INTRODUCTION

1. INTRODUCTION

Municipality is an administrative entity composed of a clearly defined territory and its population and commonly denotes a city, town, or village, or a small grouping of them. A municipality is typically governed by a mayor and a city council or municipal but is not council. The notion of municipality includes township but is not restricted to them. A municipality is a general-purpose district, as opposed to a special-purpose district.

The development of this system contains the following activities, which try to automate the entire process keeping in the view of database integration approach. User Friendliness is provided in the application with various controls provided by system Rich User Interface. It can be accessed over the Internet. The user information files can be stored in centralized database which can be maintained by the system. This can give the good security for user information because data is not in client machine. Authentication is provided for this application only registered members can access. Report generation features is provided to generate different kind of reports.

1.1 Salient Features OF Municipality Administration System.

- A complete online application: The Municipality Administration System is a complete online application (from member registration to certificate generation). The application also provides the members under the municipality to gain new memberships and it also facilitates online status tracking of certificate requests and modification requests.
- Extensive use of master data: The application provides the user with a wide range of master tables to capture most of the commonly used data like Municipality No, Municipality Name etc.). Most of the master data is made available in the system in pre-populated form following national level standards. The usage of master tables not only helps in data redundancy but also helps in data cleansing and removes the possibility of erroneous data.

1.2 DIFFERENT MODULES

Admin Module

Admin module has main role in this software. It consists of the peculiar functions like office information, office updating, certificate type and verification of birth certification and death certification request.'

User Module

The most important module in this software is the user module. It carries a major part of information to greater extent. It includes online application for a new birth and death certificates, application for membership, application for membership updating etc.

2. SYSTEM STUDY AND ANALYSIS

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

Before designing a computer system, which will satisfy the information requirements of an organization, it is important that the nature of the business and the way it currently operates are clearly understood. This process of system investigation includes several methods of gathering the required information such as system functionality, problems encountered, requirements of the proposed system, user their responsibility.

2.1 FEASIBILITY STUDY

Feasibility Study can be considered as preliminary investigation that helps the management to take decision about whether study of system should be feasible for development or not.

The following steps are to be followed while performing feasibility analysis:

- Develop system flowcharts.
- Identify the deficiencies of current system and set goals.
- Enumerate the alternative solution or potential candidate system to meet goals.
- Determine the feasibility of each alternative such as technical feasibility, operational feasibility, etc.
- Weight the performance and cost effectiveness of each candidate system.
- Rank the other alternatives and select the best candidate system.
- Prepare a system proposal of final project directive to management for approval.

There are three aspects in the feasibility study portion of the preliminary investigation.

1. Technical Feasibility

2. Economic Feasibility

3. Operational Feasibility

Technical Feasibility:

Technical Feasibility investigates the technical feasibility of each implementation alternative. It analyzes and determines whether the solution can be supported by existing technology or not. The analyst determines whether current technical resources be upgraded or added it that fulfill the new requirements. It ensures that the candidate system provides appropriate responses to what extent it can support the technical enhancement.

Economic Feasibility:

It is evaluating the effectiveness of candidate system by using cost/benefit analysis method. It demonstrates the net benefit from the candidate system in terms of benefits and costs to the organization. The main aim of Economic Feasibility Analysis (EFS) is to estimate the economic requirements of candidate system before investments funds are committed to proposal. It prefers the alternative which will maximize the net worth of organization by earliest and highest return of funds along with lowest level of risk involved in developing the candidate system.

Operational Feasibility:

Operational Feasibility determines whether the system is operating effectively once it is developed and implemented. It ensures that the management should support the proposed system and its working feasible in the current organizational environment. It analyzes whether the users will be affected and they accept the modified or new business methods that affect the possibleSystem benefits. It also ensures that the computer resources and network architecture of candidate system are workable.

2.2 SYSTEM REQUIREMENTS

❖ Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, a hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

Architecture

All computer operating systems are designed for particular computer architecture. Most software applications are limited to particular operating systems running on particular architectures. Although architecture-independent operating systems and applications exist, most need to be recompiled to run on a new architecture. See also a list of common operating systems and their supporting architectures.

Processing power

The power of the central processing unit (CPU) is a fundamental system requirement for any software. Most software running on 86X architecture define processing power as the model and the clock speed of the CPU. Many other features of a CPU that influence its speed and power, like bus speed, cache, and MIPS are often ignored. This definition of power is often erroneous, as AMD Athlon and Intel Pentium CPUs at similar clock speed often have different throughput speeds. Intel Pentium CPUs have enjoyed a considerable degree of popularity, and are often mentioned in this category.

Memory

All software, when run, resides in the random access memory (RAM) of a computer. Memory requirements are defined after considering demands of the application, operating system, supporting software and files, and other running processes. Optimal performance of other unrelated software running on a multi-tasking computer system is also considered.

Secondary storage

Data storage device requirements vary, depending on the size of software installation, temporary files created and maintained while installing or running the software, and possible use of swap space (if RAM is insufficient).

Display adapter

Software requiring a better than average computer graphics display, like graphics editors and high-end games, often define high-end display adapters

Peripherals

To Some software applications need make extensive and/or special use of some peripherals, demanding the higher performance or functionality of such peripherals. Such peripherals include CD-ROMS, keyboard, mouse, network devices etc.

CPU	: Intel Core i3-380M
RAM	: 4 GB
HARD DISK	: 1TB
KEYBOARD MOUSE	:STANDERED
MOUSE	: TOUCH PAD

❖ SOFTWARE REQUIREMENTS

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Platform

A computing platform describes some sort of framework, either in hardware or software, which allows software to run. Typical platforms include a computer's architecture, operating system, or programming languages and their runtime libraries.Operating system is one of the requirements mentioned when defining system requirements (software). Software may not be compatible with different versions of same line of operating systems, although some measure of backward compatibility is often maintained. Similarly, software designed using newer features of Linux Kernel v2.6 generally does not run or compile properly (or at all) on Linux distributions using Kernel v2.2 or v2.4.

APIs and drivers

Software making extensive use of special hardware devices, like high-end display adapters, needs special API or newer device drivers. A good example is DirectX, which is a collection of APIs for handling tasks related to multimedia, especially game programming.

Web browser

Most web applications and software depend heavily on web technologies to make use of the default browser installed on the system. Microsoft Internet Explorer is a frequent choice of software running on Microsoft Windows, while use of ActiveX controls, despite their vulnerabilities.

OS PLATFORM	: Windows 10
FROND END	: ASP.NET IN C#.NET 2015
BACK END	: SQL Server
WEB SERVER	: ASP.NET Development Server

2.3 LANGUAGE SPECIFICATION

VISUALSTUDIO.NET

Visual Studio .NET is a Microsoft-integrated development environment (IDE) that can be used for developing consoles, graphical user interfaces (GUIS), Windows Forms, Web services and Web applications.

Visual Studio is used to write native code and managed code supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silver light. Visual Studio .NET's code editor supports IntelliSense and code refactoring, while the Visual Studio .NET integrated debugger supports both source and machine-level debugging. Visual Studio .NET includes other built-in tools, like a form designer, which is useful when building GUI applications, a Web designer that creates dynamic Web pages; a class designer that is used to create custom libraries, and a schema designer for database support.

Visual Studio .NET is a Microsoft-integrated development environment (IDE) that can be used for developing consoles, graphical user interfaces (GUIS), Windows Forms, Web services and Web applications.

USE OF .NET FRAMEWORK

- **Business Functions**

  At Those days are gone when businesses were conducted without the presence of a software tool in the background. Businesses, today, make constructive use of applications and software toStreamline their business processes. Be it supply chain management, sales, or finance; business software can work wonders. .NET is regularly and extensively used in the creation of such software like CRM, supply management app, and more.

- **Re-Designing**

  An organization doesn't remain static. Your needs can alter with them, and your software needs to scale with it. .NET has been found to be a great platform for re-designing current applications in order to make it line up with the growing needs of an organization.

- **Interoperable Apps**

  If one aspires to have an interoperable app that seamlessly brings together user experience across multiple platforms, .NET can again play a pivotal role. With the range of features on offer, .NET can help programmers create apps that work in sync across platforms.

- **Gaming**

  .NET is extremely versatile, and that makes it suitable for the development of gaming applications. Its versatility can also help in the creation of web and enterprise-graded applications.

- **Communication**

  Businesses need emails and chat platforms for seamless communication between employees. .NET is one of the most popular platforms used for emails & chat. It can integrate with your Outlook for a more holistic experience. Moreover, there are various other techniques that make your communication more fool proof,

- **Multi-Tiered Software Architecture**

  .NET makes use of multi-tiered software architecture. It is known as multi-tiered because it physically separates functions for presentation, app processing, and data management. It helpsDevelopers build flexible applications. Moreover, developers can also add or edit a layer without being required to rework on the entire app.

- **Cross Platform**

  Most expert developers consider .NET to be a modular and swift platform that can be used to create server apps that run equally well across Linux, Windows, and MAC.

- **Mobile Apps**

  It lets you create apps that work smoothly across computer, mobile, and other devices.

Features of .NET framework

Code editor is like any other IDE. The .NET framework includes a code editor that supports syntax highlighting and code completion using intelligence for variables, functions, methods, loops and Linux queries. This also supports settings bookmarks in code for quick navigation.

Debugger: Visual Studio includes debugger that works both as a source level debugger and as a machine level debugger. It works with both managed codes as well as negative codes.

Designer: Visual Studio includes a host of visual designers to aid in the development of applications. Those tools include; Windows presentation foundation, Web designer/development, Class designer, Data designer, Mapping designer, Extensibility.

.NET platform Neutral framework

- .NET provides a common set of class libraries which can be accessed from any .NET based programming language
- .NET is a layer between the operating system and the programming language.
- .NET supports many programming language including VB, c#, ASP, etc.

There will not be separate set of classes and libraries for each language.

Understanding .NET Framework

Microsoft started development on the .NET Framework in the late 1990s originally under the name of Next Generation Windows Services (NGWS). By late 2001 the first beta versions of .NET 1.0 were released. The first version of .NET Framework was released on 13 February 2002, bringing managed code to Windows NT 4.0, 98, 2000, ME and XP.

Since the first version, Microsoft has released nine more upgrades for .NET Framework, seven of which have been released along with a new version of Visual Studio. Two of these upgrades, .NET Framework 2.0 and 4.0, have upgraded Common Language Runtime (CLR). New versions of .NET Framework replace older versions when the CLR version is the same.

The .NET Framework family also includes two versions for mobile or embedded device use. A reduced version of the framework, the .NET Compact Framework, is available on Windows CE platforms, including Windows Mobile devices such as smart phone. Additionally, the .NET Micro Framework is targeted at severely resource-constrained devices. .NET Framework 4.8 was the final version of .NET Framework, future work going into the rewritten and cross-platform .NET Core platform, which shipped as .NET 5 in November 2020.

Benefits of .NET Framework

- **Object-Oriented Programming (OOP)**

  One of the best things about .NET is that it is based on object-oriented programming (OOP). This is where the software is divided into smaller chunks, which then allows developers to work on them one at a time. Once the work on one part is done, they can move on to the next. When all the smaller pieces are complete, they can then be combined and managed more effectively.

- **Great Caching System**

  The simplicity of the caching system of .NET makes it reliable and easy to come temporarily store data. What's more, it allows for the cache implementation to be customized, so developers can improve performance and scale as needed.

- **Visual Studio**

  .NET comes with the Visual Studio Integrated Development Environment (IDE). This tool allows developers to create applications without much fuss, as well as debug and publish them across multiple platforms and OS.

- **Cross-Platform Development**

  Although it was not when it was first launched, .NET has since become a cross-platform implementation that can be used to develop applications on multiple OS. For developers who prefer different programming languages, having them work will not be an issue. It is also open source, which means that developers can tinker with it as much as they want.

- **Easy to Deploy and Maintain**

The deployment of applications and their maintenance could not be done easier with the help of the .NET family of development tools. Thanks to its modular design, developers can literally take apart applications and then fix the ones that need fixing or updated and then put them back together. There is no need to wade through oceans of scripts just to find that one line that is making everything go haywire.

- **Universal .NET Standard**

  Thanks to the .NET Standard that was introduced back in 2016, developing applications using the .NET family means not having to redevelop the same applications for each new platform. This is thanks to the huge class library that is composed of practically of the functions that developers could ask for. This helps in such projects that require rendering graphics and interacting with databases. Manipulating XML documents has become a lot easier, as well.

- **Large community**

  The .NET has managed to build a huge community ever since it was launched, but it was not until it was made open-source by Microsoft that it really exploded. Thanks to the sheer ubiquity of the company's products and services, the development framework already had a huge influence. Now, it's even bigger.

- **Automatic monitoring in ASP.NET**

  One of the worst things that can happen during the coding process is when something goes wrong and you don't even notice or notice too late. Thanks to the automatic monitoring feature of ASP.NET, you will be notified if something like an infinite loop pops. The same goes for memory leaks or a bunch of other issues.

- **ASP.NET**

  ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices.ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation.ASP.NET is a part of Microsoft .Net platform. ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework.

  The ASP.NET application codes can be written in any of the following languages:

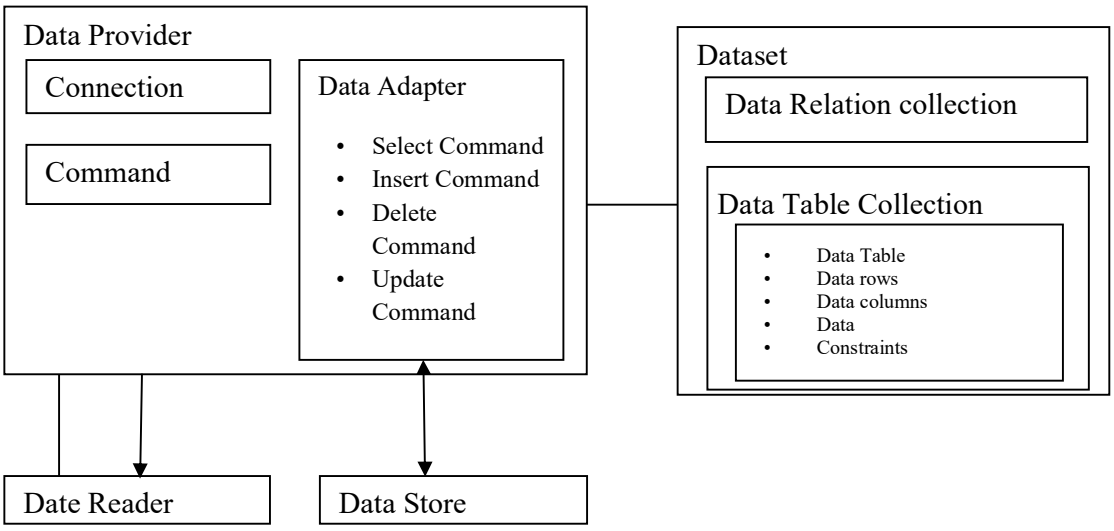
- C#
- Visual Basic. Net
- Jscript & J#

ASP.NET is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls such as text boxes, buttons, and labels for assembling, configuring, and manipulating code to create HTML pages.

- **ADO.NET**

ADO.NET provides a bridge between the front end controls and the back end database. The ADO.NET objects encapsulate all the data access operations and the controls interact with these objects to display data, thus hiding the details of movement of data.

The following figure shows the ADO.NET objects at a glance:



Architecture

ADO.NET is conceptually divided into consumers and data providers. 21 The consumers are the applications that need access to the data, and the providers are the software components that implement the interface and thereby provide the data to the consumer.

Functionality exists in Visual Studio IDE to create specialized subclasses of the Dataset classes for a particular database schema, allowing convenient access to each field in the schema through strongly typed properties. This helps catch more programming errors at

compile-time and enhances the IDE's Intelligence feature.A provider is a software component that interacts with a data source. ADO.NET data providers are analogous to ODBC drivers, JDBC drivers, and OLE DB providers.

ADO.NET providers can be created to access such simple data stores as a text file and spreadsheet, through to such complex databases as Oracle Database, Microsoft SQLServer, Myself, PostgreSQL, SQLite, IBM DB2, Sybase ASE, and many others. They can also provide access to hierarchical data stores such as email systems.

However, because different data store technologies can have different capabilities, every ADO. NET provider cannot implement every possible interface available in the ADO.NET standard. Microsoft describes the availability of an interface as "provider-specific," as it may not be applicable depending on the data store technology involved. Providers may augment theCapabilities of a data store; these capabilities are known as "services" in Microsoft parlance.

ADO.NET Components

- Windows Form: It is used to provide a user interface to a database driven application.
- Data Set: To store the data that retrieve from a data source.
- Data Adapter: To transfer data to and from a data source.
- Data Connection: To establish a connection between a dataset and a data source.
- Data Source: It generally represent database.

SQL SERVER

SQL SERVER is a relational database management system (RDBMS) developed by Microsoft. It is primarily designed and developed to compete with myself and Oracle database.

SQL Server supports ANSI SQL, which is the standard SQL (Structured Query Language) language. However, SQL Server comes with its own implementation of the SQL language, T-SQL (Transact-SQL).T-SQL is a Microsoft propriety Language known as Transact-SQL. It provides further capabilities of declaring variable, exception handling, stored procedure, etc.SQL Server Management Studio (SSMS) is the main interface tool for SQL Server, and it supports both 32-bit and 64-bit environments.SQL Server is sometimes referred to as MSSQL and Microsoft SQL server. It was originally released in 1989 as a

version 1.0 by Microsoft, in conjunction with Sybase, SQLServer and its versions where very similar to Sybase. However, the Microsoft Sybase partnership dissolved in the early 1991's, and Microsoft has released 2000, and 2008 versions which features more advanced options and better security.

SQL SERVER EDITIONS

SQL Server Enterprise: It is used in the high end, large scale and mission Critical business. It provides High-end security, Advanced Analytics, Machine Learning, etc.

SQL Server Standard: It is suitable for Mid-Tier Application and Data marts. It includes basic reporting and analytics

SQL Server WEB: It is designed for low total-cost-of-ownership options for Web hosters. It provides scalability, affordability, and manageability capabilities for small to large scale Web properties.

SQL Server Developer: It is similar to an enterprise edition for the non-production environment. It is mainly used for build, test, and demo.

SQL Server Express: It is for small scale applications and free to use.

SYSTEM DESIGN

3. SYSTEM DESIGN

System design is the process of defining the architecture, module, interface, and data for system to satisfy specified requirements. System design could overlap with the disciplines of system analysis, system architecture and system engineering.

At A systemic approach is required for a coherent and well-running system. Bottom-Up or Top-Down approach is required to take into account all related variables of the system. A designer uses the modeling languages to express the information and knowledge in a structure of system that is defined by a consistent set of rules and definitions. The designs can be defined in graphical or textual modeling languages.

3.1 ARCHITECTURE DESIGN

Architecture design of a system defines the system behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structure and behaviors of the system.

A system architecture can consist of system components and sub-system developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADL).

3.2 LOGICAL DESIGN

Logical design pertains to an abstract representation of the data flow, inputs, and outputs of the system. It describes the inputs (sources), outputs (destinations), databases (data stores), procedures (data flows) all in a format that meets the user requirements.

While preparing the logical design of a system, the system analyst specifies the user needs at level of detail that virtually determines the information flow into and out of the system and the required data sources. Data flow diagram, E-R diagram modeling are used.

3.3 PHYSICAL DESIGN

Physical design relates to the actual input and output processes of the system. It focuses on how data is entered into a system, verified, processed, and displayed as output.

It produces the working system by defining the design specification that specifies exactly what the candidate system does. It is concerned with user interface design, process design, and data design.

It consists of the following steps -

- Specifying the input/output media, designing the database, and specifying backup procedures.
- Planning system implementation.
- Devising a test and implementation plan, and specifying any new hardware and software.
- Updating costs, benefits, conversion dates, and system constraints.

a) INPUT DESIGN

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output.

Well-designed input forms and screens have following properties --

- . It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.

Input design requirements

1. To produce a cost effective method of input.
2. To achieve the high level of accuracy.
3. To ensure that the input is acceptable to and understand by the user.
4. Four objectives guiding the design of input focus on:
5. Controlling the amount of input requirement
6. Controlling error.
7. Avoiding delay
8. Keeping the step simple.

9. Input design consideration
10. The nature of input processing...
11. Flexibility and thoroughness of validation rule
12. Handling of properties within inputs...

b) OUTPUT DESIGN

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

Objectives of Output Design

The objectives of input design are --

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

3.4 DATABASE DESIGN

Database design defines the database structure used for planning, storing, and managing information. Accuracy in data can only be accomplished if a database is designed to store only valuable and necessary information.

A well-designed database is imperative in guaranteeing information consistency, eliminating redundant data, efficiently executing queries, and improving the performance of the database. Meticulously designing a database saves you from wasting time and getting frustrated during the database development phase. A good database design also allows you to easily access and retrieve data whenever needed.

The reliability of data depends on the table structure; whereas creating primary and unique keys guarantees uniformity in the stored information. Data replication can be avoided by forming a table of probable values and using a key to denote the value. So, whenever the value changes, the alteration happens only once in the main table.

As the general performance of a database depends on its design, a good database design uses simple queries and faster implementation. It is easy to maintain and update; whereas fixing trivial interruptions in a poor database design may harm stored events, views, and utilities. The organization of the data in a database aims to achieve five major objectives:

a. Data integrity
b. Data independency
c. Control redundancy
d. Privacy and security
e. Performance

Several degrees of normalization have to be applied during the process of table design. The major aim of the process of normalization is to reduce data redundancy and prevent loosingdata integrity. Redundancy refers to unwanted and unnecessary repetition of data. Data integrity has to be converted to all levels. Poor normalization can cause problems related to storage and retrieval of data. During the process of normalization which identifies causes serious problems, during deletion and updating.

Normalization also helps in simplifying of structure of the table. The theme behind a database is to handle information as an integrated whole that makes access to information easy, quick inexpensive and flexible for users. The entire package depends on how the data are maintained in the system. Each table has been designed with a perfect vision.

a) NORMALIZATION

Normalization is the process of reorganizing tables in such a way that it decreases data redundancy and dependency. Larger tables are divided into smaller tables and are linked together using relationships. Normalization is important for many reasons, but chiefly because it allows database to take up a little disk space as possible, resulting in increased performance. Normalization is also known as data normalization.

There are three main types of normalization: (“NF” refers to "normal form”)

1. INF
2. 2NF
3. 3NF

The following normal forms exist and are rarely used:

1. BCNF
2. 4NF
3. SNF

FIRST NORMAL FORM (1NF)

If a relation contains composite or multi-valued attribute, it violates first normal form or a relation is in first normal form if it does not contain any composite or multi-valued attribute. A relation is in first normal form if every attribute in that relation is singled valued attribute.

For a table to be in the First Normal Form, it should follow the following 4 rules:

Rule 1: Single Valued Attributes

Each column of your table should be single valued which means they should not contain multiple values. We will explain this with help of an example later; let's see the other rules for now.

Rule 2: Attribute Domain should not change

This is more of a "Common Sense" rule. In each column the values stored must be of the same kind or type.

For example: If you have a column 'dob' to save date of births of a set of people, then you cannot or you must not save 'names' of some of them in that column along with 'date of birth of others in that column. It should hold only 'date of birth' for all the records/rows.

Rule 3: Unique name for Attributes/Columns

This rule expects that each column in a table should have a unique name. This is to avoid confusion at the time of retrieving data or performing any other operation on the stored data.

If one or more columns have same name, then the DBMS system will be left confused.

Rule 4: Order doesn't matters

This rule says that the order in which you store the data in your table doesn't matter.

SECOND NORMAL FORM (2NF)

Second Normal Form (2NF) is based on the concept of full functional dependency. Second Normal Form applies to relations with composite keys, that is, relations with a primary key composed of two or more attributes. A relation with a single-attribute primary key is automatically in at least 2NF. A relation that is not in 2NF may suffer from the update anomalies.

To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency. A relation is in 2NF if it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

THIRD NORMAL FORM (3NF)

A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.

A relation is in 3NF if at least one of the following conditions holds in every non-trivial function dependency X -> Y:

1. X is a super key
2. Y is a prime attribute (each element of Y is part of some candidate key).

The normalization of 2NF relations to 3NF involves the removal of transitive dependencies. If a transitive dependency exists, we remove the transitively dependent attribute(s) from the relation by placing the attribute(s) in a new relation along with a copy of the determinant.

a) Tables

Table 1: admin

Field name	Type	Width	Constraints	Description
Id	varchar	30	Primary key	Username
pswd	varchar	30	NOT NULL	Password

Table 2: user1

Field name	Type	Width	Constraints	Description
id	varchar	30	Primary key	Username
pswd	varchar	30	NOT NULL	Password
name	varchar	30	NOT NULL	Name
pno	varchar	30	NOT NULL	Phone no
email	varchar	30	NOT NULL	Email

Table 3: muninfo

Field name	Type	Width	Constraints	Description
munid	varchar	30	Primary key	Municipality id
mname	varchar	30	NOT NULL	Municipality name
district	varchar	30	NOT NULL	District
address	varchar	30	NOT NULL	Address
phone	varchar	30	NOT NULL	Phone
email	varchar	30	NOT NULL	Email
fax	varchar	30	NOT NULL	Fax
chairman	varchar	30	NOT NULL	Chairman
contact	varchar	30	NOT NULL	Contact
username	varchar	30	NOT NULL	Username
password	varchar	30	NOT NULL	Password
hintq	varchar	30	NOT NULL	Hint Question
hinta	varchar	30	NOT NULL	Hint Answer

Table 4: mundep

Field name	Type	Width	Constraints	Description
depid	varchar	30	Primary key	Department id
depame	varchar	30	NOT NULL	Department name
head	varchar	30	NOT NULL	Head Name
desig	varchar	30	NOT NULL	Designation
dor	varchar	30	NOT NULL	Date of Registration
regno	varchar	30	NOT NULL	Registration No.
address	varchar	30	NOT NULL	Address
email	varchar	30	NOT NULL	Email
fax	varchar	30	NOT NULL	Fax
username	varchar	30	NOT NULL	Username
password	varchar	30	NOT NULL	Password
hq	varchar	30	NOT NULL	Hint Question
ha	varchar	30	NOT NULL	Hint Answer

Table 5: tax

Field name	Type	Width	Constraints	Description
fm	varchar	30	NOT NULL	From
t	varchar	30	NOT NULL	To
hyt	varchar	30	NOT NULL	Half Yearly Tax
deffect	varchar	30	Primary key	Date of Effect

Table 6: bank

Field name	Type	Width	Constraints	Description
acnt	varchar	30	Primary key	Account number
Bnk	varchar	30	NOT NULL	Bank
Ifc	varchar	30	NOT NULL	IFSC
amnt	varchar	30	NOT NULL	Amount

Table 7: ptei

Field name	Type	Width	Constraints	Description
empid	varchar	30	Primary key	Employee id
empname	varchar	30	NOT NULL	Employee name
sop	varchar	30	NOT NULL	Scale of Pay
depart	varchar	30	NOT NULL	Department
bp	varchar	30	NOT NULL	Basic Pay
da	varchar	30	NOT NULL	DA%
doj	varchar	30	NOT NULL	Date of Join
deb	varchar	30	NOT NULL	Date entered By
designt	varchar	30	NOT NULL	Designation
ed	varchar	30	NOT NULL	Entered Date
d	varchar	30	NOT NULL	DA Total
t	varchar	30	NOT NULL	Total

3.5 SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS)

The production of the requirements stage of the software development process is Software Requirements Specifications (SRS) (also called a requirements document). This report lays a foundation for software engineering activities and is constructing when entire requirements are elicited and analysed. SRS is a formal report, which acts as a representation of software that enables the customers to review whether it (SRS) is according to their requirements.

The SRS is a specification for a specific software product, program, or set of applications that perform particular functions in a specific environment. It serves several goals depending on who is writing it. First, the SRS could be written by the client of a system. Second, the SRS could be written by a developer of the system. The two methods create entirely various situations and establish different purposes for the document altogether. The first case, SRS, is used todefine the needs and expectation of the users. The second case, SRS, is written for various purposes and serves as a contract document between customer and developer.

Characteristics of good SRS

1. Correctness: User review is used to provide the accuracy of requirements stated in the SRS. SRS is said to be perfect if it covers all the needs that are truly expected from the system.

2. Completeness: The SRS is complete if, and only if, it includes the following elements:

- All essential requirements, whether relating to functionality, performance, design, constraints, attributes, or external interfaces.
- Definition of their responses of the software to all realizable classes of input data in all available categories of situations.
- Full labels and references to all figures, tables, and diagrams in the SRS and definitions of all terms and units of measure.

3. Consistency: The SRS is consistent if, and only if, no subset of individual requirements described in its conflict. There are three types of possible conflict in the SRS:

a) The specified characteristics of real-world objects may conflict. For example,

- The format of an output report may be described in one requirement as tabular but in another as textual.
- One condition may state that all lights shall be green while another states that all lights shall be blue.

b) One requirement may determine that the program will add two inputs, and another may determine that the program will multiply them. One condition may state that "A" must always follow "B," while other requires that "A and B" co-occurs.

c) Two or more requirements may define the same real-world object but use different terms for that object. For example, a program's request for user input may be called a "prompt" in one requirement's and a "cue" in another. The use of standard terminology and descriptions promotes consistency

4. Unambiguousness: SRS is unambiguous when every fixed requirement has only one interpretation. This suggests that each element is uniquely interpreted. In case there is a method used with multiple definitions, the requirements report should determine the implications in the SRS so that it is clear and simple to understand.

5. Ranking for importance and stability: The SRS is ranked for importance and stability if each requirement in it has an identifier to indicate either the significance or stability of that particular requirement.

Typically, all requirements are not equally important. Some prerequisites may be essential, especially for life-critical applications, while others may be desirable. Each element should be identified to make these differences clear and explicit. Another way to rank requirements is to distinguish classes of items as essential, conditional, and optional.

6. Modifiability: SRS should be made as modifiable as likely and should be capable of quickly obtain changes to the system to some extent. Modifications should be perfectly indexed and cross-referenced.

7. Verifiability: SRS is correct when the specified requirements can be verified with a cost-effective system to check whether the final software meets those requirements. The requirements are verified with the help of reviews.

8. Traceability: The SRS is traceable if the origin of each of the requirements is clear and if it facilitates the referencing of each condition in future development or enhancement documentation.

There are two types of Traceability:

a) Backward Traceability: This depends upon each requirement explicitly referencing its source in earlier documents.

b) Forward Traceability: This depends upon each element in the SRS having a unique name or reference number.

The forward traceability of the SRS is especially crucial when the software product enters the operation and maintenance phase. As code and design document is modified, it is necessary to be able to ascertain the complete set of requirements that may be concerned by those modifications.

9. Design Independence: There should be an option to select from multiple design alternatives for the final system. More specifically, the SRS should not contain any implementation details.

10. Testability: An SRS should be written in such a method that it is simple to generate test cases and test plans from the report.

11. Understandable by the customer: An end user may be an expert in his/her explicit domain but might not be trained in computer science. Hence, the purpose of formal notations and symbols should be avoided too as much extent as possible. The language should be kept simple and clear.

12. The right level of abstraction: If the SRS is written for the requirements stage, the details should be explained explicitly. Whereas, for a feasibility study, fewer analysis can be used. Hence, the level of abstraction modifies according to the objective of the SRS.

a) TESTING OF SRS AND POINT TO BE NOTICED WHILE TESTING

Correctness of SRS should be checked: Since the whole checking phase is dependent on SRS, it is very important to check its correctness. There are some standards with which we can compare and verify.

Ambiguity should be avoided: Sometimes in SRS, some words have more than one meaning and this might confuse testers making it difficult to get the exact reference. It is advisable to check for such ambiguous words and make the meaning clear for better understanding.

Requirements should be completed: When testers write test cases, what exactly is required from the applications, is the first thing that needs to be cleared. For e.g. if application needs to send the specific data of some specific size then it should be clearly mentioned in SRS that how much data and what is the size limit to send.

Consistent requirements:

The SRS should be consistent within itself and consistent to its reference documents. If u call an input “start and stop” in one place, don't call it "start/stop" in another. This sets the standard and should be followed throughout the testing phases.

Verifications of expected results: SRS should not have statements like "work as expected”, it should be clearly stated that what is expected since different testers would have different thinking aspects and may draw different results from this statement.

DATA FLOW DIAGRAMS

3.6 DATA FLOW DIAGRAMS

Also known as DFD, Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation,

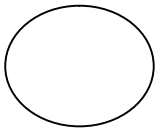
Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of logical data flow.

DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expands it to a hierarchy of detailed diagrams.

DataFlow

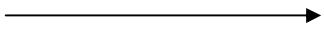
Data Source

Process

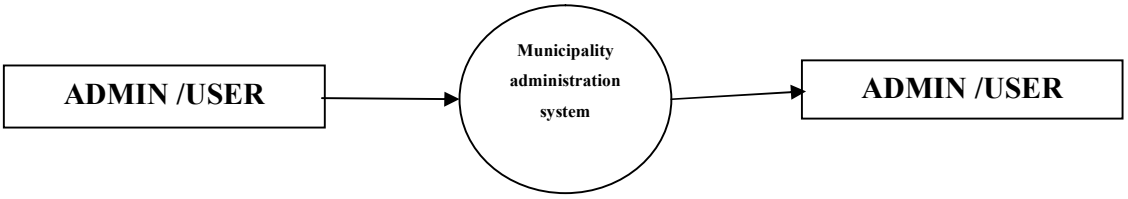


Input/output

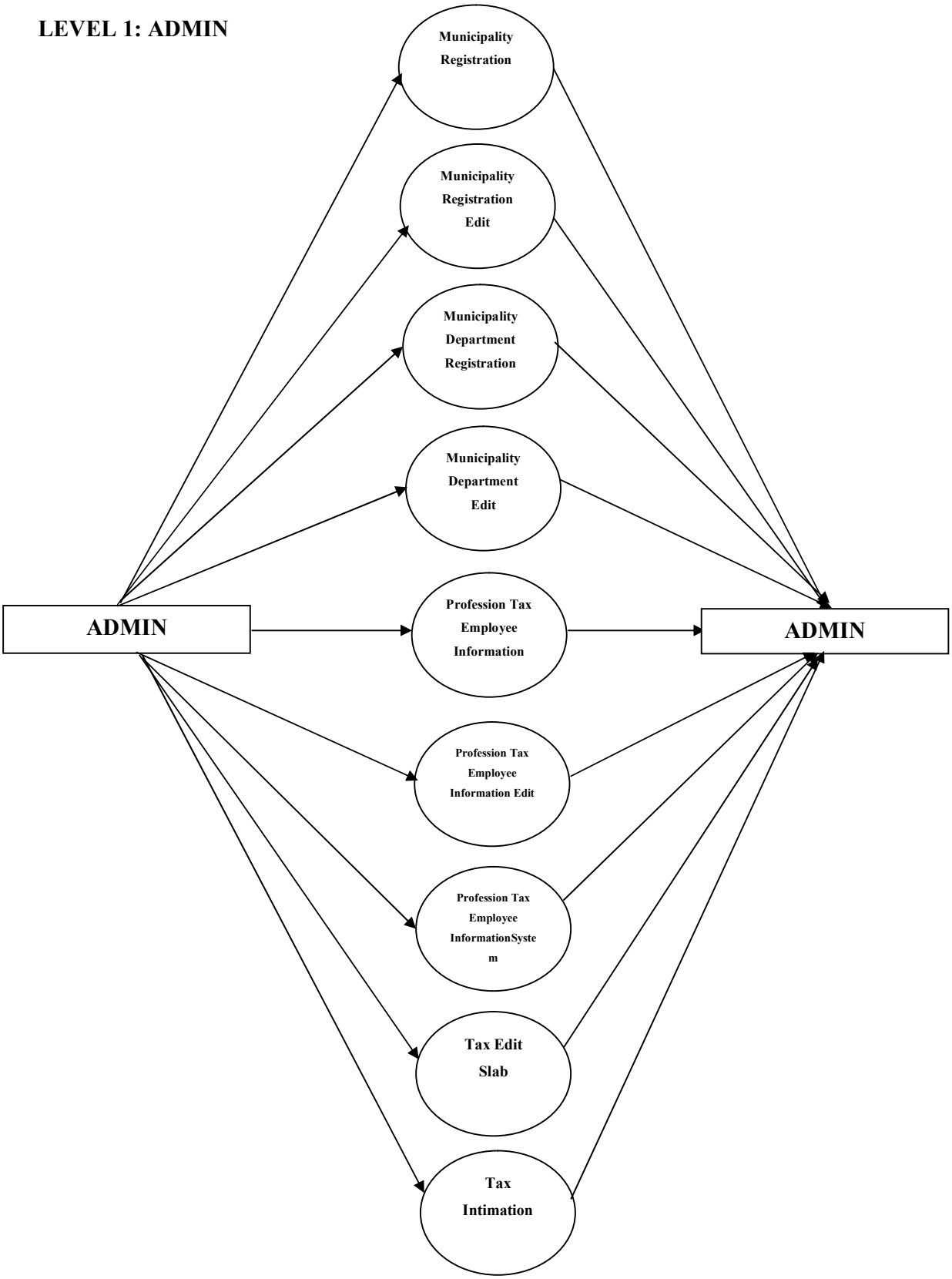
Flow



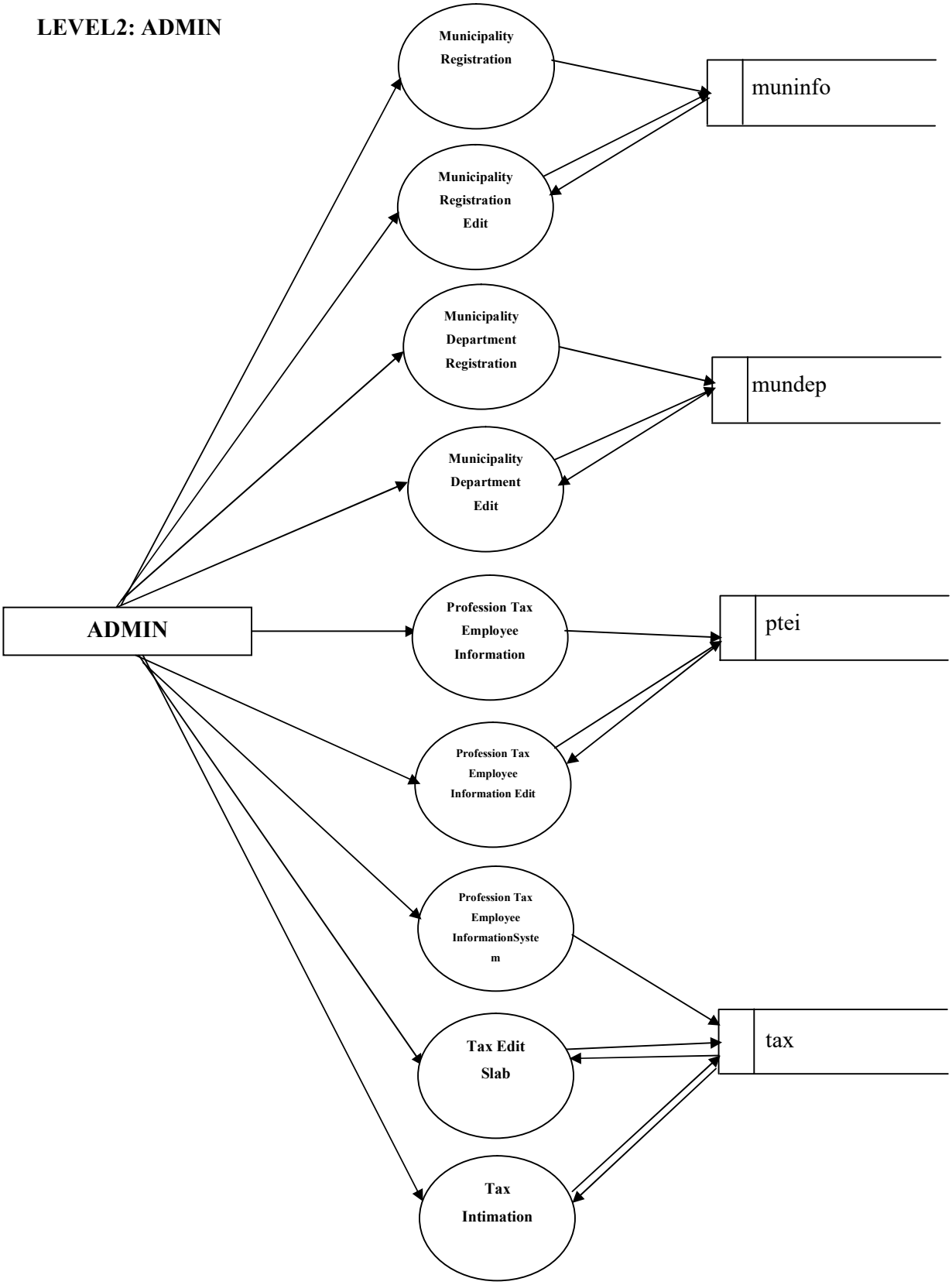
LEVEL 0:



LEVEL 1: ADMIN

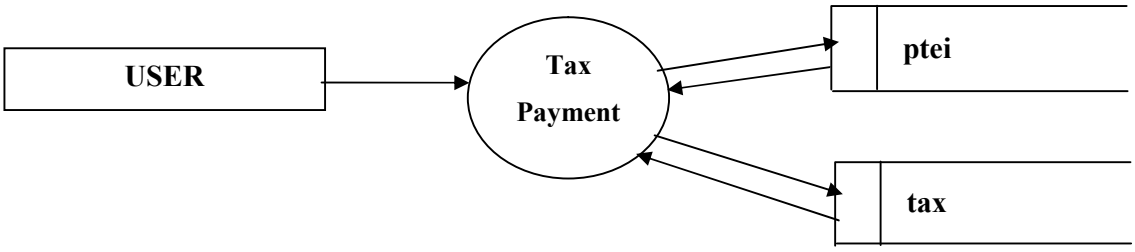


LEVEL2: ADMIN



LEVEL 1: USER

LEVEL 2: USER



SYSTEM TESTING AND IMPLIMENTATION

4. SYSTEM TESTING AND IMPLIMENTATION

4.1 System Testing

System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end to end system specifications. Usually, the software is only one element of a large computer-based system. Ultimately, the software is interfaced with other software/hardware system. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

Component or unit testing

- Invalid components are tested independently;
- Components may be functions or objects or coherentgrouping of these entities.

System Testing

- Testing of the system as a whole. Testing of emigrant properties is particularly important.

Acceptance

- Testing with costumer data to check that the system meets the customer needs.

integrated testing

Integration testing is a systematic testing for conducting tests to uncover errors associated within the interface. The objective is to take a unit tested module and build a program structure. Here, connection is difficult because the vast expense of the entire program complicates the isolation of causes. some of the incremented integration testing strategies are as follows:

1. Top-down: Top-down integration testing is an incremental approach to construction of program structure. Modules are integrated by moving downward through the control, hierarchy, beginning with main control module. Modules subordinate to the main control module are incorporated into the structure in either a depth first or breadth first manner.

2. Bottom-up: Bottom-up integration testing begins construction with automatic module. Because components are integrated from bottom-up processing required for the components subordinate to a given level is always available. Each time a new module is added as a part of integration testing, the software changes. These changes may cause problem with function they previously worked flawlessly.
3. System testing: In this phase, the entire software system was tested. After integration testing, the entire software was tested again by various clients. The software has been tested for its functionality as well as limitations. The various interfaces developed were thoroughly debugged and found to be working correctly. The main types of system tests are:

    1. Recovery testing: Recovery testing is a test those forces the software to fail in a verity of ways and that recovery is properly performed.
    2. Security Testing: Security testing attempts to verify that protection mechanisms built into a system will fact protect it from improper penetrations.
    3. Stress Testing: Stress testing execute a system in a manner that demands resources in abnormal quantity, frequency or volume.
    4. Performance Testing: Performance testing is designed to test the runtime performance of software within the context of an integrated system.

Output testing:

After performing the validation testing, the test step is output testing of the proposed system, since no system could be useful if it does not produce the required output in specific format. Asking the user about the required by them tests the outputs generated or displayed by the system under consideration. Here, the output format is considered in two ways. One is on screen and another is printed format. The output format on the screen is found to be correct as the format was designed in the system design phase according to the user needs.

User Acceptance Testing:

User acceptance of a system is key factor to the success of any system. The system under consideration was tested for user acceptance by constantly keeping in touch with the prospected system user at the time of followingpoints:

1. Input screen design.

2. Output screen design.
3. Online message to guide the user.
4. Format of the reports and other outputs.

Validation Testing

The goal of validation testing is to demonstrate the software trace ability to the requirements. There are different types of validation. They are as following:

a. Required Field Validation.
b. Range Validation.
c. Regular Express Validation.
d. Format of the reports and other outputs.

VALIDATION CONTROL	DESCRIPTION
Required Validation	Makes an input control a required field.
Range Validation	Checks that user enters a value that falls between two values.
Regular Expression Validation	Ensures that the value of an input control matches specified pattern.
Compare Validation	Compare the value of one input control to the value of another control.
Validation Summary	Display report of all validation errors occurred in webpage.

4.2 TESTING METHODS

BLACK BOX TESTING

Black Box Testing a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

The above Black-Box can be any software system you want to test. For example, an operating system like Windows, a website like Google, a database like Oracle or even your own custom application. Under Black Box Testing, you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

Advantages of Black Box Testing

- Efficient when used on large systems.
- Since the tester and developer are independent of each other, testing is balanced and unprejudiced.
- Tester can be non-technical.
- There is no need for the tester to have detailed functional knowledge of system. Tests will be done from an end user's point of view, because the end user should accept the system. (This testing technique is sometimes also called Acceptance testing.) identify vagueness and contradictions in functional specifications.
- Testing helps to Test cases can be designed as soon as the functional specifications are complete.

WHITE BOX TESTING

White box testing is a testing technique that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

Advantages of White Box Testing:

- Forces test developer to reason carefully about implementation.
- Reveals errors in "hidden" code.
- Spots the Dead Code or other issues with respect to best programming practices.

4.3 IMPLEMENTATION

System implementation projects are long difficult journeys by which organizations move from an old set of technology/methods/procedures to a new one. A software implementation method is a systematic structure approach to effectively integrate software based service or component into the workflow of an organizational structure or an individual

end-user. The complexity of implementing product software differs on several issues. Example are: the number of end users that will use the product software, the effects that the implementation has on changes on task and responsibilities for the end user, the culture and integrity of the organization where the software is going to be used and the budget available. It is vital to select the right strategy for implementing the application to assure successful results.

Implementing Strategy

1. Direct implementation

With this method of implementation, the users stop using the manual system and start using the computer system from a given date. The advantage of this method is that it is less costly in effort and time than any other method of implementation. The disadvantage of this method is that if problems occur the users do not have any alternative part from returning to a manual system which may prove difficult if it has been disconnected.

2. Parallel Running

With parallel running, the new system is introduced alongside the existing system. With parallel running both systems (manual and computer, or old computer and new computer system) will be in operation at the same time. This has theadvantage that the result from the new system can be compared at the same time. This has the advantage that the result from the system can be with those of the old system. However, it has the major disadvantage that each job is done twice and therefore it means a lot of extra work for the user.

3. Phase implementation

The phase implementation method enables us to break our project into smaller milestones. Major disadvantages are difficult to achieve due to interdependencies between modules

SYSTEM MAINTENANCE

5. SYSTEM MAINTENANCE

The result obtained from the evaluation process help the organization to determine whether its information system is effective and efficient or otherwise. The process of monitoring, evaluating, and modifying of existing information system to make required or desirable improvements may be termed as System Maintenance. System Maintenance is an ongoing activity, which covers a wide verity of activities, including removing program and design errors, updating documentation and test data and updating user support. For the purpose of convenience, maintenance may be categorized into three classes, namely:

Corrective maintenance:

This type of maintenance implies removing errors in a program, which might have crept in the system due to faulty design or wrong assumptions. Thus, in corrective maintenance, processing or performing failures are repaired.

Adaptive Maintenance:

In adaptive maintenance, program functions are changed enables the information system to satisfy the information needs of the user. This maintenance may become necessary because of organizational changes which include

5.2.1 Change in the organizational procedures.

5.2.2 Change in organizational objectives, goals, policies, etc.

5.2.3 Change in forms

5.2.4 Change in information needs of managers

5.2.5 Change in system controls and security needs, etc.

Perfective Maintenance:

Perfective maintenance means adding new programs or modify the existing programs to enhance the performance of the information system. This type of maintenance undertaken to respond to user's additional needs which may be due to the changes within or outside of the organization. Outside changes are primarily environmental changes, which may vary in the

absence of system maintenance, render the information system ineffective and inefficient. These environmental changes include:

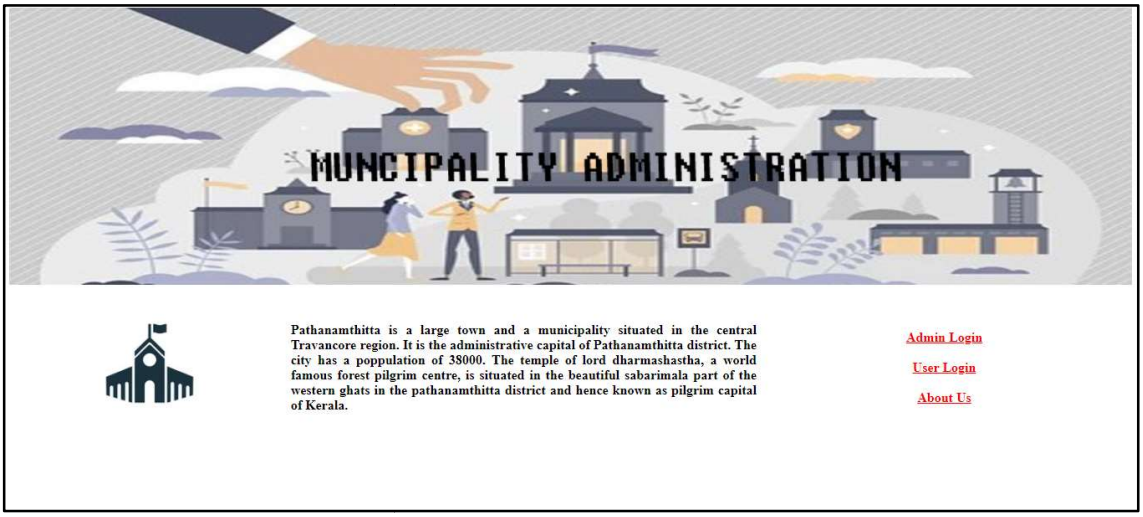
a. Changes in governmental policies, law, etc.
b. Economic and competitive conditions and new technology.

APPENDIX

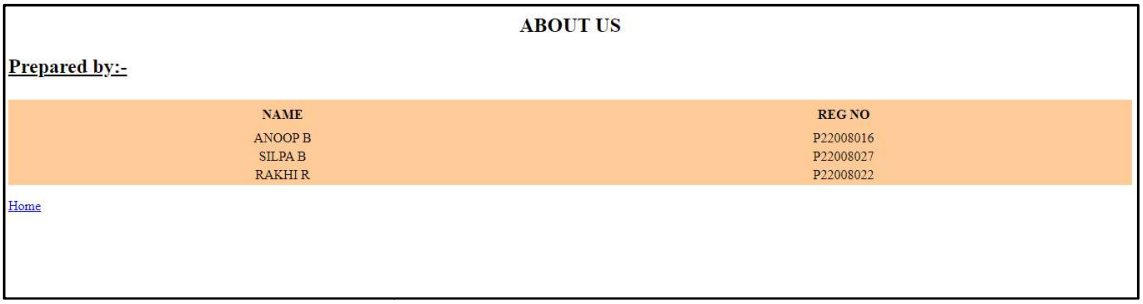
6. APPENDIX

6.1 SCREENSHOTS

index



aboutus



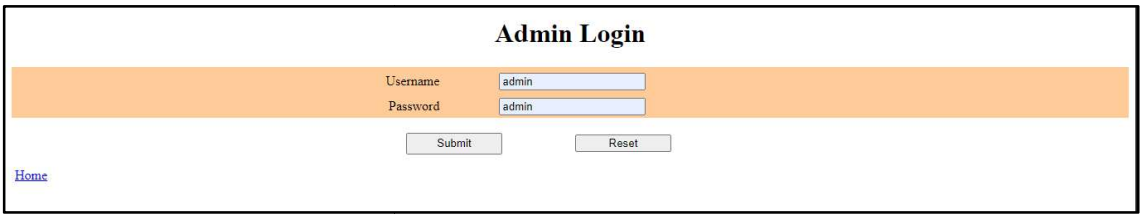
userlogin

user

homeuser

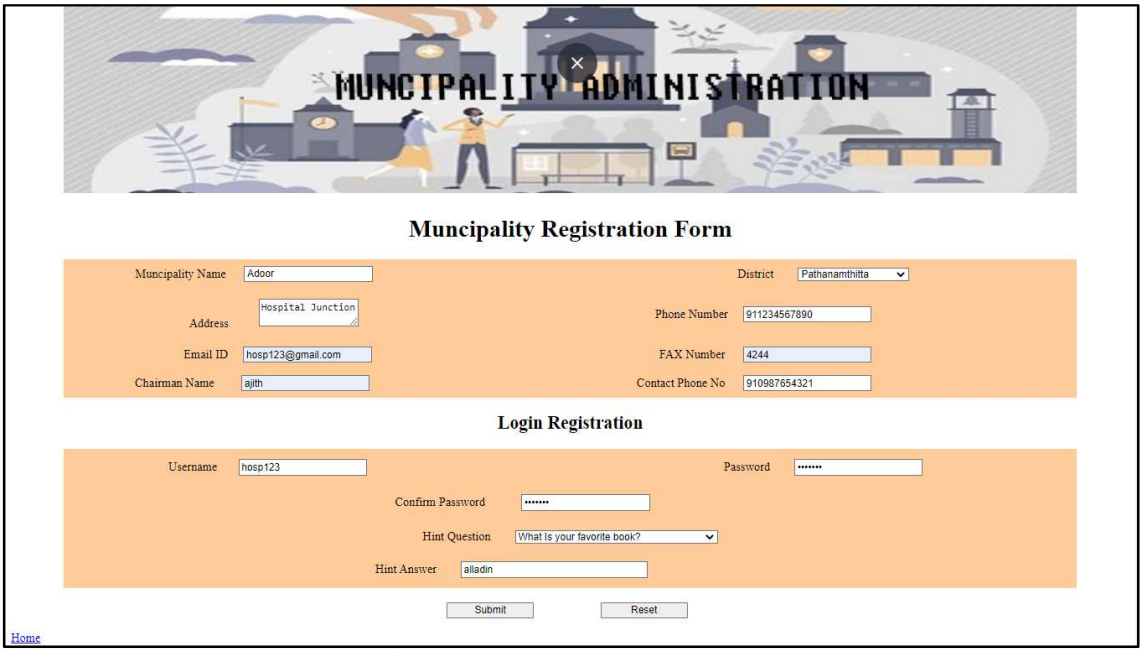
taxpay

adminlogin



home

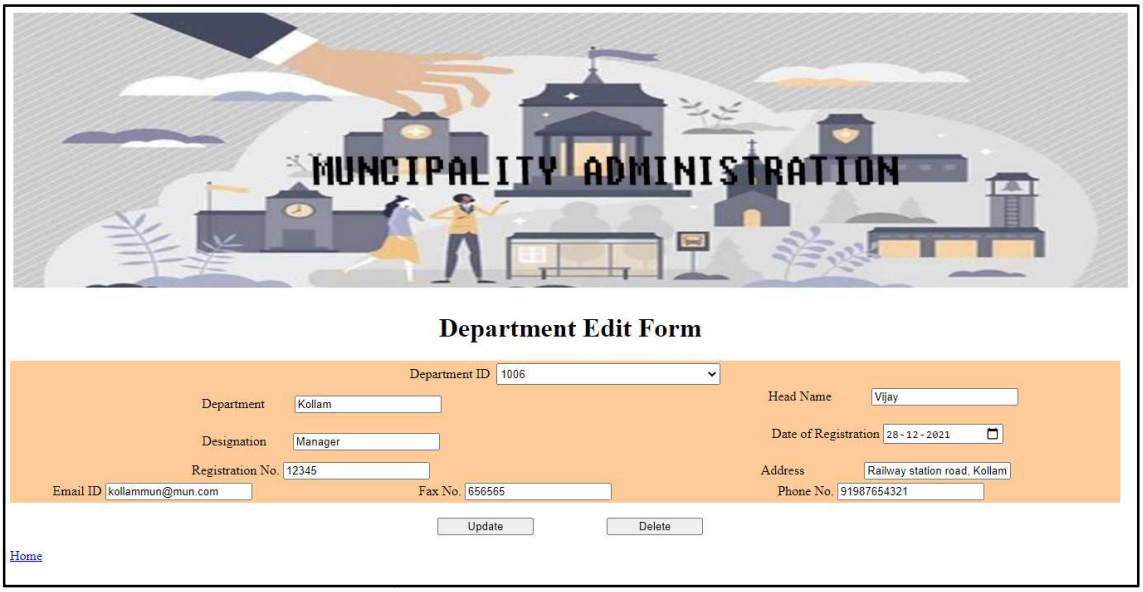
muninfo



muniedit

mundep

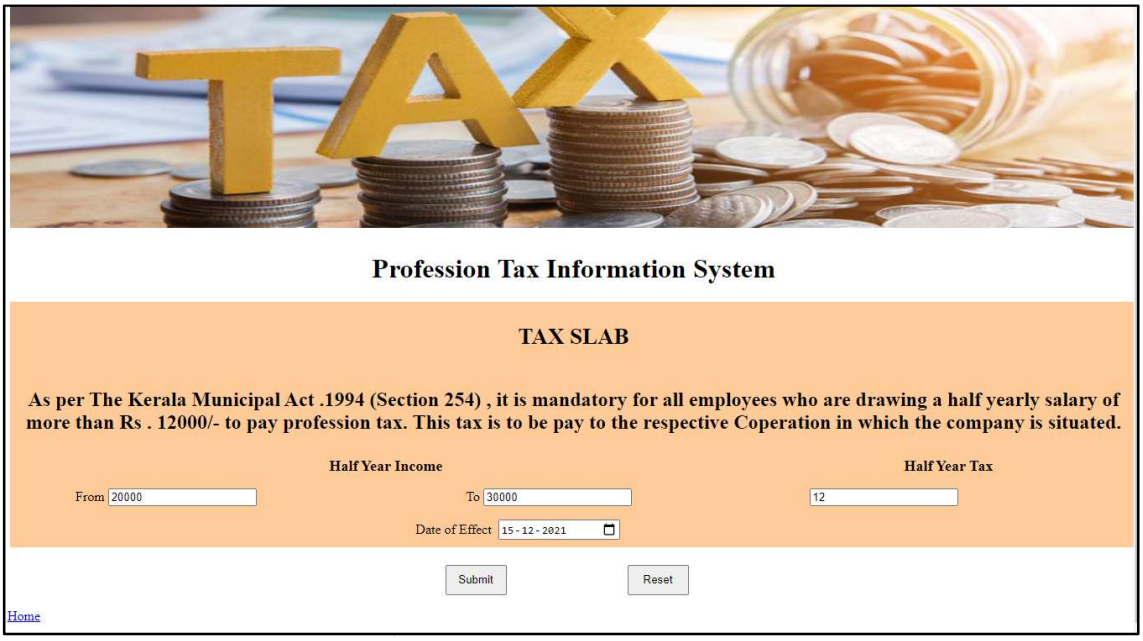
mundepedt



pteinfo

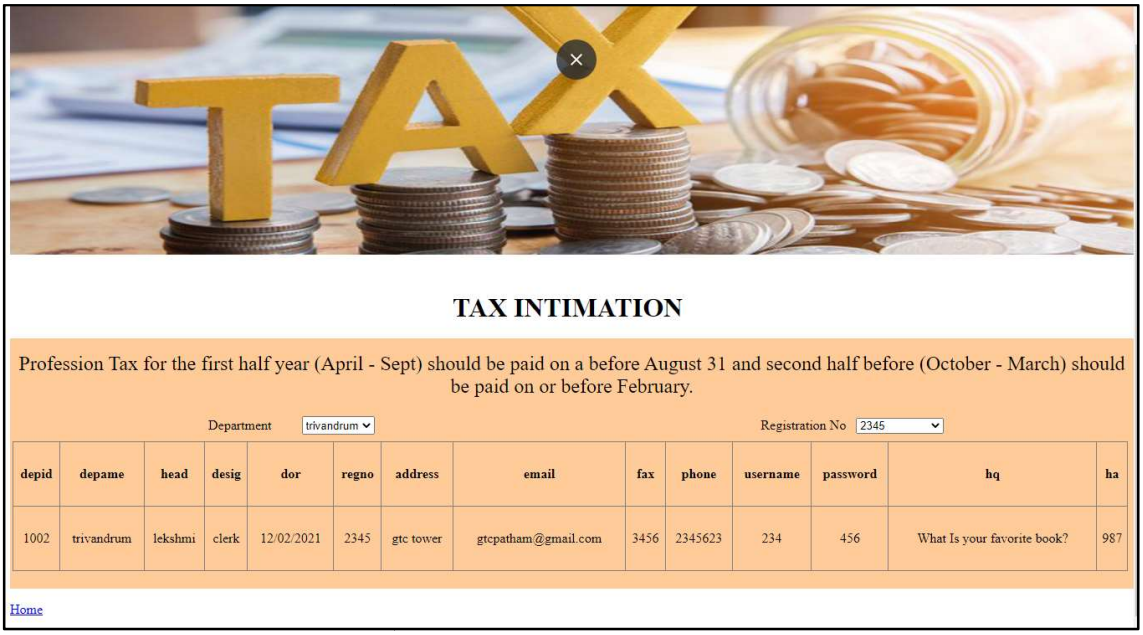
ptedit

taxinfo



taxedit

taxinti



6.2 SOURCE CODE

1.1. index.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class index :System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
   }
   protected void LinkButton1_Click(object sender, EventArgs e)
   {
Response.Redirect("adminlogin.aspx");
   }
   protected void LinkButton2_Click(object sender, EventArgs e)
   {
Response.Redirect("userlogin.aspx");
   }
   protected void LinkButton3_Click(object sender, EventArgs e)
   {
Response.Redirect("aboutus.aspx");
   }
     }
```

1.2. aboutus.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class aboutus :System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
   }
      protected void LinkButton2_Click1(object sender, EventArgs e)
   {
Response.Redirect("index.aspx");
```

```
    }
}
```

1.3. userlogin.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;
using System.Data;
using System.Data.SqlClient;

public partial class userlogin :System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();

SqlDataAdapterobjdataadapter = new SqlDataAdapter("select * from user1", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "user1");
int f = 0;
foreach (DataTable table in objdataset.Tables)
          {
foreach (DataRow row in table.Rows)
              {
                 if
(TextBox6.Text.ToString().ToUpper().Trim().Equals(row["id"].ToString().ToUpper().Trim()
) &&
(TextBox7.Text.ToString().ToUpper().Trim().Equals(row["pswd"].ToString().ToUpper().Tri
m())))
                 {
                    f = 1;
Response.Redirect("homeuser.aspx");
                    break;
                 }
              }
              if (f == 1)
                 break;
          }
          if (f == 0)
```

```
Response.Write("<marquee> sorry user not found</marquee>");
        }
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        TextBox6.Text = " ";
        TextBox7.Text = " ";
    }
        protected void LinkButton2_Click1(object sender, EventArgs e)
    {
Response.Redirect("index.aspx");
    }
    protected void LinkButton3_Click1(object sender, EventArgs e)
    {
Response.Redirect("user.aspx");
    }
}
```

1.4. user.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;
using System.Data;
using System.Data.SqlClient;

public partial class user :System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlCommand s = new SqlCommand();
s.Connection = o;
s.CommandText = "insert into user1(id,pswd,name,pno,email) values('" +
TextBox9.Text.ToString() + "','" + TextBox10.Text.ToString() + "','" +
TextBox13.Text.ToString() + "','" + TextBox12.Text.ToString() + "','" +
TextBox11.Text.ToString() + "')";
s.ExecuteNonQuery();
    }
    protected void Button2_Click(object sender, EventArgs e)
```

```
    {
        TextBox9.Text = " ";
        TextBox10.Text = " ";
        TextBox13.Text = " ";
        TextBox12.Text = " ";
        TextBox11.Text = " ";
    }
    protected void LinkButton3_Click(object sender, EventArgs e)
    {
Response.Redirect("userlogin.aspx");
    }
    protected void LinkButton2_Click1(object sender, EventArgs e)
    {
Response.Redirect("index.aspx");
    }
}
```

1.5. userhome.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class homeuser :System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void LinkButton1_Click(object sender, EventArgs e)
    {
Response.Redirect("index.aspx");
    }
}
```

1.6. taxpay.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
```

```
public partial class taxpay :System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      if (!IsPostBack)
      {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlDataAdapterobjdataadapter = new SqlDataAdapter("select* from ptei", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "ptei");
foreach (DataTable table in objdataset.Tables)
         {
foreach (DataRow row in table.Rows)
            {
DropDownList1.Items.Add(row["depart"].ToString());
DropDownList3.Items.Add(row["empid"].ToString());
            }
         }
      }
   }
   protected void Button1_Click(object sender, EventArgs e)
   {
int amnt1 = int.Parse(Label14.Text.ToString());
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlDataAdapterobjdataadapter = new SqlDataAdapter("select * from dummy where acnt='" +
TextBox5.Text.ToString() + "'", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "ptei");
int f = 0;
foreach (DataTable table in objdataset.Tables)
      {
foreach (DataRow row in table.Rows)
         {
            if
(TextBox5.Text.ToString().ToUpper().Trim().Equals(row["acnt"].ToString().ToUpper().Trim
()))
            {
               f = 1;
               string a = row["amnt"].ToString();
int amnt2 = int.Parse(a.ToString());
int d = amnt2 - amnt1;
SqlCommand s = new SqlCommand();
s.Connection = o;
s.CommandText = "update dummy set amnt='" + d.ToString() + "'where acnt='" +
TextBox5.Text.ToString() + "'";
```

```
s.ExecuteNonQuery();
                break;
            }
            if (f == 1)
                break;
        }
    }
}
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlDataAdapterobjdataadapter = new SqlDataAdapter("select * from ptei where depart='" +
DropDownList1.Text.ToString() + "'", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "ptei");
}
protected void Button2_Click(object sender, EventArgs e)
{
    TextBox5.Text = " ";
    TextBox6.Text = " ";
    TextBox7.Text = " ";
    TextBox8.Text = " ";
}
protected void DropDownList3_SelectedIndexChanged(object sender, EventArgs e)
{
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlDataAdapterobjdataadapter = new SqlDataAdapter("select * from ptei where empid='" +
DropDownList3.Text.ToString() + "'", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "ptei");
int f = 0;
foreach (DataTable table in objdataset.Tables)
    {
foreach (DataRow row in table.Rows)
        {
            if
(DropDownList3.Text.ToString().ToUpper().Trim().Equals(row["empid"].ToString().ToUpp
er().Trim()))
            {
                f = 1;
                Label13.Text = row["doj"].ToString();
                Label14.Text = row["t"].ToString();
                break;
            }
            if (f == 1)
```

```
                break;
            }
            if (f == 0)
Response.Write("<marquee> sorry data not found</marquee>");
            }
        }
    protected void LinkButton2_Click1(object sender, EventArgs e)
    {
Response.Redirect("homeuser.aspx");
    }
    protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
    {
    }
}
```

1.7. adminlogin.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class adminlogin :System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();


SqlDataAdapterobjdataadapter = new SqlDataAdapter("select * from admin", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "admin");
int f = 0;
foreach (DataTable table in objdataset.Tables)
        {
foreach (DataRow row in table.Rows)
            {
                if
(TextBox6.Text.ToString().ToUpper().Trim().Equals(row["id"].ToString().ToUpper().Trim()
)
```

```
&&(TextBox7.Text.ToString().ToUpper().Trim().Equals(row["pswd"].ToString().ToUpper().
Trim())))
                {
                f = 1;
Response.Redirect("home.aspx");
                    break;
              }
           }
             if (f==1)
                break;
       }
       if (f==0)
Response.Write("<marquee> sorry admin not found</marquee>");
   }
   protected void Button2_Click(object sender, EventArgs e)
   {
      TextBox7.Text = " ";
      TextBox6.Text = " ";
   }
   protected void LinkButton2_Click1(object sender, EventArgs e)
   {
Response.Redirect("index.aspx");
   }
}
```

1.8. home.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class home :System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
   }
   protected void TreeView1_SelectedNodeChanged(object sender, EventArgs e)
   {
   }
   protected void LinkButton1_Click(object sender, EventArgs e)
   {
Response.Redirect("index.aspx");
   }
}
```

1.9. muninfo.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;


public partial class muninfo :System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
   }
   protected void Button1_Click(object sender, EventArgs e)
   {
      Session["mname1"] = TextBox1.Text.ToString();
      Session["district1"] = DropDownList1.Text.ToString();
      Session["address1"] = TextBox3.Text.ToString();
      Session["pno1"] = TextBox4.Text.ToString();
      Session["email1"] = TextBox5.Text.ToString();
      Session["fax1"] = TextBox7.Text.ToString();
      Session["chairman1"] = TextBox6.Text.ToString();
      Session["contact1"] = TextBox8.Text.ToString();
      Session["username1"] = TextBox9.Text.ToString();
      Session["password1"] = TextBox10.Text.ToString();
      Session["h1"] = DropDownList2.Text.ToString();
      Session["hint1"] = TextBox13.Text.ToString();
Response.Redirect("munirec.aspx");
   }
   protected void Button2_Click(object sender, EventArgs e)
   {
      TextBox1.Text = " ";
      TextBox3.Text = " ";
      TextBox4.Text = " ";
      TextBox5.Text = " ";
      TextBox6.Text = " ";
      TextBox7.Text = " ";
      TextBox8.Text = " ";
      TextBox9.Text = " ";
      TextBox10.Text = " ";
      TextBox11.Text = " ";
      TextBox13.Text = " ";
   }
   protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
   {
```

```
    }
    protected void LinkButton2_Click1(object sender, EventArgs e)
    {
Response.Redirect("home.aspx");
    }
}
```

1.10. muniedit.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class muniedit :System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlDataAdapterobjdataadapter = new SqlDataAdapter("select* from muninfo", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "muninfo");
foreach (DataTable table in objdataset.Tables)
            {
foreach (DataRow row in table.Rows)
                {
DropDownList2.Items.Add(row["munid"].ToString());
                }
            }
        }
    }
    protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
    {
        {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlDataAdapterobjdataadapter = new SqlDataAdapter("select* from muninfo", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "muninfo");
int f = 0;
```

```
foreach (DataTable table in objdataset.Tables)
        {
foreach (DataRow row in table.Rows)
            {
               if
(DropDownList2.Text.ToString().ToUpper().Trim().Equals(row["munid"].ToString().ToUpp
er().Trim()))
               {
                  f = 1;
                  TextBox1.Text = row["Mname"].ToString();
                  TextBox9.Text = row["district"].ToString();
                  TextBox3.Text = row["address"].ToString();
                  TextBox4.Text = row["phone"].ToString();
                  TextBox5.Text = row["email"].ToString();
                  TextBox7.Text = row["fax"].ToString();
                  TextBox6.Text = row["chairman"].ToString();
                  TextBox8.Text = row["contact"].ToString();
                  break;
               }
               if (f == 1)
                  break;
            }
         }
      }
   }
   protected void Button1_Click(object sender, EventArgs e)
   {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlCommand s = new SqlCommand();
s.Connection = o;
s.CommandText = "update muninfo set mname='" + TextBox1.Text.ToString() + "',
district='" + TextBox9.Text.ToString() + "', address='" + TextBox3.Text.ToString() + "',
phone='" + TextBox4.Text.ToString() + "', email='" + TextBox5.Text.ToString() + "', fax='"
+ TextBox7.Text.ToString() + "', chairman='" + TextBox6.Text.ToString() + "', contact='" +
TextBox8.Text.ToString() + "' where munid='" + DropDownList2.Text.ToString() + "'";
s.ExecuteNonQuery();
   }
   protected void Button2_Click(object sender, EventArgs e)
   {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlCommand s = new SqlCommand();
s.Connection = o;
s.CommandText = "delete from muninfo where munid='" + DropDownList2.Text.ToString()
+ "'";
s.ExecuteNonQuery();
```

```
    }
    protected void LinkButton2_Click1(object sender, EventArgs e)
    {
Response.Redirect("home.aspx");
    }
}
```

1.11. mundep.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class mundep :System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
          }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Session["depame1"] = TextBox1.Text.ToString();
        Session["head1"] = TextBox2.Text.ToString();
        Session["desig1"] = TextBox3.Text.ToString();
        Session["dor1"] = TextBox4.Text.ToString();
        Session["regno1"] = TextBox5.Text.ToString();
        Session["address1"] = TextBox6.Text.ToString();
        Session["email1"] = TextBox7.Text.ToString();
        Session["fax1"] = TextBox8.Text.ToString();
        Session["phone1"] = TextBox9.Text.ToString();
        Session["username1"] = TextBox10.Text.ToString();
        Session["password1"] = TextBox11.Text.ToString();
        Session["hq1"] = DropDownList1.Text.ToString();
        Session["ha1"] = TextBox14.Text.ToString();
Response.Redirect("mundeprec.aspx");
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        TextBox1.Text = " ";
        TextBox2.Text = " ";
        TextBox3.Text = " ";
        TextBox4.Text = " ";
        TextBox5.Text = " ";
        TextBox6.Text = " ";
        TextBox7.Text = " ";
```

```
        TextBox8.Text = " ";
        TextBox9.Text = " ";
        TextBox10.Text = " ";
        TextBox11.Text = " ";
        TextBox12.Text = " ";
        TextBox14.Text = " ";
    }
    protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
    {
    }
    protected void LinkButton2_Click1(object sender, EventArgs e)
    {
Response.Redirect("home.aspx");
    }
}
```

1.12. mundepedt.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class mundepedt :System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();

SqlDataAdapterobjdataadapter = new SqlDataAdapter("select* from mundep", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "mundep");

foreach (DataTable table in objdataset.Tables)
          {
foreach (DataRow row in table.Rows)
              {
DropDownList2.Items.Add(row["depid"].ToString());
              }
          }
        }
```

```
    }
    protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
    {
        {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlDataAdapterobjdataadapter = new SqlDataAdapter("select* from mundep", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "mundep");
int f = 0;
foreach (DataTable table in objdataset.Tables)
          {
foreach (DataRow row in table.Rows)
              {
                 if
(DropDownList2.Text.ToString().ToUpper().Trim().Equals(row["depid"].ToString().ToUppe
r().Trim()))
                 {
                    f = 1;
                    TextBox1.Text = row["depame"].ToString();
                    TextBox2.Text = row["head"].ToString();
                    TextBox3.Text = row["desig"].ToString();
                    TextBox4.Text = row["dor"].ToString();
                    TextBox5.Text = row["regno"].ToString();
                    TextBox6.Text = row["address"].ToString();
                    TextBox7.Text = row["email"].ToString();
                    TextBox8.Text = row["fax"].ToString();
                    TextBox9.Text = row["phone"].ToString();
                    break;
                 }
                 if (f == 1)
                    break;
              }
          }
        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlCommand s = new SqlCommand();
s.Connection = o;
s.CommandText = "update mundep set depame='" + TextBox1.Text.ToString() + "', head='"
+ TextBox2.Text.ToString() + "', desig='" + TextBox3.Text.ToString() + "',   dor='" +
TextBox4.Text.ToString() + "', regno='" + TextBox5.Text.ToString() + "', address='" +
TextBox6.Text.ToString() + "', email='" + TextBox7.Text.ToString() + "', fax='" +
```

```
TextBox8.Text.ToString() + "', phone='" + TextBox9.Text.ToString() + "' where depid='" +
DropDownList2.Text.ToString() + "'";
s.ExecuteNonQuery();
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlCommand s = new SqlCommand();
s.Connection = o;
s.CommandText = "delete from mundep where depid='" + DropDownList2.Text.ToString() +
"'";
s.ExecuteNonQuery();
    }
    protected void LinkButton2_Click1(object sender, EventArgs e)
    {
Response.Redirect("home.aspx");
    }
}
```

1.13. pteinfo.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class pteinfo :System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Session["employee1"] = TextBox1.Text.ToString();
        Session["depart1"] = DropDownList1.Text.ToString();
        Session["sop1"] = TextBox2.Text.ToString();
        Session["bp1"] = TextBox4.Text.ToString();
        Session["da1"] = TextBox5.Text.ToString();
        Session["d1"] = TextBox6.Text.ToString();
        Session["doj1"] = TextBox7.Text.ToString();
```

```
        Session["deb1"] = TextBox8.Text.ToString();
        Session["designt1"] = TextBox9.Text.ToString();
        Session["ed1"] = TextBox10.Text.ToString();
Response.Redirect("pteview.aspx");
    }
    protected void Button3_Click(object sender, EventArgs e)
    {
      double d= double.Parse(TextBox4.Text.ToString()) +
double.Parse(TextBox4.Text.ToString()) * double.Parse(TextBox5.Text.ToString()) /100;
        TextBox6.Text = d.ToString();
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        TextBox1.Text = " ";
        TextBox2.Text = " ";
        TextBox4.Text = " ";
        TextBox5.Text = " ";
        TextBox6.Text = " ";
        TextBox7.Text = " ";
        TextBox8.Text = " ";
        TextBox9.Text = " ";
        TextBox10.Text = " ";
    }
        protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
    {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlDataAdapterobjdataadapter = new SqlDataAdapter("select * from mundep where
depame='" + DropDownList1.Text.ToString() + "'", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "mundep");
foreach (DataTable table in objdataset.Tables)
        {
foreach (DataRow row in table.Rows)
            {
DropDownList1.Items.Add(row["depame"].ToString());
            }
        }
    }
    protected void LinkButton2_Click1(object sender, EventArgs e)
    {
Response.Redirect("home.aspx");
    }
}
```

1.14. ptedit.aspx

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class ptedit :System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      if (!IsPostBack)
      {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlDataAdapterobjdataadapter = new SqlDataAdapter("select* from ptei", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "ptei");
foreach (DataTable table in objdataset.Tables)
         {
foreach (DataRow row in table.Rows)
            {
DropDownList2.Items.Add(row["empid"].ToString());
            }
         }
      }
   }
   protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
   {
      {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlDataAdapterobjdataadapter = new SqlDataAdapter("select* from ptei", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "ptei");
int f = 0;
foreach (DataTable table in objdataset.Tables)
         {
foreach (DataRow row in table.Rows)
            {
               if
(DropDownList2.Text.ToString().ToUpper().Trim().Equals(row["empid"].ToString().ToUpp
er().Trim()))
                  {
                     f = 1;
                     TextBox1.Text = row["empname"].ToString();
```

```
                    TextBox2.Text = row["sop"].ToString();
                    DropDownList1.Text = row["depart"].ToString();
                    TextBox4.Text = row["bp"].ToString();
                    TextBox5.Text = row["da"].ToString();
                    TextBox7.Text = row["doj"].ToString();
                    TextBox8.Text = row["deb"].ToString();
                    TextBox9.Text = row["designt"].ToString();
                    TextBox10.Text = row["ed"].ToString();
                    break;
                }
                if (f == 1)
                    break;
            }
        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlCommand s = new SqlCommand();
s.Connection = o;
s.CommandText = "update ptei set empname='" + TextBox1.Text.ToString() + "', sop='" +
TextBox2.Text.ToString() + "', depart='" + DropDownList1.Text.ToString() + "',   bp='" +
TextBox4.Text.ToString() + "', da='" + TextBox5.Text.ToString() + "',d='" +
TextBox6.Text.ToString() + "', doj='" + TextBox7.Text.ToString() + "', deb='" +
TextBox8.Text.ToString() + "', designt='" + TextBox9.Text.ToString() + "', ed='" +
TextBox10.Text.ToString() + "' where empid='" + DropDownList2.Text.ToString() + "'";
s.ExecuteNonQuery();
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlCommand s = new SqlCommand();
s.Connection = o;
s.CommandText = "delete from ptei where empid='" + DropDownList2.Text.ToString() + "'";
s.ExecuteNonQuery();
    }
    protected void Button3_Click(object sender, EventArgs e)
    {
        double d = double.Parse(TextBox4.Text.ToString()) +
double.Parse(TextBox4.Text.ToString()) * double.Parse(TextBox5.Text.ToString()) / 100;
        TextBox6.Text = d.ToString();
    }
    protected void LinkButton2_Click1(object sender, EventArgs e)
    {
```

```
Response.Redirect("home.aspx");
    }
}
```

1.15. taxinfo.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class taxinfo :System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Session["fm1"] = TextBox1.Text.ToString();
        Session["t1"] = TextBox2.Text.ToString();
        Session["hyt1"] = TextBox3.Text.ToString();
        Session["deffect1"] = TextBox4.Text.ToString();

Response.Redirect("taxview.aspx");
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
    }
    protected void LinkButton2_Click1(object sender, EventArgs e)
    {
Response.Redirect("home.aspx");
    }
}
```

1.16. taxedit.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class taxedit :System.Web.UI.Page
```

```
{
   protected void Page_Load(object sender, EventArgs e)
   {
      if (!IsPostBack)
      {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlDataAdapterobjdataadapter = new SqlDataAdapter("select* from tax", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "tax");
         GridView1.DataSource = objdataset;
         GridView1.DataMember = "tax";
         GridView1.DataBind();
foreach (DataTable table in objdataset.Tables)
         {
foreach (DataRow row in table.Rows)
            {
DropDownList1.Items.Add(row["deffect"].ToString());
            }
         }
      }
   }
   protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
   {
      {
      }
   }
   protected void Button1_Click(object sender, EventArgs e)
   {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlCommand s = new SqlCommand();
s.Connection = o;
s.CommandText = "update tax set fm='" + TextBox5.Text.ToString() + "', t='" +
TextBox6.Text.ToString() + "',hyt='" + TextBox7.Text.ToString() + "' where deffect='" +
DropDownList1.Text.ToString() + "'";
s.ExecuteNonQuery();
   }
   protected void TextBox4_TextChanged(object sender, EventArgs e)
   {
   }
   protected void Button2_Click(object sender, EventArgs e)
   {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlCommand s = new SqlCommand();
```

```
s.Connection = o;
s.CommandText = "delete from tax where deffect='" + DropDownList1.Text.ToString() + "'";
s.ExecuteNonQuery();
   }
   protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
   {
   }
   protected void LinkButton2_Click1(object sender, EventArgs e)
   {
Response.Redirect("home.aspx");
   }
}
```

1.17. taxinti.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;
using System.Data;
using System.Data.SqlClient;
public partial class taxinti :System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      if (!IsPostBack)
      {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();

SqlDataAdapterobjdataadapter = new SqlDataAdapter("select* from mundep", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "mundep");
foreach (DataTable table in objdataset.Tables)
         {
foreach (DataRow row in table.Rows)
            {
DropDownList1.Items.Add(row["depame"].ToString());
DropDownList2.Items.Add(row["regno"].ToString());
            }
         }
      }
   }
   protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
   {
```

```
    }
    protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
    {
SqlConnection o = new SqlConnection("Data Source=DARK\\SQLEXPRESS; Initial
Catalog = munsip; Integrated Security=true");
o.Open();
SqlDataAdapterobjdataadapter = new SqlDataAdapter("select * from mundep where
depame='" + DropDownList1.Text.ToString() + "'", o);
DataSetobjdataset = new DataSet();
objdataadapter.Fill(objdataset, "mundep");
        GridView1.DataSource = objdataset;
        GridView1.DataMember = "mundep";
        GridView1.DataBind();
    }
    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {
    }
    protected void LinkButton2_Click1(object sender, EventArgs e)
    {
Response.Redirect("home.aspx");
    }
}
```

CONCLUSION

7. CONCLUSION

This website is developed in an interactive and user friendly manner. It will meet most of necessities required. Modules are designed to be flexible so that any failure requirements can be easily added to the modules without facing many problems. There is enough security is provided here. The municipality Administration System is a complete online application (right from the membership admission to its modification, information storage, downloading certificates is completely carried out online over internet. The concept of Online Municipality Administration System comes along with lot of advantages.

Municipality Administration System is a complete Workflow based system wherein specific functionalities/privileges are assigned to departmental officials based upon the roles performed by them. We tried to make the project to a success within the prescribed time with the guidance of our teachers and friends.

BIBLIOGRAPHY

8. BIBLIOGRAPHY

- System Analysis & Design, - www.tutorialsspot.com
- Database Design, Overview, Importance, and techniqueswww.aestra.com
- System Study and analysis- www.wikipedia.org
- ASP.NET 3.5 Unleashed, by Stephen Walther.