**INDUSTRIAL TRAINING REPORT**

ON

**“Online-Food-Order”**

SUBMITTED IN PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE OF

MASTER OF SCIENCE

IN

INFORMATION TECHNOLOGY

  
 **Department of Computer Science & Informatics**

**SUBMITTED BY:**

Name: Damini

Roll no: CUHP19IT10

**FACULTY INCHARGE:**

Mr. Keshav Singh Rawat, Asst. Prof.

Mr. Manoj Dhiman, Asst. Prof.

Mr. Ajay Kumar, Asst. Prof.

CENTRAL UNIVERSITY OF HIMACHAL PRADESH

Temporary Academic Block, Shahpur, Kangra, Himachal Pradesh -176206

**December 2019**

**CONTENTS**

* Certificate of Training
* Industry Supervisor’s Report
* Declaration by student
* Acknowledgement
* Technology Used
* Project
* Project details
* Snapshots of project
* Fortnightly Report
* References

**DECLARATION**

We hereby declare that the Industrial Training Report entitled ("SLIM") is an authentic record of our own work of 4 to 6 weeks Industrial Training during the period from 07-July-2019 to 10-Aug-2019 as per the requirement of the university and for partial fulfillment for the award of degree of M.Sc. in Information Technology, under the guidance of Mr. Manoj Dhiman, Mr. Keshav Singh Rawat and Mr. Ajay Kumar.

**Dated** :

**(Signature of student)**

**(Damini)**

**(CUHP19IT10) (M.Sc. IT- III Semester)**

**ACKNOWLEDGEMENT**

“**Choose what you are passionate about & what you love to do. The success will be yours!** “

The experience of making the project had been productive and enjoyable. It proved an opportunity for us to upgrade our skills and upgraded our professional knowledge. In course of present work it has been our privilege to receive help and assistance from many quarters. Before we get into the thick of things, we would like to share few heartfelt words with the people who were the part of project in numerous ways.

My primary thanks goes to Miss.Harita ……..our project guide who poured over every inch of the manuscript with painstaking attention to detail and made a semi- infinite number of helpful suggestions.

Also would like to acknowledge our thanks to Mr. Keshav Rawat, Mr. Manoj Dhiman , Mr Ajay Kumar and Mr. MAYANK CHOPRA for being instrumental in making this project possible and successful.

Damini

(**CUHP19IT10**)

**TECHNOLOGY USED**

**Technology Implemented :** Apache Server

**Language Used                   :**  PHP&JS REACT

**Database                              :** My SQL

**User Interface Design       :**  HTML, AJAX,JQUERY,JAVASCRIPT

**Web Browser                      :** Mozilla, Google Chrome, IE8,OPERA

**Software                               :**  WAMP Server

**SOFTWARE REQUIREMENT ANALYSIS**

The analysis part of designing of the software focuses on the requirement analysis of the software. It is based on studying of what is required from the software, the nature of program to be built, behavior, performance and interface. It is based on studying the existing system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HARDWARE REQUIREMENTS | | | | |
| PROCESSOR | RAM | | DISK SPACE | |
| Pentium IV or above version | 512 MB OR Higher | | 500 MB or above | |
| SOFTWARE REQUIREMENTS | | | | |
| OPERATING SYSTEM | | DATABASE | | SERVER |
| WIN-XP,WIN-7 ,WIN-10 | | MY SQL, MYSQLI | | APACHE(WAMP) |

**LANGUAGES USED**

* **PHP** For Development
* **HTML** , **HTML5** For designing
* **CSS** And **CSS3** For style
* **My SQL** For database design
* **JavaScript** For client-side validations
* **JQuery** For client-side Scripting

**PHP Hypertext Preprocessor**

PHP is a [server-side scripting](http://en.wikipedia.org/wiki/Server-side_scripting) language designed for [web development](http://en.wikipedia.org/wiki/Web_development) but also used as a [general-purpose programming language](http://en.wikipedia.org/wiki/General-purpose_programming_language). As of January 2013, PHP was installed on more than 240 million [websites](http://en.wikipedia.org/wiki/Website) and 2.1 million servers. Originally created by [Rasmus Lerdorf](http://en.wikipedia.org/wiki/Rasmus_Lerdorf) in 1995, the [reference implementation](http://en.wikipedia.org/wiki/Reference_implementation) of PHP is now produced by The PHP Group. While PHP originally stood for *Personal Home Page*, it now stands for *PHP: Hypertext Preprocessor*, a recursive [backronym](http://en.wikipedia.org/wiki/Backronym).

PHP code is [interpreted](http://en.wikipedia.org/wiki/Interpreter_%28computing%29) by a web server with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly into an [HTML](http://en.wikipedia.org/wiki/HTML) source document rather than calling an external file to process data. It has also evolved to include a [command-line interface](http://en.wikipedia.org/wiki/Command-line_interface) capability and can be used in standalone graphical.

PHP is [free software](http://en.wikipedia.org/wiki/Free_software) released under the [PHP License](http://en.wikipedia.org/wiki/PHP_License). PHP has been widely ported and can be deployed on most web servers on almost every [operating system](http://en.wikipedia.org/wiki/Operating_system) and [platform](http://en.wikipedia.org/wiki/Computing_platform), free of charge.

**Syntax**

The PHP interpreter only executes PHP code within its [delimiters](http://en.wikipedia.org/wiki/Delimiter). Anything outside its delimiters is not processed by PHP (although non-PHP text is still subject to [control structures](http://en.wikipedia.org/wiki/Control_structures) described within PHP code). The most common delimiters are <?php to open and ?> to close PHP sections.

<script language="php"> and </script> delimiters are also available, as are the shortened forms <? or <?= (which is used to echo back a [string](http://en.wikipedia.org/wiki/String_(computer_science)) or [variable](http://en.wikipedia.org/wiki/Variable_(programming))) and ?> as well as [ASP](http://en.wikipedia.org/wiki/Active_Server_Pages)-style short forms <% or <%= and %>. While short delimiters are used, they make script files less portable as support for them can be disabled in the [PHP configuration](http://wiki.php.net/rfc/shortags), and so they are discouraged. The purpose of all these delimiters is to separate PHP code from non-PHP code, including HTML.

Variables are prefixed with a [dollar symbol](http://en.wikipedia.org/wiki/Dollar_sign), and a [type](http://en.wikipedia.org/wiki/Primitive_type) does not need to be specified in advance.

Unlike function and class names, variable names are case sensitive. PHP has hundreds of base functions and thousands more via extensions. These functions are well documented on the PHP site; however, the built-in library has a wide variety of naming conventions and inconsistencies.

**Security**

Vulnerabilities are caused mostly by not following best practice programming rules: technical security flaws of the language itself or of its core libraries are not frequent (23 in 2008, about 1% of the total). Recognizing that programmers cannot be trusted, some languages include [taint checking](http://en.wikipedia.org/wiki/Taint_checking) to detect automatically the lack of [input validation](http://en.wikipedia.org/wiki/Data_validation) which induces many issues. Such a feature is being developed for PHP, but its inclusion in a release has been rejected several times in the past. Hosting PHP applications on a server requires careful and constant attention to deal with these security risks.[[53]](http://en.wikipedia.org/wiki/PHP#cite_note-52) There are advanced protection patches such as [Suhosin](http://en.wikipedia.org/wiki/Suhosin" \o "Suhosin) and [Hardening](http://en.wikipedia.org/wiki/Hardening_(computing))-Patch, especially designed for web hosting environments. [PHPIDS](http://en.wikipedia.org/wiki/PHPIDS) adds security to any PHP application to defend against intrusions. PHPIDS detects [Cross-site scripting](http://en.wikipedia.org/wiki/Cross-site_scripting) (XSS), [SQL injection](http://en.wikipedia.org/wiki/SQL_injection), header injection, [Directory traversal](http://en.wikipedia.org/wiki/Directory_traversal), Remote File Execution, [Local File Inclusion](http://en.wikipedia.org/wiki/Local_File_Inclusion), [Denial of Service](http://en.wikipedia.org/wiki/Denial_of_Service) (DoS).

**PHP has the following benefits:**

* **Short Time to Market**

PHP enables fast implementation of complex solutions. The benefit? The faster a new application enters the market, the higher your cost-efficiency and the greater your competitive advantage.

* **Easy Integration**

PHP runs on practically any platform - not only on Linux, but also, for example, on Windows, Unix, and IBM's System i. In addition, because PHP seamlessly integrates with other technologies (e.g. Java), we can re-use your current softwarecomponents. This is a major benefit, as no re-development is required for the existing software.

* **Flexibility**

PHP also offers great flexibility during and after the initial project. This is important, since functionality often changes during a project's lifetime. A great thing about PHP is that we can implement changes even after starting development, without losing valuable time.

* **Availability of Resources**

By resources, we mean two things. First, thanks to the popularity of PHP, the number of (usually free) resources online and offline is continuously growing. No matter what your needs are, it is very likely that someone has already developed something very similar, be it related to frameworks, CMS, blogs, ecommerce, or something else. You might even find someone in the PHP community willing to help you.The second aspect of resources refers to the sheer number of PHP developers on the market.

**Hypertext Markup Language**

Hypertext Markup Language (HTML) is a language for describing how pages of text, graphics and other information are organized, formatted, and linked together. It is not really a programming language in the sense of COBOL or Visual Basic, but it does provide powerful capabilities for making text documents readily available on the Internet. Basic graphic support was added, and the things started to take off. Now we have sound, live video, retail catalogs, and much, more available to us.

HTML pages are the standard interface to the Internet. This basic language provides the necessary nuts and bolts for building Web pages.

HTML stands for Hypertext Markup Language and was developed specifically for use on the system of links, which can be non sequential in order. HTML is not a programming language.It is a marking up language used for linking one piece of information to another. Using a markup language means that tags can be added to the words in the document and web enable them. A tag is a set of descriptive formatting codes used in HTML document that instructs a web browser how to display text and graphics on a web page.

**Introduction to React**

React is a [JavaScript](https://flaviocopes.com/javascript/) library that aims to simplify development of visual interfaces.

Developed at Facebook and released to the world in 2013, it drives some of the most widely used code in the world, powering Facebook and Instagram among many, many other software companies.

Its primary goal is to make it easy to reason about an interface and its state in any point in time, by dividing the UI into a collection of components.

React is used to build single-page web applications, among with many other libraries and frameworks that were available before React came into life.

## React components:

### Custom components

There are 2 ways to define a component in React.

A stateless component does not manage internal state, and is just a function:

A stateful component is a class, which manages state in its own properties:

As they stand, they are equivalent because there is no state management yet (coming in the next couple articles).

There is a third syntax which uses the ES5 / ES2015 syntax, without the classes:

Props is how Components get their properties. Starting from the top component, every child component gets its props from the parent. In a stateless component, props is all it gets passed, and they are available by adding props as the function argument:

In a stateful component, props are passed by default. There is no need to add anything special, and they are accessible as this.props in a Component instance.

Passing props down to child components is a great way to pass values around in your application. A component either holds data (has state) or receives data through its props.

It gets complicated when:

* you need to access the state of a component from a child that’s several levels down (all the previous children needs to act as a pass-through, even if they do not need to know the state, complicating things)
* you need to access the state of a component from a completely unrelated component.

[Redux](https://flaviocopes.com/redux/) was traditionally very popular for this, and this is the reason it’s included in many tutorials.

Recently React (in version 16.3.0) introduced the **Context API**, which makes Redux redundant for this simple use case.

We talk about the Context API later in this guide.

Redux is still useful if you:

* need to move your data outside of the app altogetherm for some reason
* create complex reducers and actions to manipulate the data in any way you want

but it’s no more “required” for any React application.

### Fragment

Notice how I wrapped the return values in a div. This is because a component can only return one single element, and if you want more than one, you need to wrap it into another container tag.

This however causes an unnecessary div in the output. You can avoid this by using React. Fragment:

which also has a very nice shorthand syntax <> </> that is supported only in recent releases (and Babel 7+):

### Protypes

Since JavaScript is a dynamically typed language, we don’t really have a way to enforce the type of a variable at compile time, and if we pass invalid types, they will fail at runtime or give weird results if the types are compatible but not what we expect.

Flow and TypeScript help a lot, but React has a way to directly help with props types, and even before running the code, our tools (editors, linters) can detect when we are passing the wrong values:

**CSS-Cascading Style Sheets:**

Cascading Style Sheets (CSS) is a [style sheet language](http://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [look and formatting](http://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](http://en.wikipedia.org/wiki/Markup_language). While most often used to style [web pages](http://en.wikipedia.org/wiki/Web_page) and [interfaces](http://en.wikipedia.org/wiki/Interface_%28computing%29) written in [HTML](http://en.wikipedia.org/wiki/HTML) and [XHTML](http://en.wikipedia.org/wiki/XHTML), the language can be applied to any kind of [XML](http://en.wikipedia.org/wiki/XML) document, including [plain XML](http://en.wikipedia.org/wiki/Plain_Old_XML),SVG and XUL. CSS is a cornerstone specification of [the web b](http://en.wikipedia.org/wiki/The_web)and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the [layout](http://en.wikipedia.org/wiki/Page_layout), [colors](http://en.wikipedia.org/wiki/Color), and [fonts](http://en.wikipedia.org/wiki/Typeface). This separation can improve content [accessibility](http://en.wikipedia.org/wiki/Accessibility), provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for [table less web design](http://en.wikipedia.org/wiki/Tableless_web_design)).

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or [screen reader](http://en.wikipedia.org/wiki/Screen_reader)) and on [Braille-based](http://en.wikipedia.org/wiki/Braille_display), tactile devi

ces. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called *cascade*, priorities or *weights* are calculated and assigned to rules, so that the results are predictable.

**JQuery:**

jQuery is a [cross-platform](http://en.wikipedia.org/wiki/Cross-platform)[JavaScript library](http://en.wikipedia.org/wiki/JavaScript_library) designed to simplify the [client-side scripting](http://en.wikipedia.org/wiki/Client-side_scripting) of [HTML](http://en.wikipedia.org/wiki/HTML).It was released in January 2006 at [Bar Camp](http://en.wikipedia.org/wiki/BarCamp)NYC by [John Resig](http://en.wikipedia.org/wiki/John_Resig). It is currently developed by a team of developers led by Dave Methvin. Used by over 80% of the 10,000 most visited websites, jQuery is the most popular [JavaScript library](http://en.wikipedia.org/wiki/JavaScript_library) in use today.

jQuery is [free, open source software](http://en.wikipedia.org/wiki/Free_and_open_source_software), licensed under the [MIT License](http://en.wikipedia.org/wiki/MIT_License).JQuery's syntax is designed to make it easier to navigate a document, select [DOM](http://en.wikipedia.org/wiki/Document_Object_Model) elements, create [animations](http://en.wikipedia.org/wiki/Animation), handle [events](http://en.wikipedia.org/wiki/Event_%28computing%29), and develop [Ajax applications](http://en.wikipedia.org/wiki/Ajax_%28programming%29). jQuery also provides capabilities for developers to create [plug-ins](http://en.wikipedia.org/wiki/Plug-in_%28computing%29) on top of the JavaScript library. This enables developers to create [abstractions](http://en.wikipedia.org/wiki/Abstraction_%28computer_science%29) for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful [dynamic web pages](http://en.wikipedia.org/wiki/Dynamic_web_page) and web applications.

**JavaScript:**

JavaScript (JS) is a [dynamic](http://en.wikipedia.org/wiki/Dynamic_programming_language) computer [programming language](http://en.wikipedia.org/wiki/Programming_language). It is most commonly used as part of [web browsers](http://en.wikipedia.org/wiki/Web_browser), whose implementations allow [client-side scripts](http://en.wikipedia.org/wiki/Client-side_scripting) to [interact with the user](http://en.wikipedia.org/wiki/User_interface), control the browser, communicate [asynchronously](http://en.wikipedia.org/wiki/Ajax_%28programming%29), and alter the [document content](http://en.wikipedia.org/wiki/Document_Object_Model) that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications.

JavaScript is a [prototype-based](http://en.wikipedia.org/wiki/Prototype-based_programming)[scripting language](http://en.wikipedia.org/wiki/Scripting_language) with [dynamic](http://en.wikipedia.org/wiki/Dynamic_language) typing and has [first-class functions](http://en.wikipedia.org/wiki/First-class_functions). Its [syntax](http://en.wikipedia.org/wiki/JavaScript_syntax) was influenced by [C](http://en.wikipedia.org/wiki/C_%28programming_language%29). JavaScript copies many names and naming conventions from [Java](http://en.wikipedia.org/wiki/Java_%28programming_language%29), but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the [Self](http://en.wikipedia.org/wiki/Self_%28programming_language%29) and [Scheme](http://en.wikipedia.org/wiki/Scheme_%28programming_language%29) programming languages. It is a [multi-paradigm](http://en.wikipedia.org/wiki/Multi-paradigm) language, supporting [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming), [imperative](http://en.wikipedia.org/wiki/Imperative_programming), and [functional](http://en.wikipedia.org/wiki/Functional_programming) programming styles.

The application of JavaScript in use outside of web pages—for example, in [PDF](http://en.wikipedia.org/wiki/Portable_Document_Format) documents, [site-specific browsers](http://en.wikipedia.org/wiki/Site-specific_browser), and [desktop widgets](http://en.wikipedia.org/wiki/Desktop_widget)—is also significant.

**MySQL**

MYSQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. It is named after developer [Michael Widenius](http://en.wikipedia.org/wiki/Michael_Widenius) daughter, My. The [SQL](http://en.wikipedia.org/wiki/SQL) phrase stands for Structured Query Language.

The MySQL development project has made its [source code](http://en.wikipedia.org/wiki/Source_code) available under the terms of the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License), as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality.

**Uses**

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used [LAMP](http://en.wikipedia.org/wiki/LAMP_(software_bundle)) web application software stack—LAMP is an acronym for "[Linux](http://en.wikipedia.org/wiki/Linux), [Apache](http://en.wikipedia.org/wiki/Apache_HTTP_Server), MySQL, PHP ".

MySQL is used in some of the most frequently visited web sites on the Internet.

**SYSTEM DESIGN**

**Introduction**

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

The main objective of the system design is to make the system user friendly. System design involves various stages as:

* Data Entry
* Data Correction
* Data Deletion
* Processing

System design is the creative act of invention, developing new inputs, a database, offline files, procedures and output for processing business to meet an organization objective. System design builds information gathered during the system analysis.

**Input Design**

The input design is the link between the information system and the user. It comprises developing specification and procedure for data preparation and those steps that are necessary to put transaction data into a usable form for processing data entry. The design of inputs focuses on controlling the amount of inputs required, controlling errors, avoiding delay, avoiding extra steps and keeping the process simple.

**Output Design**

In output design, emphasis is given on producing a hard copy of the information required as the output on the CRT screen in some predefined manner. Computer output is the most important and direct source of information to the use.

Output design is a process that involves designing necessary outputs that should be given to various users according to their requirements. Efficient, intelligent output design should improve the system’s relationship with the user and help in decision making

**Process Design**

When process of large volumes of data being handled, it is important that the item be identified, stored or selected easily and quickly. To accomplish this, each data item must have unique identification and must be related to other items of the same type. Codes can provide brief identification of each item, which replace longer description that would be more awkward to store and manipulate.

Common types of codes are:

* Sequences codes
* Significant Digit code
* Alphabetic code
* Self Checking code

**Database Design**

Database design is important to understand the underlying structure of the system. Database design can make a simpler application complex as well as simpler .It refers to the study of processes of database and the data storage structure.

The main objective of the database design is to make the system user friendly. Database design involves various stages as:

* Storage structure access
* Database Authentication
* Data Entry
* Data Correction
* Data Editing
* Data Processing
* Report Generation

**TESTING**

**Software testing**

Software testing is the process used to measure the [quality](http://en.wikipedia.org/wiki/Software_quality) of developed [computer software](http://en.wikipedia.org/wiki/Computer_software). Usually, quality is constrained to such topics as [correctness](http://en.wikipedia.org/wiki/Correctness), completeness, [security](http://en.wikipedia.org/wiki/Computer_security_audit), but can also include more technical requirements as described under the [ISO](http://en.wikipedia.org/wiki/International_Organization_for_Standardization) standard [ISO 9126](http://en.wikipedia.org/wiki/ISO_9126), such as capability, [reliability](http://en.wikipedia.org/wiki/Reliability), [efficiency](http://en.wikipedia.org/wiki/Algorithmic_efficiency), [portability](http://en.wikipedia.org/wiki/Porting), [maintainability](http://en.wikipedia.org/wiki/Maintainability), compatibility, and [usability](http://en.wikipedia.org/wiki/Usability). Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate.

White box, black box, and grey box testing

[White box](http://en.wikipedia.org/wiki/White_box_testing) and [black box testing](http://en.wikipedia.org/wiki/Black_box_testing)are terms used to describe the point of view that a test engineer takes when designing test cases. Black box testing treats the software as a black-box without any understanding as to how the internals behave. Thus, the tester inputs data and only sees the output from the test object. This level of testing usually requires thorough test cases to be provided to the tester who then can simply verify that for a given input, the output value (or behavior), is the same as the expected value specified in the test case. White box testing, however, is when the tester has access to the internal data structures, code, and algorithms. For this reason, [unittesting](http://en.wikipedia.org/wiki/Unit_testing) and debugging can be classified as white-box testing and it usually requires writing code, or at a minimum, stepping through it, and thus requires more skill than the black-box tester. If the software in test is an interface or API of any sort, white-box testing is almost always required.

**IMPLEMENTATION PHASE**

The implementation phase involves installing approved applications into production environments. Primary tasks include announcing the implementation schedule, training end users, and installing the product. Additionally, organizations should input and verify data, configure and test system and security parameters, and conduct post-implementation reviews. Management should circulate implementation schedules to all affected parties and should notify users of any implementation responsibilities.

After organizations install a product, pre-existing data is manually input or electronically transferred to a new system. Verifying the accuracy of the input data and security configurations is a critical part of the implementation process. Organizations often run a new system in parallel with an old system until they verify the accuracy and reliability of the new system. Employees should document any programming, procedural, or configuration changes made during the verification process.

**PROJECT DETAILS**

**INTRODUCTION**:

The aim of developing Online Food Ordering system project is to replace the traditional way of taking orders with computerized system. Another important reason for developing this project is to prepare order summary reports quickly and in correct format at any point of time when required.

Online Food Ordering System has a very lot of scope. This PHP project can be used by any restaurants or fast foods for customers for keeping their order records. This project is easy, fast and accurate. It requires less disk space. Online Food Ordering System uses MYSQL Server as backend so there is not any chance of data loss or data security.

Modules of food delivery in php

* Admin
* Users
* Guest users

**Users—**

* User can register yourself.
* User can login with valid email and password.
* Forgot Password(user Can recover own password)
* About the fast food company
* The fast food and the services offered there
* change Password

**Admin**–

* Admin can create Package
* Online purchase
* Type of food provided.
* Change Password
* Admin Dashboard

**Guest users**

* Visit the Website
* Guest user can enquiry

**Project Introduction:**

1.1 BACKGROUND OF STUDY

The online food delivery system is one of the latest servicers most fast food restaurants in the western world are adopting. With this method, food is orderedonline and delivered to the customer. This is made possible through the use of electronic payment system. Customers pay with their credit cards, although credit card customers can be served even before they make payment either through cashor cheque. So, the system designed in this project will enable customers go onlineand place order for their food.Due to the great increase in the awareness of internet and the technologiesassociated with it, several opportunities are coming up on the web. So manybusinesses and companies now venture into their business with ease because of theinternet. One of such business that the internet introduced is an online food

ordering system. In today’s age of fast food and take out, many restaurants havechosen to focus on quick preparation and speedy delivery of orders rather thanoffering a rich dining experience. Until recently, most of this delivery orders wereplaced over the phone, but there are many disadvantages to this system.It is possible for anybody to order any goods via the internet and have the goodsdelivered at his/her doorsteps. But while trying to discuss the transfer method ofthe goods and services, attention is focused on the payment mode. In other words,how possible is it to pay for goods and services via the internet? This then leads tothe discussion of the economic consequences of digital cash.

1.2 STATEMENT OF PROBLEM

As industries are fast expanding, people are seeking for more ways to purchaseproducts with much ease and still maintain cost effectiveness. The vendors need topurchase the products in order to sell to end users. The manual method of going totheir local food sales outlets to purchase food is becoming obsolete and moretasking. Food can be ordered through the internet and payment made without goingto the restaurant or the food vendor. So there is need for a wide range of publicityand enabling direct order, processing and delivering of food through online system.

1.3 OBJECTIVES OF STUDY

1.The home page of this web interfile provides an avenue where customerswill be able to gather more and reliable information about what the fast foodindustry really does.

2.The products and services offered would provide the customers with all thedifferent categories of available products that they can choose and selectfrom.

3.This will provide a user friendly environment between the customer andemployee thus increasing the efficiency of the food ordering system.

4.There will also be an online purchase form with which valued customerswill be using to get in touch with any of their request whenever the needarises.

5.It will also help for easy retrieval of orders made by the customers.

**1.5 SIGNIFICANCE OF STUDY**

In view of the rapid development of computer technology in almost all the fieldsof operation and its use in relation to information management, it has becomeimportant to look into the development of online ordering system for firms to meetup with demands of the customers. Therefore, the food ordering and deliverysystem will help customers and management to:

1.Advertise available foods in their company

2.Reduce the workload in the present system.

3.Reduce time wasted in data processing.

4.Create a platform for online purchase and delivery of fast food.

5.Keep accurate record on purchased order and delivery.

**1.6 LIMITATIONS**

Due to time and financial constraints, the software that is developed covers onlythe aspect of food ordering and payments.

**1.7 DEFINITION OF TERMSFOOD:**

 Any nutritious substance that people or animals eat or drink, or that plantabsorbs, in order to maintain life and growth.

**ONLINE FOOD ORDERING:**Online food ordering services are websites thatfeature interactive menus allowing customers to place orders with local restaurantsand food cooperatives.

A CREDIT CARD:is a[payment card](http://en.wikipedia.org/wiki/Payment_card)issued to users as a system of[payment.](http://en.wikipedia.org/wiki/Payment) It allows the cardholder to pay for goods and services based on theholder's promise to pay for them.

ORDERING SYSTEM:This is referred to as a set of detailed methods that isbeing used in handling the ordering process.

RESTAURANT: (eating place) is a place where meals and drinks are sold andserved to customers.

CUSTOMER:Sometimes known as a client, buyer, or purchaser) is the recipientof goods, services, products or idea obtained from a seller, vendor, or supplier for amonetary or other valuable consideration.

TECHNOLOGY: It is the study of techniques or process of mobilizing resources(such as information) for accomplishing objectives that benefit man and hisenvironment.

HAMBURGERS:A hamburger is a sandwich consisting of a cooked patty ofground meat usually placed inside a sliced hamburger bun.

**Images:**

Breakfast

****Burger

****

Cupcake With Coffee

Fruit salad

Indian Thali

Indian Maggie

Samosa

Special Pizza

**CODING**

**Header**

<div class="header">Online Food Delivery System</div>

**Home**

<html>

<head>

<title>Home</title>

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

</head>

<body>

<?php

session\_start();

if(!isset($\_SESSION['Username'])){

header("Location:"."index.php");

}

?>

<?php

include('header.php');

?>

<?php

$conn=mysqli\_connect('localhost','root','',"food\_delivery");

if(!$conn){

die("Error connecting to the database!");

}

?>

<?php echo "<H1 ALIGN='center'>Welcome To Your User Dashboard"." ".$\_SESSION['Username']."!</H1>";?>

<table border="1" cellpadding="5px" cellspacing="2" align="center" width="500px" width="1000px" >

<tr>

<th>Sr. No.</th>

<th>Name</th>

<th>Image</th>

<th>Description</th>

<th>Price</th>

<th>Quantity</th>

<th>Action</th>

</tr>

<?php

$query="select \* from products";

$data=$conn->query($query);

if($data->num\_rows>0){

while($row=$data->fetch\_assoc()){

echo"<tr>";

echo"<td>";

echo $row['id']."<br>";

echo"</td>";

echo"<td>";

echo $row['product\_name']."<br>";

echo"</td>";

echo"<td>";

echo"<img src='images/$row[product\_image]'height='100px' width='100px'>";

echo"</td>";

echo"<td>";

echo $row['product\_description'];

echo"</td>";

echo"<td>";

echo "&#x20b9;".$row['product\_price'];

echo"</td>";

?>

<td>

<form action="confirm\_order.php" method="post">

<input type="hidden" value="<?php echo $row['product\_name'];?>" style="width:50px" name="hidden\_name">

<input type="hidden" value="<?php echo $row['product\_price']; ?>" style="width:50px" name="hidden\_price">

<input type="hidden" value="<?php echo $\_SESSION['Username']; ?>" style="width:50px" name="username">

<input type='hidden' value="<?php echo $row['product\_image']; ?>" style="width:50px" name="hidden\_image">

<input type="text" value="1" style="width:50px" name="qty" id="qty" required>

</td>

<td>

<input type="submit" value="Place Order" name="confirm\_order">

</form>

</td>

<?php

echo"</tr>";

}

}

?>

</table>

<?php

include\_once('sidebar.php');

?>

</body>

</html>

**Index**

<html>

<head>

<title>Online Food Delivery System</title>

<link rel="stylesheet" type="text/css" href="css\style.css">

<script type="text/javascript" src="js/validation.js"></script>

</head>

<body background="images/images3.jpg">

<?php

include('header.php') ;

?>

<?php

session\_start();

if(isset($\_POST['login'])){

$conn=mysqli\_connect('localhost','root','',"food\_delivery");

if(!$conn){

die("Error Connecting to the database!");

}

$password=sha1($\_POST['password']);

$sql="SELECT \* FROM users WHERE username='$\_POST[username]' and password='$password'";

$result=mysqli\_query($conn,$sql);

$login=mysqli\_num\_rows($result);

if($login>0){

$\_SESSION['Username']=$\_POST['username'];

header("Location:"."home.php");

}

else{

echo"<center><font color='red'>Incorrect username or password.</font></center>";

}

}

?>

<br><br><br>

<div id="title">

<span class="title">

#1 Sign Up

</span>

Sign up for free by giving the right Credentials

</div>

<div id="title">

<span class="title">

#2 Log in

</span>

Log in with the username and paasword

</div>

<div id="title">

<span class="title">

#3 Set Profile

</span>

user can change the profile and payment options once logged in

</div><div id="title">

<span class="title">

#4 Place order

</span>

Found something tasty,then place your order

</div>

<div id="content">

<div class="login">

<form action="" method="post" onsubmit="return valid()">

<span id="userlogin"><label>User Login</label></span>

<br><br><br>

&nbsp;&nbsp;<label>Username</label>

<input type="text" id="user" name="username" placeholder="Username"><br>

<br><br>

&nbsp;&nbsp;<label>Password</label>

<input type="password" name="password" id="pass" placeholder="Password">&nbsp;&nbsp;<input type="button"value="Show/Hide"id="button"onClick="showhideuserpass(this.form)"><br>

<br>

&nbsp;&nbsp;

<input type="submit"value="Login"id="submit"name="login" id="login">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

&nbsp;&nbsp;&nbsp;&nbsp;

<a href="register.php">Create an account</a>

</form>

</div>

<?php

if(isset($\_POST['admin'])){

$conn=mysqli\_connect('localhost','root','',"food\_delivery");

if(!$conn){

die("Error Connecting to the database!");

}

$sql="SELECT \* FROM admin WHERE username='$\_POST[username]' and password='$\_POST[password]'";

$result=mysqli\_query($conn,$sql);

$login=mysqli\_num\_rows($result);

if($login>0){

$\_SESSION['UserName']=$\_POST['username'];

header("Location:"."admindashboard.php");

}

else{

echo"<center><font color='red'>Incorrect username or password.</font></center>";

}

}

?>

<div class="adminlogin">

<form action="" method="post" onsubmit="return validadmin()">

<span id="adminlogin"><label>Admin Login</label></span>

<br><br><br>

&nbsp;&nbsp;<label>Username</label>

<input type="text"id="adminuser" name="username"placeholder="Username"><br>

<br><br>

&nbsp;&nbsp;&nbsp;<label>Password</label>

<input type="password"name="password"id="adminpass"placeholder="Password">&nbsp;&nbsp;<input type="button"value="Show/Hide"id="button"onClick="showhideadminpass(this.form)"><br>

<br>

&nbsp;&nbsp;<input type="submit"value="Login"id="submit"name="admin"><br>

</form>

</div>

</div>

</body>

</html>

**Edit profile**

<html>

<head>

<title>User Profile</title>

<script src="js/validation.js"></script>

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

</head>

<?php

session\_start();

if(!isset($\_SESSION['Username'])){

header("Location:"."index.php");

}

?>

<?php

include\_once('header.php');

?>

<br>

<?php

$user=$\_SESSION['Username'];

if(isset($\_POST['update\_profile'])){

$conn=mysqli\_connect('localhost','root','',"food\_delivery");

if(!$conn){

die("Error Connecting to the database!");

}

$sql="UPDATE users SET firstname='$\_POST[firstname]', lastname='$\_POST[lastname]', mobile='$\_POST[mobile]', address='$\_POST[address]', email='$\_POST[email]' WHERE username='$user'";

$result=mysqli\_query($conn,$sql);

if($result){

echo"<script type='text/javascript'>alert('Your information was updated successfully.')</script>";

}

}

?>

<?php

$user=$\_SESSION['Username'];

$conn=mysqli\_connect('localhost','root','',"food\_delivery");

if(!$conn){

die("Error Connecting to the database!");

}

$sql="SELECT \* FROM users WHERE username='$user'";

$result=mysqli\_query($conn,$sql);

if($sql){

while($row=mysqli\_fetch\_array($result,MYSQLI\_ASSOC)){

?>

<center>

<div class="content">

<div class="register">

<form action="" method="post" onsubmit="return formvalidation()">

<span id="registeruser"><label>Edit Profile </label></span><br><br><br>

<label>Firstname</label><br>

<input type="text"id="fname" name="firstname" placeholder="First Name" value="<?php echo $row['firstname'];?>" required><br>

<br>

<label>Lastname</label><br>

<input type="text"id="lname" name="lastname" placeholder="Last Name" value="<?php echo $row['lastname'];?>" required><br>

<br>

<label>Mobile</label><br>

<input type="number" id="mobile" name="mobile" placeholder="Mobile" value="<?php echo $row['mobile']; ?>" required><br>

<br>

<label>Address</label><br>

<textarea id="address" name="address" placeholder="Address" rows="5" cols="50" required><?php echo $row['address']; ?></textarea><br>

<br>

<label>Email</label><br>

<input type="email" id="email" name="email" placeholder="Email" value="<?php echo $row['email']; ?>" required><br>

<br>

<input type="submit" value="Save Changes" id="submit" name="update\_profile"><br>

</form>

</div>

</center>

</div>

<?php

break;

}

}

if(!$row){

echo"<font color='red'>Username and password don't match.</font>";

}

?>

<?php

include\_once('sidebar.php');

?>

**Delete user**

<?php

$id=$\_GET['id'];

$conn=mysqli\_connect('localhost','root','');

if(!$conn){

die("Error connecting to the database!");

}

mysqli\_select\_db($conn,"food\_delivery");

$sql="DELETE FROM products where id='$id'";

if(mysqli\_query($conn,$sql)){

header("Location:"."product\_listing.php");

}

?>

**Delete order**

<?php

$id=$\_GET['id'];

$conn=mysqli\_connect('localhost','root','');

if(!$conn){

die("Error connecting to the database!");

}

mysqli\_select\_db($conn,"food\_delivery");

$sql="DELETE FROM products where id='$id'";

if(mysqli\_query($conn,$sql)){

header("Location:"."product\_listing.php");

}

?>

**Cancel order**

<?php

$id=$\_GET['id'];

$conn=mysqli\_connect('localhost','root','');

if(!$conn){

die("Error connecting to the database!");

}

mysqli\_select\_db($conn,"food\_delivery");

$sql="DELETE FROM orders where id='$id'";

if(mysqli\_query($conn,$sql)){

header("Location:"."orders.php");

}

?>

**Admin signout**

<?php

session\_start();

session\_unset();

header("Location:"."index.php");

?>

**Order**

<html>

<head>

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

</head>

<body>

<?php

session\_start();

if(!isset($\_SESSION['Username'])){

header("Location:"."index.php");

}

?>

<?php

include('header.php');

?>

<h1 align="center">My Orders</h1>

<br>

<?php

$user=$\_SESSION['Username'];

$conn=mysqli\_connect('localhost','root','',"food\_delivery");

if(!$conn){

die("Error Connecting to the database!");

}

?>

<center>

<table border="2" cellspacing="0" align="center" width="500px" >

<tr>

<th>Product Name</th>

<th>Product Price</th>

<th>Product Image</th>

<th>Product Quantity</th>

<th>Total Cost</th>

<th>Action</th>

</tr>

<?php

$sql="SELECT \* FROM orders WHERE username='$user'";

$result=mysqli\_query($conn,$sql);

if($sql){

while($row=mysqli\_fetch\_array($result,MYSQLI\_ASSOC)){

echo"<tr>";

echo"<td>";

echo $row['hidden\_name']."<br>";

echo"</td>";

echo"<td>";

echo "&#X20B9;".$row['hidden\_price']."<br>";

echo"</td>";

echo"<td>";

echo "<img src='images/$row[hidden\_image]' height='100px' width='100px'>"."<br>";

echo"</td>";

echo"<td>";

echo $row['qty'],"<br>";

echo"</td>";

echo"<td>";

echo "&#X20B9;".$row['total'],"<br>";

echo"</td>";

echo"<td>";

echo"<a href='cancel\_orders.php?id=$row[id]'>Cancel Order</a>";

echo"</td>";

echo"</tr>";

}

}

?>

</table>

<br>

<form action="home.php" method="home.php">

<input type="submit" value="Continue Shopping" style="width:150px">

</form>

</center>

<?php

include('sidebar.php');

?>

</body>

</html>

**Profile**

<html>

<head>

<title>User Profile</title>

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

</head>

<?php

session\_start();

if(!isset($\_SESSION['Username'])){

header("Location:"."index.php");

}

?>

<?php

include\_once('header.php');

?>

<br>

<h1 align="center">Personal Information</h1>

<br>

<?php

$user=$\_SESSION['Username'];

$conn=mysqli\_connect('localhost','root','',"food\_delivery");

if(!$conn){

die("Error Connecting to the database!");

}

?>

<center>

<table border="2" cellspacing="0" align="center" width="500px" >

<tr>

<th>Name</th>

<th>Mobile</th>

<th>Address</th>

<th>Gender</th>

<th>Email</th>

<th>Username</th>

</tr>

<?php

$sql="SELECT \* FROM users WHERE username='$user'";

$result=mysqli\_query($conn,$sql);

if($sql){

while($row=mysqli\_fetch\_array($result,MYSQLI\_ASSOC)){

echo"<tr>";

echo"<td>";

echo $row['firstname']," ", $row['lastname'],"<br>";

echo"</td>";

echo"<td>";

echo $row['mobile'],"<br>";

echo"</td>";

echo"<td>";

echo $row['address'],"<br>";

echo"</td>";

echo"<td>";

echo $row['gender'],"<br>";

echo"</td>";

echo"<td>";

echo $row['email'],"<br>";

echo"</td>";

echo"<td>";

echo $\_SESSION['Username'],"<br>";

echo"</td>";

echo"</tr>";

break;

}

}

if(!$row){

echo"<font color='red'>Username and password don't match.</font>";

}

?>

</table>

</center>

<?php

include\_once('sidebar.php');

?>

**Payment info**

<html>

<head>

<title>Payment Information</title>

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

</head>

<body>

<?php

session\_start();

if(!isset($\_SESSION['Username']))

{

header("Location:"."index.php");

}

?>

<?php

include\_once('header.php');

?>

<br>

<h1 align="center">Payment Information</h1>

<br>

<?php

$user=$\_SESSION['Username'];

$conn=mysqli\_connect('localhost','root','',"food\_delivery");

if(!$conn){

die("Error Connecting to the database!");

}

?>

<center>

<table border="2" cellspacing="0" align="center" width="500px">

<tr>

<th>Aadhaar Number</th>

<th>Bank</th>

<th>Account Number</th>

<?php

$sql="SELECT \* FROM users WHERE username='$user'";

$result=mysqli\_query($conn,$sql);

if($sql){

while($row=mysqli\_fetch\_array($result,MYSQLI\_ASSOC)){

echo"<tr>";

echo"<td>";

echo $row['aadhaar\_number']."<br>";

echo"</td>";

echo"<td>";

echo $row['bank'],"<br>";

echo"</td>";

echo"<td>";

echo $row['account\_number'],"<br>";

echo"</td>";

echo"</tr>";

break;

}

}

if(!$row){

echo"<font color='red'>Username and password don't match.</font>";

}

?>

</table>

</center>

<?php

include\_once('sidebar.php');

?>

**User logout**

<html>

<head>

<title>User Logout</title>

</head>

<body>

<?php

session\_start();

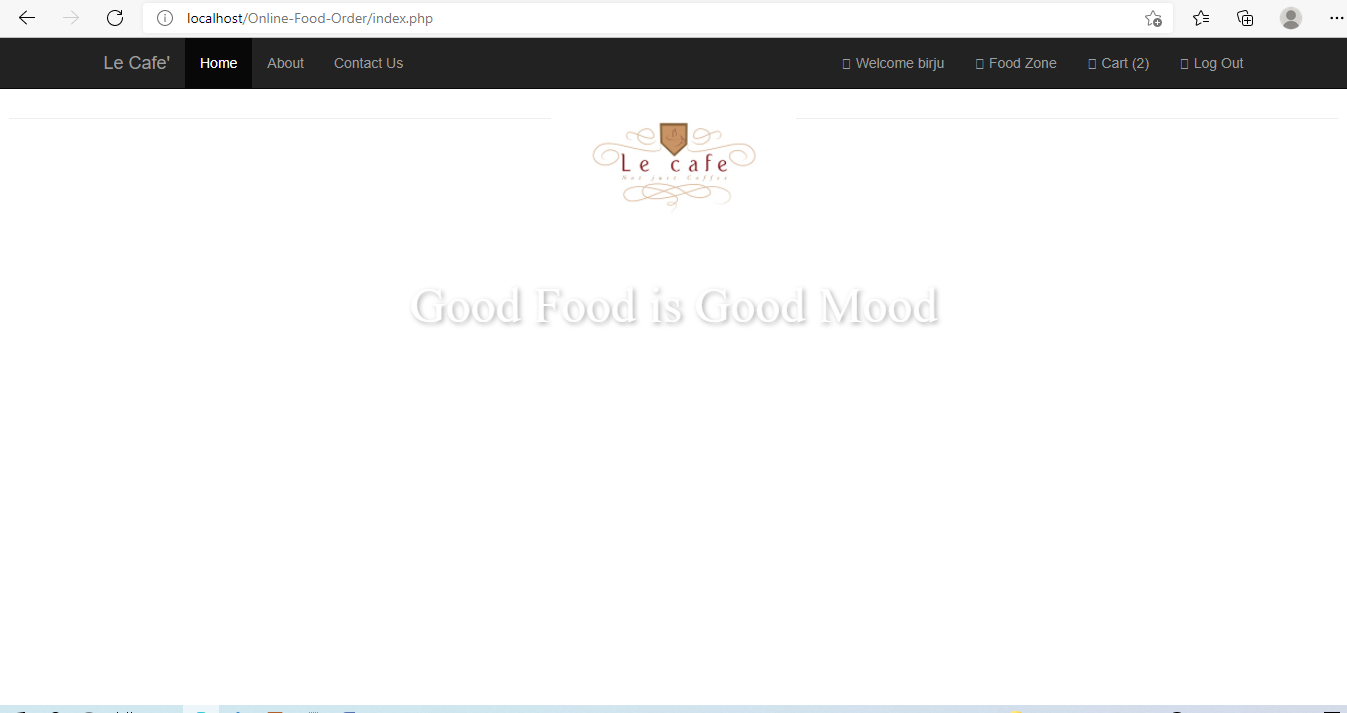
session\_unset();

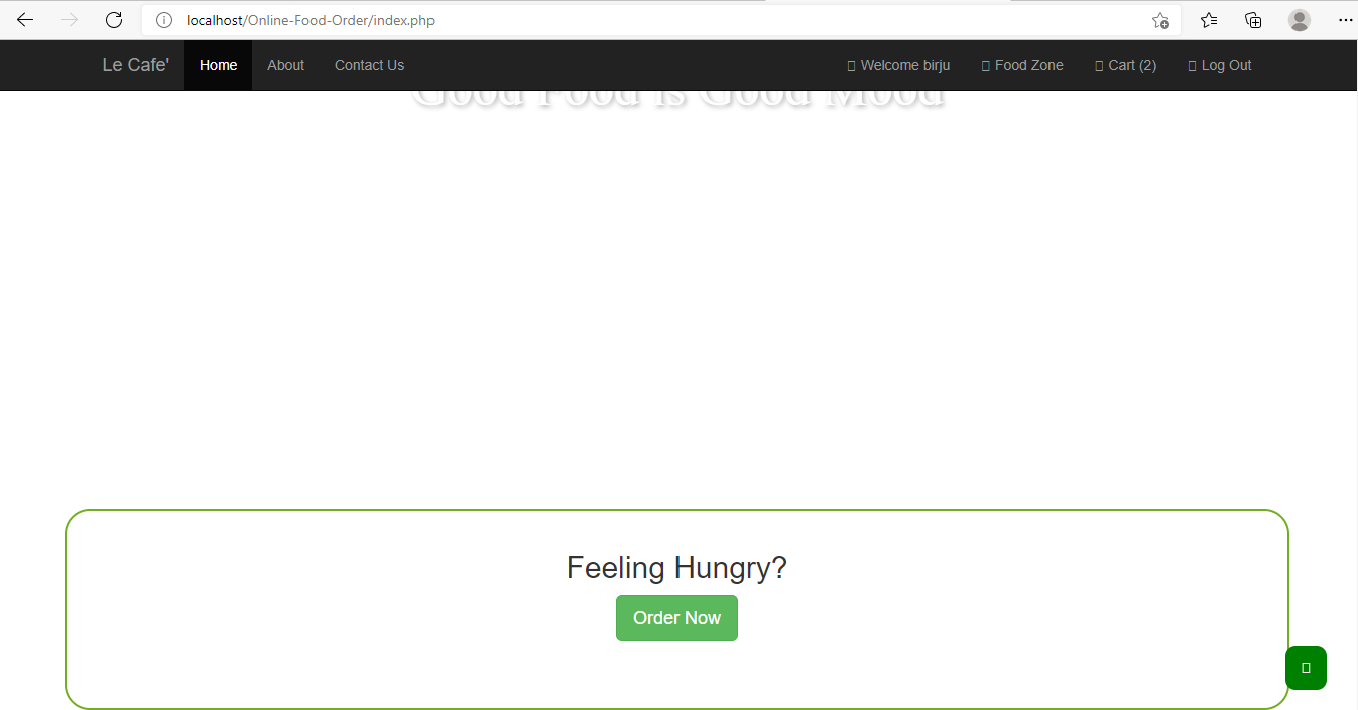
header("Location:"."index.php");

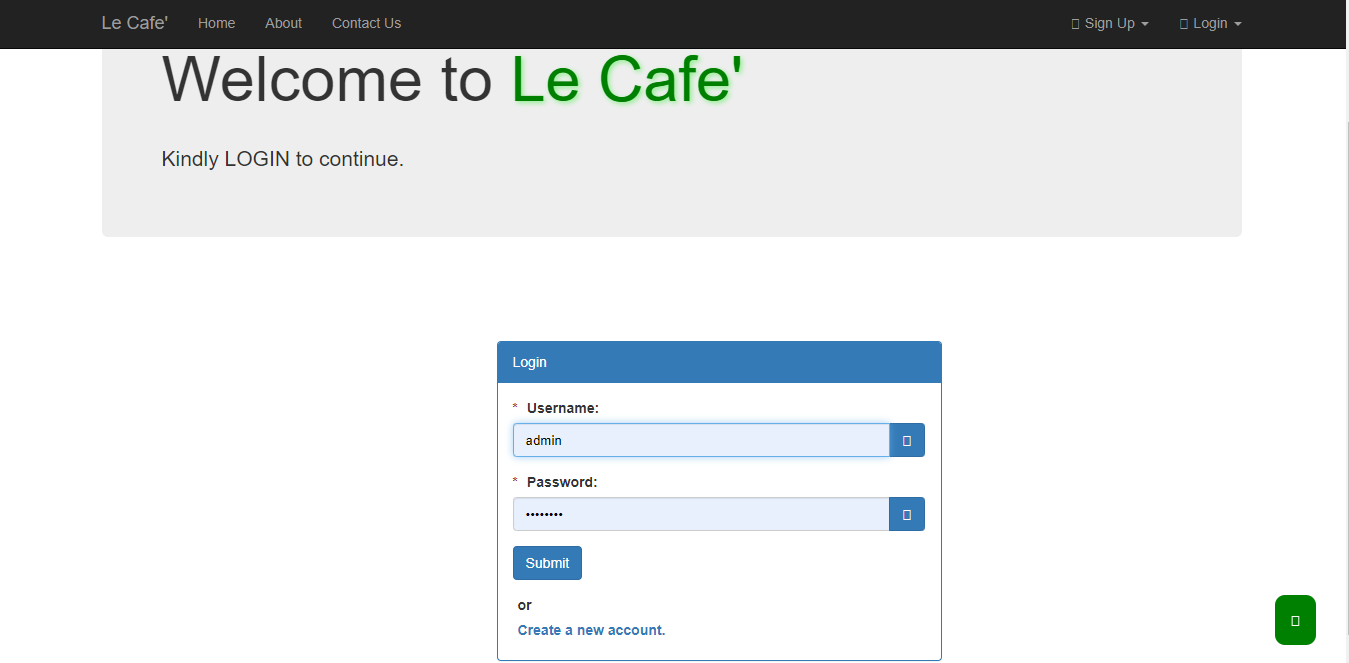
?>

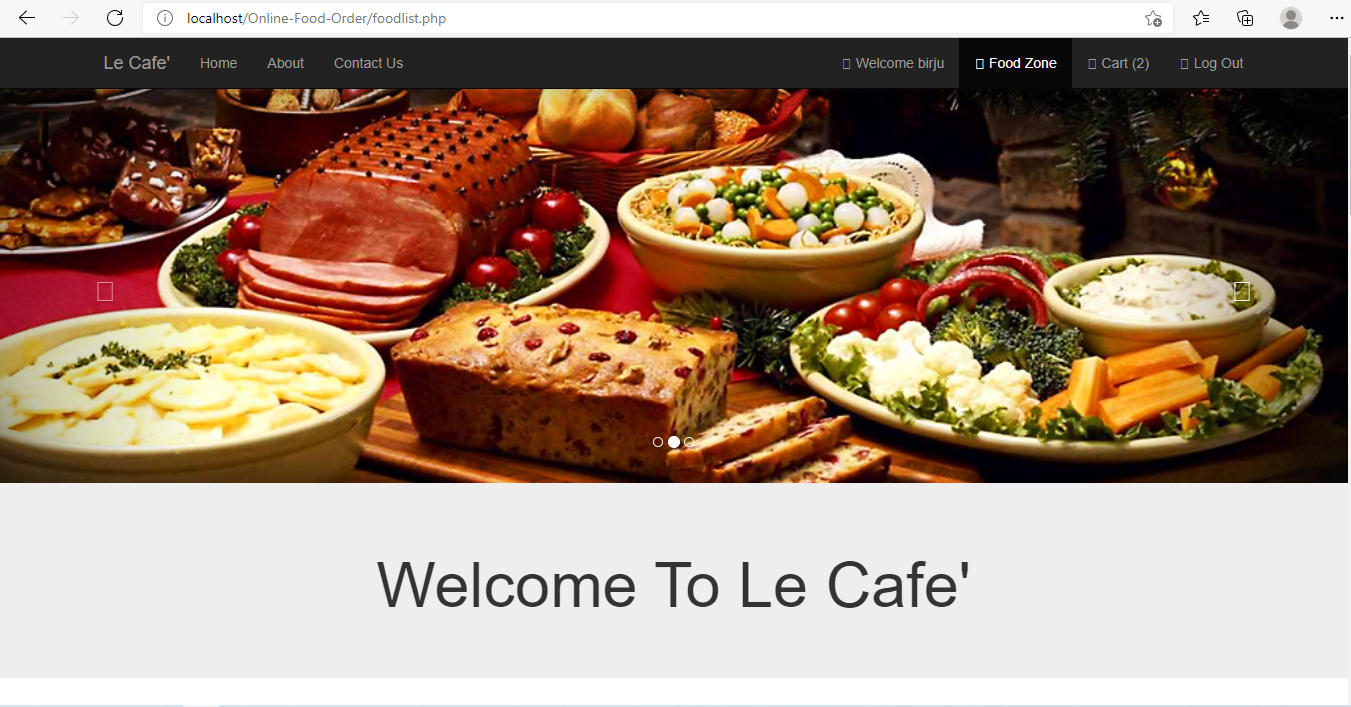
</body>

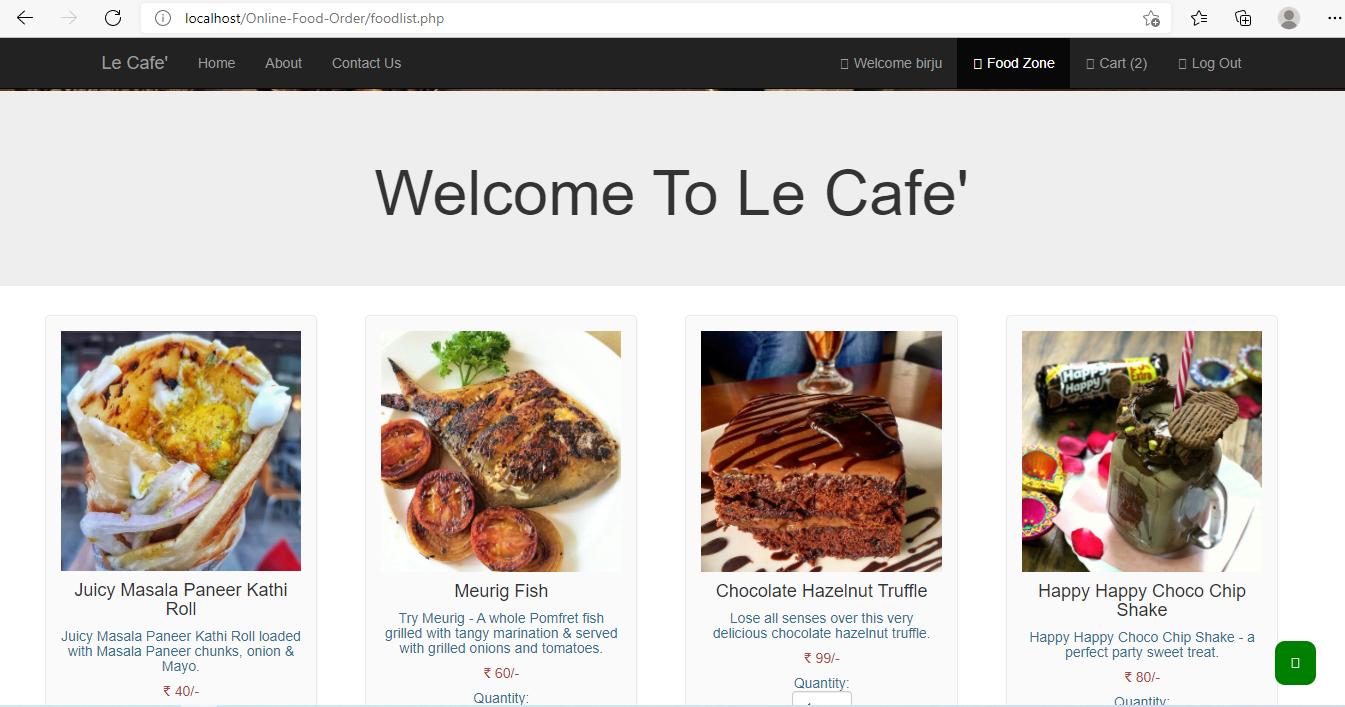
</html>

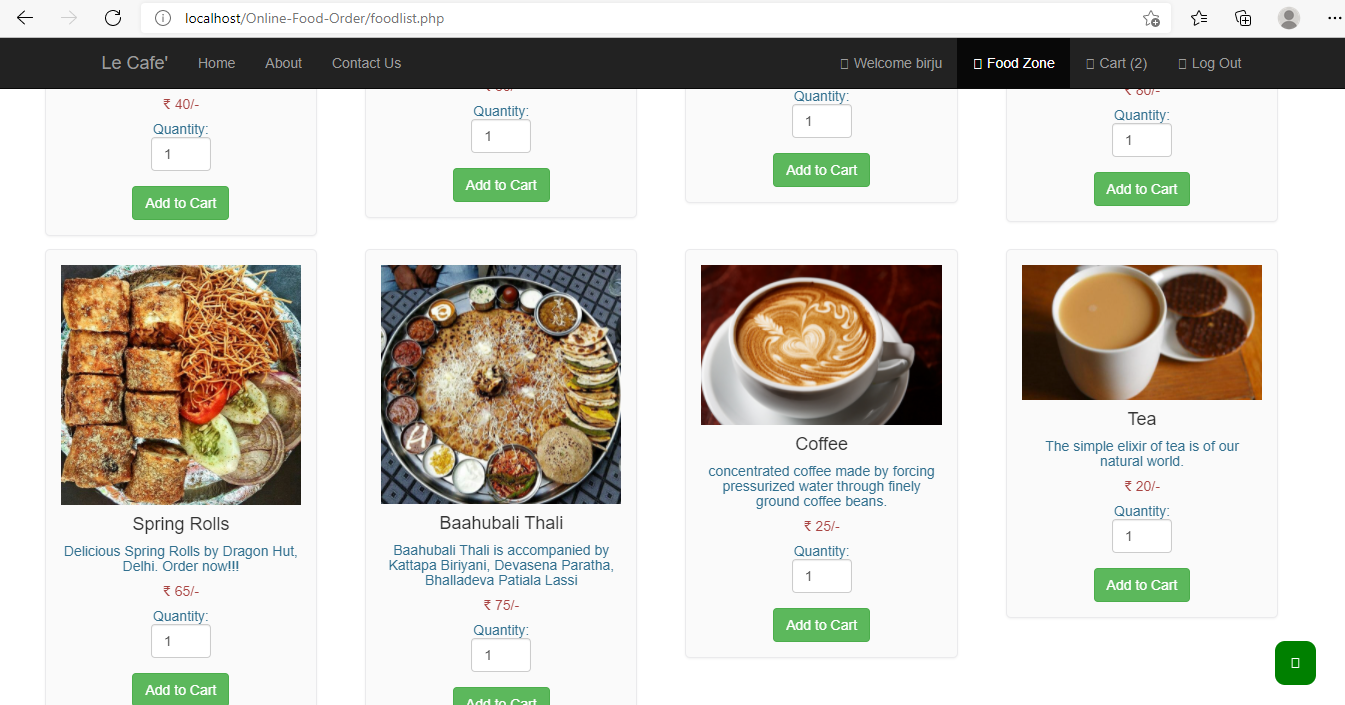
****

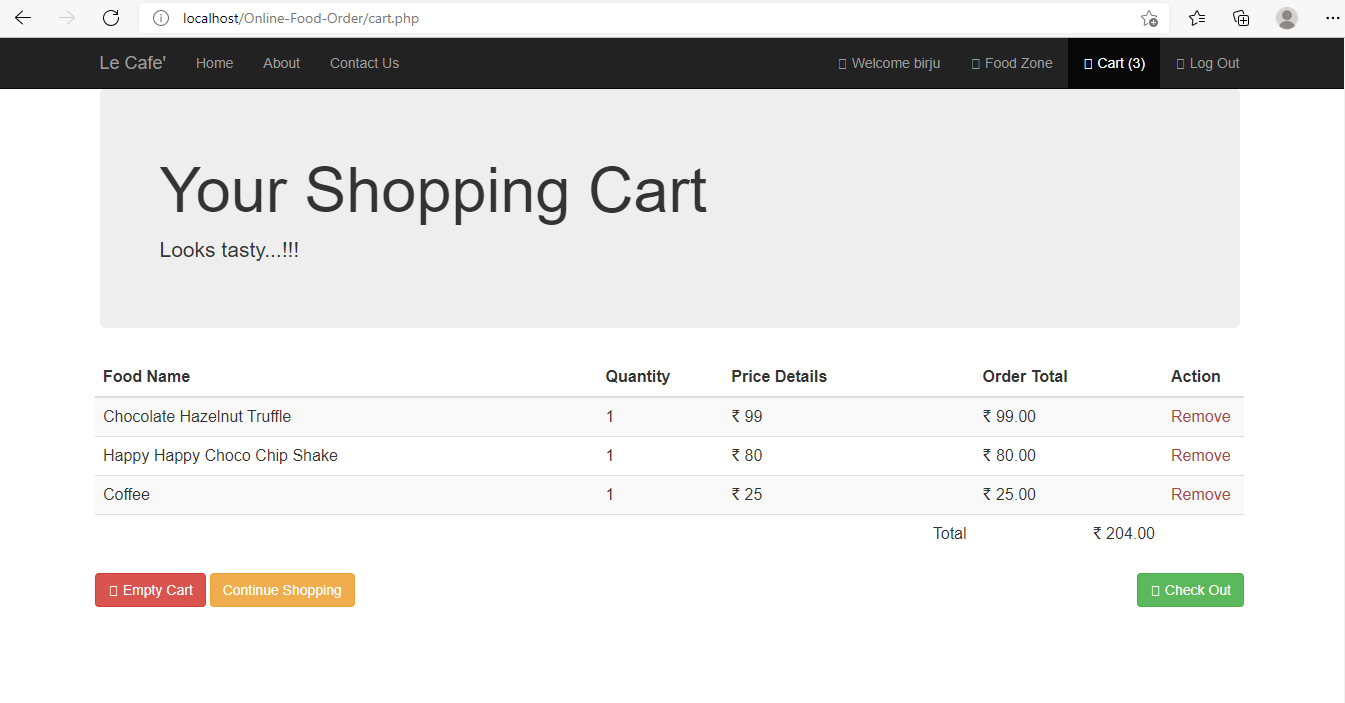
****

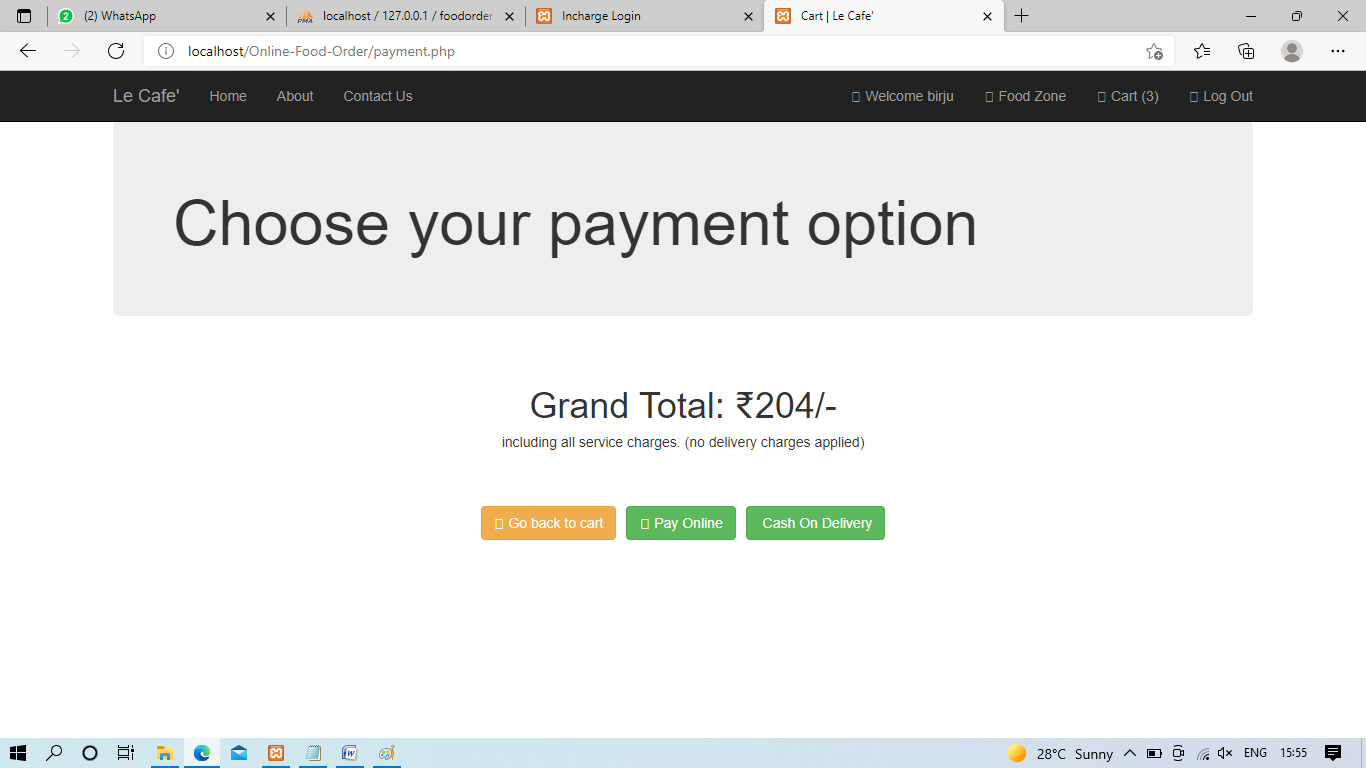
****

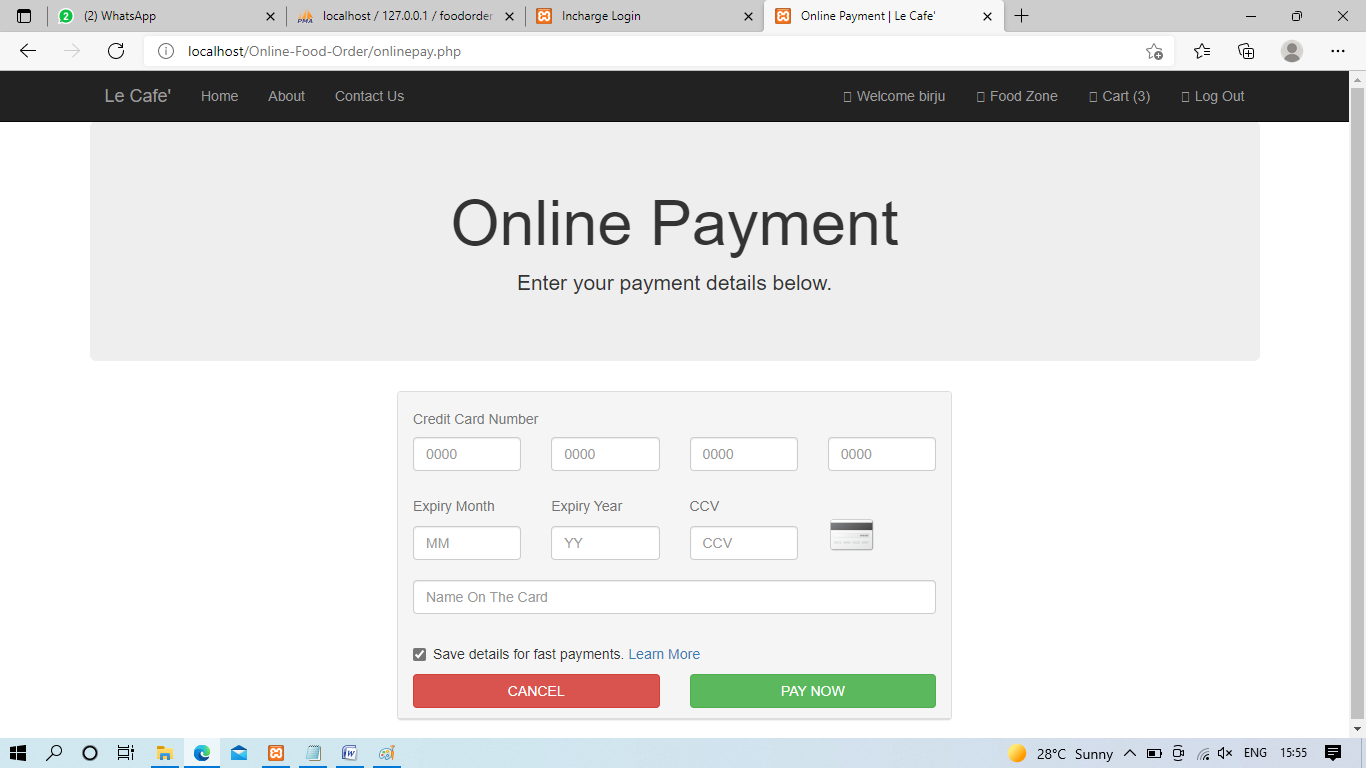
****

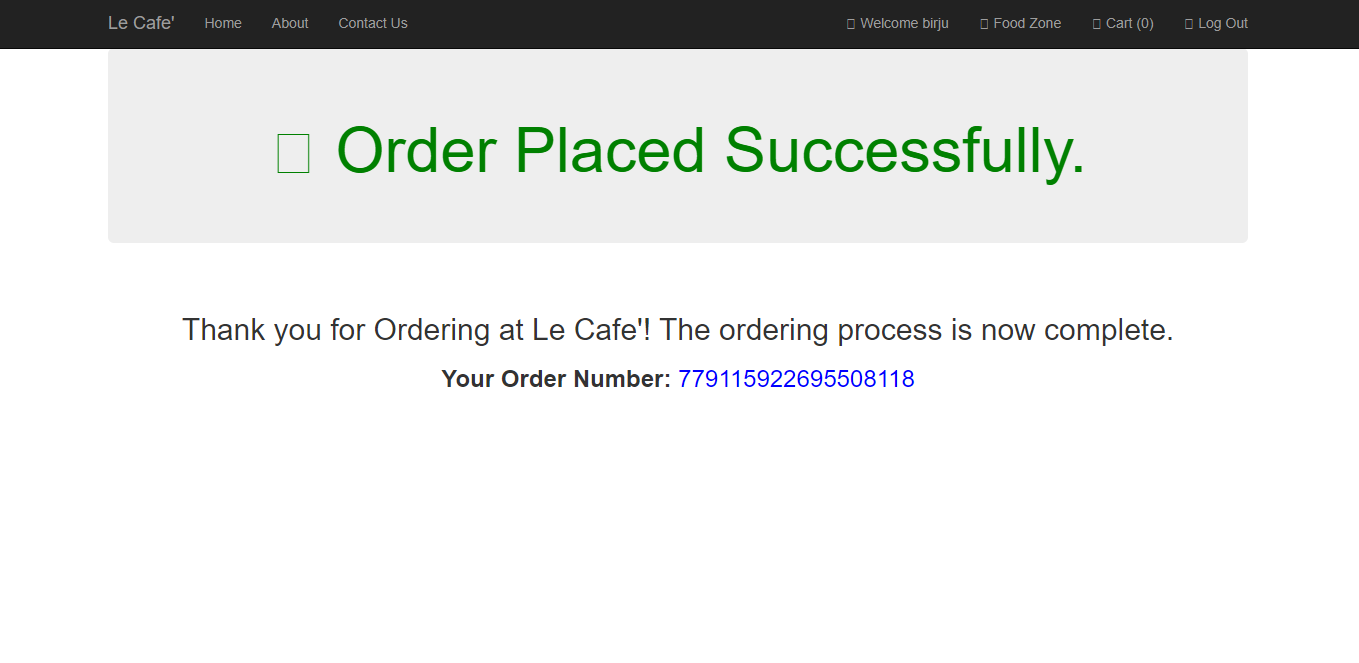
****

****

****

******

******

****

**Conclusion:**

In conclusion I would like to thanks to faculty member of the university

And all my peers for supporting me in the completion of this project.

I would also like to thanks all the teachers of the institute (I.C.M) who

helped me with all the problems that I encountered during the making of

this project.

**References:**

<https://www.softwareengineerinsider.com/programming-languages/php.html>

<https://www.softwareengineerinsider.com/programming-languages/php.html>