

MCA – 170
Project Report on
Restaurant Management System

BY

Virender Yadav

50311404420

Under Guidance

Of

MR. KAUSHAL MEHTHA

(HOD)

Submitted to the BPIBS In partial fulfillment of the
requirements For the Major Project

Master Of Computer Applications
Guru Gobind Singh Indraprastha University
Dwarka Sector – 16C , 110078

CERTIFICATE OF ORIGINALITY

This is certify that the project report entitled **Restaurant Management System** Submitted to **BHAI PARMANAND INSTITUTE OF BUSINESS STUDIES (Affiliated to GGSIPU, Dwarka, New Delhi – 110078”)** in partial fulfillment of the requirement for the award of the degree of **MASTER OF COMPUTER APPLICATION(MCA)**, is an original work carried out by **Virender Yadav Enrolment No.: 50311404420** respectively under the guidance of **MR. KAUSHAL MEHTA (HOD)**.

The matter embodied in this project is a genuine work done by the student and has not been submitted whether to this University or to any other University/ Institute for the fulfillment of the requirement of any course of study.

(.MR. KAUSHAL MEHTA)
(HOD)

CANDIDATE DECLARATION

We hereby certify that the work which is presented in the project entitled **Restaurant Management System**, in partial fulfillment of the requirements for the award of the degree of **MASTER OF Computer Application (MCA)**, with specialization in Computer Science, submitted to **BHAI PARMANAND INSTITUTE OF BUSINESS STUDIES (“Affiliated to GGSIPU, Dwarka, New Delhi – 110078”)**, is an authentic record of original work carried out from August 2019 under the supervision and guidance of **MR. KAUSHAL MEHTA (ASSISTANT PROFESSOR)**.

we have not submitted the matter embodied in this project for the award of any other degree.

Date:
Place: New Delhi

(Virender Yadav)
50311404420

ACKNOWLEDGEMENT

It is our pleasant duty to thank all those who have helped us in completing this project. we take the opportunity to express our deep sense of indebtedness and the sincerest gratitude of our guides **MR. KAUSHAL MEHTA** having extended their ever encouraging and affable guidance and constant encouragement throughout the progress of the project.

They have displayed unique tolerance and understanding at every step of progress and encouraged me incessantly. I deem it my privilege to have carried out the project work under their able guidance.

we would like to thank my all friends for their co-operation and support.

(Virender Yadav)

50311404420

CONTENT

Topic	Page No.
1. INTRODUCTION	6
2. OBJECTIVE	7
3. SOFTWARE REQUIREMENTS SPECIFICATION	8
4. PLATFORM USED	11
5. SYSTEM ANALYSIS	12
6. IDENTIFICATION OF THE NEED	14
7. PRELIMINARY INVESTIGATION.....	15
8. FEASIBILITY STUDY	17
(A) TECHNICAL FEASIBILITY	18
(B) OPERATIONAL FEASIBILITY	19
(C) ECONOMICAL FEASIBILITY	20
9. DESIGN PHASE	21
(A) WATERFALL MODEL	
(B)MODULES-	24
(C) DATA FLOW DIAGRAM	28
(D) ER-DIAGRAM	32
(E) DATA STRUCTURE	36
10. SOURCE CODE	44
11. OUTPUT SCREEN	61
11. TESTING	68
12. MAINTENANCE	71
15. FUTURE SCOPE	72
16. CONCLUSION	73
17. BIBLIOGRAPHY	74

INTRODUCTION

The project titled as “Restaurant management system” is basically a type of application System. Its aim is to Collect Data of customer and their transection with our restaurant. This system is accessible by the restaurant Staff & Management. The “Restaurant Management System” is well organized program, developed for smooth function of the restaurant. This system can take orders from customer, give them their total bill after calculating g.s.t, It display menu and provide staff duty schedule. It also has customer point system which provides points (each point is equal to 1 rupees) to customer against their purchases. The most important part of this project is authorization, authentication and privacy. It provides user id and password which ensures security of the system. Only authorized user can access the system.

OBJECTIVE

The main objectives of this project are as follows:

- The main motive of this restaurant Management project is to provide some basic facilities to management so that they can records data about customer transection & also maintains details of staff.
- This system provides the user with the facilities such as Billing system, payment system, calculate customer points, display menu items and also staff duty schedule.
- The proposed System also provides a whole organization details in one way and less time.
- This system provides greater efficiency and better services to all the users.
- It has user friendly and interactive nature.
- Information retrieval is faster and accurate.

SOFTWARE REQUIREMENT SPECIFICATION

The Software Requirements Specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional description, a representation of system behavior, an indication of performance requirement and design constraints, appropriate validation criteria, and other information pertinent to requirement.

- The Introduction to software requirements specification states the goals and objectives of the software, describing it in the context of the Computer based system.
- The Information Description provides a detailed description of the problem that the software must solve. Information content, flow, and structure are documented.
- A Description of each function required to solve the problem is presented in the Functional Description.
- Validation Criteria's probably the most important and, ironically, the most often neglected section of the Software requirement Specification.

This project is based on three tier architecture. The three tier architecture where the application is divided into three logical constituents. And the three logical constituents are as follows:

1. User Services – Provide services such as user interface. (PYTHON/Tkinter).
2. Data Services – Provide handling and validation of data.

Disadvantages of the two tier architecture :

1. The Server will bear comparatively extra load.
2. It increases the network traffic.
3. Difficult to implement incremental improvements.
4. Applications are bound to the data source.

PROJECT CATEGORY

The undergoing project falls under Internet Technologies & XAMP Server or RDBMS (Relational Database Management System) category. Since the project is mainly responsible for creation of the RESTAURANT MANAGEMENT SYSTEM with the database at backend. As we know that, the Internet is huge client server architecture. The client is the web browser; it is requesting a web based data, a file, or whatever, from some computer somewhere – anywhere – in the world. The server is that computer that holds the information you want.

PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3. Due to concern about the amount of code written for Python 2, support for Python 2.7 (the last release in the 2.x series) was extended to 2020. Language developer Guido van Rossum shouldered sole responsibility for the project until July 2018 but now shares his leadership as a member of a five-person steering council.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends,[2] consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.[3][4] Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

PLATFORM USED

REQUIREMENTS AND ANALYSIS

Hardware Requirement

Processor	Intel® Core(TM) i3-6006U CPU @ 2.00GHz 2.00GHz
Mother Board	Intel 915
Monitor	CRT (1024 X 768) color
Hard Disk	1TB or More
Cache Memory	1 MB
PEN Drive	2 GB
Key Board	Any (QWERTY)
Mouse	ANY (Optical Mouse)
Printer	Office jet color
RAM	4 GB or More

Software Requirement

Operating System	Windows 10 Pro or More
Languages	PYTHON
Front End	PYTHON & Tkinter
Platform	Shell IDLE 3.7.4
Back End	XAMP, MySql Server
Browser Program	Google Crome/ Internet explorer/Mozilla Fireworks

SYSTEM ANALYSIS

Prior to implementing the computerization of any system, the existing system must be thoroughly being understood to determine, how the computer can be best used to make its operation most effective. This knowledge is generally acquired by analyzing the existing system. Hence, SYSTEM ANALYSIS is the process of diagnosing problems, gathering and interpreting facts to improve the system performance. In this Project System Analysis and Design refers to the process of examining a business situation with the intent of improving it through better procedures and methods. System Analysis is a process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvement to the system. In brief we can say Analysis specifies what system should do. System Analysis is a management technique, which helps us in designing a new system or improving an existing system.

In This Project, I deal with four basic elements of SYSTEM ANALYSIS in the following ways:

INPUTS:-

Once we know the output, we can easily determine what the inputs should be. Sometimes, it may happen that the required information may not be readily available in the proper form. This may be because of the existing forms are not properly designed. In my project, the Data entry operator must know all the required data such as Guest Name, Email, password, phone, etc., as the system will ask for this data when adding a record to the database.

The essential elements of inputs as far as this project is concerned are:

- ✓ Accuracy
- ✓ Timeliness
- ✓ Proper Format

✓ Economy

OUTPUTS:-

My aim in this project is to achieve through my goal and what the application intends to perform. In other words, I have to determine what the objectives or goals are, what do we intend to achieve, what is the purpose of our work, i.e. what is the main aim behind the system. Defining the aim is very vital in system work.

My project can perform all the basic operations of a Commodity Trading System like Creating of an Id, Customer information, Modification of Customers record, Different Reports etc.

DATABASE:-

The data entered by the user gets stored in the Database Tables which are register, booking, query and review should be taken that these tables might not get deleted accidentally as the records are stored and loaded from those tables only. No backup copies are maintained as a replacement for those tables. So in case of loss of those tables, all the records stored will be lost forever and there is no way to recover that data.

PROCESS:-

In this section of the Project, I have deal with the matter that how inputs are converted to outputs by the code. Basically, the data that has been feed to the system is stored in the form of tables and the text “End of File” has been appended after the last record so that we can find out the end of table when we need to read data from that table. Also, the coding section can be referred to find the details about the processes by which the input gets converted into the required outputs.

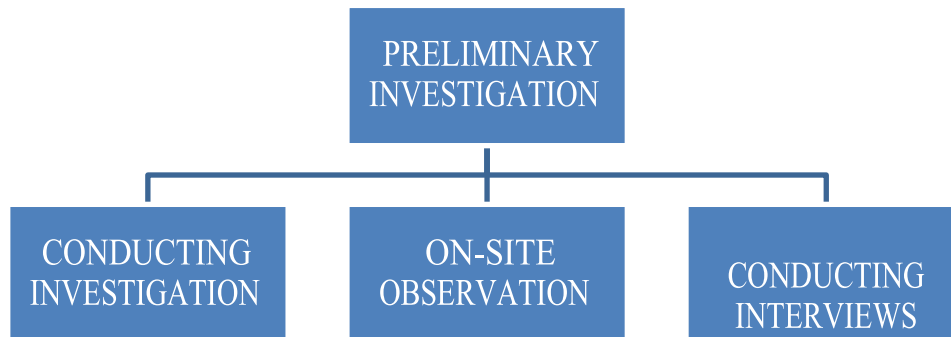
IDENTIFICATION OF THE NEED

The Identification of Need is one of the most difficult tasks of System Analysis i.e. developing a clear, in-depth understanding of the problem being investigated, without which it becomes impossible to specify the requirements for a new Project with accuracy. While Identifying the Needs for this Project I have kept the following things in my mind:

- ☐ The Performance of the System.
- ☐ The information being supplied and its form.
- ☐ The control of the information Processing.
- ☐ The efficiency of the existing System.
- ☐ The security of the Data and Software.
- ☐ The security of the equipment and personnel, etc.

After identification of Need, it is defined and a general direction or method for solving is also determined. Then Project boundaries are defined.

PRELIMINARY INVESTIGATION



The First Step in the System Development Life Cycle is the Preliminary Investigation to determine the feasibility of the system. The purpose of the preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the business system in all respect. Rather, it is the collecting of the information that helps committee members to evaluate the merits of the project request and make an informed judgment about the feasibility of the proposed project.

In my Project the following Objectives of the Preliminary investigation is accomplished:

- Clarify and understand the project request.
- Determine the size of the project.
- Assess costs and benefits of alternative approaches.
- Determine the technical and operational feasibility of alternative approaches.
- Report the findings to management; with recommendations outlining the acceptance or rejection of the proposal

CONDUCTING THE INVESTIGATION:-

The data that the analysts collect during preliminary investigations are gathered through three primary methods: reviewing organization documents, on-site observations and conducting interviews.

REVIEWING ORGANIZATION DOCUMENTS:-

The analyst conducting the investigation first learns about the organization involved in, or affected by the project. In RESTAURANT MANAGEMENT SYSTEM.

ON-SITE OBSERVATION:-

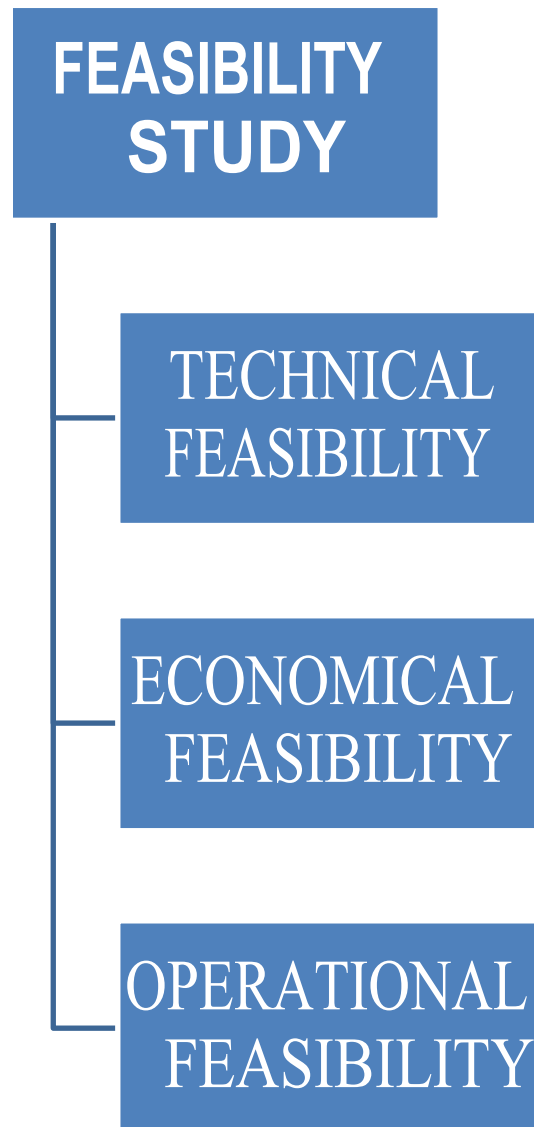
Another important technique to collect data in system. The purpose of the system observation is to get as close as possible to the real system being studied.

PROVIDING MATERIALS: -

Materials allows to sending product to Customer by the Supplier and request for submitting it. Interviews should provide details that further explain the project and show whether assistance is merited economically, operationally and technically. These were the Preliminary Investigations which I have made during the processing of this Project.

FEASIBILITY STUDY

Feasibility Study is one of the important outcomes of the preliminary investigation. The system is to be checked whether the system requested is feasible or not. In this Project I have conducted the three aspects of Feasibility Study.



TECHNICAL FEASIBILITY

The Technical Feasibility is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably, but might include:

- ☐ The facility to produce outputs in a given time.
- ☐ Response time under certain conditions.
- ☐ Ability to process a certain volume of transaction at a particular speed.
- ☐ Facility to communicate data to distant location.

In examining technical feasibility, configuration of the system is given more importance than the actual make of hardware. The configuration should give the complete picture about the system's requirements like how many workstations are required, how these units are interconnected so that they could operate and communicate smoothly. What speeds of input and output should be achieved at particular quality of printing. This can be used as a basis for the tender document against which dealers and manufacturers can later make their equipment bids. Specific hardware and software products can then be evaluated keeping in view with the logical needs

OPERATIONAL FEASIBILITY

It is mainly related to human organizational and political aspects. The points to be considered in this Project related to Operational Feasibility are as follows:

- ☐ What changes will be brought with the system?
- ☐ What organizational structures are distributed?
- ☐ What new skills will be required? Do existing staff members have these skills? If not, can they be trained in due course of time?

Generally project will not be rejected simply because of operational infeasibility but such considerations are likely to critically affect the nature and scope of the eventual recommendations. This feasibility study is carried out by a small group of people who are familiar with information system techniques, who understand the parts of the business that are relevant to the project and are skilled in system analysis and design process. As far as this project is concerned the changes which we have to be brought depends upon whether we are going to build a new project or we going to modify some establishment in the late one first of all we have make a network environment i.e establishment of a server is must. Then our focus goes towards workstations. Keeping in view of their hardware requirements like network interface card etc. Regarding this project, distribution of organizational structures is also essential because of security concerns, as there are different departments having their particular tasks I have already mentioned earlier like a system administrator should have the authentication to provide different access permission to its clients.

ECONOMICAL FEASIBILITY

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost/benefit analysis; the procedure is to determine the benefits and savings that expected from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

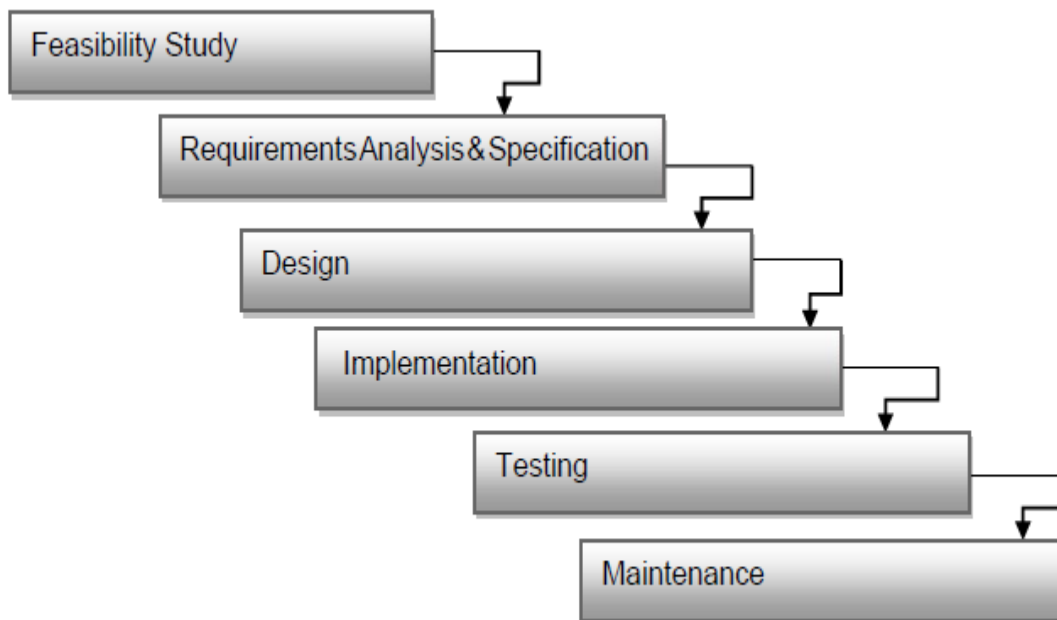
This feasibility also depends upon quality of staff hired and the proposed duration of time taken in this project sometimes it might be possible due to extension of time duration may fall the project under loss. The study of feasibility changes from phase to phase of the project development.

In this project although this feasibility study doesn't matter much in the case new setup of project because we start according to client specification but on the other hand if we have to modify over existing system we must take care of our existing resources and must analyze specially the working condition of hardware like server quality etc.

DESIGN PHASE

WATERFALL MODEL

It is the oldest and most widely used model in the field of software development.



Sequential Phases in the Waterfall Model

- **Requirements:**

The first phase involves understanding what needs to design and what is its function, purpose, etc. Here, the specifications of the input and output or the final product are studied and marked.

- **System Design:**

The requirement specifications from the first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The software code to be written in the next stage is created now.

- **Implementation:**

With inputs from system design, the system is first developed in small programs called units, which are integrated into the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

- **Integration and Testing:**

All the units developed in the implementation phase are integrated into a system after testing of each unit. The software designed, needs to go through constant software testing to find out if there are any flaw or errors. Testing is done so that the client does not face any problem during the installation of the software.

- **Deployment of System:**

Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

- **Maintenance:**

This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. The client is provided with regular maintenance and support for the developed software.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for the previous phase and it is signed off, so the name “Waterfall Model”.

ADVANTAGES OF WATERFALL MODEL

- The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.
- The waterfall model progresses through easily understandable and explainable phases and thus it is easy to use.
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- In this model, phases are processed and completed one at a time and they do not overlap. Waterfall model works well for smaller projects where requirements are very well understood.

Disadvantages of Waterfall Model

- It is difficult to estimate time and cost for each phase of the development process.
- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- Not a good model for complex and object-oriented projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.

MODULE

Modules & Their Description

1. SIGN_UP
 - SET_SDET
2. LOGIN
3. MAIN_PAGE
4. FORGOT_PASS
 - SET_PASS
5. BILLING_S
 - INS()
 - ADD
6. CMP
 - INSP
7. IT_DETAILS
8. STAFF_DET

1. SIGN UP PAGE
 - a) EMP_ID
 - b) EMP_NAME
 - c) EMP_PASSWORD
 - d) SHIFT
 - e) CONTACT_NO
 - f) Emp_address

2. LOGIN PAGE

- a) USER NAME
- b) PASSWORD
- c) LOGIN BUTTON
- d) SIGN UP BUTTON
- e) FORGOT PASSWORD BUTTON

3. FORGOT PASSWORD PAGE

- a) USER NAME
- b) PHONE NO.
- c) NEW PASSWORD
- d) REENTER PASSWORD
- e) SET BUTTON
- f) BACK BUTTON

4. MAIN PAGE

- a) CUSTOMER MEAL POINTS BUTTON
- b) BILLING SYSTEM BUTTON
- c) STAFF DETAIL BUTTON
- d) STAFF DETAIL BUTTON

5. CUSTOMER INFORMATION

- a) ORDER NO.

- b) CUSTOMER NAME
- c) PHONE NO.
- d) EMAIL
- e) CUSTOMER POINTS
- f) TOTAL POINTS
- g) SET BUTTON
- h) BACK BUTTON

6. BILLING SYSTEM

- a) LABELS
 - ITEMS
 - RATE
 - QUANTITY
 - AMOUNT
- b) FRENCH FRIES
- c) BURGER
- d) SANDWICH
- e) PASTA
- f) THALI
- g) COFFE
- h) TOTAL
- i) GST
- j) TOTAL AMOUNT
- k) CALCULATE

7. STAFF DETAILS

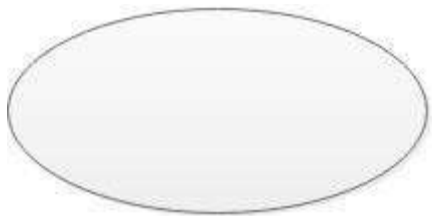
-It displays details of staff . fetched from database .

8. Menu page

-It displays all the menu items and similar as menu card serve to the Customers.

Data Flow Diagram

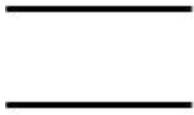
The **Data Flow Diagram** is the graphical representation that depicts information flow and the transforms that are applied as data move from input to output. DFD is a model, which gives the insight into the information domain and functional domain at the same time. DFD is refined into different levels. The more refined DFD is more details of the system are incorporated. In the process of creating a DFD, we decompose the system into different functional subsystems. The DFD refinement results in a corresponding refinement of data. After going through the current working process of the department, we can create the Data Flow Diagram (DFD). Following is the DFD of the “**RESTAURANT MANAGEMNET SYSTEM**”. We have refined the system up to two levels. Each break-up has been numbered as per the rule of DFD. We have tried to incorporate all the details of the system but there is some chance of further improvisation because of the study that is still going on for the project development.



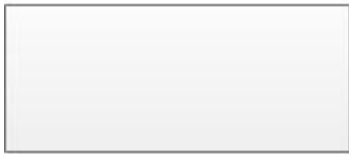
Process: Transforms of incoming data flow(s) to outgoing data flow(s).



Data Flow: Movement of data in the system.



Data Store: Data repositories for that are not moving



External Entity: Sources of destination outside the specified system boundary.



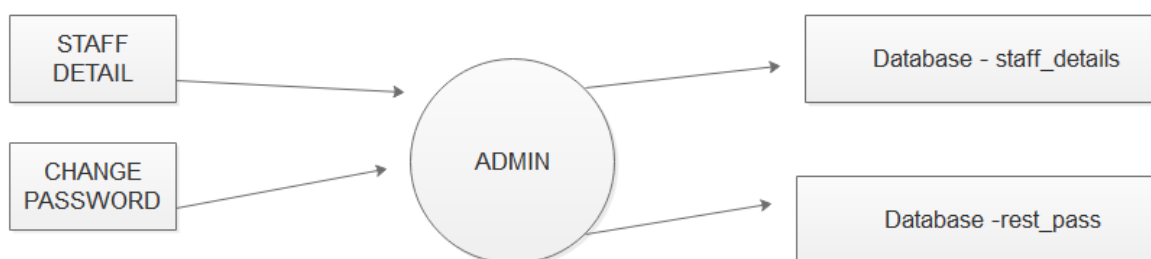
OR Gate: This symbol is work as OR Gate.

Data Flow Diagram of the project

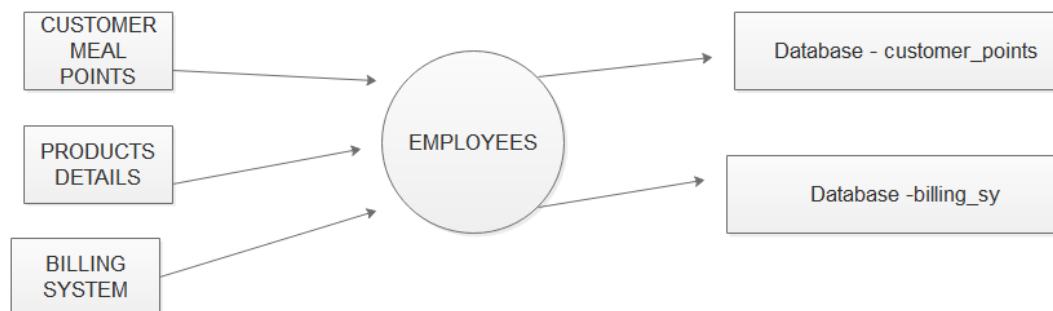
LEVEL-0



LEVEL-1 (ADMIN)



LEVEL 1 (Employee)



E-R DIAGRAM

- E-R diagram is a physical relationship between different entities.

It depicts the overall structure of an organization. E-R Model is a detailed Logical representation of entities. It is an association of data elements for an organization and business area. Basic symbols used for E-R diagram are:-

1. ENTITY:

It is a thing in real word with an independent existence. An entity may be an object with a physical existence or may be an object with a conceptual existence.



There is a type of entity, which is called weak entity:

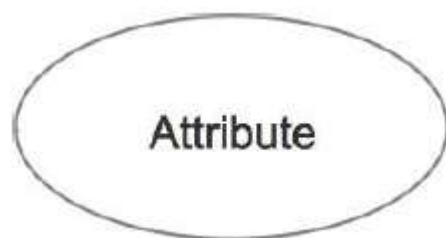
- WEAK ENTITY:

The entity, which has not any key, attributes of their own; these are called weak entity.



2. ATTRIBUTE

These are properties, which describe an entity.



TYPES OF ATTRIBUTES :

- **MULTIVALUED ATTRIBUTE:**

These attributes are shown in double Ovals. It shows the attributes that have multiple values.



- **DERIVED ATTRIBUTE:**

These attributes are shown in double dotted lines. Example, Age of the person can be a derived attribute. Because it is possible to get the age of a person when Date of Birth is known.



3. RELATIONSHIPS

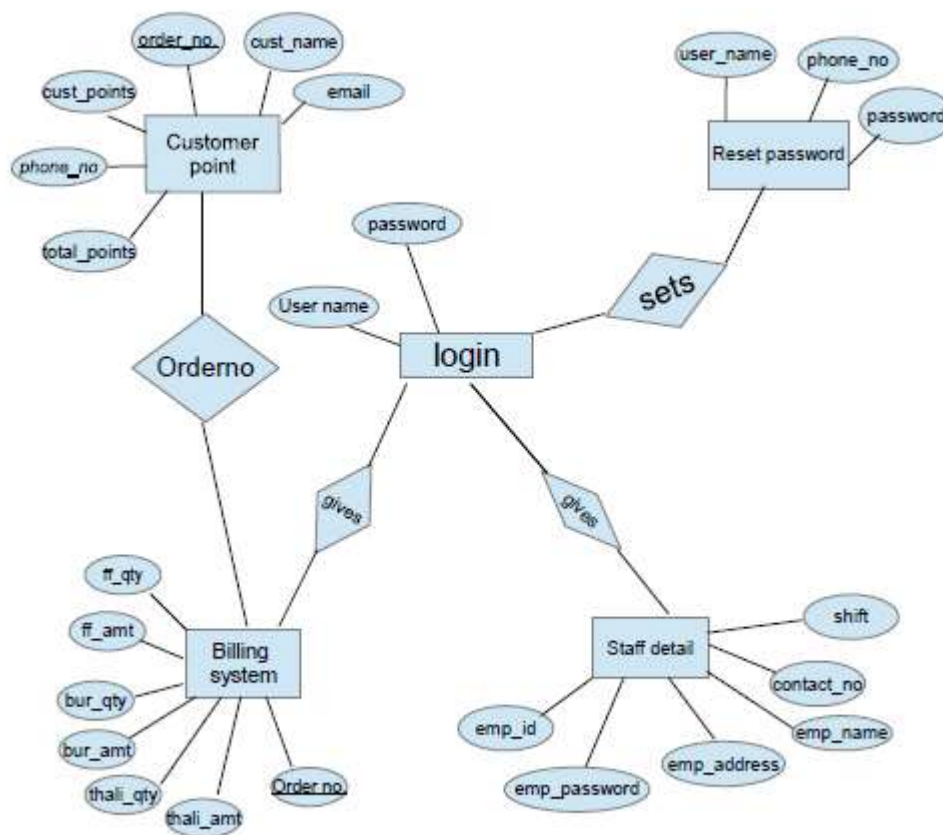
Whenever an attribute of one entity type refers to another entity type, some relationship a meaningful dependency between them. exists. It connects entities and represents a meaningful dependency between them.



4. PRIMARY KEY

It is a unique quality and is related with attribute.

E-R Diagram Of the Project



DATA STRUCTURE:-

TABLE SPECIFICANTIONS

DATABASE NAME-hotelm

TABLES NAME

Billing_sy

FIELD NAME	DATA TYPE	SIZE
order_no	INT	<u>11</u>
qty_ff	INT	<u>11</u>
amount_ff	INT	<u>11</u>
qnty_burg	INT	<u>11</u>
burg_amount	INT	<u>11</u>
qnty_san	INT	<u>11</u>
san_amount	INT	<u>11</u>

qty_pasta	INT	<u><u>11</u></u>
pasta_amount	INT	<u><u>11</u></u>
qnty_thali	INT	<u><u>11</u></u>
thali_amount	INT	<u><u>11</u></u>
qnty_coffe	INT	<u><u>11</u></u>
coffe_amount	INT	<u><u>11</u></u>
,total_qty	INT	<u><u>11</u></u>
net_amount	<u>Float</u>	
gst	<u>float</u>	
total_amount	<u>float</u>	

Customer meal points

FIELD NAME	DATA TYPE	SIZE
order_no	INT	11
customer_name	VARCHAR	30
phone_no	BIGINT	12
email	VARCHAR	30
customer_points	INT	11
total_points	INT	11

STAFF_DETAILS

FIELD NAME	DATA TYP	SIZE
emp_id	INT	10
emp_name	VARCHAR	30
emp_password	VARCHAR	30
shift	VARCHAR	30
contact_no	BIGINT	12
emp_address	VARCHAR	30

RESET_PASS

FIELD NAME	DATA TYPE	SIZE
user_name	VARCHAR	30
phone_no	BIGINT	12
Password	VARCHAR	30

Database

STRUCTURE OF DATABASE

Structure SQL Search Query Export Import Operations Privileges Routines Events						
Filters						
Containing the word: <input type="text"/>						
Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> billing_sy	Browse Structure Search Insert Empty Drop	10	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> customer_points	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> reset_pass	Browse Structure Search Insert Empty Drop	11	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> staff_details	Browse Structure Search Insert Empty Drop	11	InnoDB	latin1_swedish_ci	16 KiB	-
4 tables	Sum	40	InnoDB	latin1_swedish_ci	64 KiB	0 B
<input type="checkbox"/> Check all With selected: <input type="text"/>						

BILLING SYSTEM TABLE

Options																		
order_no	qty_tf	amount_tf	qty_burg	burg_amount	qty_san	san_amount	qty_pasta	pasta_amount	qty_thali	thali_amount	qty_coffe	coffe_amount	* 1	total_qty	net_amount	gst	total_amount	
133	8	320	5	175	6	240	2	160	3	570	1	30	25	1495	37	1532		
132	8	320	2	70	6	240	7	560	9	1710	2	60	34	2960	74	3034		
126	5	200	4	140	5	200	5	400	6	1140	3	90	28	2170	54	2224		
127	7	280	5	175	8	320	2	160	7	1330	4	120	33	2385	60	2445		
128	8	320	5	175	2	80	7	560	8	1520	6	180	36	2835	71	2906		
125	12	480	14	490	1	40	12	960	5	950	7	210	51	3130	78	3208		
130	8	320	12	420	6	240	5	400	8	1520	7	210	46	3110	78	3188		
131	15	600	7	245	6	240	7	560	3	570	7	210	45	2425	61	2486		
123	10	400	10	350	10	400	10	800	10	1900	10	300	60	4150	104	4254		
129	7	280	27	945	5	200	65	5200	22	4180	16	480	142	11285	282	11567		

CUSTOMER MEAL POINTS TABLE

Options					
order_no	customer_name	phone_no	email	customer_points	total_points
56	bhavna	965842856	choti@gmail.com	56	456
132	rama	9664665842	rama123@gmail.com	607	607
126	shayama	9664636842	shayama123@gmail.com	445	445
127	raman	9665636842	raman123@gmail.com	489	489
128	gautam	9666636842	gautam123@gmail.com	581	581
130	rajan	9662636842	rajan123@gmail.com	642	642
131	shubham	9656636842	shubham123@gmail.com	637	637
132	arjun	9659636842	arjun563@gmail.com	637	637

TESET PASSWORD TABLE

+ Options

user_name	phone_no	password
virender	9654665485	viru
virnay rawat	8956825462	vinay124
sagar aggarwal	8956568920	sag345
bhavna rati	8956568928	bahv12
sumit singh	8956568528	sumit11
sumit sharma	8956568585	sumit124
sumit sharma	8956568585	sumit124
rahul dagar	8665568520	rahul123
sumit sharma	866556520	sumit125
pramod kumar	8956856820	pramo13
sohan	8956886820	soh34

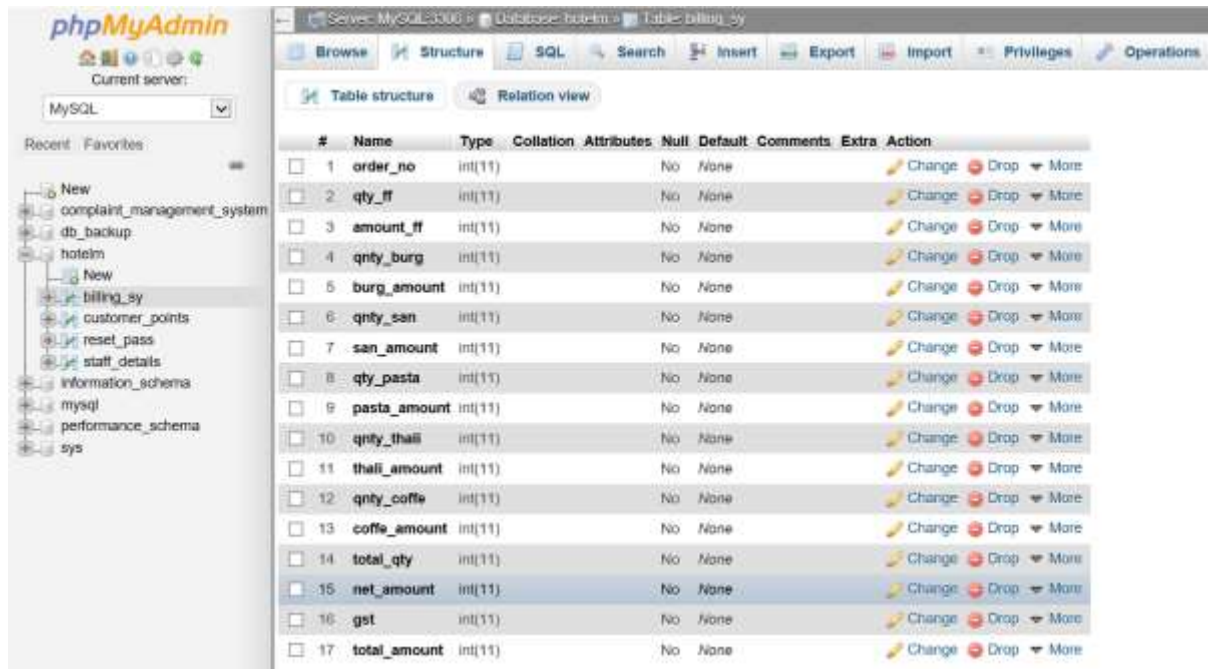
STAFF DETAILS

+ Options

emp_id	emp_name	emp_password	shift	contact_no	emp_address
25	virender	viru	morning	9654665485	saket
101	virnay rawat	vinay124	evening	8956825462	maidan giri
102	sagar aggarwal	sag345	night	8956568920	chattar pur
103	bhavna rati	bahv12	morning	8956568928	chattar pur
104	sumit singh	sumit11	morning	8956568528	chirag delhi
105	sumit sharma	sumit124	evening	8956568585	malvi nagar
105	sumit sharma	sumit124	evening	8956568585	malvi nagar
107	rahul dagar	rahul123	evening	8665568520	asian park
105	sumit sharma	sumit125	evening	866556520	asian park
106	pramod kumar	pramo13	night	8956856820	khanpur
108	sohan	soh34	night	8956886820	bangla saheb

TABLE STRUCTURES

BILLING SY



Server: MySQL:3306 • Database: hotelm • Table: billing_sy

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	order_no	int(11)		No	None			Change Drop More
<input type="checkbox"/>	2	qty_ff	int(11)		No	None			Change Drop More
<input type="checkbox"/>	3	amount_ff	int(11)		No	None			Change Drop More
<input type="checkbox"/>	4	qty_burg	int(11)		No	None			Change Drop More
<input type="checkbox"/>	5	burg_amount	int(11)		No	None			Change Drop More
<input type="checkbox"/>	6	qty_san	int(11)		No	None			Change Drop More
<input type="checkbox"/>	7	san_amount	int(11)		No	None			Change Drop More
<input type="checkbox"/>	8	qty_pasta	int(11)		No	None			Change Drop More
<input type="checkbox"/>	9	pasta_amount	int(11)		No	None			Change Drop More
<input type="checkbox"/>	10	qty_thali	int(11)		No	None			Change Drop More
<input type="checkbox"/>	11	thali_amount	int(11)		No	None			Change Drop More
<input type="checkbox"/>	12	qty_coffe	int(11)		No	None			Change Drop More
<input type="checkbox"/>	13	coffe_amount	int(11)		No	None			Change Drop More
<input type="checkbox"/>	14	total_qty	int(11)		No	None			Change Drop More
<input type="checkbox"/>	15	net_amount	int(11)		No	None			Change Drop More
<input type="checkbox"/>	16	gst	int(11)		No	None			Change Drop More
<input type="checkbox"/>	17	total_amount	int(11)		No	None			Change Drop More

CUSTOMER POINTS



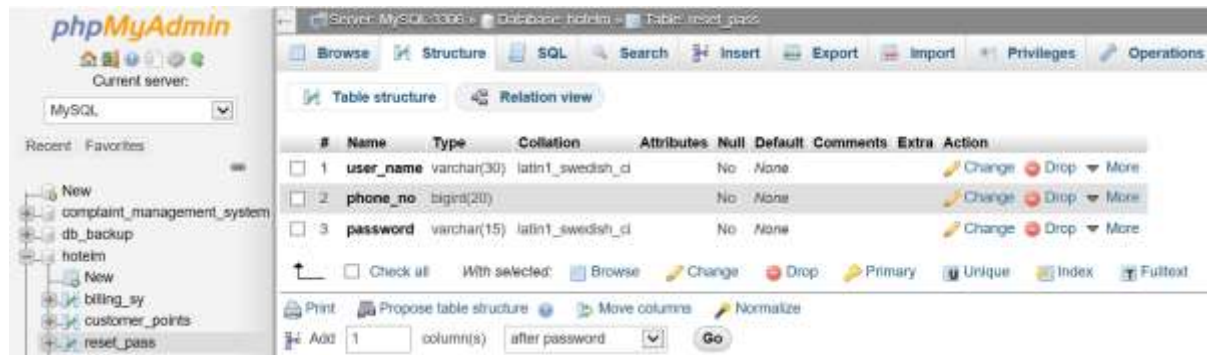
Server: MySQL:3306 • Database: hotelm • Table: customer_points

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	order_no	int(11)		No	None			Change Drop More
<input type="checkbox"/>	2	customer_name	varchar(30)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	3	phone_no	bigint(20)		No	None			Change Drop More
<input type="checkbox"/>	4	email	varchar(30)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	5	customer_points	int(11)		No	None			Change Drop More
<input type="checkbox"/>	6	total_points	int(11)		No	None			Change Drop More

☐ Check all
 With selected:
 ☐ Browse
 ☐ Change
 ☐ Drop
 ☐ Primary
 ☐ Unique
 ☐ Index
 ☐ Fulltext

RESET PASS



Server: MySQL:3306 » Database: hotelm » Table: reset_pass

Table structure

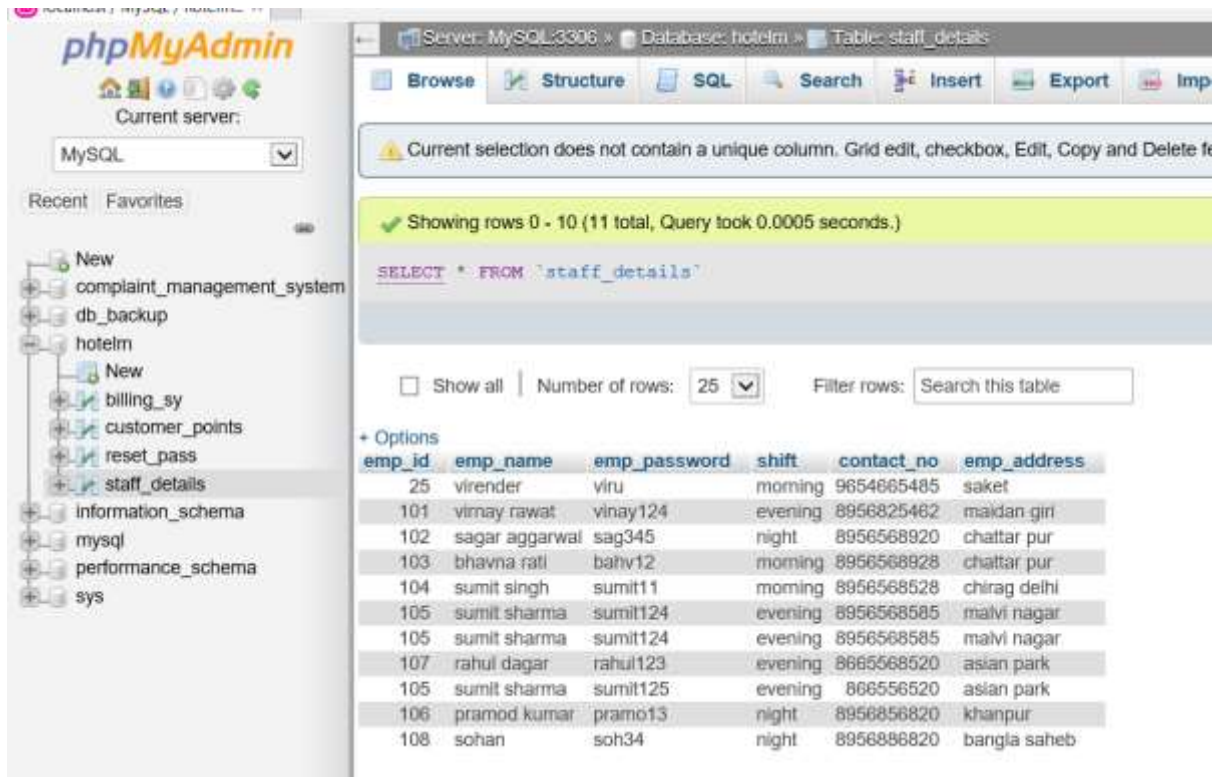
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	user_name	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
2	phone_no	bigint(20)			No	None			Change Drop More
3	password	varchar(15)	latin1_swedish_ci		No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Print Propose table structure Move columns Normalize

Add 1 column(s) after password Go

STAFF DETAIL



Server: MySQL:3306 » Database: hotelm » Table: staff_details

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete functions are disabled.

Showing rows 0 - 10 (11 total, Query took 0.0005 seconds.)

```
SELECT * FROM `staff_details`
```

Show all Number of rows: 25 Filter rows: Search this table

+ Options

emp_id	emp_name	emp_password	shift	contact_no	emp_address
25	virender	viru	morning	9654665485	saket
101	vinay rawat	vinay124	evening	8956825462	maidan giri
102	sagar aggarwal	sag345	night	8956568920	chattar pur
103	bhavna rati	bahv12	morning	8956568928	chattar pur
104	sumit singh	sumit11	morning	8956568528	chirag delhi
105	sumit sharma	sumit124	evening	8956568585	malvi nagar
105	sumit sharma	sumit124	evening	8956568585	malvi nagar
107	rahul dagar	rahul123	evening	8865568520	asian park
105	sumit sharma	sumit125	evening	8865568520	asian park
106	pramod kumar	pramo13	night	8956856820	khanpur
108	sohan	soh34	night	8956886820	bangla saheb

SOURCE CODE :-

```

import pymysql
from tkinter import *
from tkinter import messagebox

conn=pymysql.connect(host="localhost",user="root",password="",db="hotelm")
mycursor=conn.cursor()

def staff_det():
    mycursor.execute("SELECT * FROM staff_details ;")#short the rows and columns
    #to fetch a single row=> fetchone()
    #2. to fetch all rows from database=>fetchall()
    #3.To fetch any no. of rows from database=>fetchmany(size)
    # PRINT ON CONSOLE.
    res=mycursor.fetchall()

    vary=80
    sd=Toplevel()
    sd.geometry("1350x700+0+0")
    sd.config(bg="yellow")

    Label(sd,text="STAFF DETAILS",font="times 20 underline bold "fg="red",bg="yellow").place(x=342,y=10)

    Label(sd,text="EMP. id",font="times 10 underline bold ", fg="red",bg="yellow").place(x=2,y=50)
    Label(sd,text="EMP. NAME",font="times 10 underline bold ", fg="red",bg="yellow").place(x=175,y=50)
    Label(sd,text="EMP. PASSWORD ",font="times 10 underline bold ",fg="red",bg="yellow").place(x=348,y=50)
    Label(sd,text="SHIFT",font="times 10 underline bold ", fg="red",bg="yellow").place(x=521,y=50)

```

```

Label(sd,text="CONTACT_NO",font="times 10 underline bold ", fg="red",bg="yellow").place(x=640,y=50)
Label(sd,text="EMP_ADDRESS",font="times 10 underline bold ", fg="red",bg="yellow").place(x=790,y=50)

for i in range(0,mycursor.rowcount):
    Label(sd,text=res[i][0],font="times 10", fg="red",bg="yellow").place(x=50,y=vary)
    Label(sd,text=res[i][1],font="times 10", fg="red",bg="yellow").place(x=175,y=vary)
    Label(sd,text=res[i][2],font="times 10", fg="red",bg="yellow").place(x=348,y=vary)
    Label(sd,text=res[i][3],font="times 10", fg="red",bg="yellow").place(x=521,y=vary)
    Label(sd,text=res[i][4],font="times 10", fg="red",bg="yellow").place(x=640,y=vary)
    Label(sd,text=res[i][5],font="times 10", fg="red",bg="yellow").place(x=790,y=vary)

    vary+=30

def it_details():
    td=Toplevel()
    td.geometry("1350x700+0+0")
    td.config(bg="yellow")
    Label(td,text="ITEMS DETAILS",font="times 20 underline bold ",fg="red",bg="yellow").grid(row=0,
    column=1,columnspan=2,ipadx=20,ipady=20)
    Label(td,text="RATE ON THE BASIS OF PRISE",font="times 20 underline bold ",fg="red",
    bg="yellow").grid(row=1,column=0,columnspan=2,ipadx=20,ipady=20,padx=100,pady=5)

    Label(td,text="FRENCH FRIES",font="times 11 underline bold ", fg="red",bg="yellow").grid(row=2,
    column=0,ipadx=10,ipady=5,padx=10,pady=5)
    Label(td,text="40/f.plate",font="times 11", fg="red",bg="yellow").grid(row=2,column=1,ipadx=10,
    ipady=10,padx=10,pady=5)

    Label(td,text="BURGER",font="times 11", fg="red",bg="yellow").grid(row=3,column=0)
    Label(td,text="35/piece",font="times 11", fg="red",bg="yellow").grid(row=3,column=1,ipadx=20,
    ipady=20,padx=10,pady=5)

```

```
Label(td, text="SANDWICH",font="times 11", fg="red",bg="yellow").grid(row=4,column=0)
```

```
Label(td,text="40/piece",font="times 11", fg="red",bg="yellow").grid(row=4,column=1,ipadx=20,ipady=20,padx=10)
```

```
Label(td, text="PASTA",font="times 11", fg="red",bg="yellow").grid(row=5,column=0)
```

```
Label(td,text="80/f.plate",font="times 11",fg="red",bg="yellow").grid(row=5,column=1,ipadx=20,ipady=20,padx=10,pady=5)
```

```
Label(td, text="THALI",font="times 11", fg="red",bg="yellow").grid(row=6,column=0)
```

```
Label(td,text="190/f.thali",font="times 11", fg="red",bg="yellow").grid(row=6,column=1,ipadx=20,ipady=20,padx=10)
```

```
Label(td, text="COFFE",font="times 11", fg="red",bg="yellow").grid(row=7,column=0)
```

```
Label(td,text="30/cup",font="times 11", fg="red",bg="yellow").grid(row=7,column=1,ipadx=20,ipady=20 ,padx=10,pady=5)
```

```
Button(td,text="BACK",font="times 11",bg="green",bd="4",height="1",width="11",command=td.destroy).grid(row=8,column=1,columnspan=2,ipadx=20,ipady=10)
```

```
Label(td, text="* 1 RUPEES = 1 POINT",font="times 11", fg="red",bg="yellow").grid(row=9)
```

```
Label(td,text="RATE ON THE BASIS OF POINTS",font="times 20 underline bold ", fg="red",bg="yellow").grid(row=1,column=2,columnspan=2,ipadx=20,ipady=20,padx=100,pady=5)
```

```
Label(td,text="FRENCH FRIES",font="times 11", fg="red",bg="yellow").grid(row=2,column=2,ipadx=10,ipady=5,padx=10,pady=5)
```

```
Label(td,text="40/f.plate",font="times 11", fg="red",bg="yellow").grid(row=2,column=3,ipadx=10,ipady=10,padx=10,pady=5)
```

```
Label(td,text="BURGER",font="times 11", fg="red",bg="yellow").grid(row=3,column=2)
```

```
Label(td,text="35/piece",font="times 11", fg="red",bg="yellow").grid(row=3,column=3,ipadx=20,ipady=20,padx=10,pady=5)
```

```
Label(td, text="SANDWICH",font="times 11", fg="red",bg="yellow").grid(row=4,column=2)
```

```
Label(td,text="40/piece",font="times 11", fg="red",bg="yellow").grid(row=4,column=3,ipadx=20,ipady=20,padx=10)
```

```

Label(td, text="PASTA",font="times 11", fg="red",bg="yellow").grid(row=5,column=2)
Label(td,text="80/f.plate",font="times 11", fg="red",bg="yellow").grid(row=5,column=3,ipadx=20,ipady=20,
padx=10,pady=5)
Label(td, text="THALI",font="times 11", fg="red",bg="yellow").grid(row=6,column=2)
Label(td,text="190/f.thali",font="times 11", fg="red",bg="yellow").grid(row=6,column=3,ipadx=20,
ipady=20,padx=10)
Label(td, text="COFFE",font="times 11", fg="red",bg="yellow").grid(row=7,column=2)
Label(td,text="30/cup",font="times 11", fg="red",bg="yellow").grid(row=7,column=3,ipadx=20,ipady=20
,padx=10,pady=5)

```

#module of customer point game

def cmp():#here cmp means customer points game

```

def insp():
    a=orderno.get()
    b=cs_name.get()
    c=ph_no.get()
    d=mail.get()
    e=cpo.get()
    f=tpo.get()
    g=tco.get()
    def calp():
        x=int(tco.get())
        y=a*0.05
        cpo.set(y)

```

```
try:
```

```
    mycursor.execute("INSERT INTO customer_points(order_no,customer_name,phone_no,email,
customer_points,total_points)VALUES('"+a+"','"+b+"','"+c+"','"+d+"','"+e+"','"+f+"');")
    conn.commit()
    messagebox.showinfo("message","inserted successfully")
```

```
except:
```

```
    conn.rollback()
    conn.close()
```

```
cd=Toplevel()
```

```
cd.geometry("500x700+250+15")
```

```
cd.config(bg="yellow")
```

```
Label(cd,text="Customer Information",font="times 30 bold underline", fg="blue",bg="yellow"
).grid(row=0,column=0,ipadx=20,ipady=20,padx=10,pady=10,columnspan=2)
```

```
Label(cd,text="ORDER NO.",font="times 14 ", fg="red",bg="yellow").grid(row=1,column=0,ipadx=20,
ipady=20,padx=10,pady=5)
```

```
orderno=StringVar()
```

```
Entry(cd,textvariable=orderno).grid(row=1,column=1,ipadx=10,ipady=5,padx=10,pady=5)
```

```
Label(cd,text="CUSTOMER NAME",font="times 14 ", fg="red",bg="yellow").grid(row=2,column=0,ipadx=20,
ipady=20,padx=10,pady=5)
```

```
cs_name=StringVar()
```

```
Entry(cd,textvariable=cs_name).grid(row=2,column=1,ipadx=10,ipady=5,padx=10,pady=5)
```

```
Label(cd,text="PHONE NO.",font="times 14 ", fg="red",bg="yellow").grid(row=3,column=0,
ipadx=20,ipady=20,padx=10,pady=5)
```

```
ph_no=StringVar()
```

```
Entry(cd,textvariable=ph_no).grid(row=3,column=1,ipadx=10,ipady=5,padx=10,pady=5)
```



```
Label(cd,text="EMAIL",font="times 14 ", fg="red",bg="yellow").grid(row=4,column=0,ipadx=20,
ipady=20,padx=10,pady=5)
```

```
mail=StringVar()
```

```
Entry(cd,textvariable=mail).grid(row=4,column=1,ipadx=10,ipady=5,padx=10,pady=5)
```

```
Label(cd,text="TOTAL COST",font="times 14 ", fg="red",bg="yellow").grid(row=5,column=0,ipadx=20,
ipady=20,padx=10,pady=5)
```

```
tco=StringVar()
```

```
Entry(cd,textvariable=tco).grid(row=5,column=1,ipadx=10,ipady=5,padx=10,pady=5)
```

```
Label(cd,text="CUSTOMER POINTS",font="times 14 ", fg="red",bg="yellow").grid(row=6,column=0,ipadx=20,
ipady=20,padx=10,pady=5)
```

```
cpo=StringVar()
```

```
Entry(cd,textvariable=cpo).grid(row=6,column=1,ipadx=10,ipady=5,padx=10,pady=6)
```

```
Label(cd,text="TOTAL POINTS",font="times 14 ", fg="red",bg="yellow").grid(row=7,column=0,ipadx=20,
ipady=20,padx=10,pady=5)
```

```
tpo=StringVar()
```

```
Entry(cd,textvariable=tpo).grid(row=7,column=1,ipadx=10,ipady=5,padx=10,pady=5)
```

```
Button(cd,text="SET",font="times 11",bg="green",bd="4",height="1",width="11",command=insp
).grid(row=8,column=1,ipadx=20,ipady=10,padx=20,pady=20)
```

```
Button(cd,text="back",font="times 11",bg="green",bd="4",height="1",width="11",command=cd.destroy
).grid(row=8,column=0,ipadx=20,ipady=10,padx=20,pady=20)
```

```
def billing_s():
```

```
    mp=Toplevel()
```

```
    mp.geometry("1200x1200+5+5")
```

```

def ins():
    order=ordno.get()
    a=qty1.get()
    b=qty2.get()
    c=qty3.get()
    d=qty4.get()
    e=qty5.get()
    f=qty6.get()

    g=fr_fries.get()
    h=burger.get()
    i=sandwich.get()
    j=pasta.get()
    k=thali.get()
    l=coffe.get()

m=totalqty.get()
n=net_total.get()
p=gst.get()
q=total_amt.get()

#try:
mycursor.execute("INSERT INTO billing_sy(order_no,qty_ff,amount_ff,qnty_burg,burg_amount,qnty_san
,san_amount,qty_pasta,pasta_amount,qnty_thali,thali_amount,qnty_coffe,coffe_amount,total_qty
,net_amount,gst,total_amount)VALUES('"+order+"','"+a+"','"+g+"','"+b+"','"+h+"','"+c+"','"+i+"','"+d+"',
 '"+j+"','"+e+"','"+k+"','"+f+"','"+l+"','"+m+"','"+n+"','"+p+"','"+q+"');")
conn.commit()
messagebox.showinfo("message","inserted successfully")

```

```
#except:
```

```
    #conn.rollback()
```

```
    #print("will not execute")
```

```
    #conn.close()
```

```
def add():
```

```
    a=int(qty1.get())
```

```
    b=int(qty2.get())
```

```
    c=int(qty3.get())
```

```
    d=int(qty4.get())
```

```
    e=int(qty5.get())
```

```
    f=int(qty6.get())
```

```
    s=a+b+c+d+e+f
```

```
    totalqty.set(s)
```

```
tf=40*a
```

```
tb=35*b
```

```
ts=40*c
```

```
tp=80*d
```

```
tt=190*e
```

```
tc=30*f
```

```
fr_fries.set(tf)
```

```
burger.set(tb)
```

```
sandwich.set(ts)
```

```
pasta.set(tp)
```

```
thali.set(tt)
```

```
coffe.set(tc)
```

```

g=int(fr_fries.get())
h=int(burger.get())
i=int(sandwich.get())
j=int(pasta.get())
k=int(thali.get())
l=int(coffe.get())
namt=g+h+i+j+k+l
net_total.set(namt)

```

```

m=int(net_total.get())
gnt=m*(2.5/100)
gst.set(gnt)
toa=gnt+namt
total_amt.set(toa)

```

```

mp.title("BILLING PAGE")
mp.config(bg="yellow")
Label(mp,text="ORDER NO.",font="times 14 bold underline", fg="red",bg="yellow"
).grid(row=0,column=0,ipadx=20,ipady=20,padx=10,pady=5)
ordno=StringVar()
Entry(mp,textvariable=ordno).grid(row=0,column=1,ipadx=10,ipady=5,padx=10,pady=5)
Label(mp,text="ITEMS",font="times 14 bold underline", fg="red",bg="yellow").grid(row=1,
column=0,ipadx=20,ipady=20,padx=10,pady=10)
Label(mp,text="RATE",font="times 14 bold underline", fg="red",bg="yellow").grid(row=1,column=1,
ipadx=20,ipady=20,padx=10,pady=10)
Label(mp,text="QUANTITY",font="times 14 bold underline", fg="red",bg="yellow").grid(row=1,
column=2,ipadx=20,ipady=20,padx=10,pady=10)
Label(mp,text="AMOUNT",font="times 14 bold underline", fg="red",bg="yellow").grid(row=1,column=3,
ipadx=20,ipady=20,padx=10,pady=10)

```

```
Label(mp,text="FRENCH FRIES",font="times 11", fg="red",bg="yellow").grid(row=2,column=0,ipadx=10,
ipady=5,padx=10,pady=5)
```

```
Label(mp,text="40/f.plate",font="times 11", fg="red",bg="yellow").grid(row=2,column=1,
ipadx=10,ipady=10,padx=10,pady=5)
```

```
qty1=StringVar()
```

```
Entry(mp,textvariable=qty1).grid(row=2,column=2,ipadx=10,ipady=5,padx=10,pady=5)
```

```
fr_fries=StringVar()
```

```
Entry(mp,textvariable=fr_fries).grid(row=2,column=3,ipadx=10,ipady=5,padx=10,pady=5)
```

```
Label(mp,text="BURGER",font="times 11", fg="red",bg="yellow").grid(row=3,column=0)
```

```
Label(mp,text="35/piece",font="times11",fg="red",bg="yellow").grid(row=3,column=1,ipadx=20,
ipady=20,padx=10,pady=5)
```

```
qty2=StringVar()
```

```
Entry(mp,textvariable=qty2).grid(row=3,column=2,ipadx=10,ipady=5,padx=10,pady=5)
```

```
burger=StringVar()
```

```
Entry(mp,textvariable=burger).grid(row=3,column=3,ipadx=10,ipady=5,padx=10,pady=5)
```

```
Button(mp,text="CALCULATE",bd="4",height="1",width="15",command=add).grid(row=3,column=4)
```

```
Label(mp, text="SANDWICH",font="times 11", fg="red",bg="yellow").grid(row=4,column=0)
```

```
Label(mp,text="40/piece",font="times 11", fg="red",bg="yellow").grid(row=4,column=1,
ipadx=20,ipady=20,padx=10)
```

```
qty3=StringVar()
```

```
Entry(mp,textvariable=qty3).grid(row=4,column=2,ipadx=10,ipady=5,padx=10)
```

```
sandwich=StringVar()
```

```
Entry(mp,textvariable=sandwich).grid(row=4,column=3,ipadx=10,ipady=5,padx=10)
```

```
Button(mp,text='SUBMIT',bd="4",height="1",width="15",command=ins).grid(row=4,column=4,padx=10,
pady=5)
```



```

Label(mp,text="GST",font="times 11", fg="red",bg="yellow").grid(row=10,column=2)

gst=StringVar()

Entry(mp,textvariable=gst).grid(row=10,column=3,ipadx=10,ipady=5,padx=10,pady=5)


Label(mp,text="TOTAL AMOUNT",font="times 11", fg="red",bg="yellow").grid(row=11,column=2)

total_amt=StringVar()

Entry(mp,textvariable=total_amt).grid(row=11,column=3,ipadx=10,ipady=5,padx=10,pady=5)

```

```

def main_page():

    ma=Toplevel()

    ma.title("MAIN PAGE")

    ma.geometry("600x500+450+80")

    ma.config(bg="yellow")

    Label(ma,text="",bg="yellow").grid(row=0,column=0,padx=40,pady=30)

    Button(ma,text="CUSTOMER MEAL POINTS",font="times 11",bd="4",height="1",width="15",command=cmp
    ).grid(row=1,column=0,ipadx=40,ipady=20,pady=20,padx=40)

    Button(ma,text="BILLING SYSTEM",font="times 11",bd="4",height="1",width="15",
    command=billing_s).grid(row=1,column=1,ipadx=40,ipady=20,padx=40,pady=20)

    Label(ma,text="",bg="yellow").grid(row=2,column=0,padx=20,pady=10)


    Button(ma,text="PRODUCTS DETAIL",font="times 11",bd="4",height="1",width="15",
    command=it_details).grid(row=3,column=0,ipadx=40,ipady=20,padx=40,pady=20)

    Button(ma,text="STAFF DETAIL",font="times 11",bd="4",height="1",width="15",command=staff_det
    ).grid(row=3,column=1,ipadx=40,ipady=20,padx=40,pady=20)

    Label(ma,text="",bg="yellow").grid(row=4,column=0,padx=40)

    Button(ma,text="LOGOUT",font="times 11",bd="4",height="1",width="15",command=ma.destroy
    ).grid(row=5,column=0,ipadx=40,ipady=20,padx=20,pady=20,columnspan=2)

```

```

root=Tk()

def login():
    aco=userid.get()
    pn=password.get()

    if(mycursor.execute("SELECT * FROM reset_pass WHERE user_name='"+aco+"' AND password='"+pn+"' ;")==
True):
        main_page()
        conn.commit()
    else:
        conn.rollback()
        messagebox.showinfo("message","Please check user id and password")

def forgot_pass():

def set_pass():
    u=usn.get()
    p=ph.get()
    n=np.get()
    r=rnp.get()

    if(n==r):
        mycursor.execute("UPDATE reset_pass SET password='"+r+"' WHERE user_name='"+u+"'
AND phone_no='"+p+"' ;")
        conn.commit()
        messagebox.showinfo("message","PASSWORD HAS CHANGED")
    else:
        conn.rollback()
        messagebox.showinfo("message","your password not match")

```



```

cd=Toplevel()
cd.geometry("500x700+250+15")
cd.config(bg="yellow")
Label(cd,text="RESET PASSWORD",font="times 30 bold underline", fg="blue",bg="yellow").grid(row=0,
column=0,ipadx=20,ipady=20,padx=10,pady=10,columnspan=2)

Label(cd,text="USER NAME",font="times 14 ", fg="red",bg="yellow").grid(row=1,column=0,ipadx=20,
ipady=20,padx=10,pady=5)
usn=StringVar()
Entry(cd,textvariable=usn).grid(row=1,column=1,ipadx=10,ipady=5,padx=10,pady=5)

Label(cd,text="PHONE NO.",font="times 14 ", fg="red",bg="yellow").grid(row=2,column=0,ipadx=20,
ipady=20,padx=10,pady=5)
ph=StringVar()
Entry(cd,textvariable=ph).grid(row=2,column=1,ipadx=10,ipady=5,padx=10,pady=5)

Label(cd,text="NEW PASSWORD ",font="times 14 ", fg="red",bg="yellow").grid(row=3,column=0,ipadx=20,
ipady=20,padx=10,pady=5)
np=StringVar()
Entry(cd,textvariable=np).grid(row=3,column=1,ipadx=10,ipady=5,padx=10,pady=5)

Label(cd,text="REENTER PASSWORD ",font="times 14 ", fg="red",bg="yellow").grid(row=4,column=0,
ipadx=20,ipady=20,padx=10,pady=5)
rnp=StringVar()
Entry(cd,textvariable=rnp).grid(row=4,column=1,ipadx=10,ipady=5,padx=10,pady=5)

Button(cd,text="SET",font="times 11",bg="green",bd="4",height="1",width="11",command=set_pass
).grid(row=5,column=1,ipadx=20,ipady=10,padx=20,pady=20)
Button(cd,text="back",font="times 11",bg="green",bd="4",height="1",width="11",command=cd.destroy
).grid(row=5,column=0,ipadx=20,ipady=10,padx=20,pady=20)

```

```

def sign_up():

    def set_sdet():
        a=emp_id.get();
        b=emp_name.get();
        c=emp_pass.get();
        d=shift.get();
        e=contact_no.get();
        f=emp_addr.get();

        mycursor.execute("INSERT INTO staff_details(emp_id,emp_name,emp_password,shift,contact_no,
        emp_address)VALUES('"+a+"','"+b+"','"+c+"','"+d+"','"+e+"','"+f+"');")
        conn.commit()

        mycursor.execute("INSERT INTO reset_pass(user_name,phone_no,password)
        VALUES('"+b+"','"+e+"','"+c+"');")
        conn.commit()

        messagebox.showinfo("message","sign up successfully")

    cd=Toplevel()
    cd.geometry("500x700+250+15")
    cd.config(bg="yellow")
    Label(cd,text="SIGN UP",font="times 30 bold underline", fg="blue",bg="yellow").grid(row=0,column=0,
    ipadx=20,ipady=20,padx=10,pady=10,columnspan=2)

    Label(cd,text="EMP_ID",font="times 14 ", fg="red",bg="yellow").grid(row=1,column=0,ipadx=20,
    ipady=20,padx=10,pady=5)
    emp_id=StringVar()
    Entry(cd,textvariable=emp_id).grid(row=1,column=1,ipadx=10,ipady=5,padx=10,pady=5)

    Label(cd,text="EMP_NAME",font="times 14 ", fg="red",bg="yellow").grid(row=2,column=0,
    ipadx=20,ipady=20,padx=10,pady=5)

```

```

emp_name=StringVar()
Entry(cd,textvariable=emp_name).grid(row=2,column=1,ipadx=10,ipady=5,padx=10,pady=5)

Label(cd,text="EMP_PASSWORD",font="times 14 ", fg="red",bg="yellow").grid(row=3,column=0,
ipadx=20,ipady=20,padx=10,pady=5)
emp_pass=StringVar()
Entry(cd,textvariable=emp_pass).grid(row=3,column=1,ipadx=10,ipady=5,padx=10,pady=5)

Label(cd,text="SHIFT",font="times 14 ", fg="red",bg="yellow").grid(row=4,column=0,ipadx=20,
ipady=20,padx=10,pady=5)
shift=StringVar()
Entry(cd,textvariable=shift).grid(row=4,column=1,ipadx=10,ipady=5,padx=10,pady=5)

Label(cd,text="contact_no",font="times 14 ", fg="red",bg="yellow").grid(row=5,column=0,ipadx=20,
ipady=20,padx=10,pady=5)
contact_no=StringVar()
Entry(cd,textvariable=contact_no).grid(row=5,column=1,ipadx=10,ipady=5,padx=10,pady=5)

Label(cd,text="emp_address",font="times 14 ", fg="red",bg="yellow").grid(row=6,column=0,ipadx=20,
ipady=20,padx=10,pady=5)
emp_addr=StringVar()
Entry(cd,textvariable=emp_addr).grid(row=6,column=1,ipadx=10,ipady=5,padx=10,pady=5)

Button(cd,text="SET",font="times 11",bg="green",bd="4",height="1",width="11",command=set_sdet
).grid(row=7,column=1,ipadx=20,ipady=10,padx=20,pady=20)
Button(cd,text="back",font="times 11",bg="green",bd="4",height="1",width="11",command=cd.destroy
).grid(row=7,column=0,ipadx=20,ipady=10,padx=20,pady=20)

```

```

root.geometry("400x500+450+80")

root.config(bg="yellow")

Label(root,text="RESTAURANT",font="calibri 35 bold
underline",fg="blue").grid(row=0,column=0,columnspan=2,ipadx=45)


Label(root,text="USER NAME").grid(row=1,column=0,padx=20,pady=30,ipadx=20,ipady=10)
userid=StringVar()
Entry(root,textvariable=userid).grid(row=1,column=1,padx=30,pady=30,ipadx=20,ipady=10)
Label(root,text="PASSWORD").grid(row=2,column=0,padx=20,pady=20,ipadx=20,ipady=10)
password=StringVar()
Entry(root,textvariable=password,show="*").grid(row=2,column=1,padx=20,pady=20,ipadx=20,ipady=10)

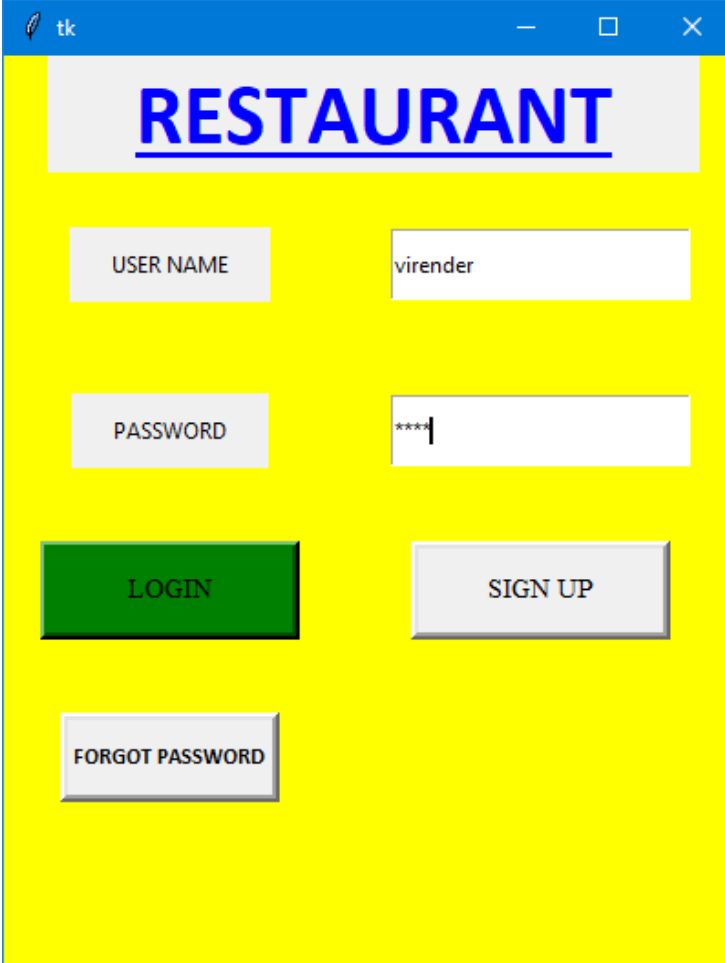

Button(root,text="LOGIN",font="times11",bg="green",bd="4",height="1",width="11",command=login).grid(ro
w=3,column=0,ipadx=20,ipady=10,padx=20,pady=20)

Button(root,text="SIGN UP",font="times11",bd="4",height="1",width="11",command=sign_up).grid(row=3,
column=1,ipadx=20,ipady=10,padx=20,pady=20)

Button(root,text="FORGOT PASSWORD",font="calibri 9
bold",bd="4",height="1",width="11",command=forgot_pass).grid(row=4,column=0,ipadx=20,ipady=10,padx=2
0,pady=20)

```

OUTPUT SCREENS :-



A screenshot of a Tkinter window titled 'tk' with a blue title bar. The window has a yellow background. At the top, the word 'RESTAURANT' is displayed in large, bold, blue, underlined capital letters. Below this, there are four input fields and three buttons. The first row has a 'USER NAME' label and a text entry field containing 'virender'. The second row has a 'PASSWORD' label and a text entry field containing '****'. Below these, there are three buttons: a green 'LOGIN' button, a grey 'SIGN UP' button, and a grey 'FORGOT PASSWORD' button.

RESTAURANT	
USER NAME	virender
PASSWORD	****
LOGIN	SIGN UP
FORGOT PASSWORD	

tk

— □ ×

SIGN UP

EMP_ID

104

EMP_NAME

sumit singh

EMP_PASSWORD

sum

SHIFT

mor

contact_no

8956568528

emp_address

chirag delhi

back

SET

message

×

i

sign up successfully

OK

RESET PASSWORD

USER NAME

VIRENDER

PHOT

NEW PA

REENTER PASSWORD

VIRU

[back](#)

SET

message

PASSWORD HAS CHANGED

OK

BILLING PAGE

ORDER NO. 123

ITEMS	RATE	QUANTITY	AMOUNT
FRENCH FRIES	40 /plate	10	400
BURGER	35 /piece	10	350
SANDWICH	40 /plate		400
PASTA	80 /plate		800
THALI	190 /plate		1900
COFFE	30 /plate		300
TOTAL		60	4150
GST			103.75
TOTAL AMOUNT			4253.75

CALCULATE

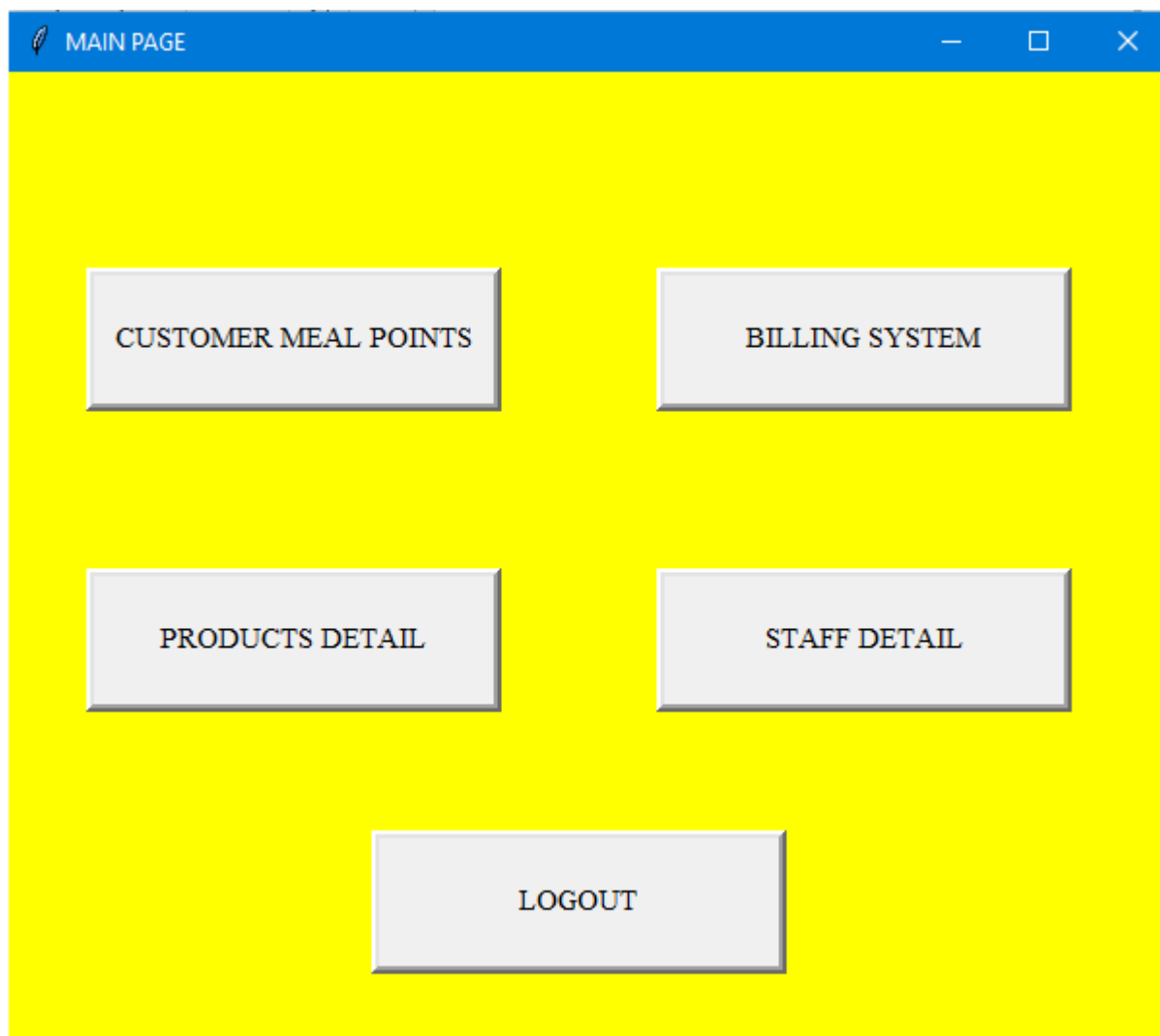
SUBMIT

message

i

inserted successfully

OK



tk

Customer Information

ORDER NO.

CUSTOMER NAME

PHONE NO.

EMAIL

TOTAL C

CUSTOMER POINTS

TOTAL POINTS

OK

message

i inserted successfully

<u>RATE ON THE BASIS OF PRICE</u>		<u>RATE ON THE BASIS OF POINTS</u>	
<u>FRENCH FRIES</u>	40 ₹/plate	FRENCH FRIES	40 ₹/plate
BURGER	35/piece	BURGER	35/piece
SANDWICH	40/piece	SANDWICH	40/piece
PASTA	80 ₹/plate	PASTA	80 ₹/plate
THALI	190 ₹/thali	THALI	190 ₹/thali
COFFE	30/cup	COFFE	30/cup
<div>BACK</div>			

EMP_ID	EMP_NAME	EMP_PASSWORD	SHIFT	CONTACT_NO	EMP_ADDRESS
25	virender	viru	morning	9654665485	saket
101	vinay rawat	vinay124	evening	8956825462	maidan giri
102	sagar aggarwal	sag345	night	8956568920	chattar pur
103	bhavna rati	bahv12	morning	8956568928	chattar pur
104	sumit singh	sumit11	morning	8956568528	chirag delhi
105	sumit sharma	sumit124	evening	8956568585	malvi nagar
105	sumit sharma	sumit124	evening	8956568585	malvi nagar
107	rahul dagar	rahul123	evening	8665568520	asian park
105	sumit sharma	sumit125	evening	866556520	asian park
106	pramod kumar	pramo13	night	8956856820	khanpur
108	sohan	soh34	night	8956886820	bangla saheb

TESTING

The main purpose of software testing is to identify defects in the software. Defects in software testing can be defined as variance from requirement or user expectation. Software testing can also be stated as the process of validating and verifying that a software program/application:

- Meets the business and technical requirements that guided its design and development;
- Works as expected; and
- Can be implemented with the same characteristics.

Testing is divided into different types. This division helps management to manage various activities effectively. Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. There are many ways in which software testing can be categorized. Some of them are described as follows:

BLACE BOX TESTING

Black box testing treats the software as a “black box”- it does not consider the internal system design. In Black box testing, tests are based on requirements and functionality i.e., focus of all the activities is only on the functionality of system and not on what is happening inside the system. Black box testing is also known as Behavior tests or Boundary value testing. Black box testing methods include:

- Equivalence partitioning,
- Boundary value analysis,
- All-pairs testing,
- Model-based testing,

WHITE BOX TESTING

White box testing is based on an analysis of internal workings, language and structure of the software. Tests are based on coverage of code statements, branches, paths and conditions. Purpose of white box testing is to make sure that

- Functionality of the system is proper
- Information on the code coverage

White box testing is also known as Glass box testing and Structural testing.

UNIT TESTING

Unit testing refers to the testing of individual software components or modules. In Unit testing, the source code is divided into modules, which in turn are divided into smaller units called units. The test done on these units of code is called Unit test. Unit test depends upon the language on which the project is developed. Unit test ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results.

SECURITY TESTING

Security testing refers to the testing of database and network software in order to keep data and resources secure from mistaken or accidental users, hackers, and other unauthorized internal and external accesses.

ALPHA TESTING

Alpha testing is done at the end of development when the code is mostly complete or contains most of the functionality and prior to users being involved. Sometimes a select group of users are involved. Still Major design changes may be made as a result of such testing.

BETA TESTING

Beta testing is done after the product code is complete. It is the final testing before releasing application for commercial purpose. Beta testing is done by the end-users or others.

SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. This phase is started after the completion of other phases like Unit, Component and Integration testing. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

INTEGRATION TESTING

In Integration testing, the modules are combined and tested as a group to verify the combined functionality of modules after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc.

Integration Testing follows unit testing

MAINTENANCE

The last part of the system development life cycle is system Maintenance which is actually the implementation of the post – implementation review plan. When systems are installed they are generally used for long periods. The average life of a system is 4 to 6 years with oldest application often in use for over 10 years. However, this period of use brings with it the need to continually maintain the system. Programmers/analyst spends sufficient time for maintaining programs. The study on the maintenance requirement for the information system of this Project Online Room Booking revealed that:

- (a) Sixty to Ninety percent of the overall cost of the software during life of a system is spent on maintenance.
- (b) In documented cases, cost of maintenance, when measured on the basis of writing each instruction in coding form, is more than the 50 times the cost of developing a system.
- (c) The software demand is increasing at a faster rate than supply many programmers are devoting more time on system maintenance than on a new software development. There is backlog of new development work.

So in this Project the maintenance can be classified as:

Corrective maintenance means repairing, processing or performance failures or making alterations because of previously ill-defined problems.

Adaptation maintenance means changing the program functions. Enhancing the performance or modifying the programs according to user additional or changing needs are included in perfective maintenance the greatest amount of maintenance work is for

user enhancement and improve documentation of the system for better efficiency. More time and money are spent on perfective than on corrective and adaptive maintenance together

FUTURE SCOPE OF RESTAURANT MANAGEMENT SYSTEM

The proposed Restaurant Management System basically designed to provide the users with a greater facility.

It simply saves the time of users by Record Restaurant Data. This system in its current scope basically enables to provide some of the basic facilities of a System and management system such as Customers info. form, Paymen. It does not provide all the services provided by an System

More User-friendly interface.

- Enhanced graphics and designing.
- Security
- Fast searching techniques.
- Supports a broad range of input and output devices.
- In future this project will support distributed database rather than centralized database.

CONCLUSION

The Restaurant Management System enables users to directly Record and view the data anytime by use of system. Users get easiest way to see data .The users or view every facility provide by the RESTAURANT MANAGEMENT SYSTEM. It saves time of users. The records of the users are directly saved to the database.

BIBLIOGRAPHY

- TechEplica, South Ex. , Delhi (Study Python Course)
- Google.com
- Youtube.com
- Faculty Guide