CHAPTER – 1

INTRODUCTION

1.1 OBJECTIVE OF THE PROBLEM

The objective of the Rural Employment Portal is to address the social welfare needs of the rural population in India, specifically those who rely on unskilled labor for their livelihood. The program aims to enhance the livelihood security of households in rural areas by providing a guaranteed minimum number of days of wage employment each year. The key objectives of the Rural Employment Portal project can be summarized as follows:

- Poverty Alleviation: The primary objective is to alleviate poverty and improve the economic conditions of rural households by ensuring access to employment opportunities.

- Livelihood Security: The project seeks to enhance the livelihood security of rural families by offering a minimum of 100 days of guaranteed wage employment annually.

- Unskilled Labor Empowerment: The program empowers individuals engaged in unskilled labor by providing them with steady employment and income opportunities.

- Effective Implementation: The portal aims to streamline the implementation of rural employment programs by providing a digital platform for work allocation, application processing, and tracking.

- Efficient Management: By dividing the system into various modules, such as Administrator, Panchayath Office, ADS Members, and Work Employees, the project intends to efficiently manage tasks, responsibilities, and communication among stakeholders.

- Transparency: The portal promotes transparency in the allocation of work, payment processing, and overall program management, reducing the chances of corruption and ensuring fair distribution of opportunities.

- Monitoring and Evaluation: Regular review, monitoring, and evaluation of processes and outcomes are essential components of the project to ensure its effectiveness and make necessary improvements.

- Communication and Collaboration: The portal facilitates communication and collaboration among different user categories, such as administrators, Panchayath offices, ADS members, and work employees, fostering a cohesive and well-coordinated approach to rural employment.

- Empowerment at Local Levels: By involving Panchayath offices and ADS members, the project aims to empower local communities and institutions to actively participate in the process of work allocation and management.

- Skill Development: While the work involves unskilled labor, the project can contribute to skill development, as participants gain experience and exposure to various types of work.

- Documentation and Records: The portal enables efficient documentation of work details, complaints, feedback, and payments, aiding in maintaining accurate records for future reference and audits.

- Socio-Economic Development: Ultimately, the Rural Employment Portal contributes to the socio-economic development of rural areas by providing employment opportunities, improving income levels, and fostering local development.

STATEMENT OF THE PROBLEM

The rural areas of India are home to a significant population of people who heavily rely on unskilled labor as their primary source of income. To address the challenges of rural poverty and lack of livelihood security, the Government of India has introduced a social welfare program known as the Rural Employment System. This program aims to provide a minimum of 100 days of guaranteed wage employment in each financial year to households in rural areas whose adult members are willing to engage in unskilled manual work. The primary objective is to enhance the economic well-being and security of rural households, thereby contributing to overall socio-economic development.

To effectively implement this program and ensure its success, the Rural Employment Portal has been developed, consisting of four interconnected modules: Administrator, Panchayath Office, ADS Members, and Work Employees, along with a User Module. Each module has specific roles and responsibilities to streamline the processes of work allocation, application verification, work details entry, communication, and payment distribution. The Administrator module acts as the central authority, overseeing and managing the entire system.

In many rural areas of India, a significant portion of the population depends on unskilled labor as their primary source of income. Despite various government initiatives, including the Rural Employment System, there are persistent challenges in effectively providing and managing 100 days of guaranteed wage employment to eligible households. These challenges include inefficient allocation of work, delays in payment distribution, and a lack of transparency and accountability in the process. As a result, many rural households continue to face economic insecurity and poverty, and the full potential of the program to contribute to socio-economic development remains unrealized.

CHAPTER - 2

SYSTEM ANALYSIS

System Analysis is a detailed study of various operation performed by a system and their relationship within and outside of the system. Here the key question is: What must be done to solve the problem? One aspect of analysis is defining the boundaries of a system and determining whether or not a candidate system should consider other related system. Analysis begins when a user or manager begins a study of the program using an existing system.

During analysis, data is collected on the various files, decision points and transactions handled by the present system. The commonly used tools in system are dataflow diagrams, interviews, onsite observations etc. System analysis is application of the system approach to the problem-solving using computers. The ingredients are the system elements, process and technology. This means that to do system works, one is to understand the system concepts and how the organizations operate as a system and the design appropriate computer-based system that will meet the organizations requirements. It is actually customized approach to the use of computer problem solving.

Analysis can be defined as the separation of a substance into parts for study an interpretation, detailed examination. System development revolves around a lifecycle that being with the recognition of user needs. The critical phase of managing system project is planning. To launch a system investigation, we need a master plan detailing the steps taken, the people to be questioned and outcome expected.

System analysis can be categorized into four parts:

- System planning and initial investigation.
- Information gathering.
- Applying Analyzing tools for structured analysis.
- Feasibility study.
- Cost/ Benefits analysis.

System study or system analysis is the first among the four life cycle phases of a system. System analysis begins when a user or manager request a studying of a program in either an existing system or a project one. It involves studying the base of the organizations currently operating, retrieving and processing data to produce information with goal of determining how to make it work better. System analysis itself breaks down into stages preliminary and detailed. During preliminary analysis, the analyst and the user list the objectives of the system. To arrive at a preliminary report, the analyst interviews key personnel in the organization and scheduling meetings with the users and the management.

If management approves preliminary report, the system analysis or study phases advantage to the second stage, the detailed analysis. If the management approves the result of a preliminary analysis, the analyst conducts a detailed analysis, gathering facts about the old system, outline objectives for a new one, estimating costs, listing possible alternatives and making recommendation. After gathering the analysis will arrange it in organized way called the feasibility study.

Thus, the objective of analysis phase of the system analysis and design exercise is the establishment of the requirement for the system to be acquired, developed and installed. Fact finding or gathering is essential of requirements. In brief analysis of the system helps an analyst to make a clear view of an existing system and thereby can give suggestions for the improvement of the new system information about the organization's policies, goals, objectives and structure explains the kind of environment that promotes the introduction of computer-based system. It is necessary that the analyst must be familiar with the objectives, activities and the functions of the organization in which the computer system is to be implemented.

2.1 EXISTING SYSTEM

In manual system, everything depends upon paper also there is no automated system for keeping the records. The process which is accessible on rural employment is paper based. All work request taken by panchayath is paper based. Hence, we realized that this paper-based system is easily vulnerable to get harmed because of several reasons and it leads to different problem i.e., panchayath couldn't have arranged records of work properly. The all process can be taken as paper work lead to time consuming and that will take more time to arrange it. The rural employment portal has information passing delay.

2.2 Disadvantages of Existing System

- Dependency on Paper: The current system relies heavily on paper-based records, which can be inefficient, cumbersome, and prone to errors. This manual process involves physically handling and storing a large volume of paperwork, making it susceptible to damage, loss, or misplacement.

- Lack of Automation: There is a notable absence of an automated system for record-keeping and data management. This absence can result in inefficiencies, delays, and difficulties in accessing and organizing essential information.

- Accessibility: The paper-based process may not be easily accessible to those who need the information, particularly in rural areas. This can lead to challenges in retrieving and utilizing data for decision-making.

- Vulnerability to Harm: Physical records are vulnerable to various risks, including damage due to environmental factors like humidity or pests, which can compromise the integrity of the data.

2.3 PROPOSED SYSTEM

Rural employment system is primarily a social welfare program of the Government of India. Majority of the poor in rural areas of the country depend mainly on the wages they earn through unskilled labor. They seek to enhance the lively hood security of the households in rural areas of the country by providing at least 100 days of guaranteed wage employment in every financial year to every household whose adult members volunteer to do unskilled manual work. It has to undertake regular review, monitoring and evaluation of processes and outcomes.

2.4 Advantages of Proposed System

- Work requesting and allocation can be more simple
- Timely communication is available
- Payment can be taken more speed
- Attendance checking can be more efficient
- Adding employee more speedily
- Request sort time will less
- Worker adding take more convenient
- Ads members adding and verification can make more convenient

2.5 FEASIBILITY STUDY

Feasibility study is a test of system proposed regarding its workability, impact on the organization, ability to meet the needs and effective use of resources. Thus, when a new project is proposed, it normally goes through a feasibility study before it is approved for development.

A feasibility study is made to see if the project on completion will serve the purpose ofthe organization for the amount of work, effort and the time that is spend on it. Feasibility study lets the developer foresee the future of the project and its usefulness.

All the projects are feasible given unlimited resources and infinite time. Unfortunately, the development of the computer-based system is more likely to be played by a security of resources and difficulty delivery dates. Feasibility and risk analysis are related in many ways. If project risk is great, the feasibility of producing the quality software is reduced.

Steps in Feasibility Study

Feasibility Study involves eight steps:

- Form a project team and appoint a project leader.
- Prepare a system flow chart.
- Enumerate potential candidate systems.
- Describe and identify characteristics of candidate systems.
- Describe and evaluate performance and cost effectiveness of each candidate systems.

- Weight system performance and cost data.
- Select the best candidate system.
- Prepare and report final project directive and management.

Mainly three key considerations are involved in the feasibility analysis.

- Economic Feasibility
- Technical Feasibility
- Operational Feasibility

2.5.1 Economical Feasibility

Economical Feasibility is the most frequently used method for evaluating theeffectiveness of the candidate system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. Otherwise, further justifications or alterationsin the proposed system will have to be made if it is having a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

2.5.2 Technical Feasibility

Technical Feasibility centers on the existing computer system (hardware, software, etc) and to what extend it can support the proposed addition. For example, if the current computer is operating at 80 percent capacity, an arbitrary ceiling, then running another application could over load the system or require additional hardware. This involves financial considerations to accommodate technical enhancements. If the budget is a serious constraint, then the project is judged not feasible.

2.5.3 Operational Feasibility

The main problem faced during development of a new system is getting acceptance from the user. People are inherently resistant to changes and computers have been known to facilitate change. It is mainly related to human organizational and political aspects.

The points to be considered are:

- What changes will be brought with the system?
- What new skills will be required? Do the existing staff members have these skills? If not, can they be trained due course of time?

Generally, project will not be rejected simply because of operational feasibility but such considerations are likely to critically affect the nature and scope of the eventual recommendations. This feasibility study is carried out by a small group of people who are familiar with information system techniques, who understand the parts of the businessthat are relevant to the project and are skilled in skilled analysis and design process.

CHAPTER 3

SYSTEM SPECIFICATIONS

1.1 HARDWARE REQUIREMENTS

Processor	:	Intel i5 7 th Gen
RAM	:	8 GB DDR4
Hard Disk	:	256 GB SSD
Display Size	:	Compatible Size (Recommend 15’inch)
Screen Resolution	:	1920 * 1080
Pixels Keyboard	:	Wireless Enabled Keyboard (Recommend: Logitech)
Mouse	:	Wireless Enabled Mouse (Recommend: Logitech)
Monitor	:	Touch Capacity LED Monitor Dedicated
Graphics Card	:	Nvidia GeForce 920m 2GB DDR4

3.2 SOFTWARE REQUIREMENTS

Operating System	:	Windows 11
Platform	:	Visual Studio Code
Language	:	Python
Framework	:	Django
Frontend	:	HTML, CSS, JS
Database	:	SQLite3
Backend	:	Python
Web Browser	:	Google Chrome

CHAPTER - 4

SYSTEM DESIGN

System Design involves translating system requirements and conceptual design into technical specifications and general flow of processing. After the system requirements have been identified, information has been gathered to verify the problem and after evaluating the existing system, a new system is proposed.

System Design is the process of planning of new system or to replace or complement an existing system. It must be thoroughly understood about the old system and determine how computers can be used to make its operations more effective.

System design sits at technical the kernel of system development. Once system requirements have been analyzed and specified system design is the first of the technical activities-design, code generation and test- that required build and verifying the software. System design is the most creative and challenging phases of the system life cycle. The term design describes the final system and the process by which it is to be developed.

System design is the high-level strategy for solving the problem and building a solution. System design includes decisions about the organization of the system into subsystems, the allocation of subsystems to hardware and software components and major conceptual and policy decision that forms the framework for detailed design.

There are two levels of system design:

- Logical design.
- Physical design.

In the logical design, the designer produces a specification of the major features of the system which meets the objectives. The delivered product of logical design includes current requirements of the following system components:

- Input design.
- Output design.

- Database design.

Physical design takes this logical design blue print and produces the program software, files and a working system. Design specifications instruct programmers about what the system should do. The programmers in turn write the programs that accept input from users, process data, produce reports, and store data in files.

Structured design is a data flow-based methodology that partitions a program into a hierarchy of modules organized top-down manner with details at the bottom. Data flow diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes may be described logically and independently of the physical components.

DATA FLOW DIAGRAM

4.1 Context Level Diagram

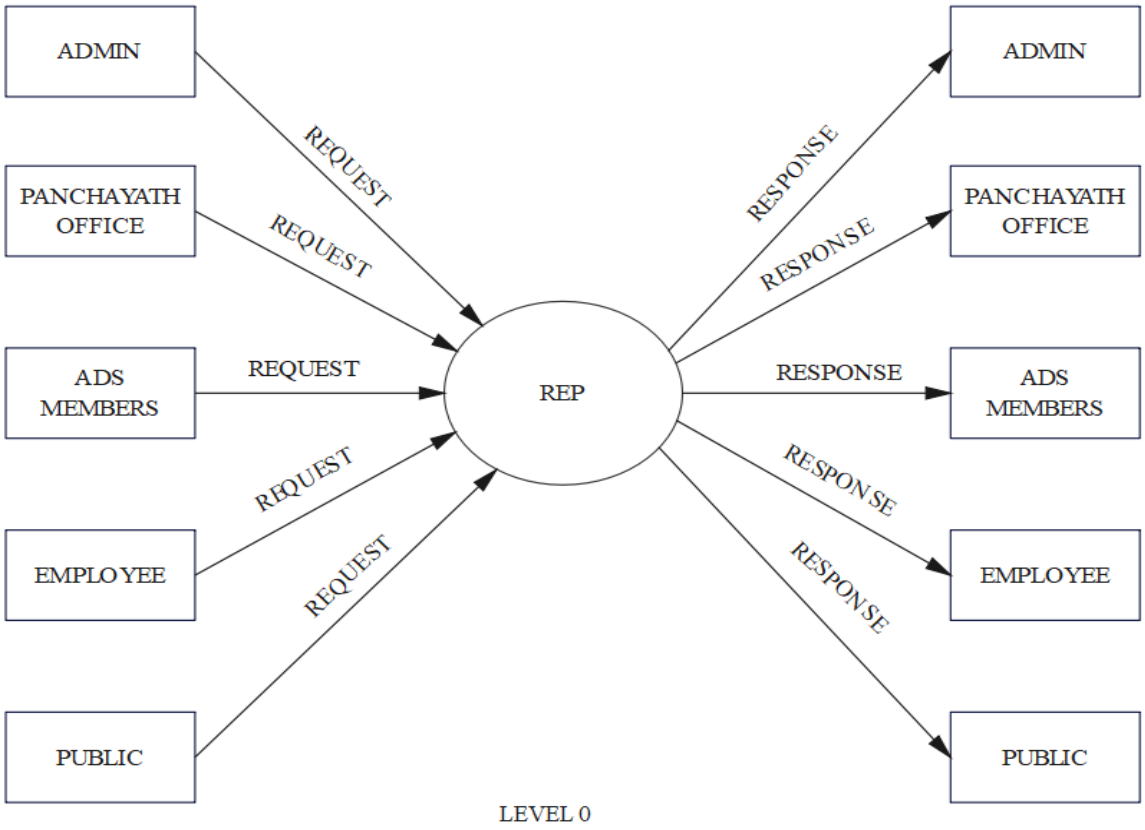


Figure 4.1 Context Level Diagram

4.2 Data Flow Diagram

4.2.1 Level 1 Admin

+

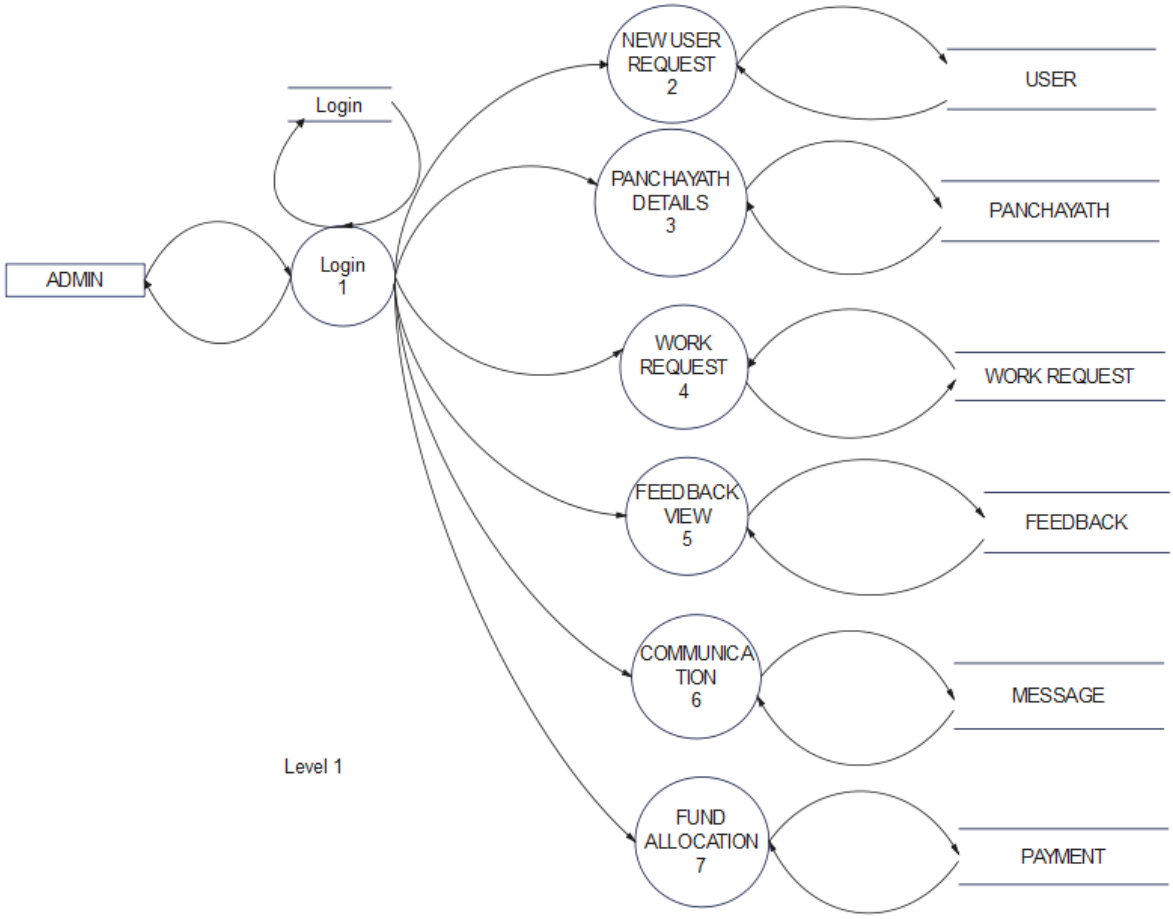


Figure 4.2.1 Admin (Level 1)

4.2.2 Level 1 Panchayath Office

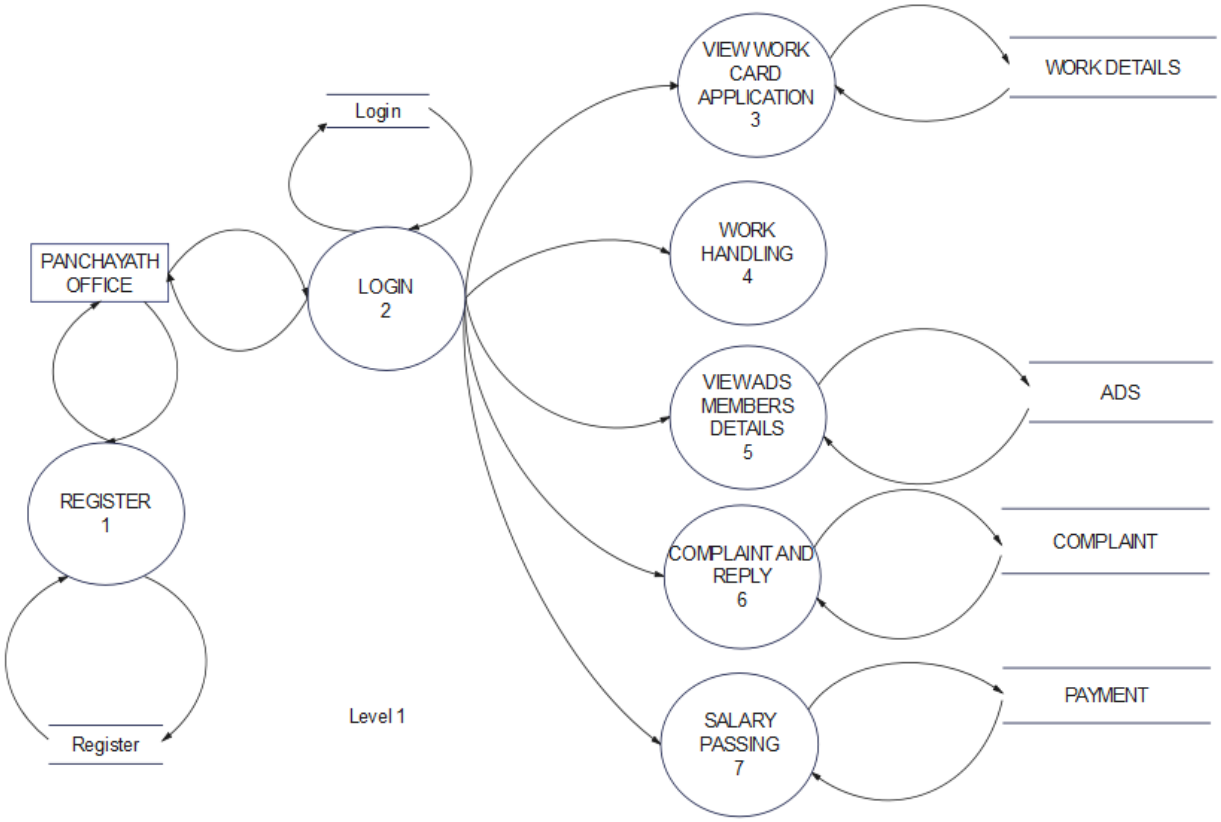


Figure 4.2.2 Panchayath Office (Level 1)

4.2.3 Level 1 Ads Members

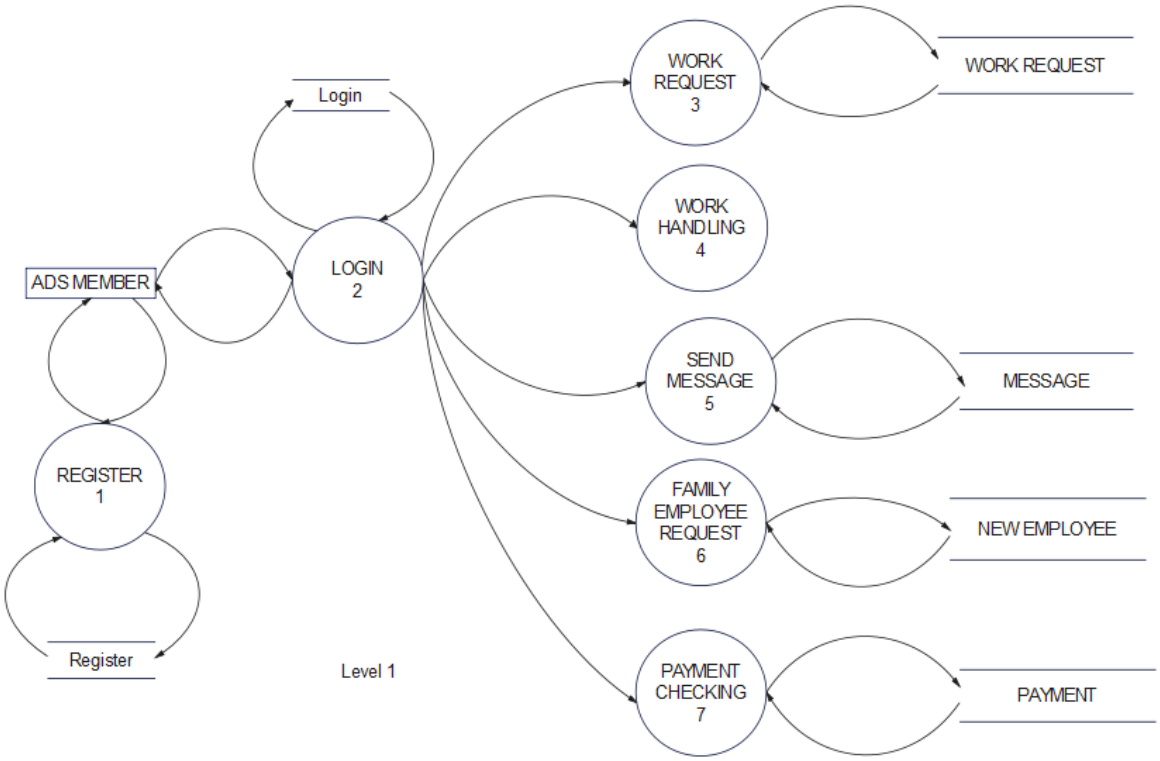


Figure 4.2.3 Ads Members (Level 1)

4.2.4 Level 1 Work Employee

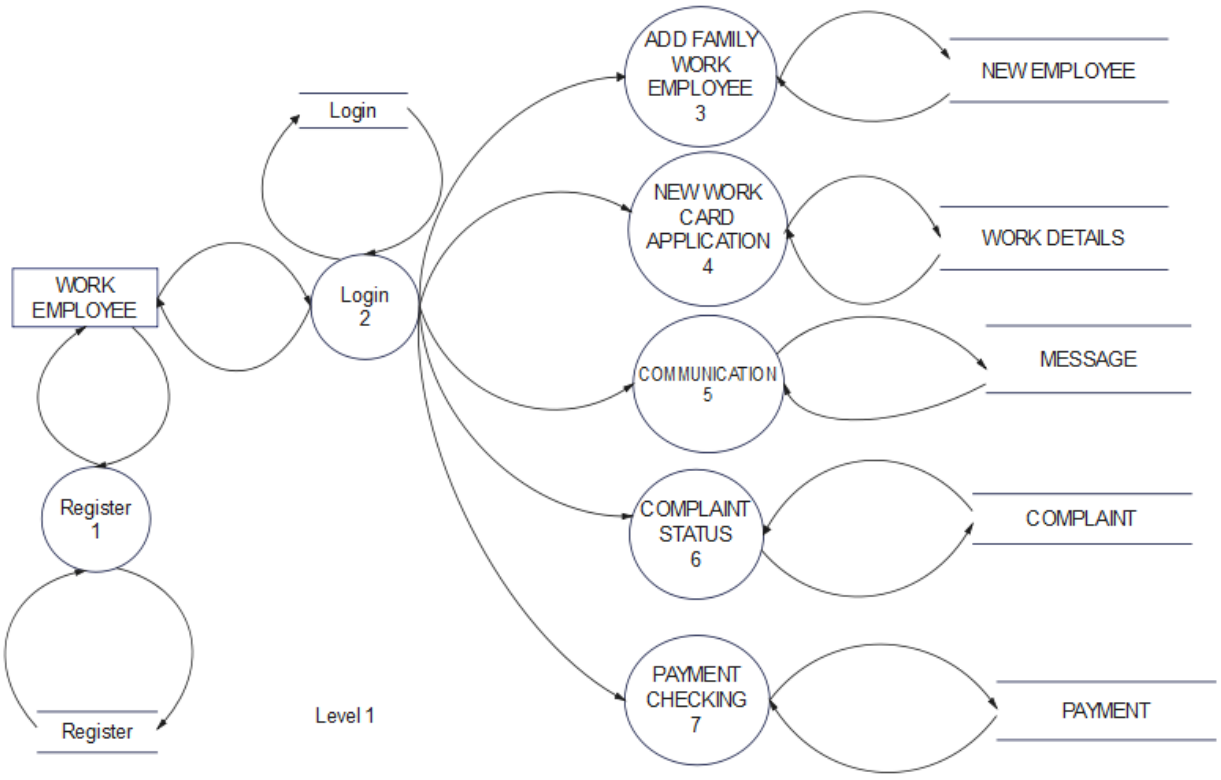


Figure 4.2.4 Work Employee (Level 1)

4.2.5 Level 1 User

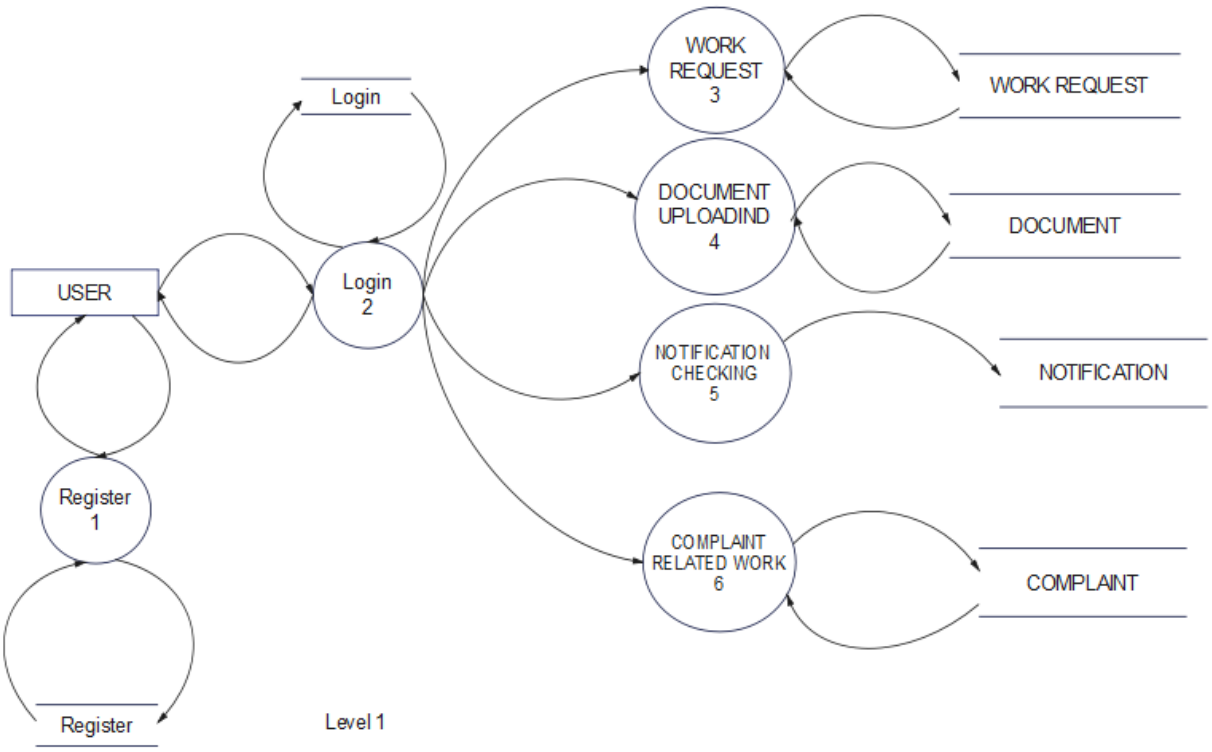


Figure 4.2.5 User (Level 1)

4.2.6 Sub-Level 1 ADS Member

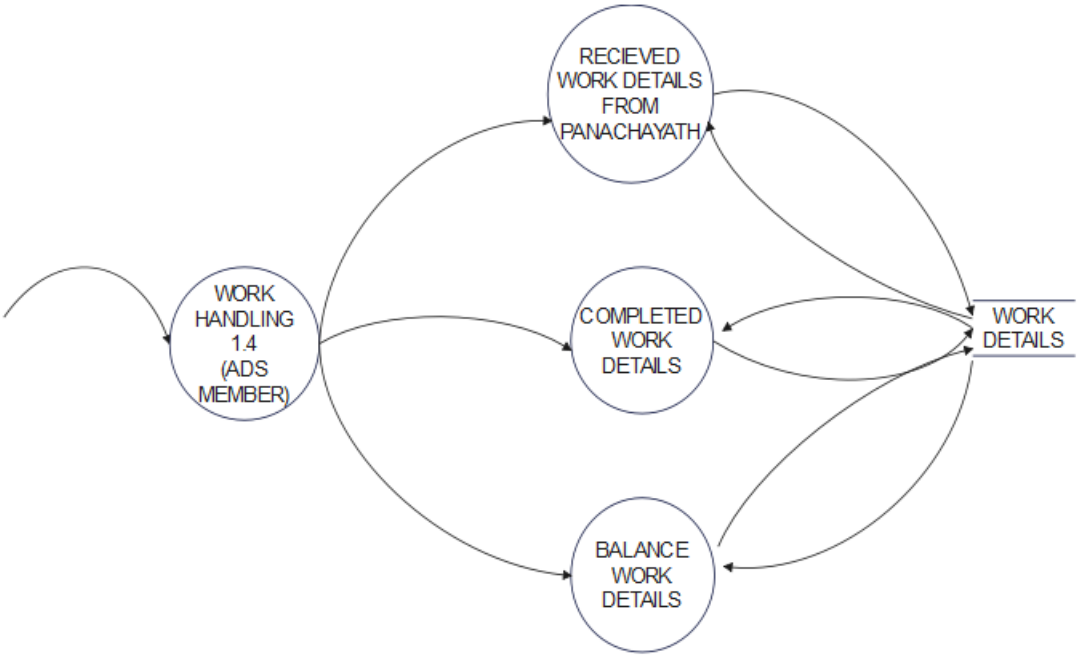


Figure 4.2.6 ADS Member (Sub-Level)

4.2.7 Sub-Level 1 Panchayath Office

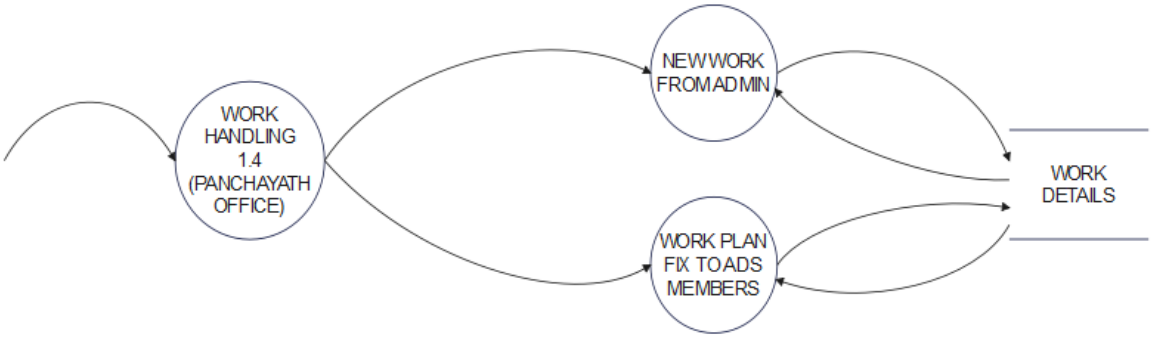


Figure 4.2.7 Panchayath Office (Sub-Level)

4.3 E R Diagram

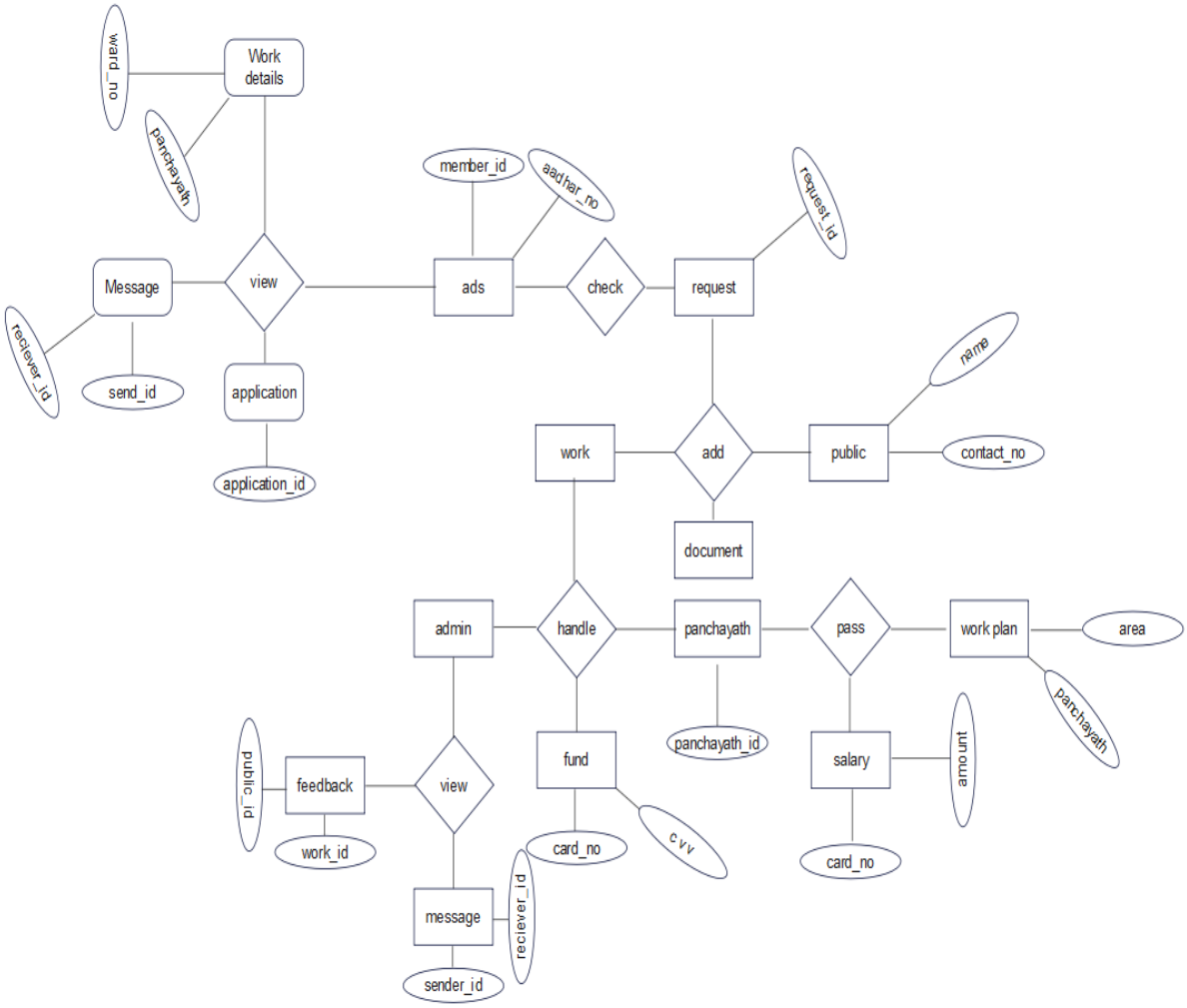


Figure 4.3.1 ER Diagram

4.3 Database Design

The most important aspect of building software systems is database design. The highest level in the hierarchy is the database. It is a set of inter-related files for real timeprocessing. It contains the necessary data for problem solving and can be used by several users accessing data concurrently. The general objective of database design is to make thedata access easy, inexpensive and flexible to the user.

Database design is used to define and then specify the structure of business used in the client/server system. A business object is nothing but information that is visible to the users of the system. The database must be a normalized one.

Database management system (DBMS) allows the data to be protected and organized separately from other resources like hardware, software and programs. DBMS is a software package, which contains components that are not found in other data management packages. The significance of DBMS is the separation of data as seen by theprograms and data as stored on the direct access storage devices, i.e. the difference between logical and physical data.

In my project, I have used Microsoft SQL Server 2005 as the database to implement the data store part. The most important part in the database design is the identification of tables to be used. Database design activity deals with the design of the physical database. A key is to determine how the access paths are to be implemented. A physical path is derived from a logical path. Pointers, chains or other mechanisms may implement it.

The tables used in this project are:

Data Integrity and Constraints

Data Integrity refers to the validity of data. Data integrity can be compromised in a number of ways:

- Human errors when data is entered
- Errors that occur when data is transmitted from one computer to another
- Software bugs or viruses
- Hardware malfunctions, such as disk crashes

- Natural disasters, such as fires and floods There are many ways to minimize these threats to data integrity. These include:
- Backing up data regularly
- Controlling access to data via security mechanisms
- Designing user interfaces that prevent the input of invalid data
- Using error detection and correction software when transmitting data

Types of Data Integrity

This section describes the rules that can be applied to table columns to enforce different types of data integrity.

- **Null Rule**

A null rule is a rule defined on a single column that allows or disallows inserts or updates of rows containing a null (the absence of a value) in that column.

- **Unique Column Values**

A unique value rule defined on a column (or set of columns) allows the insert or update of a row only if it contains a unique value in that column (or set of columns).

- **Primary Key Values**

A primary key value rule defined on a key (a column or set of columns) specifies that each row in the table can be uniquely identified by the values in the key.

- **Referential Integrity Rules**

A referential integrity rule is a rule defined on a key (a column or set of columns) in one table that guarantees that the values in that key match the values in a key in a related table (the referenced value).

Referential integrity also includes the rules that dictate what types of data manipulation are allowed on referenced values and how these actions affect dependent values. The rules associated with referential integrity are:

- Restrict: Disallows the update or deletion of referenced data.
- Set to Null: When referenced data is updated or deleted, all associated dependent data is set to `NULL`.
- Set to Default: When referenced data is updated or deleted, all associated dependent data is set to a default value.

- Cascade: When referenced data is updated, all associated dependent data is correspondingly updated. When a referenced row is deleted, all associated dependent rows are deleted.
- No Action: Disallows the update or deletion of referenced data. This differs from `RESTRICT` in that it is checked at the end of the statement, or at the end of the transaction if the constraint is deferred. (Oracle uses No Action as its default action.)
- Complex Integrity Checking

Complex integrity checking is a user-defined rule for a column (or set of columns) that allows or disallows inserts, updates, or deletes of a row based on the value it contains for the column (or set of columns).

Integrity Constraints

An integrity constraint is a declarative method of defining a rule for a column of a table. Oracle supports the following integrity constraints:

- NOT NULL constraints for the rules associated with nulls in a column
- UNIQUE key constraints for the rule associated with unique column values
- PRIMARY KEY constraints for the rule associated with primary identification values
- FOREIGN KEY constraints for the rules associated with referential integrity. Oracle supports the use of `FOREIGN KEY` integrity constraints to define the referential integrity actions, including:
    - Update and delete No Action
    - Delete `CASCADE`
    - Delete `SET NULL`
- CHECK constraints for complex integrity rules

For example, assume that you define an integrity constraint for the `salary` column of the `employees` table. This integrity constraint enforces the rule that no row in this table can contain a numeric value greater than 10,000 in this column. If an `INSERT` or `UPDATE` statement attempts to violate this integrity constraint, then Oracle rolls back the statement and returns an information error message.

The integrity constraints implemented in Oracle fully comply with ANSI X3.135-1989 and ISO 9075-1989 standards.

Advantages of Integrity Constraints

This section describes some of the advantages that integrity constraints have over other alternatives, which include:

- Enforcing business rules in the code of a database application
- Using stored procedures to completely control access to data
- Enforcing business rules with triggered stored database procedures

Declarative Ease

Define integrity constraints using SQL statements. When you define or alter a table, no additional programming is required. The SQL statements are easy to write and eliminate programming errors. Oracle controls their functionality. For these reasons, declarative integrity constraints are preferable to application code and database triggers. The declarative approach is also better than using stored procedures, because thestored procedure solution to data integrity controls data access, but integrity constraintsdo not eliminate the flexibility of ad hoc data access.

Centralized Rules

Integrity constraints are defined for tables (not an application) and are stored in the data dictionary. Any data entered by any application must adhere to the same integrity constraints associated with the table. By moving business rules from application code to centralized integrity constraints, the tables of a database are guaranteed to contain valid data, no matter which database application manipulates the information. Stored procedures cannot provide the same advantage of centralized rules stored with a table. Database triggers can provide this benefit, but the complexity of implementation is far greater than the declarative approach used for integrity constraints.

Maximum Application Development Productivity

If a business rule enforced by an integrity constraint changes, then the administrator need only change that integrity constraint and all applications automatically adhere to the modified constraint. In contrast, if the business rule were enforced by the code of each database application, developers would have to modify all applicationsource code and recompile, debug, and test the modified applications.

Immediate User Feedback

Oracle stores specific information about each integrity constraint in the data dictionary. You can design database applications to use this information to provide immediate user feedback about integrity constraint violations, even before Oracle runs and checks the SQL statement. For example, an Oracle Forms application can use integrity constraint definitions stored in the data dictionary to check for violations as values are entered into the fields of a form, even before the application issues a statement.

Superior Performance

The semantics of integrity constraint declarations are clearly defined, and performance optimizations are implemented for each specific declarative rule. The Oracle optimizer can use declarations to learn more about data to improve overall query performance. (Also, taking integrity rules out of application code and database triggers guarantees that checks are only made when necessary.)

Flexibility for Data Loads and Identification of Integrity Violations

You can disable integrity constraints temporarily so that large amounts of data can be loaded without the overhead of constraint checking. When the data load is complete, you can easily enable the integrity constraints, and you can automatically report any new rows that violate integrity constraints to a separate exceptions table.

The Performance Cost of Integrity Constraints

The advantages of enforcing data integrity rules come with some loss in performance. In general, the cost of including an integrity constraint is, at most, the same as executing a SQL statement that evaluates the constraint.

Types of Integrity Constraints

You can use the following integrity constraints to impose restrictions on the input of column values:

- NOT NULL Integrity Constraints
- UNIQUE Key Integrity Constraints
- PRIMARY KEY Integrity Constraints
- Referential Integrity Constraints
- CHECK Integrity Constraints

4.3.1 Table Design

Table Name: Public

Table Description: to store login details

Field Name	Data Type	Size	Constraint	Description
id	Auto Field	-	Primary Key	Unique identifier for pub
name	Char Field	255	-	Name of the pub
contact	Char Field	10	-	Contact number of the pub
email	Char Field	255	-	Email address of the pub
password	CharField	255	-	Password for the pub account
usertype	CharField	255	Default 'public'	User type of the pub

Table Name: Panchayath

Table Description: to store user details

Field Name	Data Type	Size	Constraint	Description
id	AutoField	-	Primary Key	Unique identifier for panchayath
panchayath_id	CharField	10	-	Panchayath ID
panchayath_name	CharField	255	-	Name of the panchayath
panchayath_address	CharField	255	-	Address of the panchayath
panchayath_district	CharField	255	-	District of the panchayath
panchayath_city	CharField	255	-	City of the panchayath
panchayath_pincode	CharField	255	-	Pincode of the panchayath
panchayath_contact	CharField	10	-	Contact number of the panchayath
panchayath_email	CharField	255	-	Email address of the panchayath
panchayath_password	CharField	255	-	Password for the panchayath account
usertype	CharField	255	Default 'panchayath'	User type of the panchayath

Table name: ADS MEMBERS

Table Description: to store hotel details

Field Name	Data Type	Size	Constraint	Description
id	AutoField	-	Primary Key	Unique identifier for ads
adsmemberid	CharField	10	-	ADS member ID
adsname	CharField	255	-	Name of the ADS member
adsgender	CharField	10	-	Gender of the ADS member
adsadharno	CharField	15	-	Aadhar number of the ADS member
adspanchayath	CharField	255	-	Panchayath of the ADS member
adsward	CharField	255	-	Ward of the ADS member
adscontact	CharField	10	-	Contact number of the ADS member
adsemail	CharField	255	-	Email address of the ADS member
adspassword	CharField	255	-	Password for the ADS member account
usertype	CharField	255	Default 'ads'	User type of the ADS member

Table Name: Work employee

Table Description: to store food details

Field Name	Data Type	Size	Constraint	Description
id	AutoField	-	Primary Key	Unique identifier for workemployee
emname	CharField	255	-	Name of the workemployee
emaddress	CharField	255	-	Address of the workemployee
empincode	CharField	255	-	Pincode of the workemployee
emdistrict	CharField	150	-	District of the workemployee
empanchayath	CharField	255	-	Panchayath of the workemployee
emward	CharField	255	-	Ward of the workemployee
emratiocard	CharField	255	-	Ratiocard of the workemployee
emadhar	CharField	255	-	Aadhar number of the workemployee
emcontact	CharField	15	-	Contact number of the workemployee
emusername	CharField	150	-	Username of the workemployee
empassword	CharField	150	-	Password for the workemployee account
usertype	CharField	255	Default 'employee'	User type of the workemployee

Table Name: Work request

Table Description: to store parking details

Field Name	Data Type	Size	Constraint	Description
id	AutoField	-	Primary Key	Unique identifier for workrequest
address	CharField	255	-	Address for the workrequest
panchayath	CharField	255	-	Panchayath for the workrequest
wardno	CharField	255	-	Ward number for the workrequest
rationcardno	CharField	100	-	Ration card number for the workrequest
taxno	CharField	255	-	Tax number for the workrequest
area	CharField	255	-	Area for the workrequest

Table Name: Notification

Table Description: to store slot availability

Field Name	Data Type	Size	Constraint	Description
id	AutoField	-	Primary Key	Unique identifier for notifition
notification	TextField	255	-	Notification text
date	DateField	-	-	Date of the notification

Table Name: workcomplete

Table Description: to store mail details

Field Name	Data Type	Size	Constraint	Description
id	AutoField	-	Primary Key	Unique identifier for workcomplete
workentryid	CharField	255	-	Work entry ID
completeworkdetails	CharField	255	-	Completed work details
currentdetails	CharField	255	-	Current work details
todate	DateField	-	-	Date of completion for work complete

Table Name: complaint

Table Description: to store complaint details

Field Name	Data Type	Size	Constraint	Description
id	AutoField	-	Primary Key	Unique identifier for complaints
subject	CharField	255	-	Subject of the complaint
complaint	CharField	255	-	Complaint details
todate	DateField	-	-	Date of the complaint

Table Name: reply

Table Description: to store reply details

Field Name	Data Type	Size	Constraint	Description
id	AutoField	-	Primary Key	Unique identifier for reply2
replyid	CharField	255	-	Reply ID
reply	CharField	255	-	Reply details
redate	DateField	-	-	Date of the reply

Table Name: user

Table Description: to store user details

Field Name	Data Type	Size	Constraint	Description
id	AutoField	-	Primary Key	Unique identifier for user
name	CharField	100	-	Name of the user
email	EmailField	100	-	Email address of the user
password	CharField	100	-	Password for the user account

Table Name: employee attendance

Table Description: to store attendance details

Field Name	Data Type	Size	Constraint	Description
id	AutoField	-	Primary Key	Unique identifier for empattendence
employeeid	IntegerField	-	-	Employee ID
attendence	CharField	1	-	Attendence status (P: Present, A: Absent)
current_date	DateField	-	-	Date of attendence

Table Name:new work

Table Description: to store new work details

Field Name	Data Type	Size	Constraint	Description
id	AutoField	-	Primary Key	Unique identifier for newwork
fullname	CharField	255	-	Full name of the newwork applicant
age	IntegerField	-	-	Age of the newwork applicant
gender	CharField	15	-	Gender of the newwork applicant (male/female/other) aadhar
contactno	CharField	255	-	Contact number of the newwork applicant
empid	CharField	255	-	Employee ID associated with the newwork applicant

Table Name:Feedback

Table Description: to store feedback details

Field Name	Data Type	Size	Constraint	Description
id	AutoField	-	Primary Key	Unique identifier for feedbacks
feedback	TextField	255	-	Feedback details
currentdate	DateField	-	-	Date of the feedback
employeeid	CharField	25	-	Employee ID associated with the feedback

Table Name:work assign

Table Description: to store work assign details

Field Name	Data Type	Size	Constraint	Description
workrequestid	TextField	100	-	Unique identifier or code for a work request.
adsid	TextField	100	-	Identifier related to an Area Development Society (ADS).
currentdate	DateField	-	Nullable	Date when the work request or assignment was created or updated. Nullable field.

Table Name:Salary

Table Description: to store Salary details

Field Name	Data Type	Size	Constraint	Description
name_of_cardholder	CharField	100	-	Name of the cardholder
amount	IntegerField	-		Salary amount
card_number	IntegerField	-	MaxValueValidator	Card number, max value limited)
cvv	IntegerField	-	MaxValueValidator	Card CVV (max value limited)
card_expiry	CharField	25	-	Card expiry date
current_date	DateField	-		Date of the transaction
worker_id	CharField	100	-	Worker's ID

Table Name:Message

Table Description: to store Message

Field Name	Data Type	Size	Constraint	Description
message	TextField	255	-	The content of the message.
sender_id	ForeignKey	-	On Delete: CASCADE, Nullable	ForeignKey referencing the sender (workemployee). Null allowed.
receiver_id	ForeignKey	-	On Delete: CASCADE, Nullable	ForeignKey referencing the receiver (ads). Null allowed.
current_date	DateField	-	Nullable	Date when the message was sent. Null allowed.

Table Name:Reply

Table Description: to store Reply

Field Name	Data Type	Size	Constraint	Description
reply	TextField	255	-	The content of the reply message.
sender_id	ForeignKey	-	On Delete: CASCADE, Nullable	ForeignKey referencing the sender (ads). Null allowed.
receiver_id	ForeignKey	-	On Delete: CASCADE, Nullable	ForeignKey referencing the receiver (workemployee). Null allowed.
current_date	DateField	-	Nullable	Date when the reply message was sent. Null allowed.

4.4 NORMALIZATION

The entities along with their attributes can be stored in many different ways into a set of tables. The methods of arranging these attributes are called normal forms. The theory behind the arrangement of attributes into table is known as normalization theory. Normalization is a series of tests which we use against the data to eliminate redundancy and make sure that the data is associated with the correct table or relationship.It helps in,

- Minimization of duplication data.
- Providing flexibility to support different functional requirements.
- Enabling the model to be translated to database design

All relations in a relational database are required to satisfy the following conditions.

1. **Data in First Normal Form**
    - Remove repeating data from table
    - From the removed data, create one or more tables and relationships.

2. **Data in Second Normal Form**
    - Identify tables and relationships with more than one key.
    - Remove data that depends on only one part of the key.
    - From the removed data, create one or more tables and relationships.

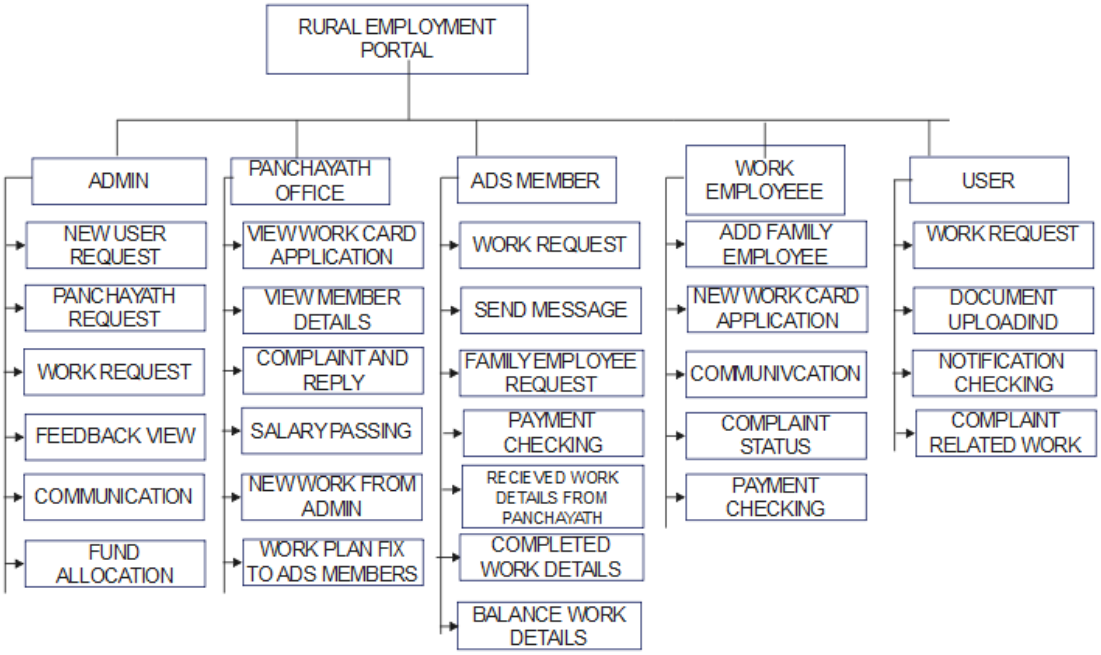
3. **Data in Third Normal Form**
    - Remove that depends on other hand in the table or relationship and not on the key.
    - From the removed data, create one or more tables and relationships.

Advantages of normalization are:

- Helps in reduction in the complexity of maintaining data integrity by removing the redundant data.
- It reduces inconsistency of data.
- Eliminate the repeating fields.
- Creates a row for each occurrence of a repeated field.
- Allows exploitation of column functions.

The second normal form has the characteristics of the first normal form and all the attributes must fully be dependent on the primary key. The proposed system is using second normal form as it is found most suitable.

4.5 Design of Each Sub System



4.6 UML **DIAGRAMS**

4.6.1 Use case Diagram

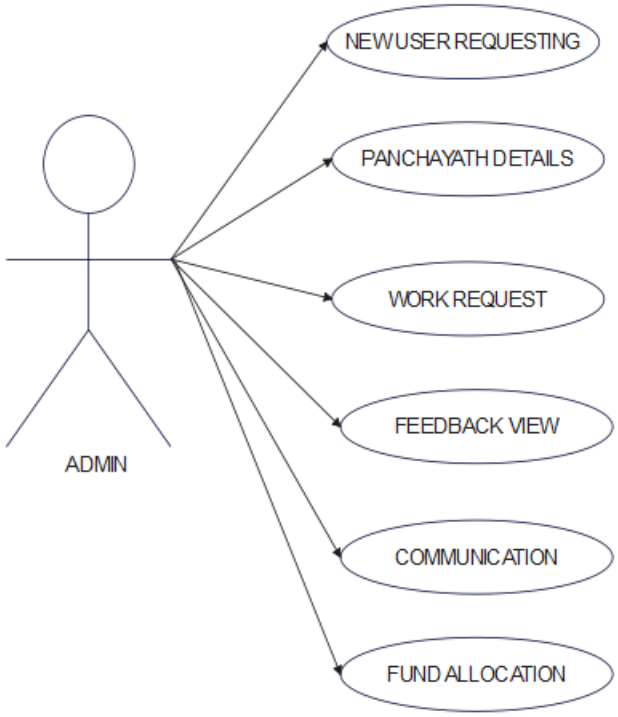


Figure 4.6.1 .1 Use Case Diagram for Admin

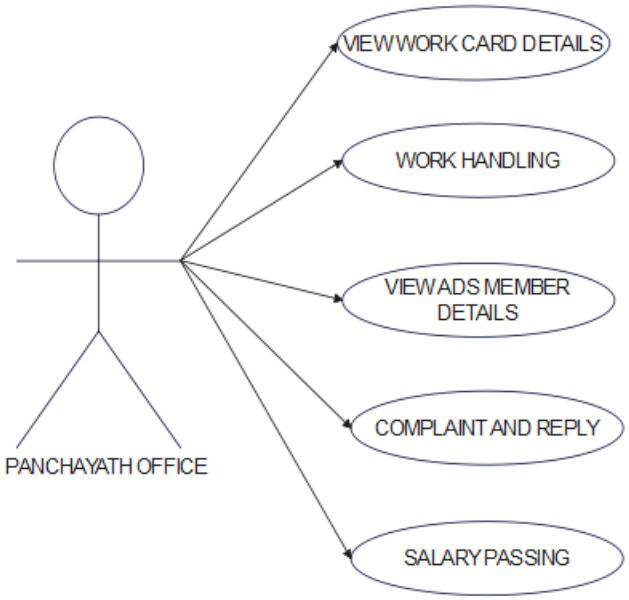


Figure 4.6.1.2 Use Case Diagram for Panchayath Office

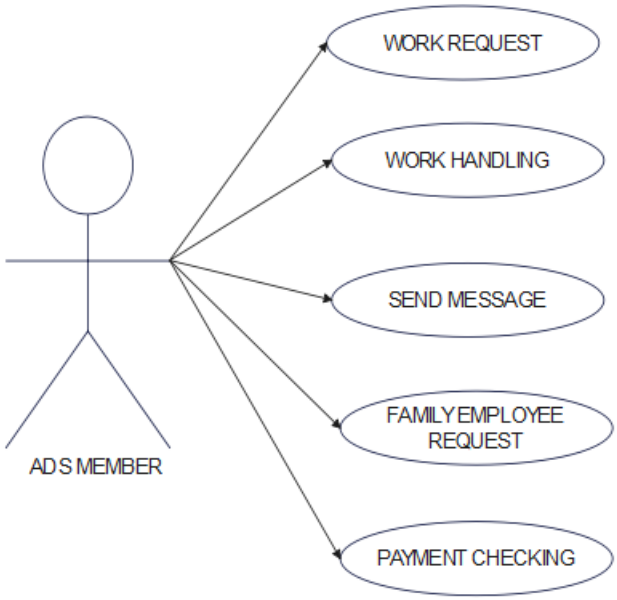


Figure 4.6.1.3 Use Case Diagram for ADS Member

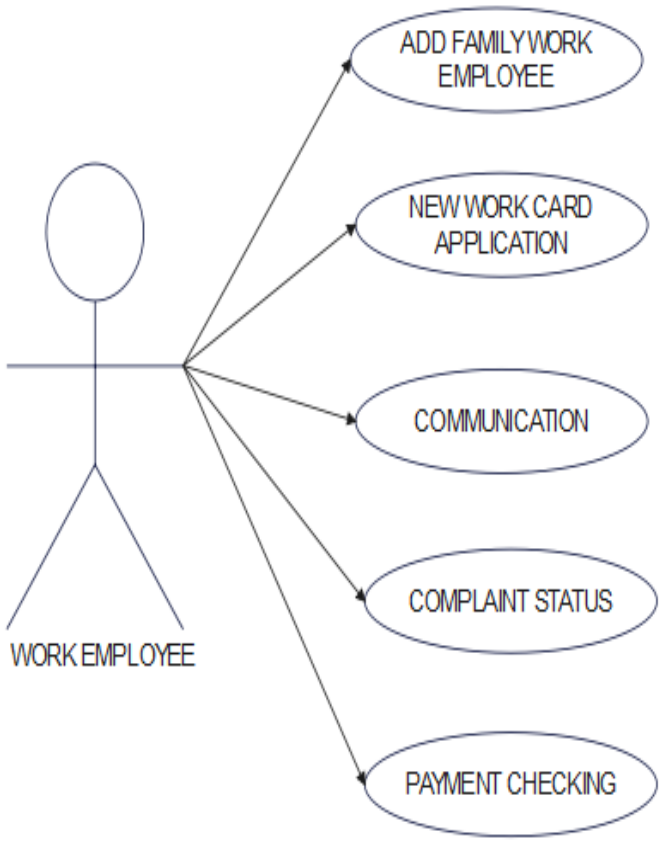


Figure 4.6.1.4 Use Case Diagram for Work Employee

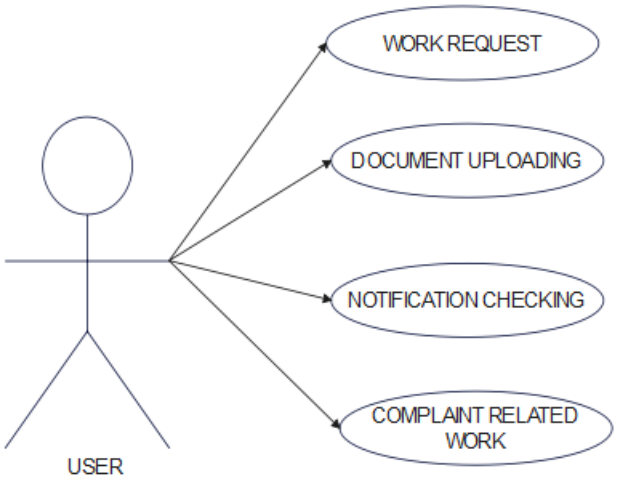


Figure 4.6.1.5 Use Case Diagram for User

4.7.1Sequence Diagram

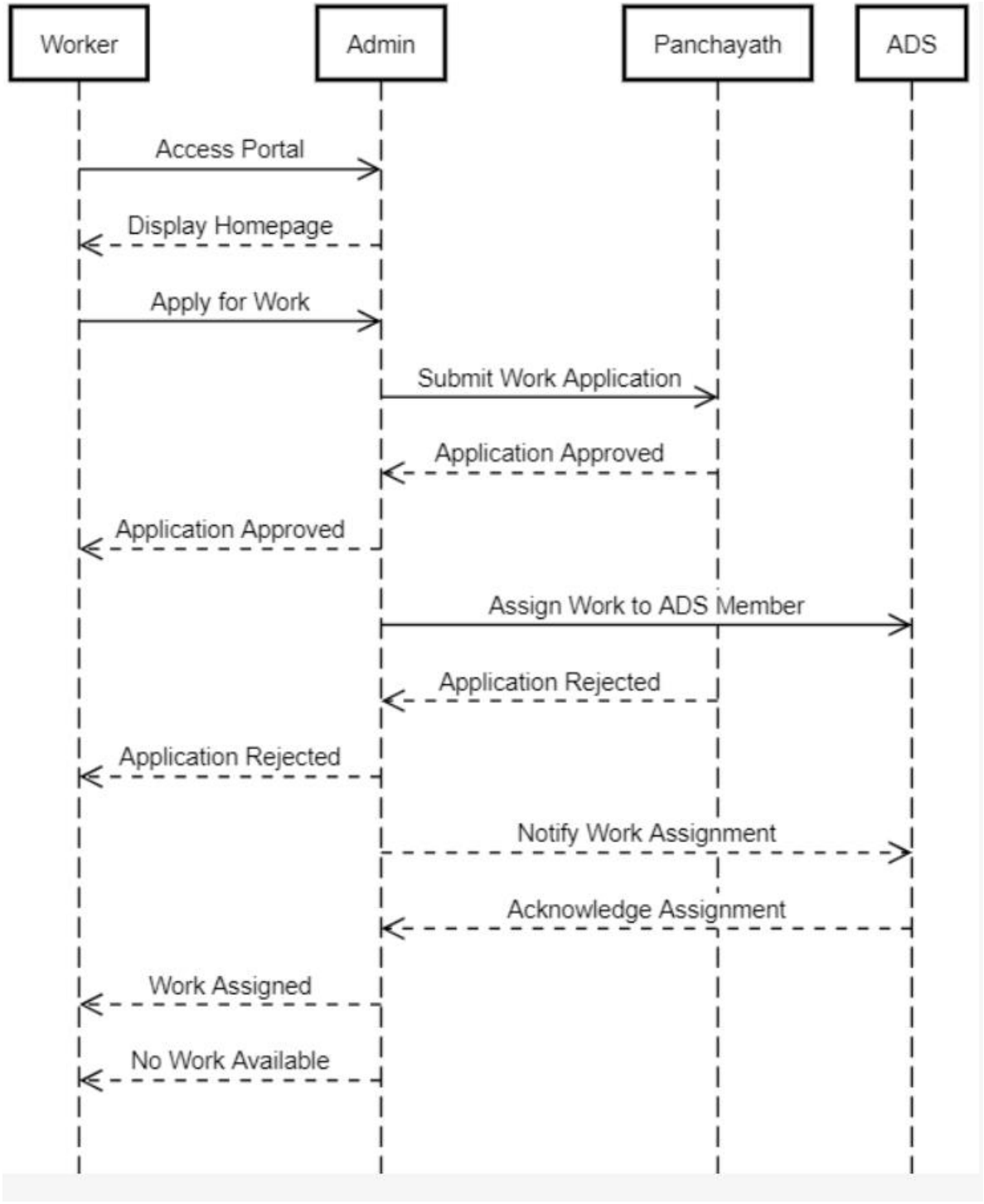


Figure 4.7.1.1 Sequence Diagram

CHAPTER-5

CODING

5.1- Language Study

HTML: stands for Hyper Text Mark-up Language. It is used to design web pages using a mark-up language. HTML is a combination of Hypertext and Mark-up language. Hypertext defines the link between web pages. A mark-up language is used to define the text document within the tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most mark-up languages (e.g., HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text. HTML is a mark-up language used by the browser to manipulate text, images, and other content, in order to display it in the required format. HTML was created by Tim Berners-Lee in 1991. The first-ever version of HTML was HTML 1.0, but the first standard version was HTML 2.0, published in 1995.

Elements and Tags: HTML uses predefined tags and elements which tell the browser how to properly display the content. Remember to include closing tags. If omitted, the browser applies the effect of the opening tag until the end of the page.

HTML page structure: The basic structure of an HTML page is laid out below. It contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.

<! DOCTYPE html>: This is the document type declaration (not technically a tag). It declares a document as being an HTML document. The doctype declaration is not case-sensitive.

<Html>: This is called the HTML root element. All other elements are contained within it

<Head>: The head tag contains the “behind the scenes” elements for a webpage. Elements within the head aren’t visible on the front-end of a webpage. HTML elements used inside the<html> element include:

- <Style>-This html tag allows us to insert styling into our webpages and make them appealing to look at with the help of CSS.
- <Title>The title is what is displayed on the top of your browser when you visit a website and contains the title of the webpage that you are viewing.

- <Base>-It specifies the base URL for all relative URL‟s in a document.
- <no script>– Defines a section of HTML that is inserted when the scripting has been turned off in the users browser
- <Script>-This tag is used to add functionality in the website with the help of JavaScript.
- <Meta>-This tag encloses the Meta data of the website that must be loaded every time the website is visited. For e.g.:- the metadata charset allows you to use the standard UTF-8 encoding in your website. T
- His in turn allows the users to view your webpage in the language of their choice. It is a self-closing tag.
- <Link>– The „link‟ tag is used to tie together HTML, CSS, and JavaScript. It is self-closing.

<Body>: The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front-end. An HTML document can be created using any text editor. Save the text file using .html or .html. Once saved as an HTML document, the file can be opened as a webpage in the browser.

5.2 Cascading Style Sheets:

Fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independently of the HTML that makes up each web page. It describes how a webpage should look: it prescribes colours, fonts, spacing, and much more. In short, you can make your website look however you want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser. While HTML uses tags, CSS uses rule sets. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

- CSS saves time: You can write CSS once and reuse the same sheet in multiple HTML pages.
- Easy Maintenance: To make a global change simply change the style, and all elements in all the webpages will be updated automatically.
- Search Engines: CSS is considered a clean coding technique, which means search engineswon‟t have to struggle to “read” its content.

- Superior styles to HTML: CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- Offline Browsing: CSS can store web applications locally with the help of an offline cache. Using this we can view offline websites.CSS Syntax: CSS comprises style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule set consists of a selector and declaration block
- The selector points to the HTML element you want to style
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.

5.3 Bootstrap:

Bootstrap is freely available for every. The main features of bootstrap is, it is very simple and easy to use, hug JavaScript plugins are available, easily design mobile friendly website.

- Easy to Use
- Mobile-Friendly
- Customizable Bootstrap
- Simple Integration
- Pre-styled Components
- Responsive Features
- Browser Compatibility
- Great Grid System
- Extensive list of Components
- Bundled JavaScript plugins
- Good Documentation

Features of bootstrap

Easy to use: Anybody with just basic knowledge of HTML and CSS can start using Bootstrap Responsive features: Bootstrap's responsive CSS adjusts to phones, tablets, and desktops

Mobile-Friendly: Mobile-first approach: In Bootstrap 3, mobile-first styles are part of the core framework

Simple Integration: Bootstrap can be simply integrated along with distinct other platforms and frameworks, on existing sites and new ones too and one more things you can also utilize particular elements of Bootstrap along with your current CSS.

Pre-styled Components: Bootstrap approaches with pre-styled components for alerts, dropdowns, nav bars, Customizable Bootstrap: The Bootstrap can be customized as per the designs of your project.

Browser compatibility: Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera) Great grid system: Bootstrap is built on responsive 12-column grids, layouts and components. Whether you need a fixed grid or a responsive, it's only a matter of a few changes.

Bundled JavaScript plugins: The components such as drop down menu are made interactive with the numerous JavaScript plugins bundled in the bootstrap package.

Extensive list of components: Whether you need drop down menus, pagination or alert boxes, Bootstrap has got your covered. Some of the components pre styled are; Dropdowns, Button Groups, Navigation Bar, Breadcrumbs, Labels & Badges, Alerts, Progress Bar, And many others.

Base styling for most HTML elements: A website has many different elements such as headings, lists, tables, buttons, forms, etc. The HTML elements for which styles are provided are; Typography Code, Tables, Forms, Buttons, Images, Icons

5.4 JavaScript:

JavaScript is a popular programming language. JavaScript features are flexible. Many open- source libraries are available. GitHub contains a large volume of JavaScript code by developers across the world. JavaScript works well in the front end and back end. JavaScript has a simple syntax. Without any settings, anyone can execute JavaScript programs and make them user-friendly. One individual having basic knowledge of HTML, CSS and coding can work with JavaScript.

Features of JavaScript

Scripting: JavaScript executes the client-side script in the browser. Interpreter The browser interprets JavaScript code. Event Handling Events are actions. JavaScript provides event-handling options.

Case Sensitive: In JavaScript, names, variables, keywords, and functions are case-sensitive.

Control Statements: JavaScript has control statements like if-else-if, switch case, and loop. Users can write complex code using these control statements.

Objects as first-class Citizens:

JavaScript arrays, functions, and symbols are objects which can inherit the Object prototype properties. Objects being first-class citizens mean Objects can do all tasks.

Supports Functional Programming:

JavaScript functions can be an argument to another function, can call by reference, and can assign to a variable.

Dynamic Typing: JavaScript variables can have any value type. The same variable can have a string value, an integer value, or any other.

Client-side Validations: JavaScript client-side validations allow users to submit valid data to the server during a form submission.

Platform Independent: JavaScript will run in the same way in all systems with any operating system.

Async Processing: JavaScript async-await and promise features provide asynchronous nature. As the processes run in parallel, it improves processing time and responsiveness.

Prototype-based: JavaScript follows 'Object. Prototype' functions instead of class inheritance.

5.5 Python:

Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, procedural programming approaches and provides dynamic memory allocation.

We have listed below a few essential features.

Easy to Learn and Use: Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.

Expressive Language: Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type print ("Hello World"). It will take only one line to execute, while Java or C takes multiple lines.

Interpreted Language: Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

Cross-platform Language: Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

Free and Open Source: Python is freely available for everyone. It is freely available on its official website www.python.org. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."

Object-Oriented Language: Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

Extensible: It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.

Large Standard Library: It provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting. There are various machine learning libraries, such as Tensor flow, Pandas, Numpy, Keras, and Pytorch, etc. Django, flask, pyramids are the popular framework for Python web development.

GUI Programming Support: Graphical User Interface is used for the developing Desktop application. PyQT5, Tkinter, Kivy are the libraries which are used for developing the web application.

Integrated: It can be easily integrated with languages like C, C++, and JAVA, etc. Python runs code line by line like C, C++ Java. It makes easy to debug the code.

Embeddable: The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well. It can embed other language into our code.

5.6 Django framework

Django is a web application framework written in Python programming language. It is based on MVT (Model View Template) design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement. By using Django, we can build web applications in very less time. Django is designed in such a manner that it handles much of configure things automatically, so we can focus on application development only. History :Django was design and developed by Lawrence journal world in 2003 and publicly released under BSD license in July 2005. Currently, DSF (Django Software Foundation) maintains

its development and release cycle. Django was released on 21, July 2005. Its current stable version is 2.0.3 which was released on 6 March, 2018.

Features of Django

Rapid Development: Django was designed with the intention to make a framework which takes less time to build web application. The project implementation phase is a very time taken but Django creates it rapidly.

Secure: Django takes security seriously and helps developers to avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery etc. Its user authentication system provides a secure way to manage user accounts and passwords.

Scalable: Django is scalable in nature and has ability to quickly and flexibly switch from small to large scale application project.

Fully loaded: Django includes various helping task modules and libraries which can be used to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds etc.

Versatile: Django is versatile in nature which allows it to build applications for differentdifferent domains. Now a day, Companies are using Django to build various types of applications like: content management systems, social networks sites or scientific computing platforms etc.

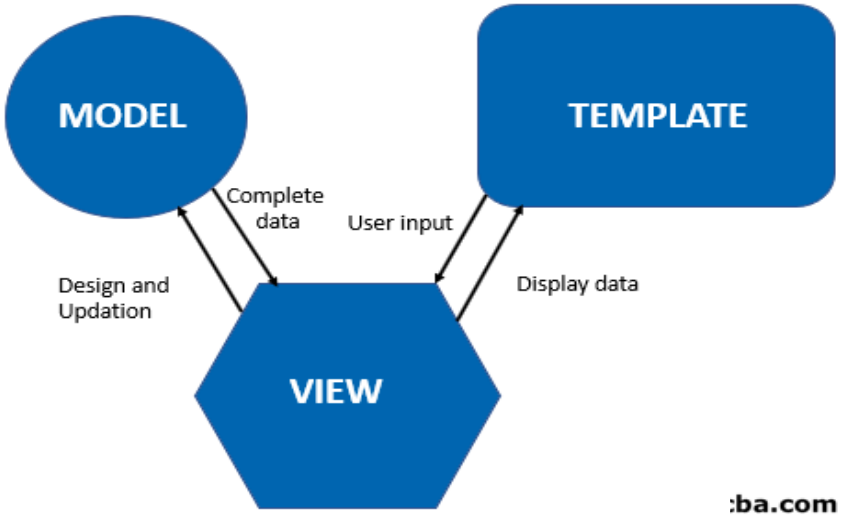
Open Source: Django is an open source web application framework. It is publicly available without cost. It can be downloaded with source code from the public repository. Open source reduces the total cost of the application development.

Vast and Supported Community: Django is a one of the most popular web framework. It has widely supportive community and channels to share and connect.

Django MVT

The MVT (Model View Template) is a software design pattern. It is a collection of three important components Model View and Template. The Model helps to handle database. It is a data access layer which handles the data. The Template is a presentation layer which handles User Interface part completely. The View is used to execute the business logic and interact with a model to carry data and renders a template. Although Django follows MVC pattern but maintains its own conventions. So, control is handled by the framework itself. There is no separate controller and complete application is based on Model View and Template. That’s

why it is called MVT application. Here, a user requests for a resource to the Django, Django works as a controller and check to the available resource in URL. If URL maps, a view is called that interact with model and template, it renders a template. Django responds back to the user and sends a template as a response.



Django Model

In Django, a model is a class which is used to contain essential fields and methods. Each model class maps to a single table in the database. Django Model is a subclass of django.db.models.Model and each field of the model class represents a database field (column). Django provides us a database-abstraction API which allows us to create, retrieve, update and delete a record from the mapped table. Model is defined in Models.py file. This file can contain multiple models.

Django Views

A view is a place where we put our business logic of the application. The view is a python function which is used to perform some business logic and return a response to the user. This response can be the HTML contents of a Web page, or a redirect, or a 404 error. All the view function are created inside the views.py file of the Django app

.

Django Templates

Django provides a convenient way to generate dynamic HTML pages by using its template system. A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted. In HTML file, we can't write python code because the code is only interpreted by python interpreter not the browser. We know that HTML is a static mark-up language, while Python is a dynamic programming language. Django template engine is used to separate the design from the python code and allows us to build dynamic web pages.

5.7. SQLite3 DATABASE

Being a very lightweight database management system, SQLite is very popular. No administration was required for operating a program in SQLite. However, it can only handle low to medium traffic HTTP requests. Also, the size of the database is usually restricted to 2GB. Even with these limitations, the SQLite advantages have gained more attention from the users. Some of the SQLite advantages are listed below:
Lightweight database management system: Easy to use as embedded software with various electronic devices. Better Performance:

- Very flexible.
- Fast reading and writing operations.
- Loads only the required data and not the entire file.
- Only overwrite the edited parts of a file and not the entire file
- Facilitates an efficient way for data storage.
- Variable column length of columns thus allows allocating only the spaces that a field needs.

Installation not needed:

- Easy to learn.
- No need to install.
- No Configuration Required.

Reliable:

- Contents are continuously updated.
- Less bug-prone than custom-written I/O code files
- Smaller queries than equivalent procedural codes.

Portable:

- Portable across all 32-bit and 64-bit operating systems and big- and little-endian architectures.
- Facilitates work on multiple databases on the same session at the same time. • Cross-platform DBMS.
- Available on both UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT).
- No compatibility issue with any programming languages.
- Facilitates API for a large range of programming languages.
- Facilitates simple and easy-to-use API

Accessible:

- Accessible through a wide variety of third-party tools.
- More likely to be recoverable if data has been lost.
- Data in SQLite lives longer than co
- Reduce Cost and Complexity:
- Free to use.
- Open-source.
- No license required to work with SQLite
- Doesn't require a different server process or system to operate and is thus Server less.
- No need for lengthy and error-prone procedural queries
- Content can be accessed and updated using concise SQL queries.

Open-Source.

- No license required to work with SQLite.
- Doesn’t require a different server process or system to operate and is thus Server less.
- No need for lengthy and error-prone procedural queries.
- Content can be accessed and updated using concise SQL queries

5.8 FUNCTIONAL DESCRIPTION

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Generally, functional requirements are expressed in the form "system must do requirement ".

Functional requirements for each of the use’s cases described below:

- Manager or supervisor should login with his credentials provided by admin for doing any activity.
- The system shall provide the admin to control all other users.
- The product can only be checkout if the product has been added.
- The box could only be transit if rack and bin were added.
- The box can be identified only by printing the box number.
- The supervisor can only login if the manager adds the supervisor.
- The employee can only login if the manager adds the employee.
- The rack could only be added if the building was added

System Coding

```
from django.shortcuts import render
from django.shortcuts import redirect
from django.http import HttpResponse
from django.template import loader
from . forms import pub1
from . forms import panchayath1
from . models1 import pub
from . models2 import panchayath
from . models3 import ads
from . forms import ads1
from . models4 import workemployee
from . forms import emp1

def Ho(request):
template=loader.get_template('index.html')
return HttpResponse(template.render())

def publ(request):
if request.method=="POST":
fm=pub1(request.POST)
if fm.is_valid():
fm.save()
return redirect('home')
else:
fm=pub1()
return render(request,'public.html',{'fm':fm})

def vi(request):
fm=pub.objects.all().values()
return render(request,'publicview.html',{'fm':fm})
def log(request):
```

```
template=loader.get_template('login.html')
return HttpResponse(template.render())
def admin1(request):
template=loader.get_template('admin.html')
return HttpResponse(template.render())
def pancha(request):
if request.method=='POST':
gf=panchayath1(request.POST)
if gf.is_valid():
gf.save()
return redirect('home')
else:
gf=panchayath1()

return render(request,'panchayath.html',{'pa':gf})



ph=panchayath.objects.all().values()
return render(request,'panchayathview.html',{'ph':ph})

def adsr(request):
if request.method=='POST':
ad=ads1(request.POST)
if ad.is_valid():
ad.save()
return redirect('home')
else:
ad=ads1()
return render(request,'ads.html',{'a':ad})

def ac(request):
```

```
adr=ads.objects.all().values()
return render(request,'adsview.html',{'q':adr})

def we(request):
if request.method=='POST':
emw=emp1(request.POST)
if emw.is_valid():
emw.save()
return redirect('home')
else:
emw=emp1()
return render(request,'employee.html',{'p':emw})
def emp(request):
emv=workemployee.objects.all().values()
return render(request,'employeeview.html',{'r':emv})


def adsregi(request):
adr=ads.objects.all().values()
return render(request,'adsregistered.html',{'q':adr})

def empre(request):
emv=workemployee.objects.all().values()
return render(request,'employeeregistered.html',{'r':emv})
def wre(request):
if request.method=='POST':
wr=workre(request.POST)
if wr.is_valid():
wr.save()
else:
wr=workre()
return render(request,'workrequest.html',{'p':wr})
```

```
def wor(request):
emv=workrequest.objects.all().values()
return render(request,'workrequestregistered.html',{'m':emv})
def edit(request,id):
if request.method=='POST':
mydata=workrequest.objects.get(id=id)
fm=workre(request.POST,instance=mydata)
if fm.is_valid():
fm.save()
return redirect('workrequestview')
else:
mydata=workrequest.objects.get(id=id)
fm=workre(instance=mydata)
return render(request,'workrequestedit.html',{'p':fm})

def worv(request):
emv=workrequest.objects.all().values()
return render(request,'workrequestview.html',{'m':emv})

def delete(request,id):
mydata=workrequest.objects.get(id=id)
mydata.delete()
return redirect('workrequestview')
def notific(request):
if request.method=='POST':
nf=notifi(request.POST)
if nf.is_valid():
nf=nf.save(commit=False)
nf.date=date.today()
nf.save()
return redirect('notification')
else:
```

```
nf=notifi()
return render(request,'notification.html',{'jw':nf})
def novi(request):
fm=notifition.objects.all().values()
return render(request,'notificationview.html',{'fm':fm})

def noviw(request):
fm=notifition.objects.all().values()
return render(request,'notificationviewreg.html',{'fm':fm})

def noviw1(request):
fm=notifition.objects.all().values()
return render(request,'notificationviewreg1.html',{'fm':fm})

def noedit(request,id):
if request.method=='POST':
mdata=notifition.objects.get(id=id)
fm=notifi(request.POST,instance=mdata)
if fm.is_valid():
fm.save()
return redirect('ee')
else:
mdata=notifition.objects.get(id=id)
fm=notifi(instance=mdata)
return render(request,'notificationedit.html',{'fm':fm})

def nodelete(request,id):
mdata=notifition.objects.get(id=id)
mdata.delete()
return redirect('ee')

def empree(request):
```

```
    emv=workemployee.objects.all().values()
    return render(request,'employeeregistered2.html',{'r':emv})



    def pavi2(request):
    ph=panchayath.objects.all().values()
    return render(request,'panchayathview2.html',{'ph':ph})
```

HTML Code

```
<!DOCTYPE html>
<html lang="en">
  {% load static %}
<head>
    <title>Rural Employment Portal</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" type="image/png" href="{% static 'Login_v16/images/icons/favicon.ico' %}

    <link rel="stylesheet" type="text/css" href="{% static 'Login_v16/vendor/bootstrap/css/bootstrap.min.css' %}">
    <link rel="stylesheet" type="text/css" href="{% static 'Login_v16/fonts/font-awesome-4.7.0/css/font-awesome.min.css' %}">
    <link rel="stylesheet" type="text/css" href="{% static 'Login_v16/fonts/Linearicons-Free-v1.0.0/icon-font.min.css' %}">
    <link rel="stylesheet" type="text/css" href="{% static 'Login_v16/vendor/animate/animate.css' %}">
    <link rel="stylesheet" type="text/css" href="{% static 'Login_v16/vendor/css-hamburgers/hamburgers.min.css' %}">
    <link rel="stylesheet" type="text/css" href="{% static
```

```
'Login_v16/vendor/animsition/css/animsition.min.css' %}">
	<link rel="stylesheet" type="text/css" href="{% static 'Login_v16/vendor/select2/select2.min.css' %}">

	<link rel="stylesheet" type="text/css" href="{% static 'Login_v16/vendor/daterangepicker/daterangepicker.css' %}">
	<link rel="stylesheet" type="text/css" href="{% static 'Login_v16/css/util.css' %}">
	<link rel="stylesheet" type="text/css" href="{% static 'Login_v16/css/main.css' %}">
</head>
<body>

	<div class="limiter">
		<div class="container-login100" style="background-image: url('/static/Login_v16/images/bg-01.jpg');">
			<div class="wrap-login100 p-t-30 p-b-50">
				<span class="login100-form-title p-b-41">
					Account Login
				</span>

				<form class="login100-form validate-form p-b-33 p-t-5" method="post">
					{% csrf_token %}
					{{ form.as_p }}

					<div class="container-login100-form-btn m-t-32">
						<button class="login100-form-btn">
							Login
						</button>
					</div>

				</form>
```

```
                    </div>
                </div>
            </div>



            <div id="dropDownSelect1"></div>

            <script src="{% static 'Login_v16/vendor/jquery/jquery-3.2.1.min.js' %}"></script>
            <script src="{% static 'Login_v16/vendor/animsition/js/animsition.min.js' %}"></script>
            <script src="{% static 'Login_v16/vendor/bootstrap/js/popper.js' %}"></script>
            <script src="{% static 'Login_v16/vendor/bootstrap/js/bootstrap.min.js' %}"></script>
            <script src="{% static 'Login_v16/vendor/select2/select2.min.js' %}"></script>
            <script src="{% static 'Login_v16/vendor/daterangepicker/moment.min.js' %}"></script>
            <script src="{% static 'Login_v16/vendor/daterangepicker/daterangepicker.js' %}"></script>
            <script src="{% static 'Login_v16/vendor/countdowntime/countdowntime.js' %}"></script>
            <script src="{% static 'Login_v16/js/main.js' %}"></script>

    </body>
    </html>
```

CHAPTER – 6

TESTING

System Testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It certifies that the whole set of program hang together. System testing requires a test plan that consists of several keys, activities and steps to run program, string, system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system.

TESTING OBJECTIVES

- Testing is the process of correcting a program with intend of finding an error. \
- A good test is one that has a high probability of finding a yet undiscovered error.
- A successful test is one that uncovers an undiscovered error.

6.1Levels of Testing

6.1.1 Unit Testing

In this testing we test each module individually and integrate the overall system. Unit testing focuses verification efforts on the smaller unit of software design in the module. This is also known as „module" testing. The modules of the system are tested separately. The testing is carried out during programming stage itself. In this testing step each module is found to work satisfactory as regard to the expected output from the module. There are some validation checks for verifying the data input given by the user. It is very easy to find error and debug the system.

6.1.2 Integration Testing

Data can be lost across an interface; one module can have an adverse effect on the other sub functions when combined by May not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. This testing was done with sample data. The need for integrated test is to find the overall system performance.

6.1.3 Black Box Testing

This testing attempts to find errors in the following areas or categories: Incorrect or missing functions, interface errors, errors in data structures, external database access, performance errors and initialization and termination errors.

6.1.4 Validation Testing

At the culmination of Black Box testing, software is completely assembled as a package, interface errors have been uncovered and corrected and final series of software tests, validation tests begins. Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably accepted by the customer.

After validation test has been conducted one of the two possible conditions exists.

- The function or performance characteristics confirm to specification and are accepted.
- A deviation from specification is uncovered and a deficiency list is created

6.1.5 Output Testing

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it doesn't produce the required data in the specific format. The output displayed or generated by the system under consideration is tested by, asking the user about the format displayed. The output format on the screen is

6.1.6 White Box Testing

White box testing is a testing case design method that uses the control structure of the procedural design to derive the test cases. The entire independent path in a module is exercised at least once. All the logical decisions are exercised at least once. Executing all the loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. In our project testing was conducted at every step. Initially each module was tested separately to check whether they gave the desired output for the given input. The forms used to enter data by user were validated and appropriate error messages were displayed if incorrect data was entered. Once the data was entered correctly, the processing was done and testing was done to check whether the correct output was obtained. Once the test cases were conducted successfully for each module, the modules were integrated together as a single system. After integration, the test cases were again applied to check whether the entire system as a whole produced the desired output. At times, the test cases failed and the shortcomings were noted down and appropriate corrections were done. Once the integration testing was performed correctly, output testing was done and it did not result in any change or correction in the system. Black box testing and white box testing was also conducted successfully. All the

loops, decisions, relations were executed at least once before giving it to the users for testing. In black box testing, it was checked whether the data in the proper format was stored in the database or not. Also, it was checked whether the interfaces were working properly or not. On successful completion of these tests, the system was then given to undergo user acceptance testing where the users entered test data to check whether the correct output was obtained. The users were satisfied with the output and thus the testing phase was completed successfully.

6.1.7 Test Data and Results

The primary goal of software implementation is the production of source code that is easy to read and understand. Clarification of source code helps in easier debugging, testing and modification. Source code clarification is enhanced by structural coding techniques, by good coding style, by appropriate supporting documents, by good internal comments and by the features provided in the modern programming language. In our implementation phase, source code contains both global and formal variables. It contains predefined functions as well as the user defined functions. The result of the new system is compared with old system and supposes if the result is wrong the error must be debugged.

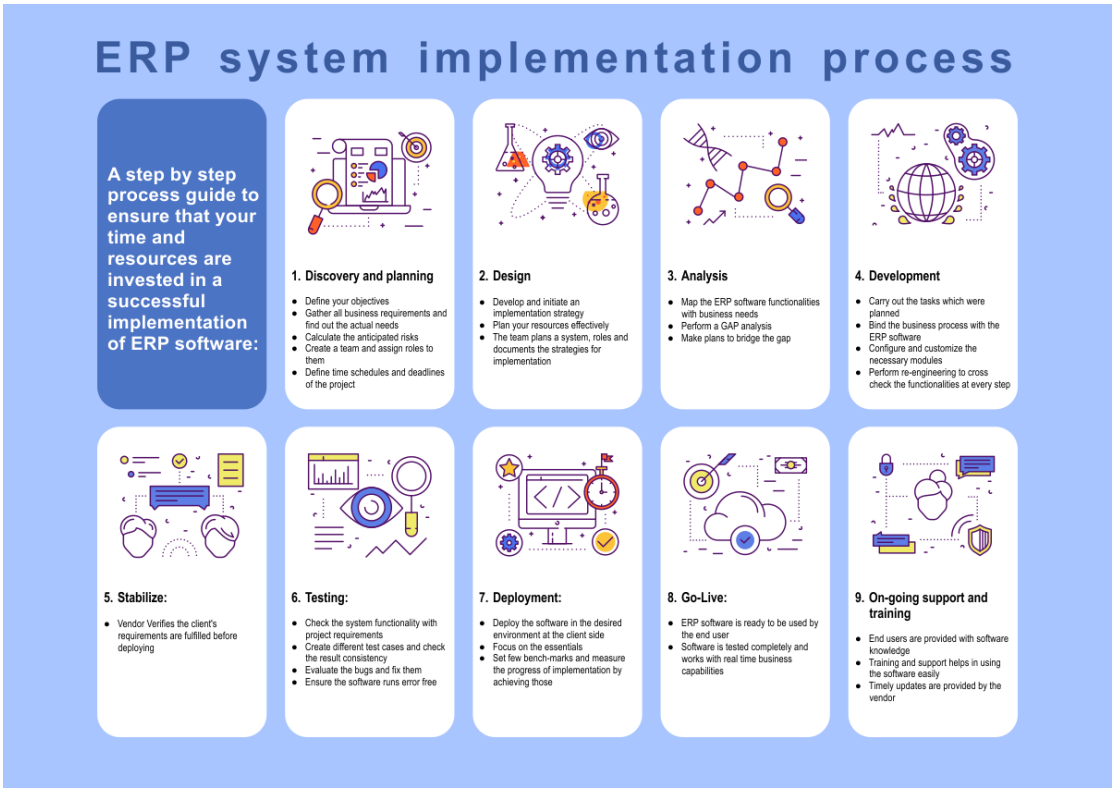
After the acceptance of the system by the user, the existing system should be replaced by this system. Any user handles this package very easily. It does not require any intensive training for the user. Procedures and functions involved in this system are very simple that anyone can understand and correspondingly act to the system with no difficulty

CHAPTER – 7

IMPLEMENTATION

7.1 Implementation of Proposed System

Here's a list of ERP Implementation steps that is sure to soar your chances of success. Follow them to ensure that your ERP implementation process is not an intimidating one.



The final and important phase in the system life cycle is the implementation of the new system. The term implementation has different meanings ranging from the conversion of a basic application to a complete replacement of a computer system. The procedure, however, is virtually the same. Implementation includes all those activities that take place to convert from old system to new. The new system may be totally new replacing existing system, manual or automated, or it may be a major modification to an existing system. The method of implementation and time scale to be adopted is found out initially. Next the system is tested properly and at the same time users are trained in the new procedure. Proper implementation is essential to provide a reliable system to meet organization requirements. Successful implementation may not guarantee improvement in the organization using the new system, but it will prevent improper installation.

The implementation involves following things:

- Careful planning

- Investigation of the system considerations.
- Design the method to achieve the changeover.
- Evaluation of change over method.

Implementation of a new system requires the operating staff installing the software and creating computer files. There are many ways in which this can be achieved. The most common methods are the following:

- Direct change over.
- Parallel running.
- Pilot running change over.

The creation of the designed system takes place in the implementation phase.

This phase activities do the following:

- Development of phase overview.
- Preparing for implementation.
- Computer program development.
- Development phase report and overview.

It also performs activities like writing, testing, debugging and documenting the programs.

There are three types of implementations:

- Implementation of a computer system to replace a manual system. The problems Encountered are converting files, training users, creating accurate files, and verifying printouts for integrity.
- Implementation of a new computer system to replace an existing one. This is usually a difficult conversion. If not properly planned, there can be many problems. Some large computer systems have taken as long as a year to convert.

CHAPTER – 8

SECURITY BACKUP AND RECOVERY MECHANISMS

Online Help

Simply speaking, a backup is a copy of data. This copy includes important parts of database such as the control file and data files. A backup is a safeguard against unexpected data loss and application errors, should we lose the original data, can use the backup to make it available again. After developing our portal, a proper backup copy is stored in a separate system or medium, like CD and drives, from the primary data to protect against the possibility of data loss due to primary hardware or software failure. Recovery from a backup typically involves restoring the data to the original location, or to an alternate location where it can be used in place of the lost or damaged data. A database is a very huge system with lose of data and transaction. The transaction in the database is executed at each seconds of time and is very critical to the database. If there is any failure or crash while executing the transaction, then it expected that no data is necessary to revert the changes of transaction to previously committed point. That means, any transaction in the system cannot be left at the stage of its failure. It should either be completed fully or rolled back to the previous consistent state.

User Manuals

A user manual is a technical document with a quite specific purpose to help non-technicalpeople pinpoint and solve problems without assistance. Since user manual translatewhat's not comprehend to a language for everyone to understand, they are essential in technical sectors and most commonly associated with software and hardware, IT systems.In our application user manual includes images and notes relating every functions inorder to help the user to understand features of our system. Admin can add complaints against hotel or their food which can help the hotel managements for trying their best solutions. The hotels can add their images about menus which can be viewed by the user and this also helps them immensely. It also includes screenshots. The main working ofthe system and its data flow is all elaborately described with the help of simplified diagrams and descriptions.

CHAPTER – 9

CONCLUSION

The Rural Employment Portal serves as a comprehensive and user-centric platform aimed at uplifting the livelihoods of individuals residing in rural areas. With its multifaceted features and modules, the portal seeks to bridge the gap between unskilled laborers and meaningful employment opportunities. By providing a robust framework that connects workers, Panchayath offices, Area Development Society (ADS) members, and administrators, the portal revolutionizes the way rural employment is managed and accessed.

Through the administrator module, the portal enables effective oversight and management of various functions, including work allocation, fund allocation, and user communication. Panchayath offices play a pivotal role in verifying and processing work requests, while ADS members facilitate work verification and reporting, ensuring transparency and accountability.

CHAPTER – 10

FUTURE ENHANCEMENT

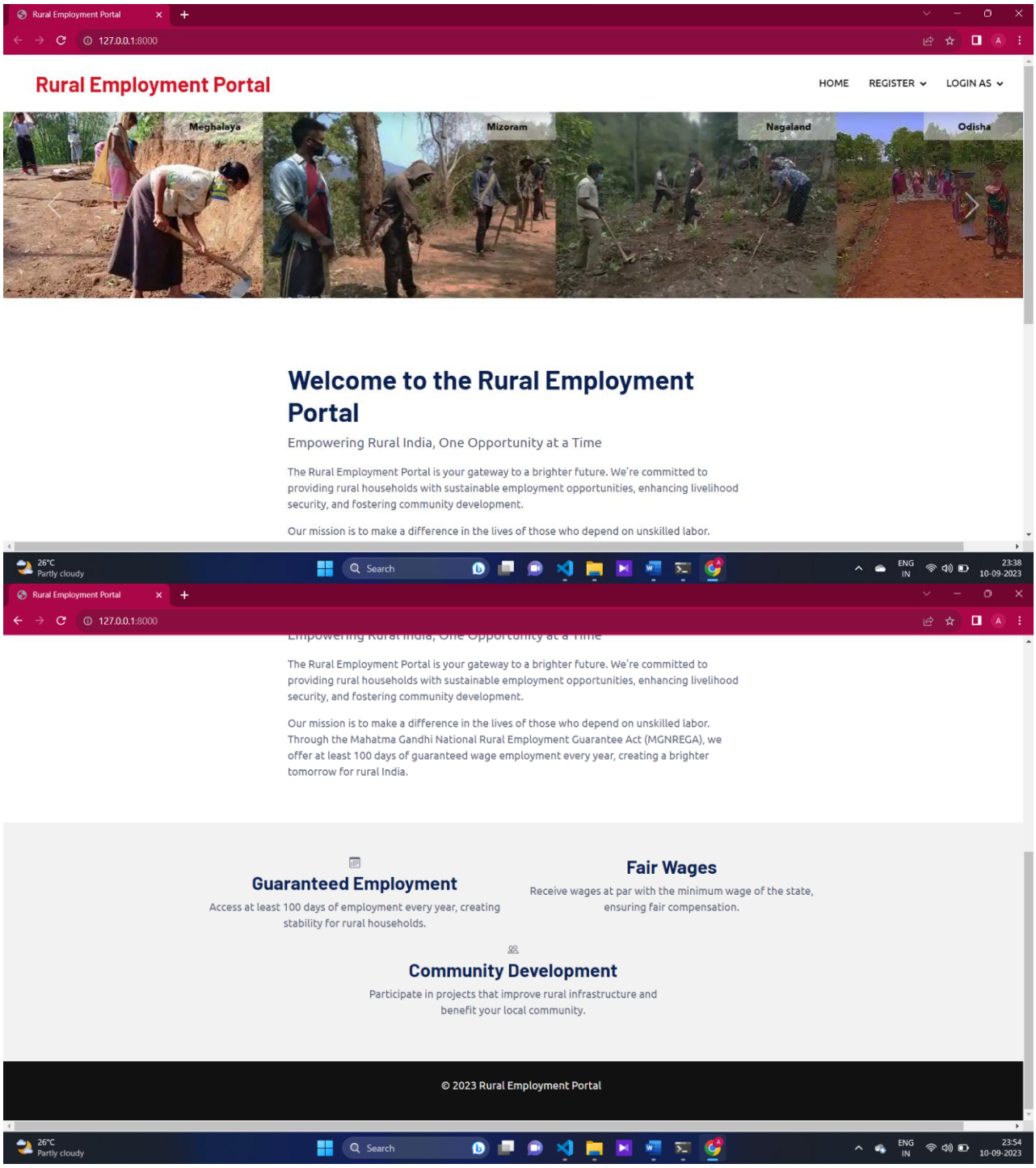
The changing requirements and the fast growth of technology will surely force the system to impose some changes forecasting those needs. Rural employment portal is developed in such a way that the enhancement can easily made without making major changes to the application. Though the system is working in certain assumptions it can be modified easily to anykind of requirements.

- Mobile Application Development
- Geolocation Integration
- Real-Time Communication
- Multi-Language Support

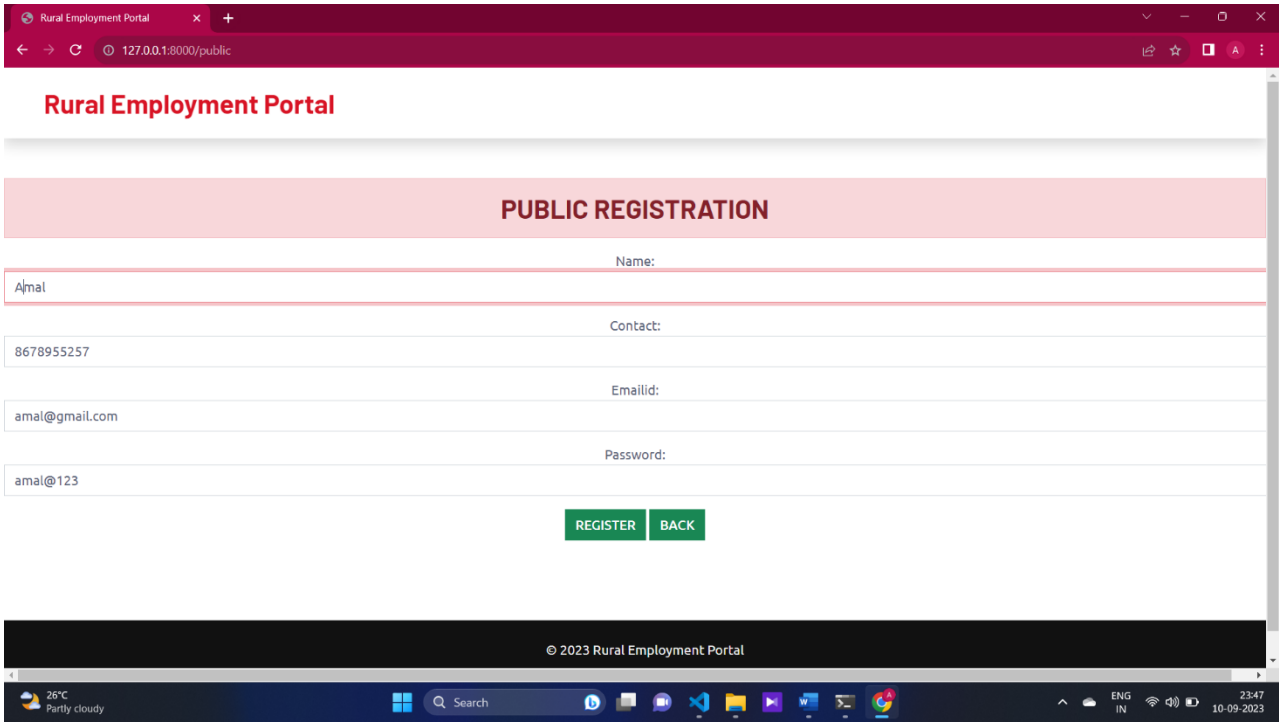
APPENDIX

INPUT AND OUTPUT FORMS

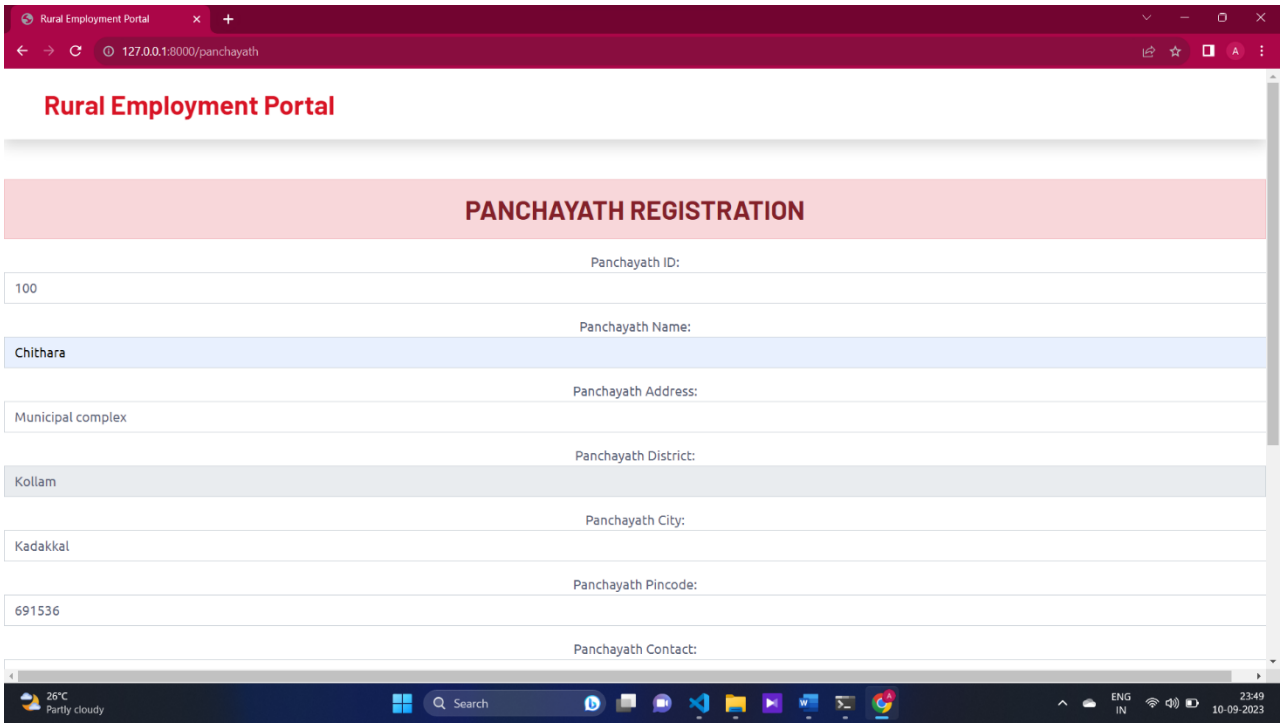
Home Page



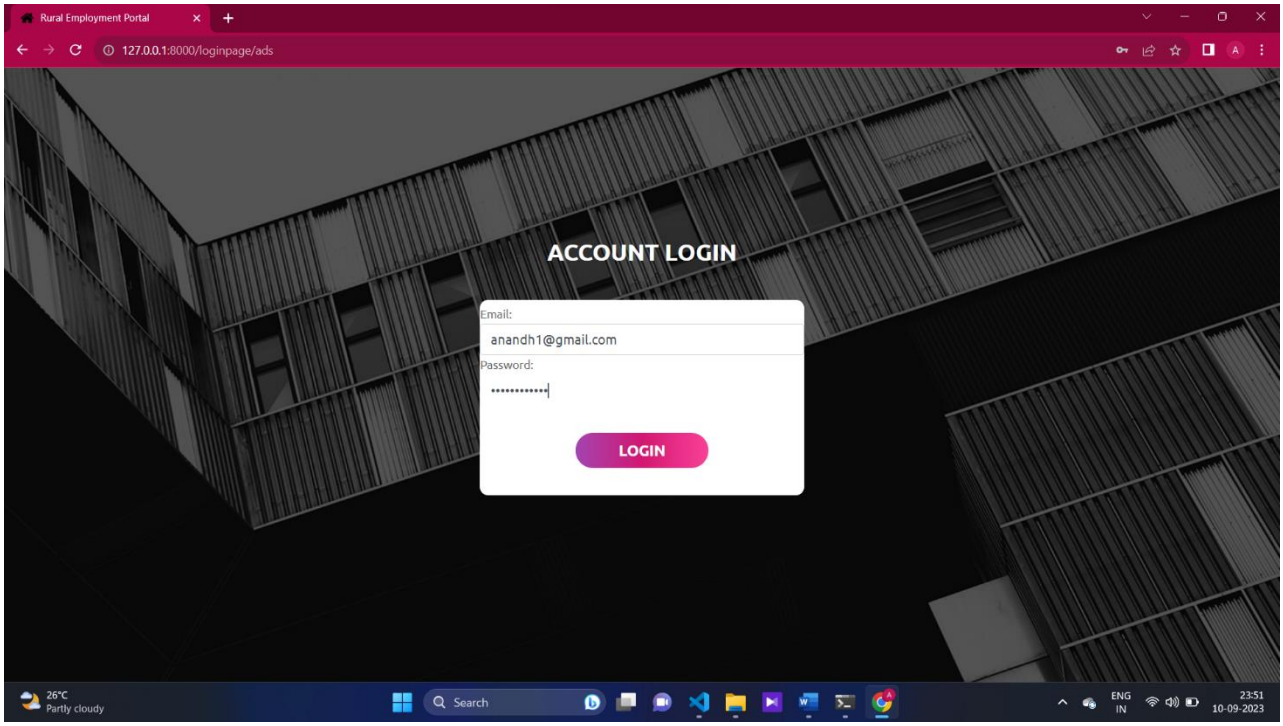
Public Register Page



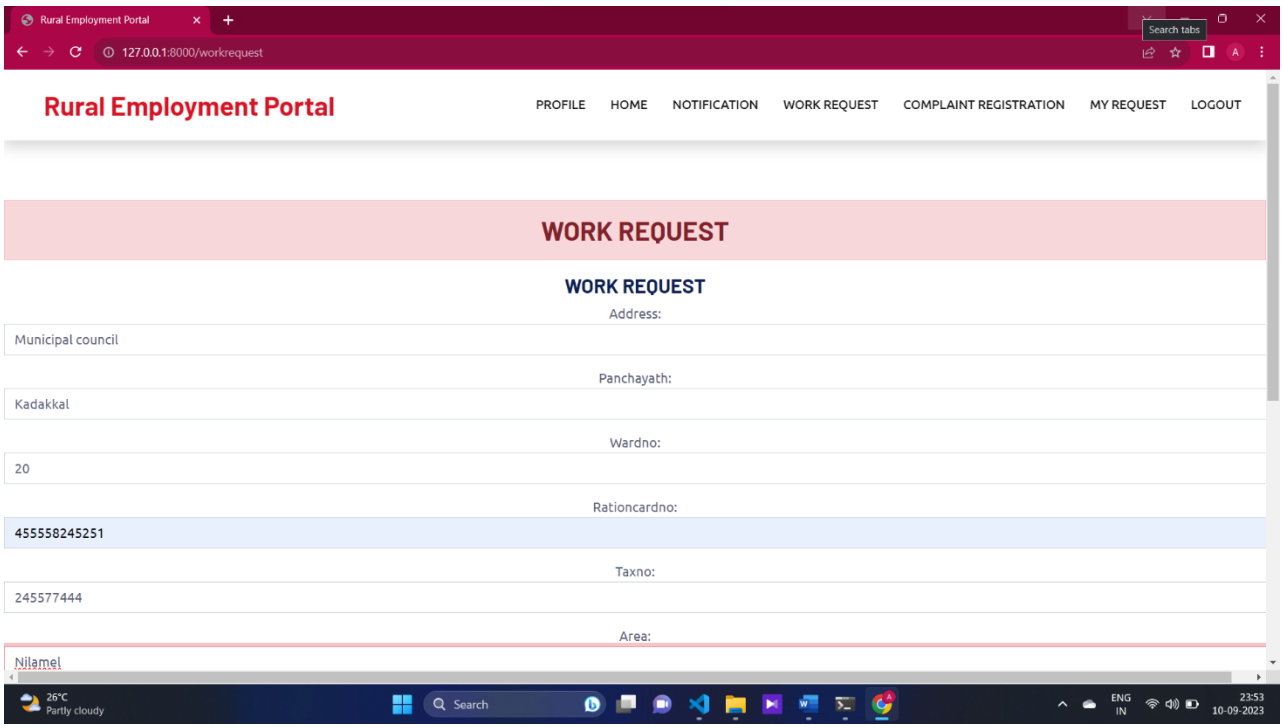
Panchayath Register Page



Login Page



Work Request Register



BIBLIOGRAPHY

1. BOOKS

- **Kenneth. S. Rubin**, Essential Scrum: A Practical Guide to the Most Popular Agile Process, First Edition.
- **Allen B. Downey**, Think Python: An Introduction to Software Design, Second Edition
- **OpenCV (2019)** OpenCV—Python Core— [online]

2. WEBSITES

- https://chat.openai.com/
- https://www.javatpoint.com/python-tutorial
- https://www.tutorialspoint.com/python/index.html
- https://api.jquery.com
- https://www.djangoproject.com
- https://www.tutorialspoint.com/mongodb/index.html
- https://searcherp.techtarget.com/definition/warehouse-management-system-WMS