1. **INTRODUCTION**

Now a days everything became online as the technology is growing faster yet there is no awareness to the farmers to improve the crop yield. So there is a need for the farmers to know everything about fertilizers to be used in accordance with the seasons and soils. By using this web application, every farmer can have a clear cut idea about the fertilizers to be used and what type of crops to be cultivated in particular soil and it also gives idea about crops to be cultivated in particular season.

There are many different ways to cultivate crops and all the farmers may or may not know all the required fertilizers to be used. They may know some of them and some ways to cultivate and this may or may not give desired output. So we providethe details to the farmers to fertilize their crops in a procedural manner and clearing their doubts through mailing or by messaging.

**2. EXISTING SYSTEM**

The existing system gives information about soils and type of crops to be cultivated. It provides various details about different crop fertilizers. It provides Static way of representation. Farmers who have knowledge about the system can use this kind of application.

**2.1 DRAWBACKS OF THE EXISTING SYSTEM**

This have several drawbacks:

* The main drawback is that user must have system knowledge then only he can use this application.
* It is not dynamic which means there is no interaction with the farmer.

So it provides poor awareness to the farmer

**3.PROPOSED SYSTEM**

The concept of developing this web application is to provide a dynamic way of representation for farmers by providing them the following

* Registration for farmers.
* Video Conferencing
* Mailing
* Messaging

**3.1 Advantages:**

* It provides a dynamic way for the farmers to interact with the experts, admin and can mail or message queries they have through this application
* They can participate in videoconference
* Farmers can receive solutions for there queries either through mail or through sms

**4. SYSTEM ANALYSIS**

**4.1 FEASIBILITY STUDY**

A feasibility study is a preliminary study undertaken to determine and document a project’s viability. The term feasibility study is also used to refer to the resulting document. These results of this study are used to make a decision whether to proceed with the project. It is an analysis of possible alternative solutions. It, for example, can decide whether an order processing be carried out by a new system more efficiently than the previous one.

There are three different aspects in the feasibility study.

* Technical feasibility
* Operational feasibility
* Economic feasibility

**4.1.1 Economic Feasibility**

Economic analysis is most frequently used for evaluation of the effectiveness of the system. More commonly knows as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions is made to design and implement the system. This is an important input to the management because very often the top management does not like to get confounded by the various technicalities that bound to be associated with a project of this kind.

**4.1.2 Technical Feasibility**

It refers to whether the software that is available in the market fully supports the present application. It studies the pros and cons of using particular software for the development and its feasibility.  It also studies the additional training needed to be given to the people to make the application work.

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on etc. A number of issues have to be considered while doing a technical analysis. Understand the different technologies involved in the proposed system before commencing the project we have to be very clear about what are the technologies that are to be required for the development of the new system. Find out whether the system currently possesses the required technologies. Is the required technology available with the system? This product is technical feasible as the technology available in market can support.

**4.1.3 operational Feasibility**

It refers to the feasibility of the product to be operational. Proposed project is beneficial only if it can be turned into information systems that will meet the operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project.

Is there sufficient support for the project from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change there may be resistance.Are the current methods acceptable to the user? If they are not Users may welcome a change that will bring about a more operational and useful systems. Have the users been involved in the implementing and development of the project.

Early involvement reduces the chances of resistance to the system and in general and increases the likelihood of successful project. Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

Some products may work very well at design and implementation but may fail in the real time environment.  It includes the study of additional human resource required and their technical expertise. This product is operationally feasible as it is a windows application and can be easily installed and run successfully with minimum requirements.

**4.** **2 SOFTWARE REQUIRMENTS SPECIFICATIONS :**

**HARDWARE REQUIREMENTS:**

Processor : Pentium IV, 2.4 GHz

Hard Disk : 320GB (min)

RAM : 2GB (min)

**SOFTWARE REQUIREMENTS:**

Operating System : Windows 7

Programming Language : ASP.NET, C#.Net

Database : SQL Server 2008

**5. SYSTEM DESIGN**

Once software requirements have been analyzed and specified, software design as the first of three technical activities design, code generation and test-that are require to build and software. Various stages in design are

* Conceptual design.
* Database design.
* Functional design.

**Conceptual design:**

The conceptual structure of a database is called schema. Schema shows the kind of data that exists in a database and how these are logically related to each other. A schema can be regarded as a blueprint that portrays, both kind of data used in building a database and logical relationship, and must correctly represent their inter -relationships. Schema is frequently depicted pictorially viz, Data Flow diagram etc.

**Database Design:**

A database management system (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems.

Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

**Description**

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. A DBMS includes: A modeling language to define the schema of each database hosted in the DBMS, according to the DBMS data model.

* The four most common types of organizations are the hierarchical, network, relational and object models. Inverted lists and other methods are also used. A given database management system may provide one or more of the four models. The optimal structure depends on the natural organization of the application's data, and on the application's requirements (which include transaction rate , reliability, maintainability, scalability, and cost).
* The dominant model in use today is the ad hoc one embedded in SQL, despite the objections of purists who believe this model is a corruption of the relational model, since it violates several of its fundamental principles for the sake of practicality and performance. Many DBMSs also support the Open Database Connectivity API that supports a standard way for programmers to access the DBMS.

Data structures (fields, records, files and objects) optimized to deal with very large amounts of data stored on a permanent data storage device (which implies relatively slow access compared to volatile main memory).

A database query language and report writer to allow users to interactively interrogate the database, analyze its data and update it according to the users privileges on data.

* It also controls the security of the database.
* Data security prevents unauthorized users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of it called subschemas. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and medical data.
* If the DBMS provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. However, it may not leave an audit trail of actions or provide the kinds of controls necessary in a multi-user organization. These controls are only available when a set of application programs are customized for each data entry and updating function.

A transaction mechanism, that ideally would guarantee the ACID properties, in order to ensure data integrity, despite concurrent user accesses (concurrency control), and faults (fault tolerance).

* It also maintains the integrity of the data in the database.
* The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time. The DBMS can help prevent duplicate records via unique index constraints; for example, no two customers with the same customer numbers (key fields) can be entered into the database. See ACID properties for more information (Redundancy avoidance).

The DBMS accepts requests for data from the application program and instructs the operating system to transfer the appropriate data. When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. New categories of data can be added to the database without disruption to the existing system. Organizations may use one kind of DBMS for daily transaction processing and then move the detail onto another computer that uses another DBMS better suited for random inquiries and analysis. Overall systems design decisions are performed by data administrators and systems analysts. Detailed database design is performed by database administrators.

Database servers are specially designed computers that hold the actual databases and run only the DBMS and related software. Database servers are usually multiprocessor computers, with RAID disk arrays used for stable storage. Connected to one or more servers via a high-speed channel, hardware database accelerators are also used in large volume transaction processing environments.

DBMSs are found at the heart of most database applications. Sometimes DBMSs are built around a private multitasking kernel with built-in networking support although nowadays these functions are left to the operating system.

**SQL SERVER**

SQL SERVER is one of the leading database management systems (DBMS) because it is the only database that meets the uncompromising requirements of today’s most demanding information systems. SQL SERVER is a truly portable, distributed, and open DBMS that delivers unmatched performance, continuous operation and support for every database.SQL SERVER RDBMS is high performance fault tolerant DBMS which is specially designed for online transactions processing and for handling large database application.

A database is a collection of related data, organized on the basis of relationship in the data rather than the convenience of storage structures. It enables sharing of data among various users as and when required.

**Relational database:**

A data base in which data is stored in tables allowing the relationship among tables and more efficient no redundant data storage and manipulation.

**Table:**

Table is a fundamental structure of Relational Database Management System.

**Field:**

An element of a table that contains a specific item of information such as Account Code. It represents a column.

**Record:**

A collection of data about a particular thing like a book. It represents a row in a table.

**Functional Design:**

Input describes the information to be supplied to this function either by the user on the screen or from any data store. Processing describes the operation to be carried out by the function. The validations necessary are also described at this point.

Output describes the information obtained from the functions or the action carried out by the function.

**Design Principles:**

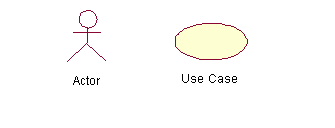
Software design is a both process and a model basic design principles enables the analyst to navigate the design process.

* The design process should not suffer from “tunnel vision”.
* The design should be traceable to analysis model.
* The design should “minimize the intellectual distance” between software and problem that exists in the real world.

**5.1 UML DIAGRAMS**

**USECASE DIAGRAM:**

A use case is a set of scenarios that describing an interaction between a user and a system.  A use case diagram displays the relationship among actors and use cases.  The two main components of a use case diagram are use cases and actors.



An actor is represents a user or another system that will interact with the system you are modeling.  A use case is an external view of the system that represents some action the user might perform in order to complete a task.

Use cases are used in almost every project.  They are helpful in exposing requirements and planning the project. During the initial stage of a project most use cases should be defined, but as the project continues more might become visible.

**USECASE DIAGRAM FOR FARMER MODULE:**

****

The Farmer has the following actions like Register, Login, View Informations, Get Solutions by mail or by sms,participate in video conferencing.

**USECASE DIAGRAM FOR THE ADMIN MODULE:**

****

In the Admin module he can view all the information, send solutions to the farmers through mailing and through sms.

**USECASE DIAGRAM FOR THE EXPERT MODULE :**



In this module the Expert adds information required for the farmer.

**CLASS DIAGRAM :**

A class diagram is graphical presentation of the static view that shows a collection of static model elements such as classes, types and their contents and relationships apart from interfaces and collaborations also. Class diagram contains class name, class attributes, operations or methods

**CLASS DIAGRAM FOR WHOLE SYSTEM:**

****

This diagram shows classes and corresponding objects their properties and methods used in the project. Classes present in this system are Farmer , Admin, Expert .

**SEQUENCE DIAGRAM :**

A sequence diagram emphasizes the time ordering of messages. A sequence diagram is a special kind of interaction diagram. Sequence diagrams have two features that distinguish them from collaboration diagram.

First there is the object lifeline. An object is the vertical dashed line that represents the existence of an object over a period of time. Most objects that appear in an interaction diagram will be in existence for some duration, so these objects are aligned at the top of the diagram, with their lifelines drawn from the top of the diagram to bottom.

Second there is focus of control is a tall thin rectangle that shows the period of time during which an object is performing an action either directly or through a subroutine procedure

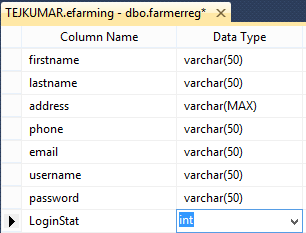
**SEQUENCE DIAGRAM FOR WHOLE SYSTEM :**



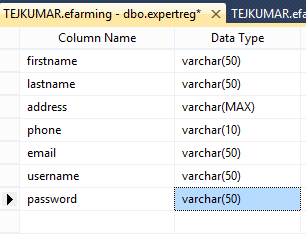
This provides the sequence of operations performed by the user and server at the time of login. This includes how the operations are done in a sequence to get appropriate results.

**5.2 DATABASE DESIGN :**

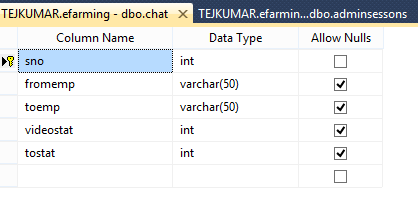
**Farmer Registration :**

****

**Expert Registration :**

****

**Chat :**

****

**6. SYSTEM IMPLEMENTATION**

**6.1 INTRODUCTION**

**Attributes**

The attributes of an element are name-value pairs, separated by "=", and written within the start label of an element, after the element's name. The value should be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML).Leaving attribute values unquoted is considered unsafe.

Most elements take any of several common attributes: id, class, style and title. Most also take language-related attributes: Lang and dir.

The id attribute provides a document-wide unique identifier for an element. This can be used by style sheets to provide presentational properties, by browsers to focus attention on the specific element or by scripts to alter the contents or presentation of an element. The class attribute provides a way of classifying similar elements for presentation purposes. For example, an HTML document (or a set of documents) may use the designation class="notation" to indicate that all elements with this class value are all subordinate to the main text of the document (or documents). Such notation classes of elements might be gathered together and presented as footnotes on a page, rather than appearing in the place where they appear in the source HTML.

An author may use the style non-attributable codes presentational properties to a particular element. It is considered better practice to use an element’s son- id page and select the element with a style sheet, though sometimes this can be too cumbersome for a simple ad hoc application of styled properties. The title is used to attach sub textual explanation to an element. In most browsers this title attribute is displayed as what is often referred to as a tooltip. The generic inline span element can be used to demonstrate these various non-attributes.

The preceding displays as HTML (pointing the cursor at the abbreviation should display the title text in most browsers).

**Advantages**

* A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
* HTML is platform independent.
* HTML tags are not case-sensitive.

**.NET**

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.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. We can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and Remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

**Features of .NET**

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There’s no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

“.NET” is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

**.NET framework**

The .NET Framework has two main parts:

* The Common Language Runtime (CLR).
* A hierarchical set of class libraries.

The CLR is described as the “execution engine” of .NET. It provides the environment within which programs run. The most important features are

* Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
* Memory management, notably including garbage collection.
* Checking and enforcing security restrictions on the running code.
* Loading and executing programs, with version control and other such features.
* The following features of the .NET framework are also worth description:

**Managed Code**

The code that targets .NET, and which contains certain extra Information - “metadata” - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

**Managed Data**

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you’re using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn’t get garbage collected but instead is looked after by unmanaged code.

**Common Type System**

The CLR uses something called the Common Type System to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn’t attempt to access memory that hasn’t been allocated to it.

**Common Language Specification**

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

**The class library**

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary.

The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity.

The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

**Languages supported by .net**

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications and XML Web services. The .NET framework supports new versions of Microsoft’s old favorites Visual Basic and C++ (as VB.NET and Managed C++), but there are also a number of new additions to the family.

Visual Basic .NET has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others. Visual Basic also now supports structured exception handling, custom attributes and also supports multi-threading.

Visual Basic .NET is also CLS compliant, which means that any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET.

Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new .NET Framework.

C# is Microsoft’s new language. It’s a C-style language that is essentially “C++ for Rapid Application Development”. Unlike other languages, its specification is just the grammar of the language. It has no standard library of its own, and instead has been designed with the intention of using the .NET libraries as its own.

Microsoft Visual J# .NET provides the easiest transition for Java-language developers into the world of XML Web Services and dramatically improves the interoperability of Java-language programs with existing software written in a variety of other programming languages.

Active State has created Visual Perl and Visual Python, which enable .NET-aware applications to be built in either Perl or Python. Both products can be integrated into the Visual Studio .NET environment. Visual Perl includes support for Active State’s Perl Dev Kit.

Other languages for which .NET compilers are available include

* FORTRAN
* COBOL
* Eiffel

|  |  |
| --- | --- |
| ASP.NET  XML WEB SERVICES | Windows Forms |
| Base Class Libraries | |
| Common Language Runtime | |
| Operating System | |

Fig1 **.**Net Framework

C#.NET is also compliant with CLS (Common Language Specification) and supports structured exception handling. CLS is set of rules and constructs that are supported by the CLR (Common Language Runtime). CLR is the runtime environment provided by the .NET Framework; it manages the execution of the code and also makes the development process easier by providing services.

C#NET is a CLS-compliant language. Any objects, classes, or components that created in C#.NET can be used in any other CLS-compliant language. In addition, we can use objects, classes, and components created in other CLS-compliant languages in C#.NET .The use of CLS ensures complete interoperability among applications, regardless of the languages used to create the application.

**CONSTRUCTORS AND DESTRUCTORS:**

Constructors are used to initialize objects, whereas destructors are used to destroy them. In other words, destructors are used to release the resources allocated to the object. In C#.NET the sub finalize procedure is available. The sub finalize procedure is used to complete the tasks that must be performed when an object is destroyed. The sub finalize procedure is called automatically when an object is destroyed. In addition, the sub finalize procedure can be called only from the class it belongs to or from derived classes.

**GARBAGE COLLECTION**

Garbage Collection is another new feature in C#.NET. The .NET Framework monitors allocated resources, such as objects and variables. In addition, the .NET Framework automatically releases memory for reuse by destroying objects that are no longer in use.

In C#.NET, the garbage collector checks for the objects that are not currently in use by applications. When the garbage collector comes across an object that is marked for garbage collection, it releases the memory occupied by the object.

**OVERLOADING**

Overloading is another feature in C#. Overloading enables us to define multiple procedures with the same name, where each procedure has a different set of arguments. Besides using overloading for procedures, we can use it for constructors and properties in a class.

**MULTITHREADING:**

C#.NET also supports multithreading. An application that supports multithreading can handle multiple tasks simultaneously, we can use multithreading to decrease the time taken by an application to respond to user interaction.

**THE .NET FRAMEWORK**

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet.

**OBJECTIVES OF.NET FRAMEWORK**

* To provide a consistent object-oriented programming environment whether object codes is stored and executed locally on Internet-distributed, or executed remotely.
* To provide a code-execution environment to minimizes software deployment and guarantees safe execution of code.
* Eliminates the performance problems.

There are different types of application, such as Windows-based applications and Web-based applications.

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services

SQL-SERVER database consist of six types of objects. They are,

1. TABLE

2. QUERY

3. FORM

4. REPORT

5. MACRO

**TABLE:**

A database is a collection of data about a specific topic.

**VIEWS OF TABLE:**

We can work with a table in two types,

* Design View
* Datasheet View

**Design View:**

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

**Datasheet View**

To add, edit or analyses the data itself we work in tables datasheet view mode.

**QUERY:**

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot (it cannot be edited).Each time we run query, we get latest information in the dynaset. Access either displays the dynaset or snapshot for us to view or perform an action on it, such as deleting or updating.

**Active server pages.net**

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web development models:

* **Enhanced Performance.** ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.
* **World-Class Tool Support.** The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.
* **Power and Flexibility.** Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.
* **Simplicity.** ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.
* **Manageability.** ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications. Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools. This "zero local administration" philosophy extends to deploying ASP.NET Framework applications as well. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.

**What is ASP.NET Web Forms?**

The ASP.NET Web Forms page framework is a scalable common language runtime programming model that can be used on the server to dynamically generate Web pages.

Intended as a logical evolution of ASP (ASP.NET provides syntax compatibility with existing pages), the ASP.NET Web Forms framework has been specifically designed to address a number of key deficiencies in the previous model. In particular, it provides:

* The ability to create and use reusable UI controls that can encapsulate common functionality and thus reduce the amount of code that a page developer has to write.
* The ability for developers to cleanly structure their page logic in an orderly fashion (not "spaghetti code").
* The ability for development tools to provide strong WYSIWYG design support for pages (existing ASP code is opaque to tools).

**Code-Behind Web Forms**

ASP.NET supports two methods of authoring dynamic pages. The first is the method shown in the preceding samples, where the page code is physically declared within the originating.aspx file. An alternative approach--known as the code-behind method--enables the page code to be more cleanly separated from the HTML content into an entirely separate file.

**Introduction to ASP.NET Server Control:**

In addition to (or instead of) using <% %> code blocks to program dynamic content, ASP.NET page developers can use ASP.NET server controls to program Web pages. Server controls are declared within an.aspx file using custom tags or intrinsic HTML tags that contain a **runat="server"** attribute value. Intrinsic HTML tags are handled by one of the controls in the **System.Web.UI.HtmlControls** namespace. Any tag that doesn't explicitly map to one of the controls is assigned the type of **System.**

**Web.UI.HtmlControls.HtmlGenericControl**.

Server controls automatically maintain any client-entered values between round trips to the server. This control state is not stored on the server (it is instead stored within an **<input type="hidden">** form field that is round-tripped between requests). Note also that no client-side script is required.

In addition to supporting standard HTML input controls, ASP.NET enables developers to utilize richer custom controls on their pages. For example, the following sample demonstrates how the **<asp:adrotator>** control can be used to dynamically display rotating ads on a page.

* ASP.NET Web Forms provide an easy and powerful way to build dynamic Web UML. ASP.NET Web Forms pages can target any browser client (there are no script library or cookie requirements).
* ASP.NET Web Forms pages provide syntax compatibility with existing ASP pages.
* ASP.NET server controls provide an easy way to encapsulate common functionality.
* ASP.NET ships with 45 built-in server controls. Developers can also use controls built by third parties.
* ASP.NET server controls can automatically project both up-level and down-level HTML.
* ASP.NET templates provide an easy way to customize the look and feel of list server controls.
* ASP.NET validation controls provide an easy way to do declarative client or server data validation.

**Database:**

A database management system (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems.

Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

**Description**

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. A DBMS includes:

* A modeling language to define the schema of each database hosted in the DBMS, according to the DBMS data model.
* The four most common types of organizations are the hierarchical, network, relational and object models. Inverted lists and other methods are also used. A given database management system may provide one or more of the four models. The optimal structure depends on the natural organization of the application's data, and on the application's requirements (which include transaction rate (speed), reliability, maintainability, scalability, and cost).
* The dominant model in use today is the ad hoc one embedded in SQL, despite the objections of purists who believe this model is a corruption of the relational model, since it violates several of its fundamental principles for the sake of practicality and performance. Many DBMSs also support the Open Database Connectivity API that supports a standard way for programmers to access the DBMS.
* Data structures (fields, records, files and objects) optimized to deal with very large amounts of data stored on a permanent data storage device (which implies relatively slow access compared to volatile main memory).
* A database query language and report writer to allow users to interactively interrogate the database, analyze its data and update it according to the users privileges on data.
* It also controls the security of the database.
* Data security prevents unauthorized users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of it called subschemas. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and medical data.
* If the DBMS provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. However, it may not leave an audit trail of actions or provide the kinds of controls necessary in a multi-user organization. These controls are only available when a set of application programs are customized for each data entry and updating function.
* A transaction mechanism, that ideally would guarantee the ACID properties, in order to ensure data integrity, despite concurrent user accesses (concurrency control), and faults (fault tolerance).
* It also maintains the integrity of the data in the database.
* The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time. The DBMS can help prevent duplicate records via unique index constraints; for example, no two customers with the same customer numbers (key fields) can be entered into the database. See ACID properties for more information (Redundancy avoidance).

The DBMS accepts requests for data from the application program and instructs the operating system to transfer the appropriate data. When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. New categories of data can be added to the database without disruption to the existing system.

Organizations may use one kind of DBMS for daily transaction processing and then move the detail onto another computer that uses another DBMS better suited for random inquiries and analysis. Overall systems design decisions are performed by data administrators and systems analysts. Detailed database design is performed by database administrators.

Database servers are specially designed computers that hold the actual databases and run only the DBMS and related software. Database servers are usually multiprocessor computers, with RAID disk arrays used for stable storage. Connected to one or more servers via a high-speed channel, hardware database accelerators are also used in large volume transaction processing environments.

DBMSs are found at the heart of most database applications. Sometimes DBMSs are built around a private multitasking kernel with built-in networking support although nowadays these functions are left to the operating system.**SQL**

Structured Query Language (SQL) is the language used to manipulate relational databases. SQL is tied very closely with the relational model.

In the relational model, data is stored in structures called relations or tables*.*

SQL statements are issued for the purpose of:

**Data definition:**

Defining tables and structures in the database (DDL used to create, alter and drop schema objects such as tables and indexes).

**Data manipulation:**

Used to manipulate the data within those schema objects (DML Inserting, Updating, Deleting the data, and Querying the Database).

A schema is a collection of database objects that can include: tables, views, indexes and sequences

List of SQL statements that can be issued against an Oracle database schema are:

1. **ALTER** - Change an existing table, view or index definition (DDL)
2. **AUDIT** - Track the changes made to a table (DDL)
3. **COMMENT** - Add a comment to a table or column in a table (DDL)
4. **COMMIT** - Make all recent changes permanent (DML - transactional)
5. **CREATE** - Create new database objects such as tables or views (DDL)
6. **DELETE**- Delete rows from a database table (DML)
7. **DROP** - Drop a database object such as a table, view or index (DDL)
8. **GRANT** - Allow another user to access database objects such as tables or views (DDL)
9. **INSERT** - Insert new data into a database table (DML)
10. **No AUDIT** - Turn off the auditing function (DDL)
11. **REVOKE** - Disallow a user access to database objects such as tables and views (DDL)
12. **ROLLBACK** - Undo any recent changes to the database (DML - Transactional)
13. **SELECT** - Retrieve data from a database table (DML)
14. **TRUNCATE** - Delete all rows from a database table (can not be rolled back) (DML)
15. **UPDATE**- Change the values of some data items in a database table (DML)

**Database design:**

After carefully understanding the requirements the entire data storage requirements are divided into tables. The tables are normalized to avoid any anomalies during the course of data entry.

**Normalization:**

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, update, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relation.

Insertion anomaly: Inability to add data to the database due to absence of other data.

Deletion anomaly: Unintended loss of data due to deletion of other data.

Update anomaly: Data inconsistency resulting from data redundancy and partial update

Normal Forms: These are the rules for structuring relations that eliminate anomalies.

**First Normal Form:**

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

**Second Normal Form**:

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

* Primary key is a not a composite primary key
* No non key attributes are present
* Every non key attribute is fully functionally dependent on full set of primary key.

**Third Normal Form**:

A relation is said to be in third normal form if their exits no transitive dependencies.

Transitive Dependency: If two non key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

**SQL SERVER**

SQL SERVER is one of the leading database management systems (DBMS) because it is the only database that meets the uncompromising requirements of today’s most demanding information systems. SQL SERVER is a truly portable, distributed, and open DBMS that delivers unmatched performance, continuous operation and support for every database.SQL SERVER RDBMS is high performance fault tolerant DBMS which is specially designed for online transactions processing and for handling large database application.

SQL SERVER is a truly portable, distributed, and open DBMS that delivers unmatched performance, continuous operation and support for every database.

SQL SERVER RDBMS is high performance fault tolerant DBMS which is specially designed for online transactions processing and for handling large database application

**Enterprise wide Data Sharing**

The unrivaled portability and connectivity of the SQL SERVER DBMS enables all the systems in the organization to be linked into a singular, integrated computing resource

**Portability**

SQL SERVER is fully portable to more than 80 distinct hardware and operating systems platforms, including UNIX, MSDOS, OS/2, Macintosh and dozens of proprietary platforms. This portability gives complete freedom to choose the database server platform that meets the system requirements.

**Open System**

SQL SERVER offers a leading implementation of industry –standard SQL. SQL Server’s open architecture integrates SQL SERVER and non –SQL SERVER DBMS with industry’s most comprehensive collection of tools, application, and third party software products SQL Server’s Open architecture provides transparent access to data from other relational database and even non-relational database.

**Sophisticated Concurrency Control**

Real World applications demand access to critical data. With most database Systems application becomes “contention bound” – which performance is limited not by the CPU power or by disk I/O, but user waiting on one another for data access. SQL Server employs full, unrestricted row-level locking and contention free queries to minimize and in many cases entirely eliminates contention wait times.

**No I/O Bottlenecks**

SQL Server’s fast commit groups commit and deferred write technologies dramatically reduce disk I/O bottlenecks. While some database write whole data block to disk at commit time, SQL Server commits transactions with at most sequential log file on disk at commit time, On high throughput systems, one sequential writes typically group commit multiple transactions. Data read by the transaction remains as shared memory so that other transactions may access that data without reading it again from disk. Since fast commits write all data necessary to the recovery to the log file, modified blocks are written back to the database independently of the transaction commit, when written from memory to disk.

**6.2 MODULES DESCRIPTION**

**MODULES**

* Admin
* Expert
* Farmer

**Admin :**

In the module after admin logins he has his own menu like viewing all the farmers registered and information about all the experts registered. He can add the information based on the soils and seasons he can participate in videoconferencing with the farmers. He can view all quries posted by user through mail and sms and can repley to them through mailing and through sms.

**Expert :**

In the module we Expert registration and after Expert registration he has to login. after his login he has his own menu such as adding soil based fertilizers, seasons based fertlizers. He can add the information .

**Farmer :**

In the module we farmer registration and after farmer registration he has to login.after his login he has his own menu such as soil, seasons, expert soil, expert season, mailing, messaging, video conferencing. He can participate in videoconference and he can send mails if he has any queries.

**7. SYSTEM TESTING**

Software testing is the process used to assess the quality of Computer Software. Software testing is an empirical technical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. This includes the process of executing a program or application with the intent of finding

Over its existence, computer software has continued to grow in complexity and size. Every software product has a target audience. For example, a video game software has its audience completely different from banking software. Therefore, when an organization develops or otherwise invests in a software product, it presumably must assess whether the software product will be acceptable to its end users, its target audience, its purchasers, and other stakeholders. Software testing is the process of attempting to make this assessment.

Different levels of testing that have to be conducted are

* Unit Testing
* Integration Testing
* System Testing
* Implementation & Maintenance Testing

**7.1 UNIT TESTING:**

Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design and process specifications testing is done to uncover errors within the boundary of the module. All the modules must be successful in the unit test before the start of the integration testing begins.

**7.2 INTEGRATION TESTING:**

After the unit testing we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interaction.

**7.3 SYSTEM TESTING:**

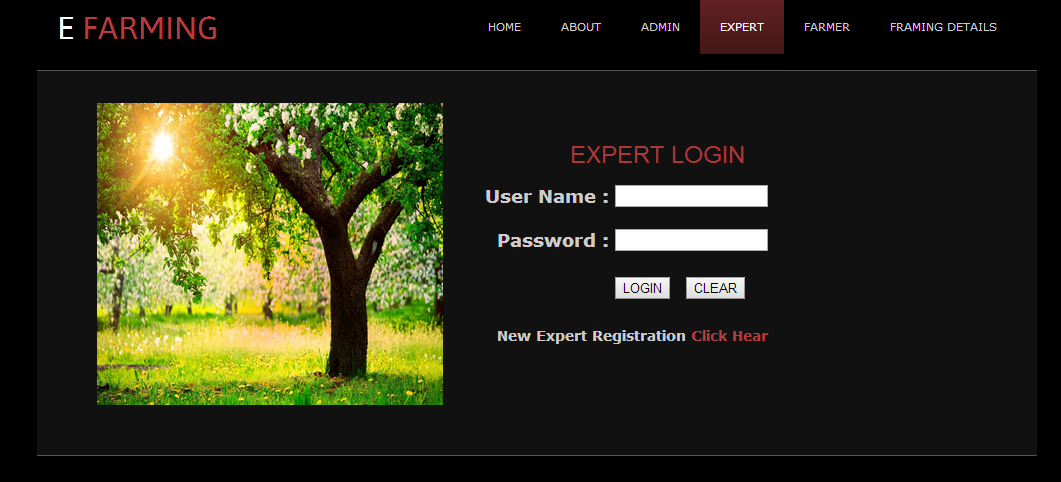
Here the entire software system is tested. The reference document for this process is the requirements document, and the goal OS to see a software meets its requirements.

**7.4 IMPLEMENTATION AND MAINTENANCE TESTING:**

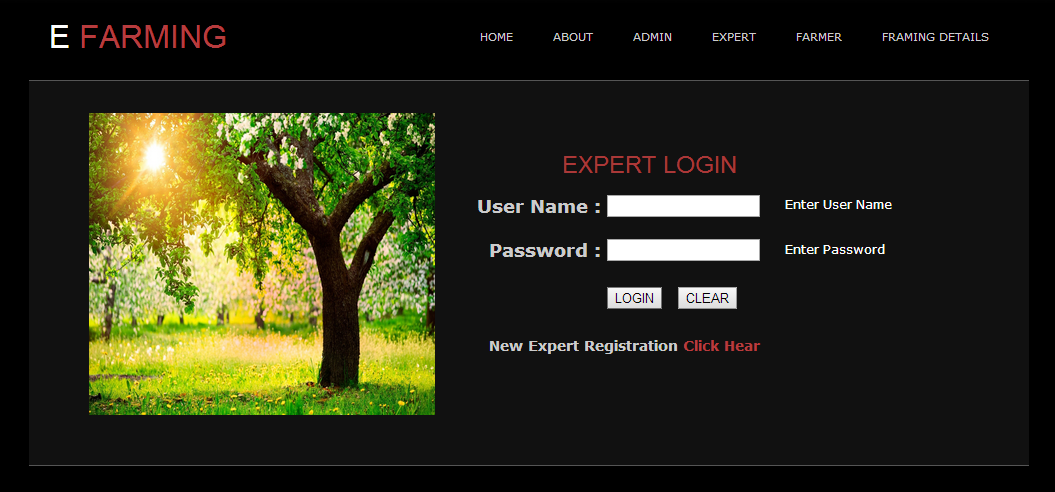
Maintenance testing is that testing which is performed to either identify equipment problems, diagnose equipment problems or to confirm that repair measures has been effective. It can be performed at either the system level, the equipment level, or the component level.

**TEST SCREENS**

**LOGIN PAGE :**

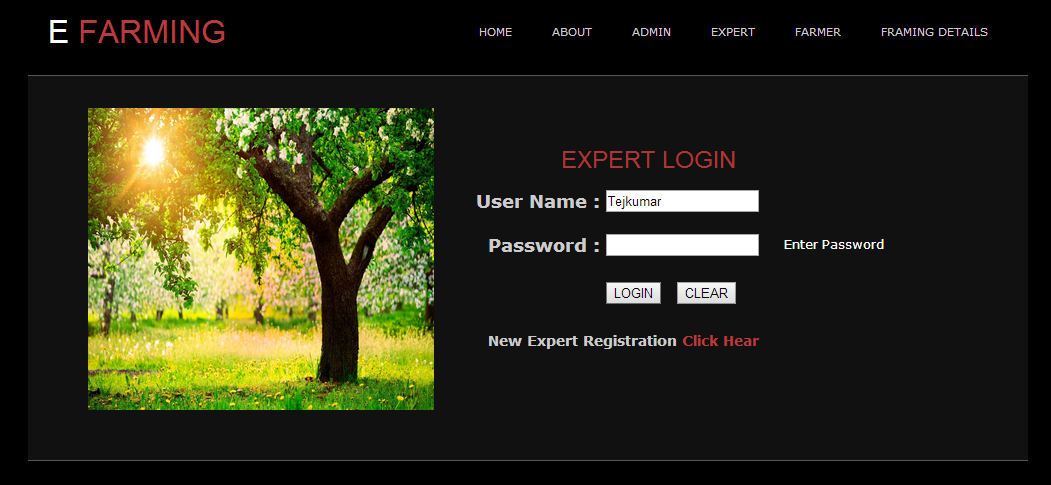
****

**ERROR PAGES :**



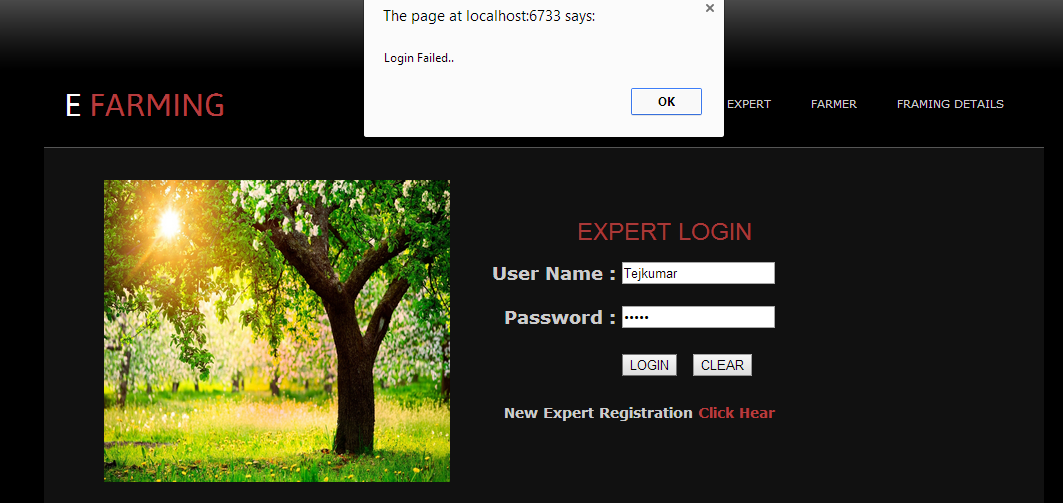
Description :

Need to enter both username and password otherwise above error message will occur.



Description :

If password is wrong then it displays an error message.



Description :

If either username or password is incorrect then the above dialog box appears.

**TEST PLAN:**

Testing commence with a test plan and terminates with acceptance testing. Test plan is a general document for the entire project that defines the scope, approach to be taken and the schedule of testing as well as identifies the test item for the entire testing process and the personal responsible for the different activities of testing. The test planning can be done in parallel with the coding and design phases. The inputs forming the test plan are

* Project plan
* Requirements document
* System design document

The project plan is need to make sure that the test plan is consistent with the overall plan for the project and the testing schedule matches that of the project plan. The requirements document and the design document are the basic documents used for selecting the test units and deciding the approaches to be use during testing. A test plan should contain the following:

* Test unit specification
* Features to be tested
* Approaches for testing
* test deliverables
* Schedule
* Personal allocation

One of the most important activities of test plan is to identify the test units. The test unit is a set of one or more modules together with associated date that are from single computer program and that are objectives of testing.

A test unit can occur at any level and can contain from a single module to the entire system. Thus a test unit may be a module, a few modules or complete activities of testing.

A test plan should contain the following

**TEST UNIT SPECIFICATION**

A test unit is a set of one or more modules with associated date which are from a single program and which are the object of testing. Test unit may be module, a few modules or a complete program. Different units are specified for unit integration and system testing.

**FEATURES TO BE TESTED**

Features to be tested include all software features and combination of features that should be tested. A software feature is software characteristics specified or simplified by the requirements of design documents. These may include functionality, performance, design constraints and attributes.

All the functional features specified in requirements document will be tested. No testing will be done for the performance.

**APPROACH FOR TESTING**

Approach for testing specifies the overall approach to be followed in the current project. This is sometimes called testing criteria.

**Test deliverable:**

Testing deliverables should be specified the test plan, before the actual testing begins. Test summary report, test log and data about the code coverage.

**Schedule:**

The test log provides chronological record of relevant details about the execution of test case. Different activities of testing and testing of different units that has identified.

**Personal Allocation:**

Personal allocation identifies the person responsible for performing the different activities

**8. SCREEN SHOTS**

**HOMEPAGE :**

****

**Desription:**

This is the home page of the system.When project is executed the above page is loaded.

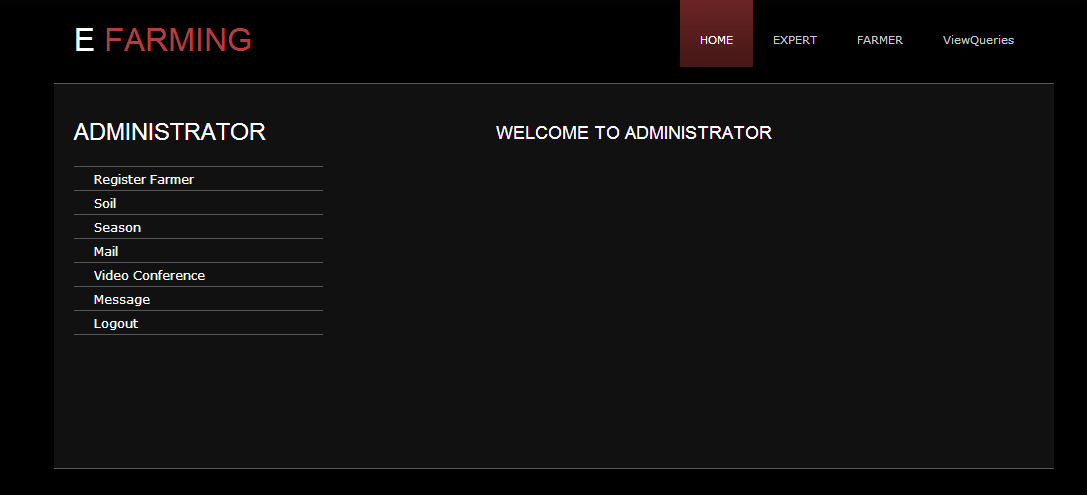
**ABOUT PAGE :**



**Desription:**

If you click on ABOUT, the above page appears which gives the description of E-Farming.

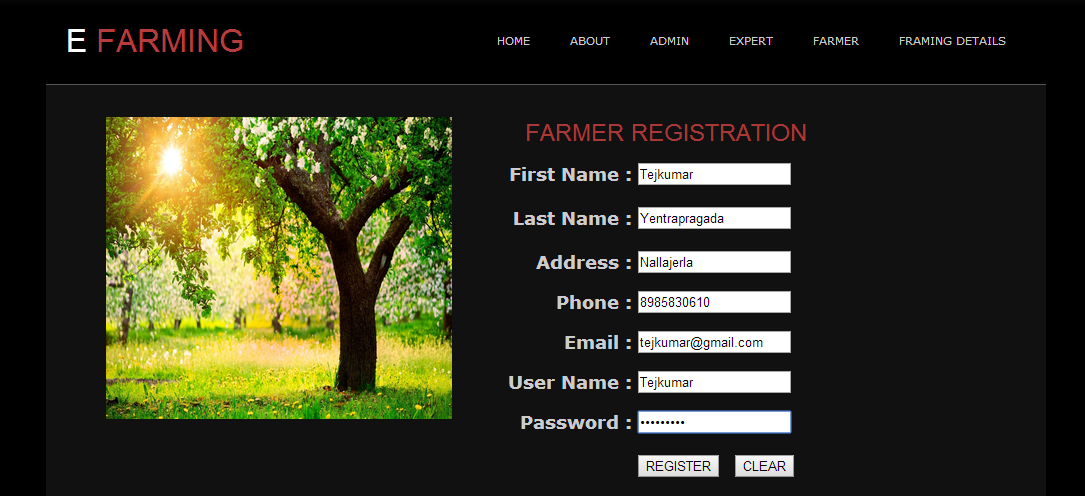
**ADMIN HOME PAGE**  :

****

**Desription:**

When Admin login the above page appears and he has his own menu.

**REGISTER FARMER :**



**Description:**

The Farmer Registers in this page and after registering login page appears.

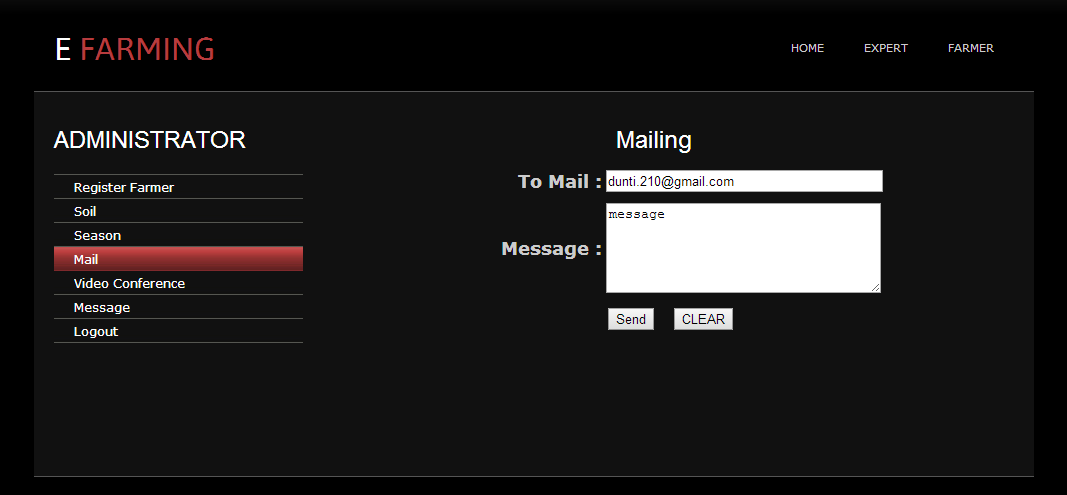
**ADMIN ADDING SEASON INFORMATION :**



**Description :**

Admin adds the information about the fertilizers based soil and type of crop

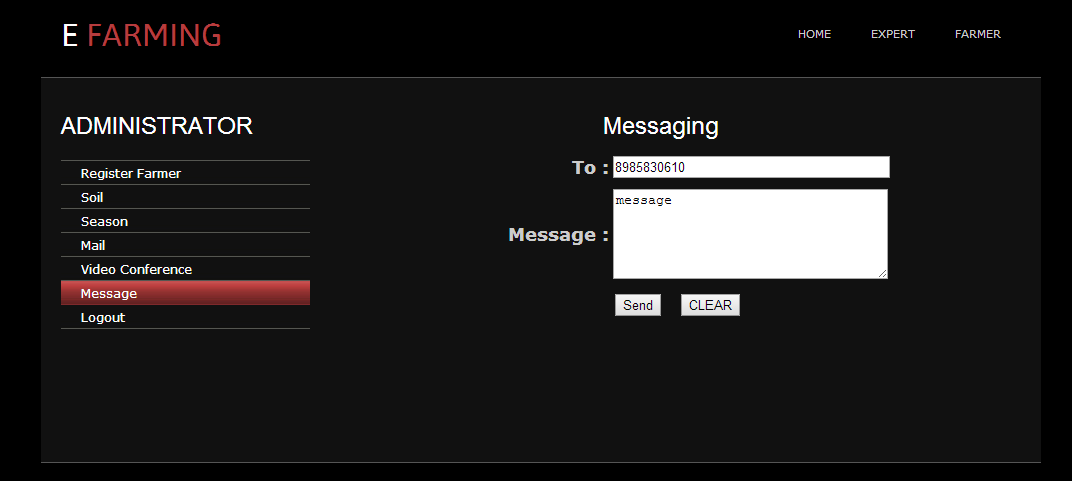
**ADMIN MAILING PAGE :**

****

**Description :**

Admin sends mail to the farmer based on the query farmer send.

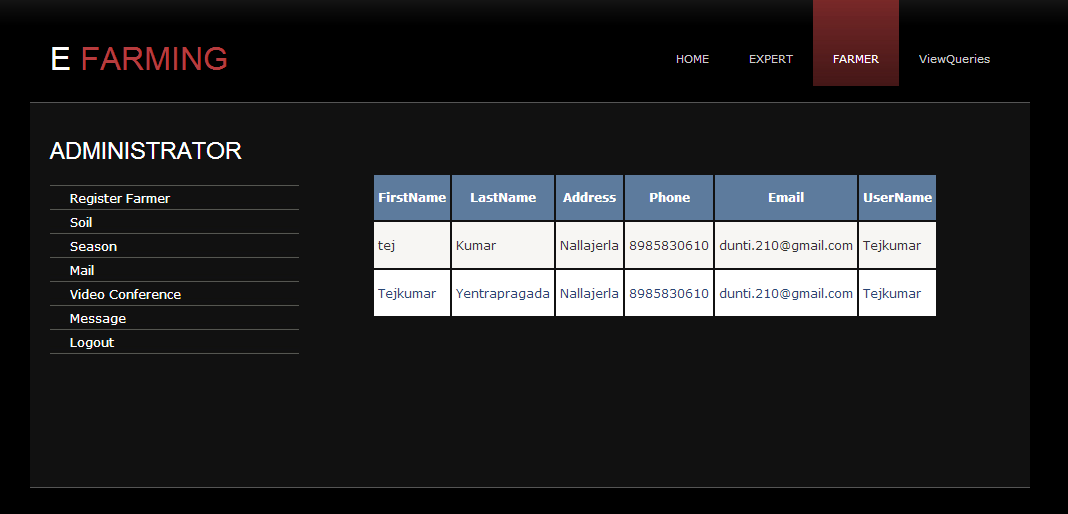
**ADMIN MESSAGING PAGE :**



Description :

Admin sends messages to the farmer based on the query farmer send.

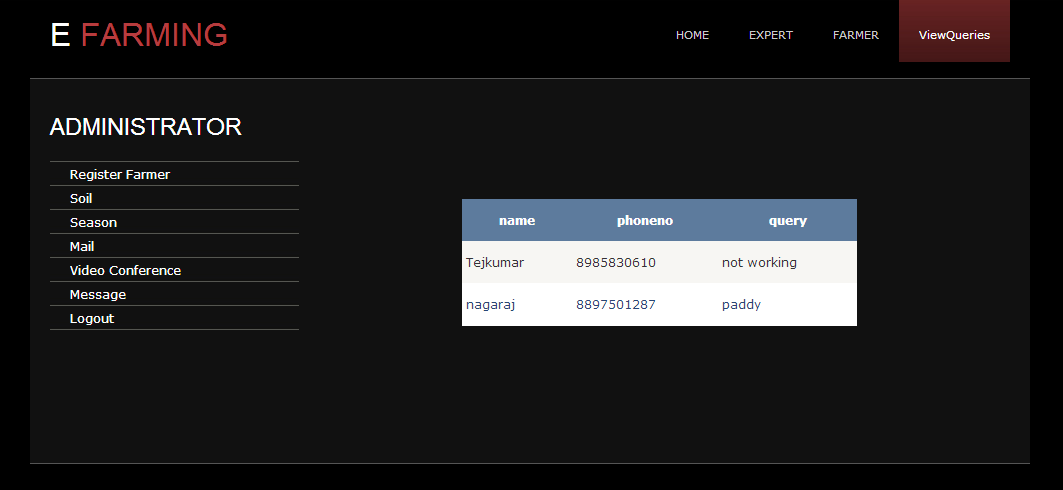
**ADMIN VIEWING FARMER INFORMATION :**



Description :

Admin can view the farmer details who registered.

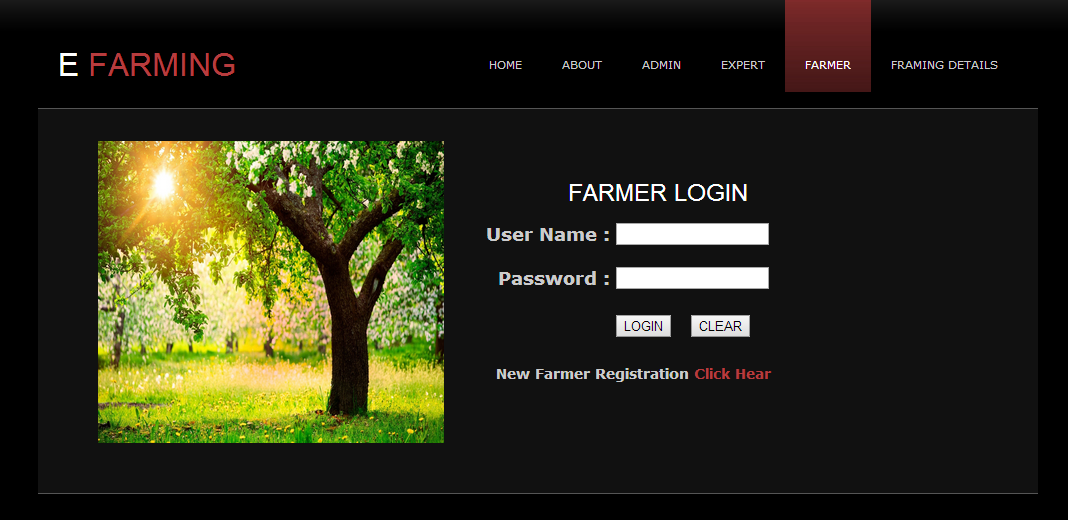
**ADMIN VIEWING QUERIES :**

****

Description :

Admin can view the queries posted by the farmers.

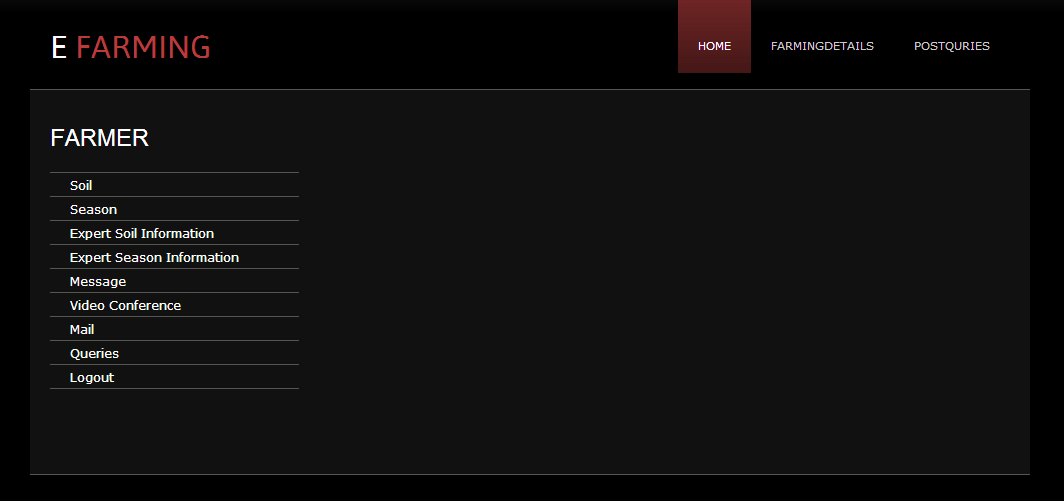
**FARMER LOGIN PAGE :**



**Desription:**

After Registration the above page appears where farmer login and check the details.

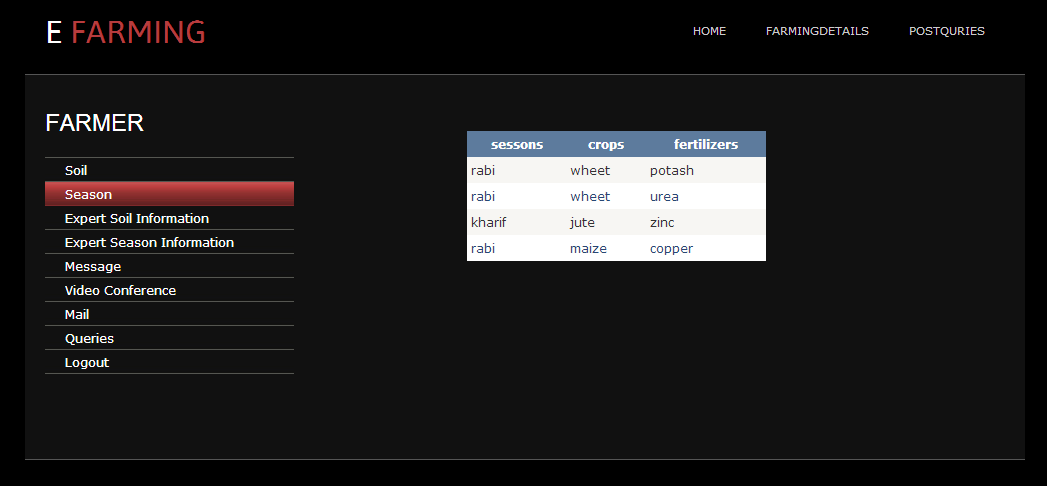
**FARMER MENU PAGE :**

****

Description :

After login farmer has the above menu by which he can perform specific tasks.

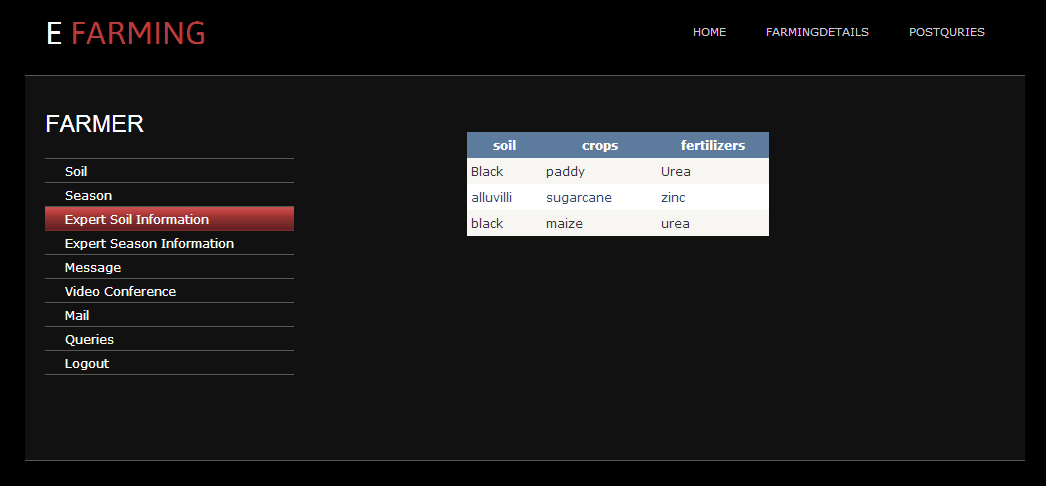
**SEASON PAGE :**



Description :

Farmer see the information regarding the seasons.

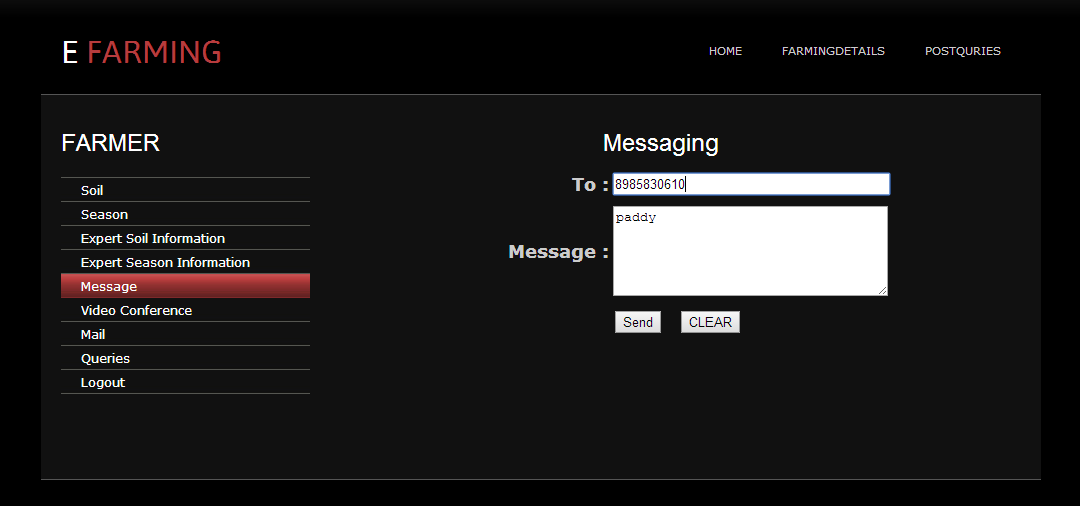
**EXPERT SOIL INFORMATION :**



Description :

Farmer see the information regarding the soils added by the experts.

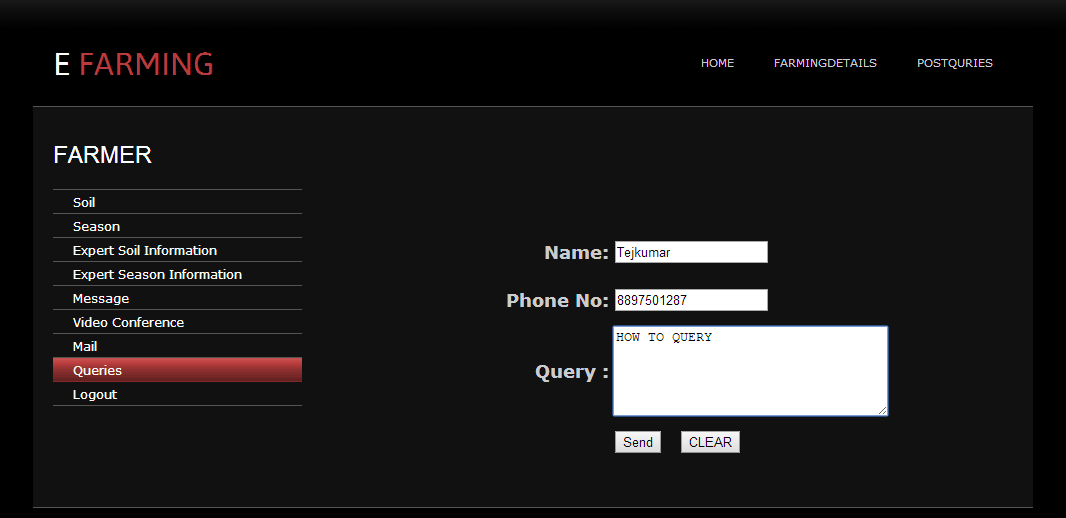
**MESSAGE :**



Description :

Farmer can send message to the expert/admin in case of queries.

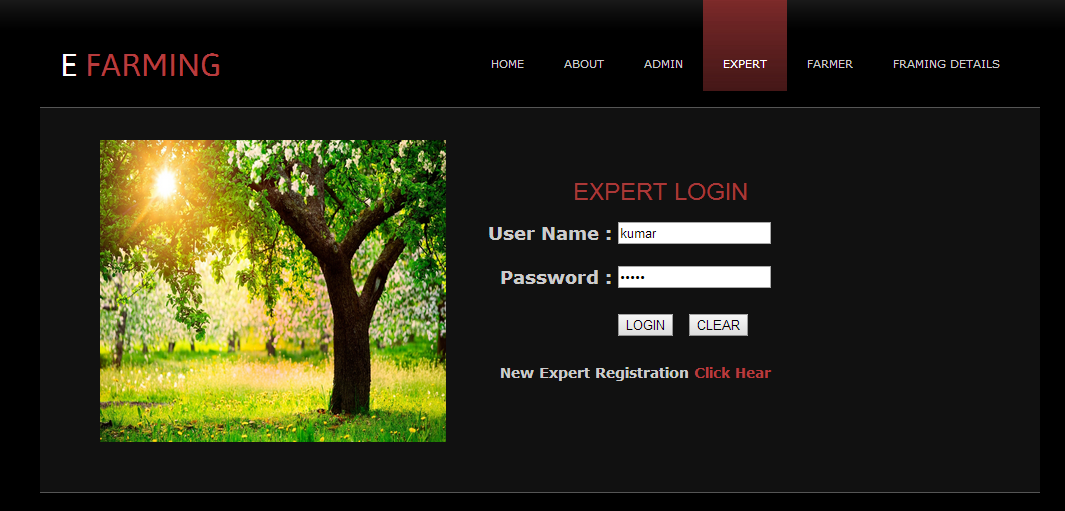
**QUERIES :**

****

Description :

Farmer can post queries that he has to get the solutions.

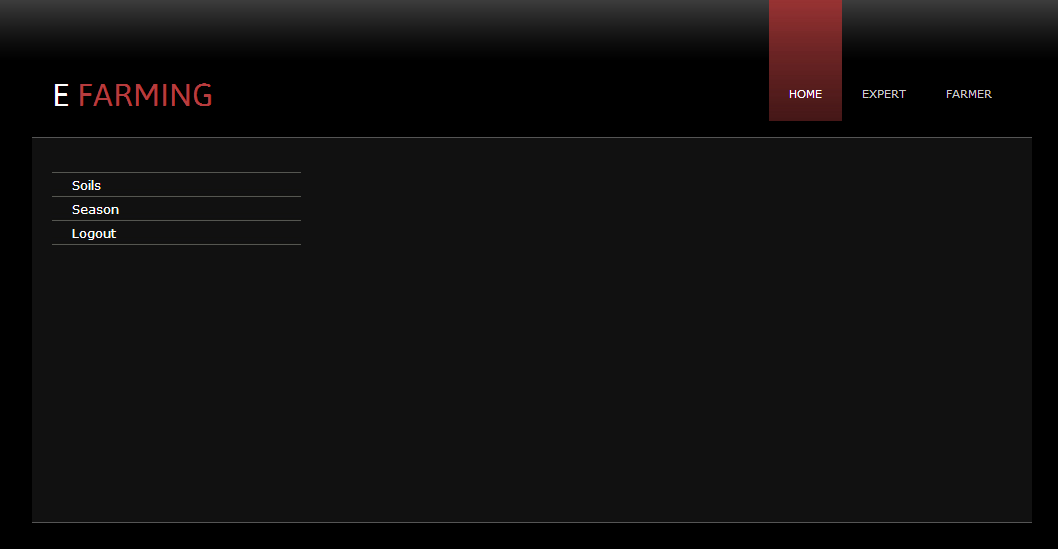
**EXPERT LOGIN :**

****

Description :

Expert login here and if new expert then he may click on click here.

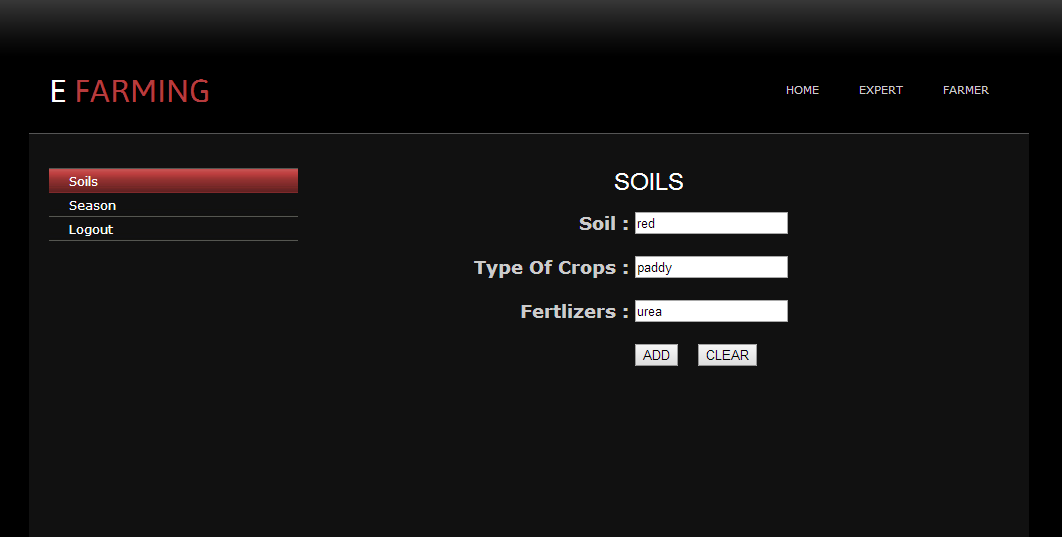
**EXPERT HOME:**

****

Description :

When expert login he will be redirected to the above page.

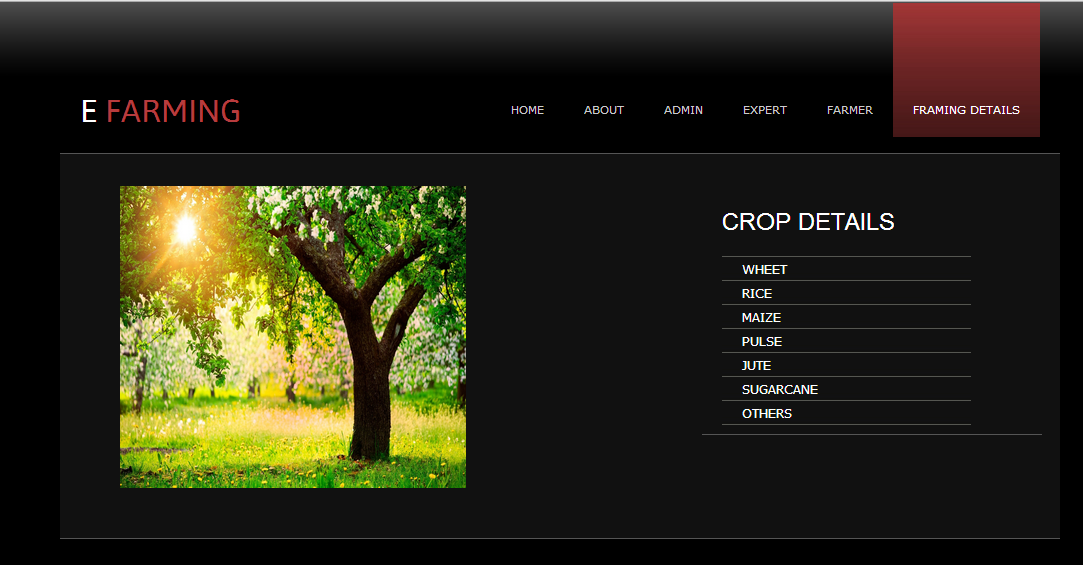
**EXPERT ADDED SOIL INFORMATION :**

****

Description :

Expert adds the soil information based on recent inventions done on farming.

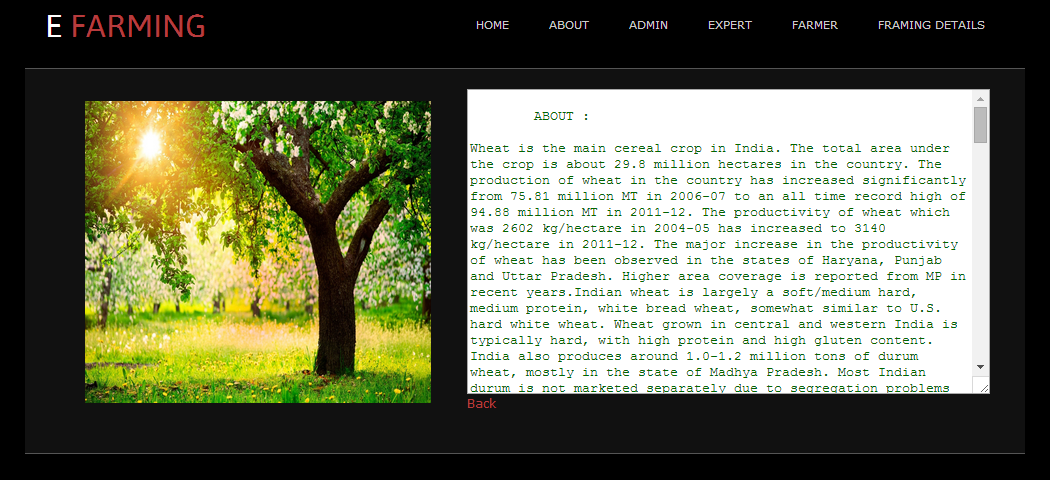
**FARMING DETAILS :**

****

Description :

Here all the required information for the farmers is available.

**WHEET :**

****

Description :

Gives details about the wheet.

**CONCLUSION**

E-Farming is a web based application designed to guide the farmers in order to improve the yield. Using this application the farmer can view any details regarding the cultivation and can post queries related to his crop cultivation.

As technology is playing a vital role these days, farmers can view any type of details related to his crop. Using video conferencing the farmer can directly talk to the admin and can have detailed description about his doubts.

**FUTURE ENHANCEMENT**

* On validation of the login one-time password can be generated and send to user’s cell phone as an SMS.
* Automatic message can be send to the farmers based on the date they entered.

**BIBLIOGRAPHY**

**URLs:**

[www.support.microsoft.com](http://www.support.microsoft.com)

[www.develop+er.com](http://www.develop+er.com)

[www.msdn.microsoft.com](http://www.msdn.microsoft.com)

[www.msdn.microsoft.com/net/quickstart/aspplus/default.com](http://www.msdn.microsoft.com/net/quickstart/aspplus/default.com)

[www.asp.net](http://www.asp.net)

[www.fmexpence.com/quickstart/applus/default.com](http://www.fmexpence.com/quickstart/applus/default.com)

[www.asptoday.com](http://www.asptoday.com)

[www.aspfree.com](http://www.aspfree.com)

[www.4guysfromrola.com/index.aspx](http://www.4guysfromrola.com/index.aspx)

[www.a.spnextgen.com/index.aspx](http://www.a.spnextgen.com/index.aspx)

**Books :**

Database and System Design by Korth

Software Engineering Concepts by Robert Pressman

**APPENDIX**

**How to install sql server**

* Double click on SQL server setup file,
* It will configuring all components
* It scan your computer configurations
* Then the sql server installation wizard will be appear,click Next button.It copies all the files
* Give the registration information like name,and company details then click next
* Select all components to install and click next.
* Select Default instance radio button option and click next.
* Then select the service account details here click on “Use the built in system account” and select all check boxes and click next button
* Then authentication mode window will be appear here select “Mixed mode radio button and enter the password it is by default “123” only. Then click next button
* Click On Next Button until you get the finish --- Rest of all the options are default only

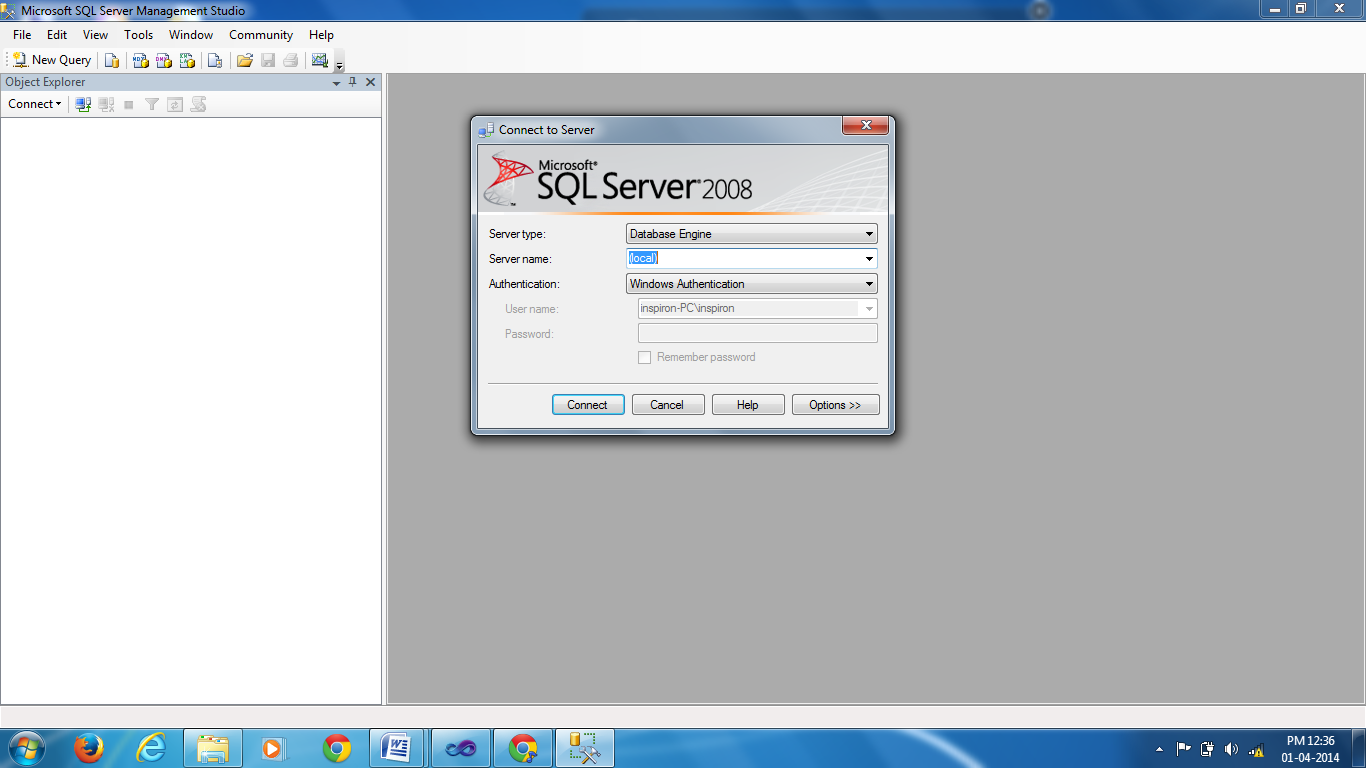


Fig: The final IDE of Sql Server 2008.

When you click on the connect button the sql server data base is connected to your .net

**How to install .net**

* Click on setup file in the .net software.
* It will take several minutes to install, give the name to the .net for the administrator name.
* Then the installing components window will be appear select all components and then click next button until finish button appear.
* Finally the .Net dictionary files are copied into your system then click close button,then the fallowing IDE will be appear which is Microsoft Visual Studio start page.

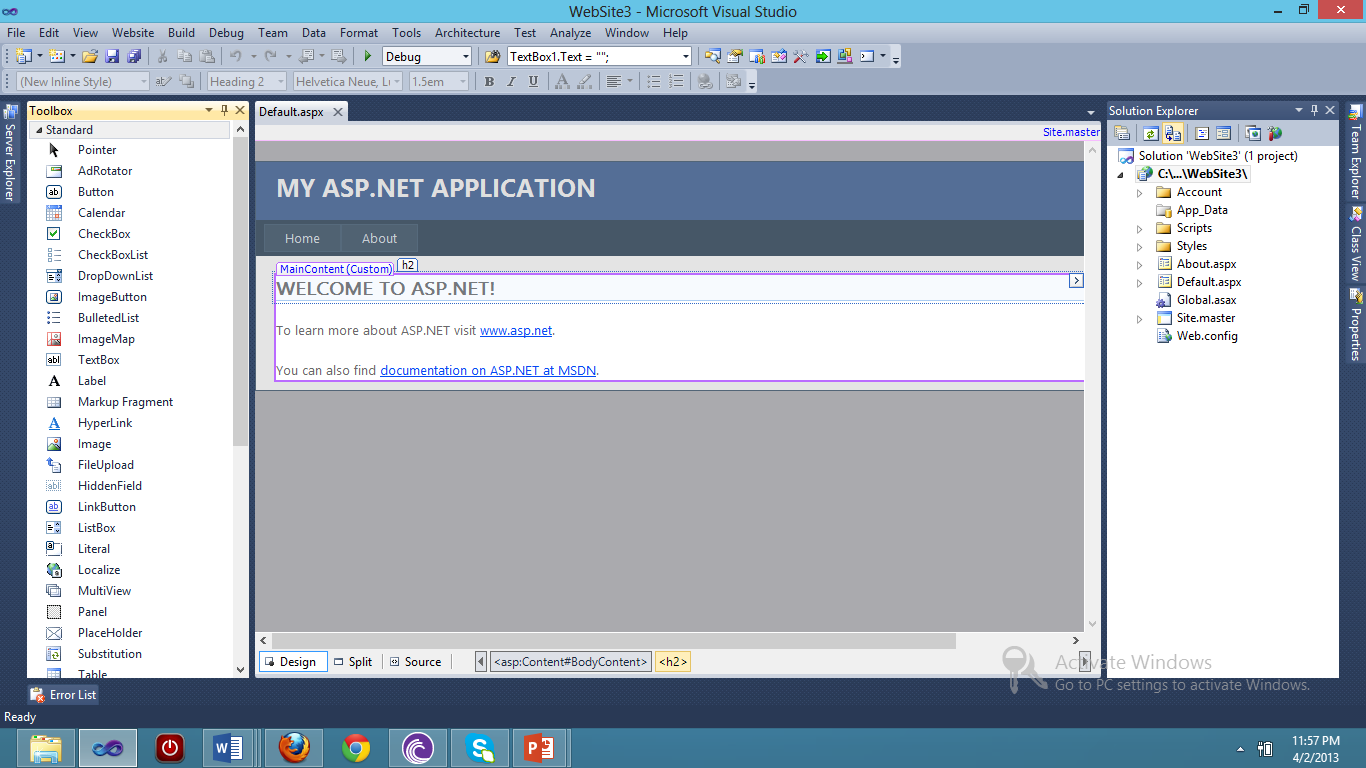


Fig: The Microsoft Visual Studio start page

When you are click on file\New\Website , give the website name,then the empty default website will be appear which has the tool box, solution explorer tabs as shown in the above figure.

**How to deploy the data base files**

* Copy the database files like mdf and ldf files into C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data
* Paste the database files into specific computer C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data
* Select Sqlserver Management Studio Option in start menu
* Connect with Windows Authentication, Click on the connect button
* Select Sqlserver and right click and then select properties
* Select Sqlserver and windows authentication mode radiobutton and click on ok
* Reset the password
* Select status page guest and enabled radio buttons
* Right click on server and select disconnect and then
* Click on connect and select database Engine
* Right click on the database and select attach
* Click on the Add button to attach the mdf files
* Select the relevant .mdf file and click on ok button
* Both the database files are attached just click on ok button
* Select the database and view its tables

**How to execute the project**

* Copy the code from the CD.
* Paste the code at C:\Users\admin\Documents\Visual Studio 2010\Websites
* Open the visual studio 2010
* Click on the file\open
* Select the project on the C:\Users\admin\Documents\Visual Studio 2010\Websites
* Right click on start page on your project click “set as start page” and press F5.