Contents

1 INTRODUCTION

1.1 Project Profile

The Collectorate management is a solution for public to get information and to receive services from Govt. offices. The normal applications or information centre at the Collectorate will give just an idea about the location and service that available by the offices. But after getting the information the public will have to go around the campus to reach the correct place. This is an online application that can be used from anywhere. The website provides details about where the exact location is, do the officer present or not, what are the services available by the office etc. So that the user can decide whether to visit the office on right day or to change the visiting date before they go. The public peoples and end users are provided with the online service for applying forms such as community, income, death and pattayam certificates and can also participate in tender offers announced by the collectorate. Online petition filing system is available for every end user even for peoples who havent registered with the system. Online petition filing is also available in registered users interface and they can check the status of their petition whether action taken for his petition or it is rejected due to some reasons. Also a map will be available at the website displaying the office locations. The main users in this system are:

1. Admin Module.
2. Office Module
3. Staff Module.
4. User Module

2 ABOUT THE DEVELOPING TOOLS

2.1 Introduction to JSP

JSP technology is used to create dynamic web applications. JSP pages are easier to maintain then a Servlet. JSP pages are opposite of Servlets as a servlet adds HTML code inside Java code, while JSP adds Java code inside HTML using JSP tags. Everything a Servlet can do, a JSP page can also do it. JSP enables us to write HTML pages containing tags, inside which we can include powerful Java programs. Using JSP, one can easily separate Presentation and Business logic as a web designer can design and update JSP pages creating the presentation layer and java developer can write server side complex computational code without concerning the web design. And both the layers can easily interact over HTTP requests.
Advantage of JSP over Servlet

1. Extension to Servlet
   JSP technology is the extension to servlet technology. We can use all the features of servlet in JSP.

2. Easy to maintain
   JSP can be easily managed because we can easily separate our business logic with presentation logic

3. Fast Development
   If JSP page is modified, we don't need to recompile and redeploy the project.

4. Less code than Servlet
   In JSP, we can use a lot of tags such as action tags, jstl, custom tags etc. that reduces the code.

2.2 HeidiSQL

HeidiSQL is a useful and reliable tool designed for web developers using the popular MySQL server, Microsoft SQL databases and PostgreSQL. It enables you to browse and edit data, create and edit tables, views, procedures, triggers and scheduled events. Also, you can export structure and data either to SQL file, clipboard or to other servers. Its codebase was originally taken from Ansgar Becker's own MySQL-Front 2.5 software. Due to having sold the MySQL-Front branding to an unrelated party, Becker chose ”HeidiSQL” as a replacement. The name was suggested by a friend as a tribute to Heidi Klum, and was further reinforced by Becker's own nostalgia for Heidi, Girl of the Alps. HeidiSQL is a useful and reliable tool designed for web developers using the popular MySQL server, Microsoft SQL databases and PostgreSQL. It enables you to browse and edit data, create and edit tables, views, procedures, triggers and scheduled events. Also, you can export structure and data either to SQL file, clipboard or to other servers. HeidiSQL is a free and open-source administration tool for MySQL and its forks, as well as Microsoft SQL Server and PostgreSQL. Its codebase was originally taken from Ansgar Becker's own MySQL-Front 2.5 software. Due to having sold the MySQL-Front branding to an unrelated party, Becker chose ”HeidiSQL” as a replacement. The name was suggested by a friend as a tribute to Heidi Klum, and was further reinforced by Becker's own nostalgia for Heidi, Girl of the Alps.A version written in Java, jHeidi, was designed to work on Mac and Linux computers. It was discontinued in March 2010 in favor of Wine support.

2.3 Netbeans

NetBeans IDE is a free, open source, integrated development environment (IDE) that enables you to develop desktop, mobile and web applications. The IDE supports application development in various languages, including Java, HTML5, PHP and C++. The IDE provides integrated support for the complete development cycle, from project creation through debugging, profiling and deployment. The IDE runs on Windows, Linux, Mac OS X, and other UNIX-based system.The IDE provides comprehensive support for JDK 7 technologies and the most recent Java enhancements. It is the first IDE that provides support for JDK 7, Java EE 7, and JavaFX 2. The IDE fully supports Java EE using the latest standards for Java, XML, Web services, and SQL and fully supports the GlassFish Server, the reference implementation of Java EE.

2.4 GitHub

Version control is a system that manages changes to a file or files.These changes are kept as logs in a history, with detailed information on what file(s) was changed, what was changed within the file, who changed it, and a message on why the change was made. This is extremely useful, especially when working in teams.To understand how incredibly powerful version control is How many files of different versions of a manuscript or thesis do you have laying around after getting feedback from your supervisor or co-authors? Have you ever wanted to experiment with your code or your manuscript and need to make a new file so that the original is not touched ?Have you ever deleted something and wish you hadnt ? Have you ever forgotten what you were doing on a project ? All these problems are fixed by using version control (git)!

HOW TO ENTER THE PROJECT VERSION TO GIT

1.Create a new repository on GitHub. To avoid errors, do not initialize the new repository with README, license, or gitignore files. You can add these files after your project has been pushed to GitHub.

2.Open Git Bash

3.Change the current working directory to your local project.

4.Initialize the local directory as a Git repository

```
$ git init
```

5.Add the files in your new local repository. This stages them for the first commit.

```
$ git add
```

6. Commit the files that you've staged in your local repository

```
$ git commit -m "First commit"
```

7.At the top of your GitHub repository's Quick Setup page, click to copy the remote repository URL.

8.In the Command prompt, add the URL for the remote repository where your local repository will be pushed

```
$git remote add origin remote repository URL
$git remote -v
```

9.Push the changes in your local repository to GitHub

```
$git push origin master
```

2.5 Apache Tomcat

Apache Tomcat, often referred to as Tomcat Server, is an open-source Java Servlet Container developed by the Apache Software Foundation (ASF).Tomcat implements several Java EE specifications including Java Servlet, JavaServer Pages (JSP), Java EL, and WebSocket, and provides a "pure Java" HTTP web server environment in which Java code can run.Tomcat is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation, released under the Apache License 2.0 license, and is open-source software.

Installing Tomcat with the Windows Installer

Using the Windows EXE distribution is by far the easiest way to get Tomcat up and running on your Windows machine. As this is a quick-start guide, it is recommended that you use this option.Simply download the "Windows Service Installer" package from the Apache website and run the .exe file, which will be called something like "apache-tomcat-6.x.x.exe". In addition to moving the binaries into place, the installer will prompt you to configure a few critical variables, such as the location of your JVM, your default HTTP connector port, and your administrator login, that you will otherwise have to configure yourself.If you don't want to use the installer you can visit our easy guide to Tomcat Configuration for more information about manual installation and configuration. After finishing your installation, you will be given the option to run Tomcat on exit. Select this option, and click Finish to launch the Tomcat server. If your installation was successful, you should be able to use your web browser to access the Tomcat welcome page at http:/localhost:8080/, and you should see a new Apache Tomcat Manager icon on your Toolbar displaying a small green arrow, indicating that your server is running.The Manager application (not to be confused with Tomcat's Manager web application) is a Windows-specific GUI tool for interacting with the Tomcat server. You can double-click it to start or shutdown the server, or set a variety of server options.

2.6 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web.[3] Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as ¡img /¿ and ¡input /¿ introduce content into the page directly. Others such as ¡p¿...¡/p¿ surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.[4] In 1980, physicist Tim Berners-Lee, a contractor at CERN, proposed and prototyped ENQUIRE, a system for CERN researchers to use and share documents. In 1989, Berners-Lee wrote a memo proposing an Internet-based hypertext system.[5] Berners-Lee specified HTML and wrote the browser and server software in late 1990. That year, Berners-Lee and CERN data systems engineer Robert Cailliau collaborated on a joint request for funding, but the project was not formally adopted by CERN. In his personal notes[6] from 1990 he listed[7] "some of the many areas in which hypertext is used" and put an encyclopedia first.

The first publicly available description of HTML was a document called "HTML Tags", first mentioned on the Internet by Tim Berners-Lee in late 1991.[8][9] It describes 18 elements comprising the initial, relatively simple design of HTML. Except for the hyperlink tag, these were strongly influenced by SGMLguid, an in-house Standard Generalized Markup Language (SGML)-based documentation format at CERN. Eleven of these elements still exist in HTML 4.

HTML is a markup language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of CSS. Many of the text elements are found in the 1988 ISO technical report TR 9537 Techniques for using SGML, which in turn covers the features of early text formatting languages such as that used by the RUNOFF command developed in the early 1960s for the CTSS (Compatible Time-Sharing System) operating system: these formatting commands were derived from the commands used by typesetters to manually format documents. However, the SGML concept of generalized markup is based on elements (nested annotated ranges with attributes) rather than merely print effects, with also the separation of structure and markup; HTML has been progressively moved in this direction with CSS.

2.7 Java

Java is a set of computer software and specifications developed by Sun Microsystems, which was later acquired by the Oracle Corporation, that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones to enterprise servers and supercomputers. Java applets, which are less common than standalone Java applications, run in secure, sandboxed environments to provide many features of native applications and can be embedded in HTML pages.

Writing in the Java programming language is the primary way to produce code that will be deployed as byte code in a Java virtual machine (JVM); byte code compilers are also available for other languages, including Ada, JavaScript, Python, and Ruby. In addition, several languages have been designed to run natively on the JVM, including Scala, Clojure and Apache Groovy. Java syntax borrows heavily from C and C++, but object-oriented features are modeled after Smalltalk and Objective-C.[10] Java eschews certain low-level constructs such as pointers and has a very simple memory model where every object is allocated on the heap and all variables of object types are references. Memory management is handled through integrated automatic garbage collection performed by the JVM.

On November 13, 2006, Sun Microsystems made the bulk of its implementation of Java available under the GNU General Public License (GPL). The latest version is Java 9, the second of the two supported (with e.g. security updates) versions as of 2017. Oracle (and others) has announced that using older versions (other than Java 8) of their JVM implementation presents serious risks, due to unresolved security issues.

The heart of the Java platform is the concept of a "virtual machine" that executes Java bytecode programs. This bytecode is the same no matter what hardware or operating system the program is running under. There is a JIT (Just In Time) compiler within the Java Virtual Machine, or JVM. The JIT compiler translates the Java bytecode into native processor instructions at run-time and caches the native code in memory during execution.

The use of bytecode as an intermediate language permits Java programs to run on any platform that has a virtual machine available. The use of a JIT compiler means that Java applications, after a short delay during loading and once they have "warmed up" by being all or mostly JIT-compiled, tend to run about as fast as native programs.[16][17][18] Since JRE version 1.2, Sun's JVM implementation has included a just-in-time compiler instead of an interpreter. Although Java programs are cross-platform or platform independent, the code of the Java Virtual Machines (JVM) that execute these programs is not. Every supported operating platform has its own JVM.

2.8 Java script

JavaScript , often abbreviated as JS, is a high-level, dynamic, weakly typed, prototype-based, multi-paradigm, and interpreted programming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production. It is used to make webpages interactive and provide online programs, including video games. The majority of websites employ it, and all modern web browsers support it without the need for plug-ins by means of a built-in JavaScript engine. Each of the many JavaScript engines represent a different implementation of JavaScript, all based on the ECMAScript specification, with some engines not supporting the spec fully, and with many engines supporting additional features beyond ECMA.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as Self and Scheme.

3 SYSTEM ANALYSIS

3.1 Introduction

System Analysis works with users to identify goals and build system to achieve them. System Analysis is an important phase of any system development process. System analysis is a step-by-step process used to identify and develop or acquire the software need to control the processing of specific application. System analysis is a continuing activity the stages of the systems development. The system is studied to the minutes details and analyzed. In analysis, a detailed study of these operation performed by a system and their relationships within and outside of the system is done

The aim of the proposed system is to develop a system with improved facilities. The proposed system can overcome all the limitation of the existing system, such as students information is maintained in the database, it gives more security to data, ensures data accuracy, reduces paper work and save time, only eligible students get chance, it makes information flow efficient and paves way for easy report generation, reduce the space. proposed system is cost effective.

3.2 Existing System

Get information from information centres, but the user cant get actual direction to reach the office. No online website to get the details of services and departments at the Collectorate. User is not aware about the officer is present or absent on the day. The important and the most significant drawback is that the system is manual. There may be errors due to carelessness or oversight that may result in loss to the data and as to the organization. The forms like community certificates, birth certificates are all applied in person by the public peoples. The complaints and petitions against officials and government officers should be submitted in person in the collector office itself, it is very time consuming in this modern world.

3.3 Feasibility Study

A feasibility study is needed to determine if a project or end result of a project is feasible and beneficial. The main objective of feasibility study is to test the technical, social and economic feasibility of developing a new computer system. Investigating the existing system in the areas under investigation and generating ideas about a new system does this. The key considerations involved in the feasibility analysis are the following:

1. Economic feasibility
2. Technical feasibility
3. Operational feasibility.

3.3.1 Economic Feasibility

Economic feasibility is a method for evaluating the effectiveness of a candidate system. This study is mainly concerned with cost-benefit analysis that is how much money the user is investing in any system and how much he is getting as a benefit in output. Our project is Economical Feasible because anyone uses this software would need only to buy the machine. Our hardware requirement is not too expensive. The money and human effort needed for the existing system is high .In the new system benefits outweigh costs. So as compare to cost the project is economically feasible. We conduct an economic feasibility study for this exam seat mapping system and it also uses minimum hardware requirements that are already used in the existing system .In existing system manual records are used for storing details. The system is cost effective because of its compatibility and effort saving nature. The cost benefit ratio is very small and hence the proposed system is feasible.

3.3.2 Technical Feasibility

Technical feasibility includes whether the technology is available in the market for the development and its availability. The assessment of technical feasibility must be based on an outline design of the system requirements in terms of input, output, files, programs and procedures. This study checks the technical aspects of system. Minimum requirements of the proposed system are a computer and internet connectivity, which will not add any additional expense in implementing the system. This software is simple to use and manage.Online Freelancer system also uses the minimum technologies for the creation of the web based application. The existing system has also required minimum technical requirements. So the proposed system is said to be technically feasible

3.3.3 Operational Feasibility

The new system is very much easier and user friendly than the existing system. It satisfies the requirements identified in the requirements analysis phase of system development. It reduces the operational time considerably. Operational cost is very less. The maintenance and modification of the new system needs very less human effort. Using command buttons throughout the application programs enhances the operational feasibility. The new system is operationally feasible and makes the operations simpler and quite easier. The proposed system exam seat mapping system does not produce any problem to existing customers, organization etc. It reduces the drawback of existing system. All these reasons make the new system operationally feasible.

3.4 Proposed system

The proposed system provides, the user can see whether the officer is present or not before visit. Lists all the services available at the department. Get the map of office location. And also department details, available forms details, online petition system public schemes for welfare and contacts of higher officials. The major advantage of the proposed system is that information is available anytime, high integrity and security, ability to incorporate newly available data, user friendly, duplication of information is curbed

We all know the importance of computerization. The world is moving ahead at lightning speed and everyone is running short of time. One always wants to get the information and perform a task he/she/they desire(s) within a short period of time and too with amount of efficiency and accuracy. The application areas for the computerization have been selected on the basis of following factors:

FEATURES Work changes and several ad hoc reports.

4 FACT FINDING TECHNIQUES

The success of any project depends upon the accuracy of available data. Accurate information can be collected with the help of certain methods / techniques. These specific methods for finding information of the system are termed as fact finding techniques. Interview, Questionnaire, Record View and Observations are the different fact finding techniques used in this project.

4.1 Interview:

This method is used to collect the information from groups or individuals. We select the people who are related with the system for the interview. In this method, we sit face to face with people and record their responses.

4.2 Record View:

The information related to the system is available in the source like companys documents, websites and other records. This record review helped me to get valuable information about the system.

4.3 Onsite observation:

Unlike the other fact finding techniques, in this method we visit the organization and observe and understand the working of the existing system, flow of the system, the users of the system etc.

5 SYSTEM SPECIFICATION

5.1 Hardware Specification

The selection of hardware configuration is very important task related to software development. The processor should be powerful to handle all the operations. The hard disk should have the sufficient capacity to solve the database and the application. The hardware requirements for developing and implementing the proposed system are given below:

- Processor - Pentium III
- Speed - 1.1 Ghz
- RAM - 256 MB(min)
- Hard Disk - 20 GB
- Floppy Drive - 1.44 MB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse

5.2 Software Specification

Windows XP server includes improved network, application, and Web services. It provides improved reliability and scalability, lowers yours cost of computing with powerful, flexible management services, and provides the best foundation for running business applications. It provides network data security by protecting data on the wire or at the network interface. It also provides stored data on the security by using data encryption. Data encryption is provided transparently within windows XP by feature known as Encrypting File System (EFS). It has the ability to run on a single PC chip with a user up to a multi-user, multi-processor network installation. The software requirements for developing and implementing the proposed system are given below:

- Operating System : Windows 8.1
- Application Server : Apache Tomcat 7.0.34.0
- Front End : HTML, Java.
- Scripts : JavaScript.
- Server side Script : Java Server Pages.
- Database : HeidiSQL
- Database Connectivity : JDBC.

6 SYSTEM DESIGN

6.1 Introduction of System Design

In this project design technique used is top-down, object- oriented dynamic modeling technique. A top-down design approach starts by identifying the major components and iterating until the desired level of details is achieved. In object oriented design technique, the modules in the design represent data abstraction. A dynamic model aim to specify new the state of various objects changes as events occur

6.2 Input Design

Input design is a part of overall system design, which requires very careful attention. Input design features can ensure the reliability of the system and produce result from accurate data, or they can result in the production or erroneous information. The input design also determines whether the user can interact efficiently with the system. Admin who was a person which they can view the normal facebook activites,view streesed user.user is a normal facebook user who can login to the system,do all the normal activites like posting photos,messaging,give friend request etc.And the user can view whether any person tag their photo on the timeline.and he or she can accept or reject the request..

6.3 Output Design

One of the important features of an information system for users is the output produces. Output is the information delivered to users through the information system. Output design is very important phase because the output will be interactive manner. In order to create the most useful output possible. To make a user friendly output and for better communication the programmer can use the features of a window. admin can view the users,streessed user,user can view the friend request,mesages,post etc.user can also view the timeline of the post

6.4 Database Design

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

- Determine the data to be stored in the database.
- Determine the relationships between the different data elements.
- Superimpose a logical structure upon the data on the basis of these relationships.

In this project database design generally the data is to be stored in the database whether it can more relation for each modules. And it provides the logical relation between them.

6.5 Architectural Design

Architectural design is of crucial importance in software engineering during which the essential requirements like reliability, cost, and performance are dealt with. Architectural design is the responsibility of developers, some other people like user representatives, systems engineers, hardware engineers, and operations personnel are also involved. All these stakeholders must also be consulted while reviewing the architectural design in order to minimize the risks and errors.

6.6 System Modules

1. Admin module

    - Offices Entry: let the admin to add offices at the campus
    - View and verify attendance: admin can view presents of officers at each department and offices.
    - Map preparation: set the map of campus with route to each office from entry point.
    - Forms and petitions: admin perform relevant actions for applied forms and petitions.

2. user module

    - Route view
    - Staff presents verification
    - Check available services
    - Apply for certificates and petition filling

3. Office Admin

    - Services entry: enters the services provided by the office.
    - Department Entry: add and update the departments in the office.
    - Staff management: add staff and set attendance for each staff.

4. Staff

    - Attendance: logging in to the system means the staff is present and whenever the staff logging out the present marker will be updated.

6.7 Form Design

A form designing means deciding the contents and layout of forms for the purpose of collecting and processing the required information economically and efficiently. The importance of forms designing can be understood because of the following points:

1. Forms are used to collect record and communicate the required information according to the expectations of the needy persons. Therefore, forms are treated as tools of office work. If the forms are badly designed, it reduces the speed of operation of office work.

2. The forms create psychological impact on the people who use it. The people may be frustrated and get tired if the forms are not designed properly.

3. The badly designed forms results in more number of mistakes in clerical work. Hence, there is a need of well-designed forms to avoid mistakes in clerical work.

4. Sometimes, the designed form may project a poor image in the minds of the customers. This may adversely affect the good will of the company.

5. System is the basis for form design. Hence, forms are designed according to the needs of the system. If forms are badly designed, they can ruin a whole system.

6. The well-designed forms contribute much to the efficiency of employees of an organization and efficiency of the system.

7. The cost of forms is less than the cost of completing office forms, transporting and filling of office forms. The ratio will be greater if the forms are badly designed.

6.8 Table Structure

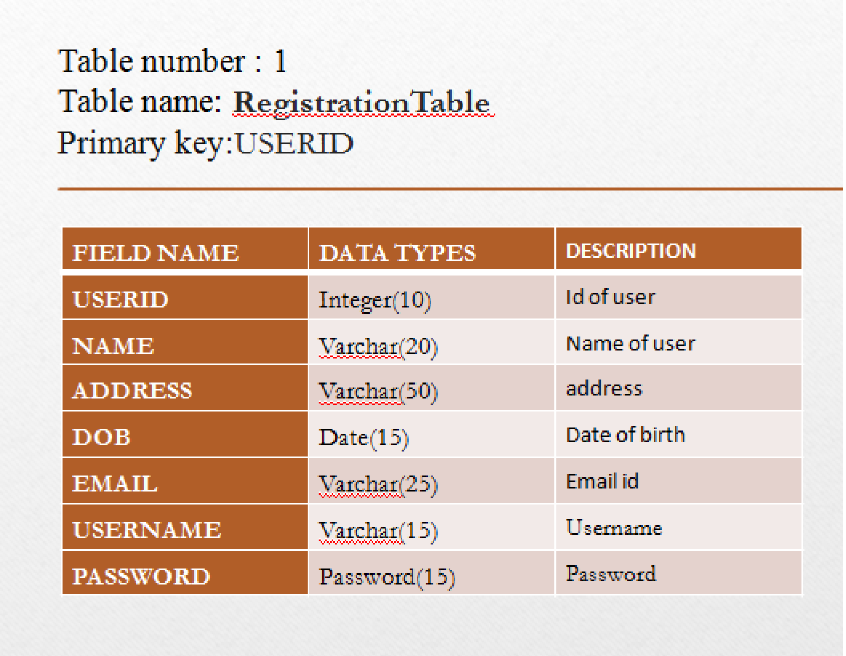


Table number : 1
Table name: RegistrationTable
Primary key:USERID

FIELD NAME	DATA TYPES	DESCRIPTION
USERID	Integer(10)	Id of user
NAME	Varchar(20)	Name of user
ADDRESS	Varchar(50)	address
DOB	Date(15)	Date of birth
EMAIL	Varchar(25)	Email id
USERNAME	Varchar(15)	Username
PASSWORD	Password(15)	Password

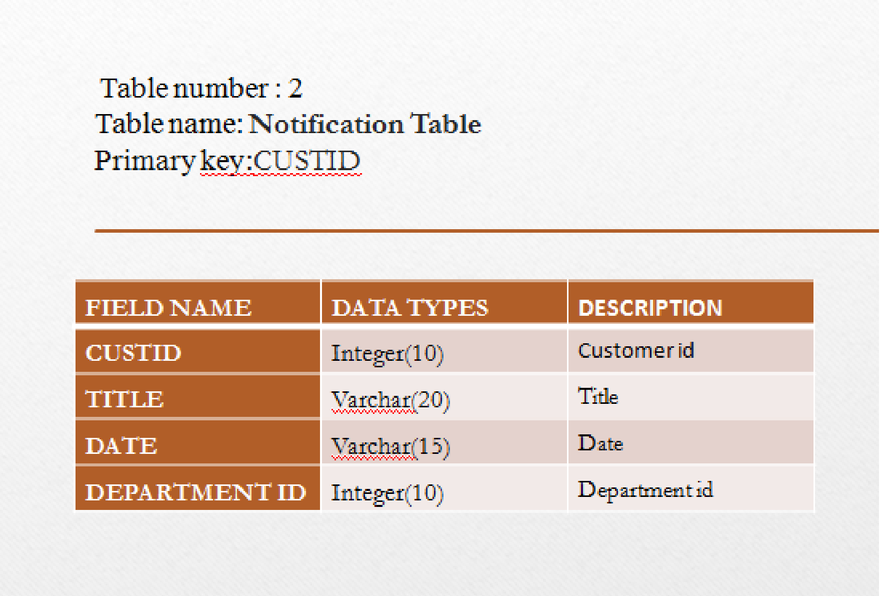


Table number : 2
Table name: Notification Table
Primary key:CUSTID

FIELD NAME	DATA TYPES	DESCRIPTION
CUSTID	Integer(10)	Customer id
TITLE	Varchar(20)	Title
DATE	Varchar(15)	Date
DEPARTMENT ID	Integer(10)	Department id

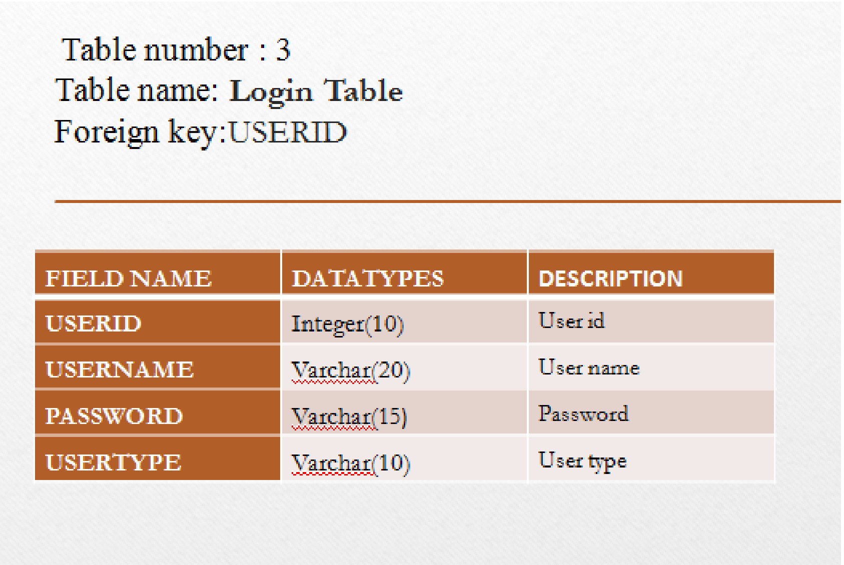


Table number : 3
Table name: **Login Table**
Foreign key:USERID

FIELD NAME	DATATYPES	DESCRIPTION
USERID	Integer(10)	User id
USERNAME	Varchar(20)	User name
PASSWORD	Varchar(15)	Password
USERTYPE	Varchar(10)	User type

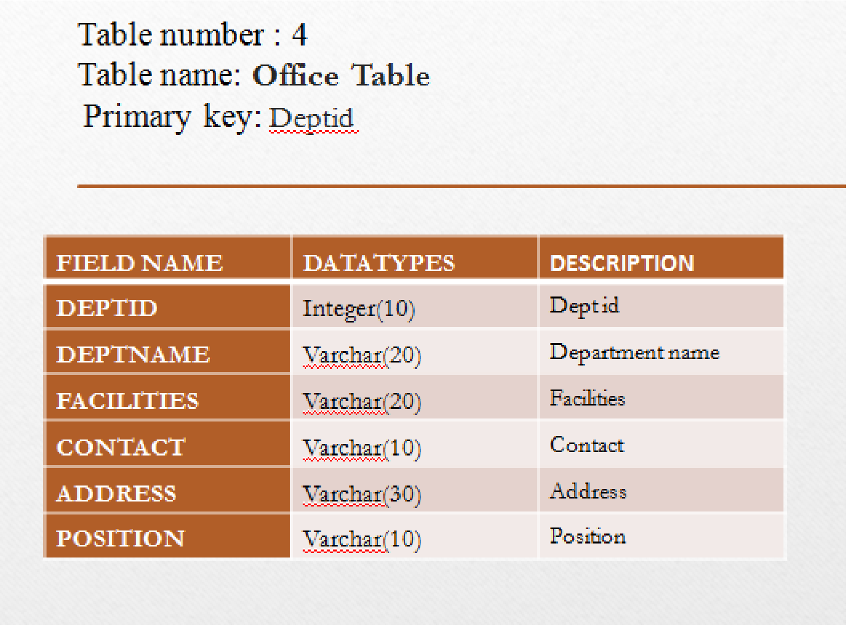


Table number : 4
Table name: **Office Table**
Primary key: Deptid

FIELD NAME	DATATYPES	DESCRIPTION
DEPTID	Integer(10)	Dept id
DEPTNAME	Varchar(20)	Department name
FACILITIES	Varchar(20)	Facilities
CONTACT	Varchar(10)	Contact
ADDRESS	Varchar(30)	Address
POSITION	Varchar(10)	Position

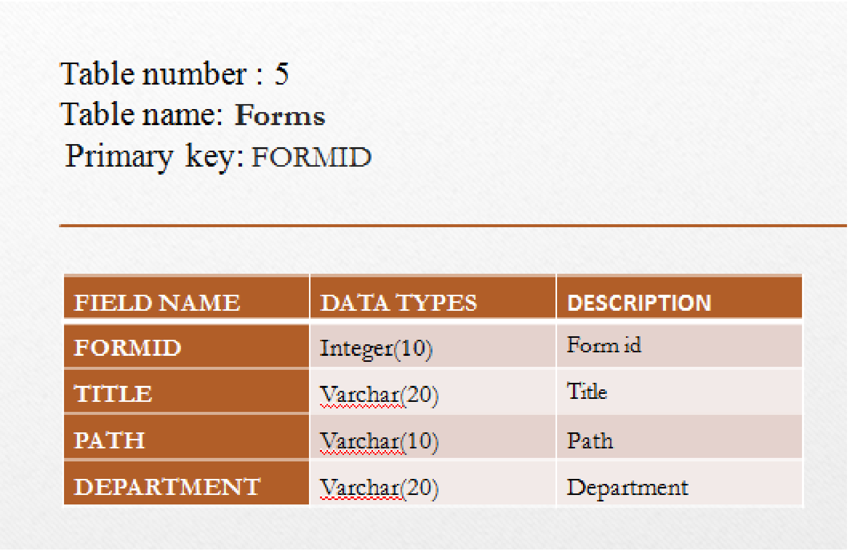


Table number : 5
Table name: **Forms**
Primary key: FORMID

FIELD NAME	DATA TYPES	DESCRIPTION
FORMID	Integer(10)	Form id
TITLE	Varchar(20)	Title
PATH	Varchar(10)	Path
DEPARTMENT	Varchar(20)	Department

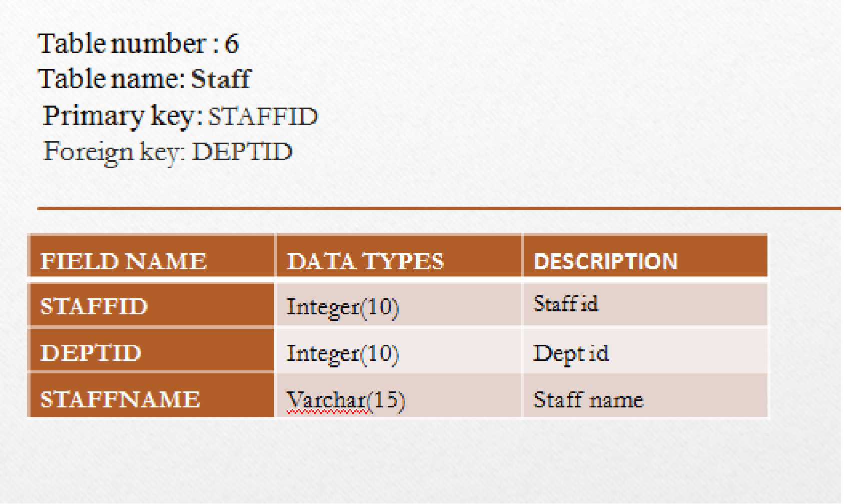


Table number : 6
Table name: **Staff**
Primary key: STAFFID
Foreign key: DEPTID

FIELD NAME	DATA TYPES	DESCRIPTION
STAFFID	Integer(10)	Staff id
DEPTID	Integer(10)	Dept id
STAFFNAME	Varchar(15)	Staff name

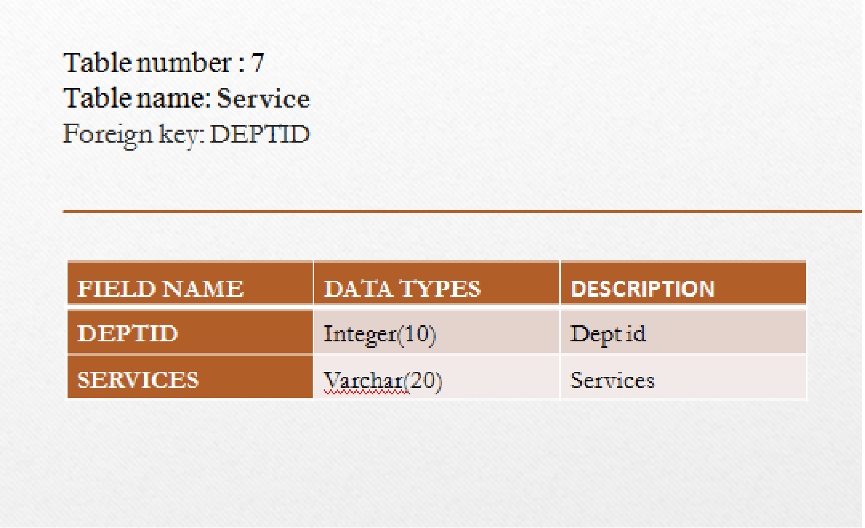


Table number : 7

Table name: **Service**

Foreign key: DEPTID

FIELD NAME	DATA TYPES	DESCRIPTION
DEPTID	Integer(10)	Dept id
SERVICES	Varchar(20)	Services

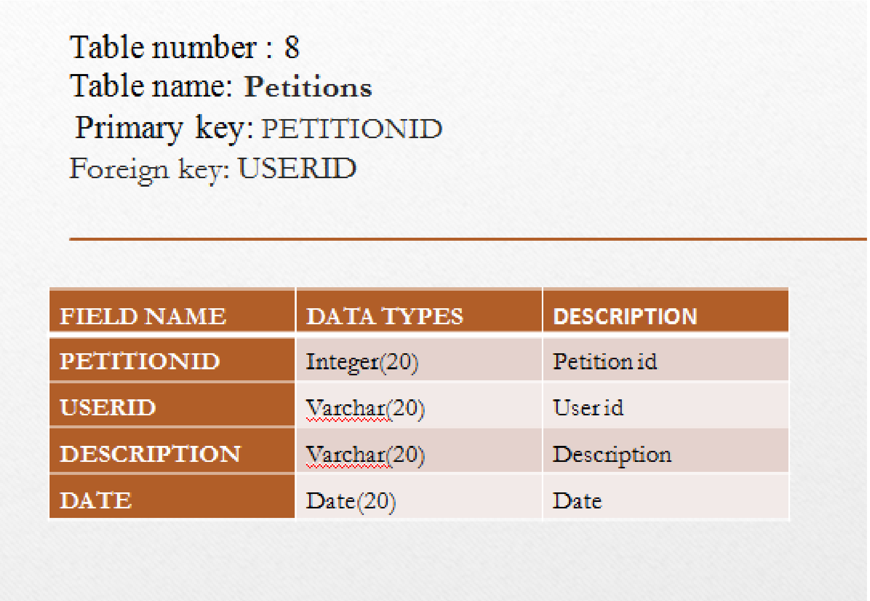


Table number : 8

Table name: **Petitions**

Primary key: PETITIONID

Foreign key: USERID

FIELD NAME	DATA TYPES	DESCRIPTION
PETITIONID	Integer(20)	Petition id
USERID	Varchar(20)	User id
DESCRIPTION	Varchar(20)	Description
DATE	Date(20)	Date

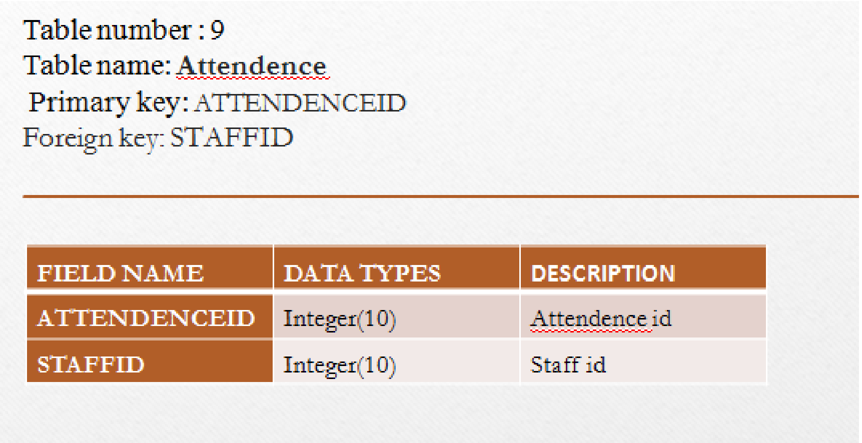


Table number : 9
Table name: Attendence
Primary key: ATTENDENCEID
Foreign key: STAFFID

FIELD NAME	DATA TYPES	DESCRIPTION
ATTENDENCEID	Integer(10)	Attendence id
STAFFID	Integer(10)	Staff id

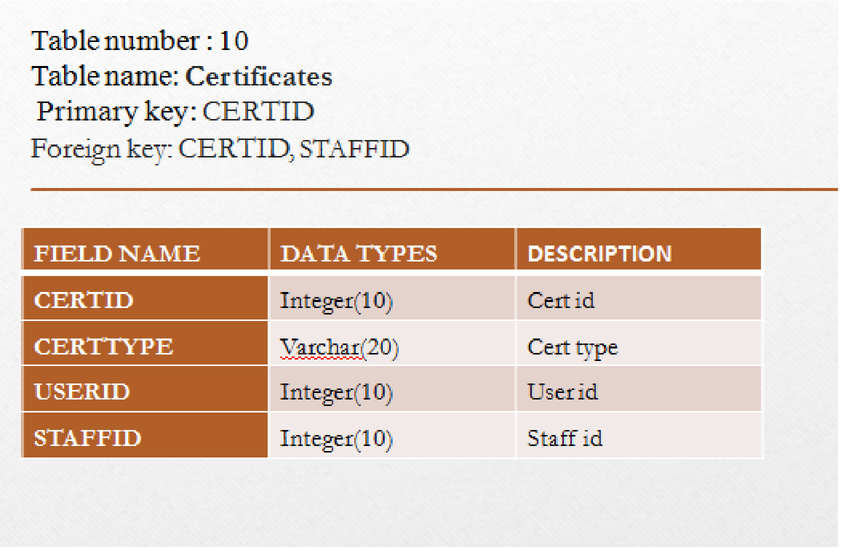
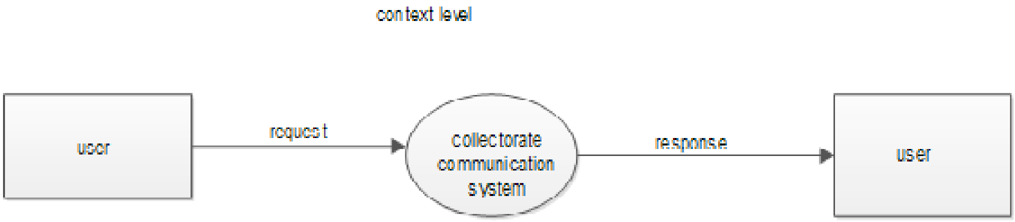


Table number : 10
Table name: Certificates
Primary key: CERTID
Foreign key: CERTID, STAFFID

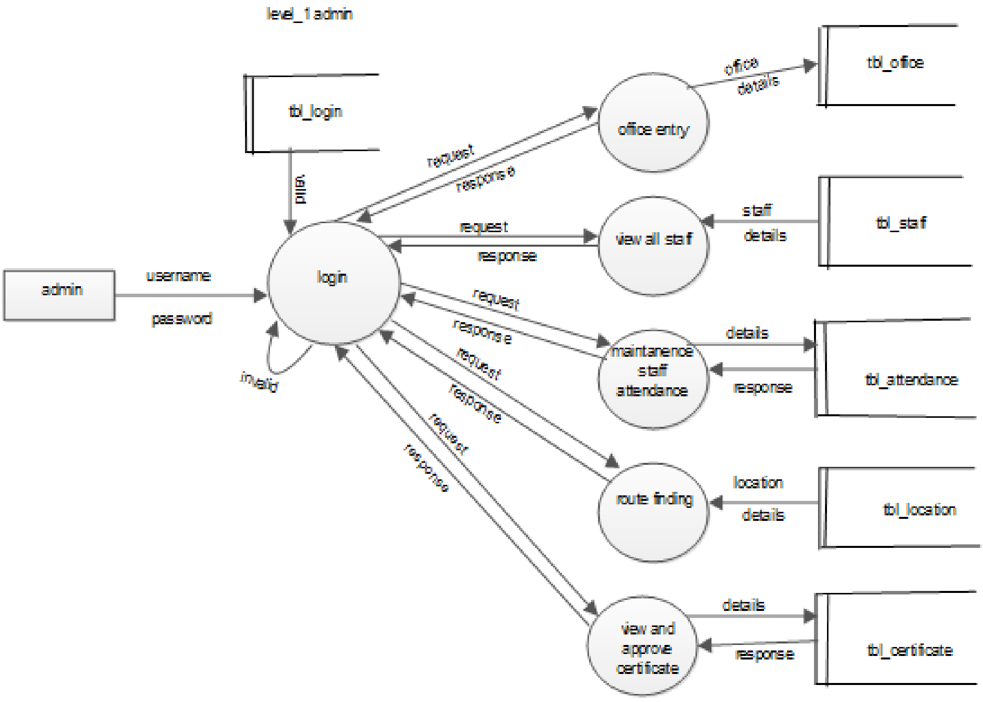
FIELD NAME	DATA TYPES	DESCRIPTION
CERTID	Integer(10)	Cert id
CERTTYPE	Varchar(20)	Cert type
USERID	Integer(10)	User id
STAFFID	Integer(10)	Staff id

6.9 DFD

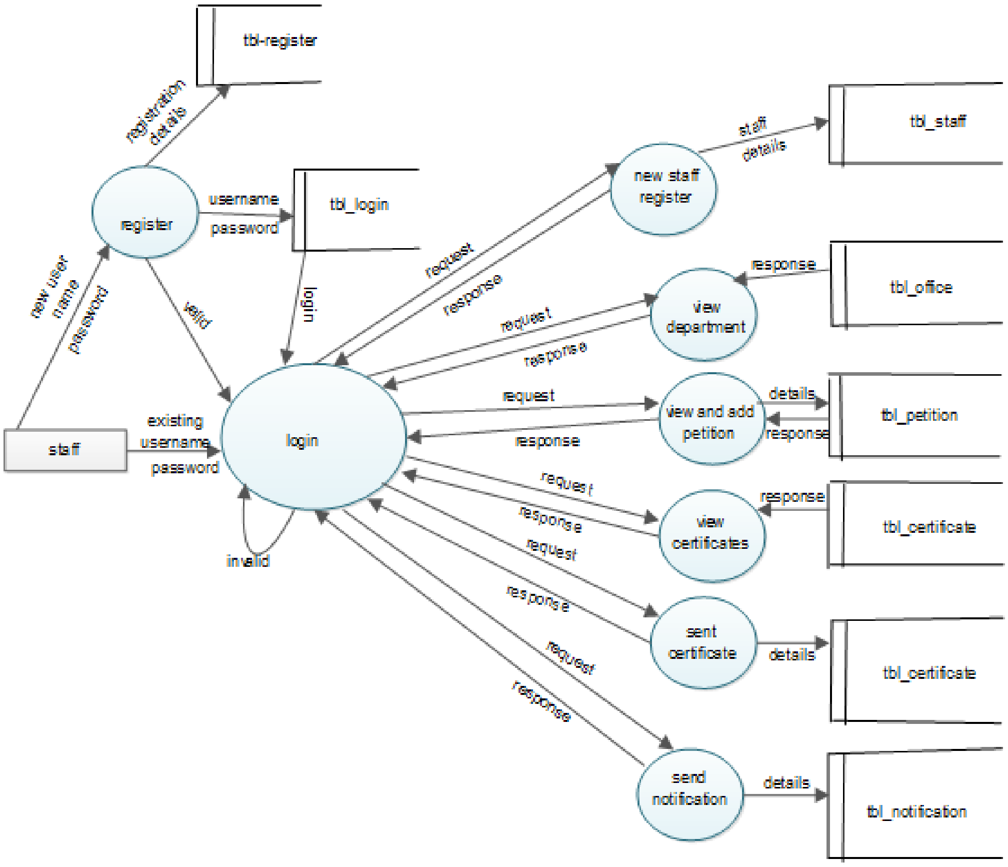
6.9.1 CONTEXT LEVEL



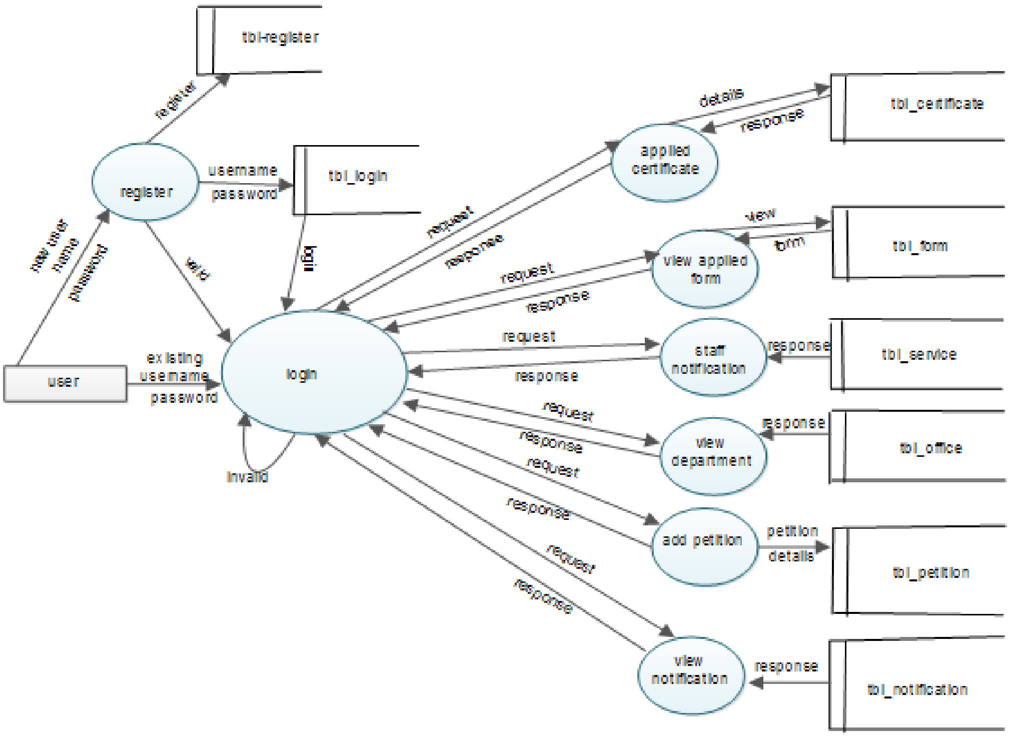
6.9.2 LEVEL-1 ADMIN



6.9.3 LEVEL-1 STAFF



6.9.4 LEVEL-1 USER



7 SYSTEM TESTING

7.1 Introduction to System Testing

Testing is the process of examining the software to compare the actual behavior with that of the excepted behavior. The major goal of software testing is to demonstrate that faults are not present. In order to achieve this goal the tester executes the program with the intent of finding errors. Though testing cannot show absence of errors but by not showing their presence it is considered that these are not present.

System testing is defined as the process by which one detects the defects in the software. Any software development organization or team has to perform several processes. Software testing is one among them. It is the final opportunity of any programmer to detect and rectify any defects that may have appeared during the software development stage. Testing is a process of testing a program with the explicit intention of finding errors that makes the program fail. In short system testing and quality assurance is a review in software products and related documentation for completion, correctness, reliability and maintainability.

System testing is the first stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct and the goal will be successfully achieved. A series of testing are performed for the proposed system before the proposed system is ready for user acceptance testing.

7.2 Unit Testing

This method of testing test the smallest unit of software called modules. It will test all the important path to find errors within the boundary of module. This has enabled the detection of errors in coding and logic. Various test cases are prepared. For each module these test cases are implemented and it is checked whether the module is executed as per the requirements and outputs the desired result. In this test each service input and output parameters are checked. In unit testing, All independent paths through the control structures are executed to ensure that all statements in the modules have been executed at least once. Error handling paths are also tested.

7.3 Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. In this testing, all individual modules were combined and module wise shifting was verified to be alright The integration testing is performed in the Myspace by combining the two modules.ie, by combining the admin, user modules and found all modules are running without any error.

7.4 Validation Testing

Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably accepted by user. Software validation is achieved through a series of tests that demonstrate conformity with requirement. Deviation or error discovered at this step in this project is corrected prior to completion of the project with the help of the user. In Myspace verifications are done correctly. So there is no chance for users to enter incorrect values. It will give error messages by using different validations. The validation testing is done very clearly and found it is error free.

7.5 Alpha Testing

Alpha testing is one of the most common software testing strategies used in software development. Its specially used by product development organizations.

1. This test takes place at the developers site. Developers observe the users and note problems.

2. Alpha testing is testing of an application when development is about to complete. Minor design changes can still be made as a result of alpha testing.

3. Alpha testing is typically performed by a group that is independent of the design team, but still within the company, e.g. in-house software test engineers, or software QA engineers.

4. Alpha testing is final testing before the software is released to the general public. It has two phases:

5. In the first phase of alpha testing, the software is tested by in-house developers. They use either debugger software, or hardware-assisted debuggers. The goal is to catch bugs quickly.

6. In the second phase of alpha testing, the software is handed over to the software QA staff, for additional testing in an environment that is similar to the intended use.

7. Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

7.6 Beta Testing

Beta Testing is also known as field testing. It takes place at users site. It sends the system/software to users who install it and use it under real-world working conditions.

- A beta test is the second phase of software testing in which a sampling of the intended audience tries the product out. (Beta is the second letter of the Greek alphabet.) Originally, the term alpha testing meant the first phase of testing in a software development process. The first phase includes unit testing, component testing, and system testing. Beta testing can be considered pre-release testing.

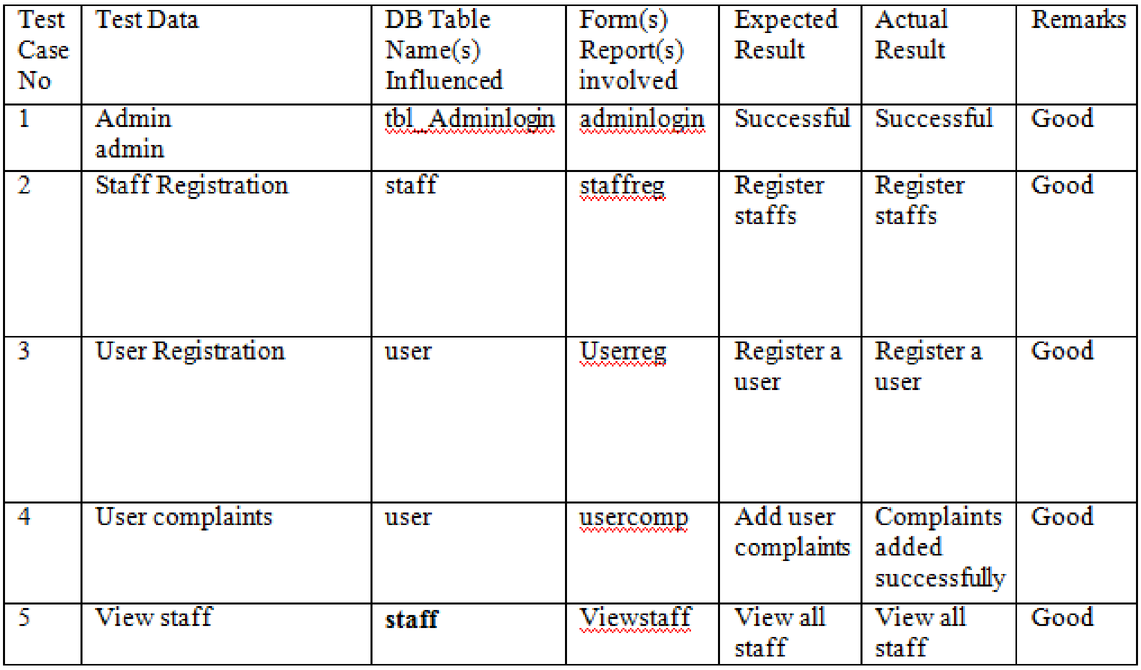
- The goal of beta testing is to place your application in the hands of real users outside of your own engineering team to discover any flaws or issues from the users perspective that you would not want to have in your final, released version of the application. Example: Microsoft and many other organizations release beta versions of their products to be tested by users.

7.7 Bugzilla

Bugzilla is an open-source issue/bug tracking system that allows developers effectively to keep track of outstanding problems with their product. It is written in Perl and uses MYSQL database.Bugzilla is a Defect tracking tool, however it can be used as a test management tool as such it can be easily linked with other Test Case management tools like Quality Center, Testlink etc. This open bug-tracker enables users to stay connected with their clients or employees, to communicate about problems effectively throughout the data-management chain Key features of Bugzilla includes

1. Advanced search capabilities
2. E-mail Notifications
3. Modify/file Bugs by e-mail
4. Time tracking
5. Strong security
6. Customization
7. Localization

7.8 Test Cases



Test Case No	Test Data	DB Table Name(s) Influenced	Form(s) Report(s) involved	Expected Result	Actual Result	Remarks
1	Admin admin	tbl_Adminlogin	adminlogin	Successful	Successful	Good
2	Staff Registration	staff	staffreg	Register staffs	Register staffs	Good
3	User Registration	user	Userreg	Register a user	Register a user	Good
4	User complaints	user	usercomp	Add user complaints	Complaints added successfully	Good
5	View staff	**staff**	Viewstaff	View all staff	View all staff	Good

8 SYSTEM IMPLEMENTATION

8.1 Introduction to System Implementation

The implementation is the final state and it is an important phase. It involves the individual programming; system testing, user training and the operational running of developed proposed system that constitutes the application subsystems. A major task of preparing for implementation is education of users, which should really have been taken place much earlier in the project when they were being involved in the investigation and design work. During the implementation phase system actually takes physical shape. In order to develop a system implemented planning is very essential. A software implementation method is a systematically structured approach to effectively integrate software based service or component into the workflow of an organizational structure or an individual end-user. This entry focuses on the process modeling (Process Modeling), a process model is a description of a process at the type level, side of the implementation of large product software, using the implementation of Enterprise Resource Planning systems as the main example to elaborate on. A product software implementation method is a blueprint to get users and/or organizations running with a specific software product. The method is a set of rules and views to cope with the most common issues that occur when implementing a software product: business alignment from the organizational view and acceptance from the human view. The implementation of product software, as the final link in the deployment chain of software production, is in a financial perspective of a major issue. The implementation phase of the software development is concerned with translating design specification into source code. The user tests the developed system and changes are made according to their needs. Our system has been successfully implemented. Before implementation several tests have been conducted to ensure that no errors are encountered during the operation. The implementation phase ends with an evaluation of the system after placing into the operation for a period of time. Implementation is the third phase of the system processe.

8.2 Training

An analysis of user training focuses on two factors:

1. User capabilities

2. Nature of the system to be installed.

Users range from the native to the highly sophisticated. They approach it as concrete learners, learning how to use the system without trying to understand which abstract principles determine which function. The distinction between concrete and formal (student type) learning says about what one can expect from trainees in general.These project also sophisticated the user capabilities and the corresponding nature of the system to be installed.

8.3 Conversion

Conversion refers to changing from one design to another system. The main objective of conversion is to put tested system into operation while holding costs, risks, and personal irritation to minimum. The various tasks involved in conversion are:

1. Creating computer compatible files.
2. Training the operating staffs.
3. Installing terminals and hardware.

The project entitled" Collectorate management agreed the conversion phases that begin with a review of the project plan, the system test documentation and the implementation plan. And also conversion portion of the implementation plan is finalized and approved. Files are converted.

8.4 Post Implementation Review

Every system requires periodic evaluation after implementation. A post implementation review measures the systems performance against predefined requirements. Unlike system testing, which determines where the system fails so that the necessary adjustments can be made, a post-implementation review determines how well the system continues to meet performances specifications. It is done after design and conversion are complete. It also provides information to determine whether major redesign is necessary.

8.5 System Maintenance

Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes. This section describes the six software maintenance processes as:

1. The implementation processes contains software preparation and transition activities, such as the conception and creation of the maintenance plan, the preparation for handling problems identified during development, and the follow-up on product configuration management.

2. The problem and modification analysis process, which is executed once the application has become the responsibility of the maintenance group. The maintenance programmer must analyze each request, confirm it (by reproducing the situation) and check its validity, investigate it and propose a solution, document the request and the solution proposal, and, finally, obtain all the required authorizations to apply the modifications.

3. The process considering the implementation of the modification itself.

4. The process acceptance of the modification, by confirming the modified work with the individual who submitted the request in order to make sure the modification provided a solution.

5. The migration process is exceptional, and is not part of daily maintenance tasks. If the software must be ported to another platform without any change in functionality, this process will be used and a maintenance project team is likely to be assigned to this task.

6. Finally, the last maintenance process, also an event which does not occur on a daily basis, is the retirement of a piece of software.

9 SYSTEM EVALUATION

Although system evaluation is an ongoing process throughout the performance testing effort, it offers greater value when conducted early in the test project. The intent of system evaluation is to collect information about the project as a whole, the functions of the system, the expected user activities, the system architecture, and any other details that are helpful in guiding performance testing to achieve the specific needs of the project.

1. Your need to evaluate and select software that meets your business requirements.

2. Your need to evaluate and select a partner that is capable of delivering the most benefit to your business from your software investment, as well as managing the risks inherent in system implementation projects.

3. Your time and ours is valuable; at each step along the way we will each decide whether or not it is beneficial to proceed.

To help you with your selection, this evaluation process is designed to give us both a clear understanding of the systems to be implemented and the corresponding benefits of the partnership.This information provides a foundation for collecting the performance goals and requirements, characterizing the workload, creating performance-testing strategies and plans, and assessing project and system risks. A thorough understanding of the system under test is critical to a successful performance-testing effort. The measurements gathered during later stages are only as accurate as the models that are developed and validated in this stage. The evaluation provides a foundation for determining acceptable performance; specifying performance requirements of the software, system, or component(s); and identifying any risks to the effort before testing even begins. System evaluation providing in these project is needed to evaluate and select the requirements and managing the risk in system implementation on project. Also it is valuable in time so that way it is beneficial in each step.

10 CONCLUSION

The project was successfully completed within the time span allotted .The drawbacks of the existing system as listed before are fully evacuated. All the existing inconsistencies are fully solved as this system is implemented. This reduced the burden of the administration of the system. All the modules are tested separately and put together to form the main system. Finally the system is tested with real data and it worked successfully. Thus the system has fulfilled the entire objective defined. In this system collector or higher officials considered as the administrator in this system. The administrate has the responsibility for approving or reject the form that have been applied by the user. The petitions applied by the public peoples are check by the administrator and taking necessary actions against the petition if it is legal. The end users are the public people. Once he/she registered in the system, then they can sign in to apply the several forms and participate in tenders. Online petition also available in registered users interface and they can check the status of their petition whether action taken for his petition or not.

10.1 SCOPE FOR FUTURE ENHANCEMENTS

In future we can expect the modified version of a collecterate management. The system is very flexible for further up gradation with additional requirement of the self working, the jsp and hidisql server makes this modifications very easily It is also possible to involve more functions into the system. This flexibility makes this system widening its scope. All day to day work can be done with much more ease and efficiency. The database and the information can be updated to the latest coming versions. There are also possibilities for enhancing and further developing the project with the latest information and needs of the portal.

11 APPENDIX

11.1 APPENDIX A

11.1.1 Sample Source Code / Pseudo Code

```
\begin{itemsize}
\item COMPANY
\end{iemsize}

Staff reg
<div class="article" id="article1">

<form method="post" action="../process/regactionemplyee.jsp">
<h2>New Staff Registration
</h2>
<table>
<tr>
<td>Full Name</td><td><input type="text" name="fullname" value="" placeholder="F
</tr>
<tr>
<td>Gender</td><td>Male<input type="radio" value="male" name="gender"></input>Fe
</tr>
<tr>
<td>Father Name</td><td><input type="text" name="fname" value="" placeholder="Fa
</tr>
<tr>
<td>Mother Name</td><td><input type="text" name="mname" value="" placeholder="Mo
</tr>


<tr>
<td>Date OF Birth</td><td><input type="date" name="dob" value="" placeholder="Da
</tr>

<tr>
<td>Address</td><td><textarea rows="5" cols="20" name="address"></textarea></td>
</tr>
```

```
<tr>
<td>Phone Number</td><td><input type="text" name="phno"></input></td>
</tr>

<tr><td>Department</td>
<td><select name="dept">
<option value="0">--select--</option>
<option value="corporation">Corporation</option>
<option value="roadt">Road Transport</option>
<option value="revenue">Revenue</option>
<option value="pwd">P.W.D</option>
</select></td>
</tr>
<tr>
<td>Email</td><td><input type="email" name="email"  placeholder="Email" required
</tr>
<tr>
<td>User Name</td><td><input type="text" name="uname"  placeholder="User Name" r
</tr>
<tr>
<td>Password</td><td><input type="password" name="pass"  placeholder="Password"
</tr>
<tr>
<td></td><td><input type="submit"  value="Register" required=""/></td>
</tr>
</table>
</form>



</div>


<div class="article" id="article1">

<form method="post" action="../process/regactionemplyee.jsp">
<h2>New Staff Registration
</h2>
<table>
<tr>
```

```
<td>Full Name</td><td><input type="text" name="fullname" value="" placeholder="F
</tr>
<tr>
<td>Gender</td><td>Male<input type="radio" value="male" name="gender"></input>Fe
</tr>
<tr>
<td>Father Name</td><td><input type="text" name="fname" value="" placeholder="Fa
</tr>
<tr>
<td>Mother Name</td><td><input type="text" name="mname" value="" placeholder="Mo
</tr>


<tr>
<td>Date OF Birth</td><td><input type="date" name="dob" value="" placeholder="Da
</tr>

<tr>
<td>Address</td><td><textarea rows="5" cols="20" name="address"></textarea></td>
</tr>

<tr>
<td>Phone Number</td><td><input type="text" name="phno"></input></td>
</tr>

<tr><td>Department</td>
<td><select name="dept">
<option value="0">--select--</option>
<option value="corporation">Corporation</option>
<option value="roadt">Road Transport</option>
<option value="revenue">Revenue</option>
<option value="pwd">P.W.D</option>
</select></td>
</tr>
<tr>
<td>Email</td><td><input type="email" name="email"  placeholder="Email" required
</tr>
<tr>
<td>User Name</td><td><input type="text" name="uname"  placeholder="User Name" r
</tr>
<tr>
```

```
<td>Password</td><td><input type="password" name="pass"  placeholder="Password" 
</tr>
<tr>
<td></td><td><input type="submit"  value="Register" required=""/></td>
</tr>
</table>
</form>



</div>
```

User reg

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.or
<!--
Design by http://www.bluewebtemplates.com
Released for free under a Creative Commons Attribution 3.0 License
-->
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Template</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link href="style.css" rel="stylesheet" type="text/css" />
<!-- CuFon: Enables smooth pretty custom font rendering. 100% SEO friendly. To d
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/arial.js"></script>
<script type="text/javascript" src="js/cuf_run.js"></script>
<!-- CuFon ends -->
</head>
<body>
<div class="main">

<div class="header">
<div class="header_resize">
<div class="logo"><h1><a href="index.html">My Collectorate<br /><small>Trusted S
<div class="menu_nav">
<ul>
<li class="active"><a href="home.jsp"><span><span>Home</span></span></a></li>
```

```
</ul>
</div>
<div class="clr"></div>
<img src="images/eDis.jpg" width="970" height="250" alt="image" />
<div class="clr"></div>
</div>
</div>

<div class="content">
<div class="content_resize">
<div class="mainbar">
<div class="article">
<center>
<h2>USER REGISTRATION </h2>
<form method="post" action="process/regactionuser.jsp">
<table>
<tr>
<td>Full Name</td><td><input type="text" name="fullname" value="" placeholder="F
</tr>
<tr>
<td>Gender</td><td>Male<input type="radio" value="male" name="gender"></input>
                                            Female<input type="radio" value="fema
</tr>
<tr>
<td>Father Name</td><td><input type="text" name="fname" value="" placeholder="Fa
</tr>
<tr>
<td>Mother Name</td><td><input type="text" name="mname" value="" placeholder="Mo
</tr>


<tr>
<td>Date OF Birth</td><td><input type="date" name="dob" value="" placeholder="Da
</tr>

<tr>
<td>Address</td><td><textarea rows="5" cols="20" name="address"></textarea></td>
</tr>

<tr>
```

```
<td>Phone Number</td><td><input type="number" name="phno" size="10" pattern="^[a
</tr>


<tr>
<td>Email</td><td><input type="email" name="email"  placeholder="Email" required
</tr>
<tr>
<td>User Name</td><td><input type="text" name="uname"  placeholder="User Name" r
</tr>
<tr>
<td>Password</td><td><input type="password" name="pass"  placeholder="Password"
</tr>
<tr>
<td></td><td><input type="submit"  value="Register" required=""/></td>
</tr>
</table>
</form>

</center>


</div>

</div>
<div class="sidebar">
<div class="gadget">
<!--<h2 class="star"><span>Sidebar</span> Menu</h2>-->
<ul class="sb_menu">
<li><a href="home.jsp">Back to Login</a></li>

</ul>
</div>
<div class="gadget">

</div>
</div>
<div class="clr"></div>
</div>
</div>
```

```
<div class="fbg">
<div class="fbg_resize">
<div class="col c1">

</div>
<div class="col c2">

</div>
<div class="col c3">
<!--<h2><span>About</span></h2>-->
<!--        <img src="images/white.jpg" width="56" height="56" alt="pix" />
<p>convallis aliquam, lectus turpis varius lorem, eu posuere nunc justo tempus l
<p><a href="#" class="lbg">Learn more...</a></p>-->
</div>
<div class="clr"></div>
</div>
</div>
<div class="footer">
<div class="footer_resize">

</div>
</div>
</div>
</body>
</html>
```

User complaints view

```
<%@page import="java.util.Iterator"%>
<%@page import="java.util.Vector"%>
<%@page import="connection.DbConnection"%>

<div class="article" id="article1">
<h2>View My Petitions</h2>

<%
        String uid = session.getAttribute("uid").toString();
        String str = "SELECT * FROM ‘pet_tb‘ WHERE uid=" + uid;
        System.out.println(str);
        DbConnection dcon = new DbConnection();
        Vector v = dcon.getData(str);
        Iterator itr = v.iterator();
        if (itr.hasNext()) {
    %>
<table border="1">
<!--<th>User Name</th>-->
<th>About</th>
<th>Complaint</th>
<th>Complaint status</th>
<%
                while (itr.hasNext()) {
                    Vector vv = (Vector) itr.next();
            %>
<tr><td><%=vv.get(4)%></td><td><%=vv.get(5)%></td><td><%=vv.get(6)%></td></tr>

<%
            }%>
</table>
<%
    } else {
    %>
<center><h2>No new Complaint replay</h2></center>
<%
            }
        %>




</div>
```

User apply tender

```
<%
connection.DbConnection con=new connection.DbConnection();
String tendid=request.getParameter("tendid").toString(),uid=session.getAttribute

String strIns="update tend_tab set uid=’"+uid+"’ where ten_id=’"+tendid+"’";
```

```
if(con.putData(strIns)>0)
{
    %>
<script>
        alert("Tender Claimed Succesfully");
        window.location="userhome.jsp";
</script>
<%
}else
{
   %>
<script>
        alert("Unable to claim");
        window.location="userhome.jsp";
</script>
<%
}


%>
```

office

```
<%@page import="java.util.Vector"%>
<%@page import="java.util.Iterator"%>
<div class="article" id="article1">
<h2>Office Details</h2>
<%
    connection.DbConnection con=new connection.DbConnection();
String strOff="select * from office order by offtype";
Iterator itOff=con.getData(strOff).iterator();
%>
<table border="2"><tr>
<th>Office Type</th><th>Office name</th><th>No of employess</th><th>Building nam
</tr>
<%
while(itOff.hasNext())
{
    Vector vOff=(Vector)itOff.next();
    %>
<tr>
<td><%
    out.println(vOff.get(1));
    %></td>
<td><%
    out.println(vOff.get(2));
    %>
<td><%
    out.println(vOff.get(3));
    %></td>
<td><%
    out.println(vOff.get(4));
    %></td>
</tr>
<%

}
%>
</table>
</div>
```

View staffs

```
<%@page import="java.util.Vector"%>
<%@page import="java.util.Iterator"%>
<div class="article" id="article1">
<h2>Staff Details</h2>
<%
    connection.DbConnection con=new connection.DbConnection();
String strOff="select * from regtable where departmnt!='user' and departmnt!='ad
Iterator itOff=con.getData(strOff).iterator();
%>
<table border="2"><tr>
<th>Name</th><th>Address</th><th>Contact</th><th>Email</th><th>Department</th>
</tr>
<%
while(itOff.hasNext())
{
    Vector vOff=(Vector)itOff.next();
    %>
<tr>
<td><%
    out.println(vOff.get(1));
    %></td>
<td><%
    out.println(vOff.get(6));
    %>
<td><%
    out.println(vOff.get(7));
    %></td>
<td><%
    out.println(vOff.get(9));
    %></td>
<td><%
    out.println(vOff.get(8));
    %></td>
</tr>
<%

}
%>
</table>
</div>
```

User notification

```
<%@page import="beanfiles.Tender"%>
<%@page import="java.util.Iterator"%>
<%@page import="connection.DbConnection"%>
<%@page import="java.util.Vector"%>

<div class="article" id="article1">


<h1>Welcome  <%out.print(session.getAttribute("uname").toString());%></h1><br></

<h2>Your Approved Nottification</h2>
```

```
<table>
<!--               <th>Tender Id</th>
<th>Bid Value</th>
<th>Details</th>
<th>Status</th>-->
<%
           String uid=session.getAttribute("uid").toString();
           String str="SELECT * FROM ‘bid_tab‘ WHERE ‘uid‘=’"+uid+"’";
           System.out.println(str);

           DbConnection dcon=new DbConnection();
           Vector v=dcon.getData(str);
           Iterator itr=v.iterator();
           if(itr.hasNext()){
           while(itr.hasNext()){
           Vector vv=(Vector)itr.next();
           %>


<!--                <tr><td>Your Tender Request To Tender Id <%=vv.get(1)%> is A

<!--<tr><td><%=vv.get(0)%></td><td><%=vv.get(3)%></td><td><textarea cols="15" ro
<%
               if(vv.get(5).toString().equals("1")){
                   //out.print("Approved");%>
<tr style="background-color:greenyellow;color: black;"><td>Your Tender Request T
<tr><td></td></tr>
<%
               }
               else if(vv.get(5).toString().equals("2"))
               { %>
<tr style="background-color:red;"><td>Your Tender Request To Tender Id <%=vv.get
<tr><td></td></tr>
<%

               }
               else
               { %>
<tr style="background-color:yellow;color: black;"><td>Your Tender Request To Ten
<tr><td></td></tr>
<%}
```

```
               %></td></tr>
<%}}
           else
                             {

           }
           %>



</table>



</div>
```

View tenders user

```
<%@page import="java.util.Vector"%>
<%@page import="java.util.Iterator"%>
<div class="article" id="article1">
<h2>Tenders View</h2>
<%
        connection.DbConnection con = new connection.DbConnection();
        String strOff = "select * from tend_tab where uid='0'";

        Iterator itOff = con.getData(strOff).iterator();
        if (itOff.hasNext()) {
    %>
<table border="2"><tr>
<th>title</th><th>closing date</th><th>Location</th><th>Tender Amount</th>
<th>Open date</th><th>More Details</th>
</tr>
<%
            while (itOff.hasNext()) {
                Vector vOff = (Vector) itOff.next();
        %>
<tr>
<td><%
                out.println(vOff.get(2));
                %></td>
<td><%
```

```
                out.println(vOff.get(3));
                %>
<td><%
                out.println(vOff.get(4));
                %></td>
<td><%
                out.println(vOff.get(5));
                %></td>
<td><%
                out.println(vOff.get(7));
                %></td>
<td><%
                out.println(vOff.get(6));
                %></td>
<td><a href="UserApplyTender.jsp?tendid=<%=vOff.get(0)%>">Apply</a></td>
</tr>
<%

            }
        } else {
        %>
<h3>No New Tenders</h3>
<%
            }
        %>
</table>
    -----------------------------------------------------------------------
<br/><br/><br/>
<h2>Your Tenders Approved by Administrator</h2>
    ----------------------------------------------------------------------------
<%
            String strOff1 = "select * from tend_tab where uid='" + session.getA
System.out.println("strOff1--"+strOff1);
            Iterator itOff1 = con.getData(strOff1).iterator();
            if (itOff1.hasNext()) {
        %>
<table border="2"><tr>
<th>title</th><th>closing date</th><th>Location</th><th>Tender Amount</th>
<th>Open date</th><th>More Details</th>
</tr>
<%
```

```
        Vector vOff = (Vector) itOff1.next();
      %>
<tr>
<td><%
          out.println(vOff.get(2));
          %></td>
<td><%
          out.println(vOff.get(3));
          %>
<td><%
          out.println(vOff.get(4));
          %></td>
<td><%
          out.println(vOff.get(5));
          %></td>
<td><%
          out.println(vOff.get(7));
          %></td>
<td><%
          out.println(vOff.get(6));
          %></td>

</tr>
<%
      } else {
      %>
<h3>You have no tenders</h3>
<%
        }
      %>


</table>
</div>
```

Petrol pump user

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.or
<!--
Design by http://www.bluewebtemplates.com
```

```
Released for free under a Creative Commons Attribution 3.0 License
-->
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>My Collectorate</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link href="style.css" rel="stylesheet" type="text/css" />
<!-- CuFon: Enables smooth pretty custom font rendering. 100% SEO friendly. To d
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/arial.js"></script>
<script type="text/javascript" src="js/cuf_run.js"></script>
<!-- CuFon ends -->
</head>
<body>

<div class="main">

<div class="header">
<div class="header_resize">
<div class="logo"><h1><a href="index.html">My Collectorate<br /><small>Trusted S
<div class="menu_nav">
<ul>
<li class="active"><a href="home.jsp"><span><span>Home</span></span></a></li>

</ul>
</div>
<div class="clr"></div>
<img src="images/eDis.jpg" width="970" height="250" alt="image" />
<div class="clr"></div>
</div>
</div>

<div class="content">
<div class="content_resize">
<div class="mainbar">
<div class="article" id="article1">
<center>
<!--jj-->
<h1>Welcome  <%out.print(session.getAttribute("uname").toString());%></h1>
<br/><br/><br/>
</center>
```

```
<!--<img src="images/58396f370c715-c6b696ed5174a4da834e23c5aace0ef1.jpg" width="
<h2>Petrol Pumb Licencing</h2>
<form method="post" action="process/PertolPumbProcess.jsp">
<table>

<tr>
<td>Owner Name</td><td><input type="text" name="ownername" value="" required=""
</tr>




<tr>
<td>Owner Address</td><td><textarea rows="5" cols="20" name="owneraddress"></tex
</tr>

<tr>
<td>Contact Number</td><td><input type="number" name="phno"></input></td>
</tr>



<tr>
<td>Email</td><td><input type="email" name="email"   required=""/></td>
</tr>
<tr>
<td>Pertol Pumb Name</td><td><input type="text" name="pname"  required=""/></td>
</tr>
<tr>
<td>Location</td><td><input type="text" name="location"  required=""/></td>
</tr>
<tr>
<td>Dealer Ship</td><td><input type="text" name="dealrship"  required=""/></td>
</tr>
<tr>
<td></td><td><input type="submit"  value="Submit" /></td>
</tr>
</table>
</form>

</div>

</div>
```

```
<div class="sidebar">
<div class="gadget">
<h2 class="star"><span>User</span> Panel</h2>
<ul class="sb_menu">
<li><a href="userhome.jsp">Home</a></li>

<!--                              <li><a href="Petrolpumb.jsp">Petrol Pumb Lic
<li><a href="CrackerShop.jsp">Cracker Shop Licence</a></li>         -->

<!--<li><a href="logoutjsp.jsp" >Logout</a></li>-->

</ul>

<ul class="sb_menu">

<!--<li><a href="#" onclick="loadpage(2)">View Feedback</a></li><li><a href="#" 


</ul>
</div>
<div class="gadget">

</div>
</div>
<div class="clr"></div>
</div>
</div>

<div class="fbg">
<div class="fbg_resize">
<div class="col c1">
<!--                        <h2><span>Image Gallery</span></h2>
<a href="#"><img src="images/pix1.jpg" width="56" height="56" alt="pix" /></a>
<a href="#"><img src="images/pix2.jpg" width="56" height="56" alt="pix" /></a>
<a href="#"><img src="images/pix3.jpg" width="56" height="56" alt="pix" /></a>
<a href="#"><img src="images/pix4.jpg" width="56" height="56" alt="pix" /></a>
<a href="#"><img src="images/pix5.jpg" width="56" height="56" alt="pix" /></a>
<a href="#"><img src="images/pix6.jpg" width="56" height="56" alt="pix" /></a>--
</div>
<div class="col c2">
<!--                        <h2><span>Lorem Ipsum</span></h2>
```

```
<p>Lorem ipsum dolor<br />Lorem ipsum dolor sit amet, consectetuer adipiscing el
</div>
<div class="col c3">
<!--                        <h2><span>About</span></h2>
<img src="images/white.jpg" width="56" height="56" alt="pix" />
<p>convallis aliquam, lectus turpis varius lorem, eu posuere nunc justo tempus l
<p><a href="#" class="lbg">Learn more...</a></p>-->
</div>
<div class="clr"></div>
</div>
</div>
<div class="footer">
<div class="footer_resize">
<!--                    <p class="lf">&copy; Copyright MyWebSite. Designed by Bl
<ul class="fmenu">
<li class="active"><a href="index.html">Home</a></li>
<li><a href="support.html">Support</a></li>
<li><a href="blog.html">Blog</a></li>
<li><a href="about.html">About Us</a></li>
<li><a href="contact.html">Contacts</a></li>
</ul>
<div class="clr"></div>-->
</div>
</div>
</div>
</body>
</html>

</td>
</tr>
<tr>
<td>
<img alt="no image" src="../profImages/<%=rsUsers.getString("profImage")%>" widt
<h3><%=rsUsers.getString("firstname") + " " + rsUsers.getString("lastname")%></h
<span>Uploaded on : <%=rsUsers.getString("dateofupload") %></span>
</td>
</tr>
<tr>
<td>
<span class="likecount"><%=rsUsers.getString("likeCount")%></span>
<%
```

```
if(rsUsers.getString("likeStatus")==null || rsUsers.getString("likeStatus").equa
%>
<a href="like_action.jsp?postid=<%=rsUsers.getString("postid")%>&status=1" class
<%
}else{
%>
<a href="like_action.jsp?postid=<%=rsUsers.getString("postid")%>&status=0" class
<%
}
%>
<span style="float:right"><a href="#" class="acomment">Comment</a></span>
</td>
</tr>
<tr>
<td><a href="viewposts.jsp" >Hide comments</a></td>
</tr>
<%
String selectComment="select c.dateofcomment,c.comment,u.firstname,u.lastname,u.
ResultSet rsComments=con.getData(selectComment);
while(rsComments.next()){
%>
<tr>
<td>
<img alt="no image" src="../profImages/<%=rsComments.getString("profImage")%>" w
<h3><%=rsUsers.getString("firstname") + " " + rsComments.getString("lastname")%>
<span><%=rsComments.getString("comment") %></span><br/>
<span>Uploaded on : <%=rsComments.getString("dateofcomment") %></span>
</td>
</tr>
<%
}
%>
<tr class="visi commentDiv">
<td>
<textarea class="commentBox" rows="8" cols="80"></textarea><br/>
<div class="button"><span><span><a href="sendcomment.jsp?postid=<%=rsUsers.getSt
</td>
</tr>
</table>
<div>
</div>
```

```
</li>
<%
}
%>
</ul>

</div>
</div>
</div>
<div class="left-bot-corner png"><div class="right-bot-corner png"><div class="b
</div>
<!-- box end -->
</div>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<jsp:useBean class="com.myspace.dataaccess.DataAccess" id="con"/>
<%@page import="java.sql.*" %>
<%

String selectUser="select fl.friendid,u.firstname,u.lastname,u.profImage from fr
" union "+
" select fl.userid as friendid,u.firstname,u.lastname,u.profImage from friendlis
ResultSet rsUsers=con.getData(selectUser);
%>
<%@include file="header.jsp" %>
<script>
$(document).ready(function(){
$('#sendmessage').on('click',function(e){
e.preventDefault();
var message=$('#messagebox').val();
if(message==''){
alert('Please enter some message');
}else
{
window.location=$(this).attr('href')+"&&message="+message+"&m=m";
}
});
});
</script>
<!DOCTYPE html>
<div class="section">
```

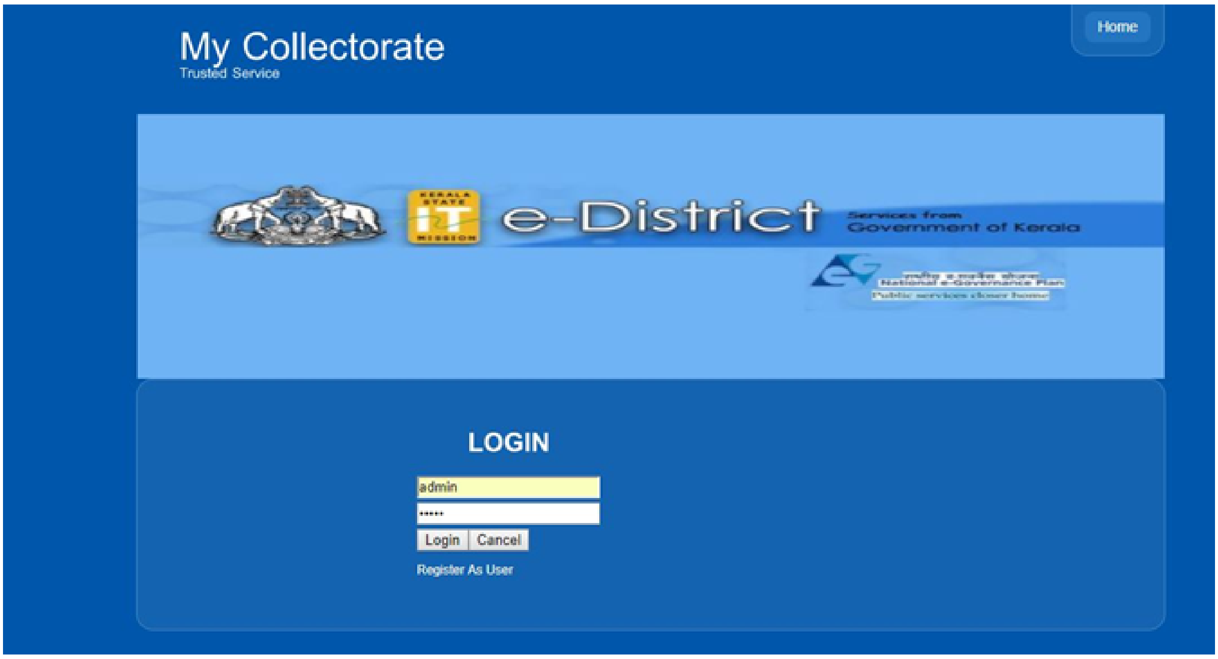
```
<!-- box begin -->
<div class="box">
<div class="left-top-corner png"><div class="right-top-corner png"><div class="b
<div class="border-left png">
<div class="border-right png">
<div class="inside png">
<%-- <textarea rows="8" cols="80" id="messagebox" name=></textarea>--%>
<h2>Users search result</h2>
<div><%
String err=request.getParameter("err");
%><%=(err!=null?err:"")%></div>
<ul class="items-list">
<%
if(rsUsers!=null){
while(rsUsers.next()){
%>
<li>
<img alt="no image" src="../profImages/<%=rsUsers.getString("profImage")%>" />
<h3><%=rsUsers.getString("firstname")+" "+rsUsers.getString("lastname")%></h3>
<%-- <div class="button"><span><span><a href="sendmessage.jsp?friendid=<%=rsUser
</li>
<%
}}else{
%>
<li>
<h3>No result</h3>
</li>
<%
}%>
</ul>

</div>
</div>
</div>
<div class="left-bot-corner png"><div class="right-bot-corner png"><div class="b
</div>
<!-- box end -->
</div>

<%@include file="footer.jsp" %>
<%@include file="footer.jsp" %>
```

```
<%@include file="AdminFooter.jsp" %>
package com.myspace.dataaccess;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
*
* @author Dinesh
*/
public class DataAccess {

Connection con = null;
Statement st = null;
ResultSet rs = null;

public DataAccess() {
try {
Class.forName("com.mysql.jdbc.Driver");
con = DriverManager.getConnection("jdbc:mysql://localhost/myspace", "root", "roo

} catch (SQLException ex) {
Logger.getLogger(DataAccess.class.getName()).log(Level.SEVERE, null, ex);
} catch (ClassNotFoundException ex) {
Logger.getLogger(DataAccess.class.getName()).log(Level.SEVERE, null, ex);
}
}

public boolean executeCommand(String query) {
boolean flag = false;
try {
st = con.createStatement();
st.executeUpdate(query);
flag = true;
} catch (SQLException ex) {
Logger.getLogger(DataAccess.class.getName()).log(Level.SEVERE, null, ex);
} finally {
return flag;
}
}

public ResultSet getData(String query) {
try {
st = con.createStatement();
rs = st.executeQuery(query);
} catch (SQLException ex) {
Logger.getLogger(DataAccess.class.getName()).log(Level.SEVERE, null, ex);
} finally {
return rs;
}
}

public int getLength(ResultSet rs) {
int count = 0;
try {

while (rs.next()) {
count++;
}
rs.first();
} catch (SQLException ex) {
Logger.getLogger(DataAccess.class.getName()).log(Level.SEVERE, null, ex);
} finally {
return count;
}
}
}

</div>
</div>
</div>
<div class="left-bot-corner png"><div class="right-bot-corner png"><div class="b
</div>
<!-- box end -->
</div>
```

11.2 APPENDIX B

11.2.1 SCREEN SHOTS

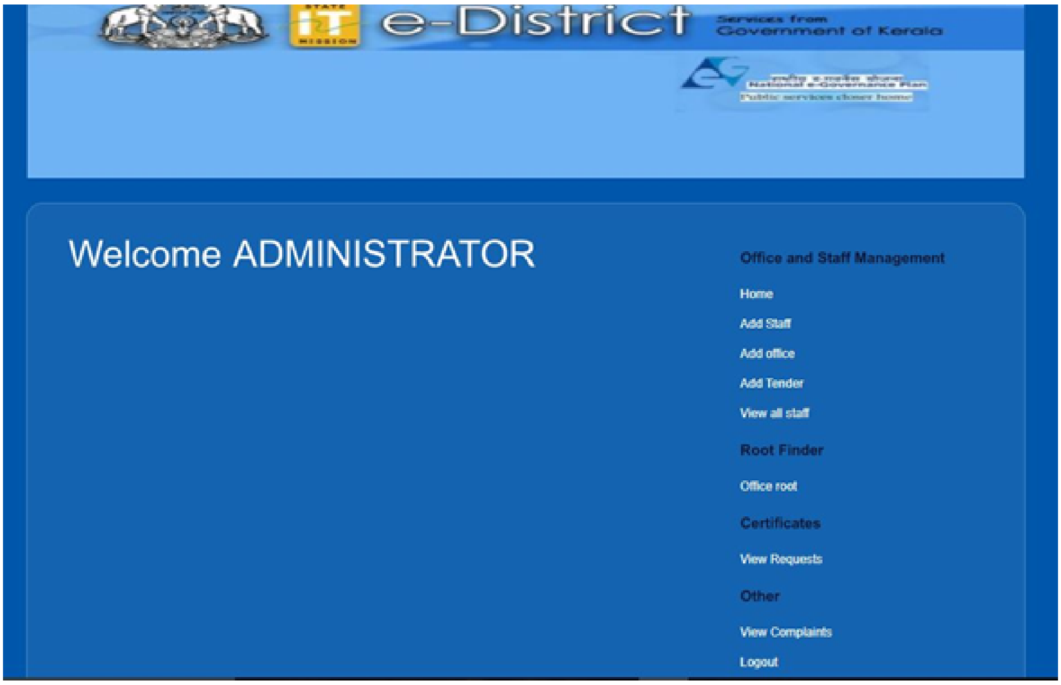
- 11.2.1.1 Admin login



admin

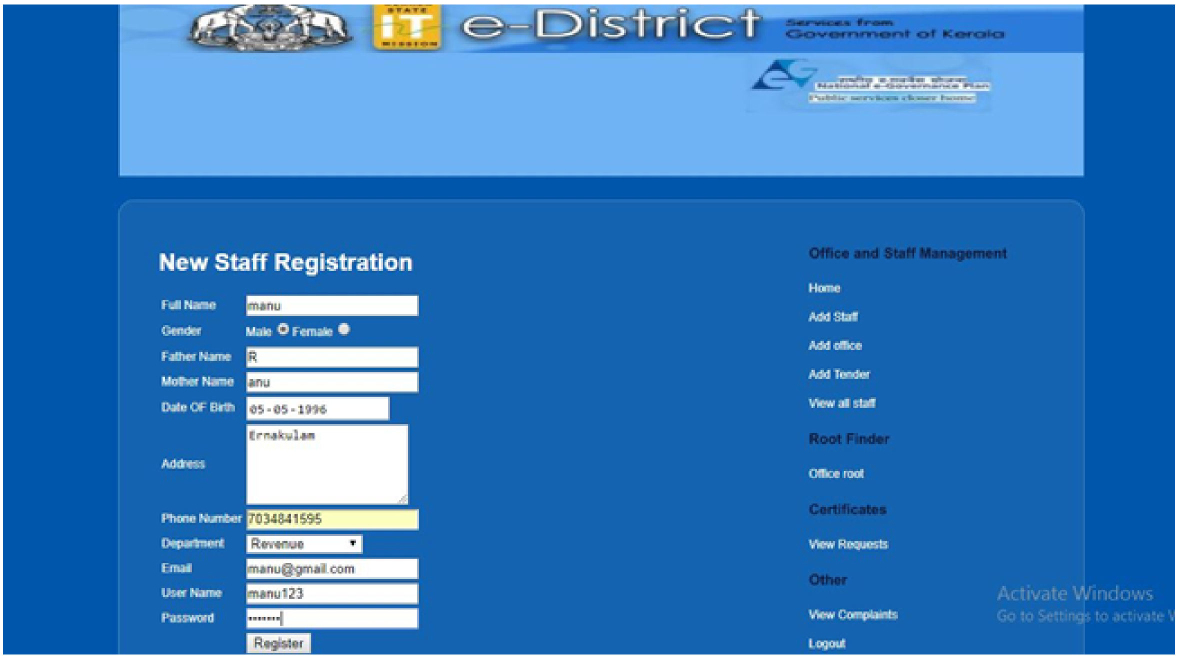
*****

- 11.2.1.2 Admin homepage



Welcome ADMINISTRATOR

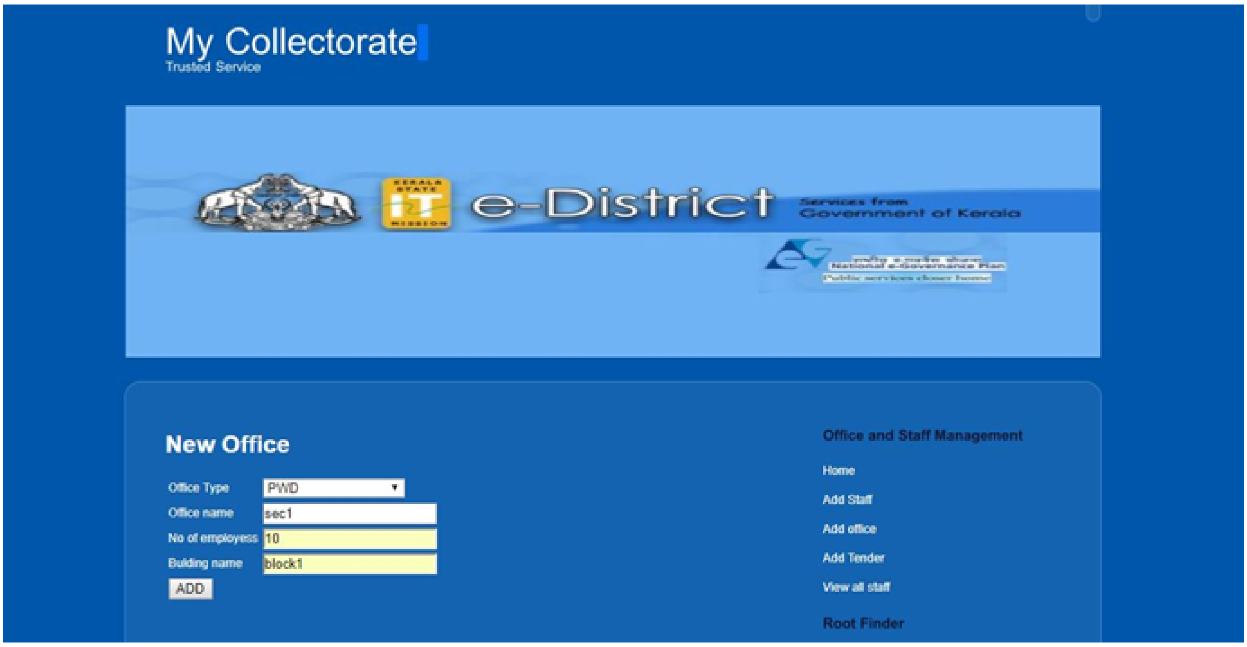
- 11.2.1.3 Staff registration



New Staff Registration

Full Name	manu
Gender	Male ● Female ○
Father Name	R
Mother Name	anu
Date OF Birth	05-05-1996
Address	Ernakulam
Phone Number	7034841595
Department	Revenue
Email	manu@gmail.com
User Name	manu123
Password	•••••••
	Register

- 11.2.1.4 Add office


e-District
Services from Government of Kerala
National e-Governance Plan
Public services closer home

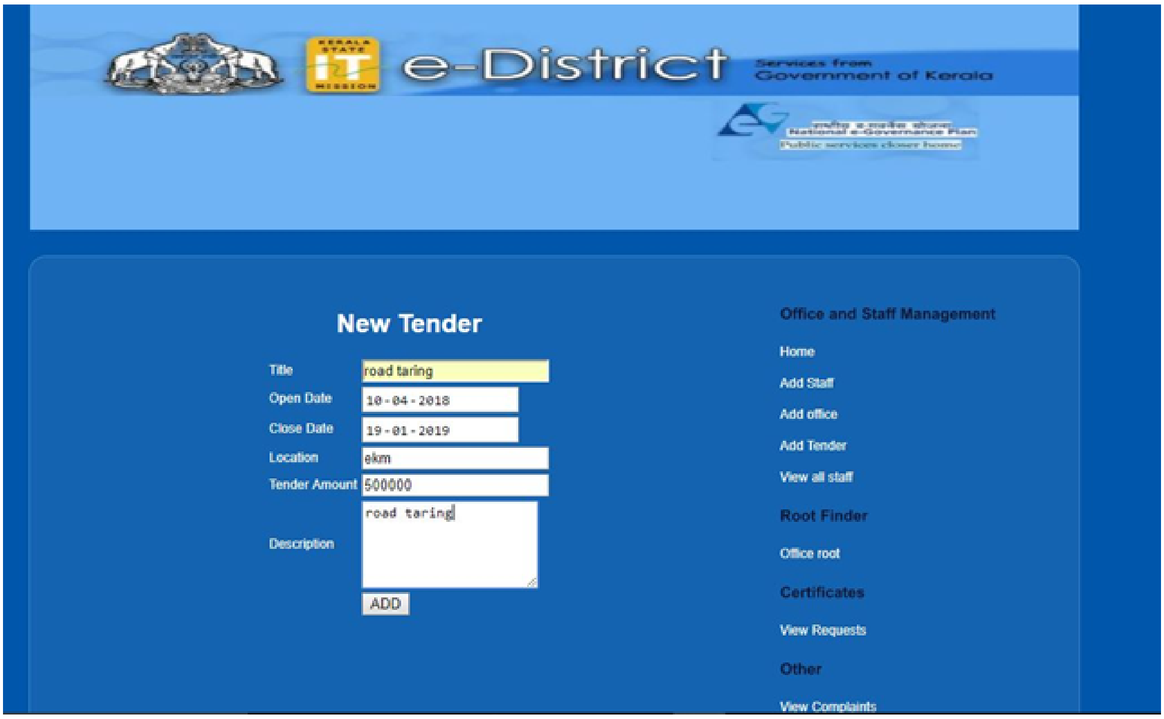
My Collectorate
Trusted Service

New Office

Office Type: PWD ▾
Office name: sec1
No of employess: 10
Bulding name: block1
ADD

Office and Staff Management
Home
Add Staff
Add office
Add Tender
View all staff
Root Finder

- 11.2.1.5 New tender

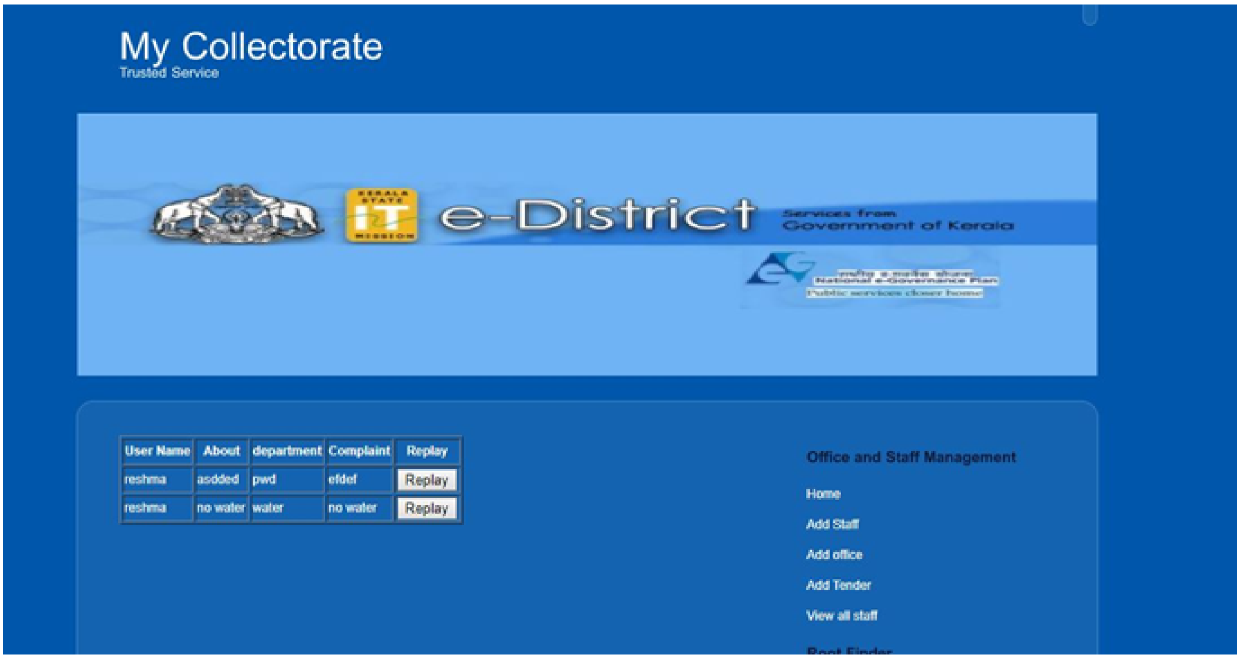


e-District
Services from Government of Kerala
National e-Governance Plan
Public services closer home

New Tender

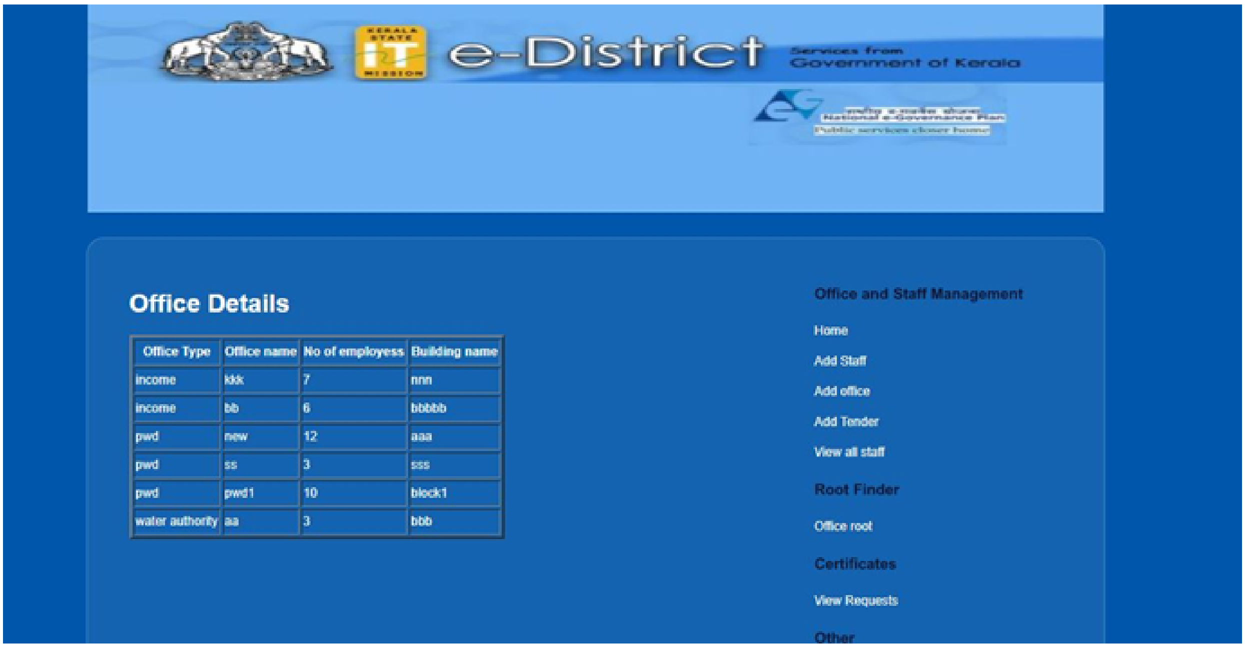
Title	road taring
Open Date	10-04-2018
Close Date	19-01-2019
Location	ekm
Tender Amount	500000
Description	road taring

- 11.2.1.6 View complaints


My Collectorate
Trusted Service
e-District
Services from Government of Kerala
National e-Governance Plan
Public services closer home
Office and Staff Management
Home
Add Staff
Add office
Add Tender
View all staff
Post Finder

User Name	About	department	Complaint	Replay
reshma	asdded	pwd	efdef	Replay
reshma	no water	water	no water	Replay

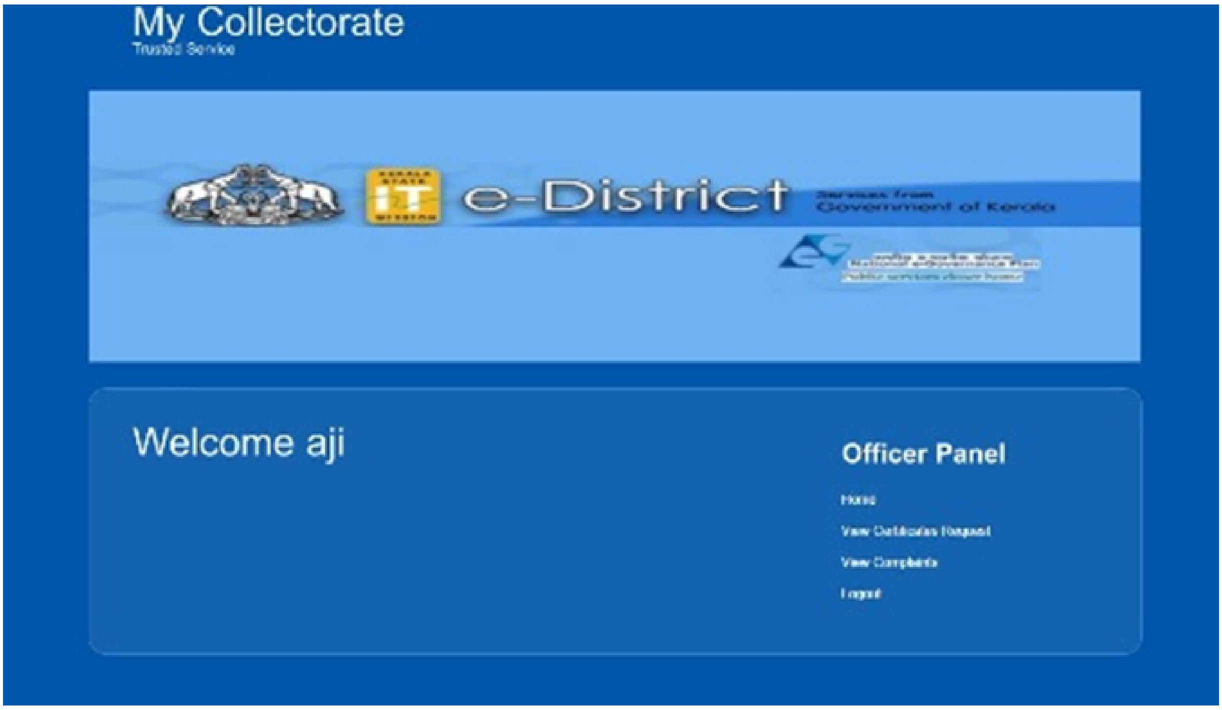
- 11.2.1.7 View staffs



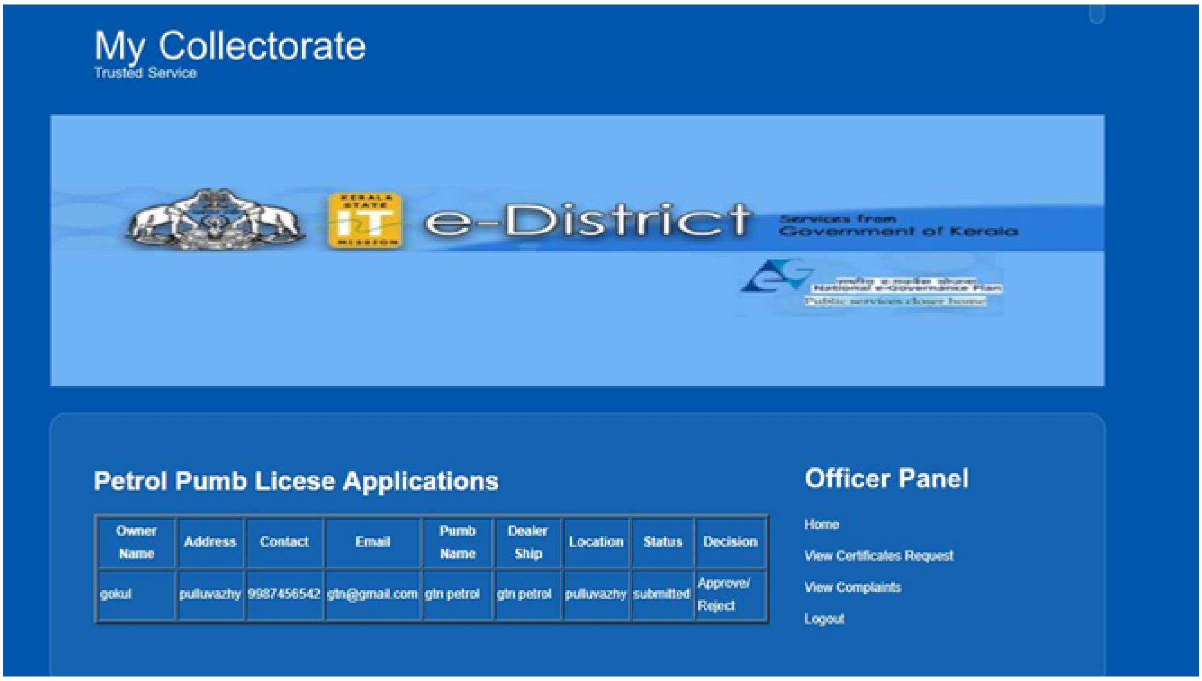
Office Details

Office Type	Office name	No of employess	Building name
income	kkk	7	nnn
income	bb	6	bbbbb
pwd	new	12	aaa
pwd	ss	3	sss
pwd	pwd1	10	block1
water authority	aa	3	bbb

- 11.2.1.7 Office home



- 11.2.1.7 Certificate rqst approve


e-District
Services from Government of Kerala
KERALA STATE IT MISSION
National e-Governance Plan
Public services closer home

My Collectorate
Trusted Service

Petrol Pumb Licese Applications

Owner Name	Address	Contact	Email	Pumb Name	Dealer Ship	Location	Status	Decision
gokul	pulluvazhy	9987456542	gtn@gmail.com	gtn petrol	gtn petrol	pulluvazhy	submitted	Approve/ Reject

Officer Panel

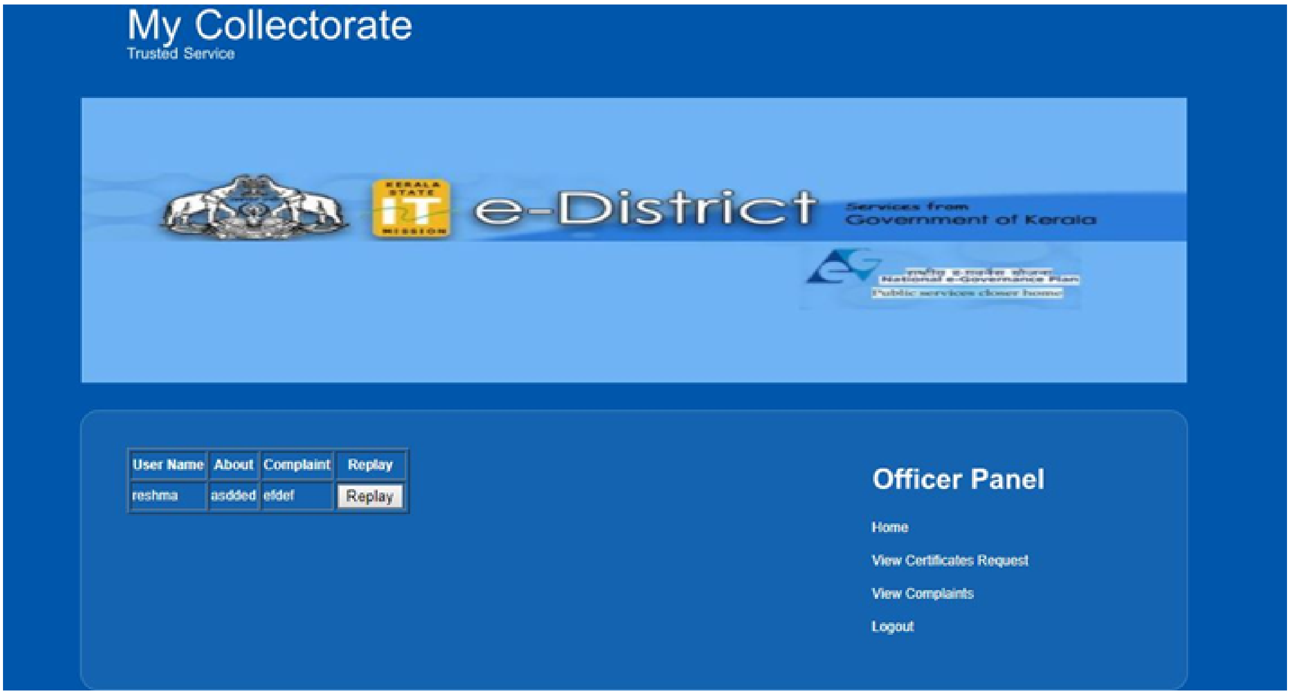
Home

View Certificates Request

View Complaints

Logout

- 11.2.1.7 View complaints



My Collectorate

Trusted Service

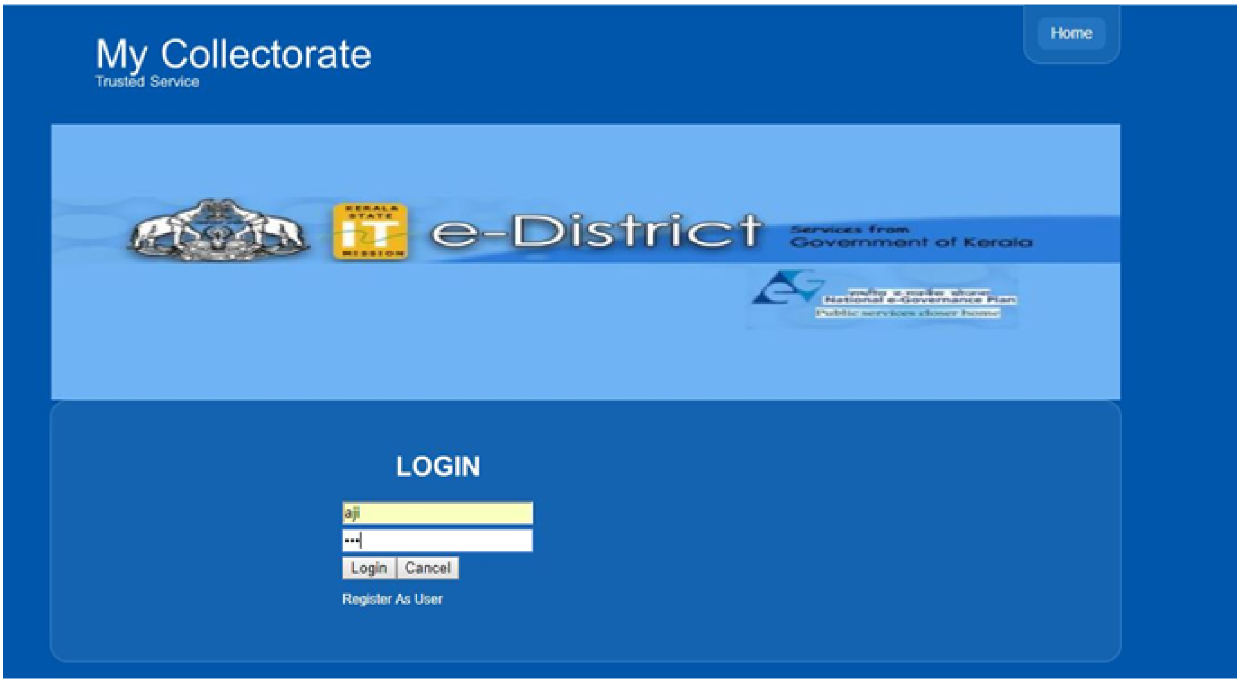
e-District
Services from
Government of Kerala
National e-Governance Plan
Public services closer home

User Name	About	Complaint	Replay
reshma	asdded	efdef	Replay

Officer Panel

Home
View Certificates Request
View Complaints
Logout

- 11.2.1.7 User login

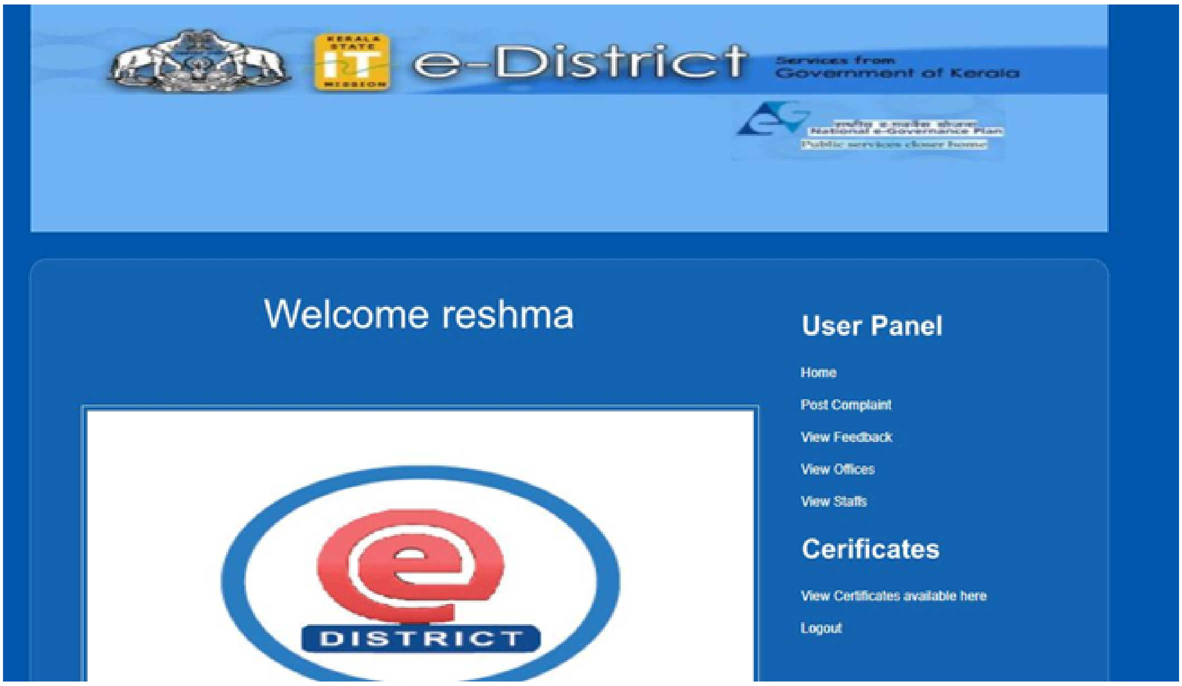


My Collectorate
Trusted Service
Home
e-District
Services from Government of Kerala
National e-Governance Plan
Public services closer home
LOGIN
Login
Cancel
Register As User

aji

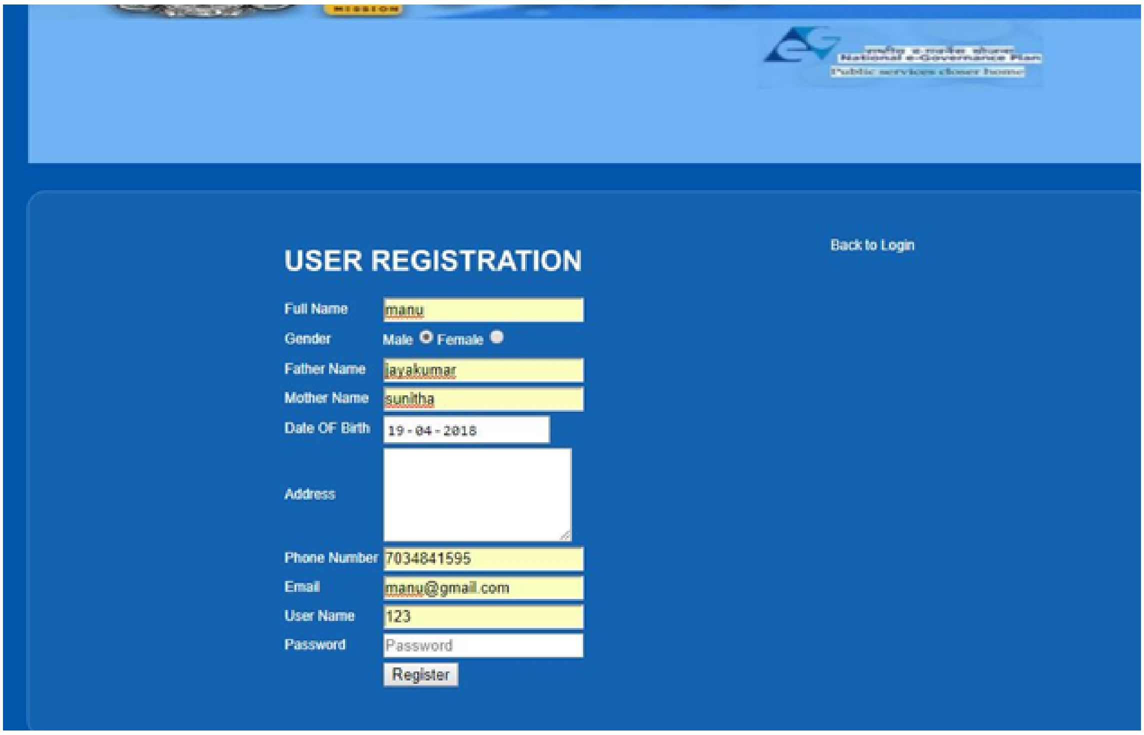
•••

- 11.2.1.7 User homepage



Welcome reshma

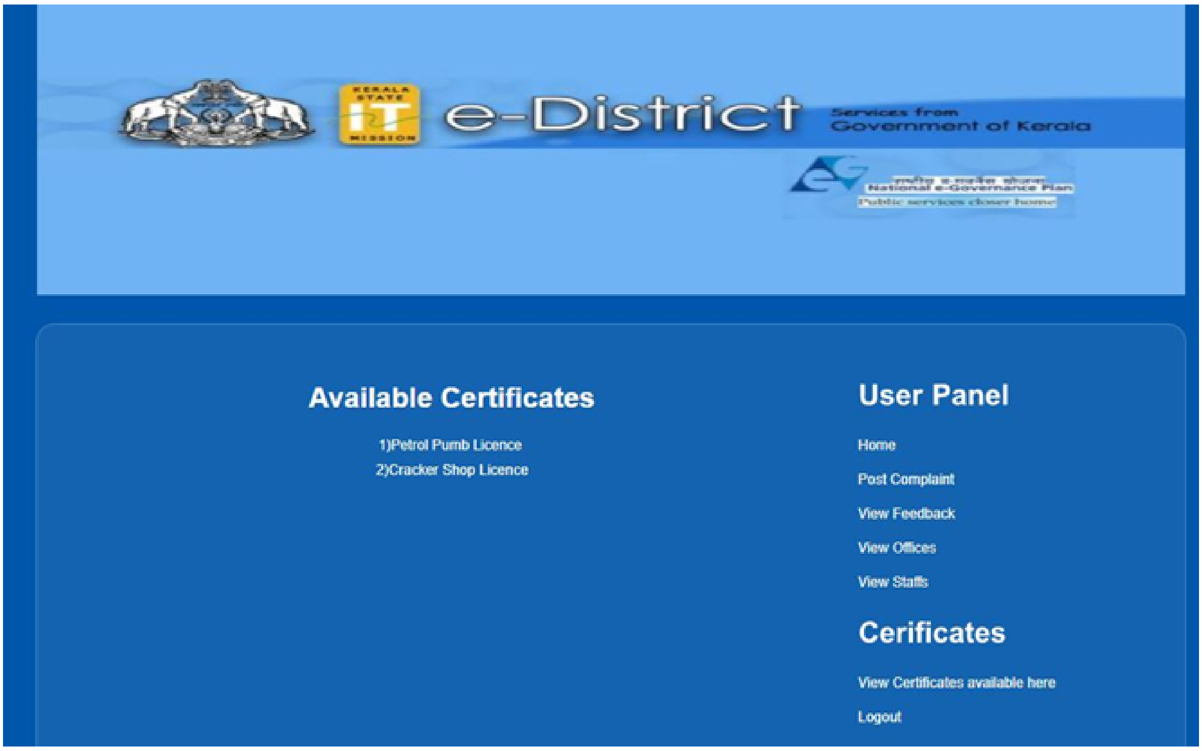
- 11.2.1.7 User registration



USER REGISTRATION

Full Name	manu
Gender	Male ● Female ○
Father Name	jayakumar
Mother Name	sunitha
Date OF Birth	19-04-2018
Address	
Phone Number	7034841595
Email	manu@gmail.com
User Name	123
Password	Password

- 11.2.1.7 Available certificates


e-District
Services from Government of Kerala
National e-Governance Plan
Public services closer home

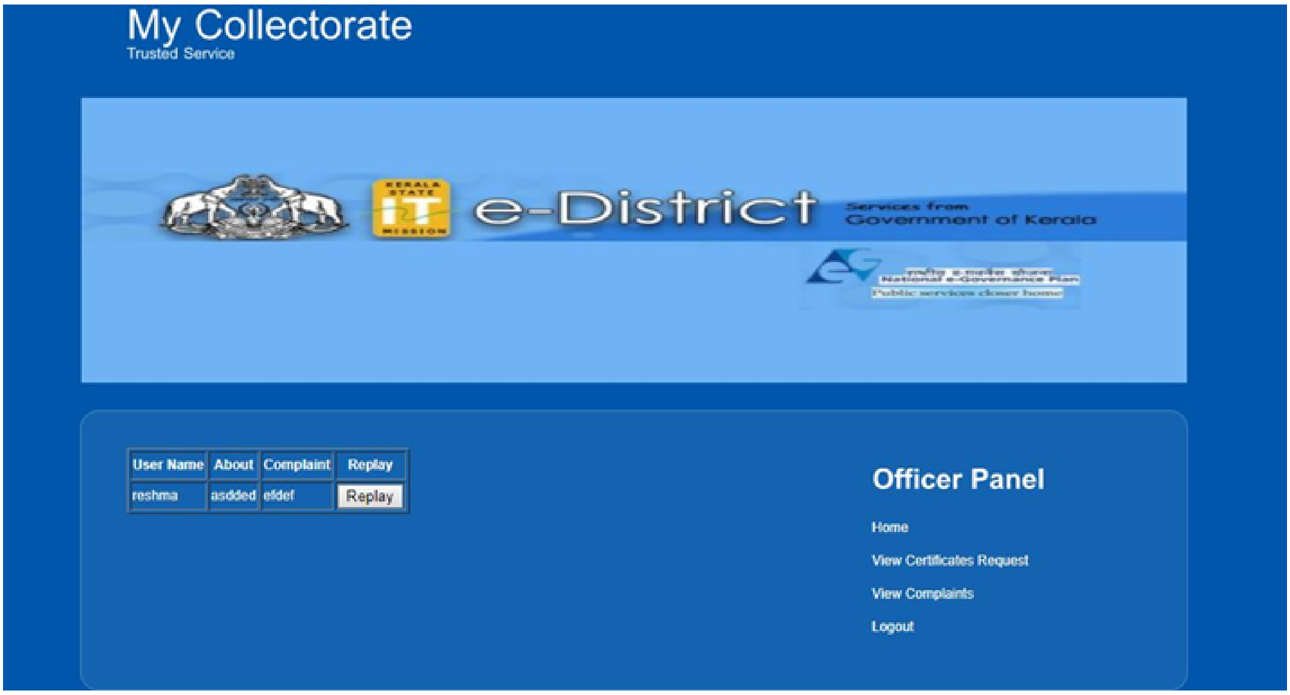
Available Certificates

1)Petrol Pumb Licence

2)Cracker Shop Licence

User Panel
Home
Post Complaint
View Feedback
View Offices
View Staffs
Cerificates
View Certificates available here
Logout

- 11.2.1.7 User complaints



My Collectorate
Trusted Service
e-District
Services from Government of Kerala
National e-Governance Plan
Public services closer home

User Name	About	Complaint	Replay
reshma	asdded	efdef	Replay

Officer Panel

Home
View Certificates Request
View Complaints
Logout

11.3 APPENDIX C

11.3.1 Acronyms

ASF-Apache Software Foundation
CSS-Cascading Style Sheets
HTML-Hyper Text Markup Language
HTTP-Hyper Text Transfer Protocol
IDE-Integrated Development Enivironment
JDK-Java Development Kit
JIT -Just In Time
JSP-Java Server Pages
JVM-Java Virtual Machine
OSNs-Online Social Networks
SGML-Standard Generalized Markup Language

11.3.2 Bibilography

References:
1-Pure JSP by James Goodwill
2-Softwareb Engineering 2005-by Roger Pressman
Websites:
http://www.google.co.in
http://www.wikipedia.org
http://www.seminarsonly.com
http://www.1000projects.org