**CHAPTER 1**

**INTRODUCTION**

* 1. **AIM**

The purpose of the project is to make an e-retailing website for the farmers to sell their products. The basic idea is to make the farmer to sell his goods to the users. The aim is to build an e-commerce website through which the farmers can fix price for their product and sell them to public.

* 1. **PROBLEM STATEMENT**

Farmers in India don’t get their justifiable rates for their goods (Rice, Wheat, Vegetables etc.) they produce in farming. Their goods are bought by distributors and manufacturers at much lower rates and sold to public with higher rates. In this case the distributors get more profits, and the farmer who puts formidable efforts end up with less profit. And people also have no way but to buy the agriculture products at the rates which distributors specify. Our Project lets the farmers fix Prices for their products and add them to our Website while the customer buys the available products in our Site.

* 1. **DESCRIPTION**

This Project acts as an interface between the farmer and the customer. The farmers can fix the rates of their products through it, and use the website as a medium to sell them. Hence the proposed system helps the farmers to get more profit for their products as the cost raised by the distributors directly benefit them. The customers also get products at reasonable rates. Another advantage of our system is that it is the first of its kind to deliver Agricultural Products at customer’s door steps.

Agriculture is the essence that keeps us alive and is the Back bone of Indian Economy. Hence it is crucial to protect it from several evils surmounting it. The Farmers in India don't get their justifiable/reasonable rates for the goods they produce in farming. Their goods are bought by distributors and manufacturers at much lower rates and sold to public with higher rates. In this case, only the distributors get more profit, while both the Farmer and the customer end up in agony. As a result of this, The Farmers in India leave Agriculture and choose some other Profession for their Livelihood.

In our project “E-Retailing of agricultural products” we introduce farmers into a common e-commerce website and make them fix prices for their products, so that they can sell them directly to public. By having them on this website we can make local farmer's market to be known worldwide.

In this way the farmers get justifiable rates for their products and earn good profit. By implementing it we can improve the farmer's economic level and also it is good for the public as they make obvious pay for their purchase.

* + 1. **PROGRAMMING LANGUAGE DESCRIPTION**
       1. **ASP.NET**

**ASP.NET** is an [open source](http://en.wikipedia.org/wiki/Open_source) [server-side](http://en.wikipedia.org/wiki/Server-side_scripting) [Web application framework](http://en.wikipedia.org/wiki/Web_application_framework) designed for [Web development](http://en.wikipedia.org/wiki/Web_development) to produce [dynamic Web pages](http://en.wikipedia.org/wiki/Dynamic_Web_pages). It was developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft) to allow [programmers](http://en.wikipedia.org/wiki/Programmer) to build dynamic [web sites](http://en.wikipedia.org/wiki/Web_site), [web applications](http://en.wikipedia.org/wiki/Web_application) and [web services](http://en.wikipedia.org/wiki/Web_service). It was first released in January 2002 with version 1.0 of the [.NET Framework](http://en.wikipedia.org/wiki/.NET_Framework), and is the successor to Microsoft's [Active Server Pages](http://en.wikipedia.org/wiki/Active_Server_Pages) (ASP) technology. ASP.NET is built on the [Common Language Runtime](http://en.wikipedia.org/wiki/Common_Language_Runtime) (CLR), allowing programmers to write ASP.NET code using any supported [.NET language](http://en.wikipedia.org/wiki/List_of_CLI_languages). The ASP.NET [SOAP](http://en.wikipedia.org/wiki/SOAP) extension framework allows ASP.NET components to process SOAP messages.

ASP.NET is built on the .NET framework, which provides an application program interface ([API](http://techterms.com/definition/api)) for software programmers. The .NET development tools can be used to create applications for both the Windows operating system and the Web. Programs like Visual Studio .NET provide a visual interface for developers to create their applications, which makes .NET a reasonable choice for designing Web-based interfaces as well. In order for an ASP.NET website to function correctly, it must be published to a Web server that supports ASP.NET applications. Microsoft's Internet Information Services (IIS) Web server is by far the most common platform for ASP.NET websites.

**1.3.1.2 SQL**

SQL (Structured Query Language) is a [special-purpose programming language](http://en.wikipedia.org/wiki/Special-purpose_programming_language) designed for managing data held in a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system)(RDBMS), or for stream processing in a [relational data stream management system](http://en.wikipedia.org/wiki/Relational_data_stream_management_system) (RDSMS). Microsoft SQL Server is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft). As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many [concurrent users](http://en.wikipedia.org/wiki/Concurrent_user). Its primary [query languages](http://en.wikipedia.org/wiki/Query_language) are [T-SQL](http://en.wikipedia.org/wiki/Transact-SQL) and [ANSI SQL](http://en.wikipedia.org/wiki/SQL).

**1.4 BENEFITS**

* By having farmers on this website we can make local farmer’s market to be known worldwide.
* Farmers get more profit for their Products and economy level would be increased.
* Customers get the agricultural products at their door steps without having to travel to crowded markets
* Provide interactive interface through which a user can interact with different areas of application easily.

**1.5 OTHER APPLICATIONS**

* Retail.
* Regional rate fixing.
* No bargaining.
* A chance for the farmer’s to get into modern internet world.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 EXISTING SYSTEM**

The problem with existing system is that many e-commerce websites like Flipkart, Amazon are available for all kinds of Product like electronics, books, textiles, etc. The existing system does not sell Agricultural products via online and also the Products reach the Customers through Distributors.

The existing e-commerce sites provide the products for agriculture like products for irrigation purposes, fertilizers for the development of crops. The basic problems with the existing systems are non-interactive environment they provide to the website users. A search engine that would display the results based on various parameters. The use of traditional and non-user friendly interfaces are hard to use.

The farmers end up with much less profit as their goods brought by the distributors at dead rates and sold in the market in large sums. The international oil price also decides market rates of the agricultural products and hence a marginal rice in it causes a steep rice in the products.

**2.1.1 DRAWBACKS**

* The existing system doesn’t provide the agricultural products for sale.
* The distributors buy the product at a much less rate from the farmer and sell them at a large sum to public.
* Farmers cannot add their product directly to the websites.

**2.2 PROPOSED SYSTEM**

The current website will remain as a platform between farmer and the consumer. The farmers can fix the rates of their product through it, and use the website as a medium. Separate Logins are provided for the farmer and the customer. The farmer can fix Prices for his product, specify the quantity and adds the Product to the Website. The added Product gets displayed in the Customer’s view and the customer can buy the product he wishes to buy.

The farmers and the normal users have their separate usage area. The farmers and users have to login separately for adding products and purchase products respectively. The farmers can add their product directly to site after the registration process get satisfied. The products will be added to the sites after a small verification process.

Our proposed system will provide a direct relation to the user interface of the website and will be viewed directly by the users. The users can easily purchase the product through our user interfaces.

Agriculture is the essence that keeps us alive and is the Back bone of Indian Economy. Hence it is crucial to protect it from several evils surmounting it.

The Farmers in India don't get their justifiable/reasonable rates for the goods they produce in farming. Their goods are bought by distributors and manufacturers at much lower rates and sold to public with higher rates. In this case, only the distributors get more profit, while both the Farmer and the customer end up in agony. As a result of this, The Farmers in India leave Agriculture and choose some other Profession for their Livelihood.

We introduce farmers into a common e-commerce website and make them fix prices for their products, so that they can sell them directly to public. By having them on this website we can make local farmer's market to be known worldwide.

In this way the farmers get justifiable rates for their products and earn good profit. By implementing it we can improve the farmer's economic level and also it is good for the public as they make obvious pay for their purchase.

Our project provides an easy interface for the farmers to add their products and also fix price for it. On the user front, customers can navigate to their desired product through categories and select a desired product along with the quantity.

**2.2.1 ADVANTAGES**

* The motive of this shopping site is to allow the farmers to fix price for his product and sell them online.
* The Farmers get more profit as the distributors don’t come in to play.
* The customers can easily track the required product and buy them as the products are listed according to their category.
* Provide interactive interface through which a user can interact with different areas of application easily.

**2.3 EXAMPLE**

Example for an Interactive interface through our website.

User - Enter the website address to visit our site.

User - Login to purchase products.

User - Add the products to the cart and proceed for further transactions.

System - Successfully purchased.

Farmers - Perform the registration process.

Farmers - Add the product details which are going to be added.

System - Successfully added.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 FEASIBILITY STUDY**

The main objective of feasibility study is to test the Technical, operational and economical for adding new modules and debugging old running system. All system is feasible for adding new modules and debugging old running system.

There are aspects in the feasibility study portion of the preliminary investigation:

* Technical feasibility
* Economic feasibility
* Operational feasibility

**3.1.1 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. The technical software requirement needed for this project is visual basic and Microsoft server SQL. The requirements here are very modest because the system here supports very basic operating system with a normal browser on it. So, this system is technically feasible because it needs only a computer with basic configurations.

**3.1.2 ECONOMIC FEASIBILITY**

Economic feasibility is the most frequently used method for evaluating the effectiveness of the proposed system. More commonly used analysis known as cost benefit- analysis, the procedure that costs for a proposed system and weights them against the tangible and intangible benefits of the system.

The system is cost effective because it is freely available as software and can be downloaded at free of cost from the Microsoft websites or on the internet. The system is economically feasible because the users can freely access the websites in a browser on their phones, computers and can run it.

**3.1.3 OPERATIONAL FEASIBILITY:**

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user and the farmers to use the system efficiently. This system will not threaten the user and the farmers instead it is friendly in its operation. All the user needs to have is an Internet connection provided to his computer or mobiles. The system is operationally feasible because the user can run the website just by typing the website address on the space provided on the browser. The whole process is done only through the user interaction process.

* 1. **SYSTEM REQUIREMENTS**
* Software requirements
* Hardware requirements
  + 1. **SOFTWARE REQUIREMENTS**
* ASP.Net Using Visual Studio 2012 web express.
* Microsoft SQL server 2012.
* Win 7/8/8.1 OS.
* Internet browser like google chrome/Mozilla firefox/Internet explorer/Safari etc…
  + - 1. **VISUAL STUDIO 2012 WEB EXPRESS**

**Microsoft Visual Studio** is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows superfamily of operating systems, as well as web sites, web applications and web services. Visual Studio supports different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010). Support for other languages such as M, Python, and Ruby among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Individual language-specific versions of Visual Studio also exist which provide more limited language services to the user: Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++.

**3.2.1.2 .NET FRAMEWORK**

**.NET Framework** is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large library and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for .NET Framework execute in a software environment (as contrasted to hardware environment), known as the Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling. The class library and the CLR together constitute .NET Framework.

**3.2.1.3 MICROSOFT SQL SERVER**

SQL (Structured Query Language) is a [special-purpose programming language](http://en.wikipedia.org/wiki/Special-purpose_programming_language) designed for managing data held in a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS), or for stream processing in a [relational data stream management system](http://en.wikipedia.org/wiki/Relational_data_stream_management_system) (RDSMS).

Microsoft SQL Server is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft). As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many [concurrent users](http://en.wikipedia.org/wiki/Concurrent_user). Its primary [query languages](http://en.wikipedia.org/wiki/Query_language) are [T-SQL](http://en.wikipedia.org/wiki/Transact-SQL) and [ANSI SQL](http://en.wikipedia.org/wiki/SQL).

* + 1. **HARDWARE REQUIREMENTS**

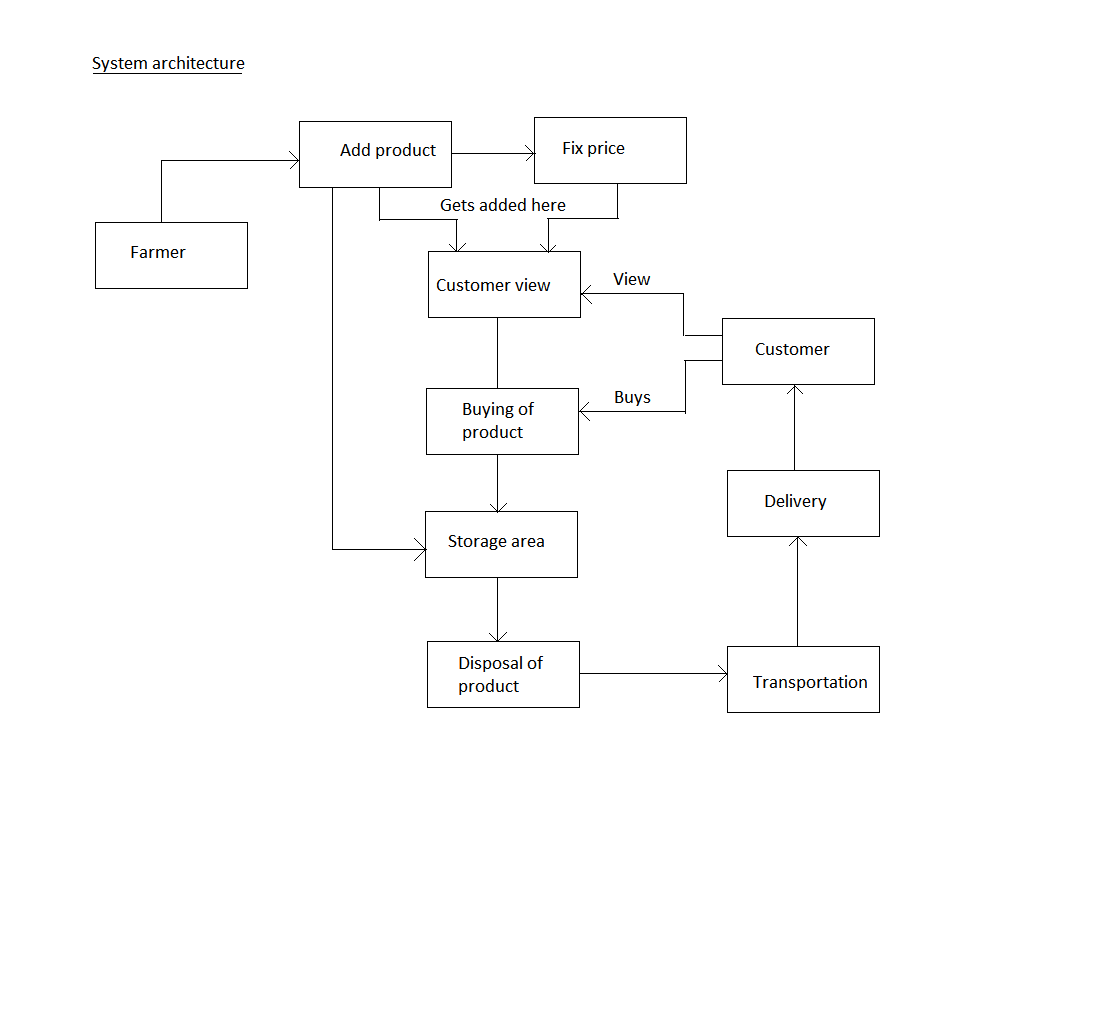
A laptop with the basic configuration of

* Core I3 Processor
* 1 GB Ram
* NIC card for internet connectivity.

**CHAPTER 4**

**DETAILED DESIGN**

**4.1 SYSTEM ARCHITECTURE**

****

**Fig 4.1 SYSTEM ARCHITECTURE**

**4.2 DESIGN GOALS**

* The design of the website involves the design of the forms for listing the products, search for products, display the complete specification for the product and design a shopping cart that is easy to use.
* The Website Design should be such that it enables the user to search the products based on different parameters.
* The Design should help the farmer to add his product without much complexity.
* The Design should help the farmer to choose category of his product that makes it easy for the product to catch the user’s eye.
  1. **MODULES**

The following modules being are used.

* + 1. **USER MODULE**
       1. **Registration**

This module contains the registration form where the new user can enter details like Name. Address, Phone Number, Mobile Number, Email Address and so on. The registered user only can purchase product through this system.

* + - 1. **User Login**

This module contains login form where the user can enter the username and password to authenticate the user from database whether the user is registered user or not in order to access this system.

* + - 1. **Purchase Product**

This module contains the list of products and prices where the user can choose the product to purchase in this system.

* + 1. **Farmer module**
       1. **Registration**

This module contains the registration form where the new user can enter details like Name. Address, Phone Number, Mobile Number, Email Address and so on. The registered user only can add product to this system.

* + - 1. **Farmer Login**

This module contains login form where the user can enter the username and password to authenticate the user from database whether the user is registered user or not in order to access this system.

* + - 1. **Product Adding/Deleting**

This module is used to add new product to this system or remove the existing product from this system based on sales, features and customer feedback.

* + 1. **Admin module**
       1. **Customer Verification**

This module is used to identify whether existing customer or new user is processing before any transaction.

* + - 1. **Farmer Verification**

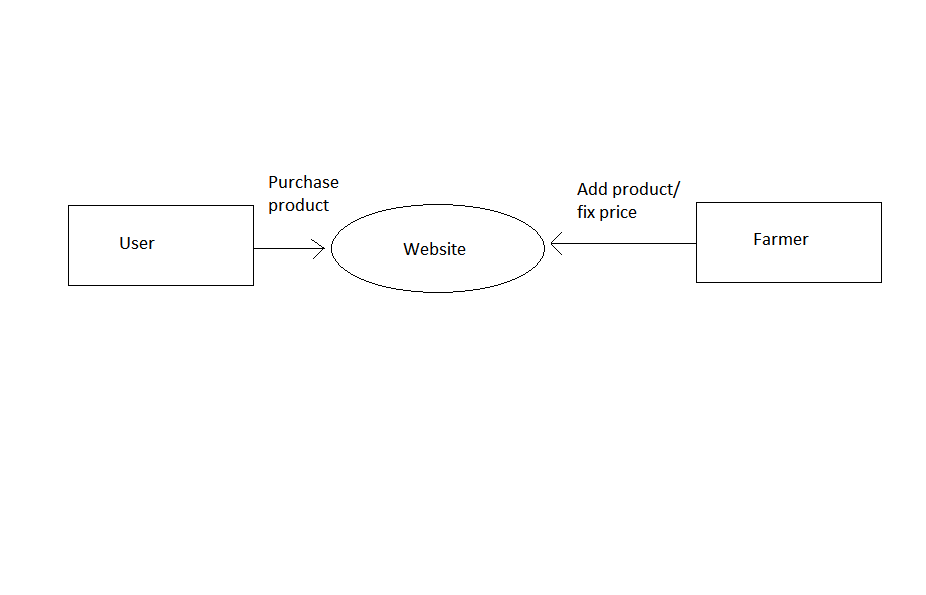
This module is used to identify whether existing farmer or new farmer is processing before any transaction.

* 1. **FLOW DIAGRAM**

The following figure explains the data flow diagrams with various levels.

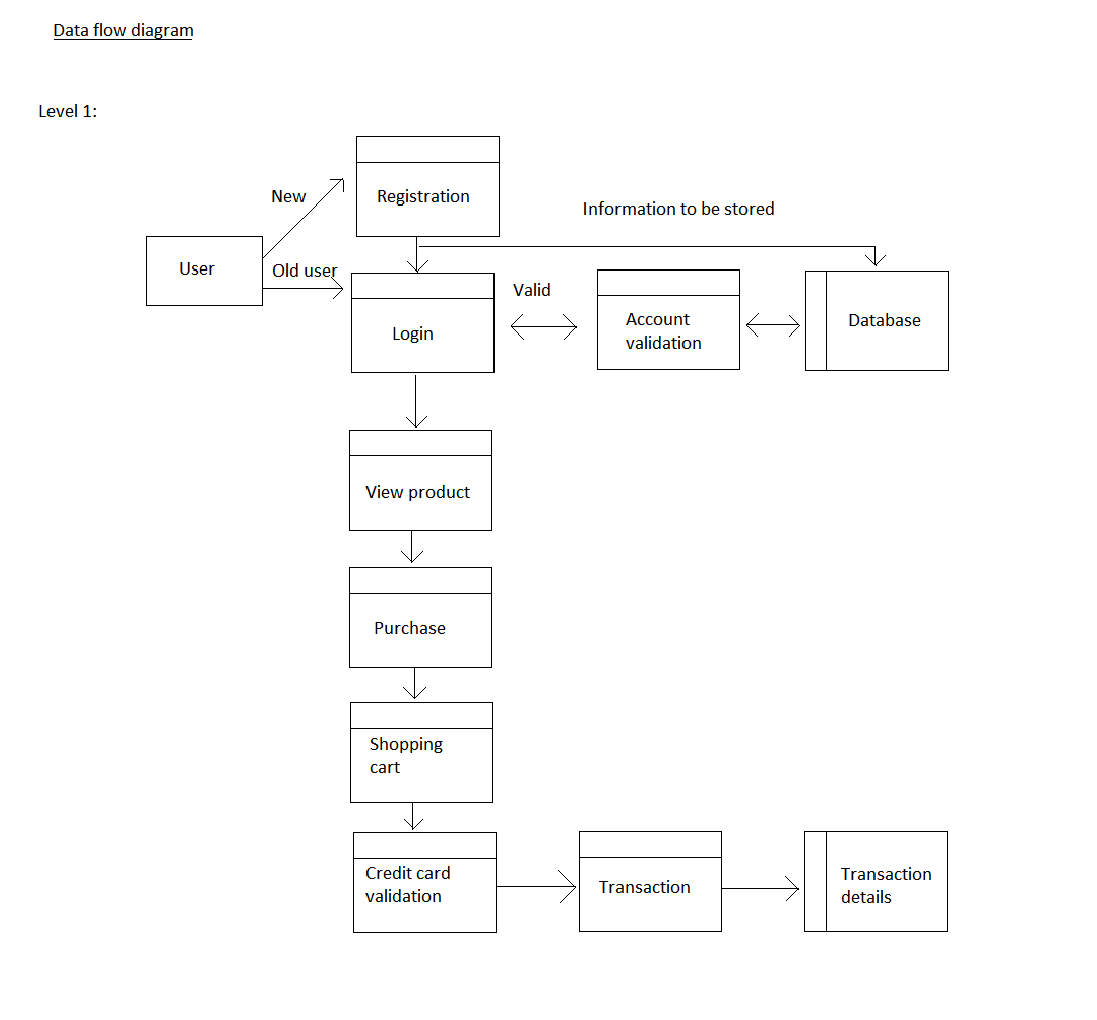
The initial level 0 explains the interaction of the farmers and the users to the website. The level 1 diagram explains the user process on the website such as user login registration, viewing products and purchasing of products etc. The level2 diagram explains the interaction of the farmers to the website and their processing steps through the websites.

**LEVEL 0**

****

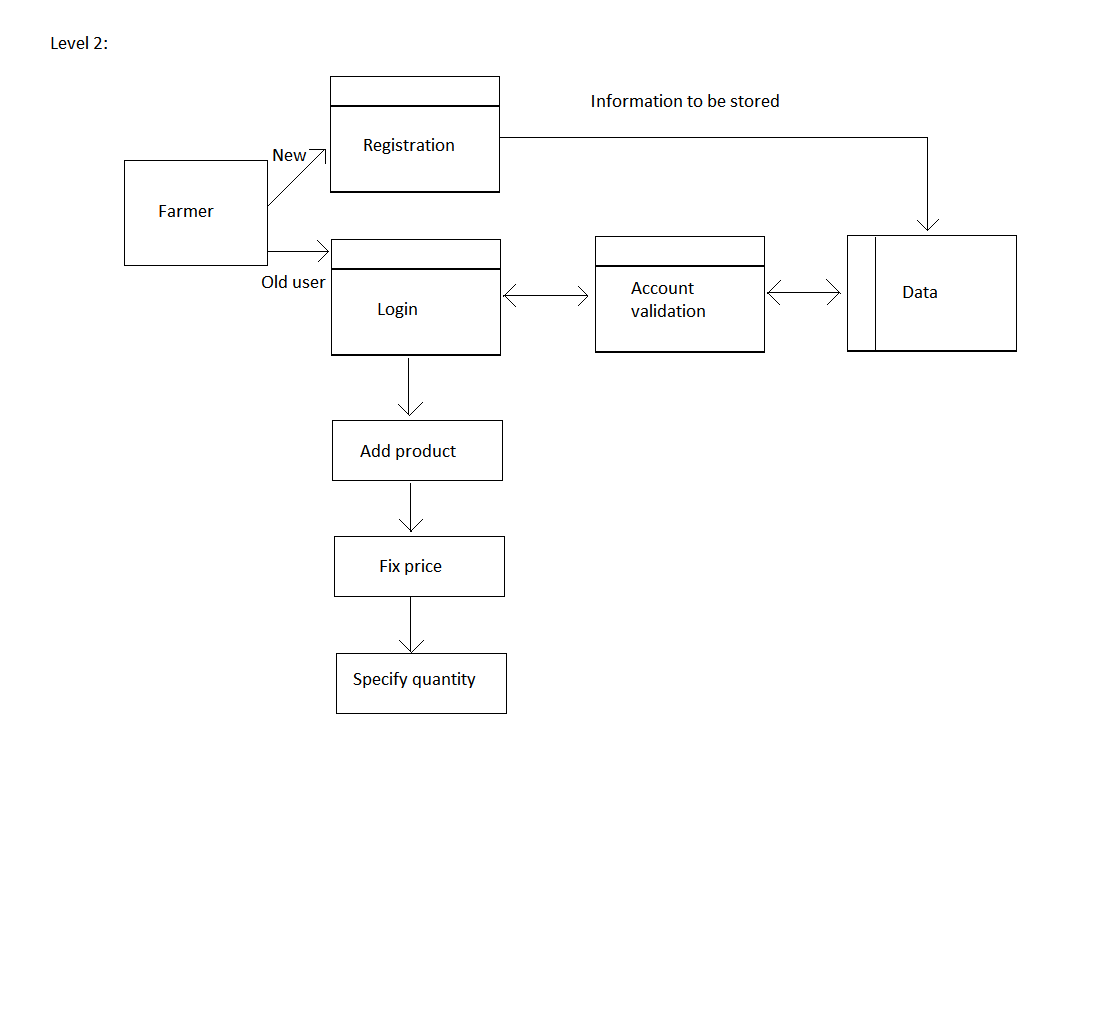
**Fig 4.2 FLOW DIAGRAM: LEVEL 0**

**LEVEL 1**



**Fig 4.2 FLOW DIAGRAM: LEVEL 1**

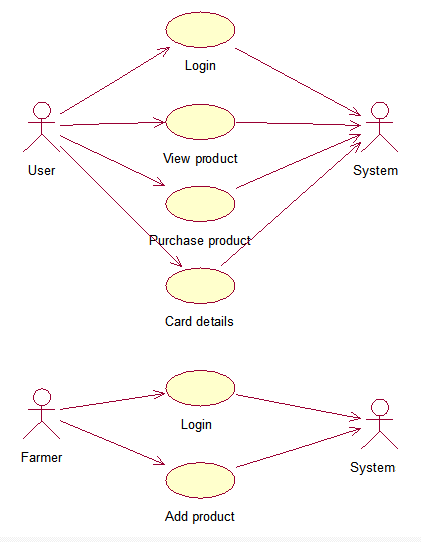
**LEVEL 2**

**Fig 4.2 FLOW DIAGRAM: LEVEL 2**

**4.5 UML DIAGRAMS**

The Unified Modelling Language (UML) is a general purpose modelling language in the field of software engineering. The basic level provides a set of graphic notation techniques to create visual methods of object-oriented software-intensive systems. Object-oriented analysis and design (OOAD) is a software engineering approach that models a system as a group of interacting objects.

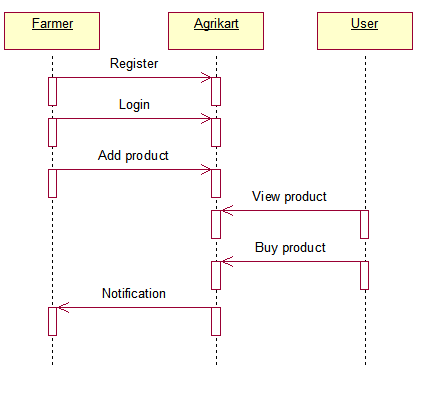
**4.5.1 USECASE DIAGRAM**



**Fig 4.3 USECASE DIAGRAM**

**4.5.2 SEQUENCE DIAGRAM**

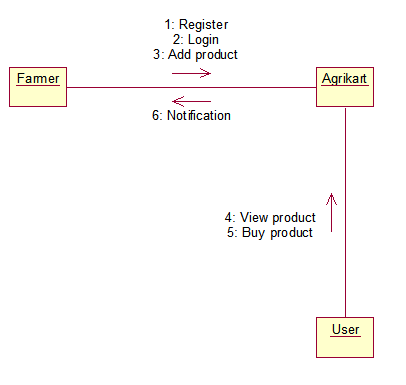
An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them. A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a message sequence chart.

****

**Fig 4.4 SEQUENCE DIAGRAM**

**4.5.3 COLLABORATION DIAGRAM**

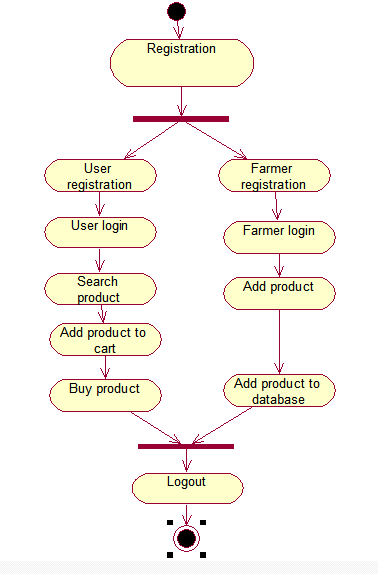
A collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Graphically, a collaboration diagram is a collection of vertices and arcs. This diagram depicts the interaction among the class and the information passed between the classes. The arrow and data suggest the information to pass among the classes.

****

**Fig 4.5 COLLABORATION DIAGRAM**

**4.5.4 ACTIVITY DIAGRAM**

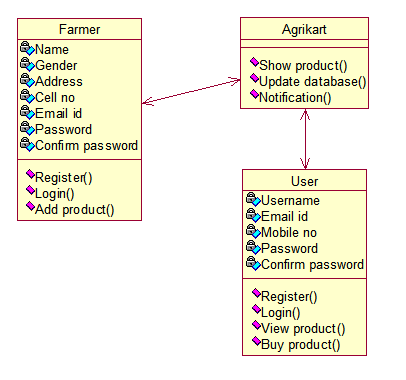
Activity diagram represents the performance of operations and transitions are triggered by completion of operations. Activity diagram are graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

****

**Fig 4.6 ACTIVITY DIAGRAM**

**4.5.5 CLASS DIAGRAM**

Class diagram are the most common diagram found in modelling object-oriented systems. A class diagram shows a set of classes, interfaces, and collaboration and their relationships. Class Diagram provides an overview of the target system by describing the objects and classes inside the system and the relationships between them. It provides a wide variety of usages; from modelling the domain-specific data structure to detailed design of the target system.

****

**Fig 4.7 CLASS DIAGRAM**

**CHAPTER 5**

**IMPLEMENTATION AND TEST PLAN**

**5.1 IMPLEMENTATION**

This project deals with interaction between the users and the farmers through our user interface. In our project the normal users and the farmers are having a separate module for usages. The farmers and the users can login separately and perform their works separately. The farmers can register themselves and get verified for the registration purposes. Then they can add their products directly to the websites by clicking add product button. The users can also login separately to view products and add the products to their cart for the buying process. The users initially have to register themselves for creating a user account and they can proceed directly for the purchase process.

The account details entered on the registration process are added to the databases. The login status of both the users and the farmers are also added to the databases. The database is connected to the visual basic 2012 web express through the Microsoft SQL server software. Whenever the login process will conducted the entered details are verified on the databases for the existence process. If it is already available means the users can be allowed to enter into further process. If the farmers decide to add product on the website they have to enter the details of the product they are going to add on it. Then automatically the details are added to the databases and are displayed on the categories options. After the product purchasing the details are entered on the databases and the quantity available on the product display will get decreased.

**Farmer Registration**

Initially farmers have to create an account for adding the products to the website. For that process they have to register themselves. The details entered should be added to the database. For that process the Microsoft SQL server connectivity is attached to the document for storing the details entered to the database. After the details were stored on the database through the connectivity the farmer were redirected to the add product page. In that page the product description for the product to be added by the farmer have to be added to the database. Then finally the product will be added to the display and the database.

**User Registration**

Initially users have to create a user account for viewing the agricultural products which were displayed on the website. The users have to enter their details on the form for registration. Like the farmer registration process the same method is followed for the registration process. After the successful registration and login the users were redirected to the buy product page. In that page the product from the database were displayed for selecting the category process. Then they were redirected to the product purchase page. The details entered for purchasing should be securely stored on the database through the Microsoft SQL server connectivity.

**5.2 TESTING**

* Software testing is a process of running with intent of finding errors in software.
* Software testing assures the quality of software and represents final review of other phases of software like specification, design, code generation etc.

**5.2.1 Unit Testing**

The functionality of the modules was also tested as separate units. Each of the two modules was tested as separate units. In each module all the functionalities were tested in isolation. Various methods have been created for the purpose of unit testing. Test cases are automatically generated for these methods. The tests run under the application context which means settings from Web.config file are automatically picked up once the test case starts running. Methods were written to retrieve all the manufacturers from the database, strings that match a certain search term, products that match certain filter criteria, all images that belong to a particular product etc. Unit test cases were automatically generated for these methods. The following table shows the test cases for the unit testing process.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test** | **Description** | **Test Inputs** | **Expected Results** | **Actual results** | **Result** |
| 1 | Check for inputting values in Email field | Email=[**shan@yahoo.com**](mailto:shan@yahoo.com)  Username=dave | Inputs should be accepted. | Inputs are accepted. | Pass |
| 2 | Check for inputting values in Email field | Email=[**knki@gmail.com**](mailto:knki@gmail.com)  Username=john | Inputs should be accepted. | Inputs are accepted. | Pass |

**Table 5.1 Unit testing**

**5.2.2 Integration Testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test** | **Description** | **Test Inputs** | **Expected Results** | **Actual results** | **Result** |
| 1 | Check for Login Screen | Enter values in Email and Username.  For Eg:  Email =[**shilpa@yahoo.com**](mailto:shilpa@yahoo.com)  Username=shilpa | Inputs should be accepted. | Inputs are accepted. | Pass |
| Backend Verification | Select email, username from the entered field. | The entered Email and Username should be displayed at SQLprompt. | The entered Email and username were displayed at SQLprompt. | Pass |
| 2 | Check for Product Information | Click product information link | It should display complete details of the product | The complete details of the product were displayed. | Pass |

**Table 5.2 Integration testing**

In Integration testing a system consisting of different modules is tested for problems arising from component interaction. Integration testing should be developed from the system specification. Firstly, a minimum configuration must be integrated and tested. I have done integration testing in a bottom up fashion i.e. in this project I have started construction and testing with atomic modules. After unit testing the modules are integrated one by one and then tested the system for problems arising from component interaction. The following table shows the test case for integration testing.

**5.2.3 Performance Testing**

Jakarta JMeter, a tool for testing applications was used to simulate the virtual users (clients) and test the performance of the system. It can be used to test performance both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more). It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. It can be used to make a graphical analysis of performance and test the server/script/object behavior under heavy concurrent load. I have done performance testing to achieve an estimate of the peak and sustained load the application. This has done with few pages like the Shop Products (extensive Database access, business logic Intensive and more Images) and the Cart Details (simple page). The tests have been conducted by running the application (server) and JMeter on same machine. These test results do not include factors like network bandwidth etc as the server is running on the same machine along with JMeter.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Case** | **Expected Output** | **Observed Output** | **Result** |
| 1 | Required details are given on the field. | The product should be purchased. | Product is purchased. | Pass |

**Table 5.3 Performance testing**

**5.3 TEST PLAN**

The project is tested to verify its correctness and identify the bugs. The test plan includes the various test cases that acts as the set of conditions or variables that determine whether the corresponding feature in the system is working as it originally established to do so. When this test plan is executed, the errors spotted are rectified and the final testing yields following result.

**5.4 TEST ANALYSIS**

In this phase of testing, the requirements for software testing are analyzed and later its feasibility is determined. In the feasibility study the possibility of project development is found through suitable test cases.

**5.5 RESULT**

All the test cases mentioned above passed successfully. No defects encountered. The website have executed successfully without no errors during the execution time.

**CHAPTER 6**

**CONCLUSION AND FUTURE ENHANCEMENT**

**6.1 CONCLUSION**

We conclude that our project “E-Retailing of agricultural products” is a great boon for the farmers to sell their products online and also fix price for the products they sell which gives them higher profits and increase their economy level. If farmers get more profits, then there is a high probability that people would leave their jobs and then move to Agriculture. And also our project lets the customers to buy their Agricultural products online and it is delivered to them at their door step.

**6.2 FUTURE ENHANCEMENT**

This project can be developed in a way that the farmers can easily sell their products through the internet marketing. This website will be developed to add additional products such as natural fertilizers, products for irrigation purposes based on the environment study such as soil, ground water supply etc. This project can be a platform between the users and the famers. The comment blog can also be attached to the website from that the farmers can share their experience and thoughts for future development process.

**APPENDIX-A**

**SAMPLE SOURCE CODE**

**HomePage.aspx**

<%@ Page Title="" Language="C#" MasterPageFile="~/Agri.Master" AutoEventWireup="true" CodeBehind="HomePage.aspx.cs" Inherits="Agrikart.HomePage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<link href="Content/Style1.css" rel="stylesheet" type="text/css" />

<div style="background-color: #C1D4E6;">

<div><br /></div>

<div style="width: 500px">

<table><tr><td>

<img src="images/Agricultural-India.jpg" width="500px" height="400px" />

</td></tr>

<tr><td> <p>&nbsp;&nbsp;&nbsp;<b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; Agriculture</b> is the cultivation of animals, plants, fungi, and other life forms for food, fiber, biofuel, medicinals and other products used to sustain and enhance human life. Agriculture was the key development in the rise of sedentary human civilization, whereby farming of domesticated species created food surpluses that nurtured the development of civilization. The study of agriculture is known as agricultural science.</p>

</td></tr></table></div></div></asp:Content>

**UserRegistration.aspx**

<%@ Page Title="" Language="C#" MasterPageFile="~/Agri.Master" AutoEventWireup="true" CodeBehind="UsersRegistration.aspx.cs" Inherits="Agrikart.UsersRegistration" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<link href="Content/Style1.css" rel="stylesheet" type="text/css" />

<center>

<table width="100%" style="background-color:#C1D4E6;">

<tr>

<td colspan="4"><br /> </td>

</tr>

<tr><td></td>

<td colspan="2" align="center" class="header">User Register</td>

<td></td></tr>

<tr><td colspan="4"><br /></td>

</tr>

<tr>

<td></td>

<td align="right"><asp:Label ID="lblFarmer" runat="server" Text="UserName:" CssClass="labelcontrol"/></td>

<td align="left"><asp:TextBox ID="txtusername" runat="server" CausesValidation="True" Width="160px"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="rfvduserid" runat="server" ForeColor="Red" ControlToValidate="txtusername" ErrorMessage="Please enter your name..." ToolTip="UserID must be fill"></asp:RequiredFieldValidator></td>

</tr>

<tr><td></td>

<td align="right"><asp:Label ID="Label2" runat="server" Text="EmailID:" CssClass="labelcontrol" /></td>

<td align="left"><asp:TextBox ID="txtemail" runat="server" CausesValidation="True" Width="160px"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="rfvdemail" runat="server" forecolor="red" ControlToValidate="txtemail" ErrorMessage="Please enter your emailid..." Tooltip="Mail must be fill"></asp:RequiredFieldValidator>

<asp:RegularExpressionValidator ID="revdmail" runat="server" forecolor="red" display="Dynamic" ControlToValidate="txtemail" ErrorMesssage="must be enter valid mail" tooltip=" must enter valid mail format" ValidationExpression="\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*"></asp:RegularExpressionValidator></td>

</tr><tr<td></td>

<td align="right"><asp:Label ID="Label3" runat="server" Text="MobileNumber:" CssClass="labelcontrol"/> </td>

<td align="left"><asp:TextBox ID="txtmobile" runat="server" CausesValidation="True" Width="160px"></asp:TextBox></td>

<td><asp:requiredfieldvalidator ID="rfvdmobile" runat="server" forecolor="red" ControlToValidate="txtmobile" ErrorMessage="Mobile Number Must not be empty" tooltip="Please enter your mobileno..."></asp:requiredfieldvalidator>

<asp:regularexpressionvalidator ID="revdmobile" runat="server" forecolor="red" ControlToValidate="txtmobile" ErrorMessage="Mobile Number Must be Integer" tooltip="allowed only numbers" ValidationExpression="^\d+$" ></asp:regularexpressionvalidator></td>

</tr><tr><td></td>

<td align="right"><asp:Label ID="Label4" runat="server" Text="Password:" CssClass="labelcontrol" /></td>

<td align="left"><asp:TextBox ID="txtpassword" runat="server" TextMode="Password" CausesValidation="true" Width="160px"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="rfvpassword" runat="server" ControlToValidate="txtpassword" ErrorMessage="Password must not be empty" forecolor="red" tooltip="Password must not be empty"></asp:RequiredFieldValidator></td>

</tr><tr><td></td>

<td align="right"><asp:Label ID="Label5" runat="server" Text="ConfirmPassword:" CssClass="labelcontrol"/></td>

<td align="left"><asp:TextBox ID="txtconfirmpassword" runat="server" TextMode="Password" CausesValidation="true" Width="160px"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="rfvconfirm" runat="server" ControlToValidate="txtconfirmpassword" ErrorMessage="Confirm Password must not be empty" ForeColor="Red" ToolTip="Confirm Password must not be empty"></asp:RequiredFieldValidator><asp:CompareValidator ID="cvconfirmpassword" runat="server" ControlToCompare="txtpassword" ControlToValidate="txtconfirmpassword" ErrorMessage="Password and Confirm Password must be match" ForeColor="red" ToolTip="Password and ConfirmPassword must be match"></asp:CompareValidator></td>

</tr>

<tr><td colspan="4"><br /></td></tr>

<tr><td></td>

<td colspan="2" align="center"><asp:Button ID="btnsubmit" runat="server" Text="Submit" BackColor="Green" style="width:80px; height:30px" OnClick="btnsubmit\_Click" />&nbsp;&nbsp;<asp:Button ID="btncancel" BackColor="Orange" runat="server" style="width:80px; height:30px" Text="Cancel" Width="80px" OnClick="btncancel\_Click" /></td>

<td></td></tr>

<tr><td colspan="4"><br /> </td></tr></table></center></asp:Content>

**FarmerRegisteration.aspx**

<%@ Page Title="" Language="C#" MasterPageFile="~/Agri.Master" AutoEventWireup="true" CodeBehind="FarmersRegistration.aspx.cs" Inherits="Agrikart.FarmersRegistration" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<link href="Content/Style1.css" type="text/css" rel="stylesheet" />

<center>

<table cellpadding="5" cellspacing="3" width="100%" style="background-color:#C1D4E6;">

<tr><td colspan="4"><br /> </td></tr>

<tr><td></td>

<td colspan="2" align="center"><asp:Label ID="Label1" runat="server" Text="Farmer Register" CssClass="header" /></td>

<td></td></tr>

<tr><td colspan="4"><br /></td></tr>

<tr><td></td>

<td align="right"><asp:Label ID="label3" runat="server" Text="Name:" CssClass="labelcontrol" /></td>

<td align="left"><asp:TextBox ID="txtname" runat="server" width="160px"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="rfvname" ForeColor="Red" runat="server" ErrorMessage="Name should not be empty" ControlTOValidate="txtname"></asp:RequiredFieldValidator></td>

</tr>

<tr><td></td>

<td align="right"><asp:Label ID="Label4" runat="server" Text="Gender:" CssClass="labelcontrol" /></td>

<td align="left"><asp:RadioButtonList ID="rdgender" runat="server" RepeatDirection="Horizontal" Height="29px">

<asp:ListItem Text="Male" Value="Male">

</asp:ListItem>

<asp:ListItem Text="Female" Value="Female">

</asp:ListItem></asp:RadioButtonList></td>

<td><asp:RequiredFieldValidator ID="GenderValidator" ForeColor="Red" runat="server" ErrorMessage="Please Select the gender!!!"

ControlToValidate="rdgender" Display="Dynamic"></asp:RequiredFieldValidator></td></tr>

<tr><td></td>

<td align="right"><asp:Label ID="Label5" runat="server" Text="Address:" CssClass="labelcontrol" /></td>

<td align="left"><asp:TextBox ID="txtaddress" runat="server" width="160px" Textmode="MultiLine"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="rfvaddress" ForeColor="Red" runat="server" ErrorMessage="Address should not be empty" ToolTip="Address should not be empty" ControlToValidate="txtaddress"></asp:RequiredFieldValidator></td></tr>

<tr><td></td>

<td align="right"><asp:Label ID="Label6" runat="server" Text="CellNo:" CssClass="labelcontrol" /></td>

<td align="left"><asp:TextBox ID="txtcontact" runat="server" MaxLength="12" Width="160px"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="rvcontact" ForeColor="Red" runat="server" ErrorMessage="Contact number should not be empty" ToolTip="Contact number should not be empty" ControlToValidate="txtcontact"></asp:RequiredFieldValidator></td></tr>

<tr><td></td>

<td align="right"><asp:Label ID="Label7" runat="server" Text="Pincode:" CssClass="labelcontrol" /></td>

<td align="left"><asp:TextBox ID="txtpin" runat="server" Width="160px" ></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="rfvpin" ForeColor="Red" runat="server" ErrorMessage="Pincode should not be empty" ToolTip="Pincode should not be empty" ControlToValidate="txtpin"></asp:RequiredFieldValidator></td></tr>

<tr><td></td>

<td align="right"><asp:Label ID="Label8" runat="server" Text="AccountNumber:" CssClass="labelcontrol" /></td>

<td align="left"><asp:TextBox ID="txtacno" runat="server" Width="160px"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="rvacno" ForeColor="Red" runat="server" ErrorMessage="Account should not be empty" ToolTip="Account should not be empty" ControlToValidate="txtacno"></asp:RequiredFieldValidator></td></tr>

<tr><td></td>

<td align="right"><asp:Label ID="Label9" runat="server" Text="EmailID:" CssClass="labelcontrol"/></td>

<td align="left"><asp:TextBox ID="txtemail" runat="server" Width="160px"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="rfvmail" runat="server" forecolor="Red" errormessage="MailID should not be empty" ControlToValidate="txtemail" ToolTip="MailID should not be empty"></asp:RequiredFieldValidator>&nbsp;&nbsp;

<asp:RegularExpressionValidator ID="rvemail" runat="server" forecolor="Red" errormessage="Must be enter valid emilid" ControlToValidate="txtemail" ToolTip="Must fill Mailid" Display="Dynamic" validationexpression="\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*" ></asp:RegularExpressionValidator></td></tr>

<tr><td></td>

<td align="right"><asp:Label ID="label10" runat="server" Text="Password:" CssClass="labelcontrol"/></td>

<td align="left"><asp:TextBox ID="txtpass" runat="server" CausesValidation="true" textmode="Password" Width="160px"></asp:TextBox></td>

<td><small style="color:green">(minimum 3-10 characters)</small>&nbsp;

<asp:RequiredFieldValidator ID="rfpassword" runat="server" ControlToValidate="txtpass" ErrorMessage="Password should not be empty" ToolTip="Password should not be empty" ForeColor="Red"></asp:RequiredFieldValidator></td></tr>

<tr><td></td>

<td align="right"><asp:Label ID="Label11" runat="server" Text="ConfirmPassword:" CssClass="labelcontrol" /></td>

<td align="left"><asp:TextBox ID="txtconfrmpassword" runat="server" CausesValidation="true" Width="160px" TextMode="Password"></asp:TextBox></td>

<td><asp:CompareValidator ID="cvdconfrmpassword" runat="server" CausesValidation="true" ForeColor="Red" ControlToCompare="txtpass" ControlToValidate="txtconfrmpassword" ErrorMessage="Password and ConfirmPassword Must be Match" ToolTip="Password and ConfirmPassword Must be Match"></asp:CompareValidator>

<asp:RequiredFieldValidator ID="rvconfrmpassword" runat="server" controltovalidate="txtconfrmpassword" errormessage="Must be enter your confirm password" forecolor="red"></asp:RequiredFieldValidator></td></tr>

<tr><td></td>

<td align="right"><asp:Label ID="Label12" runat="server" Text="Hint:" CssClass="labelcontrol"/></td>

<td align="left"><asp:dropdownlist ID="ddlhint" runat="server" Width="210px" >

<asp:ListItem>what is your favourite book?</asp:ListItem>

<asp:ListItem>what is your timepass?</asp:ListItem>

<asp:ListItem>where did you born?</asp:ListItem>

<asp:ListItem>what is your favourite food?</asp:ListItem>

<asp:ListItem>what is favorite color?</asp:ListItem>

</asp:dropdownlist></td>

<td></td></tr>

<tr><td></td>

<td align="right"><asp:Label ID="Label13" runat="server" Text="Answer:" CssClass="labelcontrol" /></td>

<td align="left"><asp:TextBox ID="txtans" runat="server" width="160px"></asp:TextBox></td>

<td></td></tr>

<tr><td></td><td></td><td></td><td><br /></td></tr>

<tr><td></td>

<td colspan="2" align="center"><asp:Button ID="btnsubmit" Text="Submit" runat="server" BackColor="Green" style="width:80px; height:30px" OnClick="btnsubmit\_Click" /> &nbsp;&nbsp;&nbsp;&nbsp;<asp:Button ID="btnCancel" BackColor="Orange" runat="server" Text="Cancel" style="width:80px; height:30px" OnClick="btnCancel\_Click" /></td>

<td></td></tr>

<tr><td></td><td></td><td></td><td></td></tr>

<tr><td colspan="4"><br /> </td></tr> </table></center></asp:Content>

**PaymentDetails.cs**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Web;

namespace Agrikart

{ public class PaymentDetails

{ static string accName = string.Empty;

static string accNo = string.Empty;

static string totalAmount = string.Empty;

static string totalQuantity = string.Empty;

static string totalPrice = string.Empty;

static string[] priceList = new string[0];

static string[] prodQuantity = new string[0];

static ArrayList productList = new ArrayList();

static DataTable dt = new DataTable();

static PaymentDetails()

{ }

public static string UserId

{ get

{ return accName; }

set

{ accName = value;

} }

public static string AccountNo

{ get {

return accNo;

}

set

{ accNo = value;

} }

public static string TotalAmount

{ get

{ return totalAmount;

}

set

{ totalAmount = value; }

}

public static void ProductList(string prodId)

{

productList.Add(prodId);

}

public static ArrayList GetProducts

{

get

{

return productList;

} }

public static void PriceList(string price)

{

priceList = new string[priceList.Length+1];

priceList[priceList.Length - 1] = price;

}

public static string[] getPriceList

{ get {

return priceList;

} }

public static void ProdQuantity(string qty)

{

prodQuantity = new string[prodQuantity.Length + 1];

prodQuantity[prodQuantity.Length - 1] = qty;

}

public static string[] getProdQuantity

{ get {

return prodQuantity;

} }

public static void storeData(DataTable dtt)

{ dt = dtt;

}

public static DataTable getData

{ get {

return dt;

}}}}

**FarmerLogin.aspx**

<%@ Page Title="" Language="C#" MasterPageFile="~/Agri.Master" AutoEventWireup="true" CodeBehind="Farmerlogin.aspx.cs" Inherits="Agrikart.Farmerlogin" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<link href="Content/Style1.css" rel="stylesheet" type="text/css" />

<Center>

<table width="100%" style="background-color:#C1D4E6;">

<tr><td colspan="4"><br /> </td></tr>

<tr>

<td></td>

<td colspan="2" align="center"><span class="header" >Farmer Login</span></td>

<td></td></tr>

<tr><td colspan="4"><br /></td></tr>

<tr><td></td>

<td align="right"><asp:Label ID="lblFarmer" runat="server" Text="UserName:" CssClass="labelcontrol"/></td>

<td align="left"><asp:TextBox ID="txtfrmname" runat="server" font-size="10pt" height="25" Width="160px"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="txtuservalidate" runat="server" Controltovalidate="txtfrmname" ErrorMessage="Please enter your name..."

tooltip="UserName cannot be blank" ForeColor="Red" Display="Dynamic"></asp:RequiredFieldValidator></td></tr>

<tr><td></td>

<td align="right"><asp:Label ID="Label1" runat="server" Text="Password:" CssClass="labelcontrol"/></td>

<td align="left"><asp:TextBox ID="txtpassword" TextMode="Password" runat="server" Height="25" Width="160px" Font-Size="10pt"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="passvalid" ForeColor="Red" runat="server" ControlToValidate="txtpassword" ErrorMessage="Please enter your password..." ToolTip="password cannot be empty" Display="Dynamic"></asp:RequiredFieldValidator></td></tr>

<tr><td colspan="4"><br /></td></tr>

<tr><td></td>

<td colspan="2" align="center"><asp:Button ID="btnSignin" Text="Login" BackColor="Green" style="width:80px; height:30px" runat="server" OnClick="btnSignin\_Click" />&nbsp;&nbsp;<asp:Button ID="btnCancel" Text="Cancel" runat="server" BackColor="Orange" style="width:80px; height:30px" OnClick="btnCancel\_Click" /></td>

<td></td></tr><tr>

<td colspan="4" align="center">Not yet Registered? Click here to

<a href="FarmersRegistration.aspx"><u>RegisterNow</u></a></td>

</tr> <tr>

<td colspan="4"><br /> </td>

</tr></table></Center></asp:Content>

**Logout.aspx**

<%@ Page Title="" Language="C#" MasterPageFile="~/Farmer.Master" AutoEventWireup="true" CodeBehind="Logout.aspx.cs" Inherits="Agrikart.Logout" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

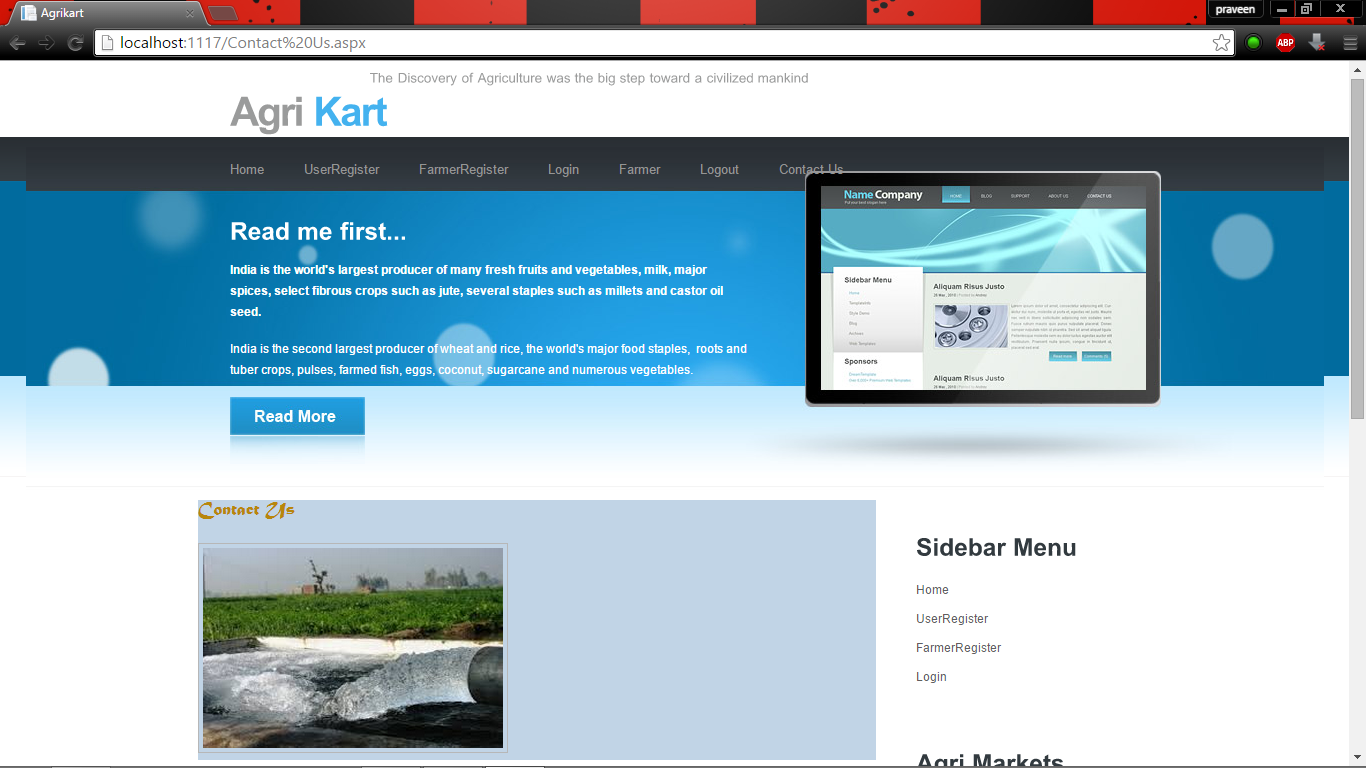
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

</asp:Content>

**APPENDIX-B**

**SCREEN SHOTS**

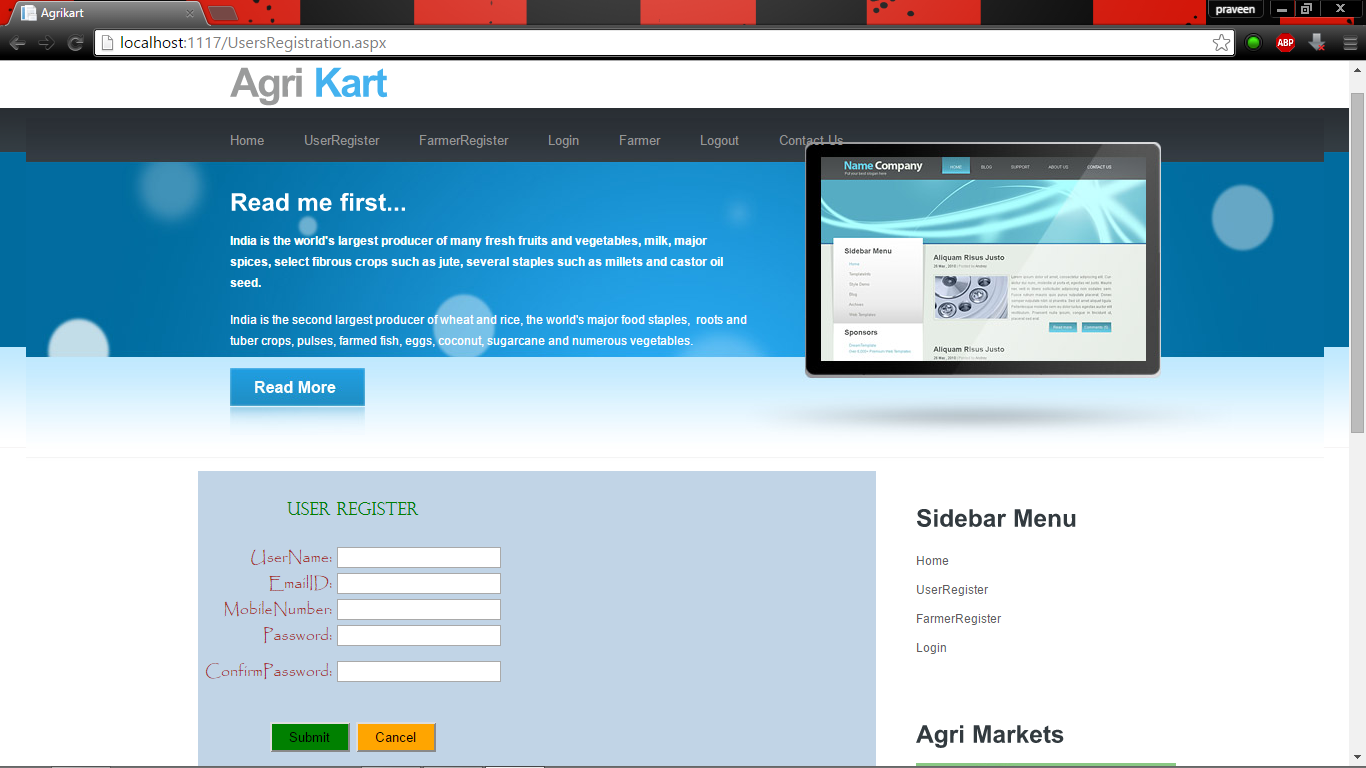
**Home**



**Fig. B.1 homepage**

The above home page screen shot shows that the contents that are present on the home page. It contains the link to other subpages like user registration, farmer registration, login page, contact details etc.

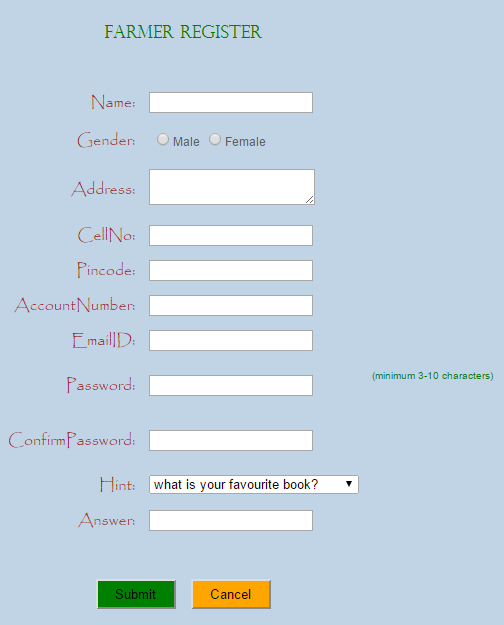
**User Registration**



**Fig.B.2 User registration**

The figure shows the screen shot for the user registration page. In this User registration page the users have to register themselves to create a user account. The details entered by the users are automatically stored to the database. The database stores the user details for future references. Whenever the user entered the username they were checked on the databases for the previous existence of the same username.

**Farmer Registration**



**Fig.B.3 Farmer registration**

The above figure shows the screen shot for farmer registration page. In this farmer registration page the farmers have to register themselves to add product on the databases. The entered details were securely added to the database for the future reference process. After the completion of this registration process the farmer were directed to the add product page.

**Buy product**

****

**Fig B.4 Buy product**

The above screen shot represents the buy product page for the users. After the successful login by the user they were redirected to the buy product page. From there they can select their product for purchase. After the selection process they were redirected to the purchase page. In there they have to enter the purchase details.

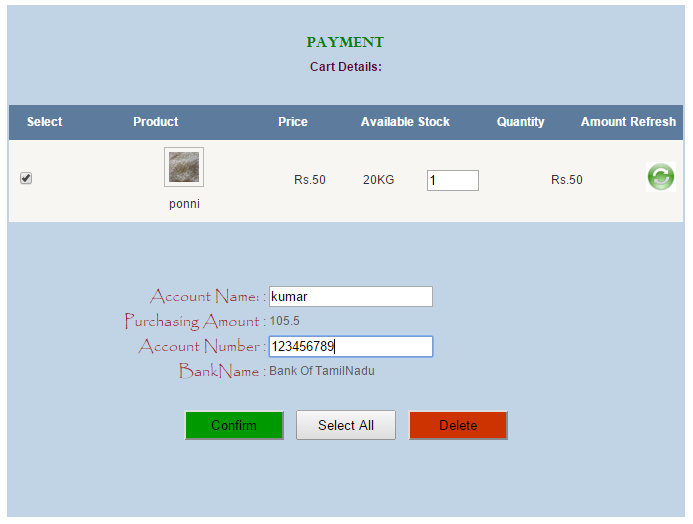
**Product details**



**Fig B.5 Product details**

The above screen shot represents the product details of the product which are selected by the user for the purchasing process. It will show the details of the product selected such as name, category, price, quantity, product image.

**Payment Details**



**Fig B.6 Payment Details**

The above screen shot represents the payment details page. The user can modify the quantity details and can reselect the required product from the cart for purchasing. Then the users have to enter their account holder name and bank details for the purchase. After the verification of the bank details the product is delivered to the user.

**REFERENCES**

1. Step by step Microsoft SQL server 2000 programming, Author*- Rioedan, Rebecca M*
2. Microsoft SQL server 7 with VB 6, Author- *Freeze, Wayne S*
3. Microsoft .NET architecting applications for the enterprise, Author- *Esposito Dino, Salta*
4. Programming c# 2008 and the .NET 3.5 platform, Author- *Troelson Andrew*
5. Programming Microsoft ASP.NET 3.5, Author-Esposito Dino.
6. *Anamika S*. (2011). Top 10 benefits of online shopping. Hubpages . Retrieved (February 14, 2012) from (http://anamikas.hubpages.com/hub/Online-shopping-sites-benefits)
7. *Vespa, M*. (2010, November 11). Online shopping: Helping or hurting local economy. Central Illinois News Center. Retrieved from (http://www.centralillinoisnewscenter.com/home/Online-Shopping--111008209.html)
8. *J Doll*. (2011, December 28). Drunk online shopping will save this wretched economy. Village Voice. Retrieved (February 14, 2012) from (http://blogs.villagevoice.com/runninscared/2011/12/drunk\_online\_shopping\_trend.php)
9. Associated Press. (2011, November 29). Reports say ‘Cyber Monday’ top online shopping day. New York (AP). Retrieved from (http://www.mail.com/business/economy/876350-reports-cyber-monday-top-online-shopping-day.html)