

SolarFlow: Multi-Tenant Solar Project & Customer Management System

Mr. Hatte Basvesh Umesh, Prof. N. N. Shaikh, Prof. H. R. Dhananjaya

Dept. of Electrical Engineering

Brahmdevdada Mane Institute of Technology, Solapur

Abstract: *Mobile manipulators, which combine mobility with robotic manipulation, are increasingly being used in industrial automation, warehouse logistics, and service robotics. This project presents the design, development, and testing of a wheeled robotic arm system capable of performing pick-and-place operations in a semi-structured environment.*

The proposed system consists of a differential drive mobile platform integrated with a 4-DOF robotic arm. The mobility base enables navigation, while the robotic arm performs manipulation tasks. The system is controlled using a microcontroller-based architecture combined with sensor feedback for positioning and obstacle detection.

A key challenge addressed in this project is the coordination between navigation and manipulation subsystems. To solve this, a modular control strategy is implemented where the robot first navigates to a target location using predefined paths and then executes arm operations using inverse kinematics.

The mechanical structure is designed using lightweight materials to ensure stability and efficiency. Electrical integration includes motor drivers, sensors, and communication modules. The software architecture is built using embedded programming and basic motion planning algorithms.

Testing results demonstrate that the robot can successfully perform object pickup and placement tasks with reasonable accuracy. The system shows potential for applications in small-scale industrial automation, warehouse sorting, and educational robotics.

This project highlights the feasibility of low-cost mobile manipulation systems and provides a foundation for further research in autonomous robotics.

Keywords: *Dynamic Showtime Allocation*

I. INTRODUCTION

Background of the Study

The global shift towards renewable energy has significantly accelerated in recent years, with solar energy emerging as one of the most promising and widely adopted sources. In India, the government's initiatives such as the National Solar Mission and various state-level subsidy programs have contributed to the rapid growth of rooftop solar installations across residential, commercial, and industrial sectors.

As the adoption of rooftop solar systems increases, solar installation companies are required to manage a large number of customer projects simultaneously. Each project involves multiple stages such as initial enquiry, site inspection, loan processing, installation, documentation, and subsidy processing. Managing these stages efficiently is critical to ensure timely project completion and customer satisfaction.

However, many small and medium-sized solar businesses still rely on traditional methods such as spreadsheets, manual registers, and fragmented software tools to manage their operations. These approaches are often inefficient, error-prone, and lack real-time visibility.

The increasing complexity of solar project workflows and the need for better operational efficiency highlight the importance of a digital, centralized, and scalable management system. This project, SolarFlow, is developed to address these challenges by providing a comprehensive SaaS-based solution tailored specifically for solar installation companies.



Problem Statement

Despite the rapid growth of the solar industry, many solar businesses face significant operational challenges due to the absence of an integrated management system.

One of the major issues is the lack of a centralized platform to manage all aspects of the business. Customer information is often scattered across multiple tools, making it difficult to track project progress and maintain consistency.

Additionally, managing the project lifecycle is complex, as each solar installation involves multiple stages and dependencies. Without a structured system, it becomes difficult to monitor progress, identify delays, and ensure timely completion.

Financial management is another critical challenge. Businesses often struggle with tracking quotations, payments, and outstanding balances, leading to inaccuracies and financial inefficiencies.

Employee coordination also becomes difficult in the absence of a unified system. Tracking attendance, managing wages, and assigning responsibilities are often handled manually, resulting in errors and lack of transparency.

Furthermore, manual systems lead to data inconsistency, redundancy, and inefficiency, making it difficult to scale operations as the business grows. These limitations highlight the need for a robust, centralized, and scalable solution.

Objectives of the Project

The primary objective of this project is to design and develop a centralized SaaS-based platform that addresses the operational challenges faced by solar installation companies.

The specific objectives of the project are as follows:

- To develop a comprehensive management system that integrates customer, project, employee, finance, and inventory management into a single platform.
- To enable end-to-end tracking of solar installation projects, covering all stages from initial enquiry to subsidy receipt.
- To implement a multi-tenant architecture that allows multiple solar companies to use the same application while ensuring complete data isolation and security.
- To improve operational efficiency by automating repetitive tasks and reducing manual effort.
- To enhance transparency and accuracy in data management and reporting.
- To provide a scalable solution that supports business growth and expansion.

Motivation

The motivation behind developing SolarFlow stems from the need to solve real-world challenges faced by solar installation businesses.

One of the key motivations is to address the inefficiencies associated with manual and fragmented systems. By automating repetitive processes such as data entry, project tracking, and reporting, the system aims to significantly reduce human effort and errors.

Another important motivation is to build a domain-specific solution tailored to the unique requirements of the solar industry, rather than relying on generic CRM or ERP systems.

Overall, SolarFlow is motivated by the vision of creating a smart, efficient, and scalable digital platform that empowers solar businesses to operate more effectively and grow sustainably.

II. LITERATURE REVIEW AND RESEARCH GAP

4.1 Existing Systems

Traditional CRM Tools:

Used for managing customer data and sales, but limited to basic tracking.



ERP Systems:

Provide integrated business management but are complex and costly for small solar businesses.

Spreadsheet-Based Tracking:

Commonly used for data storage and calculations, but prone to errors and not scalable.

Limitations of Existing Systems

- Not designed specifically for the solar industry
- No proper tracking of multi-stage project workflows
- Lack of subsidy management features
- No support for multi-tenant (SaaS) architecture

Need for Proposed System

- A domain-specific system tailored for solar businesses
- Automation of workflows to reduce manual effort
- Real-time analytics for better decision-making
- Mobile-first access for ease of use anywhere

III. METHODOLOGY

Development Approach

The development of the SolarFlow system follows an Agile methodology, which emphasizes iterative development and continuous improvement. Instead of building the entire system at once, the project is divided into smaller modules that are developed, tested, and refined in cycles. This approach allows flexibility in incorporating new features, fixing issues early, and improving the overall quality of the system based on feedback during development.

System Design

The system is designed using a client-server architecture, where the frontend and backend operate as separate but connected components. The frontend provides an interactive user interface through which users can perform various operations such as managing customers, tracking projects, and viewing analytics. The backend handles business logic, processes user requests, and communicates with the database to store and retrieve data. This structured design ensures better performance, scalability, and maintainability of the system.

Module-Based Development

To ensure organized development, the system is divided into multiple functional modules such as customer management, employee and attendance management, finance management, inventory management, and dashboard analytics. Each module is developed independently with clearly defined responsibilities. This modular approach simplifies development, testing, and future enhancements, while also improving code maintainability.

Data Handling

The system uses a centralized database to store all application data, ensuring consistency and easy access. A multi-tenant architecture is implemented, where data is segregated using a unique company identifier, allowing multiple organizations to use the same system securely. Standard database operations such as Create, Read, Update, and Delete (CRUD) are used to manage data efficiently. Proper validation and structured schemas ensure data accuracy and integrity.



IV. SYSTEM ARCHITECTURE

High-Level Architecture

The SolarFlow system is designed using a modern full-stack web architecture, consisting of three main components: frontend, backend, and database. The frontend is developed using React and Next.js, providing a responsive and interactive user interface for users to manage customers, projects, and other business operations.

The backend is implemented using Next.js API routes, which handle business logic, process user requests, and ensure secure communication between the frontend and the database. This approach eliminates the need for a separate backend server while maintaining scalability and performance.

The database is built using MongoDB, which stores all application data in a structured format. It provides flexibility in handling dynamic data such as project stages, financial records, and employee details. Together, these components form a robust and scalable architecture suitable for a SaaS-based application.

Web Application System Architecture

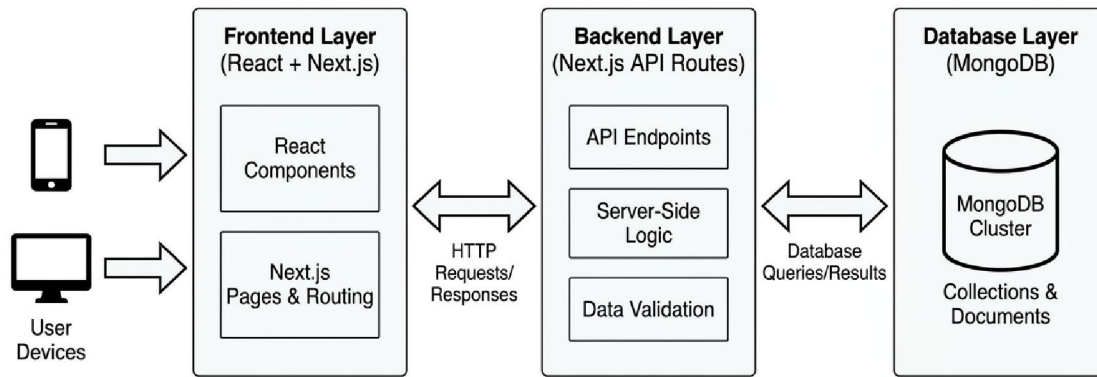


Fig. 1 System Architecture

Multi-Tenant Architecture

SolarFlow is designed as a multi-tenant SaaS application, where a single instance of the application serves multiple companies while keeping their data completely isolated. This is achieved by associating every data record with a unique companyId.

Each request made by a user is scoped to their respective company, ensuring that data is securely separated and inaccessible to other tenants. This architecture allows efficient resource utilization, centralized updates, and easy onboarding of new companies without deploying separate systems.

Additionally, feature controls and subscription limits can be managed at the tenant level, enabling customized experiences for different organizations.



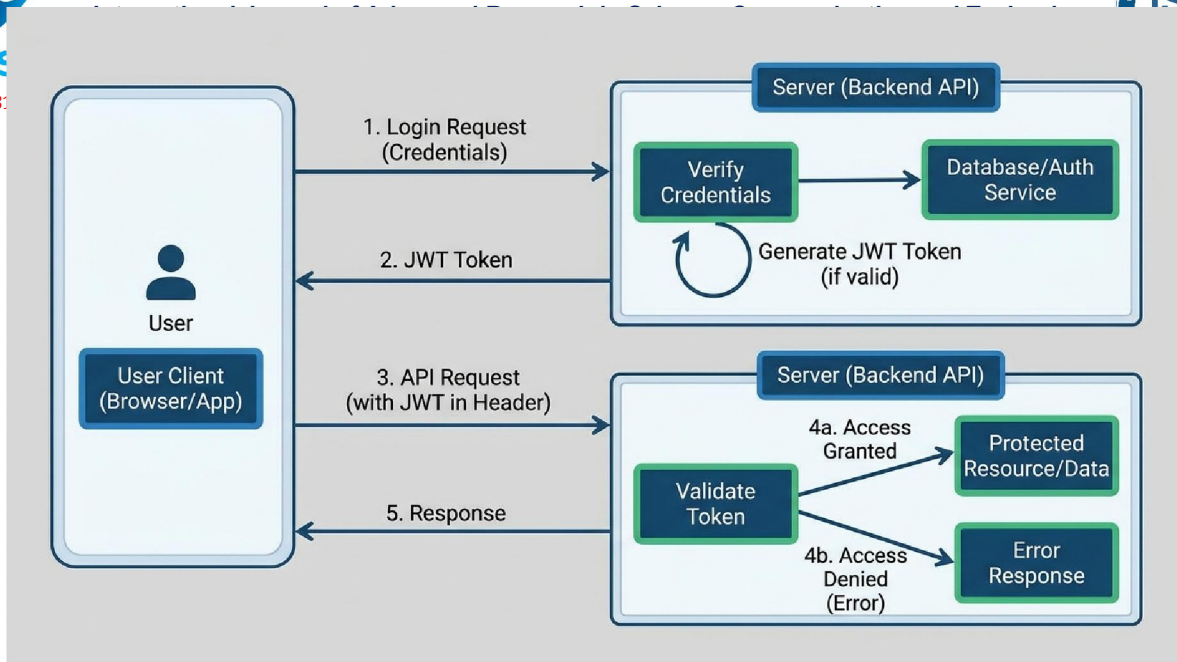


Fig. 2 Multi-tenant Architecture

Data Flow Architecture

The data flow in SolarFlow follows a structured request-response cycle. When a user performs an action on the frontend, such as adding a customer or updating project status, a request is sent to the backend API.

The backend processes the request, applies necessary business logic, and interacts with the database to store or retrieve data. Once the operation is completed, a response is sent back to the frontend, which updates the user interface accordingly.

Middleware plays a crucial role in this architecture by handling authentication and request validation before the request reaches the main API logic. This ensures security, consistency, and efficient data handling throughout the system.

V. TECHNOLOGY STACK

Frontend Technologies

The frontend of the SolarFlow system is developed using modern web technologies to ensure a responsive and user-friendly interface. React.js is used as the core library for building reusable UI components, enabling efficient rendering and state handling. Next.js is used as the framework, providing features such as server-side rendering, routing, and optimized performance.

The application is written in TypeScript, which enhances code reliability by introducing static typing and reducing runtime errors. For styling, Tailwind CSS is used, allowing rapid UI development with utility-first classes and ensuring a clean and consistent design across the application.

Backend Technologies

The backend is implemented using Node.js, integrated within Next.js through API routes. This approach allows both frontend and backend logic to be managed within a single framework, simplifying development and deployment.

The system follows a RESTful API architecture, where different endpoints handle operations such as data retrieval, creation, updates, and deletion. This ensures structured communication between the client and server and makes the system scalable and maintainable.



Database

The application uses MongoDB as its primary database, which is a NoSQL database suitable for handling flexible and dynamic data structures. It is particularly useful for storing complex data such as project workflows, customer details, and financial records.

To interact with the database, Mongoose ORM is used. It provides schema-based data modeling, validation, and an easy-to-use interface for performing database operations, ensuring data consistency and integrity.

VI. SYSTEM MODULES

Multi-Tenant Management Module

The Multi-Tenant Management Module is responsible for handling multiple companies within a single application instance. It enables the onboarding of new companies by creating dedicated workspaces for each organization. This module also provides feature toggling, allowing specific functionalities to be enabled or disabled based on company requirements. Additionally, plan limits such as the number of users and subscription validity are managed here, ensuring controlled access and scalability of the system.

Customer Management Module

The Customer Management Module is the core component of the system, designed to manage all customer-related information and project workflows. It allows the registration of new customers and maintains detailed records including contact information and project details.

A key feature of this module is the multi-stage workflow tracking, where each solar project progresses through various stages such as application, loan processing, installation, documentation, and subsidy processing. This structured approach helps in monitoring project progress, identifying delays, and ensuring efficient execution.

Employee Management Module

The Employee Management Module is used to maintain employee records and manage their roles within the system. It stores essential information such as employee details, assigned responsibilities, and work sections.

Role assignment is an important feature of this module, enabling role-based access control within the system. Additionally, it supports wage tracking by maintaining records of employee earnings, which can be used for reporting and payroll purposes.

VII. SECURITY & ACCESS CONTROL

Authentication

Authentication in the SolarFlow system ensures that only verified users can access the application. A secure login system is implemented where users must provide valid credentials to gain access. Passwords are not stored in plain text; instead, they are encrypted using hashing techniques, which enhances security and prevents unauthorized access even in case of data exposure.

Additionally, session management is handled using token-based authentication, ensuring that users remain securely logged in while interacting with the system. This approach maintains both security and usability.

Authorization

Authorization controls what actions a user is allowed to perform within the system. SolarFlow implements role-based access control (RBAC), where users are assigned roles such as Super Admin, Company Admin, or Employee. Each role has predefined permissions that restrict access to specific modules and functionalities.

For example, administrative features such as finance and employee management are accessible only to authorized users, while general users have limited access. This ensures that sensitive operations are protected and only performed by authorized personnel.



Data Security

Data security is a critical aspect of the system, especially due to its multi-tenant nature. SolarFlow ensures tenant isolation, where each company's data is securely separated using a unique identifier. This prevents any unauthorized access or data leakage between different organizations.

Furthermore, all API endpoints are secured using authentication checks and validation mechanisms. Only authenticated and authorized requests are processed, ensuring that the system remains protected against unauthorized access and potential security threats.

VIII RESULT & DISCUSSION

System Performance

The developed SolarFlow system demonstrates efficient performance in terms of both speed and scalability. The use of modern web technologies such as optimized frontend rendering and efficient API handling ensures quick response times for user actions. Operations such as customer data retrieval, project updates, and financial tracking are performed with minimal delay, providing a smooth user experience.

In terms of scalability, the system is designed using a multi-tenant architecture, allowing multiple companies to use the platform simultaneously without performance degradation. The use of a NoSQL database further enhances scalability by efficiently handling large and dynamic datasets.

IX. FUTURE SCOPE

The SolarFlow system provides a strong foundation for managing solar business operations; however, several enhancements can be implemented in the future to further improve its capabilities and scalability.

One of the major future enhancements is the integration of AI-based analytics, which can be used to predict project delays, analyze financial trends, and provide intelligent insights for better decision-making. This would help businesses optimize their operations and improve efficiency.

Another important extension is IoT integration for solar monitoring, where real-time data from solar panels and devices can be collected and analyzed within the system. This would enable performance tracking, fault detection, and improved maintenance of solar installations.

The development of a dedicated mobile application is also a potential future improvement. Although the current system supports mobile access through a Progressive Web App (PWA), a native mobile app could provide enhanced performance, better user experience, and offline capabilities.

Additionally, integration with government subsidy APIs can streamline the subsidy application and tracking process. This would reduce manual effort, improve accuracy, and ensure faster processing of subsidy-related tasks.

Overall, these future enhancements aim to transform SolarFlow into a more intelligent, connected, and comprehensive platform for the solar industry.

X. CONCLUSION

The SolarFlow system has been successfully designed and developed as a comprehensive solution for managing solar installation businesses. It integrates multiple functionalities such as customer management, project tracking, employee management, finance handling, and inventory control into a single unified platform. By adopting a multi-tenant SaaS architecture, the system ensures scalability, data isolation, and efficient resource utilization, making it suitable for multiple organizations.

The implementation of this system has a significant impact on solar businesses by improving operational efficiency, reducing manual effort, and enhancing transparency. Features such as workflow tracking, real-time analytics, and automated calculations enable better decision-making and streamlined operations. As a result, businesses can manage projects more effectively, minimize delays, and improve overall productivity.



From a learning perspective, this project provided valuable experience in applying modern web technologies and software engineering principles to solve real-world problems. It enhanced understanding of full-stack development, database design, authentication mechanisms, and scalable system architecture. Additionally, it helped in developing problem-solving skills, project management abilities, and practical knowledge required for building industry-level applications.

REFERENCES

- [1] V. K. Mehta and Rohit Mehta, Principles of Power System, 4th ed. New Delhi, India: S. Chand Publishing, 2008.
- [2] Ministry of New and Renewable Energy (MNRE), "National Solar Mission Guidelines," Government of India, 2023.
- [3] MongoDB Inc., "MongoDB Documentation," [Online]. Available: <https://www.mongodb.com/docs>
- [4] Vercel Inc., "Next.js Documentation," [Online]. Available: <https://nextjs.org/docs>
- [5] Meta Platforms Inc., "React Documentation," [Online]. Available: <https://react.dev>
- [6] NextAuth.js, "Authentication for Next.js Applications," [Online]. Available: <https://next-auth.js.org>
- [7] Tailwind Labs, "Tailwind CSS Documentation," [Online]. Available: <https://tailwindcss.com/docs>
- [8] M. Fowler, Patterns of Enterprise Application Architecture. Boston, MA, USA: Addison-Wesley, 2002.
- [9] I. Sommerville, Software Engineering, 10th ed. Boston, MA, USA: Pearson, 2015.
- [10] SheetJS, "xlsx Library Documentation," [Online]. Available: <https://docs.sheetjs.com>

