Marathwada Shikshan Prasarak Mandal’s

**Deogiri Institute of Engineering and Management Studies,**

**Aurangabad**

**Project Report**

**On**

**Online Grampanchyat**

Submitted By

**Narayan Dudhe (36204)**

**Shubham Thorat (36201)**

**Nikita Magar (36209)**

**Anjali pagar (36191)**

**Dr. Babasaheb Ambedkar Marathwada University**

**Aurangabad (M.S.)**

****

Department of Computer Science and Engineering

**Deogiri Institute of Engineering and Management Studies,**

**Aurangabad**

(2017- 2018)

**Project Report**

**On**

**Online Grampanchyat**

Submitted By

**Narayan Dudhe (36204)**

**Shubham Thorat (36201)**

**Nikita Magar (36209)**

**Anjali pagar (36191)**

**In partial fulfilment of**

**Bachelor of Engineering**

**(Computer Science & Engineering)**

Project Co-ordinator

**Prof.: Manisha Mundhe**

Department of Computer Science & Engineering

**Deogiri Institute of Engineering and Management Studies,**

**Aurangabad**

(2017- 2018)

**CERTIFICATE**

This is to certify that, the Project entitled **“Online Grampanchyat”** submitted by **Narayan Dudhe, Shubham Thorat, Nikita magar, Anjali Pagar** isa bona fide work completed under my supervision and guidance in partial fulfilment for award of Bachelor of Engineering (Computer Science and Engineering) Degree of Dr. Babasaheb Ambedkar Marathwada University, Aurangabad.

Place: Aurangabad

Date:

Prof. Manisha Mundhe Prof. S.B. Kalyankar

Project Co-ordinator Head

**Dr. Ulhas D. Shiurkar**

**Director,**

**Deogiri Institute of Engineering and Management Studies,**

**Aurangabad**

**DECLARATION**

This is to certify that, the partial project report entitled “**Online Grampanchyat**” Submitted by Group Members is a bona fide work completed under my supervision and guidance in partial fulfilment for award of Bachelor of Engineering in Computer Science and Engineering of Deogiri Institute of Engineering and Management Studies, Aurangabad under Dr. Babasaheb Ambedkar Marathwada University, Aurangabad.

Place: Aurangabad

Date:

Prof. Manisha Mundhe

External Examiner Project Co-ordinator

**Abstract**

Now a day’s people in the rural areas has to go to panchayat office in their location to apply and get their certificates provided in that office. It requires a lot of time and may result in work delay. The data in the office has to be maintained manually. There is no security for the data and faults can be encountered during entering the data mainly which require higher calculations. People also face so many problems in their area. They complain to their respective ward members but they may or may not respond quickly. There are many other problems in the present day panchayat raj system. So, the online panchayat provide solutions to all the problems in the current system. Online panchayat provides online service to the people living in that area. All the services which are done manually are made online panchayat in the project. The people can about their panchayat, activities notifications and all other information related their villages. All the applications and certificates are applied and verified online. The users on the people in the village can complain about their problem through online. Suggestions are also accepted from the people for the development of their village. The user can request any application, suggestion, and complaint at anywhere and at any time.

The grampanchayat provide birth certificate, death certificate, domicile certificate, receipts for house tax, water tax etc. They give order for construction of road, buildings, renewal of building. They keep records of their monthly & yearly budget. The technology used in developing this project is the visual studio 2010. The webpage are developed in ASP.NET frame work and the database is build up in Microsoft SQL Server database .Through which polishing in done to the project.

###### Contents

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Contents** | **Page No.** |
|  | **List of Abbreviations** | **I** |
|  | **List of Figures** | **Ii** |
|  | **List of Screenshots** | **Iii** |
|  |  |  |
| **Chapter1** | **INTRODUCTION** | **1** |
|  | 1.1 Introduction | **1** |
|  | 1.2 Objectives | **3** |
|  | 1.3 Scope | **4** |
|  |  |  |
| **Chapter2** | **LITERATURE SURVEY** |  |
|  |  |  |
| **Chapter3** | **DESCRIPTION OF TOOL** | **5** |
|  | 3.1 Visual Studio | **5** |
|  | 3.1.1 Web Application | **10** |
|  | 3.1.2 C# Programming Language | **11** |
|  | 3.2 Database | **14** |
|  |  |  |
| **Chapter4** | **IMPLEMENTATION** | **15** |
|  | 4.1 Model | **15** |
|  | 4.2 Data Flow Diagram | **15** |
|  | 4.3 ER diagram | **18** |
|  | 4.4 User Interface Design | **19** |
|  |  |  |
| **Chapter 5** | **CONCLUSIONS** | **21** |
|  | 5.1 Conclusion | **21** |
|  | 5.2 Future Scope |  |
|  |  |  |
|  | **REFERENCES** | **26** |
|  | **ACKNOWLEDGEMENT** | **27** |

**List of Abbreviation**

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Acronym** | **Abbreviation** |
| 1 | .net | Network Enabled Technology |
| 2 | ADO | ActiveX Data Object |
| 3 | ASP | Active Server Pages |
| 4 | C# | C Sharp |
| 5 | CLR | Common Language Runtime |
| 6 | CLS | Common Language Specification |
| 7 | DBMS | **Database Management System** |
| 8 | DLL | Dynamic Link Library |
| 9 | IDE | Integrated Development Environment |
| 10 | VS | Visual Studio |
| 11 | SQL | Structured Query Language |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure no.** | **Illustration** | **Page** |
| 1.1.1 | Panchayat Raj Structure | 2 |
| 3.3.1 | Architecture Of .NET Framework | 12 |
| 4.2.1 | Use Case Diagram | 18 |
| 4.3.1 | tax management system | 19 |
| 4.3.2 | application for certificate | 20 |

**List of Screenshot**

|  |  |  |
| --- | --- | --- |
| **Screenshot no.** | **Illustration** | **Page** |
| 5.3.1 | screenshot of login page | 22 |
| 5.3.2 | Screenshot of register page | 24 |

**Chapter1: INTRODUCTION**

**1.1 Introduction**

A grampanchayat is the cornerstone of a local self-government organisation in India of the panchayat raj system at the village or small-town level and has a sarpanch as its elected head. The gram panchayat is divided into wards and each ward is represented by a Ward Member, also referred to as a Panch, who is directly elected by the villagers. The panchayat is chaired by the president of the village, known as a Sarpanch. The term of the elected representatives is five years. The Secretary of the panchayat is a non-elected representative, appointed by the state government, to oversee panchayat activities.

This document is a narrative of our journey towards evolving a framework for strengthening organizational capacity of Gram Panchayats (GPs). The premise is that a strong GP institution can potentially address the issues hindering delivery of services and good governance at the last mile. Moreover, only an effective Gram Panchayat institution can deliver what our constitution mandates.

The Constitution of India, through its 73 Amendment in 1993, defines 'Panchayat', as an institution of self-government constituted under Article 243-B, for the rural areas. The Karnataka Panchayati Raj Act, 1993, established three clear mandates for the Panchayati Raj Institutions, “… greater participation of the people and more effective implementation of rural development programmes and to function as units of local-self-Government”. Accordingly, there have been many initiatives taken towards strengthening decentralized governance in Panchayati Raj Institutions (PRIs), encapsulated as devolution of functions, funds and functionaries, commonly known as the 3Fs. In addition, multiple efforts are undertaken to incentivise, supervise and monitor performances of different level of PRIs. However, despite the fact that the spirit of the devolution or its better-known synonym, federalisation is to help create institutions of self-government, most GPs presently are organizationally weak, structurally inadequate and systemically feeble to leverage and build on devolved powers and functions.

The Panchayatraj institutions have administrative and financial powers at district level (Zilla Parishad), block level (Panchayat Samiti) and village level (Gram Panchayat).



**Figure 1.1.1: Panchayat Raj Structure**

Online panchayat the vast majority of India’s population lives in the village and the Panchayat (village level governance units also known as Panchayat Raj Institution) represent the face of the governance for these villagers.

Online panchayat is an initiative for providing software solution attempting automation of Gram Panchayat functions. Benefits are improved citizen services, better transparency, streamlining of procedures and monitoring of revenues & services.

1). the state Governments have created several Panchayat Raj institutions to ensure grass root level development. At village level the Gram Panchayats are there, these are grass root level institutions.

2). the grampanchayat provide birth, death certificate, domicile certificate, receipts for house tax, water tax etc... They give order for construction of road, buildings, renewal of building. They keep records of their monthly &amp; yearly budget. Online panchayat provides online service to the people living in that area. All the services which are done manually are made online in the project.

3). the people can about their panchayat, activity notifications and all other information related their villages. All the applications and certificates are applied and verified online. The users on the people in the village can complain about their problem through online. Suggestions are also accepted from the people for the development of their village. The user can request any application, suggestion, and complaint at anywhere and at any time.

**1.2 Objective**

The Panchayat represents the low-level in the government activities having more than 60 per cent of the Indian populace and provides a large number of basic services for citizens living in rural area. The aims is to identify and overcome the challenges faced in the villages, such as delay in delivery of services to citizens, lack of flexibility in communication, low revenue mobilization for implementing schemes at the Gram Panchayat level, and lack of monitoring mechanisms for schemes.

E-Governance for Panchayat is considered as a Mission Mode Project (MMP) under NeGP with the primary objective of introducing and institutionalizing e-Governance in Panchayats. The primary aim is to transform Panchayats into efficient units and to strengthen their business functions and processes with the ultimate goal of ensuring effective local area development and prompt and efficient service delivery to common man.

**Tasks of the Panchayat Raj**

Projects and programmes for economic development and social justice under the responsibility of the Panchayati Raj institutions include:

1. Drinking water Tax
2. Fuel and fodder
3. Roads, culverts, bridges ferries, waterways and other means of communication
4. Rural electrification including distribution of electricity
5. Non-conventional energy sources
6. Poverty alleviation program
7. Education, including primary and secondary schools
8. Technical training and vocational education
9. Adult and non-formal education
10. Libraries
11. Markets and fairs
12. Public distribution systems
13. Health and sanitation, including hospitals, primary health centers and dispensaries
14. Family welfare
15. Women and child development
16. Social welfare, including welfare of handicapped and mentally retarded
17. Welfare of the weaker sections and in particular of the scheduled castes and tribes
18. Public distribution system (PDS)
19. Maintenance of community assets

**1.3 Future Scope**

1. Automation of internal workflow processes of Panchayats.
2. Improving delivery of services to citizens
3. Capacity building of Panchayat Representatives and Officials
4. Social Audit
5. Transparency, Accountability, Efficiency and RTI compliance of Panchayats.
6. Improving Governance of local self-government.
7. Agriculture, including agricultural extension
8. Land improvement, implementation of land reforms, land consolidation and soil conservation
9. Minor irrigation, water management and watershed development
10. Animal husbandry, dairying and poultry
11. Fisheries
12. Social forestry and farm forestry
13. Minor forest produce
14. Small scale industries, including food processing industries
15. Khadi, village and cottage industries
16. Rural housing

It can be concluded from the list above that the Panchayat Raj institutions have the overall responsibility for social and economic development within their jurisdictions. A large component of projects are related to the development and management of infrastructure. Several of the items listed above include infrastructure components. The Panchayat Raj institutions are given the task to identify and plan these interventions in the most effective and efficient manner.

The online panchayat project holds great promise for the rural masses as it aims to transform the Panchayati Raj Institutions (PRIs) into symbols of modernity, efficiency and transparency. This is one of the kinds of nationwide IT initiative introduced by Ministry of Panchayati Raj that is willing to ensure people’s participation in decision making programs, implementation and delivery. The project addresses all aspects of Panchayats’ functioning including Planning, Monitoring, Implementation, Budgeting, Accounting, Social Audit and delivery of citizen services like issue of certificates, licenses etc.

**Chapter 2: LITERATURE SURVEY**

**2.1.** According to the DR. M.C. Minimol “eGovernance and rural self-government is the ultimate goal of any government is the betterment of the citizens, it should have better administrative system. The government all over the world are interested in enhancing the quality and quantity of services to the citizens in cost effective manner (C.S.R. Parabhu, Cost Effective solution for effective e-Governance Panchayat 2007).

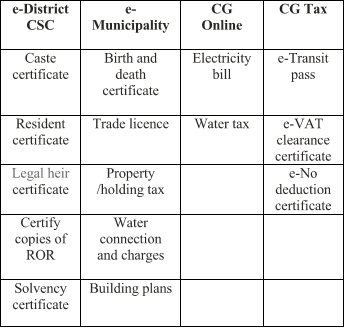
**2.2. E-Governance in Chhattisgarh State**

Chhattisgarh is now focussing critical infrastructure sectors and keeping in view the advantages in terms of resources and business environment it offers to the rural areas.

Chhattisgarh provides for a unique location advantage with direct physical access to markets of seven states: Telangana, Andhra Pradesh, Uttar Pradesh, Maharashtra, Jharkhand and Odisha. This gives the industry unique market. Approximately 70 per cent of our state’s population lives in rural areas and when it comes to literacy rate it is 74 per cent.

**2.2. E-Governance initiatives in Chhattisgarh**

To provide opportunities, information and easy access of the rural India, several steps at various levels (National, State, Block and Panchayat) have been taken in the ministry of rural development through the ICT infrastructure. Table 1- shows the different types of e-Governance services in C.G. State.



**Table 2.1.1 List of services under different categories of e-governance services in Chhattisgarh state**

**Chapter3: DESCRIPTION OF TOOL**

**3.1 Visual Studio**

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code. [3]

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).[7]

Visual Studio supports 36 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++ and C++/CLI (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), F# (as of Visual Studio 2010) and Typescript (as of Visual Studio 2013 Update 2). Support for other languages such as Python, Ruby, Node.js, and M among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Java (and J#) were supported in the past.

Microsoft provides a free version of Visual Studio called the Community edition that supports plugins and is available at no cost.

**Architecture**

Visual Studio does not support any programming language, solution or tool intrinsically; instead, it allows the plugging of functionality coded as a VSPackage. When installed, the functionality is available as a Service. The IDE provides three services: SVsSolution, which provides the ability to enumerate projects and solutions; SVsUIShell, which provides windowing and UI functionality (including tabs, toolbars and tool windows); and SVsShell, which deals with registration of VSPackages. In addition, the IDE is also responsible for coordinating and enabling communication between services. All editors, designers, project types and other tools are implemented as VSPackages. Visual Studio uses COM to access the VSPackages. The Visual Studio SDK also includes the Managed Package Framework (MPF), which is a set of managed wrappers around the COM-interfaces that allow the Packages to be written in any CLI compliant language. However, MPF does not provide all the functionality exposed by the Visual Studio COM interfaces. The services can then be consumed for creation of other packages, which add functionality to the Visual Studio IDE.

Support for programming languages is added by using a specific VSPackage called a Language Service. A language service defines various interfaces which the VSPackage implementation can implement to add support for various functionalities. Functionalities that can be added this way include syntax colouring, statement completion, brace matching, parameter information tooltips, member lists and error markers for background compilation. If the interface is implemented, the functionality will be available for the language. Language services are implemented on a per-language basis. The implementations can reuse code from the parser or the compiler for the language. Language services can be implemented either in native code or managed code. For native code, either the native COM interfaces or the Babel Framework (part of Visual Studio SDK) can be used. For managed code, the MPF includes wrappers for writing managed language services.

Visual Studio does not include any source control support built in but it defines two alternative ways for source control systems to integrate with the IDE. A Source Control VSPackage can provide its own customised user interface. In contrast, a source control plugin using the MSSCCI (Microsoft Source Code Control Interface) provides a set of functions that are used to implement various source control functionality, with a standard Visual Studio user interface. MSSCCI was first used to integrate Visual SourceSafe with Visual Studio 6.0 but was later opened up via the Visual Studio SDK. Visual Studio .NET 2002 used MSSCCI 1.1, and Visual Studio .NET 2003 used MSSCCI 1.2. Visual Studio 2005, 2008 and 2010 use MSSCCI Version 1.3, which adds support for rename and delete propagation as well as asynchronous opening.

Visual Studio supports running multiple instances of the environment (each with its own set of VSPackages). The instances use different registry hives (see MSDN's definition of the term "registry hive" in the sense used here) to store their configuration state and are differentiated by their AppId (Application ID). The instances are launched by an AppId-specific .exe that selects the AppId, sets the root hive and launches the IDE. VSPackages registered for one AppId are integrated with other VSPackages for that AppId. The various product editions of Visual Studio are created using the different AppIds. The Visual Studio Express edition products are installed with their own AppIds, but the Standard, Professional and Team Suite products share the same AppId. Consequently, one can install the Express editions side-by-side with other editions, unlike the other editions which update the same installation. The professional edition includes a superset of the VSPackages in the standard edition and the team suite includes a superset of the VSPackages in both other editions. The AppId system is leveraged by the Visual Studio Shell in Visual Studio 2008. Features Code editor Like any other IDE, it includes a code editor that supports syntax highlighting and code completion using IntelliSense for variables, functions, methods, loops and LINQ queries. IntelliSense is supported for the included languages, as well as for XML and for Cascading Style Sheets and JavaScript when developing web sites and web applications. Autocomplete suggestions appear in a modeless list box over the code editor window, in proximity of the editing cursor. In Visual Studio 2008 onwards, it can be made temporarily semi-transparent to see the code obstructed by it. The code editor is used for all supported languages.

The Visual Studio code editor also supports setting bookmarks in code for quick navigation. Other navigational aids include collapsing code blocks and incremental search, in addition to normal text search and regex search. The code editor also includes a multi-item clipboard and a task list. The code editor supports code snippets, which are saved templates for repetitive code and can be inserted into code and customized for the project being worked on. A management tool for code snippets is built in as well. These tools are surfaced as floating windows which can be set to automatically hide when unused or docked to the side of the screen. The Visual Studio code editor also supports code refactoring including parameter reordering, variable and method renaming, interface extraction and encapsulation of class members inside properties, among others. Visual Studio features background compilation (also called incremental compilation). As code is being written, Visual Studio compiles it in the background in order to provide feedback about syntax and compilation errors, which are flagged with a red wavy underline. Warnings are marked with a green underline. Background compilation does not generate executable code, since it requires a different compiler than the one used to generate executable code. Background compilation was initially introduced with Microsoft Visual Basic but has now been expanded for all included languages. Debugger Visual Studio includes a debugger that works both as a source-level debugger and as a machine-level debugger. It works with both managed code as well as native code and can be used for debugging applications written in any language supported by Visual Studio. In addition, it can also attach to running processes and monitor and debug those processes. If source code for the running process is available, it displays the code as it is being run. If source code is not available, it can show the disassembly. The Visual Studio debugger can also create memory dumps as well as load them later for debugging. Multi-threaded programs are also supported. The debugger can be configured to be launched when an application running outside the Visual Studio environment crashes.

The debugger allows setting breakpoints (which allow execution to be stopped temporarily at a certain position) and watches (which monitor the values of variables as the execution progresses). Breakpoints can be conditional, meaning they get triggered when the condition is met. Code can be stepped over, i.e., run one line (of source code) at a time. It can either step into functions to debug inside it, or step over it, i.e., the execution of the function body isn't available for manual inspection. The debugger supports Edit and Continue, i.e., it allows code to be edited as it is being debugged. When debugging, if the mouse pointer hovers over any variable, its current value is displayed in a tooltip ("data tooltips"), where it can also be modified if desired. During coding, the Visual Studio debugger lets certain functions be invoked manually from the immediate tool window. The parameters to the method are supplied at the immediate window. Designer

Visual Studio includes a host of visual designers to aid in the development of applications. These tools include: Windows Forms Designer the Windows Forms designer is used to build GUI applications using Windows Forms. Layout can be controlled by housing the controls inside other containers or locking them to the side of the form. Controls that display data (like textbox, list box and grid view) can be bound to data sources like databases or queries. Data-bound controls can be created by dragging items from the Data Sources window onto a design surface. The UI is linked with code using an event-driven programming model. The designer generates either C# or VB.NET code for the application. WPF Designer the WPF designer, codenamed Cider, was introduced with Visual Studio 2008. Like the Windows Forms designer it supports the drag and drop metaphor. It is used to author user interfaces targeting Windows Presentation Foundation. It supports all WPF functionality including data binding and automatic layout management. It generates XAML code for the UI. The generated XAML file is compatible with Microsoft Expression Design, the designer-oriented product. The XAML code is linked with code using a code-behind model. Web designer/development Visual Studio also includes a web-site editor and designer that allows web pages to be authored by dragging and dropping widgets. It is used for developing ASP.NET applications and supports HTML, CSS and JavaScript. It uses a code-behind model to link with ASP.NET code. From Visual Studio 2008 onwards, the layout engine used by the web designer is shared with Microsoft Expression Web. There is also ASP.NET MVC support for MVC technology as a separate download and ASP.NET Dynamic Data project available from Microsoft. Class designer The Class Designer is used to author and edit the classes (including its members and their access) using UML modelling. The Class Designer can generate C# and VB.NET code outlines for the classes and methods. It can also generate class diagrams from hand-written classes. Data designer the data designer can be used to graphically edit database schemas, including typed tables, primary and foreign keys and constraints. It can also be used to design queries from the graphical view. Mapping designer From Visual Studio 2008 onwards, the mapping designer is used by LINQ to SQL to design the mapping between database schemas and the classes that encapsulate the data. The new solution from ORM approach, ADO.NET Entity Framework, replaces and improves the old technology. Other tools

**Open Tabs Browser**

The open tabs browser is used to list all open tabs and to switch between them. It is invoked using CTRL+TAB. Properties Editor the Properties Editor tool is used to edit properties in a GUI pane inside Visual Studio. It lists all available properties (both read-only and those which can be set) for all objects including classes, forms, web pages and other items. Object Browser the Object Browser is a namespace and class library browser for Microsoft .NET. It can be used to browse the namespaces (which are arranged hierarchically) in managed assemblies. The hierarchy may or may not reflect the organization in the file system. Solution Explorer in Visual Studio parlance, a solution is a set of code files and other resources that are used to build an application. The files in a solution are arranged hierarchically, which might or might not reflect the organization in the file system. The Solution Explorer is used to manage and browse the files in a solution. Team Explorer Team Explorer is used to integrate the capabilities of Team Foundation Server, the Revision Control System into the IDE (and the basis for Microsoft's CodePlex hosting environment for open source projects). In addition to source control it provides the ability to view and manage individual work items (including bugs, tasks and other documents) and to browse TFS statistics. It is included as part of a TFS install and is also available as a download for Visual Studio separately. Team Explorer is also available as a stand-alone environment solely to access TFS services. Data Explorer Data Explorer is used to manage databases on Microsoft SQL Server instances. It allows creation and alteration of database tables (either by issuing T-SQL commands or by using the Data designer). It can also be used to create queries and stored procedures, with the latter in either T-SQL or in managed code via SQL CLR. Debugging and IntelliSense support is available as well. Server Explorer the Server Explorer tool is used to manage database connections on an accessible computer. It is also used to browse running Windows Services, performance counters, and Windows Event Log and message queues and use them as a data source. Pre-emptive Protection-Dotfuscator Community Edition Visual Studio includes a free 'light' version of Dotfuscator by Pre-emptive Solutions which obfuscates and hardens applications to help secure trade secrets (IP), reduce piracy/counterfeiting, protect against tampering and unauthorized debugging. Dotfuscator works with all flavours of .NET including ASP.NET, Xamarin, Unity and UWP. Text Generation Framework Visual Studio includes a full text generation framework called T4 which enables Visual Studio to generate text files from templates either in the IDE or via code. ASP.NET Web Site Administration Tool the ASP.NET Web Site Administration Tool allows for the configuration of ASP.NET websites.

**3.2 Web Application**

**ASP.NET**

ASP.NET is an open-source server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages. ASP.NET's successor is ASP.NET Core. It is a re-implementation of ASP.NET as a modular web framework, together with other frameworks like Entity Framework. The new framework uses the new open source .NET Compiler Platform (codename "Roslyn") and is cross platform. ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages) have merged into a unified MVC 6.

**Characteristics**

ASP.NET Web pages, known officially as Web Forms, are the main building blocks for application development in ASP.NET. [3] There are two basic methodologies for Web Forms, a web application format and a web site format. Web applications need to be compiled before deployment, while web sites structures allows the user to copy the files directly to the server without prior compilation. Web forms are contained in files with an ".aspx" extension; these files typically contain static (X) HTML markup or component markup. The component markup can include server-side Web Controls and User Controls that have been defined in the framework or the web page. For example, a textbox component can be defined on a page as <asp: textbox id='myid' runat='server'>, which is rendered into an html input box.

**3.3 C# Programming Language**

C# syntax is highly expressive, yet it is also simple and easy to learn. The curly-brace syntax of C# will be instantly recognizable to anyone familiar with C, C++ or Java. Developers who know any of these languages are typically able to begin to work productively in C# within a very short time. C# syntax simplifies many of the complexities of C++ and provides powerful features such as nullable value types, enumerations, delegates, lambda expressions and direct memory access, which are not found in Java. C# supports generic methods and types, which provide increased type safety and performance, and iterators, which enable implementers of collection classes to define custom iteration behaviours that are simple to use by client code. Language-Integrated Query (LINQ) expressions make the strongly-typed query a first-class language construct.

As an object-oriented language, C# supports the concepts of encapsulation, inheritance, and polymorphism. All variables and methods, including the Main method, the application's entry point, are encapsulated within class definitions. A class may inherit directly from one parent class, but it may implement any number of interfaces. Methods that override virtual methods in a parent class require the override keyword as a way to avoid accidental redefinition. In C#, a struct is like a lightweight class; it is a stack-allocated type that can implement interfaces but does not support inheritance.

In addition to these basic object-oriented principles, C# makes it easy to develop software components through several innovative language constructs, including the following:

* Encapsulated method signatures called delegates, which enable type-safe event notifications.
* Properties, which serve as accessory for private member variables.
* Attributes, which provide declarative metadata about types at run time.
* Inline XML documentation comments.
* Language-Integrated Query (LINQ) which provides built-in query capabilities across a variety of data sources.

If you have to interact with other Windows software such as COM objects or native Win32 DLLs, you can do this in C# through a process called "Interop." Interop enables C# programs to do almost anything that a native C++ application can do. C# even supports pointers and the concept of "unsafe" code for those cases in which direct memory access is absolutely critical.

The C# build process is simple compared to C and C++ and more flexible than in Java. There are no separate header files, and no requirement that methods and types be declared in a particular order. A C# source file may define any number of classes, structs, interfaces, and events.

The following are additional C# resources:

## .NET Framework Platform Architecture

C# programs run on the .NET Framework, an integral component of Windows that includes a virtual execution system called the common language runtime (CLR) and a unified set of class libraries. The CLR is the commercial implementation by Microsoft of the common language infrastructure (CLI), an international standard that is the basis for creating execution and development environments in which languages and libraries work together seamlessly.

Source code written in C# is compiled into an intermediate language (IL) that conforms to the CLI specification. The IL code and resources, such as bitmaps and strings, are stored on disk in an executable file called an assembly, typically with an extension of .exe or .dll. An assembly contains a manifest that provides information about the assembly's types, version, culture, and security requirements.

When the C# program is executed, the assembly is loaded into the CLR, which might take various actions based on the information in the manifest. Then, if the security requirements are met, the CLR performs just in time (JIT) compilation to convert the IL code to native machine instructions. The CLR also provides other services related to automatic garbage collection, exception handling, and resource management. Code that is executed by the CLR is sometimes referred to as "managed code," in contrast to "unmanaged code" which is compiled into native machine language that targets a specific system. The following diagram illustrates the compile-time and run-time relationships of C# source code files, the .NET Framework class libraries, assemblies, and the CLR.



**Figure3.3.1 Architecture Of .NET Framework**

Language interoperability is a key feature of the .NET Framework. Because the IL code produced by the C# compiler conforms to the Common Type Specification (CTS), IL code generated from C# can interact with code that was generated from the .NET versions of Visual Basic, Visual C++, or any of more than 20 other CTS-compliant languages. A single assembly may contain multiple modules written in different .NET languages, and the types can reference each other just as if they were written in the same language.

In addition to the run time services, the .NET Framework also includes an extensive library of over 4000 classes organized into namespaces that provide a wide variety of useful functionality for everything from file input and output to string manipulation to XML parsing, to Windows Forms controls. The typical C# application uses the .NET Framework class library extensively to handle common "plumbing" chores.

**3.4 Database**

**SQL server database 2014**

SQL Server Database (commonly referred to as SQL RDBMS or simply as SQL) is an object-relational database management system produced and marketed by SQL Maria DB. Larry Ellison and his two friends and former co-workers, Bob Miner and Ed Oates, started a consultancy called Software Development Laboratories (SDL) in 1977. SDL developed the original version of the Oracle software. The name Oracle comes from the code name of a CIAfunded project Ellison had worked on while formerly employed by Apex. [8]

A cloud database is a database that runs on a cloud computing platform, and access to it is provided as a service. Managed database services take care of scalability, backup, and high availability of the database. Azure SQL Database is a managed database service which is different from AWS RDS which is a container service. Microsoft Azure SQL Database includes built-in intelligence that learns app patterns and adapts to maximize performance, reliability, and data protection. It was originally announced in 2009 and released in 2010. [9]

**Pricing**

Azure SQL Database is offered either as a Standalone database or Elastic database pool, and is priced in three tiers: Basic, Standard and Premium. Each tier offers different performance levels to accommodate a variety of workloads. The resources available for Standalone databases are expressed in terms of Database Transaction Units (DTUs) and for elastic pools in terms of elastic DTUs or eDTUs. [10] A DTU is defined as a blended measure of CPU, memory, and data I/O and transaction log I/O in a ratio determined by an OLTP benchmark workload designed to be typical of real-world OLTP workloads. Databases are available as Standalone databases or in database pools which allow multiple databases to share storage and compute resources. [10]

**Chapter4: IMPLEMENTATION**

**4.1 Model**

**Iterative design**

Iterative design is a design methodology based on a cyclic process of prototyping, testing, analysing, and refining a product or process. Based on the results of testing the most recent iteration of a design, changes and refinements are made. This process is intended to ultimately improve the quality and functionality of a design. In iterative design, interaction with the designed system is used as a form of research for informing and evolving a project, as successive versions, or iterations of a design are implemented**.**

**Iterative design process**

The iterative design process may be applied throughout the new product development process. However, changes are easiest and less expensive to implement in the earliest stages of development. The first step in the iterative design process is to develop a prototype. The prototype should be evaluated by a focus group or a group not associated with the product in order to deliver non- biased opinions. Information from the focus group should be synthesized and incorporated into the next iteration of the design. The process should be repeated until user issues have been reduced to an acceptable level.

**Application: Human computer interfaces**

Iterative design is commonly used in the development of human computer interfaces. This allows designers to identify any usability issues that may arise in the user interface before it is put into wide use. Even the best usability experts cannot design perfect user interfaces in a single attempt, so a usability engineering lifecycle should be built around the concept of iteration. The typical steps of iterative design in user interfaces are as follows:

1. Complete an initial interface design

2. Present the design to several test users

3. Note any problems had by the test user

4. Refine interface to account for/fix the problems

5. Repeat steps 2-4 until user interface problems are resolved

Iterative design in user interfaces can be implemented in many ways. One common method of using iterative design in computer software is software testing. While this includes testing the product for functionality outside of the user interface, important feedback on the interface can be gained from subject testing early versions of a program. This allows software companies to release a better quality product to the public, and prevents the need of product modification following its release. Iterative design in online (website) interfaces is a more continuous process, as website modification, after it has been released to the user, is far more viable than in software design. Often websites use their users as test subjects for interface design, making modifications based on recommendations from visitors to their sites.

**Iterative design use**

Iterative design is a way of confronting the reality of unpredictable user needs and behaviours that can lead to sweeping and fundamental changes in a design. User testing will often show that even carefully evaluated ideas will be inadequate when confronted with a user test. Thus, it is important that the flexibility of the iterative design’s implementation approach extends as far into the system as possible. Designers must further recognize that user testing results may suggest radical change that requires the designers to be prepared to completely abandon old ideas in favour of new ideas that are more equipped to suit user needs. Iterative design applies in many fields, from making knives to rockets. As an example consider the design of an electronic circuit that must perform a certain task, and must ultimately fit in a small space on a circuit board. It is useful to split these independent tasks into two smaller and simpler tasks, the functionality task, and the space and weight task. A breadboard is a useful way of implementing the electronic circuit on an interim basis, without having to worry about space and weight. Once the circuit works, improvements or incremental changes may be applied to the breadboard to increase or improve functionality over the original design. When the design is finalized, one can set about designing a proper circuit board meeting the space and weight criteria. Compacting the circuit on the circuit board requires that the wires and components be juggled around without changing their electrical characteristics. This juggling follows simpler rules than the design of the circuit itself, and is often automated. As far as possible off the shelf components are used, but where necessary for space or performance reasons, custom made components may be developed. Several instances of iterative design are as follows:

Wiki - A wiki is a natural repository for iterative design. The 'Page History' facility allows tracking back to prior versions. Modifications are mostly incremental, and leave substantial parts of the text unchanged.

Common law - The principle of legal precedent builds on past experience. This makes law a form of iterative design where there should be a clear audit trail of the development of legal thought.

Evolution - There is a parallel between iterative and the theory of Natural Selection. Both involve a trial and error process in which the most suitable design advances to the next generation, while less suitable designs perish by the wayside. Subsequent versions of a product should also get progressively better as its producers learn what works and what doesn't in a process of refinement and continuous improvement.

**Benefits**

When properly applied, iterative design will ensure a product or process is the best solution possible. When applied early in the development stage, significant cost savings are possible. Other benefits to iterative design include:

1. Serious misunderstandings are made evident early in the lifecycle, when it's possible to react to them.

2. It enables and encourages user feedback, so as to elicit the system's real requirements.

3. Where the work is contracted, Iterative Design provides an incremental method for more effectively involving the client in the complexities that often surround the design process.

4. The development team is forced to focus on those issues that are most critical to the project, and team members are shielded from those issues that distract them from the project's real risks.

5. Continuous, iterative testing enables an objective assessment of the project's status.

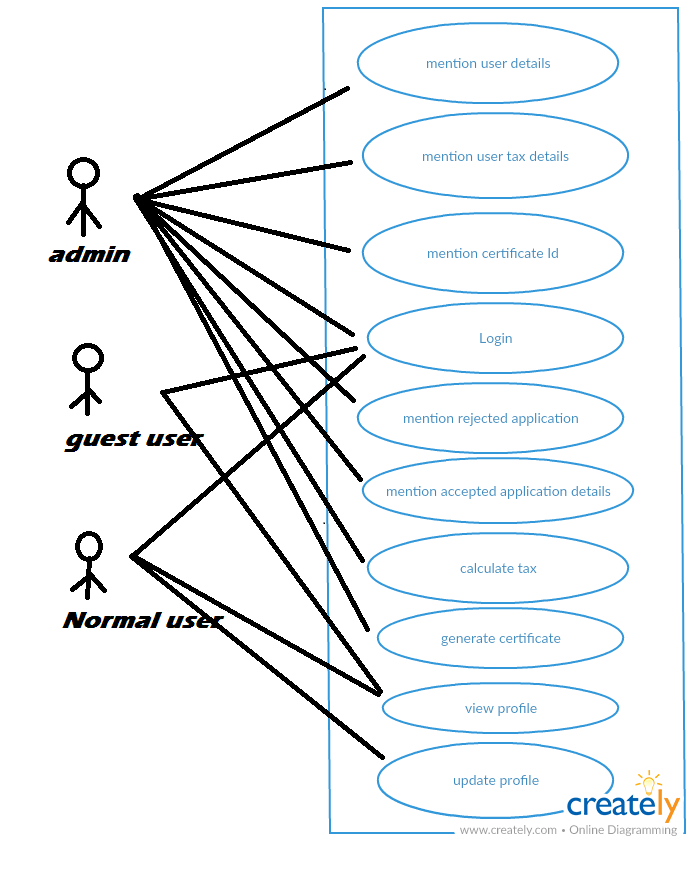
6. Inconsistencies among requirements, designs, and implementations are detected early.

7. The workload of the team, especially the testing team, is spread out more evenly throughout the lifecycle.

8. This approach enables the team to leverage lessons learned, and therefore to continuously improve the process.

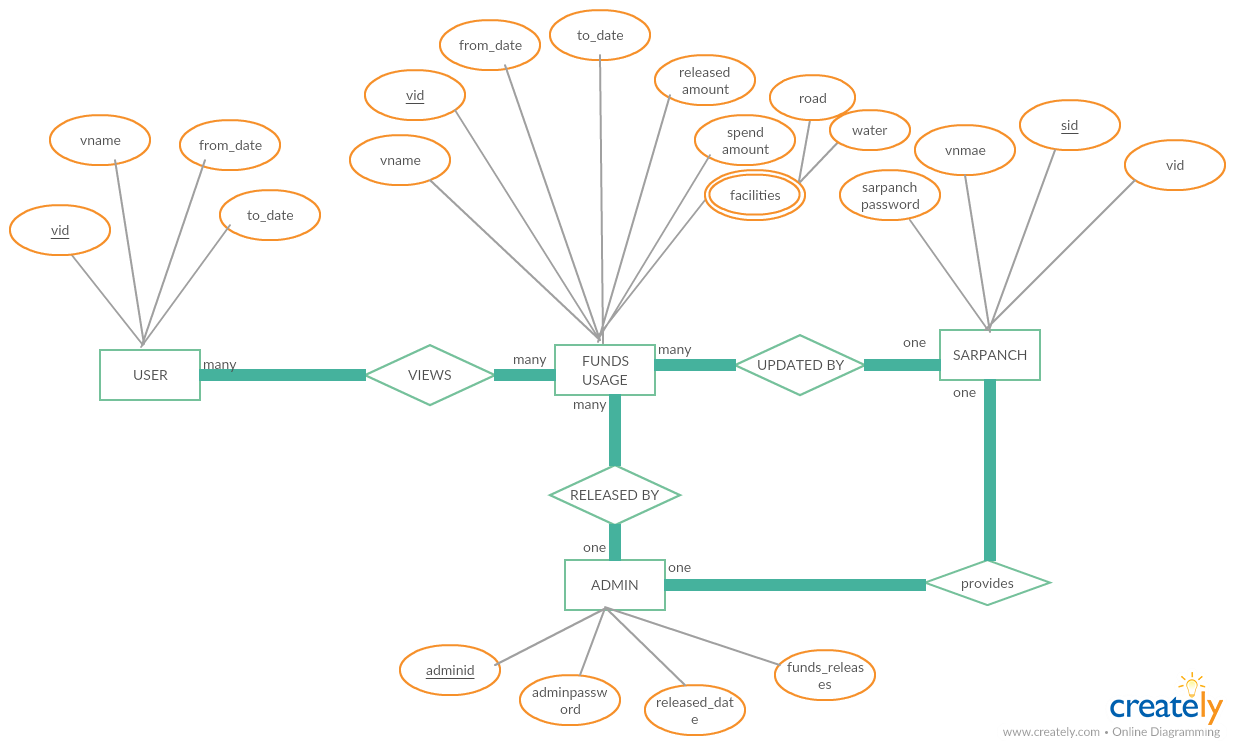
9. Stakeholders in the project can be given concrete evidence of the project's status throughout the lifecycle.

**4.2 Data Flow Diagram**

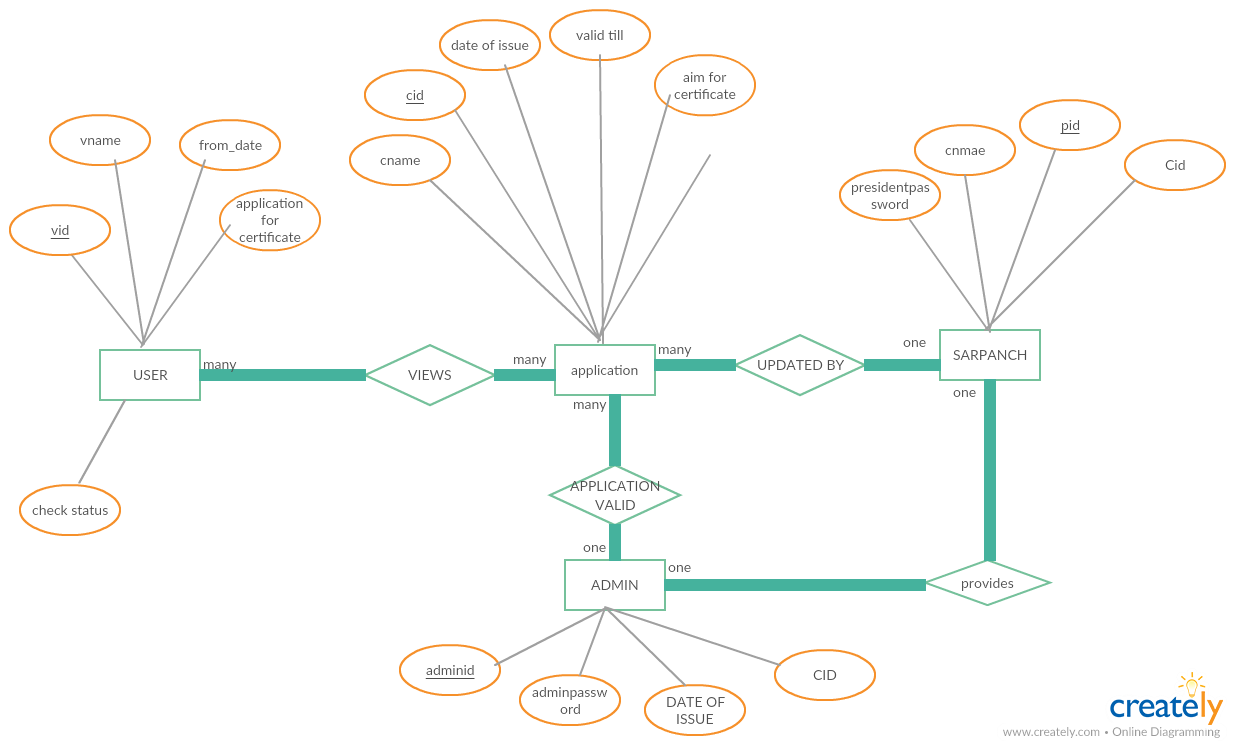
****

**Figure 4.2.1: Use Case Diagram**

**4.3 ER Diagram**

****

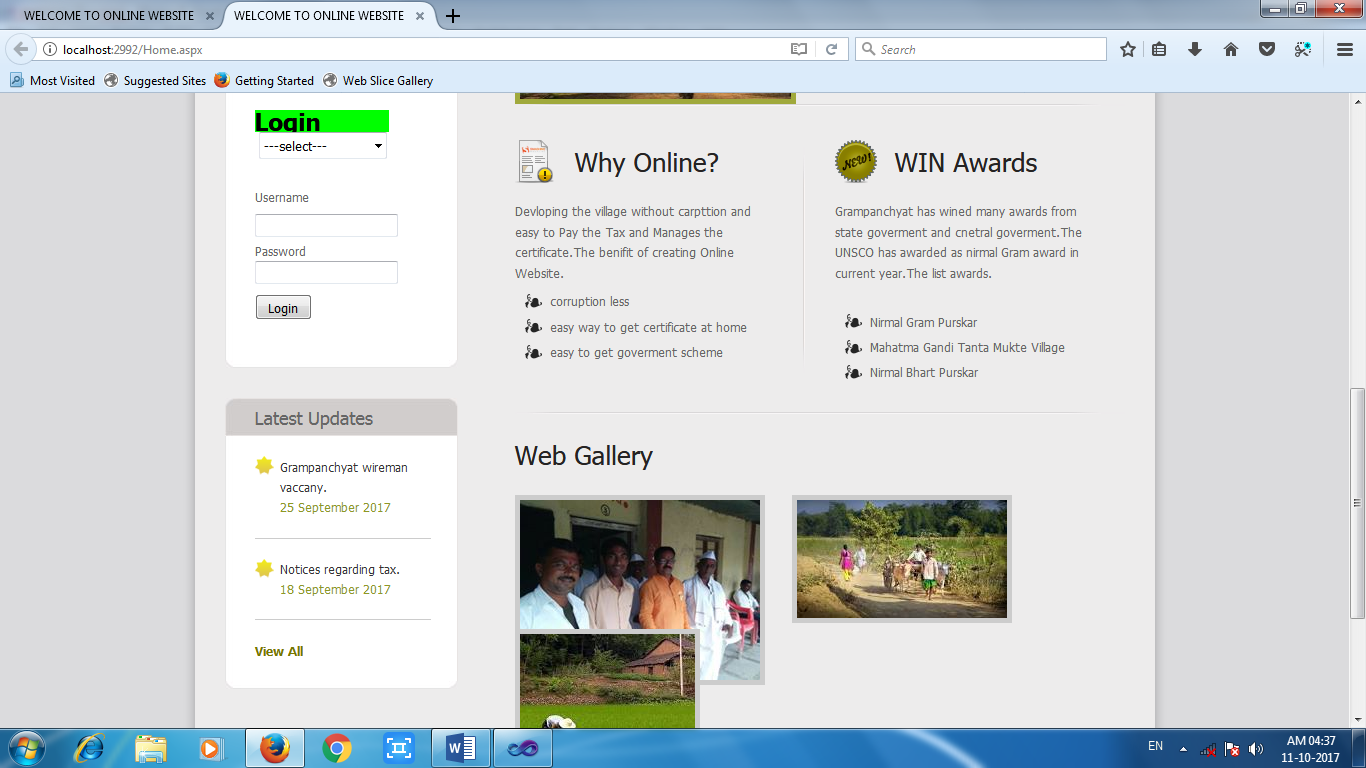
**Figure4.3.1: tax management system**

****

**Figure 4.3.2 application for certificate**

**4.4 USER INTERFACE DESIGN**

**HOME PAGE**

****

**Screenshot 5.3.1 screenshot of login page**

**DESCRIPTION**

This page allows user to log in in different type of user like admin or normal as per he registered previously by admin. This page contains abstract of the village some photo menu that includes different switchable webpage. This page has login form to get logged in.

**SOURCE CODE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Data;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Configuration;

namespace grampan

{

public partial class Site1 : System.Web.UI.MasterPage

{

SqlConnection con = new SqlConnection();

SqlCommand cmd = new SqlCommand();

SqlConnection conn = new SqlConnection();

SqlDataAdapter sda = new SqlDataAdapter();

DataSet ds = new DataSet();

DataTable dt=new DataTable();

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

{

con.ConnectionString = @"Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\NARAYAN\Documents\Grampanchyat.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True";

conn.ConnectionString = @"Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\NARAYAN\Documents\admin.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True";

if (Session["admin"] != null || Session["normal"] != null)

{

SubmitLogin.Visible = false;

Panel1.Visible = true;

}

else if (Session["admin"] == null || Session["normal"]==null)

{

SubmitLogin.Visible = true;

}

}

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

{

if (DropDownList1.Text == "admin")

{

cmd.CommandText = "select \* from admin where id='" + Username.Text + "' and password='" + Password.Text + "'";

cmd.Connection = conn;

sda.SelectCommand = cmd;

sda.Fill(ds, "admin");

if (ds.Tables[0].Rows.Count > 0)

{

Session["admin"] = ds.Tables[0].Rows[0]["id"].ToString();

SubmitLogin.Visible = false;

Panel1.Visible = true;

message.Text = "Welcome Admin";

Response.Redirect("admin.aspx");

}

else

{

ScriptManager.RegisterClientScriptBlock(Page, typeof(Page), "ClientScript", "alert('Enter Correct Username & password')", true);

}

}

else if (DropDownList1.Text == "normal")

{

cmd.CommandText = "select \* from Grampanchyat where Username='" + Username.Text + "' and Password='" + Password.Text + "'";

cmd.Connection = con;

sda.SelectCommand = cmd;

sda.Fill(ds, "Grampanchyat");

if (ds.Tables[0].Rows.Count > 0)

{

Session["normal"] = ds.Tables[0].Rows[0]["UID"].ToString();

SubmitLogin.Visible = false;

message.Text = ds.Tables[0].Rows[0]["Username"].ToString();

Response.Redirect("userhome.aspx");

}

else

{

ScriptManager.RegisterClientScriptBlock(Page, typeof(Page), "ClientScript", "alert('Enter Correct Username & password')", true);

}

}

else if(DropDownList1.Text=="select")

{

ScriptManager.RegisterClientScriptBlock(Page, typeof(Page), "ClientScript", "alert('select user from list')", true);

}

}

protected void LinkButton1\_Click(object sender, EventArgs e)

{

}

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

{

Session.RemoveAll();

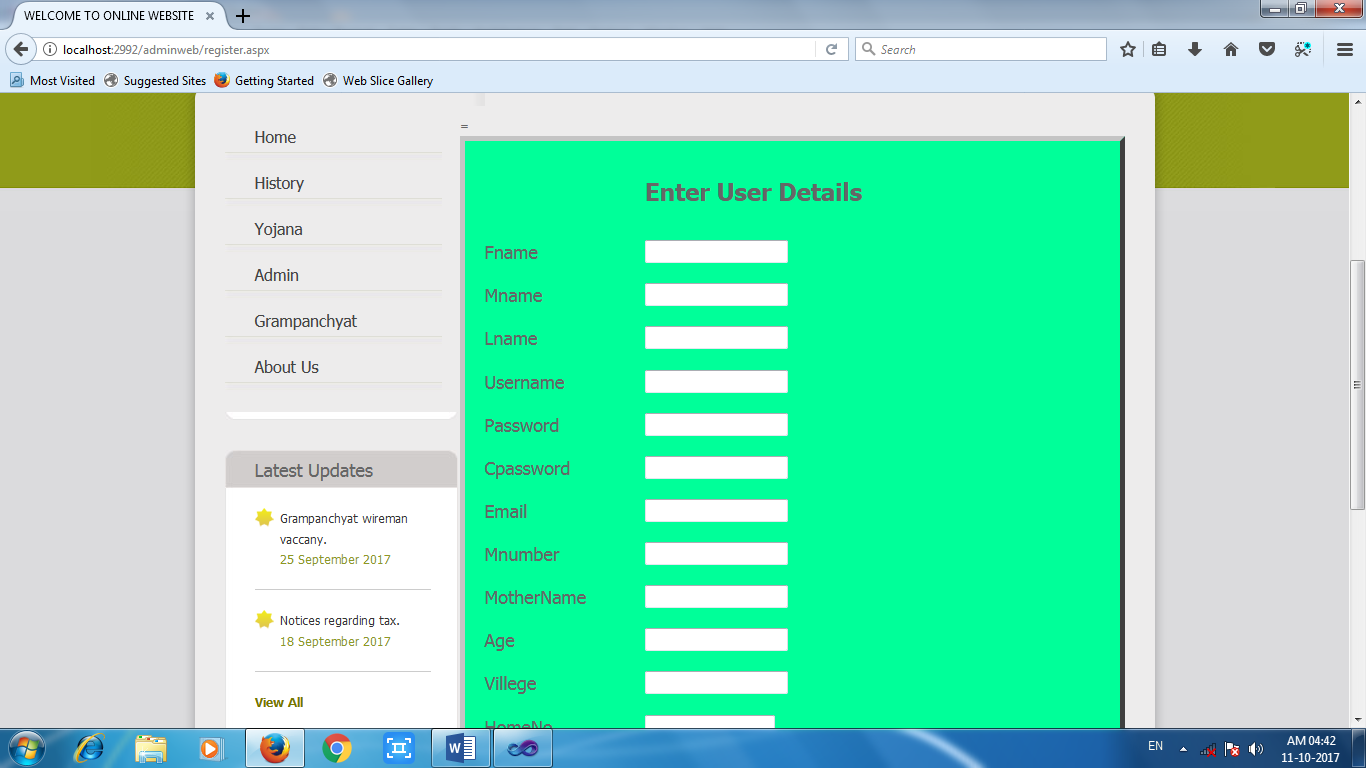
Response.Redirect("Home.aspx");

}

}

}

**REGISTER PAGE**

****

**Screenshot 5.3.2 register user**

**DESCRIPTION**

This page is for register new user. This page can only be accessed by admin. This page contains form to fill the detail of user to register. There are some validators are applied to form. And also menu is added to switch through the different pages.

**SOURCE CODE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

namespace grampan.adminweb

{

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

{

SqlConnection con;

SqlCommand cmd;

SqlDataReader rdr;

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

{

if (Session["admin"] == null)

{

Response.Redirect("~/Home.aspx");

}

con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\NARAYAN\Documents\Grampanchyat.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True");

}

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

{

long mn = Convert.ToInt64(Mnumber.Text);

int ag = Convert.ToInt32(Age.Text);

int Hm = Convert.ToInt32(HomeNo.Text);

decimal ic = Convert.ToDecimal(Income.Text);

Int32 pc = Convert.ToInt32(PostalNo.Text);

string fn = Fname.Text;

string mna = Mname.Text;

string lm = Lname.Text;

string pw = Password.Text;

string em = Email.Text;

string mon = MotherName.Text;

string vi = Villege.Text;

string oc = Occuption.Text;

string tq = Taluka.Text;

string dist = Dist.Text;

string un = Username.Text;

string str = FileUpload1.FileName;

FileUpload1.PostedFile.SaveAs(Server.MapPath(".")+str);

string path = str.ToString();

string que = "insert into Grampanchyat values('" + fn + "','" + mna + "','" + lm + "','" + un + "','" + pw + "','" + em + "','" +mn + "','" + mon + "','" + ag + "','" + vi + "','" +Hm +"','" +oc +"','" +ic +"','" +tq +"','" + dist+"','" +pc +"','"+path+"')";

cmd = new SqlCommand(que, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

ScriptManager.RegisterClientScriptBlock(Page, typeof(Page), "ClientScript", "alert('User are Saveed in Database ')", true);

}

}

}

**Chapter 5: CONCLUSIONS**

**5.1 CONCLUSION**

E-Governance for panchayat provides online services to the people living in that panchayat. It helps for the people in that area to easily complete their work which involves the action of authority of the panchayat people. As everything is made online people can request their applications from anywhere at any time. After requesting the certificate the process will be carried out normally, no need for the people to go to panchayat office every time for the completion of work. It saves people time and they can perform their daily work without any interruption.

By using this users can also post their problems directly to the higher officials and can get them solved. The people can access the website where ever they are in the world. This is user friendly and users can perform their operations easily. The e-Governance for Panchayat project promises the rural areas that it can transform the Panchayat Raj Institutions into a modern, efficient and transparent areas. The concept of online panchayat system put a step forward in digitalizing India.

**References**

**1**. [http://www.projectcorner.in/grampanchayat-automation-system-academic-project-java synopsis/](http://www.projectcorner.in/grampanchayat-automation-system-academic-project-java%20synopsis/) sited on 12th Sept 2017

2. [www.punsarigrampanchayat.in](http://www.punsarigrampanchayat.in) sited on 12th Sept 2017

3. <https://etownpanchayat.com/> sited on 17th Sept 2017

4. <http://mpanchayat.ap.gov.in> sited on 17th Sept 2017

5. <http://wikiedit.org/India/Gandhi-Nagar> sited on 18th Sept 2017

6. <http://mpanchayat.ap.gov.in> sited on 22nd Sept 2017

7. <http://epanchayat-pris.ap.nic.in> sited on 30th Sept 2017

8. <http://www.cips.org.in/online-mission-mode-project> sited on 5th Oct 2017

9. <http://shubharathi.blogspot.in/015/04/importance-of-online-for-rural.html> sited on 10th Oct 2017

**ACKNOWLEDGEMENT**

We would like to place on record our deep sense of gratitude to **Prof. Mr. S.B .Kalyankar Sir**, HOD-Dept. of Computer Science and Engineering, Deogiri Institute of Engineering and management Studies Aurangabad, for his generous guidance, help and useful suggestions.

We express our sincere gratitude to **Prof. Manisha Mundhe**, Dept. of Computer Science and Engineering, Deogiri Institute of Engineering and management Studies Aurangabad, for her stimulating guidance, continuous encouragement and supervision throughout the course of present work.

We are extremely thankful to **Dr. Ulhas Shiurkar Sir**, Director, Deogiri Institute of Engineering and management Studies Aurangabad, for providing us infrastructural facilities to work in, without which this work would not have been possible.