COST ANALYSIS AND TENDERING OF IRRIGATION DEPARTMENT

Project Report Submitted by

PRINTO MATHEW

Register No: 220021087299

In partial fulfilment of the requirements for the award of the degree

Of

BACHELOR OF COMPUTER APPLICATION (BCA)

P.G.M COLLEGE

KANGAZHA

MAHATMA GANDHI UNIVERSITY

KOTTAYAM

2022 – 2025

DEPARTMENT OF COMPUTER APPLICATIONS

P.G.M COLLEGE

KANGAZHA

CERTIFICATE

Certified that this project report **"Cost Analysis And Tendering Of Irrigation Department"** *is the bonafide work of* **Mr. PRINTO MATHEW** *with Register Number:* **220021087299** *who carried out the work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.*

Head of the Department	**Project Co-ordinator**	**Internal Guide**
Mrs. Bhagya M.U	Mrs. Bhagya M.U	Mrs. Chithra B

Submitted for the viva-voce held on.....

External Examiner

DECLARATION

I here by declare that the project report "**Cost Analysis And Tendering Of Irrigation Department**" is a bonafide work done at PGM College Devagiri, Kangazha towards the partial fulfilment of the requirements for the award of the Degree of Bachelor of Computer Applications (BCA) from Mahatma Gandhi University, Kottayam, during the academic year 2022-2025.

Date :…….

Place: Kangazha

PRINTO MATHEW

Reg No:220021087299

ACKNOWLEDGEMENT

First and foremost I thank God Almighty for His blessings for this project. I take this opportunity to express my gratitude to all those who have helped me in completing the project successfully.
It has been said that gratitude is the memory of the heart.I wish to express my sincere gratitude to our Principal **Prof. JEEN ABHRAHAM** for providing good faculty for guidance.

I owe a great depth of gratitude towards our head of the department **Mrs. BHAGYA M.U** for helping us. I am thankful to our project guide **Mrs. CHITHRA B** for her proper guidance throughout the project. I also express my gratitude towards the supporting staffs for technical assistance.

I am indebted to my beloved teachers whose cooperation and suggestions throughout the project which helped me a lot. I also thank all my friends and classmates for their interest, dedication and encouragement shown towards the project.

I convey hearty thanks to parents for the moral support, suggestion and encouragement to make this venture a success.

PRINTO MATHEW

ABSTRACT

The **Cost Analysis and Tendering of Irrigation Department** is a web-based application designed to automate the process of cost estimation, tendering, and project management for construction works under the Irrigation Department. The current manual system, which relies on Microsoft Excel for cost estimation, is time-consuming and prone to errors, creating inefficiencies and security concerns. The proposed system addresses these issues by offering an integrated platform that streamlines the estimation process, enhances data security, and improves overall project management.

The system caters to multiple user roles, including the **Irrigation Department (Super User)**, **Senior Engineer**, **Junior Engineer**, **Overseer**, **Contractor**, **Guest**, **Irrigation Officer** and **Farmer**. Each user role has distinct functionalities, ensuring smooth interaction across the hierarchy of project management. Key features include automated cost estimation, tender management, real-time notifications, fund allocation, project tracking, and secure role-based access.

This solution provides several benefits, including improved accuracy in cost estimations, faster processing of tenders, and better fund management, all while maintaining data security. Additionally, the system enhances transparency by allowing guests to view project and tender details. By automating and securing key processes, the system will optimize the efficiency of the Irrigation Department, ensuring better resource management and timely completion of irrigation projects.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER -1

INTRODUCTION

1.1 PROJECT OVERVIEW

The "**Cost Analysis and Tendering of Irrigation Department**" project is a web-based application designed to automate the cost estimation and tendering processes for irrigation department construction works. It addresses the limitations of the existing system, where cost estimation is performed manually using Excel, resulting in inefficiency and security concerns. The proposed system enhances efficiency and security by enabling the irrigation department to announce project plans and estimates online, allowing registered contractors to quote tenders while also making project details publicly viewable. The system includes various modules, each with specific functionalities: the irrigation department manages the entire system, senior engineers oversee tender approvals and project planning, junior engineers handle cost estimation and reporting, overseers manage on-site activities, contractors create and modify tenders, and guests can view project and tender details. By streamlining workflows and improving transparency, the system significantly reduces time consumption and enhances data security.

1.2 PROJECT SPECIFICATION

The project "**Cost Analysis and Tendering of Irrigation Department**" aims to develop a web-based application that automates and streamlines cost estimation and tendering processes. It consists of several modules with specific functionalities, such as the Irrigation Department (admin) managing user roles, approving project plans, and allocating funds; Senior Engineers overseeing tenders, approving contractor applications, and managing stage-wise payments; Junior Engineers preparing cost estimations, submitting plans, and reporting progress; Overseers handling on-site execution and reporting changes; Contractors quoting tenders and modifying them based on notifications; Farmer and Guests can submit their complaints and Irrigation Officer can view verify the complaints and take necessary actions.

The system focuses on performance, ensuring it handles multiple users efficiently with real-time notifications and role-specific dashboards. It incorporates robust security features, including role-based access control, data encryption, and audit trails, while maintaining scalability to accommodate increasing data and users. The technical stack includes a frontend built with HTML5, CSS3, and a backend developed using Python Django for business logic, and databases such as MySQL or MongoDB for storing project and user data.

CHAPTER-2

SYSTEM STUDY

2.1 EXISTING SYSTEM

The existing system for cost estimation and tendering in the irrigation department relies heavily on manual processes using Microsoft Excel, where separate worksheets are created for each project. This method is time-consuming, prone to errors, and lacks robust security measures for sensitive data. Additionally, there is no centralized platform for contractors or the public to access project details or submit tenders efficiently, leading to a lack of transparency and coordination. These limitations highlight the need for a streamlined and automated solution.

2.2 DRAWBACKS OF EXISTING SYSTEM

- Time-consuming processes
- Error-prone calculations
- No centralized platform
- Limited accessibility for contractors and stakeholders
- Difficulty in managing multiple projects
- No real-time notifications or updates

2.3 PROPOSED SYSTEM

The proposed system is a web-based application designed to automate and streamline cost estimation and tendering processes in the irrigation department. It provides a centralized platform where the department can manage projects, issue tenders, and allocate funds securely and efficiently. Registered contractors can easily submit and track tenders, while public users can view project and tender details to ensure transparency. The system incorporates role-based access control, real-time notifications, and secure data management to address the inefficiencies and security risks of the existing system. By replacing manual processes with automated workflows, the proposed system enhances accuracy, efficiency, and scalability.

2.4 ADVANTAGES OF PROPOSED SYSTEM

- Centralized platform for efficient project and tender management
- Real-time notifications for updates and approvals
- Reduced errors through automated calculations and workflows
- Faster decision-making with streamlined processes
- Can manage multiple projects

CHAPTER-3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For the feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are**:**

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- BEHAVIOURAL / OPERATIONAL FEASIBILITY

3.1.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development

3.1.2 TECHNICAL FEASIBILITY

Technical Feasibility deals with the hardware as well as software requirements. Technology is not a constraint to type system development. We have to find out whether the necessary technology, the proposed equipment has the capacity to hold the data, which is used in the project, should be checked to carry out this technical feasibility.

The technical feasibility issues usually raised during the feasibility stage of investigation includes these:

- This web application can run in any type of Operating System.

- The hardware required is the minimum hardware for running the applications.
- The application can be expanded easily to include more functionalities.

3.1.3 BEHAVIOURAL/OPERATIONAL FEASIBILITY

This analysis involves how it will work when the system runs in the environment in which it is implemented. The new proposed system is made with user friendly interfaces that the user can easily understand and use. The proposed system is very much useful to the users and there for it will accept broad audience from the department.

3.2 SYSTEM SPECIFICATION

The **Cost Analysis and Tendering of Irrigation Department** project is a web-based application designed to automate the cost estimation and tendering processes for the irrigation department. This system replaces the existing manual method of using Microsoft Excel for cost estimation and tender management, which is time-consuming, error-prone, and lacks security. The proposed solution centralizes project management, provides real-time updates, enhances data security, and improves the overall efficiency of the tendering and cost estimation processes. It aims to ensure transparency, streamline operations, and reduce errors associated with manual work.It will feature a intuitive and user-friendly interface accessible to both user and services provider. Some of its features are given below:

- Centralized platform to announce new projects, issue tenders, and allow registered contractors to submit quotes for approval.
- Automatic notifications and alerts for users on project updates, tender approvals, fund requests, and changes in project plans.
- Guests can view project details, tender information, and gallery showcasing ongoing and completed projects.
- Contractors can track the status of their submitted tenders and receive notifications regarding approval, rejection, or changes.

3.2.1 HARDWARE SPECIFICATION

The section of the hardware is very important in the existence and proper working of any application as the proposed system is an online application it can run in any system. The basic hardware requirements are:

- **Processor** : Intel Pentium Dual Core or equivalent

- **RAM** : 1 GB
- **Hard Disk Drive** : 40 GB
- **Key Board** : Normal or multimedia keyboard
- **Mouse** : Logitech Serial Mouse or Compatible mouse

3.2.2 SOFTWARE SPECIFICATION

The Software Specification details the software components and tools required to develop, deploy, and maintain the **Cost Analysis And Tendering of Irrigation Department** web application. The basic software requirements are:

- **Front-end :** HTML, CSS
- **Back-end :** SQLite, Python
- **Framework :** Django
- **Client on PC :** Any Operating System

3.3 SOFTWARE DESCRIPTION

3.3.1 FRONTEND DESCRIPTION & OVERVIEW OF THE LANGUAGE USED

HYPER TEXT MARKUP LANGUAGE (HTML)

To public information for global distribution, one needs a universally understood language, a kind of publishing mother tongue that all computers may potentially understood. The publishing language use by the World Wide Web is HTML. HTML is short for hypertext markup language. It defines the structure and layout of a web document by using a variety of tags and attributes.HTML pages contains a set of markup symbols or cods indented for display on a browser. The markup tells the browser how to display a webpage's words and images for the user. Each individual markup code is referred as an element or tag. Some elements comes in pairs (container tags) that indicate when some display effect is to begin and when it is to end. Some tags enable the document to display formatted text, color, a variety of fonts, graphic images, special effects, hypertext jumps to other.

Internet locations and information forms, HTML 5 is advancement over the standard specification of HTML. It extends HTML with mechanism for style sheets, scripting, frames, embedding objects, improved up port for right to left and mixed direction text, richer tables and enhancements to forms offering improved accessibility for people with disabilities.HTML 5 also takes great strides towards the

internationalization of documents, with the goal of making the Web truly World Wide. Early versions of HTML were defined with loose syntactical rules, which helped its adoption by those unfamiliar with web publishing. Overtime, the trend has been to create increasingly strict language syntax.HTML 5 is the current version of HTML specification. This project makes use of HTML 5 specification.

CSS

CSS, or Cascading Style Sheets, is a stylesheet language used for describing the presentation and formatting of a document written in HTML or XML. It enables web developers to control the layout, appearance, and style of multiple web pages simultaneously. CSS separates the structure of a document from its presentation, allowing for greater flexibility and consistency in the design of websites.

By using CSS, web developers can create visually appealing and consistent designs across various web pages while maintaining a clear separation between content and presentation. This separation enhances maintainability and makes it easier to update the look and feel of a website without altering its underlying structure.

VISUAL STUDIO CODE

Visual Studio Code (VS Code) is a free, open-source code editor developed by Microsoft, widely used for software and web development. It supports multiple programming languages and provides features like syntax highlighting, intelligent code completion, debugging, and built-in Git integration. One of its biggest strengths is its vast marketplace of extensions, allowing users to customize their development environment with tools for Python, JavaScript, Docker, and more. VS Code is lightweight yet powerful, offering excellent performance and a smooth user experience. It is available on Windows, macOS, and Linux, making it accessible to developers across different platforms. With its user-friendly interface and extensive features, VS Code has become one of the most popular code editors among developers worldwide.

3.3.1 BACKEND DESCRIPTION & OVERVIEW OF THE DATABASE USED

PYTHON

As a backend language, Python is highly favored for its simplicity, readability, and flexibility. Its robust frameworks like Django and Flask streamline the development of web applications, APIs, and microservices, allowing developers to quickly build and maintain backend systems. Python's scalability and support for asynchronous programming make it suitable for handling high-traffic

applications and real-time tasks. Additionally, its vast ecosystem of libraries for database integration, security, and task management ensures Python is a powerful and efficient tool for backend development.

DJANGO

Django is a high-level, open-source web framework for Python that encourages rapid development and clean, pragmatic design. It follows the "batteries-included" philosophy, providing built-in features like authentication, an admin panel, URL routing, and database management, which speeds up development. Django promotes the use of the Model-View-Template (MVT) architecture, making it easy to organize code and separate concerns in web applications. It also emphasizes security by providing features like protection against SQL injection, cross-site scripting (XSS), and clickjacking. With its scalability, security features, and large community, Django is a popular choice for building robust web applications and APIs.

MYSQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Wideniu's daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

FEATURES :

- Public Access to Project Details
- Fund Allocation and Tracking
- Mobile and Web Accessibility
- Data Security
- Real-Time Notifications

CHAPTER- 4

SYSTEM DESIGN

4.1 INTRODUCTION

4.1.1 DEFINITION

The most creative and challenging face of the system development is System Design. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Design goes through the logical and physical stages of development. In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfil. The first step is to determine how the output is to be produced and in what format. Second, input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing. Design of a system can be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Thus, system design is a solution to "how to" approach to the creation of a new system. Thus, important phase provides the understanding and the procedural details necessary for implementing the system recommended in the feasibility study. The design step provides a data design, architectural design, and a procedural design.

4.1.2 OUTPUT DESIGN

In the output design, the emphasis is on producing a hard copy of the information requested or displaying the output on the CRT screen in a predetermined format. Two of the most output media today are printers and the screen. Most users now access their reports from a hard copy or screen display. Computer's output is the most important and direct source of information to the user, efficient, logical, output design should improve the systems relations with the user and help in decision-making. As the outputs are the most important source of information to the user, better design should improve the system's relation and also should help in decision-making. The output device's capability, print capability, print capability, response time requirements etc. should also be considered form design elaborates the way output is presented and layout available for capturing information. It's very helpful to produce the clear, accurate and speedy information for end users.

4.1.3 INPUT DESIGN

In the input design, user-oriented inputs are converted into a computer-based system format. It also includes determining the record media, method of input, speed of capture and entry on to the screen. Data entry accepts commands and data through a keyboard. The major approach to input design is the menu and the prompt design. In each alternative, the user's options are predefined. The data flow

diagram indicates logical data flow, data stores, source and destination. Input data are collected and organized into a group of similar data. Once identified input media are selected for processing.

In this online application, importance is given to develop Graphical User Interface (GUI), which is an important factor in developing efficient and user-friendly application. For inputting user data, attractive forms are designed. User can also select desired options from the menu, which provides all possible facilities.

Also, the important input format is designed in such a way that accidental errors are avoided. The user has to input only just the minimum data required, which also helps in avoiding the errors that the users may make. Accurate designing of the input format is very important in developing efficient application. The goal or input design is to make entry as easy, logical and free from errors.

4.2 MODULE DESCRIPTION

COST ANALYSIS AND TENDERING OF IRRIGATION DEPARTMENT includes following main modules:

- **ADMIN**
- **SENIOR ENGINEER**
- **JUNIOR ENGINEER**
- **OVERSEER**
- **CONTRACTOR**
- **GUEST**
- **FARMER**
- **IRRIGATION OFFICER**

4.2.1 ADMIN

This is the super user of the whole system. Admin can manage the whole system. He/she is included in the top level management.

This module's main functionalities include:

- **Login** – Secure access to the admin dashboard.
- **Manage Users** – Admin manages the users such as Senior Engineer , Junior Engineer, Overseer and Contractor.
    - The admin can Add, update, or remove users accounts.

  - Assign Task to the user and view its completion.
- **Accept/Reject Project Plan** – Review and make decisions on submitted project plans.
- **Fund Allocation** – Allocate necessary funds for approved projects.
- **Schedule Meetings** – Set up official meetings with engineers or contractors.

4.2.2 SENIOR ENGINEER

Senior engineer comes under the supreme power. The senior engineer is included in the middle level management.

This module's main functionality includes:

- **Login** – Authenticate and access senior engineer portal.
- **Tender Issue** – Create and publish new tenders for projects.
- **Accept/Reject Application from Contractor** – Evaluate and manage contractor bids.
- **Project Planning** – Draft detailed project plans and timelines.
- **Tender Approval** – Finalize the tender and assign it to the selected contractor.
- **Stage-wise Payment** – Release payments based on project milestones.
- **Fund Request to Department** – Submit fund requests to the irrigation department.
- **Approval Changes and Send Notification** – Approve mid-project changes and notify stakeholders.
- **Approve Project Changes and Allocate Fund** – Validate modifications and provide additional funding if required.

4.2.3 JUNIOR ENGINEER

Junior managers are the first line managers who comes under the senior manager. They have less power than the senior managers.

This module's main functionality includes:

- **Login** – Secure access to the junior engineer portal.
- **Cost Estimation** – Calculate expenses for materials and labor.
- **Fund Request to Senior Engineer** – Request additional funds for project execution.
- **View Changes and Reporting** – Monitor site updates and generate reports.
- **Change Project Plan and Report/Submit to Senior Engineer** – Propose changes and submit for senior approval.

4.2.4 OVERSEER

Overseers are the bottom level management. Their main functionalities include the overall management of the sites included in the projects.

Other functions include:

- **Login** – Access overseer tools and site data.
- **Site Stage Approval** – Confirm completion of each construction stage.
- **Fund Requesting to Junior Engineer** – Ask for operational funds to continue work.
- **Change Detection and Reporting** – Observe and report any deviations or issues.

4.2.5 CONTRACTOR

They also comes under the low level management. They create the tender for each project.

This module's main functionalities include:

- **Login** – Gain access to the contractor interface.
- **Quote Tender** – Submit a bid for available projects.
- **View Status of Tender** – Track progress or decisions related to submitted tenders.
- **Make Changes** – Revise tenders as requested by the senior engineer.

4.2.6 GUEST

They are the unauthenticated users of the system. They can submit complaint.

This module's main functionalities include:

- **Register Complaints** – Submit grievances related to irrigation issues.
- **View New Project Details** – Browse newly listed irrigation projects.
- **View New Tender Details** – See open tenders and their information.

4.2.7 FARMER

Farmers are key stakeholders who can interact with the system without admin approval and can register their complaints.

This module's main functionalities include:

- **Login** – Access personalized account features.
- **Register Complaints** – Raise irrigation-related concerns to officials.
- **View Responses from Irrigation Officer** – Read official replies to submitted complaints.

- **Update Profile Details** – Edit personal and land ownership information.

4.2.8 IRRIGATION OFFICER

The Irrigation Officer plays a crucial role in handling farmer-related complaints and verifying land details.

This module's main Functionalities includes:

- **Respond to Complaints** – Address farmer-submitted problems or concerns.
- **Monitor Irrigation Issues** – Track ongoing issues and coordinate resolution.
- **Provide Recommendations** – Suggest improvements or interventions to higher authorities.

4.3 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change data throughout a system. It's a structured analysis and design tool that can be used for flowcharting in place of or in association with information. Oriented and process-oriented system flowcharts. When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply physical implementations. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

The purpose of the design is to create architecture for the evolving implementation and to establish the common tactical policies that must be used by desperate elements of the system. We begin the design process as soon as we have some reasonably completed model of the behaviour of the system. It is important to avoid premature designs, wherein develop designs before analysis reaches closer. It is important to avoid delayed designing where in the organization crashes while trying to complete an unachievable analysis model.

Throughout the project, the context flow diagrams, data flow diagrams and flow charts have been extensively used to achieve the successful design of the system. In our opinion, "efficient design of the data flow and context flow diagrams helps to design the system successfully without much major flaws within the scheduled time". This is the most complicated part in a project. In the designing process, our project took more than the activities in the software life cycle. If we design a system efficiently with all the future enhancements, the project will never become junk and it will be operational.

The data flow diagrams were first developed by Larry Constantine as way for expressing system requirements in graphical form. A data flow diagram also known as “bubble chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. It functionally decomposes the requirement specification down to the lowest level. DFD depicts the information flow, the transformation flow and the transformations that are applied as data move from input to output. Thus DFD describes what data flows rather than how they are processed.

Data Flow Diagram is quite effective, especially when the required design is unclear and the user and analyst need a notational language for communication. It is one of the most important tools used during system analysis. It is used to model the system components such as the system process, the data used by the process, any external entities that interact with the system and information flows in the system. Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

Steps to Construct Data Flow Diagrams:-

Four steps are commonly used to construct a DFD:

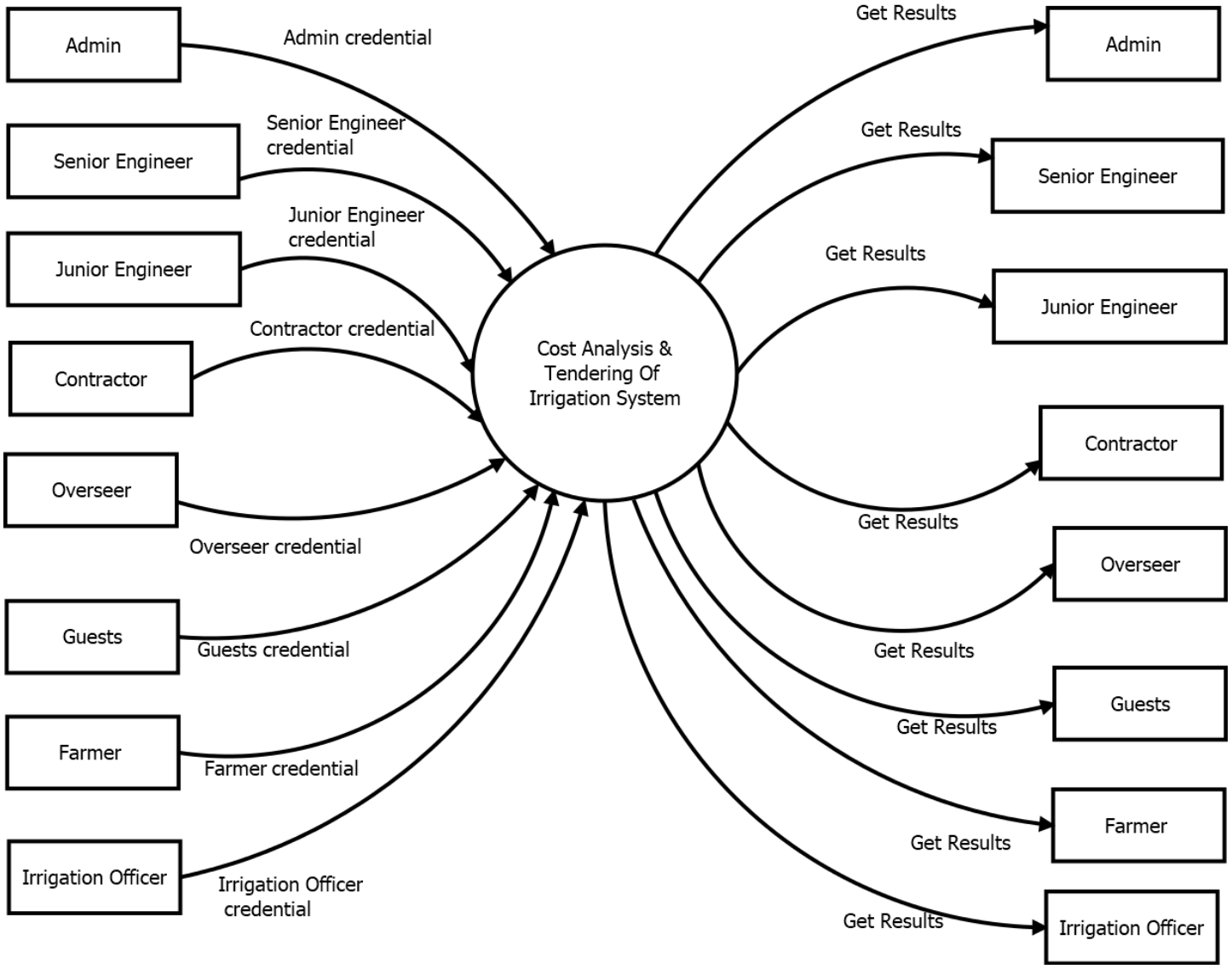
- Process should be named and numbered for easy reference. Each name should be representative of the process.
- The destination of flow is from top to bottom and from left to right.
- When a process is exploded in to lower level details they are numbered.
- The names of data stores, sources and destinations are written in capital letters.

Rules for constructing a Data Flow Diagram:-

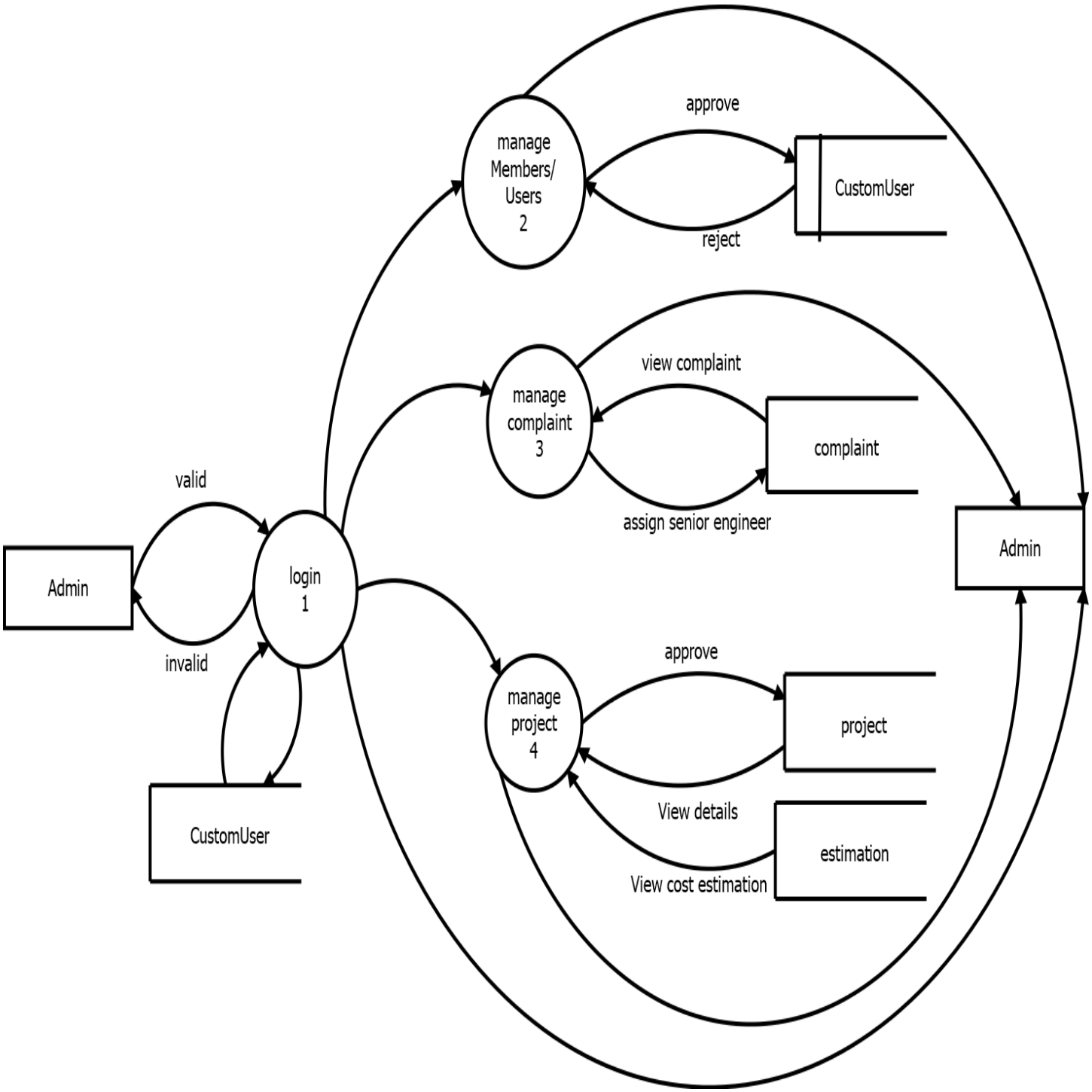
- Arrows should not cross each other.
- Squares, circles and files must bear names.
- Decomposed data flow squares and circles can have same names.
- Draw all data flow around the outside of the diagram.

Data Flow Diagram Symbols:

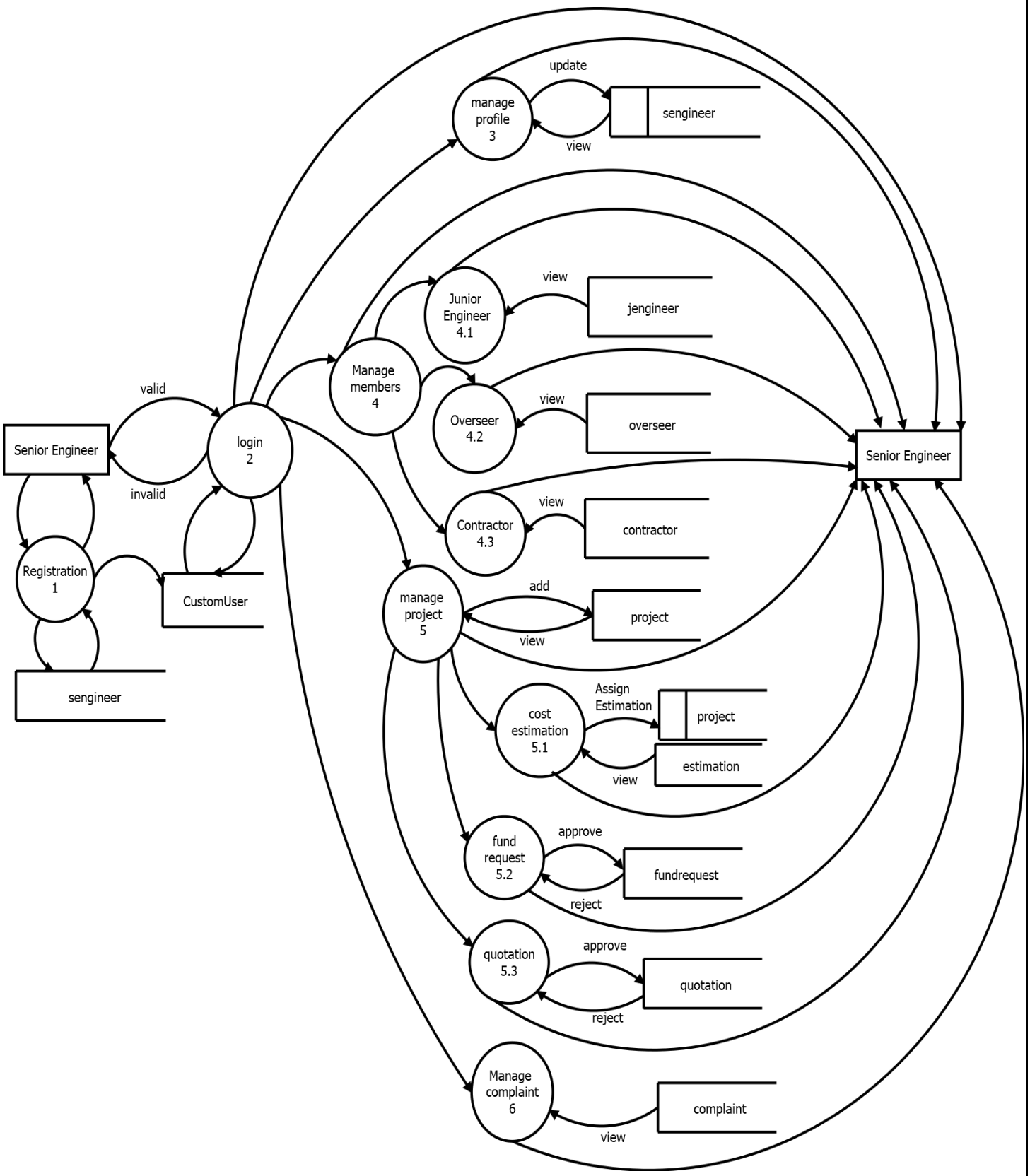
LEVEL – 0 DFD for IRRIGATION DEPARTMENT



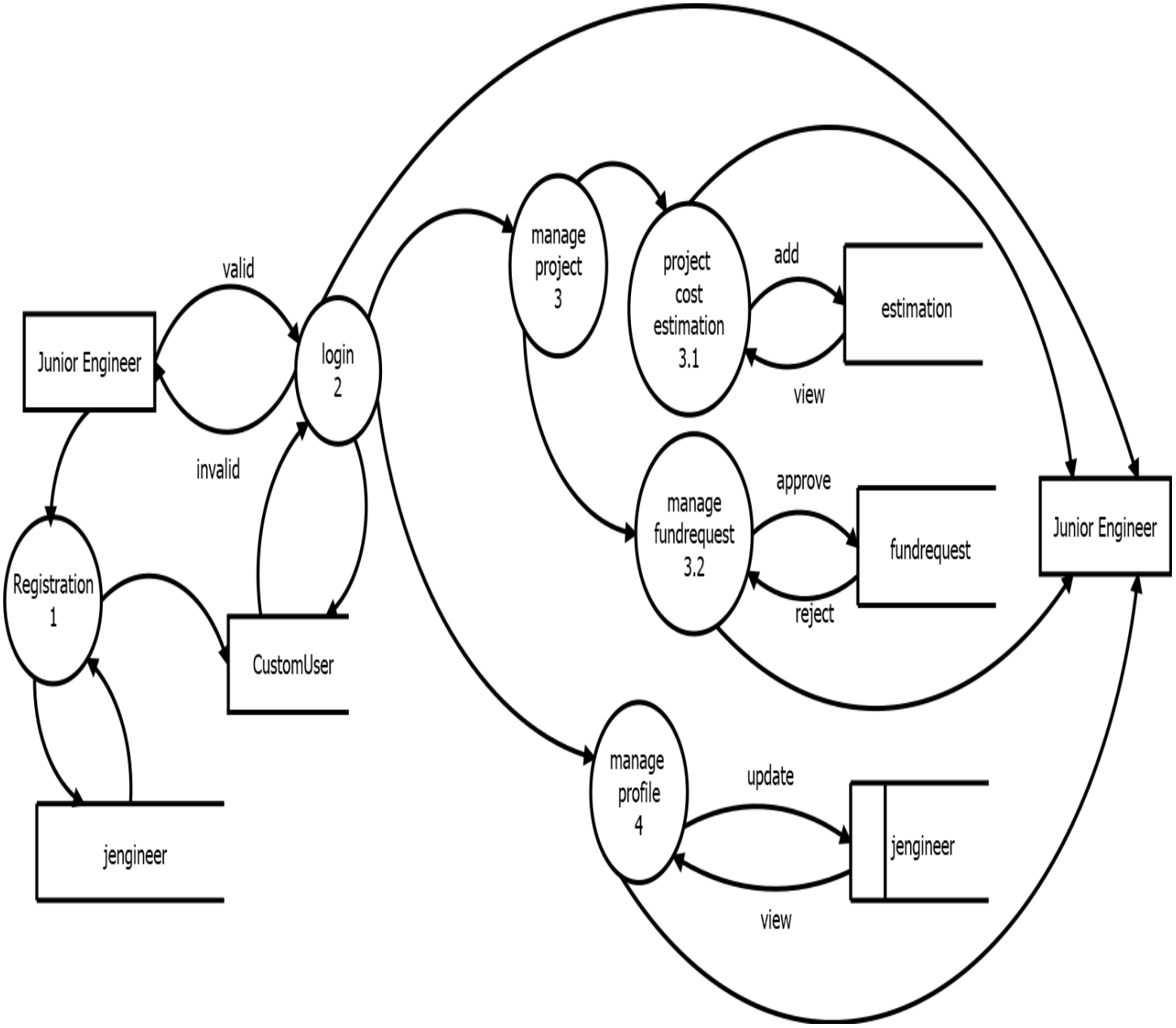
LEVEL – 1 DFD for ADMIN:



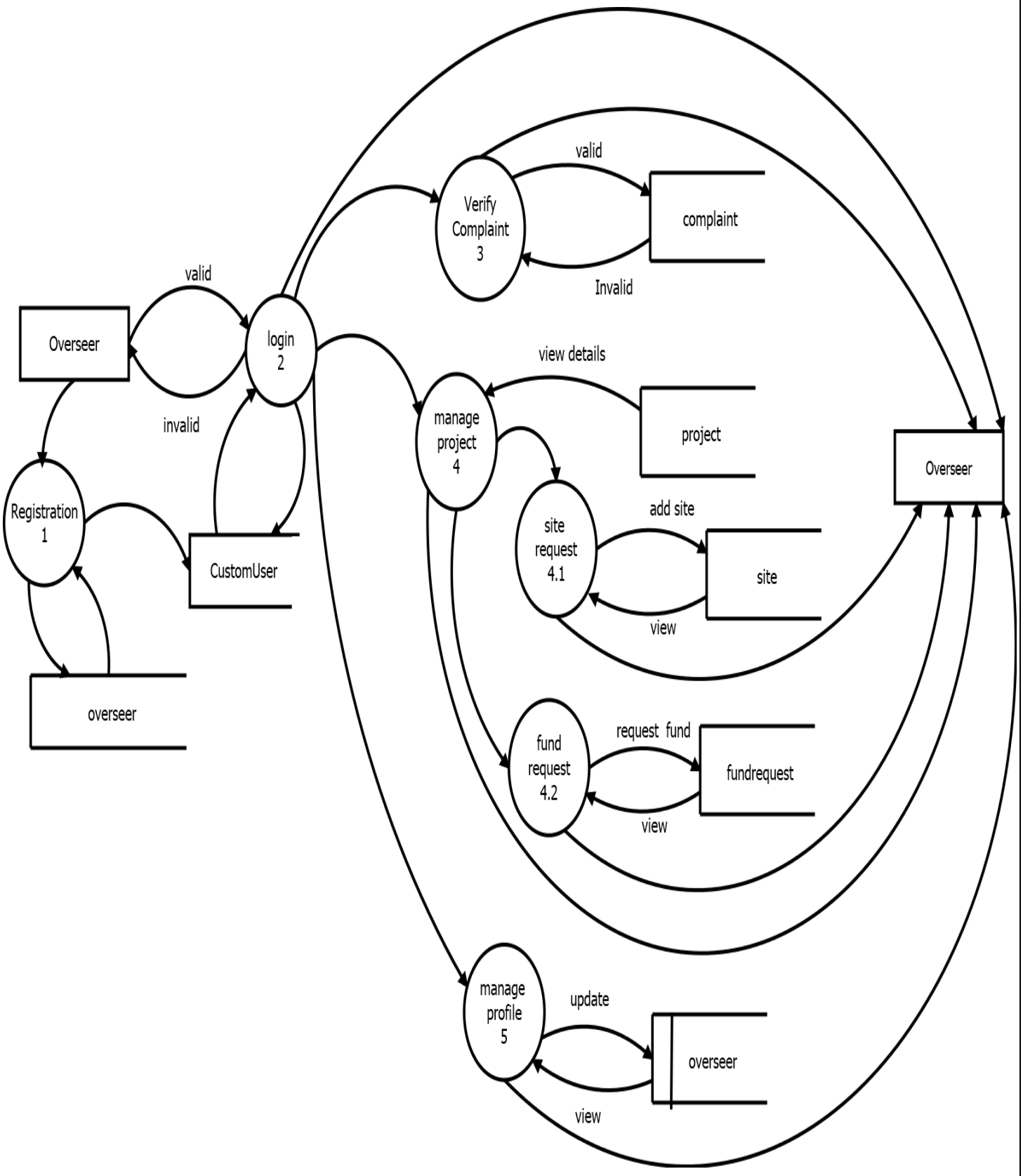
LEVEL – 1 DFD for SENIOR ENGINEER:



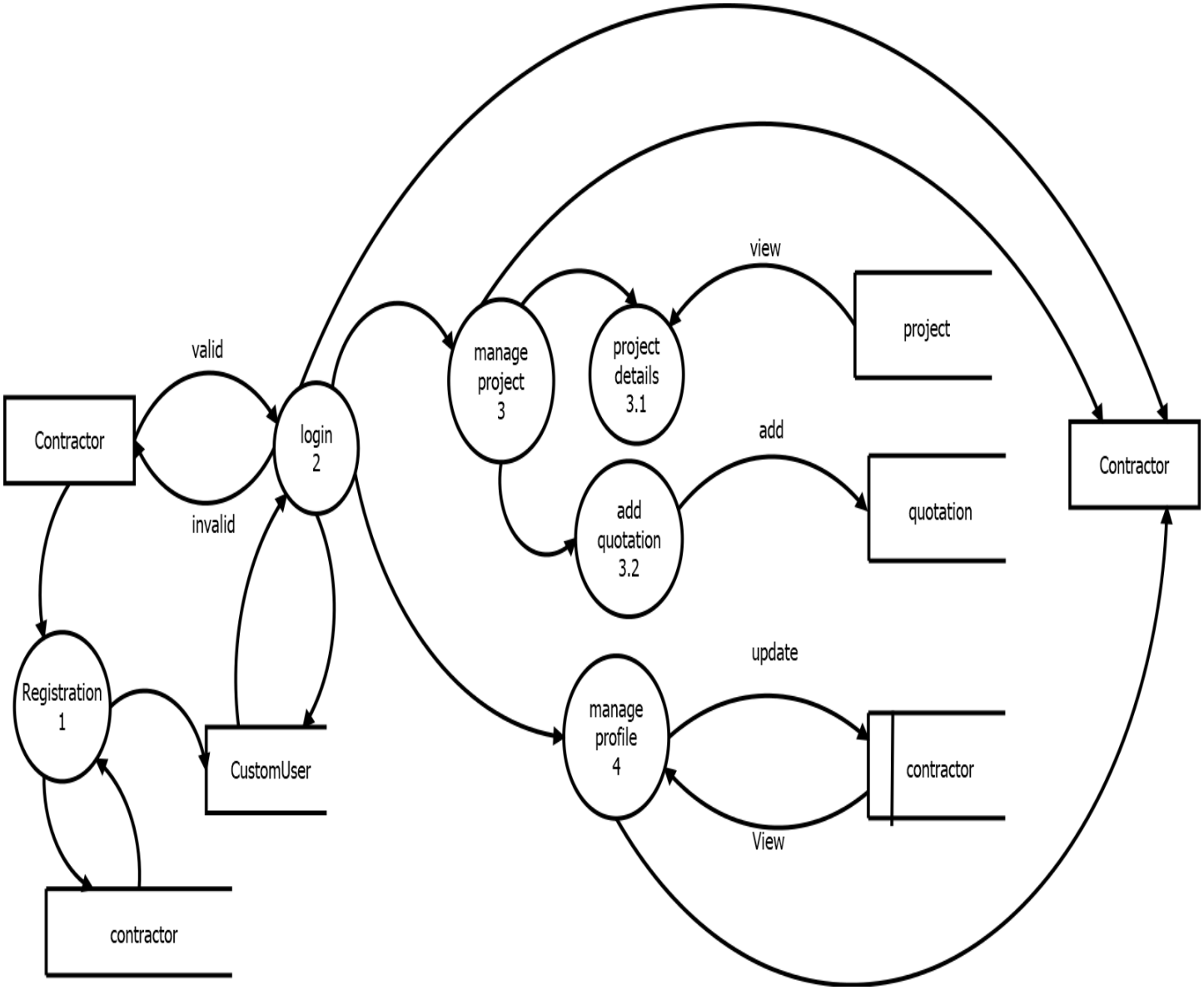
LEVEL – 1 DFD for JUNIOR ENGINEER:



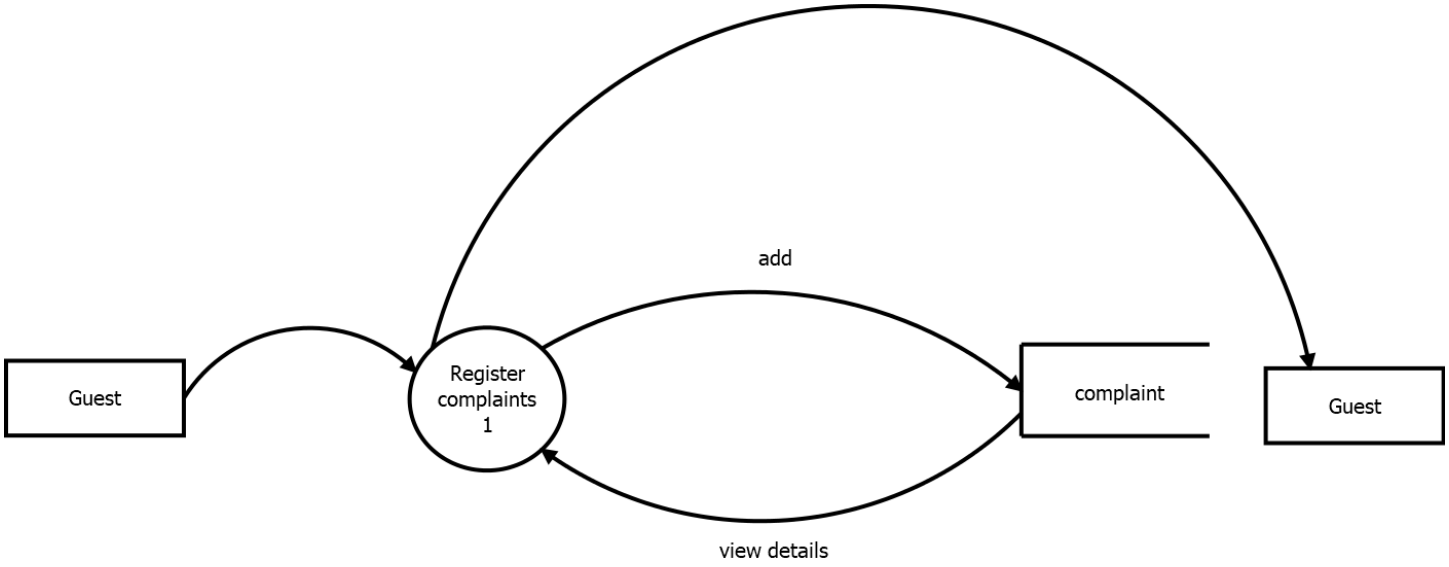
LEVEL – 1 DFD for OVERSEER:



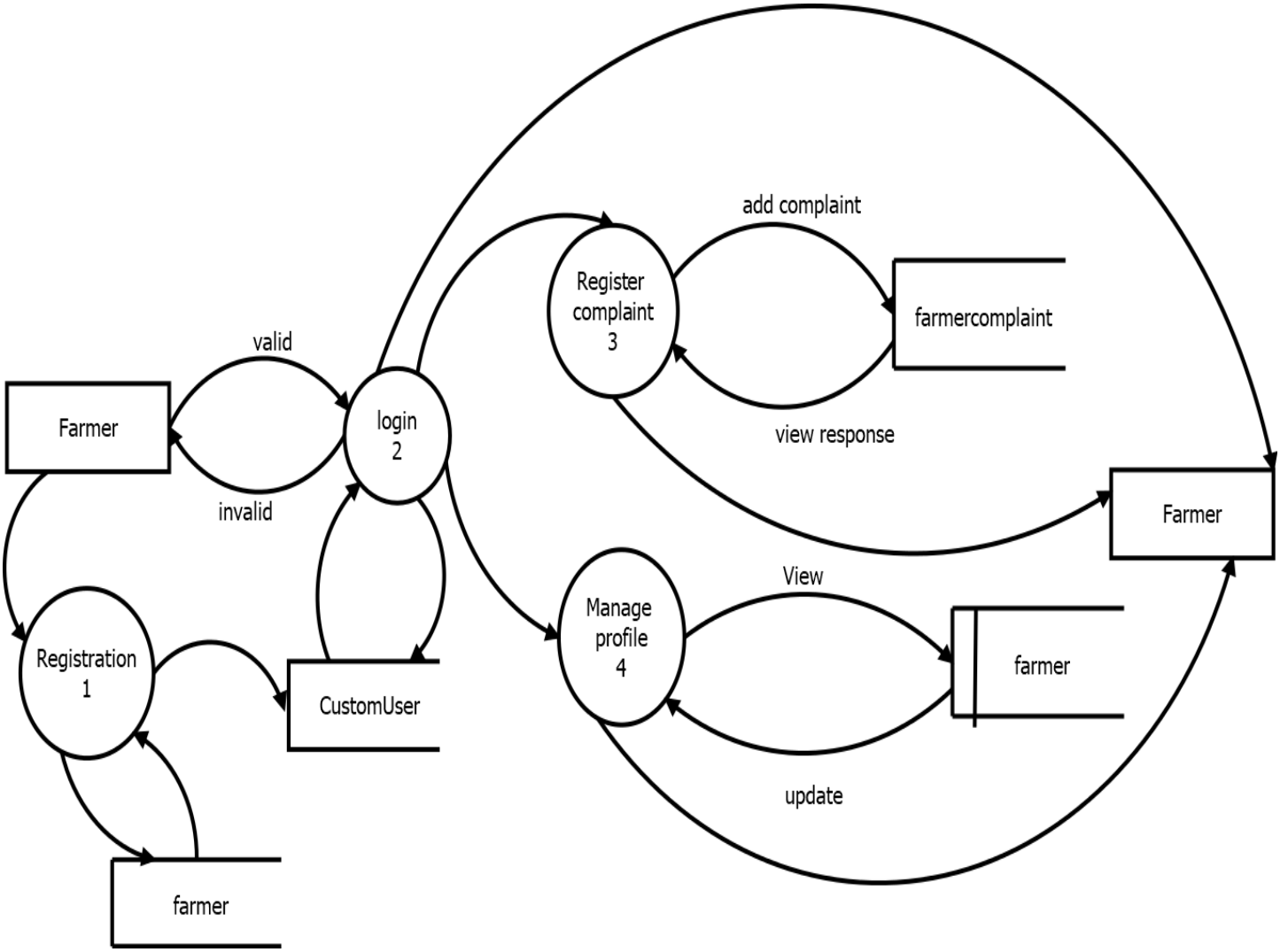
LEVEL – 1 DFD for CONTRACTOR:



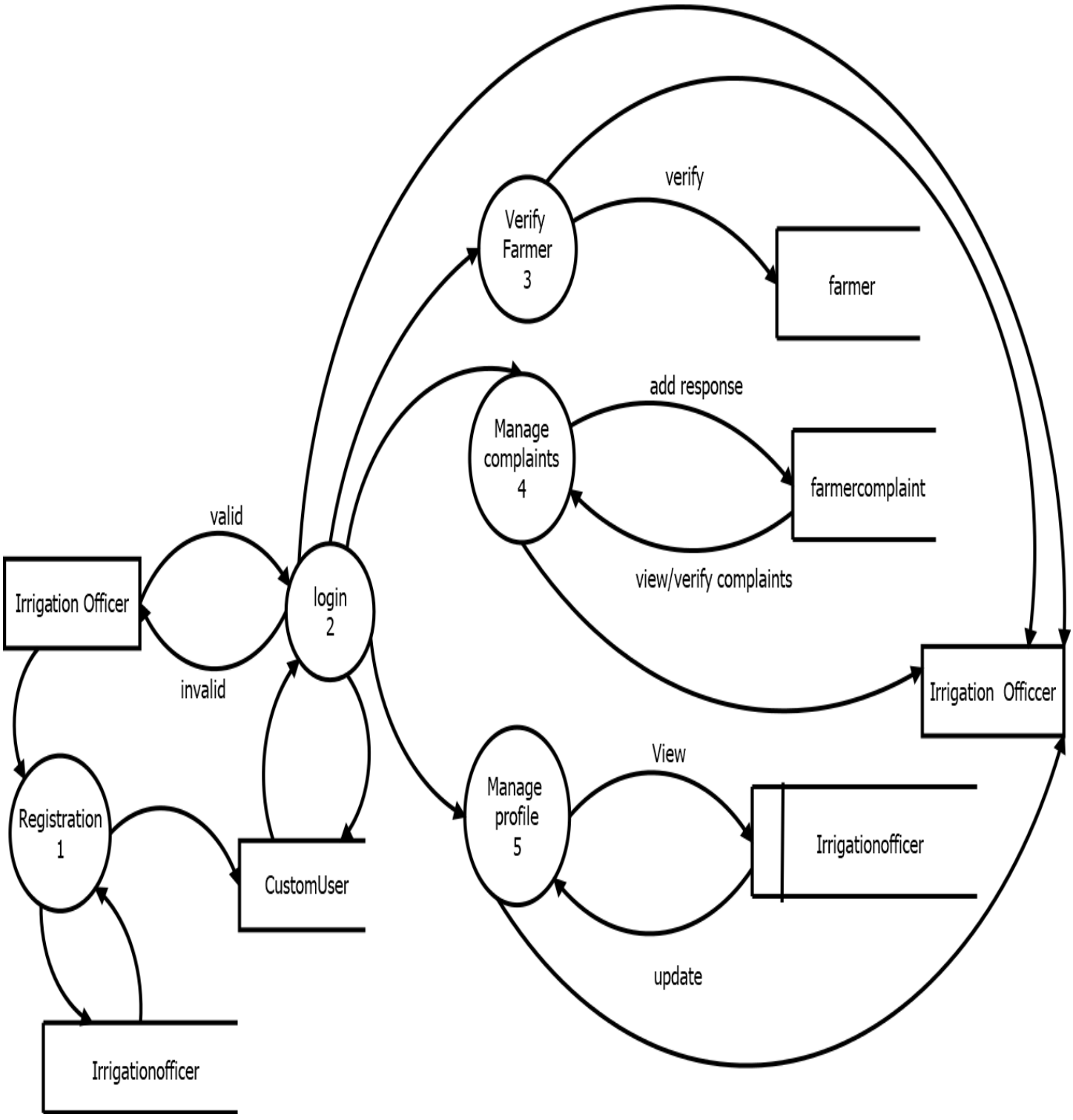
LEVEL – 1 DFD for GUEST:



LEVEL – 1 DFD for FARMER:

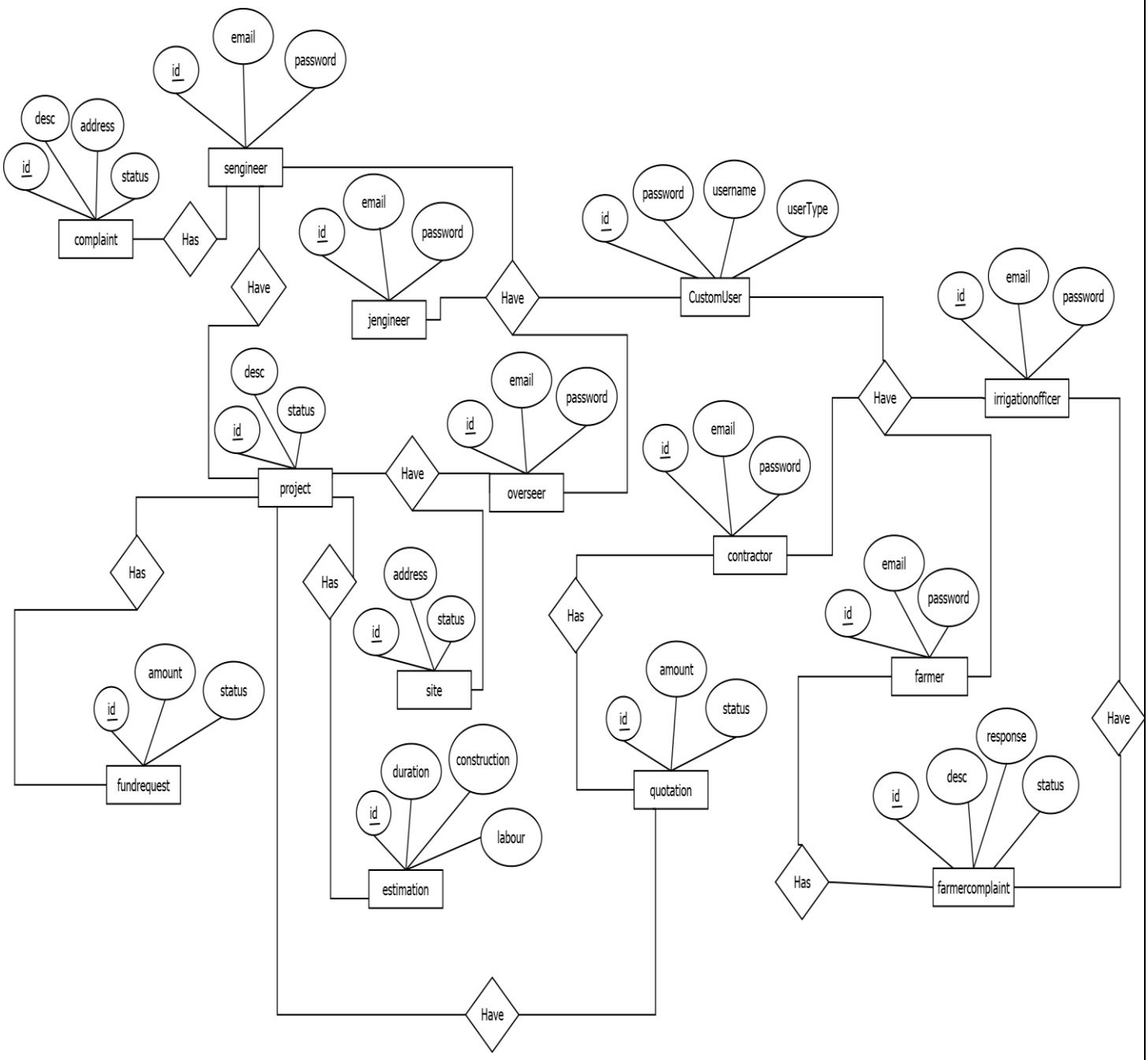


LEVEL – 1 DFD for IRRIGATION OFFICER:



4.4 ERD -ENTITY RELATIONSHIP DIAGRAM

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology(IT) system. An ERD uses data modeling techniques that can help define business processes and serve as the foundation for a relational database.



4.5 DATABASE DESIGN

Database

A database is a collection of data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers and addresses of people you know. You may have recorded this data on an indexed address book, or you may have stored it on a hard drive using a personal computer and software such as Microsoft Access or Excel. This is a collection of related data with an implicit meaning and hence is a database. A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the mini world or the Universe of Discourse (UOD). Changes to the mini world are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database. A database is designed, built and populated with data for specific purpose. It has intended group of users and some preconceived applications in which these users are interested.

Data Base Management System

A database management system (DBMS) is a collection of programs that enable users to create and maintain a database. The DBMS is hence general-purpose software system that facilitates processes of defining, constructing, manipulating, and sharing databases among various users and applications. Defining a database involves specifying the data types, structure and constraints for the data to be stored in the database. Constructing the database is the process of storing the data itself on some storage medium. Manipulating a database includes such functions as querying the database to retrieve specific data, updating the database to reflect changes in the mini world, and generating reports from the data. Sharing a database allows multiple users and programs to access the database concurrently. This section describes the structure of data and how the data are stored, processed and organized. RDBMSs have become a predominant choice for the storage of information in new databases. Here used the EAV model in the data base design. This model is an optimal approach to data modelling for a problem domain. In an EAV data model, each attribute-value pair is a fact describing an entity, and a row in an EAV table stores a single fact. EAV model helps to create “long and skinny" databases. By "long" it refers to the number of rows, by "skinny" it refers to the fewer columns.

DATA NORMALIZATION

Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like insertion, update and deletion anomalies. It is two-step process that put data into tabular form by removing duplicate data from the relation tables.
Normalization used for mainly two purpose:

- Eliminating redundant data.
- Ensuring data dependencies make sense, i.e., data is logically stored.

NORMALIZATION RULE

Normalization rules are divided into following normal forms

- First normal form
- Second normal form
- Third normal form
- Boyce codd normal form

FIRST NORMAL FORM (1NF)

A relational R is in 1NF if all underlying domains contains atomic values.
First Normal Form put two restrictions,

1) Fields of an n-set should simple, atomic values.

2) N-set should have no repeating groups.

SECOND NORMAL FORM (2NF)

A relational R is in 2NF if and only if it is1NF and every non-key attribute is fully functionally depending on the primary key. An attribute is a non-key if it does not particularly in the primary key.

THIRD NORMAL FORM (3NF)

A relation R is in 3NF if and only if it is 2NF and every non-key. Attribute is non-transitively dependent on the primary key.

BOYCE CODD NORMAL FORM (BCNF)

A table is in Boyce-Codd Normal Form (BCNF) if:

1. It is in the Third Normal Form (3NF).
2. For every non-trivial functional dependency ($X \rightarrow Y$), X is a superkey.

TABLE STRUCTURE

The main tables of **COST ANALYSIS AND TENDERING OF IRRIGATION DEPARTMENT** includes the following**:**

Table No: 1

Table Name: CustomUser

Primary Key: id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique user ID
password	varchar		128	User's encrypted password
last_login	datetime			Last login time
is_superuser	bool			Superuser status
username	varchar		150	Unique username
first_name	varchar		150	User's first name
last_name	varchar		150	User's last name
email	varchar		254	User's email
is_staff	bool			Staff member status
is_active	bool			Account active status
date_joined	datetime			User registration date
userType	varchar		20	Defines User type

Table No: 2

Table Name: Sengineer

Primary Key: id

Foreign Key: user_id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique user ID
name	varchar		50	Full name of the user
phone	varchar		20	Contact number
email	varchar		254	User's email address
address	varchar		50	Residential address
qualification	varchar		50	User's education
user_id	Big int	Foreign Key	12	References another user

Table No: 3

Table Name: jengineer

Primary Key: id

Foreign Key: user_id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique user ID
name	varchar		50	Full name of the user
phone	varchar		20	Contact number
email	varchar		254	User's email address
address	varchar		50	Residential address
qualification	varchar		50	User's education
user_id	big int	Foreign Key	12	References another user

Table No: 4

Table Name: Overseer

Primary Key: id

Foregin Key: user_id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique user ID
name	varchar		50	Full name of the user
phone	varchar		20	Contact number
email	varchar		254	User's email address
address	varchar		50	Residential address
qualification	varchar		50	User's education
user_id	big int	Foreign Key	12	References another user

Table No: 5

Table Name: contractor

Primary Key : id

Foregin Key: user_id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique user ID
name	varchar		50	Full name of the contractor
phone	varchar		20	Contact number
email	varchar		254	Contractor's email address
address	varchar		50	Residential address
license	varchar		50	Registered License of contractor
user_id	big int	Foreign Key	12	References another user

Table No: 6

Table Name: project

Primary Key: id

Foregin Key: junior_id, overseer_id , senior_id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique project ID
name	varchar		50	Project name
desc	varchar		20	Short project description
place	varchar		254	Project location
landmark	varchar		50	Nearby landmark
status	varchar		50	Current project status
junior_id	big int	Foreign Key	12	References to junior engineer
overseer_id	big_int	Foreign Key	12	References overseer
senior_id	big_int	Foreign Key	12	References senior engineer

Table No: 7

Table Name: site

Primary Key: id

Foregin Key: project_id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique site ID
name	varchar		50	Site/place name
address	varchar		100	Site location
other	varchar		100	Additional details
project_id	big int	Foreign Key	12	References project ID

Table No: 8

Table Name: estimation

Primary Key: id

Foregin Key: project_id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique estimation ID
duration	int			Estimated project duration (days)
purcahse	big int		20	Estimated material cost
labour	big int		254	Estimated labor cost
construction	big int		50	Estimated construction cost
problem	varchar		200	Potential issues/risks
project_id	big int	Foreign Key	12	References project ID

Table No: 9

Table Name: quotation

Primary Key: id

Foregin Key: contractor_id, project_id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique quotation ID
sDate	date			Quotation start date
eDate	date			Quotation end date
amount	varchar		100	Quoted project cost
quotation	varchar		100	Quotation details
status	varchar		20	Quotation status
contractor_id	big int	Foreign Key	12	References contractor ID
project_id	big int	Foreign Key	12	References project ID

Table No: 10

Table Name: fundrequest

Primary Key: id

Foregin Key: project_id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique fund request ID
reDate	datetime			Request date
amount	varchar		100	Requested fund amount
status	varchar		20	Request status
project_id	big int	Foreign Key	12	References project ID

Table No: 11

Table Name: complaint

Primary Key: id

Foregin Key: senior_id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique complaint ID
date	datetime			Complaint date
subject	varchar		100	Complaint subject
desc	varchar		100	Complaint details
name	varchar		50	Complainant's name
address	varchar		20	Complainant's address
contact	varchar		20	Contact number
status	varchar		20	Complaint status
senior_id	big int	Foreign Key	12	References senior ID

Table No: 12

Table Name: farmer

Primary Key: id

Foregin Key: user_id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique farmer ID
name	varchar		50	Farmer's name
phone	varchar		20	Contact number
email	varchar		254	Email address
address	varchar		50	Residential address
user_id	big int	Foreign Key	12	References user ID
land	varchar		100	Land details

Table No: 13

Table Name: irrigationofficer

Primary Key: id

Foregin Key: user_id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique officer ID
name	varchar		50	Officer's name
phone	varchar		20	Contact number
email	varchar		254	Email address
address	varchar		50	Residential address
qualification	varchar		50	Officer's qualifications
user_id	big int	Foreign Key	12	References user ID

Table No: 14

Table Name: farmercomplaint

Primary Key: id

Foregin Key: user_id

FIELD NAME	DATA TYPE	CONSTRAINTS	SIZE	DESCRIPTION
id	int	Primary Key	12	Unique complaint ID
date	datetime			Complaint submission date
subject	varchar		100	Complaint subject
desc	varchar		100	Complaint details
response	varchar		100	Officer's response
responseDate	datetime			Response date
Status	varchar		20	Complaint status
farmer_id	big int	Foreign Key	12	References farmer ID
officer_id	big int	Foreign Key	12	References irrigation officer ID

CHAPTER – 5

SYSTEM TESTING

5.1 TESTING

Testing is a process to show the correctness of the program. Testing is needed to show completeness, it improves the quality of the software and to provide the maintenance aid. Some testing standards are therefore necessary reduce the testing costs and operation time. Testing software extends throughout the coding phase and it represents the ultimate review of configurations, design and coding. Based on the way the software reacts to these testing, we can decide whether the configuration that has been built is study or not. All components of an application are tested, as the failure to do so many results in a series of bugs after the software is put to use.

5.1.1 BLACK BOX TESTING

Black box testing, also called behavioural testing, focuses on the functional requirements of software. This testing approach enables the software engineer to derive the input conditions that will fully exercise all requirements for a program. Black box testing attempts to find the errors like,

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behaviour or performance error
- Initialization and termination errors
- In Black box testing software is exercised over a full range of inputs and outputs are observed for correctness

5.1.2 WHITEBOX TESTING

White box testing is also called Glass box testing is a test case design control; structure of the procedural design to derive test cases using White box testing method, the software engineer can derive the test cases that guarantee that all independent paths within the module have been exercised at least once. Exercise all logic decisions on their true or false sides. Execute all loops at their boundaries and within their operational bounds. Exercise internal data structure to ensure their validity.

5.1.3 SOFTWARE TESTING STRATEGIES

Software Testing involves,

- Unit testing
- Integration testing
- System testing

The first level of test is unit testing. The purpose of unit testing is to ensure that each program is fully tested.

The second step is integration testing. In this individual program units or programs are integrated and tested as a complete system to ensure that the software requirements are met.

System testing involves a series of tests to verify that all system elements have been properly integrated it also attempts to verify the software's protection mechanisms. It also tests the run time performance of a system within the context of an integrated system. It also includes stress testing in which the system is executed in a manner that demands resources in abnormal quantity, frequency or volume.

Acceptance Testing involves planning and the execution of various types of tests in order to demonstrate that the implemented software system satisfies the requirements. Finally our project meets the requirements after going through all the levels of testing.

CHAPTER - 6

IMPLEMENTATION

6.1 IMPLEMENTATION

Implementation is the process of converting the system design into code, testing the system and giving the user training. Implementation of a new system design is a crucial phase in the system development life cycle.

Implementation simply means converting a new system design into operation. This involves creating computer compatible files, training the operating staff and installing hardware, terminals and telecommunications network before the system is up and running. A critical factor in conversion is not disrupting the functioning of e organization. There are three types of implementation,

- Implementation of computer system to replace manual system.
- Implementation of new computer system to replace existing system.
- Implementation of modified application to replace an existing one.

Implementation walkthroughs ensure that the completed system actually solves the original problem. This walkthrough occurs just before the system goes into use, and it should include careful review of all manuals, training materials and system documentation. Again, users, the analyst and the members of the computer services staff may attend this meeting. Required user training should be provided for the company to use the panel, to create database, database user, assigning user to database and for uploading to the appropriate directory.

By following this implementation plan, the **Cost Analysis And Tendering Of Irrigation Department** project will be developed systematically, ensuring all features are integrated, tested, and deployed effectively. The focus on security, user experience, and scalability will help the application serve its intended purpose efficiently and reliably.

CHAPTER - 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The **"Cost Analysis and Tendering of Irrigation Department"** project provides an efficient, secure, and automated solution to streamline the cost estimation and tendering processes within the irrigation department. By replacing the outdated manual methods with a web-based system, the proposed solution addresses the limitations of the existing system, such as time consumption, lack of security, and inefficiency.

This system ensures transparency in tender management, improves accuracy in cost estimation, and facilitates seamless communication between stakeholders through real-time notifications and role-based access control. Additionally, its scalability, user-friendly interface, and integration with third-party tools make it a robust platform for managing large-scale irrigation projects.

In conclusion, this project not only enhances the operational efficiency of the irrigation department but also ensures accountability, data security, and transparency, ultimately contributing to the timely execution of infrastructure projects that benefit the community.

7.2 FUTURE SCOPE

The **"Cost Analysis and Tendering of Irrigation Department"** platform has a significant potential for growth and development in the future, with several enhancements and expansions that could be implemented to improve functionality and user experience. Below are some areas of future scope:

With a well-structured implementation and focusing on performance, Integration with Advanced Technologies, Integrate Geographic Information System(GIS) ,Blockchain for Enhanced Transparency, Expansion to Other Departments.

By implementing these future enhancements, the system can evolve into a comprehensive and cutting-edge solution that serves not just the irrigation department but also sets a benchmark for other government and private sector projects.

By continuously gathering user feedback and iterating on the system, **Cost Analysis And Tendering Of Irrigation Department** can evolve and expand its services, ensuring long-term value and success in the market.

BIBLIOGRAPHY

REFERENCES		**AUTHOR**
Analysis and Design of Information System	:	James A Senn
Introduction to Data Base Management System	:	Naveen Prakash
Ideas to develop a literature review	:	Cooper. M
Django - Build powerful backends	:	Programming with Mosh (YouTube)
Python – Full course for Beginners	:	Programming with Mosh (YouTube)
W3school – codes	:	www.w3schools.com
Geeksforgeeks – code	:	www.geeksforgeeks.org

CHAPTER - 9

SOURCE CODE

9. SAMPLE CODE

9.1 Approval – seapproval.html

```
{% extends 'adminbase.html' %} {% block content %}
<div class="container">
<h1 class="text-center mt-3">Senior Engineer</h1>
<table  class="table table-bordered table-striped table-secondary table-hover ">
<tr>
<th>NAME</th>
<th>ADDRESS</th>
<th>CONTACT</th>
<th>EMAIL</th>
<th>QUALIFICATION</th>
<th colspan="2">ACTION</th>
</tr>
{% for d in data %}
<tr>
<td>{{d.name}}</td>
<td>{{d.address}}</td>
<td>{{d.phone}}</td>
<td>{{d.email}}</td>
<td>{{d.qualification}}</td>
<td>
{% if d.user.is_active %}
<a href="/adminapproveuser?id={{d.user_id}}&status=0&url=/adminse"class="btn btn-
danger">Reject</a>
{% else %}
<a href="/adminapproveuser?id={{d.user_id}}&status=1&url=/adminse"class="btn btn-
success">Approve</a          >
{% endif %}
</td>
</tr>
{% endfor %}
```

```
</table>
</div>
<script>
var msg = '{{msg}}';
if (msg != '') alert(msg);
</script>
{% endblock %}
```

9.2 Registration – Farmerregistration.html

```
{% extends 'commonbase.html' %}
{% block content %}
<style>
th,td{
   padding: 10px;
}
</style>
<div>
<h1 class="text-center mt-3">Farmer Registration</h1>
<form method="POST" enctype="multipart/form-data">
{% csrf_token %}
<table class="m-auto">
<tr>
<td>Name</td>
 <td><input type="text" class="form-control" name="txtName" pattern="[a-zA-Z ]+" required></td>
</tr>
<tr>
<td>Address</td>
<td><textarea name="txtAddress" class="form-control" required></textarea></td>
</tr>
 <tr>
<td>Contact</td>
<td><input type="text" class="form-control" name="txtContact" pattern="[6789][0-9]{9}"
maxlength="10" required></td>
</tr>
```

```
<tr>
<td>Land Proof</td>
<td><input type="file" class="form-control" name="txtLand" required></td>
</tr>
<tr>
<td>Email</td>
<td><input type="email" class="form-control" name="txtEmail" required></td>
</tr>
<tr>
<td>Password</td>
<td><input type="password" class="form-control" name="txtPassword" pattern="^(?=.*[A-Za-z])(?=.*\d)(?=.*[@$!%*#?&])[A-Za-z\d@$!%*#?&]{8,}$" required></td>
</tr>
<tr>
<td></td>
 <td><input type="submit" class="btn btn-primary" value="Register"></td>
 </tr>
</table>
</form>
</div>
<script>
    var msg="{{msg}}";
    if(msg!="")
        alert(msg)
</script>
{% endblock %}
```

9.3 Login.html

```
{% extends 'commonbase.html' %}
{% block content %}
<style>
   .login-container {
```

```
    max-width: 400px;
    margin: 50px auto;
    padding: 30px;
    border-radius: 10px;
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
    background: linear-gradient(to right bottom, #ffffff, #f8f9fa);
}

.login-title {
    color: #3a3a3a;
    font-size: 28px;
    margin-bottom: 25px;
    text-align: center;
    font-weight: 600;
}

.form-group {
    margin-bottom: 20px;
}

.form-label {
    font-weight: 500;
    margin-bottom: 8px;
    display: block;
    color: #555;
}

.form-control {
    border-radius: 6px;
    border: 1px solid #ced4da;
    padding: 12px 15px;
    width: 100%;
    transition: all 0.3s;
}
```

```
    .form-control:focus {
        border-color: #4d94ff;
        box-shadow: 0 0 0 3px rgba(77, 148, 255, 0.2);
        outline: none;
    }
    .btn-login {
        background-color: #4d94ff;
        border: none;
        color: white;
        padding: 12px 20px;
        border-radius: 6px;
        width: 100%;
        font-weight: 600;
        cursor: pointer;
        transition: all 0.3s;
    }
    .btn-login:hover {
        background-color: #3a85ff;
        transform: translateY(-2px);
    }
    .signup-link {
        text-align: center;
        margin-top: 20px;
        font-size: 14px;
        color: #666;
    }
    .signup-link a {
     color: #4d94ff;
     text-decoration: none;
    }
 </style>
<div class="login-container">
<h1 class="login-title">Welcome Back</h1>
<form method="POST">
```

```
{% csrf_token %}

<div class="form-group">
<label class="form-label" for="email">Username</label>
<input type="text" class="form-control" id="email" name="txtEmail" placeholder="Enter your username" required>
</div>
<div class="form-group">
<label class="form-label" for="password">Password</label>
<input type="password" class="form-control" id="password" name="txtPassword" placeholder="Enter your password" required>
</div>
<button type="submit" class="btn-login">Sign In</button>
<div class="signup-link"> Don't have an account? <a href="#">Sign up</a>  </div>
</form>
</div>

<script>

 var msg = "{{msg}}";

   if(msg != "") {
      alert(msg);
   }

</script>

{% endblock %}
```

9.4 Complaint- Commoncomplaint.html

```
{% extends 'commonbase.html' %}
{% block content %}
<style>
th,td{
   padding: 10px;
}
```

```
</style>
<div>
 <h1 class="text-center mt-3">Complaints</h1>
<form method="POST">
{% csrf_token %}
<table class="m-auto">
<tr>
<td>Subject of complaint</td>
<td><input type="text" class="form-control" name="txtSubject" required></td>
</tr>
<tr>
<td>Description</td>
<td><textarea class="form-control" required name="txtDesc"></textarea></td>
</tr><tr>
<td>Name of complainant</td>
 <td><input type="text" class="form-control" name="txtName" pattern="[a-zA-Z ]+" required></td>
</tr>
<tr>
<td>Address</td>
<td><textarea name="txtAddress" class="form-control" required></textarea></td>
</tr>
<tr>
<td>Contact</td>
<td><input type="text" class="form-control" name="txtContact" pattern="[6789][0-9]{9}"
maxlength="10" required></td>
</tr>
<tr>
<td></td>
<td><input type="submit" class="btn btn-primary" value="Submit"></td>
</tr>
</table>
</form>
</div>
<script>
```

```
    var msg="{{msg}}";
    if(msg!="")
        alert(msg)
</script>
{% endblock %}
```

CHAPTER-10

APPENDIX

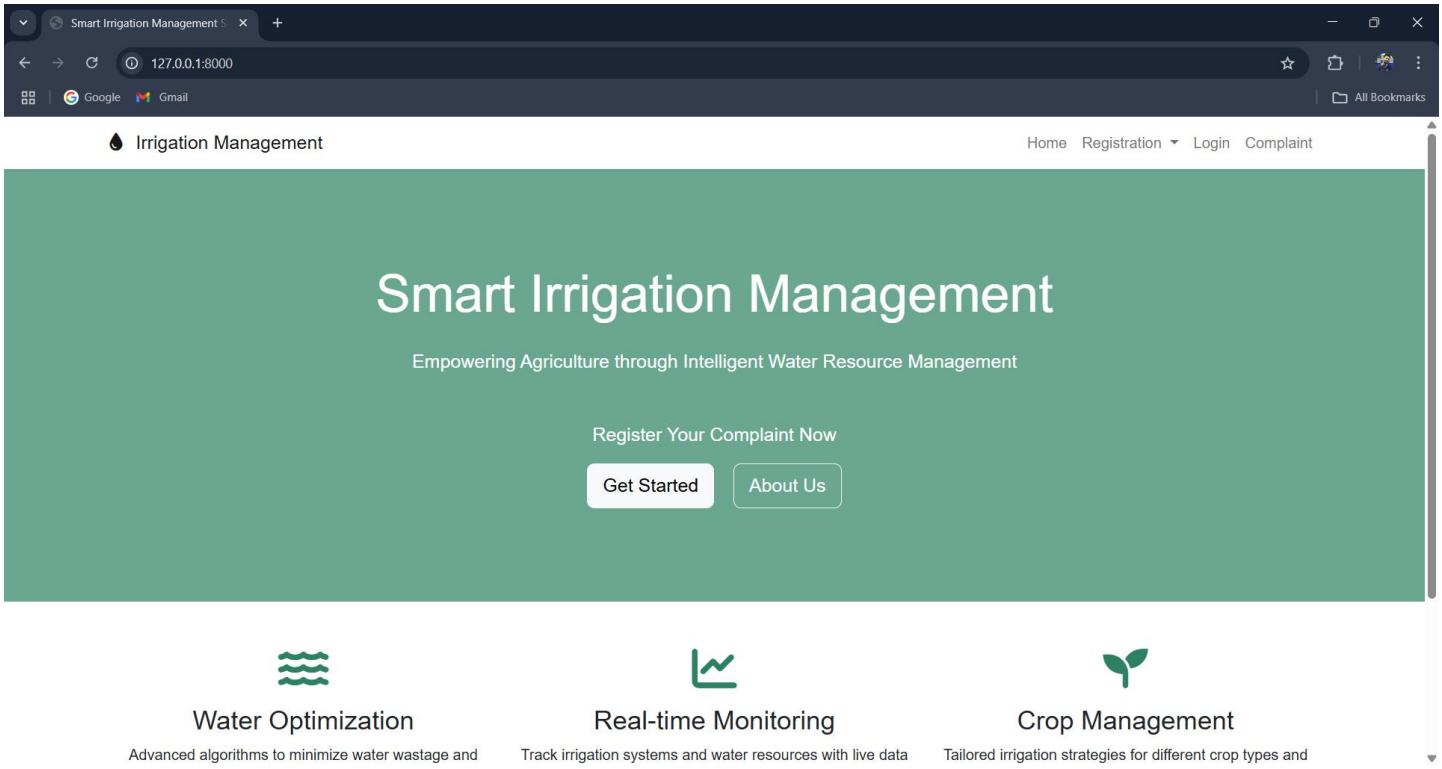


Figure 10.1 : Home Page

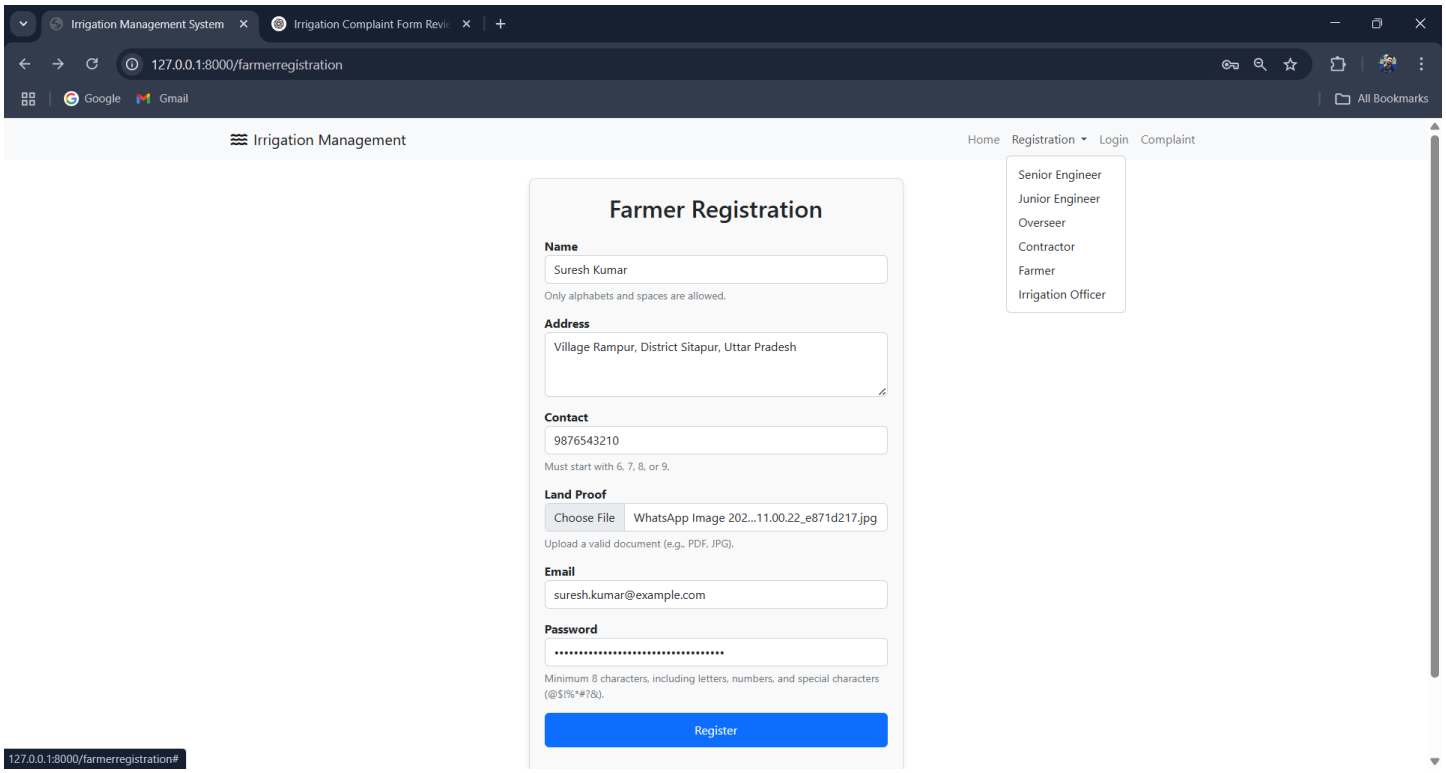


Figure 10.2: Registration – Farmer

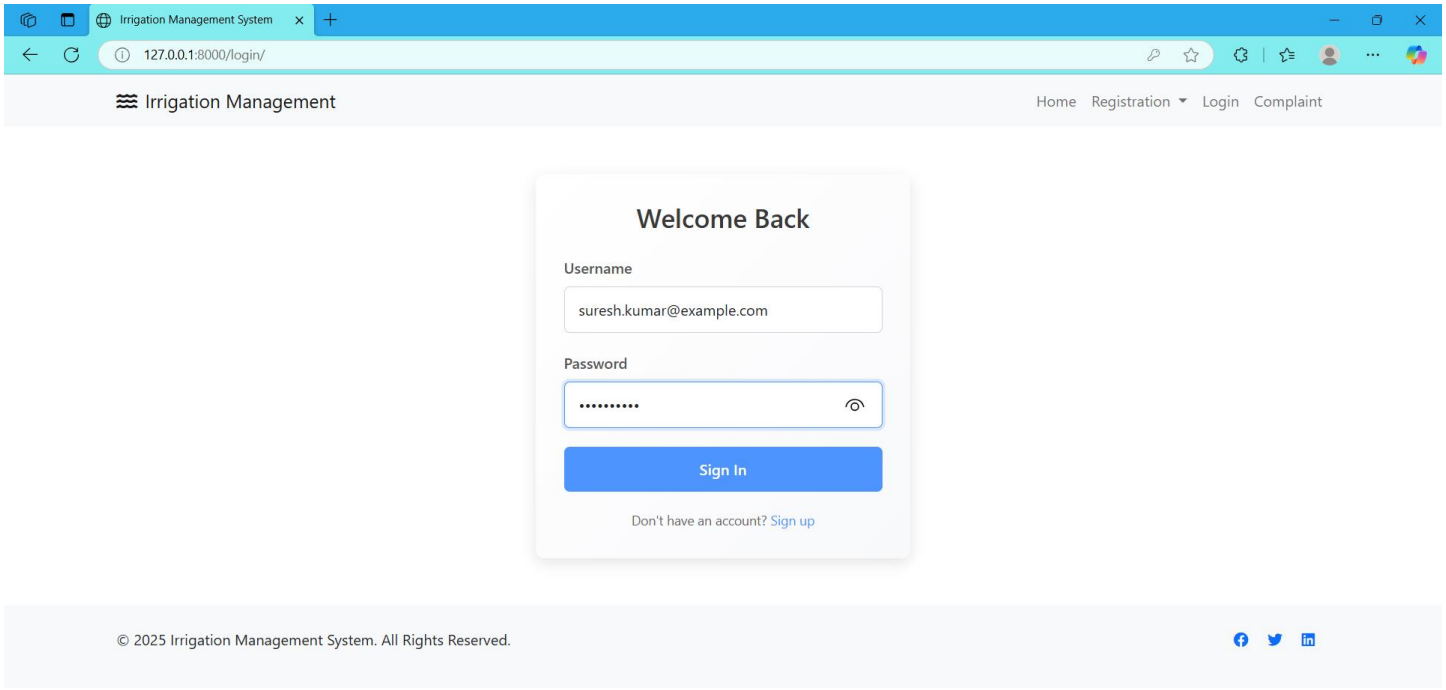


Figure 10.3 : Login

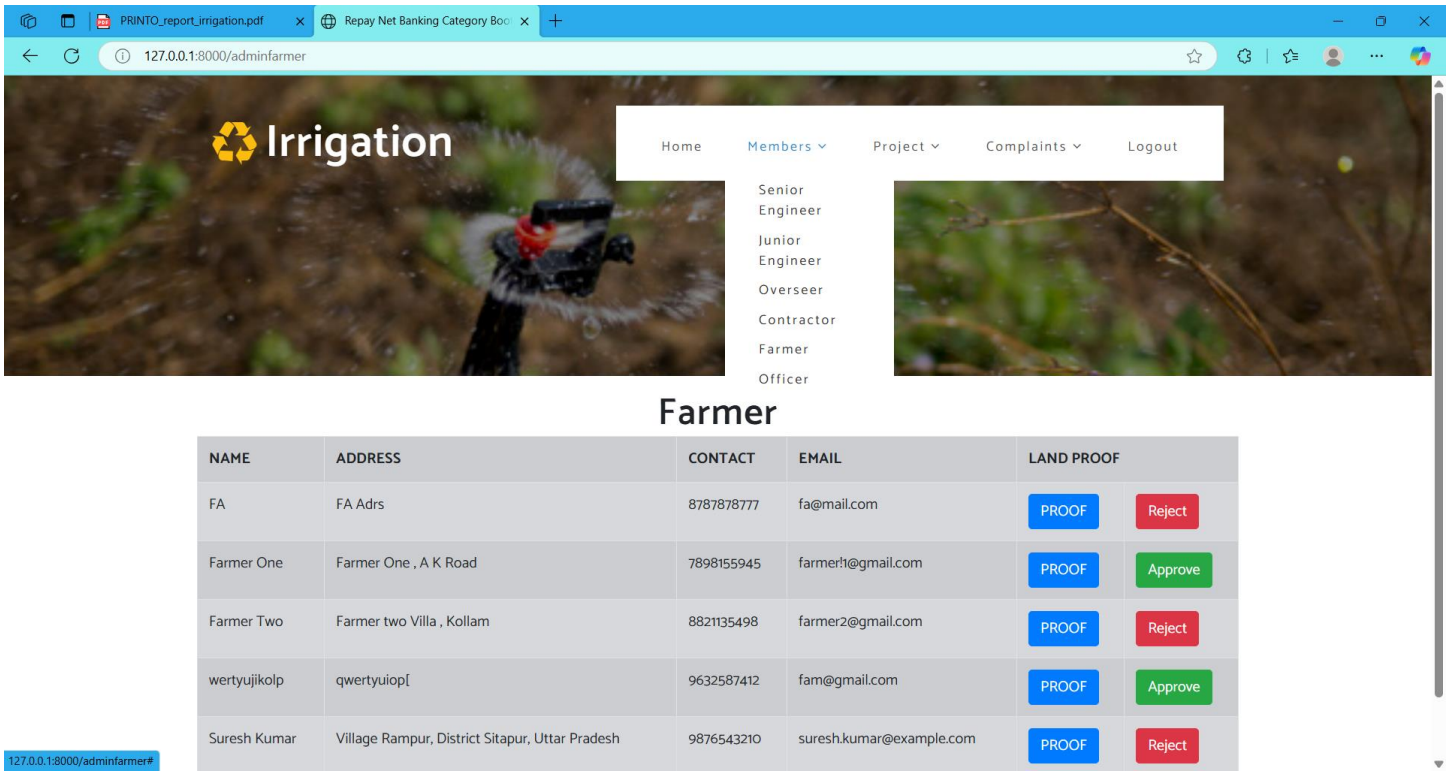


Figure 10.4 : Approval – Farmer

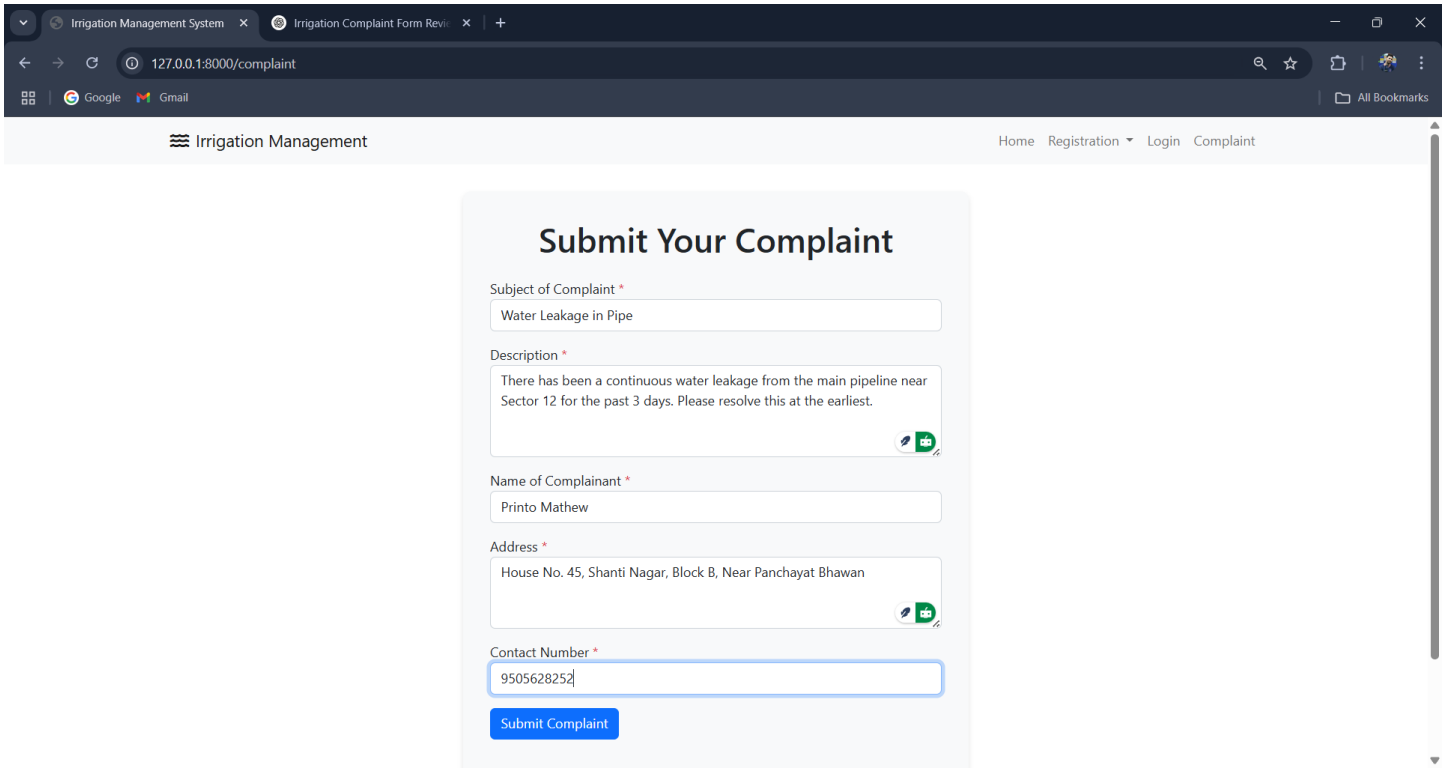


Figure 10.5 : Complaint