A PROJECT REPORT ON

AromaKart – An eCommerce Shop

Submitted By

Thanki Dwarkesh Harish

(Enrolment No. 03052101510)

In fulfilment for the award of the degree

Of

BACHELOR OF COMPUTER APPLICATION

Guided By

Mr. Thakrar Zalak

Shri V.J Modha College OF I.T-Porbandar

Bhakt Kavi Narsinh Mehta University, Junagadh

Academic Year

2021-2024

BKNMU

BHAKTA KAVI
NARSINH MEHTA
UNIVERSITY

Acknowledgement

During the preparation of the project, we have the good fortune of receiving support, in various ways, from several personal, numerous to mention here. We owe a debt of gratitude to all of them.

It is our privilege to express our sincerest regards to our project coordinator, Prof. ZALAK THAKRAR for their valuable inputs, able guidance, Encouragement, whole hearted cooperation and constructive criticism throughout the duration of our project.

It is our great pleasure to represent our project as one web application titled “Aromakart – An eCommerce Shop” and which we conceived in the 6th semester of BCA affiliated with BKNMU (Bhakt Kavi Narshinh Mehta University).

We are also thankful to the BKNMU (Bhakt Kavi Narshinh Mehta University) for including this project development subject in our syllabus. We got a golden opportunity to test and implement our creativity and programming skill simultaneously. Lastly, we would like to extend our sincere thanks to our advisors, classmates as well as all the books and websites who have directly or indirectly helped us

ABSTRACT

This Website Provide the facility to fully automate the process of food ordering for college students. This website will also provide the facility of digital menu to the students. Students can performs the following task :

- User Registration
- Login
- Can see the old forget password
- View Menu
- Make Order From that Menu
- View and edit cart
- Make payment
- Can See the order history

This website has one admin side also. Where the admin can manage the whole website. Following are some major tasks that website admin can perform :

- Admin Login
- Add Items To Menu
- View Or Edit The Menu
- Add Or Remove Category
- Can See The Orders History

Thus, in the above way, there are main two roles in this website users and admin. On this website, the users and admin have many features and facility

INDEX

Sr.No	Topic Name	Page No
Chapter - 1	**Introduction** 1.1 Project Summary 1.2 Purpose 1.3 Scope 1.4 Technology & Literature Review	1
Chapter – 2	**SYSTEM ANALYSIS** 2.1 Problem Definition 2.2 Process Model 2.3 Requirement Analysis 2.4 System Requirements and Specification	9
Chapter – 3	**SYSTEM DESIGN** 3.1 Data Flow Diagram 3.2 E-R Diagram 3.3 Use Case Diagram	18
Chapter – 4	**DATA DICTIONARY** 4.1 Data Dictionary	24
Chapter – 5	**LAYOUTS** 5.1 Admin Layout 5.2 Client Layout	36
Chapter – 6	**LIMITATIONS AND FUTURE ENHANCEMENT** 6.1 Limitations 6.2 Further Enhancement	85
Chapter – 7	**CONCLUSION** 7.1 Conclusion 7.2 Advantages	89
Chapter – 8	**BIBLIOGRAPHY & REFERENCES**	91

Chapter No: 1

Introduction

1.1 Project Summary

1.2 Purpose

1.3 Scope

1.4 Technology & Literature Review

1.1 Project Summary

Aromakart is a sophisticated eCommerce platform crafted using ASP.NET and C#, providing a seamless online shopping experience for customers seeking a diverse range of scents, perfumes, and organic oils. The platform is designed to streamline the order process, allowing clients to explore and purchase a variety of aromatic products with ease.

The user-friendly interface ensures a smooth navigation experience, while the robust backend, developed in C#, ensures secure transactions and efficient order management. Aromakart prioritizes customer satisfaction by offering a diverse selection of high-quality scents. The catalog includes perfumes, essential oils, and organic fragrances, providing clients with a broad spectrum of choices to cater to individual preferences.

The platform's innovative features include detailed product descriptions, user reviews, and a secure checkout process, enhancing the overall customer journey. With its focus on fragrances and user-centric design, Aromakart emerges as a premier destination for those seeking a personalized and convenient way to purchase aromatic products online.

1.2 Purpose

Aromakart, an innovative eCommerce shop built on the robust ASP.NET framework with C#, aims to redefine the olfactory shopping experience. In a world where scents evoke memories and emotions, Aromakart becomes a virtual haven for enthusiasts seeking a diverse range of premium perfumes, organic oils, and enticing fragrances. Our platform seamlessly combines cutting-edge technology with user-friendly design, offering clients a seamless journey from selection to checkout.

Dive into a curated collection of scents that cater to diverse preferences, allowing customers to discover and order their signature fragrances with ease. Aromakart prioritizes user satisfaction by providing a personalized shopping experience, ensuring that each client finds the perfect scent that resonates with their individual style and personality. With a secure and efficient ordering system, Aromakart transforms the act of purchasing scents into a delightful and memorable affair, all within the immersive digital realm. Elevate your senses and embrace the world of Aromakart, where fragrance meets technology for a unique and captivating eCommerce venture.

1.3 Scope

Aromakart, an innovative eCommerce platform built in ASP.NET with C#, revolutionizes the fragrance industry by offering a seamless shopping experience for clients seeking a diverse range of scents, perfumes, and organic oils. The user-friendly interface ensures easy navigation, while robust backend functionality streamlines order management. Aromakart prioritizes user satisfaction, providing a personalized approach to scent selection. Emphasizing quality and authenticity, the platform fosters trust through secure transactions. Elevating the online shopping experience, Aromakart employs cutting-edge technology to deliver aromatic delights to customers' doorsteps. Embrace the essence of luxury and convenience with Aromakart, where fragrances become an immersive journey.

1.4 Technology & Literature Review

❖ FRONT END:-

✓ ASP.Net C#

- ASP.NET is an open source, server-side web-application framework designed for web-development to produce dynamic web-pages. It was developed by Microsoft to allow programmers to build dynamic websites, applications and services. The name stands for Active Server Pages Network Enabled Technologies.

- It was first released in January 2002 with version 1.0 of the .NET Framework

- And In This We Are Using Below Technologies As Well :

✓ HTML

- HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between web pages. The markup language is used to define the text document within the tag which defines the structure of web pages.

- HTML Invent by the Tim Berners-Lee in 1993.

✓ CSS

- Cascading Style Sheets fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page. It describes how a webpage should look: it prescribes colors, fonts, spacing, and much more. In short, you can make your website look however you want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser

✓ JavaScript

- JavaScript is a lightweight, cross-platform, and interpreted compiled programming language which is also known as the scripting language for webpages.
- It is well-known for the development of web pages, and many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Serverside developments.
- JavaScript is both imperative and declarative type of language. JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.

✓ Bootstrap 5

- Bootstrap is the most popular CSS Framework for developing responsive and mobile-first websites.
- Bootstrap 5 is the newest version of Bootstrap.
- Bootstrap is an open-source front-end development framework for the creation of websites and web apps. Designed to enable responsive development of mobile-first websites, Bootstrap provides a collection of syntax for template designs.
- As a framework, Bootstrap includes the basics for responsive web development, so developers only need to insert the code into a pre-defined grid system.
- The Bootstrap framework is built on Hypertext Markup Language (HTML), cascading style sheets (CSS), and JavaScript.
- Web developers using Bootstrap can build websites much faster. Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools.

✓ FEATURES OF BOOTSTRAP

- ➢ Easy to Use
- ➢ Mobile-Friendly
- ➢ Customizable Bootstrap
- ➢ Simple Integration
- ➢ Pre-styled Components
- ➢ Responsive Features
- ➢ Browser Compatibility
- ➢ Great Grid System
- ➢ Extensive list of Components

❖ BACK-END:-

✓ SQL Server

- SQL Server is a type of database software that is used to store information for reporting, referencing and analysis.

- With SQL Server, you can analyze large amounts of data faster and more efficiently than with Excel or other types of spreadsheets.

  Access is most popular for its tables, forms and queries. The database tables are similar to spreadsheets, so you shouldn't have much trouble using the basic functions of the program.

- However, it does take time to learn the full features

Chapter No: 2

System Analysis

2.1 Problem Definition

2.2 Process Model

2.3 Requirement Analysis

2.4 System Requirement and Specification

2.1 Problem Definition

- The Aromakart is a web application for an shopping work. This web application will help the fragrance lovers to get an idea about the orders of that particular users.

- It will also help to avoid confusion between the orders as they will be placed digitally. We need an application like this to manage this vast, unorganized crowd. Users will be able to place the orders digitally. Customers will be happy after using this web application as it is very easy to use and also time saving.

- The most common problem of the manual system of canteen is that the efficiency of the management along with customer satisfaction could not be fulfilled.

- Customers have to wait in long queues, for placing orders and they have to also wait for their order to be ready on the counters.

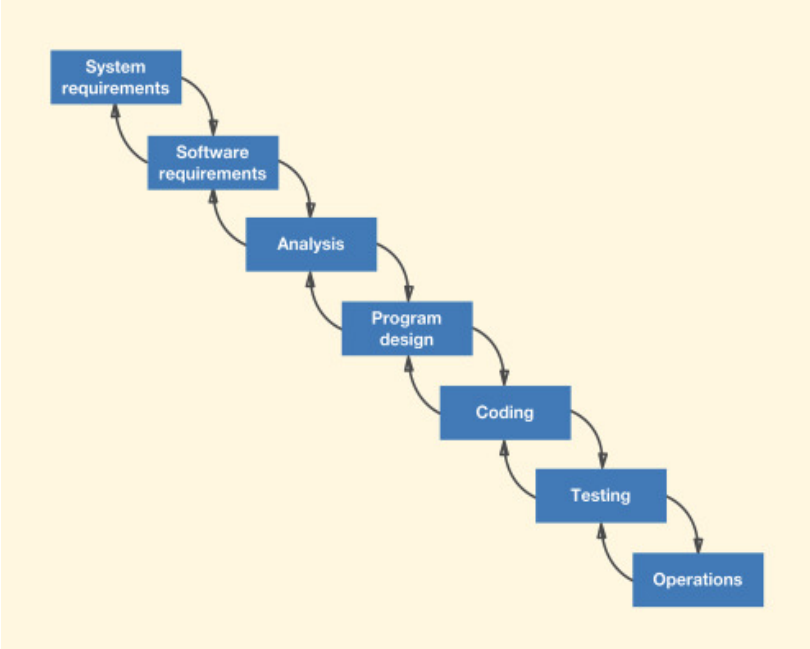
- So this website will overcome these problems.

2.2 Process Model

SDLC:-

The systems development life cycle (SDLC), also referred to as the application development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing and deploying an information system.

Iterative Waterfall Model :-



Advantages Of Iterative Waterfall Model :-

1. Simple and Easy to Understand and Each Phase has well Defined Input and Output.
2. It Works well for Project Where Requirement Are Clear And very well understood.
3. It Divides the complex task into more manageable works.

2.3 Requirement Analysis

✓ Hardware Requirement

- ➢ Operating System: 32 bits
- ➢ Ram: 2 GB

✓ Software Requirement

- ➢ Front End Tool :HTML, CSS, JavaScript & Bootstrap 5
- ➢ Back End Tool : C#, SQL Server
- ➢ Development Tool : Visual Studio 2022 Community

✓ Supported Operating Systems :

- ➢ Windows 7 (32-bit/64-bit)
- ➢ Windows 8 (32-bit/64-bit)
- ➢ Windows 10 (32-bit/64-bit)

✓ Supported Browsers :

- ➢ Microsoft Internet Explorer, Mozilla Firefox, Google Chrome

2.4 System Requirements and Specification.

✓ REQUIREMENT SPECIFICATION: USERS

- User Login & Register
- Edit Details after Login
- View Menu & Select item
- View Cart Or Manage Cart
- Make Payment Via Payment Page

✓ REQUIREMENT SPECIFICATION: ADMIN

- Admin Login
- Add New Products For Menu
- Customize Menu Or Remove Items
- Add or Remove Category
- See The Orders History
- Manage Users

REQUIREMENT SPECIFICATION: USERS

✓ User Login & Register

Description :

- On This page users are able to Login by User ID & Password. After registration users Able to Be verified and may proceed for the Login.
- Users are able to Login through Their Registered User ID and Password. If user Forgot their Password User can Click on The Forgot password Link and User Have to Fill their Details For a new A/C.

✓ View Menu & Select item

Description :

- On this page users can see the up to date menu with its item image, item name & item price.
- After that user can select item which they want to purchase and click on add to cart button. And item will added to their cart automatically.

✓ View Cart Or Manage Cart

Description :

- On this page all the items which is selected by the users will be shown.
- And also users have option to remove the items which they don't want to purchase.

✓ Make Payment Via Payment Page

Description :

- This page provides a payment option for users. Users are able to make payments using Debit Card or Cash On Delivery

REQUIREMENT SPECIFICATION: Admin

✓ Admin Login

Description :

- On this page the website admin can login to the website using their UserId and Password.

✓ Add New Items For Menu

Description :

- On this page the website admin can able to manage the menu.
- On this page admin can add items for the menu, by filling the item name, item price, item image, and item category.

✓ Customize Menu Or Remove Items

Description :

- On this page the website admin can manage the menu.
- After adding the items in menu, admin can remove those items when needed by using this page.

✓ Add or Remove Category

Description :

- By using thispage the website admin can easily add new category for the items
- And also admin can remove those added category when needed.

✓ See Order History and Status

Description :

- On this page the website admin can easily see the records of received orders.
- And this will display in users order history page.
- It will show Pending, Dispatched and Delivered Text from Database.

Chapter No: 3

System Design

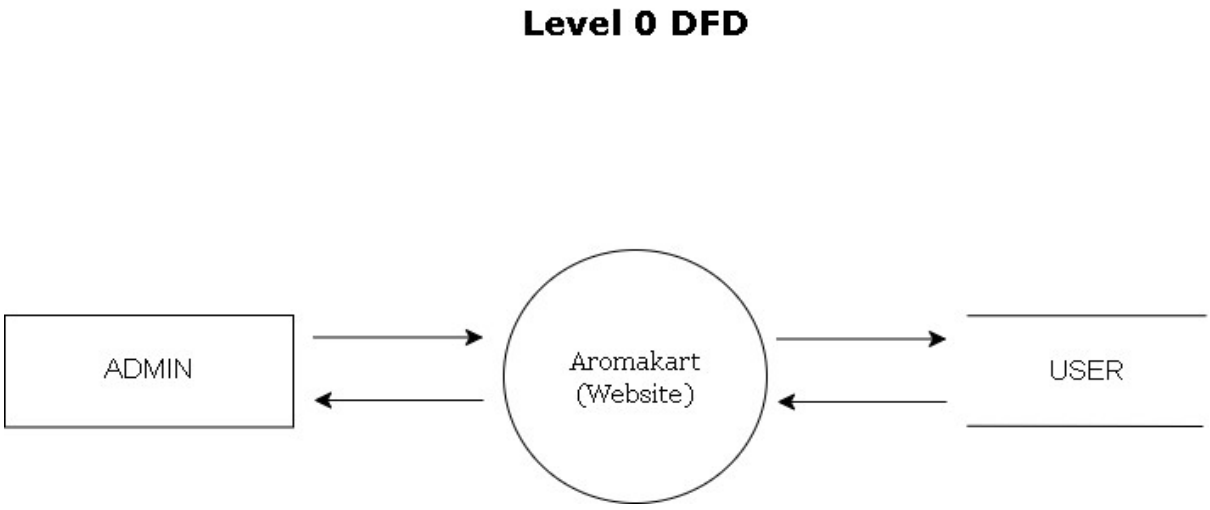
3.1 Data Flow Diagram

3.2 E-R Diagram

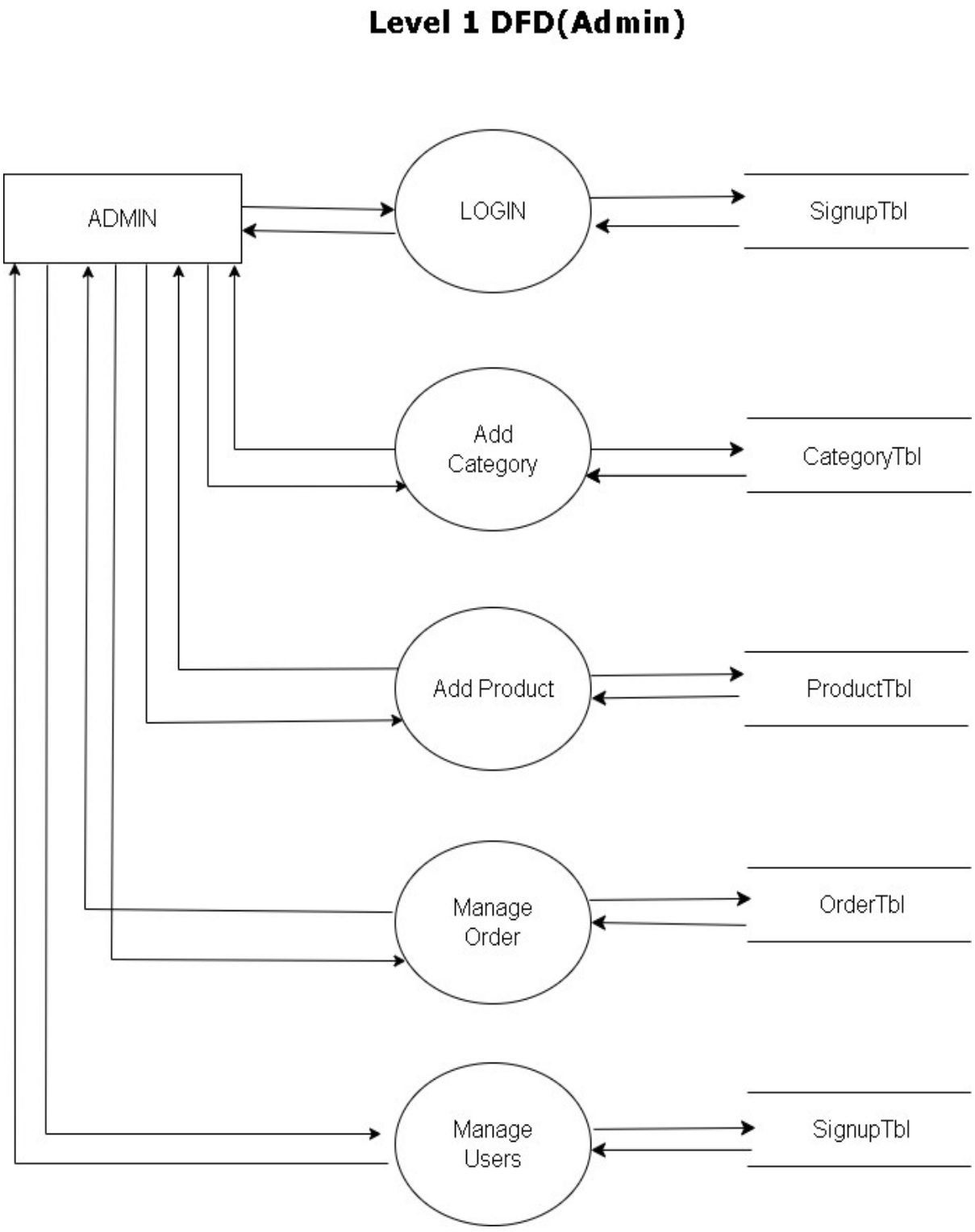
3.3 Use Case Diagram

3.1 Data Flow Diagram

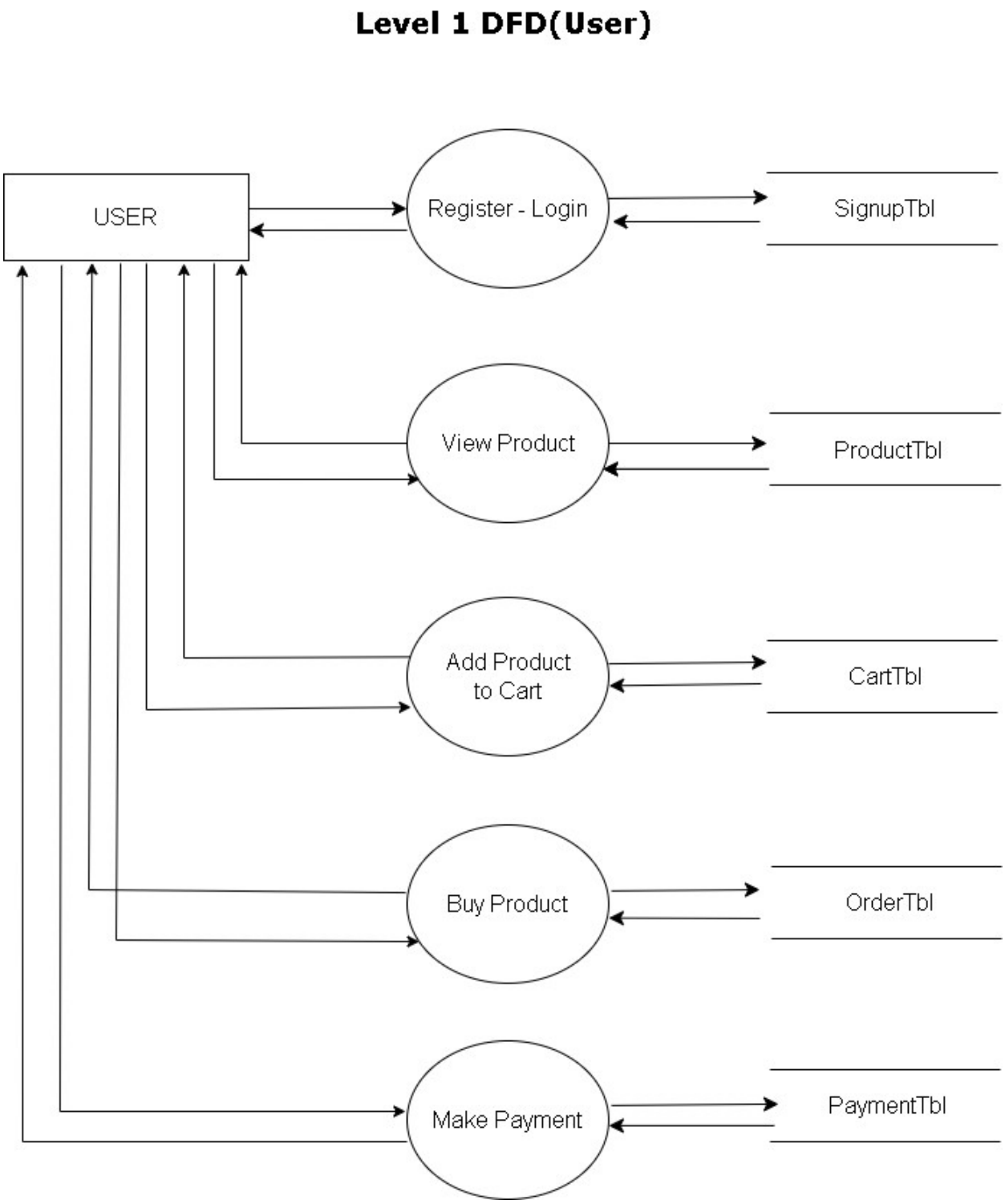
Level- 0DFD:-



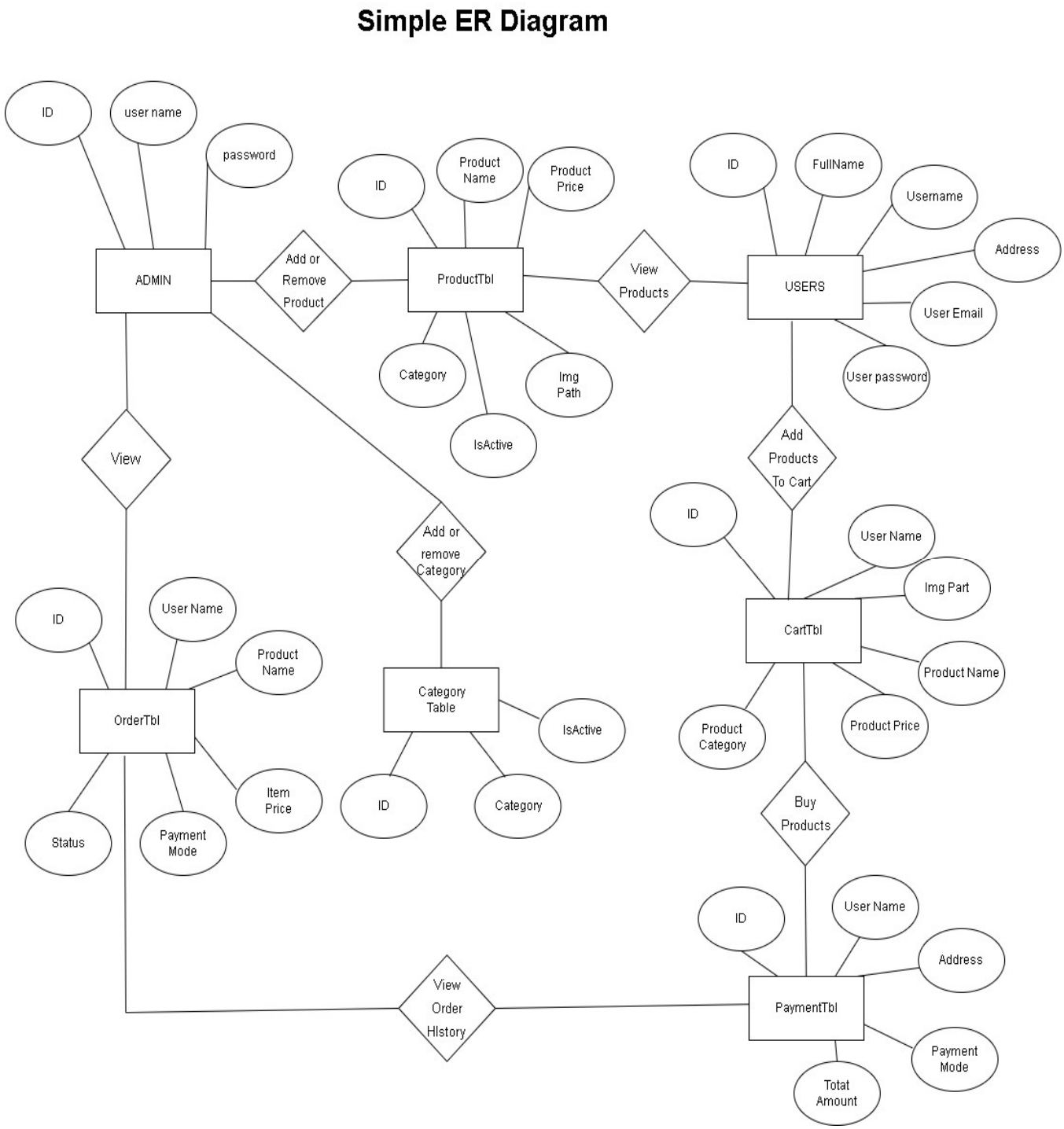
Level 1 DFD:-ADMIN



Level 1 DFD:-USER

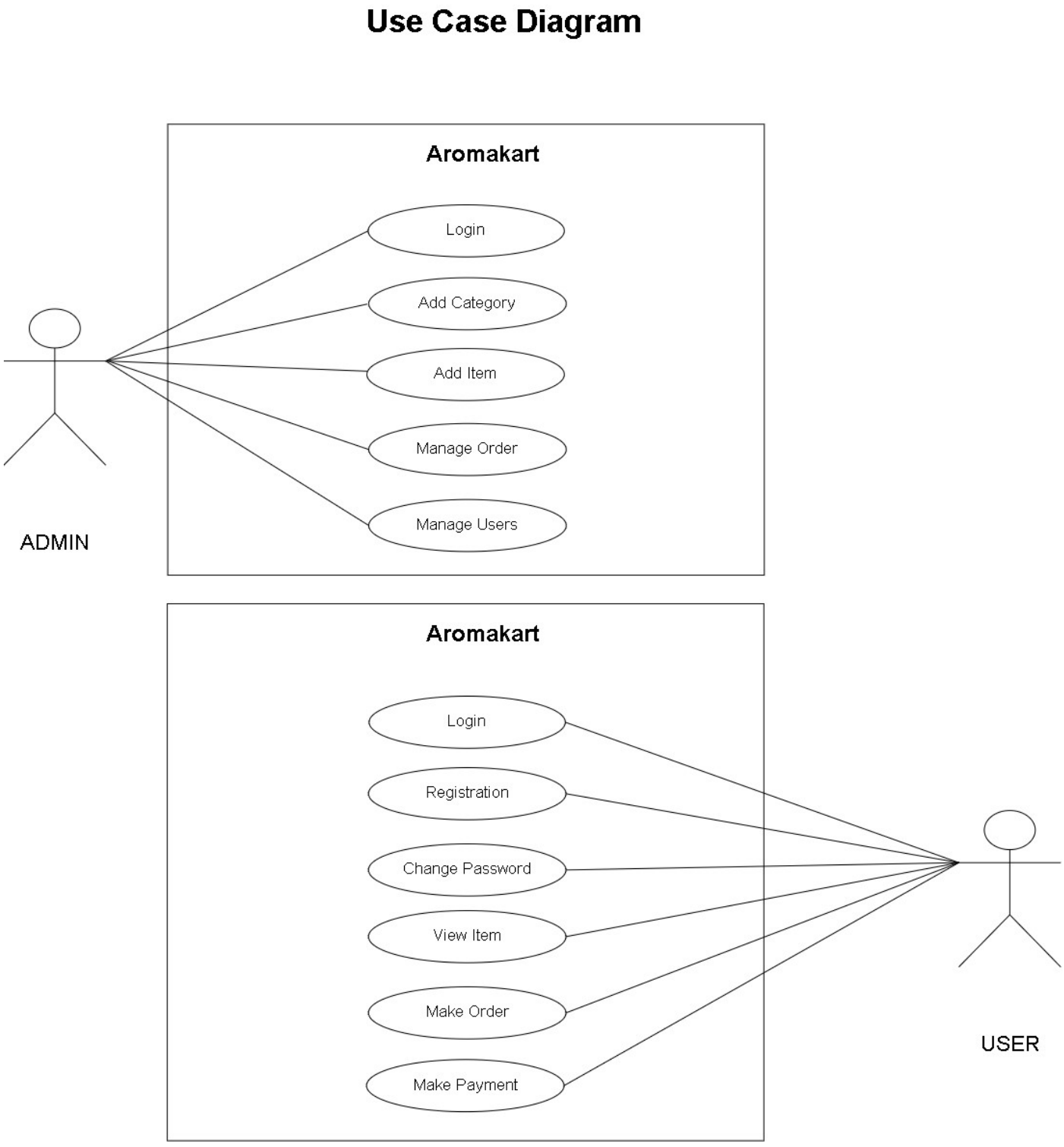


3.2 E-R Diagram



3.3 Use Case Diagram

USE CASE FOR ADMIN & USER



Chapter No: 4

Data Dictionary

4.1 Data Dictionary

Data Dictionary

- This Website Contains One Database.
- The Name of The Database Is “AromaDB”.
- This Database Contains the Following Tables.

1. USignupTbl
2. CategoryTbl
3. ProductTbl
4. CartTbl
5. PaymentTbl
6. OrderTbl

1. USignupTbl

Sr No	Fields Name	Data Types
1	Uid	Primary Key Identity(1,1)
2	Uname	VARCHAR(20)
3	Upassword	VARCHAR(20)
4	Uemail	VARCHAR(20)
5	Address	VARCHAR(MAX)
6	PostCode	INT
7	ImgUrl	VARCHAR(MAX)
8	JoinDate	DATETIME

Description **:**

- Uid
  - It is Auto Increment format, use to identify the rows number.

- Uname
  - The username field will store the admin/user information, and the data type is Text.

- Upassword
  - This Password field will used to store the admin/user password information, and the data type is Text.

- Uemail
  - This Email field will used to store the User Email, and the data type is Text.

- Address
- The Address field will store the address and the data type is Text.

- PostCode
- This PostCode fiels will used to store the city PIN, and the data type is Number.

- ImgUrl
- This Imgurl field will used to store the User Password, and the data type is Text.

- JoinDate
- This JoinDate field will used to store the Date of User joining the Account , and the data type is DATETIME.

2.CategoryTbl

Sr No	Fields Name	Data Types
1	CategoryId	Primary Key Identity(1,1)
2	CategoryName	VARCHAR(20)
3	CategoryImgUrl	VARCHAR(MAX)
4	IsActive	BIT
5	CreatedDate	JOINDATE

Description **:**

- CategoryId
- It is Number format, use to identify the rows number.

- CategoryName

- The Product Name field will store the Category Name , and the data type is Text.

- CategoryImgUrl
- This field will used to store the Category Image, and the data type is max Text.

- IsActive
- This field will used to category is active or not for the products, and the data type is BIT.

- CreatedDate
- This field will used to store the Category Date for the products, and the data type is DATETIME.

3.ProductTbl

Sr No	Fields Name	Data Types
1	ProductId	Primary Key Identity(1,1)
2	ProductName	VARCHAR(20)
3	ProductDesc	VARCHAR(MAX)
4	ProductQuantity	INT
5	ProductImgUrl	VARCHAR(MAX)
6	CategoryId	INT
7	IsActive	BIT
8	CreatedDate	DATETIME

Description **:**

- ProductId
- It is Number format, use to identify the rows number for product.

- ProductName
- The ProductName field will store the Product Name , and the data type is Text.

- ProductDesc
- This ProductDesc field will used Information for the products, and the data type is Text.

- ProductQuantity
- This ProductQuantity field will is used to store the Quantity for the cart products, and the data type is Text.

- ProductImgUrl
- This field is used to store the image for the cart products, and the data type is max text.

- CategoryId
- It is Number format, use to identify the rows number.

- IsActive
- It is BIT format, use to identify the Product active or not.

- CreatedDate
- It is DATETIME format, use to identify the date of product created.

4.Cart

Sr No	Fields Name	Data Types
1	CartId	Primary Key Identity(1,1)
2	ProductId	INT
3	Quantity	INT
4	Uid	INT

Description :

- CartId
- This field is used to Auto-Increment cart ID, and the data type is INT.

- ProductId
- It is Number format, use to identify the rows number.

- Quantity
- It is INT format, use to identify the Quantity of Product.

- Uid
- It is INT format, use to identify the user of product related.

5. PaymentTbl

Sr No	Fields Name	Data Types
1	PaymentId	Primary Key Identity(1,1)
2	Name	VARCHAR(20)
3	CardNo	INT
4	ExpiryDate	INT
5	CvvNo	INT
6	Address	VARCHAR(MAX)
7	PaymentMode	VARCHAR(20)

Description **:**

- PaymentId
- It is INT format, use to identify the rows number.

- Name
- The Payment holder name field will store in Name , and the data type is Text.

- CardNo
- This field INT will used to store the Card holders number, and the data type is INT.

- ExpiryDate
- This field INT will used identify the card expired or not.

- CvvNo

- This field CvvNo will used to store the CVV of Card and the data type is INT.

- Address

- This field Address will used to store the Address of Card-holder and the data type is Max Text.

- PaymentMode

- This field PaymentMode will used to store the holder choice of payment and the data type is Text.

6.OrdersTbl

Sr No	Fields Name	Data Types
1	OrderDetailsId	Primary Key Identity(1,1)
2	OrderNo	INT
3	ProductId	INT
4	Quantity	INT
5	Uid	INT
6	Status	VARCHAR(50)
7	PaymentId	INT
8	OrderDate	DATETIME

Description **:**

- OrderDetailsId
  - It is Auto Increment format, use to identify the rows number for an order.

- OrderNo
  - The OrderNo field will store the Unique Order number generated by system, and the data type is INT.

- ProductId
- It is Number format, use to identify the rows number for product.

- Quantity
- It is Number format, use to identify the quantity for product on how much user buy.

- Uid
  - It is User id will define which user has buy the product data type is INT.

- Status
  - The Status field will store the Status of Users product, and the data type is Text.

- PaymentId
- It is Number format, use to identify the payment number for product user paid.

- OrderDate
- It is DATETIME format, use to identify the date and time for product on which user has buy a product.

Chapter No: 5

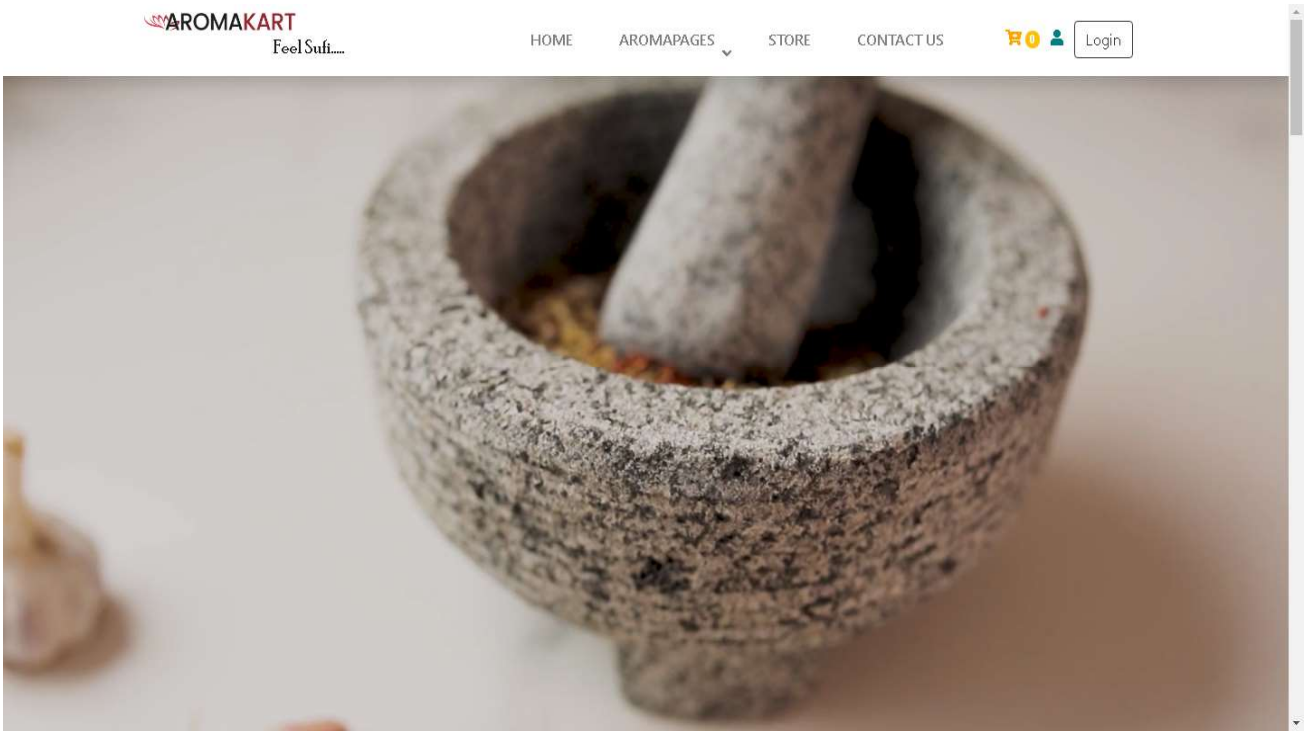
Layouts

5.1 User Layout

5.2 Admin Layout

5.1 User Layout

1. Homepage:-



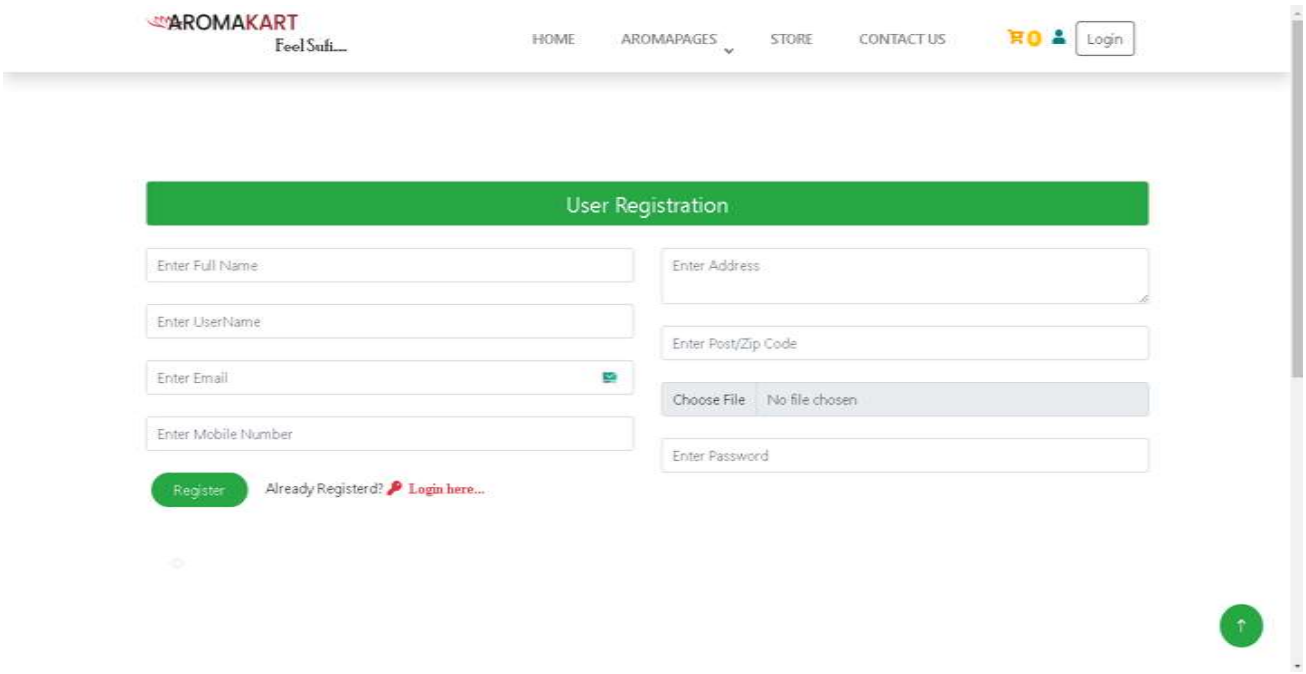
Description:-

- This is the home page of the website.
- This page has a one more Sections which are as follows.

Coding (.cs File):-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Aromakart.User
{
    public partial class User : System.Web.UI.MasterPage
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Session["userId"] == null)
            {

                lbLoginOrLogout.Text = "Login";
                Session["cartcount"] = "0";
            }
            else
            {
                lbLoginOrLogout.Text = "Logout";
                Utils utils = new Utils();
             Session["cartCount"]=
             utils.cartCount(Convert.ToInt32(Session["userId"])).ToString();

            }

        }
        protected void lbRegisterOrProfile_Click(object sender, EventArgs e)
        {
            if (Session["userId"] == null)
            {
                lbRegisterOrProfile.ToolTip = "User Registration";
                Response.Redirect("Signup.aspx");

            }
            else
            {
                lbRegisterOrProfile.ToolTip = "User profile";
                Response.Redirect("Profile.aspx");
            }
        }
        protected void lbLoginOrLogout_Click(object sender, EventArgs e)
        {
            if (Session["userId"] == null)
            {
                lbRegisterOrProfile.ToolTip = "Login.aspx";
                Response.Redirect("Login.aspx");
            }
            else
            {
                Session.Abandon();
                Response.Redirect("Login.aspx");
            }
        }}}
```

2.Register:-



Description:-

- This is the Register Section of the Home page of the website Where users can Register with us.

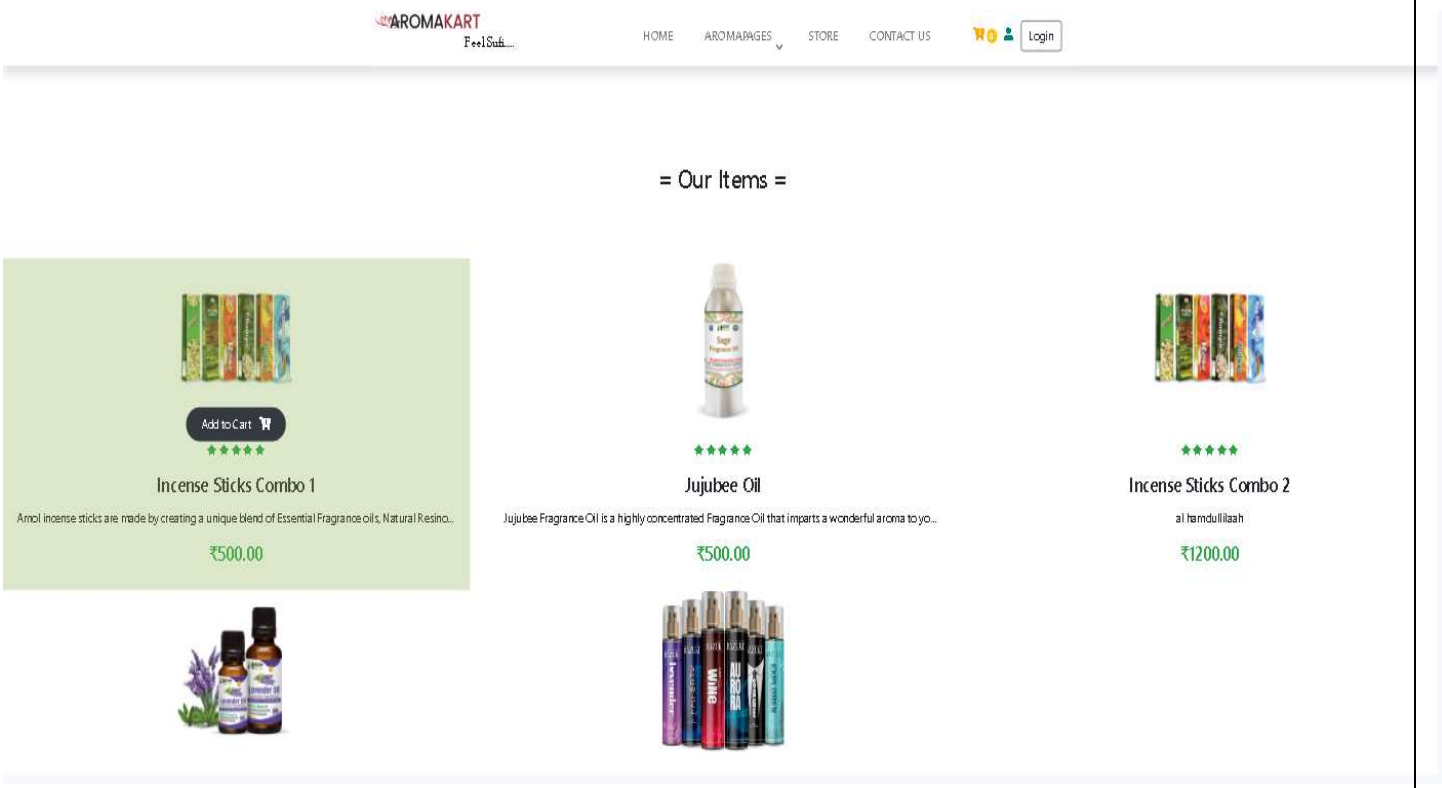
Coding:-

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;

namespace Aromakart.User
{
    public partial class WebForm4 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                if (Request.QueryString["id"] != null) /*&& Session["userId"] !=
null*/
                {
                    getUserDetails();
                }
                else if (Session["userId"]!=null)
                {
                    Response.Redirect("Default.aspx");
                }
            }
        }
        SqlConnection con;
        SqlCommand cmd;
        SqlDataAdapter sda;
        DataTable dt;
        protected void btnRegister_Click(object sender, EventArgs e)
        {
            string actionName = string.Empty, imgPath = string.Empty,
fileExtension = string.Empty;
            bool isValidToExecute = false;
            int userId = Convert.ToInt32(Request.QueryString["id"]);
            con = new SqlConnection(Utils.getConnection());
            cmd = new SqlCommand("User_Crud", con);
            cmd.Parameters.AddWithValue("@Action", userId == 0 ? "INSERT" :
"UPDATE");
            cmd.Parameters.AddWithValue("@UserId", userId);
            cmd.Parameters.AddWithValue("@Name", txtName.Text.Trim());
            cmd.Parameters.AddWithValue("@UserName", txtUsername.Text.Trim());
            cmd.Parameters.AddWithValue("@Mobile", txtMobile.Text.Trim());
            cmd.Parameters.AddWithValue("@Email", txtEmail.Text.Trim());
            cmd.Parameters.AddWithValue("@Address", txtAddress.Text.Trim());
            cmd.Parameters.AddWithValue("@PostCode", txtPostCode.Text.Trim());
            cmd.Parameters.AddWithValue("@Password", txtPassword.Text.Trim());
            if (fuUserImage.HasFile)
            {
                if (Utils.isValidExtension(fuUserImage.FileName))
```

```
                {
                    Guid obj = Guid.NewGuid();
                    fileExtension = Path.GetExtension(fuUserImage.FileName);
                    imgPath = "Images/User/" + fuUserImage.ToString() +
fileExtension;

fuUserImage.PostedFile.SaveAs(Server.MapPath("~/Images/User/") +
fuUserImage.ToString() + fileExtension);
                    cmd.Parameters.AddWithValue("@ImgUrl", imgPath);
                    isValidToExecute = true;
                }
                else
                {

                    lblMsg.Visible = false;
                    lblMsg.Text = "Please select .jpg, or .png image";
                    lblMsg.CssClass = "alert alert-danger";
                    isValidToExecute = false;
                }
            }
            else
            {
                isValidToExecute = true;
            }
            if (isValidToExecute)
            {
                cmd.CommandType = CommandType.StoredProcedure;
                try
                {
                    con.Open();
                    cmd.ExecuteNonQuery();
                    actionName = userId == 0 ?
                     "registration is successfull! <b><a href='Login.aspx'>Click
Here</a></b> to do login" : "details updated successfull! <b><a
href='Profile.aspx'>Can check here</a></b>";
                    lblMsg.Visible = true;
                    lblMsg.Text = "<b> " + txtUsername.Text.Trim() + " </b>" +
actionName;
                    lblMsg.CssClass = "alert alert-success";
                    if (userId != 0)
                    {
                        Response.AddHeader("REFRESH", "1;URL=Profile.aspx");
                    }

                    clear();
                }
                catch (SqlException ex)
                {
                    if (ex.Message.Contains("Violation of UNIQUE KEY
constraint"))
                    {
                        lblMsg.Visible = true;
                        lblMsg.Text = "<b> " + txtUsername.Text.Trim() + "</b>
username already exist, try another one..!";
                        lblMsg.CssClass = "alert alert-danger";
                    }
                }
                catch (Exception ex)
                {
                    lblMsg.Visible = true;
                    lblMsg.Text = "Error" + ex.Message;
                    lblMsg.CssClass = "alert alert-danger";
```

```
                }
                finally
                {
                    con.Close();
                }
            }
        }

        void getUserDetails()
        {
            con = new SqlConnection(Utils.getConnection());
            cmd = new SqlCommand("User_Crud", con);
            cmd.Parameters.AddWithValue("@Action", "SELECT4PROFILE");
            cmd.Parameters.AddWithValue("@UserId", Request.QueryString["id"]);
            cmd.CommandType = CommandType.StoredProcedure;
            sda = new SqlDataAdapter(cmd);
            dt = new DataTable();
            sda.Fill(dt);
            if (dt.Rows.Count == 1)
            {
                txtName.Text = dt.Rows[0]["Name"].ToString();
                txtUsername.Text = dt.Rows[0]["Uname"].ToString();
                txtMobile.Text = dt.Rows[0]["Umobile"].ToString();
                txtEmail.Text = dt.Rows[0]["Uemail"].ToString();
                txtAddress.Text = dt.Rows[0]["Address"].ToString();
                txtPostCode.Text = dt.Rows[0]["PostCode"].ToString();
                imgUser.ImageUrl =
string.IsNullOrEmpty(dt.Rows[0]["ImgUrl"].ToString()) ? "../Images/No_Image.png"
: "../" + dt.Rows[0]["ImgUrl"].ToString();
                imgUser.Height = 200;
                imgUser.Width = 200;
                txtPassword.TextMode = TextBoxMode.SingleLine;
                txtPassword.ReadOnly = true;
                txtPassword.Text = dt.Rows[0]["Upassword"].ToString();
            }
            lblHeaderMsg.Text = "<h2>Edit Profile</h2>";
            btnRegister.Text = "Update";
            lblAlreadyUser.Text = "";
        }
        private void clear() {
            txtName.Text = string.Empty;
            txtUsername.Text = string.Empty;
            txtMobile.Text = string.Empty;
            txtEmail.Text = string.Empty;
            txtAddress.Text = string.Empty;
            txtPostCode.Text = string.Empty;
            txtPassword.Text = string.Empty;

        }
    }
}
```

3.Store:-



Description:-

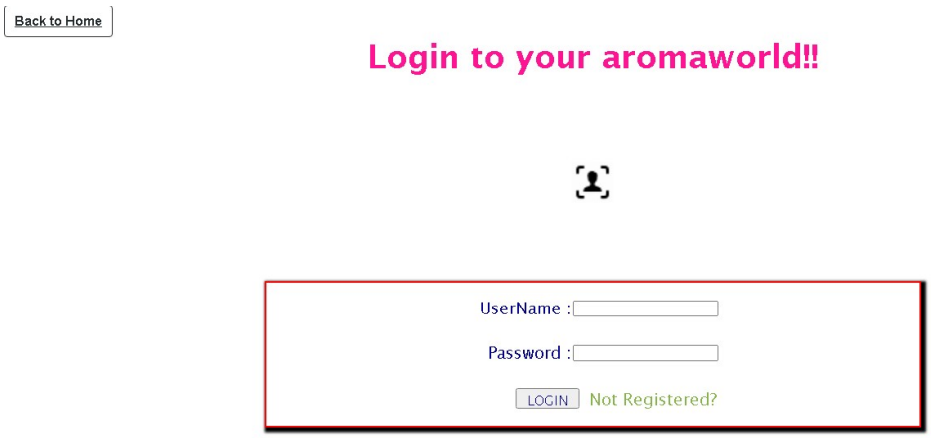
- This page is for new users who want to purchae for goods from our website.

Coding:-

```
using System;
using System.CodeDom;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Aromakart.Admin;

namespace Aromakart.User
{
    public partial class Allitem : System.Web.UI.Page
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataAdapter sda;
        DataTable dt;



        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
               getProducts();
            }
        }

  void getProducts()
  {
      con = new SqlConnection(Utils.getConnection());
      cmd = new SqlCommand("Product_Crud", con);
      cmd.Parameters.AddWithValue("@Action", "ACTIVEPROD");
      cmd.CommandType = CommandType.StoredProcedure;
      sda = new SqlDataAdapter(cmd);
      dt = new DataTable();
      sda.Fill(dt);
      rProducts.DataSource = dt;
      rProducts.DataBind();

  }
protected void rProducts_ItemCommand(object source, RepeaterCommandEventArgs e)
 {
     {
         if (Session["userId"] != null)
         {
             bool isCartItemUpdated = false;
             int i = isItemExistInCart(Convert.ToInt32(e.CommandArgument));
             if (i == 0)
             {
                 con = new SqlConnection(Utils.getConnection());
                 cmd = new SqlCommand("Cart_Crud", con);
                 cmd.Parameters.AddWithValue("@Action", "INSERT");
                 cmd.Parameters.AddWithValue("@ProductId", e.CommandArgument);
```

```
                cmd.Parameters.AddWithValue("@Quantity", 1);
                cmd.Parameters.AddWithValue("@UserId", Session["userId"]);
                cmd.CommandType = CommandType.StoredProcedure;

                try
                {
                    con.Open();
                    cmd.ExecuteNonQuery();

                }
                catch (Exception ex)
                {
                    Response.Write("<script>alert('Error - " + ex.Message + "
')</script>");
                }
                finally
                {
                    con.Close();
                }
            }
            else
            {
                Utils utils = new Utils();
                isCartItemUpdated = utils.updateCartQuantity(i + 1,
Convert.ToInt32(e.CommandArgument), Convert.ToInt32(Session["userId"]));

            }
            lblMsg.Visible = true;
            lblMsg.Text = "Item Added To cart Successfully";
            lblMsg.CssClass = "alert alert-success";
        }
        else
        {
            Response.Redirect("Login.aspx");
        }
    }
}
int isItemExistInCart(int productId)
{
    con = new SqlConnection(Utils.getConnection());
    cmd = new SqlCommand("Cart_Crud", con);
    cmd.Parameters.AddWithValue("@Action", "GETBYID");
    cmd.Parameters.AddWithValue("@ProductId", productId);
    cmd.Parameters.AddWithValue("@UserId", Session["userId"]);
    cmd.CommandType = CommandType.StoredProcedure;
    sda = new SqlDataAdapter(cmd);
    dt = new DataTable();
    sda.Fill(dt);
    int quantity = 0;
    if (dt.Rows.Count > 0)
    {
        quantity = Convert.ToInt32(dt.Rows[0]["Quantity"]);
    }
    return quantity;
}
```

4. Login:-



Description:-

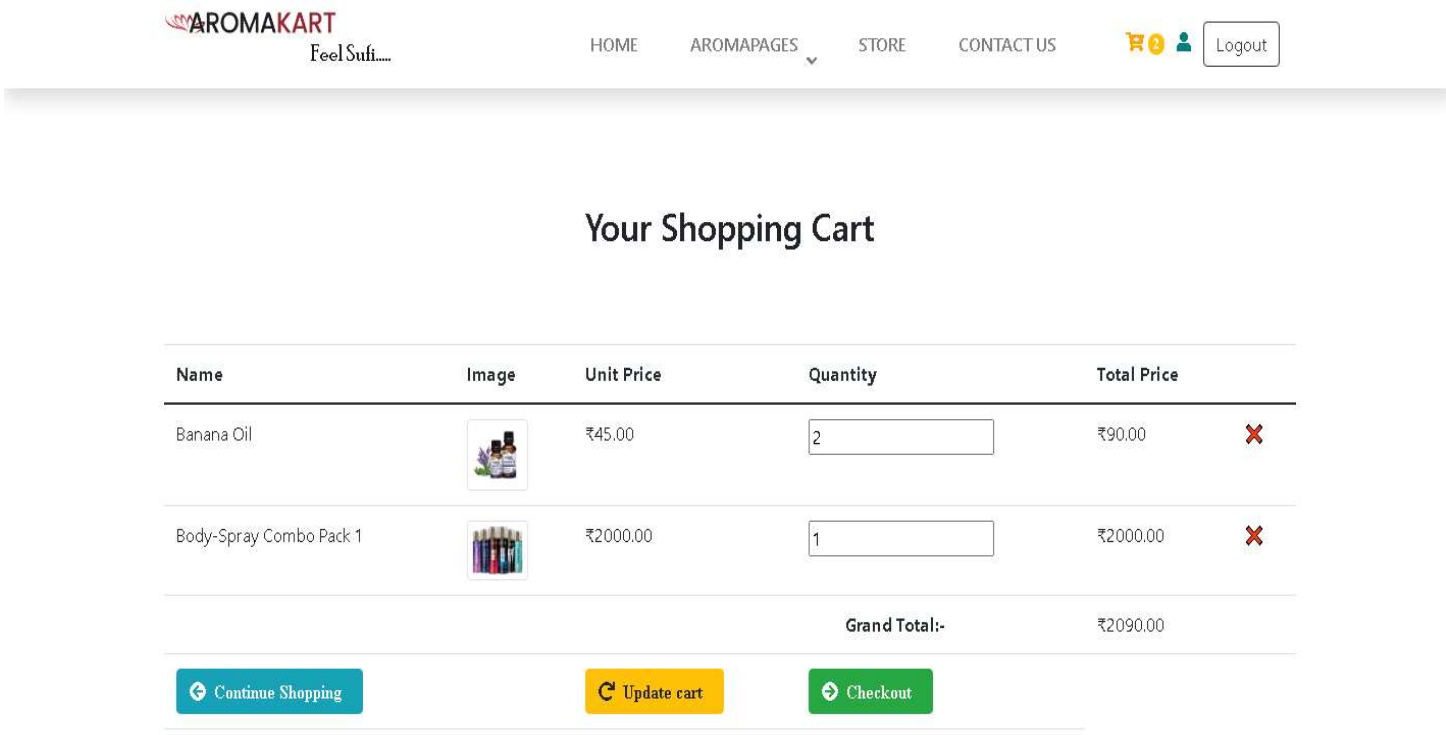
- This is Login page for the website users.
- Here /users need to their enter Username and password to Login.

Coding:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;
using System.Web.UI.WebControls.Expressions;
using System.Threading;
using System.Net;
using System.Security;

namespace Aromakart.User
{
    public partial class Login : System.Web.UI.Page
    {

        SqlConnection con;
        SqlCommand cmd;
        SqlDataAdapter sda;
        DataTable dt;

        protected void Page_Load(object sender, EventArgs e)
        {
            if (Session["userId"]!=null)
            {

                Response.Redirect("Default.aspx");

            }
        }

        protected void Button1_Click(object sender, EventArgs e)
        {

            if (txtUsername.Text.Trim()=="Admin" &&
txtPassword.Text.Trim()=="admin")
            {
                Session["admin"] = txtUsername.Text.Trim();
                Response.Redirect("../Admin/Dashboard.aspx");
            }
            else
            {
                con = new SqlConnection(Utils.getConnection());
                cmd = new SqlCommand("User_Crud", con);
                cmd.Parameters.AddWithValue("@Action", "SELECT4LOGIN");
                cmd.Parameters.AddWithValue("@Username",
txtUsername.Text.Trim());
                cmd.Parameters.AddWithValue("@Password",
txtPassword.Text.Trim());
                cmd.CommandType = CommandType.StoredProcedure;
```

```
                sda = new SqlDataAdapter(cmd);
                dt = new DataTable();
                sda.Fill(dt);
                if (dt.Rows.Count==1)
                {
                    Session["username"] = txtUsername.Text.Trim();
                    Session["userId"] = dt.Rows[0]["Uid"];
                    Response.Redirect("Default.aspx");

                }
                else
                {
                    lblMsg.Visible = true;
                    lblMsg.Text = "Invalid Credentials";
                    lblMsg.CssClass = "alert alert-danger";
                }
            }
        }
    }

}
```

5. OrderCart:-



Description:-

- This OrderCart page is used when the User Login and purchase goods.
- Here they can update and buy multiple goods and goes to checkout for payment.

Coding:-

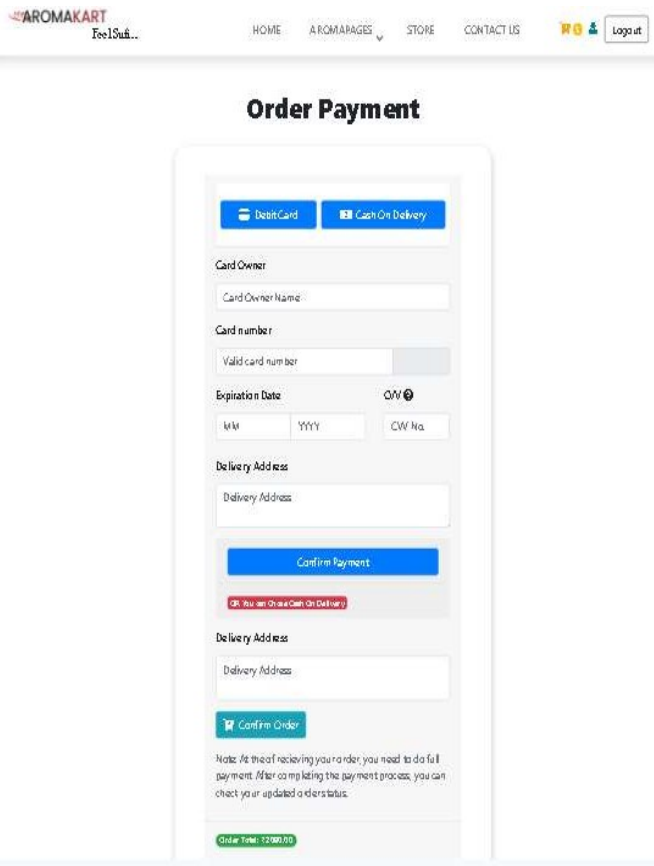
```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Aromakart.User
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataAdapter sda;
        DataTable dt;
        decimal grandTotal = 0;
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                if (Session["userId"] == null)
                {
                    Response.Redirect("Login.aspx");
                }
                else
                {

                    getCartItems();



                }
            }
        }

        void getCartItems()
        {
            con = new SqlConnection(Utils.getConnection());
            cmd = new SqlCommand("Cart_Crud", con);
            cmd.Parameters.AddWithValue("@Action", "SELECT");
            cmd.Parameters.AddWithValue("@UserId", Session["userId"]);
            cmd.CommandType = CommandType.StoredProcedure;
            sda = new SqlDataAdapter(cmd);
            dt = new DataTable();
            sda.Fill(dt);
            rCartItem.DataSource = dt;
            if (dt.Rows.Count == 0)
            {

                rCartItem.FooterTemplate = null;
                rCartItem.FooterTemplate = new
CustomTemplate(ListItemType.Footer);
            }
            rCartItem.DataBind();
        }
```

```
        protected void rCartItem_ItemCommand(object source,
RepeaterCommandEventArgs e)
        {
            Utils utils = new Utils();
            if (e.CommandName == "remove")
            {
                con = new SqlConnection(Utils.getConnection());
                cmd = new SqlCommand("Cart_Crud", con);
                cmd.Parameters.AddWithValue("@Action", "DELETE");
                cmd.Parameters.AddWithValue("@ProductId", e.CommandArgument);
                cmd.Parameters.AddWithValue("@UserId", Session["userId"]);
                cmd.CommandType = CommandType.StoredProcedure;
                try
                {
                    con.Open();
                    cmd.ExecuteNonQuery();
                    getCartItems();
                    Session["cartCount"] =
utils.cartCount(Convert.ToInt32(Session["userId"]));

                }
                catch (Exception ex)
                {
                    Response.Write("<script>alert('Error - " + ex.Message + "
')</script>");
                }
                finally
                {
                    con.Close();
                }
            }

            else if (e.CommandName == "updateCart")
            {
                bool isCartUpdated = false;
                for (int item=0; item < rCartItem.Items.Count; item++)
                {
                    if (rCartItem.Items[item].ItemType == ListItemType.Item ||
rCartItem.Items[item].ItemType == ListItemType.AlternatingItem)
                    {
                        TextBox quantity =
rCartItem.Items[item].FindControl("txtQuantity") as TextBox;
                        HiddenField _productid =
rCartItem.Items[item].FindControl("hdnProductId") as HiddenField;
                        HiddenField _quantity =
rCartItem.Items[item].FindControl("hdnQuantity") as HiddenField;
                        int quantityFromCart = Convert.ToInt32(quantity.Text);
                        int ProductId = Convert.ToInt32(_productid.Value);
                        int quantityFromDB = Convert.ToInt32(_quantity.Value);
                        bool isTrue = false;
                        int updatedQuantity = 1;
                        if (quantityFromCart > quantityFromDB)
                        {
                            updatedQuantity = quantityFromCart;
                            isTrue = true;
                        }
                        else if (quantityFromCart < quantityFromDB)
                        {
                            updatedQuantity = quantityFromCart;
                            isTrue = true;
                        }
                        if (isTrue)
```

```
                        {
                            //Update Cart Item in DB
                            isCartUpdated =
utils.updateCartQuantity(updatedQuantity, ProductId,
Convert.ToInt32(Session["userId"]));
                        }

                    }
                }
                getCartItems();
            }

            if (e.CommandName == "checkout")
            {
                bool isTrue = false;
                string pName = string.Empty;
                //First will check item quantity
                for (int item = 0; item < rCartItem.Items.Count; item++)
                {
                    if (rCartItem.Items[item].ItemType == ListItemType.Item ||
rCartItem.Items[item].ItemType == ListItemType.AlternatingItem)
                    {

                        HiddenField _productid =
rCartItem.Items[item].FindControl("hdnProductId") as HiddenField;
                        HiddenField _cartQuantity =
rCartItem.Items[item].FindControl("hdnQuantity") as HiddenField;
                        HiddenField _productQuantity =
rCartItem.Items[item].FindControl("hdnPrdQuantity") as HiddenField;
                        Label productName =
rCartItem.Items[item].FindControl("lblName") as Label;
                        int productId = Convert.ToInt32(_productid.Value);
                        int cartQuantity = Convert.ToInt32(_cartQuantity.Value);
                        int productQunatity =
Convert.ToInt32(_productQuantity.Value);

                        if (productQunatity > cartQuantity && productQunatity >
2)
                        {

                            isTrue = true;
                        }
                        else
                        {
                            isTrue = false;
                            pName = productName.Text.ToString();
                            break;
                        }

                    }
                }
                if (isTrue)
                {
                    Response.Redirect("Payment.aspx");
                }
                else
                {
                    lblMsg.Visible = true;
                    lblMsg.Text = "Item <b>'" + pName + "'</b> is out of
Stock:(";
                    lblMsg.CssClass = "alert alert-warning";
```

```
                }

            }
        }

        protected void rCartItem_ItemDataBound(object sender,
RepeaterItemEventArgs e)
        {
            if (e.Item.ItemType == ListItemType.Item || e.Item.ItemType ==
ListItemType.AlternatingItem)
            {
                Label totalPrice = e.Item.FindControl("lblTotalPrice") as Label;
                Label productPrice = e.Item.FindControl("lblPrice") as Label;
                TextBox quantity = e.Item.FindControl("txtQuantity") as TextBox;
                decimal callTotalPrice = Convert.ToDecimal(productPrice.Text) *
Convert.ToDecimal(quantity.Text);
                totalPrice.Text = callTotalPrice.ToString();
                grandTotal += callTotalPrice;
            }
            Session["grandTotalPrice"] = grandTotal;
        }
        private sealed class CustomTemplate : ITemplate
        {
            private ListItemType ListItemType { get; set; }

            public CustomTemplate(ListItemType type)
            {
                ListItemType = type;
            }

            public void InstantiateIn(Control container)
            {
                if (ListItemType == ListItemType.Footer)
                {
                    var footer = new LiteralControl("<table
class='table'><tbody><tr><td colspan='5'><b>Your cart Is Empty..</b><a
href='Store.aspx' class='btn btn-dark ml-2'>Continue
shopping</a></td></tr></tbody></table>");
                    container.Controls.Add(footer);
                }
            }
        }
    }
}
```

6.Payment-Gateway



Description:-

- Here is the payment page to our website successfully, this page is displayed when user tend to buy our product and goes for checkout.
- Here in the upper corner student can see the current login student username.
- It has Both system for transaction either COD or Debit Card.

Coding:-

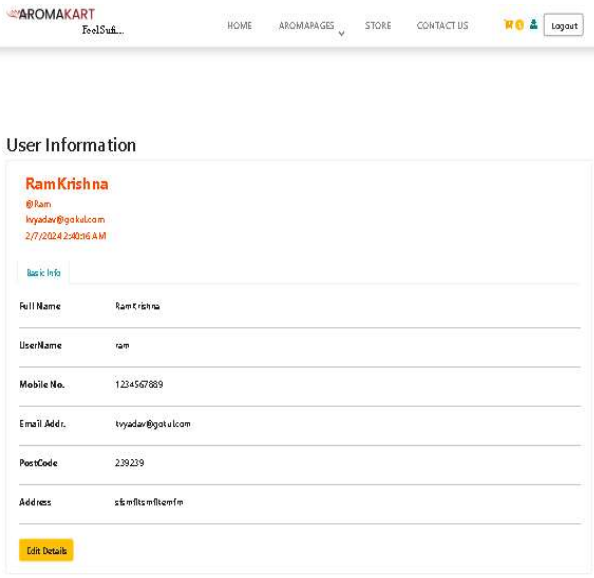
```
 using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.util;

namespace Aromakart.User
{
    public partial class Payment : System.Web.UI.Page
    {
        SqlConnection con = new SqlConnection(@"Data Source=desktop-
a8kf392\sqlexpress;Initial Catalog=AromaDB; Integrated Security=True");
        SqlCommand cmd;
        SqlDataAdapter sda;
        DataTable dt;
        SqlDataReader dr1, dr2;
        SqlTransaction transaction = null;
        string _name = string.Empty; string _cardNo = string.Empty; string
_expiryDate = string.Empty; string _cvv = string.Empty;
        string _address = string.Empty; string _paymentMode = string.Empty;

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                if (Session["userId"] == null)
                {
                    Response.Redirect("Login.aspx");
                }

            }
        }
        protected void lbCardSubmit_Click(object sender, EventArgs e)
        {
            _name = txtName.Text.Trim();
            _cardNo = txtCardNo.Text.Trim();
            _cardNo = string.Format("****************{0}",
txtCardNo.Text.Trim().Substring(12, 4));
            _expiryDate = txtExpMonth.Text.Trim() + "/" + txtExpYear.Text.Trim();
            _cvv = txtCvv.Text.Trim();
            _address = txtAddress.Text.Trim();
            _paymentMode = "card";
            if (Session["userId"] != null)
            {
                OrderPayment(_name, _cardNo, _expiryDate, _cvv, _address,
_paymentMode);
            }
            else
            {
                Response.Redirect("Login.aspx");
            }
        }
```

```
        protected void lbCodSubmit_Click(object sender, EventArgs e)
        {

            _address = txtCODAddress.Text.Trim();
            _paymentMode = "cod";
            if (Session["userId"] != null)
            {
                OrderPayment(_name, _cardNo, _expiryDate, _cvv, _address,
_paymentMode);
            }
            else
            {
                Response.Redirect("Login.aspx");
            }
        }
        void OrderPayment(string name, string cardNo, string expiryDate, string
cvv, string address, string paymentMode)
        {
            int paymentId; int productId; int quantity;
            dt = new DataTable();
            dt.Columns.AddRange(new DataColumn[7] {

                new DataColumn("OrderNo", typeof(string)),
                new DataColumn("ProductId", typeof(int)),
                new DataColumn("Quantity", typeof(int)),
                new DataColumn("UserId", typeof(int)),
                new DataColumn("Status", typeof(string)),
                new DataColumn("PaymentId", typeof(int)),
                new DataColumn("OrderDate", typeof(DateTime)),

            });


            con.Open();

            transaction = con.BeginTransaction();
            cmd = new SqlCommand("Save_Payment", con, transaction);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@Name", name);
            cmd.Parameters.AddWithValue("@CardNo", cardNo);
            cmd.Parameters.AddWithValue("@ExpiryDate", expiryDate);
            cmd.Parameters.AddWithValue("@Cvv", cvv);
            cmd.Parameters.AddWithValue("@Address", address);
            cmd.Parameters.AddWithValue("@PaymentMode", paymentMode);
            cmd.Parameters.Add("@InsertedId", SqlDbType.Int);
            cmd.Parameters["@InsertedId"].Direction = ParameterDirection.Output;
            try
            {
                cmd.ExecuteNonQuery();
                paymentId = Convert.ToInt32(cmd.Parameters["@InsertedId"].Value);


                cmd = new SqlCommand("Cart_Crud", con, transaction);
                cmd.Parameters.AddWithValue("@Action", "SELECT");
                cmd.Parameters.AddWithValue("@UserId", Session["userId"]);
                cmd.CommandType = CommandType.StoredProcedure;
                dr1 = cmd.ExecuteReader();
                while (dr1.Read())
                {
                    productId = (int)dr1["ProductId"];
                    quantity = (int)dr1["Quantity"];
                    //Update Product Quantity
```

```
                    UpdateQuantity(productId, quantity, transaction, con);
                    //Delete Cart Item
                    DeleteCartItem(productId, transaction, con);

                    dt.Rows.Add(Utils.GetUniqueId(), productId, quantity,
(int)Session["userId"], "Pending", paymentId, Convert.ToDateTime(DateTime.Now));
                }
               dr1.Close();
                if (dt.Rows.Count > 0)
                {
                    cmd = new SqlCommand("Save_Orders", con, transaction);
                    cmd.Parameters.AddWithValue("@tblOrders", dt);
                    cmd.CommandType = CommandType.StoredProcedure;
                    cmd.ExecuteNonQuery();
                }
                transaction.Commit();
                ClientScript.RegisterClientScriptBlock(this.GetType(), "d",
"swal(\"YOUR ITEM ORDER SUCCESSFULL!\", \"Payment Received\", \"success\");",
true);
                lblMsg.Visible = true;
                lblMsg.Text = "YOUR ITEM ORDER SUCCESSFUL!!";
                lblMsg.CssClass = "alert alert-success";
                Response.AddHeader("REFRESH", "3;URL=Invoice.aspx?id=" +
paymentId);
            }
            catch (Exception e)
            {
                try
                {
                    transaction.Rollback();
                }
                catch (Exception ex)
                {

                }
            }
            finally
            {
                con.Close();
            }
        }
        void UpdateQuantity(int _productId, int _quantity, SqlTransaction
sqlTransaction, SqlConnection sqlConnection)
        {

            int dbQuantity;
            cmd = new SqlCommand("Product_Crud", sqlConnection, sqlTransaction);
            cmd.Parameters.AddWithValue("@Action", "GETBYID");
            cmd.Parameters.AddWithValue("@ProductId", _productId);
            cmd.CommandType = CommandType.StoredProcedure;
            try
            {
                dr2 = cmd.ExecuteReader();
                while (dr2.Read())
                {
                    dbQuantity = (int)dr2["Quantity"];

                    if (dbQuantity > _quantity && dbQuantity > 2)
                    {
                        dbQuantity = dbQuantity - _quantity;
                        cmd = new SqlCommand("Product_Crud", sqlConnection,
sqlTransaction);
```

```
                        cmd.Parameters.AddWithValue("@Action", "QTYUPDATE");
                        cmd.Parameters.AddWithValue("@ProductQuantity",
dbQuantity);
                        cmd.Parameters.AddWithValue("@ProductId", _productId);
                        cmd.CommandType = CommandType.StoredProcedure;
                        cmd.ExecuteNonQuery();
                    }
                }
                dr2.Close();

            }
            catch (Exception ex)
            {

            }

        }

        Utils utils = new Utils();
        void DeleteCartItem(int _productId, SqlTransaction sqlTransaction,
SqlConnection sqlConnection)
        {
            cmd = new SqlCommand("Cart_Crud", sqlConnection, sqlTransaction);
            cmd.Parameters.AddWithValue("@Action", "DELETE");
            cmd.Parameters.AddWithValue("@ProductId", _productId);
            cmd.Parameters.AddWithValue("@UserId", Session["userId"]);
            cmd.CommandType = CommandType.StoredProcedure;

            try
            {
                cmd.ExecuteNonQuery();
            }
            catch (Exception ex)
            {

            }
        }
    }
}
```

7.Profile(User):-



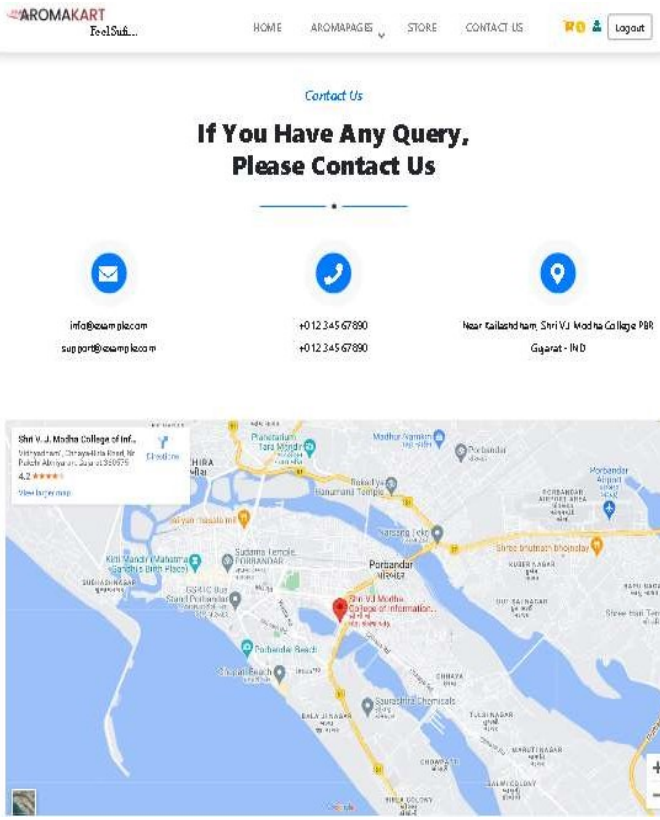
Description:-

- This is a User Profile page for User side.
- Here User can edit the details which don't want and by clicking on Edit Details button, the details will goes to the edit Profile page through session.

Coding:-

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Xml.Linq;

namespace Aromakart.User
{
    public partial class Profile : System.Web.UI.Page
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataAdapter sda;
        DataTable dt;
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                if (Session["userId"] == null)
                {
                    Response.Redirect("Login.aspx");
                }
                else
                {
                    getUserDetails();
                }
            }
        }
        void getUserDetails()
        {
            con = new SqlConnection(Utils.getConnection());
            cmd = new SqlCommand("User_Crud", con);
            cmd.Parameters.AddWithValue("@Action", "SELECT4PROFILE");
            cmd.Parameters.AddWithValue("@UserId", Session["userId"]);
            cmd.CommandType = CommandType.StoredProcedure;
            sda = new SqlDataAdapter(cmd);
            dt = new DataTable();
            sda.Fill(dt);
            rUserProfile.DataSource = dt;
            rUserProfile.DataBind();
            if (dt.Rows.Count == 1)
            {

                Session["name"] = dt.Rows[0]["Name"].ToString();
                Session["email"] = dt.Rows[0]["Uemail"].ToString();
                Session["imgurl"] = dt.Rows[0]["ImgUrl"].ToString();
                Session["createdDate"] = dt.Rows[0]["JoinDate"].ToString();

            }}}}
```

8.Contact:-



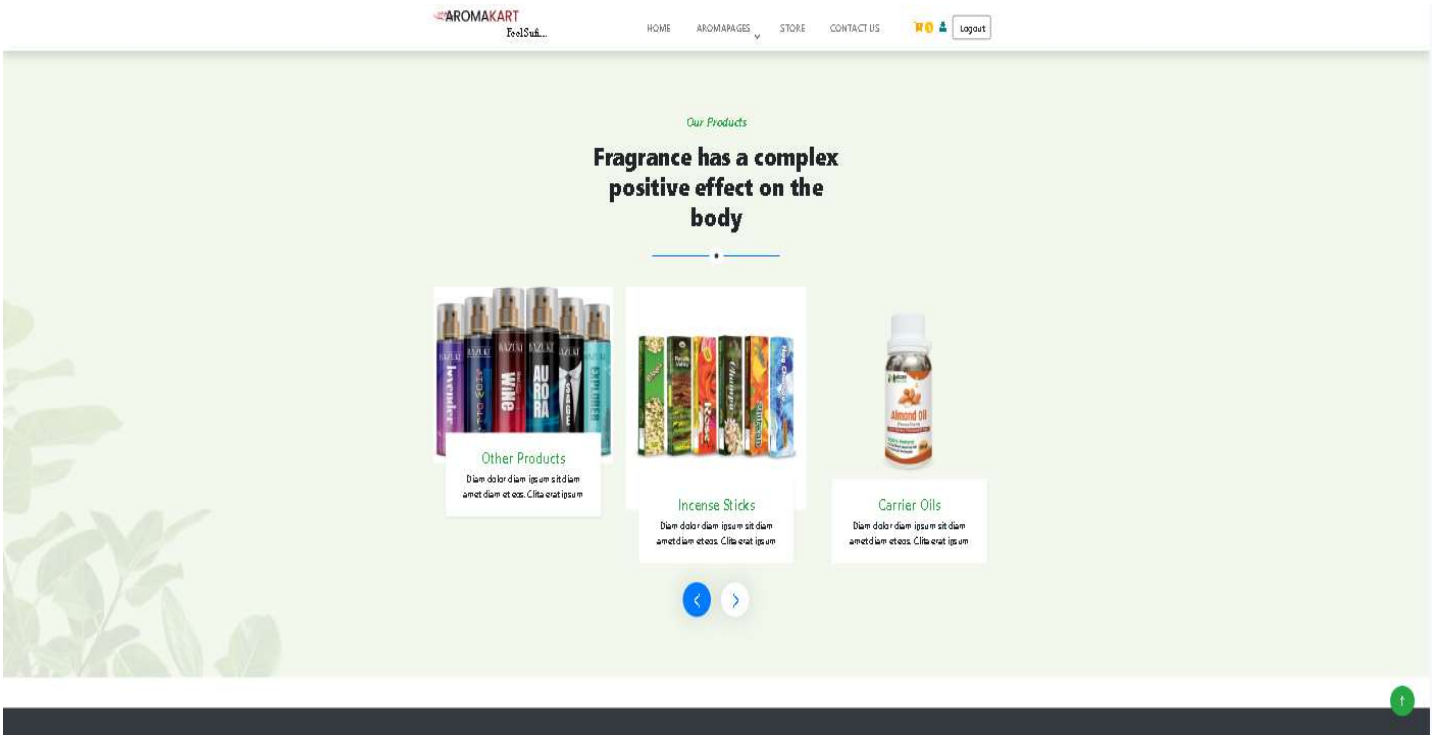
Description:-

- This is a Contact page.
- Here User can reach us by different details provided on Site.

Coding:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Aromakart.User
{
    public partial class Contact : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
      }
```

9.Product Display Slider:-



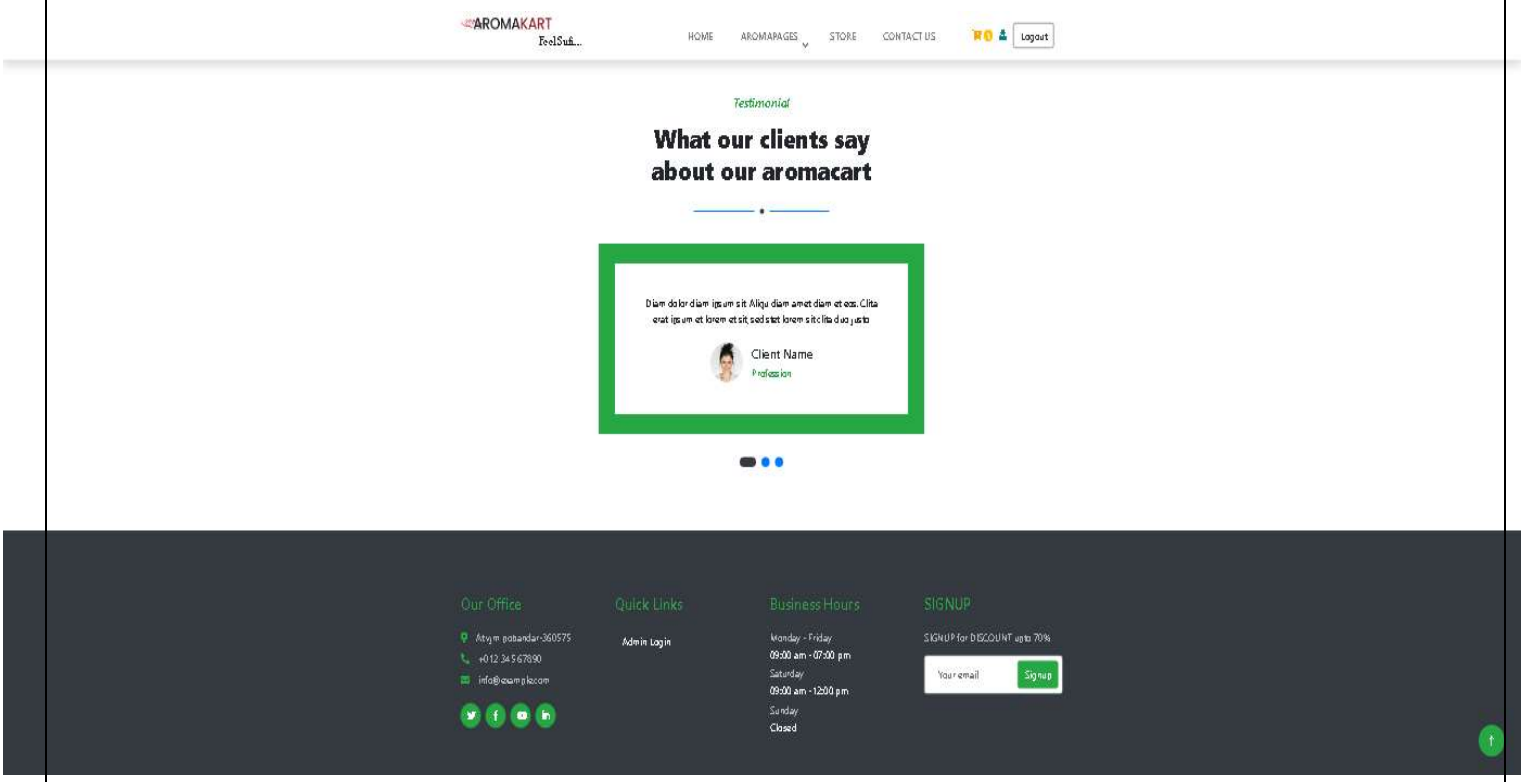
Description:-

- This is a product display slider page for the attraction.
- Onclick on items page lands to store page for purchase.

Coding:-

```
ing System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace Aromakart.User
{
    public partial class WebForm5 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
```

10.Testimonial Page:-



Description:-

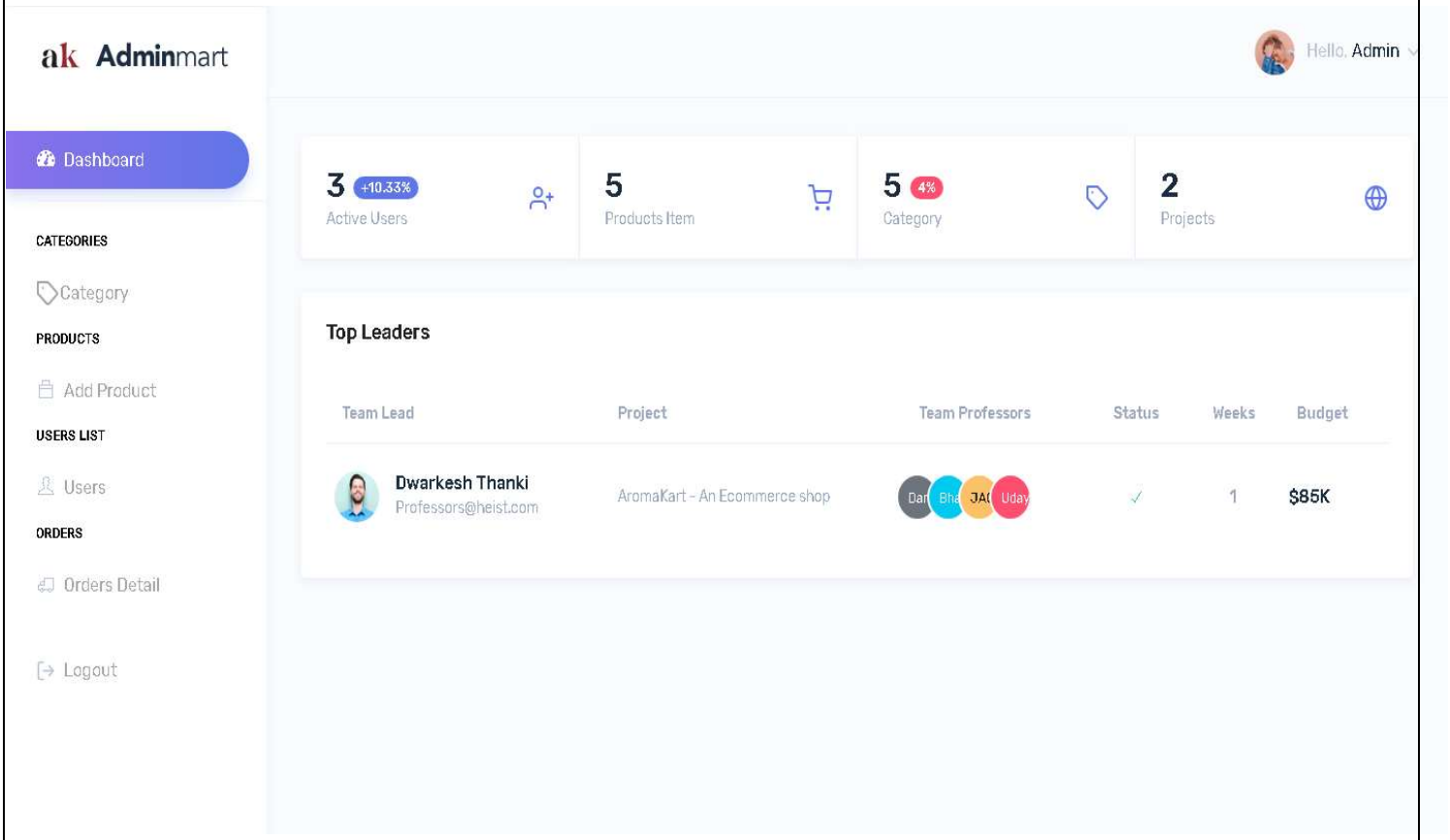
- This is a Testimonial page, where our Client Honest feedback received and that are featured here.

Coding:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Aromakart.User
{
    public partial class Testimonial : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

5.2 Admin Layout

1. Admin Entry Page(After Login):-



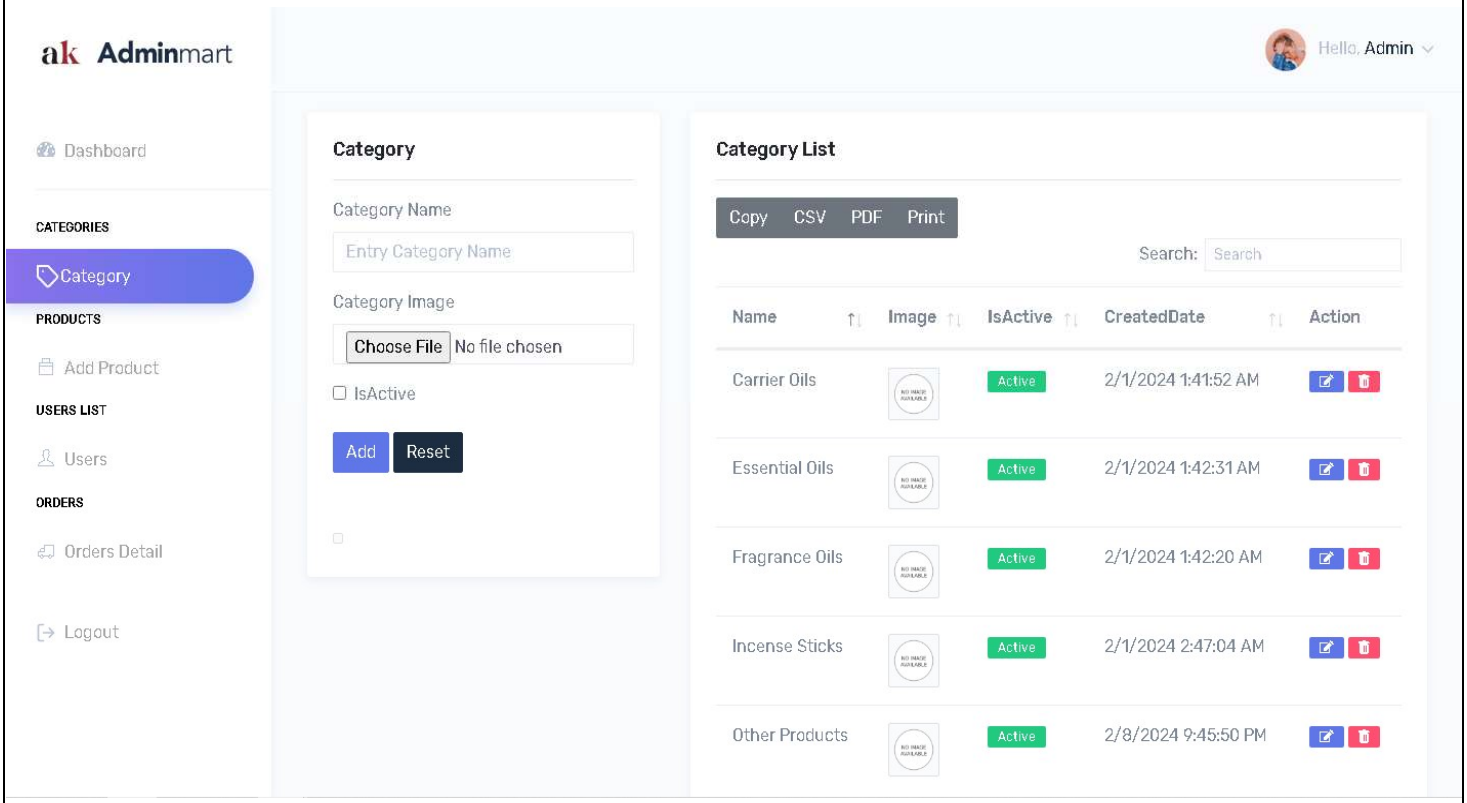
Description:-

- This is a Dashboard for the website Admin.
- By using this page the website admin can easily know the roadmap for business.

Coding:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Aromakart.User
{
    public partial class Testimonial : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

2. Adding Category:-



Description:-

- This Page is used to add Category to the Product.
- Here Admin have to enter the below information to add new Category in the Menu.
- Category Name
- Category Image is Optional
- Is Category Active or Not?
- And Select Add you can also perform edit and delete here.

Coding:-

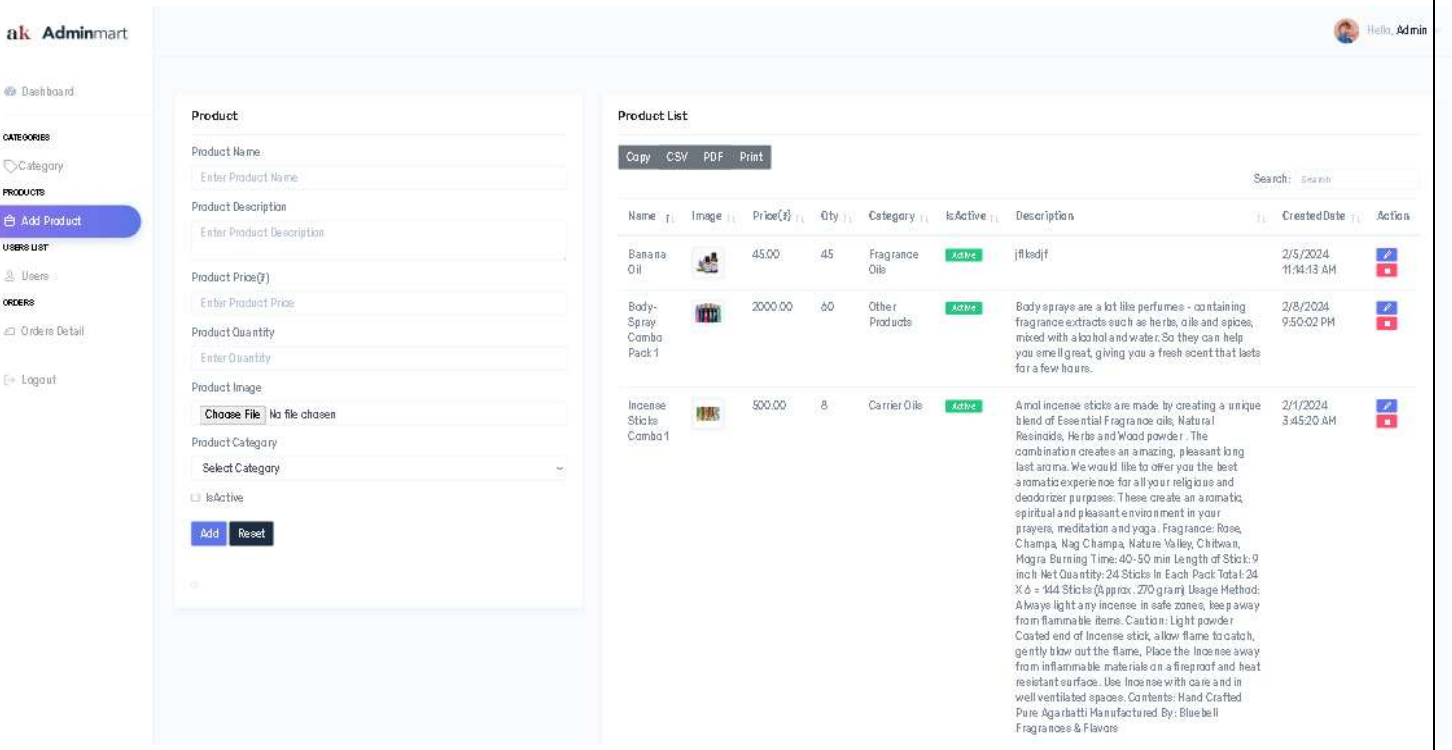
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.IO;

namespace Aromakart.Admin
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataAdapter sda;
        DataTable dt;
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack) {
                Session["breadCumbTitle"] = "Manage Category";
                Session["breadCumbPage"] = "Category";
                lblMsg.Visible = false;
                getCategories();
            }
            lblMsg.Visible = false;
        }
        void getCategories()
        {
            con = new SqlConnection(Utils.getConnection());
            cmd = new SqlCommand("Category_Crud", con);
            cmd.Parameters.AddWithValue("@Action","GETALL");
            cmd.CommandType = CommandType.StoredProcedure;
            sda = new SqlDataAdapter(cmd);
            dt = new DataTable();
            sda.Fill(dt);
            rCategory.DataSource = dt;
            rCategory.DataBind();

        }


        protected void btnAddOrUpdate_Click(object sender, EventArgs e)
        {
            string actionName = string.Empty, imgPath = string.Empty,
fileExtension = string.Empty;
            bool isValidToExecute = false;
            int categoryId = Convert.ToInt32(hfCategoryId.Value);
            con = new SqlConnection(Utils.getConnection());
            cmd = new SqlCommand("Category_Crud", con);
            cmd.Parameters.AddWithValue ("@Action", categoryId == 0 ? "INSERT" :
"UPDATE");
            cmd.Parameters.AddWithValue("@CategoryId", categoryId);
```

```
        cmd.Parameters.AddWithValue("@CategoryName",
txtCategoryName.Text.Trim());

            cmd.Parameters.AddWithValue("@IsActive", cbIsActive.Checked);
            if (fuCategoryImage.HasFile)
            {
                if (Utils.isValidExtension(fuCategoryImage.FileName))
                {
                    string newImageName = Utils.getUniqueId();
                    fileExtension = Path.GetExtension(fuCategoryImage.FileName);
                    imgPath = "Images/Category/" + newImageName.ToString() +
fileExtension;

fuCategoryImage.PostedFile.SaveAs(Server.MapPath("~/Images/Category/") +
newImageName.ToString() + fileExtension);
                    cmd.Parameters.AddWithValue("@CategoryImgUrl",imgPath);
                    isValidToExecute = true;
                }
                else
                {
                    lblMsg.Visible = false;
                    lblMsg.Text = "Please select .jpg, or .png image";
                    lblMsg.CssClass = "alert alert-danger";
                    isValidToExecute = false;
                }
            }
            else
            {
                isValidToExecute = true;
            }
            if (isValidToExecute)
            {
                cmd.CommandType = CommandType.StoredProcedure;
                try
                {
                    con.Open();
                    cmd.ExecuteNonQuery();
                    actionName = categoryId == 0 ? "inserted" : "updated";
                    lblMsg.Visible = true;
                    lblMsg.Text = "Category " + actionName + " 
successfully!";
                    lblMsg.CssClass = "alert alert-success";
                    getCategories();
                    clear();
                }

                catch (Exception ex)
                {
                    lblMsg.Visible = true;
                    lblMsg.Text = "Error" + ex.Message;
                    lblMsg.CssClass = "alert alert-danger";

                }
                finally
                {
                    con.Close();
                }
            }
        }
```

```
        protected void btnClear_Click(object sender, EventArgs e)
        {
            clear();
        }


         void clear()
        {
            txtCategoryName.Text = string.Empty;
            cbIsActive.Checked = false;
            hfCategoryId.Value = "0";
            btnAddOrUpdate.Text = "Add";
            ImagePreview.ImageUrl = string.Empty;
        }


        protected void rCategory_ItemCommand(object source,
RepeaterCommandEventArgs e)
        {
            lblMsg.Visible = false;
            if (e.CommandName == "Edit")
            {
                con = new SqlConnection(Utils.getConnection());
                cmd = new SqlCommand("Category_Crud", con);
                cmd.Parameters.AddWithValue("@Action", "GETBYID");
                cmd.Parameters.AddWithValue("@CategoryId", e.CommandArgument);
                cmd.CommandType = CommandType.StoredProcedure;
                sda = new SqlDataAdapter(cmd);
                dt = new DataTable();
                sda.Fill(dt);
                txtCategoryName.Text = dt.Rows[0]["CategoryName"].ToString();
                cbIsActive.Checked = Convert.ToBoolean(dt.Rows[0]["IsActive"]);
                ImagePreview.ImageUrl =
string.IsNullOrEmpty(dt.Rows[0]["CategoryImgUrl"].ToString()) ?
"../Images/No_Image.png" : "../" + dt.Rows[0]["CategoryImgUrl"].ToString();
                ImagePreview.Height = 200;
                ImagePreview.Width = 200;
                hfCategoryId.Value = dt.Rows[0]["CategoryId"].ToString();
                btnAddOrUpdate.Text = "Update";
            }

            else if (e.CommandName == "Delete")
            {
                con = new SqlConnection(Utils.getConnection());
                cmd = new SqlCommand("Category_Crud", con);
                cmd.Parameters.AddWithValue("@Action", "DELETE");
                cmd.Parameters.AddWithValue("@CategoryId", e.CommandArgument);
                cmd.CommandType = CommandType.StoredProcedure;
                try {

                    con.Open();
                    cmd.ExecuteNonQuery();
                    lblMsg.Visible = true;
                    lblMsg.Text = "Category Deleted Successfully!";
                    lblMsg.CssClass = "alert alert-success";
                    getCategories();

                }
```

```
            catch (Exception ex)
            {
                lblMsg.Visible = true;
                lblMsg.Text = "Error" + ex.Message;
                lblMsg.CssClass = "alert alert-danger";

            }
            finally
            {
                con.Close();
            }
        }

    }
}
    }
```

3. Adding Product :-



Description:-

- This Page is used to Display Item with all fields related above.
- Here Evey field can be Updated, Created and Deleted.
- Here Admin can decide which items can be in Menu and which item can not.

Coding:-

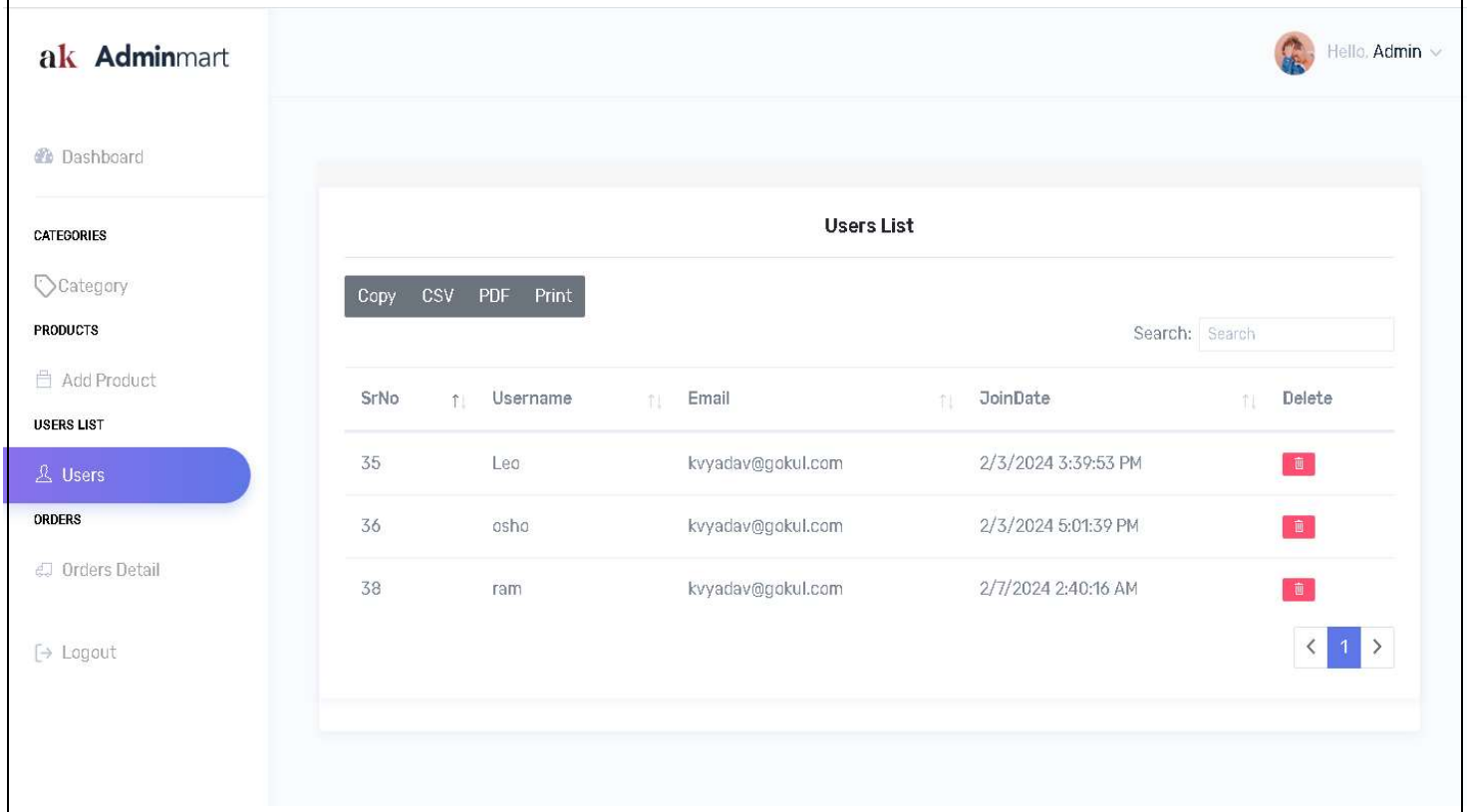
```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;
using System.Reflection.Emit;


namespace Aromakart.Admin
{
    public partial class WebForm3 : System.Web.UI.Page
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataAdapter sda;
        DataTable dt;
        protected void Page_Load(object sender, EventArgs e)
        {

                Session["breadCumbTitle"] = "Manage Category";
                Session["breadCumbPage"] = "Product";
                lblMsg.Visible = false;
                getProducts();


            lblMsg.Visible = false;
        }

        protected void btnAddOrUpdate_Click(object sender, EventArgs e)
        {
            {
                string actionName = string.Empty, imgPath = string.Empty,
fileExtension = string.Empty;
                bool isValidToExecute = false;
                int productId = Convert.ToInt32(hdnId.Value);
                con = new SqlConnection(Utils.getConnection());
                cmd = new SqlCommand("Product_Crud", con);
                cmd.Parameters.AddWithValue("@Action", productId == 0 ? "INSERT"
: "UPDATE");
                cmd.Parameters.AddWithValue("@ProductId", productId);
                cmd.Parameters.AddWithValue("@ProductName", txtName.Text.Trim());
                cmd.Parameters.AddWithValue("@ProductDesc", txtDesc.Text.Trim());
                cmd.Parameters.AddWithValue("@ProductPrice",
txtPrice.Text.Trim());
                cmd.Parameters.AddWithValue("@ProductQuantity",
txtQuantity.Text.Trim());
                cmd.Parameters.AddWithValue("@CategoryId",
ddlCategories.SelectedValue);
                cmd.Parameters.AddWithValue("@IsActive", cbIsActive.Checked);
                if (fuProductImg.HasFile)
                {
```

```
                    if (Utils.isValidExtension(fuProductImg.FileName))
                    {
                        string newImageName = Utils.getUniqueId();
                        fileExtension = Path.GetExtension(fuProductImg.FileName);
                        imgPath = "Images/Product/" + newImageName.ToString() +
fileExtension;

fuProductImg.PostedFile.SaveAs(Server.MapPath("~/Images/Product/") +
newImageName.ToString() + fileExtension);
                        cmd.Parameters.AddWithValue("@ProductImgUrl", imgPath);
                        isValidToExecute = true;
                    }
                    else
                    {
                        lblMsg.Visible = false;
                        lblMsg.Text = "Please select .jpg, or .png image";
                        lblMsg.CssClass = "alert alert-danger";
                        isValidToExecute = false;
                    }
                }
                else
                {
                    isValidToExecute = true;
                }
                if (isValidToExecute)
                {
                    cmd.CommandType = CommandType.StoredProcedure;
                    try
                    {
                        con.Open();
                        cmd.ExecuteNonQuery();
                        actionName = productId == 0 ? "inserted" : "updated";
                        lblMsg.Visible = true;
                        lblMsg.Text = "Product " + actionName + " 
successful!";
                        lblMsg.CssClass = "alert alert-success";
                        getProducts();
                        clear();
                    }
                    catch (Exception ex)
                    {
                        lblMsg.Visible = true;
                        lblMsg.Text = "Error" + ex.Message;
                        lblMsg.CssClass = "alert alert-danger";

                    }
                    finally
                    {
                        con.Close();
                    }
                }
            }
        }
        void getProducts()
        {
            con = new SqlConnection(Utils.getConnection());
            cmd = new SqlCommand("Product_Crud", con);
            cmd.Parameters.AddWithValue("@Action", "SELECT");
            cmd.CommandType = CommandType.StoredProcedure;
            sda = new SqlDataAdapter(cmd);
            dt = new DataTable();
            sda.Fill(dt);
```

```
            rProduct.DataSource = dt;
            rProduct.DataBind();

        }
        void clear()
        {
            txtName.Text = string.Empty;
            txtDesc.Text = string.Empty;
            txtQuantity.Text = string.Empty;
            txtPrice.Text = string.Empty;
            ddlCategories.ClearSelection();
            cbIsActive.Checked = false;
            hdnId.Value = "0";
            btnAddOrUpdate.Text = "Add";
            imgProduct.ImageUrl = string.Empty;
        }

        protected void btnClear_Click(object sender, EventArgs e)
        {
            clear();
        }

        protected void rProduct_ItemCommand(object source,
RepeaterCommandEventArgs e)
        {
            lblMsg.Visible = false;

            if (e.CommandName == "Edit")
            {
                con = new SqlConnection(Utils.getConnection());
                cmd = new SqlCommand("Product_Crud", con);
                cmd.Parameters.AddWithValue("@Action", "GETBYID");
                cmd.Parameters.AddWithValue("@ProductId", e.CommandArgument);
                cmd.CommandType = CommandType.StoredProcedure;
                sda = new SqlDataAdapter(cmd);
                dt = new DataTable();
                sda.Fill(dt);
                txtName.Text = dt.Rows[0]["ProductName"].ToString();
                txtDesc.Text = dt.Rows[0]["ProductDesc"].ToString();
                txtPrice.Text = dt.Rows[0]["ProductPrice"].ToString();
                txtQuantity.Text = dt.Rows[0]["ProductQuantity"].ToString();
                ddlCategories.SelectedValue =
dt.Rows[0]["CategoryId"].ToString();
                cbIsActive.Checked = Convert.ToBoolean(dt.Rows[0]["IsActive"]);
                imgProduct.ImageUrl =
string.IsNullOrEmpty(dt.Rows[0]["ProductImgUrl"].ToString()) ?
"../Images/No_Image.png" : "../" + dt.Rows[0]["ProductImgUrl"].ToString();
                imgProduct.Height = 200;
                imgProduct.Width = 200;
                hdnId.Value = dt.Rows[0]["ProductId"].ToString();
                btnAddOrUpdate.Text = "Update";
            }
            else if (e.CommandName == "Delete")
            {
                con = new SqlConnection(Utils.getConnection());
                cmd = new SqlCommand("Product_Crud", con);
                cmd.Parameters.AddWithValue("@Action", "DELETE");
                cmd.Parameters.AddWithValue("@ProductId", e.CommandArgument);
                cmd.CommandType = CommandType.StoredProcedure;
                try
                {
                    con.Open();
```

```
                cmd.ExecuteNonQuery();
                lblMsg.Visible = true;
                lblMsg.Text = "Product Deleted Successfully!";
                lblMsg.CssClass = "alert alert-success";
                getProducts();

            }
            catch (Exception ex)
            {
                lblMsg.Visible = true;
                lblMsg.Text = "Error" + ex.Message;
                lblMsg.CssClass = "alert alert-danger";

            }
            finally
            {
                con.Close();
            }
        }
    }
```

4. Manage Users:-



Description:-

- This is Users List Page where Admin can have right to abandon users A/C
- This Page is used to store user details for future reference.

Coding:-

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Security.Cryptography;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Aromakart.Admin
{
    public partial class WebForm4 : System.Web.UI.Page
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataAdapter sda;
        DataTable dt;
        protected void Page_Load(object sender, EventArgs e)
        {

            getUsers();

        }
        protected void rUsers_ItemCommand(object source, RepeaterCommandEventArgs
e)
        {
              if (e.CommandName == "Delete")
            {
                con = new SqlConnection(Utils.getConnection());
                cmd = new SqlCommand("User_Crud", con);
                cmd.Parameters.AddWithValue("@Action", "DELETE");
                cmd.Parameters.AddWithValue("@UserId", e.CommandArgument);
                cmd.CommandType = CommandType.StoredProcedure;
                try
                {
                    con.Open();
                    cmd.ExecuteNonQuery();
                    lblMsg.Visible = true;
                    lblMsg.Text = "User Deleted Successfully!";
                    lblMsg.CssClass = "alert alert-success";
                    getUsers();

                }
                catch (Exception ex)
                {
                    lblMsg.Visible = true;
                    lblMsg.Text = "Error" + ex.Message;
                    lblMsg.CssClass = "alert alert-danger";

                }
                finally
                {
```

```
                con.Close();
            }


        }
         }

         void getUsers()
         {
             con = new SqlConnection(Utils.getConnection());
             cmd = new SqlCommand("User_Crud", con);
             cmd.Parameters.AddWithValue("@Action", "SELECT4ADMIN");
             cmd.CommandType = CommandType.StoredProcedure;
             sda = new SqlDataAdapter(cmd);
             dt = new DataTable();
             sda.Fill(dt);
             rUsers.DataSource = dt;
             rUsers.DataBind();

         }


    }
}
```

5. Order Status:-

Description:-

- This page is used to display all the orders which is received by the User side.
- Here Admin can submit the received orders Status, which will after displayed in the User orders history page.

Coding:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.OleDb;
public partial class _Default :System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
DLbind();
    }

    protected void DLbind()
    {
        string sql = "";
OleDbConnection conn = new OleDbConnection();
        // establish connection
conn.ConnectionString = Session["cstring"].ToString();
conn.Open(); // connection open
        // sql query
sql = "select * from OrdersHistory";
OleDbCommandcmdd = new OleDbCommand(sql, conn);
OleDbDataReader reader = cmdd.ExecuteReader();
reader.Read();
        if (reader.HasRows == true)
        {
            DataList1.DataSource = reader;
            DataList1.DataBind();
        }
        //Label3.Text += reader["password"].ToString();
conn.Close(); // connection close

    }
          string sqlQuery = "";
          string queryf = "";
          protected void DataList1_ItemCommand(object source,
      DataListCommandEventArgs e)
    {


        if (e.CommandName == "Submitbtn")
        {
            try
            {
```

```
                string status =
((DropDownList)e.Item.FindControl("DropDownList1")).SelectedItem.Text;
                string ID = ((Label)e.Item.FindControl("Label3")).Text;
sqlQuery = Session["cstring"].ToString();
OleDbConnectionconnectionC = new OleDbConnection(sqlQuery);

                // queryf = "INSERT INTO OrdersHistory (Status)
VALUES('" + status + "') WHERE ID=" + ID;
queryf = "UPDATE OrdersHistorySET  Status=('" + status + "') WHERE ID="
+ ID;
OleDbCommandoledmcmd = new OleDbCommand(queryf, connectionC);
connectionC.Open();
oledmcmd.ExecuteNonQuery();
connectionC.Close();

            }
            catch (Exception ex)
            {
Response.Write("Error " + ex.Message + "<br>" + sqlQuery);
            }

        }

    }
}
```

Chapter No: 6

Limitations and Future Enhancement

6.1 Limitations

6.2 Further Enhancement

6.1 Limitations

There are some Limitations to this Project. Which are as the Follows:-

- The Website Admin cannot change his password.
- There is only one admin on this website and admin login details are static in the database. We are not able to register the new admin for the admin panel.
- Right now the user Order status is not active and is under development.

6.2 Further Enhancement

- In Future I Will add an option to change password for the website admin.

- In the future I will add the registration system for the admin. By adding this feature admin will be able to divide his work. it will help him to Manage the website Easily

- In Future I Will Add an option to send reply to the students who have given feedbacks to the admin.

- In Future I will Make the Users Website more mobile-friendly.

Chapter No: 7

Conclusion

7.1 Conclusion

7.2 Advantage

7.1 Conclusion

In the End , the Project “AromaKart – An eCommerce Shop” is developed very well. All requirements of the client are full filled. Students can able to visit the website. students are able to register by email.

Users are able to log in & able to make order as per their requirements and able to pay for it using payment option.

Users are able to manage their cart. Users are able to manage their cart. Admin is able to log in to the Admin Panel of the Website and able to add remove & manage the menu of the website.

Admin is able to see the full details of the received orders by different students. Admin is also able to add or remove category for the items. Admin is able to manage menu to make It up to date.

Admin is also able to see the received feedbacks from the students. At last but not the least client gets satisfaction as per their requirements the website is completed.

7.2 Advantages

- Completely automated fragrance and organic oil web application.
- Detailed summary of orders placed, update status, etc.
- User can also order a Special Combo Pack which contains multiple Perfume items.
- Aromakart is design as easy way. So maintenance is also easy.
- Simple user-interface Admin Panel for creation and configuration of menu groups, menu items, etc.

Chapter No: 8

Bibliography & Reference

8.1 Bibliography

8.2 Reference

8.1:Bibliography

- First Started with the Design of the user side.
- Make the user-side Design Dynamic.
- Started with the Design of the Admin side.
- Make the Admin Panel Dynamic.
- Write a back end code for the admin side.
- Write the back end Code for the User side.

8.2:Reference

I am thankful to my guide Prof. ZALAK THAKRAR for guiding me. I am also Thankful to the whole staff of the computer Department to gives us huge support for my project.

Sites:-

- https://mdbootstrap.com
- https://getbootstrap.com/docs/5.0/getting-started/introduction/
- https://www.c-sharpcorner.com/
- www.stackoverflow.com
- https://youtube.com/@TechTipsUnlimited?si=Op3vyjmgyV3buHgu