*A DBMS Mini Project on*

**ONLINE GROCERY SHOP**

*Submitted by*

**Name\_of\_student Roll\_no**

**Anjan Kumar 14CO104**

**Deepak KV 14CO116**

**Pashupati Nath Verma 14CO130**

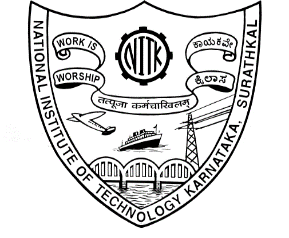
**V Sem B.Tech**

*in partial fulfillment for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

*in*

**COMPUTER SCIENCE AND ENGINEERING**



**Department of Computer Science and Engineering**

**National Institute of Technology Karnataka, Surathkal**.

***November 2016***

**ONLINE GROCERY SHOP**

**ABSTRACT**

The business-to-consumer aspect of electronic commerce (e-commerce) is the most visible business use of the World Wide Web. The primary goal of an e-commerce site is to sell goods and services online. This project deals with developing an e-commerce website for Online Grocery Shop. It provides the user with a catalog of different Groceries available for purchase in the store.

In order to facilitate online purchase a shopping cart is provided to the user. The system implemented using a 3-tier approach, with a backend database, a middle tier of Microsoft Internet Information Services (IIS) and ASP.NET, and a web browser as the front end client.

In order to develop an e-commerce website, a number of Technologies must be Studied and understood. These include multi-tiered architecture, server and client side scripting techniques, implementation technologies such as ASP.NET, programming language (such as C#, VB.NET), relational databases (such as MySQL, Access). This is a project with the objective to develop a basic website where a consumer is provided with a shopping cart application and also to know about the technologies used to develop such an application.

This document will discuss each of the underlying technologies to create and implement an e-commerce website. To implement this as a web application we used ASP.NET as the Technology. ASP.NET has several advantages such as enhanced performance, scalability, built-in security and simplicity. To build any web application using ASP.NET we need a programming language such as C#, VB.NET, J# and so on. C# was the language used to build this application. For the client browser to connect to the ASP.NET engine we used Microsoft’s Internet Information Services (IIS) as the Web Server. ASP.NET uses ADO.NET to interact with the database as it provides in-memory caching that eliminates the need to contact the database server frequently and it can easily deploy and maintain an ASP.NET application. MySQL was used as back-end database since it is one of the most popular open source databases, and it provides fast data access, easy installation and simplicity.

**INDEX**

1. INTRODUCTION……………………………………………… 5
2. PROJECT DESIGN…………………………………………….. 5
3. USER INTERFACE DESIGN………………………………….. 9
4. IMPLEMENTATION TECHNOLOGIES……………………… 11
5. THE SHOPPING CART APPLICATION……………………... 17
6. CONCLUSION…………………………………………………. 20
7. REFERENCES………………………………………………….. 21

**1. Introduction**

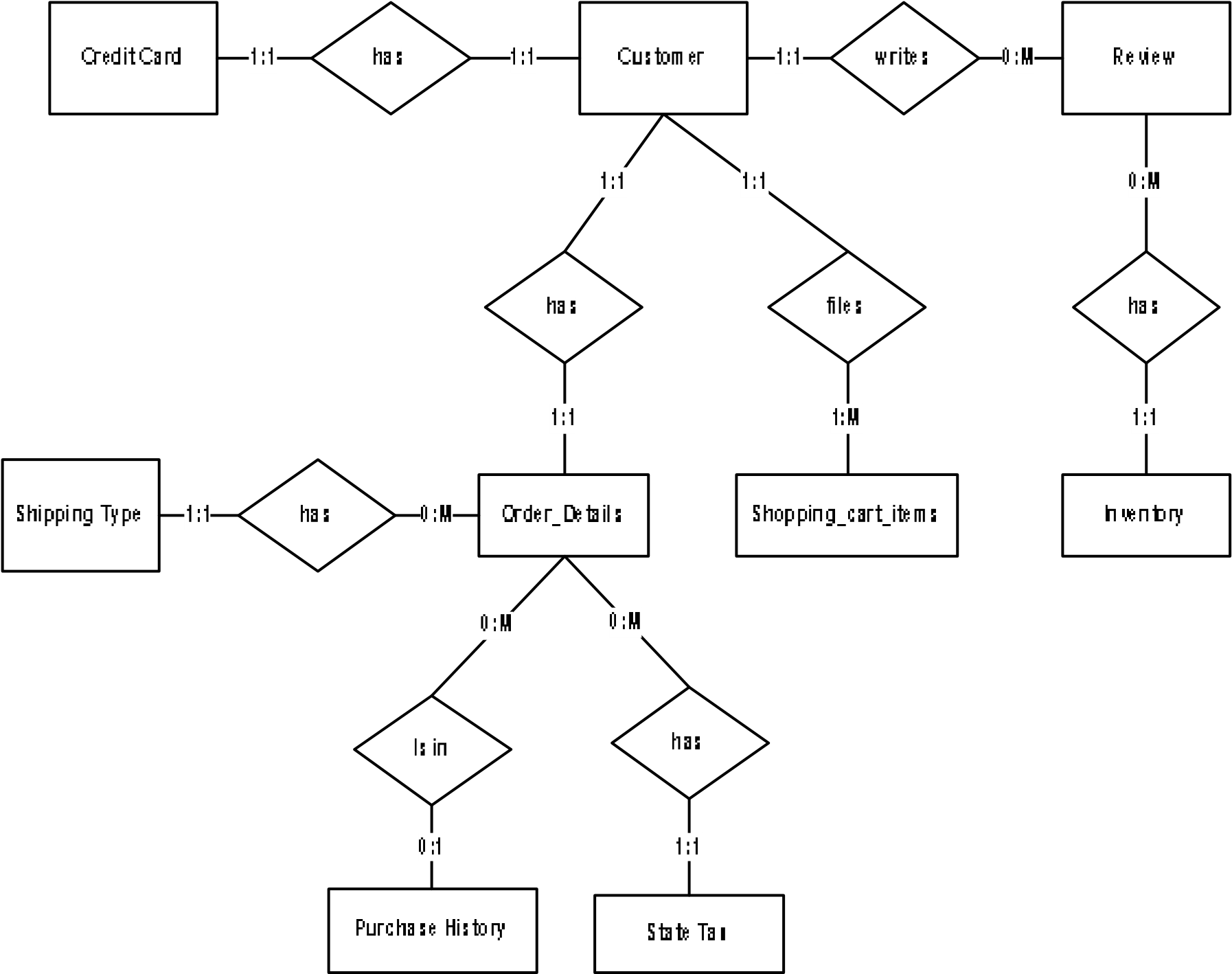
E-commerce is fast gaining ground as an accepted and used business paradigm.More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace. The objective of this project is to develop a general purpose e-commerce store.Where any product (such as fruits, vegetables, grocery stables etc) can be bought from the comfort of home through the Internet. However, for implementation purposes, this document will deal with an online Grocery store. An online store is a virtual store on the Internet where customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as credit card number. An e- mail notification is sent to the customer as soon as the order is placed.

**2. Project Design**

In order to design a web site, the relational database must be designed first. Conceptual design can be divided into two parts: The **data model** and the **process** **Model**. The data model focuses on what data should be stored in the database while the process model deals with how the data is processed. To put this in the context of the relational database, the data model is used to design the relational tables. The process model is used to design the queries that will access and perform operations on those tables.

**2.1 Data Model**

A data model is a conceptual representation of the data structures that are required by a database. The first step in designing a database is to develop an Entity-Relation Diagram (ERD). The ERD serves as a blue print from which a relational database maybe deduced. Figure 1 shows the ERD for the project and later we will show the transformation from ERD to the Relational model.



**2.2Database Design**

In this section, the basic structure of the tables composing the database for the project are shown along with information about primary and foreign keys.

**A) Customer**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | UserID | Varchar | Primary key for Customer identification |
| 2 | Password | Varchar | Security for Customer |
| 3 | First\_Name | Varchar |  |
| 4 | Last\_Name | Varchar |  |
| 5 | Address | Varchar |  |
| 6 | City | Varchar |  |
| 7 | Zip | Integer |  |
| 8 | State | Varchar |  |
| 9 | Email Address | Varchar |  |
| 10 | Phone\_Number | Varchar |  |

**B) Order Details**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | OrderID | Integer | Primary key for Order identification |
| 1 | UserID | Char | Foreign key to Customer |
| 2 | Receiver’s Name | Char | If order is to be sent to other address rather than to the customer, we need that address |
| 3 | Address | Char |  |
| 4 | City | Char |  |
| 5 | Zip | Integer |  |
| 6 | State | Char | Foreign key to State Tax |
| 7 | Type of Shipping | Char | Foreign key to Shipping Type |
| 8 | Date of Purchase | Date |  |

**C) Shopping\_cart\_items**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | ShoppingCartID | Integer | Primary key for Shopping Cart  Identification |
| 2 | InventoryID | Varchar | Foreign key to Inventory |
| 3 | Price | Double |  |
| 4 | Date | Date |  |
| 5 | UserID | Varchar | Foreign key to Customer |
| 6 | Quantity | Integer |  |

**D) Credit\_Card\_Details**

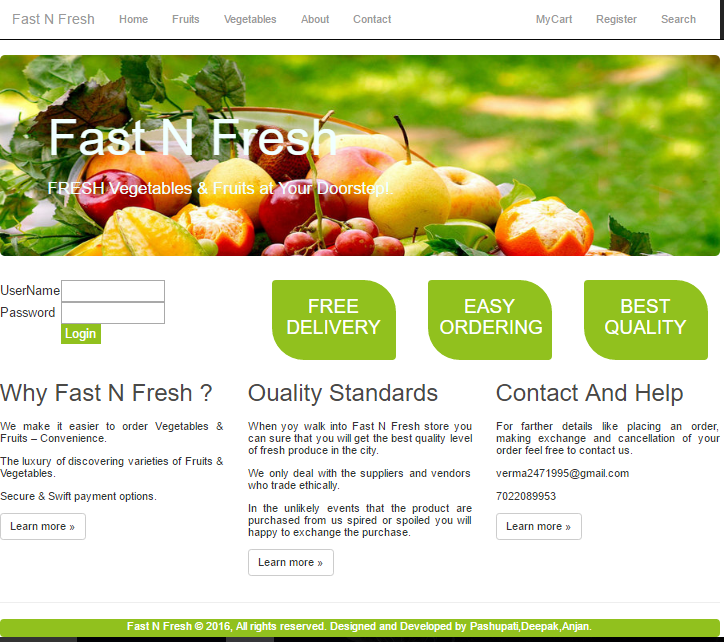
|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | Credit Username | Varchar | Primary key for Customer Identification |
| 2 | Credit Card Number | Varchar |  |
| 3 | Card Type | Varchar | Master Card, Visa, Discover |
| 4 | CVV Number | Integer | Number present on the back of the card for extra security |
| 5 | Expiry Date | Date |  |
| 6 | UserID | Varchar | Foreign key to Customer |

**E) Purchase\_History**

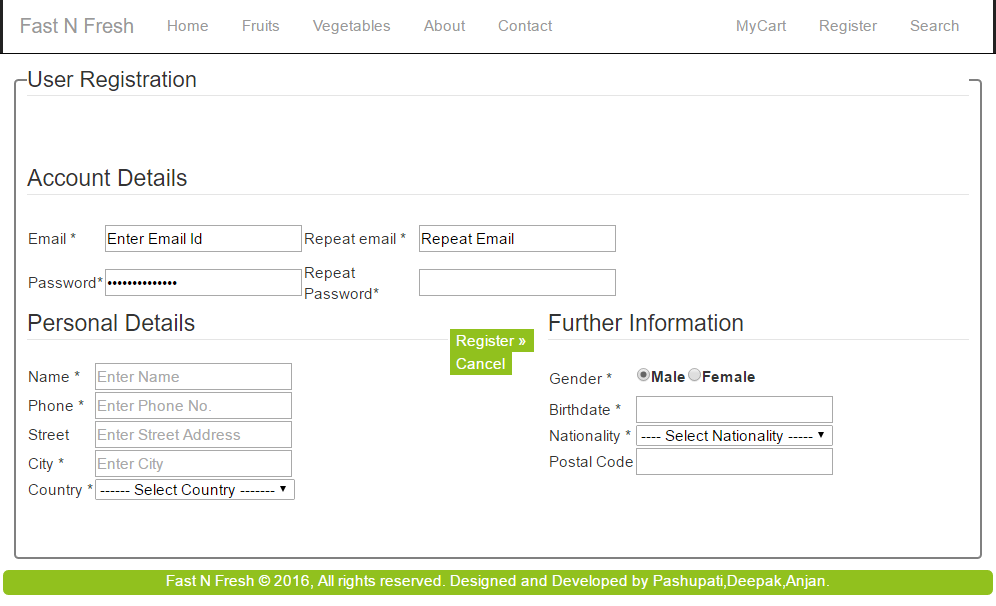
|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **NAME** | **TYPE** | **DESCRIPTION** |
| 1 | UserID | Varchar | Primary key for Customer Identification |
| 2 | InventoryID | Varchar | Book purchased by the user |
| 3 | Date of Purchase | Date |  |
| 4 | OrderID | Integer | Foreign key to Order\_details |
| 5 | Quantity | Integer |  |
| 6 | Price | Double |  |

**3. USER INTERFACE DESIGN**

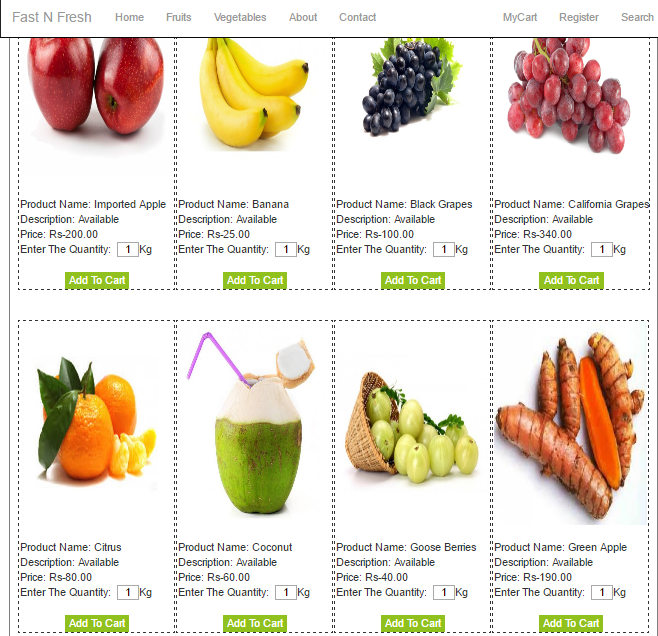
**1) Home Page**

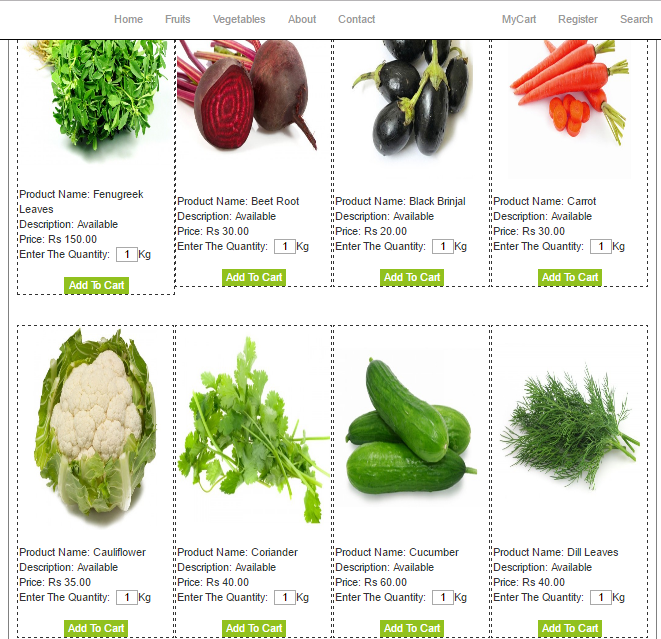


**2)** **Registration of the new user**

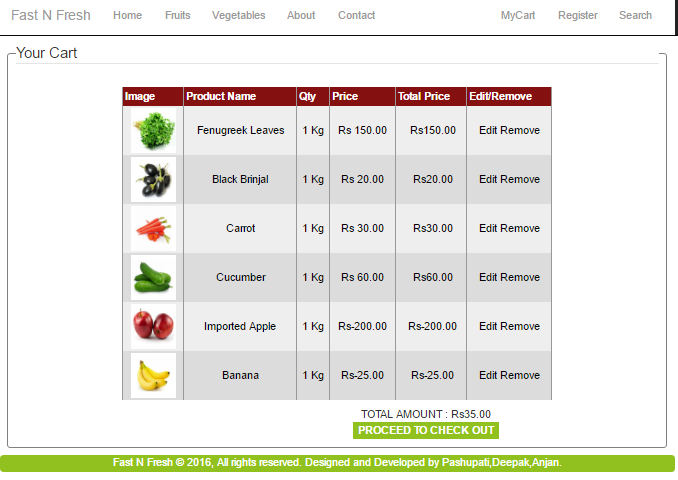


3) Fruits and Vegetables Page





4) Shopping Cart for the user



**4. Implementation Technologies**

The objective of this project is to develop an online Grocery store. When the user types in the URL of the Grocery Store in the address field of the browser, a Web Server is contacted to get the requested information. In the .NET Framework, IIS (Internet Information Service) acts as the Web Server. The sole task of a Web Server is to accept incoming HTTP requests and to return the requested resource in an HTTP response. The first thing IIS does when a request comes in is to decide how to handle the request. Its decision is based upon the requested file's extension. For example, if the requested file has the .asp extension, IIS will route the request to be handled by asp.dll. If it has the extension of .aspx, .ascx, etc, it will route the request to be handled by ASP.NET Engine.

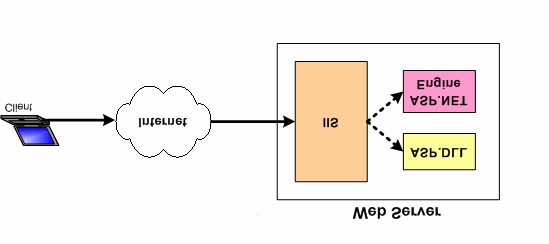


Figure 4.1 Relation between IIS and ASP.NET

The ASP.NET Engine then gets the requested file, and if necessary contacts the database through ADO.NET for the required file and then the information is sent back to the Client’s browser. Figure 4.1 shows how a client browser interacts with the Web server and how the Web server handles the request from client.There are two separate authentication layers in an ASP.NET application. All requests flow through IIS before they are handed to ASP.NET, and IIS can decide to deny access before ASP.NET even knows about the request. Here is how the process works :

1. IIS checks to see if an incoming request is coming from an IP address that is allowed access to the domain. If not, the request is denied.
2. IIS performs its own user authentication, if it is configured to do so. By default, IIS allows anonymous access and requests are authenticated automatically.
3. When a request is passed from IIS to ASP.NET with an authenticated user, ASP.NET checks to see whether impersonation is enabled. If so, ASP.NET acts as though it were the authenticated user. If not, ASP.NET acts with its own configured account.
4. Finally, the identity is used to request resources from the operating system. If all the necessary resources can be obtained, the user's request is granted; otherwise the request is denied.

**4.1 MySQL Database**

In this project, MySQL is used as the backend database. MySQL is an open source database management system. The features of MySQL are given below:

1. MySQL is a relational database management system. A relational database stores information in different tables, rather than in one giant table. These tables can be referenced to each other, to access and maintain data easily.

2. MySQL is open source database system. The database software can be used and modify by anyone according to their needs.

3.It is fast, reliable and easy to use. To improve the performance, MySQL is multithreaded database engine. A multithreaded application performs many tasks at the same time as if multiple instances of that application were running simultaneously.

In being multithreaded MySQL has many advantages. A separate thread handles each incoming connection with an extra thread that is always running to manage the connections. Multiple clients can perform read operations simultaneously, but while writing, only hold up another client that needs access to the data being updated. Even though the threads share the same process space, they execute individually and because of this separation, multiprocessor machines can spread the thread across many CPUs as long as the host operating system supports multiple CPUs. Multithreading is the key feature to support MySQL’s performance design goals. It is the core feature around which MySQL is built. MySQL database is connected to ASP.NET using an ODBC driver. Open

Database Connectivity (ODBC) is a widely accepted application-programming interface (API) for database access. The ODBC driver is a library that implements the functions supported by ODBC API. It processes ODBC function calls, submits SQL requests to MySQL server, and returns results back to the application. If necessary, the driver modifies an application's request so that the request conforms to syntax supported by MySQL.

**4.2 Web Page Programming Options**

An e-commerce organization can create data-based Web pages by using server-side and client-side processing technologies or a hybrid of the two. With server-side processing, the Web server receives the dynamic Web page request, performs all processing necessary to create the page, and then sends it to the client for display in the client’s browser. Client-side processing is done on the client workstation by having the client browser execute a program that interacts directly with the database.

**4.2.1 Server-side processing**

Generally dynamic or data-driven Web pages use HTML forms to collect user inputs, submitting them to a Web server. A program running on the server processes the form inputs, dynamically composing a Web page reply. This program, which is called, servicing program, can be either a compiled executable program or a script interpreted into machine language each time it is run.

When a user submits HTML-form data for processing by a compiled server program, the Web Server invokes the servicing program. The servicing program is not part of the Web server but it is an independent executable program running on the Web server; it processes the user input, determines the action which must be taken, interacts with any external sources (Eg: database) and finally produces an HTML document and terminates. The Web server then sends the HTML document back to the user’s browser where it is displayed.

**4.2.2 Client-Side Processing**

Client-side web page processing is achievable through compiled programs downloaded, installed, and executed on the client workstation or by creating scripts with the HTML Web page commands interpreted by the client browser.

Although basic client-side scripts cannot be used by a Web page to interact with a remote database, they are often used to validate user inputs entered on HTML forms submitted for processing by a server-side program; for example, a script running on a client workstation might check the inputs users submit to a Web page to make sure they entered all required data and appropriate data values. This approach avoids transmitting inputs to the Web server that are incomplete or include errors, while offloading error checking and handling from the Web server program to the client workstation.

Client-side scripts can also be used to create advanced Web page features, including: animations, calculations, playing sound and video, and image maps allowing users to move their cursors over an image and click to access different Web page links. JavaScript is the most commonly used client-side scripting language.

**4.3 Database Connectivity**

In e-commerce applications it is very typical for the Web server to contact the database to get information as needed. ASP.NET uses a technology called ActiveX Data Objects.NET (ADO.NET) to connect to the database. .

In ADO.NET there are two core objects that allow us to work with data initially: the Data reader and the Dataset. In any .NET data access page, before we connect to a database, we first have to import all the necessary namespaces that will allow us to work with the objects required. Namespace in .NET is a set of classes that can be used while creating an application. The .NET Framework has about 3,500 classes which can be accessed through a namespace. The application will be using a technology known as Open Database Connectivity (ODBC) to access the database; therefore we must first import necessary namespaces. Below is a sample namespace declaration used by .NET.

<%@ Import Namespace="System" %>

<%@ Import Namespace="System.Data" %>

<%@ Import Namespace="System.Data.Odbc" %>

After all the necessary namespaces are imported, a connection to the database is made.

OdbcConnetion odbcCon = new OdbcConnection ("DRIVER = {MySQL ODBC 3.51

Driver}; SERVER=localhost; DATABASE=project; UID=root; PASSWORD=pwd");

The above statement creates a connection to the database with an

Odbc Connection object. This object tells ASP.NET where to go to get the data it needs. Since the data is stored in the same computer as the application, the SERVER is given as *localhost.*

**4.3.1 Connecting ASP.NET to the database.**

The steps required to connect our ASP.NET application to the MySQL database and access the data are given below:

1. Import the required namespaces.

using System;

using System.Data;

using System.Data.Odbc;

1. Create a connection object.

string myConnectionString;

myConnectionString = “DRIVER = {MySQL ODBC 3.51 Driver}; SERVER = localhost; DATABASE = project; UID = root; PASSWORD = ‘’ ”

OdbcConnection odbcCon = new odbcConnection(myConnectionString)

1. Create a SQL query

string str;

str="Select \* from Customer where UserID='admin’;

1. Create a Command object to run the SQL query

odbcCmd=new OdbcCommand(str,odbcCon);

1. DataReader to read the result

OdbcDataReader odbcreader;

String text,text2;

while(odbcReader.Read)

{ text = odbcReader["UserID"].ToString();

text2 = odbcReader[“FirstName”].ToString(); }

1. Close odbcReader and odbcConnection odbcReader.Close(); odbcCon.Close();
2. The data can now be used as desired by the application.

**5. The Shopping Cart Application**

The objective of this application is to provide the user an online website where they can buy Groceries from the comfort of their home. A shopping cart is used for the purpose. The user can select the desired Groceries, place them in the shopping cart and purchase them using a Credit Card. The user’s order will be shipped according to the type of shipping selected at the time of placing the order.

Website consists of the following web pages:

1. AddGrocery.aspx
2. GroceryDetails.aspx
3. GroceryReview.aspx
4. Grocerys.aspx
5. ChangePassword.aspx
6. CheckOut.aspx
7. FinalOrder.aspx
8. Footer.ascx
9. ForgotPassword.aspx
10. Login.aspx
11. LogOff.aspx
12. Menu.ascx
13. Order.aspx
14. PurchaseHistory.aspx
15. Registration.aspx
16. Search.aspx
17. ShoppingCart.aspx
18. UserDetails.aspx

As explained earlier, the user need not be logged in to add Groceries to the shopping cart. When the user adds Groceries without logging in, a GUID (Globally Unique Identifier) is obtained from the system and stored in Session ["login id”] variable. This GUID is stored in the shopping\_cart\_items table along with the selected Groceries by the user. If the user logs in, the GUID in the table is replaced with the actual User ID of the user. Session variables are used to transfer data from one page to another. As soon as the user closes the window, the session variables are cleared.

**5.1 Transactions in the Application**

A transaction is a group of database commands that are treated as a single unit. Transaction must pass what is known as the ACID test:

**Atomic:** All operations in the transaction are executed properly or none. In other words, they make up a single unit of work. For example, if a customer moves and a transaction is used to reflect that change in the database, all parts of the address (street, city, state, etc) must be changed as an atomic action, rather than changing street, then city, then state, and so on.

**Consistent**: The execution of a single transaction preserves the consistency of the database. All the relationships between data in a database are maintained correctly. For example, if customer information uses a tax rate from a state tax table, the state entered for the customer must exist in the state tax table.

**Isolation:** Each transaction is unaware of the other transactions occurring concurrently. Changes made by other clients cannot affect the current changes. For example, if two data entry operators try to make a change to the same customer at the same time, one of two things occurs: either one operator's changes are accepted and the other is notified that the changes were not made, or both operators are notified that their changes were not made. In either case, the customer data is not left in an indeterminate state.

**Durability:** Changes the transaction has performed persist in the database. Once a change is made, it is permanent. If a system error or power failure occurs before a set of commands is complete, those commands are undone and the data is restored to its original state once the system begins running again. Transaction processing is particularly important for Web applications that use data access, since Web applications are distributed among many different clients. In a Web application, databases are a shared resource, and having many different clients distributed over a wide area can present these key problems:

* Contention for resources. Several clients might try to change the same record at the same time. This problem gets worse the more clients you have.
* Unexpected failures. The Internet is not the most reliable network, even if your Web application and Web server are 100 percent reliable. Clients can be unexpectedly disconnected by their service providers, by their modems, or by power failures.
* Web application life cycle. Web applications do not follow the same life cycle as Windows applications—Web forms live for only an instant, and a client can leave your application at any point by simply typing a new address in their browser.

Transaction processing follows these steps:

1. Begin a transaction.
2. Process database commands.
3. Check for errors.
4. If errors occurred, restore the database to its state at the beginning of the transaction. If no errors occurred, commit the transaction to the database.

**6. Conclusion**

The Internet has become a major resource in modern business, thus electronic shopping has gained significance not only from the entrepreneur’s but also from the customer’s point of view. For the entrepreneur, electronic shopping generates new business opportunities and for the customer, it makes comparative shopping possible. As per a survey, most consumers of online stores are impulsive and usually make a decision to stay on a site within the first few seconds. “Website design is like a shop interior. If the shop looks poor or like hundreds of other shops the customer is most likely to skip to the other site. Hence we have designed the project to provide the user with easy navigation, retrieval of data and necessary feedback as much as possible.

In this project, the user is provided with an e-commerce web site that can be used to buy Groceries online. To implement this as a web application we used ASP.NET as the Technology. ASP.NET has several advantages such as enhanced performance, scalability, built-in security and simplicity. To build any web application using ASP.NET we need a programming language such as C#, VB.NET, J# and so on. C# was the language used to build this application. For the client browser to connect to the ASP.NET engine we used Microsoft’s Internet Information Services (IIS) as the Web Server. ASP.NET uses ADO.NET to interact with the database as it provides in-memory caching that eliminates the need to contact the database server frequently and it can easily deploy and maintain an ASP.NET application. MySQL was used as back-end database since it is one of the most popular open source databases, and it provides fast data access, easy installation and simplicity.

A good shopping cart design must be accompanied with user-friendly shopping cart application logic. It should be convenient for the customer to view the contents of their cart and to be able to remove or add items to their cart. The shopping cart application described in this project provides a number of features that are designed to make the customer more comfortable.

This project helps in understanding the creation of an interactive web page and the technologies used to implement it. The design of the project which includes Data Model and Process Model illustrates how the database is built with different tables, how the data is accessed and processed from the tables. The building of the project has given me a precise knowledge about how ASP.NET is used to develop a website, how it connects to the database to access the data and how the data and web pages are modified to provide the user with a shopping cart application.

**7. References**

1. http://encyclopedia.laborlawtalk.com/IIS for information on IIS
2. http://aspnet.4guysfromrolla.com/articles/020404-1.aspx for relationship between IIS and ASP.NET.
3. http://216.15.201.66/dpec/course.htm?fullpg=http%3A//216.15.201.66/dpec/cours es/wac312/wah006.htm&acro=wac312 for security authentication in ASP.NET
4. http://samples.gotdotnet.com/quickstart/aspplus/doc/mtstransactions.aspx for information on Transactions in ASP.NET.
5. http://www.x-cart.com/articles/design\_development.htmlfor online customer behavior.
6. http://aspnet.4guysfromrolla.com/articles/011404-1.aspx for relation between IIS and ASP.NET.
7. http://www.informatik.uni-bremen.de/uniform/gdpa\_d/methods/m-fctd.htm for definition of Functional Decomposition.
8. http://www.agilemodeling.com/artifacts/dataFlowDiagram.htm for definition of Data Flow Diagram.
9. http://www.startvbdotnet.com/ado/default.aspx for information on ADO.NET
10. http://mypage.iusb.edu/~hhakimza/505/index.html for ADO.NET objects.
11. http://msdn.microsoft.com for ADO.NET objects.