MBC- BUILDING AND CONSTRUCTION SITE MANAGMENT

PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of the degree of Master of Computer Applications of A P J Abdul Kalam Technological University

Submitted by:

AMAL MANOJ (SJC23MCA-2011)

MASTER OF COMPUTER APPLICATIONS

ST. JOSEPH’S COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI

CHOONDACHERRY P.O, KOTTAYAM

KERALA

NOVEMBER 2024

ST. JOSEPH’ S COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI

(An ISO 9001: 2008 Certified College)

CHOONDACHERRY P.O, KOTTAYAM KERALA

CERTIFICATE

This is to certify that the project work entitled **MBC- BUILDING AND CONSTRUCTION SITE MANAGEMENT** submitted by **Amal Manoj** student of **Third** semester **MCA** at **ST. JOSEPH’S COLLEGE OF ENGINEERING AND TECHNOLOGY**, **PALAI** in partial fulfilment for the award of Master of Computer Applications is a bonafide record of the project work carried out by him under guidance and supervision. This report in any form has not been submitted to any other university or institute for any purpose.

Mr. Akhil Sekharan	**Dr. Rahul Shajan**	**Dr. Rahul Shajan**
(Project Guide)	**(Project Coordinator)**	**(Head , Department of Computer Applications)**

Submitted for the Viva-Voce Examination held on ____________________________

Examiner 1:

Examiner 2:

DECLARATION

Me **Amal Manoj** , do hereby declare that the project titled **"MBC"** is a record of work carried out under the guidance **Mr. Akhil Sekharan**, Asst.Professor, Department of Computer Applications, SJCET, Palai as per the requirement of the curriculum of Master of Computer Applications Programme of A P J Abdul Kalam Technological University, Thiruvananthapuram. Further, I also declare that this report has not been submitted, full or part thereof, in any University / Institution for the award of any Degree / Diploma.

Place: Choondacherry Amal Manoj

Date: (SJC23MCA-2011)

ACKNOWLEDGEMENT

The success of any project depends largely on the encouragement and guidelines of many others. I would like to take this opportunity to express my gratitude to those people who have been instrumental in the successful completion of this project.

First and foremost, I give all glory, honor and praise to **God Almighty** who gave me wisdom and enabled me to complete the project successfully.

I also express sincere thanks, from the bottom of my heart, to my parents for their encouragement and support in all my endeavors and especially in this project.

Words are inadequate to express my deep sense of gratitude to **Dr. V. P Devasia, Principal, SJCET, Palai** for allowing me to utilize all the facilities of our college and for his encouragement.

I extend my sincere gratitude to **Dr. Rahul Shajan, HOD, Department of Computer applications**, **SJCET, Palai** who has been a constant source of inspiration and without his tremendous help and support this project would not have been materialized.

I owe a particular debt of gratitude to my project guide, **Mr.Akhil Sekharan , Asst. Professor, Department of Computer Applications, SJCET, Palai** for all the necessary help and support that he has extended to me. His valuable suggestions, corrections, and the sincere efforts to accomplish my project even under a tight time schedule were crucial in the successful completion of this project.

I extend my sincere thanks to all our teachers and non-teaching staff of SJCET, Palai for the knowledge they have imparted to me over the last three years.

I would also like to express my appreciation to all my friends for their comments, help and support.

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS

VISION

To emerge as a centre of excellence in the field of computer education with distinct identity and quality in all areas of its activities and develop a new generation of computer professionals with proper leadership, commitment, and moral values.

MISSION

- ❖ Provide quality education in Computer Applications and bridge the gap between the academia and industry.
- ❖ Promoting innovation research and leadership in areas relevant to the socio-economic progress of the country.
- ❖ Develop intellectual curiosity and a commitment to lifelong learning in students, with societal and environmental concerns.

PEO'S (Program Educational Objectives)

1. MCA Graduates will be able to progress career productively in software industry, academia, research, entrepreneurship pursuits, government, consulting firms and other IT enabled services.
2. MCA Graduates will be able to achieve peer-recognition as an individual or in a team by adopting ethics and professionalism, and communicate effectively to excel well in crisis and inter-disciplinary teams.

3. MCA Graduates will be able to continue life-long professional development in computing and in management that contributes in self and societal growth.

20MCA245 MINI PROJECT

Co No	CO	Blooms Category
CO1	Identify a real-life project which is useful to society / industry	Level 2: Understand
CO2	Interact with people to identify the project requirements	Level 3: Apply
CO3	Apply suitable development methodology for the development of the product / project	Level 3: Apply
CO4	Analyse and design a software product / project	Level 4: Analyse
CO5	Test the modules at various stages of project development	Level 5: Evaluate
CO6	Build and integrate different software modules	Level 6: Create
CO7	Document and deploy the product / project	Level 3: Apply

SYNOPSIS

The **Construction Site Management System** is a comprehensive and user-focused platform designed to revolutionize construction project management by enabling efficient tracking, communication, and transparency. Tailored for clients who cannot regularly visit the site, this system offers remote access to real-time data on project progress, budget allocation, workforce activity, and materials management. This platform empowers clients to stay informed, make approvals, and gain an accurate view of construction activities, all from a distance.

Through a centralized dashboard, clients can navigate through active worksites, each with detailed daily logs and updates. They can review and approve budget allocations, attendance records, and material purchases, providing essential oversight. By increasing transparency in daily operations, such as budget usage, work hours, and resource allocation, the system helps prevent potential fraud and mismanagement, fostering trust between clients and contractors. To ensure a seamless experience, the platform features a straightforward scheduling system for regular updates and feedback, allowing for effective communication and accountability. Contractors, acting as the admin, can oversee worksite details, manage workforce assignments, and track resources while keeping clients informed. Automated notifications and reminders help both clients and contractors stay up-to-date on project milestones and approval requests.

With a strong emphasis on data security, the Construction Site Management System employs robust measures to protect sensitive information, ensuring confidentiality in project management. This platform aims to redefine construction project management by bridging the gap between clients and contractors, offering a transparent, accessible, and effective solution for remote construction oversight in today’s digital era.

TABLE OF CONTENTS

INTRODUCTION

Introduction

1.1 About The Project

The Construction Site Management System is an innovative digital solution designed to streamline and optimize the management of construction projects. This web-based platform provides a centralized and user-friendly environment that supports remote tracking, communication, and project transparency, catering specifically to clients who cannot physically visit the worksite regularly. By allowing clients to monitor project progress, approve budgets, track labor attendance, and review material purchases, the system enables effective oversight and fosters trust in project management. Contractors benefit from tools to manage resources, schedule tasks, and communicate with clients, resulting in a more organized, accountable, and efficient worksite operation. This project aims to revolutionize construction project management by bridging the gap between on-site activities and remote client supervision, ensuring transparency and accuracy in construction workflows

1.2 Objective Of the Project

The **Construction Site Management System** is designed as a dual-benefit platform, enhancing project oversight for clients while facilitating streamlined management for contractors. This system allows clients to remotely review and approve daily budgets, attendance records, and material usage, providing transparency and preventing financial discrepancies. By digitizing project tracking and reporting, the system reduces paperwork and minimizes human error, offering a reliable, efficient, and secure way to manage construction projects. It aims to strengthen trust and communication between clients and contractors, making the construction process more accessible and accountable, and contributing to high-quality project outcomes.

1.3 Scope Of the Project

The Construction Site Management System encompasses a comprehensive platform for remotely managing various aspects of construction projects. This includes tracking project budgets, daily attendance, material purchases, and overall worksite progress, all accessible through an intuitive dashboard. Clients can browse through live data from multiple worksites, review detailed reports, and approve expenditures, providing essential oversight even without on-site visits. The system also includes automated notifications for project milestones and approvals, enhancing communication between clients and contractors. Future features may include advanced data analytics for project trends, secure payment processing, and project completion certification, aiming to offer a complete solution for transparent, efficient, and accessible construction management.

FEASIBILITY STUDY

2. Feasibility Study

System feasibility is a test or evaluation of the complete system plan. Such an evaluation is necessary to define the application area along with the extend and capability to provide the scope of computerization together with suggested output and input format and potential benefits. Feasibility study is a proposal according to the work ability, impact on the organization, ability to meet user's needs and efficient use of resources. The feasibility study is conducted to determine if the proposed system is feasible or not. Feasibility analysis evaluates the candidate systems and determines the best system that needs performance requirements. The purpose of feasibility study is to investigate the present system, evaluate the possible application of computer-based methods, select a tentative system, evaluate the cost and effectiveness of the proposed system, evaluate impact of the proposed system on existing personnel and ascertain the need for new personnel. All projects are feasible when given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. A feasibility study is not warranted for systems in which economic justification is obvious, technical risk is low, few legal problems are expected and no reasonable alternative exists. An estimate is made of whether the identified user needs may be satisfied using current software and hardware technologies. The study will decide if the proposed system will be cost effective from the business point of view and if it can be developed in the given existing budgetary constraints. The feasibility study should be relatively cheap and quick. The result should inform the decision of whether to go ahead with a more detailed analysis. Feasibility study may be documented as a separated report to higher officials of the top-level management and can be included as an appendix to the system specification. Feasibility and risk analysis are related in many ways. If there is more project risk then the feasibility of producing the quality software is reduced.

The key combinations are involved in the feasibility study:

- Economic Feasibility.
- Technical Feasibility.
- Behavioral Feasibility.

- Operational Feasibility

2.1 Economic Feasibility

Economic feasibility is assessed through cost-benefit analysis, which evaluates the financial implications of implementing the Construction Site Management System. The primary goal of this project is to reduce costs and time associated with managing construction projects while increasing transparency regarding budgets, work hours, and labor on-site. The system is designed to be cost- effective, requiring minimal initial investment while providing significant long-term savings by optimizing resource allocation and reducing inefficiencies. By enabling clients to remotely monitor projects, the system minimizes the need for physical site visits, which can incur travel costs and time delays. The anticipated benefits, including improved oversight and reduced risk of financial discrepancies, suggest that the system will be economically advantageous for construction firms and their clients.

2.2 Technical Feasibility

➢ Does the necessary technology exist to do what is been suggested?

➢ Does the proposed equipment have the technical capacity for using the new system?

➢ Are there technical guarantees of accuracy, reliability and data security?

Technical feasibility focuses on the technology and infrastructure required to implement the Construction Site Management System. The system will utilize widely available web-based technologies to ensure compatibility across various devices, allowing users to access project information from anywhere. Key technical requirements include a reliable internet connection, user-friendly interfaces, and secure data management protocols to protect sensitive project information. The platform will incorporate features such as real-time progress tracking, budget monitoring, and communication tools to facilitate collaboration between clients and contractors. Given the current technological landscape, the proposed system can be effectively developed using existing software and hardware, making it technically feasible.

2.3 Behavioral feasibility

Behavioral feasibility evaluates the likelihood that users will adopt and effectively utilize the Construction Site Management System after its implementation. Understanding the needs of both clients and contractors is essential to ensure that the system is user-friendly and meets their expectations. The project will prioritize user experience by incorporating intuitive design elements and providing comprehensive training resources. Engaging potential users during the development process through surveys and feedback sessions will help identify and address any concerns, fostering acceptance of the system. By demonstrating the system's value in enhancing transparency and reducing the potential for fraud, the project is expected to gain broad support among stakeholders in the construction industry.

2.4 Operational Feasibility

Question that going to be asked are:

- Will the system be used if it developed and implemented?
- If there was sufficient support for the project from the management and from the users.
- Have the users been involved in planning and development of the project.
- Will the system produce poorer result in any respect or area?

This application can be implemented in an organization because there is adequate support from management and users. And application will be used by them since it doesn’t generate poorer results or problems in any area. Therefore, the implementation of this application is operationally feasible. Operational feasibility is concerned with human, organizational and political aspects. The issues considered are the job changes that will be brought about, the organizational structures that will be distributed and the new skills that will be required**.**

SYSTEM ANALYSIS

3. System Analysis

Analysis is a structured method for identifying and solving problems. Analysis implies breaking something into its parts so that the whole may be understood. The definition of system analysis not only process analysis but also that of synthesis, which implies the process of putting parts together to form a new whole. All the activities relating to the life cycle phase must be performed managed and document. To design a system, we need requirements of the system and the specification document are prepared in this phase. The purpose of this documents to specify the functional requirement of the software that is to build. The specifications are intended to guide the activities, relationships and all other objectives. The main thing is to find what is to be done to solve the problems with the current system. In the phase the problems or drawbacks of the current system is identified and the necessary actions to solve these problems are recommended

3. Existing System

The existing system for managing construction projects from a client’s perspective is generally limited and lacks the features needed for an interactive, transparent experience. Most construction companies provide a simple web page or form where clients can contact staff or leave feedback. While this serves as a basic communication tool, it doesn’t offer clients much control over their project experience or access to updates on their work. Current systems often rely on emails or sporadic updates rather than a structured platform where clients can easily track progress. These systems tend to have outdated, poorly designed user interfaces that can be difficult to navigate, making it frustrating for clients to find relevant information about their projects. For instance, clients often lack individual accounts, meaning they don’t have a dedicated space to manage or review their specific project details; instead, they may have to request updates from staff directly, leading to delays and inefficiency. This lack of personalization makes it challenging for clients to differentiate between the general company information provided on these pages and the specific details related to their own projects. Moreover, in many of these systems, clients cannot view ongoing updates, budgets, or timelines directly. They are often unable to track their project’s daily progress or expenses online, which can cause anxiety and dissatisfaction due to the lack of transparency and control. Ultimately, this outdated approach doesn’t meet modern expectations for instant access to information, especially for clients looking for real-time insights into their projects, limiting their overall satisfaction and engagement.

Disadvantages of Existing System are:

- Hard to use
- Bad user interface
- Users doesn’t have an individual account
- Hard for users to find their project details
- Tracking the work online is not possible

3.2 Proposed System

Proposed System

The proposed construction management system introduces a modern, efficient solution by giving each customer a secure, personalized account. Upon logging in, customers can access a dashboard where all their assigned worksites and project details are organized for easy viewing. This interface allows users to search for and monitor daily updates, including work progress, expenses, and timelines, with just a few clicks. By simplifying access to critical project information, the system improves transparency and enables clients to stay informed about every aspect of their construction project. Additionally, the system incorporates features for scheduling and attending virtual or in-person sessions with designated trainers or contractors. Customers can choose their preferred session time and location, making it easier for them to get hands-on or virtual updates and support as needed.

Advantages of the Proposed System

The proposed system offers numerous benefits to enhance the client experience. First, it features a **user-friendly interface** that is easy to navigate, designed to minimize confusion and provide clear pathways to information. With an updated, **good user interface**, clients can intuitively find the data they need and interact with the platform without hassle. The system also enables **efficient tracking of work progress** with real-time data, empowering clients to monitor their project status at any moment. Furthermore, it includes an **easy registration process** for new users, enabling quick access without unnecessary delays. Finally, the platform provides comprehensive **day-to-day details** on project activities, financials, and milestones, ensuring that clients are well-informed throughout the project lifecycle. This streamlined, user-centred approach not only improves transparency but also strengthens the client's trust in the company by offering reliable, continuous access to important project information.

Advantages of the proposed System are:

- User friendly
- Good user interface
- Efficient work progress tracking
- Easy Registration process
- Availability of Day-to-day details

MODULES

4 MODULES

4.1 ADMIN

Admin is responsible for the overall management of the system. Admin can view the details of very other modules. Admin can see the customer requests, customer details, worksite details, Budget details, Customer queries and every other details. Admin is responsible for assigning a new worksite to profile.

Here's a detailed list of **Admin's responsibilities** in each module of the system

4.1.1 Customer Management Module

- View customer profiles, details, and registration information.

- Track and manage customer requests, including requests for new worksites or project updates.

- Address and resolve customer queries and issues.

4.1.2. Worksite Management Module

- Review details for each worksite, including location, project scope, and assigned personnel.

- Assign new worksites to relevant customer profiles upon approval.

- Monitor worksite progress and ensure project milestones are updated.

4.1.3. Budget Management Module

- Review and manage budget details for each worksite, including daily expenses and overall project costs.

- Approve or adjust budget requests submitted by customers or site managers.

- Ensure budget transparency and accurate reporting for customers.

4.1.4. Customer Query Management Module

- Access and respond to customer queries or issues related to worksites, budgets, and timelines.

- Provide support and communicate directly with customers to resolve issues or provide updates.

- Track and analyze common queries to improve system functionality and customer support.

4.1.5. Reporting and System Oversight Module

- Generate reports on system activities, such as budget allocations, worksite progress, and customer satisfaction.
- Oversee the overall system performance, ensuring that all modules are functioning as expected.
- Implement system improvements based on feedback and operational data to optimize performance and usability.

4.2 Customer

At the heart of the User Module lies the process of registration. This initial step allows individuals to establish their digital identity within the platform. By providing essential information such as username, email address, and password, users gain entry into the system, unlocking a realm of possibilities for interaction and participation.

4.2.1 Registration

Allows users to create accounts within the system by providing necessary personal information such as name, email address, and password. Enables user authentication and authorization for accessing system functionalities.

4.2.2 Site Checking:

It allows users who are already registered to check whether the work progress on work site whether it is going well based on the daily uploaded data

4.2.3 Review On sites:

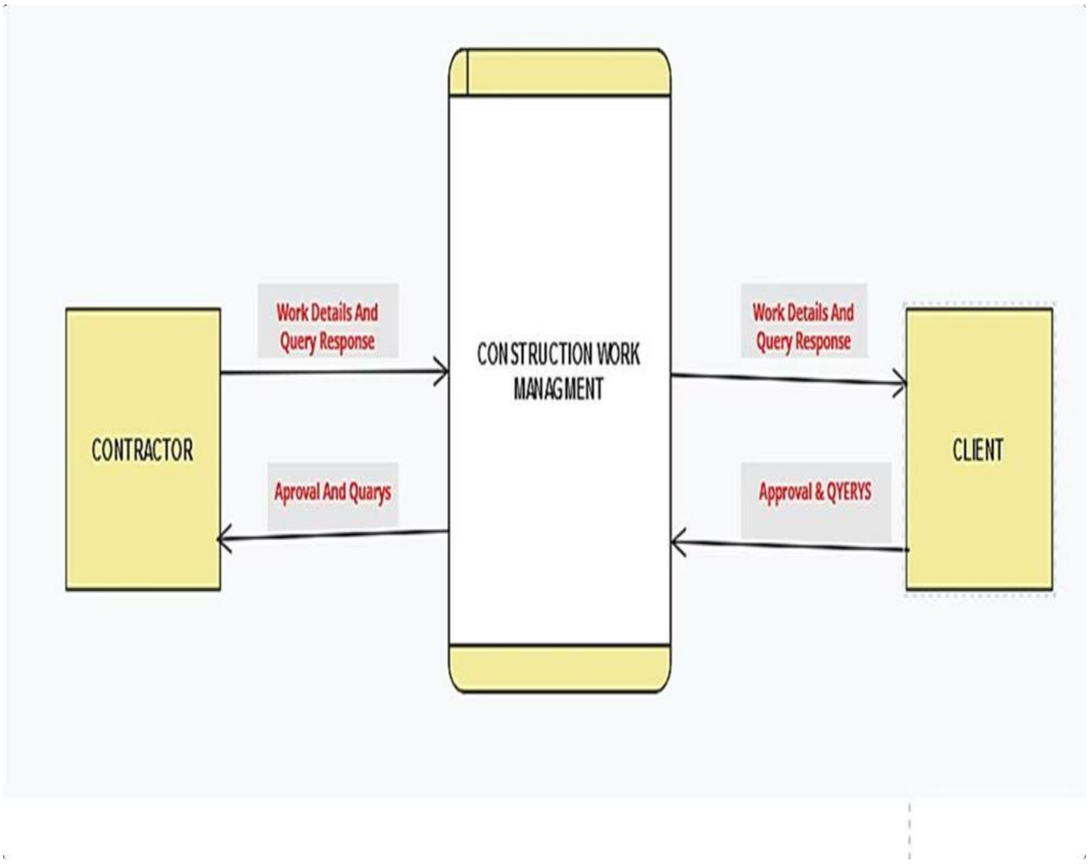
The users can leave review on the work and approve the daily work reports if everything looks good and links which they have already come across, whether they are good or they include any malicious contents

DATA FLOW DIAGRAM

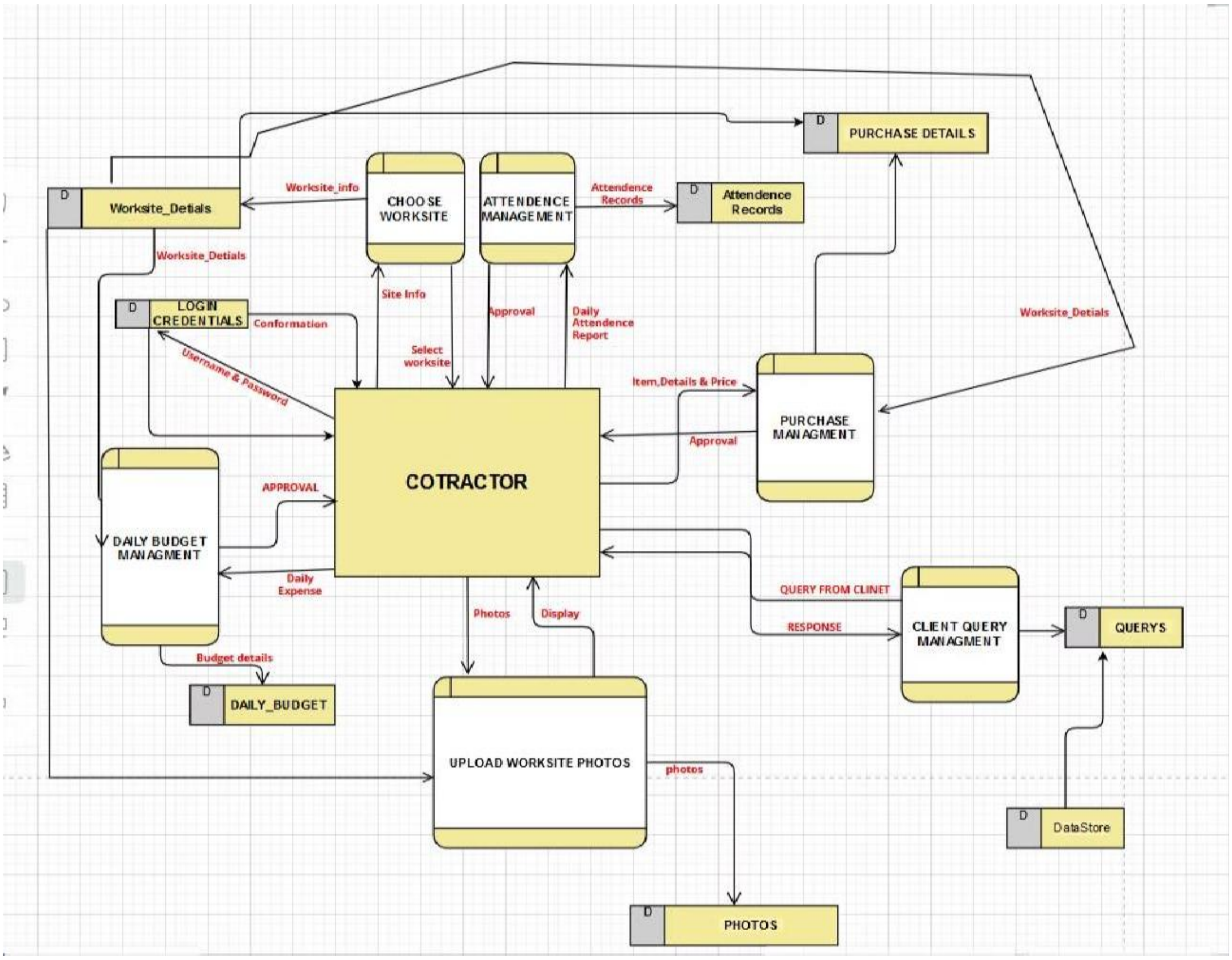
5. Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical technique that depicts information flow and transforms that are applied as data move from input to output. It is otherwise known as bubble chart. It has the purpose of clarifying system requirements and identifying major transformations that will become programs in the system design. It is the major starting point in the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. A DFD consists of a series of bubbles joined by lines. The bubble represents data flow in the system. The DFD is used to represent increasing information flow and functional details. A Level 0 DFD is called a fundamental system model represents the entire software elements as single bubble with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in the next level, i.e., Level 1 DFD. Each of the processes represented at Level 1 are sub functions of overall system depicted in the context model. Any processes which are complex in Level 1 will be further represented into sub functions in the next level, in Level 2. Data Flow diagram is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes and data sources. The purpose of data flow diagram is to provide a semantic bridge between users and system developers. The diagram are graphical, eliminating thousands of words, logical representation, modelling what system does; hierarchical, showing systems at any level of details; and jargon less, allowing user understanding and reviewing. The goal of data flow diagram is to have a commonly understood model of a system. Data flow diagram area supported by other techniques of structured system analysis such as data structured diagrams, data dictionaries and procedure representing techniques such as decision tables, decision trees.

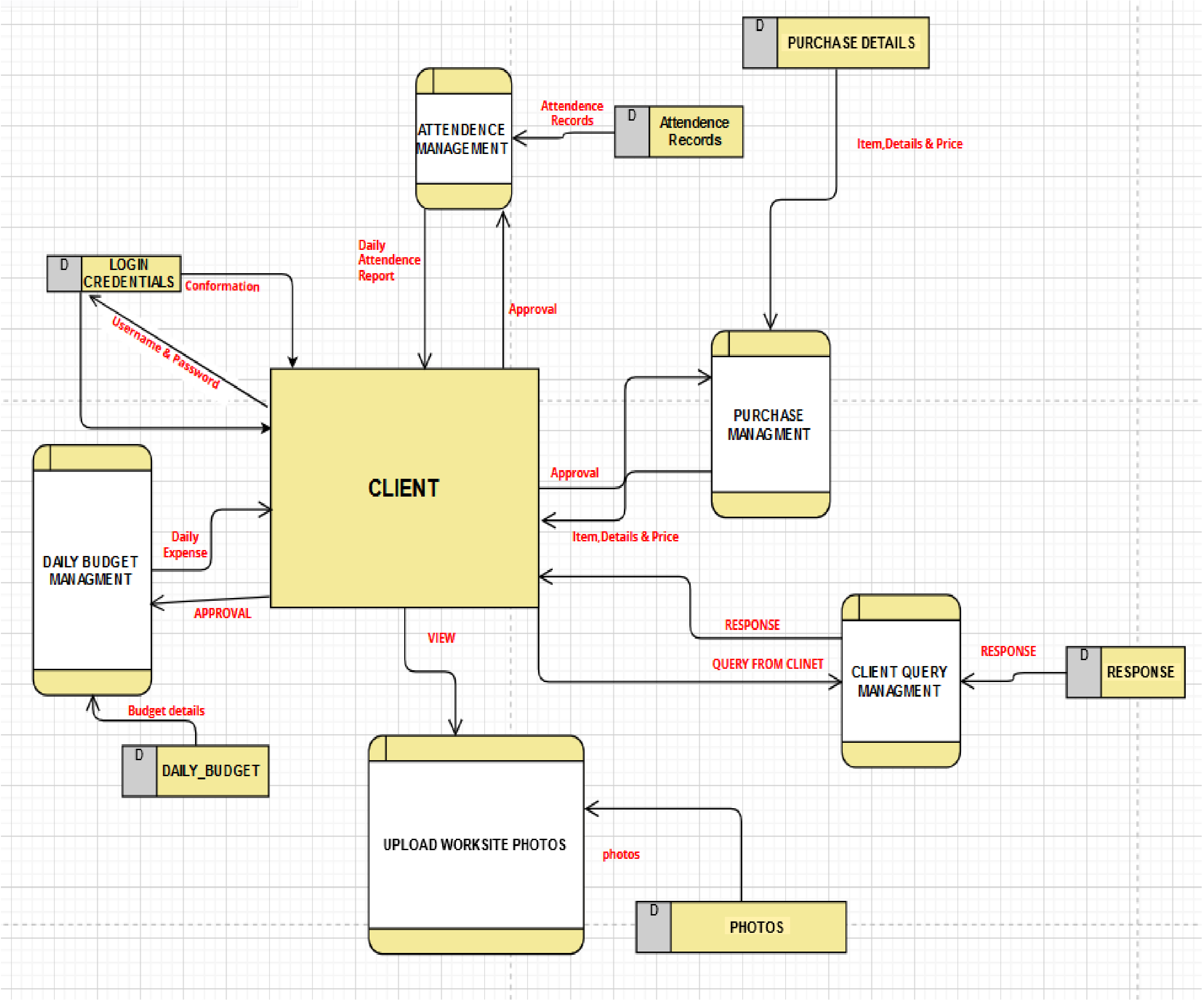
5.1 CONTEXT LEVEL-LEVEL 0 DFD


Work Details And Query Response
Aproval And Quarys
CONTRACTOR
CONSTRUCTION WORK MANAGMENT
Work Details And Query Response
Approval & QYERYS
CLIENT

5.1 LEVEL 1 – Admin



LEVEL 1 – CUSTOMER



SYSTEM DESIGN

6. System Design

System designing is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. it is a solution to a "how to" approach compared to system analysis which is a "what is" orientation. it translates the system requirements into ways of making them operational. The design phase focuses on the detailed implementation of the system recommended in the feasibility study. The system which is in making is developed by working on two different modules and combining them to work as a single unit. That single unit is the one which is known as the new software. We go through the different design strategies to design the system we are talking about. In the input design we decide which type of input screens are going to be used for the system in making. In the output design we decide the output screens and the reports that will be used to give the output and in the database design we decide what all tables will be required and what all fields will be there in those tables. Each of them discussed briefly below.

6.1 Input Design

Input design converts user-oriented inputs to computer-based formats, which requires careful attention. The collection of input data is the most expensive part of the system in terms of the equipment used and the number of people involved. In input design, data is accepted for computer processing and input to the system is done through mapping via a map support or links. Inaccurate input data is the most common cause of errors in data processing. The input screens need to be designed more carefully and logically. A set of menus is provided which help for better application navigation. While entering data in the input forms, proper validation checks are done and messages will be generated by the system if incorrect data has been entered. The objective of input design is to create an input layout that is easy to follow and prevent operator errors. It covers all phases of input from creation of initial data into actual entry of the data to the system for processing. The input design is the link that ties the system into world of its users. The user interface design is very important for any application. The interface design defines how the software communication within itself, to system that interpreted with it and with human who use it. The input design requirements such as user friendliness, consistent format and interaction dialogue for giving the right message and help for the user at right time are also considered for the development of the project.

6.2 Output Design

Outputs are the most important and useful information to the user and to the department. Intelligent output designs will improve systems relationships with the user and help much in decision-making. Outputs are also used to provide a permanent hard copy of the results for later use. The forms used in the system are shown in the appendix. The outputs also vary in terms of their contents, frequency, timing and format. The users of the output, its purpose and sequence of details to be printed are all considered. The output forms a system in the justification for its existence. If the outputs are inadequate in any way, the system itself is inadequate. The basic requirements of output are that it should be accurate, timely and appropriate, in terms of content, medium and layout for its intended purpose. Hence it is necessary to design output so that the objectives of the system are met in the best possible manner.

6.3 Table Design

The efficiency of an application using SQLITE-3 Server is mainly dependent upon the database tables, the fields in each table and joined using the fields contained in them to retrieve the necessary information. A table is a set of data elements that is organized using a model of vertical columns and horizontal rows. A table has a specified number of columns, but can have any number of rows. Each row is identified by the values appearing in a particular column subset which has been identified as a unique key index. The primary objective of a database design is fast response time to inquiries, more information at low cost, control of redundancy, clarity and ease of use, accuracy and integrity of the system fast recovery and availability of powerful end-user language.

Table Login

6.3.1 Name of table : tbl login

6.3.2 Primary key : login_id

6.3.3 *Description:* To store login details

SL.NO	FIELD NAME	DATA TYPE	CONSTRAINT
1	login_id	int	Primary Key
2	username	varchar	Not Null
3	password	Varchar	Not Null
4	usertype	Varchar	Not Null

Worksite

6.1.1 Name of table: table Worksite

6.1.2 Primary key: primary key

6.1.3 *Description:* To store worksite details.

SL.NO	FIELD NAME	DATA TYPE	CONSTRAINT
1	Worksite_id	int	Primary Key
2	username	varchar	Unique
3	Startdate	date	Not Null
4	End_date	date	Not Null
5	Current_status	varchar	Not null

Table Purchase Details

Name of table : tbl Purchase

Primary key : worksite_id

Description: To store Purchase details

SL.NO	FIELD NAME	DATA TYPE	CONSTRAINT
1	Worksite_id	int	Primary Key
2	Date	date	Unique
3	cost	int	Not Null
4	Details	varchar	Not Null
5	Approval_status	Boolean	Not null

Table for DAILY DETAILS

Name of table : tbl daily_report

Primary key: worksite_id

Description: To store daily reports

SL.NO	FIELD NAME	DATA TYPE	CONSTRAINT
1	Worksite_id	int	Primary Key
2	Date	date	Unique
3	UserId	int	Foregin key
4	Details	varchar	Not Null
5	Approval_status	Boolean	Not null

Table Approval

Name of table : tbl Approval

Primary key : worksite_id

Description: To store approved details

SL.NO	FIELD NAME	DATA TYPE	CONSTRAINT
1	Worksite_id	int	Primary Key
2	Date	date	Unique
3	UserId	int	Foreign key
4	Details	varchar	Not Null
5	Approval_status	Boolean	Not null

Table USER TABLE

Name of table : tbl Users

Primary key : Userid

Description: To store User details

SL.NO	FIELD NAME	DATA TYPE	CONSTRAINT
1	Userid	int	Primary Key
2	Username	Varchar	Unique
3	Password	Varchar	Unique
4	Role	varchar	Not Null
5	Created	Date	Not null

TOOLS AND PLATFORM

7.Tools And Platform

7.1 Front End: JAVASCRIPT, HTML, CSS

JavaScript, HTML, and CSS are the core technologies for front-end web development. HTML provides the structure and content of the web pages, CSS styles and positions elements, and JavaScript adds interactivity and dynamic behavior to the user interface. Together, they create responsive and engaging web experiences across various devices.

7.1.1 HTML

HTML is the backbone of the front end, providing the structural layout for the web pages. It will be used to define the following elements:

- Page Structure: Layouts for various pages, such as the home page, login and registration forms, dashboards, project detail views, attendance logs, and approval interfaces.
- Forms: Interactive forms for login, registration, attendance logging, daily updates, budget submissions, and feedback collection.
- Content Display: Structured content blocks to display data such as project details, worksite information, task lists, and budgets in a clear, organized manner.
- Responsive Design Elements: By using semantic HTML tags (like <header>,
  <footer>, <nav>, <section>, and <article>), the pages will be structured to ensure accessibility and SEO optimization.

6.3.4 CSS (Cascading Style Sheets)

CSS is essential for styling the application to create an appealing and professional look that enhances user experience. It will be applied in the following ways:

- Layout and Design: CSS will be used to create a visually appealing layout for all web pages. Consistent spacing, padding, and layout grids will make the interface more readable and organized.
- Styling Components: Buttons, forms, tables, and input fields will be styled for visual clarity and ease of use. Different colors and themes can be applied to highlight action buttons, approvals, and notifications.
- Responsive Design: Using CSS media queries, the application will be made responsive, allowing users to access it on various devices (desktop, tablet, mobile) without compromising usability.
- Animations and Effects: CSS transitions and animations will be used for smoother interactions, such as hover effects on buttons, pop-ups for notifications, and smooth scrolling.
- Branding: The overall color scheme, font styles, and logos will reflect the branding of the construction company, creating a cohesive look across the website.

7.1.3 JAVASCRIPT

JavaScript is critical for interactivity and enhancing user experience on the website. Its use in this project includes:

- Form Validation: JavaScript will perform client-side validation on forms (login, registration, budget submission, etc.) to ensure data accuracy before submitting to the backend.
- Dynamic Content Loading: JavaScript will allow data to be fetched asynchronously from the backend using AJAX or Fetch API, enabling seamless data updates (e.g., project updates, approvals, and attendance logs) without page reloads.
- Interactive Dashboards: JavaScript libraries like Chart.js or D3.js may be used to create visual charts and graphs for tracking project progress, budget usage, and attendance statistics.
- Modal and Pop-up Windows: JavaScript will enable modals and pop-up windows for displaying important information or actions (e.g., confirming budget approval, viewing task details).
- Real-time Notifications: For features like real-time notifications on approvals, budget changes, or project milestones, JavaScript will handle dynamic updates and alert the users.
- Navigation and Interactivity: JavaScript will enhance navigation by providing drop-down menus, sidebar toggling, and smooth transitions, creating a more intuitive and engaging user experience

7.2Backend:

7.2.1 Django:

Django is a high-level Python framework used as a backend for building scalable, secure web applications. It follows the Model-View-Template (MVT) architecture, enabling developers to handle data and business logic efficiently. Django's built-in ORM simplifies database interactions, while its authentication system and security features protect against common web vulnerabilities. With Django REST Framework (DRF), Django can also serve as a backend for single-page applications (SPAs) and mobile apps by providing RESTful APIs. Its modularity and robust admin interface make it ideal for projects ranging from simple websites to complex, data-driven applications.

7.3 Data Base: SQLite3

MySQL is a multithreaded, multi-user SQL Database Management System. My SQL's implementation of a relational database is an abstraction on top of a computer’s file system. The relational database abstraction allows collection of data items to be organized as a set of formally described tables. Data can be accessed or reassembled from these tables in many different ways, which do not require any reorganization of the database tables themselves. MySQL database has become the world's most popular Open-source database because of its consistency, fast performance, high reliability and ease of use.

7.4 IDE: VISUAL STUDIO CODE V1.69

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go, .NET). In normal terms, it facilitates users to write the code in an easy manner. Many people say that it is half of an IDE and an editor, but the decision is up to the coders.

7.4.1 Why Visual studio code?

Visual Studio Code is a very popular coding editor used by millions of developers around the world. If you are a Web Developer, you probably use Visual Studio Code for coding your

projects. Its popularity is due to the growth of the web development field in these years and the need of the developers of having a lightweight well-done editor, with few features but less. complex than the others available on the market. It's also free and it's developed and maintained by Microsoft with a modern approach using Electron. These are some characteristics that bring VS Code to the top of the coding editors

7.4.1.1 Open-source

7.4.1.2 Simplicity

7.4.1.3 Familiarity

7.4.1.4 Minimal Design

7.4.1.5 Extensions

7.4 WINDOWS 11

Windows 11 offers a comprehensive development environment for building and testing this Django- based construction management system. With Visual Studio Code, developers have a powerful IDE for coding, debugging, and managing the Django framework, while the Windows Subsystem for Linux (WSL) provides compatibility with Linux commands, creating a production-like environment within Windows. PostgreSQL and other databases can be managed through GUI tools like pgAdmin, and front-end design is facilitated by software such as Adobe XD or Figma, ideal for creating an intuitive user interface. The operating system’s Edge browser and developer tools allow for effective cross-browser testing, ensuring that the web application performs consistently across platforms. Windows 11’s integration of Git and GitHub, along with collaboration tools like Microsoft Teams, enhances teamwork and version control, while virtualization support (via Hyper- V) allows for testing across different OS configurations, making Windows 11 a robust choice for full-stack development and deployment.

SYSTEM TESTING

8 System Testing

8.1 Unit Testing

Here we test each module individually and integrated the overall system. Unit testing focuses verification efforts even in the smallest unit of software design in each module. This is known as” module testing”. The modules are tested separately. This testing is carried out in the programming style itself. In this testing each module is focused to work satisfactorily as regard to expected output from the module. There are some validation checks for the fields. Unit testing gives stress on the modules independently of one another, to find errors. Different modules are tested against the specifications produced during the design of the modules. Unit testing is done to test the working of individual modules with test servers. Program unit is usually small enough that the programmer who developed it can test it in a great detail. Unit testing focuses first on that the modules to locate errors. These errors are verified and corrected and so that the unit perfectly fits to the project

8.2 Integration Testing

Data can be lost across an interface, one module can have an adverse effect on the other sub functions, when combined they may not perform the desired functions. Integrated testing is the systematic testing to uncover the errors within the interface. This testing is done with simple data and the developed system has run successfully with this simple data. The need for integrated system is to find the overall system performance. After splitting the programs into units, the units were tested together to see the defects between each module and function. It is testing to one or more modules or functions together with the intent of finding interface defects between the modules or functions. Testing completed at as part of unit or functional testing, integration testing can involve putting together of groups of modules and functions with the goal of completing and verifying meets the system requirements.

8.3 System Testing

System testing focuses on testing the system as a whole. System Testing is a crucial step in Quality Management Process. In the Software Development Life Cycle, System Testing is the first level where the System is tested as a whole. The System is tested to verify whether it meets the functional and technical requirements. The application/System is tested in an environment That closely resembles the production environment where the application will be finally deployed. The perquisites for System Testing are: -

- All the components should have been successfully Unit Tested.
- All the components should have been successfully integrated.

Testing should be completed in an environment closely resembling the production environment. When necessary, iterations of System Testing are done in multiple environments

8.4 Acceptance Testing

The system was tested by a small client community to see if the program met the requirements defined the analysis stage. It was fond to be satisfactory. In this phase, the system is fully tested by the client community against the requirements defined in analysis and design stages, corrections are made as required, and the production system is built. User acceptance of the system is key factor for success of the system.

8.5 Validation Testing

Data validation is the process of testing the accuracy of data. A set of rules we can apply to a control to specify the type and range of data that can enter. It can be used to display error alert when users enter incorrect values in to a form. Now performing validation testing in system Centralized Social Welfare by undergoing validation for each tool and the validation succeeded when the software function in a manner that can be reasonably accepted, by the user.

8.6 Black box Testing

testing Knowing the specified function that a product has been designed to perform, test can be conducted that demonstrates each function that is fully operational, at the same time searching for errors in each function. Black Box testing focuses on functional requirement of the software.

8.7 White box Testing

testing Knowing the internal working of a product test can be conducted to ensure that “all gears mesh” that is internal operation performs according to specification and all internal components have been adequately exercised.

SYSTEM IMPLEMENTATION

9 System Implementation

Implementation includes placing the system into operation and providing the users and operation personnel with the necessary documentation to use and maintain the new system. Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new, replacing an existing system. Proper implementation is essential to provide a reliable system to meet the organizational requirements. Successful implementation may not guarantee improvement in the organization using the new system, as well as, improper installation will prevent. There are four methods for handling a system conversion. The Implementation Plan describes how the information system will be deployed, installed and transitioned into an operational system. The plan contains an overview of the system, a brief description of the major tasks involved in the implementation, the overall resources needed to support the implementation effort, and any site-specific implementation requirements. The plan is developed during the Design Phase and is updated during the Development Phase the final version is provided in the Integration and Test Phase and is used for guidance during the Implementation Phase. The implementation phase ends with an evaluation of the system after placing it into operation of time. The validity and proper functionality of all the modules of the developed application is assured during the process of implementation. Implementation is the process of assuring that the information system is operational and then allowing user to take over its operation for use and evaluation. Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operated the new system. The most crucial stage in achieving a new successful system is that it works effectively and efficiently. There are three types of implementations:

9.1 Implementation Procedure

Implementation is the process of personnel check out, install the required equipment and application and train user accordingly. Depending on the size of the organization and its requirements, the implementation is divided into three parts.

. Stage Implementation

Here system is implemented in stages. The whole system is not implemented at once. Once the user starts working with the system and is familiar with it, then a stage is introduced and implemented. Also, the system is usually updated, regularly until a final system is sealed.

• Direct Implementation

The proposed new system is implemented directly and the user starts working on the new system. The shortcoming, if any, faced are then rectified later. 8.1.3 PARALLEL IMPLEMENTATION The old and new systems are not used simultaneously. This helps in comparison of the results from two systems. Once the user is satisfied and his intended objectives are achieved by the new system, he stops using the old one.

9.2 PROBLEM STATEMENT

In the construction industry, project management is a complex process involving various stakeholders, including contractors, clients, and workers. Each project requires careful coordination, monitoring, and documentation of tasks, budgets, and daily activities. Traditional management practices, which rely on paper records and manual tracking, are time-consuming and prone to errors, leading to inefficiencies, communication breakdowns, and potential financial losses.

This project aims to develop a **web-based construction management system** that streamlines communication, documentation, and task tracking across all project stakeholders. The system will enable contractors to manage project details, clients to monitor project progress and approve expenses, and workers to log daily updates. By centralizing project data and enabling real-time access, the system intends to improve transparency, efficiency, and accountability on construction sites.

PROBLEM DEFINITION

Construction companies face several challenges in project management due to the industry's multifaceted nature.

Specifically:

- adds Inefficient Communication: Communication between clients, contractors, and workers often relies on phone calls, emails, and in-person meetings, which are inefficient and prone to misunderstandings or delays.
- Time-Consuming Documentation: Daily reports, budget tracking, and incident logs are typically maintained on paper or spreadsheets, resulting in delays, data redundancy, and risks of data loss.
- Budget Overruns: Due to poor tracking of expenses and lack of timely approvals, project budgets are often exceeded. Delays in approvals from clients further complicate financial planning.
- Difficulty in Tracking Project Progress: Project stakeholders, especially clients, lack real- time visibility into project updates, attendance records, and daily budget usage, resulting in reduced trust and dissatisfaction.

CONCLUSION

10 Conclusion

This system will be a help for those who are suffering from different mental situations. They can easily find a counselor using our system. Anyone can easily register to our system because the registration process is simple. We are providing a good user interface, and because of such a good user interface, even a layman can simply register and use our system. This will offer more job opportunities for those people who want to work as a counselor. Users can register and attend counseling sessions with qualified counselors. This system will offer the customers more trusted services. Because a trainer is added to the system, only if he satisfies a particular selection criterion. This will help to increase the loyalty of trainers. In short, this system will help to connect those people who need counseling to increase their own mental health with good quality trainers. This system is a very relevant one in the current corporate world filled with people struggling with different mental issues.

APPENDICES

11 Bibliography

11.1. Book References

- Bahrami, A. Object Oriented System Development. Tata Mc Graw Hill.
- Bayross, I. SQL,PL/SQL programming language of Oracle. BPB.
- Python for Beginners. Harsh Bhasin
- Navathe, E. &. Fundamentals of Database System. Addision Wesley.
- Python Django Web Development: The Ultimate Django web framework guide for Beginners. William Jordon
- Somerville, I. Software Engineering. Pearson Education.

11.2 Website References

- Retrieved from https://www.w3schools.com
- Retrieved from www.stackoverflow.com

11.2 HARDWARE AND SOFTWARE REQUIRMENTS

11.2.1 HARDWARE REQUIRMENTS

Development Machine

- Processor: Dual-core CPU (Intel i3 or equivalent and above)
- RAM: Minimum 4GB (8GB recommended for smooth performance)
- Storage: 10GB free disk space (for codebase, database, and dependencies)
- Display: 1024x768 resolution or higher

Production Server

- Processor: Quad-core CPU (Intel Xeon or equivalent for higher loads)
- RAM: Minimum 4GB for light traffic; 8GB or more recommended for moderate to heavy traffic
- Storage: SSD with at least 20GB free for files, media, logs, and database storage
- Network: Reliable internet connection with 1 Gbps speed or higher for optimal performance

11.2.2 SOFTWARE REQUIRMENTS

Operating System

- **Development**: Windows, macOS, or Linux (Ubuntu recommended)
- **Production**: Linux (Ubuntu Server 20.04 LTS or later recommended)

Programming Language and Framework

- Python 3.8 or higher
- Django 3.2 or later (up to latest stable release)

Web Server

- **Development**: Django’s built-in server (for local testing)
- **Production**: Nginx or Apache (as a reverse proxy) paired with Gunicorn or uWSGI for application serving

Database

- PostgreSQL (recommended for production) or SQLite (for development/testing purposes)

Other Software and Dependencies

- **Pip**: Python package manager to install dependencies
- **Virtualenv**: For creating an isolated Python environment
- **Node.js and npm** (optional): If using frontend build tools or frameworks (e.g., for asset management)
- **Redis** (optional): For caching and session management if required
- **Git**: For version control

Libraries and Dependencies

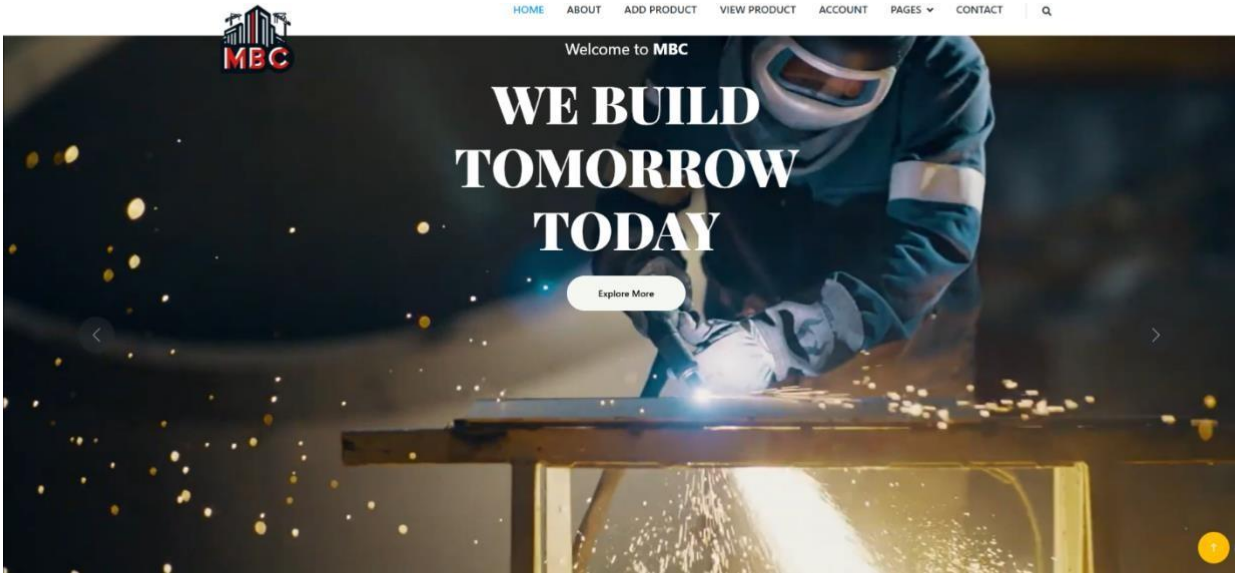
- Django Rest Framework (if building an API)
- Pillow (for handling images)
- psycopg2 (for PostgreSQL support)
- Celery (if background tasks are needed)
- Any other Django plugins specific to construction project needs (e.g., for user authentication or data visualization)
-

Optional Tools

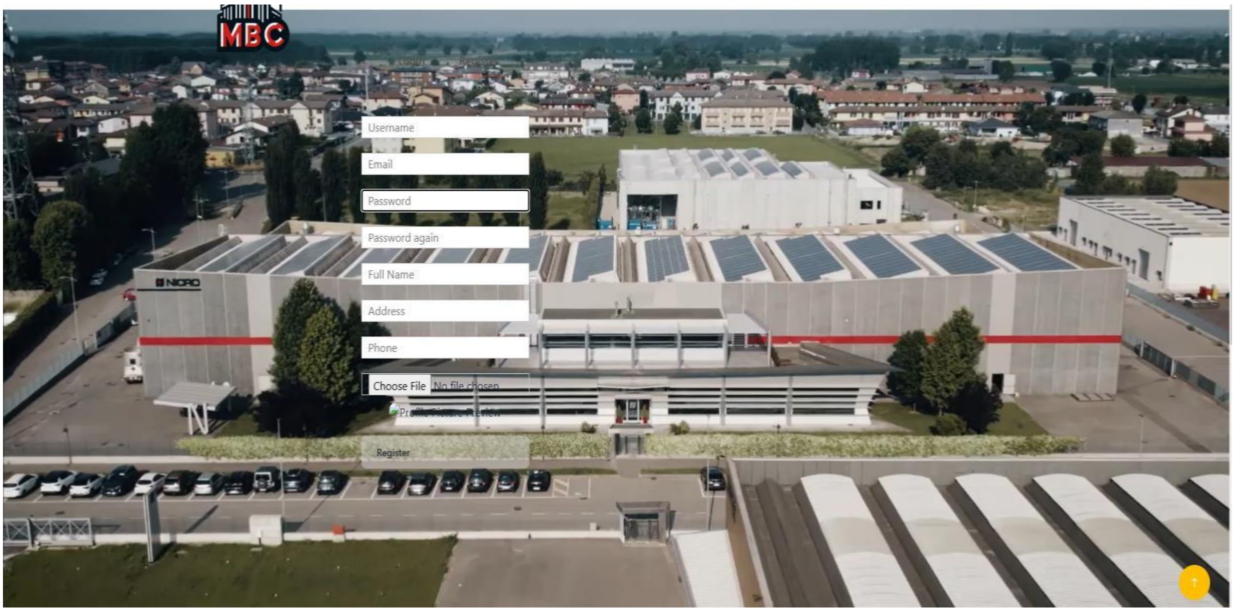
- **Docker**: To containerize the application for easy deployment and scalability
- **CI/CD Tools**: Jenkins, GitHub Actions, or GitLab CI for automated testing and deployment

APPENDIX A: FORM SCREENSHOT

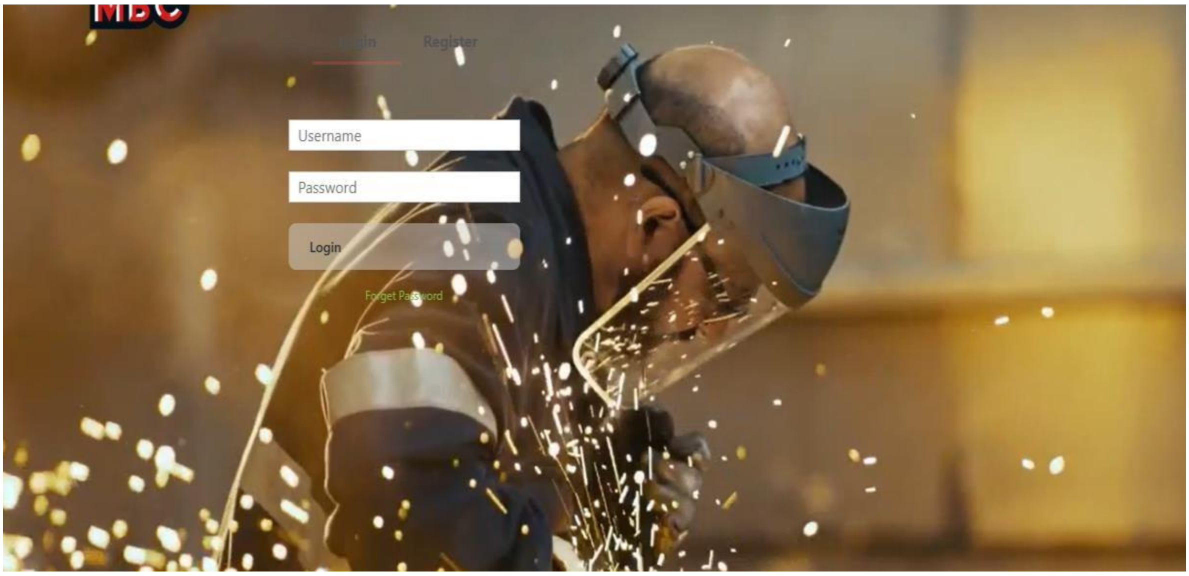
Homepage:



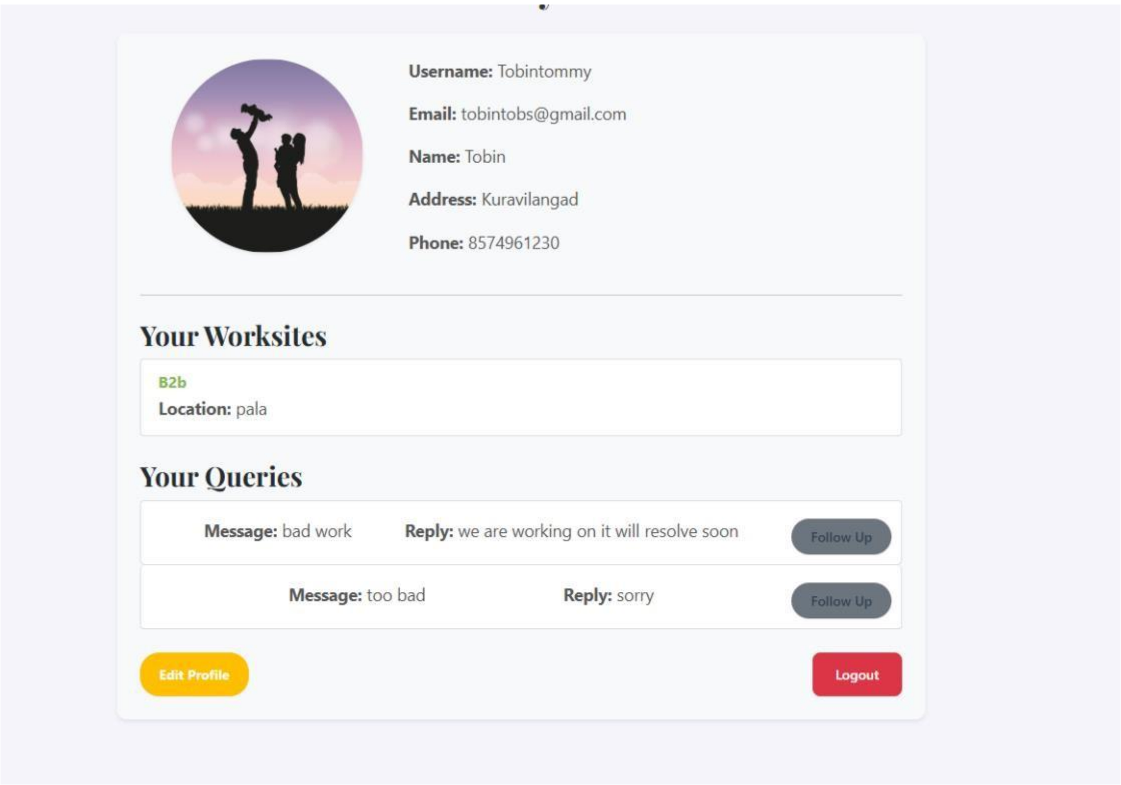
Register:



LOGIN:



User Homepage:



Username: Tobintommy

Email: tobintobs@gmail.com

Name: Tobin

Address: Kuravilangad

Phone: 8574961230

Your Worksites

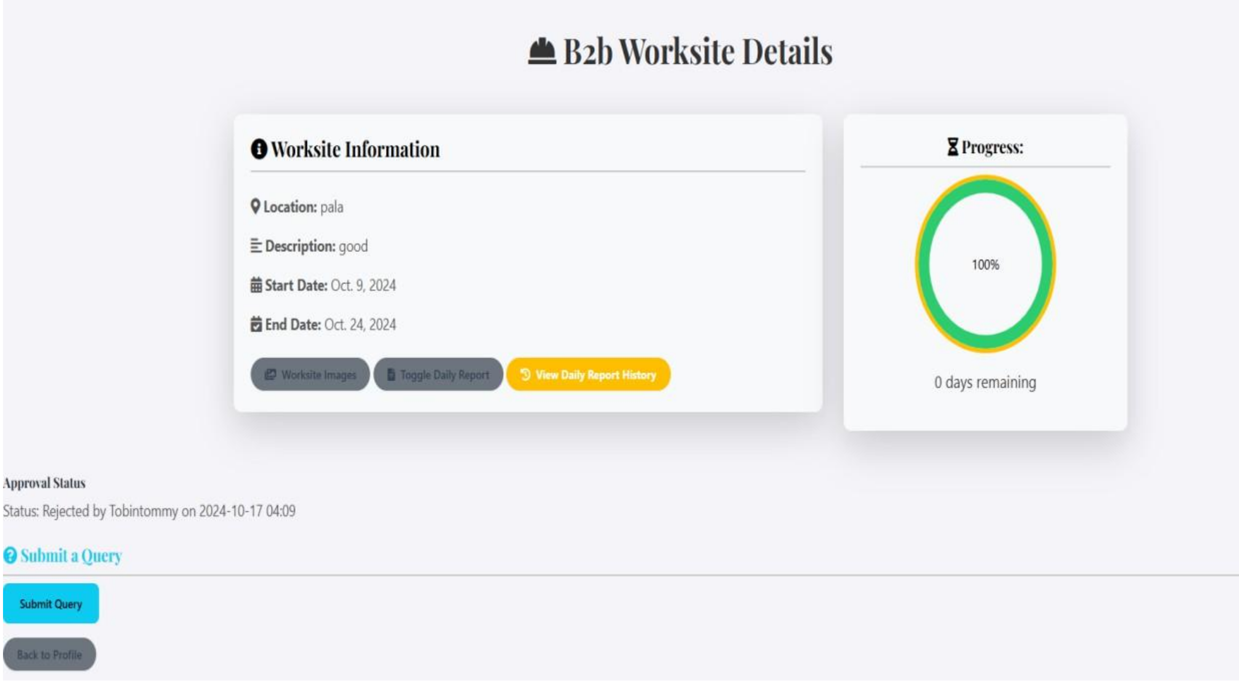
B2b
Location: pala

Your Queries

Message: bad work **Reply:** we are working on it will resolve soon

Message: too bad **Reply:** sorry

Worksite Interface:



B2b Worksite Details

Worksite Information

Location: pala

Description: good

Start Date: Oct. 9, 2024

End Date: Oct. 24, 2024

Progress:

100%

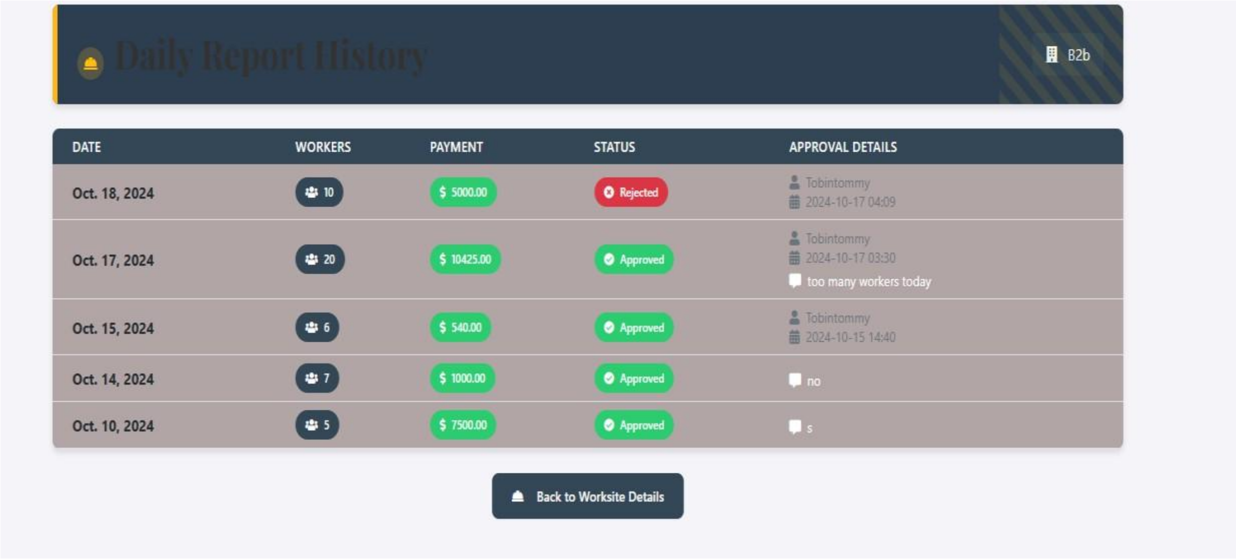
0 days remaining

Approval Status

Status: Rejected by Tobintommy on 2024-10-17 04:09

Submit a Query

Daily Report History:

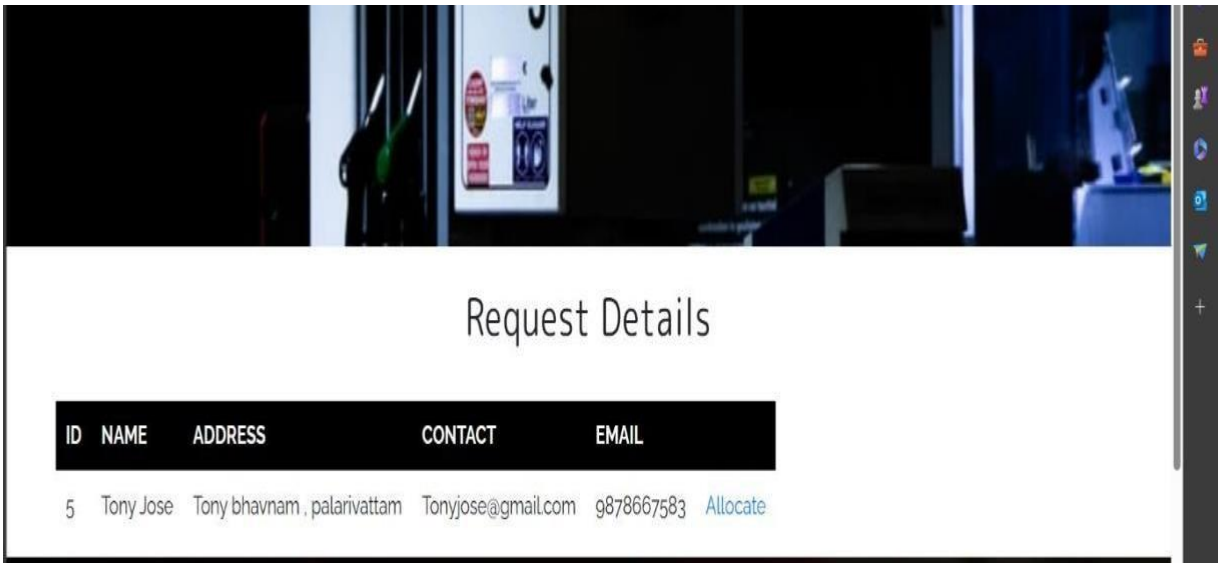


Daily Report History

B2b

DATE	WORKERS	PAYMENT	STATUS	APPROVAL DETAILS
Oct. 18, 2024	10	$ 5000.00	Rejected	Tobintommy 2024-10-17 04:09
Oct. 17, 2024	20	$ 10425.00	Approved	Tobintommy 2024-10-17 03:30 too many workers today
Oct. 15, 2024	6	$ 540.00	Approved	Tobintommy 2024-10-15 14:40
Oct. 14, 2024	7	$ 1000.00	Approved	no
Oct. 10, 2024	5	$ 7500.00	Approved	s

User Request:



Request Details

ID	NAME	ADDRESS	CONTACT	EMAIL	
5	Tony Jose	Tony bhavnam , palarivattam	Tonyjose@gmail.com	9878667583	Allocate

SAMPLE CODE

```
from django.db import models

from django.contrib.auth.models import User


class Product(models.Model):

    name = models.CharField(max_length=100)

    description = models.TextField()

    price = models.DecimalField(max_digits=10, decimal_places=2)

    image = models.ImageField(upload_to='images/', null=True, blank=True)

    created_at = models.DateTimeField(auto_now_add=True)


    def __str__(self):

        return self.name


from django.db import models

from django.contrib.auth.models import User


class customer(models.Model):

    LIVE = 1

    DELETE = 0

    STATUS_CHOICES = (

        (LIVE, 'Live'),

        (DELETE, 'Delete')

    )
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    name = models.CharField(max_length=200)
    address = models.TextField()
    phone = models.CharField(max_length=15, null=True, blank=True)
    user = models.OneToOneField(User, on_delete=models.CASCADE, related_name='customer_profile')
    profile_picture = models.ImageField(upload_to='profile_pics/', null=True, blank=True)


    def __str__(self):
        return self.name


from django.db import models
from django.contrib.auth.models import User


class customer(models.Model):
    LIVE = 1
    DELETE = 0
    STATUS_CHOICES = (
        (LIVE, 'Live'),
        (DELETE, 'Delete')
    )
    status = models.IntegerField(choices=STATUS_CHOICES, default=LIVE)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    name = models.CharField(max_length=200)
    address = models.TextField()
    phone = models.CharField(max_length=15, null=True, blank=True)
    user = models.OneToOneField(User, on_delete=models.CASCADE, related_name='customer_profile')
    profile_picture = models.ImageField(upload_to='profile_pics/', null=True, blank=True)

    def __str__(self):
        return self.name

class Supervisor(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE, related_name='supervisor_profile')
    phone = models.CharField(max_length=15, blank=True, null=True)
    address = models.TextField(blank=True, null=True)
    profile_picture = models.ImageField(upload_to='supervisor_pics/', null=True, blank=True)
    designation = models.CharField(max_length=100, blank=True, null=True)  # Optional field for job title or designation
    joined_date = models.DateField(null=True, blank=True)

    def __str__(self):
        return self.user.username

class Worksite(models.Model):
    name = models.CharField(max_length=255)
    location = models.CharField(max_length=255)
    description = models.TextField(blank=True, null=True)
    start_date = models.DateField()

    end_date = models.DateField(blank=True, null=True)

    user = models.ForeignKey(User, on_delete=models.CASCADE)  # Owner of the worksite

    worksite_picture = models.ImageField(upload_to='worksite_pics/', null=True, blank=True)

    supervisor = models.ForeignKey(Supervisor, on_delete=models.SET_NULL, null=True, related_name='worksites')


    def __str__(self):

        return self.name


class WorksiteImage(models.Model):

    worksite = models.ForeignKey(Worksite, on_delete=models.CASCADE, related_name='images')

    image = models.ImageField(upload_to='worksite_images/')


    def __str__(self):

        return f"Image for {self.worksite.name}"


class Material(models.Model):

    worksite = models.ForeignKey(Worksite, on_delete=models.CASCADE, related_name='materials')  # Link materials to worksites

    name = models.CharField(max_length=100)

    cost = models.DecimalField(max_digits=10, decimal_places=2)

    image = models.ImageField(upload_to='materials/', null=True, blank=True)

    quantity = models.IntegerField(default=1)  # Quantity of material used

    is_approved = models.BooleanField(default=False)

    is_rejected = models.BooleanField(default=False)
    feedback = models.TextField(null=True, blank=True)


    def __str__(self):
        return self.name



class DailyReport(models.Model):
    worksite = models.ForeignKey(Worksite, on_delete=models.CASCADE)
    date = models.DateField()
    workers_count = models.IntegerField()
    total_payment = models.DecimalField(max_digits=10, decimal_places=2)
    materials = models.ManyToManyField(Material)  # Use ManyToManyField to link multiple materials
    material_cost = models.DecimalField(max_digits=10, decimal_places=2)
    material_image = models.ImageField(upload_to='materials/', null=True, blank=True)
    is_approved = models.BooleanField(default=False)
    is_rejected = models.BooleanField(default=False)
    feedback = models.TextField(null=True, blank=True)
    approved_by = models.ForeignKey(User, on_delete=models.SET_NULL, null=True, related_name='approved_reports')
    approved_at = models.DateTimeField(null=True, blank=True)


from django.db import models
from django.contrib.auth.models import User
```

```
class Incident(models.Model):
    SEVERITY_CHOICES = [
        ('LOW', 'Low'),
        ('MEDIUM', 'Medium'),
        ('HIGH', 'High'),
        ('CRITICAL', 'Critical'),
    ]

    TYPE_CHOICES = [
        ('SAFETY', 'Safety Incident'),
        ('EQUIPMENT', 'Equipment Failure'),
        ('DELAY', 'Project Delay'),
        ('WEATHER', 'Weather Related'),
        ('OTHER', 'Other'),
    ]

    daily_report = models.ForeignKey(DailyReport, on_delete=models.CASCADE, related_name='incidents')  # Corrected here
    incident_type = models.CharField(max_length=20, choices=TYPE_CHOICES)
    severity = models.CharField(max_length=10, choices=SEVERITY_CHOICES)
    description = models.TextField()
    time_occurred = models.TimeField()
    location = models.CharField(max_length=255)
    people_involved = models.TextField(blank=True, null=True)
    immediate_actions = models.TextField()
    incident_image = models.ImageField(upload_to='incidents/', null=True, blank=True)
    reported_by = models.ForeignKey(User, on_delete=models.SET_NULL, null=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    followup_required = models.BooleanField(default=False)
    followup_notes = models.TextField(blank=True, null=True)
    resolved = models.BooleanField(default=False)
    resolved_at = models.DateTimeField(null=True, blank=True)

    class Meta:
        ordering = ['-time_occurred']

    def __str__(self):
        return f"Incident on {self.daily_report.date} - {self.incident_type}"

from django.db import models
from django.contrib.auth.models import User

class Query(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE, null=True, blank=True)
    name = models.CharField(max_length=100)
    worksite = models.ForeignKey(Worksite,null=True, on_delete=models.CASCADE)
    email = models.EmailField()
    message = models.TextField()
    reply = models.TextField(null=True, blank=True)  # Field for admin's reply
    is_replied = models.BooleanField(default=False)  # To check if the query has been replied
to
    submitted_at = models.DateTimeField(auto_now_add=True)
    # New field for referencing the original query
    original_query = models.ForeignKey('self', on_delete=models.SET_NULL, null=True, blank=True, related_name='follow_ups')

    def __str__(self):
        return f"Query from {self.name or self.user.username} on {self.submitted_at}"



class WorksiteApproval(models.Model):
    worksite = models.ForeignKey(Worksite, on_delete=models.CASCADE, null=True, blank=True)

    approved_by = models.ForeignKey(User, on_delete=models.SET_NULL, null=True)
    is_approved = models.BooleanField(default=False)
    approved_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f"Approval for {self.worksite.name} by {self.approved_by.username}"

# models.py
from django.db import models
from django.contrib.auth.models import User
from django.utils import timezone

class Incident(models.Model):
    SEVERITY_CHOICES = [
        ('LOW', 'Low'),
        ('MEDIUM', 'Medium'),
        ('HIGH', 'High'),
        ('CRITICAL', 'Critical'),
    ]

    TYPE_CHOICES = [
        ('SAFETY', 'Safety Incident'),
        ('EQUIPMENT', 'Equipment Failure'),
        ('DELAY', 'Project Delay'),
        ('WEATHER', 'Weather Related'),
        ('OTHER', 'Other'),
    ]

    daily_report = models.ForeignKey(DailyReport, on_delete=models.CASCADE, related_name='incidents')
    incident_type = models.CharField(max_length=20, choices=TYPE_CHOICES)
    severity = models.CharField(max_length=10, choices=SEVERITY_CHOICES)
    description = models.TextField()
    time_occurred = models.TimeField()
    location = models.CharField(max_length=255)
    immediate_actions = models.TextField()
    incident_image = models.ImageField(upload_to='incidents/', null=True, blank=True)
    reported_by = models.ForeignKey(User, on_delete=models.SET_NULL, null=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    followup_required = models.BooleanField(default=False)

    followup_notes = models.TextField(blank=True, null=True)

    resolved = models.BooleanField(default=False)

    resolved_at = models.DateTimeField(null=True, blank=True)


    class Meta:

        ordering = ['-time_occurred']

        verbose_name = 'Incident'

        verbose_name_plural = 'Incidents'


    def __str__(self):

        return f"Incident on {self.daily_report.date} - {self.get_incident_type_display()}"


    def save(self, *args, **kwargs):

        if self.resolved and not self.resolved_at:

            self.resolved_at = timezone.now()

        super().save(*args, **kwargs)


    @property

    def status(self):

        if self.resolved:

            return "Resolved"

        elif self.followup_required:

            return "Follow-up Required"

        return "Open"


class ContactSubmission(models.Model):
```

```
name = models.CharField(max_length=100)

email = models.EmailField()

subject = models.CharField(max_length=200)

message = models.TextField()

submission_date = models.DateTimeField(default=timezone.now)

is_read = models.BooleanField(default=False)


def __str__(self):

   return f"{self.name} - {self.subject}"


class Meta:

   ordering = ['-submission_date']
```