**RANI CHANNAMMA UNIVERSITY BELAGAVI**



B.L.D.E.A’s

Commerce, BHS Arts and TGP Science College, Jamakhandi



A Project Report On

“Annapurna Food Fantacy”

Submitted in partial fulfillment of requirement for

The award of the degree

BACHELOR OF COMPUTER APPLICATION

**Guides**

Prof. P.V.Tilavalli and Prof. V.N.Tubachi

Submitted By

Miss. PRIYANKA .S. KALLIMATH (M1423629).

Miss.RASHMI.R.SONAR(M1423631)

Mr.Prashant Sindhur(M1423625)

**RANI CHANNAMMA UNIVERSITY BELAGAVI**



B.L.D.E.A’s

Commerce, BHS Arts and TGP Science College, Jamakhandi



A Project Report On

“Annapurna Food Fantacy”

**CERTIFICATE**

**This is to certify that Miss. Priyanka S. Kallimath (M1423629)**

**have satisfactorily completed Project work entitled “**Annapurna Food Fantacy**” for the partial fulfillment of Bachelor of Computer Application (BCA) by Rani Channamma University, Belagavi for the Academic Year 2016-17.**

### Prof. P.V.Tilavalli and V.N.Tubachi Prof. B.K.Galagali Dr.S.C.Hiremath

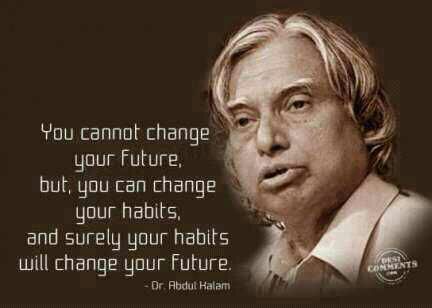
Guides Co-ordinator Principal

Examiners: 1) 2)

*My Respected Parents……*

*&*

*Lovely Friends,sister&Brother…!!!*



# ACKNOWLEDGEMENT

The successful presentation of this project is an acknowledgement of the immense support expended by B.L.D.E.A’s Commerce, B.H.S. Arts and T.G.P. Science College’s BCA Department which has provided an opportunity to fulfill the most cherished desire to reach my goal.

I express my sincere thanks to our reverend Principal **Dr.S.C.Hiremath** for extending all the facilities to carry out this project.

I express my sincere gratitude to our beloved **H.O.D Prof.B.K.Galagali** who has been a constant source of inspiration and motivation

I would also like to express out special thanks to my project guides **Prof.P.V.Tilavalli and Prof.V.N.Tubachi** , who have spent their precious time by guiding and encouraging me throughout the development of the project. Further I would like to thank our lecturers **Prof.S.S.Umadi**, **Prof.K.A.Patil**, **Prof.Praksh .Chanaveer and Prof.S.K.Danagar** for their support.

I am highly grateful to our guides **Prof.P.V.Titlavalli and Prof.V.N.Tubachi**  who helped me throughout the project. Last but not the least; I am grateful to my parents, my friends and all the people who have helped me directly or indirectly to make this project a success.

“A problem is a chance for you to do your best”.

Priyanka.S.Kallimath

**CONTENTS**

* **Objective & Scope of the Project**
* **Theoretical Background Definition of Problem**

* **System Analysis & Design**
* **User Requirements System Planning (PERT Chart)**
* **Methodology adopted**
* **System Implementation & Details of Hardware & Software used.**
* **System Maintenance & Evaluation**
* **Cost and benefit Analysis**
* **Detailed Life Cycle of the Project**

**ERD, DFD**

**Input and output screen design**

**Process involved**

**Methodology used for testing**

**Test Report, Printout of the Reports**

* + - * **User/Operational Manual - including security aspects, access rights, back up, controls, etc.**

**Objective and Scope of the Project**

The project **“Annapurna Food Fantacy”** is developed to develop an application which gives provision to the restaurant owners to flourish their business by uploading menus at no cost and will invariably lead to higher customer retention and acquisition rates.

The basic framework of the project is developed in .NET. Making use of this application the administrator can perform their activities through it.

**Scope of the Project:**

The project Annapurna Food Fantacy can be use to jamakhandi living people where as if they want to order the food online And for working women’s who don’t have time to prepare a quality food.

**Theoretical Problem and Definition of the Problem**

**EXISTING SYSTEM**

Online ordering system will be a web based application whose main language of programming will be c#. Its main aim is to simplify and improve the efficiency of the ordering process for both customer and restaurant, minimize manual data entry and ensure data accuracy and security during order placement process. Customers will also be able to view product menus and there ingredients and be able to have a visual confirmation that the order was place correctly.

**Problems in Existing System**

* No notification is sent when a customer cancel the order
* No notification is sent when a customer order the food
* Present System is very Time Consuming and Complicated.

**PROPOSED SYSTEM**

* Our project is used to send the message of order number to customer.
* Ordered food will reach to customer on time after ordering a food.
* By using this application the work of the Waiter is reduced and we can also say that the work is nullified. The benefit of this is that if there is rush in the Restaurant then there will be chances that the waiters will be unavailable and the users can directly order the food to the chef online by using this application

**Advantages**

* Reduce time-consuming phone orders and eliminate illegible fax orders.
* No more busy phones or the requirement for extra phone lines.
* Broader customer reaches across regions.
* Builds a customer database.
* Helps in improved service.
* Greater customer satisfaction!!!

**System Analysis and Design**

**System Requirement Specification**

A **software requirements specification** (SRS) is a description of a [software system](https://en.wikipedia.org/wiki/Software_system) to be developed. It lays out [functional](https://en.wikipedia.org/wiki/Functional_requirement) and [non-functional requirements](https://en.wikipedia.org/wiki/Non-functional_requirements), and may include a set of [use cases](https://en.wikipedia.org/wiki/Use_case) that describe user interactions that the software must provide.

**Functional Requirement:**

In [Software engineering](https://en.wikipedia.org/wiki/Software_engineering) and [systems engineering](https://en.wikipedia.org/wiki/Systems_engineering), a **functional requirement** defines a function of a [system](https://en.wikipedia.org/wiki/System) or its component. A function is described as a set of inputs, the behavior, and outputs.

**Features:**

The proposed web application serves the following activities.

* Online menus (original and searchable format)
* Easy lookup of restaurants in your area
* Check Ratings and Review the restaurants
* Simple, fast and convenient ordering of food
* Accurate – no more spelling out the dishes’ names.
* Menu with the actual pictures of the product thereby adding to the uniqueness of your online presence.
* Prior knowledge of time for delivery helps prepare and provide better service.
* Receive direct customer feedback and suggestions.
* Keep the customers informed.

**Non-Functional Requirement:**

In [systems\_engineering](https://en.wikipedia.org/wiki/Systems_engineering)  and  [requirements\_engineering](https://en.wikipedia.org/wiki/Requirements_engineering), a **non-functional requirement** is a [requirement](https://en.wikipedia.org/wiki/Requirement) that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

**Performance:**

**Computer performance** is the amount of work accomplished by a computer system. Depending on the context, high computer performance may involve one or more of the following:

* Short [response time](https://en.wikipedia.org/wiki/Response_time_(technology)) for a given piece of work
* High [throughput](https://en.wikipedia.org/wiki/Throughput) (rate of processing work)
* Low utilization of [computing resource](https://en.wikipedia.org/wiki/Computing_resource)s
* [High availability](https://en.wikipedia.org/wiki/High_availability) of the computing system or application
* Fast or highly compact [data compression](https://en.wikipedia.org/wiki/Data_compression) and decompression
* High [bandwidth](https://en.wikipedia.org/wiki/Bandwidth_(computing))
* Short [data transmission](https://en.wikipedia.org/wiki/Data_transmission) time

**System Requirement**

Computer system is made up of units that are put together to work as one in order to achieve a common goal. The requirements for the implementation of the new system are:

* The Hardware
* The software

## Hardware Requirements:

The selection of hardware is very important in the existence and proper working of any software. When selecting hardware, the size and requirements are also important.

The proposed System is developed on:

* + - Processor : Pentium IV
    - RAM : 2GB
    - Hard Disk Drive : 250GB
    - LAN Cable
    - Inter Connection Requirement
    - LAN Driver
    - Optical Mouse
    - Standard USA Layout Keyboard
    - Monitor: colour 15inch

**Software Requirements:**

The software’s on which the project is proposed to be developed has the following configuration:

* Operating System : Windows 7 on words
* IDE: Visual Studio 2008or above
* Design: ASP.NET, HTML, CSS, Framework 3.5
* DBMS SQL Server 2005
* Other Tools: Microsoft Office, Visual Paramedic

All of the application data is stored in a Oracle database, and therefore a Oracle Database must also be installed on the host computer. As with Apache2, this software is freely available and can be installed and run under most operating systems. The server hardware can be any computer capable of running both the web and database servers and handling the expected traffic. For a small scale restaurant that is not expecting to see much web traffic, an average personal computer may be appropriate. Once the site starts generating more hits, though, it will likely be necessary to upgrade to a dedicated host to ensure proper performance. The exact cutoffs will need to be determined through a more thorough stress testing of the system.

**Use Case Model**

The use-case concept is used to present functionality of the system described in a number of use-cases, each of which represents a specific flow of events in the system. A use-case diagram is a graph of actors, a set of use cases enclosed by a system boundary, and generalization among the use-cases**.**

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap and synergy with the disciplines of systems analysis, systems architecture and systems engineering.

Use case diagram consists of use cases and actors and shows the interaction between them. The key points are:

* The main purpose is to show the interaction between the use cases and the actor.
* To represent the system requirement from user’s perspective.
* The use cases are the functions that are to be performed in the module.
* An actor could be the end-user of the system or an external system.

**Use Case Model**

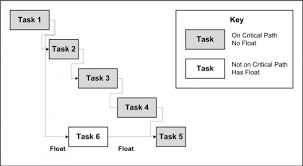
**Admin**

Customer

**User Requirements System Planning (PERT Chart)**

A PERT chart is a project management tool used to schedule, organize, and coordinate tasks within a project. PERT stands for Program Evaluation Review Technique. PERT charts allow the tasks in a particular project to be analyzed, with particular attention to the time required to complete each task, and the minimum time required to finish the entire project.

A PERT chart is a graph that represents all of the tasks necessary to a project's completion, and the order in which they must be completed along with the corresponding time requirements. Certain tasks are dependent on serial tasks, which must be completed in a certain sequence. Tasks that are not dependent on the completion of other tasks are called parallel or concurrent tasks and can generally be worked on simultaneously. PERT charts are preferable to Gantt charts because they more clearly identify task dependencies; however, the PERT chart is often more challenging to interpret. As such, project managers frequently employ both methodologies



**Methodologies Adopted**

The **waterfall model**  is a [sequential](https://en.wikipedia.org/wiki/Sequence) (non-iterative) [design](https://en.wikipedia.org/wiki/Design)  process, used in [software development processes](https://en.wikipedia.org/wiki/Software_development_process), in which progress is seen as flowing steadily downwards (like a [waterfall](https://en.wikipedia.org/wiki/Waterfall)) through the phases of conception, initiation, [analysis](https://en.wikipedia.org/wiki/Analysis), [design](https://en.wikipedia.org/wiki/Software_design), construction, [testing](https://en.wikipedia.org/wiki/Software_testing),  [production/implementation](https://en.wikipedia.org/wiki/Implementation) and [maintenance](https://en.wikipedia.org/wiki/Software_maintenance).

**Diagram of Waterfall - model:**

The following illustration is a representation of the different phases of the Waterfall Model.



The sequential phases in Waterfall model are −

* **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
* **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
* **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
* **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

## Modules Description of the Project:

* **ADMIN**
* **CUSTOMER**

**ADMIN Module:**

* Admin having the all the privileges over the project.
* He can add the items in the menus. So that customer can be viewed and put a order.
* He can delete, update, view the items.
* He can add the new Offers. So that customer can be view and place a order.
* He can delete, update, view the items.
* He can view the Ordered list of the customer.
* He can view the orders who have cancel the orders.

**Customer Module:**

* Customer can resister and use the web application.
* He can view the new offers and place an order.
* He can search the items and place order apart from new offer.
* He can give a feed back and can view the feed back in main page.
* He want to cancel the order the he can cancel order within half an hour.

**System Implementation & Details of Hardware & Software used.**

Implementation is the carrying out, execution, or practice of a plan, a method, or any design for doing something. As such, implementation is the action that must follow any preliminary thinking in order for something to actually happen. In an information technology context, implementation encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, running, testing, and making necessary changes. The word deployment is sometimes used to mean the same thing. In this sense implementation is an ongoing process to implement this project “Pay Roll Manager”.

## The features of the tool used to design the front end

Windows Form application for the Microsoft.NET framework using C#. The sample application demonstrates how to create and layout controls on a simple form and the handling of mouse click events. The application displays a form showing attributes of a file. This form is similar to the properties dialog box of a file (Right click on a file and Click on Properties menu item). Since attributes of a file will be shown, the sample will show how to use File IO operations in .NET framework.

Most of the C# core functionality is implemented in the System namespace. For forms application, the functionality is included in the System.WinForms namespace. Therefore, right at the top of our source file we need to define these namespaces.

Using System;

Using System.WinForms;

Using System.Data.Sql;

These can be used to create an user friendly GUI, graphical user interface for this the application will use front end as C# .NET which involves the drag and drop options to design forms

The forms are design using labels, textboxes, buttons, image, image button, combo box, datagridview, radio buttons etc.

# The features of the db used to design the back end

For back end using MS SQL Server 2008 supports having a wide range of users access it at the same time. An instance of MS SQL Server 2008 includes the files that make up a set of database and copy of the DBMS software. Applications running on separate computer use a SQL Server 2008 communications component to transmit commands over a network to the SQL Server 2008; it can reference any of the databases in that instance that the user is authorized to access. The communication component also allows communication between an instance of SQL Server 2008 and an application running on the computer.

## The connectivity process, installation process, execution process

## Connectivity:

* The MS SQL Server 2008 is used for provider for connection purpose.
* Using System.Data.Client; is used for namespace operations
* SqlConnection class is used for Connection to sqlserver.

**How to load and Run:**

* First we create exe file and we execute this file in client machine.
* Double click on the executable file the application will run.
* Software testing is a critical element of software quality assurance and represents the ultimate review of specification,design and code generation. Testing stratergies used are.

**Microsoft. NET Framework**

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

* To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
* To provide a code-execution environment that minimizes software deployment and versioning conflicts.
* To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
* To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
* To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
* To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

**Features of the Common Language Runtime:**

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally featuring rich.

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety.

In addition, the managed environment of the runtime eliminates many common software issues. For example, the runtime automatically handles object layout and manages references to objects, releasing them when they are no longer being used. This automatic memory management resolves the two most common application errors, memory leaks and invalid memory references.

**.NET Framework Class Library**

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is Object Oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access. In addition to these common tasks, the class library includes types that support a variety of specialized development scenarios. For example, you can use the .NET Framework to develop the following types of applications and services:

* Console applications.
* Scripted or hosted applications.
* Windows GUI applications (Windows Forms).
* ASP.NET applications.
* XML Web services.
* Windows services.

**Client Application Development**

Client applications are the closest to a traditional style of application in Windows-based programming. These are the types of applications that display windows or forms on the desktop, enabling a user to perform a task. Client applications include applications such as word processors and spreadsheets, as well as custom business applications such as data-entry tools, reporting tools, and so on. Client applications usually employ windows, menus, buttons, and other GUI elements, and they likely access local resources such as the file system and peripherals such as printers.

Another kind of client application is the traditional ActiveX control deployed over the Internet as a Web page. This application is much like other client applications: it is executed natively, has access to local resources, and includes graphical elements.

In the past, developers created such applications using C/C++ in conjunction with the Microsoft Foundation Classes (MFC) or with a rapid application development (RAD) environment such as Microsoft® Visual Basic®. The .NET Framework incorporates aspects of these existing products into a single, consistent development environment that drastically simplifies the development of client applications.

The Windows Forms classes contained in the .NET Framework are designed to be used for GUI development. You can easily create command windows, buttons, menus, toolbars, and other screen elements with the flexibility necessary to accommodate shifting business needs.

For example, the .NET Framework provides simple properties to adjust visual attributes associated with forms. In some cases the underlying operating system does not support changing these attributes directly, and in these cases the .NET Framework automatically recreates the forms. This is one of many ways in which the .NET Framework integrates the developer interface, making coding simpler and more consistent.

Unlike ActiveX controls, Windows Forms controls have semi-trusted access to a user's computer. This means that binary or natively executing code can access some of the resources on the user's system without being able to access or compromise other resources. Because of code access security, many applications that once needed to be installed on a user's system can now be safely deployed through the Web. Your applications can implement the features of a local application while being deployed like a Web page.

**Introduction to c#.NET**

C# programming language and .NET base class libraries using a friendly and approachable tone. The first edition of C# and .NET was published in conjunction with the release of .NET Beta 2, circa the summer of 2001.

Given that .NET is such a radical departure from the current thoughts of day, Microsoft has developed a new programming language (C#) specifically for this new platform. C# is a programming language that looks very similar (but not identical) to the syntax of java. For example, like java, a C# class definition is contained within a single-source code file (\*.cs) rather than the C++-centric view of splitting a class definition into discrete header (\*.h) and implementation (\*.cpp) files.

NET framework is completely new model for building systems on the Windows family of operating systems and in future.

The numerous benefits provided by C# and the .NET platform:

**Life As a C/Win32 API (Application Programming Interface) Programmer**.

Developing software for the Windows family of operating systems involved using the C programming language in conjunction with the Windows API.

**Life AS a C++/MFC (Microsoft Foundation Classes) Programmer.**

C++ can be thought of as an object-oriented layer on top of c. Thus, even though C++ programmers benefit from the famed “pillars of OOP”. The main role of MFC is to wrap a “sane subset” of the raw Win32 API behind a number of classes, magic macros and numerous code wizards.

**Life As a Visual Basic 6.0 Programmer.**

Many programmers have shifted away from the world of C(++)-based frameworks to kinder, gentler languages such as Visual Basics 6.0. VB is popular due to its ability to build complex user interfaces, code libraries (e.g., COM servers) and data access logic with minimal fuss and bother.

**Life As a Java/J2EE Programmer.**

Enter Java. The Java programming language is (almost) completely object oriented and has syntactic roots in C++. Although Java is a very elegant language.

**Life As a COM (Component Object Model) Programmer.**

COM is architecture that says in effect, “If you build your classes in accordance with the rules of COM, you end up with up with a block of reusable binary code”. The beauty of COM server is that it can be accessed in a language-independent manner.

**Standard controls:**

C# .NET has various drag and drop controls which are user friendly like textbox to enter the data, button is used to perform some action etc.

**Usage:**

* Text boxes are used in login page to fetch username and password from user.
* Dropdown menu is used to select topic when a admin is necessary.
* Button is used for save, update, close, delete etc.

**Data bound controls:**

Data bound controls are used when information to be displayed is more than one row. Ex, search results then we go for data bound controls. It has many such controls like- grid view, repeater, data list, details view and form view. They all used to display information but not to insert it.

**Usage:**

* Grid view to display short all the details.
* Grid view is also used here to show the all rows information.

**Validation controls:**

C# .NET facilitates us with providing standard controls for validation. It provides validation controls like-required field validation, regular expression validation, range validation, compare validation &on.

**Usage:**

* In login page, user name and password are validated for required field’s validation.
* In change password page required fields validation is kept for old password and new password in New profile page
* In Customer page, first name, middle name, last name, phone numbers are validated for required field’s validation.
* For cell double click in data grid view in all pages are validated wherever it takes.

# SQL Server

C# .NET can interact with any database like Access, Oracle, MySQL, and SQL-Server. Etc, but the most popular in combination with ASP.NET is SQL-Server. This is because:

* In Visual Studio 2010 a built-in SQL-Server called SQL Express Edition is given hence we need not install any other database externally.
* .NET with SQL-Server works 30% faster than other database.

**Connectivity:**

For connectivity purpose SQL server needs 2 objects:

**Connection Object:** Connection object is used to connect any front-end application of .NET to a particular database like SQL Server.

For SQL Server provider is SQLOLEDB.

**Command Object:** It is use to execute any SQL Statement or stored procedures against database.

It has got 3 important methods-

i) **Execute Non-Query**- when we use commands like insert, update, Delete to add a record, to change a record or to remove a record then we should use this method along with command object.

ii) **Execute Reader**- When we use simple select statement to fetch one or more records. This method is used along with command object. To store the fetched rows we have to use an oledbreader object (dr).we can fetch first row by dr[0].

iii) **Execute Scalar**- when we use select statement with group Functions like Sum, Average, Min, Max, and count which return only one value from database.

**Usage:**

* Execute non query is used to perform insert, delete and update functions for all add, delete and edit forms respectively.
* Execute reader is used to fetch values from table.

**ASP.NET**

ASP.NET is part of the whole. NET framework, built on top of the Common Language Runtime (also known as the CLR) - a rich and flexible architecture, designed not just to cater for the needs of developers today, but to allow for the long future we have ahead of us. What you might not realize is that, unlike previous updates of ASP, ASP.NET is very much more than just an upgrade of existing technology – it is the gateway to a whole new era of web development.

ASP.NET is a feature at the following web server releases

* Microsoft IIS 5.0 on WINDOWS 2000 Server
* Microsoft IIS 5.1 on WINDOWS XP

**ASP.NET has several advantages over ASP.**

 The following are some of the benefits of ASP.NET:

* Make code cleaner.
* Improve deployment, scalability, and reliability.
* Provide better support for different browsers and devices.

**ActiveX**

ActiveX is a specification develops by Microsoft that allows ordinary Windows programs to be run within a Web page. ActiveX programs can be written in languages such as Visual Basic and they are complied before being placed on the Web server.

ActiveX application, called controls, are downloaded and executed by the Web browser, like Java applets. Unlike Java applets, controls can be installed permanently when they are downloaded; eliminating the need to download them again. ActiveX’s main advantage is that it can do just about anything.

This can also be a disadvantage:

Several enterprising programmers have already used ActiveX to bring exciting new capabilities to Web page, such as “the Web page that turns off your computer” and “the Web page that formats disk drive”.

Fortunately, ActiveX includes a signature feature that identifies the source of the control and prevents controls from being modified. While this won’t prevent a control from damaging system, we can specify which sources of controls we trust.

ActiveX has two main disadvantages

It isn’t as easy to program as scripting language or Java.

ActiveX is proprietary.

It works only in Microsoft Internet Explorer and only Windows platforms.

**ADO.NET**

ADO.NET provides consistent access to data sources such as Microsoft SQL Server, as well as data sources exposed via OLE DB and XML. Data-sharing consumer applications can use ADO.NET to connect to these data sources and retrieve, manipulate, and update data.

ADO.NET cleanly factors data access from data manipulation into discrete components that can be used separately or in tandem. ADO.NET includes .NET data providers for connecting to a database, executing commands, and retrieving results. Those results are either processed directly, or placed in an ADO.NET Dataset object in order to be exposed to the user in an ad-hoc manner, combined with data from multiple sources, or remote between tiers. The ADO.NET Dataset object can also be used independently of a .NET data provider to manage data local to the application or sourced from XML.

**Why ADO.NET?**

As application development has evolved, new applications have become loosely coupled based on the Web application model. More and more of today's applications use XML to encode data to be passed over network connections. Web applications use HTTP as the fabric for communication between tiers, and therefore must explicitly handle maintaining state between requests. This new model is very different from the connected, tightly coupled style of programming that characterized the client/server era, where a connection was held open for the duration of the program's lifetime and no special handling of state was required.

In designing tools and technologies to meet the needs of today's developer, Microsoft recognized that an entirely new programming model for data access was needed, one that is built upon the .NET Framework. Building on the .NET Framework ensured that the data access technology would be uniform—components would share a common type system, design patterns, and naming conventions.

**Leverage Current ADO Knowledge:**

Microsoft's design for ADO.NET addresses many of the requirements of today's application development model. At the same time, the programming model stays as similar as possible to ADO, so current ADO developers do not have to start from scratch in learning a brand new data access technology. ADO.NET is an intrinsic part of the .NET Framework without seeming completely foreign to the ADO programmer.

ADO.NET coexists with ADO. While most new .NET applications will be written using ADO.NET, ADO remains available to the .NET programmer through .NET COM interoperability services.

ADO.NET provides first-class support for the disconnected, n-tier programming environment for which many new applications are written. The concept of working with a disconnected set of data has become a focal point in the programming model. The ADO.NET solution for n-tier programming is the Dataset.

**XML Support**

XML and data access are intimately tied—XML is all about encoding data, and data access is increasingly becoming all about XML. The .NET Framework does not just support Web standards—it is built entirely on top of them

**Information Super Highway**

A set of computer networks, made up of a large number of smaller networks, using different networking protocols. The world's largest computing network consisting of over two million computers supporting over 20 millions users in almost 200 different countries. The Internet is growing a phenomenal rate between 10 and 15 percent. So any size estimates are quickly out of date.

**HTML:**

HTML (Hyper Text Markup Language) is the language that is used to prepare documents for online publications. HTML documents are also called Web documents, and each HTML document is known as Web page.

A page is what is seen in the browser at any time. Each Web site, whether on the Internet or Intranet, is composed of multiple pages. And it is possible to switch among them by following hyperlinks. The collection of HTML pages makes up the World Wide Web.

A web pages is basically a text file that contains the text to be displayed and references of elements such as images, sounds and of course hyperlinks to other documents. HTML pages can be created using simple text editor such as Notepad or a WYSIWYG application such as Microsoft FrontPage.

In either case the result is a plain text file that computers can easily exchange. The browser displays this text file on the client computer.

"Hypertext" is the jumping frog portion. A hyperlink can jump to any place within your own page(s) or literally to anyplace in the world with a 'net address (URL, or Uniform Resource Locator.) It's a small part of the html language.

**INTERNET INFORMATION SERVER (IIS):**

Internet Information Services (IIS) - IIS is the web server of choice for running ASP.NET web applications. You'll need your copy of the Windows CD to install and configure it. Unfortunately, some versions of Windows (such as Windows XP Home Edition) don't support IIS. If you're one of those users, there's Cassini. Cassini is a small web server designed for hobbyists who are looking to build ASP.NET web sites. It isn't as robust, powerful, or user-friendly as IIS, but it will be sufficient for our purposes.

A modern web browser - Throughout this book, we'll be using Internet Explorer 6, but you can use other browsers during development if you wish. Any modern browser will do.NET Framework 2.0 - As we've already discussed, the .NET Framework drives ASP.NET. When you install the .NET Framework, you'll automatically install the files necessary to run ASP.NET. You're likely to have the .NET Framework already, as it installs automatically through the Windows Update service.

.NET Framework Software Development Kit (SDK) - The .NET Framework 2.0 Software Development Kit (SDK) is a free download that contains the necessary Web Application development tools, a debugger for error correcting, and a suite of samples and documentation.

Microsoft Internet Information server (IIS) version 4.0 offers technologies to do this:

1. Windows scripting Host (WSH)

2. IIS Admin objects built on top of Active Directory service Interface(ADS))

**System Implementation**

The design of a Management Information System may seem to management to be an expensive project, the cost of getting the MIS on line satisfactorily may often be comparable to that of its design, and the implementation has been accomplished when the outputs of the MIS are continuously utilized by decision makers.

Once the design has been completed, there are four basic methods for implementing the MIS.   
These areas  
1. Install the system in a new operation or organization.   
2. Cut off the old system and install the new   
This produces a time gap during which no system is in operation. Practically, installation requires one or two days for small companies or small systems.   
3. Cut over by segments   
This method is also referred as” phasing in” the new system. Small parts or subsystems are substituted for the old. In the case of upgrading old systems, this may be a very desirable method.   
4. Operate in parallel and cut over.

The new system is installed and operated in parallel with the current system until it has been checked out, then only the current system is cut out. This method is expensive because of personal and related costs. Its big advantages are that the system is fairly well debugged when it becomes the essential information system.

**Plan the implementation**   
The three main phases in implementation take place in series.   
These are   
1. The initial installation   
2. The test of the system as a whole   
3. The evaluation, maintenance and control of the system.

Many implementation activities should be undertaken in parallel to reduce implementation time. Training of personnel and preparation of software may be in parallel with each other and with other implementation activities.   
The first step in the implementation procedure is to plan the implementation. Some analyst includes the planning of the implementation with the design of the system, the planning and the action to implement the plan should be bound closely together. Planning is the first step of management, not the last. The MIS design and the urgent need for the system at the time the design is completed will weigh heavily on the plan for implementation.

**Implementation Tasks**   
The major implementation tasks consists of-   
1. Planning the implementation activities   
2. Acquiring and laying out facilities and offices   
3. Organizing the personnel for implementation   
4. Developing procedures for installation and testing   
5. Developing the training program for operating personnel.   
6. Completing the system’s software   
7. Acquiring required hardware   
8. Generating files   
9. Designing forms   
10. Testing the entire system   
11. Completing cutover to the new system   
12. Documenting the system   
13. Evaluating the MIS

14. Providing system maintenance(debugging and improving)

### Cost & Benefit Analysis

**Cost–Benefit Analysis** (**CBA**), sometimes called **Benefit–Cost Analysis** (**BCA**), is a systematic approach to estimating the strengths and weaknesses of alternatives that satisfy transactions, activities or functional requirements for a business. It is a technique that is used to determine options that provide the best approach for the adoption and practice in terms of benefits in labor, time and cost savings etc.[[1]](http://en.wikipedia.org/wiki/Cost%E2%80%93benefit_analysis#cite_note-1) The CBA is also defined as a systematic process for calculating and comparing benefits and [costs](http://en.wikipedia.org/wiki/Cost) of a project, decision or [government policy](http://en.wikipedia.org/wiki/Government_policy) (hereafter, "project").

Broadly, CBA has two purposes:

1. To determine if it is a sound investment/decision (justification/feasibility),
2. To provide a basis for comparing projects. It involves comparing the total expected cost of each option against the total expected benefits, to see whether the benefits outweigh the costs, and by how much.[[2]](http://en.wikipedia.org/wiki/Cost%E2%80%93benefit_analysis#cite_note-2)

CBA is related to, but distinct from [cost-effectiveness](http://en.wikipedia.org/wiki/Cost-effectiveness) analysis. In CBA, benefits and costs are expressed in monetary terms, and are adjusted for the [time value of money](http://en.wikipedia.org/wiki/Time_value_of_money), so that all flows of benefits and flows of project costs over time (which tend to occur at different points in time) are expressed on a common basis in terms of their "[net present value](http://en.wikipedia.org/wiki/Net_present_value)."

Closely related, but slightly different, formal techniques include [cost-effectiveness](http://en.wikipedia.org/wiki/Cost-effectiveness) analysis, [cost–utility analysis](http://en.wikipedia.org/wiki/Cost%E2%80%93utility_analysis), [risk–benefit analysis](http://en.wikipedia.org/wiki/Risk%E2%80%93benefit_ratio), [economic impact analysis](http://en.wikipedia.org/wiki/Economic_impact_analysis), fiscal impact analysis, and [Social return on investment](http://en.wikipedia.org/wiki/Social_return_on_investment) (SROI) analysis.

**Process:**

The following is a list of steps that comprise a generic cost–benefit analysis.

1. List alternative projects/programs.
2. List [stakeholders](http://en.wikipedia.org/wiki/Stakeholder_(corporate)).
3. Select measurement(s) and measure all cost/benefit elements.
4. Predict outcome of cost and benefits over relevant time period.
5. Convert all costs and benefits into a common currency.
6. Apply [discount rate](http://en.wikipedia.org/wiki/Annual_effective_discount_rate).
7. Calculate [net present value](http://en.wikipedia.org/wiki/Net_present_value) of project options.
8. Perform [sensitivity analysis](http://en.wikipedia.org/wiki/Sensitivity_analysis).
9. Adopt recommended choice.

**Evaluation:**

CBA attempts to measure the positive or negative consequences of a project, which may include:

1. Effects on users or participants
2. Effects on non-users or non-participants
3. [Externality](http://en.wikipedia.org/wiki/Externality) effects
4. [Option value](http://en.wikipedia.org/wiki/Option_value) or other social benefits.

A similar breakdown is employed in environmental analysis of [total economic value](http://en.wikipedia.org/wiki/Total_economic_value). Both costs and benefits can be diverse. Financial costs tend to be most thoroughly represented in cost-benefit analyses due to relatively abundant market data. The net benefits of a project may incorporate cost savings or public [willingness to pay](http://en.wikipedia.org/wiki/Willingness_to_pay) compensation (implying the public has no legal right to the benefits of the policy) or [willingness to accept](http://en.wikipedia.org/wiki/Willingness_to_accept) compensation (implying the public has a right to the benefits of the policy) for the welfare change resulting from the policy. The guiding principle of evaluating benefits is to list all (categories of) parties affected by an intervention and add the (positive or negative) value, usually monetary, that they ascribe to its effect on their welfare.

**Time and Discount:**

CBA usually tries to put all relevant costs and benefits on a common temporal footing using [time value of money](http://en.wikipedia.org/wiki/Time_value_of_money) calculations. This is often done by converting the future expected streams of costs and benefits into a [present value](http://en.wikipedia.org/wiki/Present_value) amount using a discount rate. Empirical studies and a technical framework suggest that in reality, people do discount the future like this.

The choice of discount rate is subjective. A smaller rate values future generations equally with the current generation. Larger rates (e.g. a market rate of return) reflects humans' attraction to [time inconsistency](http://en.wikipedia.org/wiki/Time_inconsistency)—valuing money that they receive today more than money they get in the future. The choice makes a large difference in assessing interventions with long-term effects. One issue is the [equity premium puzzle](http://en.wikipedia.org/wiki/Equity_premium_puzzle), in which long-term returns on equities may be rather higher than they should be. If so then arguably market rates of return should not be used to determine a discount rate, as doing so would have the effect of undervaluing the distant future.

**Risk and Uncertainty:**

Risk associated with project outcomes is usually handled using probability theory. This can be factored into the discount rate (to have uncertainty increasing over time), but is usually considered separately. Particular consideration is often given to [risk aversion](http://en.wikipedia.org/wiki/Risk_aversion)—the irrational preference for avoiding loss over achieving gains. [Expected return](http://en.wikipedia.org/wiki/Expected_return) calculations does not account for the detrimental effect of uncertainty

Uncertainty in CBA parameters (as opposed to risk of project failure etc.) can be evaluated using a sensitivity analysis, which shows how results respond to parameter changes. Alternatively a more formal risk analysis can be undertaken using [Monte Carlo simulations](http://en.wikipedia.org/wiki/Monte_Carlo_simulation)

### Detailed Life Cycle of the Project

**Entity Relationship Diagram:**

* For database application, the entity relationship approach can be used effectively for modeling some part of the problem. The ER modeling approach is used to help design information system. The main focus of ER modeling is the data items in the system and relationship between them. It aims to create a conceptual schema for the data from the user perspective. The model created is the high-level data model.
* An ER diagram consist of the following components

It represents entity sets.

It represents attribute

It represents relationship set

**High Level Design:**

**Entities and Attributes:**

**Place Order Table:**

Place Order

**Order Table:**

Order table

**Cancle Order:**

Cancle\_table

**Registration Table:**

Customer reg\_table

**Admin Table:**

Admin\_table

**ItemMaster Table:**

ItemMaster

**NewOffer Table:**

Newoffer

**Ordernumber Table:**

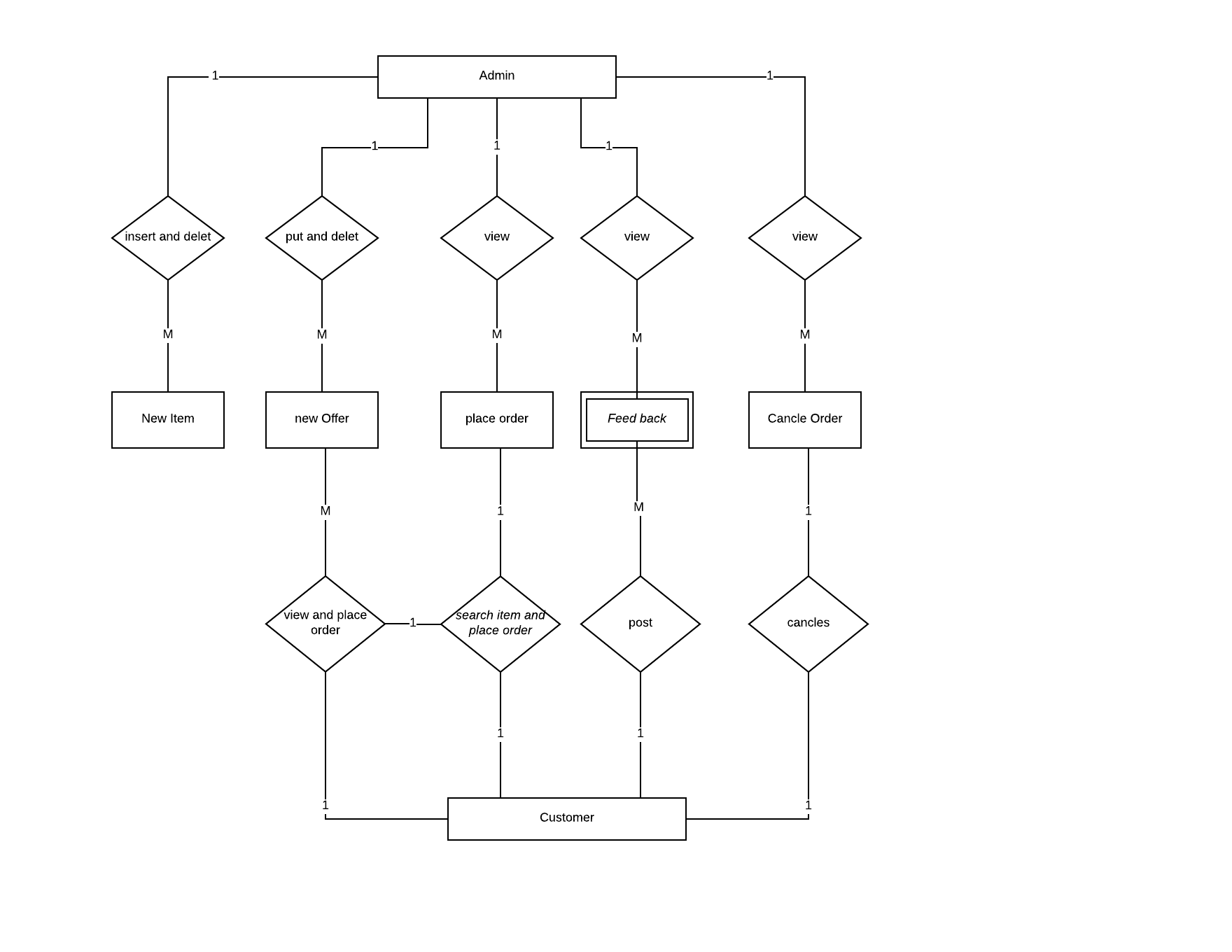
ordernum

**FeedBack Table:**

feedback

feedback

**ER Diagram:**

****

**Data Flow Diagram**

The data flow diagram is an important modeling tool. It shows the use of data pictorially. DFD represents the flow of data between different transformations and processes in the system. The DFD shows logical flow of data.

**Different notations used in DFD are**

**Process**

Processes show the system does. Each process has one or more inputs and none or more outputs. If a process does not have an output then it is considered to a Black Hole. The notation for the process is

**Functional Processing**

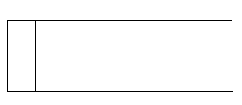
It is represented by an oval. The processing or main transactions are specified by this notation.

**Data Flow**

It is represented by an arrow line and name of the data is specified by the side of the line as label. This is used for the data movement.

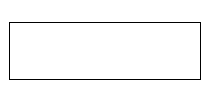
**Data Store**

It is represented by anyone open-end rectangle. The database used in the system is specified by this notation.

****

**Source or Sink**

It is represented by one open-end rectangle. It is used for specifying from where data comes and where it reaches.

****

**Low Level DFD:**

It is best compared to high level design. It is only the eloborated to the degree needed to satisfy the needs.

Admin

FOOD ITEMS

**High Level DFD :**

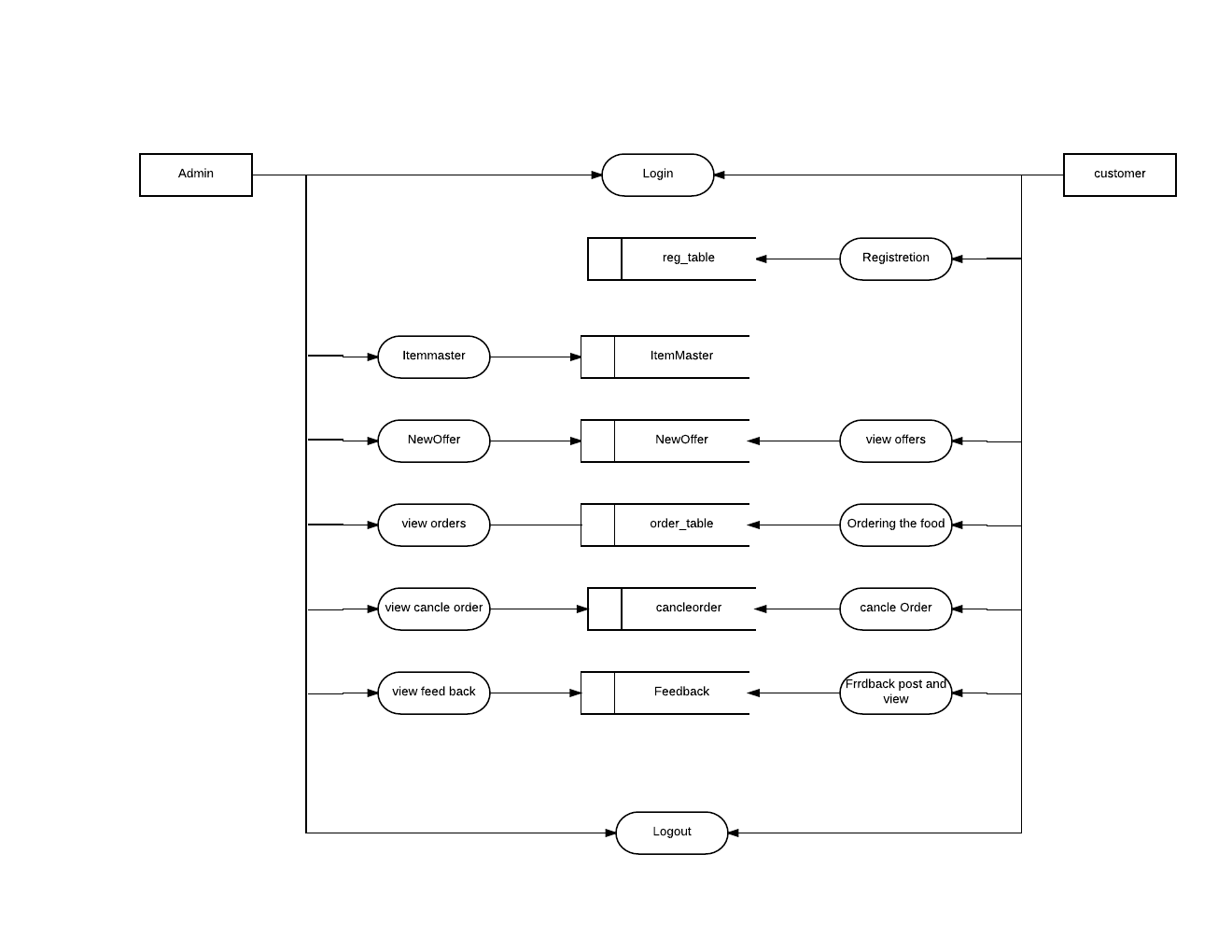
It is a complete view of entire system breaking it down in to smaller parts that are easily understood

ADMIN

New Offer

Food Items

**Overall Data Flow Diagram :**



## Low Level Design

### Relational Model

Relational Schema is the core idea to describe a database as a collection of predicates over a finite set of predicate variables, describing constrains on the possible values and combinations of values. The content of the database at any given times is a finite model (logic) of the database i.e. a set of relations, one per predicate variables such that all predicates are satisfied. A request for information from the database (a database query) is also predicates.

The purpose of the relational model is to provide a declarative method for specifying data and queries we directly state what information the database contains and what information we want from it, and let the

Following are the steps which can be used to convert ER-Components to a relational model.

**Step 1: For regular strong entity type E in the ER-schema**

1. Create a relation R to represent strong entity E and include all simple attributes of E as attributes of R.
2. Include only simple component of composite attribute if exist in E as attribute of R.
3. Choose one of the key attribute of E as the primary key of R.
4. If the chosen key attribute of E is composite, the set of simple attribute that from it will together from the primary key of R.
5. If E contains attribute of type multivalued or complex then follow step no 6 to convert it into relational concepts.

**Step 2: For every weak entity ‘w’ in the ER schema with owner entity type ‘E’**

1. Create a new relation R to represent ‘W’. Include all simple attribute (simple components of composite attributes) of W as attribute of relation R.
2. Include as foreign key attributes in R the primary key attribute of relation that corresponds to the owner entity type.
3. The primary key of R is the combination of foreign key and the partial key of ‘W’.

**Step 3: For each binary 1:1 relationship type R in the ER relational model**

1. Identify relations S and T that corresponds to the entity types participating in R.
2. Choose one of the relation say S and include as foreign key in the primary key of T.
3. It’s better to choose an entity type with total participation in R in the role of S.
4. Include all simple attribute (or simple components of composite attributes) of the 1:1 relationship type R as attribute of ‘S’.

**Step 4: For each binary 1:N relationship type R in the ER schema or ER model**

1. Identify the relation ‘S’ that represents the participating entity type at the N side of the relationship type R.
2. Include the foreign key in S the primary key of the relation ‘T’ that represents the other entity type participating in the relationship type R.
3. Include all simple attributes (simple components of composite attributes) of the 1: N relationship type as the attribute of ‘S’.

**Step 5: For each binary M: N relationship type R in the ER schema**

1. Create a new relation ‘S’ to represent R.
2. Include as foreign key in ‘S’ the primary key of relations that represents that participating entity type in the relationship R.
3. Include any simple attribute of R as attributes of ‘S’.
4. The primary key of ‘S’ is the combination of foreign key including S.

**Step 6: For each multivalued attribute A**

1. Create a new relation R.
2. This relation R will include an attribute corresponding to A plus the primary key attribute K as foreign key in R of the relation that represents the entity type or relationship type has A as an attribute.
3. The primary key of R is the combination of A and K. If A is composite we include its simple component as attribute of R.

**Step 7: For each n-ary relationship type R where n>2**

1. Create new relation S to represent R.
2. Include as foreign key in S the primary key of relations that represent or correspond to the participating entity type in R.
3. Include any simple attribute of R as attribute of S.
4. The primary key of S is the combination of all foreign key, if each entity associates with relation N.

**DATABASE TABLES:**

**Admin Table :**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE NAME** | **DATATYPE** | **CONSTRAINTS** |
| aid | int | Allow null |
| Username | Varchar(50) | Primary key |
| Password | Varchar(50) | Not null |

**ordernum Table:**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE NAME** | **DATATYPE** | **CONSTRAINTS** |
| **oid** | **int** | **Not null** |
| **Onum** | **Varchar(50)** | **Primary key** |

**Reg Table :**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE NAME** | **DATATYPE** | **CONSTRAINTS** |
| Id | Int | Allow null |
| uname | Varchar(50) | Not null |
| uloginname | Varchar(50) | Primary key |
| password | Varchar(50) | Not null |
| Location | Varchar(MAX) | Allow null |
| mobno | Numeric(18,0) | Allow null |
| email | Numeric(18,0) | Not null |

**FeedBack:**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE NAME** | **DATATYPE** | **CONSTRAINTS** |
| **uloginname** | **Varchar(50)** | **Not null** |
| **feedback** | **Varchar(MAX)** | **Not null** |

**Placeorder Table**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE NAME** | **DATATYPE** | **CONSTRAINTS** |
| iid | Numeric(18,0) | Primary key |
| itemName | Varchar(50) | Allow null |
| itemType | Varchar(50) | Not null |
| imageImage | Varchar(50) | Not null |
| price | int | Not null |

**Order table:**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE NAME** | **DATATYPE** | **CONSTRAINTS** |
| pid | int | Not null |
| oitemname | nvarchar(50) | Not null |
| image | Nvarchar(MAX) | Allow null |
| price | int | Allow null |
| date | Varchar(50) | Allow null |
| discription | Nvarchar(MAX) | Allow null |

**PlaceOrder table:**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE NAME** | **DATATYPE** | **CONSTRAINTS** |
| Grandtotal | Numeric(18,0) | Allow null |
| ordno | int | Not null |
| date | Varchar(50) | Allow null |
| Mobno | Numeric(18,0) | Allow null |
| adress | Varchar(MAX) | Not null |
| pid | Varchar(MAX) | Allow null |
| itemname | Varchar(50) | Allow null |
| itemtype | Varchar(50) | Allow null |
| price | Numeric(18,0) | Allow null |
| quantity | Numeric(18,0) | Allow null |
| total | Numeric(18,0) | Allow null |

**NewOffer:**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE NAME** | **DATATYPE** | **CONSTRAINTS** |
| pid | int | Not null |
| oitemname | nvarchar(50) | Not null |
| image | Nvarchar(MAX) | Allow null |
| price | int | Allow null |
| date | Varchar(50) | Allow null |
| discription | Nvarchar(MAX) | Allow null |

**Cancle order:**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTE NAME** | **DATATYPE** | **CONSTRAINTS** |
| id | int | Not null |
| onum | int | Not null |
| Product\_id | int | Allow null |
| Item\_Name | Varchar(50) | Allow null |
| date | Varchar(50) | Allow null |
| price | int | Allow null |
| Quntity | int | Allow null |
| Total | int | Allow null |
| resturant | Varchar(50) | Allow null |
| Del\_date | Varchar(50) | Allow null |
| Del\_time | Varchar(50) | Allow null |
| Del\_adress | Varchar(MAX) | Allow null |
| mobno | Numeric(18,0) | Allow null |

**Methodology Used for Testing**

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of conditions known as test cases and the output is evaluated to determine whether the program is performing as expected.

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

**Levels of Testing:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  |  |  |  |  |  |  | | --- | --- | --- | |  |  |  |  |  |  |  | | --- | --- | --- | |  |  |  |   Units Units Units  i/P integration o/p i/p integration o/P |   Module1 module2 module3 |   System Testing: Presentation +business + database  UAT: user acceptance testing |

**Types of Testing:**

**Unit Testing**

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

**Each module can be tested using the following two strategies:**

**Black Box Testing**

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories:

1. Incorrect or missing functions
2. Interface errors
3. Errors in data structure or external database access
4. Performance errors
5. Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked

**White Box testing**

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases.

It has been uses to generate the test cases in the following cases:

1. Guarantee that all independent paths have been executed.
2. Execute all logical decisions on their true and false sides.
3. Execute all loops at their boundaries and within their operational bounds.
4. Execute internal data structures to ensure their validity.

**Integrating Testing**

Integration testing ensures that software and subsystems work together as a whole. It tests the interface ofall the modules to make sure that the modules behave properly when integrated together.

**System Testing**

Involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user the system meets all requirements of the client's specifications.

**Acceptance Testing**

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

**Validation**

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

**Compilation Test:**

It was a good idea to do our stress testing early on, because it gave us time to fix some of the unexpected deadlocks and stability problems that only occurred when components were exposed to very high transaction volumes.

**Execution Test:**

This program was successfully loaded and executed. Because of good programming there was no execution error.

##### **Output Test:**

The successful output screens are placed in the output screens section above.

**TEST CASES:**

|  |  |  |
| --- | --- | --- |
| Test Case No | 1 | |
| Module | Admin | |
| User Form | Login form | |
| Input | Enter admin name and password | |
| **Admin name** | **Password** |
| Correct | Correct |
| Excepted Result | Successful login | |
| Test Result | Tested-Ok | |

|  |  |  |
| --- | --- | --- |
| Test Case No | 2 | |
| Module | Customer | |
| User Form | Login form | |
| Input | Enter Customer name and password | |
| **Customer user login name** | **Password** |
| Correct | Correct |
| Excepted Result | Successful login | |
| Test Result | Tested-Ok | |

|  |  |  |
| --- | --- | --- |
| Test Case No | 3 | |
| Page | Home Page | |
| User Form | Registration form | |
| Input | Enter customer user login name and password | |
| **User login name** | **Password** |
| Correct | Correct |
| Excepted Result | Successful Registration succefull | |
| Test Result | Tested-Ok | |

|  |  |  |
| --- | --- | --- |
| Test Case No | 4 | |
| Module | Admin | |
| User Form | Admin New offer form | |
| Input | Enter Product id and Item Name | |
| **Product id** | **Item Name** |
| Correct | Correct |
| Excepted Result | Successfully added | |
| Test Result | Tested-Ok | |

|  |  |  |
| --- | --- | --- |
| Test Case No | 5 | |
| Module | Customer | |
| User Form | Cancle form | |
| Input | Enter admin name and password | |
| **Order number** |  |
| Correct |  |
| Excepted Result | Successfully cancled orderss | |
| Test Result | Tested-Ok | |

|  |  |  |
| --- | --- | --- |
| Test Case No | 6 | |
| Module | Customer | |
| User Form | Feedback form | |
| Input | Enter user Login name and feedback | |
| **Userlogin name** | **Feedback** |
| Correct | Correct |
| Excepted Result | Successful post | |
| Test Result | Tested-Ok | |

## Test cases and results

**Test Case: 1**

**Description:**Admin has to login into login page.

|  |  |
| --- | --- |
| Input: | Correct username and password. |
| Desired output: | Admin login into his window login page. |
| On Error: | A message indicating invalid user. |
| If fields kept blank | A message indicating to fill the desired fields. |
| Remark | Ok. |

**Test case 2**

**Description**:Registretion user login name and password.

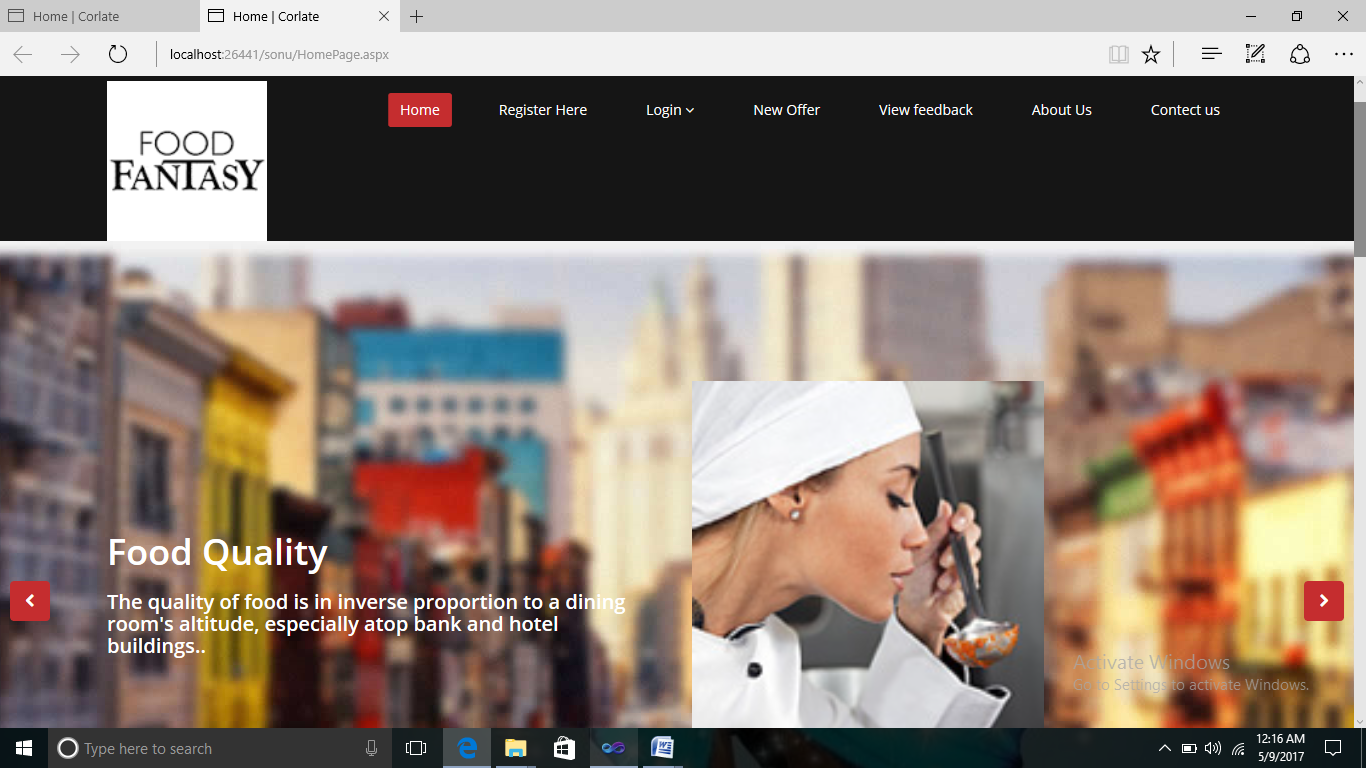
|  |  |
| --- | --- |
| Input: | Correct user login name and password. |
| Desired output: | User login into his window login page. |
| On Error: | A message indicating invalid user login name. |
| If fields kept blank | A message indicating to fill the desired fields. |
| Remark | Ok. |

**GLOSSARY OF TERMS**

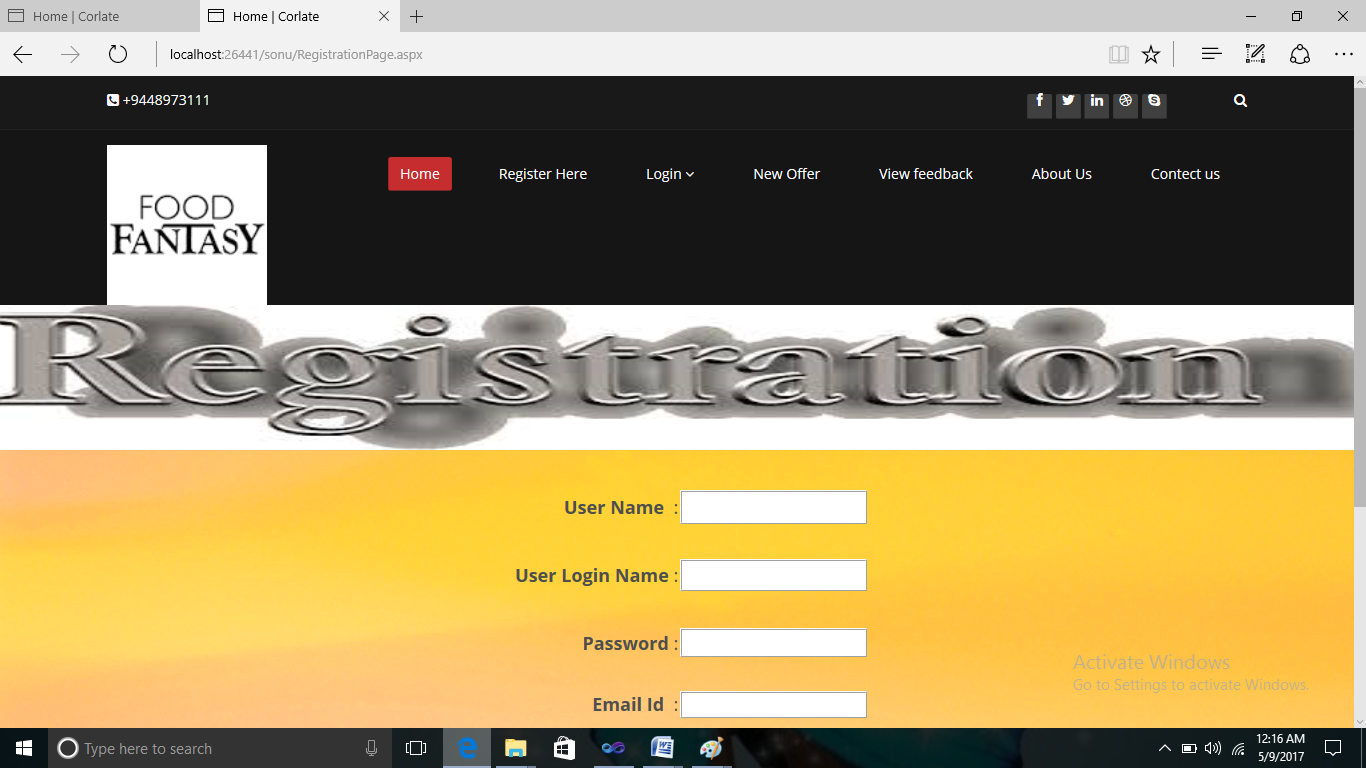
|  |  |
| --- | --- |
| ER Model | -Entity Relationship Model. |
| DFD  API  MFC  MDI  COM  SQL  DTS  CLR  CTS  CLS  MSIL  RAD  SDK  UI | -Data Flow Diagram.  -Application Programming Interface.  -Microsoft Foundation Classes.  -Multi Documents Interface.  -Component Object Model.  -Structured Query Language.  -Data Transformation Services.  -Common Language Runtime.  -Common Types System.  -Common Language Specification.  -Microsoft Intermediate Language.  -Rapid Application Development.  -Software Development Kit.  -User Interface. |

### User Manuals

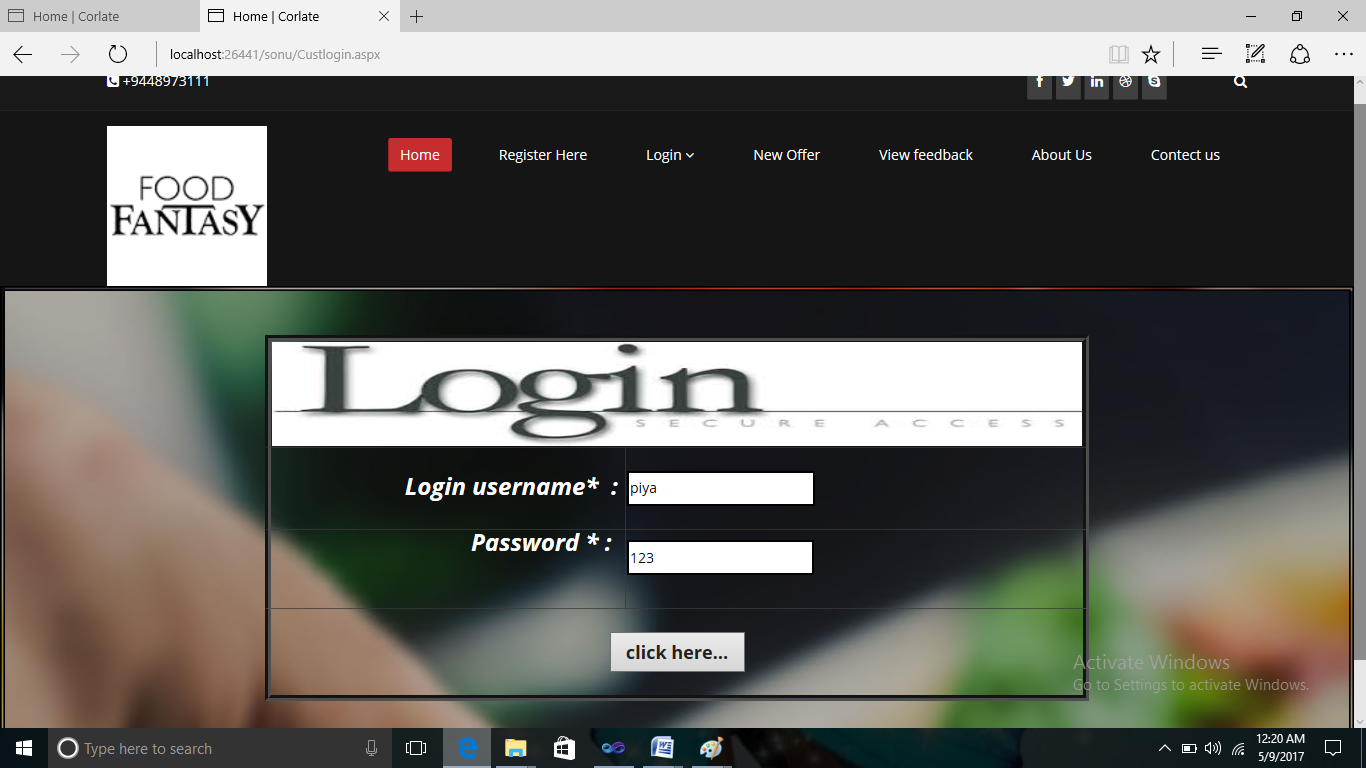
**Home Page:**

****

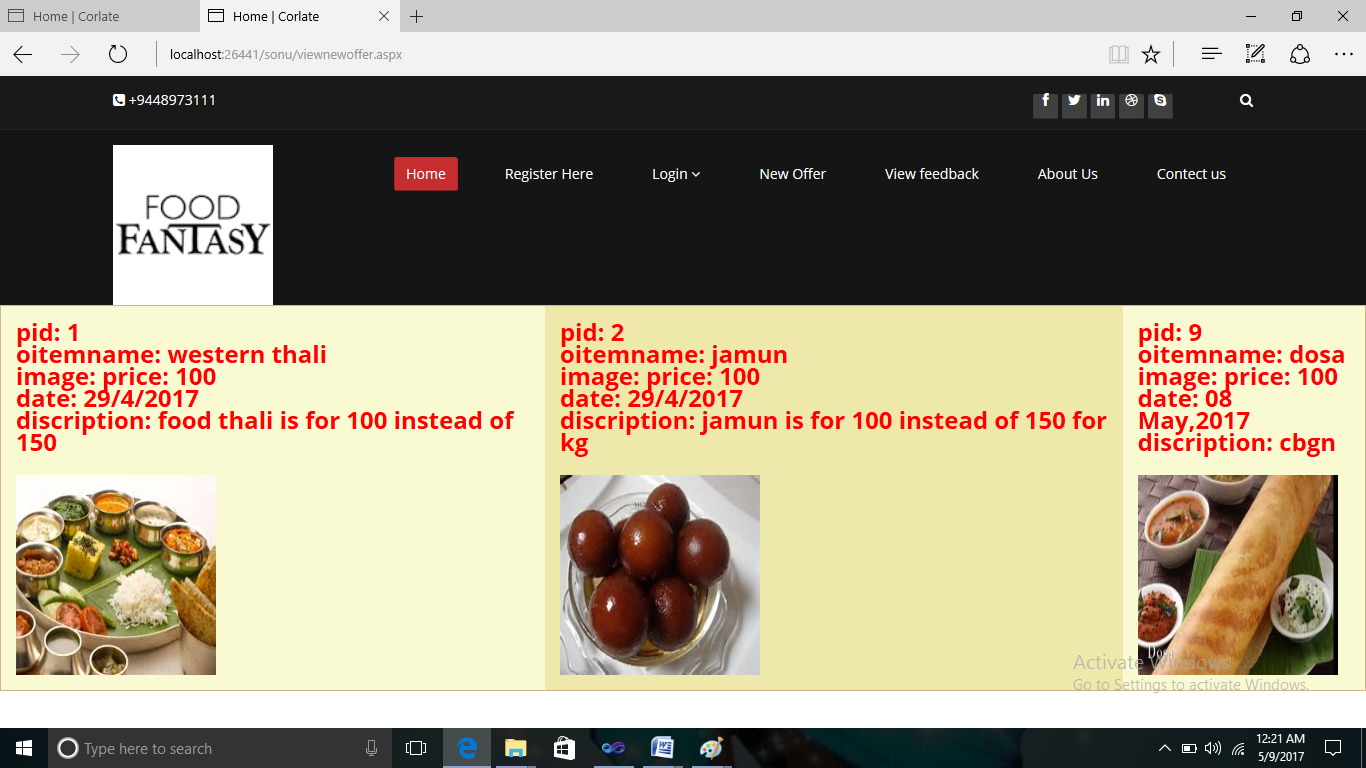
**Registretion page for customer:**

****

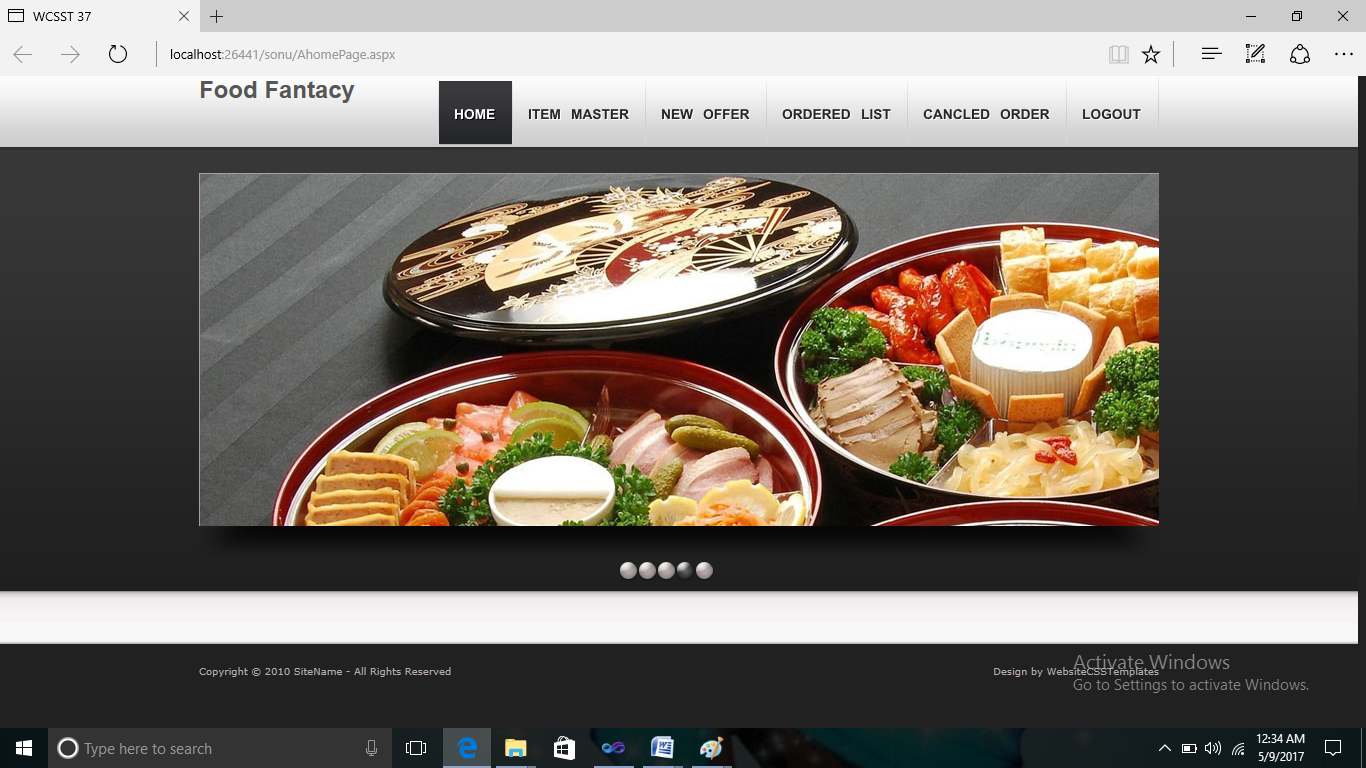
**Customer Login:**

****

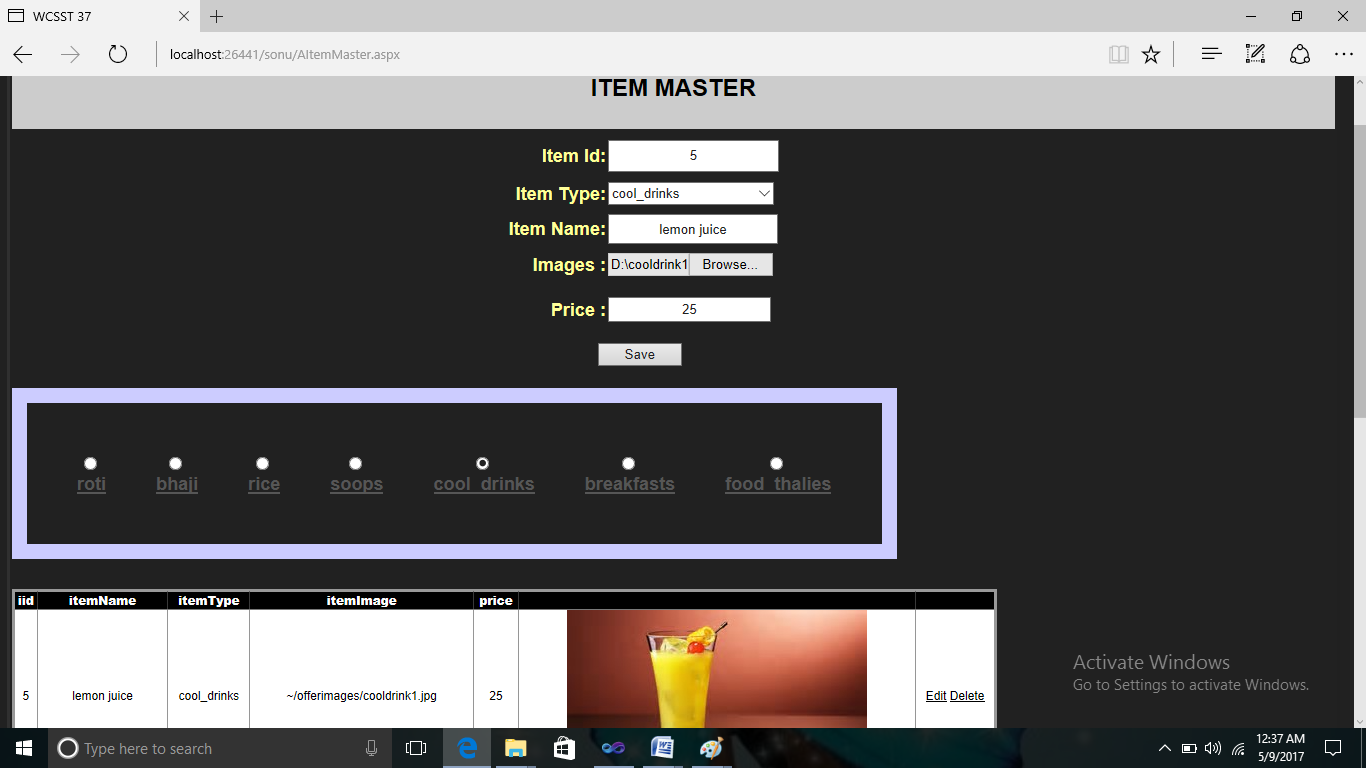
**New Offer view Page:**

****

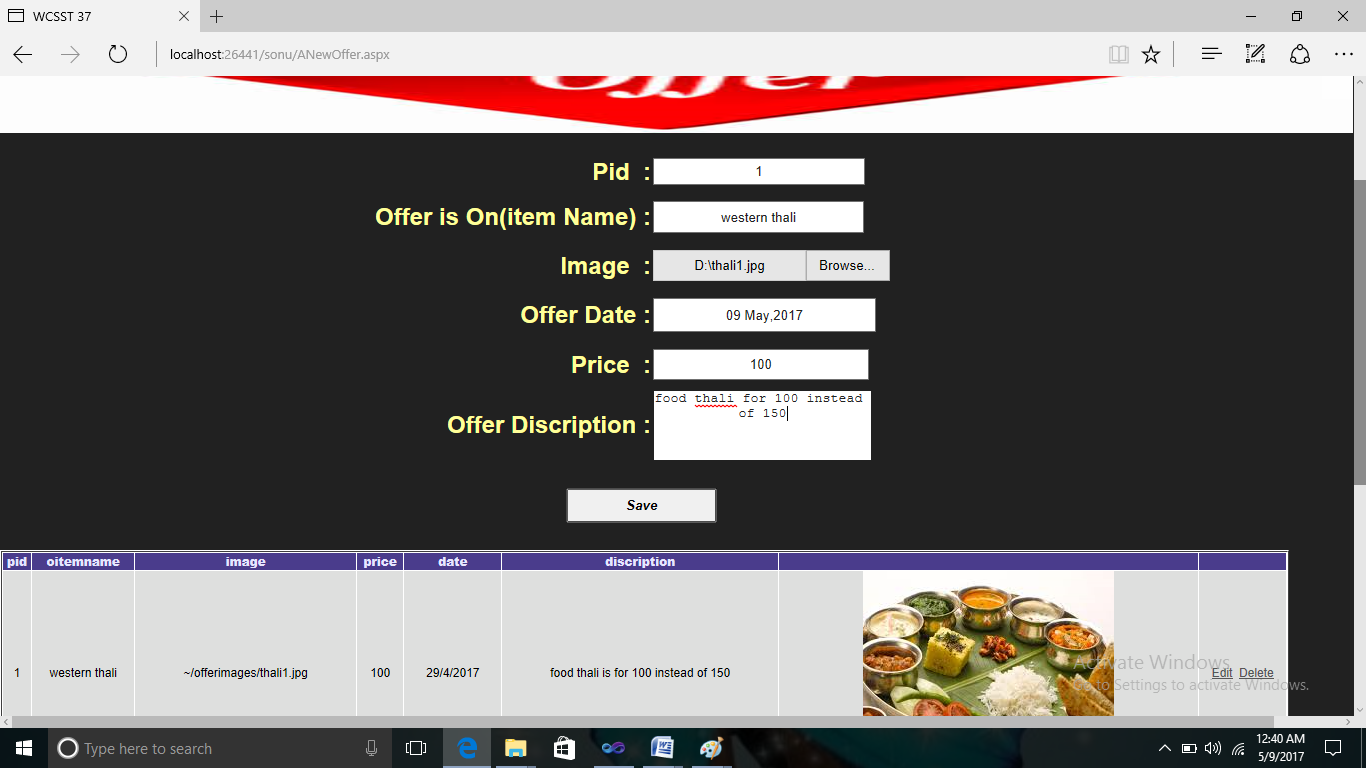
**Admin Home Page:**

****

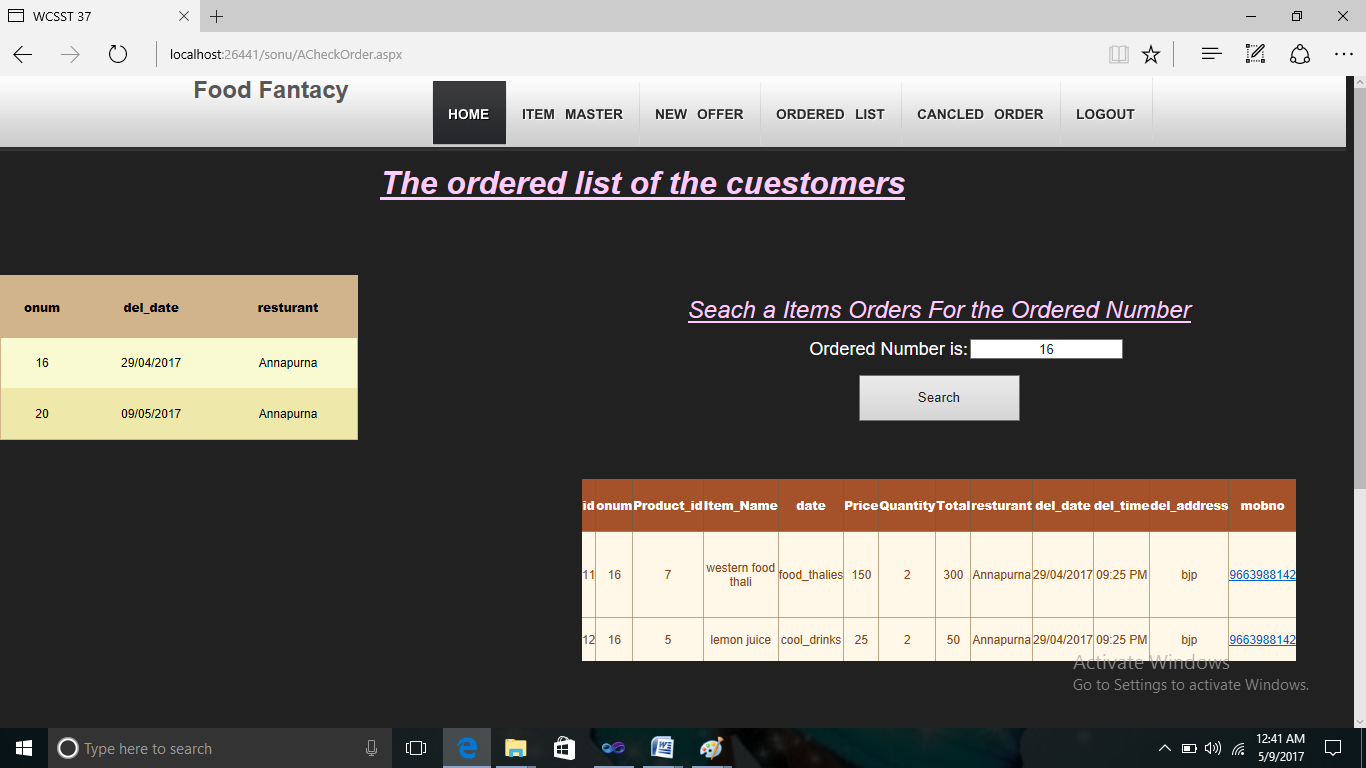
**Admin ItemMaster Page:**

****

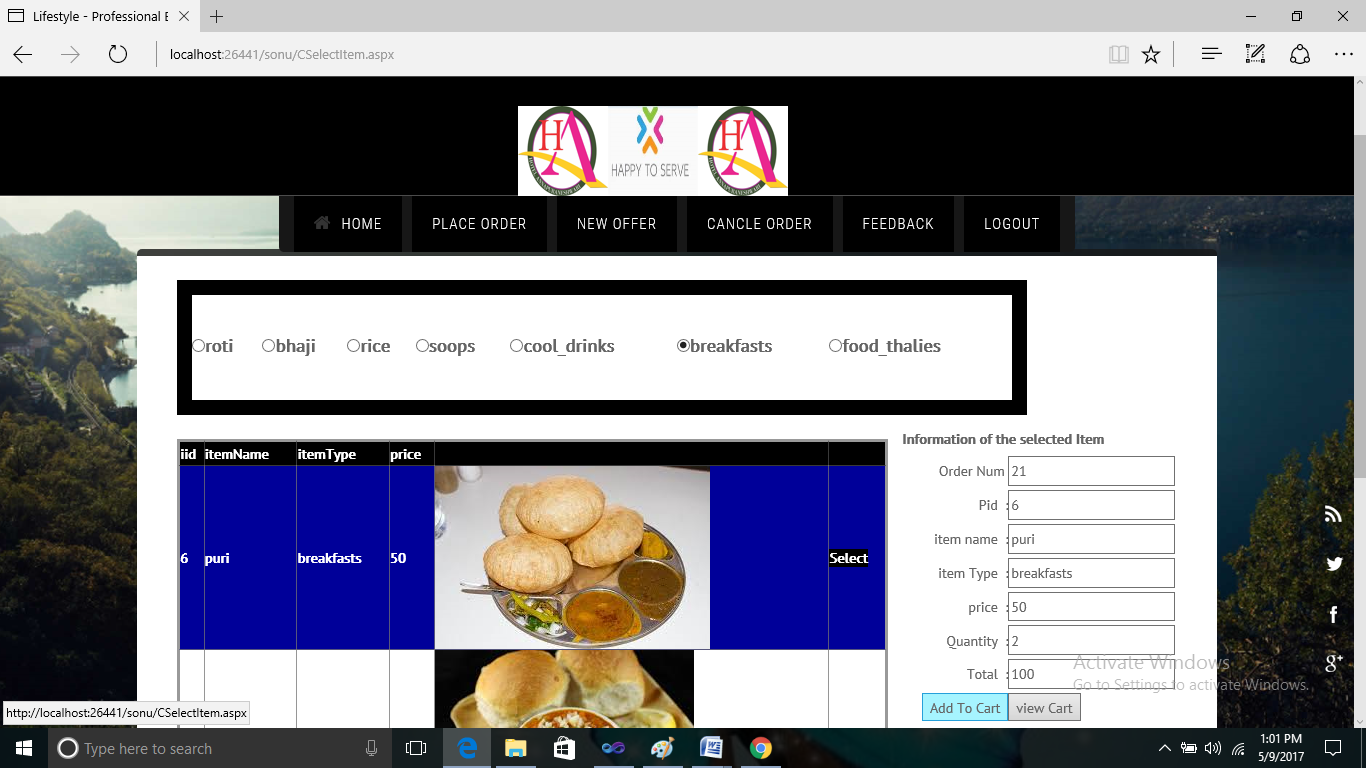
**Admin NewOffer Page:**

****

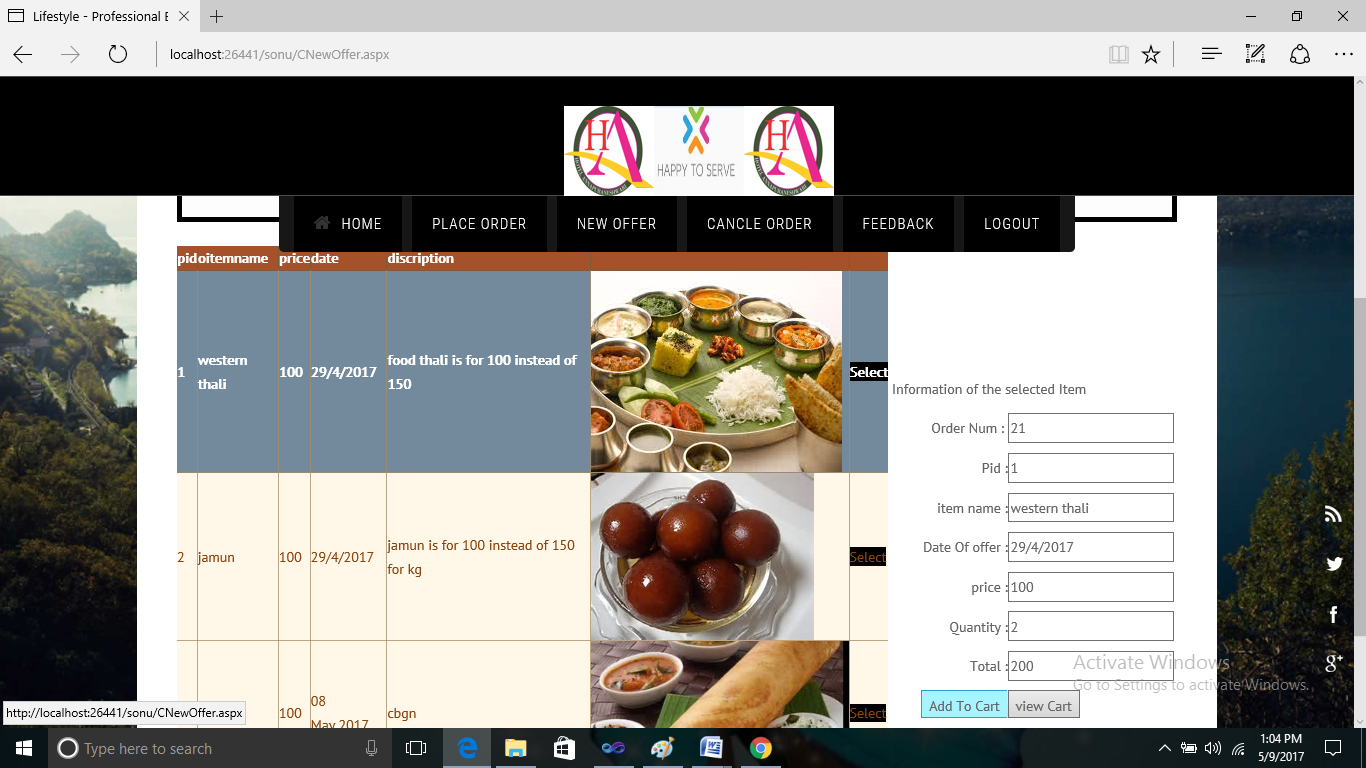
**Admin Viewing Ordered List Of Customer:**

****

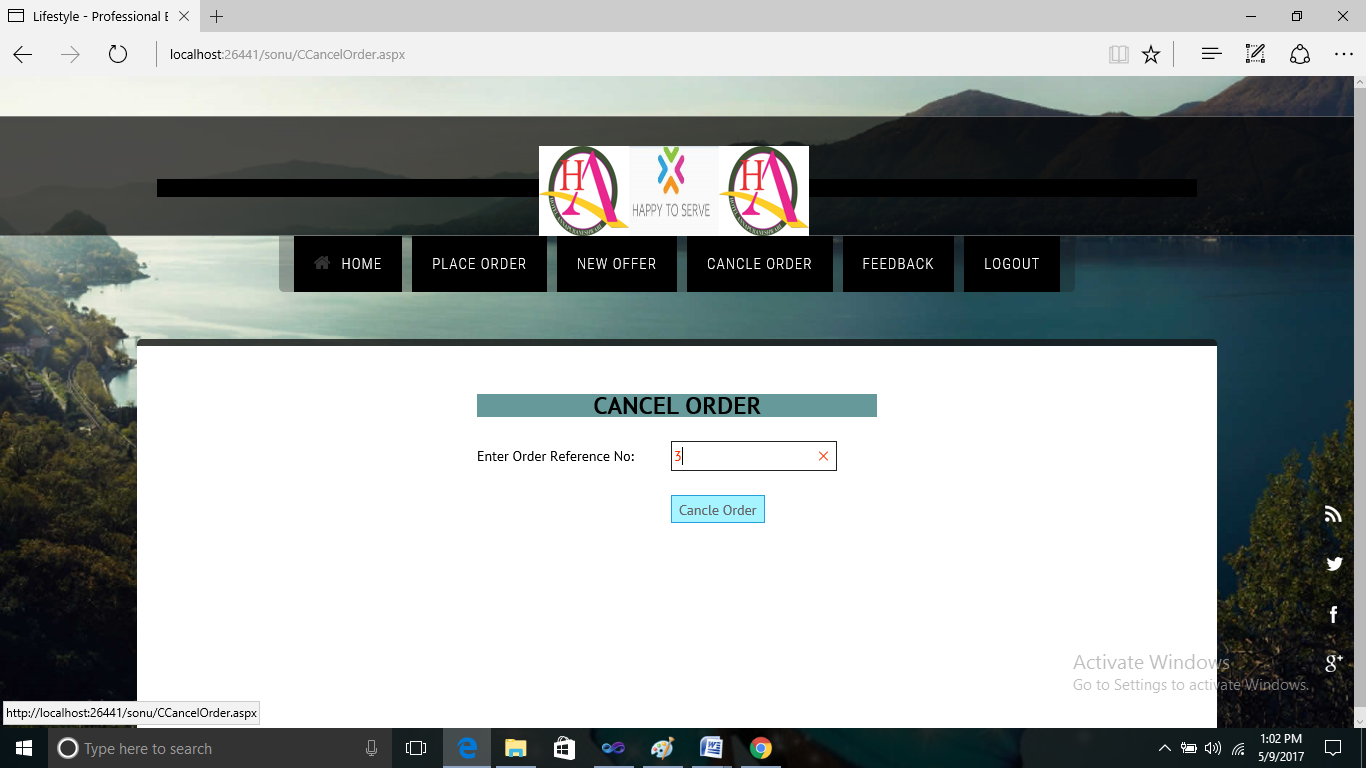
**PlaceOrder Page of Customer:**

****

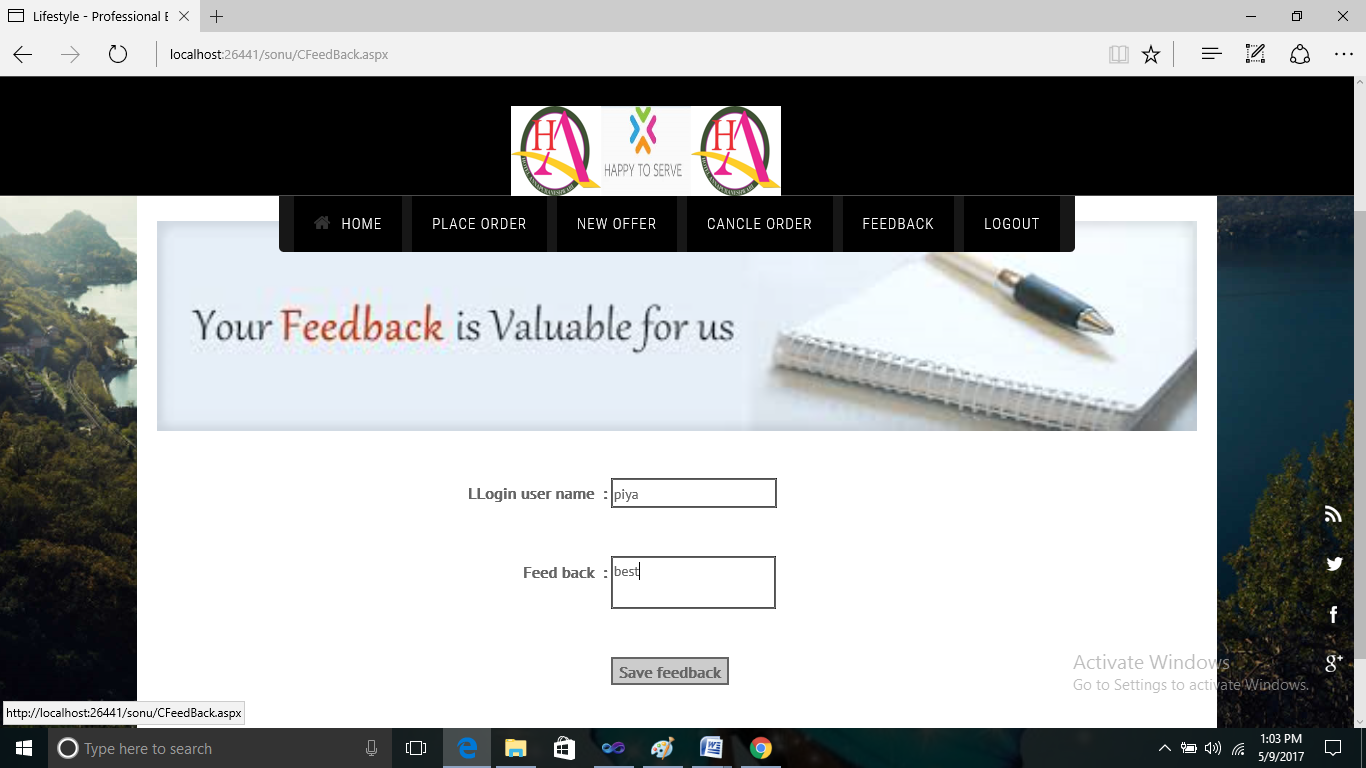
**New Offer Page Of Customer:**

****

**Cancle Order Page Of Customer:**

****

**FeedBack page Of Customer:**

****

**Coding:**

**PlaceOrder code:**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

using System.Net;

using System.Text;

using System.IO;

public partial class CCartPage : System.Web.UI.Page

{

string ord;

int order1;

foodfantacy obj = new foodfantacy();

string smsURL;

protected void Page\_Load(object sender, EventArgs e)

{

txttime.Text = DateTime.Now.ToString("hh:mm:ss tt");

txtdate.Text = DateTime.Now.ToString("dd MMMM,yyyy");

ord = Session["ord\_no"].ToString();

txt\_ord\_no.Text = ord;

order1 = Convert.ToInt32(ord);

DateTime today = DateTime.Today;

TimeSpan current\_time = DateTime.Now.TimeOfDay;

DateTime time = DateTime.Today.Add(current\_time);

string displayTime = time.ToString("hh:mm tt");

txtdate.Text = today.ToString("dd/MM/yyyy");

txttime.Text = displayTime;

BindGrid();

}

protected void ImageButton1\_Click(object sender, ImageClickEventArgs e)

{

Response.Redirect("CPlaceOrder.aspx");

}

protected void GridView1\_RowDeleting(object sender, GridViewDeleteEventArgs e)

{

int index = Convert.ToInt32(e.RowIndex);

DataTable dt = Session["cart"] as DataTable;

dt.Rows[index].Delete();

Session["dt"] = dt;

BindGrid();

}

protected void BindGrid()

{

DataTable dt = Session["cart"] as DataTable;

GridView1.DataSource = dt;

GridView1.DataBind();

int total = dt.AsEnumerable().Sum(row => row.Field<int>(dt.Columns["Total"]));

GridView1.FooterRow.Cells[5].Text = "Total";

GridView1.FooterRow.Cells[5].HorizontalAlign = HorizontalAlign.Right;

GridView1.FooterRow.Cells[6].Text = total.ToString("N2");

}

protected void GridView1\_PageIndexChanging(object sender, GridViewPageEventArgs e)

{

GridView1.PageIndex = e.NewPageIndex;

this.BindGrid();

}

protected void Button1\_Click(object sender, EventArgs e)

{

foreach (GridViewRow g1 in GridView1.Rows)

{

string prod = g1.Cells[1].Text;

int prod\_id = Convert.ToInt32(prod);

string pr = g1.Cells[4].Text;

int price = Convert.ToInt32(pr);

string qty = g1.Cells[5].Text;

int quantity = Convert.ToInt32(qty);

string tot = g1.Cells[6].Text;

int total = Convert.ToInt32(tot);

DataTable dt1 = new DataTable();

string querry2 = "select \* from order\_table where del\_date='" + txtdate.Text + "' and Product\_id='" + prod\_id + "' and Item\_Name='" + g1.Cells[2].Text + "' and Price='" + price + "' and Quantity='" + quantity + "' and Total='" + total + "'";

dt1 = obj.FoodS(querry2);

if (dt1.Rows.Count > 0)

{

ClientScript.RegisterStartupScript(GetType(), "alert", "alert('Order Exists ');", true);

txt\_ord\_no.Text = "";

DropDownList1.SelectedIndex = -1;

txtdate.Text = "";

txttime.Text = "";

txtaddress.Text = "";

return;

}

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\FoodFanctsy.mdf;Integrated Security=True;User Instance=True");

SqlCommand com = new SqlCommand("insert into order\_table(onum,Product\_id,Item\_Name,date,Price,Quantity,Total,resturant,del\_date,del\_time,del\_address,mobno) values (" + order1 + ",'" + prod\_id + "','" + g1.Cells[2].Text + "','" + g1.Cells[3].Text + "','" + price + "','" + quantity + "','" + total + "','" + DropDownList1.SelectedItem.Text + "','" + txtdate.Text + "','" + txttime.Text + "','" + txtaddress.Text + "','" + TextBox1.Text + "')", con);

con.Open();

com.ExecuteNonQuery();

con.Close();

string save = "insert into temp\_table(onum,Product\_id,Item\_Name,date,Price,Quantity,Total,resturant,del\_date,del\_time,del\_address) values (" + order1 + ",'" + prod\_id + "','" + g1.Cells[2].Text + "','" + g1.Cells[3].Text + "','" + price + "','" + quantity + "','" + total + "','" + DropDownList1.SelectedItem.Text + "','" + txtdate.Text + "','" + txttime.Text + "','" + txtaddress.Text + "')";

foodfantacy ob = new foodfantacy();

ob.FoodI(save);

}

SqlConnection con1 = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\FoodFanctsy.mdf;Integrated Security=True;User Instance=True");

SqlCommand com1 = new SqlCommand("insert into ordernum(onum) values (" + order1 + ")", con1);

con1.Open();

com1.ExecuteNonQuery();

con1.Close();

txt\_ord\_no.Text = "";

DropDownList1.SelectedIndex = -1;

txtdate.Text = "";

txttime.Text = "";

txtaddress.Text = "";

TextBox1.Text = "";

GridView1.Visible = false;

ClientScript.RegisterStartupScript(GetType(), "alert", "alert('Order Placed Sucessfully');", true);

sendsms();

}

public void sendsms()

{

string added\_id = "";

string admid = "";

string msg = "YOUR ORDER Number is" + order1 + "" + "Thanks For Ordering the Food ";

if (msg != "")

{

string mobile = "";

string stud\_mobile = "";

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\FoodFanctsy.mdf;Integrated Security=True;User Instance=True");

con.Open();

string q;

q = "select distinct(onum),mobno from order\_table where onum='"+order1+"'";

SqlCommand cmd = new SqlCommand(q, con);

SqlDataReader dr;

dr = cmd.ExecuteReader();

while (dr.Read() == true)

{

stud\_mobile = dr["mobno"].ToString();

if (stud\_mobile != "")

{

smsURL = sms(stud\_mobile, msg);

WebRequest request = null;

HttpWebResponse response = null;

try

{

smsURL = sms(stud\_mobile, msg);

request = WebRequest.Create(smsURL);

response = (HttpWebResponse)request.GetResponse();

Stream stream = response.GetResponseStream();

Encoding ec = System.Text.Encoding.GetEncoding("utf-8");

string content;

using (StreamReader reader = new StreamReader(stream, ec))

{

content = reader.ReadToEnd();

}

stream.Close();

}

catch (Exception exp)

{

}

finally

{

if (response != null)

response.Close();

}

}

}

dr.Close();

}

}

public static string sms(string mobile, string sms)

{

string url = "http://basava.siegsms.in/SendingSms.aspx?userid=basava&pass=basava@123&title=BETMBL&phone=" + mobile + "&msg=" + sms + "";

return url;

}

protected void Button2\_Click(object sender, EventArgs e)

{

Response.Redirect("CCancelOrder.aspx");

}

}

**AppData:**

using System;

using System.Data;

using System.Configuration;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

public class foodfantacy

{

public void FoodI(string InsertQuery)

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\FoodFanctsy.mdf;Integrated Security=True;User Instance=True");

con.Open();

SqlCommand cmd = new SqlCommand(InsertQuery, con);

cmd.ExecuteNonQuery();

con.Close();

}

public DataTable FoodS(string SelectQuery)

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\FoodFanctsy.mdf;Integrated Security=True;User Instance=True");

con.Open();

SqlCommand cmd = new SqlCommand(SelectQuery, con);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable ta = new DataTable();

da.Fill(ta);

con.Close();

return ta;

}

public

string

checkRole(string SelectQuery)

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\FoodFanctsy.mdf;Integrated Security=True;User Instance=True");

con.Open();

SqlDataReader dr;

SqlCommand cmd = new SqlCommand(SelectQuery, con);

dr = cmd.ExecuteReader();

string

role1 =

"";

while

(dr.Read())

{

role1 = dr[

"role"].ToString();

}

dr.Close();

con.Close();

return

role1;

}

public int getOdernum(string selectQuery)

{

SqlConnection con = new SqlConnection("Data Source=.\\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\FoodFanctsy.mdf;Integrated Security=True;User Instance=True");

con.Open();

SqlCommand cmd = new SqlCommand(selectQuery, con);

int ord = (int)cmd.ExecuteScalar();

con.Close();

return ord;

}

}

**Registration Code:**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

public partial class RegistrationPage : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

string selectquery = "select \* from Reg\_Table where UloginName='" + txtulname.Text + "'";

foodfantacy obj = new foodfantacy();

DataTable ta1 = obj.FoodS(selectquery);

if (ta1.Rows.Count >= 1)

{

string msg1 = "<script>alert('User Already Exist!,');</script>";

ScriptManager.RegisterStartupScript(this, typeof(Control), "alertmsg", msg1, false);

}

else

{

string save = "insert into Reg\_Table(uname,uloginname,password,location,mobno,email)values('" + txtrname.Text + "','" + txtulname.Text + "','" + txtpass.Text + "','" + txtloca.Text + "','" + txtmobno.Text + "','" + txtemail.Text + "')";

foodfantacy ob = new foodfantacy();

ob.FoodI(save);

ClientScript.RegisterStartupScript(GetType(), "alert", "alert('Registred Sucessfully');", true);

txtulname.Text = "";

txtulname.Text="";

txtpass.Text="";

txtloca.Text="";

txtmobno.Text="";

txtemail.Text = "";

}

}

}

**Reference Books and sites**

**Book:**

* C# and the .NET Platform Second Edition 2005 by Andrew Troelsen.

**Sites:**

* [www.Knowdotnet.com](http://www.Knowdotnet.com)
* [www.triconsole.com](http://www.triconsole.com)
* [www.stackoverflow.com](http://www.stackoverflow.com)
* [www.dotnetspider.com](http://www.dotnetspider.com)