A
Project Report
On

Online Food Ordering System

B. TECH-Sem VI

Prepared By: -

IT122-Shaikh Shahid S.

**DEPARTMENT OF INFORMATION
TECHNOLOGY FACULTY OF TECHNOLOGY,
DHARMSINH DESAI UNIVERSITY
COLLAGE ROAD, NADIAD - 387001**

April,2021

A
Project Report
On

Online Food Ordering System

B. TECH-Sem VI

In Partial fulfillment of requirements for
Bachelor of Technology
in
Information Technology

Submitted By:
Shaikh Shahid S .

Under the guidance of
Prof. Deepak C. Vegda

DEPARTMENT OF INFORMATION TECHNOLOGY
FACULTY OF TECHNOLOGY, DHARMSINH DESAI
UNIVERSITY COLLEGE ROAD, NADIAD-387001

CANDIDATE'S DECLARATION

We declare that the pre-final semester report entitled **"Online food ordering system"** is our own work conducted under the supervision of the guide **Prof. Deepak C. Vegda**. We further declare that to the best of our knowledge the report for B. Tech. VI semester does not contain part of the work which has been submitted either in this or any other university without proper citation.

Candidate's Name:- Shaikh Shahid S .
Student ID:- 18ITUOS088
Roll NO : IT122
Sem : 6

Submitted To: **Prof. Deepak C. Vegda**

Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Gujarat.

DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project carried out in the subject of Software Design

Projects, entitled "Online food ordering system" and recorded in this report is a bonafide report of work of

1)Name : Shaikh Shahid Roll No: IT122 Student ID No: 18ITUOS088

of the Department of Information Technology, semester VI.

They were involved in Project work during the academic year 2020 -2021.

Prof. Deepak C. Vegda
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:30 /03 / 2021

Prof. (Dr.) V. K. Dabhi,
Head , Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad.
Date:30 /03 / 2021

Acknowledgment

This project would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

It gives us an immense pleasure submitting this report towards the partial fulfillment of our academics. Success in any mission cannot be achieved single handedly. It is the team effort that sails the ship to shore.

We would like to express our sincere thanks to our Head of Department **Prof. (Dr.) Vipul K. Dabhi** and our project guide **Prof. Deepak C. Vegda,** who gave us an opportunity to undertake such a great challenging and innovative work, without whose help and encouragement, it would be infeasible for us to carry out the desired work. We are grateful to them for their guidance, encouragement, understanding and insightful support in the development process.

With sincere regards,
Shahid Shaikh

TABLE OF CONTENT

ABSTRACT

This system is designed to connect the restaurant and customer online through web application .
It contains three applications for three users such as admin , restaurant and customer according to their functionality.
So the main task of the whole system is when customers order some food then that food and all the information related to order will transfer to a particular restaurant which is chosen by the customer .

So this system contains a relational database and to interact with the database it uses the **mysql** language to perform **CRUD** operations on the database .

The role of **Admin** to authenticate a restaurant . After authentication, the admin has to register the restaurant by filling necessary details . After that, create the database for the restaurant .

After successfully registering the restaurant by admin. Now the **Restaurant** can login with credentials provided by the admin . Simply restaurant name , and password etc. then the restaurant can access their database . Now restaurants can add categories , delete categories , add food , delete food , active-deactive food , orders and all operations can perform .

The **Customer** first needs to create the account . After that, customers can see all the foods along with restaurant names . For place order customers need to add foods into cart and from cart customer needs to checkout and provide address and all necessary details in order to place food . after successfully placing order , order received by restaurant and display on the homepage of restaurants with necessary order details .

LIST OF FIGURES

LIST OF TABLES

Chapter 1:- Introduction

1.1 Project Overview:

Project Title: Online Food Ordering System

Here we propose an Online Food Ordering system supporting the needs of the current society. During this pandemic many of the restaurants need their own online food ordering and delivering service so this is useful for time and also for health . The system can be used in any food delivery industry. This simplifies the process of food ordering for both the customer and the restaurant as the entire process is automated and digitized .

1.2 Purpose

This system is particularly a multi vendor system . there is the service multiple food provider with delivery for their own restaurant . There are lots of functionality involved in this . and there might be some regular updates in the enhancement functionality as development grows .

1.3 Scope

Process start when a customer ordered something and then admin will receive all the order information with the who is ordered , where to deliver , payment and all other relevant information . and this order will be assigned to deliver to make delivery . This whole process is automated .
Technology will be used :

1.4 Technology and Literature Review

Front-End : HTML , CSS , JavaScript , BOOTSTRAP
Back-End : PHP
Database : MYSQL

Chapter 2:- Project Management

2.1 Feasibility Study

2.1.1 Technical Feasibility

A technical feasibility study evaluates the details of how you intend to build a system or solution. The technical feasibility means the study or evaluation of current equipment, existing software technology and the current knowledge that we are going to use to provide the solution or system. Technical analysis evaluates technical merits of the system at the same time collects additional information about performances, reliability, maintainability and productivity.

2.1.2 Time Schedule Feasibility

The project has simple working and the basic requirement can be satisfied within allotted time period so the time development feasibility is satisfied.

2.1.3 Operational Feasibility

The proposed system will increase the operational efficacy of all the users. Therefore, the throughput of the system and reduction of time is as desired. This system will ensure and satisfy the requirements identified by the requirement analysis phase system. Operational feasibility deals with the acceptance of the user and their willingness to use the system.

The web-application we are developing will interact with these main actors of our application.

The system and UI will be very user-friendly, i.e., it will be very useful and easy to operate
by any new user. We will be particular and will follow the below standards: Universal Design, Convenience, Safe and secured, distinct styling.

2.1.4 Implementation feasibility

After adding a front-end web application to our project, users can directly access the system and can implement directly from the web application. Implementation will result in all the important information about all the available professionals and customers can hire any of them for their personal activities/needs.

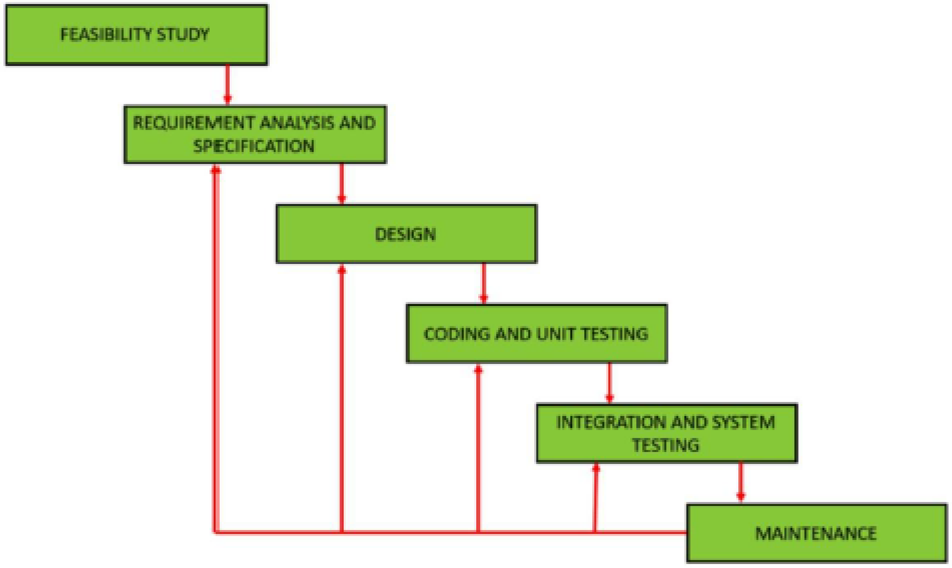
2.2 Project Planning

2.2.1 Project Development Approach and Justification

For project development, the Iterative waterfall model is used. It is a particular implementation of a software development life cycle that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete.

This model divides the cycle into the phases mentioned below:

1. Feasibility Study
2. Requirement analysis and specification
3. Design
4. Coding and unit testing
5. Integration and system testing
6. Maintenance


FEASIBILITY STUDY
REQUIREMENT ANALYSIS AND SPECIFICATION
DESIGN
CODING AND UNIT TESTING
INTEGRATION AND SYSTEM TESTING
MAINTENANCE

2.3 Project Scheduling

Scheduling the project tasks is an important project planning activity. It involves deciding which task should be taken up and when. In order to schedule the project activities; a software project manager needs to do the following:

1. Identify all the tasks needed to complete the project.
2. Break down large tasks into smaller activities.
3. Determine dependencies among different activities
4. Establish most likely estimates for the time duration necessary to complete the activities.

Chapter3:- System Requirement Study

3.1 User Characteristics

There are four types of users dealing with this system:

1. Customer (end Users)
2. Restaurant
3. Admin

3.2 Hardware and Software Requirements

Computer

- RAM 4 GB , Storage 512 SSD
- Processor i5 8th generation

Software

- xampp
- visual studio code
- chrome browser

3.3 Constraints

- Minimum response time can’t be satisfied all time.
- Servers incapable of carrying high traffic bursts.
- It will work only on a web based system.
- There are no other systems that use this application as an interface.

3.4 Assumptions and Dependencies

The user must have connected to the internet to use the system and must have a stable network configuration in order to have a successful payment transaction. The users must have Window 7/8/10 installed on system and the browser should be Google chrome. The Service person will be assumed to take four services per day. The accuracy of the information of the users is the responsibility of all users. It is also assumed that all the clients running this software are not blocked by Firewalls, proxies etc.

Chapter 4:- System Analysis

4.1 System Requirements (SRS Document)

1) Core Functional Requirements.

R.1 User management :

R.1.1 Registration / signup :-
⇒ I/p : user has to provide user name , email-id , mobile number , set password .
⇒ O/p : user will redirect to home page .
⇒ desc : email verification required .

R.1.2 Login :-
⇒ I/p : user has to provide email-id and password .
⇒ O/p : user will redirect to home page .
⇒ desc : email id and password must match .

R.1.3 sorting food category :-
⇒ I/p : select the categories of food .
⇒ O/p : it will show all foods belonging to those categories .
⇒ desc : it is helpful to order the food .

R.1.4 Add to the cart :-
⇒ I/p : user will add the food on cart .
⇒ O/p : all the food chosen by the user will be added to the cart .
⇒ desc : for placing the order the user must need to add the food in the cart.

R.2 Cart Management :

R.2.1 delete the foods on cart :

⇒ I/p remove the foods added on cart .

⇒ O/p : food removed successfully .

⇒ desc :

R.2.2 add more foods on cart :

⇒ I/p : add more foods

⇒ O/p : user redirect to home page .

⇒ desc : before checkout if user want to add more foods .

R.2.3 Checkout cart :

⇒ I/p : click the checkout button .

⇒ O/p : redirect to checkout page .

⇒ desc : confirms the food to order .

R.3 Checkout :

R.3.1 address info :

⇒ I/p : provide the address where food will be deliver .

⇒ O/p : food will delivered on this address .

⇒ desc : fill the correct address .

R.3.2 coupon code :

⇒ I/p : choose the coupon code to apply .

⇒ O/p : order price will reduce according to coupon code .

⇒ desc : coupon code must be valid .

R.3.2 place order :

⇒ I/p : click on the place the order .

⇒ O/p : user will redirect to the payment page .

⇒ desc : provided information in the checkout page must be correct .

R.4 Payment :

R.4.1 make a payment :
⇒ I/p : select the appropriate interface for payment .
⇒ O/p : order will confirm after payment .
⇒ desc : order will confirm if and only if payment done or cash on delivery(COD) otherwise status is pending .

R.5 Food Management by admin :

R.5.1 Add food :
⇒ I/p : provide food name , detail , category , image , price etc .
⇒ O/p : all information will be stored in database .
⇒ desc : information stored if and if further information provided .

R.5.2 food availability :
⇒ I/p : update the status of food available or not .
⇒ O/p : food status will show on home page .
⇒ desc : this is helpful to user that food is available at that time when they order .

R.5.3 Remove food :
⇒ I/p : select the food to remove .
⇒ O/p : food and related information remove successfully from the whole database .

R.5.4 Update food details :
⇒ I/p : provide updated information .
⇒ O/p : information updated successfully on all modules .
⇒ desc : using this functionality you can update the food price , details Etc.

R.6 Order management by restaurant :

R.6.1 Confirm order :

⇒ I/p : order requested by user and waiting for confirmation .
⇒ O/p : send the confirmation message with order id .
⇒ desc : new orders arrives and order id generated for all orders to identify uniquely .

R.6.2 Pending order :

⇒ I/p : payment not completed by user .
⇒ O/p : show that your order in pending state .
⇒ desc : when payment is done then order will change the state pending to confirmed .

R.7 Customer support :

R.7.1 Contact us :

⇒ I/p : provide the mobile no . , email_add , and issue in text area .
⇒ O/p : our executive will contact customers .
⇒ desc : if customer need some help about food quality or hygiene it can complain via this module .

2) Non-Functional Requirements

⇒ **Security :**

● Login requirements : access level , CRUD levels

● Password requirements : length , special characters , expiry , encrypted .

⇒ **Performans :**

● Response time : Application loading , screen open , refresh time .

● Processing time : functions , calculation , import , export etc .

⇒ **Recovery :**

● recovery process : what is the process of recovery .

● recovery time : how quickly to recover .

● back-up frequencies : how often the transaction data , set-up , system code backup .

⇒ **Reliability :**

● Recovery time : if system failure occurs , how much time it will take to recover and up again .

⇒ **Consistency :**

● Databases must preserve the consistency whatever operations occurs .

⇒ **Compatibility :**

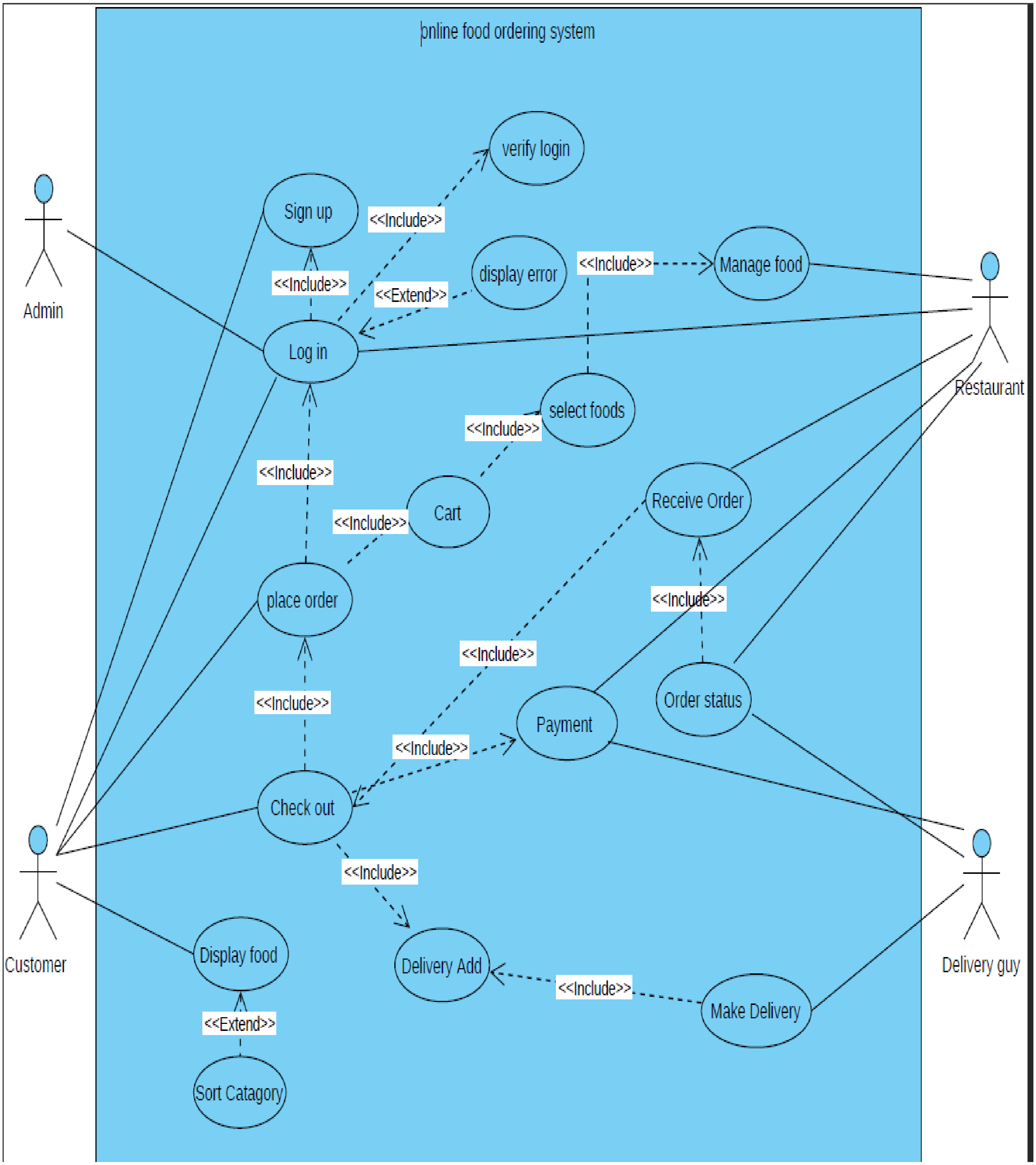
● Compatible in different browsers and environments .

● Compatibility with shared applications .

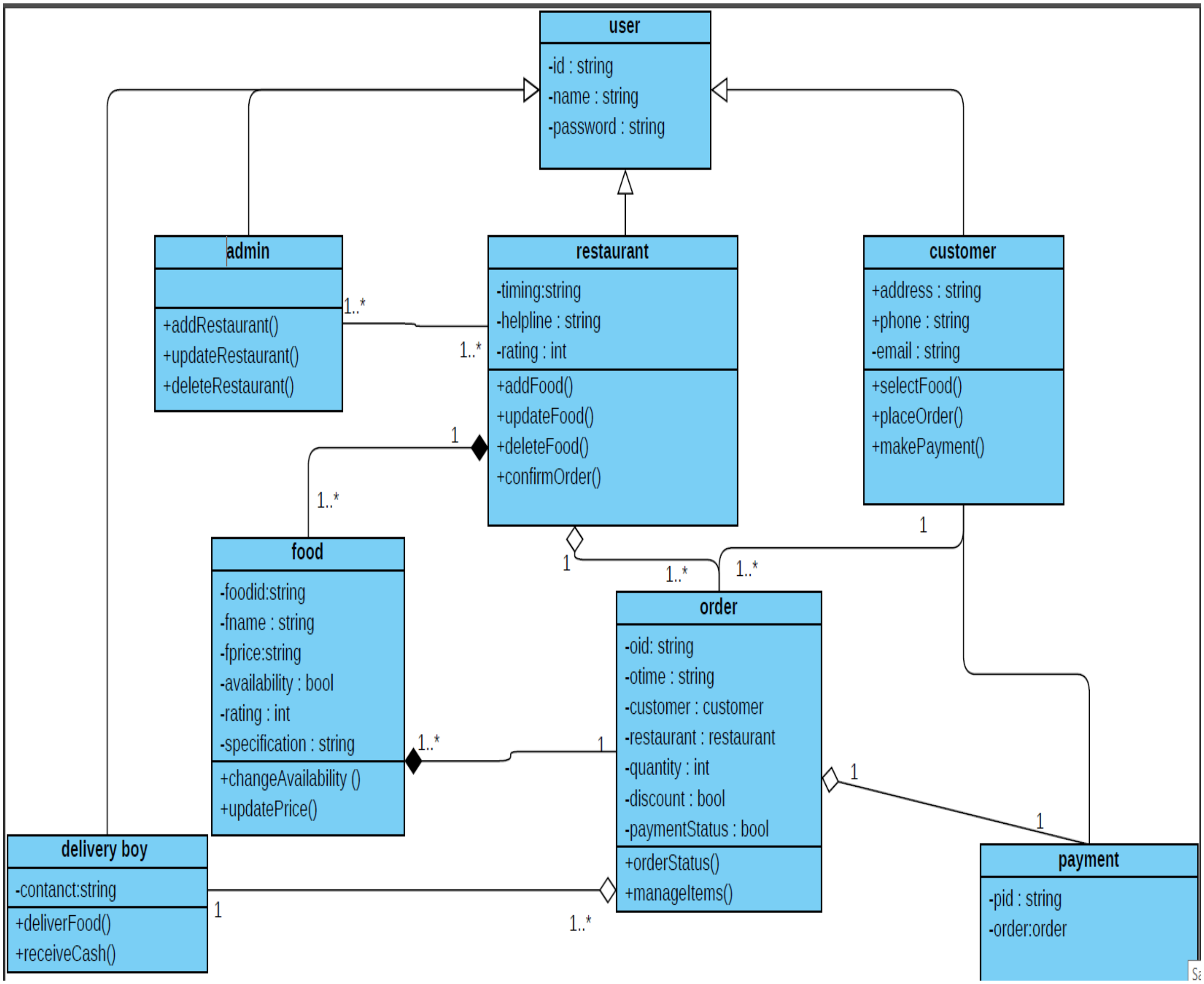
● Compatibility on different platforms .

Chapter 5:- System Design

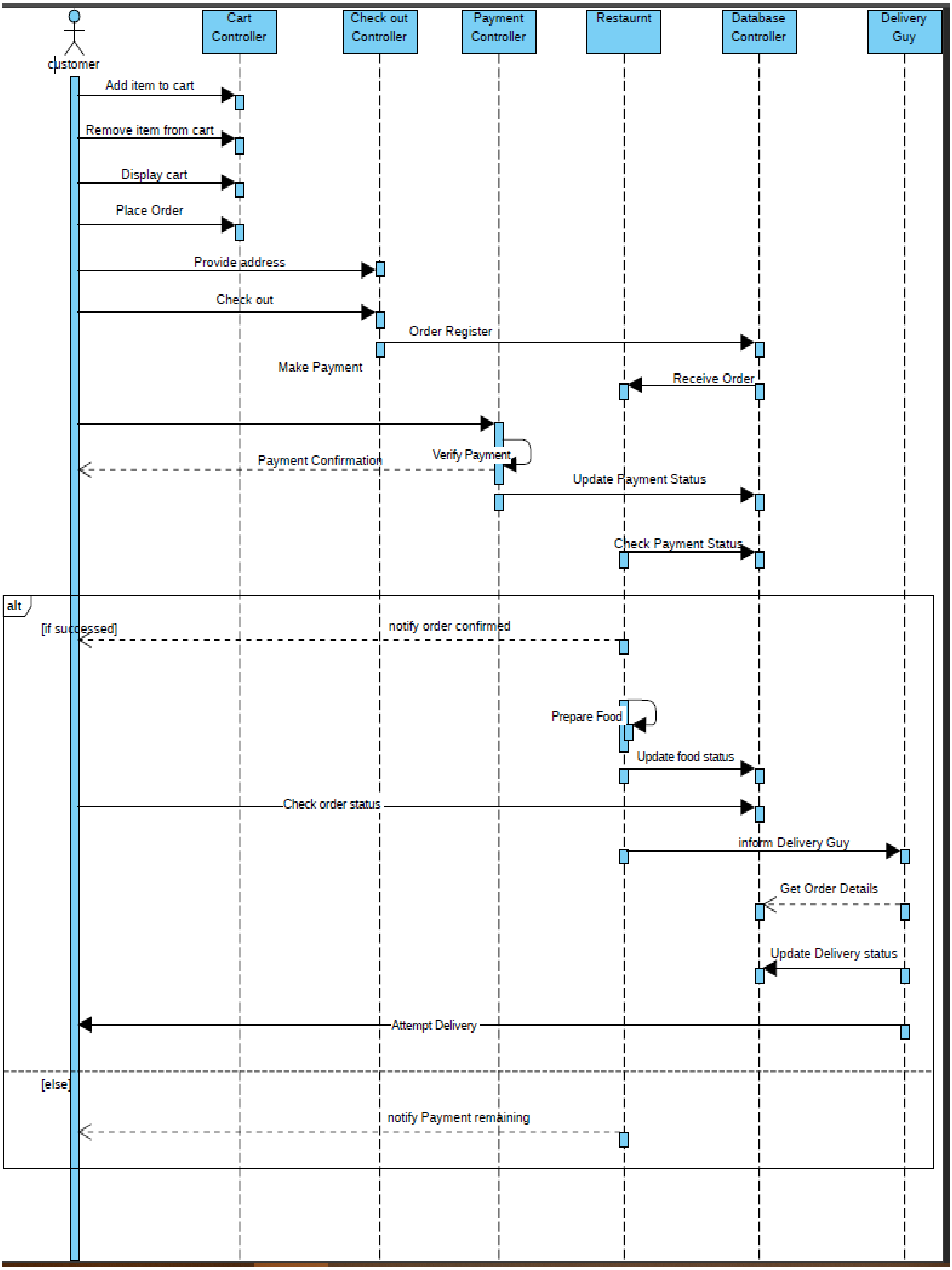
5.1 Use-case diagram :



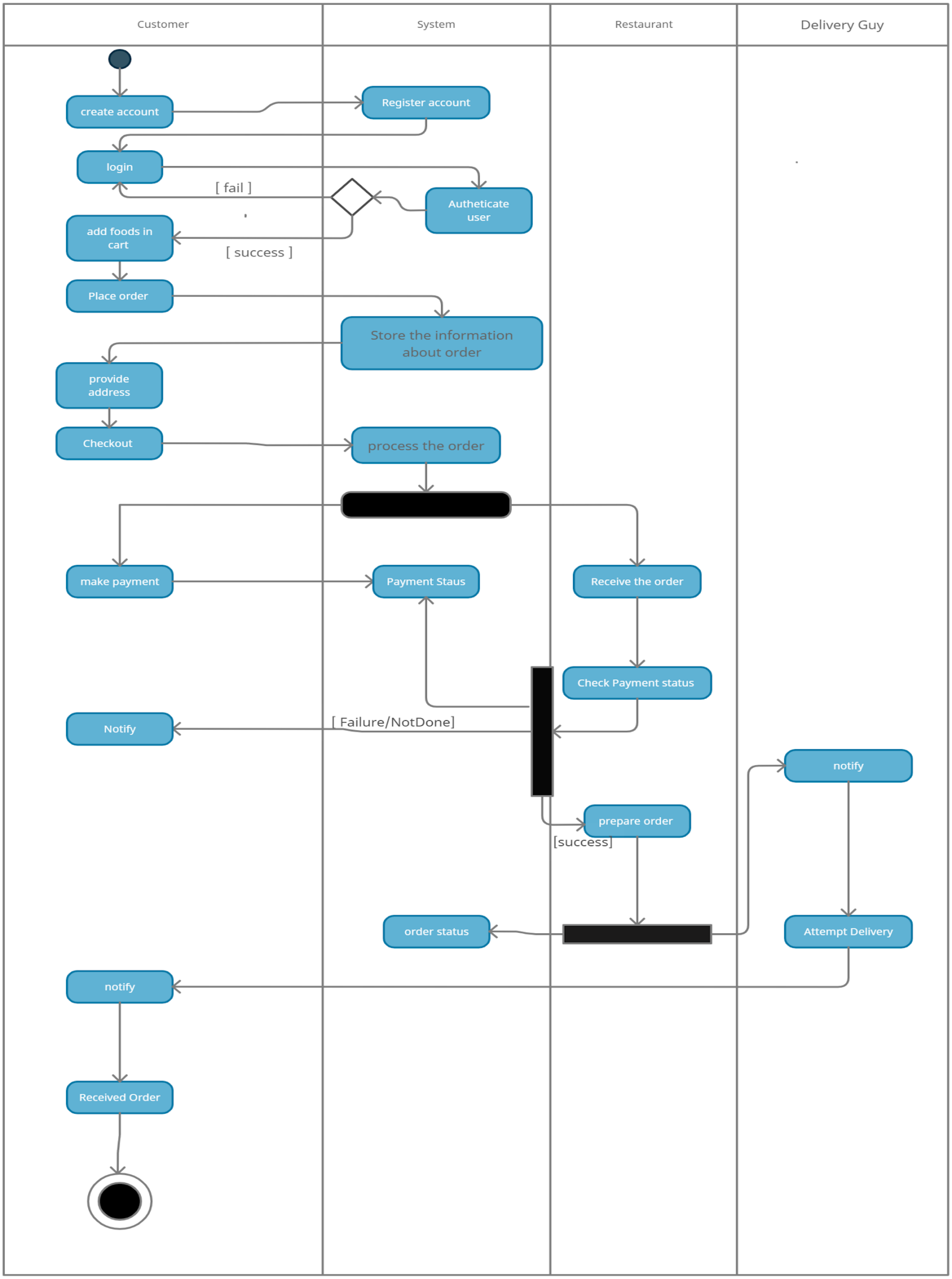
5.2 Class Diagram :-



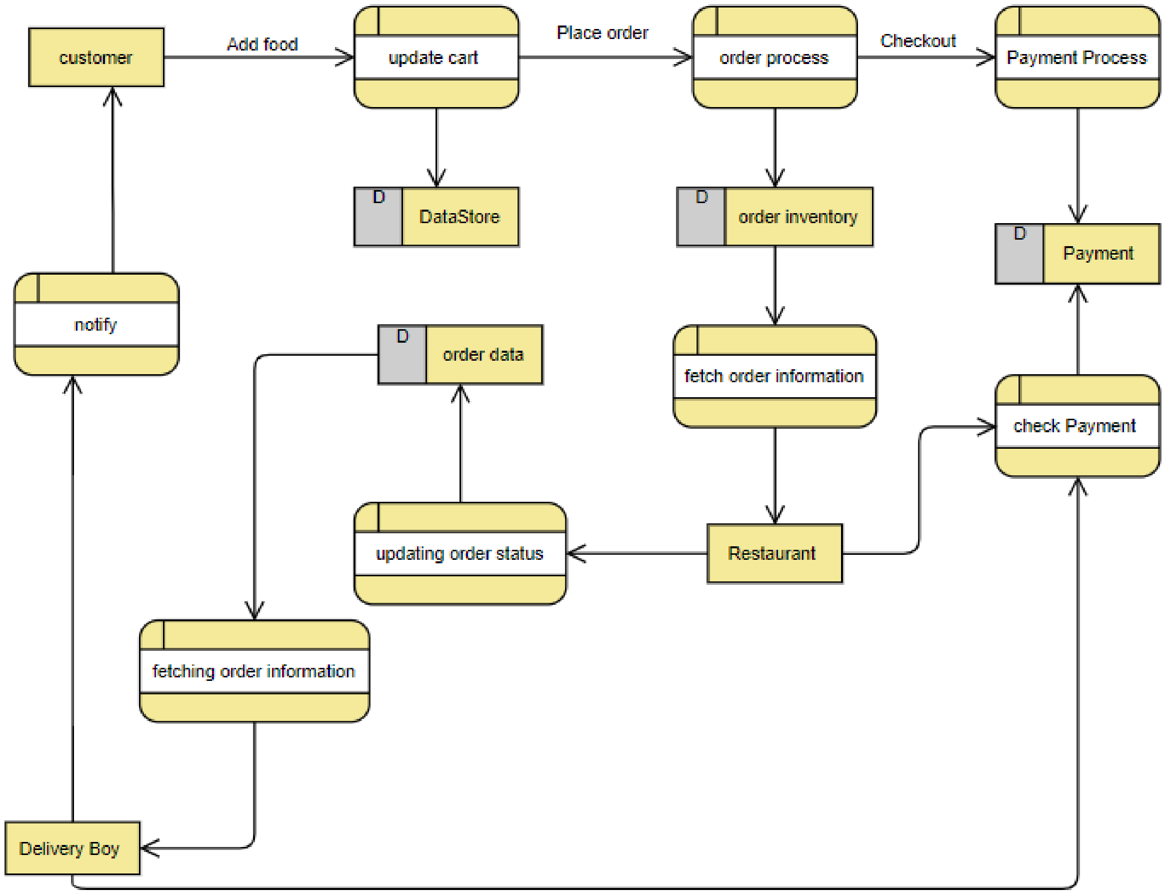
5.3 Sequence Diagram:-



5.4 Activity Diagram:-



5.5 DataFlow Diagram :-



Chapter 6 : IMPLEMENTATION PLANNING

6.1 Implementation Environment

For Implementation of this web-application we have used:

1. Visual Studio Code
2. Sublime Text

We have used HTML5 , CSS , JAVASCRIPT , BOOTSTRAP4 ,and CSS frameworks for implementation of Frontend and PHP For Backend, and we have used Mysql as database in our web-application.

6.2 Program/Modules Specification

Admin : can add restaurant , update restaurant , delete restaurant .

Restaurant : add food , category .
delete food , category .
active , deactivate food .

Customer : order food .

6.3 Coding Standards

FRONTEND : HTML & CSS &
JAVASCRIPT & BOOTSTRAP5
BACKEND : PHP

Chapter 7 : Testing Phase

7.1 Testing Plan

Software Testing:

Testing involves operation of a system or application under controlled conditions and evaluating the results. The controlled conditions should include both normal and abnormal conditions. Testing should intentionally attempt to make things go wrong to determine if things happen when they don’t happen when they should. It is oriented to
‘detection’.

Why Important:

No matter how good a programmer is, no application will ever be one hundred percent correct. Testing was important to us in order to ensure that the application works as efficient as possible and conforms to the needs of the system. Testing was carried out throughout the development of the application, not just the application has been developed, as at this stage it took a great deal of effort to fix any bugs or design problems that were occurred.

We Started our project **Campus connect system** with just a brief idea of the technologies that we are going to implement. So at first, we designed the login and the sign up page just as in front end development and later on when we created the database we started with the testing of the values. This states that we started our
testing phase with the addition of sign up entries in the database values.

7.2 Testing Strategy

When our application was configured and customized in the system, the test was observed that this configuration or customization does not cause any improper processing or violation. The following care was taken when the application was developed at the local machine. The interface may have something not proper, which can be tested by this checklist:

- Number of input parameter equal to number of arguments
- Parameter and argument attributes match
- Number of arguments transmitted to called forms equal to number of parameters
- Attributes of arguments transmitted to called forms to attributes of parameters
- Number of attributes and order of arguments to built-in functions correct
- The local data structures for a form are common source of errors. The following types of errors should be searched for,
- Improper or inconsistent typing
- Erroneous initialization or default values
- Incorrect places name
- Underflow, overflow and addressing exception
- As far as unit testing is concerned; we did it at the time of coding in an informal but extensive way, so as to reduce number of problems arising out of incorrect syntax, incorrect variable, function names etc.
- Close the database connection when not required.
- Care was taken to check for any infinite loop that exists in code before executing the code.

7.3 Testing Methods

While Box Testing

Also known as glass box, structural, clear box and open box testing. A software testing technique whereby explicit knowledge of the internal workings of the item being tested are used to select the test data. Unlike black box testing, white box testing uses specific knowledge of programming code to examine outputs. The test is
accurate only if the tester knows what the program is supposed to do; it means that he
must be completely aware that for particular input a particular output must be obtained. The main benefit of this type of testing is Tester can see if the program diverges from its intended goal. This test concentrates on the examination of the code rather than the specification. We have included three different forms of white box testing.

Working process of white box testing:

Input: Requirements, Functional specifications, design documents, source code. Processing: Performing risk analysis for guiding through the entire process.

Proper test planning: Designing test cases so as to cover entire code. Execute rinse- repeat until error-free software is reached. Also, the results are communicated.

Output: Preparing final report of the entire testing process.

White-box testing's basic procedures require the tester to have an in-depth knowledge of the source code being tested. The programmer must have a deep understanding of the application to know what kinds of test cases to create so that every visible path is exercised for testing. Once the source code is understood then the source code can be analyzed for test cases to be created.

Statement Coverage Criterion:

This is the simplest coverage criterion. We are checking in it that each statement of the program was executed “at least once”.

Branch Coverage Criterion:

An improvement over statement is **Branch Coverage**. In that we are running a series
Of test to ensure that all branches are tested at least once.

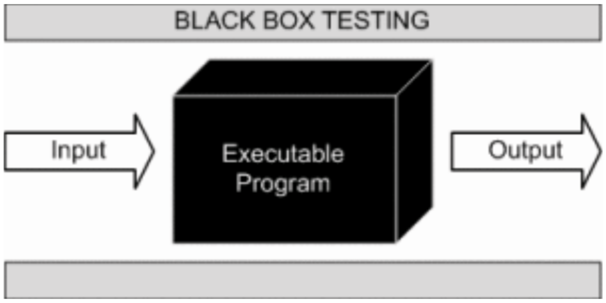
Path Coverage Criterion:

There are many errors which were not detected by statement or branch testing. The reason is that some errors are related to some combination of branches and it may not check in other test. We are checking in this test if all paths of programs are executed or not.

Black Box Testing

Black-box and white-box are test design methods. Black-box test design treats the system as a "black-box", so it doesn't explicitly use knowledge of the internal structure. Black-box test design is usually described as focusing on testing functional requirements. Also known as behavioral, functional, opaque-box and closed-box. Black Box Testing was helpful us to find error such as:

1. Interface error
2. Incorrect or missing functions.
3. Errors in data structures or external database access.
4. Performance Errors.



Black box test design technique: Procedure to derive and/or select test cases based on an analysis of the specification, either functional or non-functional, of a component or system without reference to its internal structure.
Black Box Testing method is applicable to the following levels of software

testing: System Testing

Integration Testin

Acceptance Testing

The higher the level, and hence the bigger and more complex the box, the more black- box testing method comes into use.

Unit Testing

Unit testing is a method of testing the correctness of a particular module of source code. The idea is to write test cases for every non-trivial function or method in the module so that each test case is separate from the others if possible. The developers mostly do this type of testing. In this method of testing we test all individual components to ensure that they operate correctly. Each component is tested independently without other system components.

Integration Testing

It is the phase of software testing in which individual software modules are combined and tested as a group. It follows unit testing and precedes system testing. The purpose of Integration testing is to verify functional, performance and reliability requirements placed on major design items.
It takes as its input modules that have been checked out by unit testing, groups them in larger aggregates, applies tests defined in an Integration test plan to those aggregates and delivers as its output the integrated system ready for system testing.

Interface Testing

The implemented system is user friendly and does not require any extra knowledge for using the application and hence it passes the interface testing

7.4 Test Cases :

Sr no	Purpose	input	State	Expected output	Actual output	Test result
1	Admin login	Username , password , submit	Logout	Login successful	Login successful	Pass
2	Add restaurant	Restaurant reg number,name, address,service timing , helpline number	Admin logged in	Restaurant database created successfully . also register successfully .	Restaurant database created successfully . also register successfully .	Pass
3	Modify restaurant	Restaurant id	Admin logged In	Fetch all detail of restaurant for this id so admin can modify.	Fetch all detail of restaurant for this id so admin can modify .	Pass
4	Delete restaurant	Restaurant name	Admin logged In	Delete database successfully	Delete database successfully	Pass
5	Check feedbacks	Click on feedback	Admin logged In	Display all feedbacks	Display all feedbacks	Pass
6	Set password for restaurant	Enter password , reenter password	Admin logged In	Password is set	Password is set by admin	Pass
8	Restaurant login	Restaurant name , password ,submit	Login page appear	Restaurant login successful.	Restaurant login successful.	Pass
9	Add category	Enter category id , category name , register category	Restaurant logged in	Category added successfully .	Category added successfully .	pass
10	Delete category	Clicking on delete button	Restaurant logged	Category will deleted and page refresh	Category will deleted and page refresh	Pass

			in			
11	Add foods	food id , name , select category , specification , price , rating , upload image .	Restaurant logged in	Food registered successfully .	Food registered successfully .	pass
12	Activate food	Click on active button	Restaurant logged in , food is disable	Food activated	Food activated	Pass
13	Delete food	Click on disable button	Food is active	Food is disable .	Food is disable .	Pass
14	Check orders	-	Restaurant logged in	Display all orders on home page	Display all orders on home page	Pass
15	Customer registration	Name , mobile no. , email , password	Login page , new user	Customer registration successfully	Customer registration successfully	Pass
16	Customer registration	Name , mobile , already registered email , password	Login page , new user	Fail to register , error : email id already registered.	Fail to register , error : email id already registered.	Pass
17	Customer login	User name , password	Login page	Customer logged in	Customer logged in	Pass

18	Select the quantity of wanted food	Select from drop down menu	Home page of customer logged in	Quantity selected	Quantity selected	Pass
19	Add food into the cart	Click on cart symbol for particular food	Custom logged in	Food added successfully into the cart	Food added successfully into the cart	Pass
20	Checkout	Click on checkout button fro m cart	Customer logged in	Redirect to checkout page .	Redirect to checkout page .	Pass
21	Place order	Provide address details and review the order price and all details	Customer logged in and checkout page	Order has been successfully placed	Order has been successfully placed .	Pass
22	Logout	Click on logout button	Logged in customer / restaurant / admin	Customer / admin / restaurant logout successfully .	Customer / admin / restaurant logout successfully .	Pass

8.0 USER MANUAL

→ you can check user manual here by following below link :-

https://drive.google.com/file/d/1E2gOzm-L1CKielr1ExD8QjoRp-bn-9u1/view?usp=sharing

9.0 Limitation and Future Enhancement

9.1 Limitation :

1) The location fetching api does not provide a 100% accurate location .

2) The admin has to share the password of the restaurant while creating a database for the restaurant.

9.2 Future enhancement :

1) integrating delivery system in the current system .

2) payment gateway for online payment

3) verifying email through confirmation mail .

4) notification about order status to customers .

5) real time tracking of order status .

6) introduce search functionality on customers home page .

10 Conclusion and Discussion

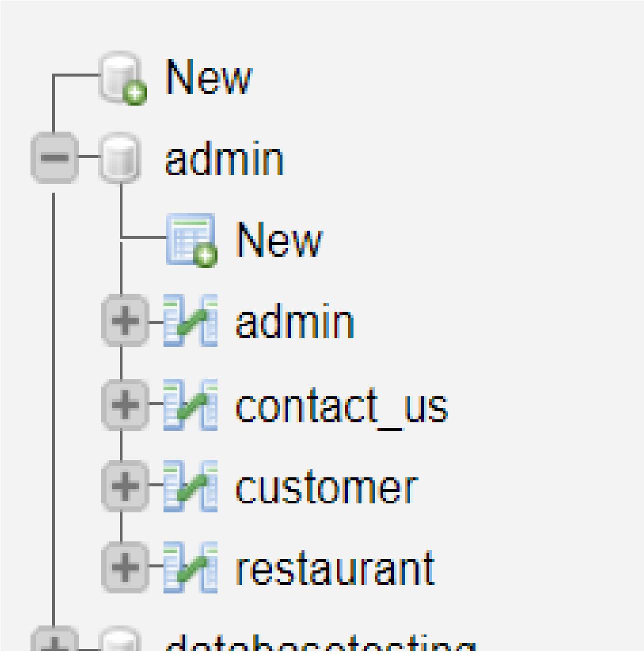
→ From the project we made these are the following things which can be concluded:

1) There is a lot of scope for enhancements, as there is a vast sea of knowledge waiting to be understood .

2) i am more focused on the implementation of basic functionalities from registration to place order . Now I need to more focus on optimization of basic functionality as well as system .

11 .DATABASE SCHEMA

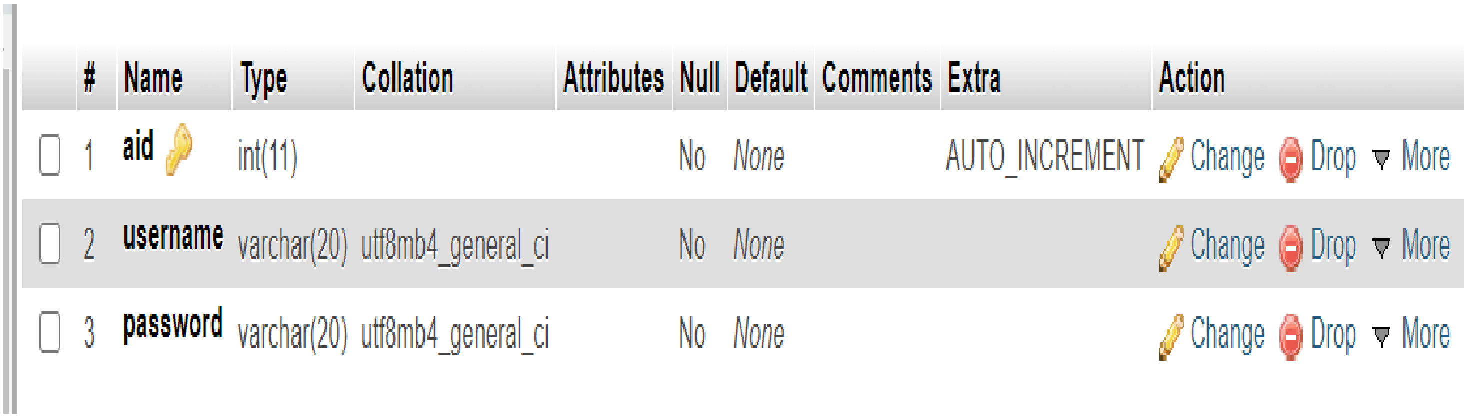
→ Admin database :



- New
- admin
  - New
  - admin
  - contact_us
  - customer
  - restaurant
- databasetesting

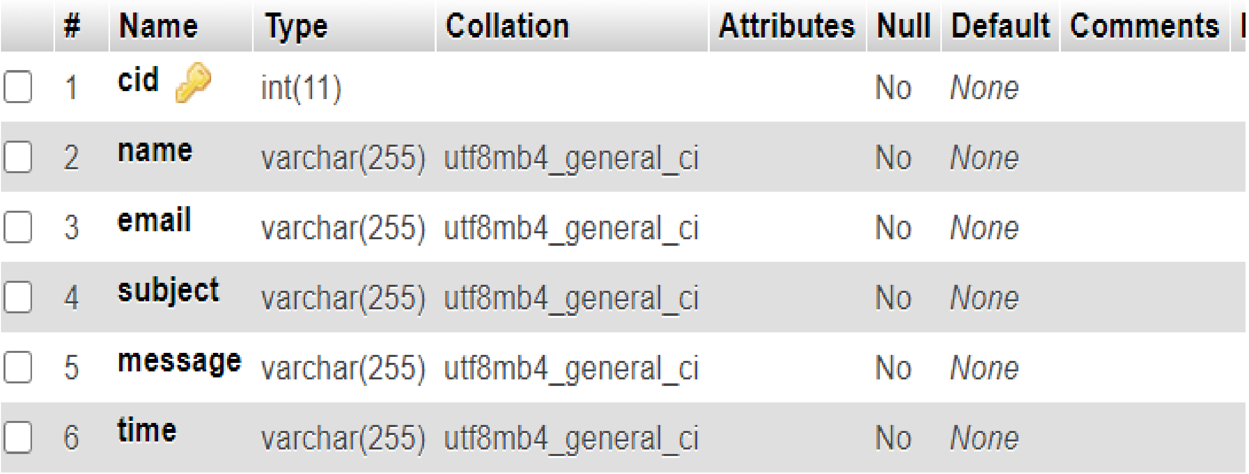
→ Table structure in admin database :

→ admin :-



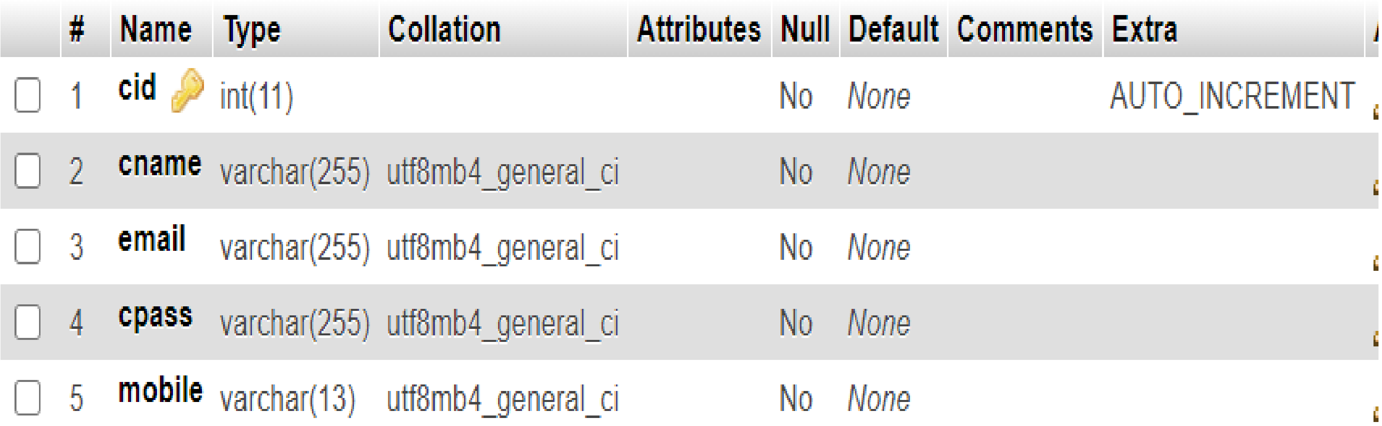
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**aid**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop ▾ More
□	2	**username**	varchar(20)	utf8mb4_general_ci		No	*None*			Change Drop ▾ More
□	3	**password**	varchar(20)	utf8mb4_general_ci		No	*None*			Change Drop ▾ More

→ contact_us :-



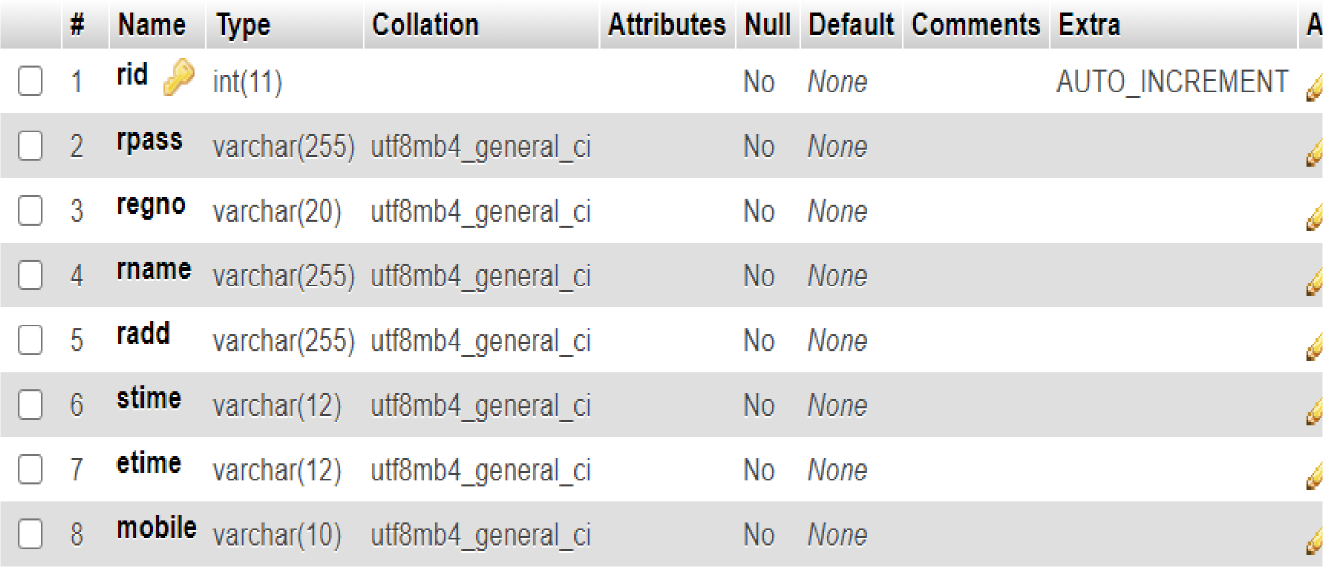
	#	Name	Type	Collation	Attributes	Null	Default	Comments
☐	1	**cid** 🔑	int(11)			No	*None*	
☐	2	**name**	varchar(255)	utf8mb4_general_ci		No	*None*	
☐	3	**email**	varchar(255)	utf8mb4_general_ci		No	*None*	
☐	4	**subject**	varchar(255)	utf8mb4_general_ci		No	*None*	
☐	5	**message**	varchar(255)	utf8mb4_general_ci		No	*None*	
☐	6	**time**	varchar(255)	utf8mb4_general_ci		No	*None*	

→ customer :-



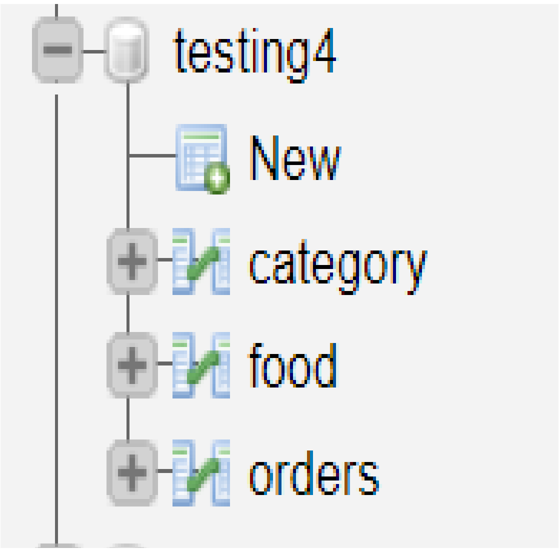
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	**cid** 🔑	int(11)			No	*None*		AUTO_INCREMENT
☐	2	**cname**	varchar(255)	utf8mb4_general_ci		No	*None*		
☐	3	**email**	varchar(255)	utf8mb4_general_ci		No	*None*		
☐	4	**cpass**	varchar(255)	utf8mb4_general_ci		No	*None*		
☐	5	**mobile**	varchar(13)	utf8mb4_general_ci		No	*None*		

→ restaurant :-



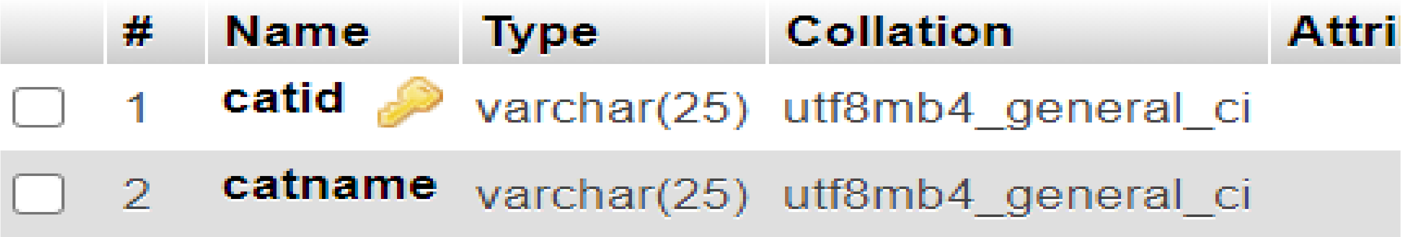
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	A
☐	1	rid 🔑	int(11)			No	*None*		AUTO_INCREMENT	✎
☐	2	rpass	varchar(255)	utf8mb4_general_ci		No	*None*			✎
☐	3	regno	varchar(20)	utf8mb4_general_ci		No	*None*			✎
☐	4	rname	varchar(255)	utf8mb4_general_ci		No	*None*			✎
☐	5	radd	varchar(255)	utf8mb4_general_ci		No	*None*			✎
☐	6	stime	varchar(12)	utf8mb4_general_ci		No	*None*			✎
☐	7	etime	varchar(12)	utf8mb4_general_ci		No	*None*			✎
☐	8	mobile	varchar(10)	utf8mb4_general_ci		No	*None*			✎

→ Restaurant database :



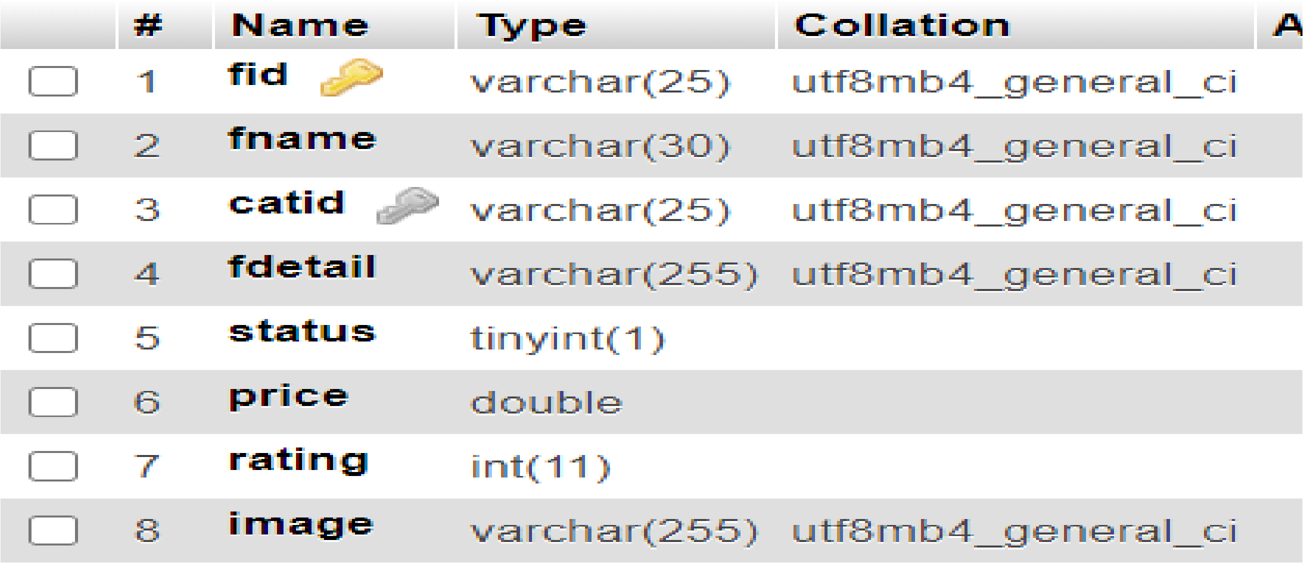
testing4
- New
- category
- food
- orders

→ category table :-



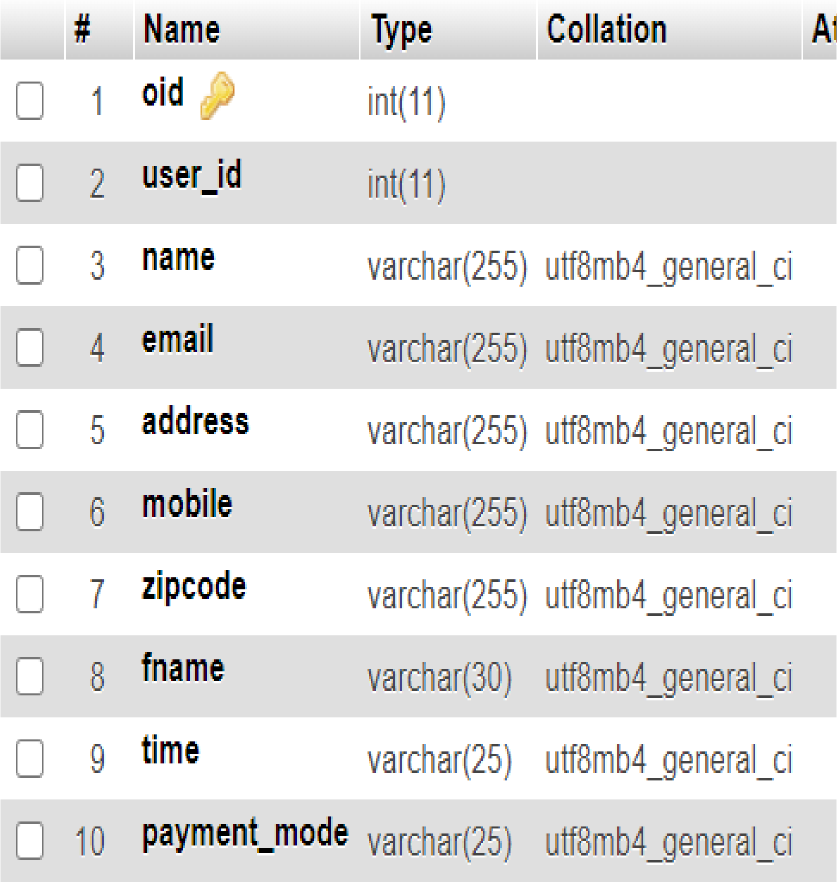
	#	Name	Type	Collation	Attri
□	1	catid	varchar(25)	utf8mb4_general_ci	
□	2	catname	varchar(25)	utf8mb4_general_ci	

→ food table :-



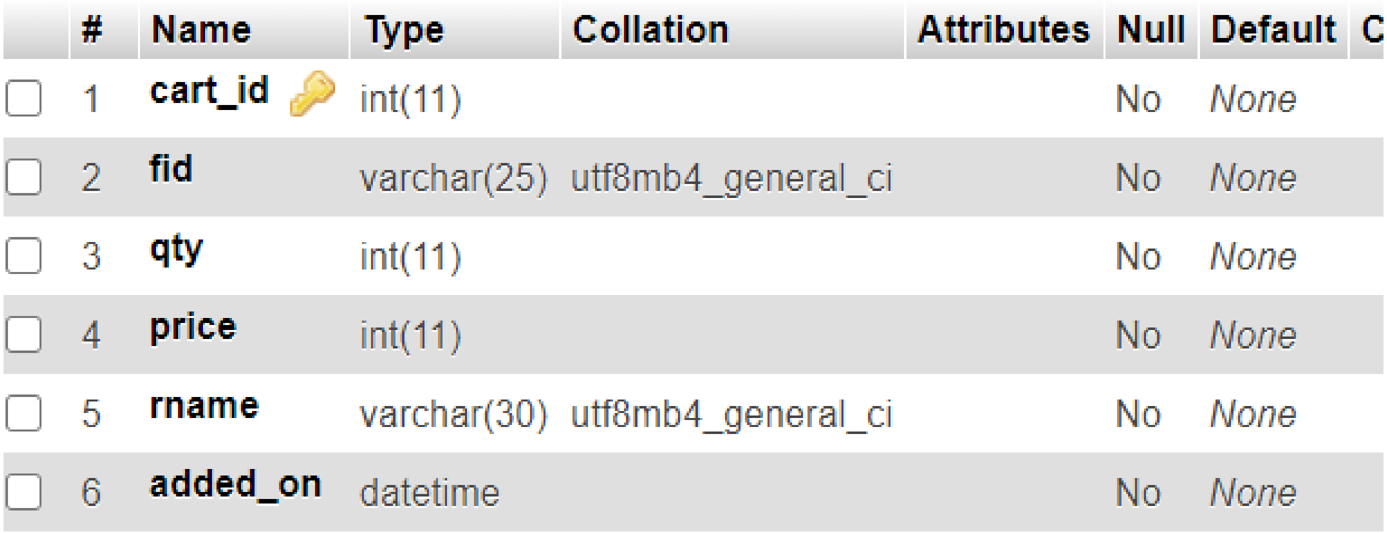
	#	Name	Type	Collation	A
□	1	fid 🔑	varchar(25)	utf8mb4_general_ci	
□	2	fname	varchar(30)	utf8mb4_general_ci	
□	3	catid 🔑	varchar(25)	utf8mb4_general_ci	
□	4	fdetail	varchar(255)	utf8mb4_general_ci	
□	5	status	tinyint(1)		
□	6	price	double		
□	7	rating	int(11)		
□	8	image	varchar(255)	utf8mb4_general_ci	

→ orders table :-



	#	Name	Type	Collation	At
□	1	oid 🔑	int(11)		
□	2	user_id	int(11)		
□	3	name	varchar(255)	utf8mb4_general_ci	
□	4	email	varchar(255)	utf8mb4_general_ci	
□	5	address	varchar(255)	utf8mb4_general_ci	
□	6	mobile	varchar(255)	utf8mb4_general_ci	
□	7	zipcode	varchar(255)	utf8mb4_general_ci	
□	8	fname	varchar(30)	utf8mb4_general_ci	
□	9	time	varchar(25)	utf8mb4_general_ci	
□	10	payment_mode	varchar(25)	utf8mb4_general_ci	

→ customer cart table :-



	#	Name	Type	Collation	Attributes	Null	Default	C
□	1	**cart_id** 🔑	int(11)			No	*None*	
□	2	**fid**	varchar(25)	utf8mb4_general_ci		No	*None*	
□	3	**qty**	int(11)			No	*None*	
□	4	**price**	int(11)			No	*None*	
□	5	**rname**	varchar(30)	utf8mb4_general_ci		No	*None*	
□	6	**added_on**	datetime			No	*None*	

12 . References

→ YouTube

→ Stack Overflow - Where Developers Learn, Share, & Build Careers

→ W3Schools PHP Tutorial

→ XAMPP Installers and Downloads for Apache Friends

→ Learn MySQL Tutorial - javatpoint