ONLINE FOOD ORDERING SYSTEM

PROJECT REPORT

Submitted in the partial fulfilment of the requirements for the award of degree in

MASTER OF SCIENCE IN COMPUTER SCIENCE

FOURTH SEMSTER

Submitted By,

SUJITH. V

Register Number : 180011012596

SWAMY SASWATHIKANANDA COLLEGE

POOTHOTTA

(Affiliated to Mahatma Gandhi University)

2020

SWAMY SASWATHIKANANDA COLLEGE, POOTHOTTA

(Affiliated to Mahatma Gandhi University)

CERTIFICATE

This is to verify that project entitled "**ONLINE FOOD ORDERING SYSTEM**" submitted in the partial fulfilment of the requirement for the award of the degree in "**MASTER OF SCIENCE IN COMPUTER SCIENCE**" is a bonafied report of the project done by **SUJITH .V (180011012596)**, during the fourth semester, year 2020.

Internal Guide **Head of the department**

Examiners : 1…….

2…….

College Seal **Department Seal**

DECLARATION

I hereby declare that this project work entitled "**ONLINE FOOD ORDERING SYSTEM**" is a record of original work done by us under the guidance of **Mrs RAJALAKSHMI K.R** Asst.Professor, Department of Computer Science, **SWAMY SASWATHIKANANDA COLLEGE**, and the project work has not formed the basic for the award of any Degree/Diploma or similar title to any candidate of University.

Internal Guide **Signature of Candidate**

MRS RAJALAKSHMI K.R **SUJITH.V**

ACKNOWLEDGEMENT

I express my gratitude to prof. Dr. S.BABU SUNDAR the principal of **SWAMY SASWATHIKANANDA COLLEGE POOTHOTTA** for providing me with adequate facilities, ways and means by which I was able to complete the project work.

I express my sincere thanks to **ASST PROF Mrs. DIVYA.R**, Head of the Department of Computer Science, and my project coordinator and my project guide **Mrs. RAJALAKSHMI K.R** who has been showing deep interest in my project and inspired me through development by valuable suggestion, and all the faulty members of the Development of Computer Science for their and support.

Last but not the least, I also express my profound gratitude to all other members of the faculty and well wishers who assisted in various occasion during the project work.

CONTENTS

1. SYNOPSIS

The main Objective of this project on "ONLINE FOOD ORDERING SYSTEM" is to manage the details of Food item, Category, Cart, Order and Customer. It manages all information about Food item, Delivery, Customer. The project is built at administrative end and thus only the administrator is guaranteed the access. The purpose of Online Food Ordering System is to build an application system to reduce the manual work for managing the Food item, Category, Cart and Delivery. It tracks all the details about the Cart, Order and Customer. This system is developed in Java language with NetBeans IDE as platform and Wampserver as backend.

2. INTRODUCTION

The project entitled "ONLINE FOOD ORDERING SYSTEM" is a process of ordering food from a local restaurant or food cooperative through internet. A customer will search for a favourite restaurant, usually filtered via type of cuisine and choose from available items, and choose delivery or pick-up. This Online Food Ordering System intends different types of forms with different of food varieties provides to user to buy online. Online Food Ordering System users can give order from any place or pay cash on delivery. The system deals with ordering, process and delivering food products. Ordering is done by valid customer with appropriate identity.

The main Objective of this system is to manage the details of Food item, Category, Cart, Order and Customer. It manages all information about Food item, Delivery, Customer. The project is built at administrative end and thus only the administrator is guaranteed the access. The purpose of Online Food Ordering System is to build an application system to reduce the manual work for managing the Food item, Category, Cart and Delivery. It tracks all the details about the Cart, Order and Customer.

3. SYSTEM ANALYSIS

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information about the Online Food Ordering System to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest details and analysed. The system is viewed as a whole and the input to the system are identified. A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The conclusion is an understanding of how the system functions.

3.1 EXISTING SYSTEM

In existing system for giving any orders users should visit hotels or restaurants to know about food items and them give order and pay advance. In this method time and manual work is required. Maintaining critical information in the files and manuals is full of risk and a tedious process.

3.2 PROPOSED SYSTEM

The proposed system helps in many ways. It helps to do billing very easily. Account maintenance also becomes easier. They can keep track of their purchases of inventories, staff details, sales of foods and account details etc. the software is provided with the facilities to find out the favourite food of the customers, and customers to add or modify and delete their feedbacks and suggestions. It helps

in managing data of different type of orders like party order, home delivery or the normal order. Managing data of daily customers, managing data of staffs, managing data of daily expenses. It eliminates the drawbacks of existing system and able includes some more features.

ADVANTAGE OF PROPOSED SYSTEM

The proposed system has many advantages over the existing. They are

- All the records of the current and staffs will be stored in the database.
- All the shift timings of staffs, salary, extra working time and charge etc. are also updated daily.
- Managers can view the kitchen side, dinner room and counter side details simultaneously and also can get details and updates from various branches.
- Any cancelations of parties are also updated.
- All the expenses per day will updated in the database daily.
- Also keeps the record of food items prepared and the sales of food and also the record of the balance food.
- Profits and losses will be updated every month and also shows the variations.
- All the food order details like the order types (normal, home delivery, party order etc.) are stored daily.

3.3 MODULES

The three main modules of this projects are

1. User /customer

2. Employee/staff
3. Administrator
4. Restaurant

   Homepage has following menu

   - Home
   - Gallery
   - About
   - Contact

USER MODULES

1 **User registration**: Any users can register on website using the registration form.
2 **User login**: User can login into the system.
3 **Food items details**: User can see all food items from the form.
4 **Cart**: This is the cart form in the project.
5 **User shipping**: This is the user shipping form in the project.
6 **User payment**: This is the user payment from in the project.
7 **User order**: This is the user order from in the project.
8 **User order confirmation**: This is the user order confirmation form where user will be able to confirm order.

ADMINISTRATOR MODULES

1 Admin can add the employee records.
2 Admin can see the lists of employee details.
3 Admin can edit and update the records of employees.
4 Admin will be able to delete the records of employees.
5 All employee forms are validated on client side using JavaScript.

EMPLOYEE MODULES

6 Employee can add the food item.
7 Employee can see the details of all the food items.
8 Employee can edit and update the records of food items.
9 Employee will be able to delete the records of food items.
10 Employee can see the user registered details.

11 Employee can see the view orders.

RESTAURANT MODULES

1 **The list of restaurants**: User can choose all restaurant lists (e.g: KFC, Dominos and McDonalds etc.)
2 **Food items details**: User can see all food items from the restaurants.
3 **Cart**: This is the cart form in the project.
4 **User shipping**: This is the user shipping form in the project.
5 **User payment**: This is the user payment from in the project.
6 **User order**: This is the user order from in the project.

3.4 FEASIBILITY STUDY

In the project online food ordering system, study and analysing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible given unlimited resources and infinite time.

Feasibility study includes consideration of all the possible ways to provide a solution to given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so the future changes can be easily done based on the future upcoming requirements.

There are three aspects in the feasibility study portion of the preliminary investigation

- Technical feasibility
- Economic feasibility
- Operational feasibility

3.4.1 TECHNICAL FEASIBILITY

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System

Requirement Specification (SRS) and checked if everything was possible using different type of frontend and backend platforms.

3.4.2 ECONOMIC FEASIBILITY

This is very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor.

- All hardware and software cost has to be borne by the organization.
- Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost form system.

3.4.3 OPERATIONAL FEASIBILITY

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self- explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with new system. As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

4. REQUIREMENT ANALYSIS

Requirement analysis is a process of discovery refinement modeling and specification. The analysis can be divided into:

- Problem recognition
- Problem evaluation and synthesis
- Modelling

4.1 PROBLEM RECOGNITION

The goal of this step is recognition of basic problems of elements as indicated by customer. The basic purpose of this activity is to obtain a through understanding of the needs of client and user what exactly is desired from the software is the constraints on solution. The main problem of existing system are:

- Time consuming.
- More man power is wasted.
- Less accuracy, and data may be lost.

4.2 PROBLEM EVALUATION AND SYNTHESIS

In this step analyst must define all externally observable object, evaluating flow and control of step of information, define and elaborate all software function, understand the software behavior and design constraints etc. Evaluation and synthesis continues until both analyst and customer felt confident about the project. Once the problem identified, the evaluation process begins. After evaluation of current problem and the desired information, the analyst may synthesis one or more solution.

- Cost effectiveness.
- Faster and records are accurate.

- More secure and easy retrieval of data.

4.3 MODELLING

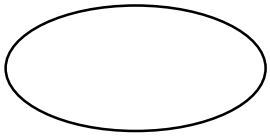
During a software requirement analysis, we create models to gain better understanding of actual logical entity to be built. The model helps the analyst to understand information and function of the system. The model became the main reference for the review to determine completeness, the main method used for this is DFD (Data Flow Diagram)

DATA FLOW DIAGRAM (DFD)

Data Flow Diagram (DFD) provides a visual representation of the flow of information (i.e. data) within a system. By drawing a Data Flow Diagram, you can tell the information provided by and delivered to someone who takes part in system processes, the information needed to complete the processes and the information needed to be stored and accessed.

- It represent a source or destination data of the system.

- It represent flow of data.



- It represent a process that transforms data.

- It represent data store (eg : table).

5. SYSTEM SPECIFICATION

5.1. SOFTWARE SPECIFICATION

OPERATING SYSTEM	:	WINDOW 10
PLATFORM	:	NETBEANS IDE 8.0.2
LANGUAGE	:	JAVA LANGUAGE
BACK-END	:	WAMPSERVER 64
USER INTERFACE DESIGN	:	ADOBE PHOTOSHOP CS5

5.2. HARDWARE SPECIFICATION

PROCESSOR	:	INTEL CORE I5 8TH GEN
RAM	:	8GB
HARD DISK	:	1TB
SYSTEM TYPE	:	64-BIT OPERATING SYSTEM
SCREEN RESOLUTION	:	1920 X 1080 PIXELS
GRAPHICS CARD	:	NVIDIA

6. SYSTEM DESIGN

System design is a process of developing specification for a candidate system that meet the criteria established in the system analysis. Major steps in design are the preparation of the input forms and output reports in a form applicable to user.

The main objective of the system design is to use the package easily be any computer operator. System design is the creative act of invention, developing new inputs, a database, offline files, method, procedure and output for processing business to meet an organization objective. System design builds information gathered during the system analysis.

6.1 DATA DESIGN

Data design creates a model of data or information that is represented at a higher level of abstraction. The structure of data has always been an important part of software design. The software design activity translates this requirement model into the data structure at the software component level.

Data design is required to manage the large volume of information. In this system, normalization process, the redundant field will be eliminated and finally produce the efficient table. "**ONLINE FOOD ORDERING SYSTEM**" has developed an efficient database that satisfies the condition below

- Ease of use
- Privacy and security
- Data dependent
- Performance

6.2 ARCHITECTURAL DESIGN

Architectural design is a comprehensive frame work that describes its form and how they fit together. The properties of components interact with other components. Architectural design focuses on the representation of the structure of the software.

6.3 PROCEDURAL DESIGN

Procedural design or component level design occurs after data, architectural and interface design must be translated into operational software. The procedural design for each component, represented in graphical, tabular or text based notation, is the primary work product produced during component level design.

6.4 INTERFACE DESIGN

Interface design creates and effective communication between a human and computer. Design identifies objects and actions then creates a screen layout that forms the basis for user interface.

Interface design focus on:

1. The design of interfaced between software components and non-producers and consumers of information.
2. The design of interface between software components.
3. The design of interface between a human and computer.

7. CODING

A code provides a brief identifications of data item and replaces longer description that would be more awkward to store and manipulate. A code can be defined as a group of characters used to identify an item of data, while identification is the main function of a code. A code may also show relationship between items of data.

A code plan identifies the particular characteristics that needed to be contained within the code. Only information that makes possible efficient identification and retrieval of coded items should be chosen. The method chosen must have the following features:

- Expandable: Code must provide space for additional entires that may be required.
- Precise: The code must identify the specific item.
- Concise: The code must be brief, yet it should adequately describe the item.
- Meaningful: The code must be useful to that people dealing with it. If possible, it should indicate some characteristics of the item.
- Operable: The code should be compatible with present and anticipated methods of data processing.

A code dictionary is often developed to make it easier for human to work with the codes. It is a listening of code and their corresponding data items. The dictionary allows one to translate the code into identification of the data or to determine the code for a particular item.

The code of "**ONLINE FOOD ORDERING SYSTEM**" is shown in the appendix.

8. SYSTEM TESTING

INTRODUCTION

The purpose of the system testing is to identify and correct errors in the candidate system. Testing is an important element of the software quality assurance and represent the ultimate review of specification, design and coding. The increasing visibility of the software as a system element and the costs associated with a software failure are motivated for well planned, through testing.

Software testing is a critical element of software quality assurance and represents the ultimate quality review of specifications, design and code generations. Once the source code has been generated, the program should be executed before the customer gets it with the specific intend of finding and removing all errors, test must be designed using disciplined techniques. Testing technique provide the systematic guidance for designing tests. To uncover the errors in the program behavior function and performance the following steps to be done:

- Execute the integral logic of the software components
- Execute the input and output domains of the program to uncover errors
- During testing the system is used experimentally to ensure that the software does not fail, i.e, it will run according to the specifications and in the way the user exports.
- Preparation of test data plays a vital rule in the system testing. Different set of test data are generated and the system under study is tested using that data.

- While testing using test data errors are again uncovered and corrected using different testing techniques.

System testing was conducted in order to detect errors and for comparing the final system with the requirement specification report. That is, whether the system meets requirements. During testing the software was executed with a set of test case and the output of the program for the cases was evaluated to determine if the program is performing as it was expected to.

Testing presents, an interesting challenge for the software engineer attends to hold software from an abstract concept to an acceptable implementation. In testing engineer creates a series of test cases that occurs when errors are uncovered. Testing is the process of executing a program for finding errors. A good test is one that has high probability of finding an uncovered error.

The turn error is used to refer the difference between the actual outputs of the software to fail to perform its required function. Software reliability is defined as the required function. Software reliability is defined as the probability that the software will not undergo failure for a specified time under specified condition. Failure is the inability of a system or a component to perform a required function according to its specifications. Different levels of testing were employed for software to make and error free, fault free, reliable. Basically in software testing three types of testing methods are adapted

- Unit Testing.
- Integration Testing.
- Validation Testing.

8.1 UNIT TESTING

In unit testing the analysis tests the programs making a system. Unit testing gives stress on the modules independently of one another, to find errors. This helps the tester in detecting errors in coding and logic that are within that module alone. The errors resulting from the interaction between modules are initially avoided. Unit testing can be performed from the bottom up, starting with smallest and lowest-level modules and proceeding one at a time. For each module in bottom-up testing a short program is used to execute the module and provides the needed data, so that the module is asked to perform the way it will when embedded with the larger system.

Unit testing deals with testing a unit testing a unit as a whole. This would test the interaction of many functions but confine the test within one unit. The exact scope of a unit is left to interpretation. Supporting test code, sometimes called scaffolding, may be necessary to support an individual test. This type of testing is driven by the architecture and implementation teams. This focus is also called back-box testing because only the details of the interface are visible to the test. Limits that are global to a unit are tested here.

Sometimes the scaffolding software becomes larger than the system software being tested. Usually the scaffolding software is not of the same quality as the system software and frequency is quite fragile. A small change in the change may larger changes in the scaffolding.

8.2 INTEGRATION TESTING

This testing level can be simply defined as integrating and then testing i.e., here, many unit tested modules are combined into subsystem, which are then tested. Integration testing aims at whether the modules can be integrated property.

Hence, the emphasis is on testing interfaces between modules. This testing activity can be considered testing the design.

8.3 VALIDATION TESTING

Validation testing is done to ensure complete assembly of the error free software. Validation can be termed successfully if it functions in manner that is reasonably expected by the customer.

9. SYSTEM IMPLEMENTATION

9.1 IMPLEMENTATION PROCEDURE

Implementation is the stage of the project when the theoretical design is turned into a working system. The implementation stage is a system project in its own right. It includes careful planning, investigation of current system and its constraints on implementation, design of methods to achieve the changeover, training of the staff in the changeover procedure and evaluation of changeover method.

The first task in implementation is planning-deciding on the methods and time-scale to be adopted. Once the planning has been completed, the major effort is to ensure that the programs in the system are working properly. At the same time concentrate on training the staff. When the staffs have been trained, the complete system, involving both computer and user can be executed effectively.

When the transport service system is linked to terminals on remote sites, the telecommunication network and tests of the network along with the system are also include under implementation. Depending upon the nature of the system, extensive user training may be required. Programming itself is a design work. The initial parameters of the management information system should be modified as a result of programming efforts; programming provides a reality test for the assumption made by the analyst.

System testing check the readiness and accuracy of the system access update and retrieve data from new files. Once the program becomes available, the test data are read into the computer and processed. In this system, Conventional Parallel Run was conducted to establish the efficiency of the system.

Implementation is used to mean the process of converting a new or a revised system design into an operational one. Conversion means changing from one system to another. The objective is to put the tested system into operation while holding costs, risks and personal irritation to a minimum

10. SOFTWARE MAINTENANCE

Maintenance is the enigma of system development. Maintenance covers a wide range of activities including correcting code and design errors, updating documentation and test data and upgrading users support.

Software maintenance means restoring something to its original condition. Maintenance can be classified as corrective, adaptive or perfective maintenance means repairing, processing or performance failures or making changes because of previously uncorrected problem or false assumption. Adaptive maintenance means changing the additional or changing needs.

Any system developer should be secured and protected against possible hazards. Security measures are provided to prevent unauthorized access of the database at various levels. An uninterrupted power supply should be that the power failure or voltage fluctuations will not erase the files.

Password protection and simple procedures to prevent the unauthorized access are provides to the users. The system allows the users to enter the system only through proper username and password.

11. CONCLUSION

The project entitled "**ONLINE FOOD ORDERING SYSTEM**" is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manager to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points:

- A description of the background and context of the project and its relation to work already done in the area.
- Made statement of the aims and objectives of the projects.
- The description of purpose and scope.
- We define the problem on which we are working in the project.
- We describe the requirement specifications of the system and the actions that can be done in these things.
- We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
- We included features and operations in details, including screen layouts.
- We designed user interface and security issues related to system.

12.1 APPENDIX –A (TABLES)

Table name: Register

Primary key: cust_id

Field	Type	Description
cust_id	int(10)	Customer id
cust_name	varchar(50)	Customer name
phone	varchar(20)	Phone number
email	varchar(50)	Email address

Table name: Login

Field	Type	Description
username	varchar(50)	Username
password	varchar(50)	password
type	varchar(10)	type
name	varchar(50)	Full name

Table name: Staff

Primary key: staff_id

Field	Type	Description
staff_id	int(10)	staff id
staff_name	varchar(50)	Staff name
phone	varchar(20)	Phone number
email	varchar(50)	Email address

Table name: Food

Primary key: food_id

Foreign key: staff_id

Field	Type	Description
food_id	int(10)	Item id
foodname	varchar(50)	Item name
category	varchar(50)	Food category
price	varchar(20)	Price
quantity	varchar(20)	Number of items
image	varchar(300)	Image
Staff_id	Int(10)	Staff id

Table name: Bank

Primary key: bank_id

Foreign key: cust_id

Field	Type	Description
bank_id	int(10)	Bank id
bankname	varchar(50)	Bank name
accountno	varchar(50)	Account number
ifsccode	varchar(10)	Pin code
branch	varchar(50)	Branch
balance	varchar(20)	Balance
cust_id	Int(10)	Customer id

Table name: Cart

Primary key: cart_id

Foreign key: food_id

Field	**Type**	**Description**
cart_id	int(10)	Cart id
food_id	int(10)	Item id
total_price	varchar(20)	Total price
date	date	Date
status	varchar(20)	Status
cust_id	Int(10)	Customer id

Table name: ShippingAddress

Primary key: ship_id

Foreign key: cust_id

Field	**Type**	**Description**
ship_id	int(10)	Shipping id
Name	varchar(50)	Full name
housename	varchar(50)	House name
street	varchar(50)	Street
city	varchar(50)	City
dist	varchar(50)	District
state	varchar(50)	State
pincode	varchar(20)	Pin code
cust_id	Int(10)	Customer id

Table name: UserOrder

Primary key: order_id

Foreign key: ship_id

Field	Type	Description
order_id	int(10)	Order number
date	date	Date
amount	varchar(10)	Amount
status	varchar(20)	Status
ship_id	int(10)	Shipping id

Table name: RegisterRestaurant

Primary key: rest_id

Field	Type	Description
rest_id	int(10)	Restaurant id
rest_name	varchar(50)	Restaurant name
address	varchar(100)	Address

Table name: FoodRestaurant

Primary key: foodrest_id

Foreign key: rest_id

Field	Type	Description
foodrest_id	int(10)	Food restaurant id
foodname	varchar(50)	Item name
restaurant	varchar(50)	restaurant
category	varchar(50)	Food category
price	varchar(20)	Price

quantity	varchar(20)	Number of items
image	varchar(300)	Image
rest_id	int(10)	Restaurant id

Table name: OrderRestaurant

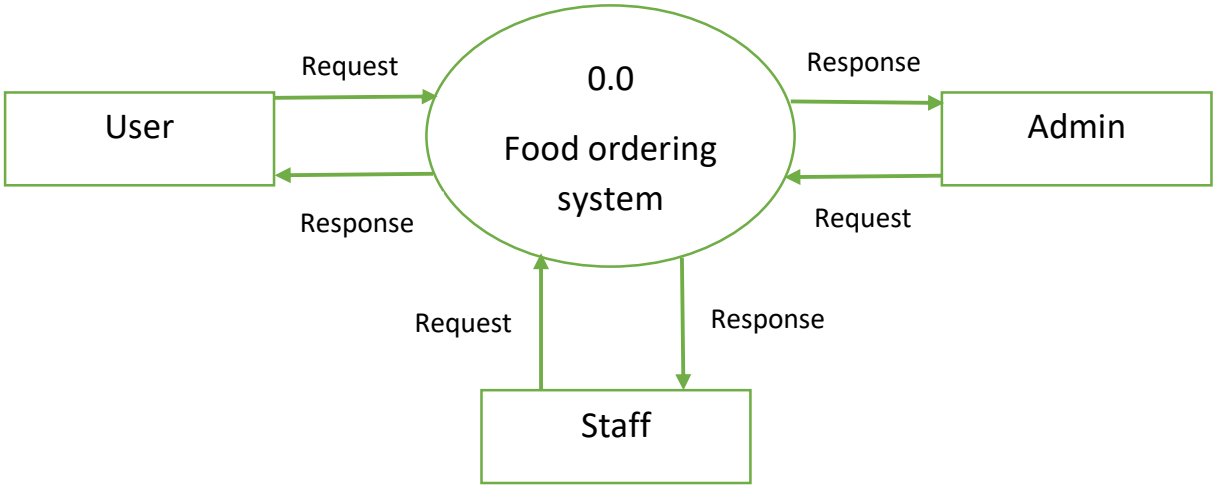
Primary key: restorder_id

Foreign key: ship_id

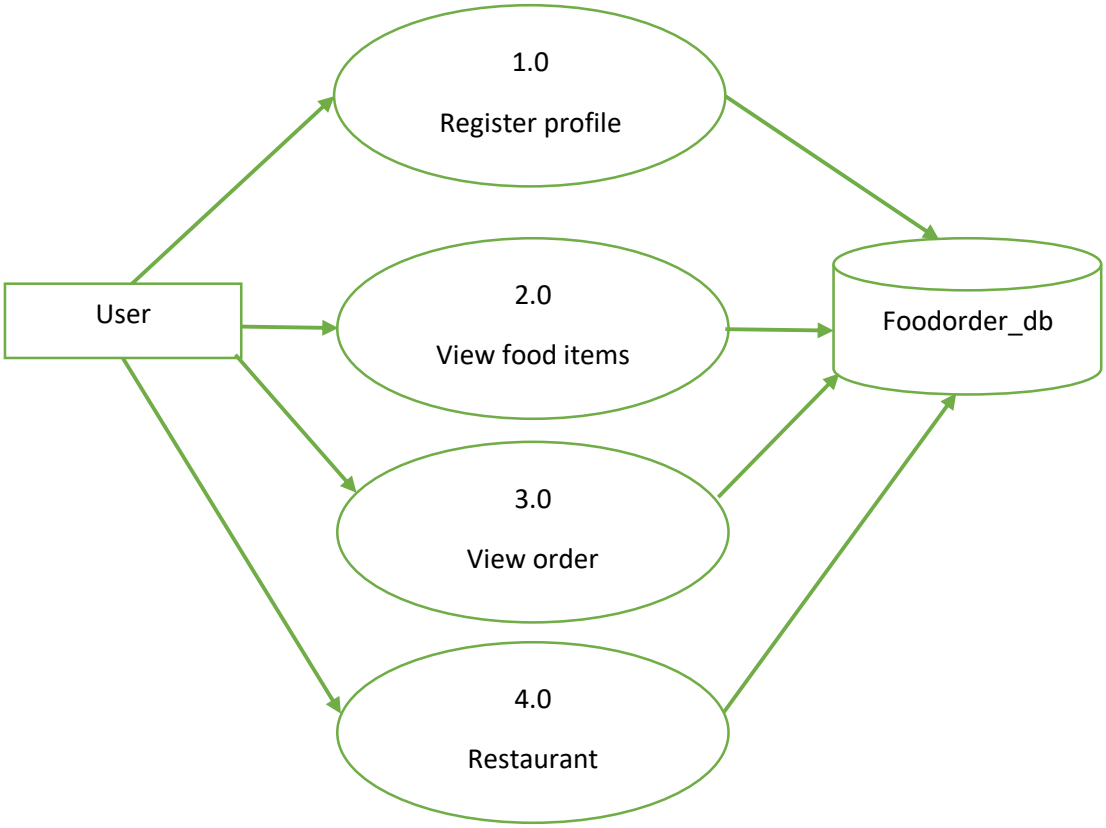
Field	**Type**	**Description**
restorder_id	int(10)	Order restaurant id
date	date	Date
amount	varchar(10)	Amount
status	varchar(20)	Status
ship_id	int(10)	Shipping id
rest_id	int(10)	Restaurant id

12.2 APPENDICES-B (DFD)

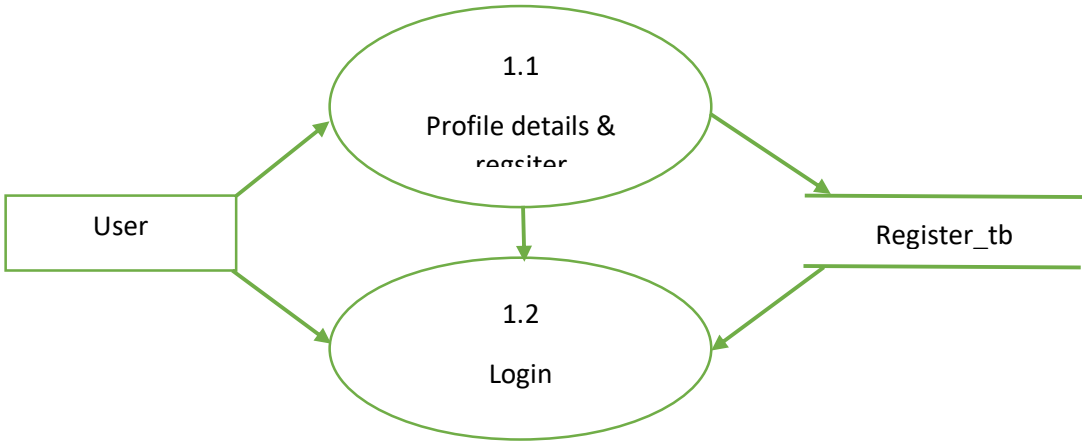
CONTEXT LEVEL DFD FOR FOOD ORDERING SYSTEM



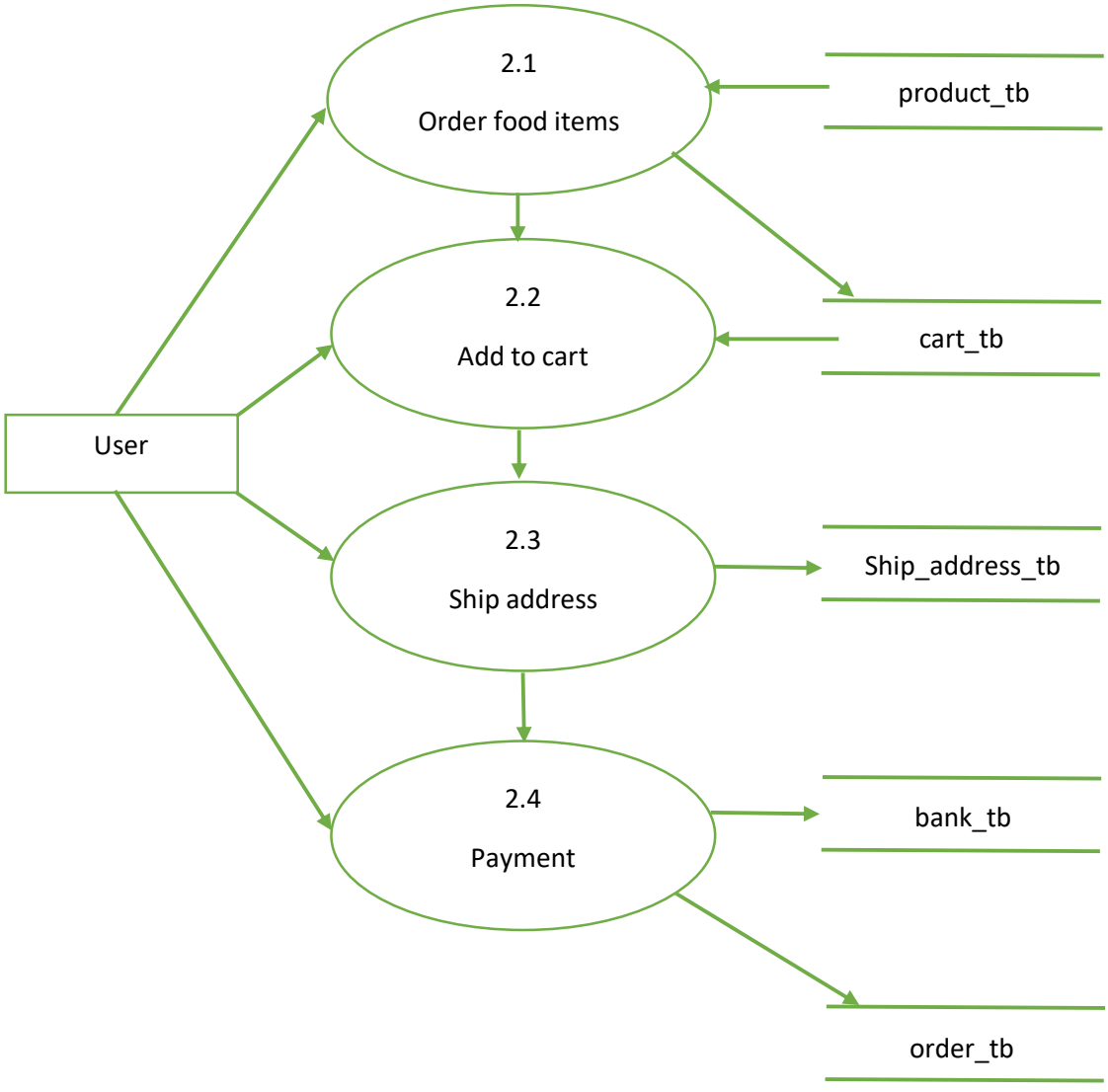
LEVEL 1 DFD FOR USER



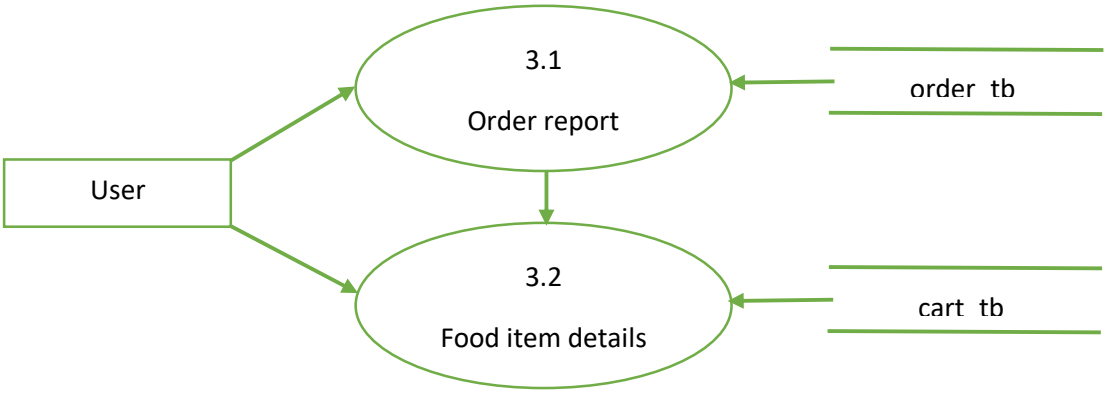
LEVEL 1 (REGISTER) USER



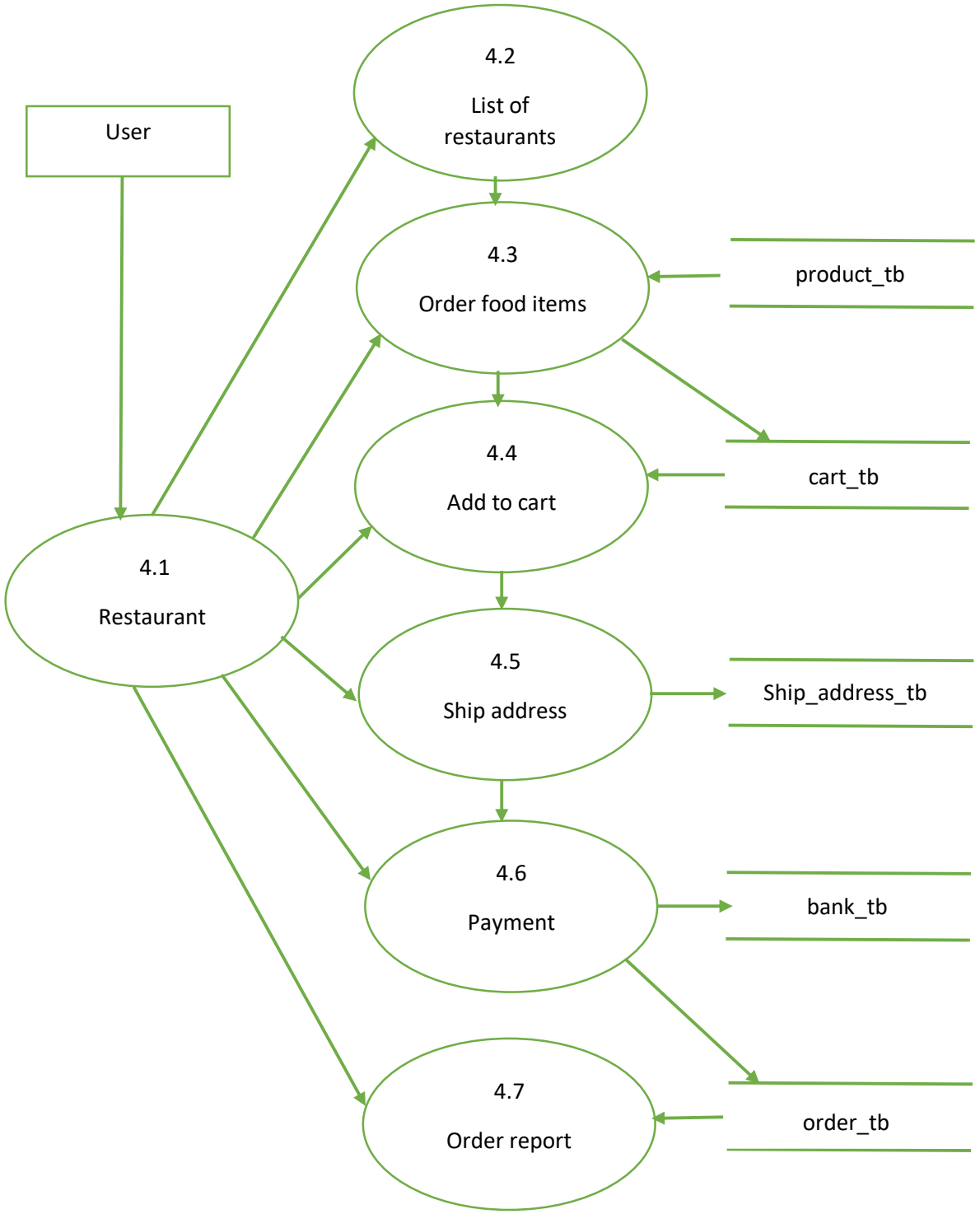
LEVEL 1 (VIEW FOOD ITEMS) USER



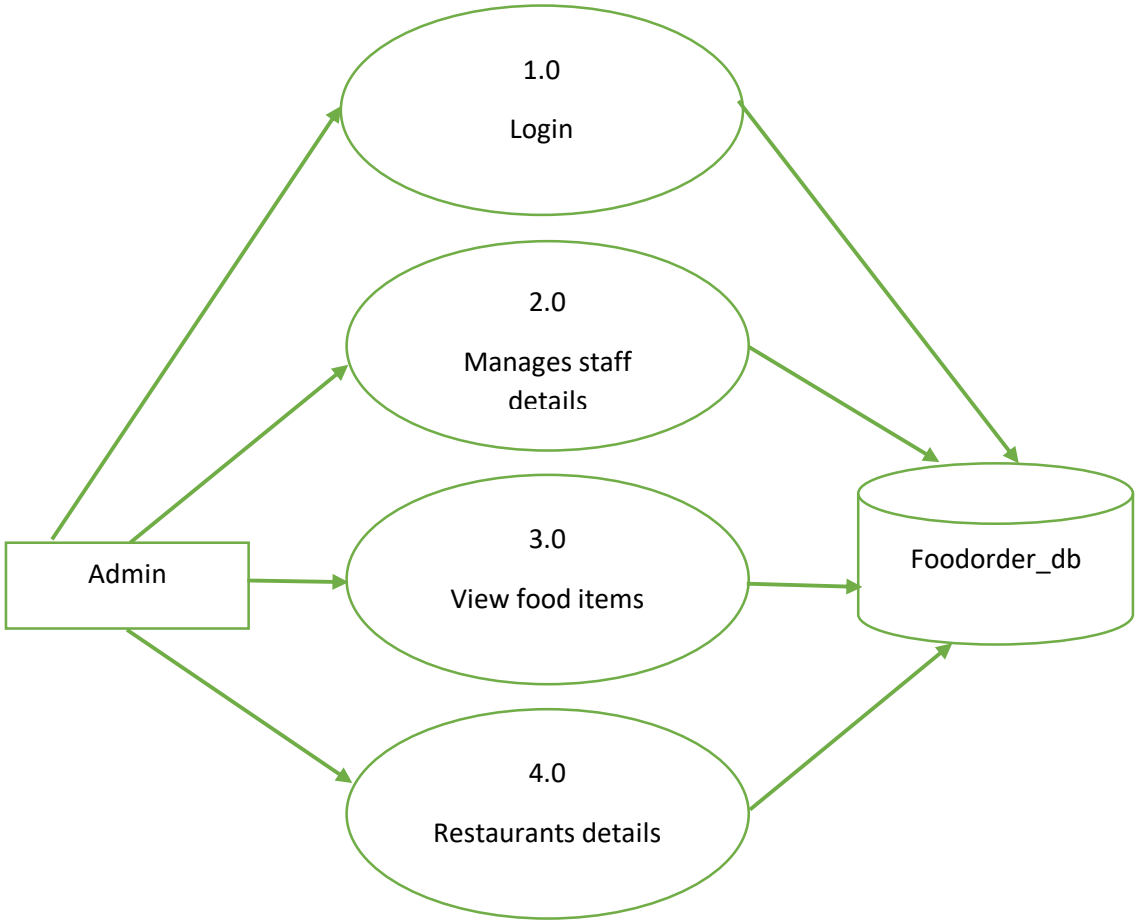
LEVEL 1 (VIEW ORDER) USER



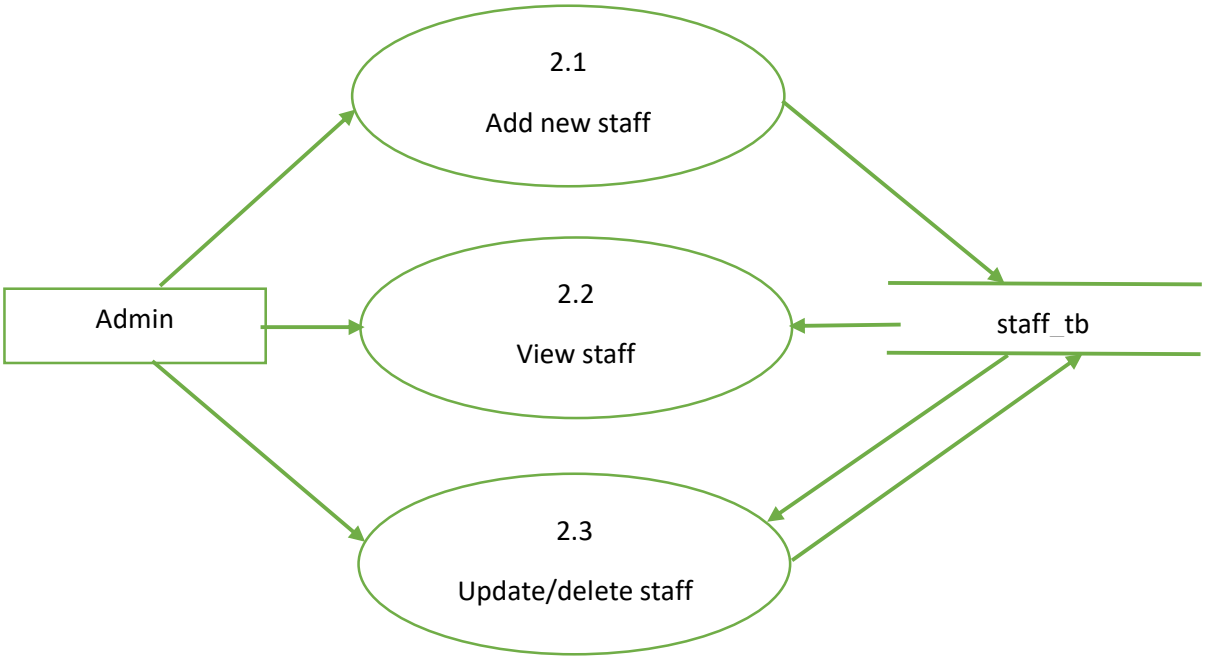
LEVEL 1 (RESTAURANT) USER



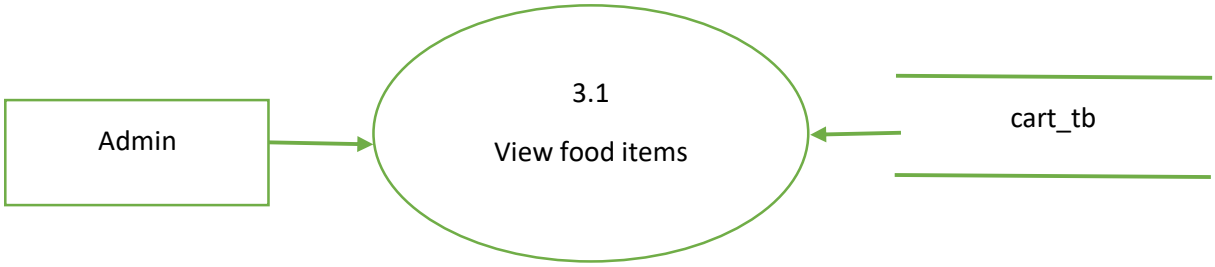
LEVEL 1 DFD ADMIN



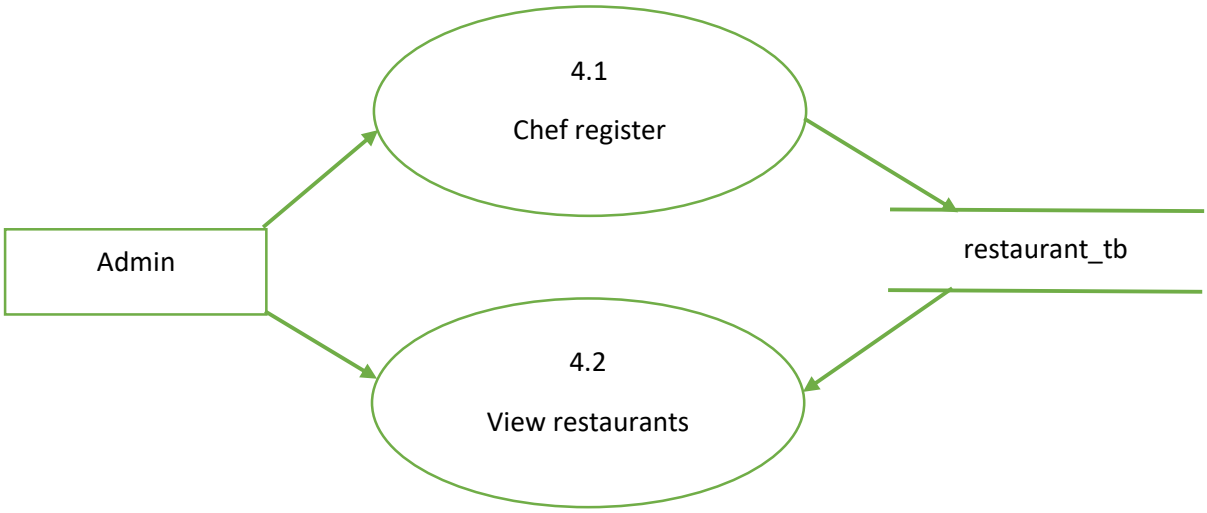
LEVEL 1 (MANAGE STAFF DETAILS) ADMIN



LEVEL 1 (VIEW FOOD ITEMS) ADMIN

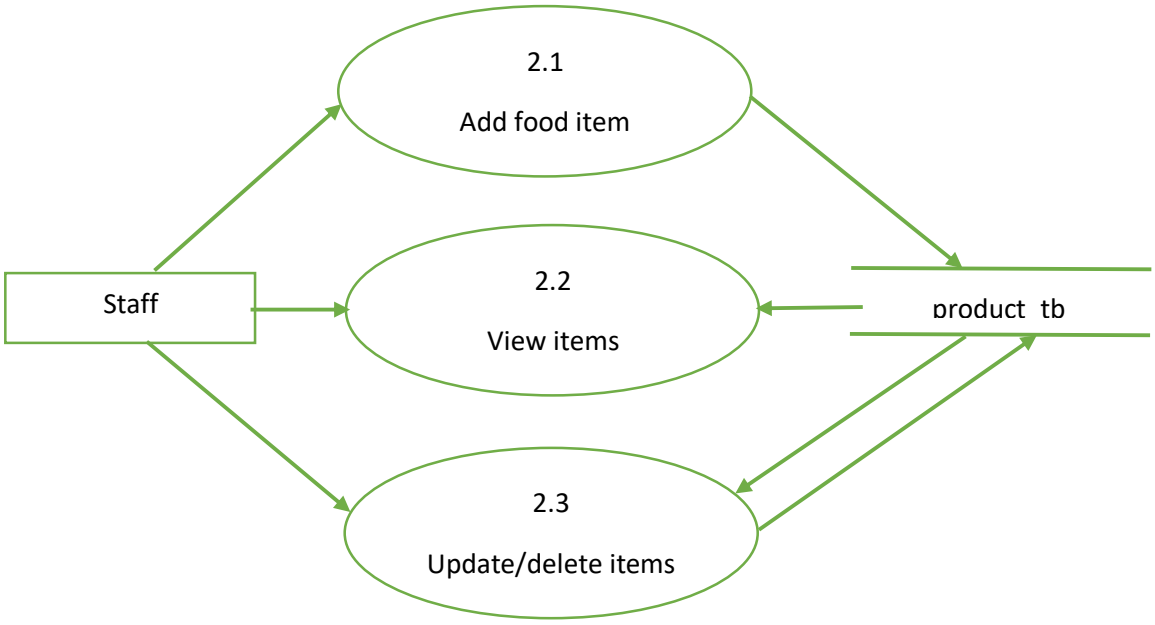


LEVEL 1 (RESTAURANT DETAILS) ADMIN

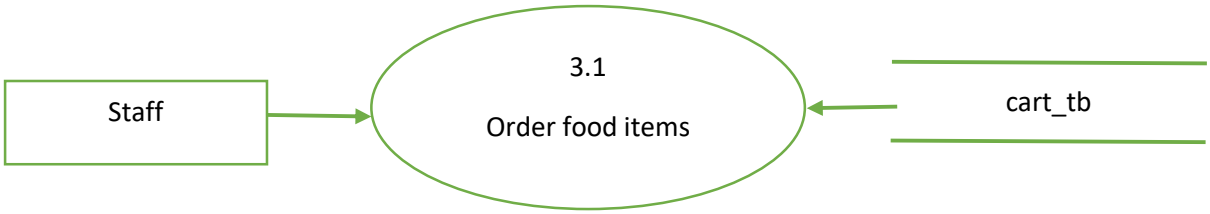


LEVEL 1 DFD FOR STAFF

LEVEL 1 (MANAGE FOOD DETAILS) STAFF



LEVEL 1 (VIEW ORDER) STAFF

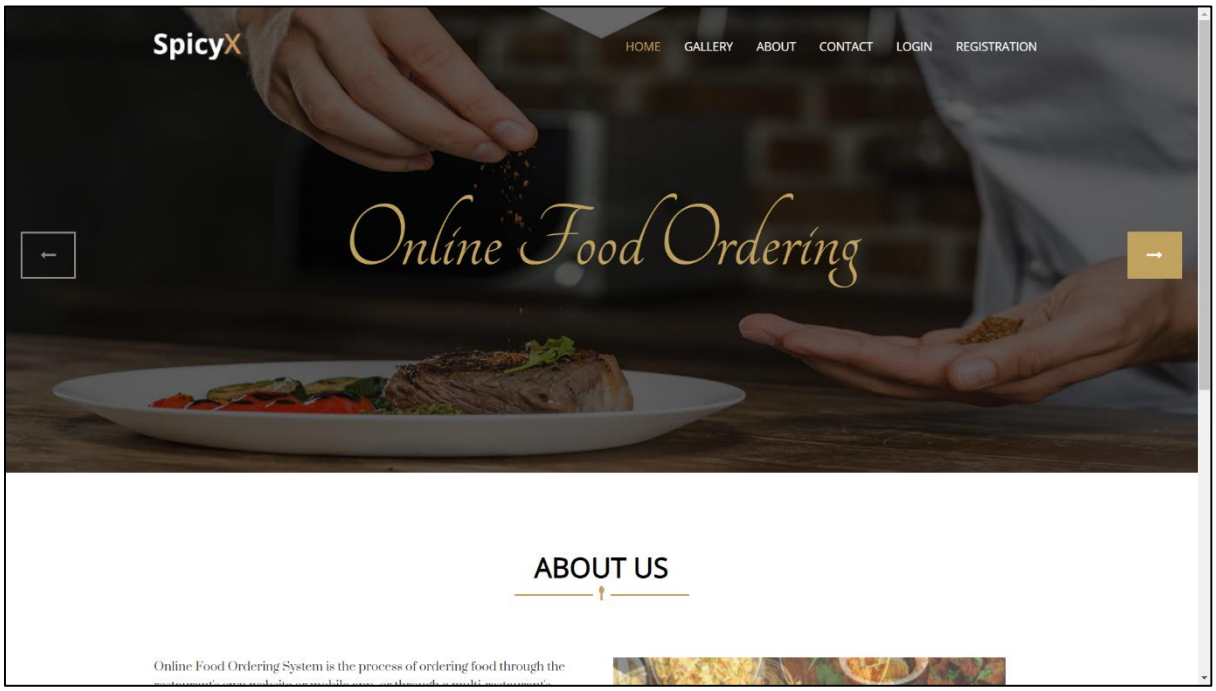


LEVEL 1 (SHIPPING DETAILS) STAFF

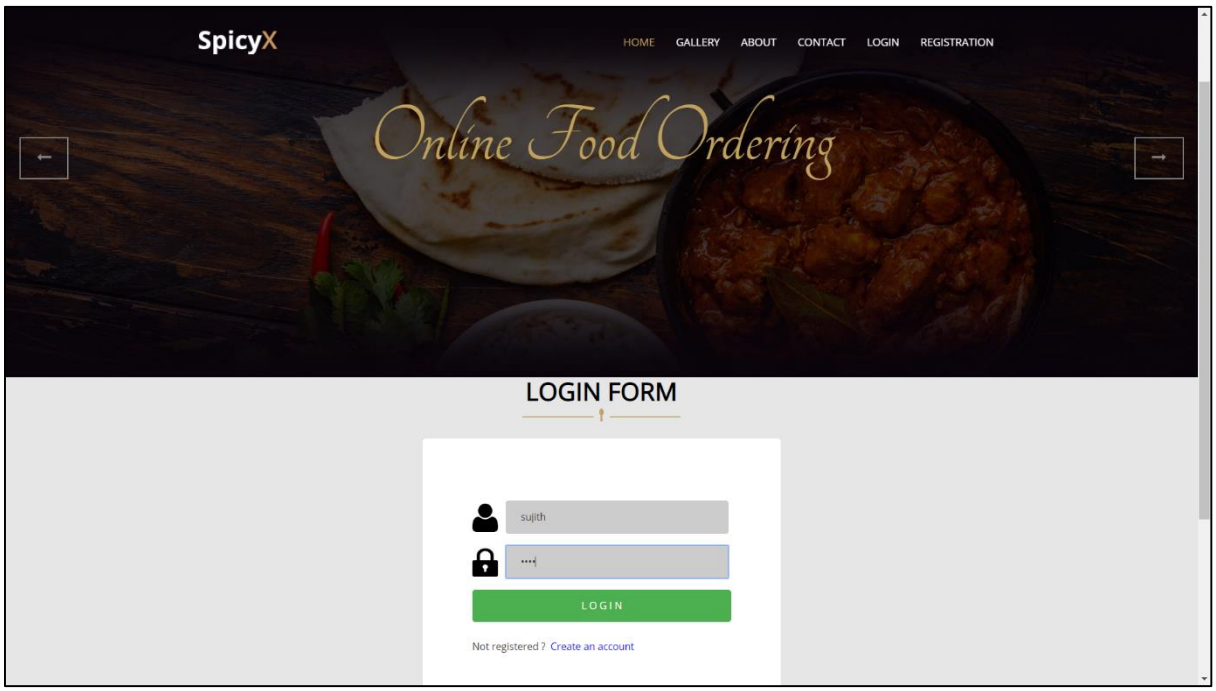
LEVEL 1 (USER REGISTERED) STAFF

12.3 APPENDICES-C (INPUT FORMS AND OUTPUT FORMS)

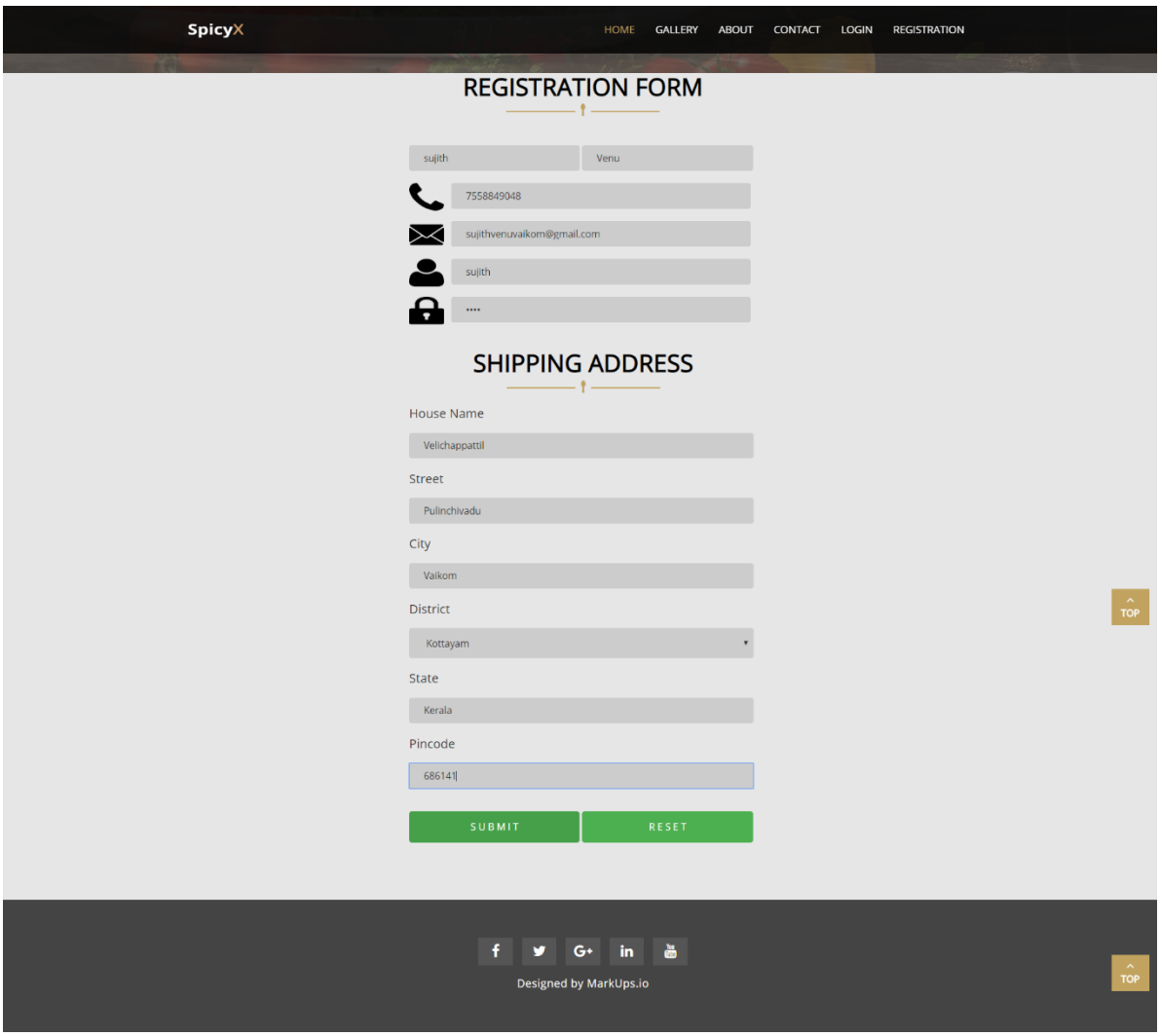
HOME PAGE



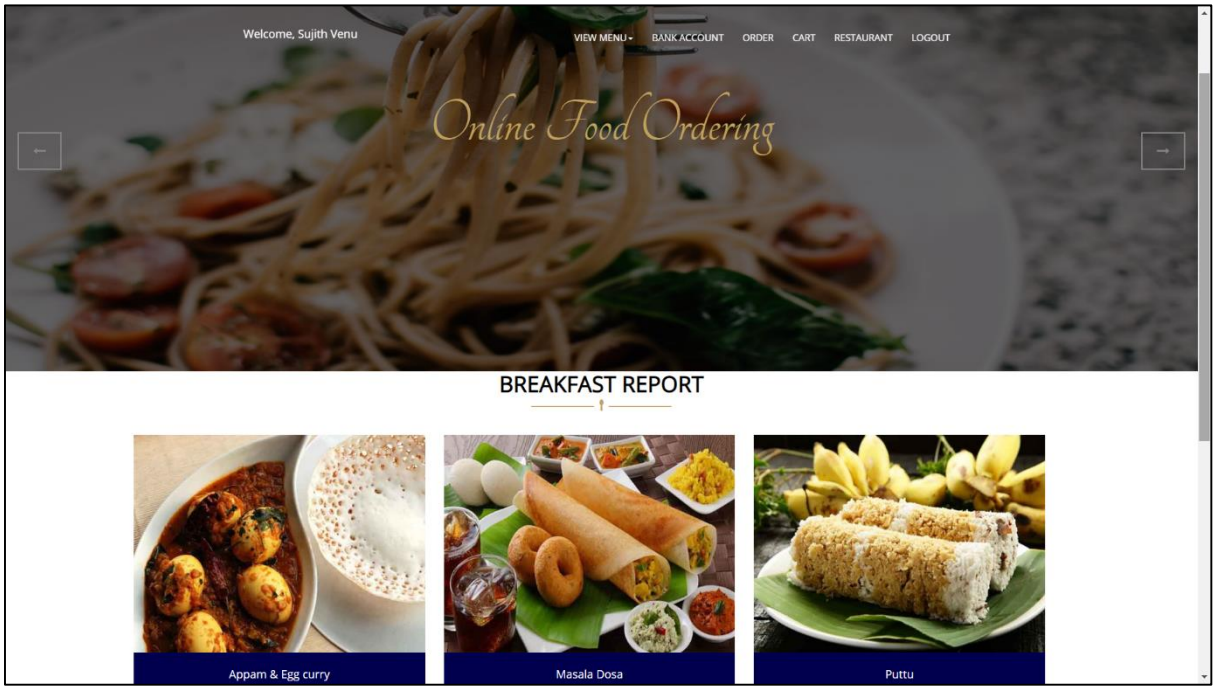
LOGIN

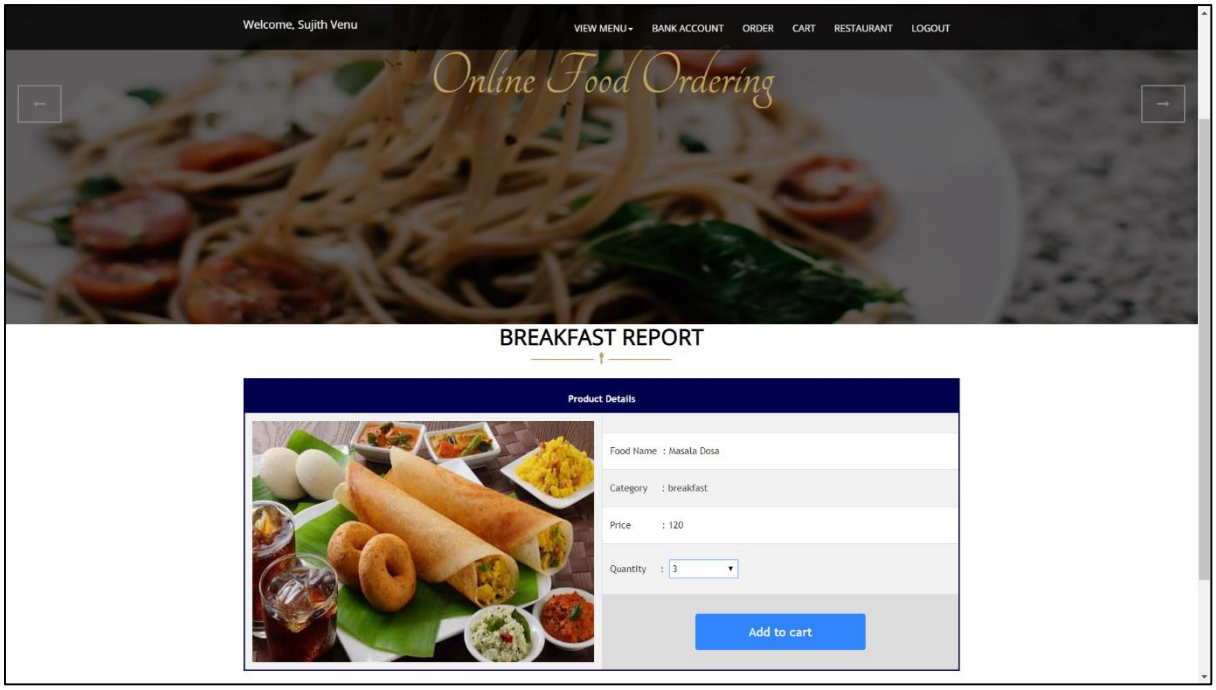


REGISTER

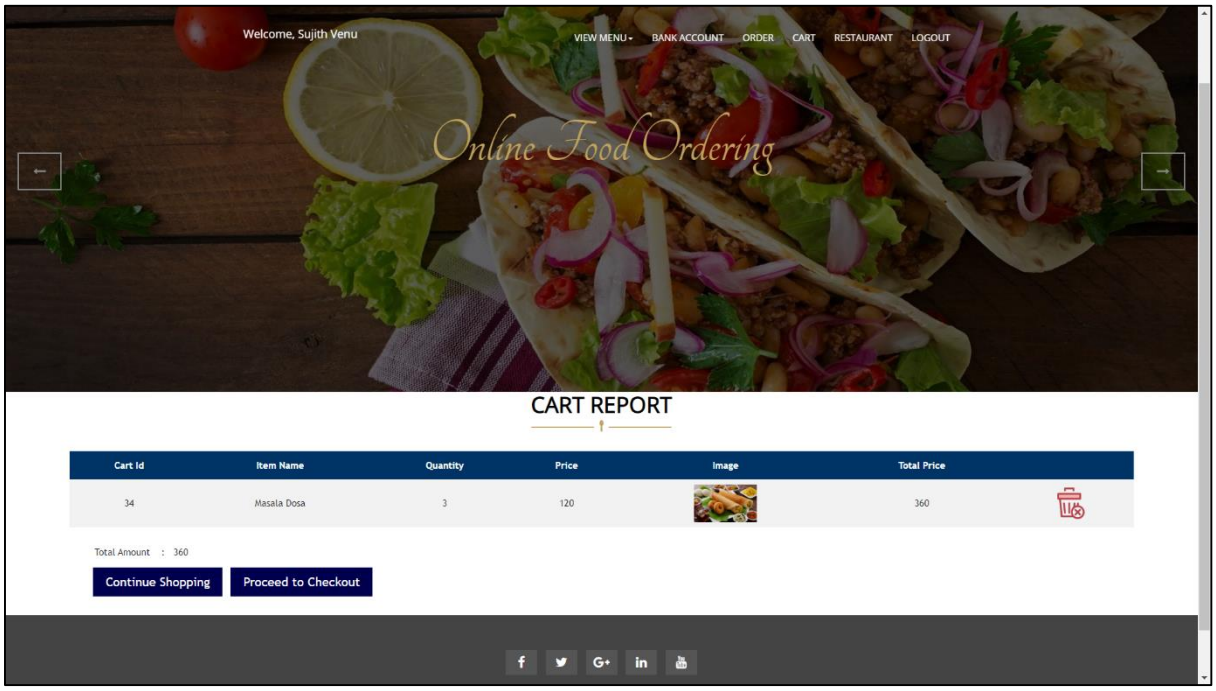

SpicyX
HOME
GALLERY
ABOUT
CONTACT
LOGIN
REGISTRATION
REGISTRATION FORM
sujith
Venu
7558849048
sujithvenuvaikom@gmail.com
sujith
....
SHIPPING ADDRESS
House Name
Velichappattil
Street
Pulinchivadu
City
Vaikom
District
Kottayam
State
Kerala
Pincode
686141
SUBMIT
RESET
TOP
Designed by MarkUps.io
TOP

USER - FOOD DETAILS

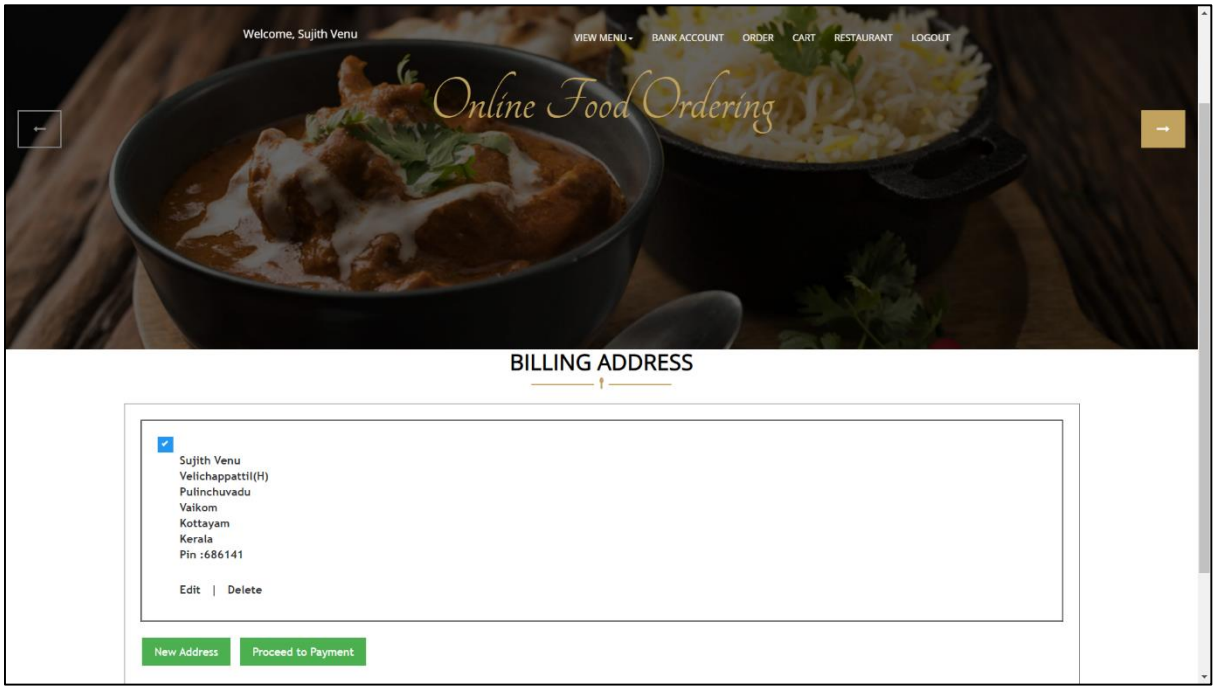




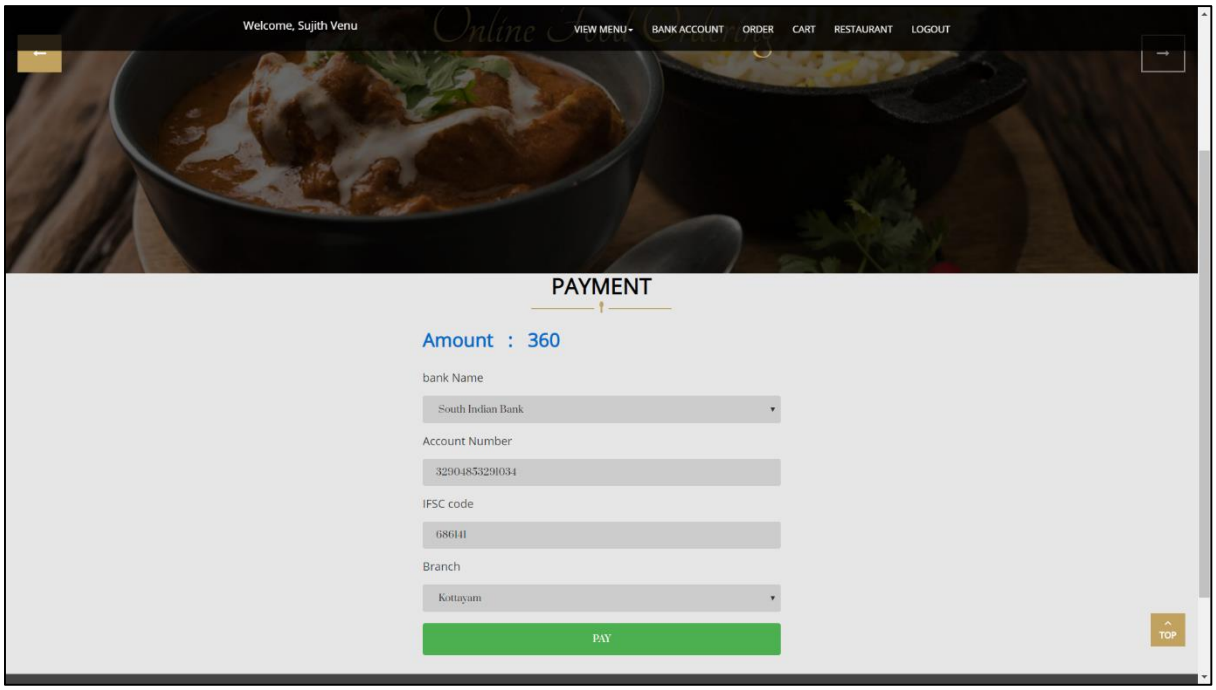
USER - CART



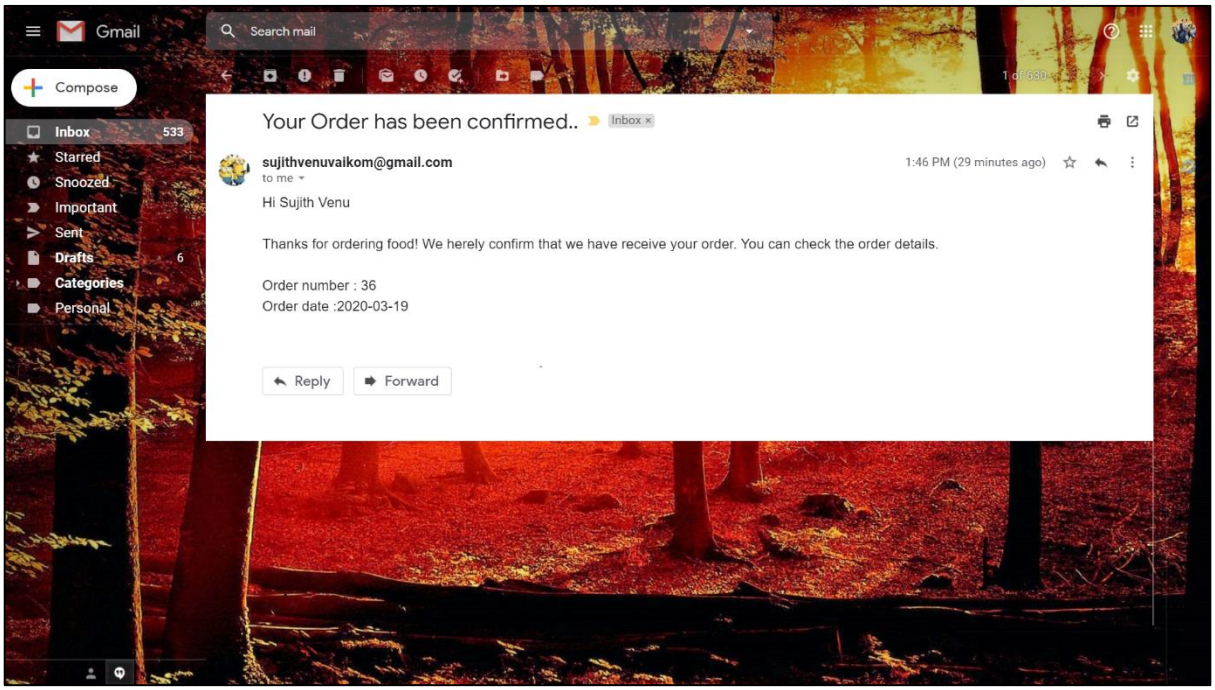
USER - SHIPPING ADDRESS



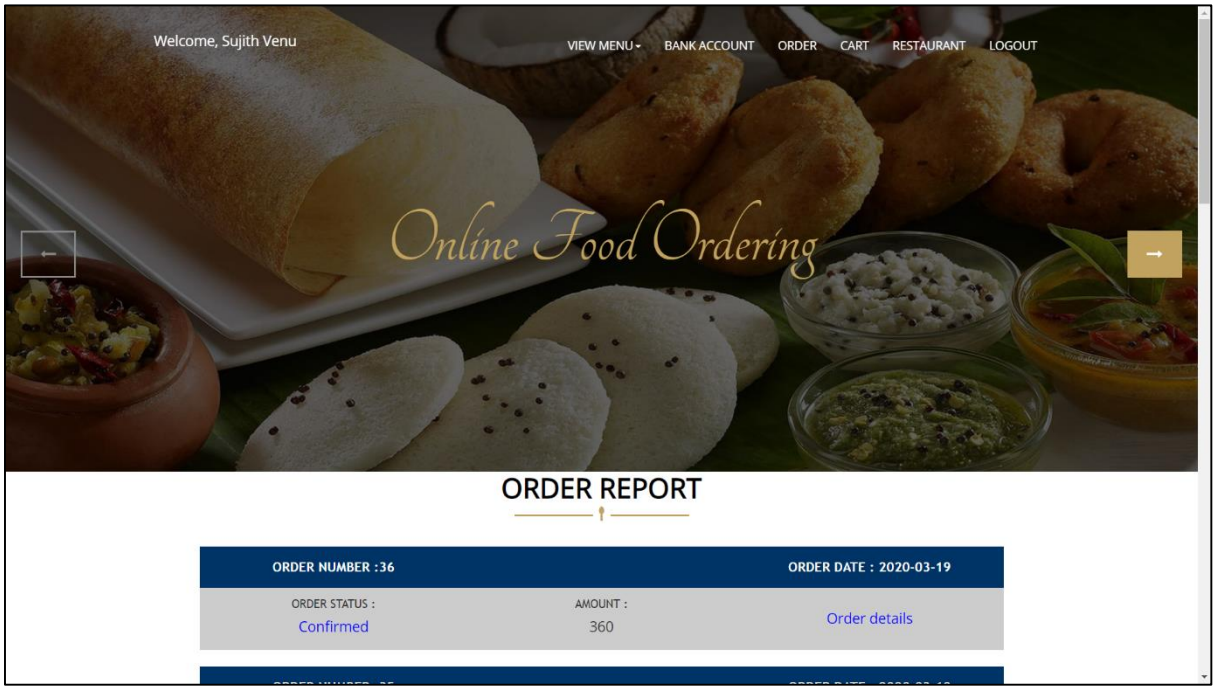
USER - PAYMENT



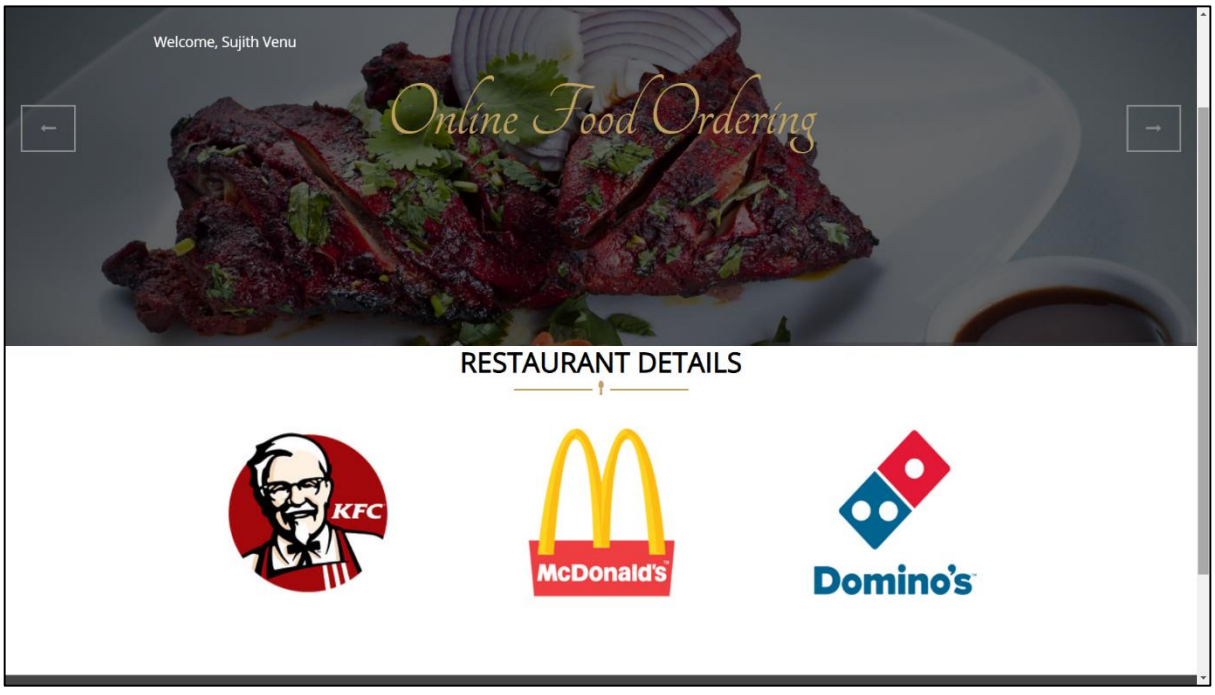
USER – EMAIL DETAILS



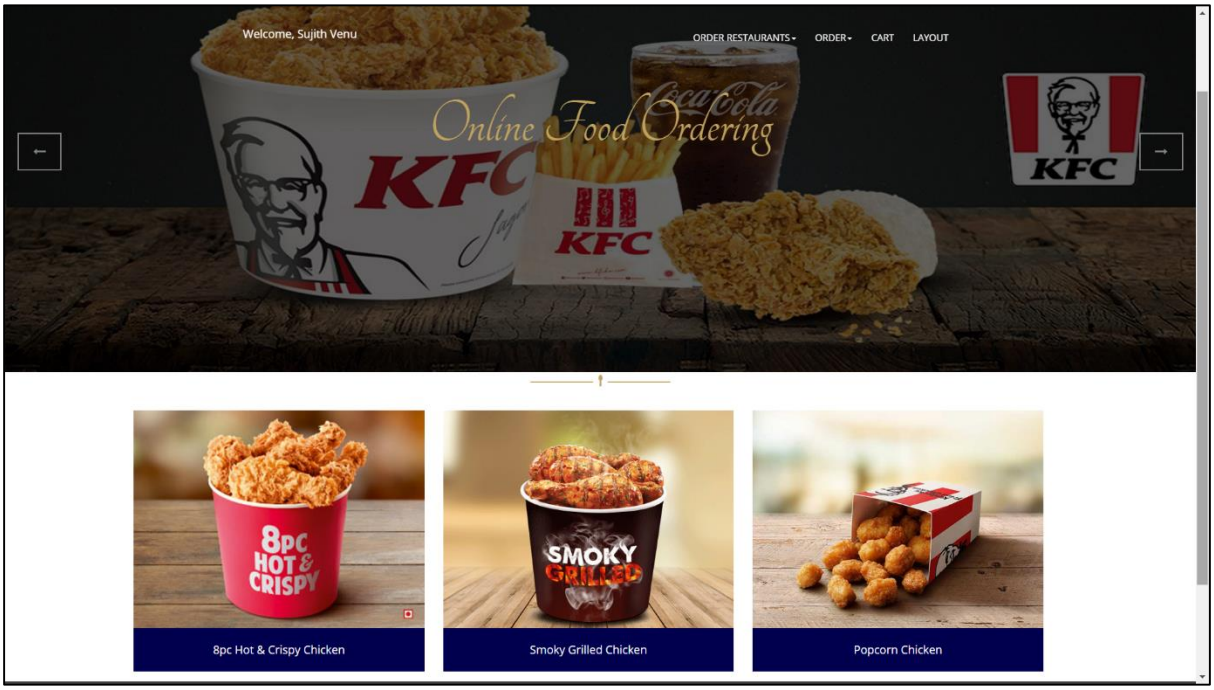
USER – ORDER REPORT



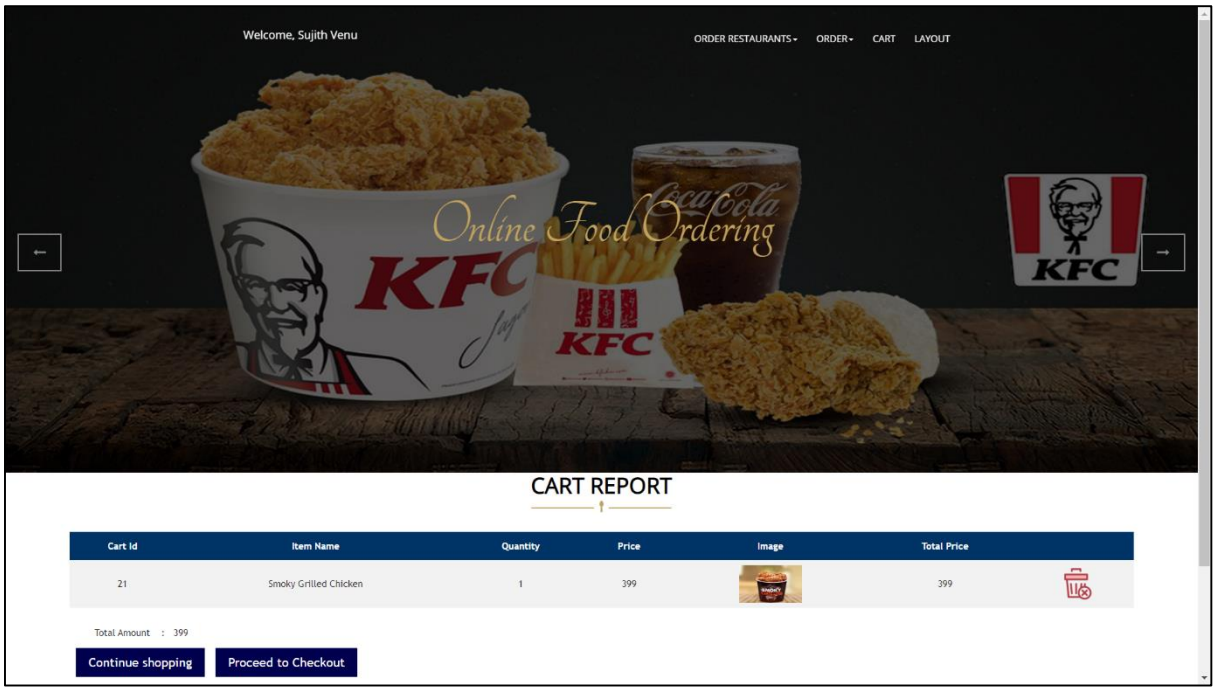
USER - RESTAURANT DETAILS



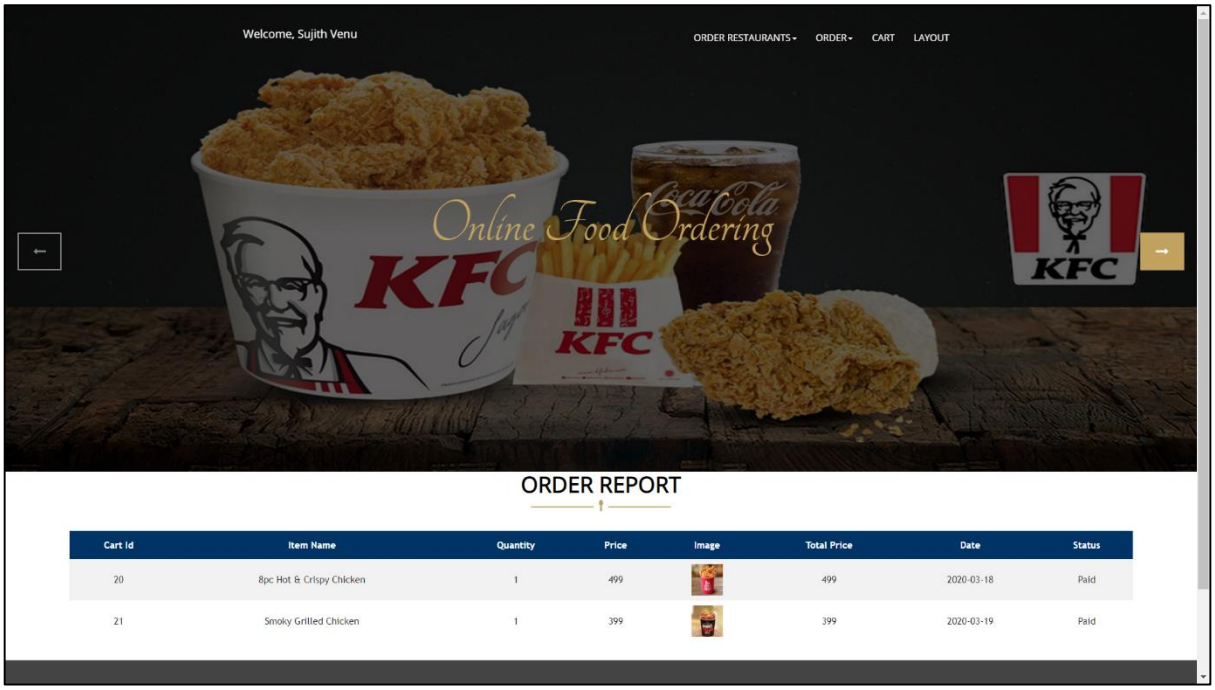
USER – RESTAURANT FOOD DETAILS



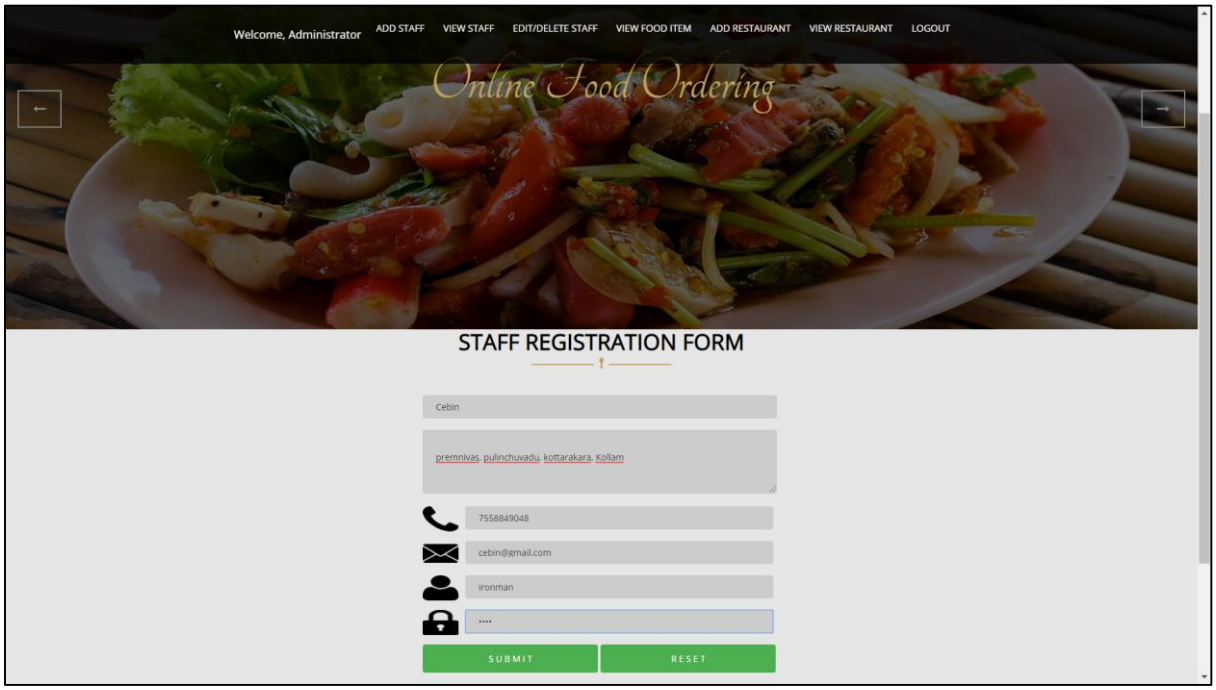
USER – RESTAURANT CART



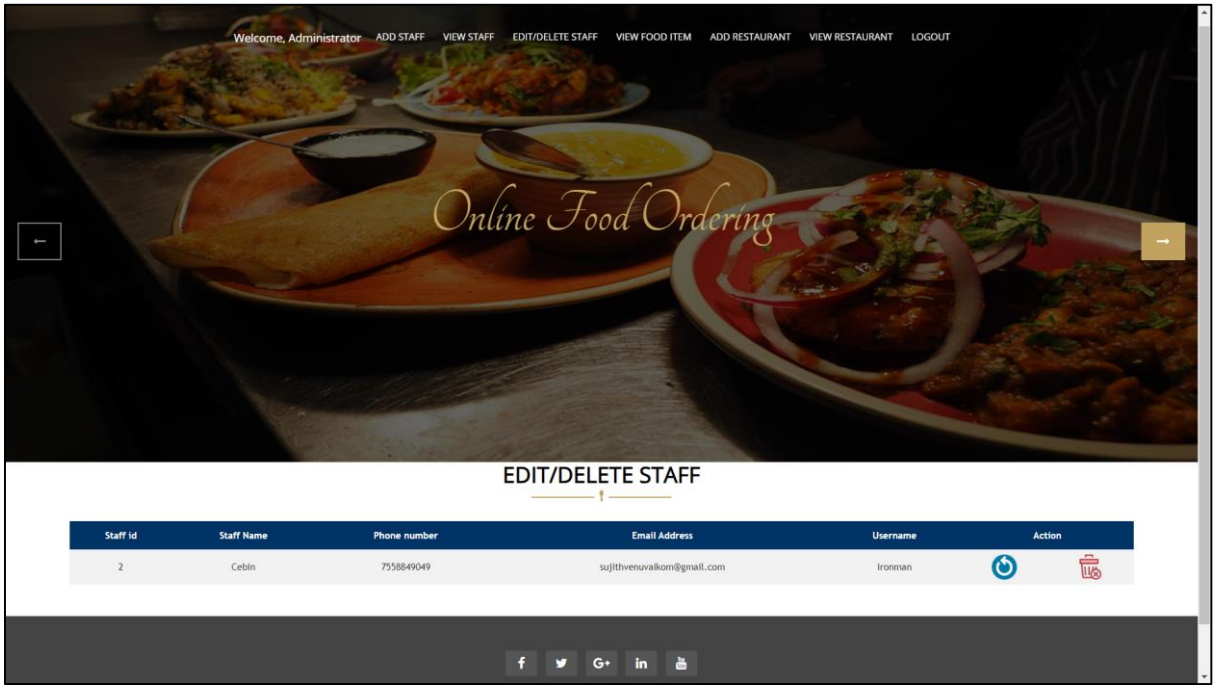
USER – RESTAURANT ORDER



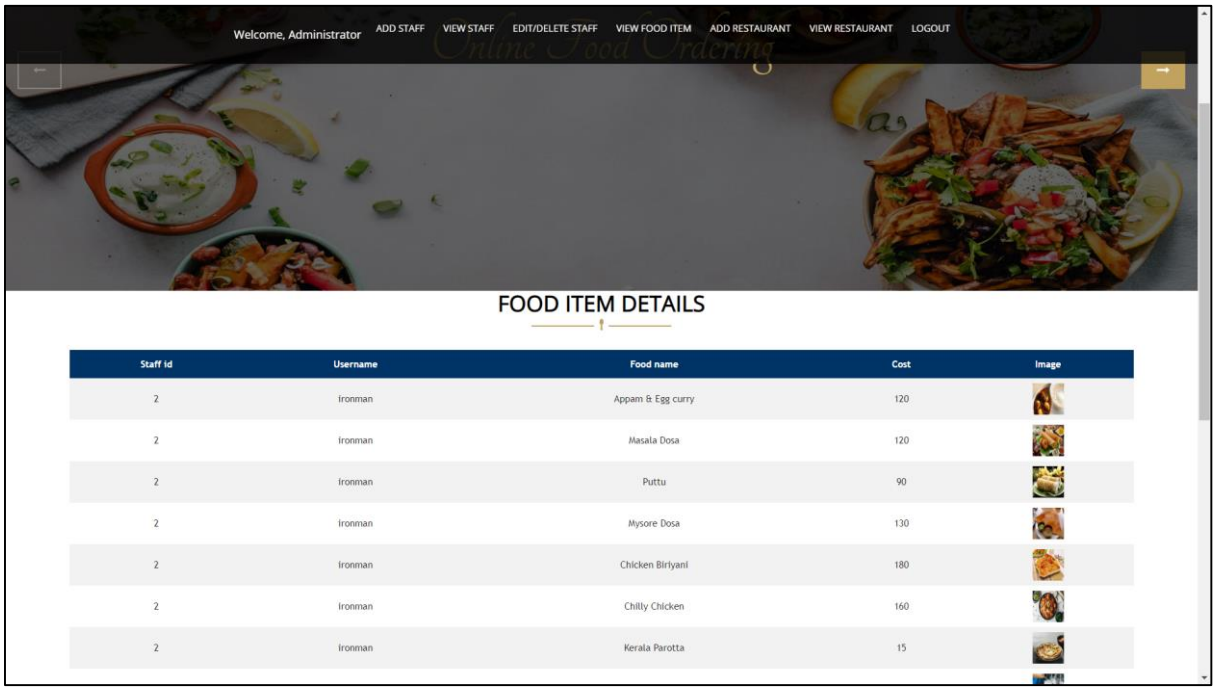
ADMIN – STAFF REGISTER



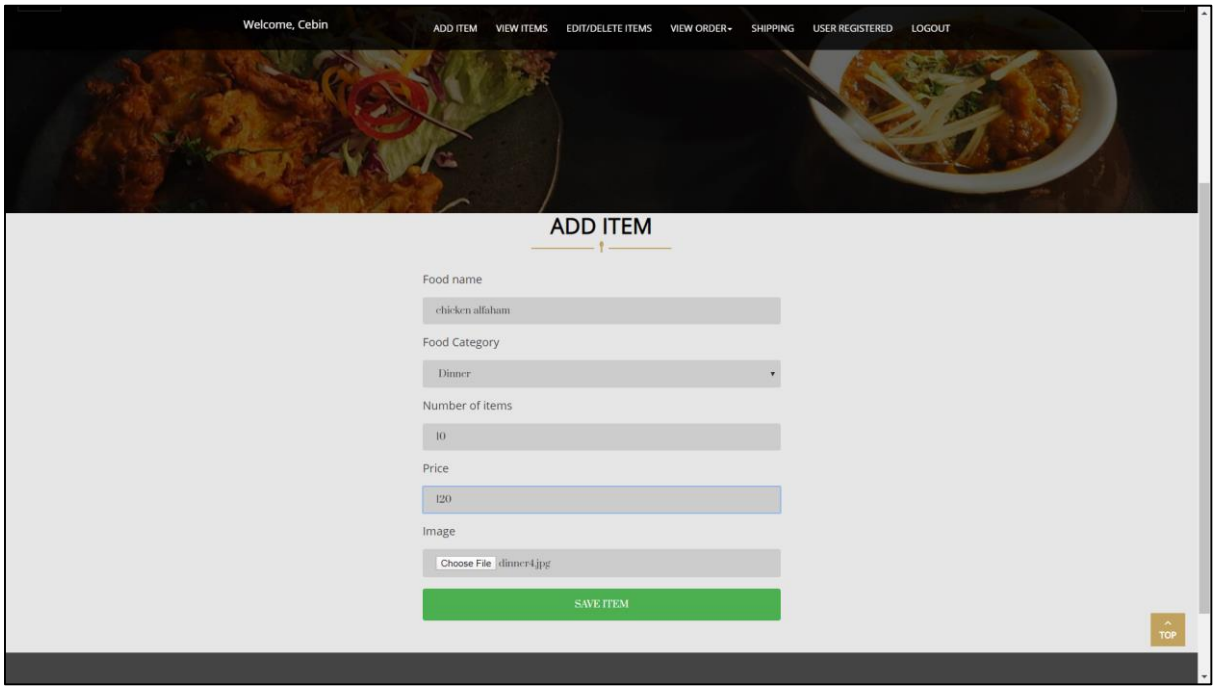
ADMIN – STAFF DETAILS



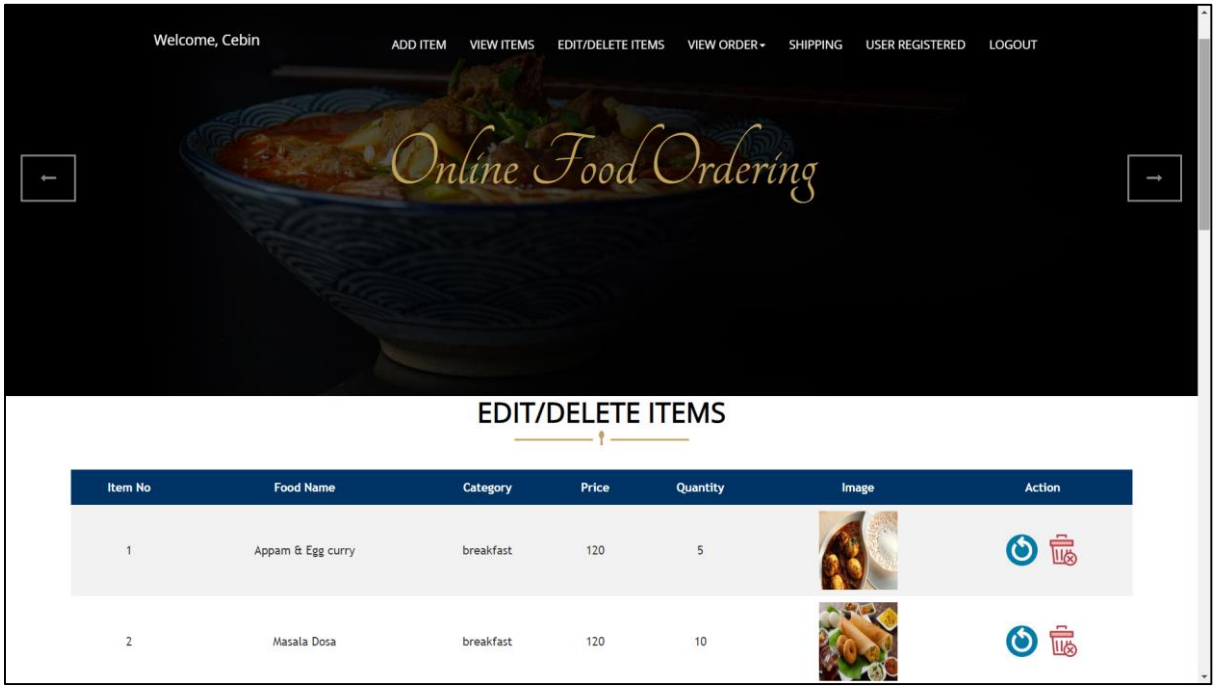
ADMIN – FOOD DETAILS



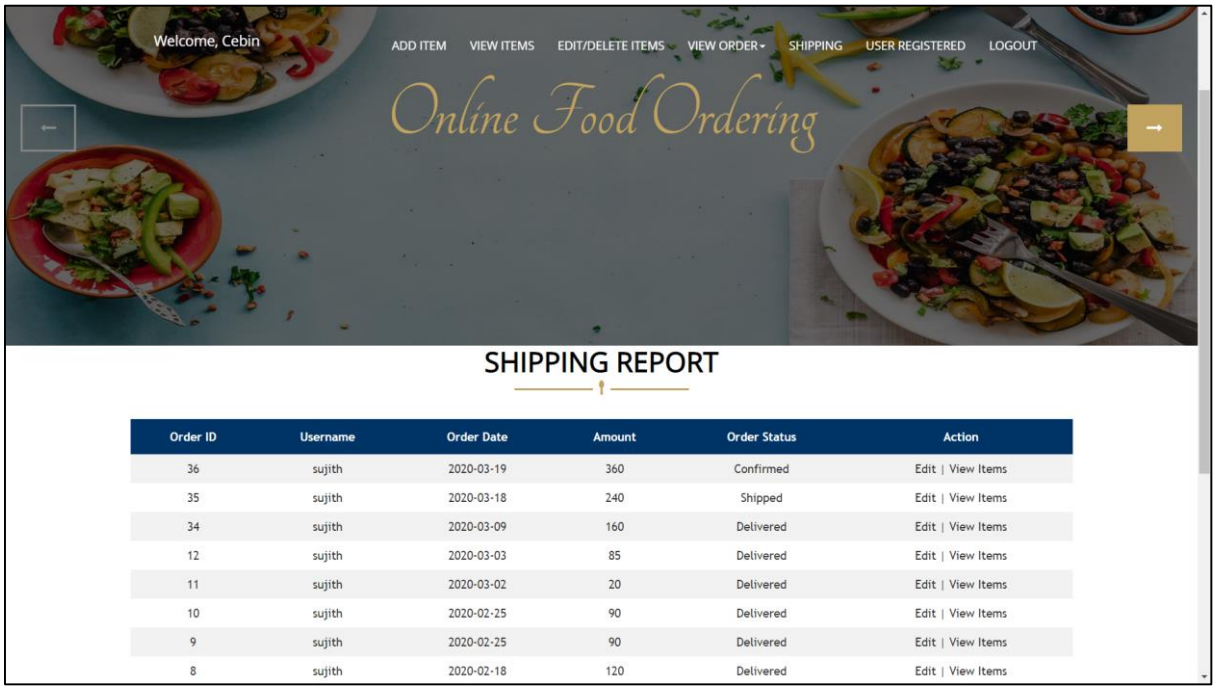
STAFF – ADD FOOD ITEM



STAFF – FOOD ITEM DETAILS

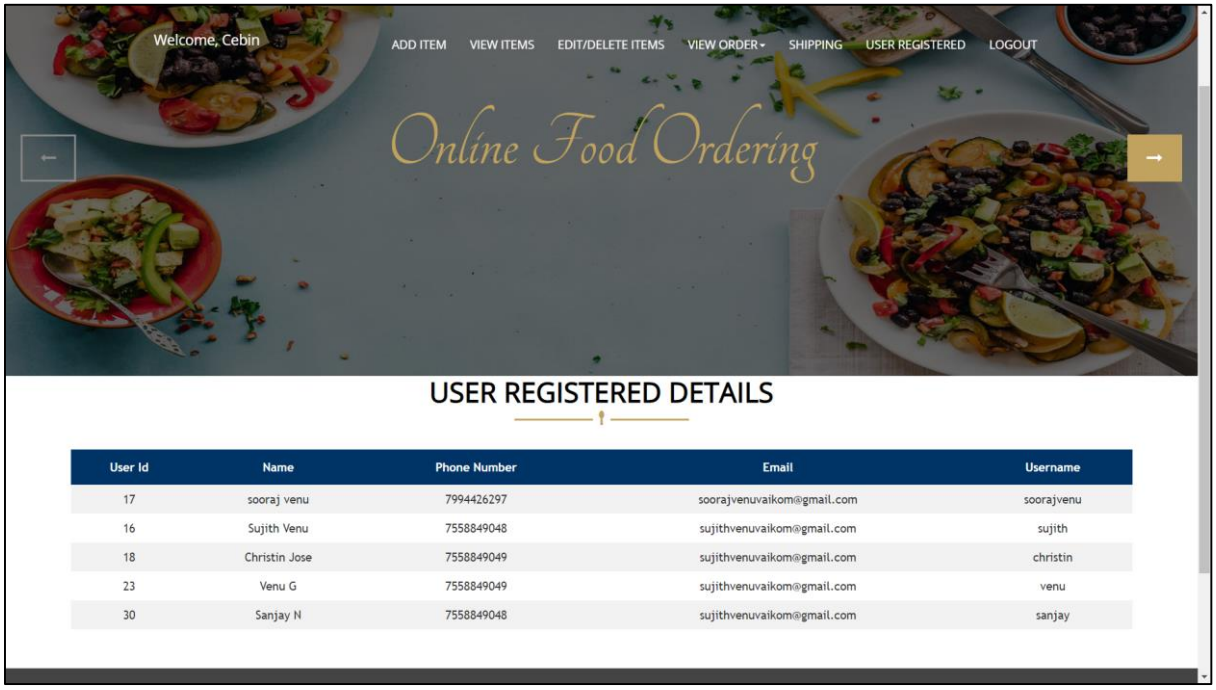


STAFF – ORDER DETAILS



Order ID	Username	Order Date	Amount	Order Status	Action
36	sujith	2020-03-19	360	Confirmed	Edit | View Items
35	sujith	2020-03-18	240	Shipped	Edit | View Items
34	sujith	2020-03-09	160	Delivered	Edit | View Items
12	sujith	2020-03-03	85	Delivered	Edit | View Items
11	sujith	2020-03-02	20	Delivered	Edit | View Items
10	sujith	2020-02-25	90	Delivered	Edit | View Items
9	sujith	2020-02-25	90	Delivered	Edit | View Items
8	sujith	2020-02-18	120	Delivered	Edit | View Items

STAFF – USER REGISTERED



User Id	Name	Phone Number	Email	Username
17	sooraj venu	7994426297	soorajvenuvaikom@gmail.com	soorajvenu
16	Sujith Venu	7558849048	sujithvenuvaikom@gmail.com	sujith
18	Christin Jose	7558849049	sujithvenuvaikom@gmail.com	christin
23	Venu G	7558849049	sujithvenuvaikom@gmail.com	venu
30	Sanjay N	7558849048	sujithvenuvaikom@gmail.com	sanjay

12.4 APPENDICES-D (CODE)

REGISTER

```
<%@include file="header.jsp" %>
<div class="mu-title">
   <h2>REGISTRATION FORM</h2>
</div>
<html>
   <head>
      <link her="assets/css/account.css" rel="stylesheet" type="text/css"/>
   </head>
<body>
   <table width="100%">
   <tr>
      <td width="35%"></td>
      <td width="30%">
<div class="rectangle">
     <h5>
      <form action="register_action.jsp" method="post">
         <input type="textname" name="fname" placeholder="F I R S T N A M E" required>
         <input type="textname" name="lname" placeholder="L A S T N A M E" required>
         <img src="assets/img/phone.png" width="10%" height="40px">  
         <input type="text" name="phone" placeholder="P H O N E  N U M B E R" pattern="[789][0-9]{9}" required>
         <img src="assets/img/envelope.png" width="10%" height="40px">  
         <input type="text" name="mail" placeholder="E M A I L  A D D R E S S" pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$" required>
         <img src="assets/img/user.png" width="10%" height="40px">  
         <input type="text" name="uname" placeholder="U S E R N A M E" required>
         <img src="assets/img/pwd.png" width="10%" height="40px">  
```

```
        <input type="password" name="pwd" placeholder="P A S S W O R D" required>
        <br><br><br>
<div class="mu-title">
<h2>SHIPPING ADDRESS</h2>
<i class="fa fa-spoon"></i>
<span class="mu-title-bar"></span>
</div>
        <h4>House Name</h4>
        <input type="text1" name="hname" placeholder="Enter your house name" required>
        <h4>Street</h4>
        <input type="text1" name="street" placeholder="Enter your street" required>
        <h4>City</h4>
        <input type="text1" name="city" placeholder="Enter your city" required>
        <h4>District</h4>
        <select name="dist" required>
                <option value="0">---</option>
                <option value="Thiruvananthapuram">Thiruvananthapuram</option>
                <option value="Kollam">Kollam</option>
                <option value="Pathanamthitta">Pathanamthitta</option>
                <option value="Alappuzha">Alappuzha</option>
                <option value="Kottayam">Kottayam</option>
                <option value="Idukki">Idukki</option>
                <option value="Ernakulam">Ernakulam</option>
                <option value="Thrissur">Thrissur</option>
                <option value="Palakkad">Palakkad</option>
                <option value="Malappuram">Malappuram</option>
                <option value="Kozhikode">Kozhikode</option>
                <option value="Wayand">Wayand</option>
                <option value="Kannur">Kannur</option>
                <option value="Kasargod">Kasargod</option>
        </select><br>
```

```
        <h4>State</h4>
        <input type="text1" name="state" placeholder="Enter your state" required>
        <h4>Pincode</h4>
        <input type="text1" name="pin" placeholder="Enter your pincode" required>
        <Br><br>
        <input type="submit" value="S U B M I T" onclick="register()"> <input type="reset" value="R E S E T">
      </form>
    </h5>
</div>
      </td>
      <td width="35%"></td>
  </table>
  <br><br><br>
</body>
</html>
<%@include file="footer.jsp" %>
```

LOGIN

```
<%@include file="header.jsp" %>
<div class="mu-title">
          <h2>LOGIN FORM</h2>
</div>
<html>
  <head>
    <link rel="stylesheet" type="text/css"  href="assets/css/text.css">
  </head>
<body>
<form action="login_action.jsp" method="post">
  <table width="100%">
```

```
   <tr>
      <td width="35%"></td>
      <td width="30%">
    <!--   login section  -->
 <div class="rectangle">
     <h5>
        <img src="assets/img/user.png" width="10%" height="40px">  
        <input type="text" id="fname" name="log_user" placeholder="U S E R N A M
E" required>
        <img src="assets/img/pwd.png" width="10%" height="40px">  
        <input type="password" id="lname" name="log_pwd" placeholder="P A S S W
O R D" required>
        <input type="submit" value="L O G I N" onclick="login()">
     </h5>
     <label><h5>Not registered ?<a href="registration.jsp" style="color:
blue">  Create an account</a></h5></label>
 </div>
      </td>
      <td width="35%"></td>
   </table>
</form>
   <br><br><br><br><br>
</body>
</html>
<%@include file="footer.jsp" %>
```

FOOD DETAILS

```
<%@page import="dbconnect.dbconnection"%>
<%@page import="java.sql.ResultSet"%>
<%@include file="header.jsp" %>
```

```
<div class="mu-title">
   <h2>BREAKFAST REPORT</h2>
   <i class="fa fa-spoon"></i>
   <span class="mu-title-bar"></span>
</div>
<html>
<head>
        <link href="assets/css/table.css" rel="stylesheet" type="text/css"/>
<script>
   function cart(){
      alert("Added to cart");
   }
</script>
</head>
<body>
<table width="100%" height="110px">
<tr>
<td width="20%"></td>
<td width="60%" >
<form method="post" action="cart_action.jsp">
<table id="customers">
   <%
                String id= request.getParameter("id");
                String sql="SELECT * FROM product WHERE item_id="'+id+'"";
                new dbconnection().setConnection();
                ResultSet rs=new dbconnection().getData(sql);
                if(rs.next())
                {%>
                   <tr><th colspan="2">Product Details</th></tr>
                   <tr><td rowspan="6" width="50%"><img
src="../upload/<%=rs.getString(6)%>"></td>
```

```
                    <td width="50%"><input type="hidden" name="id" value="<%
out.println(rs.getString(1)); %>" ></td></tr>

                <tr><td>Food
Name  : <%=rs.getString(2)%></td></tr>
<tr><td>Category     : <%=rs.getString(3)%></
td></tr>
<tr><td>Price          
 : <%=rs.getString(4)%></td></tr>

                <tr><td>Quantity     : 

                <select name="qty"><option value="1">1</option>

                        <option value="2">2</option>

                        <option value="3">3</option>

                        <option value="4">4</option>

                        <option value="5">5</option></select></td></tr>

                <tr><td><center><input type="submit" value="Add to
cart"></center></td></tr>

              <%}

   %>

</table>

</form>

</td>

<td width="20%"></td>

</tr>

</table>

<br><br>

</body>

</html>

<%@include file="footer.jsp" %>
```

CART

```
<%@page import="dbconnect.dbconnection"%>

<%@page import="java.sql.ResultSet"%>

<%@include file="header.jsp" %>
```

```
<div class="mu-title">
   <h2>CART REPORT</h2>
</div>
<html>
   <head>
      <link href="assets/css/table.css" rel="stylesheet" type="text/css"/>
      <script>
         function remove(){
               alert("Deleted cart");
         }
      </script>
   </head>
<body>
<form  method="post">
   <table class="customers" width="100%">
   <tr>
   <td width="5%"></td>
   <td width="90%">
   <table width="100%">

   <tr>
   <th>Cart Id</th>
   <th>Item Name</th>
   <th>Quantity</th>
   <th>Price</th>
   <th>Image</th>
   <th>Total Price</th>
   <th></th>
   </tr>
    <%
         String uname=session.getAttribute("username").toString();
         Integer total=0;
```

```
            String sql="select * from cart where status='Selected' and
username='"+uname+"'";
            new dbconnection().setConnection();
            //out.println(sql);
            ResultSet rs=new dbconnection().getData(sql);
           while(rs.next())
            {%>
            <tr><td><%=rs.getString(1)%></td>
              <td><%=rs.getString(3)%></td>
              <td><%=rs.getString(5)%></td>
              <td><%=rs.getString(4)%></td>
               <td><img src="../upload/<%=rs.getString(6)%>" width="100px"
height="60px"></td>
                <td><%=rs.getString(7)%></td>
                <td><a href="cart_delete.jsp?id=<%=rs.getString(1)%>"
onclick="remove()"><img src="assets/img/delete.png" width="50px"
height="50px"></a></td></tr>
            <%
           total=total+Integer.valueOf(rs.getString(7));
            }
           session.setAttribute("total",total);
    %>
            <tr><td colspan="7" style="text-
align:left;"><p><br>       Total Amount
   :   <% out.println(total); %>    </p>
                    <p>      <a href="breakfast.jsp"
class="button">Continue Shopping</a>  <a
href="user_ship_address.jsp" class="button">Proceed to
Checkout</a></p></td></tr>
    </table>
    </td>
    <td width="5%"></td>
    </tr>
</table>
</form>
```

```
</body>
</html>
<%@include file="footer.jsp" %>
```

SHIPPING ADDRESS

```
<%@page import="dbconnect.dbconnection"%>
<%@page import="java.sql.ResultSet"%>
<%@include file="header.jsp" %>
<div class="mu-title">
   <h2>BILLING ADDRESS</h2>
</div>
<html>
   <head>
      <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
   </head>
<body>
<div class="ship">
<table width="100%">
<tr>
   <td width="10%"></td>
<td class="border" width="80%">
   <table width="100%" border="2">
   <tr>
   <%
         String uname=session.getAttribute("username").toString();
         String sql="select * from shipping_address where username='"+uname+"'";
         new dbconnection().setConnection();
         ResultSet rs=new dbconnection().getData(sql);
         Integer total=0;
         while(rs.next())
         {%>
```

```
<td class="border" width="30.5%"><label class="cont">
        <input type="checkbox" name="address" value="id=<%=rs.getString(1)%>"
class="product-
list"><p><br><%=rs.getString(9)%><br><%=rs.getString(3)%><br><%=rs.getString(
4)%><br><%=rs.getString(5)%><br><%=rs.getString(6)%><br><%=rs.getString(7)%
><br>Pin :<%=rs.getString(8)%></p>
                                <span class="check"></span>
                        <br>
                        <p><a
href="ship_edit.jsp?id=<%=rs.getString(1)%>">Edit</a>   | 
  <a href="ship_delete.jsp?id= <%=rs.getString(1)%>">Delete</a>
                                </label>
</td>
<script type="text/javascript">
$('.product-list').on('change', function() {
   $('.product-list').not(this).prop('checked', false);
});
</script>
      <%}
%>
</tr>
</table>
<br>
<p><a href="ship_address.jsp" class="button">New
Address</a>   <a href="payment.jsp" class="button">Proceed to
Payment</a></p>
</td>

<td width="10%"></td>
</tr>
</table>
<br>
</div>
</body>
</html>
```

```
<%@include file="footer.jsp" %>


PAYMENT
<%@include file="header.jsp" %>
<div class="mu-title">
   <h2>PAYMENT</h2>
   <i class="fa fa-spoon"></i>
   <span class="mu-title-bar"></span>
</div>
<br>
<html>
   <head>
      <link href="assets/css/items.css" rel="stylesheet" type="text/css"/>
      <script>
         function myPay(){
            alert("Paid your order suceessful");
         }
      </script>
   </head>
   <body>
     <table width="100%">
         <tr>
            <td width="35%"></td>
            <td width="30%">
               <form action="payment_action.jsp" method="post">
               <%
                  String total=session.getAttribute("total").toString();
               %>
               <h2 style="color:
#006dcc">Amount   :   <%out.println(total);%>
</h2>
               <br>
               <h4>bank Name</h4>
```

```
                <select name="bname" required>
                    <option value="0">---</option>
                    <option value="SBI Bank">SBI Bank</option>
                    <option value="South Indian Bank">South Indian Bank</option>
                    <option value="Axis Bank">Axis Bank</option>
                    <option value="Union Bank">Union Bank</option>
                    <option value="Federal Bank">Federal Bank</option>
                    <option value="Dhanlaxmi Bank">Dhanlaxmi Bank</option>
                    <option value="Canara Bank">Canara Bank</option>
                </select>
                <h4>Account Number</h4>
                <input type="text" name="account" placeholder="Enter the account
number" required>
                <h4>IFSC code</h4>
                <input type="text" name="pin" placeholder="Enter the pincode"
required>
                <h4>Branch</h4>
                <select name="branch" required>
                    <option value="0">---</option>
                    <option value="Thiruvananthapuram">Thiruvananthapuram</option>
                    <option value="Kollam">Kollam</option>
                    <option value="Pathanamthitta">Pathanamthitta</option>
                    <option value="Alappuzha">Alappuzha</option>
                    <option value="Kottayam">Kottayam</option>
                    <option value="Idukki">Idukki</option>
                    <option value="Ernakulam">Ernakulam</option>
                    <option value="Thrissur">Thrissur</option>
                    <option value="Palakkad">Palakkad</option>
                    <option value="Malappuram">Malappuram</option>
                    <option value="Kozhikode">Kozhikode</option>
                    <option value="Wayand">Wayand</option>
                    <option value="Kannur">Kannur</option>
                    <option value="Kasargod">Kasargod</option>
```

```
                </select><br>
                <input type="submit" name="submit" value="PAY" onclick="MyPay()">
                </form>
              </td>
              <td width="35%"></td>
            </tr>
          </table>
    </body>
</html>
<br>
<%@include file="footer.jsp" %>
```

ORDER DETAILS

```
<%@page import="dbconnect.dbconnection"%>
<%@page import="java.sql.ResultSet"%>
<%@include file="header.jsp" %>
<div class="mu-title">
    <h2>ORDER REPORT</h2>
    <i class="fa fa-spoon"></i>
    <span class="mu-title-bar"></span>
</div>
<br>
<html>
    <head>
      <!--  <link href="assets/css/table.css" rel="stylesheet" type="text/css"/>  -->
    </head>
<body>

<form  method="post" action="payment.jsp">
    <table width="100%">
    <tr>
    <td width="15%"></td>
```

```
<td width="70%">
    <%
        String uname=session.getAttribute("username").toString();
        String sql="select max(date) from cart where username='"+uname+"'";
        ResultSet rs=new dbconnection().getData(sql);
        if(rs.next())
        {
            String date=rs.getString(1).toString();
            //out.println(date);
            String order="Select order_id,date,amount,status from user_order order by
order_id DESC";
            ResultSet rm=new dbconnection().getData(order);
            while(rm.next())
            {%>
            <table width="100%">
            <tr>
                <th>ORDER NUMBER :<%=rm.getString(1)%></th>
                <th></th>
                <th>ORDER DATE : <%=rm.getString(2)%></th>
            </tr>
            <tr>
                <td class="order">ORDER STATUS :
                 <% if(rm.getString(4).equals("Delivered"))
                   {  %>
                      <h4 style="color: green"><%=rm.getString(4)%></h4>
                 <% }
                   if(rm.getString(4).equals("Confirmed"))
                   {  %>
                      <h4 style="color: blue"><%=rm.getString(4)%></h4>
                <% }
                   if(rm.getString(4).equals("Shipped"))
                   {  %>
```

```
                    <h4 style="color: orangered"><%=rm.getString(4)%></h4>
                <% }
                    %>
                </td>
                <td class="order">AMOUNT : <h4><%=rm.getString(3)%></h4></td>
                <td class="order"><a style="color:blue"
href="order_item.jsp?date=<%=rm.getString(2)%>"><h4>Order details</a></td>
            </tr>
            </table>
            <br>
            <%}
         }
        %>
   </td>
   <td width="15%"></td>
   </tr>
</table>
</form>
   <br>
</body>
</html>
<%@include file="footer.jsp" %>
```

RESTAURANT

```
<%@page import="dbconnect.dbconnection"%>
<%@page import="java.sql.ResultSet"%>
<%@include file="header.jsp" %>

<html>
<head>
<script>
   function cart(){
```

```
        alert("Added to cart");
    }
</script>
</head>
<body>
<table width="100%" height="110px">
<tr>
<td width="20%"></td>
<td width="60%" >
<form method="post" action="cart_action.jsp">
<table id="customers">
    <%
                String id= request.getParameter("id");
                String sql="SELECT * FROM addfood_restaurant WHERE
food_id="'+id+'"";
                new dbconnection().setConnection();
                ResultSet rs=new dbconnection().getData(sql);
                if(rs.next())
                {%>
                    <tr><th colspan="2">Product Details</th></tr>
                    <tr><td rowspan="6" width="50%"><img
src="../upload/<%=rs.getString(6)%>"></td>
                        <td width="50%">Product No  : <input
type="text" name="id" value="<% out.println(rs.getString(1)); %>"></td></tr>

                    <tr><td>Food
Name  : <%=rs.getString(2)%></td></tr>

<tr><td>Category     : <%=rs.getString(3)%></
td></tr>
<tr><td>Price          
 : <%=rs.getString(4)%></td></tr>
                    <tr><td>Quantity     : 
                    <select name="qty"><option value="1">1</option>
                            <option value="2">2</option>
```

```
                                        <option value="3">3</option>
                                        <option value="4">4</option>
                                        <option value="5">5</option></select></td></tr>
                        <tr><td><center><input type="submit" value="Add to cart"
onclick="cart()"></center></td></tr>
                    <%}
    %>
</table>
</form>
</td>
<td width="20%"></td>
</tr>
</table>
<br><br>
</body>
</html>
<%@include file="footer.jsp" %>
```

ADD FOOD ITEM

```
<%@include file="header.jsp" %>
<div class="mu-title">
    <h2>ADD ITEM</h2>
    <i class="fa fa-spoon"></i>
    <span class="mu-title-bar"></span>
</div>
<br><br>
<html>
    <head>
        <link href="assets/css/items.css" rel="stylesheet" type="text/css"/>
        <script>
            function myAdd(){
                alert("Added item");
```

```
        }
    </script>
</head>
<body>
    <table width="100%">
        <tr>
            <td width="35%"></td>
            <td width="30%">
                <form action="additem_action.jsp" method="post"
enctype="multipart/form-data">
                <h4>Food name</h4>
                <input type="text" name="name" placeholder="">
                <h4>Food Category</h4>
                <select name="category">
                    <option value="">---</option>
                    <option value="breakfast">Breakfast</option>
                    <option value="lunch">Lunch</option>
                    <option value="dinner">Dinner</option>
                    <option value="beverage">Beverage</option>
                </select>
                <h4>Number of items</h4>
                <input type="text" name="qty" placeholder="">
                <h4>Price</h4>
                <input type="text" name="price" placeholder="">
                <h4>Image</h4>
                <input type="file" name="image" placeholder="">
                <input type="submit" name="submit" value="SAVE ITEM"
onclick="myAdd()">
                </form>
            </td>
            <td width="35%"></td>
        </tr>
    </table>
```

```
    </body>
</html>
<br><br>
<%@include file="footer.jsp" %>
```

VIEW FOOD ITEMS

```
<%@page import="dbconnect.dbconnection"%>
<%@page import="java.sql.ResultSet"%>
<%@include file="header.jsp" %>
<html>
    <head>
        <link href="assets/css/table.css" rel="stylesheet" type="text/css"/>
    </head>
<body>
<div class="mu-title">
    <h2>VIEW ITEMS</h2>
    <i class="fa fa-spoon"></i>
    <span class="mu-title-bar"></span>
</div>
<form  method="post">
<table class="customers" width="100%">

    <tr>
    <td width="5%"></td>

    <td width="90%">
    <table width="100%">
    <th>Item No</th>
    <th>Food Name</th>
    <th>Category</th>
    <th>Price</th>
    <th>Quantity</th>
```

```
   <th>Image</th>
   </tr>
    <%
          String sql="select * from product ";
          new dbconnection().setConnection();
          ResultSet rs=new dbconnection().getData(sql);
          while(rs.next())
          {%>
          <tr><td><%=rs.getString(1)%></td>
             <td><%=rs.getString(2)%></td>
             <td><%=rs.getString(3)%></td>
             <td><%=rs.getString(4)%></td>
             <td><%=rs.getString(5)%></td>
             <td><img src="../upload/<%=rs.getString(6)%>" width="50px"
height="50px"></td>
          <%}
     %>
   </table>
   </td>
   <td width="5%"></td>
   </tr>
</table>
</form>
<br><br>
</body>
</html>
<%@include file="footer.jsp" %>
```

SHIPPING DETAILS

```
<%@page import="dbconnect.dbconnection"%>
<%@page import="java.sql.ResultSet"%>
<%@include file="header.jsp" %>
```

```
<div class="mu-title">
   <h2>SHIPPING REPORT</h2>
</div>
<html>
   <head>
   </head>
<body>
<br><br>
<table class="customers" width="100%">
   <form action="post">
<tr>
   <td width="10%"></td>
   <td width="80%">
   <table width="100%">
      <tr>
         <th>Order ID</th>
         <th>Username</th>
         <th>Order Date</th>
         <th>Amount</th>
         <th>Order Status</th>
         <th>Action</th>
      </tr>
      <%
         //String uname=session.getAttribute("username").toString();
         String sql="select max(date) from cart";
         ResultSet rs=new dbconnection().getData(sql);
         if(rs.next())
         {
            String date=rs.getString(1).toString();
            //out.println(date);

            String order="Select * from user_order order by order_id desc";
```

```
            ResultSet rm=new dbconnection().getData(order);
            while(rm.next())
            {%>
        <tr>
            <td><%=rm.getString(1)%></td>
            <td><%=rm.getString(5)%></td>
            <td><%=rm.getString(2)%></td>
            <td><%=rm.getString(3)%></td>
            <td><%=rm.getString(4)%></td>
            <td><a href="staff_order_status.jsp?id=<%=rm.getString(1)%>">Edit</a> |
<a href="staff_order_item.jsp?date=<%=rm.getString(2)%>">View Items</a></td>
        </tr>
              <%}
            }
           %>
    </table>
    </form>
    </td>
    <td width="10%"></td>
</tr>
</table>
<br>
</body>
</head>
</html>
<%@include file="footer.jsp" %>
```

13 REFERENCE

www.w3schools.com

https://en.wikipedia.org/wiki/Online_food_ordering

www.slideshare.net

https://www.freeprojectz.com/paid-projects/online-food-ordering-system

https://www.scribd.com

http://www.wampserver.com/en

https://developer.mozilla.org

https://colorlib.com

https://www.freeprojectz.com

https://1000projects.org/food-ordering-management-system-php-mysql-project.html

https://www.youtube.com/watch?time_continue=4&v=Sy3o4yv8oM4

http://services.lovelycoding.org/food-ordering-system

https://meeraacademy.com/dfd-diagram-for-online-food-ordering-system