

PROFESSIONAL TRAINING REPORT

at

**Sathyabama Institute of Science and Technology
(Deemed to be University)**

Submitted in partial fulfillment of the requirements for the
award of Bachelor of Technology in Information Technology

By

**SOWMIYA S
REG. NO. 39120117**



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

NOVEMBER 2021



**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**
Accredited with Grade “A” by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI– 600119
www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SOWMIYA S (Reg. No: 39120117)** who carried out the project entitled “**Online Cake Ordering System on AWS**” under my supervision from June 2021 to November 2021.

Internal Guide

Dr. Senduru. Srinivasulu, M.E., Ph.D.,

Head of the Department

Dr. R. Subhashini, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **SOWMIYA S** hereby declare that the project report entitled “**Online Cake Ordering System on AWS**” done by me under the guidance of **Dr. Senduru. Srinivasulu, M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr. R. Subhashini, M.E., Ph.D. Head of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Senduru. Srinivasulu, M.E., Ph.D.,** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project

TRAINING CERTIFICATE



Elite NPTEL Online Certification

(Funded by the Ministry of HRD, Govt. of India)



This certificate is awarded to
SOWMIYA S
for successfully completing the course

Cloud Computing

with a consolidated score of **70** %

Online Assignments	23.75/25	Proctored Exam	46.52/75
--------------------	----------	----------------	----------

Total number of candidates certified in this course: **3189**


Prof. G P Raja Sekhar
Dean, Continuing Education
IIT Kharagpur

Aug-Oct 2021
(8 week course)


Prof. Debjani Chakraborty
Coordinator, NPTEL
IIT Kharagpur



Indian Institute of Technology Kharagpur



Roll No: NPTEL21CS62S23790197

To validate and check scores: <https://npTEL.ac.in/noc>

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	i
	LIST OF FIGURES	ii
	LIST OF TABLES	iii
	LIST OF ABBREVIATIONS	iv
1	INTRODUCTION	1
	1.1 Project Overview	1
2	AIM & SCOPE	2
	2.1 Aim	2
	2.2 Scope of the project	2
3	SYSTEM SPECIFICATIONS	2
	3.1 Hardware requirements	2
	3.2 Software requirements	2
4	LITERATURE REVIEW	3
	4.1 PDA- based system	3
	4.2 RFID-based system	4
	4.3 Mobile based application	5
	4.4 Web based application	6
	4.5 Evaluation of past works	7
5	WEB APPLICATION DEVELOPMENT TECHNOLOGIES	7
	5.1 HTML	7
	5.2 CSS	8
	5.3 Bootstrap	9
	5.4 PHP	10

	5.5 MySQL	10
	5.6 AWS	11
6	ONLINE CAKE ORDERING SYSTEM	12
	6.1 Components	12
	6.2 Overview of the components	12
	6.2.1 Home	12
	6.2.2 Products	13
	6.2.3 Contact us	13
	6.2.4 Cart	13
	6.2.5 Checkout	13
	6.2.6 Database	13
	6.2.7 AWS Hosting	13
7	CONCLUSION	14
	REFERENCES	14
	APPENDIX	15
	A. Screenshots	15
	B. Source code	22

ABSTRACT

An Online Cake Ordering System on AWS is an online cake ordering system hosted on AWS(Amazon Web Services). AWS(Amazon Web Services) is a comprehensive cloud computing platform provided by Amazon that includes a mixture of Infrastructure as a Service(IaaS), Platform as a Service(PaaS) and Storage as a Service(SaaS) offerings.

The System has been proposed here simplifies the cake ordering process. The proposed system shows an user interface and update the products with all available options so that it eases the customer work. Customer can choose more than one item to make an order and can view order details before proceeding to buy. The order confirmation is displayed on the screen to the customer. The order is updated in the database and returned in real time. This system assists the staff to go through the orders in real time and process it efficiently with minimal errors.

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
A.1	AWS EC2 instance	
A.2	Remote desktop/virtual cloud	
A.3	XAMPP server on remote desktop	
A.4	phpMyAdmin on remote desktop	
A.5	Product database	
A.6	Order details database	
A.7	Running website on remote desktop	
A.8	Deploying website on cloud	
A.9	Home page	
A.10	Products page-1	
A.11	Products page-2	
A.12	Contact us	
A.13	Cart	
A.14	Payment form	
A.15	Thank you page	

LIST OF TABLES

TABLE NO. NO.	TABLE NAME	PAGE
2.1	Scope of the Project	

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSIONS
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
PHP	Hypertext Preprocessor
AWS	Amazon Web Services
EC2	Elastic Compute Cloud

1. INTRODUCTION

1.1. Project Overview

The “Online Cake Ordering System on AWS” can be defined as a simple and convenient way for customers to purchase cake online, without having to go to the restaurant. This system is enabled by the internet – it is the internet that connects the bakery on one hand, and the customer on other hand. Therefore, as per this system, the customer visits the bakery’s website, browses through the various items available there and goes ahead and selects and purchases the items he or she needs.

The labour rates are increasing steadily year on year thus making it difficult to find employees. The food industry is highly labour intensive and the biggest expense in the food industry is the cost of employing the right kind of people to do the work. One of the ways to reduce this expense is to use modern technology to replace some of the jobs done by human beings and make machines do the work. Here we propose an “Online Cake Ordering System on AWS” that has been designed for Bakery. The system can also be used in any food delivery industry. This simplifies the process of cake ordering for both the customer and the bakery employee, as the entire process of taking orders is automated. This system has been hosted on AWS(Amazon Web Services) EC2 cloud so that it can be accessible from anywhere. This system will decrease the customer time to order a cake online than going to the bakery in person.

The project requires a complete web application for bakery that can be used to manage online orders. All the technologies that will be used to develop the system are open-sourced, such as PHP, HTML, CSS, Bootstrap 4, and Apache Server. Hence, the technologies can be used without any cost.

2. AIM & SCOPE

2.1. Aim

The main purpose of an online cake ordering system is to provide customers a way to place an order over the internet. With a website , customers can easily browse all the dishes the bakery has available, customize items to their requirements and place an order.

2.2. Scope of the Project

The following table shows the scope of the project:

Table 2.1 : scope of the project

Scope in	Scope out
1. Web application for bakery 2. Online bakery management system 3. Online cake order system 4. Written report on the Project	1. IoT based food ordering system 2. Physical smart system for bakery 3. Mobile application

3. SYSTEM SPECIFICATIONS

3.1. Hardware Requirements

Processor : Processor Intel CORE i3 and above
Internet Connection : Existing telephone lines, Data card, Fiber net
RAM : 4 GB

3.2. Software Requirements

Operating System : Windows/ Mac/ Linux

Software : XAMPP, Sublime Text/ VScode
Cloud Platform : AWS (Amazon Web Services)

4. LITERATURE REVIEW

This chapter discusses the different approaches for automating the ordering processes of bakery. After that, it explains the web application development tools and technologies used in this project.

4.1. PDA- based system

Xiang, Zhou and Chowdhury (2004) developed a PDA based in-house meal ordering system for restaurant named iMenu. It is one of the first systems that introduced PDA- based ordering in restaurant industry. The developers used .NET platform for developing the system. Patel, Patel and Obersnel (2007) discussed about implementation of PDA-based ordering. The proposed system by Patel, Patel and Obersnel (2007) was a web-based stand-alone wireless application. Connectivity between client and server is continuous in two-tiered web-based client-server architecture. Customers in restaurant can order through PDAs. Hence, the bill of foods can be generated and paid within a short time. Hongzhen, Bin and Wenlin (2009) developed a web service and wireless system which can be used through desktop and personal digital assistant (PDA). In this system, the access of data to the servers can be wired and wireless. The food ordering functions served through both desktop computers and PDAs over a wired/wireless integrated local area network. These PDAs were used to see menus and make orders. After the customer made orders through PDA, the manager can see the orders through computer. They implemented the system using .NET and C#.

There are many limitations of these PDA-based systems. Such as:

- In a PDA-based food ordering system, restaurant owners must have enough number of PDAs to serve the number of the customers especially during peak hours. Accommodation and maintenance of enough PDAs can increase the restaurant expenditure which is a big challenge for restaurants as most of the restaurants are categorised as small medium enterprise (SME) (Cheong, Chiew

and Yap, 2010; Samsudin et al., 2011).

- Continuous use of PDAs leads to recharging batteries multiple times which is a big challenge (Patel, Patel and Obersnel, 2007).
- Sharing same PDAs among many customer may help in spreading contagious diseases (Samsudin et al., 2011).
- Customers are required to be present physically to make order using the PDA-based system. Thus, customers have no alternatives to make early order before coming to the restaurant. This implies that customers have to accommodate waiting time for food preparation (Samsudin et al., 2011).
- Today, PDA based in-house ordering system is an obsolete technology as PDAs lost the attraction of the people and not used anymore widely because of the advancement of cell phone technology (Smith and Wempen, 2011).

4.2. RFID- based system

Many researchers also proposed and implemented wireless communication based system within the restaurant. Ngai, Suk and Lo (2008) developed a radio frequency identification (RFID) based sushi management system in a conveyor-belt sushi restaurant. Their case study showed that RFID technology helps improve food safety, inventory control, service quality, operational efficiency, and data visibility in sushi restaurants. Unfortunately, this system does not support customer-centred service because it cannot actively identify customers. To overcome this problem, Tan, Chang and Chen (2012) proposed a RFID based smart food recommender system, where customers will have a RFID based card. The system integrates radio frequency identification (RFID), wireless local area network, database technologies, and a menu recommender. This system enables waiters to immediately identify customers via RFID-based membership cards and then actively recommend the most appropriate menus through menu recommender for customers. The customer will be identified by this RFID based card and recommended food menu based on his/her past choice of food. For taking orders, the waiters use PDAs that are connected the system via local area network (LAN). Gayatri, Chaitanya and Harikrishna (2014) also proposed a RFID based model

almost the same system like Tan et al. (2012). But they added an extra feature to the system, where customers can order food from their own mobile through short text message (SMS).

There are some problems with this system, which are follows:

- In this systems, the customer needs to bring his/her card or smartphone to restaurant. If s/he failed to bring the card to the restaurant they cannot use many facilities of the system.
- These systems is focused only the customers who have the membership cards; not all customers get the service as well.

4.3. Mobile based application

Samsudin et al. (2011) recognised the limitations of PDA based food ordering system and developed a food ordering system with Microsoft Access database and ASP.NET and VB.NET for Microsoft Windows based smartphones. Customers can order foods remotely from this application. As Android based smartphones took over Windows based smartphones, Shinde et al. (2014) developed an application for Android operating system based smartphones where they used Java and SQLite. The order processing speed of this Android application is faster than PDA based ordering system. The application can play a media between customer, restaurant manager and kitchen. Dhore et al. (2014) also proposed an Android based system to automate ordering process of the restaurant, through which one or more customers can order food, book table, and also make payment remotely. Both of these to works implemented almost same functionalities. Leong (2016) also developed a system for Android operating system, which can be used for ordering food online. In this system, computers need to be installed in kitchen and manager's desk, so they can view and check orders and payment. Recently, Saeed et al. (2016) implemented an Android based smart restaurant system, which helps customers finding free parking lot and free table, ordering foods and paying bills from their mobile phone. On the other hand, the management can monitor the whole work properly. This system is comprised of different technologies such as Mobile and Web applications, Internet of Things (IoT), Near-Field Communications (NFC) sensors, and cloud computing. This Smart

Restaurant Management System provides with two interfaces for the two types of users in restaurants; an Android mobile application for customers and a web application for restaurant staff members.

These proposed and developed smartphone based systems let customers order foods online before arriving to restaurant. But these systems have also some limitations as follows:

- These applications are focused on specific operating system of smartphone. So, who uses a smartphone running on different operating system cannot use the application until there is another version of application is developed for that specific platform. For example, an application developed for Windows smartphone will not run on an Android smartphone.

4.4. Web based application

To overcome the device and operating system centric approach for automating restaurant operation, K et al. (2016) proposed an online web application which can be accessed through internet. For using this application, computer screen will be placed on each table of restaurant for customers to order. Customers can order food for take away and make payment through the application. However, the idea is innovative, but a little bit weird as computers need to be installed in every table. Sometimes it may create complex problem rather than making a solution. Patel (2015) developed an online food ordering system with Java in backend system and HTML with CSS in frontend. It was a robust and nice application. But the responsive development method was not used in the application. Hence, the application cannot viewed properly with a smart phone. Tan (2013) developed a web based application with ASP .Net and Bootstrap to automate the ordering process of the restaurant and make it paperless. But the system is only focused on the management of the restaurant. There is no option for customers to order food online. The system was responsive.

These web applications, however, overcame the problem of being dependent on specific device (e.g. PDA) or operating system. These systems also have few limitations, such as:

- None of them tried to make the online ordering system more trustable to

customers.

- They did not integrated social media to the system which can make customer registration and login process easier.

4.5. Evaluation of the Past Works

The previous lines illustrate that many researchers worked for implementing IT based systems in restaurants to gain more revenue and reduce order processing time. But most of them did not focus on the online based restaurant system where customers can order food over internet from their home. With the expansion of internet technology in this era, it is not feasible to stick with only technologies demonstrated above as most of the proposed system only tried to automate the in-house ordering system. On the other hand, the systems which is developed to order food remotely are device or operating systems oriented. For obtaining the benefits of the system, customers need to use a smartphone with specific operating system like Windows or Android. No one tried to find a solution which does not depend on any specific device, while a web application can be accessed by through kind of devices with internet connection from anywhere. Moreover, there is no need for installing extra software for browsing a web application. If the web application is developed using modern mobile friendly manner properly, it will give a good performance on any kind and size of device.

5. WEB APPLICATION DEVELOPMENT TECHNOLOGIES

5.1. HTML

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

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>` 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), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

5.2. CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.^[1] CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.^[2]

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.^[3] This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.^[4]

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) `text/css` is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.^[5]

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

5.3. Bootstrap

As the number of Internet users on mobile devices is growing every moment, websites are no longer built only for the desktop machines. The mobile-first philosophy demands the sites to be fully compatible for all available and future mobile devices. Bootstrap allows and easily enables to design and develop websites congenial to all devices including various screen readers.

Bootstrap is a free CSS and JavaScript framework that allows developers to rapidly build responsive web interfaces. The framework was first conceived by Mark Otto and Jacob Thornton while working at Twitter in 2010. Since its release as an open source project in August of 2011, Bootstrap has become a household name among frontend web developers (Jakobus and Marah, 2016).

Bootstrap is the most starred project in GitHub since 2012. Bootstrap is largely based on CSS3 that is currently the latest version of Cascading Style Sheet (CSS) and included with powerful capabilities like Media queries, Selectors, Cascading and Inheritance, Template layouts, Namespaces, Flexible and Grid layouts etc. (Bhaumik, 2015). Bootstrap 4 has improved grid system and inclusion of flexbox.

For regular grid, another regular grid level is introduced for better targeting smartphones (Lambert, 2016).

There are many other frontend frameworks like Zurb, Foundation etc. But the most powerful thing of Bootstrap is its popularity. Hence, it is much easier to find solutions with bootstrap (Gerchev, 2017).

5.4. PHP

PHP is acronym for PHP: Hypertext Preprocessor. PHP is an “open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML” (The PHP Group, 2018). PHP is mainly used in dynamic web page, including CLI (command line interface) and GUI (graphical user interface) program. It has the feature of good across-platforms and easy transplant (Lei, Ma and Tan, 2014). PHP is not an object-oriented language like Java. But according to Haydar (2007), PHP 5 implements almost complete object-oriented features. 83.1% of all the websites use PHP as server-side programming language and 14.1 of all websites use ASP.NET as server side language by January 2018 (W3Techs, 2018).

Along with this, nearly 60% of web servers are running on Apache with PHP (Hayder, 2007). Hence, many researchers tried to compare between these two languages. According to a study of MVC frameworks from PHP and .NET by Jailia et al. (2016), .NET gives better result for page loading time, time to first byte and time to last byte compared to PHP based web applications. While PHP based website gives much more promising result for request and response transfer speed. Bounnady et al., (2016) also made a comparison between PHP and .NET in five different parameters: load static webpage, algorithms processing, managing database, uploading file and reading/writing files. The study concluded that PHP performs better than .NET in managing database operations and uploading files.

5.5. MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter,

and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation).¹ In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for *Linux, Apache, MySQL, Perl/PHP/Python*. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress.

5.6. AWS(Amazon Web Services)

Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. These cloud computing web services provide a variety of basic abstract technical infrastructure and distributed computing building blocks and tools. One of these services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, available all the time, through the Internet. AWS's virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing;

local/RAM memory; hard-disk/SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM).

The AWS technology is implemented at server farms throughout the world, and maintained by the Amazon subsidiary. Fees are based on a combination of usage (known as a "Pay-as-you-go" model), hardware, operating system, software, or networking features chosen by the subscriber required availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either. As part of the subscription agreement, Amazon provides security for subscribers' systems. AWS operates from many global geographical regions including 6 in North America.

Amazon markets AWS to subscribers as a way of obtaining large scale computing capacity more quickly and cheaply than building an actual physical server farm. All services are billed based on usage, but each service measures usage in varying ways. As of 2017, AWS owns 33% of all cloud (IaaS, PaaS) while the next two competitors Microsoft Azure and Google Cloud have 18%, and 9% respectively, according to Synergy Group.

6. ONLINE CAKE ORDERING SYSTEM

6.1. Components

- Home
- Products
- Cart
- Contact us
- Checkout
- Database
- AWS hosting

6.2. Overview of the Components

6.2.1. Home

The Home page will have a header which contains the links to all the pages of the website like products, cart, contact us etc.,

6.2.2. Products

This page will display all the items which has been stored on the database. The PHP code will retrieve the data from the database and display it on the website.

6.2.3. Contact us

The details of the bakery will be displayed here. This is the footer of the homepage.

6.2.4. Cart

On this page the items which have been added on the cart will be displayed. The total price and the link for placing the order will be shown.

6.2.5. Checkout

This session will provide the user to fill a form for placing the order, the data which has been collected in this form will be store on the database which the bakery employee can use to process the order and will display a thank you message.

6.2.6. Database

There will be two databases for this particular website. The first one have the details of the items to be displayed on the products page i.e. product name, price and image. The bakery person can edit and access these databases. The second one which stores the order details which has been received from the website.

6.2.7. AWS hosting

For the website to be accessed from anywhere it has to be hosted on the cloud. For this project, the bakery website has been hosted on AWS EC2 cloud. By creating an EC2 instance and creating a private virtual cloud desktop we can deploy the website online. An IP address is being created while doing this which can be used to access the website from anywhere. If we close the virtual desktop then we can't access the website, we have to login again to the virtual cloud and the website will be accessed.

7. CONCLUSION

An online cake ordering system is developed where the customers can make an order for the cake easily with less time. In conclusion an online cake ordering system is proposed which is useful for many home bakers who have been running small home bakeries and also for large bakeries too. This helps the bakeries to maintain a healthy customer relationships by taking and delivering the order on correct time. Many functionalities can be added to this website for it's better performance. The system can also be used in any food delivery industry. This simplifies the process of cake ordering for both the customer and the bakery employee, as the entire process of taking orders is automated.

REFERENCES

1. Patel, Mayurkumar, "Online Food Order System for Restaurants" (2015). Technical Library. Paper 219.
2. Beltis, A. J. (2016) 9 Advantages of an Online Food Ordering System. Available at: <https://pos.toasttab.com/blog/online-food-ordering-system> (Accessed: 7 July 2017).

3. Awojide, Simon, I. M. Omogbhemhe, O. S. Awe, and T. S. Babatope, "Towards the digitalization of Restaurant Business Process for Food Ordering in Nigeria Private University: The Design Perspective. A Study of Samuel Adegboyega University Edo State Nigeria," Int. J. Sci. Res. Publ., vol. 8, no. 5, pp. 46–54, 2018.
4. igp.com/cakes website has been taken as reference.
5. For PHP- www.w3schools.com
6. Varsha Chavan, Priya Jadhav, Snehal Korade, Priyanka Teli, "Implementing Customizable Online Food Ordering System Using Web Based Application", International Journal of Innovative Science, Engineering Technology(IJSET) 2015.

APPENDIX

A. Screenshots

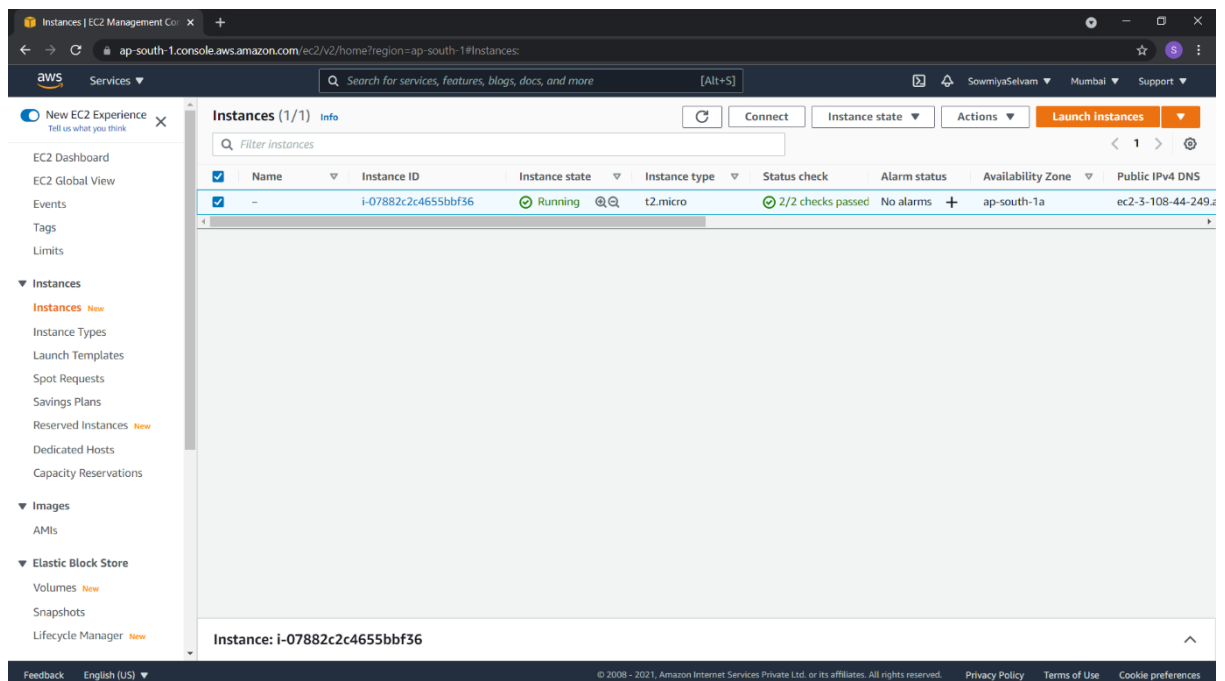


Fig: A.1: AWS EC2 instance

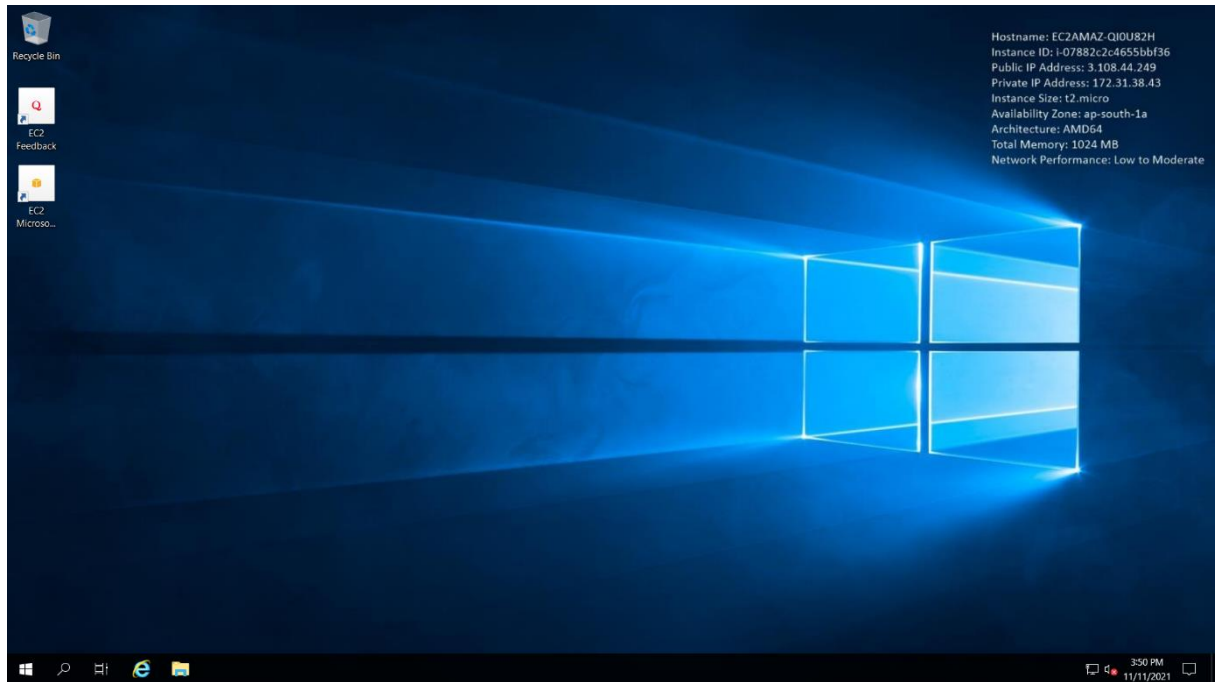


Fig: A.2: Remote desktop / Virtual cloud

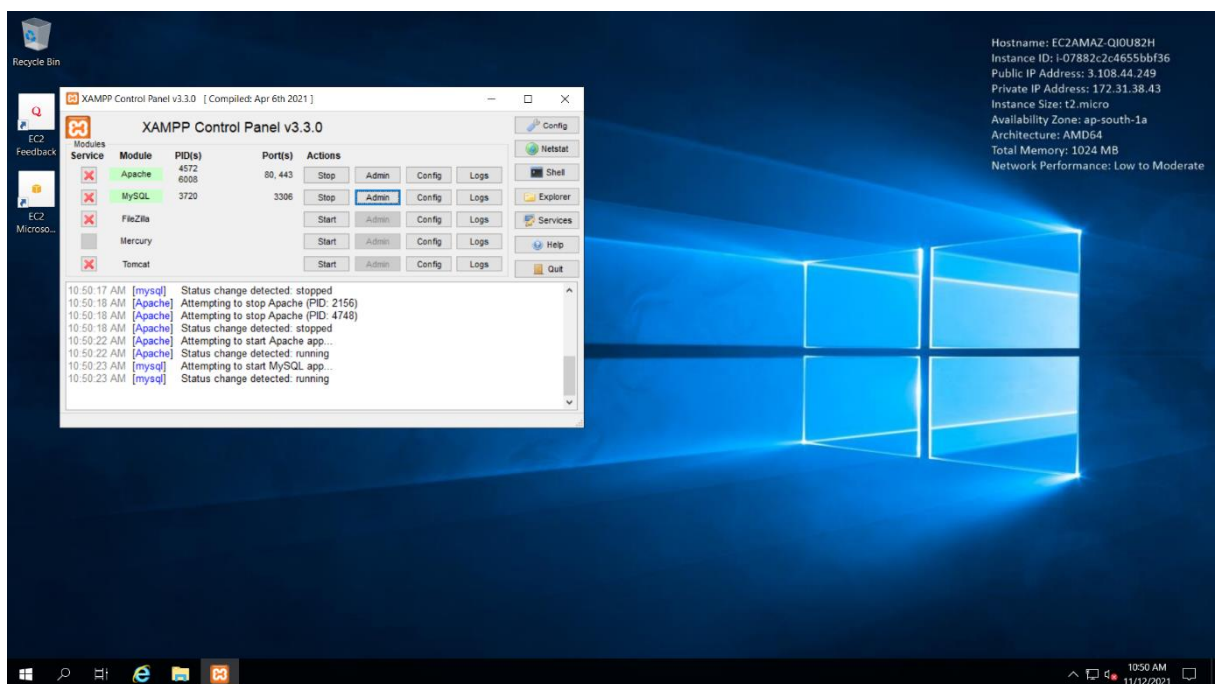


Fig: A.3: XAMPP control panel on remote desktop

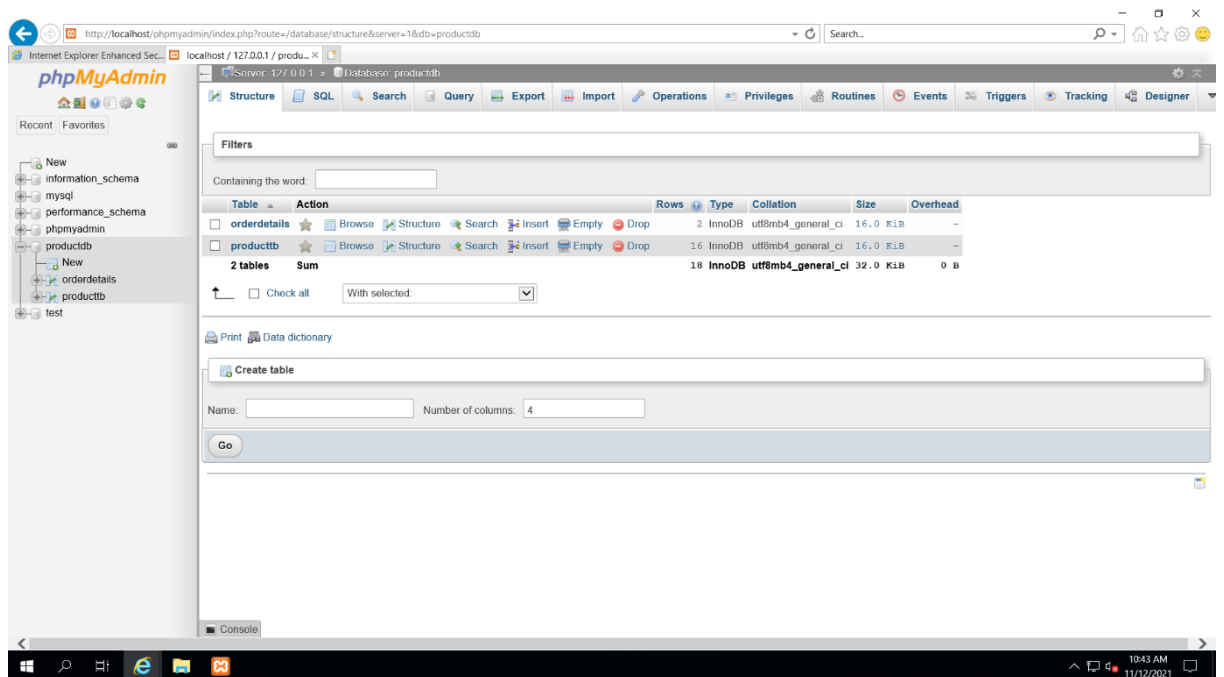


Fig: A.4: phpMyAdmin on remote desktop

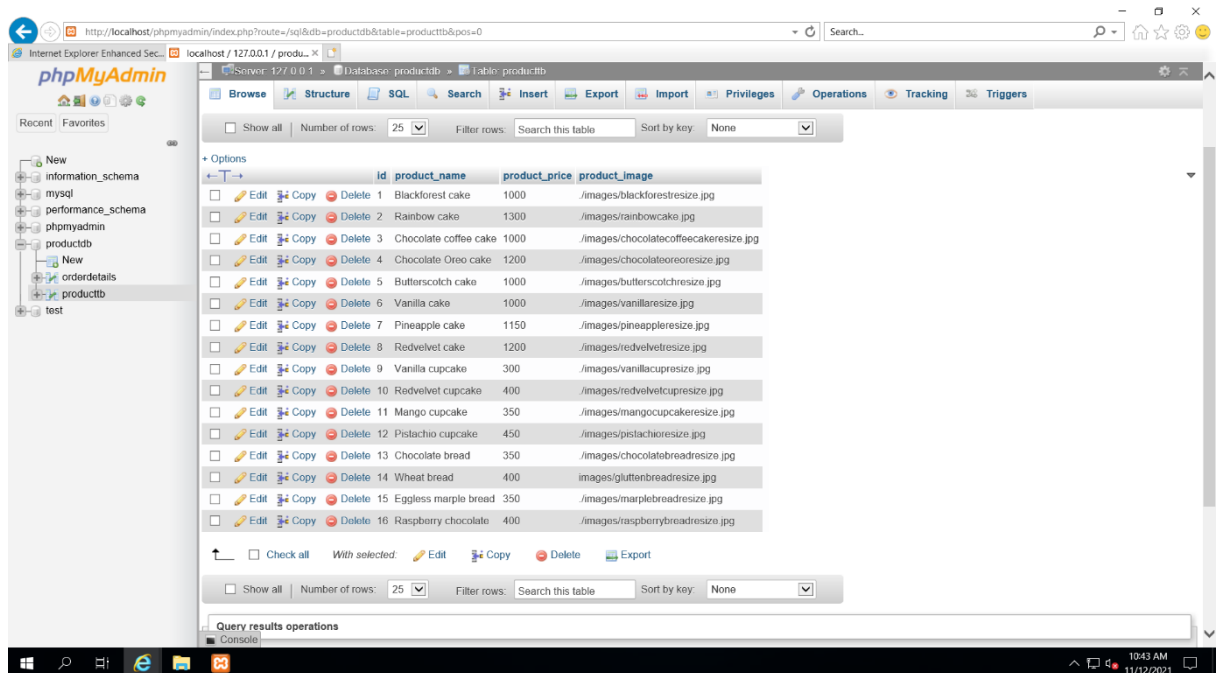


Fig: A.5: Product database

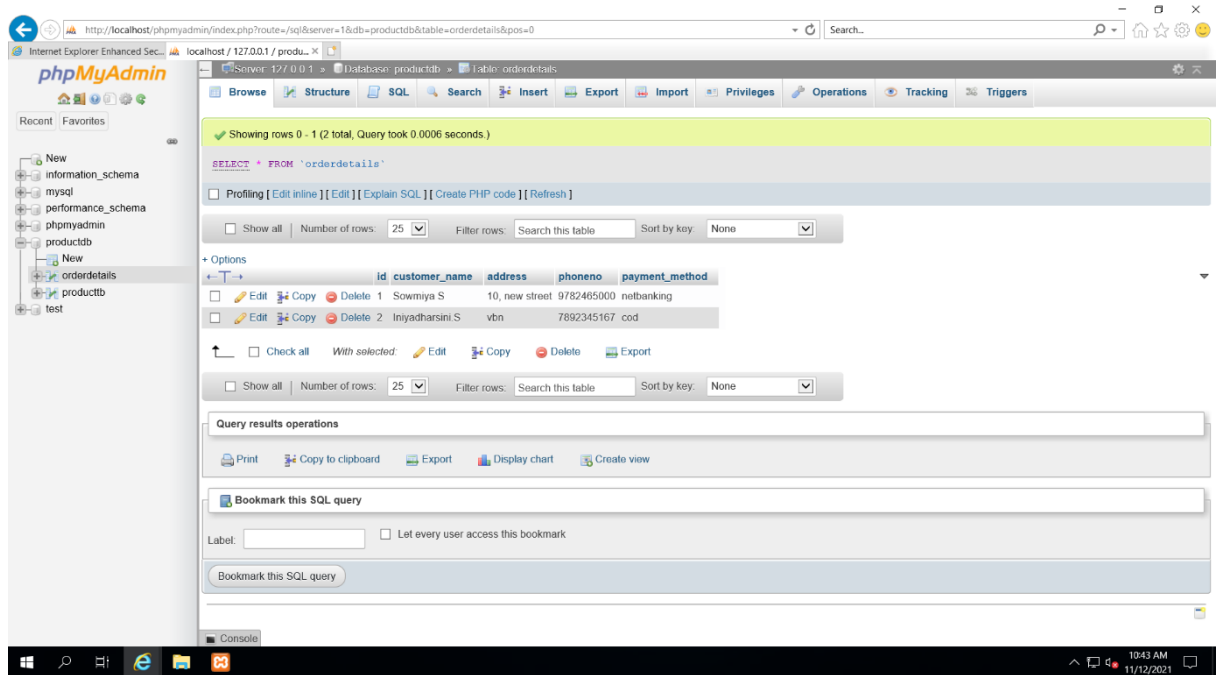


Fig: A.6: Order details database

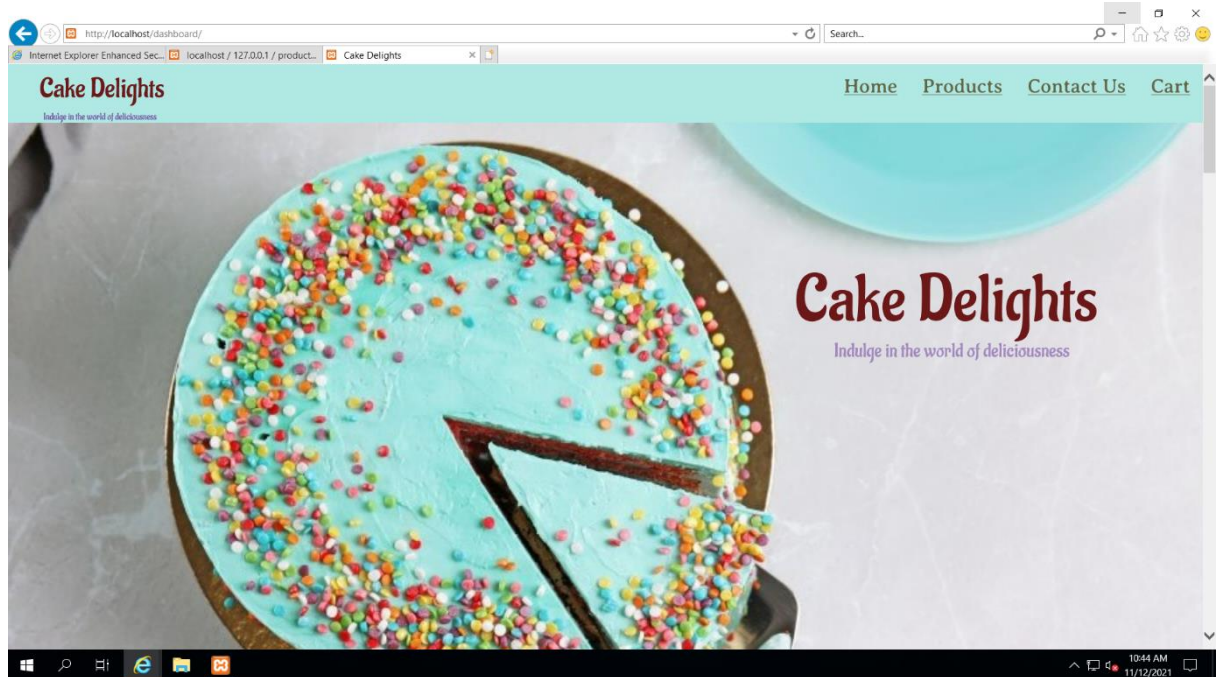


Fig: A.7: Running website on remote desktop

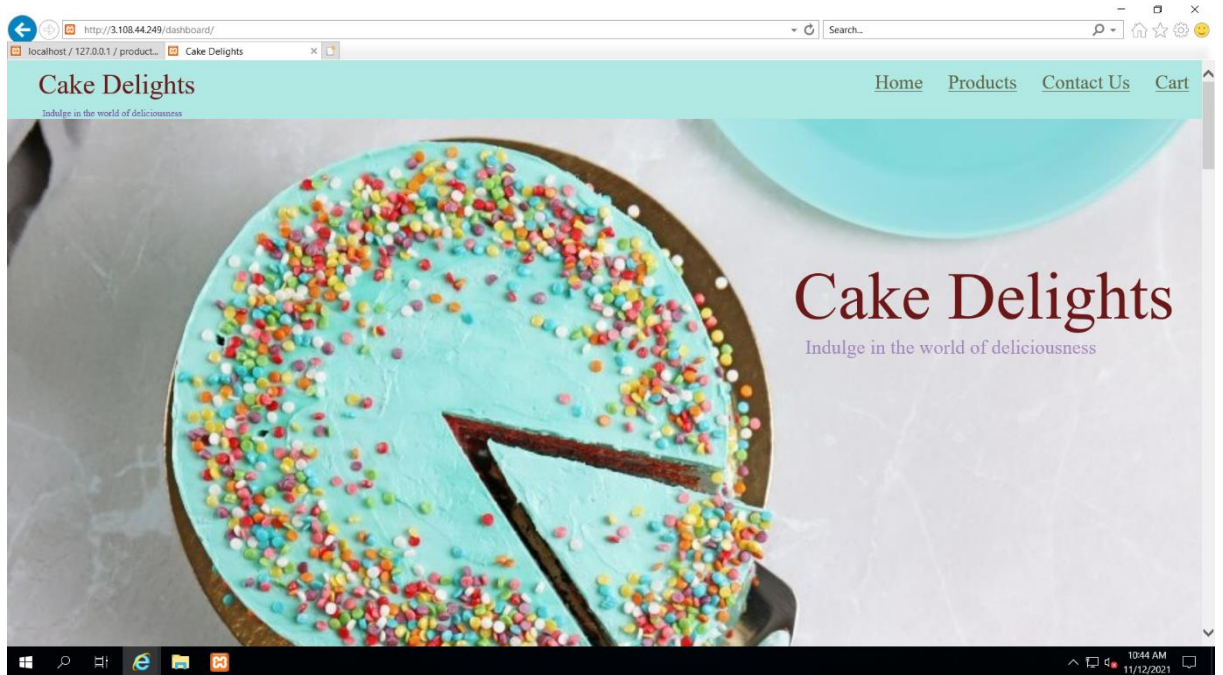


Fig: A.8: Deploying website on cloud

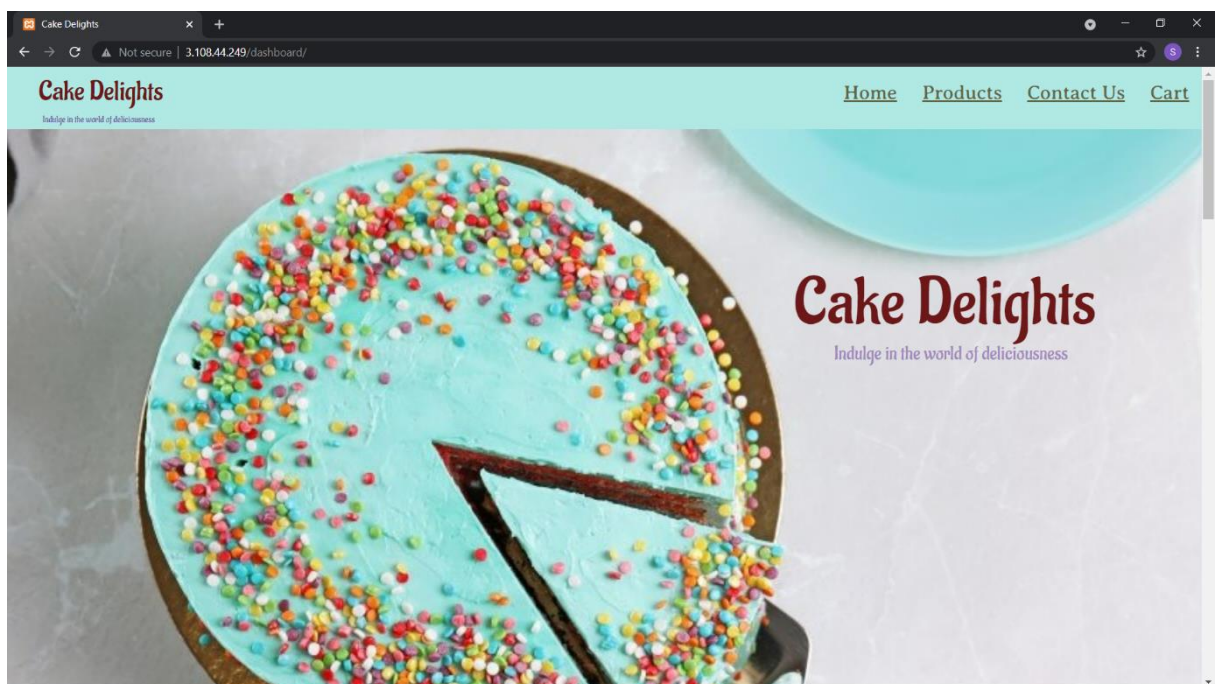


Fig: A.9: Home Page

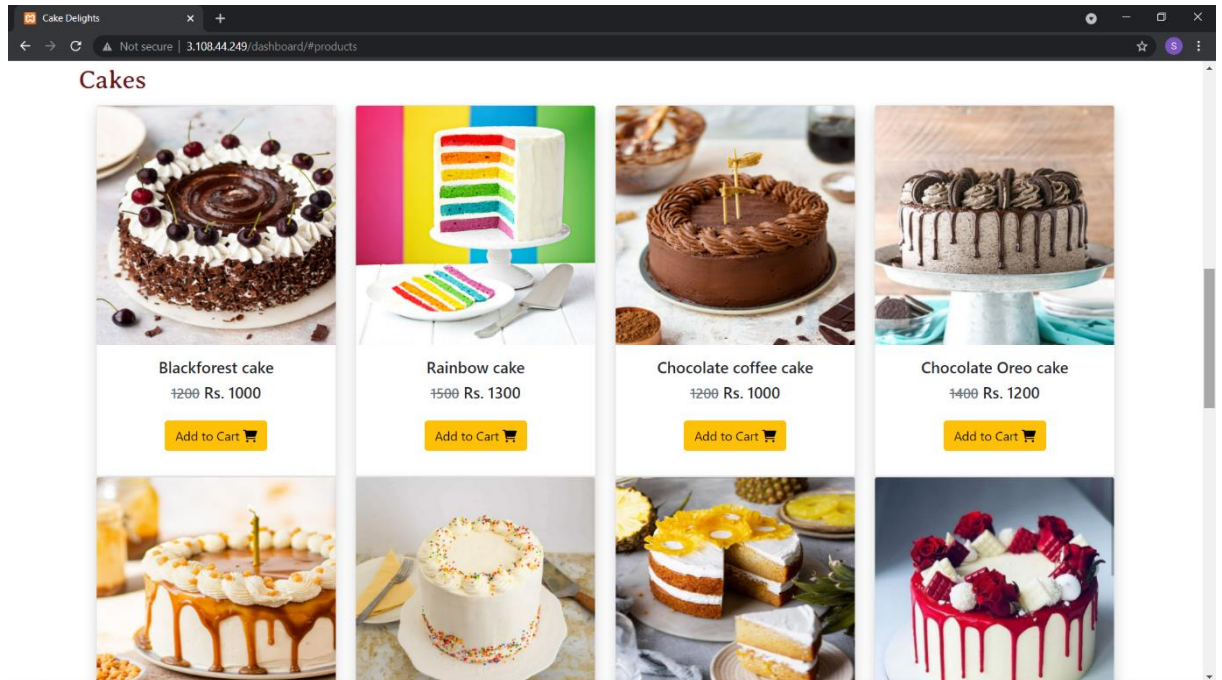


Fig: A.10: Products Page -1

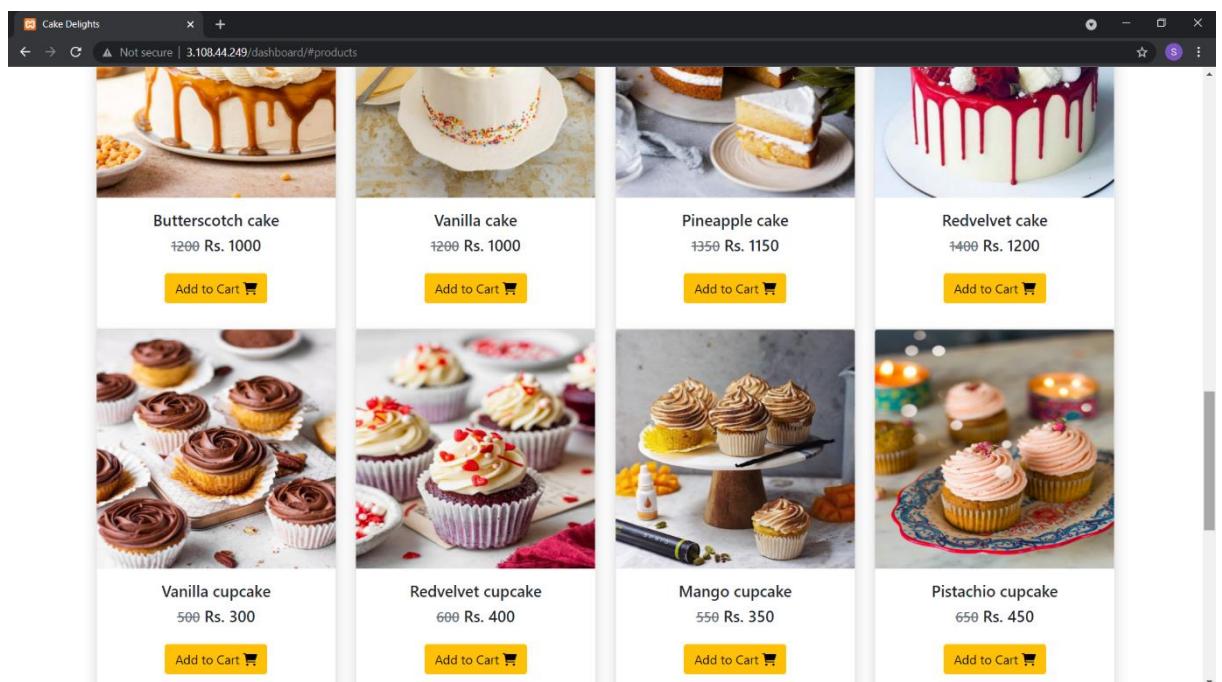


Fig: A.11: Products Page- 2

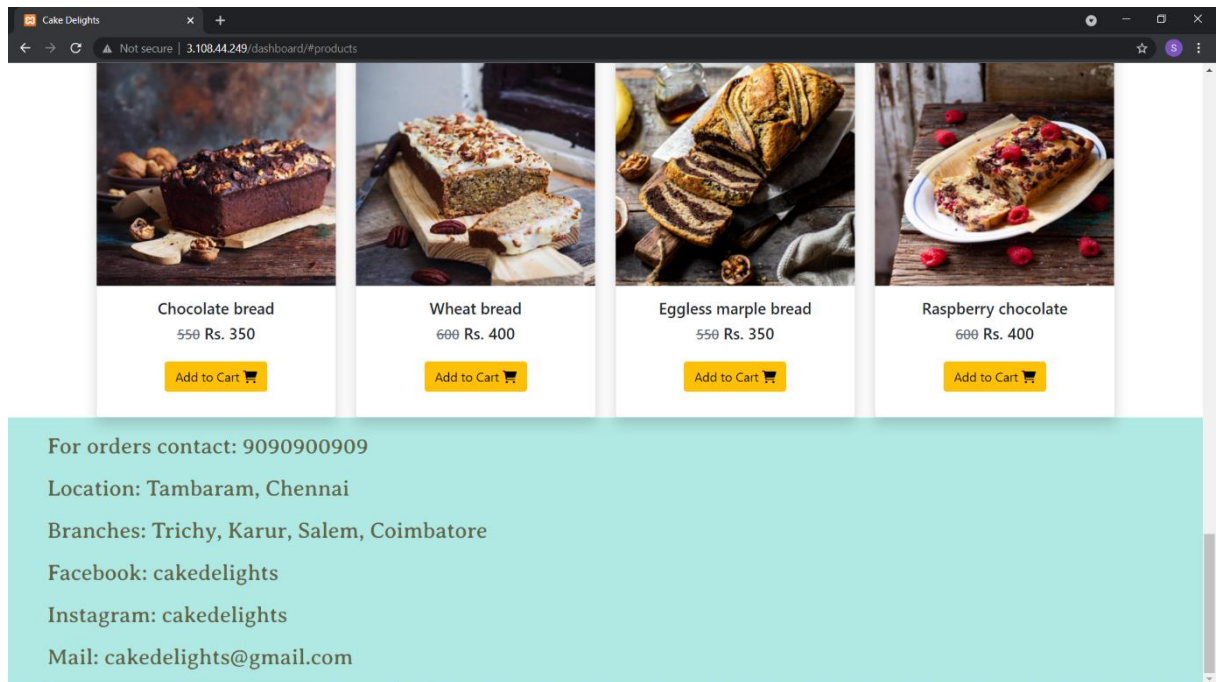


Fig: A.12: Contact us

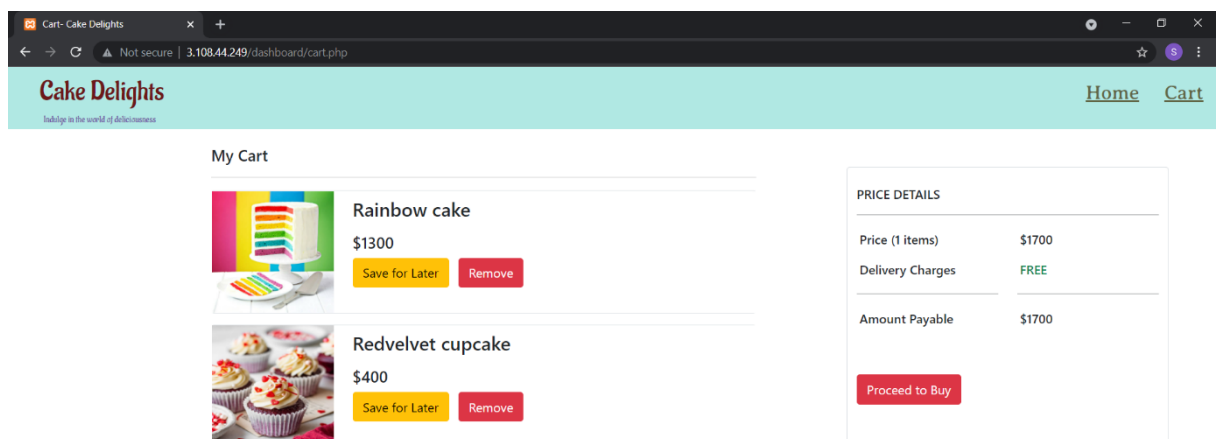
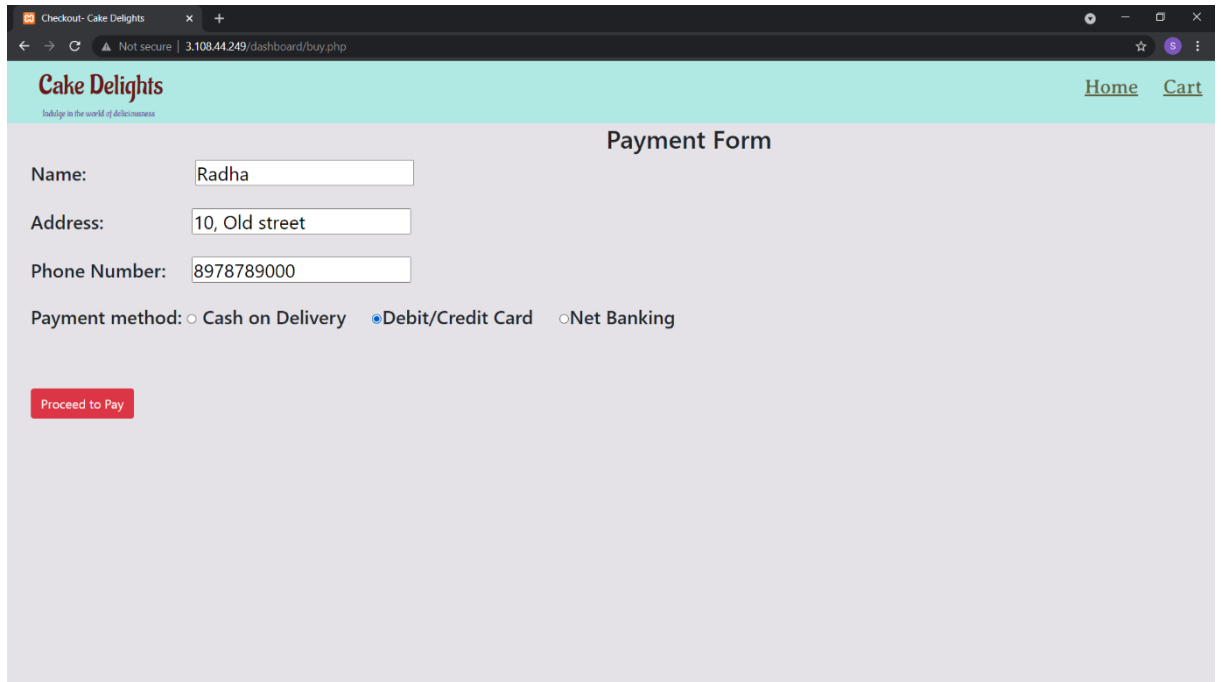


Fig: A.13: Cart



The screenshot shows a web browser window with the title "Checkout- Cake Delights". The address bar shows "3.108.44.249/dashboard/buy.php". The page has a teal header with the "Cake Delights" logo and the tagline "Indulge in the world of deliciousness". Navigation links for "Home" and "Cart" are in the top right. The main content area is titled "Payment Form" and contains the following fields and options:

- Name:
- Address:
- Phone Number:
- Payment method: ☐ Cash on Delivery ☒ Debit/Credit Card ☐ Net Banking
- A red "Proceed to Pay" button.

Fig: A.14: Payment form

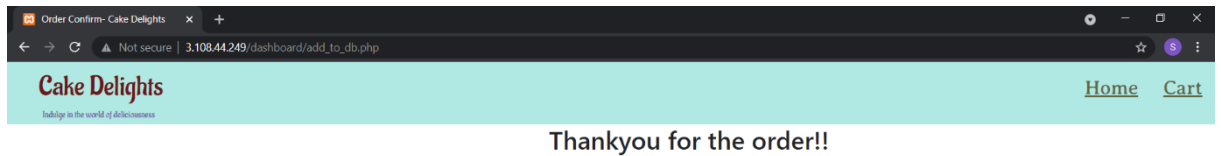


Fig: A.15: Thank you page

B. Source Code

B.1. index.php

```

<?php

session_start();

require_once ('./createdb.php');
require_once ('./component.php');

// create instance of Createdb class
$databse = new CreateDb("Productdb", "Producttb");

if (isset($_POST['add'])){
    /// print_r($_POST['product_id']);
    if(isset($_SESSION['cart'])){

        $item_array_id = array_column($_SESSION['cart'], "product_id");

        if(in_array($_POST['product_id'], $item_array_id)){
            echo "<script>alert('Product is already added in the
cart..!')</script>";
            echo "<script>window.location = './cart.php'</script>";
        }else{

            $count = count($_SESSION['cart']);
            $item_array = array(
                'product_id' => $_POST['product_id']
            );

            $_SESSION['cart'][$count] = $item_array;
            echo "<script>alert('Added to the cart..!')</script>";
            echo "<script>window.location = './cart.php'</script>";

        }

    }else{

        $item_array = array(
            'product_id' => $_POST['product_id']
        );

        // Create new session variable
        $_SESSION['cart'][0] = $item_array;
    }
}

?>

```

```

<!DOCTYPE html>
<head>
  <title>Cake Delights</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href='https://fonts.googleapis.com/css?family=Aladin'
rel='stylesheet'>
  <link href='https://fonts.googleapis.com/css?family=Averia Serif Libre'
rel='stylesheet'>
  <link rel="stylesheet" type="text/css" href="./cakestyling.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0-beta2/css/all.min.css" integrity="sha512-
YWzhKL2whUzgiheMoBFwW8CKV4qpHQA Euvilg9FAn5VJUDwKZZxkJNuGM4XkWuk94WCrrwslk8yWNG
mY1EduTA==" crossorigin="anonymous" referrerpolicy="no-referrer" />
  <!-- bootstrap cdn -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
</head>
<body>
  <header class="topheader">
    <div class="header1">
      <h2>Cake Delights</h2>
      <h3>Indulge in the world of deliciousness</h3>
    </div>
    <div class="home">

      <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#products">Products</a></li>
        <li><a href="#footer">Contact Us</a></li>
        <li><a href="./cart.php">Cart</a></li>
      </ul>

    </div>
  </header>
  <div id="home">
    <div class="middle">
      <h1>Cake Delights</h1>
      <div class="caption"><h1>Indulge in the world of
deliciousness</h1></div>
    </div>
  </div>
  <div id="products">
    <section>
      <div class="subtitle">

```

```

        <h2>Cakes</h2>
    </div>
    <div class="container">
        <div class="row text-center mx-mt-5">
            <?php
                $result = $database->getData();
                while ($row = mysqli_fetch_assoc($result)){
                    component($row['product_name'], $row['product_price'],
$row['product_image'], $row['id']);
                }
            ?>
        </div>
    </div>
</section>
</div>
<div id="footer">
    <footer>
        <h3>For orders contact: 9090900909</h3>
        <h3>Location: Tambaram, Chennai</h3>
        <h3>Branches: Trichy, Karur, Salem, Coimbatore</h3>
        <h3>Facebook: cakedelights</h3>
        <h3>Instagram: cakedelights</h3>
        <h3>Mail: cakedelights@gmail.com</h3>
    </footer>
</div>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js
" integrity="sha384-
U02eT0CpHqdSJK6hJty5KVphtPhzWj9W01clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaEfF/nJGzIxFDs4x0xIM+B07jRM"
crossorigin="anonymous"></script>
</body>

</html>

```

B.2. createdb.php

```
<?php

class CreateDb
{
    public $servername;
    public $username;
    public $password;
    public $dbname;
    public $tablename;
    public $con;

    // class constructor
    public function __construct(
        $dbname = "Newdb",
        $tablename = "Productdb",
        $servername = "localhost",
        $username = "root",
        $password = ""
    )
    {
        $this->dbname = $dbname;
        $this->tablename = $tablename;
        $this->servername = $servername;
        $this->username = $username;
        $this->password = $password;

        // create connection
        $this->con = mysqli_connect($servername, $username, $password);

        // Check connection
        if (!$this->con){
            die("Connection failed : " . mysqli_connect_error());
        }

        // query
        $sql = "CREATE DATABASE IF NOT EXISTS $dbname";

        // execute query
        if(mysqli_query($this->con, $sql)){

            $this->con = mysqli_connect($servername, $username, $password,
$dbname);
```

```

        // sql to create new table
        $sql = " CREATE TABLE IF NOT EXISTS $tablename
                (id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
                 product_name VARCHAR (100) NOT NULL,
                 product_price FLOAT,
                 product_image VARCHAR (100)
                );";

        if (!mysqli_query($this->con, $sql)){
            echo "Error creating table : " . mysqli_error($this->con);
        }

        }else{
            return false;
        }
    }

    // get product from the database
    public function getData(){
        $sql = "SELECT * FROM $this->tablename";

        $result = mysqli_query($this->con, $sql);

        if(mysqli_num_rows($result) > 0){
            return $result;
        }
    }
}

```

B.3. component.php

```

<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="./cakestyling.css">
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
</html>
<?php
function component($productname, $productprice, $productimg, $productid){
    $extra=200;
    $old=$productprice+$extra;
    $element = "

```

```

        <div class=\"col-md-3 col-sm-6 my-3 my-md-0\">
            <form action=\"index.php\" method=\"post\">
                <div class=\"card shadow\">
                    <div>
                        <img src=\"$productimg\" alt=\"Image1\"
class=\"img-fluid card-img-top\">
                    </div>
                    <div class=\"card-body\">
                        <h5 class=\"card-title\">$productname</h5>

                        <h5>
                            <small><s class=\"text-
secondary\">$old</s></small>
                            <span class=\"price\">Rs. $productprice</span>
                        </h5>
                        <button type=\"submit\" class=\"btn btn-warning
my-3\" name=\"add\">Add to Cart <i class=\"fas fa-shopping-
cart\"></i></button>
                        <input type='hidden' name='product_id'
value='$productid'>
                    </div>
                </div>
            </form>
        </div>
        ";
        echo $element;
    }

function cartElement($productimg, $productname, $productprice, $productid){
    $element = "

        <form action=\"cart.php?action=remove&id=$productid\" method=\"post\"
class=\"cart-items\">
            <div class=\"border rounded\">
                <div class=\"row bg-white\">
                    <div class=\"col-md-3 pl-0\">
                        <img src=$productimg alt=\"Image1\"
class=\"img-fluid\">
                    </div>
                    <div class=\"col-md-6\">
                        <h4 class=\"pt-2\">$productname</h4>
                        <small class=\"text-secondary\">Seller:
dailytuition</small>
                        <h5 class=\"pt-2\">$$productprice</h5>
                        <button type=\"submit\" class=\"btn btn-
warning\">Save for Later</button>
                        <button type=\"submit\" class=\"btn btn-danger
mx-2\" name=\"remove\">Remove</button>

```

```

        </div>

        </div>
    </div>
</form>

";
echo $element;
}

```

B.4. cart.php

```

<?php

session_start();

require_once ("./createdb.php");
require_once ("./component.php");

$db = new CreateDb("Productdb", "Producttb");

if (isset($_POST['remove'])){
    if ($_GET['action'] == 'remove'){
        foreach ($_SESSION['cart'] as $key => $value){
            if($value["product_id"] == $_GET['id']){
                unset($_SESSION['cart'][$key]);
                echo "<script>alert('Product has been Removed...!')</script>";
                echo "<script>window.location = './cart.php'</script>";
            }
        }
    }
}

if(isset($_POST['buy'])){
    echo"<script>window.location='./buy.php'</script>";
}

?>

<?php
    require_once ('./header.php');

```

```

?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Cart- Cake Delights</title>
</head>
</html>

<div class="container-fluid">
    <div class="row px-5">
        <div class="col-md-7">
            <div class="shopping-cart">
                <h5>My Cart</h5>
                <hr>

                <?php

                    $total = 0;
                    if (isset($_SESSION['cart'])){
                        $product_id = array_column($_SESSION['cart'],
'product_id');

                        $result = $db->getData();
                        while ($row = mysqli_fetch_assoc($result)){
                            foreach ($product_id as $id){
                                if ($row['id'] == $id){
                                    cartElement($row['product_image'],
$row['product_name'],$row['product_price'], $row['id']);
                                    $total = $total +
(int)$row['product_price'];
                                }
                            }
                        }
                    }else{
                        echo "<h5>Cart is Empty</h5>";
                    }

                ?>

            </div>
        </div>
        <div class="col-md-4 offset-md-1 border rounded mt-5 bg-white h-25">

            <div class="pt-4">
                <h6>PRICE DETAILS</h6>
                <hr>

```

```

        <div class="row price-details">
            <div class="col-md-6">
                <?php
                    if (isset($_SESSION['cart'])){
                        $count = count($_SESSION['cart'])-1;
                        echo "<h6>Price ($count items)</h6>";
                    }else{
                        echo "<h6>Price (0 items)</h6>";
                    }
                ?>
                <h6>Delivery Charges</h6>
                <hr>
                <h6>Amount Payable</h6>
            </div>
            <div class="col-md-6">
                <h6>${<?php echo $total; ?></h6>
                <h6 class="text-success">FREE</h6>
                <hr>
                <h6>${<?php
                    echo $total;
                ?></h6>

            </div>
        </div>
        <form action="buy.php" method="post">
            <button type="submit" class="btn btn-danger my-5"
name="buy" >Proceed to Buy</button>
        </form>
    </div>

</div>

</div>

</div>
</div>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>

```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js
" integrity="sha384-
U02eT0CpHqdsJQ6hJty5KVphtPhzWj9W01c1HTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/njGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
</body>
</html>

```

B.5 add_to_db.php:

```

<?php
require_once('./header.php');
$conn=mysqli_connect("localhost","root","","Productdb");

if($conn === false){
    echo "connection failed";
    exit();
}

$name=$_REQUEST['name'];
$address=$_REQUEST['address'];
$phoneno=$_REQUEST['phno'];
$paymethod=$_REQUEST['paymethod'];

$sql = "INSERT INTO orderdetails VALUES (NULL,'$name','$address',$phoneno,
'$paymethod')";
if(mysqli_query($conn,$sql)){
    echo "<center><h2>Thankyou for the order!!</h2><center>";
    echo "<center><h2>Order Placed Successfully!!</h2></center>";
}
else{
    echo "Error while placing the order!!".mysqli_error($conn);
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Order Confirm- Cake Delights</title>
</head>
</html>

```

```
<?php
    require_once('header.php');
?>

<!DOCTYPE html>
<html>
<head>
    <title>Checkout- Cake Delights</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="cakestyling.css">
</head>
<body>
    <h2><center>Payment Form</center></h2>

    <h4>
    <div class="form">
        <form action="./add_to_db.php" name="paymentdetails" method="post">
            <label> Name:      <br><br>
            <input type="text" name="name"><br><br>
            <label> Address:      <br><br>
            <input type="text" name="address"><br><br>
            <label> Phone Number:      <br><br>
            <input type="text" name="phno" maxlength="10"><br><br>
            <label> Payment method: </label>
            <input type="radio" name="paymethod" value="cod"> Cash on
            Delivery      <br><br>
            <input type="radio" name="paymethod" value="card">Debit/Credit
            Card      <br><br>
            <input type="radio" name="paymethod" value="netbanking">Net
            Banking<br><br>
            <button type="submit" class="btn btn-danger my-5" >Proceed to
            Pay</button>
        </form>
    </div>
</h4>
```

```
</body>
</html>
```

B.7 header.php

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="./cakestyling.css">

  <link href='https://fonts.googleapis.com/css?family=Aladin'
rel='stylesheet'>
  <link href='https://fonts.googleapis.com/css?family=Averia Serif Libre'
rel='stylesheet'>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0-beta2/css/all.min.css" integrity="sha512-
YWzhKL2whUzgiheMoBFwW8CKV4qpHQA Euvilg9FAn5VJUDwKZZxkJNuGM4XkWuk94WCrrwslk8yWNG
mY1EduTA==" crossorigin="anonymous" referrerpolicy="no-referrer" />
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
</head>
</html>
<header class="topheader">
  <div class="header1">
    <h2>Cake Delights</h2>
    <h3>Indulge in the world of deliciousness</h3>
  </div>
  <div class="home">

    <ul>
      <li><a href="./index.php">Home</a></li>
      <li><a href="./cart.php">Cart</a></li>
    </ul>

  </div>
  <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
    <div class="mr-auto"></div>
    <div class="navbar-nav">
      <a href="./cart.php" class="nav-item nav-link active">
        <h5 class="px-5 cart">
```

```

        Cart
        <?php

            if (isset($_SESSION['cart'])){
                $count = count($_SESSION['cart'])-1;
                echo "<span id=\"cart_count\" class=\"text-warning
bg-light\">$count</span>";
            }else{
                echo "<span id=\"cart_count\" class=\"text-warning
bg-light\">0</span>";
            }

        ?>
    </h5>
    </a>
</div>
</div>

</nav>
</header>

```

B.8. cakestyling.css

```

html{
    scroll-padding-top: 100px;
}

*{
    margin-top: 0;
    margin-bottom: 0;
    padding: 0;
}
body{
    background: #e4e2e7;
    scroll-padding-top: 120px;
}

.header1 h2{
    font-family: 'Aladin';
    color: #6c1819;
}

```

```

    font-size: 35px;
    text-align: left;
    padding-top: 10px;
    padding-left: 40px;
}

.header1 h3{
    font-family: 'Aladin';
    font-size: 12px;
    color: rgba(102, 51, 153, 0.8);
    padding-left: 45px;
}

.topheader{
    font-family: 'Averia Serif Libre';
    margin-left: 0;
    margin-right: 0;
    padding: 0;
    border: 11px;
    height: 10vh;
    width: 100%;
    background-color: #b0e8e3;
}

.middle{
    background-image: url("../images/topcake11.jpg");
    height: 120vh;
    background-size: cover;
    background-position: center;
}

.middle h1{
    position: absolute;
    left: 1000px;
    top: 250px;
    font-family: 'Aladin';
    color: #6c1819;
    font-size: 85px;
}

.caption h1{
    font-family: 'Aladin';
    font-size: 25px;
    color: rgba(102, 51, 153, 0.5);
    width: 600px;
    left: 900px;
    top: 350px;
    text-align: center;
    word-wrap: break-word;
}

.word{

```

```

    font-size: 28px;
    color: rgba(102, 51, 153, 0.8) ;
}
div > ul{
    align-items: right;
    list-style-type: none;
    margin: 0;
    padding: 0;
}
div li{
    float: left;
}
li a{
    display: inline-block;
    border-width: 1.5px;
    background-color: #b0e8e3;
    color: #616646;
    font-size: 25px;
    text-align: center;
    padding: 14px 16px;
}
li a:hover{
    background-color: white;
    color: black;
}

.header1{
    float: left;
}
.home{
    float: right;
}
section{
    padding-top: 65px;
    margin-left: 90px;
    margin-right: 90px;
}

.subtitle{
    font-family: 'Averia Serif Libre';
    text-align: left;
    color: #6c1819;
    padding-top: 18px;
    padding-bottom: 5px;
    font-size: 17px;
}

```

```

img{
  max-width: 100%;
  height: 100%;
}
/* .card-shadow img{
  width: 300px;
  height: 200px;
}*/
#cart_count{
  text-align: center;
  padding: 0 0.9rem 0.1rem 0.9rem;
  border-radius: 3rem;
}

.shopping-cart{
  padding: 3% 0;
}

.cart-items + .cart-items{
  padding: 2% 0;
}

.price-details h6{
  padding: 3% 2%;
}

footer{
  position: fixed;
  height: 100%;
  width: 100%;
  background-color: #b0e8e3;
  display: inline-block;
  position: static
}
footer > h3{
  padding-left: 30px;
  font-family: 'Averia Serif Libre';
  color: #616646;
  margin: 20px;
}

.form{
  margin-left: 10px;
  padding-left: 20px;
}

```

B.9 producttb.sql

```
-- phpMyAdmin SQL Dump
-- version 5.0.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Nov 11, 2021 at 05:09 PM
-- Server version: 10.4.11-MariaDB
-- PHP Version: 7.3.15

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `productdb`
--

--
-- Table structure for table `producttb`
--

CREATE TABLE `producttb` (
  `id` int(11) NOT NULL,
  `product_name` varchar(25) NOT NULL,
  `product_price` float DEFAULT NULL,
  `product_image` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `producttb`
--

INSERT INTO `producttb` (`id`, `product_name`, `product_price`,
`product_image`) VALUES
(1, 'Blackforest cake', 1000, './images/blackforestresize.jpg'),
(2, 'Rainbow cake', 1300, './images/rainbowcake.jpg'),
(3, 'Chocolate coffee cake', 1000,
'./images/chocolatecoffeecakeresize.jpg'),
```

```

(4, 'Chocolate Oreo cake', 1200,
 './images/chocolateoreoresize.jpg'),
(5, 'Butterscotch cake', 1000, './images/butterscotchresize.jpg'),
(6, 'Vanilla cake', 1000, './images/vanillaresize.jpg'),
(7, 'Pineapple cake', 1150, './images/pineappleresize.jpg'),
(8, 'Redvelvet cake', 1200, './images/redvelvetresize.jpg'),
(9, 'Vanilla cupcake ', 300, './images/vanillacupresize.jpg'),
(10, 'Redvelvet cupcake ', 400, './images/redvelvetcupresize.jpg'),
(11, 'Mango cupcake ', 350, './images/mangocupcakeresize.jpg'),
(12, 'Pistachio cupcake ', 450, './images/pistachioresize.jpg'),
(13, 'Chocolate bread', 350, './images/chocolatebreadresize.jpg'),
(14, 'Wheat bread', 400, 'images/gluttenbreadresize.jpg'),
(15, 'Eggless marple bread', 350, './images/marplebreadresize.jpg'),
(16, 'Raspberry chocolate ', 400,
 './images/raspberrybreadresize.jpg');

--
-- Indexes for dumped tables
--

--
-- Indexes for table `producttb`
--
ALTER TABLE `producttb`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `producttb`
--
ALTER TABLE `producttb`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=17;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```