

ONLINE CATERING SERVICES WEBPAGE

Submitted in partial fulfilment of the requirements for the
award of

Bachelor of in Computer Science

BY

I.KARKUZHALI (38290032)

R.GAYATHIRI (38290019)



DEPARTMENT OF COMPUTER SCIENCE

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A” by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119**

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **I.KARKUZHALI (Reg no 38290032)** and **R.GAYATHIRI (Reg no 38290019)** who have done the project work as a team who carried out the project entitled "**ONLINE CATERING SERVICES WEBPAGE**" under my supervision from November 2020 to March 2021.

Internal Guide

Dr.J.ALBERT.MAYAN.ME.Ph.D.

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I.KARKUZHALI (Reg no 38290032) and R.GAYATHIRI (Reg no 38290019) hereby declare that the project report entitled **ONLINE CATERING SERVICES WEBPAGE** done by me under the guidance of **Ms.J.Refonaa** is submitted in partial fulfilment of the requirements for the award of Bachelor of degree in Computer Science.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** 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. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering 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 **Ms.J.REFONAA** 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 Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT OF PROJECT:

Food is one of the most important things in our life. We all live for food and with food. And for a good food we need a good chef and for a good chef we need a good catering service and for a good catering service we need a best website for that to interact and we are providing you the best website in our today life. By talking about life we need a basic necessary thing like to have luxuries, so as the websites we also need the basics so in our web site we are providing sign in details to interact with the customer and to give the best we have and after that we have a home page full of info what the catering service has by their own to offer others and further in our website the user can pay by them self and so we have a payment option too. All our details are clearly displayed on our website. Some points we get from the users are given below

Customer details and clear view of orders are accepted.

Mainly concentrated for the use of all the customers and easy for use.

Users can sign up and have their separate password for their account.

Users can check all our catering service details on our official website. In case of any queries they can contact us through email or phone number etc.

LIST OF ABBREVIATIONS

HTML - Hypertext Markup Language

CSS - Cascading Style Sheets

XHTML - Extensible Hypertext Markup Language

SEO - Search Engine Optimization

RAM – Random Access Memory

IDE - Integrated Development Environment

AVD - Android Virtual Device

ADT - Android Development Tools

SDK - Software Development Kit

	TABLE OF CONTENTS	
CHAPTER NO	TITLE	PAGE NO
	ABSTRACT OF PROJECT	5
	LIST OF ABBREVIATIONS	
1	INTRODUCTION	7
	1.1 GENERAL	
	1.2 BACKGROUND OF STUDY	
	1.3 OVERVIEW	8
2	LITERATURE SURVEY	9
	1.1 GENERAL	
	1.2 SOME OF THE RELATED WORKS LIKE US	
3	AIM AND SCOPE TO PRESENT	12
	3.1 GENERAL	
	3.2 AIM	
	3.3 OVERVIEW OF THE PLATFORM	
	3.4 ADVANTAGE	

	3.5 DISADVANTAGE	
	3.6 OBJECTIVE AND SCOPE	
	3.7 HARDWARE REQUIREMENTS	
	3.8. SOFTWARE REQUIREMENTS	
4	EXPERIMENTAL OR MATERIAL METHOD	14
	4.1 DESIGN METHODOLOGY	
	4.2 EXISTING SYSTEM	
	4.3. PROPOSED SYSTEM	
	4.4 MODULE DESCRIPTION	
	4.5. ARCHITECTURE	
	4.6. FLOWCHART	
5	RESULT AND PERFORMANCE	18
	5.1 GENERAL	
6	CONCLUSION	20
	6.1. CODING	
	6.2. OUTPUT	

CHAPTER 1

INTRODUCTION

1.1 GENERAL: -

The catering industry comprises businesses that provide food, beverages and other services to a variety of clients, usually for special events. Catering industry jobs vary from small intimate affairs to large events that involve providing not only food and beverage service but also tableware, linens, service personnel and other aspects of the event. Companies that provide catering services might be as large as a business with several locations or as small as an individual who provides catering services on his or her own. A caterer might be part of a restaurant or have his or her own business. Caterers might make use of independent contractors for some aspects of their catering service.

1.2 Background of study: -

Catering is important as a large part of any event or party revolves around food. The food you present to your guests can either make or break your event, hence investing in a catering company is the best way out. Hiring an experienced catering company will not only save your time but also take off a huge burden from your shoulders by providing delicious food and quality service.

Why do we need a catering website?

The logo, interior design, signage, website, app, everything is consistent, professional and hyper-targeted towards the millennial Chicago market.

In 2020, every small business, caterers included, should have incredible design. If you don't invest in design, your business is susceptible to any company that does.

Great design is the foundation of a great catering website.

Website in our day-to-day life: -

Facebook, Myspace, Yahoo, Google, and Twitter are all websites that people use every day, but how did they get there and how're they created? Website design is a process of assorting and linking codes and graphics into a text editor such as notepad and then displaying it onto a web browser. There are a large amount of things to remember for website design. Ranging from [HTML](#) , CSS, and XHTML codes to scripting languages like JavaScript. Website design has a vast history considering it is a fairly young profession

From a developer's point of view, the website will be created successfully by following the client's clear instructions as well as meeting the four major requirements of every passable or effective website. This includes that of the readability, functionality, design and accuracy. The website is required to have six webpages including the homepage, about us, history, communication, contact us and help. Once the website is transferred to a web server and up and running there should be something to evaluate how many users use the website and take into account customer feedback to make the website better. As instructed by the client, the Indigenous Language Trust, the team created a website to raise awareness towards the topic of Indigenous cultures. This site included multiple web pages such as a home page, which includes a supporting image towards the topic and some space for the client details, a page called history, which gives a brief history of the topic and a supporting image, a page called communications, used to teach the correct terminology and languages spoken by the Aboriginal culture.

1.3 Overview: -

The third and main difference between on-site and off-site catering is presentation. When it comes to catering on-site, the sky's the limit in presentation. Off-site catering sometimes suffers in presentation due to the moving, jostling, and re-plating. If you're using fresh ingredients that need to be served in a time-sensitive way, or if your dishes must be kept at a certain temperature, using an on-site caterer may be important for presentation. Even the most delicious food prepared by the best caterer can quickly become a disappointment if it is lukewarm, for instance.

Regardless of whether you choose on-site or off-site catering, your caterer can make all the difference. And the right caterer will work with you to create an event that you'll want to talk about for years to come

CHAPTER -2

LITERATURE SURVEY

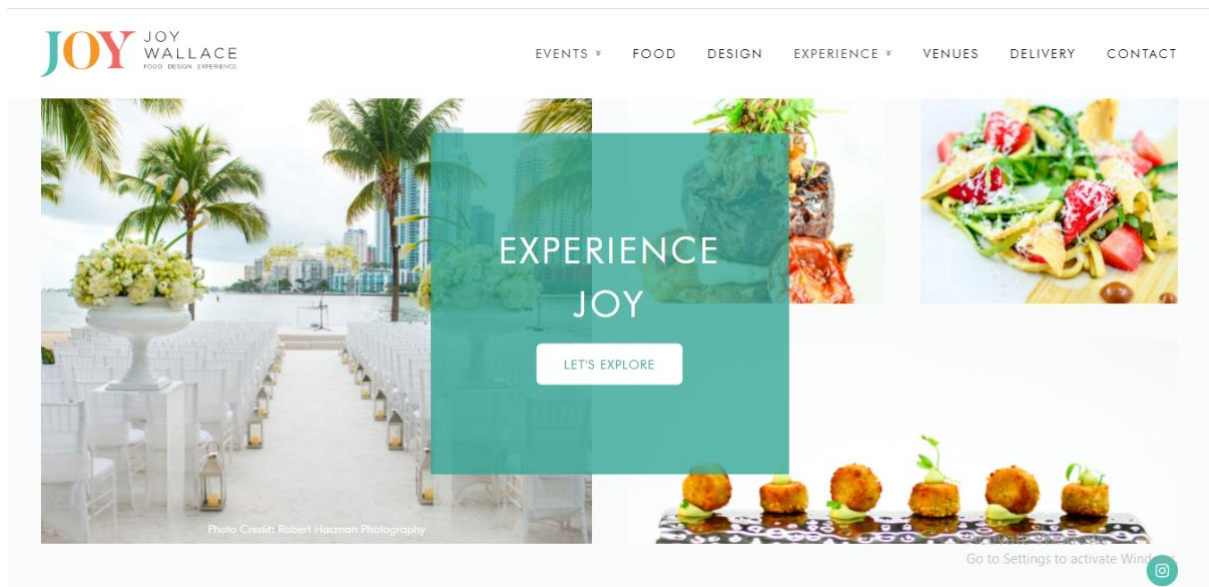
2.1 GENERAL: -

For developing any application, the ground work must be done. The ground work includes the study of various available resources for developing the applications. This chapter deals with the literature survey done for developing the application.

RELATED WORKS: -

2.2 SOME OF THE RELATED WORKS LIKE US

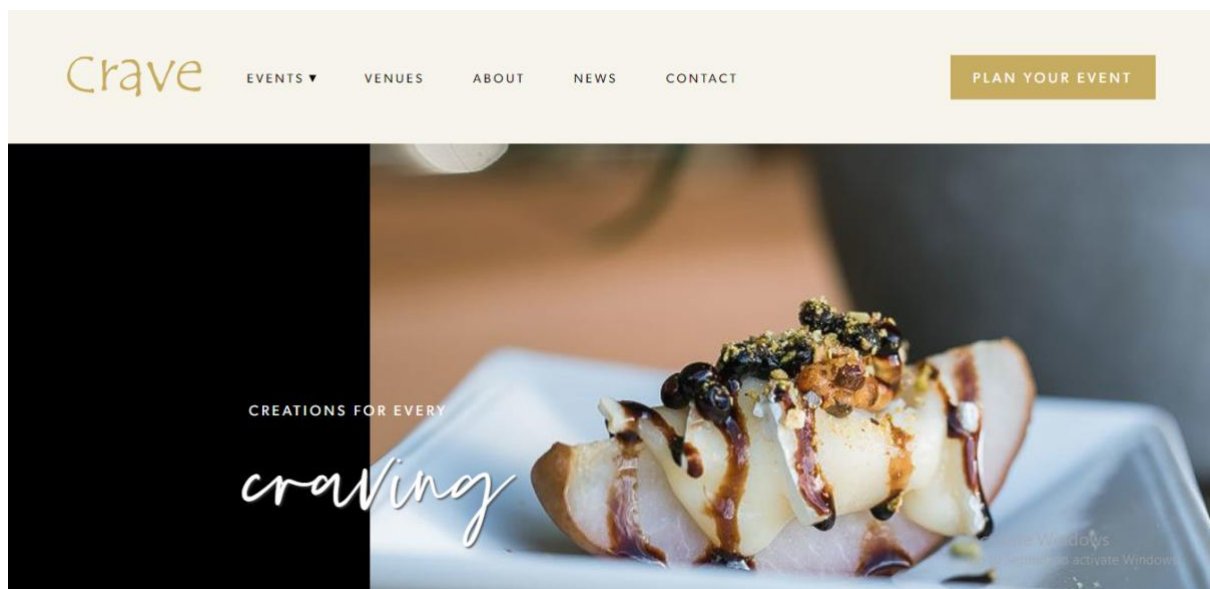
- i) **JOY WALLACE**
Joy Wallace Catering has been revered as one of Miami's most trusted Information Technology Director at A Joy Wallace Catering Production & Design Team



- ii) **ABOVE AND BELOW CATERING SERVICE: -**



iii)Cave catering service: -



iv)QUEEN OF HEARTS CATERING: -

WEDDINGS EVENTS CORPORATE

QUEEN OF HEARTS
CATERING

ABOUT VENUES CONTACT



v) FORK&SPOON PRODUCTIONS

PRIVATE EVENTS WEDDINGS CORPORATE

FORK & SPOON
PRODUCTIONS

VENUES ABOUT CONTACT

EFFORTLESS
ENTERTAINING

SAN FRANCISCO

LEARN MORE

Activate Windows
Go to Settings to activate Windows.

CHAPTER 3

AIM AND SCOPE TO PRESENT: -

3.1 GENERAL: -

System analysis as the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way. Another view system as a problem solving technique that breaks down a system into its component pieces for the purpose of studying how well these component parts work and interact to accomplish their purposes.

3.2 AIM: -

The basic aim of our project is to provide the best catering service website. We are aiming that our website will be user friendly and it will also be a good and fast website among the websites of now. We provide a scope of making the better to the best and we do it with our full effort.

3.3 Overview of the platform: -

This platform is useful for every user who is willing to book a bulk amount of orders to a specific catering service, which can be done in a simpler manner and a must easy way than booking it by voice call or direct talking. It is also full to the catering service who use this platform where we get a well- defined information about the things that the customer wants and they can market themselves with creative or attractive coverings in their web site. A most important thing in this platform is that it is the best among marketing and advertising. I think that framing our work in this way lets us web designers explore exciting new possibilities. Instead of throwing a big carousel on the page and being done with it, thinking about making a website like a catering company encourages to focus on the tough questions. We have a home, we have to provide evidence for our statements, we have a conclusion, etc. This way we don't have to get so caught up in the same old patterns that we've tried again and again in our work.

3.4 ADVANTAGE: -

- This is user friendly
- Gives a well -defined information about the catering services to the user
- Can identify the user by the catering service
- Attract people and make their work easy
- Users can see all the details about the catering service in one page and access through it.

3.5 DISADVANTAGE: -

***reliability information can be shown by the catering service which uses this platform.**

*** a website can be hacked or can also be crashed if we didn't do our work in the right manner.**

***by using the internet to access the webpage we can get to attack with spam.**

3.6 OBJECTIVE AND SCOPE: -

The objective of a catered conference meal is to feed a large group of people quickly and by an easy method of booking online, so they can get back to their conference activities. For this type of event, it often makes sense to have multiple lines serving the same selection of food so more people can get served in minimal time.

And the main scope of this project was to make the people happy by lowering their work and making a good feel of the catering service. The main objective of a good catering service website is user-friendly to the right audience, easy for that audience to find, and provides information the targeted audience wants and needs. If these things can be stated and planned out, having good project scope is halfway defined.

3.7 HARDWARE REQUIREMENTS: -

- Processor Pentium P4**
- Motherboard Genuine Intel**
- RAM Min 1GB**
- Hard disk 80GB**
- Mobile Android Based Smartphones**

3.8 SOFTWARE REQUIREMENTS: -

- Operating system Windows XP**
- Technology used Android**
- IDE Android studio**
- Emulators AVD**
- Plug-in ADT plug-in**
- Database Firebase**
- Tools used Android SDK**

CHAPTER 4

EXPERIMENTAL OR MATERIAL METHOD

4.1 DESIGN METHODOLOGY: -

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to an effective system. The term "design" is defined as "the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

1. **Goal identification:** Where I work with the client to determine what goals the new website needs to fulfil. I.e., what its purpose is.
2. **Scope definition:** Once we know the site's goals, we can define the scope of the project. I.e., what web pages and features the site requires to fulfil the goal, and the timeline for building those out.
3. **Sitemap and wireframe creation:** With the scope well-defined, we can start digging into the sitemap, defining how the content and features we defined in scope definition will interrelate.
4. **Content creation:** Now that we have a bigger picture of the site in mind, we can start creating content for the individual pages, always keeping search engine optimization (SEO) in mind to help keep pages focused on a single topic. It's vital that you have real content to work with for our next stage:
5. **Visual elements:** With the site architecture and some content in place, we can start working on the visual brand. Depending on the client, this may already be well-defined, but you might also be defining the visual style from the ground up. Tools like style tiles, mood boards, and element collages can help with this process.
6. **Testing:** By now, you've got all your pages and defined how they display to the site visitor, so it's time to make sure it all works. Combine manual browsing of the site on a variety of devices with automated site crawlers

to identify everything from user experience issues to simple broken links.

7. **Launch:** Once everything's working beautifully, it's time to plan and execute your site launch! This should include planning both launch timing and communication strategies. After that's time to release the giant.

4.2 EXISTING SYSTEM: -

Used for describing something that exists now, especially when it might be changed or replaced. The existing system needs to be changed. the demolition of existing buildings to make way for new office blocks. In the existing system only a crime can be reported, also it has more workload for the authorized person, but in the case of Proposed System, the user can register the site and send the first information report or complaint about a particular situation or person. The disadvantage of the existing system is More man power Time consuming, consumes large volume of paperwork, Needs manual calculations, No direct role for the higher officials.

To avoid all these limitations and make the working more accurate the system needs to be computerized in a better way.

4.3 PROPOSED SYSTEM:

The aim of the proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The existing system has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system will help the user to reduce the workload and mental conflict. The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has the following features; Minimize manual data entry, Minimum time needed for the various processing, Greater efficiency, Better service, User friendliness and interactive, Minimum time required. The transactions should take place in a secured format between various clients in the network. It provides flexibility to the user to transfer the data through the network very easily by compressing the large amount of file. It should also identify the user and provide the communication according to the prescribed level of security with transfer of the file requested and run the required process at the server if necessary. In this system the data will be sent through the network as an audio file. The user who received the file will do the operations like de embedding, decryption, and decompress in their level of hierarchy etc.

4.4 MODULE DESCRIPTION: -

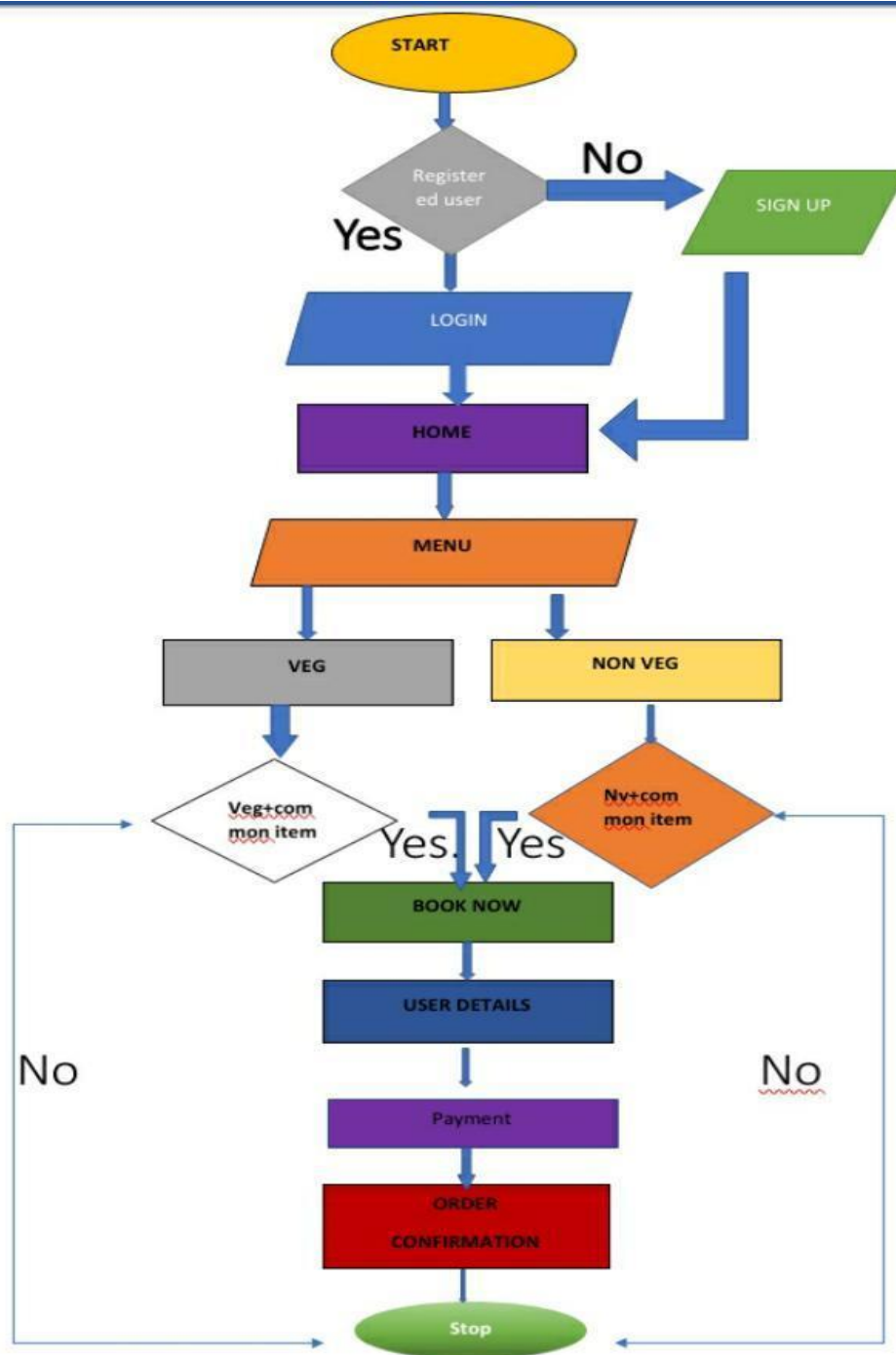
A module description provides detailed information about the module and its supported components, which is accessible in different manners. The included description is available by reading directly, by generating a short html-description, or by making an environment check for supported components to check if all needed types and services are available in the environment where they will be used. This environment check could take place during registration/installation or during a separate consistency check for a component.

4.5 ARCHITECTURE: -

Site architecture refers to the structure that organizes and delivers the content on your website. It includes the hierarchy of pages where users find content and the technical considerations that let search engine bots crawl your pages.

In our architecture we provide a website with design and code knowledge. This interface contains an editor and necessary tools to select items such as vegetarian, non- vegetarian, home buttons, background images, buttons, tables, boxes, forms etc.

4.6 FLOW CHART



EXPLANATION OF THE FLOW CHART:

At first we start and we go with login. At login you need to sign up by entering your email password. And then you go to the home page, where you can book the food you need and you are provided with the information of veg and non-veg.

After that you can book your orders by giving your user details payment options and order confirmation.

CHAPTER 5

Result and performance: -

5.1 GENERAL

This chapter explains about the implementation details of this project. Implementation is the most crucial stage in achieving a successful system and gives the user confidence that the new system is workable and effective. Implementation is the stage of the project where the theoretical design is turned out into a working system.

LOGIN PAGE: -

Login page has boxes containing input information's of

i) User name

ii) Password

HOME PAGE: -

Home page consists of the basic details of the catering services like,

i) Background design

ii) Contact details

iii) Menu

iv) About us

MENU PAGE: -

Menu page contains user interaction details,

i) Veg

ii) non-veg

iii) Book now

USER DETAILS PAGE: -

It contains,

i)First name

ii) Last name

iii)Email ID

iv)Password

v)Confirm password

vi)Phone number

vii)Food type

viii)Number of member

ix)Click register

CONFIRM PAGE: -

Confirm page consists of elements like,

i)First name

ii)Last name

iii)Email ID

iv)Phone number

v)Food type

vi)No. of. Members

vii)Click home link. its go to the home page.

CHAPTER 6

CONCLUSION

CONCLUSION: -

This website can perform the following operations and the result also relies on the functioning of the system as expected. The website is able to login and select the food that the user needs and book the food and also they can do the payment through online. We can use the website to book the food and pay through online. It is confirmed that it works as per expected. We have used advanced Java and created a catering service webpage. We have clearly explained our project using flowchart and PPT. We have connected our project using servlet

6.1 CODING:

```
<html>

<head>

    <title>TODO supply a title</title>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <style>

        Body{

            background-image:url(6.jpg);

            background-repeat: no-repeat;

            background-size:cover;

            height:140px;

            width:100px;

        }

        .signup{

            position:absolute;top:15%;left:30%;

        }
```

```

.ss{
border:21px ridge red;padding-left:80px;padding-right:80px;border-radius:40px 40px
40px;
    }
    h3{
        position:absolute;
        top:17%;
        left:43%;
        font-weight:italic;
font-size:32px;
        font-family:"forte";
    }
    label{
        font-family:"forte";
        font-size:24px;
    }
</style>
</head>
<body>
    <h3>SIGN UP</h3><br><br>
    <div class="signup">
        <div class="ss">
            <form action="ServletHome"> <br><br><br>
<br><br> <label><b>USER NAME</b></label>
            <input type="text" name="username" id="uname1" placeholder="username">
<br><br> <label><b>EMAIL ID</b></label>
            <input type="email" name="email" id="eee" placeholder="enter your email
id"><br><br>

```

```

<label> <b> PASSWORD </b></label>

<input type="password" name="password" id="pass1"
placeholder="password"><br><br>

<label> <b> CONFIRM PASSWORD </b></label>

<input type="password" name="password1" id="pass11" placeholder="password"
required><br><br>

<input type="submit" name="go to next page" id="next" style="color:black;border-
radius:17px;width:40%;height:200%;text-align:center;background-color:lightblue;font-
style:italic;font-weight:bold;" value="GO TO NEXT PAGE">

    <br><br><br><br>

</form>

</div></div>

</body>

</html>

```

```

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author ELCOT
 */

public class ServletHome extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and
     <code>POST</code>

```

```

    * methods.
*
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */

    protected void processRequest(HttpServletRequest request re
quest, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) /* TODO output your page here.
You may use the following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet ServletHome</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet ServletHome at " + request.getContextPath() +
"</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">

    /**

```

```

    * Handles the HTTP <code>GET</code> method.
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */

    @Override

    protected void doGet(HttpServletRequest request, HttpServletResponse
response)

        throws ServletException, IOException    //processRequest(request,
response);

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

RequestDispatcher dispatcher=request.getRequestDispatcher("/home.html");
dispatcher.forward(request,response); }

}

<html>

<head>

<title>Catering</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

    body{

        background-image: url(home.jpg);

        background-repeat:no-repeat;

        background-size: cover;

        height:70px;

        width:60px;

```

```

}
h2{
    position:absolute;
    font-family: castellar;
    top:40%;left:55%;
    font-size: 45px;
    font-style: italic;
    background-color:skyblue;
}
h4{
    position:absolute;
    font-family: times new roman;
    top:85%;right:80%;
    font-size: 22px;
    font-style: italic;
    border-color: white;
    background-color:orange;
}
h3{
    position:absolute;
    font-family: castellar;
    top:50%;left:70%;
    font-size: 20px;
    font-style: italic; }
#aboutus{
    background-color:yellow;
    font-weight: bold;

```

```
font-size: 20px;
border-radius: 17px;
top:5%;
text-align: center;
}

#contactus{
background-color:yellow;
font-weight: bold;
font-size: 20px;
border-radius: 17px;
top:5%;
text-align: center;
}

#menu{
background-color:yellow;
font-weight: bold;
font-size: 20px;
border-radius: 17px;
top:5%;
text-align: center;
}

#home{
background-color:yellow;
font-weight: bold;
font-size: 20px;
border-radius: 17px;
top:5%;
```

```

        text-align: center;
    }
</style>
</head>
<body>
    <form action="Servlet4">
        <div class="pp">
            <table>
                <tr>
                    <td> <input type="submit" name="oo" id="home" value="HOME"><br><hr></td>
                    <td><input type="submit" name="oo" id="contactus" value="CONTACT
US"><br><hr></td>
                    <td> <input type="submit" name="oo" id="menu" value="MENU"><br><hr></td>
                    <td> <input type="submit" name="oo" id="aboutus" value="ABOUT US"><hr></td>
                </tr>
            </table>
        </div></form>
<h2> <br>WELCOME TO K.K CATERING SERVICE<br></h2>
<h4>EMAIL:KK123@gmail.com<br>
    CONTACT No:9876543210</h4>
</body></html>

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;

```

```

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author ELCOT
 */
@WebServlet(urlPatterns = {"/Servlet4"})
public class Servlet4 extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and
     POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse
    response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        try (PrintWriter out = response.getWriter()) {

            /* TODO output your page here. You may use the following sample code. */
            out.println("<!DOCTYPE html>");

            out.println("<html>");

            out.println("<head>");

```

```

        out.println("<title>Servlet Servlet4</title>");

        out.println("</head>");

        out.println("<body>"); out.println("<h1>Servlet Servlet4 at " +
request.getContextPath() + "</h1>");

        out.println("</body>");

        out.println("</html>");

    }}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">

/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response)

    throws ServletException, IOException {

    // processRequest(request, response);

    String rr=request.getParameter("oo");

    if(rr.equals("HOME")){

        RequestDispatcher di=request.getRequestDispatcher("/home.html");

        di.forward(request,response);

    }

    if(rr.equals("CONTACT US")){

```

```

        RequestDispatcher di=request.getRequestDispatcher("/cont2.html");
        di.forward(request,response);
    }
    if(rr.equals("MENU")){
        RequestDispatcher di=request.getRequestDispatcher("/menu2.html");
        di.forward(request,response);
    }
    else{
RequestDispatcher di=request.getRequeSt Dispatcher("/about2.html");
        di.forward(request,response);
    }
}
}
}
<html>
<head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        body{
            background-image:url(2.jpg);
            background-repeat: no-repeat;
            background-size:cover;
            height:70px;
            width:60px;
        }
        h2{

```

```

        position:absolute;
        font-family:castellar;
        top:35%;left:40%;
        font-size:50px;
        font-style:italic;
        background-color:yellowgreen;
    }
    h4{
        position:absolute;
        top:45%;
        left:40%;
        font-family:times new roman,sansserif;
        font-style:italic;
        color:black;
        font-size:40px;
        border-color:white;
        background-color:palevioletred;
    }
</style>
</head>
<body>
    <form action="Servletcon">
<h2>CONTACT US:</h2>
<h4>NO:12 1st crosss street Thennampet,ch-34<br>
    CONTACT NO:9876543210 <br>
    FOLLOW US ON:<br>instagram_kk_catering service<br>
    EMAIL:KK123@gmail.com<br>

```

```

</h4>

<input type="submit" value="HOME" id="btn" name="HOME" />

</form>

</body>

</html>

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author ELCOT
 */

public class Servletcon extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and
     <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */

```

```
protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
```

```
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample code. */
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet Servletcon</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet Servletcon at " + request.getContextPath() +
"</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

```
// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
```

```
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs

```

```

*/

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse
response)

        throws ServletException, IOException {

    // processRequest(request, response)
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
RequestDispatcher
dispatcher=request.getRequestDispatcher("/home.html");
dispatcher.forward(request,response);

    }

}
<html>

<head>

    <title>TODO supply a title</title>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <style>

        body{

            background-image:url(food1.jpg);

            background-repeat: no-repeat;

            background-size:cover;}

        .veg1{

            line-height: 30px;

                word-spacing:8px;

            top:40%;

```

```

        font-family: Times new roman;
color:red;font-size:30px;font-weight:bold;font-style: italic;
    }
    .non1{
        line-height: 30px;
        word-spacing:8px;
        font-size: 30px;
font-family: Times new romancolor:red;font-size:30px;font-weight:bold;font-style:
italic;
    }
    .both{
        line-height: 30px;
        word-spacing:8px;
        font-family: Times new roman;
color:red;font-size:30px;font-weight:bold;font-style: italic;
    }
    .hh{
font-size:30px;text-align:center;text-decoration:underline;
        color:green;
        border-color: white;
        background-color:orange;  }
</style>
</head>
<body>

    <br><br>

    <br>

```



```

function checkPassword(form) {
    password = form.password.value;
    cpassword = form.cpassword.value;
    // var number = /^\d{10}$/;
    // If password not entered
    if (password == ""){
        alert ("Please enter Password");
        return false;
    }
    // If confirm password not entered
    else if (cpassword == ""){
        alert ("Please enter confirm password");
        return false;
    }
    // If Not same return False.
    else if (password != cpassword) {
        alert ("\nPassword did not match: Please try again...");
        return false;
    }
    //else if((number.value.match(number)){
    //    alert ("\n pls vandhuru.....");
    //    //return false;
    // }
    // If same return True.
    else{
        alert("Success!!");
        return true;
    }
}

```

```

    }
}
</script>
<style>
html{
    height:100px;
}
body{
    margin:0;
    padding:0;
    font-family:arial;
    background:black;
    background-image:url(2.jpg);
}
h2{
    text-align: center;
    color: black;
    padding: 20px;
}
.log{
    color:black;
    font-size: 23px;
    display: block;
    text-align: center;
}
.out{
    position:absolute;

```

```

    top:80%;
    left:50%;
    width:400px;
    padding:40px;
    transform:translate(-50%,-50%);
    background:lightgray;
    box-sizing:border-box;
    box-shadow:0 15px 25px rgb(169,169,169);
    border-radius:10px
}

.out .user-box{
    position:relative;
}

.out .user-box input{
    width:100%;
    padding:10px 0;
    font-size:16px;
    color:black;
    margin-bottom:30px;
    border:none;
    border-bottom:1px solid window;
    outline:none;
    background:transparent;
}

.out .user-box label{
    position:absolute;
    top:0;

```

```

        left:0;
        padding:10px 0;
        color:black;
        pointer-events:none;
        transition:.6s;
        font-size:14px;
    }

    .out .user-box input:focus~label,
    .out .user-box input:valid~label{
        top:-20px;
        left:0px;
        color:#03e9f4;
        font-size:12px;
    }

    #register{
        color:black;
    }

    label{
        color:red;
        margin-top:0px;
        font-size:11px;
        color:#03e9f4;
    }

    .nav
{
    background-color:black;
    overflow: hidden;

```

```

    text-align: center;
}
/* Style the links inside the navigation bar */
.nav a
{
    float: left;
    color: #f2f2f2;
    text-align:center;
    padding: 14px 18px;
    text-decoration: none;
    font-size: 17px;
}
</style>
</head>
<body>
    <div class="out">

<form name="det" id="det" action="userservlet" method="GET" onSubmit = "return
checkPassword(this)">
<div class="user-box">

    <input type="text" name="Firstname" id="uname" />
        <label> First name</label>
    </div>
    <div class="user-box">
<input type="text" name="lastname" id="lname" />
        <label>Last name</label>
    </div>
    <div class="user-box">

```

```

<input type="email" name="emailid" id="email" />
    <label> Email ID</label>
</div>

<div class="user-box">
<input type="password" name="password" id="pwd" />
    <label>Password</label>
</div>

<div class="user-box">
    <input type="password" name="cpassword" id="cpwd" />
    <label>Confirm Password</label>
</div>

<div class="user-box">
    <input type="number" name="number" id="number" />
    <label>Phone number</label>
</div>

<div class="user-box">
    <br/><br/>
    <select name="foodtype">
        <option>Veg</option>
        <option>Non-Veg</option>
    </select>
    <label> Food Type</label>
</div><br/><br/>

<div class="user-box">
    <input type="number" name="number1" id="number1" />
    <label>No.Of.Members</label>
</div>

```

```

        <input type="submit" id="register" value="register" />
    </form>
</div>
</body>
</html>

```

```

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```

```
/**
```

```
*
```

```
* @author ELCOT
```

```
*/
```

```
public class userservlet extends HttpServlet {
```

```
    /**
```

```
        * Processes requests for both HTTP <code>GET</code> and
        <code>POST</code>

```

```
        * methods.

```

```
        *

```

```
        * @param request servlet request

```

```
        * @param response servlet response

```

```
        * @throws ServletException if a servlet-specific error occurs

```

```
        * @throws IOException if an I/O error occurs

```

```
    */

```

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
```

```
throws ServletException, IOException {
```

```
response.setContentType("text/html;charset=UTF-8");
```

```
try (PrintWriter out = response.getWriter()) {
```

```
    /* TODO output your page here. You may use the following sample code. */
```

```
    out.println("<!DOCTYPE html>");
```

```
    out.println("<html>");
```

```
    out.println("<head>");
```

```
    out.println("<title>Servlet userservlet</title>");
```

```
    out.println("</head>");
```

```
    out.println("<body>");
```

```
    out.println("<h1>Servlet userservlet at " + request.getContextPath() + "</h1>");
```

```
    out.println("</body>");
```

```
    out.println("</html>");
```

```
}
```

```
}
```

```
// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +  
sign on the left to edit the code.">
```

```
/**
```

```
 * Handles the HTTP <code>GET</code> method.
```

```
 *
```

```
 * @param request servlet request
```

```
 * @param response servlet response
```

```
 * @throws ServletException if a servlet-specific error occurs
```

```
 * @throws IOException if an I/O error occurs
```

```
 */
```

```
@Override
```

```

protected void doGet(HttpServletRequest request, HttpServletResponse
response)

    throws ServletException, IOException {

    //processRequest(request, response);
response.setContentType("text/html;charset=UTF-8");

    PrintWriter out = response.getWriter();

    String Firstname=request.getParameter("Firstname");
    String lastname=request.getParameter("lastname");
    String emailid=request.getParameter("emailid");
    String number=request.getParameter("number");
    String foodtype=request.getParameter("foodtype");
    String number1=request.getParameter("number1");

    out.print("<html>\n" +
"  <head>\n" +
"    <title>register success</title>\n" +
"    <meta charset=\"UTF-8\">\n" +
"    <meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0\">\n"
+
"    <style>\n" +
" body{\n" +
"   background:black;\n" +
"     padding: 0;\n"+
"     margin: 0;\n"+
"     font-family:Helvetica;\n"+
"}\n" +
".log{\n" +
"  color:black; \n" +
"  font-size: 23px;\n" +

```

```

" display: block;\n" +
" text-align: center;\n" +
"}\n" +
".out{\n" +
" position:absolute;\n"+
"    background-color:lightgray;\n"+
"border-radius:3px;\n"+
"padding:70px 100px;\n"+
"    margin-left: 400px;\n"+
"    margin-top: 100px;\n"+
"    height: 400px;\n"+
"    width: 300px;\n"+
"}\n" +
".input-container{\n" +
" position:relative;\n" +
" margin-bottom:25px;\n" +
"}\n" +
".btn{\n" +
" text-align: center;\n" +
"    color: black;\n" +
"    padding: 10px;\n"+
"}    </style>\n" +
" </head>\n" +
" <body>\n" +
"    <div class=\"out\">\n" +
"        <form name=\"det\" id=\"det\" action=\"sign\" method=\"POST\">\n" +
" <div class=\"input-container\">\n" +

```

```

"    <h2> You are Successfully Registered!!.... </h2><BR>\n" +
"    First Name :<br> <input type=\"text\" name=\"fname\" value=\"+Firstname+\" />\" +
"    <br>Last Name :<br> <input type=\"text\" name=\"lname\" value=\"+lastname+\" />\" +
"    <br>Enail ID :<br> <input type=\"email\" name=\"email\" value=\"+emailid+\" />\" +
"    <br>Phone Number : <br> <input type=\"number\" name=\"number\" value=\"+number+\" />\" +
"    <br> Food Type :<br> <input type=\"text\" name=\"des\" value=\"+foodtype+\" />\" +
"    <br> No.Of.Members: <br> <input type=\"number\" name=\"number1\" value=\"+number1+\" />\" +
"    <br><br><a href=\"home.html\" id=\"register\"><b>*HOME*</b></a>\"+
"\n" +
" \n" +
"</form> \n" +
"    </div>\n" +
"</body>\n" +
"</html>");
}
}
<html>
<head>
<title></title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
body{
    padding: 0;

```

```
        margin: 0;
        background-image: url(about.jpg);
font-family:arial black;
}
h1{
        position:absolute;
        font-family: times new roman;
        top:20%;right:70%;
        font-size: 35px;
        font-style: italic;
        border-color: white;
        color:red;
}
h3{
        position:absolute;
        font-family: times new roman;
        top:30%;right:50%;
        font-size: 30px;
        font-style: italic;
}
h2{
        position:absolute;
        font-family: times new roman;
        top:1%;left:30%;
        font-size: 35px;
        font-style: italic;
        border-color: white;
```

```

        background-color:orange; }

    btn{

        text-align: center;

        color: green;

        bottom:90%;right:90%

    }

    label{

        position:absolute;

        font-family: times new roman;

        top:90%;left:30%;

        font-size: 35px;

        font-style: italic;

        border-color: white;

        background-color:blueviolet;

    }

</style>

</head>

<body>

<form name="Catering" action="Servletabo" method="GET">

    <h1>ABOUT US</h1>

    <h2>K.K CATERING SERVICE</h2>

    <h3>We are very glad to say that the name KK Catering Service was first introduced
    by us (KK four Brothers). Greetings from k.k Catering Service on the occasion of the
    marriage/auspicious function to be celebrated in your family. We offer the best
    services to celebrate according to your preference without compromising on the basic
    religious requirements. We have completed a decade of service in this event
    management and we have more than 6000 satisfied customers on our records.
    Quality and courtesy are our keywords.</h3>

    <input type="submit" value="HOME" id="btn" name="HOME" />

```

```

<label>MAKE YOUR OCCASION TASTEFUL</label>

    </form>

</body>

</html>

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
 *
 * @author ELCOT
 */
public class Servletabo extends HttpServlet {
    /**
     * Processes requests for both HTTP <code>GET</code> and
     <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)

```

```

        throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");

    try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use the following sample code. */
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet Servletabo</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet Servletabo at " + request.getContextPath() +
"</h1>");
        out.println("</body>");
        out.println("</html>");
    }

} // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">

/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response)

```

```

        throws ServletException, IOException {

        // processRequest(request, response);

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

RequestDispatcher dispatcher=request.getRequestDispatcher("/home.html");

dispatcher.forward(request,response);

    }

<html>
    <head>
        <title>payment</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <style>

body{
    padding: 0;
    margin: 0;
    background-image:url(1.jpg);
font-family:arial black;
}
h2{

    text-align: center;
    color: black;
    padding: 20px;

    }

.log{
color:black;
    font-size: 23px;
    display: block;
    text-align: center;
}
.out{

    position:absolute;
    background-color:appworkspace;
    border-radius:3px;
    padding:70px 100px;
margin-left: 400px;
margin-top: 100px;
height: 1000px;
width: 700px;

}
.input-container{

```

```

        width: 500px;
        overflow: hidden;
        margin: auto;
        margin: 20,0,0,450px;
        padding: 70px;
background:mediumaquamarine;
        button: center;
        border-radius: 4px;
        padding-left: 9px;
    }

.btn{

        text-align: center;
        margin-left: 50px;
color:black;
        padding: 10px;
    }
#bn{
    margin-left: 200px;
}
</style>
</head>
<body>
<div class="out">
<form name="det" id="det" action="Servletfi" method="GET">
    <div class="input-container">
<h2>PAYMENT METHOD</h2>
<br>
<table>
<tr>
<td>Payment:</td>
<td><select name="CARD TYPE" class="brderTrans" >

<option>Credit card</option>
<option>Debit card</option>

</select></td>
</tr>
</table>
Card.No:<br><input type="text" name="id" id="id" /><br>
Cvv.No:<br><input type="text" name="name" id="name"/><br>
<table>
<tr>
<td>Expiration Date:</td>
<td><select name="MONTH" class="brderTrans" >
    <option>Month</option>
<option>January</option>
<option>February</option>
<option>March</option>

```

```

<option>April</option>
<option>May</option>
<option>June</option>
<option>July</option>
<option>August</option>
<option>September</option>
<option>October</option>
<option>November</option>
<option>December</option>

</select></td>
</tr>
</table> <label>Year</label>
      <input type="date" name="date" id="eee" >
<table>
<tr>
<td>Food Type:</td>
<br>
<td><select name="FOOD TYPE" class="brderTrans" >
<option>Veg</option>
<option>Non-Veg</option>

</select></td>
</tr>
</table>

      No.of.Food:<br><input type="text" name="food" id="name"/><br><br>

      <input type="submit" name="Pay" id="bn" class="btn" value="Pay" />

```

```

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

```

```

</body>
</html>

```

Servlet Final:

```

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```

```

/**
 *
 * @author ELCOT

```

```

*/
public class Servletfi extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet Servletfi</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet Servletfi at " + request.getContextPath() + "</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to
    edit the code.">
    /**
     * Handles the HTTP GET method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // processRequest(request, response);
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        RequestDispatcher dispatcher=request.getRequestDispatcher("/suc.html");
        dispatcher.forward(request,response);
    }

}

```

Success Page(Html):

```

<html>
<head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
  body{
    background-color: blue;
  }
  .ee{
    height:282px;
    width:301px;
    margin-top: 100px;
    margin-left: 440px;
    background-color: red;
    background-image: url("suc.jpg");
  }
  h1{
    font-size: 40px;
    color: white;
    margin-left: 350px;
  }
</style>
</head>
<body>
  <div class="ee">

  </div>
  <h1> PAYMENT SUCCESSFULL...!<br>
  THANKS FOR YOUR ORDER...!</h1>
</body>
</html>

```

6.2 OUTPUT:

Sign Up page



SIGN UP

USER NAME

EMAIL ID

PASSWORD

CONFIRM PASSWORD

[GO TO NEXT PAGE](#)

Enter valid email id and correct password

- Login page



- Home page



- **Contact us page**



Click the home button. it's go to the home page

- **Menu page**





Click the book now button. it's go to the user details page

- *User details page*

First name

Last name

Email ID

Password

Confirm Password

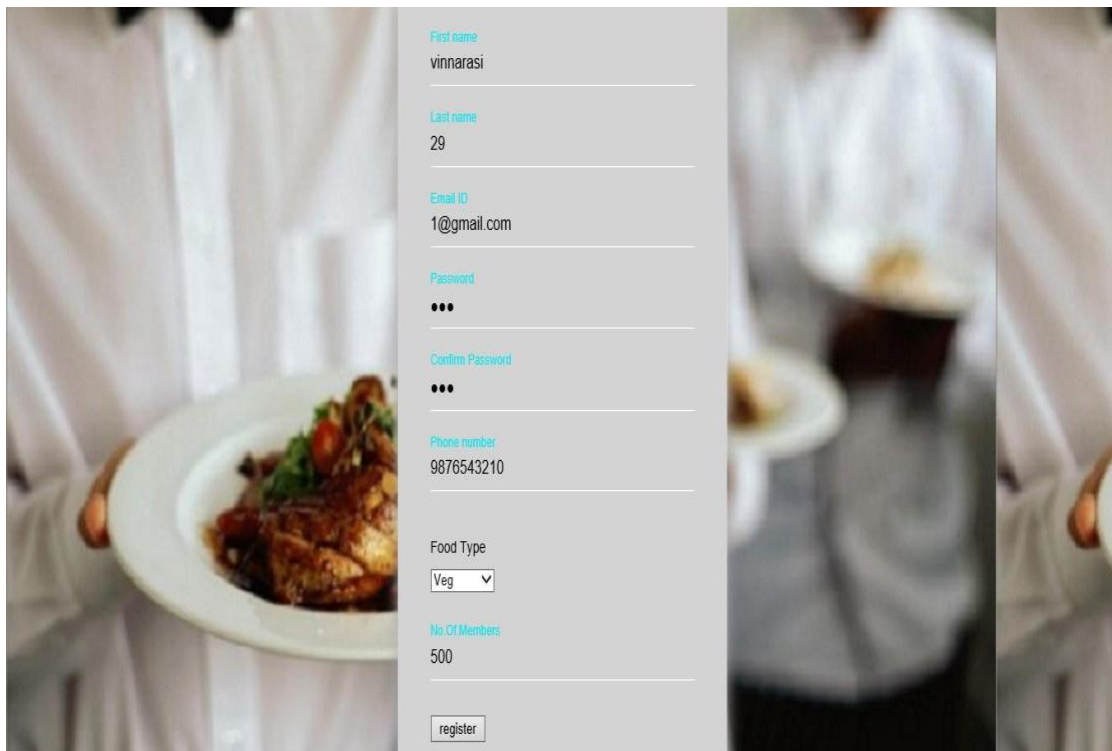
Phone number

Food Type
Veg

No. Of Members

register

- Enter all details



First name
vinnarasi

Last name
29

Email ID
1@gmail.com

Password
...

Confirm Password
...

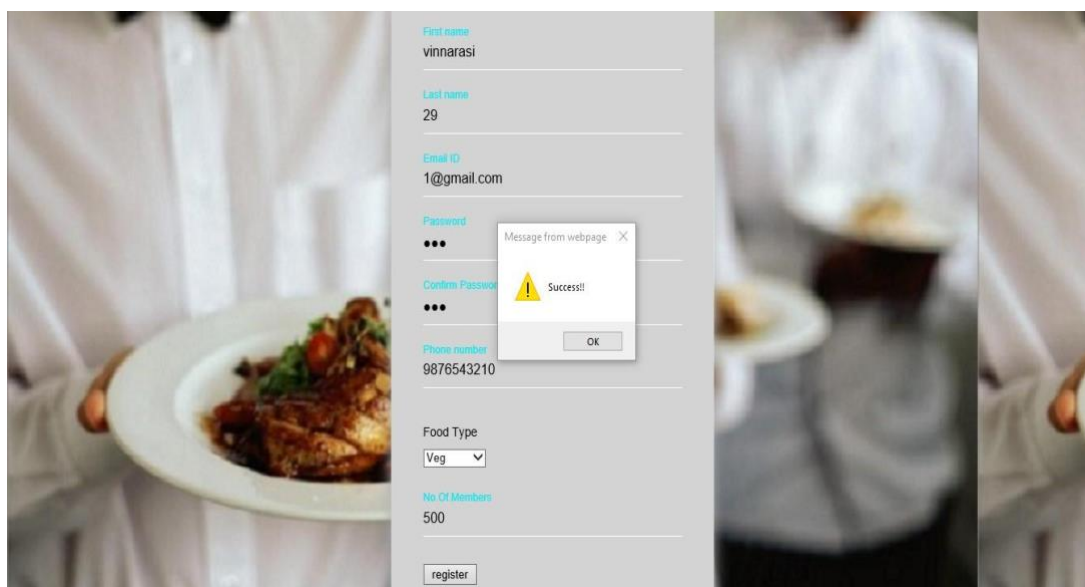
Phone number
9876543210

Food Type
Veg

No Of Members
500

register

- Message box



First name
vinnarasi

Last name
29

Email ID
1@gmail.com

Password
...

Confirm Password
...

Phone number
9876543210

Food Type
Veg

No Of Members
500

register

Message from webpage X
! Success!!
OK

- Confirm page

You are Successfully Registered!!....

First Name :
vinnarasi

Last Name :
29

Email ID :
1@gmail.com

Phone Number :
9876543210

Food Type :
Veg

No.Of.Members:
500

[*HOME*](#)

Click home link. it's go to the home page

- **PAYMENT METHOD**

PAYMENT METHOD

Payment: Credit card ▼

Card.No:
[Text Input]

Cvv.No:
[Text Input]

Expiration Date: Month ▼

Year [Text Input]

Food Type: Veg ▼

No.of.Food:
[Text Input]

- Payment page



**PAYMENT SUCCESSFULL...!
THANKS FOR YOUR ORDER...!**

- About page

[HOME](#)

K.K CATERING SERVICE

ABOUT US

We are very glad to say that the name KK Catering Service was first introduced by us (KK four Brothers). Greetings from k.k Catering Service on the occasion of the marriage/auspicious function to be celebrated in your family. We offer the best services to celebrate according to your preference without compromising on the basic religious requirements. We have completed a decade of service in this event management and we have more than 6000 satisfied customers on our records. Quality and courtesy are our keywords.

MAKE YOUR OCCASION TASTEFUL



Click the home button. it's go to the home page

