE1074

Kommareddy Saketh Reddy, 20BCE1042

Subramanian Nachiappan, 20BCE1019

Nilavan I, 20BCE1080

Course Faculty: Dr. L.M. Jenila Livingston

1. Abstract

Online Food Ordering and Delivery platforms such as Swiggy and Zomato were highly sought after, during the COVID-19 outbreak due to its flawless website and database functionality. The online food ordering platforms proved to be useful for the customers who needn't step out to buy their food. However, sources claim that [1] Swiggy partners with restaurants on a 12 to 18-month contract and as the contracts come up for renewal, it raises the charges up to 18-23% of the total value from 12-18% initial deal. Also, it has been claimed that at [2]Zomato the commission rates vary anywhere between 18-40% of the order value. This proves to be detrimental for the restaurants as they earn less-to-no profit in the orders they serve via any online platforms. [3] In fact, prices of food items in Zomato or Swiggy menu are 25 to 50% more than actual price of the same at restaurant. It is to be noted that around 70% consumers agree with the concept of online food ordering system because there's a high possibility that they may be paying more than the food's price while using a third-party ordering system. Hence it is beneficial for a restaurant to have its own website for its flawless performance.

Therefore, this project is inspired from this idea and hopes to fulfil the requirements by developing a website prototype capable of managing food order from the sides of the customers, admin and the delivery person. The main highlight of the project is its database functionality. This food ordering website will be able to attract more customers by having a user-friendly Interface and a strong database. The website makes use of a powerful tech stack containing HTML, CSS, Java Script, PHP, MySQL and Bootstrap for efficient querying and user-friendly themes. The outcome of the project will throw light on the fact that database functionality of a website is necessary for its seamless performance.

2. Introduction

The Online Food Ordering and delivery system has been developed to override the problems prevailing in the practicing manual system and most importantly the problems faced while integrating with a third-party agent for online management. Every organization irrespective of its size and capability faces challenges to overcome and manage the information of Food Category, Food Item, Order, Payment, Delivery Person, so on and so forth. Therefore, this project comprises of an admin section to manage over all the aspects of the website, a delivery person section to manage the deliveries, and obviously the user section where they can make their favourite orders. This application is reduced as much as possible to avoid any

sort of errors viz., redundancy and data insertion, deletion and modification. Thus, it will help both the organization and the user in the better utilization of the resources. The main goal is to maintain the functionalities of the restaurants in an effective and accurate manner and also it is reducing the use of manual entries. The web application keeps a proper record of the restaurant's transactions in the database and paves way to the efficient functioning of the restaurant.

3. Project Motive

The main motive of the project on Online Food Ordering and Delivery system is to manage the details of food item, category, user, order and invoice. It manages all the information about Food item, Payment, Order details and its respective invoice. The project is totally built at administrative end and thus only the admin is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the food item, category, payment and user. It tracks all the details about the customer, order and its invoice. The functionalities that are integrated the system are listed below.

- Create, Retrieve, Update and Delete the records with respect to the food ordering system is improved.
- Search Facilities based on food items, categories and types.
- The groups of user(citizens) that had been identified to use the system are client (customer) the administrator and the delivery person.
- The user will register to be a member to use the food ordering system. The user can place orders and manage their account online.
- Administrator is the person who will manage the entire system. This type of user will also do maintenance and control the application of this system. Admin takes a responsibility to register a new user, delivery boy, and new menu into the database records.
- The delivery person is the person who will deliver the required order to the user and update the delivery status which is synced with the admin and the user.
- This web application can sell food products in the restaurant. It provides the convenient way to manage transactions online.

4. Database Design

4.1. ER Diagram
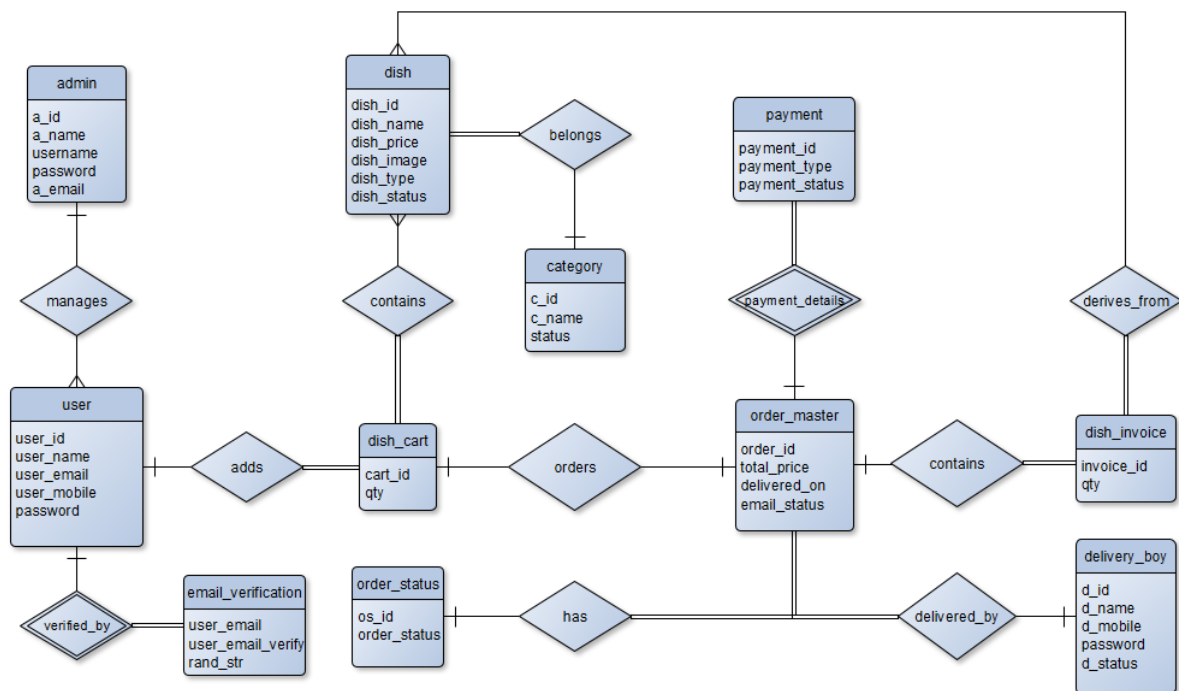
4.2. Reduction to Tables (Relational Schema)

Note: ____ →Primary Key; ____ → Foreign Key; ⁓⁓⁓→Unique Key

- **Admin** (aid, aname, username, password, a_email)
- **Category** (cid, cname, status)
- **Delivery_boy** (d_id, dname, dmobile, password, dstatus)
- **Dish** (dish_id, c_id, dish_name, dish_price, dish_image, dish_type, dish_status)
- **Dish_cart** (cart_id, user_id, dish_id, qty)
- **Dish_invoice** (invoice_id, order_id, dish_id, qty)
- **Email_verification** (user_id, user_email, email_verify, rand_str)
- **Order_master**(order_id, userid, d_id, os_id, total_price, delivered_on, email_status)
- **Order_status** (os_id, order_status)
- **Payment** (payment id, order id, payment type, payment status)
- **User** (user_id, user_name, user_email, user_mobile, password)

4.3. Normalization

1. Admin (<u>aid</u>, aname, username, password, a_email)

Aid	Aname	username	password	email
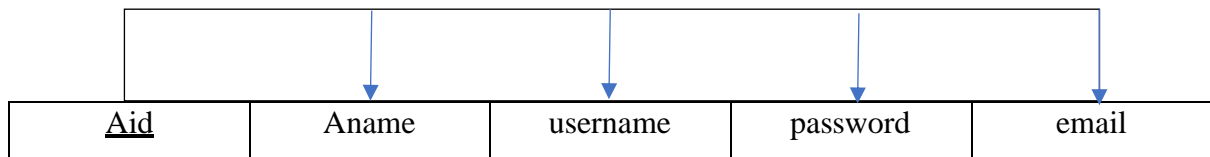
Foreign Keys: None || Primary Key: Aid || Candidate Key: Aid

Functional Dependencies:

{<u>Aid</u>}→ {A_name, Username, Password, Email}

First Normal Form

- In this Relation, No Multivalued Attribute is present. Hence this relation is in first normal form.

Second Normal Form:

- This Relation is in First Normal Form.
- There is no partial dependency present in this relation, ie only <u>aid</u> determines all attributes.
- Hence this relation is in Second Normal Form.

Third Normal Form:

- This Relation is in Second Normal Form.
- There is no transitive dependency present in this relation.
- Hence this relation is in Third Normal Form.

Boyce Codd Normal Form:

- This Relation is in Third Normal Form
- All the dependents are determined only by the candidate key. No other attribute other than <u>aid</u> is determining other attributes.
- Hence this relation is in BCNF.

Hence the finalised relation is:

Admin (<u>aid</u>, aname, username, password, a_email)

2. Order_master(<u>order_id</u>, <u>userid, d_id, os_id</u>, total_price, delivered_on, email_status)

Order_id	User_id	D_id	Os_id	Total_price	Delivered_on	Email_status
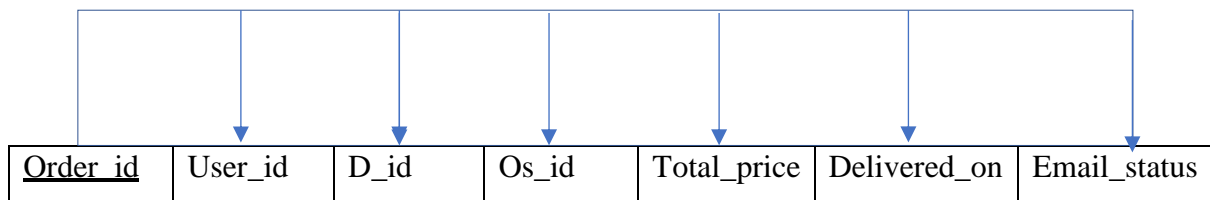
Foreign Keys: user_id, d_id, os_id || Primary Key: Order_id || Candidate Key: Order_id

Functional Dependencies:

{<u>order_id</u>}→ { <u>userid, d_id, os_id</u>, total_price, delivered_on, email_status }

First Normal Form

- In this Relation, No Multivalued Attribute is present. Hence this relation is in first normal form.

Second Normal Form:

- This Relation is in First Normal Form.
- There is no partial dependency present in this relation, ie only <u>order_id</u> determines all attributes.
- Hence this relation is in Second Normal Form.

Third Normal Form:

- This Relation is in Second Normal Form.
- There is no transitive dependency present in this relation.
- Hence this relation is in Third Normal Form.

Boyce Codd Normal Form:

- This Relation is in Third Normal Form
- All the dependents are determined only by the candidate key. No other attribute other than <u>order_id</u> is determining other attributes.
- Hence this relation is in BCNF.

Hence the finalised relation is:

Order_master(<u>order_id</u>, <u>userid, d_id, os_id</u>, total_price, delivered_on, email_status)

3. Dish (<u>dish_id, c_id</u>, dish_name, dish_price, dish_image, dish_type, dish_status)

Dish_id	C_id	Dish_name	Dish_price	Dish_image	Dish_type	Dish_status
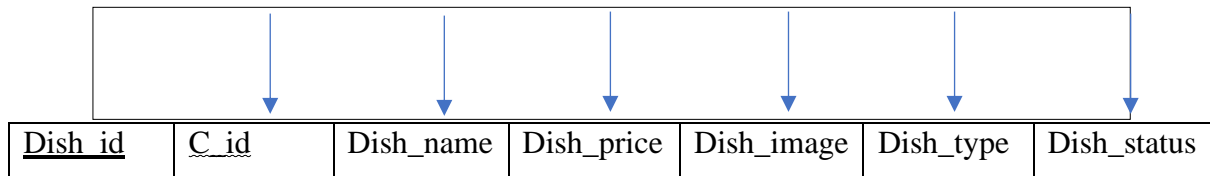
Foreign Keys: C_id || Primary Key: Dish_id || Candidate Key: Dish_id

- Here there can be multiple dishes belonging to a same category, hence category id is just a foreign key and not a candidate key.
- Whereas, <u>dish id</u> can uniquely identify each and every attribute in the relation, which is a candidate key

Functional Dependencies:

{<u>Dish_id</u>} → {_c_id, dish_name, dish_price, dish_image, dish_type, dish_status_}

First Normal Form

- In this Relation, No Multivalued Attribute is present. Hence this relation is in first normal form.

Second Normal Form:

- This Relation is in First Normal Form.
- There is no partial dependency present in this relation, ie only <u>Dish_id</u> determines all attributes.
- Hence this relation is in Second Normal Form.

Third Normal Form:

- This Relation is in Second Normal Form.
- There is no transitive dependency present in this relation.
- Hence this relation is in Third Normal Form.

Boyce Codd Normal Form:

- This Relation is in Third Normal Form

- All the dependents are determined only by the candidate key. No other attribute other than <u>dish_id</u> is determining other attributes.
- Hence this relation is in BCNF.

Hence the finalised relation is:

Dish (<u>dish_id, c_id</u>, dish_name, dish_price, dish_image, dish_type, dish_status)

4. Dish_invoice (<u>invoice_id</u>, <u>order_id, dish_id</u>, qty)

<u>Invoice_id</u>	Order_id	Dish_id	qty
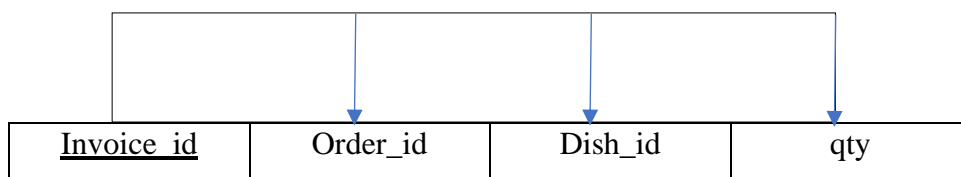
Foreign Keys: Order_id, Dish_id || Primary Key: Invoice_id || Candidate Key: None

<u>Invoice_id</u>	Order_id	Dish_id	qty
1	1	3	5
2	1	4	3
3	2	3	6

- An order can have multiple rows containing the corresponding dish ids. Hence order_id cannot be a candidate key.
- All the attributes in this relation can be determined by invoice_id which is a primary as well as the candidate key.

Functional Dependencies:

{ <u>Invoice_id</u> } → { Order_id, Dish_id, qty }

First Normal Form

- In this Relation, No Multivalued Attribute is present. Hence this relation is in first normal form.

Second Normal Form:

- This Relation is in First Normal Form.
- There is no partial dependency present in this relation, ie only Invoice_id determines all attributes.
- Hence this relation is in Second Normal Form.

Third Normal Form:

- This Relation is in Second Normal Form.
- There is no transitive dependency present in this relation.
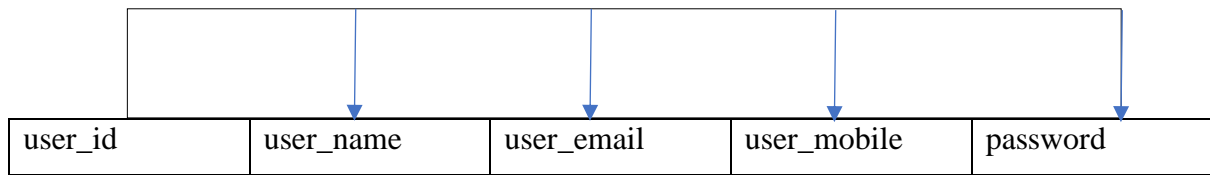- Hence this relation is in Third Normal Form.

Boyce Codd Normal Form:

- This Relation is in Third Normal Form
- All the dependents are determined only by the candidate key. No other attribute other than Invoice_id is determining other attributes.
- Hence this relation is in BCNF.

Hence the finalised relation is:

Dish_invoice (invoice_id, order_id, dish_id, qty)

5. Dish_cart (cart_id, user_id, dish_id, qty)

Cart_id	User_id	Dish_id	qty
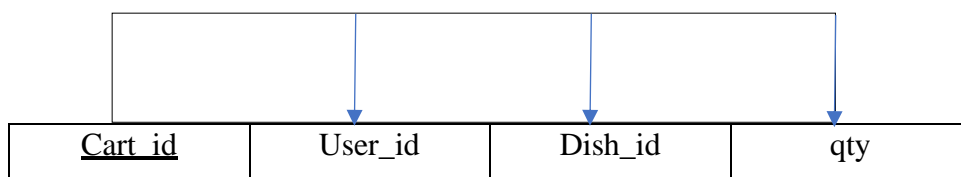
Foreign Keys: User_id, Dish_id || Primary Key: Cart_id || Candidate Key: None

Cart_id	User_id	Dish_id	qty
1	1	3	5
2	1	4	3
3	2	3	6

- An User_id can have multiple rows containing the corresponding dish ids. Hence User_id cannot be a candidate key.
- Here there can be multiple times same dish_id added to cart by different users. Hence Dish_id which is only foreign key cannot be a candidate key.
- All the attributes in this relation can be determined by Cart_id which is a primary as well as the candidate key.

Functional Dependencies:

{Cart_id}→ {_User_id, Dish_id, qty }

First Normal Form

- In this Relation, No Multivalued Attribute is present. Hence this relation is in first normal form.

Second Normal Form:

- This Relation is in First Normal Form.
- There is no partial dependency present in this relation, ie only Cart_id determines all attributes.
- Hence this relation is in Second Normal Form.

Third Normal Form:

- This Relation is in Second Normal Form.
- There is no transitive dependency present in this relation.
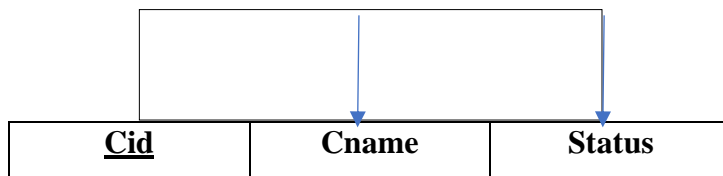- Hence this relation is in Third Normal Form.

Boyce Codd Normal Form:

- This Relation is in Third Normal Form
- All the dependents are determined only by the candidate key. No other attribute other than Cart_id is determining other attributes.
- Hence this relation is in BCNF.

Hence the finalised relation is:

Dish_cart (cart_id, user_id, dish_id, qty)

6. Email_verification (user_id, user_email, email_verify, rand_str)

User_id	User_email	Email_verify	Rand_str
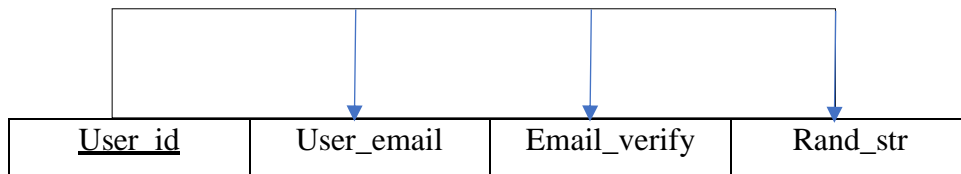
Foreign Keys: User_id || Primary Key: User_id || Candidate Key: user_id

Functional Dependencies:

{ User_id } → { User_email, Email_verify, Rand_str }

First Normal Form

- In this Relation, No Multivalued Attribute is present. Hence this relation is in first normal form.

Second Normal Form:

- This Relation is in First Normal Form.
- There is no partial dependency present in this relation, ie only User_id determines all attributes.
- Hence this relation is in Second Normal Form.

Third Normal Form:

- This Relation is in Second Normal Form.
- There is no transitive dependency present in this relation.
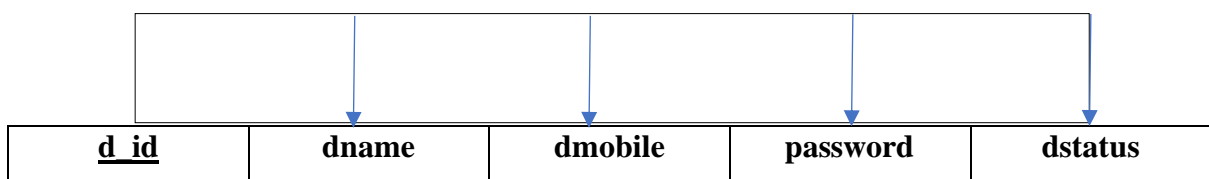- Hence this relation is in Third Normal Form.

Boyce Codd Normal Form:

- This Relation is in Third Normal Form
- All the dependents are determined only by the candidate key. No other attribute other than User_id is determining other attributes.
- Hence this relation is in BCNF.

Hence the finalised relation is:

Email_verification (user_id, user_email, email_verify, rand_str)

7. Order_status (os_id, order_status)
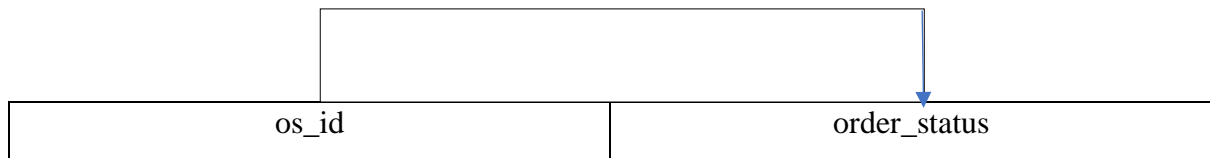
os_id	order_status

Foreign Keys: None || Primary Key: os_id || Candidate Key: os_id

Functional Dependencies:

{os_id}→ {order_status }

First Normal Form

- In this Relation, No Multivalued Attribute is present. Hence this relation is in first normal form.

Second Normal Form:

- This Relation is in First Normal Form.
- There is no partial dependency present in this relation, ie only os_id determines all attributes.
- Hence this relation is in Second Normal Form.

Third Normal Form:

- This Relation is in Second Normal Form.
- There is no transitive dependency present in this relation.
- Hence this relation is in Third Normal Form.

Boyce Codd Normal Form:

- This Relation is in Third Normal Form
- All the dependents are determined only by the candidate key. No other attribute other than os_id is determining other attributes.
- Hence this relation is in BCNF.

Hence the finalised relation is:

Order_status (os_id, order_status)

8. User (<u>user_id</u>, user_name, user_email, user_mobile, password)

user_id	user_name	user_email	user_mobile	password

Foreign Keys: None || Primary Key: user_id || Candidate Key: user_id

Functional Dependencies:

{<u>user_id</u>}→ {user_name, user_email, user_mobile, password }

First Normal Form

- In this Relation, No Multivalued Attribute is present. Hence this relation is in first normal form.

Second Normal Form:

- This Relation is in First Normal Form.
- There is no partial dependency present in this relation, ie only <u>user_id</u> determines all attributes.
- Hence this relation is in Second Normal Form.

Third Normal Form:

- This Relation is in Second Normal Form.
- There is no transitive dependency present in this relation.
- Hence this relation is in Third Normal Form.

Boyce Codd Normal Form:

- This Relation is in Third Normal Form
- All the dependents are determined only by the candidate key. No other attribute other than <u>user_id</u> is determining other attributes.
- Hence this relation is in BCNF.

Hence the finalised relation is:

User (<u>user_id</u>, user_name, user_email, user_mobile, password)

9. Payment (payment id, order id, payment type, payment status)

order_id	payment id	payment type	payment status
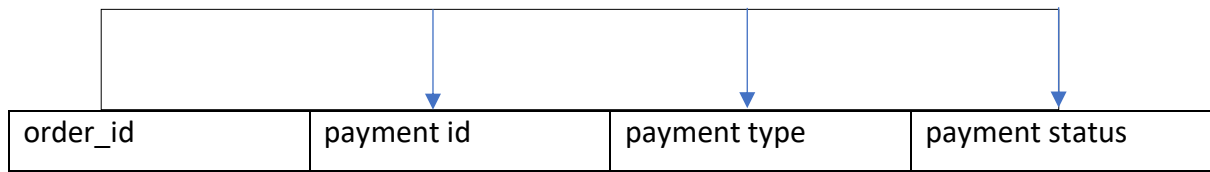
Foreign Keys: order_id || Primary Key: None || Candidate Key: order_id || Unique Key: payment id

- Here payment id is not the candidate key because, for payments involving Cash on Delivery there is no need of payment id. Hence it will remain Null.
- This implies that payment id is a underline unique key.
- Here Order_id which is the foreign key referencing order_master will be candidate key.

Functional Dependencies:

{order_id} → {payment id, payment type, payment status }

First Normal Form

- In this Relation, No Multivalued Attribute is present. Hence this relation is in first normal form.

Second Normal Form:

- This Relation is in First Normal Form.
- There is no partial dependency present in this relation, ie only order_id determines all attributes.
- Hence this relation is in Second Normal Form.

Third Normal Form:

- This Relation is in Second Normal Form.
- There is no transitive dependency present in this relation.
- Hence this relation is in Third Normal Form.

Boyce Codd Normal Form:

- This Relation is in Third Normal Form

- All the dependents are determined only by the candidate key. No other attribute other than order_id is determining other attributes.
- Hence this relation is in BCNF.

Hence the finalised relation is:

Payment (payment id, order id, payment type, payment status)

10. Category (Cid,Cname,Status)

Cid	Cname	Status

Foreign Keys:None || Primary Key: C_id || Candidate Key: C_id

Functional Dependencies:

{ Cid }→ { Cname, Status }

First Normal Form
- In this Relation, No Multivalued Attribute is present. Hence this relation is in first normal form.

Second Normal Form:
- This Relation is in First Normal Form.
- There is no partial dependency present in this relation, ie only **Cid** determines all attributes.
- Hence this relation is in Second Normal Form.

Third Normal Form:
- This Relation is in Second Normal Form.
- There is no transitive dependency present in this relation.
- Hence this relation is in Third Normal Form.

Boyce Codd Normal Form:

- This Relation is in Third Normal Form
- All the dependents are determined only by the candidate key. No other attribute other than **Cid** is determining other attributes.
- Hence this relation is in BCNF.

Hence the finalised relation is:

Category (**Cid**, Cname,Status)

11. Delivery_boy (**d_id**,dname,dmobile,password,dstatus)

d_id	dname	dmobile	password	dstatus

Foreign Keys: None || Primary Key: d_id || Candidate Key: d_id

Functional Dependencies:

{ d_id } → { dname,dmobile,password,dstatus}

First Normal Form

- In this Relation, No Multivalued Attribute is present. Hence this relation is in first normal form.

Second Normal Form:

- This Relation is in First Normal Form.
- There is no partial dependency present in this relation, ie only d_id determines all attributes.
- Hence this relation is in Second Normal Form.

Third Normal Form:

- This Relation is in Second Normal Form.
- There is no transitive dependency present in this relation.
- Hence this relation is in Third Normal Form.

Boyce Codd Normal Form:

- This Relation is in Third Normal Form
- All the dependents are determined only by the candidate key. No other attribute other than d_id is determining other attributes.
- Hence this relation is in BCNF.

Hence the finalised relation is:

Delivery_boy (d_id,dname,dmobile,password,dstatus)

6. Methodology with Modules

6.1. Methodology

The following methods/measures were adapted to achieve the goal of the project:

1. The primary step was to identify the deliverables of the project i.e, the project modules.

2. The next step was to identify the scope of the project viz., target area and target users.
3. Next step was to design an appropriate database which is comprehensive to maintain the functionalities of the restaurant based on the project scope and project deliverables.

4. Correspondingly, a front end which is relevant to the database design and user-friendly was designed.

5. Appropriate backend codes for each front-end page was developed and integrated.

6. All the frontend and backend were integrated and tested for any errors or rectifications.

7. Additional features (or) optimization methods were added (or) performed respectively based on what was observed while testing.

8. The final step was to test the whole full-stack application after final modifications.

6.2 Modules

Module 1: Creating User Login

- User Login
  To order foods from the website customer should register to the site. After registration customer can login to the website by entering User-ID and password.

Module 2: Developing the Home page (Food categories and related details)

- Food categories and details
  Customer will be able to view all the available foods categorised under many categories and their related details.
- Add to cart
  This module will add the food to the shopping cart

Module 3: Developing the Order Page (Cart summary)

- Order placement
  This module includes the reviewing of the cart which contains the food items and will enable the user to proceed and confirm the order

Module 4: Developing the Transaction Page (Order Confirmation)

- Payment
  This module enables the user to select between Online payment and Cash on Delivery for the payment.

Module 5: Developing the Order Invoice Page

- Order invoice
  This module will generate a summary of the order placed and will provide an option to download as a PDF file.

Module 6: Creating Admin Login

- Admin Login
  To manage the website, the admin should register to the site. After registration admin can login to the website by entering User-ID and password.

Module 7: Developing the Admin Dashboard Page (Management of the website)

- Dashboard
  This module is used for the maintenance and management of the website by the admin. This module consists of information about the products and services. This includes sub-modules such as **dish master, category master, order master and delivery person integration.**

- Dish Master

  This module contains information about particular dish, such as dish number, dish name, category, images of dishes and description.
- Category Master

  This module will store and maintain the details of the categories of the dishes which could be differentiated so that it is easier to search for the dish.
- Order Master

  This Module contains information about all the confirmed orders such as order id, customer details, quantity, total price, payment and order status, delivery boy status etc. This also enables the admin to update the order status.
- Delivery Person Integration

  This module will enable the delivery person to take up an order and update its status.

Module 8: Email Integration and Payment Gateway Integration

- This Module will enable the functionality of mailing the invoice details (report) to the user. Also, the online payment functionality is enabled using stripe payment gateway.

Module 9: Developing the editing page (for editing user details) and Forgot Password

- This module enables the user to modify their details and login credentials. Most importantly, when user forgets their password, a new temporary password will be mailed to their registered mail-id.

Module 10: Testing and Improving the website

7. System Specifications

7.1 Software Requirements

- Software: XAMPP server (Cross Platform)

- Operating System: Any (Platform Independent)

- Frontend: HTML, CSS, Boot Strap

- Backend: PHP, JavaScript(jQuery)

- Database: MySQL (Via PHPMyAdmin)

7.2 Hardware Requirements

- Processor: Intel Pentium III or equivalent or new

- Disk Space: 70MB or more

- Note: The web application is compatible to run in a smartphone ecosystem as well.

8. Results

Welcome Page (8.1)

User Login Page (8.2)

User Registration page (8.3)

Landing Page – Home page of the website (8.4)

Checkout Page (8.5)

Online Payment Page (8.6)

Order Confirmation Page (8.7)

Email of the invoice sent to the user after successful order placement (8.8)

Admin Login Page (8.9)

Admin Dashboard Page with sidebar for navigation (8.10)

Dish management page (8.11)

 (Same design followed for all the other options except order in menu bar)

(8.12)

Order Management Page (8.13)

Delivery Person Login Page (8.14)

Delivery Management Page (8.15)

Email after the order is successfully delivered (8.16)

Forgot Password Page (8.17)

Email of the newly generated password (8.18)

User Credentials Modification Page (8.19)

About Us Page (8.20)

9. Conclusion:

This project is a humble venture to satisfy the needs to manage the food ordering in online. Several user-friendly coding has also been adopted. A rigid efficient database with the lowest possible redundancy is built. The backend code is comprehensive enough to cover all the corner test cases as well. The main aim of the project to build a comprehensive website to cater the needs of the restaurant digitally has been successfully achieved. This project is capable enough to replace the dependency on third party vendors, provided the restaurant have their own workforce or manpower. This project might not certainly be called "perfect" for which it has its own advantages and limitations.

9.1 Advantages:

- Works easy and faster than current applications.

- Easily-maintainable online record of the data.

- Reduction of manual works

- Create, Retrieve, Update, Delete operations are performed seamlessly.

- Any cancellation of orders can be easily monitored.

- Keeps the record of food items, its sales and the balance of the stock

- Recreates the offline functionality of restaurant in a digital world.

9.2 Limitations:

- Application won't be able to send some notification via SMS.

- Application requires PHP, MySQL and three different system should be connected to one network for using one application to its full functionality.

- User won't able to change the skin or any functionality of application.

- User can get the reports in some formats only.

10. References

[1] Is Swiggy Charging Higher Commissions From Restaurants? Charging Extra Delivery Fees From Customers? (trak.in)

[2] Hotel association writes to Swiggy, Zomato to cut commission charges on takeaway services - The Financial Express

[3] Swiggy explains why its online food price might be higher than restaurant rates (livemint.com)