Chapter-1

INTRODUCTION

1.1. Introduction of the System

Foodservice or catering service defines those businesses, institutions, and companies responsible for any meal prepared outside the home. Catering is the business of providing food services at a remote site or a site such as hotel , hospital, pub, aircraft, cruise ship, park, filming site or studio, entertainment site, or event venue. The companies that supply foodservice operators are called foodservice distributors. Foodservice distributors sell goods like small wares (kitchen utensils) and foods. Some companies manufacture products in both consumer and foodservice versions.

In today’s world, almost all kinds of businesses have started opening shops online. You can see shoes and clothes being sold online. So, it is only a matter of time before food was sold online. The revenue got by selling food online is going to increase in every major country. Thus, promoters who can develop a superb online food ordering software will be able to reap profits.

In Dakshina Kannada and Udupi district food and DJ/Lighting services are becoming emerging business domains. People are started preferring food service providers for various reasons such as cost effective, responsibility etc. In this electronic world, people prefer to check food providers information such as type of food, categories of food available, price list of food etc. through electronic media whenever they need to know. So it is better to design a system that provides every information regarding the food service/ catering services and lighting services which helps both owners to get more profit and to reach maximum customers.

1.1.1. Project Title

“Dream Organizer”

1.1.2. Category

- Web based Relational Database Management System.

1.1.3 Overview

The Catering system is the business of providing food service to the people at the events that are conducted. User will get a complete information about the food and he can order the food through this system. User will get a image of the food with the description of the food. Also,system allows users to find decorators, lighting persons.

1.2 Background

The main purpose of development of this web application is to expose various service availability of a person who owned a catering service, Sound and Lighting services. This system allows a food service person to provide all information of various food and allow the customer to selectively order the items of his choice for functions. Catering owner will add various food details to the system including its type and category

1.3. Objectives of the System

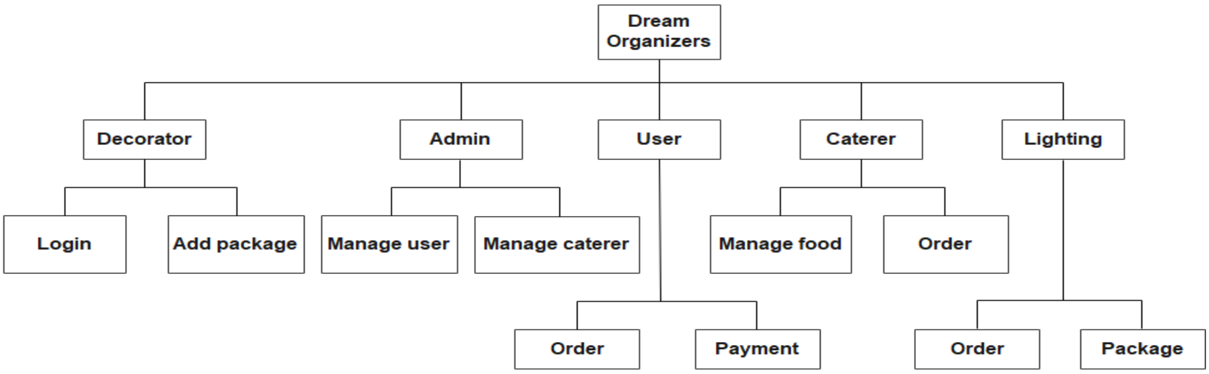
The main objectives of proposed system are:

- To expose various service availability of a person who owned a catering service, Sound and Lighting services.
- To provide all information of various food and allow the customer to selectively order the items of his choice for functions

1.4. Scope of the System

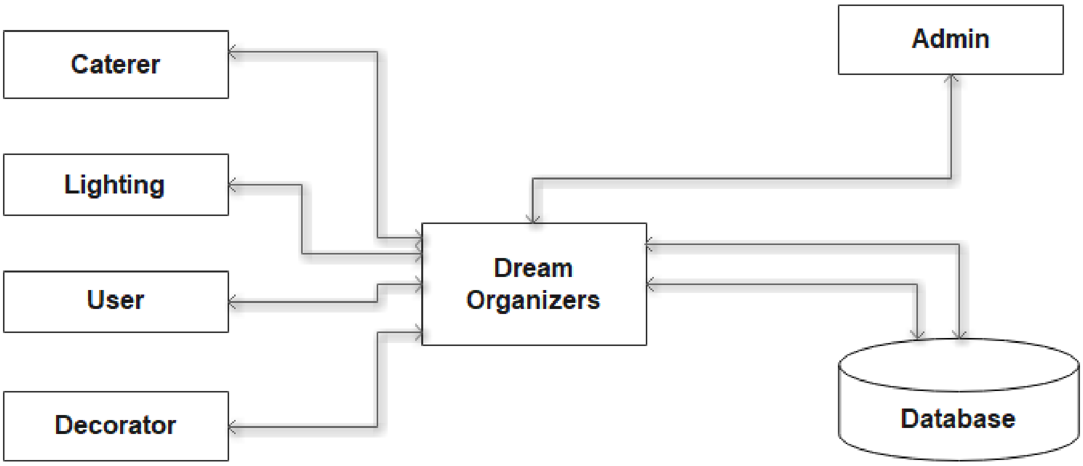
India is a country of numerous festivals and ceremonies. There are numerous social gatherings, both personal and professional. And so, with every possible social gathering, it is inevitable for someone to host the gathering. And thus, we see a boom in the catering industry. The catering industry in the country is finally extending its reach beyond marriages. Till a few years ago, the members of the Indian families usually cooked during social gatherings. Now they are hiring caterers for the same. For ensuring quality standards, such people hire professional catering services. Moreover, apart from the social gatherings and festive occasions, the formal events, seminars, conferences and other business related events consider professional catering services. This adds to the success of the catering industry.

1.5. Structure of the System



1.6. System Architecture

Architectural design is the process of decomposing a large complex system into small subsystem. These sub systems are meant for providing some related services. The architectural design is basically a layout framework for the system for subsystem control and communication.



1.7 End Users

- Admin
- Caterer, Decorator, Lighter
- User

1.7.1. Module

Admin Module:- Admin will be managing the website. He can perform the following activities:

- Manage Users
- Manage Caterer
- Manage Feedback
- Manage Payment
- Manage Advertisement
- Manage Reports

User Module:- User can login to this website by providing his Email ID and Password. He can perform following activities: -

- Login
- Give Feedback
- Search Food
- Make Advance Payment
- Select Food

Caterer Module:- Caterer can get registered to this website and can perform following activities: -

- Login
- Payment
- Provide Caterer Information
- Add Food

Decorator Module:- Decorator can get registered to this website and can perform following activities: -

- Login
- Payment
- Provide Decoration Information

Lighting Module:- Lighting can get registered to this website and can perform following activities: -

- Login
- Payment
- Provide lighting Information

1.7.2 Module Description

- Login Module: This module is used to login into the system by admin and user, Food owner and lighting owner.
- Registration module: This module is used to register a new user, food owner and lighting owner to the system. System is to be provided with basic information of user during registration.

- Add food-item Module: The food owner can add food-item into the database after logging in to the website. The food-item are classified under various sections and according to types of food-item.
- Update food-item Module: Food owner can update the food-item information based on the modifications in terms of food price.
- Add Cart Module : This module is used to place the selected food items into the cart for future use.
- View Module: The administrator will have a list of valid user, food owner and lighting owner that have been already registered to the system.
- View cart Module: This module is used to show the stored food-items to the user from his cart.
- Search food-item Module: This module is used by the user to know about a particular food item.. User can search food items based on their category.
- Order module: This module allows the end user to place an order to the food owner.
- Order Status Module: This module is used to show the order status to the user. After placing the order, user can verify whether the order has accepted or rejected by the food owner.
- Add Lighting equipment: This module is used by the light owner to add different types of lighting devices to system.
- View Lighting Order: This is used to display lighting orders from end user to lighting owner.

1.8. Software / Hardware used for the development

- Operating System : Windows 10/ Linux
- Web Server : XAMPP
- Programming Languages : PHP, CSS, Java Script
- Web browsers : Google chrome/Mozilla Firefox/internet explorer
- Database : MYSQL

1.9. Software / Hardware required for the implimentation

1.9.1. Hardware Requirements:

Processor	:1 GHZ or higher CPU
Hard disk	: 500 MB available internal storage
Memory	:512 MB of RAM is minimally recommended
Display	: 2.8 inches or larger

1.9.2. Software Requirements:

Language	:	PHP 7.1
Web components	:	HTML 5
Web Server	:	XAMPP
Data base (backend)	:	MySQL

Chapter-2

SOFTWARE REQUIREMENT AND SPECIFICATIONS

2.1 Introduction:

SRS stands for Software Requirements Specification, which is a document that fully describes the expected behaviour of a software system. Functional requirements are documented in an SRS, as are non-functional requirements such as performance goals and descriptions of quality attributes. A software requirements specification (SRS) is a description of a software system to be developed. The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide to the user for perfect interaction.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market-driven project, these roles may be played by the marketing and development divisions).

2.2 Overall Description

2.2.1 Product Perspective:

The catering industry in the country is finally extending its reach beyond marriages. Till a few years ago, the members of the Indian families usually cooked during social gatherings. Now they are hiring caterers for the same. For ensuring quality standards, such people hire professional catering services. Moreover, apart from the social gatherings and festive occasions, the formal events, seminars, conferences and other business-related events consider professional catering services.

2.2.2 Product Functions:

Systems functionalities can be explained as follows.

User Registration

User of the system must register themselves to the system. System takes following data during user registration

- Name
- Username
- Password
- Mobile Number
- Email Id
- Address
- Gender

Above registration forms should be validated as below.

- All field should contain values
- Email ID should be given in correct form.
- Mobile number should be unique and should contain 10 digits.

Login

This is used to login into the system. User and administrator should provide their credentials. It should contain following fields

- Username
- Password

Form is validated as below.

- Both fields should contain data.
- If credentials are correct home page should be displayed and login page otherwise.

Food Module

This module is used to add, delete and update and view food information. Food add form should contain following fields.

- Food name
- Food type
- Food description
- Cost

Form is validated as below.

- All fields should contain data.
- Confirmation message should be displayed after adding test data successfully.

User Manager

This module is used to add, delete and update and view user information. View module of this manager should contain following fields.

- Edit
- Delete

Order Manager

This module is used to manage orders. This module receives user orders and update the user with owner confirmation.

2.2.3 User Characteristics:

Catering Management System is a web-based application project that helps the catering owner to manage his customer and business. It includes following end users.

Administrator

The administrator will add food related information into the database after logging in to the website. The food data are classified under various sections such as breakfast item, lunch and dinner. Administrator can update the food information based on the modifications in terms of food price. Administrator can also manage user and generate bills by applying the available offers to the bill.

User

Users are the customers who orders food items for their functions. User need to register the system by providing basic information. After the successful login, user can search food of their choice and add selected food into cart. Finally, they can order the foods from their cart and specify the delivery address.

Decorators

The decorators will add package related information into the database after logging in to the website. Decorators can update the package information based on the modifications in terms of package price. Also, they can see user orders.

Lighting

The lighting owners will add package related information into the database after logging in to the website. Lighting owners can update the package information based on the modifications in terms of package price. Also, they can see user orders.

2.2.4 General Constraints:

The following are the general constraints

- System will not allow unauthorized access.
- Username must be the Email ID or Username of the user.
- Password must be of minimum 8 characters.
- System must be accessible from anywhere through internet.
- The privacy of the users must be protected against unauthorized use.
- Every user must be given a unique identification number.
- Only administrator can control food add and deletion.

2.2.5 Assumptions and Dependencies:

The user should have basic knowledge of computers. They must be trained well to handle the features provided by the system. Some of the details are required to be entered by the user and may not be generated automatically. The project depends on the user’s ability to understand the features of the “catering” application and able to use the best of his or her use.

2.3 Special Requirements

2.3.1 User Interfaces

- Web Application: Admin will use web application to login and communicate with system. System has well defined, user friendly interfaces that allow the admin to operate the system very easily.
- System provides attractive and well-defined interfaces to users.

2.3.1 Hardware Interfaces

• Processor	:1 GHZ or higher CPU
• Hard disk	: 500 MB available internal storage
• Memory	:512 MB of RAM is minimally recommended
• Display	: 2.8 inches or larger

2.3.3 Software Requirements:

• Operating System	: Windows 7 or above
• Web Server	: XAMPP
• Programming Languages	: PHP, . CSS, Java Script
• Web browsers	: Google chrome/Mozilla Firefox/internet explorer
• Database	: MYSQL

2.4 Functional Requirements

This system has Five main Modules: Caterer, Lighting, Decorator, User and Admin

2.4.1 Caterer Module

- Login Module: This module is used to login into the system.
- Registration module: This module is used to register a caterer to the system.
- Add food-item Module: The caterer can add food-item into the database after logging in to the website.

- View Order: This is to view orders from users.
- View/Update Food: This is used to view or update food.
- Logout: To logout from system

2.4.2 Lighting Module

- Login Module: This module is used to login into the system.
- Registration module: This module is used to register a light owner to the system.
- Add package Module: The light owner can add packages into the database after logging in to the system.
- View Order: This is to view orders from users.
- View/Update Food: This is used to view or update packages.
- Logout: To logout from system

2.4.3 Decorator Module

- Login Module: This module is used to login into the system.
- Registration module: This module is used to register a decorator to the system.
- Add package Module: The decorator can add packages into the database after logging in to the system.
- View Order: This is to view orders from users.
- View/Update Food: This is used to view or update packages.
- Logout: To logout from system

2.4.4 User Module

- Login Module: This module is used to login into the system.
- Registration module: This module is used to register a user to the system.
- View food items: To view all food items added by caterer.
- View lighting packages: To view all packages added by lighting owners.
- View decorator: To view decoration packages added by decorator.
- Oder module: To order food, lighting and decorators for the event.
- Logout: To logout from system

2.4.5 Admin Module

- Login Module: This module is used to login into the system.
- User management: To manage all users of system.
- Decorator management: To manage all decorators of the system.
- Lighting management: To manage all light owner of the system.

2.5 Design Constraints

System has following design and implementation Constraints

- System should allow the user to login to use it only after their registration.
- System should allow admin to update the system with new food and category.
- System will display list of food based on the category selected by the end user.

2.6 System Attributes

The quality of website is maintained in such a way so that it can be very user friendly to all the users of the website.

- **Security:**

  Security of the application is maintained by giving the privilege of managing the application to administrator. The user will not have the provision of managing the application by himself. User can just use the application.

- **Reliability:**

  The application must provide reliable and correct result to the user.

- **Re usability:**

  With the use of this application, the developers can enhance it with other improved or additional features.

- **Maintainability:**

  This system can update and add new functionality whenever required because it is designed in a modular way.

- **Portability:**

  This system can be run in any operating system and browser.

2.7 Other Requirements

2.7.1 Performance Requirements

- Usability: Software should be usable, without any effort and it should have appropriate user interface.
- Response Time: Software should response within the estimated time to display results to the user.

- Software development life cycle: Here agile method is used which combines the advantages of waterfall approach and iterative model.

2.7.2 Safety Requirements

- People other than the registered user should not be able to process product reviews.
- Only the respective user can change their details.
- Authentication is done during the login.
- Whenever user has to edit his profile the UI prompts to enter the user password.
- The system searches for the email id and mobile number during add user. If user Already Exists With Same Email Id And Mobile Number, Then A Message Is Shown.

2.7.3 Software Quality Attributes

- Performance: The database used here is robust, reliable & fast. So, users will have to wait for the output very short time.
- Reliability: There is no case of redundancy in the database so it will not take extra memory space.
- Availability: System can be made use of at any time.
- Maintainability: Maintenance is easy and economical.

Chapter-3

SYSTEM DESIGN

3.1 Introduction

System design is the process of defining the architecture, modules, interfaces and data for a system to satisfy specified requirements. System design could be seen as the application of systems theory to product development. System design implies a systematic approach to the design of a system. It may take a bottom-up or top-down approach, but either way the process is systematic wherein it takes into account all related variables of the system that needs to be created from a architecture, to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system.

3.2 Assumptions and Constraints

3.2.1 Assumptions

One assumption about the product is that it will always be used on systems that have enough performance. If the system does not have enough hardware resources available for the application there may be scenarios where the application does not work as intended or even at all.

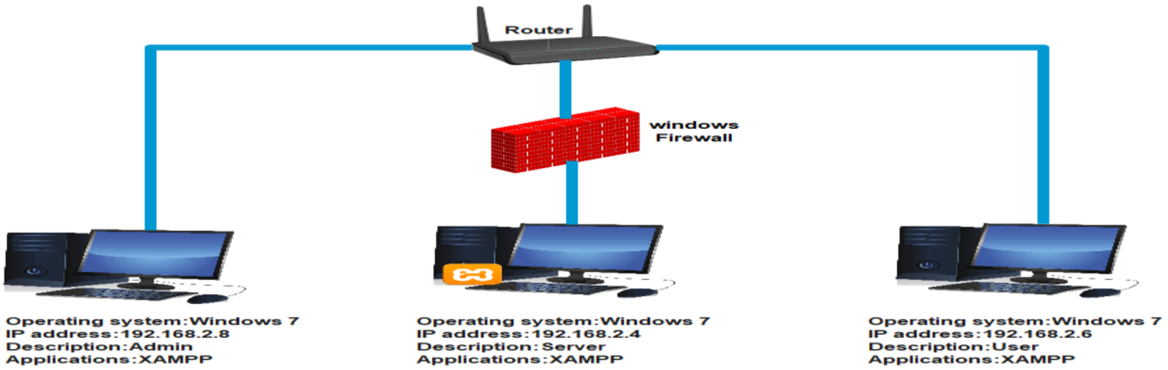
3.2.2 Constraints

- The developed system should run under any platform i.e., unix, linux, mac, windows etc.
- All mandatory field should be filled by an individual.
- Details provided by the individual during his signup should be stored in database.
- More space is required to keep all the records. Database should not be overloaded.
- Tables of the database are design normalized table.

3.3 Functional Decomposition

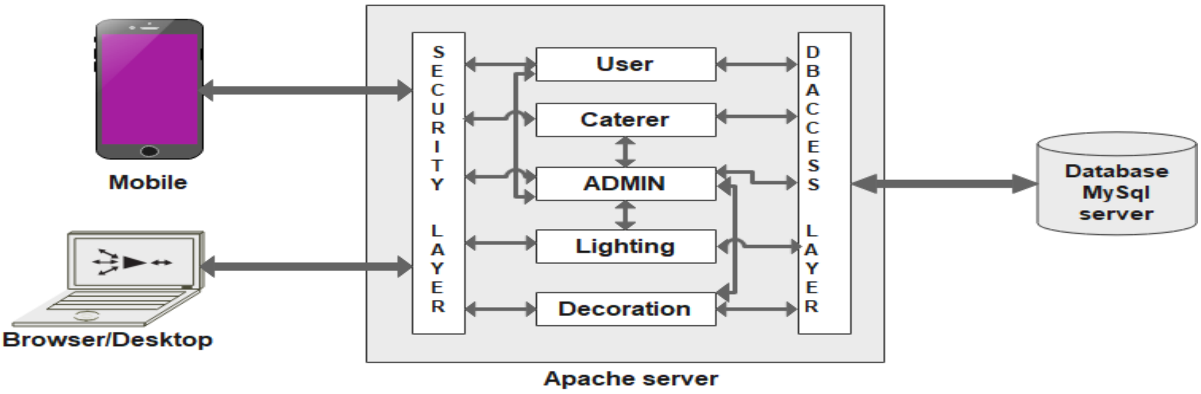
3.3.1 System Software Architecture

The software architecture of a software system is the set of structures needed to reason about the system, which comprise software components, relations among them, and properties of both.



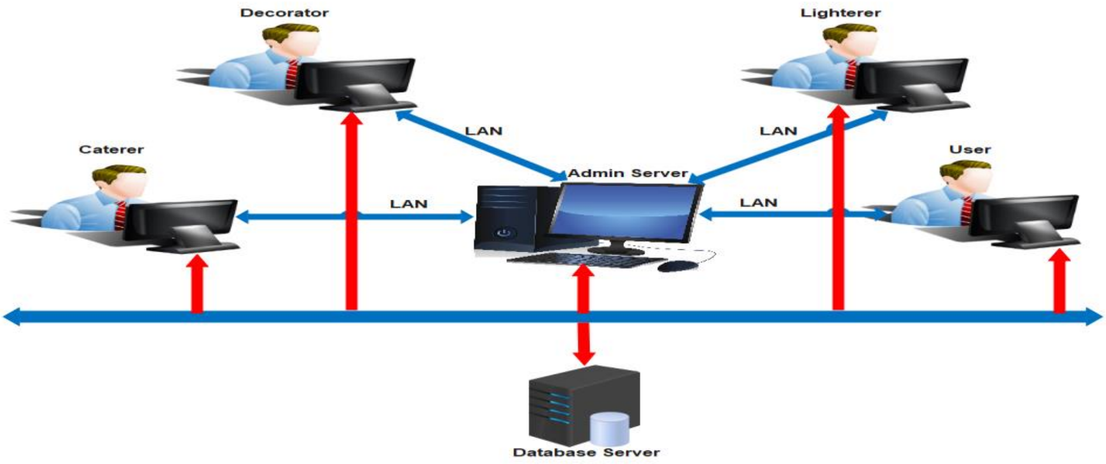
3.3.2 System Technical Architecture

Hardware architecture refers to the identification of a system's physical components and their interrelationships. This description often called a hardware design module, allow hardware designers to understand how their components fit into system architecture and provide software component designer important information needed for software development and integration



3.3.3 System Hardware Architecture

Hardware architecture refers to the identification of a system's physical components and their interrelationships. This description often called a hardware design module, allow hardware designers to understand how their components fit into system architecture and provide software component designer important information needed for software development and integration.



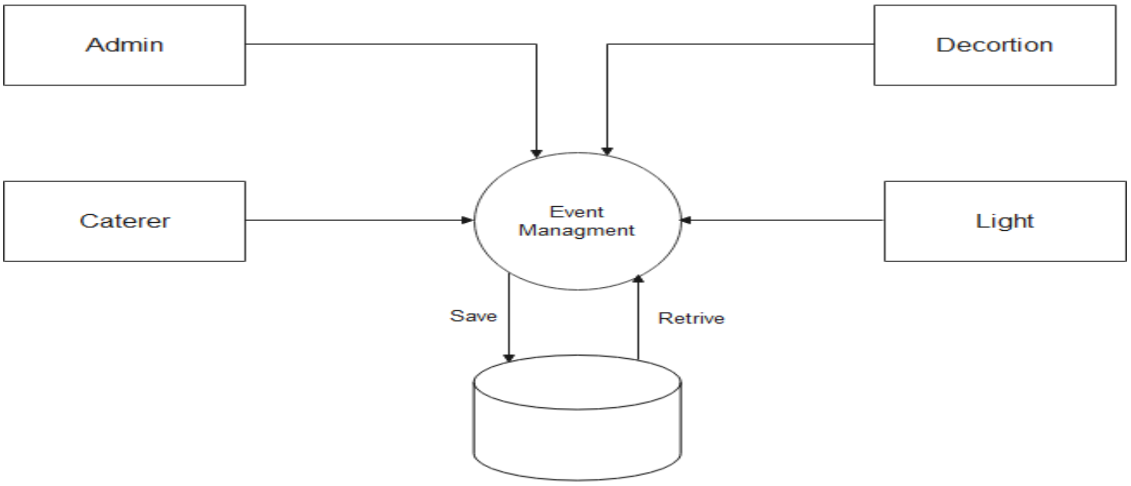
3.3.4 External Interfaces

- Name of the application:
- Details of Interface: Admin panel, Caterer panel, Decorator panel, Lighting panel, user panel.
- Type of Interface: Form based interface, Menu driven Interface.

3.4 Description of programs

3.4.1 Context Flow Diagram

Context flow diagram is top level data flow diagram. It only contains one process node that generalizes the function of the entire system in relationship to external entities. In context diagram entire system is shown as a single process and all its inputs, outputs, sinks and sources are identified and shown.



3.4.2 Data Flow Diagrams (DFDs-Level 0, Level 1, Level 2)

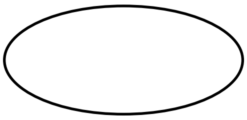
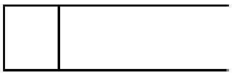
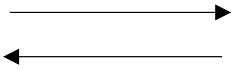
Data Flow Diagram:

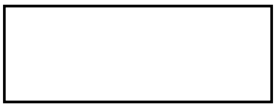
Data Flow Diagram is a graphical representation of a system or a portion of the system. It consists of data flows, process, sources and sink and stores all the description through the use of easily understandable symbols. DFD is one of the most important modeling tools. It is used to model the system, components that interact with the system, uses the data and information flows in the system. DFD shows the information moves through the and how it is modified by a series of transformations

Rules Regarding DFD Construction:

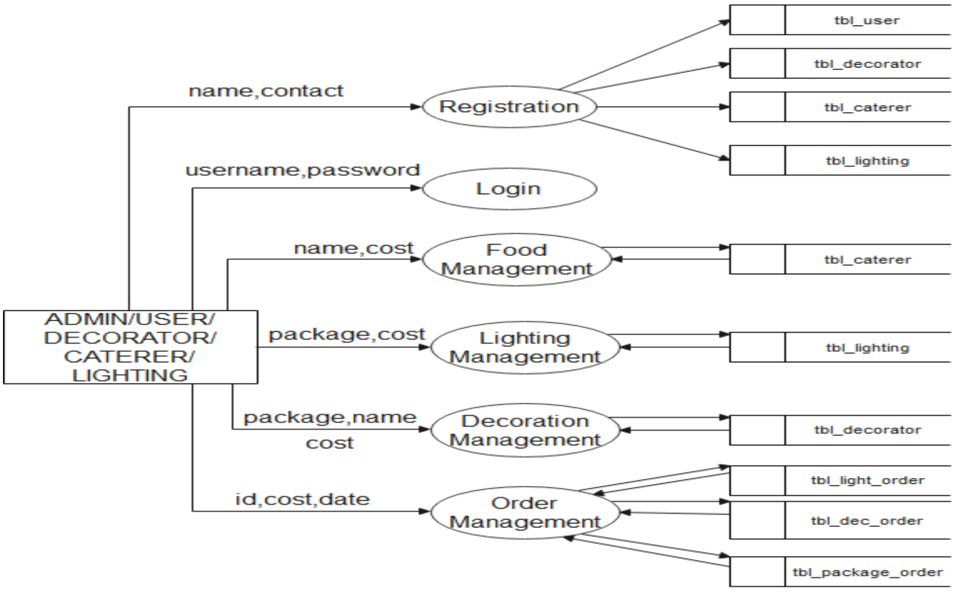
- A process cannot have only outputs.
- A process cannot have only inputs.
- The inputs to a process must be sufficient to produce the outputs from the process.
- A data flow can have only one direction of flow. Multiple data flows to and/or from the same process and data store must be shown by separate arrows.
- Data cannot flow directly back into the process it has just left. All data flows must be named using a noun phrase.

DFD Symbols:

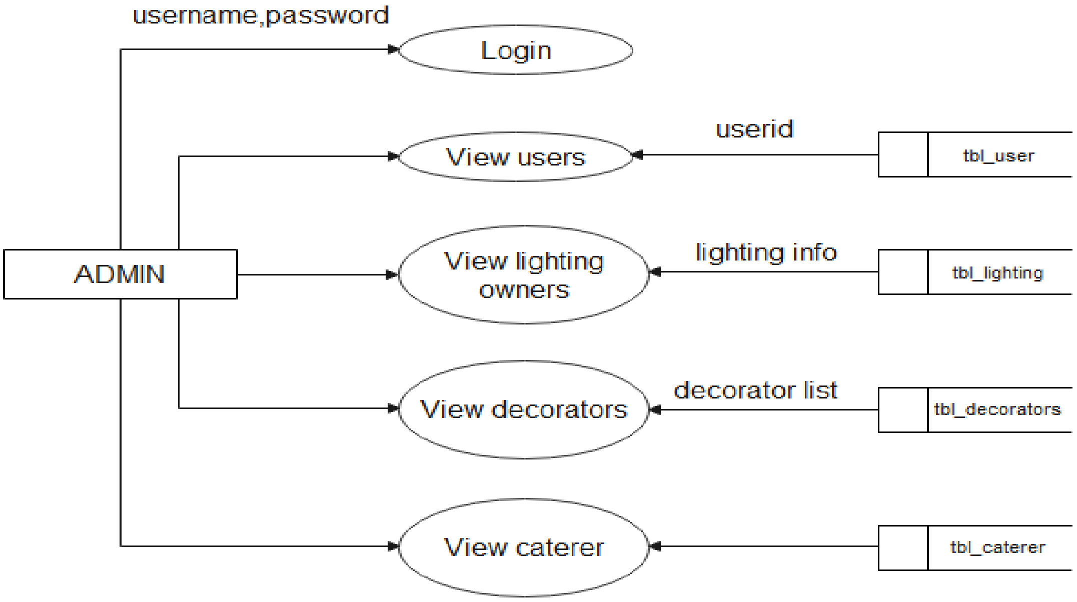
Name	Notation	Description
Process		A process transforms incoming data flows into outgoing data flows. The processes are shown by named circles.
Data store		Data stores are repositories of data in the system. They are sometimes also referred to as files
Data flows		Data flow are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.

External Entity		External entities are objects outside the system with which the system communicates. External entities are sources and destinations of the system's inputs and outputs.

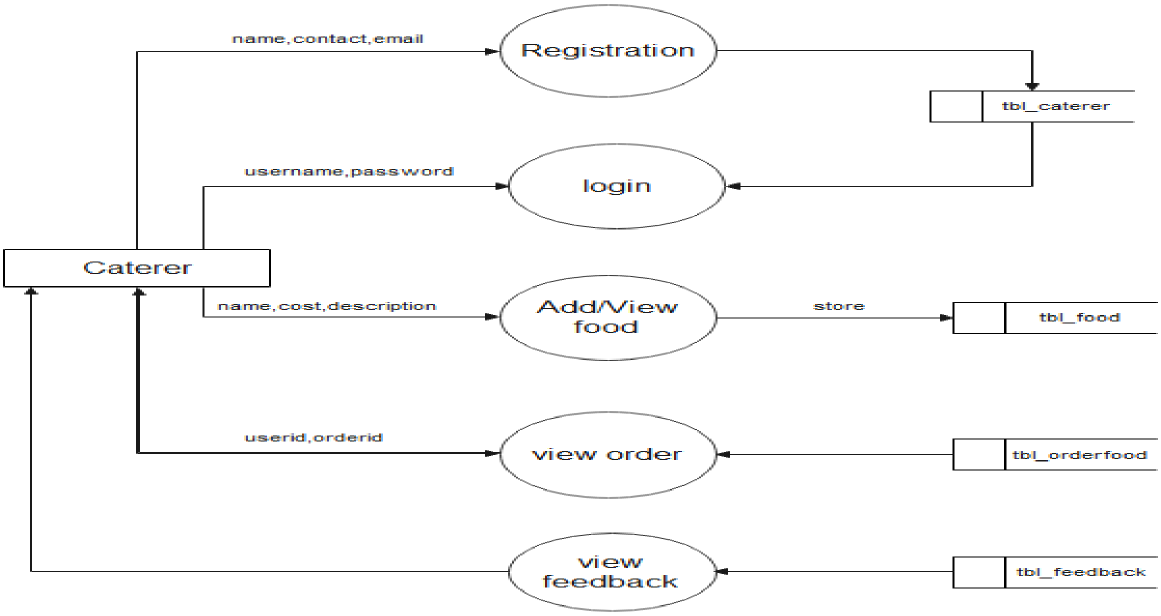
3.4.2.1 DFD Level 1



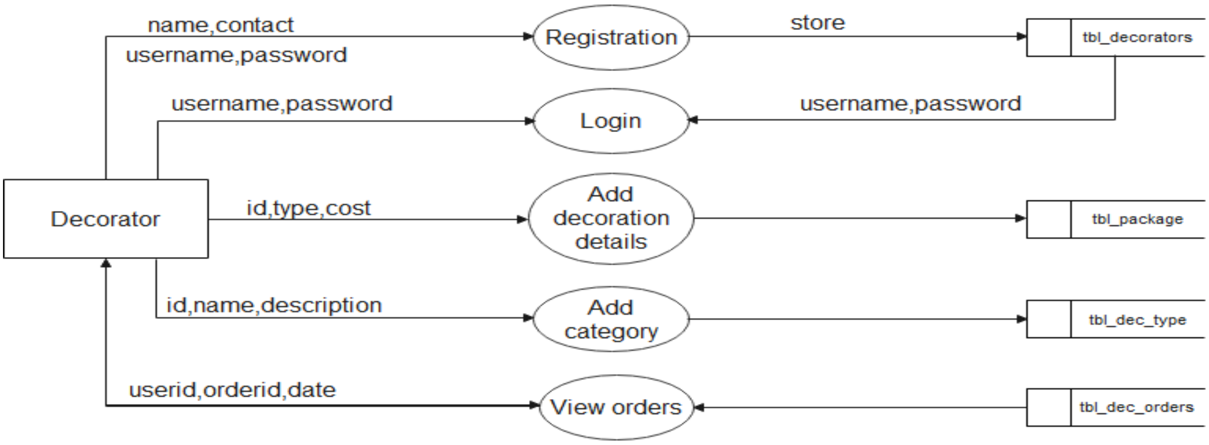
3.4.2.2 DFD Level 2 for Admin



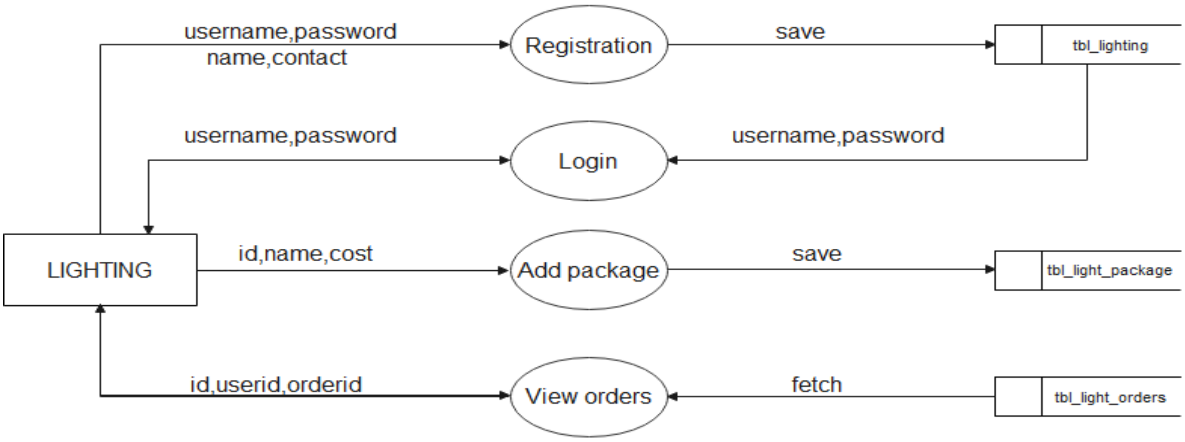
3.4.2.3 DFD Level 2 for Caterer



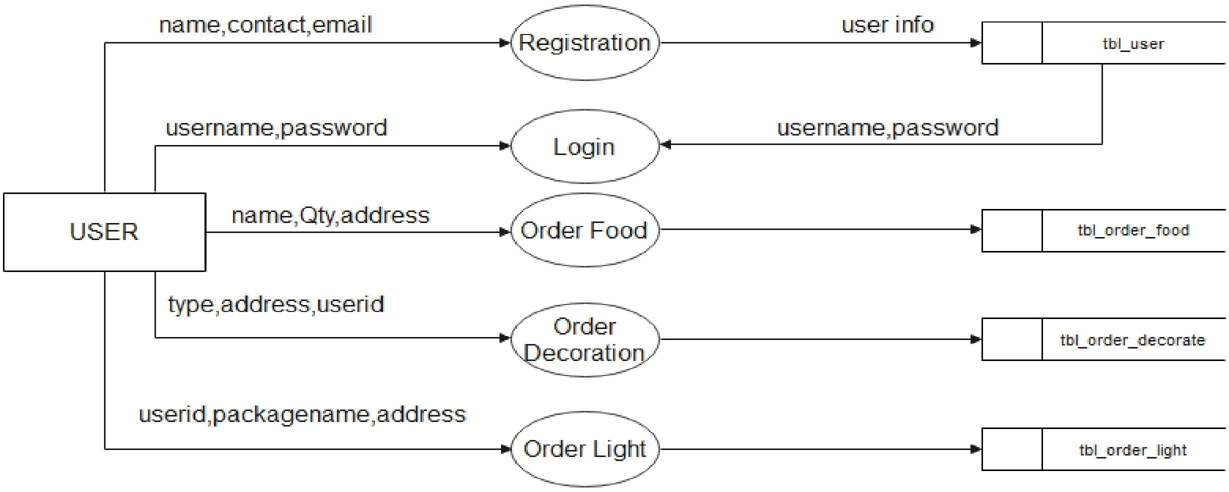
3.4.2.4 DFD Level 2 for Decorator



3.4.2.5 DFD Level 2 for Lighting



3.4.2.6 DFD Level 2 for User



3.5 Description of components

3.5.1 Functional components 1: Login

Input : Username, password.

Process : The username and password are verified to check its correctness.

Output : On successful login, home page is displayed.

3.5.2 Functional components 2: User

Input : Name, Password, confirm password, Email id, Gender, phone number, address.

Process : The users register themselves in order to gain access into the application.

Output : On entering valid input, the customer is successfully registered into the application and it creates his own profile

3.5.3 Functional components 3: Caterer

Input : Caterer name, password, confirm password, Email id, phone number, address, details.

Process : The Caterer and admin add and stores the caterer details, which can be viewed by the users.

Output : The Caterer details are displayed.

3.5.4 Functional components 4: Decorator

Input : Decorator name, password, confirm password, Email id, phone number, address, details.

Process : The Decorator and admin add and stores the Decorator details, which can be viewed by the users.

Output : The Decorator details are displayed.

3.5.5 Functional components 5: Lighting Provider

Input : Lighterer name, password, confirm password, Email id, phone number, address, details.

Process : The Lighterer and admin add and stores the lighterer details, we can be viewed by the users.

Output : The Lighterer details are displayed.

3.5.6 Functional components 6: Feedback

Input : Name, Email Id, Description.

Process : The users can give feedback. The data is stored in the feedback table.

Output : The feedback list is displayed.

3.5.7 Functional components 7: Payment

Input : Source of payment.

Process : The user can make payment online to the Caterer or decorator or lighterer

Output : The payment form is displayed.

3.5.8 Functional components 8: Add food

Input : Food type, food name, price, description

Process : The system will add food to the table

Output : The food added by the caterer are displayed to the user.

Chapter-4

DATABASE DESIGN

4.1 Introduction

The word “database” is used to describe everything from a single set of data, to a complex set of tools such as SQL server and a whole lot in between. The term data model defines how data is connected to each other and how they are processed and stored inside the system. This includes the definition of entities, their attributes and the entity constraints. The data model also includes a description of the relationship between entity and any constraints on those relationships. It is the translation of the conceptual model into a physical representation, which shall be implemented using a database management system.

4.2 Purpose and Scope

4.2.1 Purpose:

The benefits of a database design include its ability to handle huge volumes of data and multiple concurrent users. Unlike flat the systems, a DBMS maintains data integrity, consistency, security, and appreciable system performance. The database is used to store the following records.

4.2.2 Scope:

Managing information means taking care of it so that it works for us and is useful for the tasks we perform. By using a DBMS, the information we collect and add to its database is no longer subject to accidental disorganization. It becomes more accessible and integrated with the rest of our work. Managing information using a database allows us to become strategic users of the data we have.

4.3 Database Identification

- Database table name and column names defined without leaving space.
- Lower case used to create database tables name and columns names.
- Primary key and foreign key defined with same name. And the table in the database is as follows.

4.4 Schema Information

A Schema is a pictorial representation of the relationship between the database tables in the database that is created. The database schema of a database system is its structure described in a formal language supported by the database management (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed.. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language.

4.5 Table Structure

Table name: Users

Field Name	**Data Type**	**Constraint Type**
UserID	Integer	Primary Key
FName	Varchar2	Not Null
LName	Varchar2	Not Null
Email	Varchar2	Not Null
MobNo	BIGINT	Not Null
Password	Varchar2	Not Null
Address	Varchar2	Not Null

Table name: Food

Field Name	**Data Type**	**Constraint Type**
FoodID	Integer	Primary Key
FName	Varchar2	Not Null
FDesc	Varchar2	Not Null
FCate	Varchar2	Not Null
FType	Varchar2	Not Null
Price	Integer	Not Null

Table name: Cart

Field Name	Data Type	Constraint Type
CartID	Integer	Primary Key
UserID	Integer	Foreign Key
FoodID	Integer	Foreign Key
Quantity	Integer	Not Null

Table name: Order

Field Name	Data Type	Constraint Type
OrderID	Integer	Primary Key
UserID	Integer	Foreign Key
CartID	Integer	Foreign Key
TotalPrice	Integer	Not Null
DeliveryAdd	Varchar2	Not Null

Table name: Offer

Field Name	Data Type	Constraint Type
OfferID	Integer	Primary Key
OrderID	Integer	Foreign Key
Discount	Integer	Not Null
Date	Date	Not Null

Table name: Feedback

Field Name	Data Type	Constraint Type
FeedbackID	Integer	Primary Key
UserID	Integer	Foreign Key
Feedback	Varchar2	Not Null
Date	date	Not Null

Table name: Decoration_package

Field Name	Data Type	Constraint Type
ID	Integer	Primary Key
Name	Varchar2	Not Null
Type	Varchar2	Not Null

Description	Varchar2	Not Null
Price	integer	Not Null
Is_active	Integer	Not Null

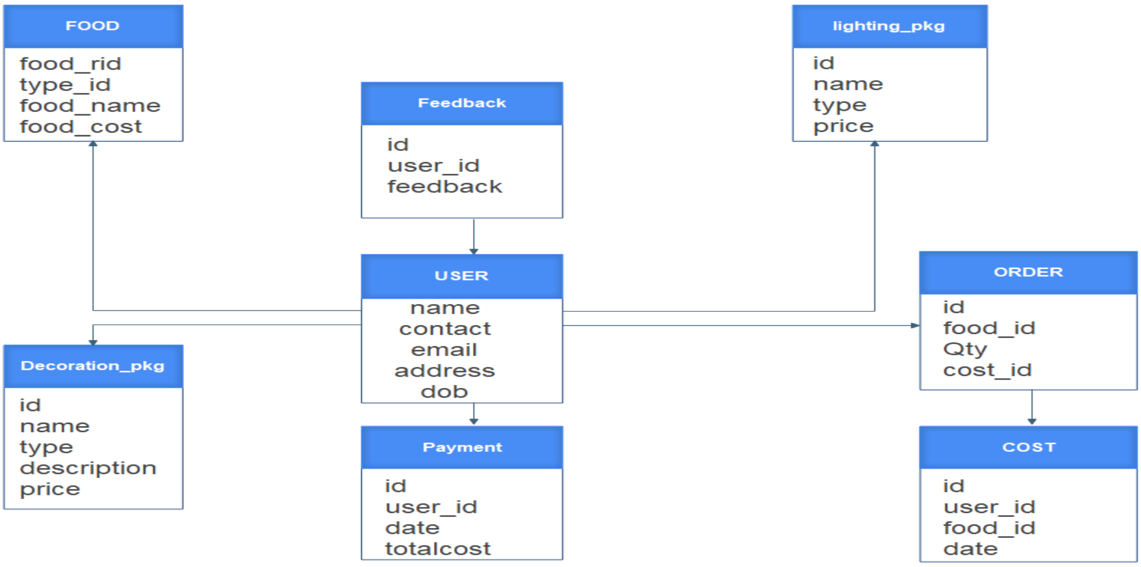
Table name: Lighting Package

Field Name	Data Type	Constraint Type
ID	Integer	Primary Key
Name	Varchar2	Not Null
Type	Varchar2	Not Null
Description	Varchar2	Not Null
Price	integer	Not Null
Is_active	Integer	Not Null

Table name: Payment

Field Name	Data Type	Constraint Type
payment_rid	Integer	Primary Key
payment_oh_id	Integer	Not Null
payment_userid	Integer	Not Null
payment_date	Date	Not Null
payment_status	integer	Not Null
payment_totcost	Integer	Not Null

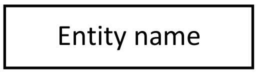
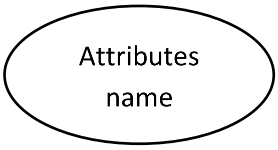
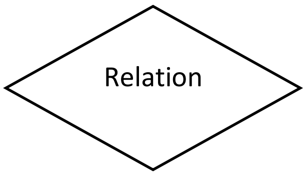
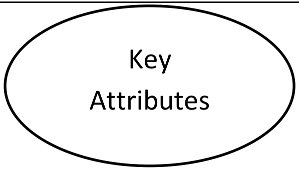
4.6 Physical Design



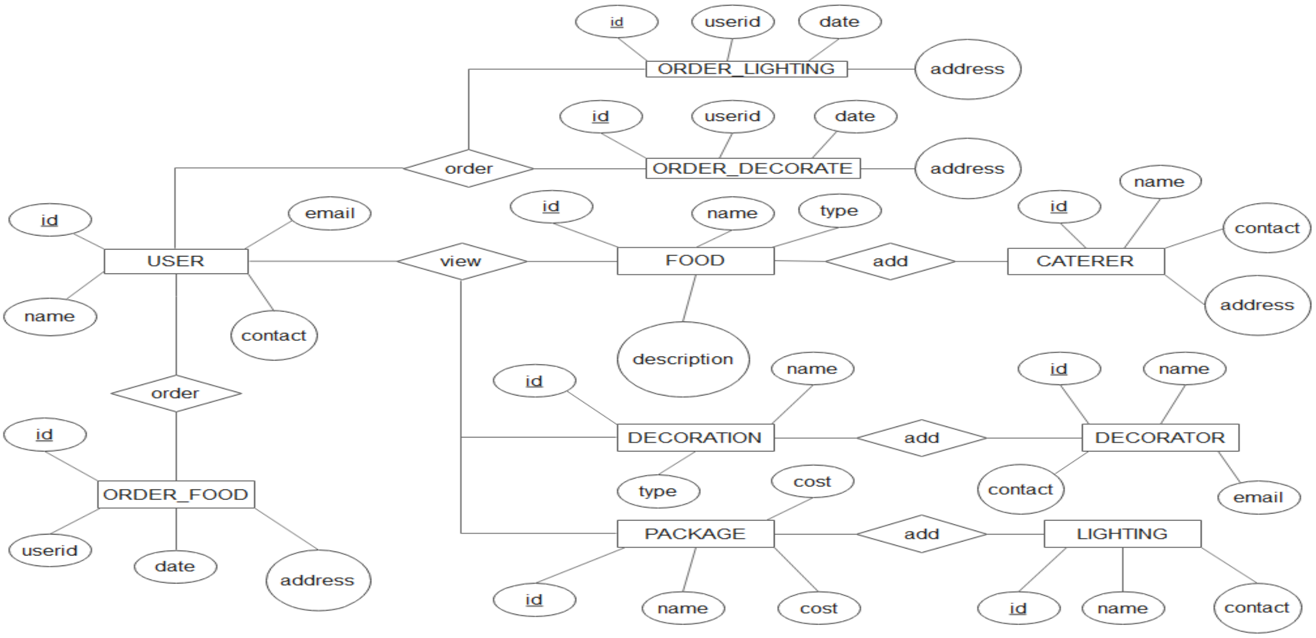
4.7 Entity-Relationship Diagram

The basic objective of the ER model representation is an entity which is a “thing” in a real word with an independent existence. Entities are physical items or aggregations of data items that are important to the business we analyze or to the system; we intend to build. An entity represents an object defined within the information system about which you want to store information. Entities are named as singular nouns and are shown in rectangles in an ER-Diagram.

ER-Diagram symbols

Name	Notation	Description
Entity		It may be an object with the physical existence or conceptual existence. It is represented by a Rectangle.
Attributes		The properties of the entity can be an attribute. It a represented by an Ellipse.
Relationship		Whenever an attribute of one entity refers to another entity, some relationship exists. It is represented by a diamond.
Link		Lines link attributes to entity sets and entity set to relation.
Key Attributes		An entity type usually has an attribute whose values are distinct for each individual entry in the entity set. It is represented by an underlined word in ellipse.

ER-Diagram:



4.8 Database Administrations

4.8.1 System Information

- **Server:** Localhost via TCP/IP
- **Server type:** MySQL
- **Server version:** 5.6.12-log-MySQL Community Server (GPL)
- **Protocol version:**10
- **User:** root@ localhost
- **Server charset:** UTF-8 Unicode(utf8)

4.8.2 DBMS configuration

Part of the DBMS installation process in the connection of the DBMS to other system software components that must interact with the DBMS. Typical infrastructure software that may need to be configured to work with the DBMS includes network, transaction processing monitors, message queues, other type of middleware, programming languages, systems Management software, operations and job control software, Web servers, and application servers.

4.8.3 Support software required

The system installs MYSQL server while installing XAMPP software. All the backup content stores in MYSQL data folder.

4.8.4 Storage requirements

- Storage for databases constraints a set of compatible software and hardware where database files are stored.
- Compatibility must be carefully tested on ensure eliminating bottlenecks and the possibility of data corruption.
- Failure free work in high load conditions and the redundancy of vulnerable components must be provided. Such a set must meet the highest standards in terms of performance and reliability, to ensure continuous and fast access to important data

4.8.5 Backup and Recovery

A volatile storage like RAM stores all the active logs, disk buffers, and related data. In addition, it stores all the transaction that are being currently executed. If such a volatile storage crashes abruptly, it would take away all the logs and active copies of the database. I make recovery almost impossible, as everything that is required to recover the data is lost.

Following techniques may be adopted in case of loss of volatile storage –

- We can have checkpoints at multiple stages so as to save the contents of the database periodically.
- A state of active database in the volatile memory can be periodically dumped into a stable storage, which may also contain logs and active transactions and buffer blocks.
- When the system recovers from a failure, it can restore the latest dump.
- It can maintain a redo-list and undo-list as checkpoints

Chapter-5

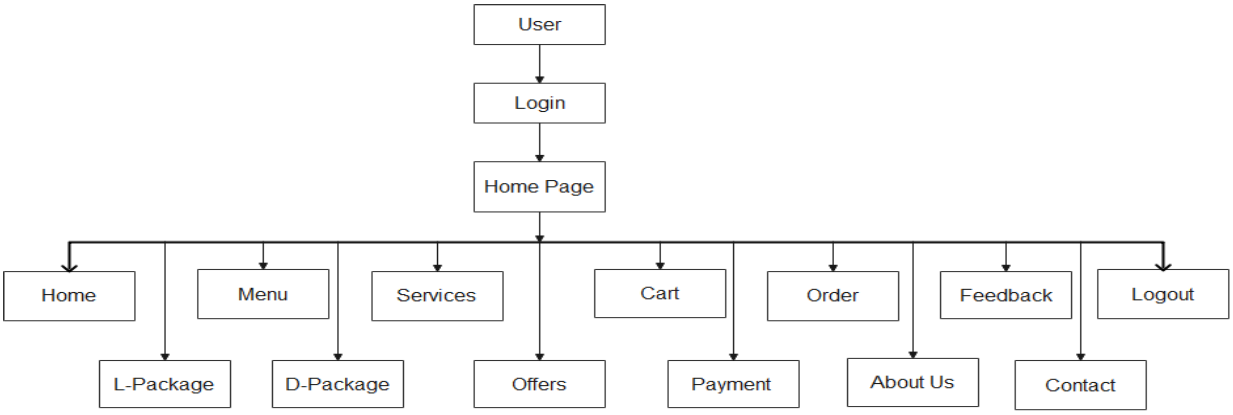
DETAILED DESIGN

5.1 Introduction

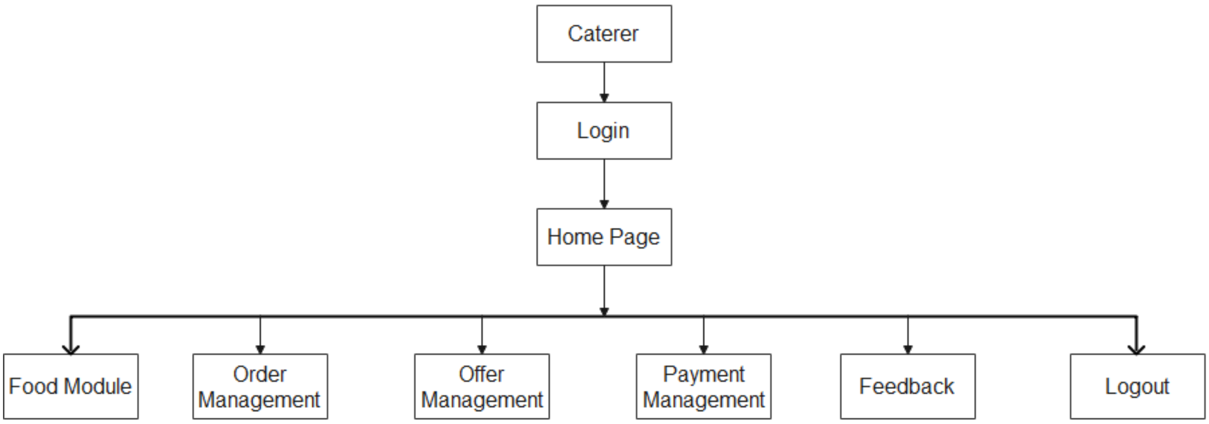
The purpose of the design phase is to plan a solution of the problem specified by the requirements document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed; design takes us towards how to satisfy the needs. The design of the system is perhaps the most critical factor affecting the quality of the software; it has a major impact on the later phases, particularly testing and maintenance.

5.2 Structure of Software Package

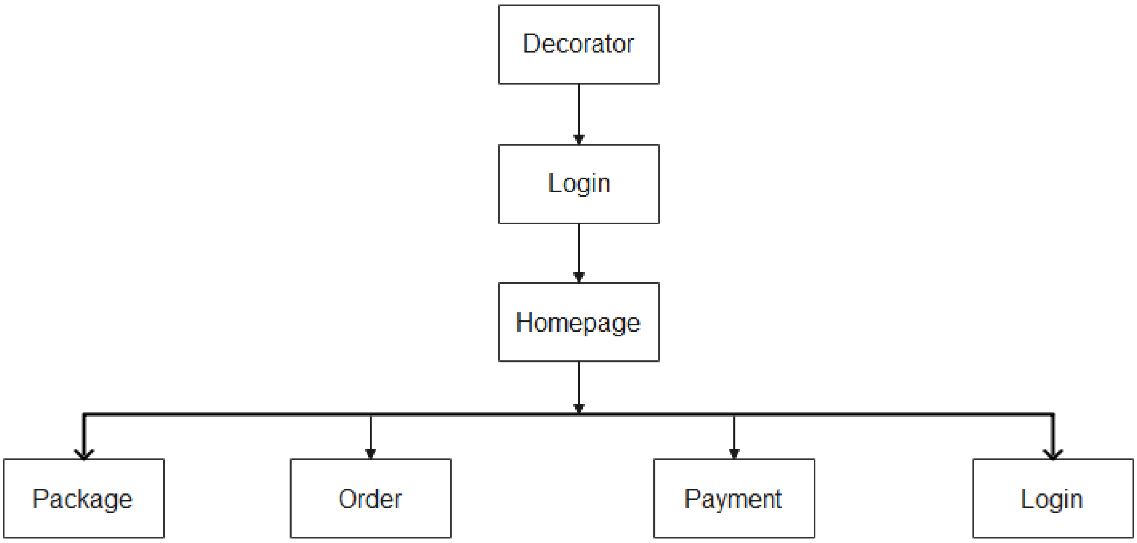
User Module:



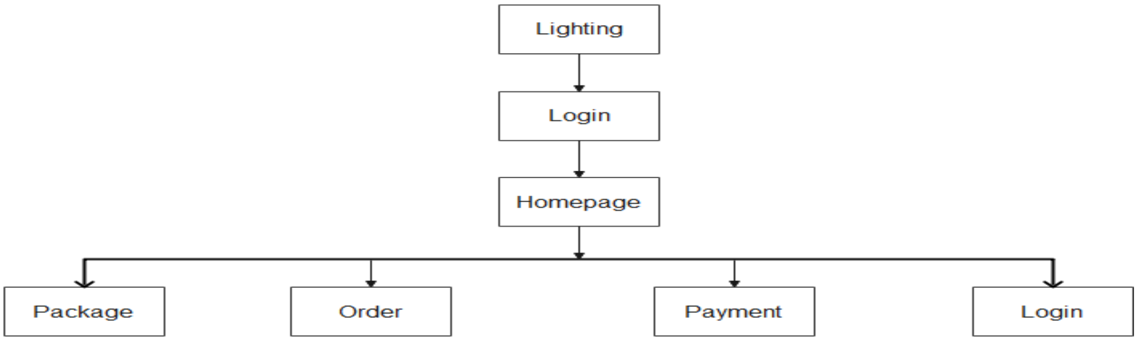
Caterer Module:



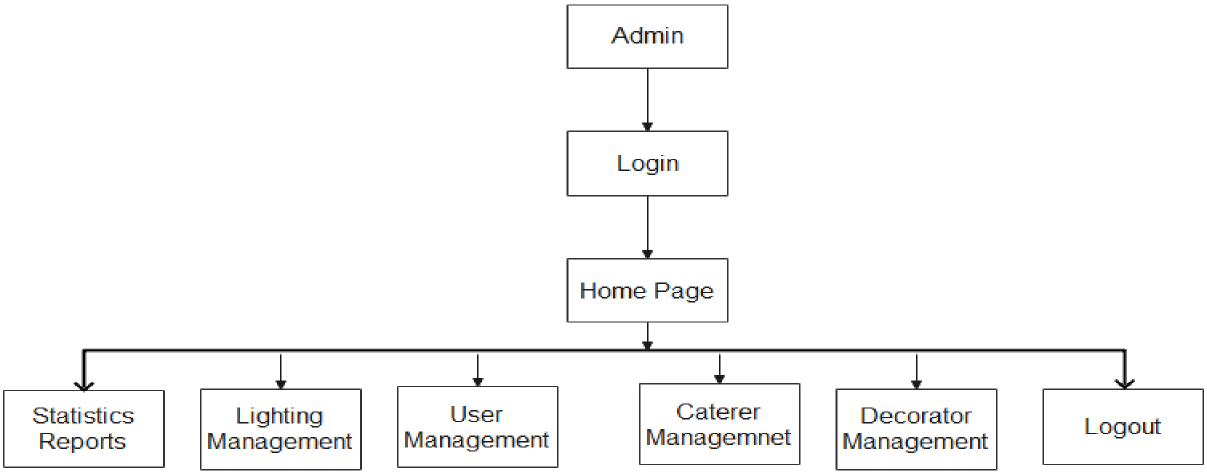
Decorater Module:



Lighting Module:



Admin Module:

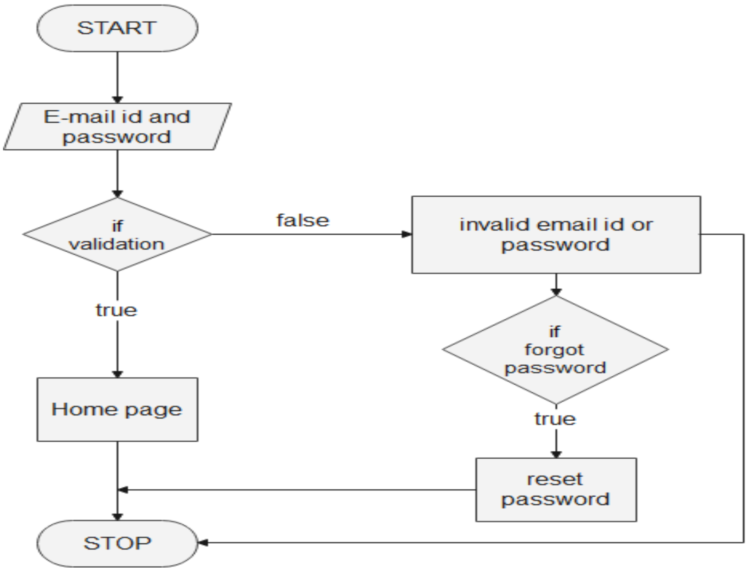


5.3 Modular Decomposition of Components

5.3.1 User

5.3.1.1 User Login

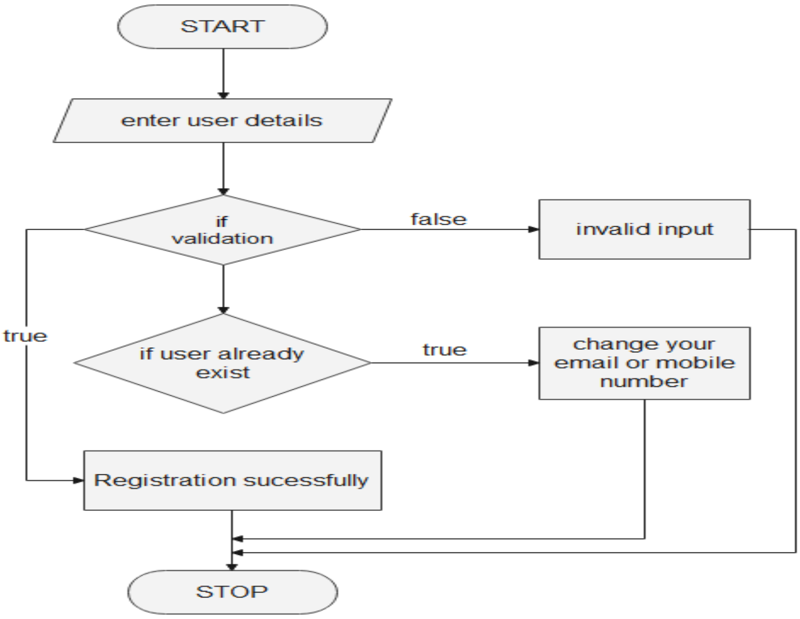
- **Input:** Email id, password
- **Procedure details**:



- **File I/O interface**: Graphical user interface to e-mail id and password. Data is stored in the user Table.
- **Output**: Entered email id and password will be checked for validity if it is valid user will be redirected to Homepage.

5.3.1.2 User Register

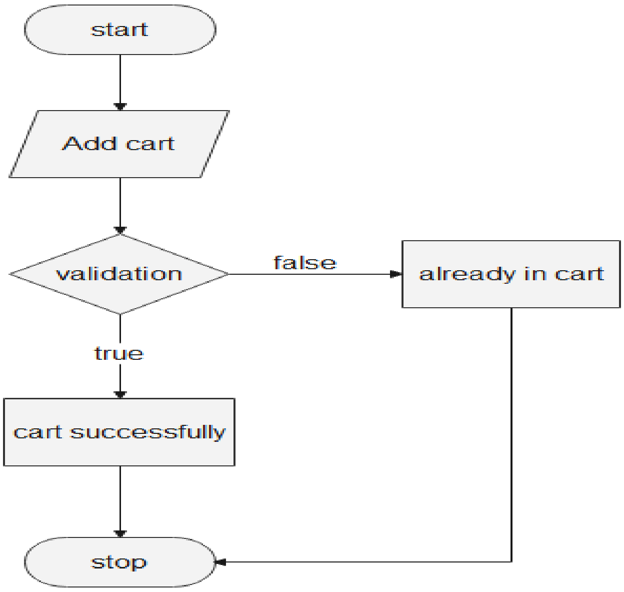
- **Input**: Name, Mobile No, Email, Address, Gender, Date of birth, Username, Password, Confirm-Password.
- **Procedure Details:**



- **File I/O interface**: The user has interface to their details and register themselves into the application.
- **Output:** The user is successfully registered and added to database.

5.3.1.3 User Cart

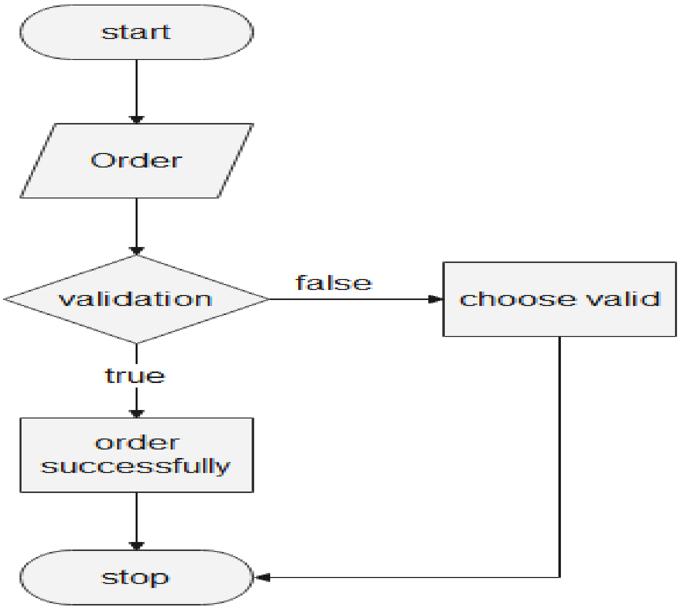
- **Input**: User id, food type.
- **Procedure Details:**



- **File I/O interface:** User has the interface to add food to cart.
- **Output**: The food is added and stored in the database. The data is stored in the cart table.

5.3.1.4 User Order

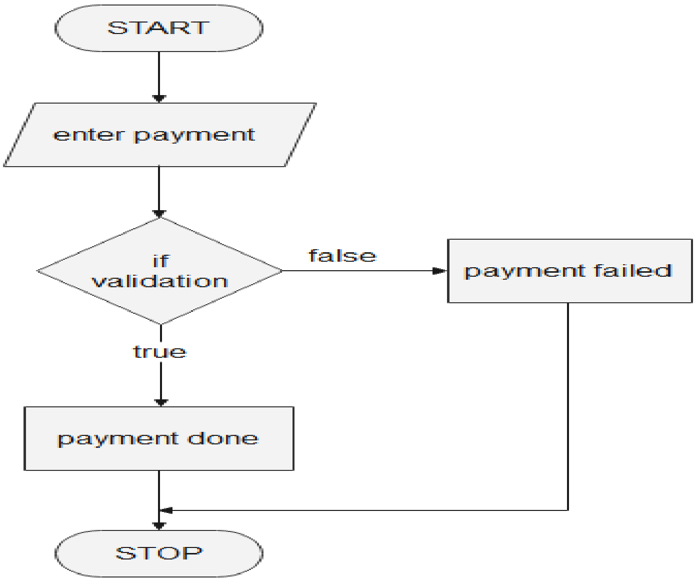
- **Input**: food id, cart id, Quantity, date, address
- **Procedure Details:**



- **File I/O interface:** User has the interface to order food.
- **Output**: The order is added and stored in the database. The data is stored in the order table.

5.3.1.5 User Payment

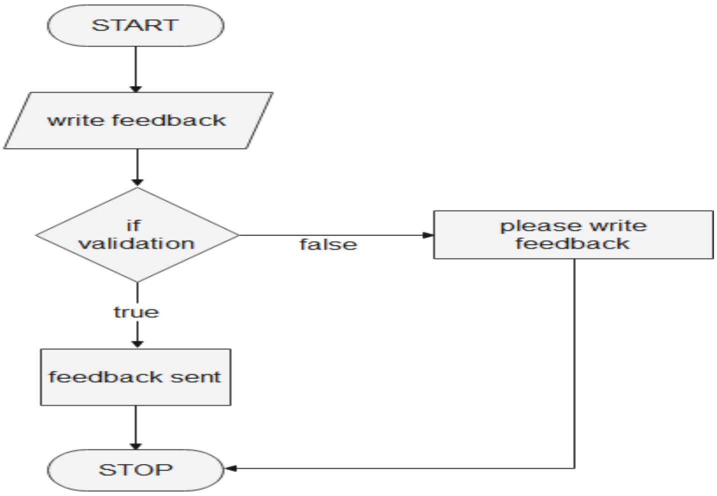
- **Input**: Click on payment
- **Procedure Details:**



- **File I/O interface:** User has the interface to do payment.
- **Output**: The payment is done and stored in the database. The data is stored in the payment table.

5.3.1.6 User Feedback

- **Input**: Click on write feedback
- **Procedure Details:**

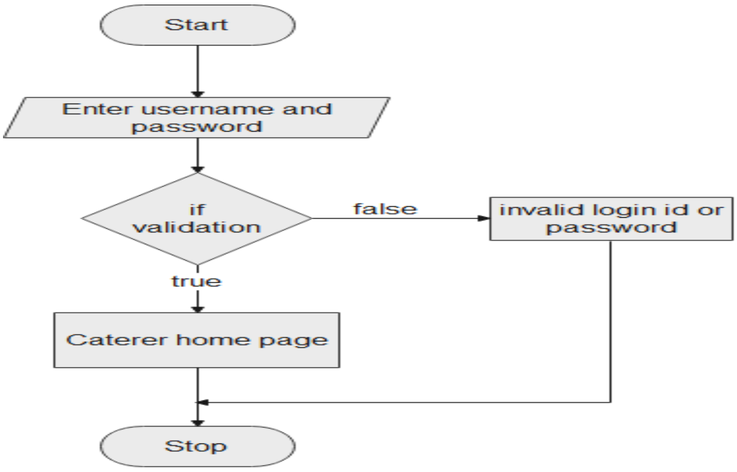


- **File I/O interface:** User has the interface to write feedback.
- **Output**: The write feedback is done and stored in the database. The data is stored in the feedback table.

5.3.2 Caterer

5.3.2.1 Caterer Login

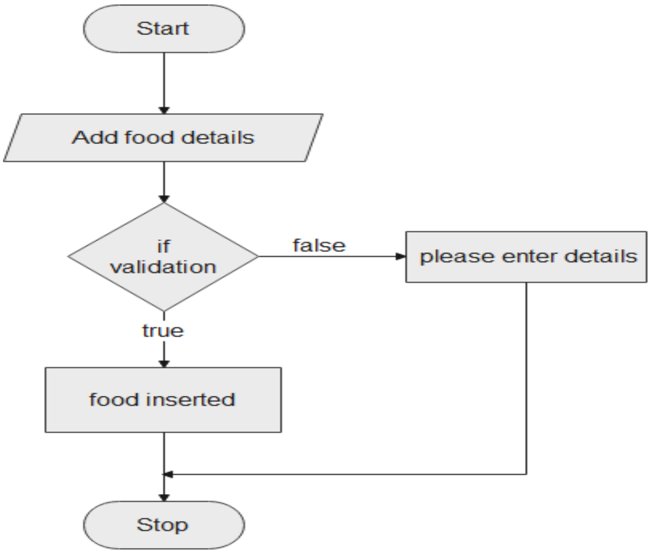
- **Input:** Username, password
- **Procedure details**:



- **File I/O interface:** Username and password. Data is stored in the caterer Table.
- **Output**: Entered username and password will be checked for validity if it is valid caterer will be redirected to caterer Homepage.

5.3.2.2 Caterer Add food

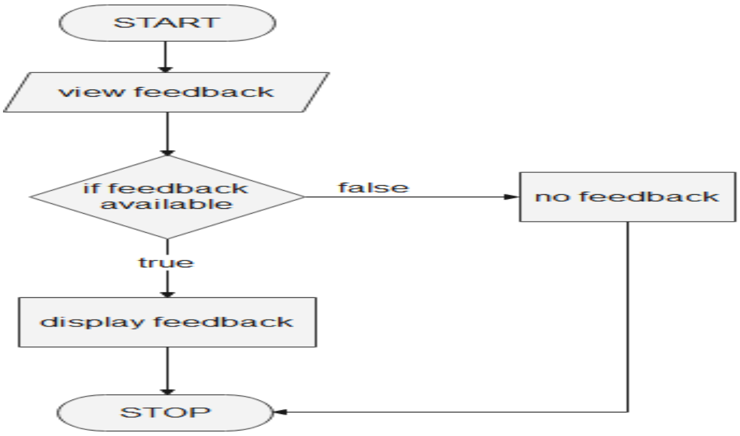
- **Input:** Food Type, Food Name, Food Description, Cost, Photo.
- **Procedure details**:



- **File I/O interface:** If the food details verified then caterer add the food.
- **Output**: Entered food details stored in database. The data is stored to food table

5.3.2.3 Caterer View Feedback

- **Input**: Click on view feedback
- **Procedure Details:**

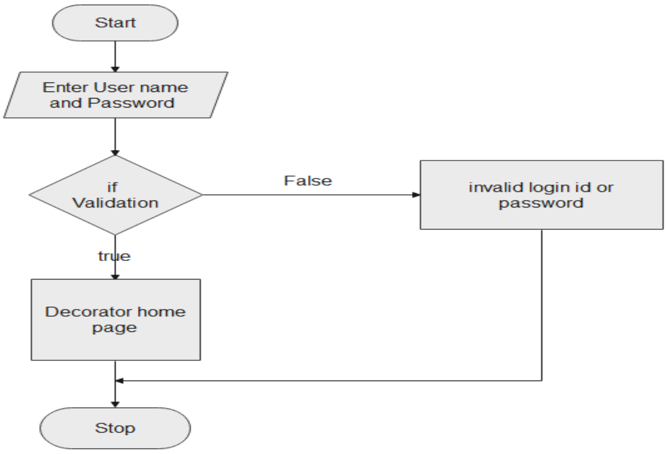


- **File I/O interface:** User has the interface to view feedback.
- **Output**: The view feedback is stored in the database. The data is stored in the view feedback table.

5.3.3 Decorator

5.3.3.1 Decorator Login

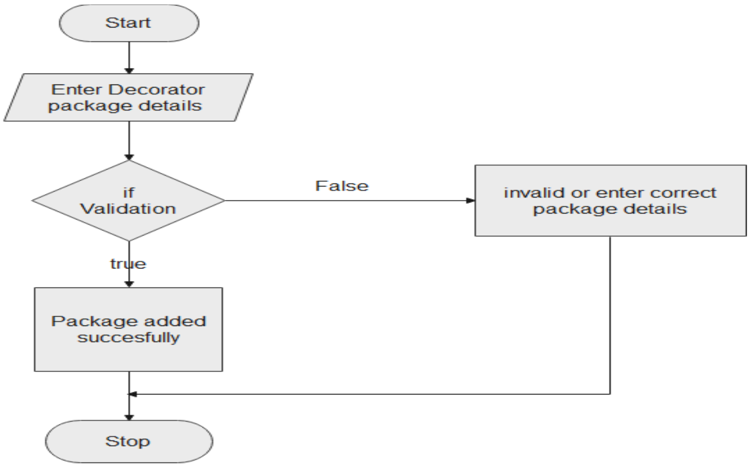
- **Input:** Username, password
- **Procedure details**:



- **File I/O interface:** Username and password. Data is stored in the Table.
- **Output**: Entered username and password will be checked for validity if it is valid decorator will be redirected to decorator Homepage

5.3.3.2 Decorator Add Package

- **Input**: Package name, Package type, decoration Description, Price
- **Procedure Details:**

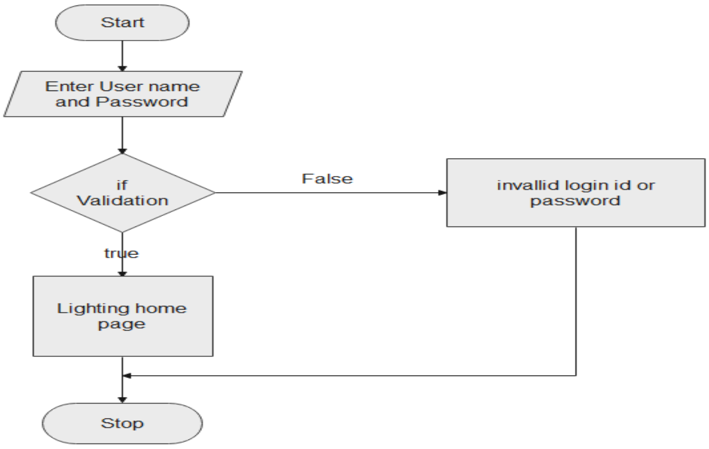

Start
Enter Decorator package details
if Validation
False
invalid or enter correct package details
true
Package added succesfully
Stop

- **File I/O interface:** If the package details verified then decorator adds the package
- **Output**: Entered package details stored in database. The data is stored to package table.

5.3.4 Lighting

5.3.4.1 Lighting Login

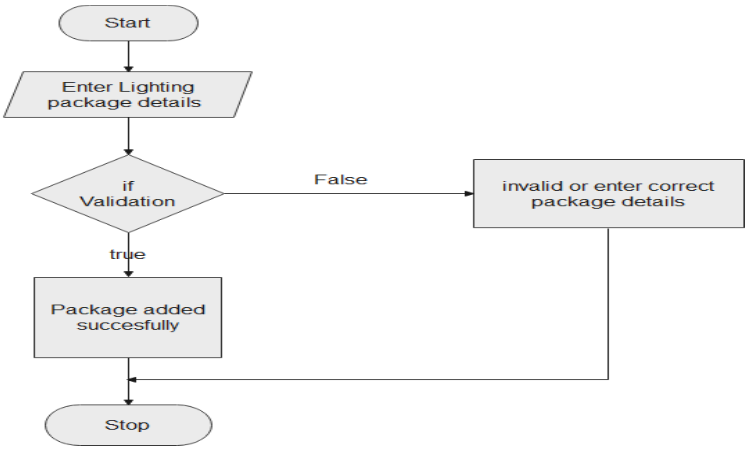
- **Input**: Username, password
- **Procedure Details:**



- **File I/O interface:** Username and password. Data is stored in the lighting Table.
- **Output**: Entered username and password will be checked for validity if it is valid lightning will be redirected to lighting Homepage

5.3.4.2 Lighting Add Package

- **Input**: Package name, Package type, lighting Description, Price
- **Procedure Details:**

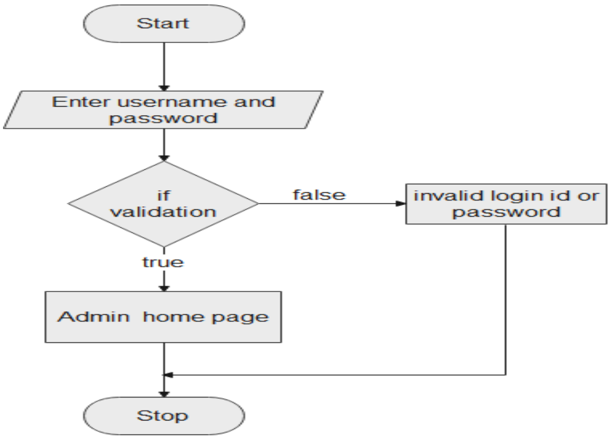

Start
Enter Lighting package details
if Validation
False
invalid or enter correct package details
true
Package added succesfully
Stop

- **File I/O interface** If the package details verified then lighting adds the package
- **Output**: Entered package details stored in database. The data is stored to package table.

5.3.5 Admin

5.3.5 Admin Login

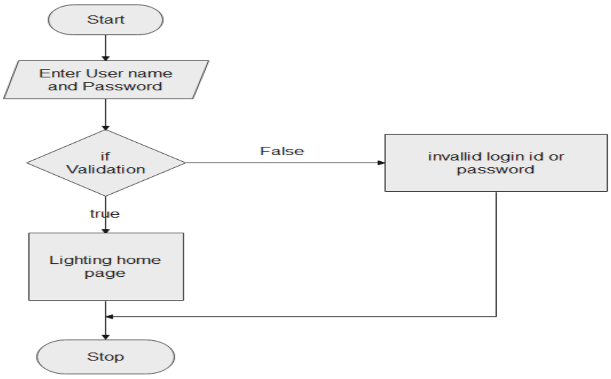
- **Input**: Username, password
- **Procedure Details:**



- **File I/O interface:** Username and password. Data is stored in the Admin Table.
- **Output**: Entered username and password will be checked for validity if it is valid Admin will be redirected to lighting Homepage.

5.3.5.2 view user

- **Input**: Username, password
- **Procedure Details:**



- **File I/O interface:** Username and password. Data is stored in the Admin Table.
- **Output**: Entered username and password will be checked for validity if it is valid Admin will be redirected to lighting Homepage.

Chapter-6

CODING

6.1 Database Connection

```
<?php
$hostname="localhost";
$username="root";
$password="";
$db="catering";
$link=mysqli_connect($hostname,$username,$password,$db) or die("error connecting db");
?>
```

6.2 Authorization/Authentication

6.2.1 login & Register form for User, Caterer, Decorator, Lighting and Admin:

- **User_Login.php**

```
<?php
session_start();
$error = "";
if (isset($_POST['submit'])) {
$username = $_POST['username'];
password = $_POST['password'];
require_once '../db/db.class.php';
$db = new DB();
$sql1 = "SELECT * FROM `user` WHERE user_email='$username' AND
user_pass='$password'";
$check_user = $db->executeSelect($sql1);
$sql2 = "SELECT * FROM `user` WHERE user_email='$username' AND
user_pass='$password' AND is_enabled=1";
$active_user = $db->executeSelect($sql2);

$sql3 = "SELECT * FROM `user` WHERE user_email='$username' AND
user_pass='$password' AND is_verified=1";
$valid_user = $db->executeSelect($sql3);
if (empty($check_user)) {
$_SESSION['userRid'] = 0;
$error = "Invalid login credentials!";
} else if (empty($active_user)) {
$_SESSION['userRid'] = 0;
$error = "Please contact admin!";
} else if (empty($valid_user)) {
$_SESSION['u_email'] = $active_user[0]['user_email'];
$_SESSION['u_password'] = $active_user[0]['user_pass'];
header('location: ../user/index2.php');
} else {
$_SESSION['userRid'] = $valid_user[0]['user_rid'];
header("location: ../user/user_home.php");
}}
?>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
<title></title>
</head>
<body class="user-logbody">
<div class="row">
<div class="col-md-4"></div>
<div class="col-md-4">
<div class="card card1 carduser">
<div class="card-body">
<h1 class="text-center">User Login</h1>
<form method="post" action="#" onsubmit="return validateuserlogin();">
<div class="form-group">
<label>Email ID</label>
<div class="border">
```

```
<input type="text" class="form-control" placeholder="Email ID" name="username"
id="username" autocomplete="off">
</div></div>
<div class="form-group">
<label>Password</label>
<div class="border">
<input type="password" class="form-control" placeholder="Password" name="password"
id="password" autocomplete="off">
</div></div>
<div class="form-group">
<?php
if (!empty($error)) {
?>
<div class="alert alert-danger alert-dismissible fade show"      role="alert">
<?php echo $error; ?>
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button></div>
<?php }
?>
</div>
<div class="form-group text-center">
<button type="submit" class="btn btn-primary " name="submit">Login</button>
</div>
<div class="form-group text-center">
<a href="../user/user_register.php">Create New Account</a>
<div class="form-group text-center">
<a href="#" data-toggle="modal" data-target="#userForgotPwdModal">Forgot
Password</a>
</div></div></form></div></div></div>
<div class="col-md-4"></div>
</div>
<div class="modal fade" id="userForgotPwdModal" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
```

```
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title">Forgot Password</h5>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button></div>
<div class="modal-body">
<form id="ForgotPwdForm" method="post" action="../actions/user_action.php">
<input type="hidden" name="command" value="postEmail" />
<div class="form-group">
<label>Enter Email</label>
<input type="email" class="form-control " id="frgtPwdEmailUser"
name="frgtPwdEmailUser" placeholder="Enter Email">
</div
<button type="submit" id="btnSubmitEmailUser" class="btn btn-primary">Submit</button>
</form></div></div></div></div>
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/popper.min.js"></script>
<script src="../assets/jquery-ui/jquery-ui.min.js"></script>
<script src="../assets/js/bootstrap.min.js"></script>
<script src="../assets/js/user.js"></script>
<script src="../assets/js/login.js"></script>
</body>
</html>
```

- **User_register.php**

```
<?php
require_once '../db/db.class.php';
$db = new DB();
?>
<html>
<head>
```

```
<meta charset="UTF-8">
<title>User Registration</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/jquery-ui/jquery-ui.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
<body>
<div class="container-fluid">
<div class="row justify-content-center">
<div class="col-md-5">
<div class="card" id="compCardview">
<form id="addUserForm" method="post" action="../actions/user_action.php">
<h3 class="text-center">Customer Registration</h3>
<div class="card-body">
<input type="hidden" name="command" value="addUser" />
<div class="form-group">
<label>Name</label>
<input type="text" autocomplete="off" class="form-control boxstyle" id="u_name"
name="u_name" placeholder="Enter name"></div>
<div class="form-group">
<label>Mobile No</label>
<input type="text" autocomplete="off" class="form-control boxstyle" id="u_mob"
name="u_mob" placeholder="Enter mobile no" maxlength="10"></div>
<div class="form-group">
<label>Email Id</label>
<input type="text" autocomplete="off" class="form-control boxstyle" id="u_email"
name="u_email" placeholder="Enter email"></div>
<div class="form-group">
<label>Address</label>
<textarea class="form-control boxstyle" autocomplete="off" id="u_add" name="u_add"
rows="5" placeholder="Enter Address"></textarea></div>
<div class="form-group">
<label>Gender</label>
<select class="form-control boxstyle" autocomplete="off" id="u_gender" name="u_gender">
```

```
<option value="male">Male</option>
<option value="female">Female</option>
</select></div>
<div class="form-group">
<label>Date Of Birth</label>
<input type="text" id="datepickerDOB"  autocomplete="off" class="form-control boxstyle"
name="u_date"></div>
<div class="form-group">
<label>Username</label>
<input type="text" class="form-control boxstyle" autocomplete="off" id="u_uname"
name="u_uname" placeholder="Enter Username"></div>
<div class="form-group">
<label>Confirm Password</label>
<input type="password" class="form-control boxstyle " autocomplete="off" id="u_con_pass"
name="u_con_pass" placeholder="Re-enter Password"></div>
<div class="form-group text-center">
<button type="submit" id="btnSubmitUser" class="btn btn-
primary">Submit</button></div>
<div class="form-group text-center">
<a href="../index.php">Back to Login</a>
</div></div></form></div></div></div></div
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/bootstrap.min.js"></script>
<script src="../assets/js/user.js"></script>
</body>
</html>
```

- **Caterer_login.php**

```
<?php
session_start();
require_once '../db/db.class.php';
$db = new DB();
$error = "";
```

```
if (isset($_POST['submit'])) {
$username = $_POST['username'];
$password = $_POST['password'];
$sql = "SELECT * FROM `cateror` WHERE u_name='$username' AND
u_pass='$username'";
$res = $db->executeSelect($sql);
$sql2 = "SELECT * FROM `cateror` WHERE u_name='$username' && u_pass='$username'
AND is_enabled=1";
$validaUser = $db->executeSelect($sql2);
if (!empty($res)) {
if (count($validaUser) > 0) {
$_SESSION['userRid'] = 1;
header("location: admin_home.php");
} else {
$error = "Please Contact Admin!";}
} else {
$_SESSION['userRid'] = 0;
$error = "Invalid login id or password";
}}
?>
<html>
<head>
<title></title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
<body class="logbody">
<div class="row">
<div class="col-md-4"></div>
<div class="card card1 cardback">
<div class="card-body">
<h1 class="text-center head">Caterer Login</h1>
<form method="post" action="#">
<div class="form-group mt-5 shadow-none">
```

```
<div class="border">
<input type="text" class="form-control" placeholder="Username" name="username"
autocomplete="off"></div></div>
<div class="form-group mt-5 shadow-none">
<div class="border"><input type="password" class="form-control" placeholder="Password"
name="password" autocomplete="off">
</div></div>
<div class="form-group">
<?php
if (!empty($error)) {
?>
<div class="alert alert-danger alert-dismissible fade show" role="alert">
<?php echo $error; ?>
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span></button></div>
<?php }
?>
</div>
<div class="form-group text-center mt-4">
<button type="submit" class="btn btn-dark px-5" name="submit">Submit</button></div>
<div class="form-group text-center">
<a href="cateror_signin.php">Sign Up?</a>
</div></form</div></div></div>
<div class="col-md-4"></div></div>
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/popper.min.js"></script>
<script src="../assets/js/bootstrap.min.js"></script>
</body>
</html>
```

- **decorator_login.php/lighting_login.php**

```
<?php
session_start();
```

```
require_once '../db/db.class.php';
$db = new DB();
$error = "";
if (isset($_POST['submit'])) {
$username = $_POST['username'];
$password = $_POST['password'];
$sql = "SELECT * FROM `decorator` WHERE u_name='$username' AND
u_pass='$password'";
$res = $db->executeSelect($sql);
$sql2 = "SELECT * FROM `decorator` WHERE u_name='$username' &&
u_pass='$username' AND is_enabled=1";
$validaUser = $db->executeSelect($sql2);
if (!empty($res)) {
if (count($validaUser) > 0) {
$_SESSION['userRid'] = 1;
header("location: decorator_home.php");
} else {
$error = "Please Contact Admin!";}
} else {
$_SESSION['userRid'] = 0;
$error = "Invalid login id or password";}}
?>
<html>
<head>
<title>Decorator Login</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
<body class="logbody">
<div class="row">
<div class="col-md-4"></div>
<div class="card card1 cardback">
<h1 class="text-center head">Decorator Login</h1>
<form method="post" action="#">
```

```
<div class="form-group mt-5 shadow-none">
<div class="border">
<input type="text"
class="form-control" placeholder="Username" name="username" autocomplete="off">
</div></div>
<div class="form-group mt-5 shadow-none">
<div class="border">
<input type="password" class="form-control" placeholder="Password" name="password"
autocomplete="off">
</div></div>
<div class="form-group">
<?php
if (!empty($error)) {?>
<div class="alert alert-danger alert-dismissible fade show" role="alert">
<?php echo $error; ?>
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button></div>
<?php }?></div>
<div class="form-group text-center mt-4">
<button type="submit" class="btn btn-dark px-5" name="submit">Submit</button></div>
<div class="form-group text-center">
<a href="decorator_signin.php">Sign Up?</a>
</div></form></div></div></div>
<div class="col-md-4"></div></div>
<script src="../assets/js/popper.min.js"></script>
<script src="../assets/js/bootstrap.min.js"></script>
</body>
</html>
```

- **Admin_login.php**

```
<?php
session_start();
```

```
$error = "";
if (isset($_POST['submit'])) {
$username = $_POST['username'];
$password = $_POST['password'];
if ('admin' == $username && 'admin' == $password) {
$_SESSION['userRid'] = 1;
header("location: admin_home.php");
} else {
$_SESSION['userRid'] = 0;
$error = "Invalid login id or password";}}?>
<html
<head>
<meta name="viewport" content="width=device-width, initial-scale=1"><title></title
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head
<body class="logbody">
<div class="row">
<div class="card-body">
<h1 class="text-center head">Admin Login</h1>
<form method="post" action="#">
<div class="form-group mt-5 shadow-none">
<div class="border">
<input type="text" class="form-control" placeholder="Username" name="username"
autocomplete="off"></div></div>
<div class="form-group mt-5 shadow-none"><div class="border">
<input type="password" class="form-control" placeholder="Password" name="password"
autocomplete="off"></div></div>
<div class="form-group">
<?php
if (!empty($error)) {
?>
<div class="alert alert-danger alert-dismissible fade show" role="alert">
<?php echo $error; ?>
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button></div>
<?php }?></div>
<div class="form-group text-center mt-4">
<button type="submit" class="btn btn-dark px-5" name="submit">Submit</button>
</div></form></div></div></div>
<div class="col-md-4"></div></div>
 <script src="../assets/js/jquery-3.3.1.min.js">
</script>
<script src="../assets/js/popper.min.js"></script>
<script src="../assets/js/bootstrap.min.js"></script>
</body>
</html>
```

6.3 Data store/ Retrieval/ Update

- **User_home.php**

```
<?php
require_once '../user/session.php';
?>
<html>
<head>
<meta charset="UTF-8">
<title>Home Page</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css" integrity anonymous" referrerpolicy="no-referrer" />
</head>
<body >
<?php
require_once './user_nav.php';
require_once './user_modal.php';
<div id="carouselExampleIndicators mt-0 pt-0" class="carousel slide" data-ride="carousel"
style="margin-left: 150px;">
<ol class="carousel-indicators">
<li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
<li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
<li data-target="#carouselExampleIndicators" data-slide-to="2"></li></ol>
<div class="carousel-inner">
<img src="../images/breakfast1.jpg" class="d-block img-fluid" alt="..."></div>
<div class="carousel-item">
<img src="../images/breakfast1.jpg" class="d-block img-fluid" alt="..."></div>
<div class="carousel-item">
<img src="../images/soup.jpg" class="d-block img-fluid" alt="..."></div></div>
<a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-
slide="prev">
<span class="carousel-control-prev-icon" aria-hidden="true"></span>
<span class="sr-only">Previous</span></a>
<a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-
slide="next">
<span class="carousel-control-next-icon" aria-hidden="true"></span>
<span class="sr-only">Next</span></a></div>
<div class="padding" style="margin-left: 160px;">
<div class="container shadows  ml-4">
<div class="row  pt-5">
<div class="col-lg-3 col-md-3 col-sm-6 col-xs-12" onclick="location.href =
'user_menu.php?foodId=1';" style="cursor: pointer;">
<div class="card card-one border" style="width: 15rem;" >
<div class="card-body rounded d-block" style="background-image:
url(../assets/img/p06skw9l.jpg);background-position: center;">
<h3 class="card-title text-white text-center font-weight-bold mt-5" ><br><br>
Breakfast</h3></div></div></div>
<div class="col-lg-3 col-md-3 col-sm-6 col-xs-12  " onclick="location.href =
'user_menu.php?foodId=2';" style="cursor: pointer;">
<div class="card card-one border " style="width: 15rem;">
```

```
<div class="card-body rounded" style="background-image:
url(../assets/img/lunch.jpg);background-position: center;">
</div></div></div>
<div class="col-lg-3 col-md-3 col-sm-6 col-xs-12  " onclick="location.href =
'user_menu.php?foodId=3';" style="cursor: pointer;">
<div class="card card-one border " style="width: 15rem;">
<div class="card-body rounded" style="background-image:
url(../assets/img/snacks.jpg);background-position: center;">
<h3 class="card-title  text-center font-weight-bold mt-5"><br><br>
Snacks</h3></div></div></div>
<div class="col-lg-3 col-md-3 col-sm-6 col-xs-12  " onclick="location.href =
'user_menu.php?foodId=4';" style="cursor: pointer;">
<div class="card card-one border " style="width: 15rem;">
<div class="card-body rounded" style="background-image:
url(../images/breakfast1.jpg);background-position: center;">
<h3 class="card-title  text-center font-weight-bold mt-5"><br><br>
</h3></div></div</div></div></div
<div class="container shadows">
<div class="row  pt-5 justify-content-center ml-5">
<div class="col-lg-6 col-md-6 col-sm-6 col-xs-12" onclick="location.href =
'searchpackage_view.php?varname=lightingpackage_table';" style="cursor: pointer;">
<div class="card card-one border" style="width: 25rem;height: 14rem;">
<div class="card-body rounded" style="background-image:
url(../assets/img/light.jpg);background-position: center;background-size: cover;">
<h3 class="card-title  text-white text-center font-weight-bold mt-4"><br>
</h3></div></div></div>
<div class="col-lg-6 col-md-6 col-sm-6 col-xs-12  " onclick="location.href =
'searchpackage_view.php?varname=decorpackage_table';" style="cursor: pointer;">
<div class="card card-one border " style="width: 25rem;height: 14rem;">
<div class="card-body rounded" style="background-image:
url(../assets/img/decoration.jpeg);background-position: center;background-size: cover;">
<h3  class="card-title  text-center font-weight-bold mt-4"><br>
Decorator
</h3></div></div></div></div></div>
```

```
<div class="container">
<div class="row  pt-5 ml-5">
<div class="col-lg-3 col-md-3 col-sm-6 col-xs-12">
<h5 style="color: red;">About Us</h5>
<div class="col-lg-3 col-md-3 col-sm-6 col-xs-12  ">
<h5 class="" style="color: red;">Location</h5>
<div class="col-lg-3 col-md-3 col-sm-6 col-xs-12  ">
<h5 style="color: red;">Services</h5>
<h6>Marrrige Function</h6>
<h6>House Warming</h6>
<h6>Birthday Party</h6>
<h6>Reception</h6></div>
<div class="col-lg-3 col-md-3 col-sm-6 col-xs-12  ">
<h5 style="color: red;">Contact Us</h5>
</div></div></div></div>
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/popper.min.js"></script>
<script src="../assets/js/bootstrap.min.js"></script>
<script src="../assets/js/food.js"></script>
<script language="JavaScript" type="text/javascript">
$(document).ready(function () {
$('#carouselExampleIndicators').carousel({interval: 3000})});
</script>
</body>
</html>
```

- **User_menu.php**

```
<?php
require_once './session.php';
require_once '../db/db.class.php';
$db = new DB();
$userRid = $_SESSION['userRid'];
$foods = array();
```

```
$sql = "";
$foodId = 0;
if (isset($_GET['foodId'])) {
$foodId = $_GET['foodId'];}
?>
<html>
<head>
<title></title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
</head>
<?php
require_once '../user/user_nav.php';
?>
<body class="body1">
<input type="hidden" id="userRid" value="<?php echo $userRid; ?>"/>
<div class="container"  style="margin-left: 180px;">
<form action="#" method="get">
<input class="search" type="text" name="searchQuery"/>
<button class="btn textcol" type="submit">Search</button></form>
<div class="row" >
<?php
if (count($foods) > 0) {
foreach ($foods as $row) {
?>
<div class="col-md-3">
<div class="card">
<h6 class="card-title text-center" style="font-weight: bold; font-size: 25px;">
<?php echo $row['food_name']; ?></h6>
<img src="../uploads/<?php echo $row['food_img']; ?>" class="img-thumbnail-home"
alt="...">
<div class="card-body">
<h6 class="text-center">
<a  href="#" onclick="insertCart('<?php echo $row['food_rid']; ?>')" id="btnInsertCart"
class="btn btn-danger"><i class="fa fa-shopping-cart" style="font-size:24px"></i> Add to
Cart  </a></h6></div>
<div class="card-footer text-center">
PRICE:   <?php echo $row['food_cost']; ?></div>
</div></div>
<?php}
} else {
?>
<div class="col-12">
<div class="alert alert-danger text-center alerts" role="alert">
No Such Item Found
</div></div>
<?php }
?>
</div></div>
<?php
require_once '../user/user_modal.php';?>
<script src="../assets/js/user.js"></script>
<script src="../assets/js/food.js"></script>
</body>
</html>
```

- **user_doFinalPayment**

```
<?php
require_once '../user/session.php';
require_once '../db/db.class.php';
$db = new DB();
$userRid = $_SESSION['userRid'];
$ohRid = 0;
$grandTot = 0
```

```
$sql = "SELECT * FROM `payment` WHERE payment_status=0 AND payment_userid =
'$userRid'";
$showRecord = $db->executeSelect($sql);
?>
<html>
<head>
<meta charset="UTF-8">
<title>payment</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
<body class="body1">
<?php
require_once './user_nav.php';
?>
<div class="container" style="margin-left: 190px;">
<div class="text-center mt-3 mb-3">
<h3 class="text-center text-light d-inline bg-primary font-weight-bold px-4 py-1">Payment
Your Order</h3></div>
<div class="card" id="compCardview">
<table class="table table-bordered table-hover">
<thead>
<tr style="text-align: center;">
<th>Sl No</th>
<th>Total Cost</th>
<th>Payment Action</th></tr>
</thead>
<tbody>
<?php
$i = 0;
foreach ($showRecord as $row) {
?>
<tr>
<td><?php echo ++$i; ?></td>
<td class="text-center"><?php echo $row['payment_totcost']; ?></td
<td class="text-center"><a  onclick="makePayment('<?php echo $row['payment_rid']; ?>',
'<?php echo $userRid ?>')"
id="btnMakePayment" class="btn btn-primary">Make Payment</a>
</td></tr>
<?php}
?>
</tbody>
</table></div></div></div>
<?php
require_once '../user/user_modal.php';
?>
<script src="../assets/js/user.js"></script>
<script src="../assets/js/food.js"></script>
</body>
</html>
```

- **user_viewcart.php**

```
<?php
require_once './session.php';
require_once '../db/db.class.php';
$db = new DB();
userRid = $_SESSION['userRid'];
$sql = "SELECT * FROM cart "
. " JOIN food ON (food_rid = cart_foodid)"
. " WHERE (cart_userid = $userRid && cart_status = '0')";
$carts = $db->executeSelect($sql);
?>
<html>
<head>
<meta charset="UTF-8">
<title>food</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
```

```
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
<body class="body1">
<?php
require_once './user_nav.php';
?>
<input type="hidden" id="sessionRid" value="<?php echo $userRid; ?>"/>
<div class="container" style="margin-left: 190px;">
<div class="text-center mt-3 mb-3">
<h3 class="text-center text-light d-inline bg-primary font-weight-bold px-4 py-1">My
Cart</h3></div>
<div class="row">
<?php
if (count($carts)) {
foreach ($carts as $row) {
?>
<div class="col-md-3">
<div class="card">
<h6 class="card-title text-center" style="font-weight: bold; font-size: 25px;">
<?php echo $row['food_name']; ?></h6>
<img src="../uploads/<?php echo $row['food_img']; ?>" class="img-thumbnail-home"
alt="...">
<div class="card-body text-center">
<input class="form-control" type="checkbox" name="selectOrder" value="<?php echo
$row['food_rid']; ?>">
<input type="hidden" name="cartIds" value="<?php echo $row['cart_rid']; ?>"/>
<input type="hidden" name="userIds" id="userIds" value="<?php echo $userRid; ?>"/>
</div>
<div class="card-footer text-center">PRICE:   <?php echo $row['food_cost']; ?>
</div></div></div>
<?php
}}
?></div>
<?php if (count($carts)) { ?>
<div class="form-group align-bottom ">
<a href="#" onclick="openCartModal()"  id="btnOrderCart"
class="btn btn-primary">
Order
</a></div>
<?php } ?>
</div>
<?php
require_once '../user/user_modal.php';
?>
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/popper.min.js"></script>
<script src="../assets/js/bootstrap.min.js"></script>
<script src="../assets/jquery-ui/jquery-ui.min.js"></script>
<script src="../assets/js/user.js"></script>
<script src="../assets/js/food.js"></script>
</body>
</html>
```

- **Userorder.php**

```
<?php
require_once './session.php';
require_once '../db/db.class.php';
$db = new DB();
$userRid = $_SESSION['userRid'];
$packageid = $_GET['packageid'];
$tabledata = $_GET['varname'];
$table_data = $_GET['varname'];
$packagename = $_GET['packagename'];
$sql = "SELECT * FROM user WHERE `user_rid` = $userRid";
$userdetails = $db->executeSelect($sql);
?>
<html>
```

```
<head>
<meta charset="UTF-8">
<title>Packages</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
<body class="body1">
<?php
require_once './user_nav.php';
?>
<div class="container" style="margin-left: 160px;">
<div class="card" id="compCardview">
<div class="card-body">
<form id="orderpackage" method="post" action="../actions/food_action.php"
enctype="multipart/form-data">
<div class="row justify-content-center border">
<div class="col-12">
<input readonly type="hidden" name="command" value="orderpackage" />
<input readonly type="hidden" name="userid" value="<?php echo $userRid ?>" />
<input readonly type="hidden" name="packageid" value="<?php echo $packageid ?>" />
<input readonly type="hidden" name="tablename" value="<?php echo $table_data ?>" />
<?php
foreach ($userdetails as $array) {
?>
<div class="d-inline">Name: <input readonly class="bg-transparent noedit border-0 "
type="text" name="username" id="username" value="<?php echo $array['user_uname'] ?>"
/></div>
<div class="d-inline">PACKAGE SELECTED: <input readonly class="bg-transparent
noedit border-0 " type="text" name="packagename" id="packagename" value="<?php echo
$packagename ?>" /></div>
<?php }
?>
</div>
<div class="col-6 mt-3">
```

```
<div class="form-group mt-3">
<label>Delivary Date</label>
<input type="date" class="form-control textstyle" id="delivarydate" name="delivarydate"
placeholder="enter name" value=""></div>
<div class="form-group">
<label>Number of Guets</label>
<input type="number" class="form-control textstyle" id="noofguests" name="noofguests"
placeholder="enter number of guest" value=""></div>
<div class="w-100 align-center">
<button type="submit" id="btnorderpackage" class="btn btn-primary px-
5">Submit</button></div></div>
<div class="col-6 mt-3">
<div class="form-group">
<label>Delivary Details</label>
<textarea class="form-control textstyle" id="address" name="address" rows="5"
placeholder="enter description">
<?php
foreach ($userdetails as $array) {
echo $array['user_add'];
}?>
</textarea></div></div></div></form></div></div></div>
<script src="../assets/js/bootstrap.min.js"></script>
<script src="../assets/js/food.js"></script>
```

- **Cateter_home.php**

```
<?php
require_once './session.php';
?>
<html>
<head>
<meta charset="UTF-8">
<title>caterer</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
```

```
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
<body class="img-thumbnail user-logbody">
<?php
require_once '../cateror/admin_nav.php';
require_once '../cateror/admin_modal.php';
?>
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/food.js"></script>
</body>
</html>
```

- **caterer_viewfood.php**

```
<?php
require_once './session.php';
require_once '../db/db.class.php';
$db = new DB();
$sql = "SELECT * FROM food"
. " JOIN food_type ON(food_type_id = food_type_rid)";
$viewFood = $db->executeSelect($sql);
?>
<html>
<head>
<title>View Product</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
<body>
<?php
require_once '../cateror/admin_nav.php';
?>
<div class="container-fluid">
<div class="card" id="compCardview">
```

```
<div class="card-body">
<table class="table table-bordered table-hover">
<thead>
<tr>
<th>Sl No</th>
<th>Food Type</th>
<th>Food Name</th>
<th>Cost</th>
<th>Image</th>
</tr></thead>
<tbody>
<?php
$i = 0;
foreach ($viewFood as $row) {
?>
<tr>
<td><?php echo ++$i; ?></td>
<td><?php echo $row['food_type_name']; ?></td>
<td><?php echo $row['food_name']; ?></td>
<td><?php echo $row['food_cost']; ?></td>
<td><a class="\1" data-lightbox="\1" href="../uploads/<?php echo
"$row[food_img]">></a></td>
<a class="btn btn-sm btn-dark text-white px-3" href="#"
data-toggle="modal" data-target="#updateFoodModal" onclick="updateProduct(
'<?php echo $row['food_rid']; ?>',
'<?php echo $row['food_type_id']; ?>',
'<?php echo $row['food_name']; ?>',
'<?php echo $row['food_decri']; ?>',
'<?php echo $row['food_cost']; ?>',
'<?php echo $row['food_img']; ?>')">Edit</a>
<a href="#" class="btn btn-sm btn-dark text-white px-2" onclick="deleteFood('<?php echo
$row['food_rid']; ?>')">Delete</a>
</td></tr>
<?php}
```

```
?>
</tbody></table></div></div></div>
<?php
require_once '../cateror/admin_modal.php';
?>
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/bootstrap.min.js"></script>
<script src="../assets/js/food.js"></script>
</body>
</html>
```

- **caterer_addoffer.php**

```
<?php
require_once './session.php';
require_once '../db/db.class.php';
$db = new DB();
$sql = "SELECT * FROM offer"
. " JOIN food ON (food_rid = offer_foodId)";
$viewOffer = $db->executeSelect($sql);
?>
<html>
<head>
<meta charset="UTF-8">
<title>View Offer</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
<body>
<?php
require_once '../cateror/admin_nav.php';
?>
<div class="container-fluid">
<div class="card" id="compCardview">
```

```
<div class="card-body">
<table class="table table-bordered table-hover">
<thead>
<tr>
<th>Sl No</th>
<th>Food Name</th>
<th>Cost</th>
<th>Discount</th>
<th>Discount Amount</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<?php
$i = 0;
foreach ($viewOffer as $row) {
?>
<tr>
<td><?php echo ++$i; ?></td>
<td><?php echo $row['food_name']; ?></td>
<td><?php echo $row['offer_cost']; ?></td>
<td><?php echo $row['offer_disc']; ?> %</td>
<td><?php echo $row['offer_discAmt']; ?></td>
<td>
<a class="colch" href="#"onclick="updateOffer(
'<?php echo $row['offer_rid']; ?>',
'<?php echo $row['offer_foodId']; ?>',
'<?php echo $row['offer_cost']; ?>',
'<?php echo $row['offer_disc']; ?>',
'<?php echo $row['offer_discAmt']; ?>')">Edit</a>
<a href="#" class="btn btn-link" onclick="deleteOffer('<?php echo $row['offer_rid'];
?>')">Delete</a>
</td></tr>
<?php
}
?>
</tbody></table></div></div></div>
<?php
require_once '../cateror/admin_modal.php';
?>
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/food.js"></script>
</body>
</html>
```

- **decorator_home.php**

```
<?php
require_once './session.php';
?>
<html>
<head>
<meta charset="UTF-8">
<title>Decorator</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
<body class="img-thumbnail user-logbody">
<?php
require_once '../decorator/decorator_nav.php';
require_once '../decorator/decorator_modal.php';
?>
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/food.js"></script>
</body>
</html>
```

- **Editpackage.php**

```
<?php
require_once '../db/db.class.php';
$db = new DB();
$id = $_POST['id'];
$name = $_POST['name'];
$type = $_POST['type'];
$price = $_POST['price'];
$description = $_POST['description'];
$selectFoodType = "SELECT `type` FROM packagetype";
$foodChoose = $db->executeSelect($selectFoodType);
?>
<html>
<head>
<meta charset="UTF-8">
<title>Edit Packages</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
<body>
<?php
require_once '../decorator/decorator_nav.php';
?
<div class="container-fluid">
<div class="card" id="compCardview">
<div class="card-body"><div class="row justify-content-center">
<form id="PackageForm" method="post" action="../actions/food_action.php"
enctype="multipart/form-data">
<input type="hidden" name="command" value="updatedecoratorpackage" />
<input type="hidden" name="id" value="<?php echo $id ?>" />
<div class="row">
<div class="form-group">
<label>package Name</label>
```

```
<input type="text" class="form-control textstyle" id="packagename" name="packagename"
placeholder="enter name" value="<?php echo $name ?>"></div>
<div class="form-group">
<label>Package Type</label>
<select class="form-control textstyle" id="packagetype" name="packagetype">
<option value="select">Choose...</option>
<?php
foreach ($foodChoose as $array) {
?>
<option value="<?php echo $array['type']; ?>">
<?php echo $array['type']; ?>
</option>
<?php}
?>
</select></div>
<div class="form-group">
<label>Price</label>
<input type="number" class="form-control textstyle" id="packageprice"
name="packageprice" placeholder="enter Price" value="<?php echo $price ?>"></div>
<div class="w-100 align-center">
<button type="submit" id="btnSubmitpackage" class="btn btn-primary px-
5">Submit</button></div></div>
<div class="col-6">
<div class="form-group">
<label>package Description</label>
<textarea class="form-control textstyle" id="packagedescri" name="packagedescri"
rows="5" placeholder="enter description"><?php echo $description ?></textarea></div>
<div class="form-group mt-4 pt-3">
<select name="active" id="" class="form-control textstyle">
<option value="1">active</option>
<option value="0">de activate</option>
</select></div></div></div></form>
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/popper.min.js"></script>
```

```
<script src="../assets/js/bootstrap.min.js"></script>
<script src="../assets/js/food.js"></script>
</body>
</html>
```

- **view_order.php**

```
<?php
require_once './session.php';
require_once '../db/db.class.php';
$db = new DB();
$sql = " SELECT o.id,u.user_name,d.name,o.Order_Date,o.Delivery_Date,"
. "o.Delivery_address,o.Noofpeople,o.order_status FROM orderpackage AS o "
. "JOIN `user` AS u ON  o.User_Id=u.user_rid "
. "JOIN decorpackage_table AS d ON d.id=o.packageid "
. "WHERE table_name='decorpackage_table'";
$viewOrder = $db->executeSelect($sql);
?>
<html>
<had>
<title>View Packages</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head
<body>
<?php
require_once '../decorator/decorator_nav.php';
require_once '../decorator/decorator_modal.php';
?>
<div class="container-fluid">
<div class="card" id="compCardview">
<div class="card-body">
<table class="table table-bordered table-hover">
<thead>
```

```
<tr>
<th>Sl No</th>
<th>User Name</th>
<th>Package Name</th>
<th>Order Date</th>
<th>Delivery Date</th>
<th>No Of People</th>
<th>Address</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<?php
$i = 0;
if (count($viewOrder) > 0) {
foreach ($viewOrder as $row) {
?>
<tr>
<td><?php echo ++$i; ?></td>
<td><?php echo $row['user_name']; ?></td>
<td><?php echo $row['name']; ?></td>
<td><?php echo $row['Order_Date']; ?></td>
<td><?php echo $row['Delivery_Date']; ?></td>
<td><?php echo $row['Noofpeople']; ?></td>
<td><?php echo $row['Delivery_address']; ?></td>
<td>
<select id="state" name="state" class="form-control rounded my-1 form-control-md"
onchange="update_order_status(<?php echo "$row[id]" ?>,<?php echo "$(this).val()" ?>);"
<?php
echo ";$order_status = $row['order_status'];
if ($order_status == 0) {
?>
<option value="0" selected>Under Process</option>
<?php } else {?>
<option value="0">Under Process</option>
<?php}
if ($order_status == 1) {
?>
<option value="1" selected>Accepted</option>
<?php }
<?php if ($order_status == 2) { ?>
<option value="2" selected>Rejected</option>
<?php } else {?>
<?php }?>
</select></td></tr>
<?php
}} else {
?>
<tr>
<td colspan="7" class="alert alert-danger text-center">No Order</td></tr>
<?php } ?>
</tbody></table></div></div</div>
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/order_status.js"></script>
</body>
</html>
```

- **lighting_home.php**

```
<?php
require_once './session.php';
?>
<html>
<head>
<title>Lighting</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
```

```
<body class="img-thumbnail user-logbody">
<?php
require_once '../lighting/lighting_nav.php';
require_once '../lighting/lighting_modal.php';
?>
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/food.js"></script>
</body>
</html>
```

- **admin_home.php**

```
<?php
require_once './session.php';
require_once '../db/db.class.php';
$db = new DB();
$sql1 = "SELECT * FROM `lightingpackage_table` WHERE isactive=1 ORDER BY id
DESC LIMIT 4";
$lighting_package = $db->executeSelect($sql1);
$sql2 = "SELECT * FROM `decorpackage_table` WHERE isactive=1 ORDER BY id DESC
LIMIT 4";
$decorator_package = $db->executeSelect($sql2)
$sql3 = "SELECT * FROM `food` AS f "
. "JOIN food_type AS ft ON f.food_type_id=ft.food_type_rid "
. "ORDER BY food_rid DESC LIMIT 4";
$food_data = $db->executeSelect($sql3);
?>
<html>
<head>
<title>Admin</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
<body class="img-thumbnail user-logbody">
```

```
<?php
require_once '../admin/admin_nav.php';?>
<div class="container"  style="margin-left: 180px;">
<div class="row">
<?php
if (!empty($lighting_package)) {
foreach ($lighting_package as $row) {?>
<div class="col">
<div class="card" style="width: 15rem;height: 22rem;background-color: #ffffff" >
<img src="../uploads/<?php echo "$row[image]" ?>" class="card-img" alt="...">
<h4 class="card-title text-dark text-center font-weight-bold" >
<?php echo "$row[name]" ?></h4>
<h6 class="card-title text-dark text-center font-weight-bold" >Price: <?php echo
"$row[price]" ?>/-</h6>
<h6 class="card-title text-dark text-center font-weight-bold" >
<?php echo "$row[type]" ?>
</h6></div</div></div>
<?php
}}
?>
</div>
<div class="row">
<?php
if (!empty($decorator_package)) {
foreach ($decorator_package as $row) {?>
<div class="col">
<div class="card" style="width: 15rem;height: 22rem;background-color: #ffffff" >
<img src="../uploads/<?php echo "$row[image]" ?>" class="card-img" alt="...">
<div class="card-body rounded d-block">
<h4 class="card-title text-dark text-center font-weight-bold" >
<?php echo "$row[name]" ?></h4>
<h6 class="card-title text-dark text-center font-weight-bold" >
Price: <?php echo "$row[price]" ?>/-</h6>
<h6 class="card-title text-dark text-center font-weight-bold" >
<?php echo "$row[type]" ?>
</h6></div></div></div>
<?php
}}
?></div>
<div class="row">
<?php
if (!empty($food_data)) {
foreach ($food_data as $row) {?>
<div class="col">
<div class="card" style="width: 15rem;height: 22rem;background-color: #ffffff" >
<img src="../uploads/<?php echo "$row[food_img]" ?>" class="card-img" alt="...">
<div class="card-body rounded d-block">
<h4 class="card-title text-dark text-center font-weight-bold" >
<?php echo "$row[food_name]" ?></h4>
<h6 class="card-title text-dark text-center font-weight-bold" >
<?php echo "$row[food_type_name]" ?>
</h6></div></div></div>
<?php
}}?>
</div><br><br></div>
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/food.js"></script>
</body>
</html>
```

- **Logout.php**

```
<?php
session_start();
session_destroy();
$_SESSION['userRid'] = 0
header("location: index.php");
die();
```

6.4 Data Validation

```
<form method="post" action="#" onsubmit="return validateuserlogin();">
<div class="form-group">
<label>Email ID</label>
<div class="border">
<input type="text" class="form-control" placeholder="Email ID"
value="<?php echo $u_email ?>"  name="username" id="username" autocomplete="off">
</div></div>
<div class="form-group">
<label>Password</label>
<div class="border">
<input type="password" class="form-control" placeholder="Password"
value="<?php echo $u_password ?>" name="password" id="password" autocomplete="off">
</div></div>
<div class="form-group">
<label>OTP</label>
<input class=form-control type="password" id="otp" name="otp"
placeholder="Enter OTP" autocomplete="off" required/></div>
<div class="form-group">
<?php
if (!empty($error)) {
?>
<div class="alert alert-danger alert-dismissible fade show" role="alert">
<?php echo $error; ?>
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button></div>
<?php }
?></div>
<div class="form-group text-center">
<button type="submit" class="btn btn-primary " name="submit">Login</button></div>
<div class="form-group text-center">
<a href="../user/user_register.php">Create New Account</a>
```

```
<div class="form-group text-center">
<a href="#" data-toggle="modal" data-target="#userForgotPwdModal">Forgot
Password</a></div></div></form></div></div></div>
<div class="col-md-4"></div></div>
<div class="modal fade" id="userForgotPwdModal" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title">Forgot Password</h5>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button></div>
<div class="modal-body">
<form id="ForgotPwdForm" method="post" action="../actions/user_action.php">
<input type="hidden" name="command" value="postEmail" />
<div class="form-group">
<label>Enter Email</label>
<input type="email" class="form-control " id="frgtPwdEmailUser"
name="frgtPwdEmailUser" placeholder="Enter Email"></div>
<button type="submit" id="btnSubmitEmailUser" class="btn btn-primary">Submit</button>
</form>
```

6.5 Search

- **searchpackage_view.php**

```
<?php
require_once './session.php';
require_once '../db/db.class.php';
$db = new DB();
$userRid = $_SESSION['userRid'];
$table_data = $_GET['varname'];
$sql = "SELECT * FROM $table_data";
$packages = $db->executeSelect($sql);
```

```
$beforeunderscore = current(explode("package", $table_data));
?>
<html>
<head>
<meta charset="UTF-8">
<title>Packages</title>
<link rel="stylesheet" type="text/css" href="../assets/css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/jquery-ui/jquery-ui.min.css" />
<link rel="stylesheet" type="text/css" href="../assets/css/custom.css" />
</head>
<body>
<?php
require_once 'user_nav.php';?>
<input type="hidden" id="sessionRid" value="<?php echo $userRid; ?>" />
<div class="container pb-3"  style="margin-left: 180px;">
<h3 class="text-center text-light mt-4 text-uppercase"><?php echo
$beforeunderscore;Package</h3>
<div class="row">
<?php
if (count($packages)) {
foreach ($packages as $array) {
?>
<div class="col-md-3">
<div class="card">
<img src="../uploads/<?php echo "$array[image]" ?>" class="card-img" alt="...">
<input type="text" readonly class="card-title text-center"<?php echo $name = $array['name'];
?>">
<div class="card-body text-center">
<form action="" method="POST">
<input readonly class="form-control border-0 bg-transparent noedit" type="text"
name="selectOrder" value="<?php echo $type = $array['type']; ?>">
<input readonly class="form-control border-0 bg-transparent noedit" type="text"
name="cartIds" value="<?php echo $price = $array['price']; ?>" />
<input readonly class="form-control border-0 bg-transparent noedit" type="hidden"
name="packageid" id="packageid" value="<?php echo $packageid = $array['id']; ?>" />
<a href="userorder.php?varname=<?php echo $table_data ?>&packageid=<?php echo
$packageid ?>&packagename=<?php echo $name ?>" class="btn btn-primary px-3 py-
1">Book Package</a>
</form></div></div></div>
<?php}}?>
</div></div>
<?php
require_once '../user/user_modal.php';
?>
<script src="../assets/js/jquery-3.3.1.min.js"></script>
<script src="../assets/js/user.js"></script>
<script src="../assets/js/food.js"></script>
</body>
</html>
```

6.6 Backup/Recovery

```
CREATE TABLE `cart` ( `cart_rid` int(10) NOT NULL AUTO_INCREMENT, `cart_userid`
int(20) DEFAULT NULL,`cart_foodid` int(20) DEFAULT NULL,`cart_status` tinyint(1)
DEFAULT NULL,cart_date` date DEFAULT NULL,  PRIMARY KEY (`cart_rid`)
)ENGINE=InnoDB AUTO_INCREMENT=32 DEFAULT CHARSET=latin1;
/*Data for the table `cart` */
insert  into `cart`(`cart_rid`,`cart_userid`,`cart_foodid`,`cart_status`,`cart_date`) values
(1,3,16,1,'2021-09-01'),(2,3,22,1,'2021-09-01'),(3,3,19,1,'2021-09-01'),(4,3,21,1,'2021-09-
11'),(5,3,15,1,'2021-09-11'),(6,3,31,1,'2021-09-11'),(7,3,16,1,'2021-09-11'),(8,3,17,1,'2021-
09-11'),(9,3,27,1,'2021-09-11'),(10,3,16,1,'2021-09-13'),(11,3,10,1,'2021-09-
20'),(27,3,13,0,'2021-09-20'),(28,3,22,1,'2021-09-20'),(29,3,16,1,'2021-09-
20'),(30,3,19,1,'2021-09-20'),(31,3,32,0,'2021-09-20');

/*Table structure for table `cateror` */
```

```
CREATE TABLE `cateror` (  `c_uid` int(5) NOT NULL AUTO_INCREMENT,`u_name`
varchar(20) DEFAULT NULL, `u_pass` varchar(20) DEFAULT NULL,
`is_enabled` int(5) DEFAULT '1' COMMENT '1-enabled,0-disabled',
  PRIMARY KEY (`c_uid`)) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT
CHARSET=latin1;

/*Data for the table `cateror` */
insert  into `cateror`(`c_uid`,`u_name`,`u_pass`,`is_enabled`) values (2,'cateror','cateror',1);
/*Table structure for table `d_payment` */
CREATE TABLE `d_payment` (  `payment_rid` int(10) NOT NULL AUTO_INCREMENT,
`payment_oh_id` int(10) NOT NULL,`payment_userid` int(10) NOT NULL,`payment_time`
time NOT NULL, `payment_date` date NOT NULL,`payment_status` tinyint(1) NOT
NULL,`payment_totcost` double NOT NULL,PRIMARY KEY (`payment_rid`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
/*Data for the table `d_payment` */
insert into`d_payment`(`payment_rid`,`payment_oh_id`,`payment_userid`,`payment_time`,
`payment_date`,`payment_status`,`payment_totcost`) values (1,3,3,'12:08:51','2021-09-
13',1,5000);
/*Table structure for table `food` */
CREATE TABLE `food` ( `food_rid` int(11) NOT NULL AUTO_INCREMENT,
`food_type_id` varchar(20) NOT NULL,`food_name` varchar(20) NOT NULL, `food_decri`
varchar(100) NOT NULL,`food_cost` double NOT NULL, `food_img` text NOT NULL,
 PRIMARY KEY (`food_rid`)) ENGINE=InnoDB AUTO_INCREMENT=33 DEFAULT
CHARSET=latin1;
/*Data for the table `food` */
insert  into
`food`(`food_rid`,`food_type_id`,`food_name`,`food_decri`,`food_cost`,`food_img`) values
(10,'1','Lemon Tea','It is south Indian special tea, it contains 23,'2','curd rice','It is more often
eaten by south indians. It is a part of meal, it makes a great summer food as Juice','It is a
special indian watermelon juice and it contains special pune pepper
masala',90,'18778627245c93231852fb3.jpg'),(31,'1','dosaa','dedasafas',324,'12115085610268
1b23b7e.jpg'),(32,'2','Biriyani','Biriyani is a non-veg item, prepared with basamtii
rice.',100,'1088147670612f6277356fd.jpg');
/*Table structure for table `food_type` */
```

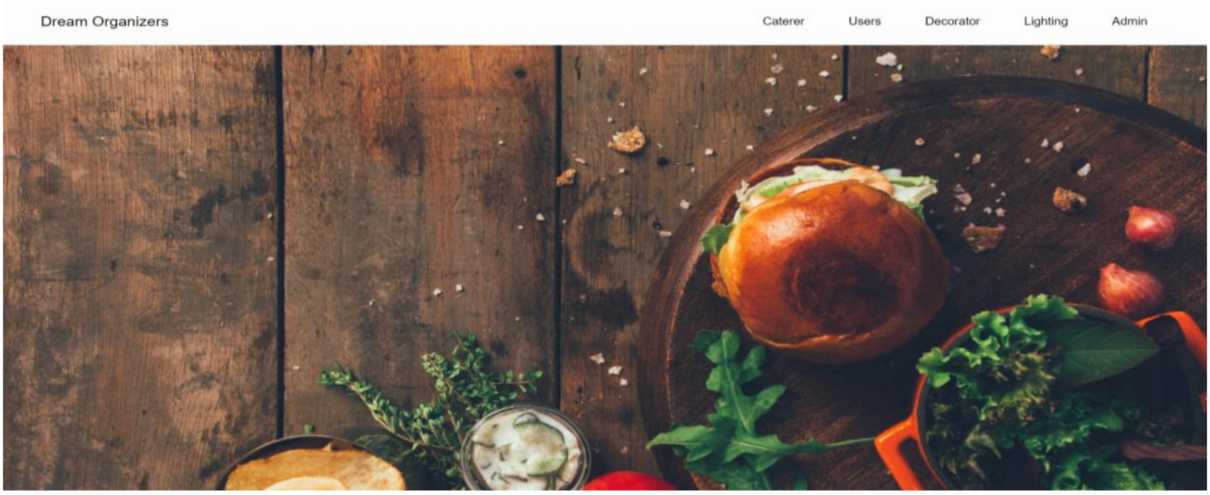
```
CREATE TABLE `food_type` ( `food_type_rid` int(10) NOT NULL AUTO_INCREMENT,
`food_type_name` varchar(20) NOT NULL,PRIMARY KEY (`food_type_rid`)
ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
/*Data for the table `food_type` */
insert  into
`food_type`(`food_type_rid`,`food_type_name`)values(1,'Breakfast'),(2,'Lunch'),(3,'Snaks'),(4
,'Dinner');
/*Table structure for table `l_payment` */
CREATE TABLE `l_payment` (`payment_rid` int(10) NOT NULL AUTO_INCREMENT,
 ‘ayment_oh_id` int(10) NOT NULL,userid` int(10) NOT NULL`payment_time` time NOT
NULL`payment_date` date NOT NULL,payment_status` tinyint(1) NOT`payment_totcost`
double NOT NULL PRIMARY KEY (`payment_rid`))
ENGINE=InnoDBAUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
/*Data for the table `l_payment` */
insert  into
`l_payment`(`payment_rid`,`payment_oh_id`,`payment_userid`,`payment_time`,`payment_da
te`,`payment_status`,`payment_totcost`) values (1,2,3,'12:10:05','2021-09-13',1,14900);
/*Table structure for table `lighting` */
CREATE TABLE `lighting` (`l_uid` int(5) NOT NULL AUTO_INCREMENT, `u_name`
varchar(20) DEFAULT NULL,`u_pass` varchar(20) DEFAULT NULL,is_enabled` int(5)
DEFAULT '1' COMMENT '1-enabled,0-disabled', PRIMARY KEY (`l_uid`)
ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
.jpg'),(7,'Package55','birthday','dgfdg',8900,1,'1581427914613cb52b82c1d.jpg');
/*Table structure for table `offer` */
CREATE TABLE `offer` ( `offer_rid` int(10) NOT NULL AUTO_INCREMENT
`offer_foodId` int(10) NOT NULL,  `offer_cost` float NOT NULL,`offer_disc` float NOT
NULL,`offer_discAmt` float NOT NULL,PRIMARY KEY (`offer_rid`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;
/*Data for the table `offer` */
insert  into `offer`(`offer_rid`,`offer_foodId`,`offer_cost`,`offer_disc`,`offer_discAmt`) values
(2,4,200,10,180),(3,3,100,7,93),(4,6,200,5,190),(5,8,20,2,19.6),(6,5,400,3,388),(7,18,60,10,5
4),(8,19,50,7,46.5),(9,32,100,5,95);
/*Table structure for table `order_details` */
```

```
CREATE TABLE `order_details` ( `od_rid` int(10) NOT NULL AUTO_INCREMENT
`od_header_id` int(10) NOT NULL,`od_food_id` int(10) NOT NULL,  `od_qty` int(10) NOT
NULL,`od_cart_id` int(10) NOT NULL,  PRIMARY KEY (`od_rid`)
) ENGINE=InnoDB AUTO_INCREMENT=30 DEFAULT CHARSET=latin1;
/*Data for the table `order_details` */
insert  into `order_details`(`od_rid`,`od_header_id`,`od_food_id`,`od_qty`,`od_cart_id`)
values
(1,1,15,26,5),(2,1,31,26,6),(3,1,16,26,7),(4,1,17,26,8),(5,2,27,89,9),(6,2,16,89,10),(7,3,15,29,
12),(8,3,16,29,13),(9,4,16,27,),(29,12,19,30,30);
/*Table structure for table `order_header` */
CREATE TABLE `order_header` ( `oh_rid` int(10) NOT NULL AUTO_INCREMENT,
`oh_user_id` int(10) NOT NULL,`oh_date` date NOT NULL,`oh_status` tinyint(1)
DEFAULT NULL `oh_order_no` varchar(10) NOT NULL,`oh_payment_status` tinyint(1)
DEFAULT NULL`oh_del_date` date NOT NULL,`oh_address` varchar(100) NOT NULL,
  `advanced_payment_status` int(5) DEFAULT '0',PRIMARY KEY (`oh_rid`)
ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=latin1;
/*Data for the table `order_header` */
insert  into
`order_header`(`oh_rid`,`oh_user_id`,`oh_date`,`oh_status`,`oh_order_no`,`oh_payment_stat
us`,`oh_del_date`,`oh_address`,`advanced_payment_status`) values (1,3,'2021-09-
11',0,'167043168',1,'2021-09-11','yuytu',1),(2,3,'2021-09-13',0,'1077836728',1,'2021-09-
22','jgjvgj',1),(3,1,'2021-20',0,'1660075122',1,'2021-09-24','Karkala',1),(12,3,'2021-09-
20',0,'1774067957',0,'2021-09-24','Karkala',0);
/*Data for the table `user` */
insert  into
`user`(`user_rid`,`user_name`,`user_mob`,`user_email`,`user_add`,`user_gender`,`user_dob`,`
user_uname`,`user_pass`,`is_enabled`,`is_verified`,`otp`) values
(3,'Suneetha','9876543210','suneetha090@gmail.com','Karkala','female','1970-01-
01','sun','1234',1,1,3668),(4,'Sunil','9898789878','suneethamoolya090@gmail.com','Karkala','
male','2000-09-01','sunil','5678',1,1,4442);
```

Chapter-7

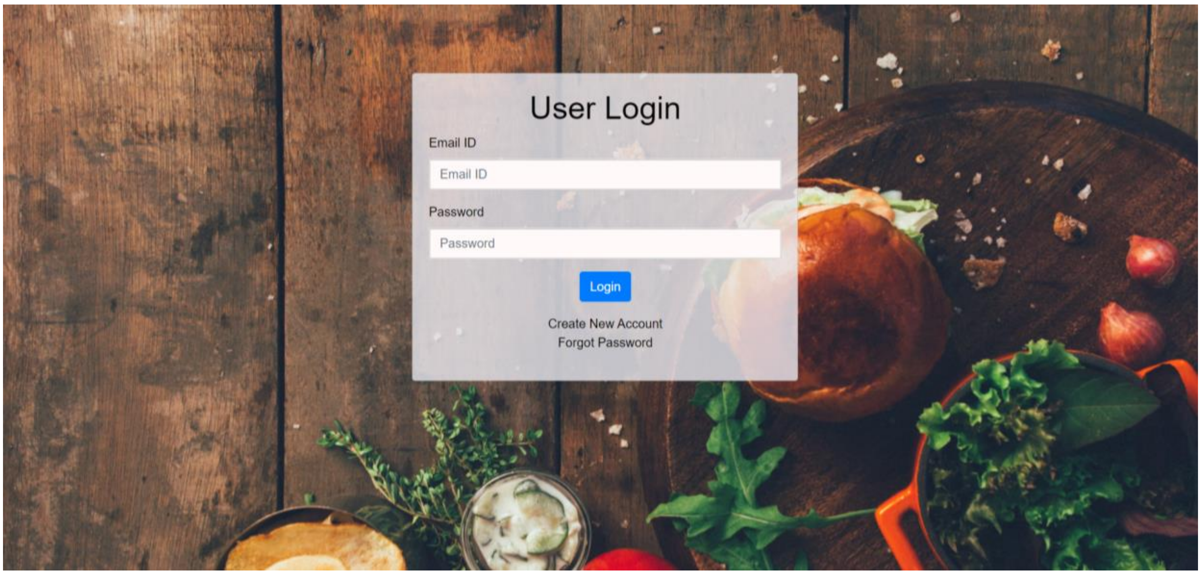
USER INTERFACE

7.1. User or Caterer or Decorator or Lighting or Admin to choose login



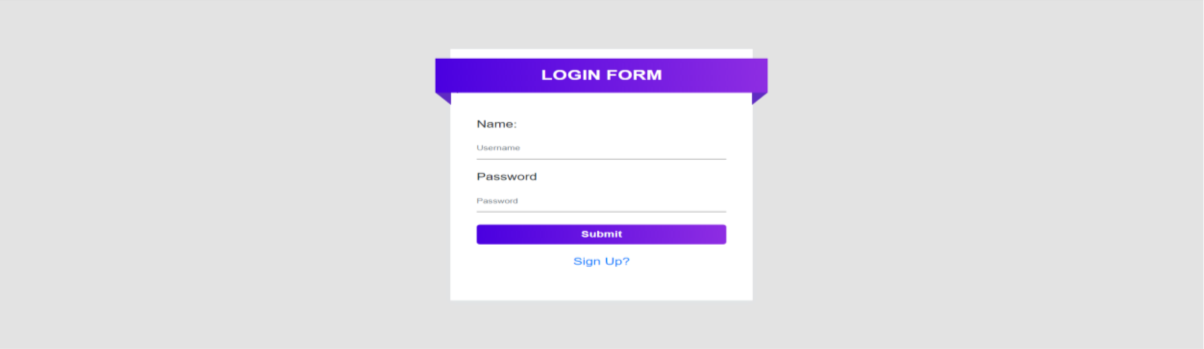
7.1.2 User login

A page for user login.



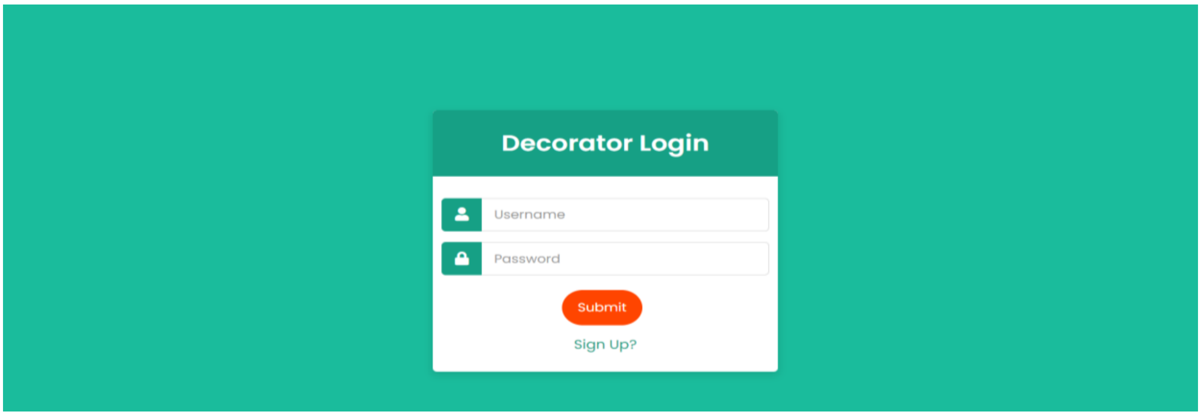
7.1.3 Caterer Login

Login page for caterer login.



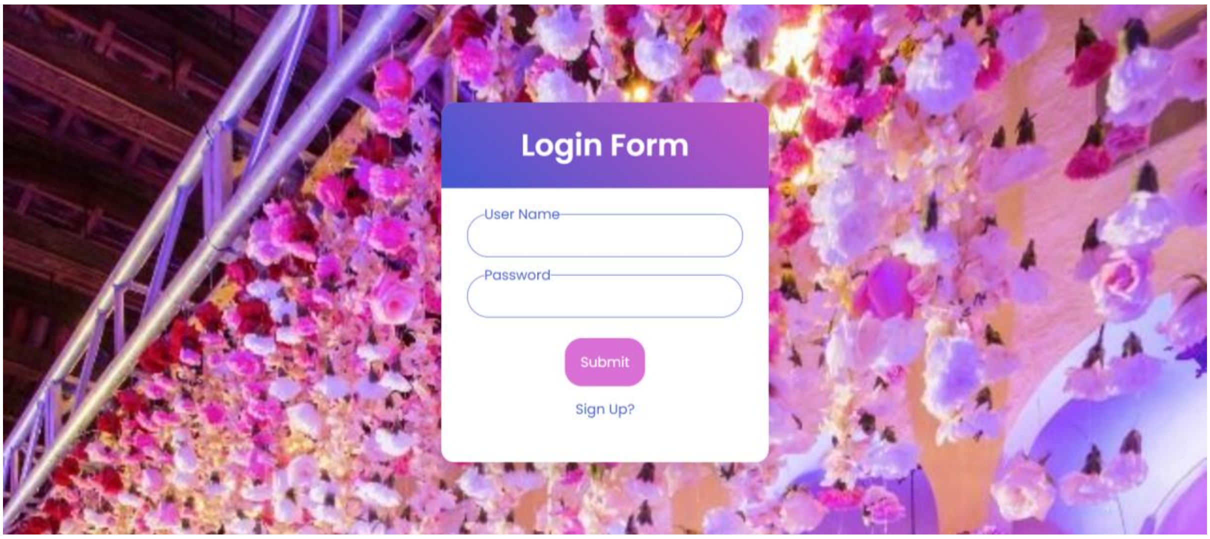
7.1.4 Decorator Login

Login page for decorator login.



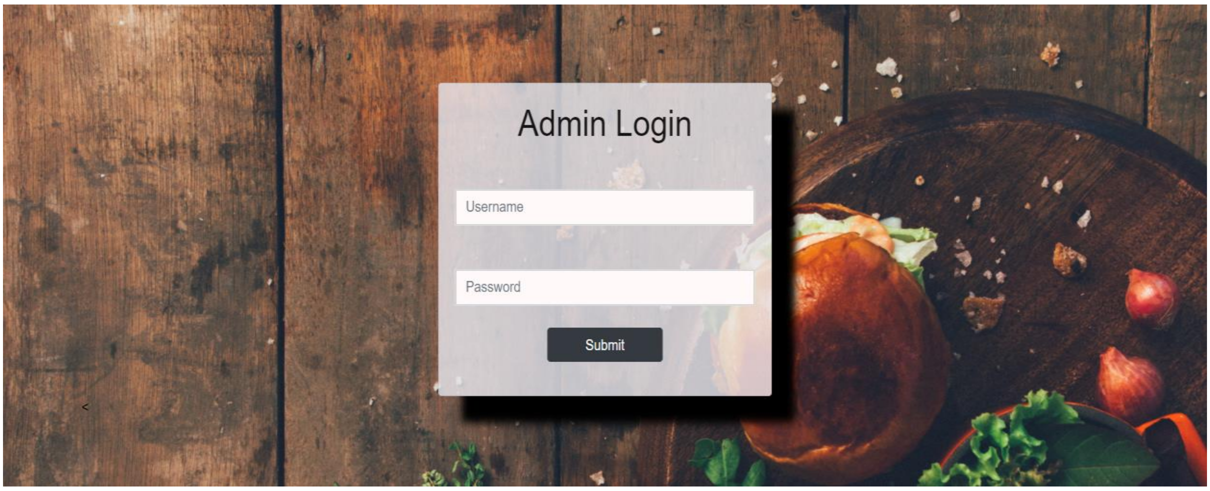
7.1.5 Lighting Login

Login page for lighting login.



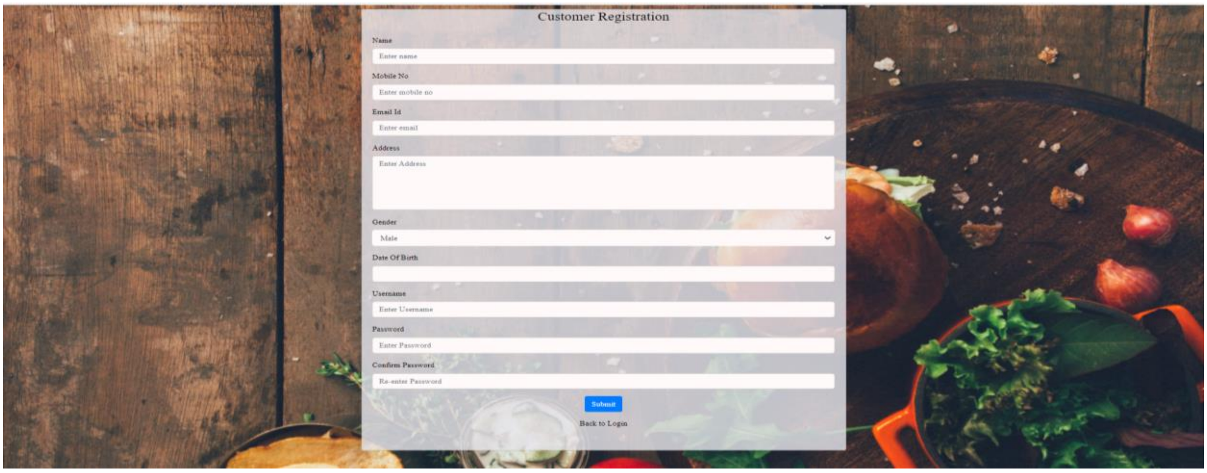
7.1.6 Admin Login

Login page for admin.



7.1.7 User Register

A page for user Register.



Customer Registration

Name

Enter name

Mobile No

Enter mobile no

Email Id

Enter email

Address

Enter Address

Gender

Male

Date Of Birth

Username

Enter Username

Password

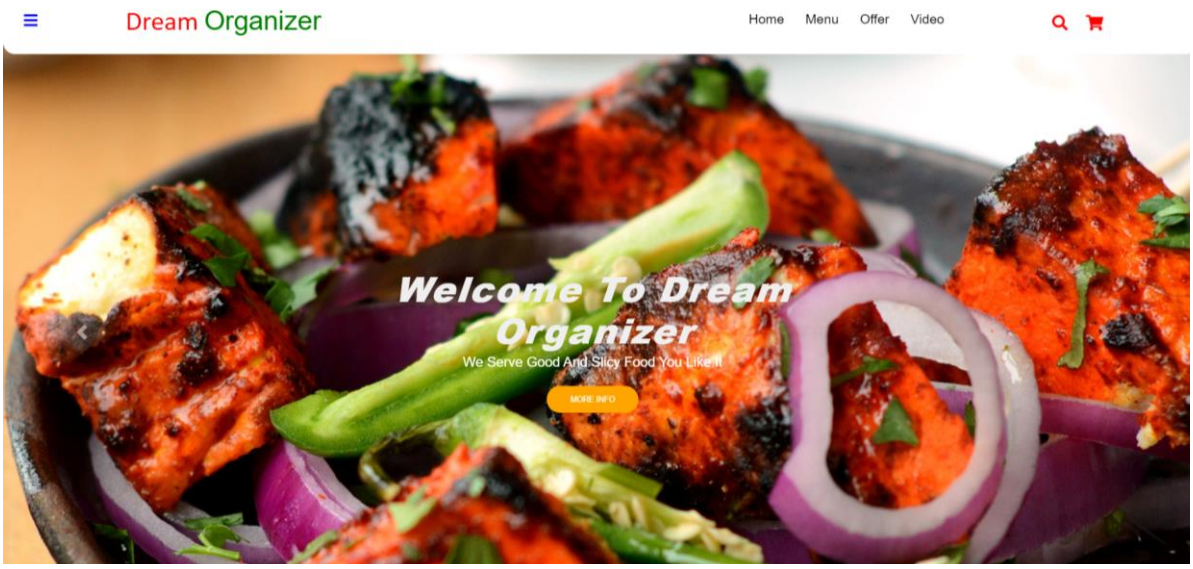
Enter Password

Confirm Password

Re-enter Password

7.2 User Home

A page for user Homepage.

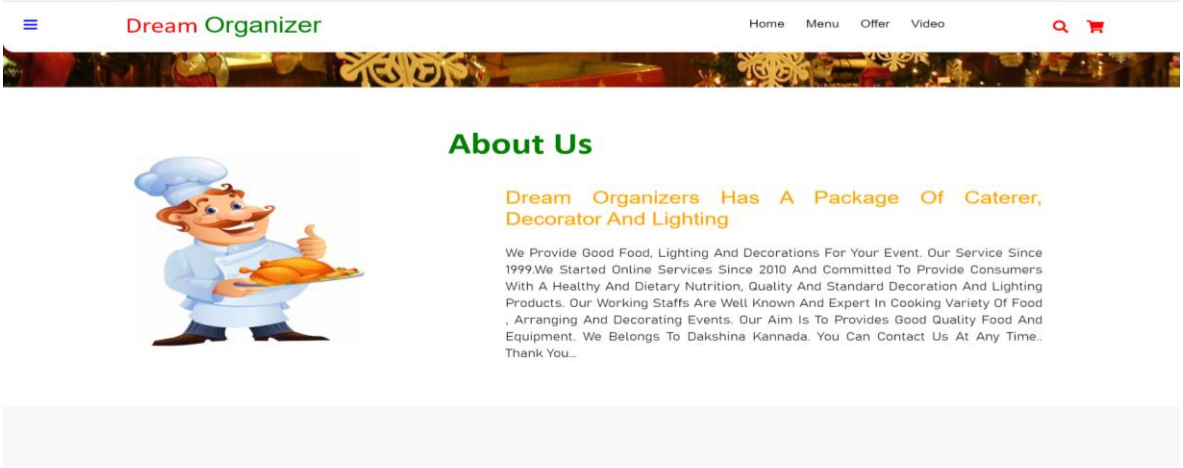


Welcome To Dream Organizer

We Serve Good And Slicy Food You Like It

7.2.1 About us

A page for about us of dream organizer.



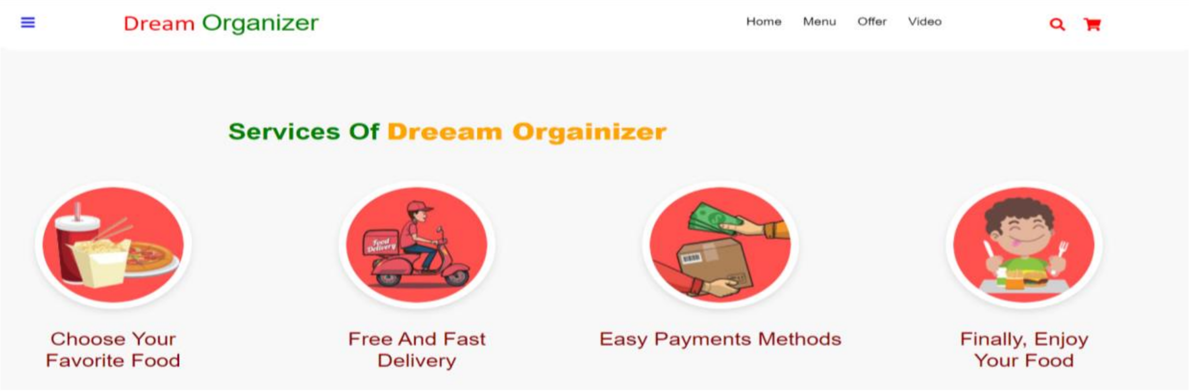
About Us

Dream Organizers Has A Package Of Caterer, Decorator And Lighting

We Provide Good Food, Lighting And Decorations For Your Event. Our Service Since 1999.We Started Online Services Since 2010 And Committed To Provide Consumers With A Healthy And Dietary Nutrition, Quality And Standard Decoration And Lighting Products. Our Working Staffs Are Well Known And Expert In Cooking Variety Of Food , Arranging And Decorating Events. Our Aim Is To Provides Good Quality Food And Equipment. We Belongs To Dakshina Kannada. You Can Contact Us At Any Time.. Thank You...

7.2.2 Services

A page about services of dream organizer.



Services Of Dreeam Orgainizer

Choose Your Favorite Food

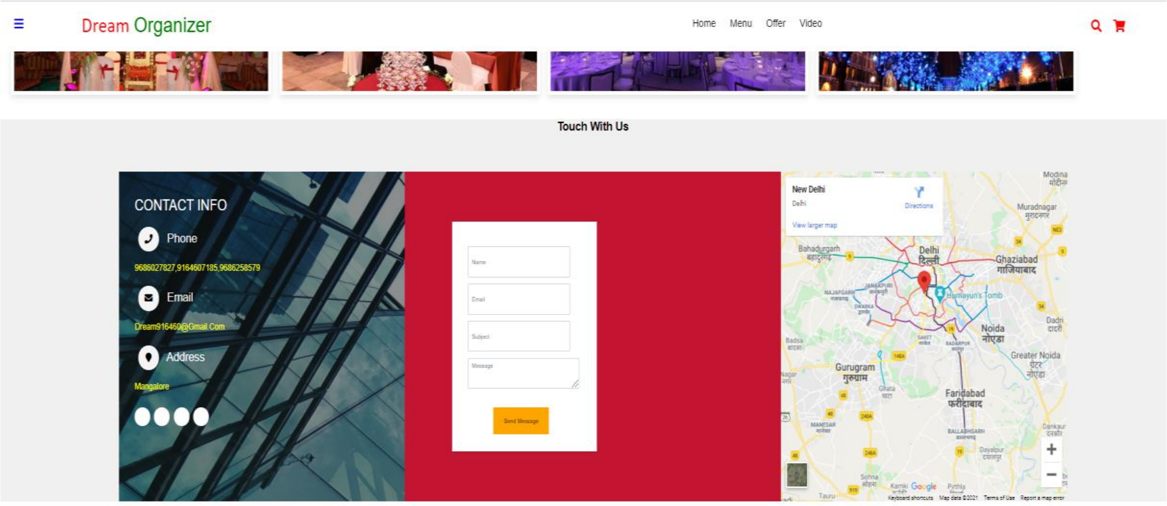
Free And Fast Delivery

Easy Payments Methods

Finally, Enjoy Your Food

7.2.3 Contact us

A page about contact us of dream organizer.



Touch With Us

CONTACT INFO

Phone

9686027827,9164607185,9686258579

Email

Dream916460@Gmail.Com

Address

Mangalore

Name

Email

Subject

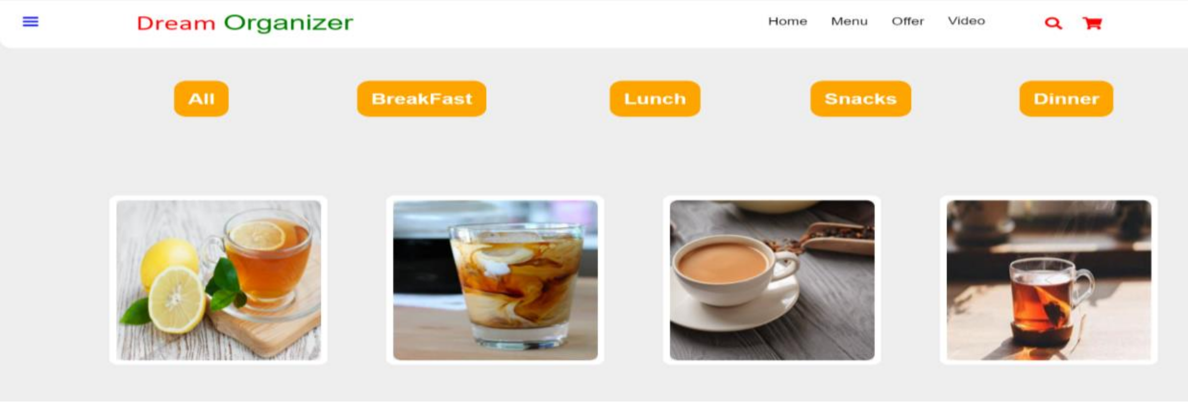
Message

Send Message

7.3 Menu

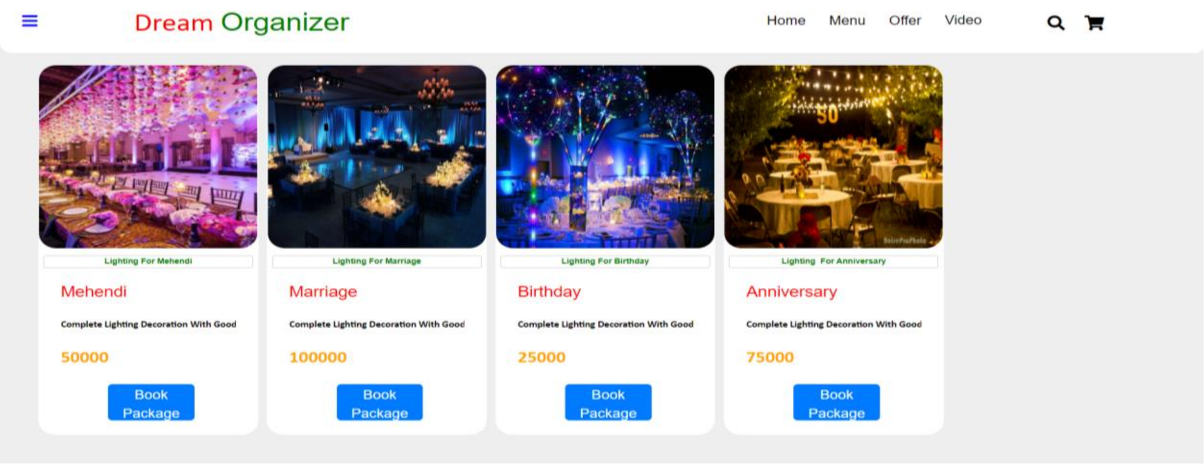
7.3.1 Food Menu

A page about food menu of dream organizer.



7.3.2 Lighting package

A page for lihgthing package.



Lighting For Mehendi
Mehendi
Complete Lighting Decoration With Good
50000

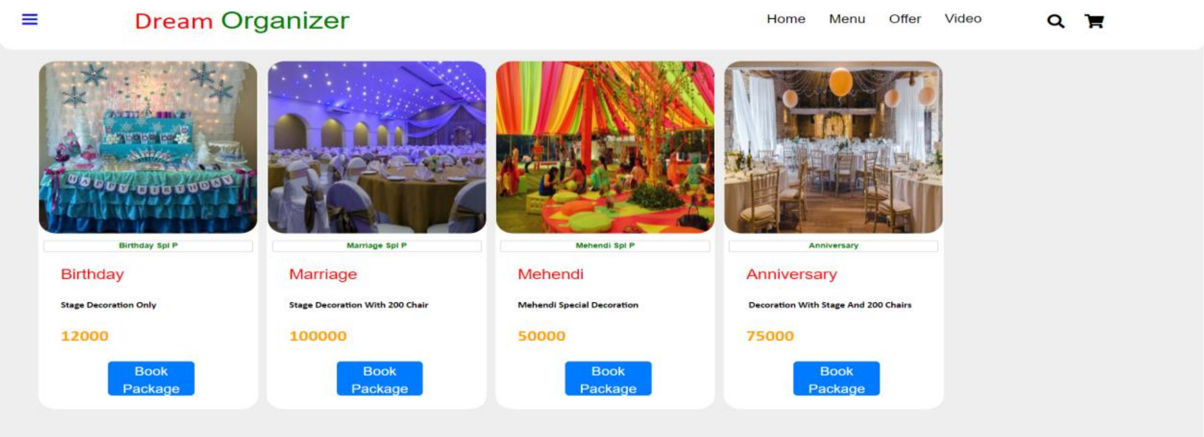
Lighting For Marriage
Marriage
Complete Lighting Decoration With Good
100000

Lighting For Birthday
Birthday
Complete Lighting Decoration With Good
25000

Lighting For Anniversary
Anniversary
Complete Lighting Decoration With Good
75000

7.3.2 Decorator package

A page for Decorator package.



Birthday Spl P
Birthday
Stage Decoration Only
12000

Marriage Spl P
Marriage
Stage Decoration With 200 Chair
100000

Mehendi Spl P
Mehendi
Mehendi Special Decoration
50000

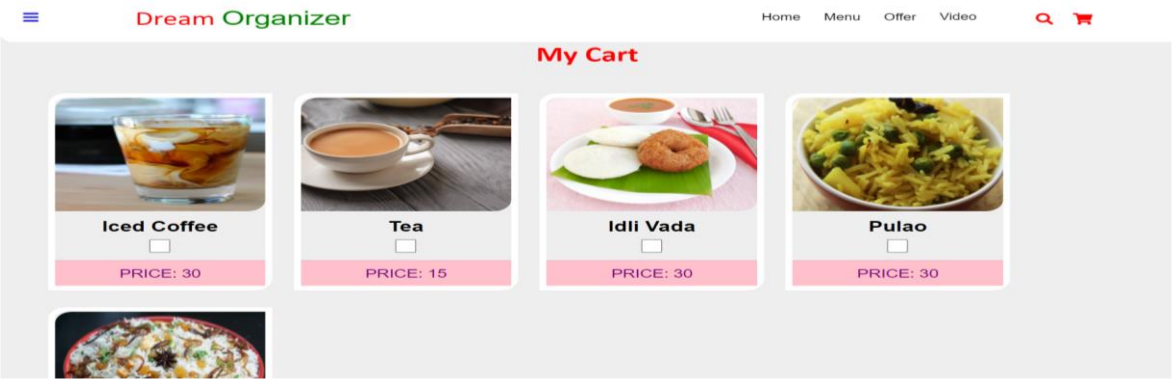
Anniversary
Anniversary
Decoration With Stage And 200 Chairs
75000

7.4 Data store/Retrieval/ Update

7.4.1 For user

7.4.1.1 User cart

A page for user cart.



My Cart

Iced Coffee
□
PRICE: 30

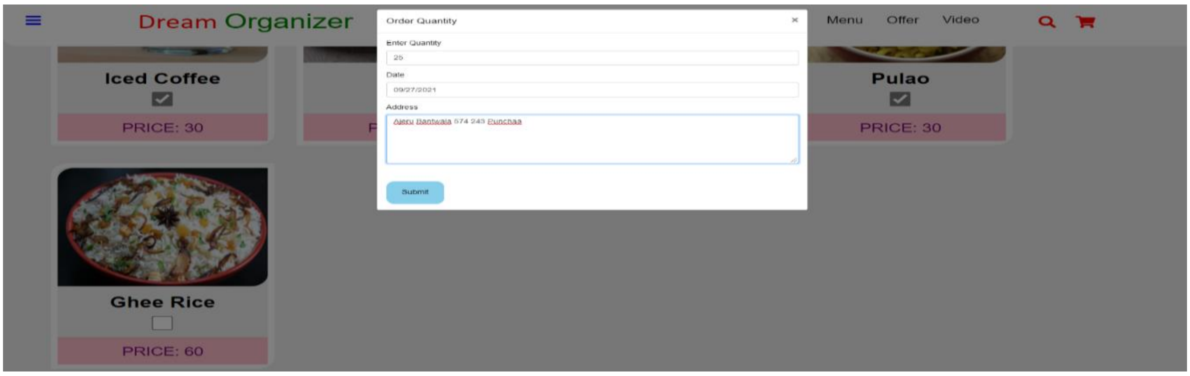
Tea
□
PRICE: 15

Idli Vada
□
PRICE: 30

Pulao
□
PRICE: 30

7.4.1.2 Order

A page for order.



Order Quantity ×

Enter Quantity
25

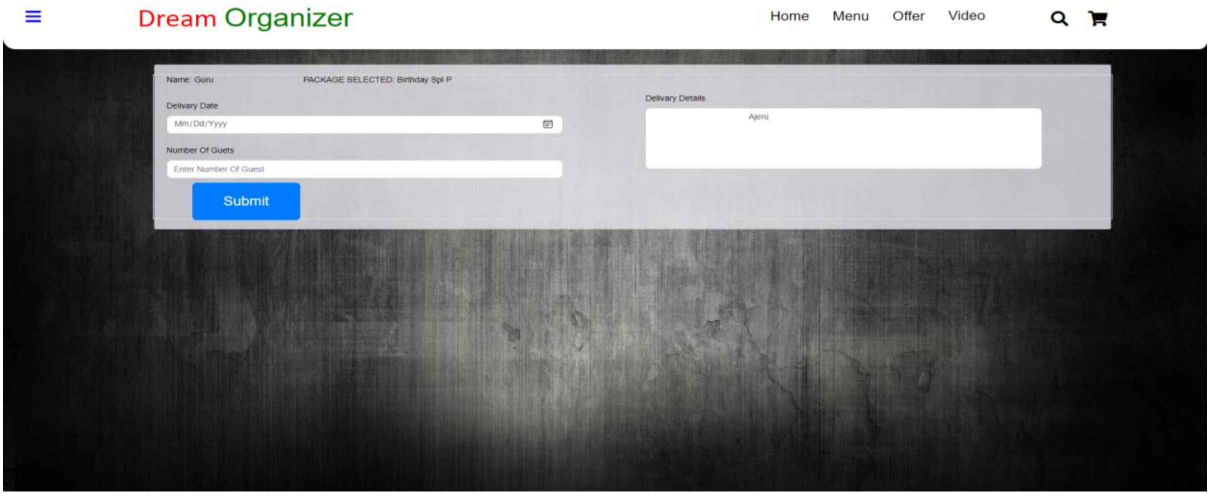
Date
09/27/2021

Address
Ajeru Bantwala 574 243 Punchaa

Submit

7.4.1.3 Package Order

A page for package order,here user can order package.



Name: Guru PACKAGE SELECTED: Birthday Spl P

Delivary Date
Mm/Dd/Yyyy

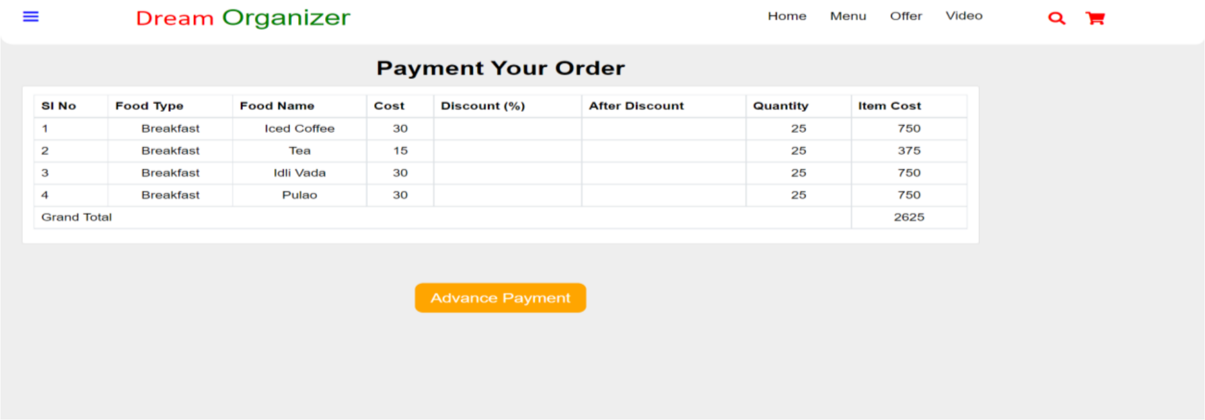
Number Of Guets
Enter Number Of Guest

Submit

Delivary Details
Ajeru

7.4.1.4 Payment

A page for payment, here user can pay according to his order.



Payment Your Order

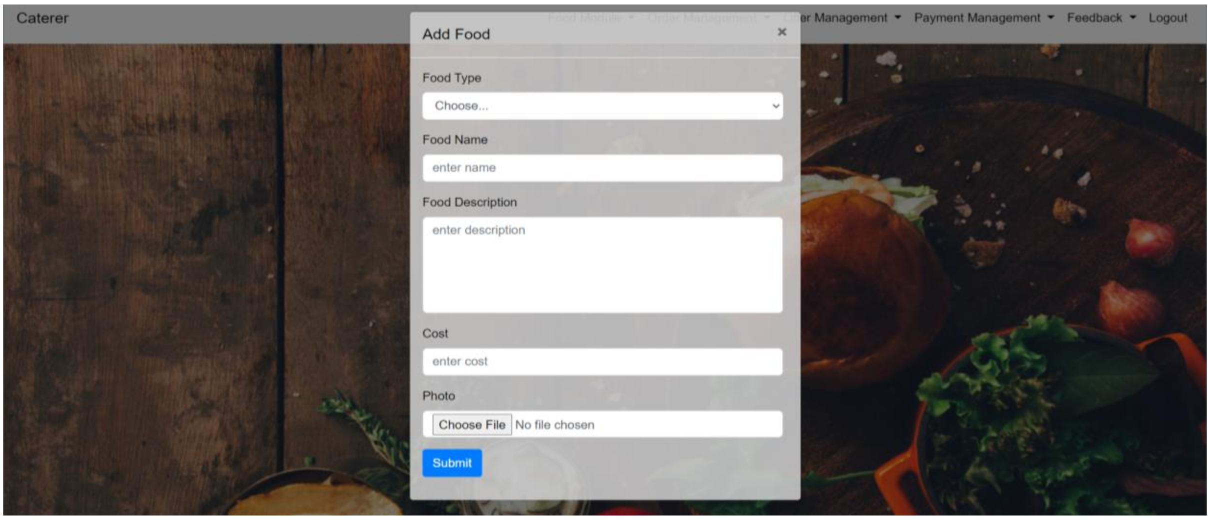
Sl No	Food Type	Food Name	Cost	Discount (%)	After Discount	Quantity	Item Cost
1	Breakfast	Iced Coffee	30			25	750
2	Breakfast	Tea	15			25	375
3	Breakfast	Idli Vada	30			25	750
4	Breakfast	Pulao	30			25	750
Grand Total							2625

Advance Payment

7.4.2 For Caterer

7.4.2.1 Add food

Add food page for caterer, he can add food.



Add Food

Food Type
Choose...

Food Name
enter name

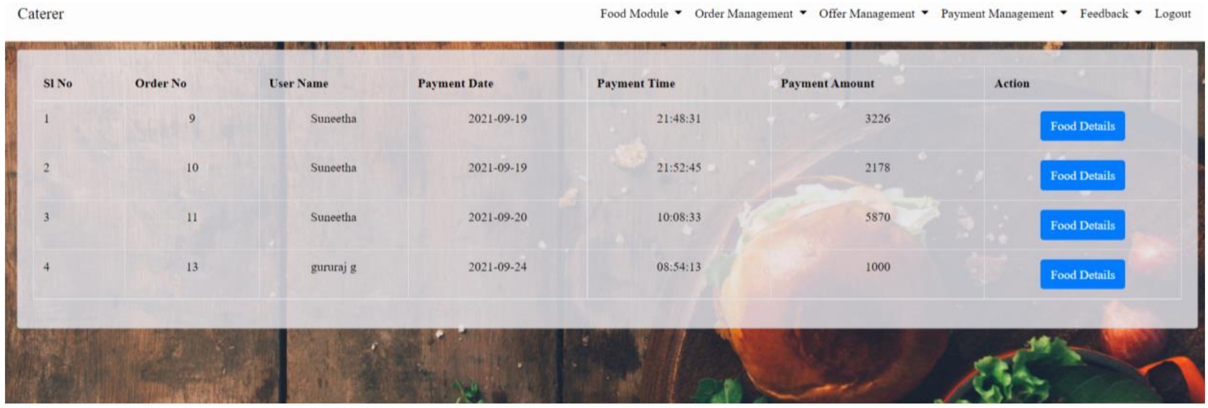
Food Description
enter description

Cost
enter cost

Photo
No file chosen

7.4.2.2 Payment Management

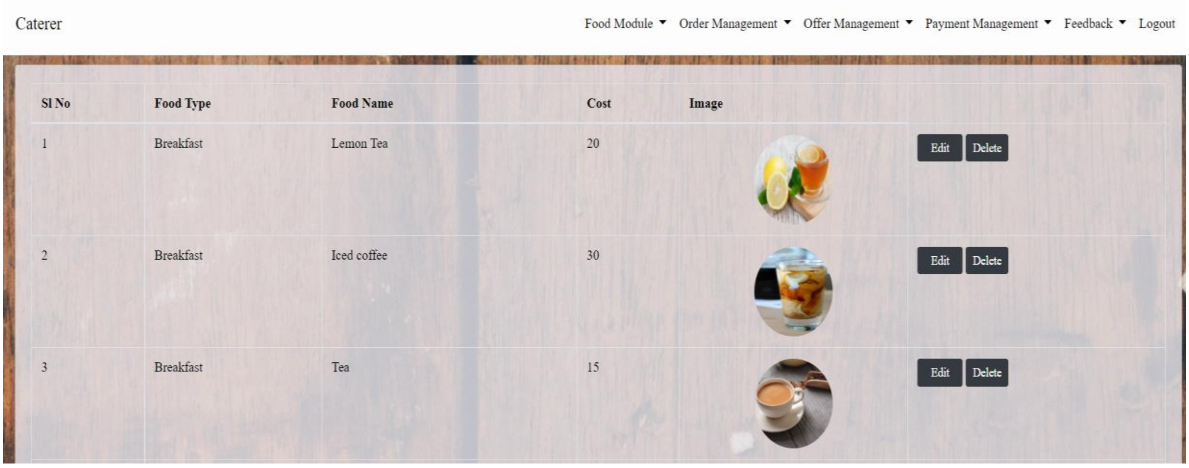
Payment management for caterer.



Sl No	Order No	User Name	Payment Date	Payment Time	Payment Amount	Action
1	9	Suneetha	2021-09-19	21:48:31	3226	Food Details
2	10	Suneetha	2021-09-19	21:52:45	2178	Food Details
3	11	Suneetha	2021-09-20	10:08:33	5870	Food Details
4	13	guruaj g	2021-09-24	08:54:13	1000	Food Details

7.4.2.3 View food

View food page for caterer, he can edit or delete food.

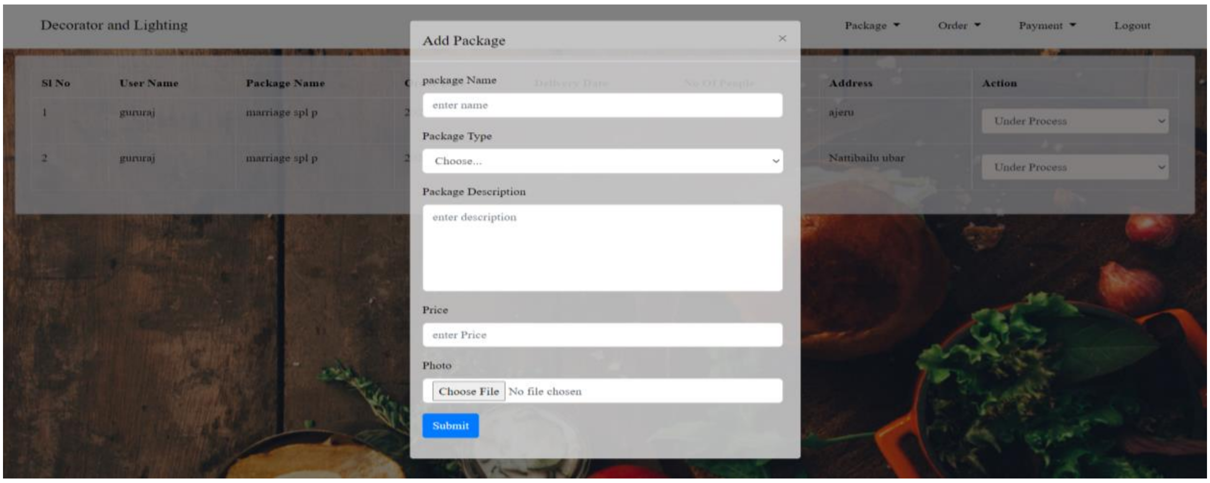


Sl No	Food Type	Food Name	Cost	Image	
1	Breakfast	Lemon Tea	20		Edit Delete
2	Breakfast	Iced coffee	30		Edit Delete
3	Breakfast	Tea	15		Edit Delete

7.4.3 For Decorator & Lighting

7.4.3.1 Add Package

Add package page of decorator & lighting, here they can add packages.



package Name

enter name

Package Type

Choose...

Package Description

enter description

Price

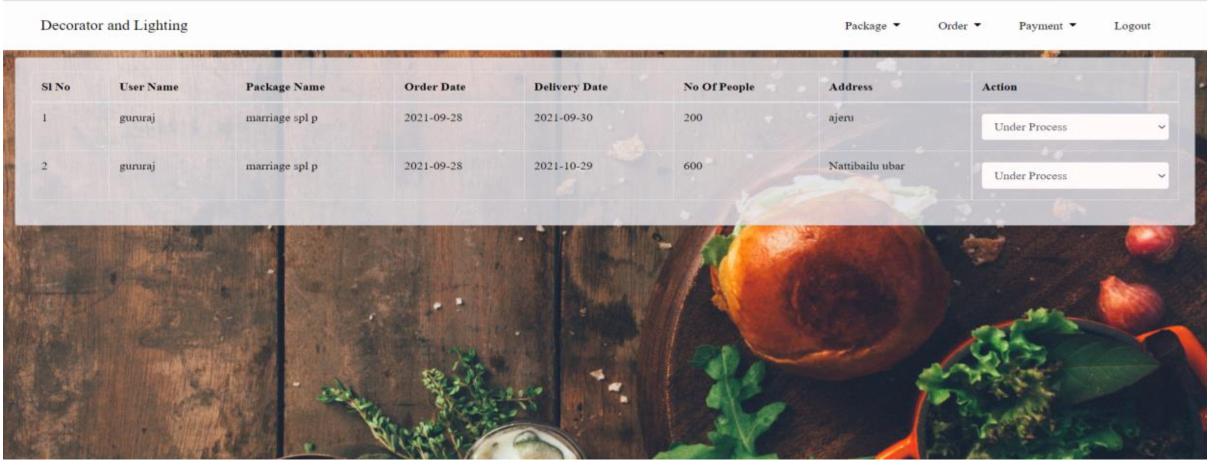
enter Price

Photo

Choose File No file chosen

7.4.3.2 Order Management

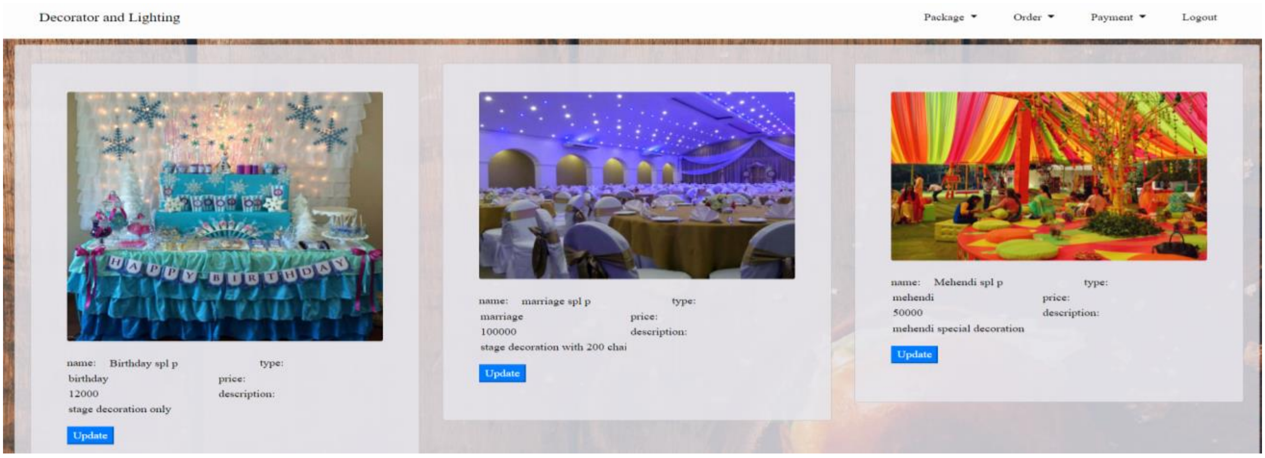
Order Management page of decorator & lighting, here they can do accept or reject order.



Sl No	User Name	Package Name	Order Date	Delivery Date	No Of People	Address	Action
1	gururaj	marriage spl p	2021-09-28	2021-09-30	200	ajeru	Under Process
2	gururaj	marriage spl p	2021-09-28	2021-10-29	600	Nattibailu ubar	Under Process

7.4.3.3 View Package

View Package page of decorator & lighting, here they can view and update package.



name: Birthday spl p type:
birthday price:
12000 description:
stage decoration only

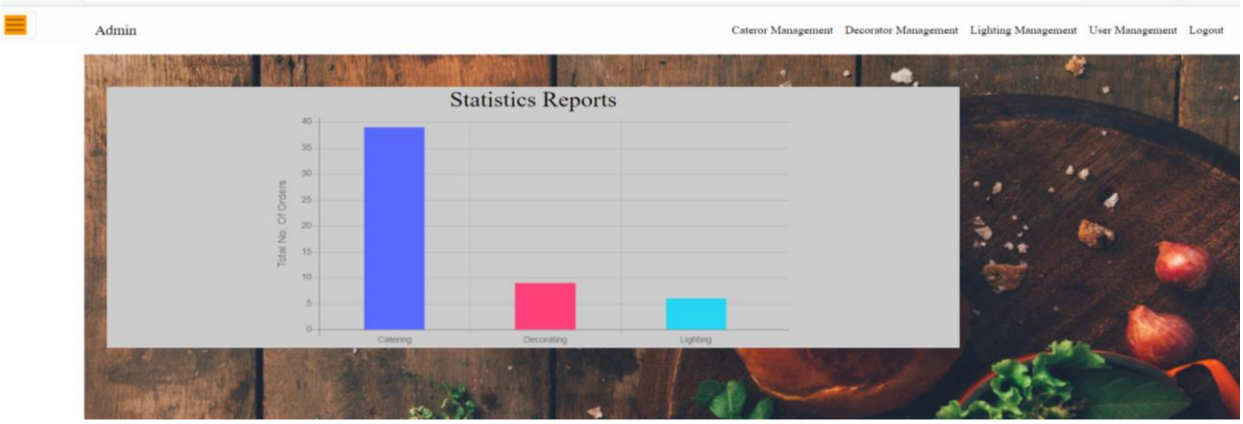
name: marriage spl p type:
marriage price:
100000 description:
stage decoration with 200 chai

name: Mehendi spl p type:
mehendi price:
50000 description:
mehendi special decoration

7.4.4 Admin

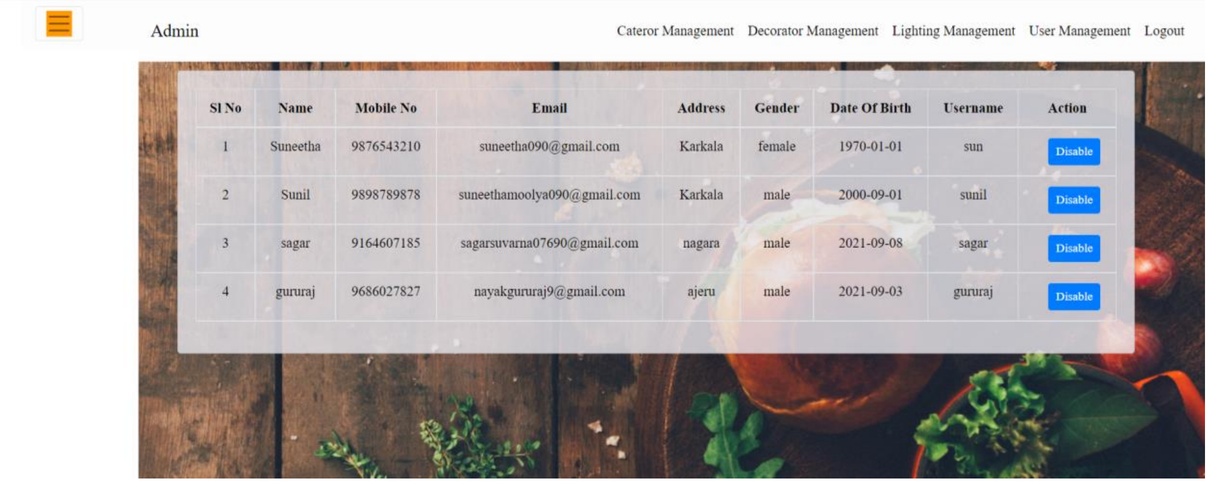
7.4.4.1 Statistics Report

Statistics report of dream organizer.



7.4.4.2 User Management

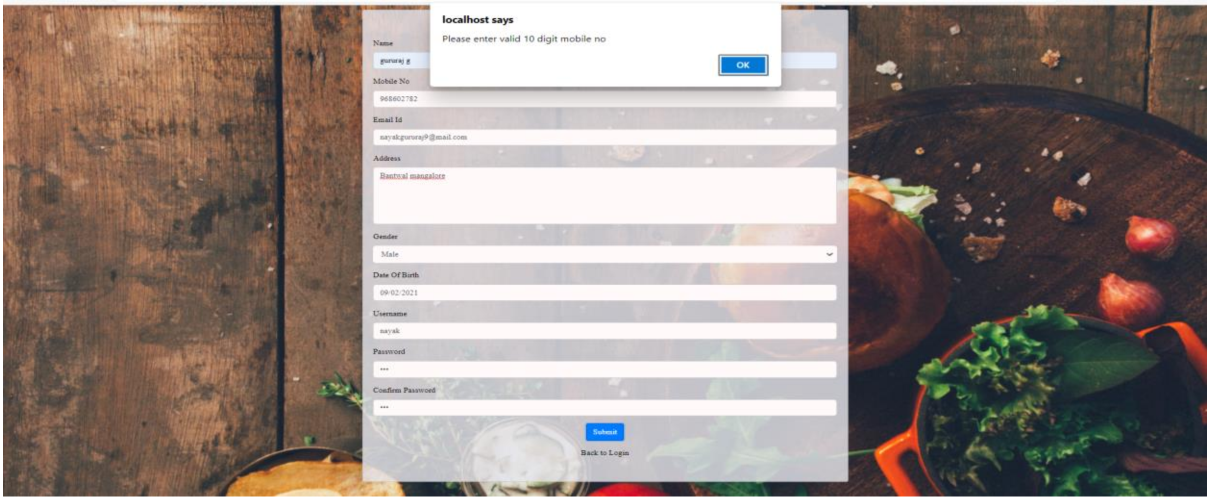
This is the page for user manage, admin can disable the user.



Sl No	Name	Mobile No	Email	Address	Gender	Date Of Birth	Username	Action
1	Suneetha	9876543210	suneetha090@gmail.com	Karkala	female	1970-01-01	sun	Disable
2	Sunil	9898789878	suneethamoolya090@gmail.com	Karkala	male	2000-09-01	sunil	Disable
3	sagar	9164607185	sagarsuvarna07690@gmail.com	nagara	male	2021-09-08	sagar	Disable
4	gururaj	9686027827	nayakgururaj9@gmail.com	ajeru	male	2021-09-03	gururaj	Disable

7.5 Data Validation

Checks if values are entered incorrect.



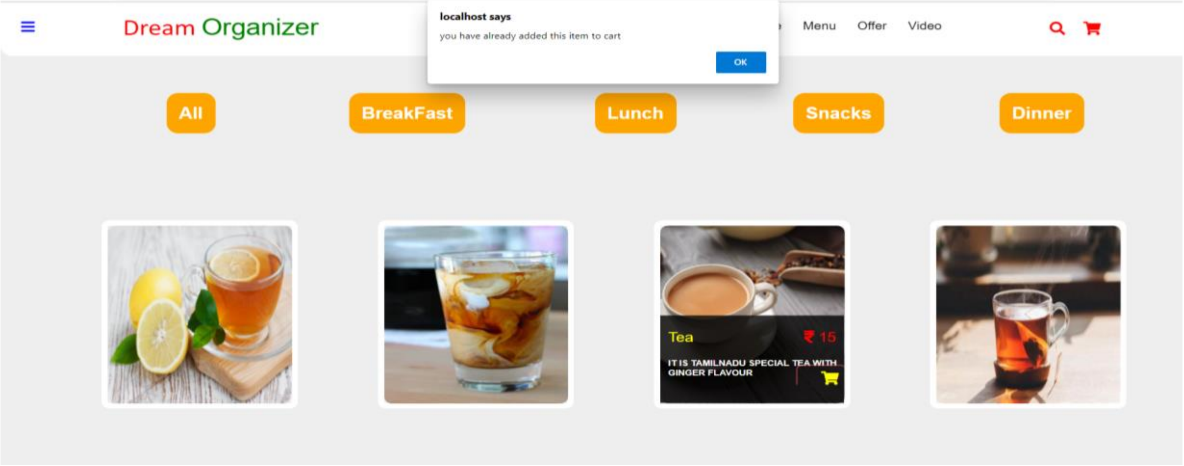
localhost says

Please enter valid 10 digit mobile no

Name	gururaj g
Mobile No	968602782
Email Id	nayakgururaj9@mail.com
Address	Bantwal mangalore
Gender	Male
Date Of Birth	09/02/2021
Username	nayak
Password	•••
Confirm Password	•••

7.6 Alerts

If already food in cart , it display alert message.



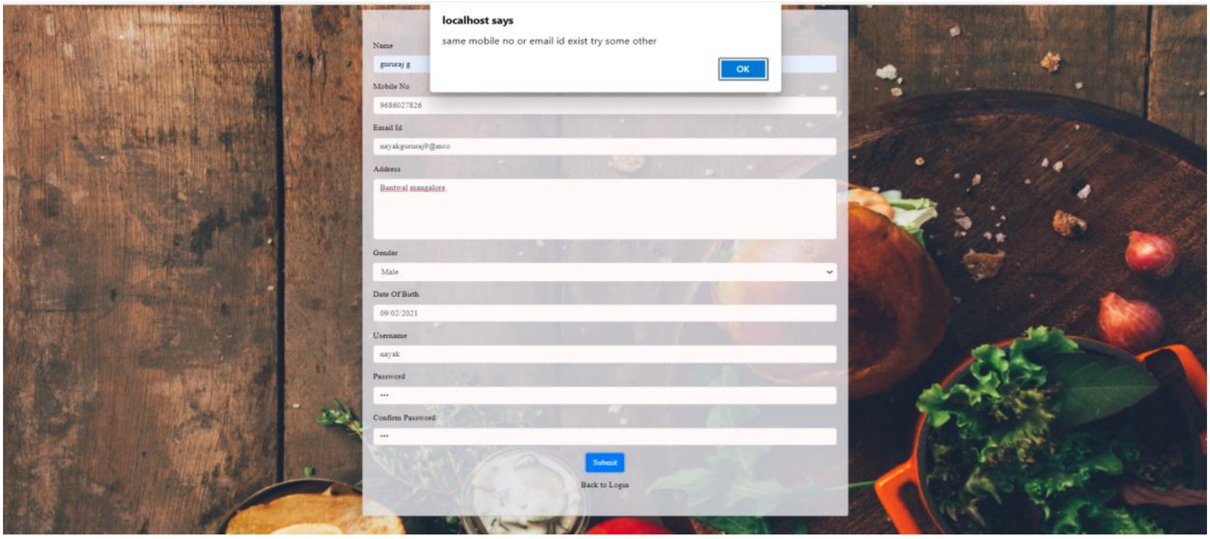
localhost says

you have already added this item to cart

OK

7.7 Error Message

If we entered incorrect values ,it shows error messages.



localhost says

same mobile no or email id exist try some other

OK

Name

gururaj g

Mobile No

9686027826

Email Id

nayakgururaj9@mco

Address

Bantwal mangalore

Gender

Male

Date Of Birth

09/02/2021

Username

nayak

Password

•••

Confirm Password

•••

Submit

Back to Login

Chapter-8

SYSTEM TESTING

8.1 INTRODUCTION

Testing is a process of executing a program with the explicit intention of finding error. It is a process used to identify correctness, completeness and quality of developed computer software. There are many approaches to software testing, but effective testing of complex product is essential a process of investigating. Testing helps in verifying and validating if he software working as it is intended to work. This involves using static and dynamic methodologies to test the application. There are two methods for test case design.

White Box testing:

White box testing strategy deals with the internal logic and structure of the code. It is also called as glass, structural, open and clear box testing. The test that are written based on the white box testing strategy incorporate coverage of the code written, branches, statements and internal logic of the code etc i.e. Internal working of the code.

Advantages:

- As the knowledge of the internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively
- It helps in optimizing the code
- It helps in removing the extra line of code, which introduce defect in the code

Disadvantage

- As the knowledge of code and internal structure is a prerequisite, a skilled tester is needed to carry out this type of testing, and this, in turn, increase the cost of the software
- It is nearly impossible into every bit of code to find out the hidden errors, which may create problems, resulting in failure of application

Black Box testing

Black box testing takes the internal perspective of the test object to derived test cases. These tests can be functional or non-functional though usually functional. The test designer selects valid and invalid inputs and determines the correct input. There is no knowledge of the test object’s internal structure. This method of test design is applicable to all levels of software testing: unit, internal, functional and system and acceptance

Advantages:

- Black box test are reproducible
- The environment in which the program is running is also tested
- The invested effort can be used multiple times

Disadvantages:

- The results are often over estimated
- Not all properties of the software can be tested
- The reason for failure is not found

8.2 Test Report:

- Unit Testing
- Integration Testing
- System Testing

8.2.1 Unit Testing

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage producers, are tested to determine if they are fit to use. Intuitively, one can view a unit as the smallest testable part of an application, In procedural programming a unit could be an entire module but is more commonly an individual function or procedure.

8.2.2 Integration Testing

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These design items, i.e. assemblages (or group of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs.

8.2.3 System Testing

A system testing of software or hardware is testing conducted on a complete, integrated system to evaluate system’s compliance with its specified requirements. System testing falls within the scope of black box testing, and such as, should require no knowledge of the inner design of the integrated software components that have successfully passed integration testing and also the software components itself integrated with any applicable hardware system.

8.3 Test cases

- **User Login**

S.No	Action	Inputs	Expected Output	Actual Output	Test Result
1	If user clicks on login button without entering username and password.	Inputs are not given	Please enter username and password	Please enter username and password	Pass
2	Enter correct username & Password and hit login button	username test@xyz.com Password: ******	Login success	Login success	Pass
3	If username is blank but password is entered.	Password: ******	Please enter username	Please enter username	Pass
4	If password is blank but username is entered.	Username: abcd	Please enter password	Please enter password	Pass
5	If the username or password is incorrect.	If the username or password is incorrect.	Invalid login id or Password	Invalid login id or Password	Pass

- **User Registration**

S.No	Action	Inputs	Expected Output	Actual Output	Test Result
1	When user click on the Create new account	Inputs are not given	Displays registration form	Displays registration form	Pass

2	user Enters Name, Mobile No, email, Address, Gender, DOB, and clicks on signup.	Abcd 8989898989 abcd@gmail.com Mumbai Male 15/05/1998 avi	Successfully registered	Successfully registered	Pass
3	If any fields are blank	Xyz Shetty Password: *******	Please enter Mobile no	Please enter Mobile no	Pass

- **Caterer Login**

SI NO	Action	Inputs	Expected Output	Actual Output	Test Result
1	If the Caterer clicks on button “Login”	Inputs are not given	Please enter username and password	Please enter username and password	Pass
2	When the Caterer clicks on “Login” with valid username and password	username Caterer Password: ******	Login success	Login success	Pass
3	If username is blank but password is entered.	Password: ******	Please enter username	Please enter username	Pass
4	If password is blank but username is entered.	Username: Caterer	Please enter password	Please enter password	Pass

5	If the username or password is incorrect.	If the username or password is incorrect.	Invalid login id or Password	Invalid login id or Password	Pass

- **Decorator Login**

SI NO	Action	Inputs	Expected Output	Actual Output	Test Result
1	If the Decorator clicks on button "Login"	Inputs are not given	Please enter username and password	Please enter username and password	Pass
2	When the Decorator clicks on "Login" with valid username and password	username decorator Password: ******	Login success	Login success	Pass
3	If username is blank but password is entered.	Password: ******	Please enter username	Please enter username	Pass
4	If password is blank but username is entered.	Username: Decorator	Please enter password	Please enter password	Pass
5	If the username or password is incorrect.	If the username or password is incorrect.	Invalid login id or Password	Invalid login id or Password	Pass

➢ Admin Login

SI NO	Action	Inputs	Expected Output	Actual Output	Test Result
1	If the Admin clicks on button "Login"	Inputs are not given	Please enter username and password	Please enter username and password	Pass
2	When the admin clicks on "Login" with valid username and password	username lighting Password: ******	Login success	Login success	Pass
3	If username is blank but password is entered.	Password: ******	Please enter username	Please enter username	Pass
4	If password is blank but username is entered.	Username: Decorator	Please enter password	Please enter password	Pass
5	If the username or password is incorrect.	If the username or password is incorrect.	Invalid login id or Password	Invalid login id or Password	Pass

➢ Feedback

SI NO	Action	Inputs	Expected Output	Actual Output	Test Result
1	If the select food field and write here is empty	Inputs are not given	Please fill out this field	Please fill out this field	Pass

2	If the select food field is empty Write here is wrote	Select food: empty Write here: super	Please select food	Please select food	Pass
3	If the select food field is selected Write here is empty	Select food: idli Write here: empty	Please write here	Please write here	Pass
4	If the select food field is selected Write here is wrote	Select food: idli Write here: Super	Feedback Sent	Feedback Sent	Pass

- **Add Food**

SI NO	Action	Inputs	Expected Output	Actual Output	Test Result
1	If the add food field and image field is empty	Inputs are not given	Please fill out this field	Please fill out this field	Pass
2	If the add food field is empty and image is selected	Add food: empty Image: choose	Please fill the add food field	Please fill the add food field	Pass
3	If the add food field is filled and image is empty	Add food: filled Image: empty	Please choose image	Please choose image	Pass

4	If the add food field is selected and image is added	Add food: Filled Image: selected	Food added	Food added	Pass

➢ Add Package

SI NO	Action	Inputs	Expected Output	Actual Output	Test Result
1	If the add package field and image field is empty	Inputs are not given	Please fill out this field	Please fill out this field	Pass
2	If the add package field is empty and image is selected	Add package: empty Image: choose	Please fill the add package field	Please fill the add package field	Pass
3	If the add package field is filled and image is empty	Add package: filled Image: empty	Please choose image	Please choose image	Pass
4	If the add package field is selected and image is added	Add package: Filled Image: selected	Package added	Package added	Pass

- **Contact us:**

SI NO	Action	Inputs	Expected Output	Actual Output	Test Result
1	If the name field is empty	Name is empty. Email ,Subject and message is filled	Please enter name field	Please enter name field	Pass
2	If the email field is empty	Email is empty. Name ,Subject and message is filled	Please enter email field	Please enter email field	Pass
3	If the subject field is empty	Subject is empty. Name, Email and message is filled	Please enter subject field	Please enter subject field	Pass
4	If the Message field is empty	Message is empty. Name, Email and subject is filled	Please enter Message field	Please enter Message field	Pass
5	If all the fields are filled	Name, Email ,subject and message is filled	Mail sent	Mail sent	Pass

CONCLUSION

The system that has been presented is mainly used in largescale catering enterprises, decoration and lightings for functions and festival, such as wedding and parties. The system is committed to provide consumers with a healthy and dietary nutrition, quality and standard decoration and lighting products. As for dream organizer, it provides an automatic ordering food, decoration and lighting products and checkout to cut down costs of variety of foods and to checkout different types of packages available, and still can provide a transparent management, sold out statistics. So, the system can reduce the unsalable food and different types of packages to improve the Services.

LIMITATIONS

- Proposed system should not limit the user from order based on their location. System predicts that users are staying in the nominal distance to which food and lighting equipment can be delivered.
- User of the system should approach the caterer and lighting owner to cancel the order.
- User must contact owner to add additional food if it is not listed in menu.

SCOPE FOR ENHANCEMENT

- This web application can be extended to support different types of payments modes that helps the user.
- An android application can be developed to help the users to access the system easily and effectively.
- We can link this project to internet, so anyone can use this.
- The system is developed with the intrusion of future expansion of the concern.
- The concern can go for any future developments, as per their needs. If so, they can easily
- develop the modules and integrated with the existing system.

ABBREVIATION AND ACRONYMS

I/O	-	Input / Output.
OS	-	Operating System.
DFD	-	Data Flow Diagram.
CSS	-	Cascading style sheet.
ADMIN	-	The Administrator.
CFD	-	Context Flow Diagram.
PHP	-	Hypertext Preprocessor.
GUI	-	Graphical User interface.
RAM	-	Random Access Memory.
SQL	-	Structured Query Language.
HTML	-	Hyper Text Markup Language.
SRS	-	Software Requirement Specifications.
IEEE	-	Institute of Electrical and Electronic Engineering.

BIBLIOGRAPHY

- **Books**

➢ AN INTEGRATED APPROCH TO SOFTWARE ENGNEERING.

Author-Pankaj Jalote.

Publication: Narosa Publishing House.

- **Websites**

➢ http://www.w3schools.com

➢ http://www.phpbuddy.com

➢ http://www.mysql.com

➢ http://www.youtube.com

➢ http://www.codeacademy.com

➢ http://www.bootstrap.com

➢ http://www.kashipara.com