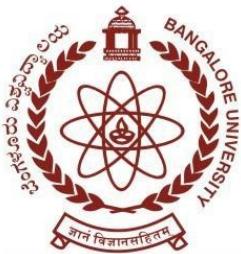


**A Project Report on**  
**“FOOD ORDERING SYSTEM”**



**Submitted to the Bangalore University in partial fulfillment  
of the requirement for the award of the degree in  
Bachelor of Computer Applications**

**Submitted by**

**Charan PT**

**USN : 16VFSB7012**

**Under the guidance of  
Ms. APOORVA A  
Asst. Professor**



**Ring Road, Marathalli, Bangalore – 560 103**

## **DECLARATION**

I hereby declare that the project report entitled “FOOD ORDERING SYSTEM” was prepared by me during the year 2018-2019 and was submitted to the Bangalore University, in partial fulfillment of the requirements for the award of the Degree in Bachelor of Computer Applications. I also declare that this project report is original and genuine and it has not been submitted to any other University for the award of any degree, diploma or other similar titles or purposes.

**Date:**

**Signature of the Candidate**

**Place: NewHorizonCollege**

**Charan PT**

**(Marathalli)**

**[ USN: 16VFSB7012 ]**

## **CERTIFICATE**

This is to certify that the project report entitled "**“FOOD ORDERING SYSTEM”**" Submitted by **Charan PT[16VFSB7012]**. This report is an outcome of genuine project work and has been submitted in partial fulfillment for the award of the Degree of Bachelor of Computer Applications, awarded by Bangalore University, during the academic year 2018-19.

**ProjectGuide**

Ms. APOORVA A

**Head of theDepartment**

Dr. Nagaraju Kilari

**Signature of theExaminers**

1. \_\_\_\_\_

2. \_\_\_\_\_

## **ACKNOWLEDGEMENT**

I express my deep sincere thanks to **Dr. Mohan Maghnani, Chairman of New Horizon Educational Institutions** for providing the platform and infrastructure to do this project. This project would not have been possible without the help and co-operation of many. At the outset, I wish to record my profound gratitude to our principal **Dr. R. Bodhisatvan**

I am extremely thankful to **Dr Nagaraju Kilari**, HOD, Department of Computer Applications for his unending support and encouragement during the development of this project.

I would like to acknowledge the interest and the support extended by our project guide **Ms. APOORVA A**, Asst. Professor, Department of Computer Applications, to make this project implementation successful.

Above all I would thank God for the blessings and my parents also for their valuable suggestion and support in my project Report.

Finally, I extend my deep sense of gratitude to all those who made this project come alive and encouraged and guided me from the start to finish.

# INDEX

CHAPTER		CONTENTS	PAGE NO.
01.		<b>INTRODUCTION</b>	1
	1.1	Abstract	2
	1.2	Problem Statement	3
	1.3	Module Description	4-6
02.		<b>SYSTEM ANALYSIS</b>	7-10
	2.1	Introduction to System Analysis	6-13
	2.1.1	Existing System	14
	2.1.2	Proposed System	15-16
	2.2	Software Requirement Specification(SRS)	17
	2.3	Hardware Requirements	28-30
	2.4	Software Requirements	
	2.5	Introduction to HTML	31-32
	2.6	Introduction to JavaScript	33-35
	2.7	Introduction to PHP	36-38
	2.8	Introduction to BootStrap	39-41
	2.9	Introduction to WebServer	42-45
03.		<b>SYSTEM DESIGN</b>	46-48
	3.1	Data Flow Diagram	37-50
	3.2	Table Structure	50-55
04.		<b>TESTING AND IMPLEMENTATION</b>	56
	4.1	Introduction to Testing	57
	4.2	Types of Testing	58

05.		CODING AND DEVELOPMENT	
	5.1	Form Design	59-63
	5.2	Source Code	64-113
06.		CONCLUSION AND FUTURE ENHANCEMENT	114-116
07.		BIBLIOGRAPHY	117-118

## **INTRODUCTION**

Nowadays, digital business platforms are very popular and save us much effort and time in our daily life. E-commerce companies such as Amazon and Ebay could deliver goods to customers very efficiently. On the one hand, customers could select goods and place orders online without visiting the shop, which is usually time-consuming. Besides, they do not need to carry the goods to home. Instead, the shop would deliver the goods and save customers' efforts. On the other hand, using digital business platforms could make it more convenient for shop owners to manage orders, collect and analyse data and provide better service. In the catering industry, the demand of combining the convenience of digital business with their traditional delivery service is increasingly growing. Unlike common e-commerce companies, the restaurants usually could deliver food in less than half an hour and actually saves customers' time when compared to visiting the restaurants.

In order to stand out in the digital business trend of catering industry and provide more satisfying service, we designed this Pizza Hot project. Customers usually expect fast delivery and food in good condition to eat. So in order to adapt to customers' expectation and earn more profits for restaurant owners, we improved the traditional digital business platform. In addition to the general functionalities of e-commerce platform, we developed a hardware system which is intended to be put in the delivery box. After the restaurant receives customers' orders and the dishes are ready to deliver, the delivery man put the portable system in the delivery box. The system could record the temperature in the box and the time of delivery. After the dishes arrives, it will calculate a suggested tip for customers based on temperature, time and distance the delivery man covered. If the food's temperature is suitable, the tips could be higher and otherwise the system gives a discount. In this way, we could realize a win-win relationship between customer and restaurants. Customers can expect fast delivery and hot food. Restaurants can make more money by providing good service, which also helps them stands out among competitors. Delivery staffs will be provided information for more efficient operations such as turn by turn navigation.

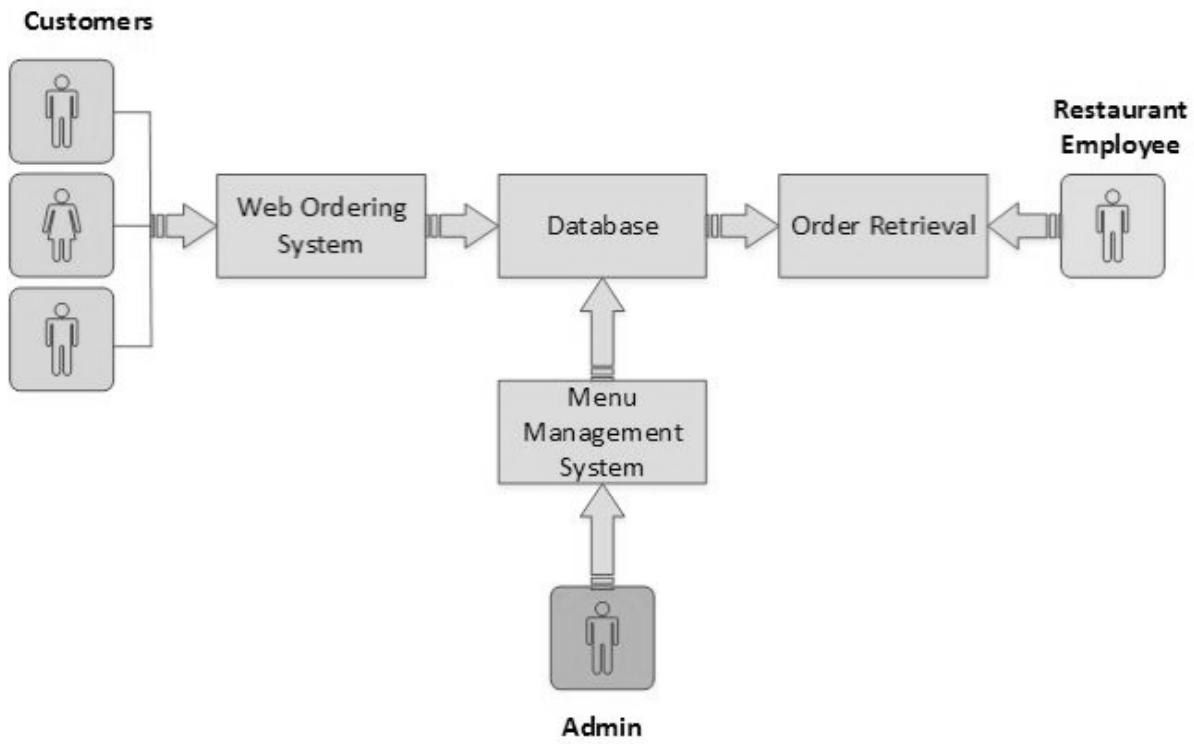
The online food ordering system provides convenience for the customers. It overcomes the disadvantages of the traditional queuing system. This system increases the takeaway of foods than visitors. Therefore, this system enhances the speed and standardization of taking the order from the customer.

It provides a better communication platform. The user's details are noted electronically.

The online food ordering system set up menu online and the customers easily places the order with a simple mouse click. Also with a food menu online you can easily track the orders, maintain customer's database and improve your food delivery service. This system allows the user to select the desired food items from the displayed menu. The user orders the food items. The payment can be made online or pay-on-delivery system. The user's details are maintained confidential because it maintains a separate account for each user. An id and password is provided for each user. Therefore it provides a more secured ordering.

It is known globally that, in today's market, it is extremely difficult to start a new small-scale business and live-through the competition from the well-established and settled owners. In fast paced time of today, when everyone is squeezed for time, the majority of people are finicky when it comes to placing a food order. The customers of today are not only attracted because placing an order online is very convenient but also because they have visibility into the items offered, price and extremely simplified navigation for the order.

Online food ordering system that I am proposing here, greatly simplifies the ordering process for both the customer and the restaurant. System presents an interactive and up-to-date menu with all available options in an easy to use manner. Customer can choose one or more items to place an order which will land in the Cart. Customer can view all the order details in the cart before checking out. At the end, customer gets order confirmation details. Once the order is placed it is entered in the database and retrieved in pretty much real time. This allows Restaurant Employees to quickly go through the orders as they are received and process all orders efficiently and effectively with minimal delays and confusion.



## OBJECTIVE

**This Project is aimed to provide**

- ✓ An order system on multi-platforms for customers to select dishes and place orders
- ✓ A convenient management dashboard for restaurant manager to easily manage the whole system
- ✓ A smart delivering system for helping delivery staff improves the quality of delivery service.

## **REQUIREMENTS AND CONSTRAINTS**

As can be seen in the system model diagramed above, each of the three system components essentially provides a layer of isolation between the end user and the database. The motivation behind this isolation is twofold. Firstly, allowing the end user to interact with the system through a rich interface provide a much more enjoyable user experience, particularly for the non-technical users which will account for the majority of the system's users. In addition, this isolation layer also protects the integrity of the database by preventing users from taking any action outside those which the system is designed to handle. Because of this design pattern, it is essential to enumerate exactly which functions a user will be presented and these functions are outlined below, grouped by component.

### **Web Ordering System**

Users of the web ordering system, namely restaurant customers, must be provided the following functionality:

- Create an account.
- Manage their account.
- Log in to the system.
- Navigate the restaurant's menu.
- Select an item from the menu.
- Customize options for a selected item.
- Add an item to their current order.
- Review their current order.
- Remove an item/remove all items from their current order.
- Provide delivery and payment details.
- Place an order.
- Receive confirmation in the form of an order number.

As the goal of the system is to make the process of placing an order as simple as possible for the customer, the functionality provided through the web ordering system is restricted to that

which most pertinent to accomplish the desired task. All of the functions outlined above, with the exceptions of account creation and management, will be used every time a customer places an order. By not including extraneous functions, I am moving towards my goal of simplifying the ordering process.

## **Menu Management system**

The menu management system will be available only to restaurant employees and will, as the name suggests, allow them to manage the menu that is displayed to users of the web ordering system. The functions afforded by the menu management system provide user with the ability to, using a graphical interface:

- Add a new/update/delete vendor to/from the menu.
- Add a new/update/delete food category to/from the menu.
- Add a new/update/delete food item to/from the menu.
- Add a new/update/delete option for a given food item.
- Update price for a given food item.
- Update default options for a given food item.
- Update additional information (description, photo, etc.) for a given food item.

It is anticipated that the functionality provided by this component will be one of the first things noted by the restaurant user, as they will have to go through it to configure their menu, etc. before beginning to actually take orders. Once everything is initially configured, however, this component will likely be the least used, as menu updates generally do not occur with great frequency.

Of the three components, the order retrieval system is functionally the simplest. Like the menu management system, it is designed to be used only by restaurant employees, and provides the following functions:

- Retrieve new orders from the database.
- Display the orders in an easily readable, graphical way.
- Mark an order as having been processed and remove it from the list of active orders.

## **SYSTEM ANALYSIS**

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis 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 viewed as a whole, the inputs are identified and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

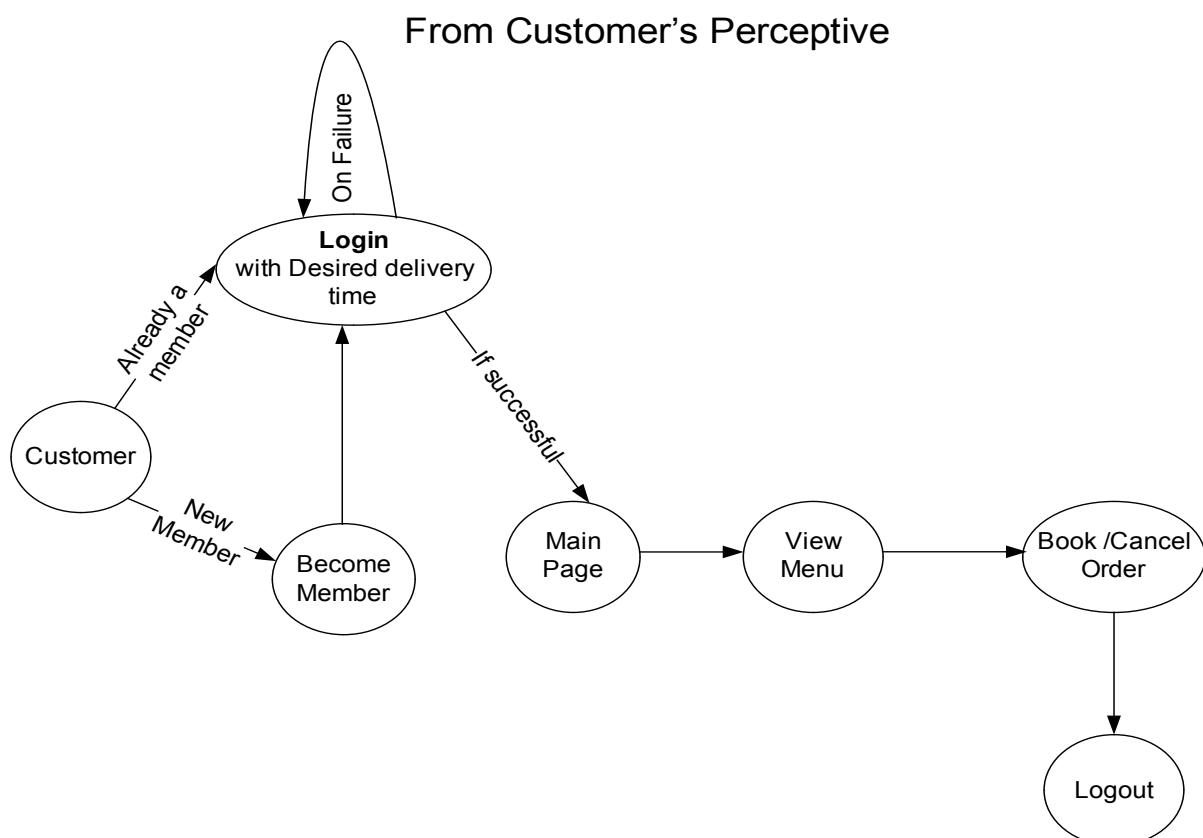
### **Problem Analysis**

As discussed earlier our main problem area focuses on the “Meal reservation/booking system”, there are lot of problems in that area which are associated with both the customer and the restaurant staff.

We would like to analyze some of the problems here:

- a. Initial problem is that the customer has to get connected over the phone, it would be harder if the restaurant is very popular and busy.
- b. As customer won't have the menu list with him, it would be harder for him to remember the entire list (with price as well...!) and come to a decision, i.e. customer is provided with less time to make decision.
- c. The chances of committing mistakes at the restaurant side in providing a menu list for a specific time would be more.

- d. There might be some communication problems or sometimes language might be a barrier.
- e. As entire booking has to be done manually at the restaurant end, the chances of occurrence of mistakes is high as well.
- f. Most of restaurants have single phone line and a single operator to handle incoming calls, so they can accept limited orders.
- g. If the restaurant is of busy type, than the operator is left with no time to decide over the priority of the order fulfillment.
- h. Even assigning orders (or some menu from the order) to a specific cook can be cumbersome if it is done parallel with the bookings of the order.
- i. All the calls will not be intended for booking, as some calls might be for canceling the order or to fetch the status as well, this eats up the productive time at the restaurant side.

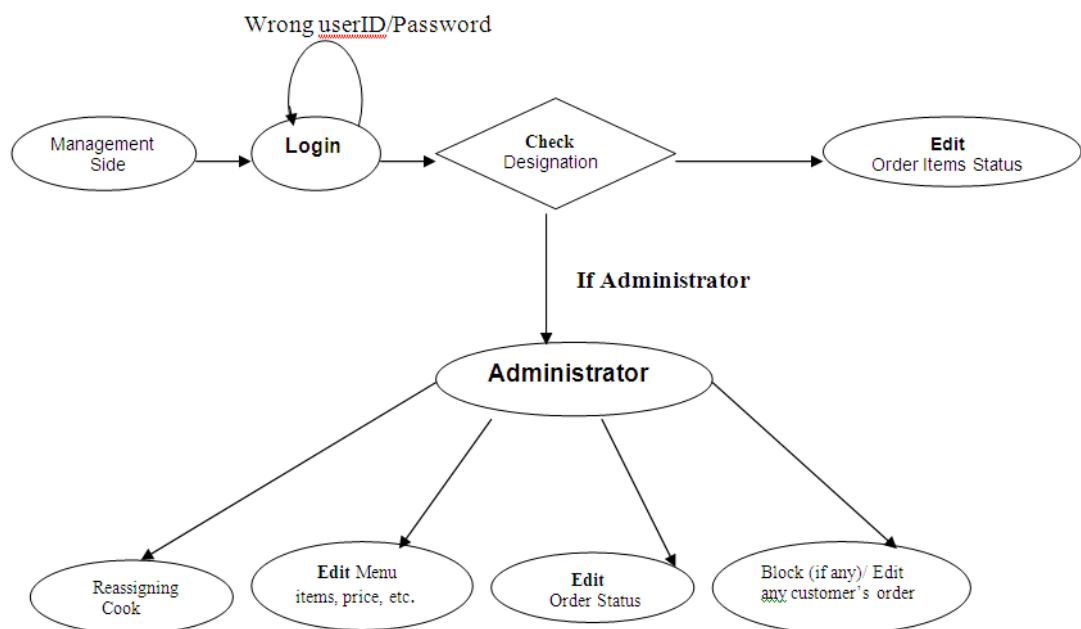


To register a meal online, the customer has to become a member first then he can access the later part of the site. The option of becoming member was only an attempt to avoid (to some extent) placing the fake bookings.

During login the user has to enter his email/UserID, password and desired time of order delivery. After successful login the customer can access the menu page with the items listed according to the desired time. Later within the available items he can search for a menu according to his choice i.e. according to price range and category of food and later he can order a meal.

If the customer later wants to cancel the order, he is permitted to do this only within a specific time period. The customer is also given with the facility to view the status of the order and if the order is ready then he can go and get it.

### **From Management's Perspective**

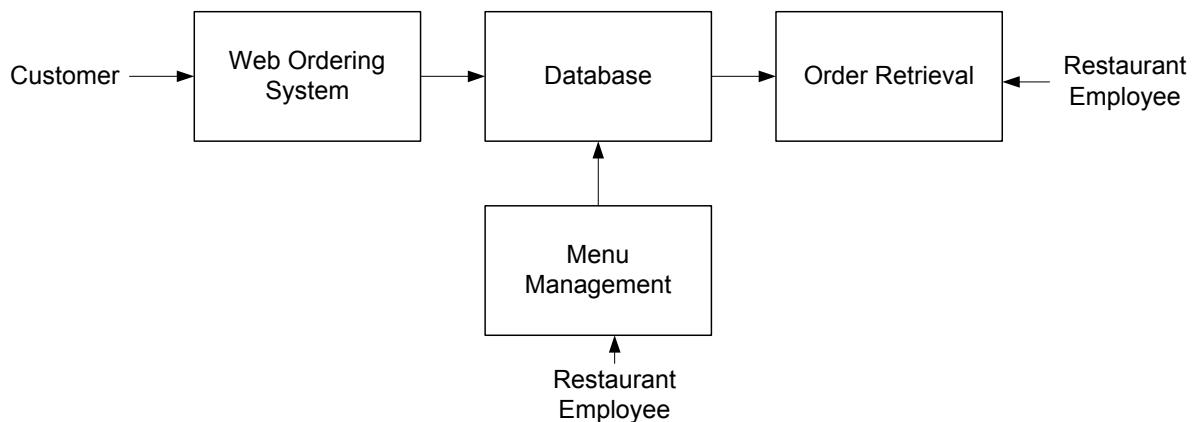


At Management side, initially the staff member has to login, and according to his designation the privileges are set. If the staff member is a cook, then he is allowed to edit only the order items status, indicating which menu items he has prepared.

If suppose the member is an administrator then, he is allowed to reassign the cook according to his priority, he can edit the menu information such as its price, items available currently, etc. He can also change the status of the order (in some special cases), and can also block (if any customer exists)/Edit any customer's order according to his priority.

## System Model

The structure of the system can be divided into three main logical components. The first component must provide some form of menu management, allowing the restaurant to control what can be ordered by customers. The second component is the web ordering system and provides the functionality for customers to place their order and supply all necessary details. The third and final logical component is the order retrieval system. Used by the restaurant to keep track of all orders which have been placed, this component takes care of retrieving and displaying order information, as well as updating orders which have already been processed.



## **FEASIBILITY STUDY**

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are listed below.

- ✓ To analyze whether the software will meet organizational requirements
- ✓ To determine whether the software can be implemented using the current technology and within the specified budget and schedule
- ✓ To determine whether the software can be integrated with other existing software

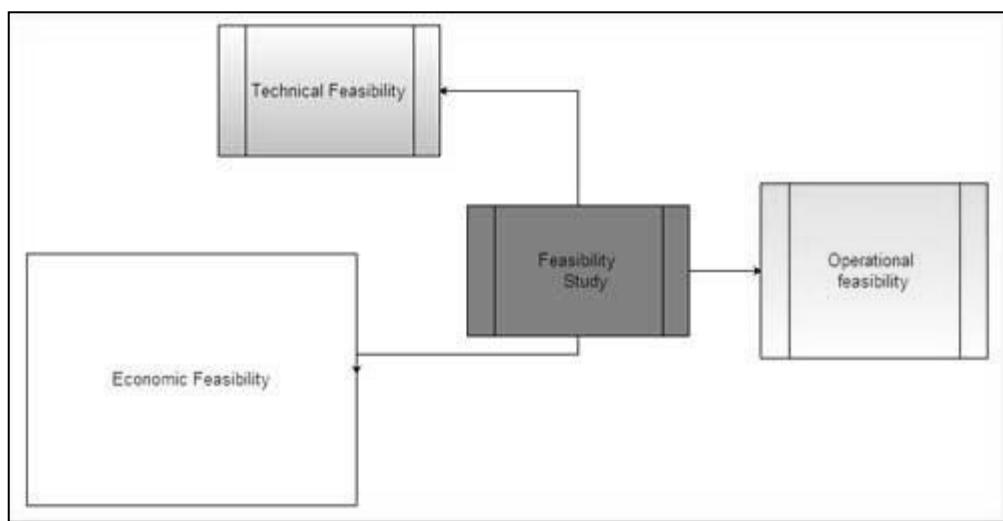
**Consequently, costs and benefits are described with greater accuracy at this stage. It consists of the following:**

- ✓ Statement of the problem: A carefully worded statement of the problem that led to analysis.

- ✓ Summary of finding and recommendations: A list of the major findings and recommendations of the study. It is ideal for the user who requires quick access to the results of the analysis of the system under study. Conclusion are stated , followed by a list of the recommendation and a justification for them.
- ✓ Details of findings :An outline of the methods and procedures under-taken by the existing system, followed by coverage of the objectives and procedures of the candidate system. Included are also discussions of output reports, file structures, and costs and benefits of the candidate system.
- ✓ Recommendations and conclusions: Specific recommendations regarding the candidate system, including personnel assignments, costs, project schedules, and target dates.

## Types of Feasibility

Various types of feasibility that are commonly considered include technical feasibility, operational feasibility, and economic feasibility.



### Technical feasibility

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. Technical feasibility also performs the following tasks.

- ✓ Analyzes the technical skills and capabilities of the software development team members
- ✓ Determines whether the relevant technology is stable and established
- ✓ Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

## **Operational feasibility**

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

- ✓ Determines whether the problems anticipated in user requirements are of high priority
- ✓ Determines whether the solution suggested by the software development team is acceptable
- ✓ Analyzes whether users will adapt to a new software
- ✓ Determines whether the organization is satisfied by the alternative solutions proposed by the software development team

## **Economic feasibility**

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

- ✓ Cost incurred on software development to produce long-term gains for an organization
- ✓ Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis)
- ✓ Cost of hardware, software, development team, and training

# **EXISTING SYSTEM AND PROPOSED SYSTEM**

After analyzing the necessities of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the necessities and domain of the new system. Both the behaviors are equally significant, but the first movement serves as a basis of giving the purposeful specifications and then winning design of the proposed system. Understanding the properties and necessities of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

## **Existing System**

### **Drawbacks of Existing System**

- ❖ As the current system is totally manual
- ❖ Existing system is manually, so it increases the chances of errors.
- ❖ Lot of the time consumed for each report generation
- ❖ Immediate response to the query's is difficult
- ❖ More stationary use so they are expensive
- ❖ Manual systems are takes more time
- ❖ More man power.
- ❖ Consumes large volume of pare work.
- ❖ Damage of machines due to lack of attention.

## **Proposed system**

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper

security and reduces the manual work. The existing system has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system will help the user to reduce the workload and mental conflict. The proposed system helps the user to work user friendly and he can easily do his jobs without time lagging.

### **Expected Advantages of Proposed System**

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features

- ❖ System can generate immediately getting the data and report.
- ❖ Avoid stationary expense
- ❖ New system provide online payment facility
- ❖ Any record is easy to store and manage
- ❖ Easy to solve customer query
- ❖ Provide better security in new system
- ❖ Give the feedback answer
- ❖ Ensure data accuracy's.
- ❖ Proper control of the higher officials.
- ❖ Reduce the damages of the machines.
- ❖ Minimize manual data entry.
- ❖ Minimum time needed for the various processing.
- ❖ Greater efficiency.
- ❖ Better service.
- ❖ User friendliness and interactive.
- ❖ Minimum time required.

## **Important Features**

- ✓ Accuracy
- ✓ User Friendly
- ✓ Availability
- ✓ Efficiency
- ✓ Reliable
- ✓ Durable

## **SPECIFICATION REQUIREMENT**

Requirement analysis for web applications encompasses three major tasks: formulation, requirements gathering and analysis modeling. During formulation, the basic motivation and goals for the web application are identified, and the categories of users are defined. In the requirements gathering phase, the content and functional requirements are listed and interaction scenarios written from end-user's point-of-view are developed. This intent is to establish a basic understanding of why the web application is built, who will use it, and what problems it will solve for its users.

### **SOFTWARE REQUIREMENT SPECIFICATION**

Operating System	:	Windows XP/7
Presentation layer	:	PHP, HTML, JS, CSS, FLASH & PHOTOSHOP
Database	:	My SQL

### **HARDWARE REQUIREMENT SPECIFICATION**

Processor	:	Standard processor with a speed of 2.0GHz
RAM	:	2BG
Hard Disk	:	50 GB or more
Monitor	:	Standard color monitor
Keyboard	:	Standard keyboard
Mouse	:	Standard mouse

## TECHNOLOGIES USED

**HTML** provides the *basic structure* of sites, which is enhanced and modified by other technologies like CSS and JavaScript.

**CSS** is used to control *presentation, formatting, and layout*.

**JavaScript** is used to control the *behavior* of different elements.

### HTML

**Hypertext Markup Language (HTML)** is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <img/> and <input/> directly introduce content into the page. Other tags such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

## CSS

CSS Stands for "Cascading Style Sheet." Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.

CSS helps Web developers create a uniform look across several pages of a Web site. Instead of defining the style of each table and each block of text within a page's HTML, commonly used styles need to be defined only once in a CSS document. Once the style is defined in cascading style sheet, it can be used by any page that references the CSS file. Plus, CSS makes it easy to change styles across several pages at once. For example, a Web developer may want to increase the default text size from 10pt to 12pt for fifty pages of a Web site. If the pages all reference the same style sheet, the text size only needs to be changed on the style sheet and all the pages will show the larger text.

While CSS is great for creating text styles, it is helpful for formatting other aspects of Web page layout as well. For example, CSS can be used to define the cell padding of table cells, the style, thickness, and color of a table's border, and the padding around images or other objects. CSS gives Web developers more exact control over how Web pages will look than HTML does. This is why most Web pages today incorporate cascading style sheets.

### Example

CSS is used to style and lay out web pages — for example, to alter the font, colour, size and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module gets you started on the path to CSS mastery with the basics of how it works, including selectors and properties, writing CSS rules, applying CSS to HTML, how to specify length, colour, and other units in CSS, cascade and inheritance, and debugging CSS.

```
<html>
<head>
  <meta charset="utf-8">
  <title>My CSS experiment</title>
```

```
<link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Hello World!</h1>
  <p>This is my first CSS example</p>
</body>
</html>
```

**Now let's look at a very simple CSS example containing two rules:**

```
h1 {
  color: blue;
  background-color: yellow;
  border: 1px solid black;
}
```

```
p {
  color: red;
}
```



At this point you have some choices of how to use the CSS, either internally or externally.

## **Internal Style sheet**

First we will explore the internal method. This way you are simply placing the CSS code within the `<head></head>` tags of each (X)HTML file you want to style with the CSS. The format for this is shown in the example below.

```
<head>
<title><title>
<style type="text/css">
CSS Content Goes Here
</style>
</head>
<body>
```

With this method each (X)HTML file contains the CSS code needed to style the page. Meaning that any changes you want to make to one page, will have to be made to all. This method can be good if you need to style only one page, or if you want different pages to have varying styles.

## **External Style sheet**

Next we will explore the external method. An external CSS file can be created with any text or HTML editor such as “Notepad” or “Dreamweaver”. A CSS file contains no (X)HTML, only CSS. You simply save it with the .css file extension. You can link to the file externally by placing one of the following links in the head section of every (X)HTML file you want to style with the CSS file.

```
<link rel="stylesheet" type="text/css" href="Path To stylesheet.css" />
```

Or you can also use the `@import` method as shown below

```
<style type="text/css">@import url(Path To stylesheet.css)</style>
```

Either of these methods are achieved by placing one or the other in the head section as shown in example below.

```

<head>
<title><title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>

or

<head>
<title><title>
<style type="text/css"> @import url(Path To stylesheet.css) </style>
</head>
<body>

```

By using an external style sheet, all of your (X)HTML files link to one CSS file in order to style the pages. This means, that if you need to alter the design of all your pages, you only need to edit one .css file to make global changes to your entire website.

Here are a few reasons this is better.

- Easier Maintenance
- Reduced File Size
- Reduced Bandwidth
- Improved Flexibility

## JavaScript

**JavaScript** often abbreviated as **JS**, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content engineering. It is used to make dynamic web pages interactive and provide online programs, including video games. The majority of websites employ it, and all modern web browsers support it without the need for plug-ins by means of a built-in JavaScript engine. Each of the many JavaScript engines represent a different implementation of JavaScript, all based on the ECMAScript specification, with some engines not supporting the spec fully, and with many engines supporting additional features beyond ECMA.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as Self and Scheme.

## **What is a Web Server?**

Wondering what the heck is a web server? Well a web server is like a restaurant host. When you arrive in a restaurant, the host greets you, checks your booking information and takes you to your table. Similar to the restaurant host, the web server checks for the web page you have requested and fetches it for your viewing pleasure. However, A web server is not just your host but also your server. Once it has found the web page you requested, it also serves you the web page. A web server like Apache, is also the Maitre D' of the restaurant. It handles your communications with the website (the kitchen), handles your requests, makes sure that other staff (modules) are ready to serve you. It is also the bus boy, as it cleans the tables (memory, cache, modules) and clears them for new customers.

So basically a web server is the software that receives your request to access a web page. It runs a few security checks on your HTTP request and takes you to the web page. Depending on the page you have requested, the page may ask the server to run a few extra modules while

generating the document to serve you. It then serves you the document you requested. Pretty awesome isn't it.

## What is Apache?

The Apache web server, more popular as simply Apache, represents an open-source web server platform lying in the basis of most of the websites we see today on the World Wide Web. Looking back at the time when it was introduced in mid 90's and gradually adopted as a preferred server platform on the web, we could state that Apache acted as the main driving force behind today's web expansion. As a web server 'pioneer', Apache has turned into a standard for the development of other successful web server platforms.

The Apache web server is a work of the Apache Software Foundation open source community. Namely the fact that it is backed up by the efforts of many supporters worldwide keeps it so well maintained and regularly updated with new useful features and functionalities up to the latest quality and security requirements in HTTP service delivery.

Where does that strange name of the popular server come from? There are two intriguing and radically different stories behind its origin. The more popular one says that naming the server this way is a kind of a tribute to the Native American Indian tribe Apache, known for its fighting strength and dauntless spirit. According to the other story, the name 'Apache server' represents a sound analogue to 'a patchy server', with 'patchy' referring to the bundle of patches that are attached to the codebase of NCSA HTTPd 1.3.

## MySQL

MySQL is a database management system that is used by WordPress to store and retrieve all your blog information. Think of it this way. If your database is a filing cabinet that WordPress uses to organize and store all the important data from your website (posts, pages, images, etc), then MySQL is the company that created this special type of filing cabinet.

MySQL is an open source relational database management system. It runs as a server and allows multiple users to manage and create numerous databases. It is a central component in the LAMP stack of open source web application software that is used to create websites.

LAMP stands for Linux, Apache, MySQL, and PHP. Most WordPress installations use the LAMP stack because it is open source and works seamlessly with WordPress.

WordPress requires MySQL to store and retrieve all of its data including post content, user profiles, and custom post types. Most web hosting providers already have MySQL installed on their web servers as it is widely used in many open source web applications such as WordPress.

WordPress uses the PHP programming language to store and retrieve data from the MySQL database. To retrieve data from the database, WordPress runs SQL queries to dynamically generate content. SQL stands for Structured Query Language and is the programming language typically used to query databases.

For users that are not comfortable writing their own PHP and SQL scripts, most web hosting providers offer easy to use web applications to manage databases. One such web application is phpMyAdmin which allows users to manage their database using a web based graphical interface. You can manipulate your tables visually while phpMyAdmin runs the SQL queries for you.

The screenshot shows the phpMyAdmin interface for managing a MySQL database named 'base2'. The left sidebar lists various databases and tables, with 'base2' currently selected. The main area displays the 'persons' table structure. The table has five fields: 'id' (tinyint(4), auto\_increment), 'person\_name' (varchar(200)), 'town\_code' (varchar(5)), 'country\_code' (char(1)), and 'car\_code' (char(3)). Below the table structure, there are sections for 'Indexes' (listing PRIMARY, town\_code, country\_code, and pays-ville indexes) and 'Space usage' (showing Data, Index, and Total sizes). A warning message at the bottom states 'More than one INDEX key was created for column country\_code'.

Out of the box MySQL doesn't come with any graphical interface and advanced level users may prefer to just manage their databases from the command line.

## **What is PHP?**

PHP is a server side scripting language. that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages.

PHP scripts can only be interpreted on a server that has PHP installed.

The client computers accessing the PHP scripts require a web browser only.

A PHP file contains PHP tags and ends with the extension ".php".

## **What is a Scripting Language?**

A script is a set of programming instructions that is interpreted at runtime.

A scripting language is a language that interprets scripts at runtime. Scripts are usually embedded into other software environments.

The purpose of the scripts is usually to enhance the performance or perform routine tasks for an application.

Server side scripts are interpreted on the server while client side scripts are interpreted by the client application.

PHP is a server side script that is interpreted on the server while JavaScript is an example of a client side script that is interpreted by the client browser. Both PHP and JavaScript can be embedded into HTML pages.

## **What does PHP stand for?**

PHP means - **P**ersonal **H**ome **P**age, but it now stands for the recursive backronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks.

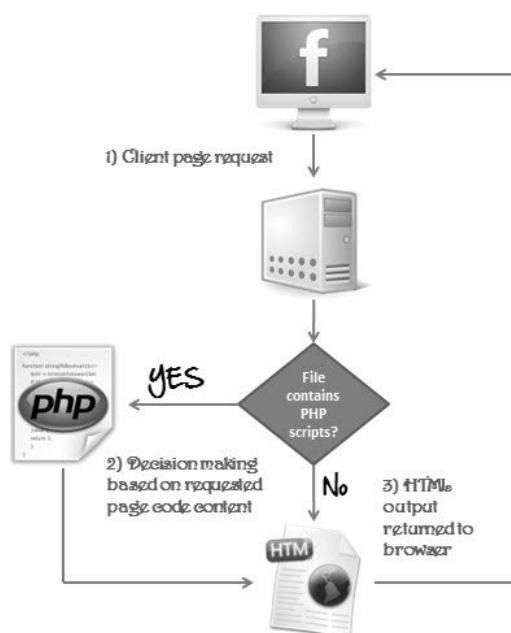
## **Php Syntax**

```
<?php
    echo 'Hello World';
?>
```

A PHP file can also contain tags such as HTML and client side scripts such as JavaScript.

- **HTML is an added advantage** when learning PHP Language. You can even learn PHP without knowing HTML but it's recommended you at least know the basics of HTML.
- **Database management systems** DBMS for database powered applications.
- For more advanced topics such as interactive applications and web services, you will need **JavaScript and XML**.

The flowchart diagram shown below illustrates the basic architecture of a PHP web application and how the server handles the requests.



## Why use PHP?

You have obviously heard of a number of programming languages out there; you may be wondering why we would want to use PHP as our poison for the web programming. Below are some of the compelling reasons.

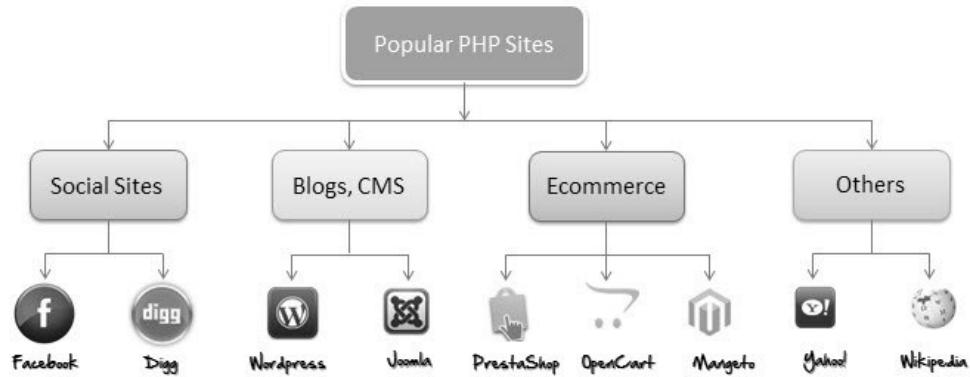
- PHP is **open source and free**.
- Short learning curve compared to other languages such as JSP, ASP etc.
- Large community document
- Most web hosting servers support PHP by default unlike other languages such as ASP that need IIS. This makes PHP a cost effective choice.
- PHP is regularly updated to keep abreast with the latest technology trends.
- Other benefit that you get with PHP is that it's a **server side scripting language**; this means you only need to install it on the server and client computers requesting for resources from the server do not need to have PHP installed; only a web browser would be enough.
- PHP has **in built support for working hand in hand with MySQL**; this doesn't mean you can't use PHP with other database management systems. You can still use PHP with
  - Postgres
  - Oracle
  - MS SQL Server
  - ODBC etc.
- PHP is **cross platform**; this means you can deploy your application on a number of different operating systems such as windows, Linux, Mac OS etc.

## What is PHP used for & Market share

In terms of market share, there are over 20 million websites and applications on the internet developed using PHP scripting language.

This may be attributed to the points raised above;

The diagram below shows some of the popular sites that use PHP



## PHP vs Asp.Net VS JSP VS CFML

ASP – Active Server Pages, JSP – Java Server Pages, CFML – Cold Fusion Markup language

The table below compares the various server side scripting languages with PHP

FEATURE	PHP	ASP	JSP	CFML
Learning curve	short	Longer than PHP	Longer than PHP	Longer than PHP
Web hosting	Supported by almost all hosting servers	Needs dedicated server	Fairly supported	Needs dedicated server
Open source	Yes	No	Yes	Both commercial and open source
Web services support	Built in	Uses the .NET framework	Uses add on libraries	Built in
Integration with HTML	Easy	Fairly complex	Fairly complex	Easy
MySQL support	Native	Needs third party drivers	Needs third party drivers	Current version has native support. Older versions use ODBC

Easily extended by other languages	Yes	No	Extended using Java classes and libraries.	Yes
------------------------------------	-----	----	--	-----

## PHP File Extensions

*File extension and Tags* In order for the **server** to **identify** our **PHP files** and **scripts**, we must **save the file** with the “**.php**” **extension**. Older PHP file extensions include

- .phtml
- .php3
- .php4
- .php5
- .phps

PHP was designed to work with HTML, and as such, it can be embedded into the HTML code.

**<HTML> <PHP CODE> </HMTL>**

You can create PHP files without any html tags and that is called Pure PHP file.

The server interprets the PHP code and outputs the results as HTML code to the web browsers.

In order for the server to identify the PHP code from the HTML code, we must always enclose the PHP code in PHP tags.

A PHP tag starts with the less than symbol followed by the question mark and then the words “php”.

PHP is a case sensitive language, “VAR” is not the same as “var”.

The PHP tags themselves are not case-sensitive, but it is strongly recommended that we use lower case letter. The code below illustrates the above point.

```
<?php ... ?>
```

We will be referring to the PHP lines of code as statements. PHP statements end with a semi colon (;). If you only have one statement, you can omit the semi colon. If you have more than one statement, then you must end each line with a semi colon. For the sake of consistency, it is recommended that you always end your statement(s) with a semi colon. PHP scripts are executed on the server. The output is returned in form of HTML.

## **What is Bootstrap?**

Bootstrap is a powerful front-end framework for faster and easier web development. It includes HTML and CSS based design templates for common user interface components like Typography, Forms, Buttons, Tables, Navigations, Dropdowns, Alerts, Modals, Tabs, Accordion, Carousel and many other as well as optional JavaScript extensions.

Bootstrap also gives you ability to create responsive layout with much less efforts.

## **Advantages of Bootstrap**

The biggest advantage of using Bootstrap is that it comes with free set of tools for creating flexible and responsive web layouts as well as common interface components.

Additionally, using the Bootstrap data APIs you can create advanced interface components like Scrollspy and Typeahead without writing a single line of JavaScript.

## **Here are some more advantages, why one should opt for Bootstrap:**

**Save lots of time** — You can save lots of time and efforts using the Bootstrap predefined design templates and classes and concentrate on other development work.

**Responsive features** — Using Bootstrap you can easily create responsive designs. Bootstrap responsive features make your web pages to appear more appropriately on different devices and screen resolutions without any change in markup.

**Consistent design** — All Bootstrap components share the same design templates and styles through a central library, so that the designs and layouts of your web pages are consistent throughout your development.

**Easy to use** — Bootstrap is very easy to use. Anybody with the basic working knowledge of HTML and CSS can start development with Bootstrap.

**Compatible with browsers** — Bootstrap is created with modern browsers in mind and it is compatible with all modern browsers such as Mozilla Firefox, Google Chrome, Safari, Internet Explorer, and Opera.

**Open Source** — And the best part is, it is completely free to download and use.

## **Content Management System**

A content management system or CMS is software that facilitates creating, editing, organizing, and publishing content. WordPress is a Content Management System that allows you to create and publish your content on the web. Although it is mostly used for web publishing, it can be used to manage content on an intranet, or in a single computer.

WordPress allows users to have full control over the files, documents, as well as the design and display of the content. You don't have to know a single line of code to publish content using WordPress. The beauty of a good content management system is to allow any user to create and manage their content without any technical know-how.

In the earlier days, an average user or a small company had to rely on static HTML sites because they could not afford a content management system which would cost hundreds of thousands of dollars. That problem is now solved. WordPress is open source and free for anyone to use.

WordPress is being used in all sort of creative ways. We have seen WordPress being used to power small business websites, blogs, large university websites, portfolios, real estate

property listing site, internal communication system for companies, web directories, movie databases, application infrastructure base, arcade sites, and basically anything else you can think of.

**A content management system may have the following functions:**

- ✓ Making publishing easier and more consistent with existing structure/design
- ✓ Allowing the input of data that classifies content (e.g. keywords) so that it can be searched for and retrieved
- ✓ Tracking changes to pages and, if necessary, allowing previous versions to be accessed.
- ✓ Make it easy to edit content
- ✓ Allow for collaborative work on content
- ✓ Integrated document management systems
- ✓ Workflow management: Allowing for parallel content development
- ✓ Provide extensions and plug-ins for increased functionality Etc.

# **SYSTEM DESIGN**

## **INTRODUCTION TO SYSTEM DESIGN**

The design phase is the life cycle phase in which the detailed design of the selected system in the study phase is accomplished. In the design phase, the technical specifications are prepared for the performance of all allocated tasks. It also includes the construction of programs and program testing. In the design phase, the first step is to determine the output is to be produced and in what format. Second, input data and master files have to be designed to meet the requirements of proposed output. The system analyst has to define the methods of capturing and input programs and format of the output and its use by the users.

## **SYSTEM FLOW CHART**

A graphic representation of a system showing the overall flow of control in the processing at the job level; specifies what activities must be done to convert from a physical to logical model is known as a system flowchart. Thus it summarizes what operations are undertaken and where and when they take place. Normally in a system flowchart input from outside are shown to the left and outputs to the right. Symbols representing the operations undertaken and the documents used are then placed in the appropriate places which gives a general flow of data from top to bottom and left to right. Arrows are used on the connecting lines to indicate the logical flow or sequence where the flow is not in the standard direction. No interaction is implied by crossing lines. Decisions which lead to different actions can also be shown

## **DATA FLOW DAGRAM**

A data flow diagram is graphic representation of a system that shows data flows to, from and within the system, processing functions that change the date in some manner, and the storage of this data. They are networks of related system function that indicated form where information is revived and to where it is sent. An external entity is the originator or receiver of data or information.

A data store symbol portraits a file or database in which data resides. A process is depicted by a circle sometimes it is called a bubble or transform. Process portraits the transformation of the content of status of data

## **DATABASE DESIGN**

This activity deals with the design of the physical database. The designer begins to concentrate on file design or how data should be organized around user requirements. How data are organized depends on the data and response requirements that determine hardware configurations.

An integrated approach to file design is the database. The general theme is to handle information as an integrated whole, with a minimum of redundancy and improved performance, type and size of data structure used. The objectives of data base are accuracy and integrity, privacy and security of data etc.

## **CODE DESIGN**

Codes can provide brief identification of data items and replace longer descriptions that would be more awkward to store and to manipulate.

## **INPUT DESIGN**

Input design is processing of converting the user oriented description of the inputs of the system. The goal of designing input data to make data entry as easy logical and free from errors as possible. In entering data, operates need to know the following.

1. The allocated space for each field.
2. Field sequence which must match that in the source document.
3. The format in which data fields are entered for example, filling out the data field is required through the edited format mm/dd/ yy

When we approach input data design, we design source documents that capture the data and then selected the media use to enter them into the computer. There are different ways in which data can be introduced into the system such as

- a. The data is converted into a machine sensible form by some realistic source document and types in the relevant items using a keyboard connected to the system.
- b. The document can be read directly by a machine and this converts information held in the human sensible form into a machine readable form without need for human investigations.
- c. Data entered into a system through a keyboard. This is done interactively by the person using the system.
- d. Data is presented in a form suitable to computer as a result of some of the processing.
- e. The data entry in the system has been designed so as to make to user friendly and also to incorporate certain validation checks.

The field name must be documented. The field name must be known to data entry operator or users so that the data entry will not exceed the allocated space. Our system contains the following inputs.

## **OUTPUT DESIGN**

The primary consideration in the design of all output is the information requirement and other objective of the users. It is the most important and direct source of information to the user. A major form of output is a hard copy. Print out should be designed around the output requirements of the user. Each output should be given a specific name or title. The output data is displayed on the visual display unit and output can be redirected to printers and or sorted in a file for later use.

Here, in this system, program is designed so as to generate a number of relevant outputs displayed in various kinds of user-defined tables in an easily readable and comprehensive manner which can be readily read and understood by the user. So no further attempt has been

made to generate reports which of course could have been easily implemented into the system.

## **PROCEDURE DESIGN**

When program become very long, they are divided into smaller programs or modules. These smaller programs can be written, tested and debugged separately. This technique of programming is known as modular programming. The advantages of modular programming are.

1. It is easy to write, test and debug a module.
2. Generally the modules of common nature are prepared, which can be used at many places.
3. The programmer can use the previously written programs.
4. If a change is to be made, it is made in the particular module; the entire program is not affected.

## **DATA FLOW DIAGRAM**

The Data Flow Diagram (DFD) is the graphical representation of the processes and the flow of data among them. A data flow diagram illustrates the processes, data stores, external entities and the connecting data flows in a system. It is a common practice to draw a context-level Data Flow Diagram first which shows the interaction between the system and outside entities.

A data flow diagram is a graphical tool used to describe and analyze movement of data through a system. They are the central tool and forms the basis from which the other components are developed. The transformation of data from input to output may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual

implements and movements of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagram. Each component in a DFD is labelled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD's is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process. Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this lead to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

## **DFD SYMBOLS**

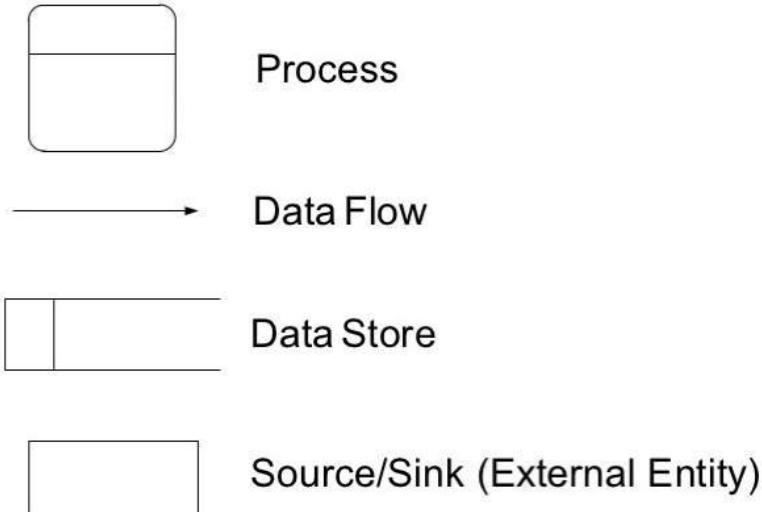
In the DFD, there are four symbols

A square defines a source (originator) or destination of system data.

An arrow defines data flow. It is the pipeline through which the information flows.

A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.

An open rectangle is a data store, data at rest or a temporary repository of data.



## CONSTRUCTING A DFD

There are several rules of thumb that are used in drawing DFD's. They are

- ✓ Process should be named and numbered for an easy reference. Each name should be representative of the process.
- ✓ The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD, it is marked with a short diagonal.
- ✓ When a process is exploded into lower level details, they are numbered.

## RULES GOVERNING THE DFD's

### PROCESS

- ✓ No process can have only outputs.
- ✓ No process can have only inputs. If an object has only inputs then it must be a sink.
- ✓ A process has a verb phrase label.

### DATA STORE

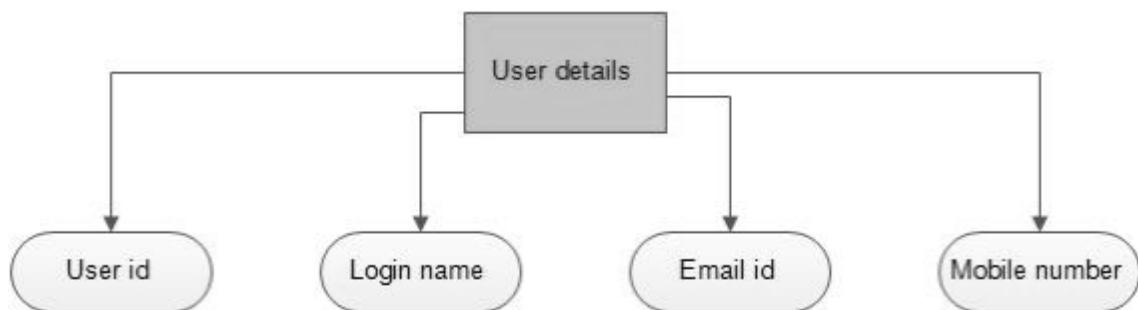
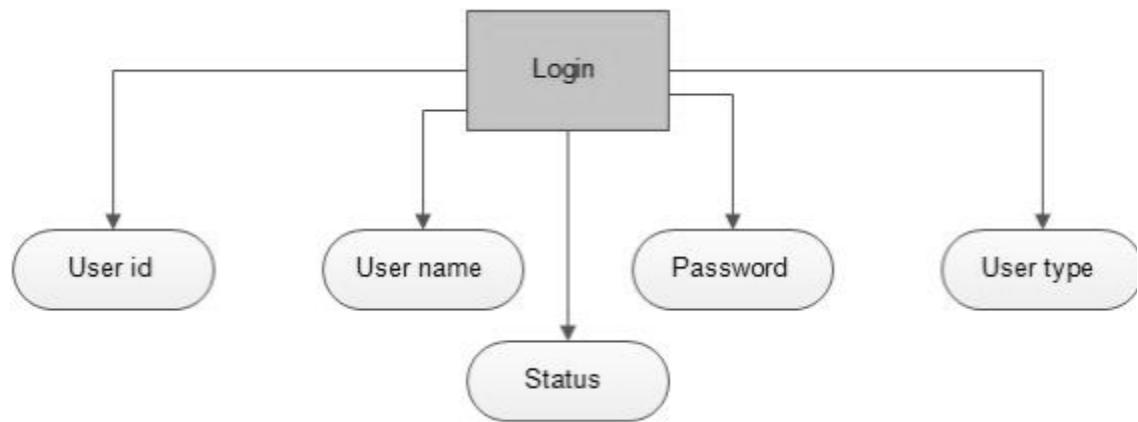
- ✓ Data cannot move directly from one data store to another data store, a process must move data.
- ✓ Data cannot move directly from an outside source to a data store. Data must be moved by a process that receives data from the source and places the data into the data store.
- ✓ Data cannot move directly to an outside sink from a data store. Data must be moved by a process.
- ✓ A data store has a noun phrase label.

## SOURCE OR SINK

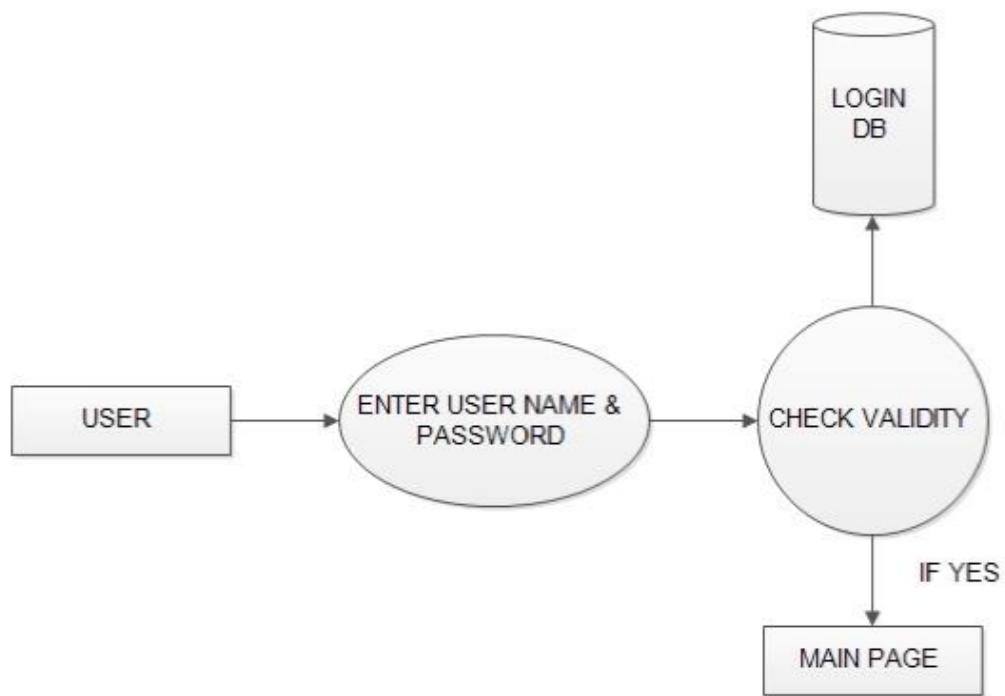
- ✓ Data cannot move directly from a source to a sink. It must be moved by a process.
- ✓ A source or sink has a noun phrase label.

## DATA FLOW

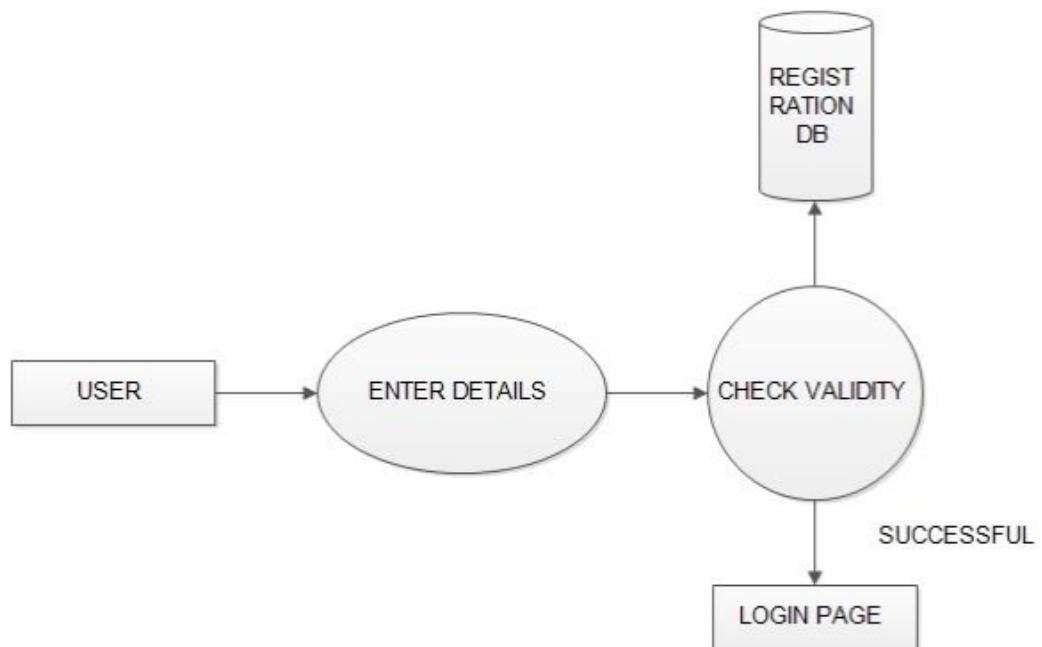
- ✓ A data flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update.
- ✓ A fork in a data flow means that exactly the same data goes from a common location to two or more different processes, data stores, or source/sinks.
- ✓ A join in a data flow means that exactly the same data come from any of two or more different processes, data stores, or source/sinks to a common location
- ✓ A data flow cannot go directly back to the same process it leaves.
- ✓ A data flow to a data store means update.
- ✓ A data flow from a data store means retrieve or use.
- ✓ A data flow has a noun phase label.



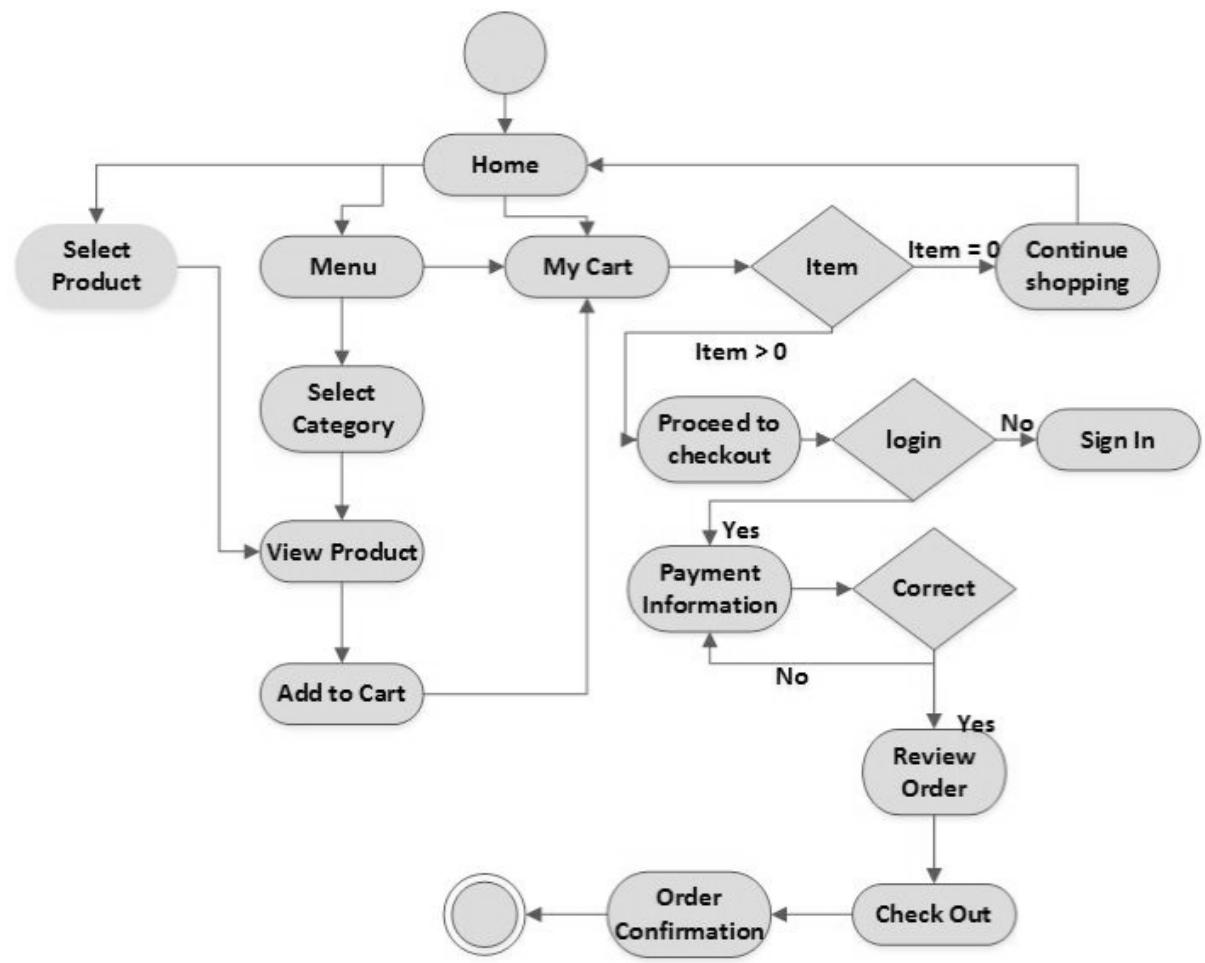
## DFD



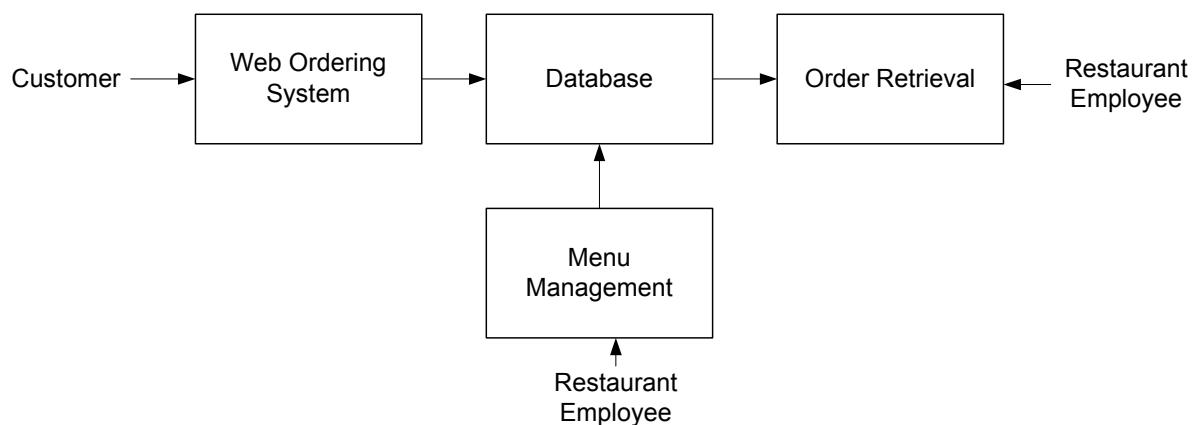
**REGISTRATION DFD**



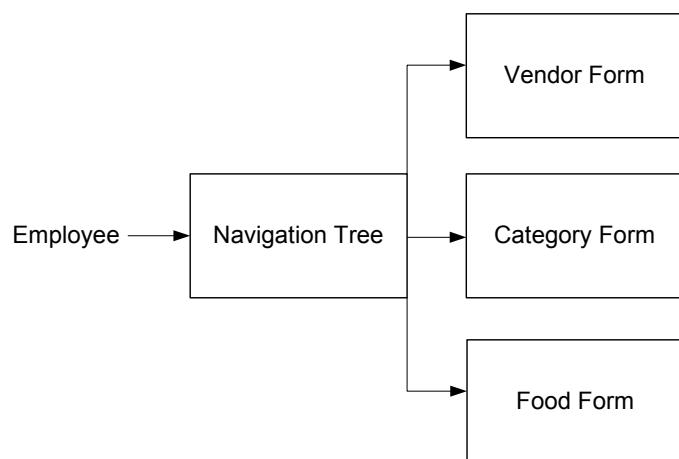
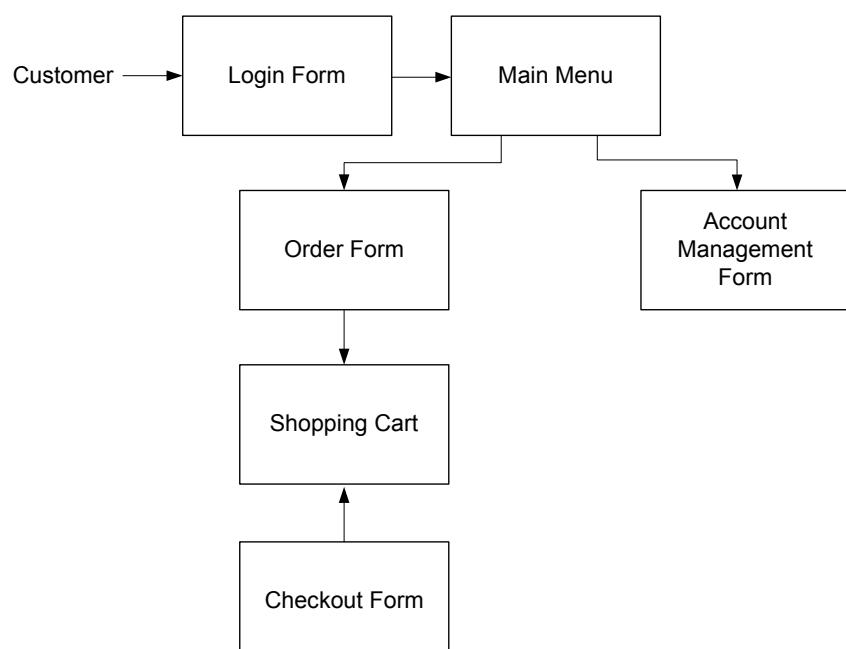
**Activity Diagram**



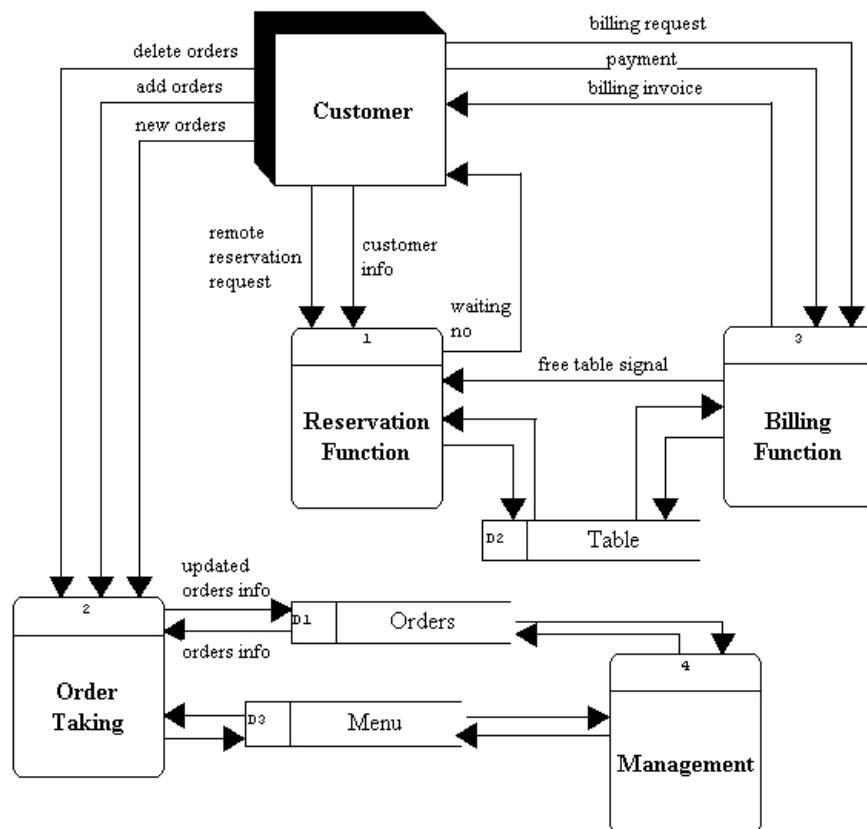
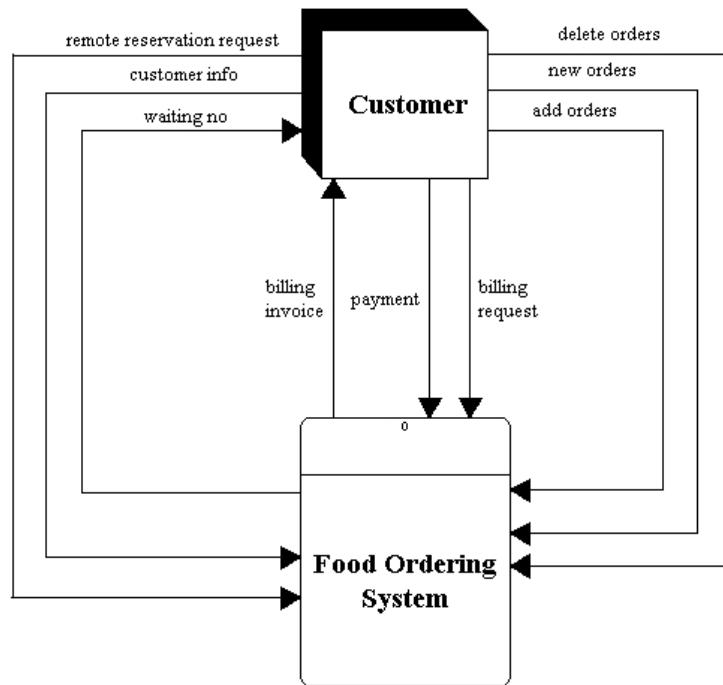
## DFD

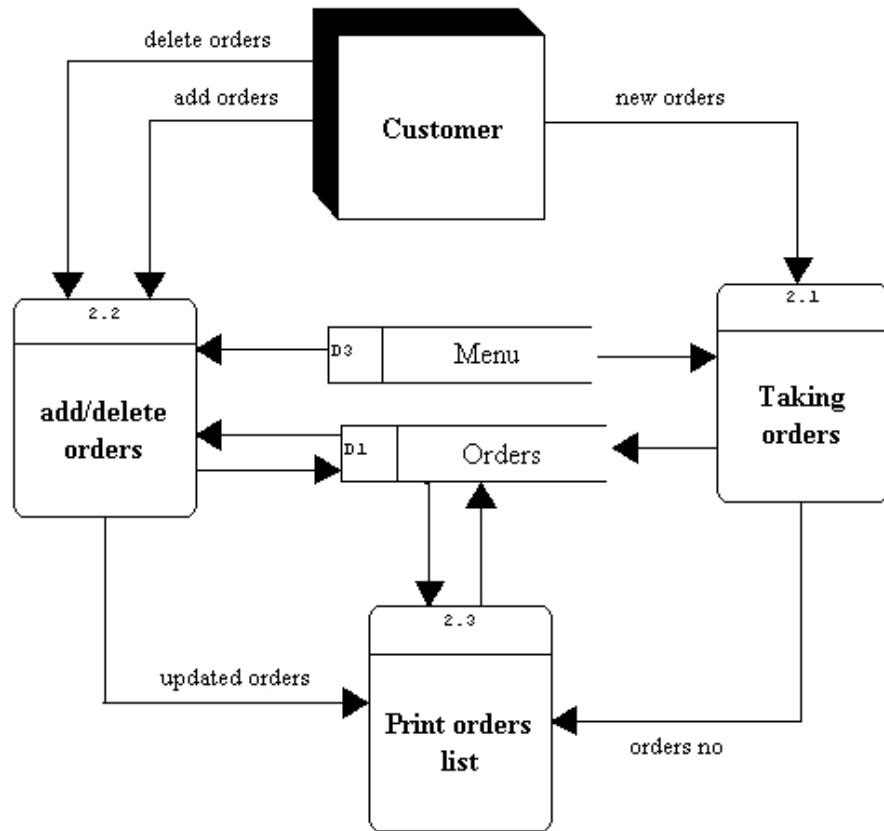


## DFD



## Content Diagram





## E-R Diagrams

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.

It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.

In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

### ***Connectivity and Cardinality***

The basic types of connectivity for relations are: one-to-one, one-to-many, and many-to-many. A *one-to-one* (1:1) relationship is when at most one instance of entity A is associated with one instance of entity B. For example, "employees in the company are each assigned their own office. For each employee there exists a unique office and for each office there exists a unique employee.

A *one-to-many* (1:N) relationships is when for one instance of entity A, there are zero, one, or many instances of entity B, but for one instance of entity B, there is only one instance of entity A. An example of a 1:N relationships is

A department has many employees

Each employee is assigned to one department

A *many-to-many* (M:N) relationship, sometimes called non-specific, is when for one instance of entity A, there are zero, one, or many instances of entity B and for one instance of entity B there is zero, one, or many instances of entity A. The connectivity of a relationship describes the mapping of associated

### ***ER Notation***

There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used, among the more common are Bachman, crow's foot, and IDEFIX.

All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

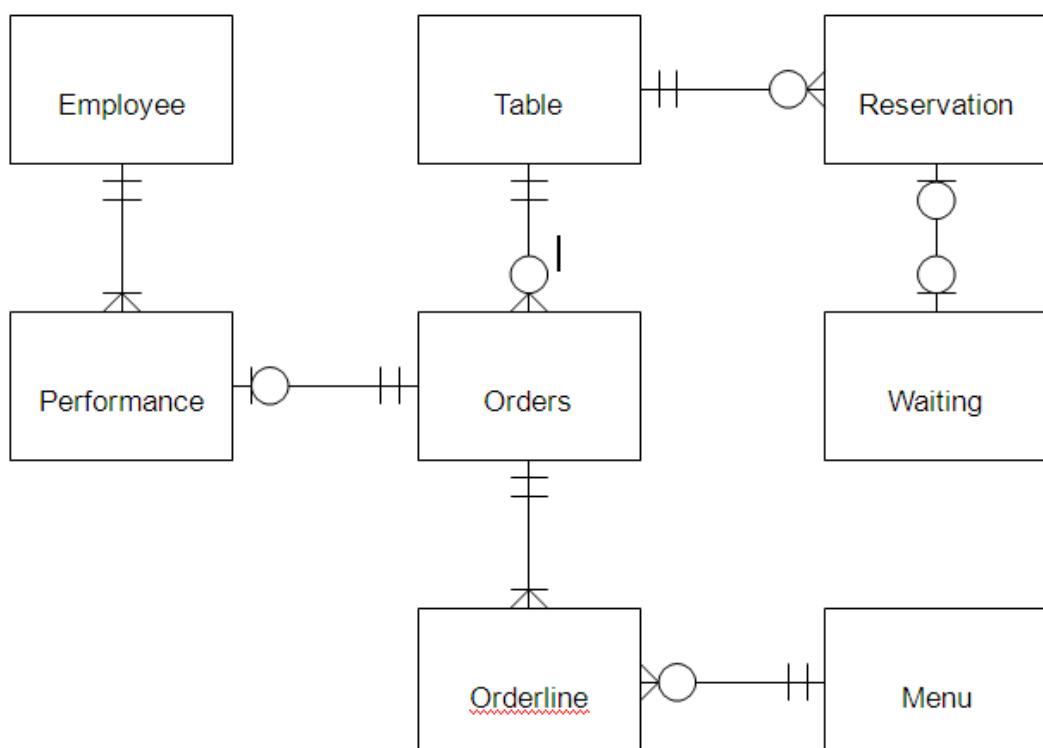
Entities are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.

Relationships are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs

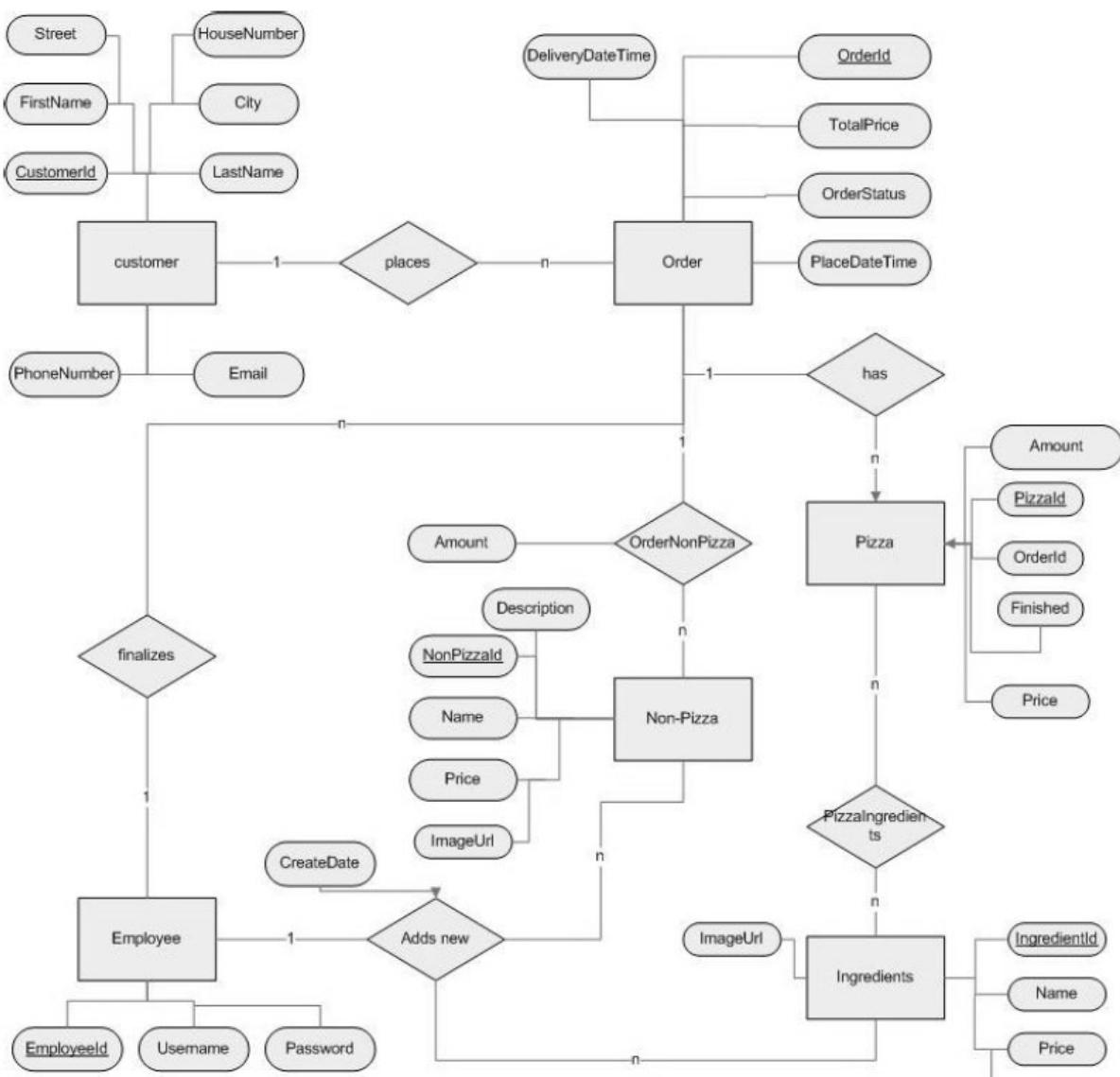
Attributes, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.

Cardinality of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.

Existence is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional



**ER Diagram**



**ER DIAGRAM**

## **DATABASE DESIGN**

The general theme behind a database is to handle information as an integrated whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. After designing input and output, the analyst must concentrate on database design or how data should be organized around user requirements. The general objective is to make information access, easy quick, inexpensive and flexible for other users.

**During database design the following objectives are concerned**

- ✓ Controlled Redundancy
- ✓ Data independence
- ✓ Accurate and integrating
- ✓ More information at low cost
- ✓ Recovery from failure
- ✓ Privacy and security
- ✓ Performance
- ✓ Ease of learning and use

localhost / 127.0.0.1 / or +

localhost/phpmyadmin/#PMAURL-105:db\_structure.php?db=online\_rest&table=&server=1&target=&token=46c4fc1b4375ad4ade19e590bd2b347c

**Structure** **SQL** **Search** **Query** **Export** **Import** **Operations** **Privileges** **Routines** **Events** **More**

**Table** **Action** **Rows** **Type** **Collation** **Size** **Overhead**

admin	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	2	InnoDB	latin1_swedish_ci	16 KiB	-
admin_codes	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	6	InnoDB	latin1_swedish_ci	16 KiB	-
dishes	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	7	InnoDB	latin1_swedish_ci	16 KiB	-
remark	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	11	InnoDB	latin1_swedish_ci	16 KiB	-
restaurant	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	6	InnoDB	latin1_swedish_ci	16 KiB	-
res_category	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	5	InnoDB	latin1_swedish_ci	16 KiB	-
users	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	6	InnoDB	latin1_swedish_ci	16 KiB	-
users_orders	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	16	InnoDB	latin1_swedish_ci	16 KiB	-
<b>8 tables</b>	<b>Sum</b>	<b>59</b>	<b>InnoDB</b>	<b>latin1_swedish_ci</b>	<b>128 KiB</b>	<b>0 B</b>

[Check All](#) [With selected...](#)

[Print view](#) [Data Dictionary](#)

**Create table**

Name:  Number of columns: 4 [Go](#)

Windows I'm Cortana, Ask me anything. 04:08 AM 27-04-2019

localhost / 127.0.0.1 / or +

localhost/phpmyadmin/#PMAURL-107:tbl\_structure.php?db=online\_rest&table=admin&server=1&target=&token=46c4fc1b4375ad4ade19e590bd2b347c

**Structure** **SQL** **Search** **Insert** **Export** **Import** **Privileges** **Operations** **Tracking** **More**

**#** **Name** **Type** **Collation** **Attributes** **Null** **Default** **Extra** **Action**

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	adm_id	int(222)	latin1_swedish_ci		No	None	AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a>
2	username	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a>
3	password	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a>
4	email	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a>
5	code	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a>
6	date	timestamp			on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	<a href="#">Change</a> <a href="#">Drop</a>

[Check All](#) [With selected:](#) [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#)

[Print view](#) [Relation view](#) [Propose table structure](#) [Track table](#) [Move columns](#)

[Add](#) 1 column(s)  At End of Table  At Beginning of Table  After adm\_id [Go](#)

[Indexes](#)

**Information**

Space usage		Row statistics	
Data	16 KiB	Format	Compact
Index	0 B	Collation	latin1_swedish_ci
Total	16 KiB	Next autoindex	9
		Creation	Apr 15, 2019 at 12:58 AM

Windows I'm Cortana, Ask me anything. 04:09 AM 27-04-2019

localhost / 127.0.0.1 / or +

localhost/phpmyadmin/#PMAURL-113:tbl\_structure.php?db=online\_rest&table=dishes&server=1&target=&ttoken=46c4fc1b4375ad4ade19e590bd2b347c

**phpMyAdmin**

Recent | Favorites

Filter databases by name or regex:

- online\_medicine
- online\_rest
  - New
  - admin
  - admin\_codes
  - dishes
  - remark
  - restaurant
  - res\_category
  - users
  - users\_orders
- orderingdb
- ourweb
- performance\_schema
- phpmyadmin
- pms
- postoffice
- prisonpro
- project
- projects
- property
- reyansdata

Server: 127.0.0.1 » Database: online\_rest » Table: dishes

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | More

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	d_id	int(22)	latin1_swedish_ci	No	None	AUTO_INCREMENT		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">More</a>
2	rs_id	int(22)	latin1_swedish_ci	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">More</a>
3	title	varchar(222)	latin1_swedish_ci	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">More</a>
4	slogan	varchar(222)	latin1_swedish_ci	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">More</a>
5	price	decimal(10,2)	latin1_swedish_ci	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">More</a>
6	img	varchar(222)	latin1_swedish_ci	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">More</a>

Check All With selected: Browse Change Drop Primary Unique Index

Print view | Relation view | Propose table structure | Track table | Move columns

Add 1 column(s) At End of Table At Beginning of Table After d\_id Go

+ Indexes

Information

Space usage	Row statistics
Data 16 KIB	Format Compact
Index 0 B	Collation latin1_swedish_ci
Total 16 KIB	Next autoindex 18
Creation Apr 15, 2019 at 12:58 AM	

04:09 AM 27-04-2019

localhost / 127.0.0.1 / or +

localhost/phpmyadmin/#PMAURL-113:tbl\_structure.php?db=online\_rest&table=remark&server=1&target=&ttoken=46c4fc1b4375ad4ade19e590bd2b347c

**phpMyAdmin**

Recent | Favorites

Filter databases by name or regex:

- online\_medicine
- online\_rest
  - New
  - admin
  - admin\_codes
  - dishes
  - remark
  - restaurant
  - res\_category
  - users
  - users\_orders
- orderingdb
- ourweb
- performance\_schema
- phpmyadmin
- pms
- postoffice
- prisonpro
- project
- projects
- property
- reyansdata

Server: 127.0.0.1 » Database: online\_rest » Table: remark

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | More

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)	latin1_swedish_ci	No	None	AUTO_INCREMENT		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">More</a>
2	frm_id	int(11)	latin1_swedish_ci	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">More</a>
3	status	varchar(255)	latin1_swedish_ci	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">More</a>
4	remark	mediumtext	latin1_swedish_ci	No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">More</a>
5	remarkDate	timestamp	latin1_swedish_ci	No	CURRENT_TIMESTAMP			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">More</a>

Check All With selected: Browse Change Drop Primary Unique Index

Print view | Relation view | Propose table structure | Track table | Move columns

Add 1 column(s) At End of Table At Beginning of Table After id Go

+ Indexes

Information

Space usage	Row statistics
Data 16 KIB	Format Compact
Index 0 B	Collation latin1_swedish_ci
Total 16 KIB	Next autoindex 73
Creation Apr 15, 2019 at 12:58 AM	

04:09 AM 27-04-2019

localhost / 127.0.0.1 / or +

localhost/phpmyadmin/#PMAURL-115:tbl\_structure.php?db=online\_rest&table=restaurant&server=1&target=&token=46c4fc1b4375ad4ade19e590bd2b347c

**phpMyAdmin**

Server: 127.0.0.1 » Database: online\_rest » Table: restaurant

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	rs_id	int(222)			No	None	AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	c_id	int(222)			No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
3	title	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
4	email	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
5	phone	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
6	url	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
7	o_hr	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
8	c_hr	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
9	o_days	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
10	address	text	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
11	image	text	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
12	date	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Check All    With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#)

[Print view](#) [Relation view](#) [Propose table structure](#) [Track table](#) [Move columns](#)

Add 1 column(s)  At End of Table  At Beginning of Table  After rs\_id [Go](#)

+ Indexes

Information

I'm Cortana. Ask me anything. 04:09 AM 27-04-2019

localhost / 127.0.0.1 / or +

localhost/phpmyadmin/#PMAURL-117:tbl\_structure.php?db=online\_rest&table=res\_category&server=1&target=&token=46c4fc1b4375ad4ade19e590bd2b347c

**phpMyAdmin**

Server: 127.0.0.1 » Database: online\_rest » Table: res\_category

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	c_id	int(222)			No	None	AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	c_name	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
3	date	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Check All    With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#)

[Print view](#) [Relation view](#) [Propose table structure](#) [Track table](#) [Move columns](#)

Add 1 column(s)  At End of Table  At Beginning of Table  After c\_id [Go](#)

+ Indexes

Information

Space usage		Row statistics	
Data	16 KiB	Format	Compact
Index	0 B	Collation	latin1_swedish_ci
Total	16 KiB	Next autoindex	10
		Creation	Apr 15, 2019 at 12:58 AM

I'm Cortana. Ask me anything. 04:09 AM 27-04-2019

The screenshot shows the phpMyAdmin interface for the 'online\_rest' database. The left sidebar lists databases and tables, with 'users' selected. The main area displays the structure of the 'users' table, which has 10 columns:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	u_id	int(22)			No	None	AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a>
2	username	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a>
3	f_name	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a>
4	l_name	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a>
5	email	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a>
6	phone	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a>
7	password	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a>
8	address	text	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a>
9	status	int(22)			No	1		<a href="#">Change</a> <a href="#">Drop</a>
10	date	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	<a href="#">Change</a> <a href="#">Drop</a>

Below the table structure, there are buttons for 'Add', 'Check All', and 'With selected'. The 'Information' panel shows 'Space usage' (Data: 16 KIB, Index: 0 B) and 'Row statistics' (Format: Compact). The status bar at the bottom right shows '04:10 AM 27-04-2019'.

The screenshot shows the phpMyAdmin interface for the 'online\_rest' database. The left sidebar lists databases and tables, with 'users\_orders' selected. The main area displays the structure of the 'users\_orders' table, which has 7 columns: o\_id, u\_id, title, quantity, price, status, and date. The 'status' column is defined as varchar(222) with a default value of 'NULL'. The 'date' column is defined as timestamp with a default value of CURRENT\_TIMESTAMP and an update value of CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP. Below the table structure, there are sections for 'Information', 'Space usage', and 'Row statistics'. The 'Space usage' section shows Data at 16 KiB, Index at 0 B, and Total at 16 KiB. The 'Row statistics' section shows Format as Compact, Collation as latin1\_swedish\_ci, Next autoindex as 54, and Creation as Apr 15, 2019 at 12:58 AM.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	<b>o_id</b>	int(222)			No	None	AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	<b>u_id</b>	int(222)			No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
3	<b>title</b>	varchar(222)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
4	<b>quantity</b>	int(222)			No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
5	<b>price</b>	decimal(10,2)			No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
6	<b>status</b>	varchar(222)	latin1_swedish_ci		Yes	NULL		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
7	<b>date</b>	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

[Print view](#) [Relation view](#) [Propose table structure](#) [Track table](#) [Move columns](#)

Add 1 column(s)  At End of Table  At Beginning of Table  After **o\_id** [Go](#)

+ Indexes

Information

Space usage		Row statistics	
Data	16 KiB	Format	Compact
Index	0 B	Collation	latin1_swedish_ci
Total	16 KiB	Next autoindex	54
		Creation	Apr 15, 2019 at 12:58 AM

# TESTING

## What is Software Testing?

Software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.

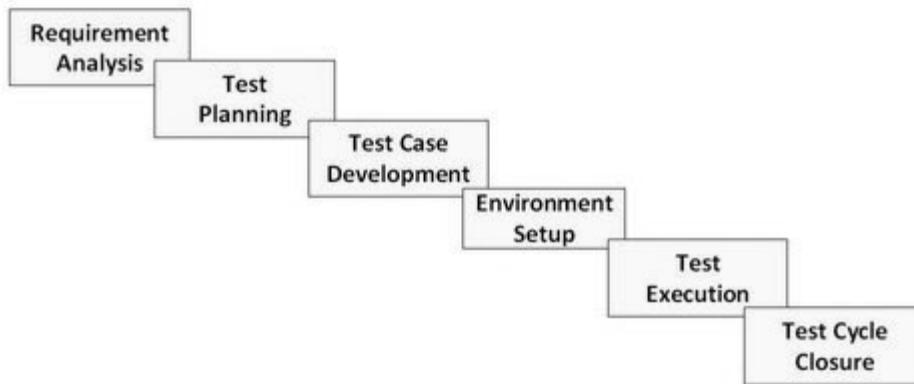
Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a white box and Black Box Testing.

## What is Software Testing Life Cycle (STLC)?

Software Testing Life Cycle (STLC) is defined as a sequence of activities conducted to perform Software Testing.

It consists of series of activities carried out methodologically to help certify your software product.

**Diagram - Different stages in Software Test Life Cycle**



Software testing is the process of executing a program with intension of finding errors in the code. It is a process of evolution of system or its parts by manual or automatic means to verify that it is satisfying specified or requirements or not. Generally, no system is perfect due to communication problems between user and developer, time constraints, or conceptual mistakes by developer. The purpose of system testing is to check and find out these errors or faults as early as possible so losses due to it can be saved.

Testing is the fundamental process of software success. Testing is not a distinct phase in system development life cycle but should be applicable throughout all phases i.e. design development and maintenance phase. Testing is used to show incorrectness and considered to success when an error is detected.

## **OBJECTIVES OF SOFTWARE TESTING**

The software testing is usually performed for the following objectives

### **Software Quality Improvement**

The computer and the software are mainly used for complex and critical applications and a bug or fault in software causes severe losses. So a great consideration is required for checking for quality of software.

### **Verification and Validation**

Verification means to test that we are building the product in right way .i.e. are we using the correct procedure for the development of software so that it can meet the user requirements.

Validation means to check whether we are building the right product or not.

### **Software Reliability Estimation**

The objective is to discover the residual designing errors before delivery to the customer. The failure data during process are taken down in order to estimate the software reliability.

## **PRINCIPLES OF SOFTWARE TESTING**

Software testing is an extremely creative and challenging task. Some important principles of software testing are as given:-

- ✓ All tests should be traceable to customer requirements.
- ✓ Testing time and resources should be limited i.e. avoid redundant testing.
- ✓ It is impossible to test everything.
- ✓ Use effective resources to test.
- ✓ Test should be planned long before testing begins i.e. after requirement phase.

- ✓ Test for invalid and unexpected input conditions as well as valid conditions.
- ✓ Testing should begin in “in the small” and progress towards testing “in the large”.
- ✓ For the most effective testing should be conducted by an independent party.
- ✓ Document test cases and test results.
- ✓ Examining what the software not doing which it expected to do and also checking what it is doing that was not expected to do.

## **STRATEGY FOR SOFTWARE TESTING**

**Different levels of testing are used in the test process; each level of testing aims to test different aspects of the system.**

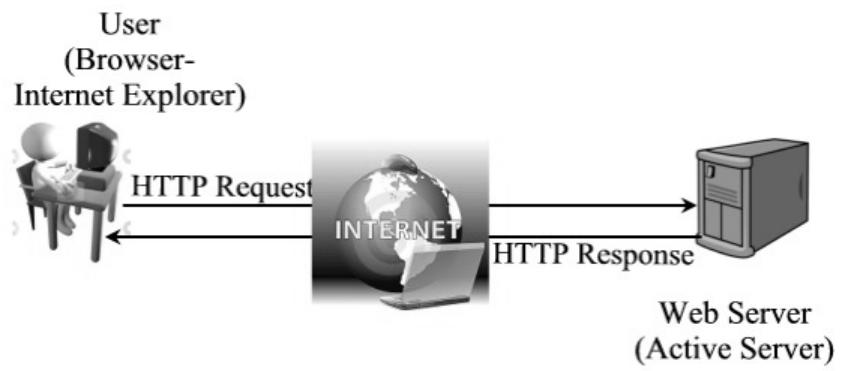
The First Level is **unit testing**. In this testing, individual components are tested to ensure that they operate correctly. It focuses on verification efforts.

The Second Level is **integration testing**. It is a systematic technique for constructing the program structure. In this testing, many tested modules are combined into the subsystems which are then tested. The goal here is to see if the modules can be integrated properly.

The Third Level is **system testing**? System testing is actually a series of different tests whose primary purpose is to fully exercise computer based system.

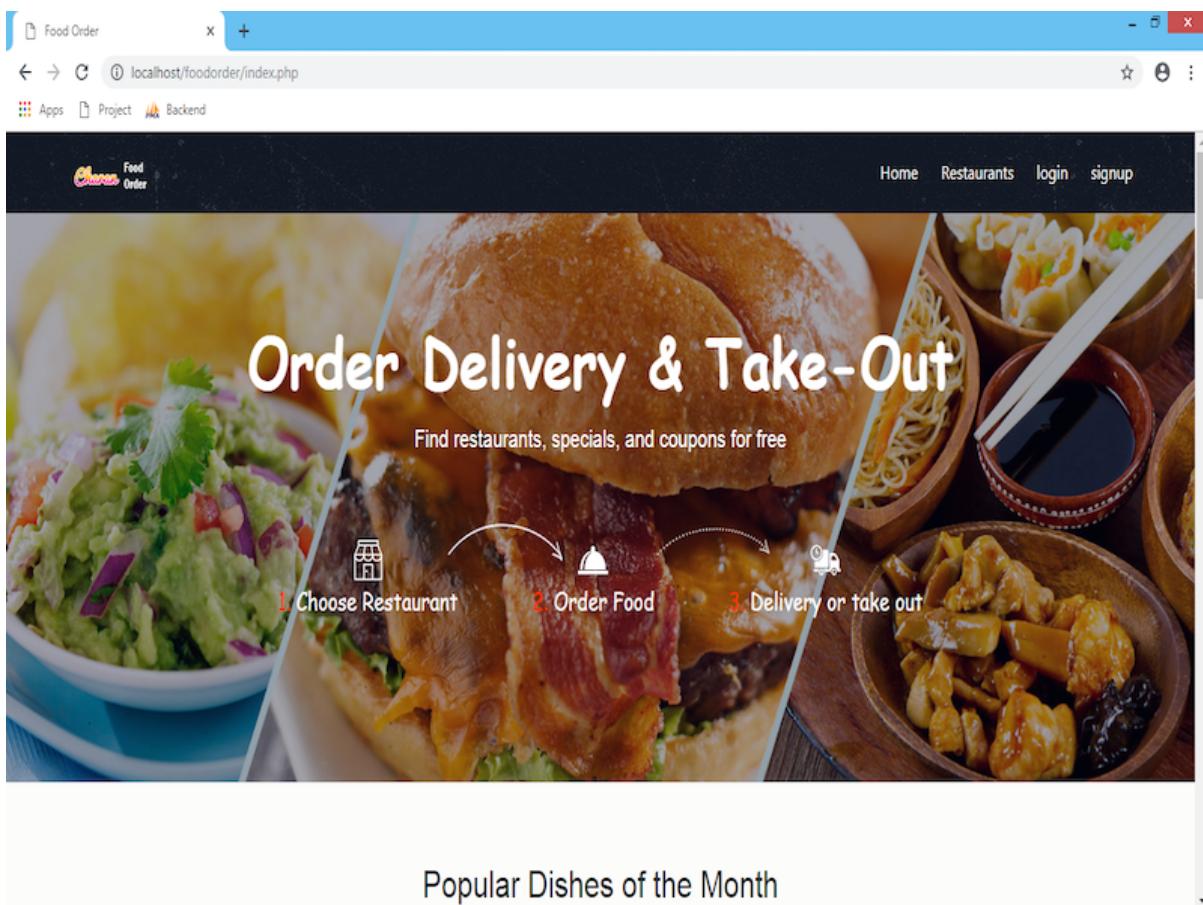
## **IMPLEMENTATION**

A customer’s web browser issues an HTTP request from the Contact page. On clicking the button, the content of the fields are posted from the customer’s browser as a request to the web server. On receiving the request, the web server retrieves the file, Contacts.asp from its disk or memory and passes it to the php, php.dll, after processing the file php sends the HTML page to the server.



## SCREEN SHOTS

### Home Page :



## Featured Restaurants :

The screenshot shows a web browser window with the title 'Food Order' at the top. The address bar indicates the URL is 'localhost/foodorder/index.php'. The browser interface includes standard controls like back, forward, and search. Below the header, there's a navigation bar with links for 'Home', 'Restaurants', 'login', and 'signup'. The main content area is titled 'Featured restaurants' and features a grid of six restaurant cards. Each card includes a small thumbnail image, the restaurant name, its address, delivery information (min Rs 100, 30 min delivery), and a star rating (4.5 stars based on 122 reviews). The cards are arranged in two rows of three.

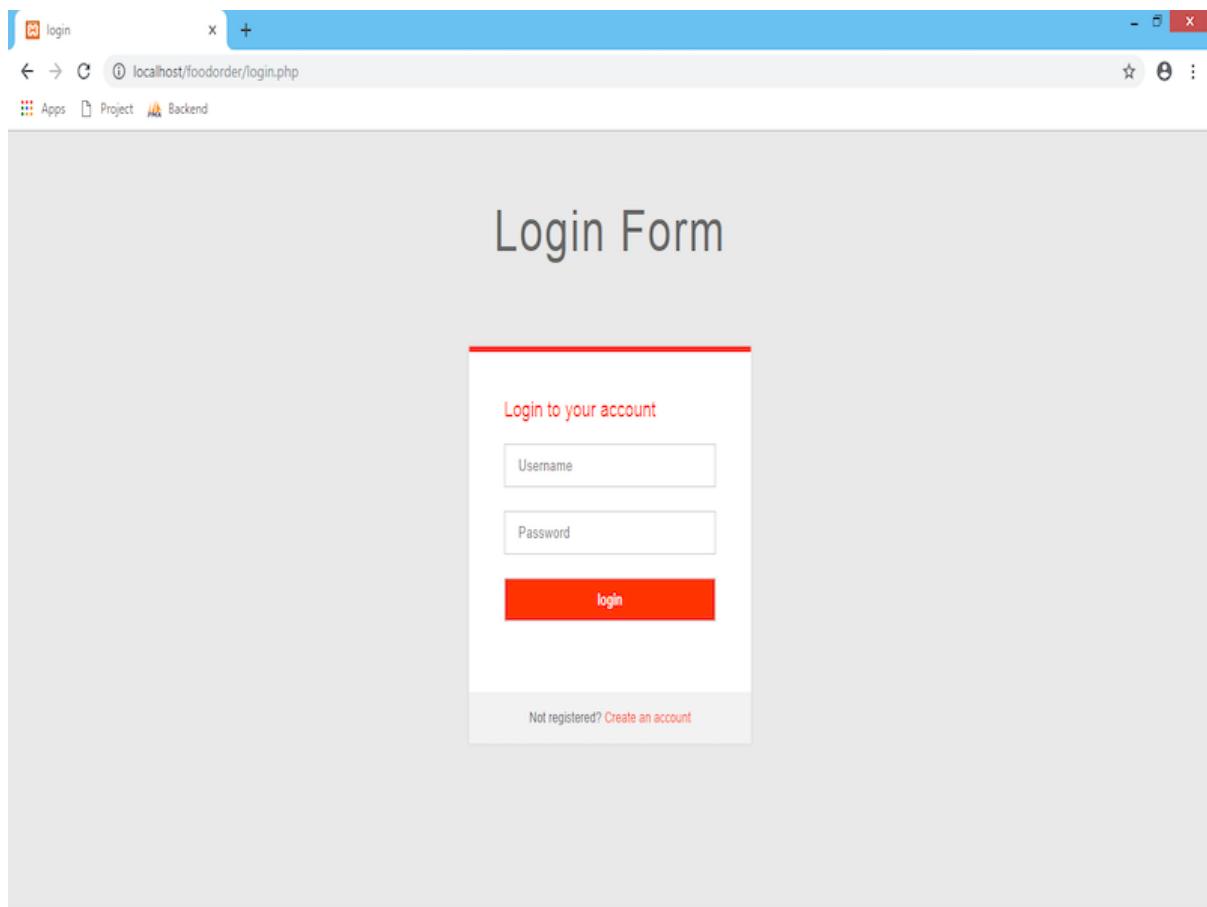
Restaurant	Address	Delivery Info	Rating
Hari Burger	Palace, natwar jalandhar	✓ Min Rs 100   30 min	★★★★☆ (122)
The Great Kabab Factory	Radisson Blu Plaza Hotel, Delhi Airport, NH-8, New Delhi, 110037	✓ Min Rs 100   30 min	★★★★☆ (122)
Aarkay Vaishno Dhaba	Bhargav Nagar, Jalandhar - Nakodar Rd, Jalandhar, Punjab 144003	✓ Min Rs 100   30 min	★★★★☆ (122)
Martini	399 L Near Apple Showroom, Model Town,	✓ Min Rs 100   30 min	★★★★☆ (122)
hudson	Opposite Lovely Sweets, Nakodar Road, Jalandhar, Punjab 144001	✓ Min Rs 100   30 min	★★★★☆ (122)
kriyana store	near kalu gali hotel india what everrrr.	✓ Min Rs 100   30 min	★★★★☆ (122)

## Restaurant List :

The screenshot shows a web browser window for a food ordering system. The URL in the address bar is `localhost/foodorder/restaurants.php`. The page has a header with the logo "Chores Food Order" and navigation links for "Home", "Restaurants", "login", and "signup". Below the header, there are three numbered steps: 1. Choose Restaurant, 2. Pick Your favorite food, and 3. Order and Pay online. A background image of various dishes is visible. On the left, there is a "Popular tags" section with a placeholder image. The main content area displays three restaurant cards:

Restaurant Name	Address	Rating	Reviews	Action
Hari Burger	Palace, natwar jalandhar ...	★★★★★	245 Reviews	<a href="#">View Menu</a>
The Great Kabab Factory	Radisson Blu Plaza Hotel, Delhi Airport, NH-8, New Delhi, 110037	★★★★★	245 Reviews	<a href="#">View Menu</a>
Aarkay Vaishno Dhaba	Bhargav Nagar, Jalandhar - Nakodar Rd, Jalandhar, Punjab	★★★★★	245 Reviews	

## Login Form :



## Registration Page :

The screenshot shows a web browser window with the title "Food Order". The address bar displays "localhost/foodorder/registration.php". The page header includes a logo for "Choco Food Order" and navigation links for "Home", "Restaurants", "login", and "signup". The main content area is a registration form:

User-Name	
UserName	<input type="text"/>
First Name	Last Name
First Name	<input type="text"/>
Email address	Phone number
Enter email	<input type="text"/>
We'll never share your email with anyone else.	
Password	Repeat password
Password	<input type="text"/>
Delivery Address	
<input type="text"/>	

# CODING

## Home / Index Page

```
<!DOCTYPE html>
<html lang="en">
<?php
include("connection/connect.php"); //include connection file
error_reporting(0); // using to hide undefined index errors
session_start(); //start temp session until logout/browser closed

?>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<!-- The above 3 meta tags *must* come first in the head; any other head content must
come *after* these tags --&gt;
&lt;meta name="description" content=""&gt;
&lt;meta name="author" content=""&gt;
&lt;link rel="icon" href="#"&gt;
&lt;title&gt;Food Order&lt;/title&gt;
<!-- Bootstrap core CSS --&gt;
&lt;link href="css/bootstrap.min.css" rel="stylesheet"&gt;
&lt;link href="css/font-awesome.min.css" rel="stylesheet"&gt;
&lt;link href="css/animstion.min.css" rel="stylesheet"&gt;
&lt;link href="css/animate.css" rel="stylesheet"&gt;
<!-- Custom styles for this template --&gt;
&lt;link href="css/style.css" rel="stylesheet"&gt; &lt;/head&gt;</pre>
```

```

<body class="home">
  <div class="site-wrapper animsition" data-animsition-in="fade-in" data-animsition-
out="fade-out">
    <!--header starts-->
    <header id="header" class="header-scroll top-header headrom">
      <!-- .navbar -->
      <nav class="navbar navbar-dark">
        <div class="container">
          <button class="navbar-toggler hidden-lg-up" type="button" data-
toggle="collapse" data-target="#mainNavbarCollapse">&#9776;</button>
          <a class="navbar-brand" href="index.php"></a>
          <div class="collapse navbar-toggleable-md float-lg-right"-
id="mainNavbarCollapse">
            <ul class="nav navbar-nav">
              <li class="nav-item"><a class="nav-link active" href="index.php">Home-
<span class="sr-only">(current)</span></a></li>
              <li class="nav-item"><a class="nav-link active"-
href="restaurants.php">Restaurants <span class="sr-only"></span></a></li>

```

(adsbygoogle = window.adsbygoogle || []).push({});

```

              <?php
                if(empty($_SESSION["user_id"])) // if user is
                  not login
                {
                  echo '<li class="nav-item"><a
                    href="login.php" class="nav-link active">login</a></li>
                    <li class="nav-item"><a
                    href="registration.php" class="nav-link active">signup</a></li>';
                }
                else

```

```

    {
        //if user is login

        echo '<li class="nav-
item"><a href="your_orders.php" class="nav-link active">your orders</a> </li>';
        echo '<li class="nav-
item"><a href="logout.php" class="nav-link active">logout</a> </li>';
    }

?>

</ul>

</div>
</div>
</nav>
<!-- /.navbar -->
</header>
<!-- banner part starts -->
<section class="hero bg-image" data-image-src="images/img/main.jpeg">
    <div class="hero-inner">
        <div class="container text-center hero-text font-white">
            <h1>Order Delivery & Take-Out </h1>
            <h5 class="font-white space-xs">Find restaurants, specials, and coupons for
            free</h5>
            <div class="banner-form">
                <form class="form-inline">
                    </form>
                </div>

```

```

<div class="steps">
  <div class="step-item step1">
    <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 483 483"
      width="512" height="512">
      <g fill="#FFF">
        <path d="M467.006 177.92c-.055-1.573-.
          469-3.321-1.233-4.755L407.006 62.877V10.5c0-5.799-4.701-10.5-10.5-10.5h-310c-5.799
          0-10.5 4.701-10.5 10.5v52.375L17.228 173.164a10.476 10.476 0 0 0-1.22 4.938h-.
          014V472.5c0 5.799 4.701 10.5 10.5 10.5h430.012c5.799 0 10.5-4.701
          10.5-10.5V177.92zM282.379 76l18.007 91.602H182.583L200.445 76h81.934zm19.391
          112.602c-4.964 29.003-30.096 51.143-60.281 51.143-30.173
          0-55.295-22.139-60.258-51.143H301.77zm143.331 0c-4.96 29.003-30.075 51.143-60.237
          51.143-30.185 0-55.317-22.139-60.281-51.143h120.518zm-123.314-21L303.78
          76h86.423l48.81 91.602H321.787zM97.006 55V21h289v34h-289zm-4.198
          21h86.243l-17.863 91.602h-117.2L92.808 76zm65.582 112.602c-5.028 28.475-30.113
          50.19-60.229 50.19s-55.201-21.715-60.23-50.19H158.39zM300
          462H183V306h117v156zm21 0V295.5c0-5.799-4.701-10.5-10.5-10.5h-138c-5.799 0-10.5
          4.701-10.5 10.5V462H36.994V232.743a82.558 82.558 0 0 0 3.101 3.255c15.485 15.344
          36.106 23.794 58.065 23.794s42.58-8.45 58.065-23.794a81.625 81.625 0 0 0
          13.525-17.672c14.067 25.281 40.944 42.418 71.737 42.418 30.752 0 57.597-17.081
          71.688-42.294 14.091 25.213 40.936 42.294 71.688 42.294 24.262 0 46.092-10.645
          61.143-27.528V462H321z"></path>
      <path d="M202.494 386h22c5.799 0 10.5-4.701
          10.5-10.5s-4.701-10.5-10.5-10.5h-22c-5.799 0-10.5 4.701-10.5 10.5s4.701 10.5 10.5
          10.5z"></path>
    </g>
  </svg>
  <h4><span>1. </span>Choose Restaurant</h4> </div>
  <!-- end:Step -->
  <div class="step-item step2">

```

```

<svg xmlns="http://www.w3.org/2000/svg" width="512" height="512"
viewbox="0 0 390.721 390.721">
  <g fill="#FFF">
    <path d="M58.727 281.236c.32-5.217.657-10.457 1.319-15.709
1.261-12.525 3.974-25.05 6.733-37.296a543.51 543.51 0 0 1 5.449-17.997c2.463-5.729
4.868-11.433 7.25-17.01 5.438-10.898 11.491-21.07 18.724-29.593 1.737-2.19 3.427-4.328
5.095-6.46 1.912-1.894 3.805-3.747 5.676-5.588 3.863-3.509 7.221-7.273 11.107-10.091
7.686-5.711 14.529-11.137 21.477-14.506 6.698-3.724 12.455-6.982 17.631-8.812
10.125-4.084 15.883-6.141 15.883-6.141s-4.915 3.893-13.502 10.207c-4.449 2.917-9.114
7.488-14.721 12.147-5.803 4.461-11.107 10.84-17.358 16.992-3.149 3.114-5.588
7.064-8.551 10.684-1.452 1.83-2.928 3.712-4.427 5.6a1225.858 1225.858 0 0 1-3.84
6.286c-5.537 8.208-9.673 17.858-13.995 27.664-1.748 5.1-3.566 10.283-5.391
15.534a371.593 371.593 0 0 1-4.16 16.476c-2.266 11.271-4.502 22.761-5.438 34.612-6.68
4.287-1.022 8.633-1.383 12.979 94 .023 166.775.069
268.589.069.337-4.462.534-8.97.534-13.536 0-85.746-62.509-156.352-142.875-165.705
5.17-4.869 8.436-11.758 8.436-19.433-.023-14.692-11.921-26.612-26.631-26.612-14.715
0-26.652 11.92-26.652 26.642 0 7.668 3.265 14.558 8.464 19.426-80.396 9.353-142.869
79.96-142.869 165.706 0 4.543.168 9.027.5 13.467 9.935-.002 19.526-.002 28.926-.002zM0
291.135h380.721v33.59H0z" /> </g>
  </svg>
  <h4><span>2. </span>Order Food</h4> </div>
  <!-- end:Step -->
  <div class="step-item step3">
    <svg xmlns="http://www.w3.org/2000/svg" width="512" height="512"
viewbox="0 0 612.001 612">
      <path d="M604.131
440.17h-19.12V333.237c0-12.512-3.776-24.787-10.78-35.173l-47.92-70.975a62.99 62.99 0
0 0-52.169-27.698h-74.28c-8.734 0-15.737 7.082-15.737 15.738v225.043h-121.65c11.567
9.992 19.514 23.92 21.796 39.658H412.53c4.563-31.238 31.475-55.396 63.972-55.396
32.498 0 59.33 24.158 63.895 55.396h63.735c4.328 0 7.869-3.541 7.869-7.869V448.04c-
001-4.327-3.541-7.87-7.87-7.87zM525.76 312.227h-98.044a7.842 7.842 0 0

```

1-7.868-7.869v-54.372c0-4.328 3.541-7.869 7.868-7.869h59.724c2.597 0 4.957 1.259 6.452  
3.305l38.32 54.451c3.619 5.194-0.079 12.354-6.452 12.354zM476.502 440.17c-27.068  
0-48.943 21.953-48.943 49.021 0 26.99 21.875 48.943 48.943 48.943 26.989 0  
48.943-21.953 48.943-48.943 0-27.066-21.954-49.021-48.943-49.021zm0 73.495c-13.535  
0-24.472-11.016-24.472-24.471 0-13.535 10.937-24.473 24.472-24.473 13.533 0 24.472  
10.938 24.472 24.473 0 13.455-10.938 24.471-24.472 24.471zM68.434 440.17c-4.328  
0-7.869 3.543-7.869 7.869v23.922c0 4.328 3.541 7.869 7.869 7.869h87.971c2.282-15.738  
10.229-29.666 21.718-39.658H68.434v-.002zm151.864 0c-26.989 0-48.943 21.953-48.943  
49.021 0 26.99 21.954 48.943 48.943 48.943 27.068 0 48.943-21.953  
48.943-48.943.001-27.066-21.874-49.021-48.943-49.021zm0 73.495c-13.534  
0-24.471-11.016-24.471-24.471 0-13.535 10.937-24.473 24.471-24.473s24.472 10.938  
24.472 24.473c0 13.455-10.938 24.471-24.472 24.471zm117.716-363.06h-91.198c4.485  
13.298 6.846 27.54 6.846 42.255 0 74.28-60.431 134.711-134.711 134.711-13.535  
0-26.675-2.045-39.029-5.744v86.949c0 4.328 3.541 7.869 7.869 7.869h265.96c4.329 0  
7.869-3.541 7.869-7.869V174.211c-.001-13.062-10.545-23.606-23.606-23.606zM118.969  
73.866C53.264 73.866 0 127.129 0 192.834s53.264 118.969 118.969 118.969 118.97-53.264  
118.97-118.969-53.265-118.968-118.97-118.968zm0 210.864c-50.752  
0-91.896-41.143-91.896-91.896s41.144-91.896 91.896-91.896c50.753 0 91.896 41.144  
91.896 91.896 0 50.753-41.143 91.896-91.896 91.896zm35.097-72.488c-1.014 0-2.052-  
131-3.082-.407L112.641 201.5a11.808 11.808 0 0 1-8.729-11.396v-59.015c0-6.516  
5.287-11.803 11.803-11.803 6.516 0 11.803 5.287 11.803 11.803v49.971l29.614 7.983c6.294  
1.698 10.02 8.177 8.322 14.469-1.421 5.264-6.185 8.73-11.388 8.73z" fill="#FFF" /></svg>  

#### <span>3. </span>Delivery or take out</h4> </div>

<!-- end:Step -->

</div>

<!-- end:Steps -->

</div>

</div>

<!--end:Hero inner -->

</section>

<!-- banner part ends -->

```

<!-- Popular block starts -->
<section class="popular">
  <div class="container">
    <div class="title text-xs-center m-b-30">
      <h2>Popular Dishes of the Month</h2>
      <p class="lead">The easiest way to your favourite food</p>
    </div>
    <div class="row">

      <?php
        // fetch records from database to display popular
        first 3 dishes from table
        $query_res= mysqli_query($db,"select * from
dishes LIMIT 3");

      while($r=mysqli_fetch_array($query_res))
      {
        echo ' <div class="col-xs-12 col-
sm-6 col-md-4 food-item">

          <div class="food-item-wrap">

            <div class="figure-wrap bg-image" data-image-src="admin/Res_img/dishes/.
$r['img']."'>

```

```
<div class="distance"><i class="fa fa-pin"></i>1240m</div>

<div class="rating pull-left"> <i class="fa fa-star"></i> <i class="fa fa-star"></i> <i class="fa fa-star"></i> <i class="fa fa-star"></i> <i class="fa fa-star-o"></i>
</div>

<div class="review pull-right"><a href="#">198 reviews</a> </div>

</div>

<div class="content">

<h5><a href="dishes.php?res_id='.$r['rs_id'].'">'.$r['title'].'</a></h5>

<div class="product-name">'.$r['slogan'].'</div>

<div class="price-btn-block"> <span class="price">Rs'.$r['price'].'

```

}

?>

```
</div>
</div>
</section>
<!-- Popular block ends -->
<!-- How it works block starts -->
<section class="how-it-works">
  <div class="container">
    <div class="text-xs-center">
      <h2>Easy 3 Step Order</h2>
      <!-- 3 block sections starts -->
      <div class="row how-it-works-solution">
        <div class="col-xs-12 col-sm-12 col-md-4 how-it-works-steps white-txt
          coll">
          <div class="how-it-works-wrap">
            <div class="step step-1">
              <div class="icon" data-step="1">
```

```

<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 483 483"
width="512" height="512">
    <g fill="#FFF">
        <path d="M467.006 177.92c-.055-1.573-
469-3.321-1.233-4.755L407.006 62.877V10.5c0-5.799-4.701-10.5-10.5-10.5h-310c-5.799
0-10.5 4.701-10.5 10.5v52.375L17.228 173.164a10.476 10.476 0 0 0-1.22 4.938h-
014V472.5c0 5.799 4.701 10.5 10.5 10.5h430.012c5.799 0 10.5-4.701
10.5-10.5V177.92zM282.379 76l18.007 91.602H182.583L200.445 76h81.934zm19.391
112.602c-4.964 29.003-30.096 51.143-60.281 51.143-30.173
0-55.295-22.139-60.258-51.143H301.77zm143.331 0c-4.96 29.003-30.075 51.143-60.237
51.143-30.185 0-55.317-22.139-60.281-51.143h120.518zm-123.314-21L303.78
76h86.423l48.81 91.602H321.787zM97.006 55V21h289v34h-289zm-4.198
21h86.243l-17.863 91.602h-117.2L92.808 76zm65.582 112.602c-5.028 28.475-30.113
50.19-60.229 50.19s-55.201-21.715-60.23-50.19H158.39zM300
462H183V306h117v156zm21 0V295.5c0-5.799-4.701-10.5-10.5-10.5h-138c-5.799 0-10.5
4.701-10.5 10.5V462H36.994V232.743a82.558 82.558 0 0 0 3.101 3.255c15.485 15.344
36.106 23.794 58.065 23.794s42.58-8.45 58.065-23.794a81.625 81.625 0 0 0
13.525-17.672c14.067 25.281 40.944 42.418 71.737 42.418 30.752 0 57.597-17.081
71.688-42.294 14.091 25.213 40.936 42.294 71.688 42.294 24.262 0 46.092-10.645
61.143-27.528V462H321z" />
    <path d="M202.494 386h22c5.799 0 10.5-4.701
10.5-10.5s-4.701-10.5-10.5-10.5h-22c-5.799 0-10.5 4.701-10.5 10.5s4.701 10.5 10.5 10.5z" />
</g>
</svg>
</div>
<h3>Choose a restaurant</h3>
<p>We've got your covered with menus from over 345 delivery
restaurants online.</p>
</div>
</div>
</div>

```

```

<div class="col-xs-12 col-sm-12 col-md-4 how-it-works-steps white-txt

col2">

<div class="step step-2">
    <div class="icon" data-step="2">
        <svg xmlns="http://www.w3.org/2000/svg" width="512" height="512"
viewbox="0 0 380.721 380.721">
            <g fill="#FFF">
                <path d="M58.727 281.236c.32-5.217.657-10.457 1.319-15.709
1.261-12.525 3.974-25.05 6.733-37.296a543.51 543.51 0 0 1 5.449-17.997c2.463-5.729
4.868-11.433 7.25-17.01 5.438-10.898 11.491-21.07 18.724-29.593 1.737-2.19 3.427-4.328
5.095-6.46 1.912-1.894 3.805-3.747 5.676-5.588 3.863-3.509 7.221-7.273 11.107-10.091
7.686-5.711 14.529-11.137 21.477-14.506 6.698-3.724 12.455-6.982 17.631-8.812
10.125-4.084 15.883-6.141 15.883-6.141s-4.915 3.893-13.502 10.207c-4.449 2.917-9.114
7.488-14.721 12.147-5.803 4.461-11.107 10.84-17.358 16.992-3.149 3.114-5.588
7.064-8.551 10.684-1.452 1.83-2.928 3.712-4.427 5.6a1225.858 1225.858 0 0 1-3.84
6.286c-5.537 8.208-9.673 17.858-13.995 27.664-1.748 5.1-3.566 10.283-5.391
15.534a371.593 371.593 0 0 1-4.16 16.476c-2.266 11.271-4.502 22.761-5.438 34.612-.68
4.287-1.022 8.633-1.383 12.979 94 .023 166.775.069
268.589.069.337-4.462.534-8.97.534-13.536 0-85.746-62.509-156.352-142.875-165.705
5.17-4.869 8.436-11.758 8.436-19.433-.023-14.692-11.921-26.612-26.631-26.612-14.715
0-26.652 11.92-26.652 26.642 0 7.668 3.265 14.558 8.464 19.426-80.396 9.353-142.869
79.96-142.869 165.706 0 4.543.168 9.027.5 13.467 9.935-.002 19.526-.002 28.926-.002zM0
291.135h380.721v33.59H0z" /> </g>
        </svg>
    </div>
    <h3>Choose a tasty dish</h3>
    <p>We've got your covered with menus from over 345 delivery
restaurants online.</p>
</div>
</div>

```

```

<div class="col-xs-12 col-sm-12 col-md-4 how-it-works-steps white-txt
col3">
    <div class="step step-3">
        <div class="icon" data-step="3">
            <svg xmlns="http://www.w3.org/2000/svg" width="512" height="512"
viewbox="0 0 612.001 612">
                <path d="M604.131
440.17h-19.12V333.237c0-12.512-3.776-24.787-10.78-35.173l-47.92-70.975a62.99 62.99 0
0 0-52.169-27.698h-74.28c-8.734 0-15.737 7.082-15.737 15.738v225.043h-121.65c11.567
9.992 19.514 23.92 21.796 39.658H412.53c4.563-31.238 31.475-55.396 63.972-55.396
32.498 0 59.33 24.158 63.895 55.396h63.735c4.328 0 7.869-3.541 7.869-7.869V448.04c-.
001-4.327-3.541-7.87-7.87zM525.76 312.227h-98.044a7.842 7.842 0 0
1-7.868-7.869v-54.372c0-4.328 3.541-7.869 7.868-7.869h59.724c2.597 0 4.957 1.259 6.452
3.305l38.32 54.451c3.619 5.194-.079 12.354-6.452 12.354zM476.502 440.17c-27.068
0-48.943 21.953-48.943 49.021 0 26.99 21.875 48.943 48.943 48.943 26.989 0
48.943-21.953 48.943-48.943 0-27.066-21.954-49.021-48.943-49.021zm0 73.495c-13.535
0-24.472-11.016-24.472-24.471 0-13.535 10.937-24.473 24.472-24.473 13.533 0 24.472
10.938 24.472 24.473 0 13.455-10.938 24.471-24.472 24.471zM68.434 440.17c-4.328
0-7.869 3.543-7.869 7.869v23.922c0 4.328 3.541 7.869 7.869 7.869h87.971c2.282-15.738
10.229-29.666 21.718-39.658H68.434v-.002zm151.864 0c-26.989 0-48.943 21.953-48.943
49.021 0 26.99 21.954 48.943 48.943 48.943 27.068 0 48.943-21.953
48.943-48.943.001-27.066-21.874-49.021-48.943-49.021zm0 73.495c-13.534
0-24.471-11.016-24.471-24.471 0-13.535 10.937-24.473 24.471-24.473s24.472 10.938
24.472 24.473c0 13.455-10.938 24.471-24.472 24.471zm117.716-363.06h-91.198c4.485
13.298 6.846 27.54 6.846 42.255 0 74.28-60.431 134.711-134.711 134.711-13.535
0-26.675-2.045-39.029-5.744v86.949c0 4.328 3.541 7.869 7.869 7.869h265.96c4.329 0
7.869-3.541 7.869-7.869V174.211c-.001-13.062-10.545-23.606-23.606-23.606zM118.969
73.866C53.264 73.866 0 127.129 0 192.834s53.264 118.969 118.969 118.969 118.97-53.264
118.97-118.969-53.265-118.968-118.97-118.968zm0 210.864c-50.752
0-91.896-41.143-91.896-91.896s41.144-91.896 91.896-91.896c50.753 0 91.896 41.144
91.896 91.896 0 50.753-41.143 91.896-91.896 91.896zm35.097-72.488c-1.014 0-2.052-.

```

```

131-3.082-.407L112.641 201.5a11.808 11.808 0 0 1-8.729-11.396v-59.015c0-6.516
5.287-11.803 11.803-11.803 6.516 0 11.803 5.287 11.803 11.803v49.971l29.614 7.983c6.294
1.698 10.02 8.177 8.322 14.469-1.421 5.264-6.185 8.73-11.388 8.73z" fill="#FFF" /></svg>

</div>
<h3>Pick up or Delivery</h3>
<p>Get your food delivered! And enjoy your meal! Pay online on pickup or delivery</p>

</div>
</div>
</div>
</div>
</div>
<!-- 3 block sections ends -->
<div class="row">
    <div class="col-sm-12 text-center">
        <p class="pay-info">Pay by Cash on delivery</p>
    </div>
</div>
</div>
</section>
<!-- How it works block ends -->
<!-- Featured restaurants starts -->
<section class="featured-restaurants">
    <div class="container">
        <div class="row">
            <div class="col-sm-4">
                <div class="title-block pull-left">
                    <h4>Featured restaurants</h4> </div>
                </div>
            <div class="col-sm-8">
                <!-- restaurants filter nav starts -->
                <div class="restaurants-filter pull-right">

```

```

<nav class="primary pull-left">
    <ul>
        <li><a href="#" class="selected" data-filter="*">>all</a> </li>
        <?php
            // display categories here
            $res=
            mysqli_query($db,"select * from res_category");
            while($row=mysqli_fetch_array($res))
            {
                echo
                '<li><a href="#" data-filter="'. $row['c_name'].'">'. $row['c_name']. '</a> </li>';
            }
            ?>
        </ul>
    </nav>
</div>
<!-- restaurants filter nav ends -->
</div>
</div>
<!-- restaurants listing starts -->
<div class="row">
    <div class="restaurant-listing">
        <?php //fetching records from table and filter
using html data-filter tag
        $ress= mysqli_query($db,"select * from
        restaurant");

```

```

while($rows=mysqli_fetch_array($ress))
{
    // fetch records from res_category table according to category ID

    $query= mysqli_query($db,"select * from res_category where c_id='".$rows['c_id']."' ");
    $rowss=mysqli_fetch_array($query);

    echo '<div class="col-xs-12 col-sm-12 col-md-6 single-restaurant all '.$rowss['c_name'].'>
        <div class="restaurant-wrap">
            <div class="row">
                <div class="col-xs-12 col-sm-3 col-md-12 col-lg-3 text-xs-center">
                    <a class="restaurant-logo" href="dishes.php?res_id='.$
$rows['rs_id'].'"></a>
                </div>
            <!--end:col -->
            <div class="col-xs-12 col-sm-9 col-md-12 col-lg-9">
                <h5><a href="dishes.php?res_id='.$rows['rs_id'].'">'.
$rows['title'].'
                </a></h5> <span>'.$rows['address'].'
                </span>
            </div>
        </div>
    </div>';
}

```

```
<div class="bottom-part">

    <div class="cost"><i class="fa fa-check"></i> Min Rs
    100</div>

    <div class="mins"><i class="fa fa-motorcycle"></i> 30
    min</div>

    <div class="ratings"> <span>

        <i class="fa fa-star"></i>

        <i class="fa fa-star"></i>

        <i class="fa fa-star"></i>

        <i class="fa fa-star"></i>

        <i class="fa fa-star-o"></i>

    </span> (122) </div>

</div>

</div>

<!-- end:col -->

</div>
```

```
<!-- end:row -->

</div>

<!--end:Restaurant wrap -->

</div>;
```

}

?>

```
</div>

</div>

<!-- restaurants listing ends -->

</div>

</section>

<!-- Featured restaurants ends -->

<section class="app-section">
  <div class="app-wrap">
    <div class="container">
      <div class="row text-img-block text-xs-left">
        <div class="container">
          <div class="col-xs-12 col-sm-5 right-image text-center">
            <figure>  </figure>
```



```

<p>Join the thousands of other restaurants who benefit from having their
menus on TakeOff</p>
</div>
</div>
</div>
<!-- bottom footer ends -->
</div>
</footer>
<!-- end:Footer -->
</div>
<!--/end:Site wrapper -->
<!-- Bootstrap core JavaScript
=====
-->
<script src="js/jquery.min.js"></script>
<script src="js/tether.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/animstion.min.js"></script>
<script src="js/bootstrap-slider.min.js"></script>
<script src="js/jquery.isotope.min.js"></script>
<script src="js/headroom.js"></script>
<script src="js/foodpicky.min.js"></script>
</body>

</html>

```

## Login Page

```

<!DOCTYPE html>
<html lang="en" >

<head>

```

```

<meta charset="UTF-8">
<title>login</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/
reset.min.css">

<link rel='stylesheet prefetch' href='https://fonts.googleapis.com/css?family=Roboto:
400,100,300,500,700,900|RobotoDraft:400,100,300,500,700,900'>
<link rel='stylesheet prefetch' href='https://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/
css/font-awesome.min.css'>

<link rel="stylesheet" href="css/login.css">

<style type="text/css">
#buttn {
    color:#fff;
    background-color: #ff3300;
}
</style>

</head>

<body>
<?php
include("connection/connect.php"); //INCLUDE CONNECTION
error_reporting(0); // hide undefined index errors
session_start(); // temp sessions
if(isset($_POST['submit'])) // if button is submit
{
    $username = $_POST['username']; //fetch records from login form
    $password = $_POST['password'];

```

```

if(!empty($_POST["submit"])) // if records were not empty
{
    $loginquery ="SELECT * FROM users WHERE username='".$username' &&
password='".$md5($password)."'; //selecting matching records
    $result=mysqli_query($db, $loginquery); //executing
    $row=mysqli_fetch_array($result);

    if(is_array($row)) // if matching records in the array & if everything is
right
    {
        $_SESSION["user_id"] = $row['u_id']; // put user id into temp
session
        header("refresh:
1;url=index.php"); // redirect to index.php page
    }
    else
    {
        $message = "Invalid Username or Password!"; // throw error
    }
}
?>

```

```

<!-- Form Mixin-->
<!-- Input Mixin-->
<!-- Button Mixin-->
<!-- Pen Title-->
<div class="pen-title">

```

```

<h1>Login Form</h1>
</div>
<!-- Form Module-->
<div class="module form-module">
<div class="toggle">

</div>
<div class="form">
<h2>Login to your account</h2>
<span style="color:red;"><?php echo $message; ?></span>
<span style="color:green;"><?php echo $success; ?></span>
<form action="" method="post">
<input type="text" placeholder="Username" name="username"/>
<input type="password" placeholder="Password" name="password"/>
<input type="submit" id="buttn" name="submit" value="login" />
</form>
</div>

<div class="cta">Not registered?<a href="registration.php" style="color:#f30;"> Create an
account</a></div>
</div>
<script src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>

</body>

</html>

```

## Registration Page

```
<!DOCTYPE html>
<html lang="en">
<?php

session_start(); //temp session
error_reporting(0); // hide undefined index
include("connection/connect.php"); // connection
if(isset($_POST['submit'])) //if submit btn is pressed
{
    if(empty($_POST['firstname']) || //fetching and find if its empty
       empty($_POST['lastname']) ||
       empty($_POST['email']) ||
       empty($_POST['phone']) ||
       empty($_POST['password']) ||
       empty($_POST['cpassword']) ||
       empty($_POST['cpassword']))
    {
        $message = "All fields must be Required!";
    }
    else
    {
        //cheching username & email if already present
        $check_username= mysqli_query($db, "SELECT username FROM users where
username = '".$_POST['username']."' ");
        $check_email = mysqli_query($db, "SELECT email FROM users where email = "".
$_POST['email']."' ");
    }
}
```

```

if($_POST['password'] != $_POST['cpassword']){ //matching passwords
$message = "Password not match";
}

elseif(strlen($_POST['password']) < 6) //cal password length
{
    $message = "Password Must be >=6";
}

elseif(strlen($_POST['phone']) < 10) //cal phone length
{
    $message = "invalid phone number!";
}

elseif (!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) // Validate email
address
{
    $message = "Invalid email address please type a valid email!";
}

elseif(mysqli_num_rows($check_username)> 0) //check username
{
    $message = 'username Already exists!';
}

elseif(mysqli_num_rows($check_email)> 0) //check email
{
    $message = 'Email Already exists!';
}

else{

//inserting values into db

```

```

$mql = "INSERT INTO
users(username,f_name,l_name,email,phone,password,address) VALUES(".
$_POST['username']."'","'".$_POST['firstname']."'","'".$_POST['lastname']."'","'.
$_POST['email']."'","'".$_POST['phone']."'","'".md5($_POST['password'])."'",
$_POST['address']."'");

mysqli_query($db, $mql);

$success = "Account Created successfully! <p>You will be redirected in
<span id='counter'>5</span> second(s).</p>

<script type='text/javascript'>

function countdown() {

    var i = document.getElementById('counter');

    if (parseInt(i.innerHTML)<=0) {

        location.href = 'login.php';

    }

    i.innerHTML = parseInt(i.innerHTML)-1;

}

setInterval(function(){ countdown(); },1000);

</script>'";

```

```

        header("refresh:5;url=login.php"); // redirected once inserted success
    }
}

}

?>

<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<!-- The above 3 meta tags *must* come first in the head; any other head content must
come *after* these tags --&gt;
&lt;meta name="description" content=""&gt;
&lt;meta name="author" content=""&gt;
&lt;link rel="icon" href="#"&gt;
&lt;title&gt;Food Order&lt;/title&gt;
<!-- Bootstrap core CSS --&gt;
&lt;link href="css/bootstrap.min.css" rel="stylesheet"&gt;
&lt;link href="css/font-awesome.min.css" rel="stylesheet"&gt;
&lt;link href="css/animstion.min.css" rel="stylesheet"&gt;
&lt;link href="css/animate.css" rel="stylesheet"&gt;
<!-- Custom styles for this template --&gt;
&lt;link href="css/style.css" rel="stylesheet"&gt; &lt;/head&gt;

&lt;body&gt;
&lt;div class="site-wrapper animstion" data-animstion-in="fade-in" data-animstion-
out="fade-out"&gt;
</pre>

```

```

<!--header starts-->
<header id="header" class="header-scroll top-header headrom">
    <!-- .navbar -->
    <nav class="navbar navbar-dark">
        <div class="container">
            <button class="navbar-toggler hidden-lg-up" type="button" data-
                toggle="collapse" data-target="#mainNavbarCollapse">&#9776;</button>
            <a class="navbar-brand" href="index.html"></a>
            <div class="collapse navbar-toggleable-md float-lg-right"
                id="mainNavbarCollapse">
                <ul class="nav navbar-nav">
                    <li class="nav-item"><a class="nav-link
                        active" href="index.php">Home <span class="sr-only">(current)</span></a></li>
                    <li class="nav-item"><a class="nav-link active"
                        href="restaurants.php">Restaurants <span class="sr-only"></span></a></li>

                    <?php
                    if(empty($_SESSION["user_id"]))
                    {
                        echo '<li class="nav-item"><a
                            href="login.php" class="nav-link active">login</a></li>
                            <li class="nav-item"><a
                            href="registration.php" class="nav-link active">signup</a></li>';
                    }
                    else
                    {
                        echo '<li
                            class="nav-item"><a href="your_orders.php" class="nav-link active">your orders</a></li>';
                    }
                </ul>
            </div>
        </div>
    </nav>
</header>

```

```

echo '<li class="nav-
item"><a href="logout.php" class="nav-link active">logout</a> </li>';
}

?>

</ul>
</div>
</div>
</nav>
<!-- /.navbar -->
</header>
<div class="page-wrapper">
<div class="breadcrumb">
<div class="container">
<ul>
<li><a href="#" class="active">
<span style="color:red;"><?php echo $message; ?></
span>
<span style="color:green;">
<?php echo $success; ?>
</span>

</a></li>

</ul>
</div>
</div>
<section class="contact-page inner-page">
<div class="container">
<div class="row">

```

```

<!-- REGISTER -->
<div class="col-md-8">
  <div class="widget">
    <div class="widget-body">

      <form action="" method="post">
        <div class="row">
          <div class="form-group col-sm-12">
            <label for="exampleInputEmail1">User-Name</label>
            <input class="form-control" type="text" name="username" id="example-text-input" placeholder="UserName">
          </div>
          <div class="form-group col-sm-6">
            <label for="exampleInputEmail1">First Name</label>
            <input class="form-control" type="text" name="firstname" id="example-text-input" placeholder="First Name">
          </div>
          <div class="form-group col-sm-6">
            <label for="exampleInputEmail1">Last Name</label>
            <input class="form-control" type="text" name="lastname" id="example-text-input-2" placeholder="Last Name">
          </div>
          <div class="form-group col-sm-6">
            <label for="exampleInputEmail1">Email address</label>
            <input type="text" class="form-control" name="email" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter email"> <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else.</small>
          </div>
          <div class="form-group col-sm-6">

```

```

        <label for="exampleInputEmail1">Phone number</label>
        <input class="form-control" type="text" name="phone"
id="example-tel-input-3" placeholder="Phone"> <small class="form-text text-muted">We'll
never share your email with anyone else.</small>
    </div>

    <div class="form-group col-sm-6">
        <label for="exampleInputPassword1">Password</label>
        <input type="password" class="form-control" name="password"
id="exampleInputPassword1" placeholder="Password">
    </div>

    <div class="form-group col-sm-6">
        <label for="exampleInputPassword1">Repeat password</label>
        <input type="password" class="form-control" name="cpassword"
id="exampleInputPassword2" placeholder="Password">
    </div>

<div class="form-group
col-sm-12">
        <label for="exampleTextarea">Delivery Address</label>
        <textarea class="form-control" id="exampleTextarea"
name="address" rows="3"></textarea>
    </div>

</div>

<div class="row">
    <div class="col-sm-4">
        <p> <input type="submit" value="Register" name="submit"
class="btn theme-btn"> </p>
    </div>
    </div>
</form>

```

```

        </div>

    <!-- end: Widget -->

</div>
<!-- /REGISTER -->
</div>
<!-- WHY? -->

<!-- /WHY? -->
</div>
</div>
</section>
<section class="app-section">
<div class="app-wrap">
<div class="container">
<div class="row text-img-block text-xs-left">
<div class="container">
<div class="col-xs-12 col-sm-6 right-image text-center">
<figure>  </figure>
</div>
<div class="col-xs-12 col-sm-6 left-text">
<h3>The Best Food Delivery App</h3>
<p>Now you can make food happen pretty much wherever you are thanks
to the free easy-to-use Food Delivery & Takeout App.</p>
<div class="social-btns">
<a href="#" class="app-btn apple-button clearfix">
<div class="pull-left"><i class="fa fa-apple"></i> </div>
<div class="pull-right"> <span class="text">Available on the</span>
<span class="text-2">App Store</span> </div>
</a>
<a href="#" class="app-btn android-button clearfix">

```



```
</div>
</div>
<!-- bottom footer ends -->
</div>
</footer>
<!-- end:Footer -->
</div>
<!-- end:page wrapper -->
</div>
<!--/end:Site wrapper -->
<!-- Bootstrap core JavaScript
=====
<script src="js/jquery.min.js"></script>
<script src="js/tether.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/animstion.min.js"></script>
<script src="js/bootstrap-slider.min.js"></script>
<script src="js/jquery.isotope.min.js"></script>
<script src="js/headroom.js"></script>
<script src="js/foodpicky.min.js"></script>
</body>

</html>
```

## Dishes Page

```
<!DOCTYPE html>
<html lang="en">
<?php
include("connection/connect.php"); // connection to db
error_reporting(0);
session_start();

include_once 'product-action.php'; //including controller

?>

<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<!-- The above 3 meta tags *must* come first in the head; any other head content must
come *after* these tags --&gt;
&lt;meta name="description" content=""&gt;
&lt;meta name="author" content=""&gt;
&lt;link rel="icon" href="#"&gt;
&lt;title&gt;Food order&lt;/title&gt;
<!-- Bootstrap core CSS --&gt;
&lt;link href="css/bootstrap.min.css" rel="stylesheet"&gt;
&lt;link href="css/font-awesome.min.css" rel="stylesheet"&gt;
&lt;link href="css/animstion.min.css" rel="stylesheet"&gt;
&lt;link href="css/animate.css" rel="stylesheet"&gt;
<!-- Custom styles for this template --&gt;</pre>
```

```

<link href="css/style.css" rel="stylesheet"> </head>

<body>
  <div class="site-wrapper animsition" data-animsition-in="fade-in" data-animsition-out="fade-out">
    <!--header starts-->
    <header id="header" class="header-scroll top-header headrom">
      <!-- .navbar -->
      <nav class="navbar navbar-dark">
        <div class="container">
          <button class="navbar-toggler hidden-lg-up" type="button" data-toggle="collapse" data-target="#mainNavbarCollapse">&#9776;</button>
          <a class="navbar-brand" href="index.html"> </a>
          <div class="collapse navbar-toggleable-md float-lg-right" id="mainNavbarCollapse">
            <ul class="nav navbar-nav">
              <li class="nav-item"><a class="nav-link active" href="index.php">Home <span class="sr-only">(current)</span></a> </li>
              <li class="nav-item"><a class="nav-link active" href="restaurants.php">Restaurants <span class="sr-only"></span></a> </li>
                <?php
                  if(empty($_SESSION["user_id"]))
                    {
                      echo '<li class="nav-item"><a href="login.php" class="nav-link active">login</a> </li>
                            <li class="nav-item"><a href="registration.php" class="nav-link active">signup</a> </li>';
                    }
                  else
                
```

```

        {
echo '<li
class="nav-item"><a href="your_orders.php" class="nav-link active">your orders</a> </li>';
echo '<li class="nav-
item"><a href="logout.php" class="nav-link active">logout</a> </li>';
}

?>

</ul>
</div>
</div>
</nav>
<!-- /.navbar -->
</header>
<div class="page-wrapper">
<!-- top Links -->
<div class="top-links">
<div class="container">
<ul class="row links">

<li class="col-xs-12 col-sm-4 link-item"><span>1</span><a
href="restaurants.php">Choose Restaurant</a></li>
<li class="col-xs-12 col-sm-4 link-item active"><span>2</span><a
href="dishes.php?res_id=<?php echo $_GET['res_id']; ?>">Pick Your favorite food</a></li>
<li class="col-xs-12 col-sm-4 link-item"><span>3</span><a href="#">Order
and Pay online</a></li>
</ul>
</div>

```

```

</div>
<!-- end:Top links -->
<!-- start: Inner page hero -->
<?php $ress= mysqli_query($db,"select * from restaurant where
rs_id='$_GET[res_id]'");
$rows=mysqli_fetch_array($ress);

?>
<section class="inner-page-hero bg-image" data-image-src="images/img/dish.jpeg">
<div class="profile">
<div class="container">
<div class="row">
<div class="col-xs-12 col-sm-12 col-md-4 col-lg-4 profile-img">
<div class="image-wrap">
<figure><?php echo '</figure>
</div>
</div>

<div class="col-xs-12 col-sm-12 col-md-8 col-lg-8 profile-desc">
<div class="pull-left right-text white-txt">
<h6><a href="#"><?php echo $rows['title']; ?></a></h6>
<p><?php echo $rows['address']; ?></p>
<ul class="nav nav-inline">
<li class="nav-item"><a class="nav-link active" href="#"><i
class="fa fa-check"></i> Min Rs 100</a> </li>
<li class="nav-item"> <a class="nav-link" href="#"><i class="fa fa-
motorcycle"></i> 30 min</a> </li>
<li class="nav-item ratings">
<a class="nav-link" href="#"> <span>

```

```
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star-o"></i>
</span> </a>
</li>
</ul>
</div>
</div>

</div>
</div>
</div>
</section>
<!-- end:Inner page hero -->
<div class="breadcrumb">
<div class="container">

</div>
</div>
<div class="container m-t-30">
<div class="row">
<div class="col-xs-12 col-sm-4 col-md-4 col-lg-3">

<div class="widget widget-cart">
<div class="widget-heading">
<h3 class="widget-title text-dark">
Your Shopping Cart
</h3>
```

```

<div class="clearfix"></div>

</div>
<div class="order-row bg-white">
    <div class="widget-body">

        <?php
$item_total = 0;

foreach ($_SESSION["cart_item"] as $item) // fetch items define current into session ID
{
?>

        <div class="title-row">
            <?php echo
$item["title"]; ?><a href="dishes.php?res_id=<?php echo $_GET['res_id']; ?
>&action=remove&id=<?php echo $item["d_id"]; ?>">
                <i class="fa fa-
trash pull-right"></i></a>
            </div>

        <div class="form-group row no-gutter">
            <div class="col-xs-8">
                <input type="text" class="form-control b-r-0" value=<?php
echo "Rs". $item["price"]; ?> readonly id="exampleSelect1">
            </div>
            <div class="col-xs-4">

```

```

<input class="form-control" type="text" readonly value='<?php
echo $item["quantity"]; ?>' id="example-number-input"> </div>

</div>

<?php
$item_total += ($item["price"]*$item["quantity"]); // calculating current price into cart
}

?>

</div>
</div>

<!-- end:Order row -->

<div class="widget-body">
  <div class="price-wrap text-xs-center">
    <p>TOTAL</p>
    <h3 class="value"><strong><?php echo "Rs". $item_total; ?></strong></h3>
    <p>Free Shipping</p>
    <a href="checkout.php?res_id=<?php echo $_GET['res_id'];?>&action=check" class="btn theme-btn btn-lg">Checkout</a>
  </div>
</div>

```

```

</div>

</div>

<div class="col-xs-12 col-sm-8 col-md-8 col-lg-6">

    <!-- end:Widget menu -->
    <div class="menu-widget" id="2">
        <div class="widget-heading">
            <h3 class="widget-title text-dark">
                POPULAR ORDERS Delicious hot food! <a class="btn btn-link pull-right" data-toggle="collapse" href="#popular2" aria-expanded="true">
                    <i class="fa fa-angle-right pull-right"></i>
                    <i class="fa fa-angle-down pull-right"></i>
                </a>
            </h3>
            <div class="clearfix"></div>
        </div>
        <div class="collapse in" id="popular2">
            <?php // display values and item of food/dishes
                $stmt = $db->
            >prepare("select * from dishes where rs_id='$_GET[res_id]'");
                $stmt->execute();
                $products = $stmt->
            >get_result();
                if (!empty($products))
                {
                    foreach($products as
                        $product)
                {

```

```

?>

<div class="food-item">
  <div class="row">
    <div class="col-xs-12 col-sm-12 col-lg-8">
      <form
method="post" action='dishes.php?res_id=<?php echo $_GET['res_id'];?>
>&action=add&id=<?php echo $product['d_id']; ?>'>
        <div class="rest-logo pull-left">
          <a class="restaurant-logo pull-left" href="#"><?php echo
'; ?></a>
        </div>
        <!-- end:Logo -->
        <div class="rest-descr">
          <h6><a href="#"><?php echo $product['title']; ?></a></h6>
          <p> <?php echo $product['slogan']; ?></p>
        </div>
        <!-- end:Description -->
      </div>
      <!-- end:col -->
      <div class="col-xs-12 col-sm-12 col-lg-4 pull-right item-cart-info">
        <span class="price
pull-left" >Rs<?php echo $product['price']; ?></span>
        <input class="b-
r-0" type="text" name="quantity" style="margin-left:30px;" value="1" size="2" />
        <input
type="submit" class="btn theme-btn" style="margin-left:40px;" value="Add to cart" />
      </div>
    </form>
  </div>

```

```
<!-- end:row -->
</div>
<!-- end:Food item -->

<?php
    }
}

?>

</div>
<!-- end:Collapse -->
</div>
<!-- end:Widget menu -->

</div>
<!-- end:Bar -->

<!-- end:Right Sidebar -->
</div>
<!-- end:row -->
</div>
<!-- end:Container -->
<section class="app-section">
    <div class="app-wrap">
        <div class="container">
            <div class="row text-img-block text-xs-left">
                <div class="container">
```



<p>Concept design of oline food order and deliveye,planned as  
 restaurant direcotry</p>

<h5>Phone: <a href="tel:+080000012222">080 000012 222</a></h5>  
 </div>

<div class="col-xs-12 col-sm-5 additional-info color-gray">  
 <h5>Addition informations</h5>  
 <p>Join the thousands of other restaurants who benefit from having  
 their menus on TakeOff</p>  
 </div>  
 </div>  
 </div>  
 </div>  
 <!-- bottom footer ends -->  
 </div>  
 </footer>  
 <!-- end:Footer -->  
 </div>  
 <!-- end:page wrapper -->  
 </div>  
 <!--/end:Site wrapper -->  
 <!-- Modal -->  
<div class="modal fade" id="order-modal" tabindex="-1" role="dialog" aria-hidden="true">  
 <div class="modal-dialog" role="document">  
 <div class="modal-content">  
 <button type="button" class="close" data-dismiss="modal" aria-label="Close">  
 <span aria-hidden="true">&times;</span> </button>  
 <div class="modal-body cart-addon">  
 <div class="food-item white">  
 <div class="row">  
 <div class="col-xs-12 col-sm-6 col-lg-6">

```

<div class="item-img pull-left">
    <a class="restaurant-logo pull-left" href="#"></a>
</div>
<!-- end:Logo -->
<div class="rest-descr">
    <h6><a href="#">Sandwich de Alegranza Grande Menü (28 - 30
cm.)</a></h6> </div>
    <!-- end:Description -->
</div>
<!-- end:col -->
<div class="col-xs-6 col-sm-2 col-lg-2 text-xs-center"> <span class="price
pull-left">Rs 2.99</span></div>
<div class="col-xs-6 col-sm-4 col-lg-4">
    <div class="row no-gutter">
        <div class="col-xs-7">
            <select class="form-control b-r-0" id="exampleSelect2">
                <option>Size SM</option>
                <option>Size LG</option>
                <option>Size XL</option>
            </select>
        </div>
        <div class="col-xs-5">
            <input class="form-control" type="number" value="0" id="quant-
input-2"> </div>
        </div>
    </div>
    <!-- end:row -->
</div>
<!-- end:Food item -->

```

```

<div class="food-item">
    <div class="row">
        <div class="col-xs-12 col-sm-6 col-lg-6">
            <div class="item-img pull-left">
                <a class="restaurant-logo pull-left" href="#"></a>
            </div>
            <!-- end:Logo -->
            <div class="rest-descr">
                <h6><a href="#">Sandwich de Alegranza Grande Menü (28 - 30
cm.)</a></h6> </div>
                <!-- end:Description -->
            </div>
            <!-- end:col -->
            <div class="col-xs-6 col-sm-2 col-lg-2 text-xs-center"> <span class="price
pull-left">Rs 2.49</span></div>
            <div class="col-xs-6 col-sm-4 col-lg-4">
                <div class="row no-gutter">
                    <div class="col-xs-7">
                        <select class="form-control b-r-0" id="exampleSelect3">
                            <option>Size SM</option>
                            <option>Size LG</option>
                            <option>Size XL</option>
                        </select>
                    </div>
                    <div class="col-xs-5">
                        <input class="form-control" type="number" value="0" id="quant-
input-3"> </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

<!-- end:row -->
</div>
<!-- end:Food item -->
<div class="food-item">
<div class="row">
<div class="col-xs-12 col-sm-6 col-lg-6">
<div class="item-img pull-left">
<a class="restaurant-logo pull-left" href="#"></a>
</div>
<!-- end:Logo -->
<div class="rest-descr">
<h6><a href="#">Sandwich de Alegranza Grande Menü (28 - 30
cm.)</a></h6> </div>
<!-- end:Description -->
</div>
<!-- end:col -->
<div class="col-xs-6 col-sm-2 col-lg-2 text-xs-center"> <span class="price
pull-left">Rs 1.99</span></div>
<div class="col-xs-6 col-sm-4 col-lg-4">
<div class="row no-gutter">
<div class="col-xs-7">
<select class="form-control b-r-0" id="exampleSelect5">
<option>Size SM</option>
<option>Size LG</option>
<option>Size XL</option>
</select>
</div>
<div class="col-xs-5">
<input class="form-control" type="number" value="0" id="quant-
input-4"> </div>

```

```

        </div>
    </div>
</div>
<!-- end:row -->
</div>
<!-- end:Food item -->
<div class="food-item">
    <div class="row">
        <div class="col-xs-12 col-sm-6 col-lg-6">
            <div class="item-img pull-left">
                <a class="restaurant-logo pull-left" href="#"></a>
            </div>
            <!-- end:Logo -->
            <div class="rest-descr">
                <h6><a href="#">Sandwich de Alegranza Grande Menü (28 - 30
cm.)</a></h6> </div>
            <!-- end:Description -->
        </div>
        <!-- end:col -->
        <div class="col-xs-6 col-sm-2 col-lg-2 text-xs-center"> <span class="price
pull-left">Rs 3.15</span></div>
        <div class="col-xs-6 col-sm-4 col-lg-4">
            <div class="row no-gutter">
                <div class="col-xs-7">
                    <select class="form-control b-r-0" id="exampleSelect6">
                        <option>Size SM</option>
                        <option>Size LG</option>
                        <option>Size XL</option>
                    </select>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

<div class="col-xs-5">
    <input class="form-control" type="number" value="0" id="quant-
input-5"></div>
</div>
</div>
</div>
<!-- end:row -->
</div>
<!-- end:Food item -->
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
    <button type="button" class="btn theme-btn">Add to cart</button>
</div>
</div>
</div>
<!-- Bootstrap core JavaScript
=====
<script src="js/jquery.min.js"></script>
<script src="js/tether.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/animstion.min.js"></script>
<script src="js/bootstrap-slider.min.js"></script>
<script src="js/jquery.isotope.min.js"></script>
<script src="js/headroom.js"></script>
<script src="js/foodpicky.min.js"></script>
</body>

</html>

```

## **FUTURE ENHANCEMENT**

We think that not a single project is ever considered as complete forever. Because our mind is always thinking something new and our necessities also are growing day by day. We always want something more than what we have. Our application also,

If you see at the first glance then you find it to be complete but we want to make it still mature and fully automatically.

This project can be easily implemented under various situations. We can add new features as and when we require. Reusability is possible as and when required in this project. There is flexibility in all the modules. We can also include online payment services to this project using PayPal. This will help customers to pay online for their purchases using Credit/Debit card. There are many features which could be added to this project for making this project more productive.

- ✓ Providing online Placement Record through our site
- ✓ Providing personalized inbox to the user
- ✓ Providing video conferencing with the Artists
- ✓ Providing links to news which will elaborate more information about them
- ✓ Providing Ajax technology refreshing in our website

## **FUTURE WORK**

**The following section describes the work that will be implemented with future releases of the software.**

- ✓ Customize orders: Allow customers to customize food orders

- ✓ Enhance User Interface by adding more user interactive features. Provide Deals and promotional
- ✓ Offer details to home page. Provide Recipes of the Week/Day to Home Page
- ✓ Payment Options: Add different payment options such as PayPal, Cash, Gift Cards etc. Allow to save payment details for future use.
- ✓ Allow to process an order as a Guest
- ✓ Delivery Options: Add delivery option
- ✓ Order Process Estimate: Provide customer a visual graphical order status bar
- ✓ Order Status: Show only Active orders to Restaurant Employees.
- ✓ Order Ready notification: Send an Order Ready notification to the customer
- ✓ Restaurant Locator: Allow to find and choose a nearby restaurant
- ✓ Integrate with In store touch screen devices like iPad
- ✓ Making a mobile app for the user is also a first step development.

## CONCLUSION

The software development is never completed. There is always a need for modification. There could have been other approaches to implement the system. I have tried to my level best to make the system an interactive as possible. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop a web application for purchasing items from a shop.

This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using html & css, usage of responsive templates, designing of android applications, and management of database using mysql. The entire system is secured. Also the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project.

This project has given us great satisfaction in having designed an application which can be implemented to any nearby shops or branded shops selling various kinds of products by simple modifications.

## About this Project

1. It is simple and user friendly
2. Platform independent
3. vide scope for future expansion
4. manual as well as paper works can be fully eliminated in the billing branch
5. accuracy and reliability are surely increased
6. it make sure that unauthorized personal cannot execute this program

System security refers to the technical innovations and procedures applied to hardware and operating system. To protect against deliberate or accidental damage from a defined threat. In contrast, data security is the protection of data, some loss, disclosure, modification and destruction. The system security problem can be provided into four related issues.

1. Security
2. integrity
3. privacy
4. access procedures

To implement this application the web server used is Microsoft IIS and the server side technology used to create the web pages is PHP. PHP has several advantages such as enhanced performance, scalability, built-in security and simplicity. The database used to store the data is MySQL database. In the course of the implementation of this application many lessons have been learned including designing an interface, database access technique and

programming for the web. This application helped in understanding the different technologies used to create interactive web pages. The implementation of the project has given a precise knowledge of how PHP is used to create web pages and how to connect to MySQL database. Overall the implementation of this project was an excellent learning opportunity.

## **BIBLIOGRAPHY**

During the development of our system, we have taken the reference from Books and journals, which we would like to mention in this section.

These books acted as our tutors during the system development..

### **System Analysis And Design**

- Kenneth E. Kendall, Julie E. Kendall

### **An Analysis and Design of Information Systems**

- Grayce M. Booth

## **Software Engineering**

- Roger S. Pressman

## **Database Management System**

- James A. Larson

## **PHP: A Beginner's Guide**

- RiwantoMegosinarso

These are the following links which assist me at each and every step in completing this project, without them

- ✓ [www.google.com](http://www.google.com)
- ✓ [www.mysql.com](http://www.mysql.com)
- ✓ <http://en.wikipedia.org/wiki/Recruitment>
- ✓ [www.w3schools.com](http://www.w3schools.com)
- ✓ [www.google.co.in](http://www.google.co.in)
- ✓ <http://www.recruitmentsystems.com/>
- ✓ [http://www.codeproject.com/KB/architecture/OOP\\_Concepts\\_and\\_manymore.aspx](http://www.codeproject.com/KB/architecture/OOP_Concepts_and_manymore.aspx)
- ✓ [http://en.wikipedia.org/wiki/Human\\_resource\\_management](http://en.wikipedia.org/wiki/Human_resource_management)
- ✓ [http://en.wikipedia.org/wiki/Object-oriented\\_programming](http://en.wikipedia.org/wiki/Object-oriented_programming)