



BVB Campus, Vidyanagar, Hubballi – 580031, Karnataka, INDIA.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

An Industry Project Report

on

Full Stack Web Development of Armtronix Management System Using MERN Stack

Submitted

in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Name USN

Under the guidance of

Dr. Karibasappa K G

Designation

School of Computer Science and Engineering

KLE Technological University, Hubballi



KLE Technological
University
Creating Value
Leveraging Knowledge

BVB Campus, Vidyanagar, Hubballi – 580031, Karnataka, INDIA.

2024-2025

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2024-25

CERTIFICATE

This is to certify that Industry Project entitled “Full Stack Web Development of Armtronix Management System Using MERN Stack” is a bonafide work carried out by the student Mr/Ms<student name here> bearing USN <student USN here> in partial fulfillment of the completion of 8th semester B. E. course during the year 2024 – 25 at Armntronix IoT Pvt. Ltd. The Industry Project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said course.

Guide Name

Dr. Karibasappa K G

Head SoCSE

Dr. Vijayalakshmi M.

External Viva-Voce

Name of the examiners

Signature with date

1 _____

2 _____

DECLARATION

I hereby declare that the Industry Project Report entitled “Full Stack Web Development of Armtronix Management System Using MERN Stack” is an authentic record of my own work as requirements of Industry, during the period from <starting date of project> to <end date of project> for the award of the degree of B.E. Under the guidance of Dr. Karibasappa K G.

Swarna Patil
01FE21BCS145

DATE:

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of a number of individuals whose professional guidance and encouragement helped me in the successful completion of this report work.

I also take this opportunity to thank Dr. Vijayalakshmi M, Professor and Head, Department of Computer Science and Engineering for having provided us academic environment which nurtured our practical skills contributing to the success of our project.

I sincerely thank our guide Dr. Karibasappa K G, for his guidance, inspiration and wholehearted co-operation during the course of completion.

I sincerely thank Mr.Hemanth Surakod, Mr. Naren Nayak, and Mr.Samarth Handral, Armtronix IoT Pvt. Ltd. for their support, inspiration and wholehearted co-operation during the course of completion.

My gratitude will not be complete without thanking our beloved parents, our seniors and our friends who have been a constant source of aspirations.

Swarna Patil
01FE21BCS145

ABSTRACT

The Armtronix Management System is a comprehensive, full-stack web application designed to optimize operational workflows across user, employee, and administrative domains. Built with the MERN (MongoDB, Express.js, React, Node.js) stack, the system incorporates modern web technologies to deliver enhanced real-time data synchronization, fortified security, and an intuitive user experience. The architecture is designed to be modular and scalable, supporting seamless CRUD (Create, Read, Update, Delete) operations facilitated through RESTful APIs.

The application is structured into three primary portals: The User Portal, which allows for secure authentication, product catalog browsing, order management, and real-time order status tracking via an intuitive interface. The Employee Portal, which streamlines internal communications, task assignments, inventory management, and facilitates form submissions for claims and leaves. Real-time updates are powered by WebSockets, ensuring immediate reflection of status changes, while automated PDF generation simplifies documentation processes. The Admin Portal serves as the central management console, providing control over user, employee, order, and event management. It includes a dynamic Calendar of Events (COE) for efficient event scheduling and monitoring.

The backend architecture is fortified with JWT-based Authentication, Role-Based Access Control (RBAC), and robust middleware for validation and error handling. Security features such as Cross-Site Scripting (XSS) Protection, are integrated to ensure data integrity and platform security. Deployment is streamlined through Docker containerization, managed via Docker Compose, facilitating consistent and isolated environments. Additionally, real-time data synchronization is powered by WebSockets, while Nodemailer handles automated email notifications efficiently.

The Armtronix Management System provides a scalable, secure, and real-time operational platform that enhances administrative oversight, employee productivity, and user engagement. Its modular design and robust security mechanisms make it a viable solution for modern enterprise applications, guaranteeing reliability, security, and high performance.

Keywords :Secure Authentication, Role-Based Access Control (RBAC), JWT Authentication, Docker Containerization, WebSockets.

CONTENTS

DECLARATION	1
ACKNOWLEDGEMENT	2
ABSTRACT	i
CONTENTS	iii
LIST OF TABLES	iv
LIST OF FIGURES	vi
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Literature Review / Survey	2
1.3 Objectives	4
1.4 Problem Statement	5
2 REQUIREMENT ANALYSIS	6
2.1 System Model	6
2.2 Functional Requirements	6
2.2.1 User Portal	7
2.2.2 Employee Portal	7
2.2.3 Admin Portal	7
2.3 Non Functional Requirements	8
2.4 Database Requirements	9
3 SYSTEM DESIGN	11
3.1 Architecture Design	11
3.2 Data Flow Diagram	12
3.3 Use Case Diagram	13
3.3.1 User Portal Use Cases:	13
3.3.2 Employee Portal Use Cases:	13
3.3.3 Admin Portal Use Cases:	14

4 IMPLEMENTATION	15
4.1 Introduction	15
4.2 System Architecture	15
4.3 Middleware Implementation	15
4.4 Data Flow Diagrams (DFD)	16
4.5 REST API Structure	16
4.6 Frontend Integration	16
4.7 PDF Generation Logic	17
4.8 WebSocket Event Flow	17
4.9 Session Management and Token Expiry	18
4.10 Deployment Strategy	18
4.11 Security Enhancements	18
4.12 Error Handling and Logging	18
4.13 Conclusion	19
5 TESTING	20
5.1 Integration Testing	20
5.2 Acceptance Testing	21
6 RESULTS AND DISCUSSIONS	23
6.1 User Portal	23
6.2 Employee Portal	30
6.3 Admin Portal	35
7 CONCLUSION AND FUTURE SCOPE OF THE WORK	41
7.1 Conclusion	41
7.2 Future Scope:	41
REFERENCES	43

LIST OF TABLES

LIST OF FIGURES

6.1	Home Page	23
6.2	Home Page	24
6.3	Products Page	25
6.4	Product Page	26
6.5	Solutions Page	27
6.6	About Page	28
6.7	Contact Page	29
6.8	Login Page	30
6.9	Employee Dashboard	30
6.10	Git Portal	30
6.11	Cloudflare portal	31
6.12	AI/GPT portal	31
6.13	Annotation portal	31
6.14	Claim Form	32
6.15	Leave Form	33
6.16	Leave Status	33
6.17	Calendar of Events(COE)	33
6.18	Employee Profile	34
6.19	Tasks Assigned	34
6.20	Attendance and Leave Tracking	34
6.21	Admin ClaimForm	35
6.22	Payment Details	36
6.23	Reports	36
6.24	Register page	36
6.25	User Info	37
6.26	Employee Details	37
6.27	Employee Details	37
6.28	Employee Leave settings	38
6.29	Leave settings	38
6.30	Calendar of Events(COE)	38
6.31	Leave Notifications	39
6.32	Admin Profile	39
6.33	Tasks Assignment	39
6.34	Add Tasks	40

6.35 Attendance and Leave Tracking	40
--	----

Chapter 1

INTRODUCTION

The rapid evolution of web technologies has paved the way for powerful, interactive, and scalable applications that enhance user experiences. Among the numerous technology stacks available, the MERN stack—comprising MongoDB, Express.js, React, and Node.js—stands out as a preferred choice for building full-stack web applications. The MERN stack's unique advantage lies in its end-to-end JavaScript usage, simplifying both server-side and client-side development while ensuring faster data exchange and seamless integration.

MongoDB, as a NoSQL database, provides high scalability and flexibility, allowing developers to store and manage vast amounts of unstructured data with ease. Its schema-less nature enables quick iterations during development, making it a popular choice for agile projects. Express.js, a lightweight and minimalist web application framework, streamlines backend development with its powerful routing and middleware capabilities. React, maintained by Meta (formerly Facebook), has redefined front-end development through its component-based architecture and virtual DOM, optimizing rendering efficiency and enhancing user experiences. Node.js, a server-side runtime environment, allows JavaScript to run natively on the server, enabling asynchronous processing for faster response times and real-time updates.

This internship project focused on harnessing the capabilities of the MERN stack to develop a comprehensive web application. The project included designing intuitive user interfaces, implementing secure and efficient REST APIs, managing real-time database operations, and deploying the application for practical use. This hands-on experience facilitated a deeper understanding of full-stack development and modern web application architecture.

1.1 Motivation

The motivation behind choosing the MERN stack for this internship project is rooted in its widespread industry adoption and efficient development lifecycle. In the current digital age, the demand for dynamic, scalable, and responsive web applications has surged, prompting developers to seek technology stacks that are both flexible and powerful. MERN stack's unified JavaScript ecosystem eliminates the need for context switching between different pro-

gramming languages, thereby optimizing development speed and minimizing errors. Moreover, the asynchronous, non-blocking nature of Node.js ensures that applications remain highly performant even under heavy load, which is crucial for modern web platforms. MongoDB's ability to handle massive data operations without rigid schema constraints allows for rapid prototyping and scalable data management. Express.js, with its minimalistic structure, streamlines backend logic, while React enhances user interaction with its reactive UI components.

Undertaking this project was driven by the desire to master full-stack development using industry-standard technologies. Gaining proficiency in MERN stack not only improves employability but also lays a strong foundation for contributing to large-scale applications in the tech industry. This project marks the beginning of exploring scalable solutions, cloud-based deployment, and API integrations with real-world implications.

1.2 Literature Review / Survey

The MERN stack has emerged as a dominant technology for building single-page applications (SPAs) and real-time web platforms. Several studies have explored the components and applications of the MERN stack, highlighting its advantages, scalability, and ease of development in modern web applications. With the shift towards JavaScript-based full-stack development, MERN has become a preferred choice for developers seeking consistency across server-side and client-side coding.

Sravani et al. [1] developed a dynamic web application named "Study Buddy" using the MERN stack to foster collaboration among students. The application includes features like authentication, Q&A forums, complaint posting, idea sharing, and chatbot assistance, demonstrating the stack's capability to build comprehensive educational platforms. The study also highlights the real-time communication enabled by WebSockets in a Node.js environment, showcasing enhanced user engagement.

Sahni et al. [2] designed a themed education platform for placement preparation using the MERN stack. Their work emphasizes the stack's effectiveness in creating interactive and responsive educational tools, catering to the evolving needs of students preparing for placements. The application leveraged React for modular front-end development, while MongoDB handled scalable data storage with flexible schema designs.

Kulkarni et al. [3] explored the development of an e-learning platform utilizing the MERN stack. Their research underscores the stack's flexibility and scalability in building platforms that offer flexible, accessible, and engaging educational experiences. The study also demonstrates how MERN applications can integrate third-party APIs seamlessly, enhancing learning

platforms with external resources.

Prakash and Pathak [4] discussed the implementation of a social media application using the MERN stack. Their study highlights the stack's proficiency in handling dynamic content and real-time interactions, essential features for social networking platforms. They emphasize the use of Redux for state management, enabling efficient handling of user interactions and live data updates.

Vasan Subramanian's book [5] provides an in-depth guide to building full-fledged web applications using the MERN stack. The author discusses industry best practices and tools like React Router, React-Bootstrap, Redux, Babel, and Webpack, offering valuable insights into the stack's components and their integration. Furthermore, the book covers optimization techniques for enhancing performance and reducing load times.

Porter et al. [6] designed and implemented a RESTful IoT service using the MERN stack. Their work demonstrates the stack's applicability beyond traditional web applications, extending its use to IoT services requiring efficient data handling and real-time communication. The integration of MQTT with Node.js in their project highlights MERN's capability in supporting lightweight, fast communication protocols.

Vasanthi et al. [7] developed a car rental website employing the MERN stack. Their project showcases the stack's capability to manage complex functionalities like booking systems, user authentication, and payment integrations in commercial applications. The authors emphasized the integration of payment gateways such as Stripe for seamless transactions, leveraging Express for secure backend processing.

Hemalatha et al. [8] focused on developing a flexible car sharing and leasing platform using the MERN stack. Their research illustrates the stack's adaptability in creating applications that require robust backend processing and user-friendly interfaces. They also incorporated Google Maps APIs for real-time location tracking and route optimization.

Vinothini et al. [9] conducted a comprehensive survey on NBA accreditation and the implementation of a web portal using the MERN stack. Their study emphasizes the stack's suitability for developing administrative and informational portals in educational institutions. MongoDB's flexibility enabled dynamic schema modifications, aligning with changing accreditation standards.

Kanchanamala et al. [10] explored automated programming evaluation using the MERN stack. Their work highlights the stack's potential in building platforms that assess coding assignments, providing immediate feedback and enhancing the learning experience. The integration of code compilers within the Node.js environment enabled real-time evaluation and error detection.

Chauhan et al. [11] developed a food ordering website using the MERN stack. Their project demonstrates the stack's effectiveness in creating responsive and interactive e-commerce applications, handling tasks like menu management, order processing, and user authentication.

They optimized database queries using MongoDB's aggregation framework for faster transaction processing.

Furthermore, recent studies have explored MERN's use in decentralized applications (DApps), where its architecture is leveraged alongside blockchain technology for building secure, distributed platforms. Researchers have also examined its role in microservices architecture, enhancing modularity and scalability for large-scale applications.

These studies collectively underscore the MERN stack's versatility and effectiveness in modern web development, making it a valuable choice for developers aiming to build comprehensive and responsive web applications. Its consistent use of JavaScript across all tiers—database, server, and client—streamlines the development process and enhances maintainability, further contributing to its widespread adoption.

1.3 Objectives

1. To design and develop a fully functional web application for Armtronix that includes three major portals:
 - User Portal: Allows customers to explore products, solutions, and access support services.
 - Employee Portal: Facilitates internal communication, task management, and resource sharing for Armtronix employees.
 - Admin Portal: Manages overall operations, including user management, employee monitoring, and data analytics.
2. To ensure seamless data flow and communication between the three portals using REST APIs built with Express.js and Node.js.
3. To implement secure authentication and authorization mechanisms for different user roles (User, Employee, Admin) using JWT (JSON Web Token).
4. To create an interactive, responsive, and user-friendly interface with React and CSS frameworks for optimal user experience across all devices.
5. To establish a scalable and efficient database structure with MongoDB for storing user, employee, and product information securely.
6. To optimize performance and enhance real-time communication with modern development practices, ensuring faster load times and smooth navigation.
7. To integrate real-time notifications and updates across all portals for better communication and user engagement.

1.4 Problem Statement

In the rapidly evolving technological landscape, Armtronix requires a robust and efficient web platform that can streamline its operations, enhance customer interaction, and improve employee productivity. The current system lacks centralized management, real-time communication, and integrated user portals that can handle diverse functionalities effectively. This project aims to build a full-stack web application leveraging the MERN (MongoDB, Express.js, React, Node.js) stack to overcome these challenges.

The proposed solution will include three dedicated portals:

- The User Portal will enable customers to interact with Armtronix services, view product & solution details, and access customer support with ease.
- The Employee Portal will streamline internal communication, task allocation, and resource management, enhancing operational efficiency.
- The Admin Portal will provide a centralized dashboard to monitor activities, manage users, handle inventory, and generate analytical reports for data-driven decision-making.

This unified platform is designed to enhance productivity, improve user experience, and enable real-time updates, thereby aligning with Armtronix's growth and digital transformation goals.

Chapter 2

REQUIREMENT ANALYSIS

The requirement analysis phase is essential for defining the core functionalities and performance metrics of the system. By clearly outlining both functional and non-functional requirements, we ensure that the project meets the business objectives and user needs while providing a solid technical foundation.

2.1 System Model

The system model of the Armtronix web application consists of three primary portals: **User Portal**, **Employee Portal**, and **Admin Portal**. These portals are designed to serve different stakeholders and provide tailored functionality for each group while utilizing a shared backend infrastructure for efficient management.

- **User Portal:** A customer-facing interface where users can explore products, solutions, and access customer support.
- **Employee Portal:** Used by employees to manage inventory, and communicate within the company.
- **Admin Portal:** A management-level interface where administrators can oversee user and product data, and generate reports.

The application follows a **Client-Server Architecture** where the frontend (React) interacts with the backend (Node.js/Express) via RESTful APIs, and MongoDB serves as the database. The system is scalable, flexible, and modular to allow future enhancements, like integrating new user roles or modules.

2.2 Functional Requirements

The functional requirements describe the essential features the system must support. These requirements are categorized into three main portals: User Portal, Employee Portal, and Admin Portal.

2.2.1 User Portal

The User Portal provides functionality for end-users and customers:

1. **User Authentication:** The system shall allow users to sign up, log in, and reset passwords securely.
2. **Product Catalog:** The system shall display an interactive product catalog with details such as name, description, price, and images.
3. **Customer Support:** The system shall provide live chat, FAQs, and email support for users.

2.2.2 Employee Portal

The Employee Portal supports internal staff operations:

1. **Employee Authentication:** The system shall provide secure login for employees with role-based access control.
2. **Inventory Management:** The system shall allow employees to add, update, and remove product details.
3. **Task Management:** The system shall allow task assignments to employees and monitor the completion status.
4. **Internal Communication:** The system shall provide a messaging system for employees to communicate with each other.

2.2.3 Admin Portal

The Admin Portal provides system management capabilities:

1. **Admin Authentication:** The system shall authenticate admins with secure login credentials and provide full access control over the system.
2. **User Management:** The system shall allow admins to view, manage, and delete user accounts.
3. **Reporting:** The system shall allow admins to generate reports on user activity, sales, and inventory.
4. **Role Management:** The system shall allow admins to manage employee roles, permissions, and access levels.

2.3 Non Functional Requirements

Non-functional requirements define the overall system attributes like performance, scalability, and security. These should be **quantitative** to ensure the system meets specific performance targets:

1. Performance:

- The system should support a minimum of **500 concurrent users** without significant performance degradation.
- The average page load time should not exceed **3 seconds** for product pages.
- Search queries for product listings should be completed within **2 seconds**.

2. Scalability:

- The system should be able to support the addition of **1,000 new users per day** without performance impact.
- The backend should scale horizontally to support increased load by adding additional nodes as required.

3. Security:

- All user passwords should be encrypted using **bcrypt** with a minimum of **12 rounds of hashing**.
- The application should utilize **HTTPS** for secure data transmission.
- The system should be protected against **SQL Injection, XSS, and CSRF** attacks through proper validation and sanitization.

4. Usability:

- The system should be responsive and function on devices with screen sizes ranging from **320px to 1920px**.
- The application should pass the **WCAG 2.1** accessibility guidelines to ensure it is usable by people with disabilities.

5. Availability:

- The system should ensure **99.9% uptime**, excluding scheduled maintenance, and must have a disaster recovery plan to restore service within **4 hours** after any system failure.

6. Compliance:

- The system should comply with **GDPR** for data protection, ensuring user data privacy and security.
- The system should implement **audit trails** for sensitive operations, allowing traceability of actions performed by users and admins.

2.4 Database Requirements

The database plays a critical role in storing and managing application data. The requirements for the database are as follows:

1. User Data:

- The system shall store user authentication data, including securely hashed passwords.
- The database shall efficiently support querying user information based on user IDs and email addresses.

2. Product Data:

- The system shall store product details such as name, description, price, stock quantity, and category.
- Product data shall be indexed on fields like **product name** and **category** for fast searches.

3. Employee Data:

- The system shall store employee details, including roles, tasks, and permissions.
- Employee data shall be securely stored and accessible only by authorized employees or admins.

4. Analytics Data:

- The system shall log user activities and orders for reporting and analytics purposes.
- The data shall be aggregated periodically to generate reports on sales, user engagement, and product performance.

5. Backup and Redundancy:

- The system shall implement daily **automatic backups** of all data, ensuring that no more than **24 hours** of data is lost in case of a failure.

- Database replication shall be implemented across multiple nodes to ensure high availability and fault tolerance.

Chapter 3

SYSTEM DESIGN

The system design defines the overall structure and behavior of the application. It includes the architecture, data flow, and interaction between different modules of the system, ensuring seamless communication between the User Portal, Employee Portal, and Admin Portal.

3.1 Architecture Design

The architecture of the Armtronix Web Application is based on a **3-tier architecture** that includes:

1. Presentation Layer (Frontend):

- Built with **React** to create dynamic, responsive user interfaces for User, Employee, and Admin portals.
- Handles user interactions, form submissions, and real-time updates.

2. Application Layer (Backend):

- Implemented with **Node.js** and **Express**.
- Manages business logic, routing, and API calls.
- Provides secure communication with the database and processes client requests.

3. Data Layer (Database):

- Utilizes **MongoDB** for storing application data, such as user details, employee records, and administrative information.
- Ensures data consistency and integrity through ACID transactions.

Communication Flow:

- The frontend communicates with the backend using **REST APIs**.
- The backend interacts with the database to fetch, update, or delete information as required by user actions.
- Data is sent back to the frontend in JSON format, which is rendered dynamically.

3.2 Data Flow Diagram

The Data Flow Diagram (DFD) represents the flow of information within the system, covering all implemented features:

1. User Registration and Login:

- User enters credentials → Frontend sends request → Backend validates → MongoDB stores/retrieves data → Response sent back to frontend.

2. Admin Dashboard Navigation:

- Admin logs in → Access to Git, Cloudflare, AI/GPT, Annotation, Claim Form, and Leave Form portals → Backend fetches corresponding data from MongoDB → Displays on frontend.

3. Employee Portal Access:

- Employees log in → Frontend sends request → Backend verifies credentials → MongoDB fetches associated tasks and profile data.

4. Claim Form Submission:

- Employee submits a claim → Frontend sends data → Backend validates and stores it in MongoDB → Admin reviews and processes the claim → PDF is generated, and UTR tracking is enabled.

5. Leave Form Submission:

- Employee submits a leave request → Frontend sends data → Backend processes → Admin approves or rejects → Real-time notification sent to the employee.

6. Check-in/Check-out and Attendance Tracking:

- Employee checks in/out → Frontend captures timestamp → Backend logs entry into MongoDB → Data is processed for attendance records.

7. COE System Management:

- Admin adds or updates events and holidays → Frontend updates in real-time → Employees view the COE calendar.

8. Profile Management:

- Admin registers employees → Assigns roles and updates profile details → Frontend reflects changes in real-time.

9. Notification System:

- Real-time updates are sent for claim approvals, leave status, and event updates → WebSocket integration ensures live synchronization.

3.3 Use Case Diagram

The Use Case Diagram visually represents the interactions between users (User, Employee, Admin) and the system, including the new features:

3.3.1 User Portal Use Cases:

- Register Account
- Log In
- View Profile
- Update Profile
- Check-in and Check-out
- View COE (Calendar of Events)

3.3.2 Employee Portal Use Cases:

- Log In
- View Assigned Tasks
- Submit Claim Form
- Submit Leave Form
- Track Leave Status
- Receive Real-time Notifications
- Update Personal Information
- Check Attendance Records

3.3.3 Admin Portal Use Cases:

- Log In
- Manage Users
- Update Employee Profiles
- Approve/Reject Claims
- Approve/Reject Leave Requests
- Modify COE (Calendar of Events)
- Monitor System Logs
- Navigate to Git, Cloudflare, AI/GPT, Annotation portals
- Modify Leave Settings
- Track Attendance and Check-ins

Chapter 4

IMPLEMENTATION

4.1 Introduction

The implementation of the **Armtronix Management System** was carried out using the MERN (MongoDB, Express.js, React, Node.js) stack. This full-stack JavaScript solution facilitated smooth development across both the client and server sides, ensuring scalability, maintainability, and high performance.

4.2 System Architecture

The system is divided into three major portals:

1. **User Portal**
2. **Employee Portal**
3. **Admin Portal**

Each of these portals is interconnected through RESTful APIs, allowing seamless data exchange and real-time updates.

4.3 Middleware Implementation

Middleware is implemented for several critical operations in the backend to ensure security and data integrity:

- **Authentication Middleware:** JWT-based token verification is used to authenticate users before granting access to protected routes. Middleware checks for valid tokens, and unauthorized attempts are rejected with a 401 status.
- **Role-Based Access Control (RBAC):** Specific routes are restricted based on user roles (e.g., Admin, Employee). This ensures that only authorized users can perform specific CRUD operations.

- **Validation Middleware:** All incoming requests are validated for required parameters. For example, during registration, user details are validated for format and completeness before database insertion.
- **Error Handling Middleware:** Centralized error handling is configured to capture runtime exceptions, database errors, and unhandled promise rejections, ensuring proper HTTP responses.

4.4 Data Flow Diagrams (DFD)

To illustrate the interaction between the frontend, backend, and database, the following data flow diagrams represent key modules:

- **User Authentication DFD:** Displays the flow from login form submission → backend validation → JWT generation → client-side token storage.
- **Claim Submission DFD:** Represents data flow from employee claim form → backend validation → MongoDB storage → status update and email notification.
- **Leave Request DFD:** Shows the path from leave form submission → database insertion → admin approval → real-time notification → mail service trigger.
- **COE Update DFD:** Outlines the process of creating or modifying events → database update → real-time sync with all portals using WebSockets.

4.5 REST API Structure

The backend services follow REST principles for clear and structured communication. Each API is protected with JWT authentication and role-based access control to prevent unauthorized access.

4.6 Frontend Integration

The frontend of the system is built using React with state management handled through `useState` and `useEffect`. Components are structured to make REST API calls using Axios. Real-time updates are achieved through WebSockets, ensuring that any backend changes are instantly reflected on the UI.

- **Data Fetching:** Components fetch data during the component mount lifecycle using Axios.

- **UI Rendering:** React dynamically updates the UI based on state changes, ensuring a smooth experience.
- **WebSocket Sync:** Whenever inventory, claims, or leaves are updated, WebSocket events trigger frontend state updates.
- **Form Submission:** Claim forms and leave requests are submitted using Axios, with real-time status updates displayed upon submission.

4.7 PDF Generation Logic

For claim forms, the backend uses the `pdfkit` and `fs` modules to dynamically generate PDF documents. When a claim form is submitted:

1. Backend receives the form data.
2. Generates a structured PDF document with all form details.
3. Saves the PDF to a secure server location.
4. Sends the download link to the user's registered email using `Nodemailer`.

This automation simplifies documentation and improves accessibility for end-users.

4.8 WebSocket Event Flow

WebSockets are used extensively for real-time data synchronization:

- **Task Assignments:** When a new task is assigned, it is pushed to the assigned employee's dashboard instantly.
- **Leave Status Updates:** Admin approval or rejection of leaves is instantly updated for the employee.
- **Inventory Updates:** Changes to product quantities are reflected in real-time.
- **COE Events:** Any addition or modification to calendar events is synced across all user dashboards.

4.9 Session Management and Token Expiry

Session management is handled via JWT tokens:

- **Token Expiry:** Each token is configured with an expiry time to enhance security.
- **Auto-Logout:** Frontend listens for token expiry events and auto-logs out users securely.
- **Refresh Logic:** Secure refresh mechanisms are in place to extend session validity upon user interaction.

4.10 Deployment Strategy

The application is containerized using Docker for simplified deployment:

- **Dockerfiles:** Configured for both frontend and backend services.
- **Docker Compose:** Orchestrates multi-container applications for frontend, backend, and database.
- **Service Isolation:** Each service runs in isolated containers, enhancing modularity.

4.11 Security Enhancements

To prevent vulnerabilities:

- **XSS Protection:** Input fields are sanitized to prevent cross-site scripting attacks.
- **CSRF Protection:** Anti-CSRF tokens are implemented for all form submissions to prevent unauthorized state changes.
- **SQL Injection Prevention:** MongoDB's parameterized queries prevent injection attacks.
- **Rate Limiting:** API routes are rate-limited to prevent brute-force attacks on authentication routes.

4.12 Error Handling and Logging

- **Error Logging:** All errors are captured using Winston for structured logging and stored in separate log files.

- **Global Error Handler:** A centralized error handler in Express.js catches unhandled errors and prevents server crashes.
- **Request Logging:** All API calls are logged with timestamps and user information for debugging.

4.13 Conclusion

The implementation of the Armtronix Management System was executed with a modular and scalable architecture. The three portals are well-integrated, ensuring real-time data synchronization and a seamless user experience. MERN stack's unified JavaScript environment significantly accelerated development while maintaining a high level of efficiency and security.

Chapter 5

TESTING

5.1 Integration Testing

Integration Testing focuses on verifying the seamless interaction between different modules of the Armtronix Management System. It ensures that components like the User Portal, Employee Portal, and Admin Portal communicate effectively with backend services and the MongoDB database.

Approach:

- The testing was performed using the Top-Down Integration Approach, beginning with high-level modules such as User Authentication, Admin Dashboard, and progressing towards lower-level components like Notification Systems and Calendar Management.
- Mock data was used to validate REST API interactions, ensuring that data retrieval, submission, and updating processes worked as expected.
- Modules were tested for data consistency, security, and response times to confirm smooth integration without loss of data integrity.

Key Modules Tested:

1. User Authentication: Verifying login, registration, and role-based access control.
2. Employee Portal Navigation: Testing navigation to Git, Cloudflare, AI/GPT, Annotation, Claim Form, and Leave Form portals.
3. Admin Dashboard Operations: Ensuring admin functionalities like employee registration, role assignment, and data modification work flawlessly.
4. Notifications System: Testing real-time updates for leave requests, claim submissions, and task assignments.
5. Calendar of Events (COE): Verifying real-time synchronization of events across user dashboards.
6. Leave and Claim Forms: Ensuring form submissions reflect correctly in the database and update status in real-time.

Test Results:

- All integrated modules performed as expected, with no loss of data integrity.
- Response times for data retrieval and updates were within the acceptable range (< 2 seconds).
- Authentication and session management were secure, preventing unauthorized access.

5.2 Acceptance Testing

Acceptance Testing was conducted to ensure the complete system meets the business requirements and end-user expectations. It involved real-world scenarios to validate the functionality, usability, and reliability of the application.

Approach:

- User Acceptance Testing (UAT) was carried out with real employees and admin users from Armtronix to validate the system against practical use cases.
- Test cases were designed to simulate everyday operations like employee check-ins, claim submissions, leave requests, and task updates.
- Real-time notifications, COE updates, and role-based access were verified during live sessions.

Key Scenarios Tested:

1. Employee check-in and check-out logging.
2. Real-time status updates for leave approvals and claims processing.
3. Secure role-based access to Git, Cloudflare, and AI/GPT portals.
4. Accurate synchronization of calendar events in the COE module.
5. Error-free navigation and task assignments through the Admin Dashboard.

Test Results:

- All business requirements were successfully validated.
- Users were able to navigate the system smoothly and access data without delays.
- System reliability was confirmed with zero crashes during testing.

The implementation of the Armtronix Management System was executed with a modular and scalable architecture. The three portals are well-integrated, ensuring real-time data synchronization and a seamless user experience. MERN stack's unified JavaScript environment significantly accelerated development while maintaining a high level of efficiency and security.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 User Portal

Home Page



Figure 6.1: Home Page

The Home Page serves as the landing page for users accessing the Armtronix application. It showcases key features, recent announcements, and quick navigation links to important sections such as **Products**, **Solutions**, **About**, and **Contact**.

Some of Our Products



Master Control Unit

The Master Control Unit (MCU) is a versatile system capable of simultaneously receiving data from multiple data acquisition and edge control devices.

[Read More about MCU](#)



Modular 4 DIO

Modular 4 DIO is designed specially for use with our Master devices for adding 4 digital inputs & outputs & expanding 8 additional programmable I/O points.

[Read More about MDIO](#)



Retro to IIoT

Retro to IIoT is developed for industrial automation applications to reduce the problem of laying cables for remote monitoring and control.

[Read More about R-IIOT](#)



IndusPi

IndusPi is an Industrial IoT device for automation, collecting and processing data from sensors and control devices.

[Read More about IndusPi](#)



Improvar

An Image Processing Variant system captures and analyzes visual data to extract information, make decisions, or control other systems.

[Read More about Improvar](#)



Presenter

This device is developed to interact with peripheral like barcode scanners, RFID readers, displays like TV or monitor.

[Read More about Presenter](#)

Our Esteemed Customers



TATA



AEQUUS[®]
ecosystems of efficiency



TATA MOTORS
Connecting Aspirations



What Our Clients Say About Us?

"We had a problem with the bending machine resulting in its inefficient utilization. The Armtronix team provided us with a machine monitoring system which was economical as well as industry 4.0 based. This solution has helped our production team to utilize the machine efficiently as they are able to track total production per shift, machine idle time and also generate reports by sitting at their desk and downloading it from the browser. Armtronix systems have truly enabled us to save time and work more efficiently."

- Assistant Manager (Electrical Maintenance)
Automotive Industry

"There was a very specific requirement by our production head. Our unit has many lines and it is practically very difficult to track what is happening in our unit with respect to each line. Our ask from Armtronix was to provide us a solution that provides an update on a TV/display on each line with the type of product to be manufactured per shift and the total quantity to be manufactured. Moreover, this system had to provide feedback to the line members on the quantity that they had produced and the efficiency of the production line. The system also had to show on a big display placed in our production head's cabin on what is happening on each line."

- Deputy Manager (Electrical Maintenance)
Automotive Industry

"We are highly satisfied with the solutions provided by Armtronix IoT Pvt. Ltd. The Production and efficiency monitoring system, along with the machine idle time monitoring system, have greatly improved our manufacturing processes. Real-time monitoring, operator tracking, and machine idle time insights have enhanced decision-making and productivity. Features like real-time data processing and intuitive dashboards have optimized operations. Alerts for idle times, while historical reports help track performance and identify improvement areas. Overall, Production and efficiency monitoring system, have increased productivity, reduced downtime, and saved costs. We appreciate Armtronix's support in delivering these effective solutions."

- R D Khobare (Director)
Fine Blanking India Pvt Ltd

"We are highly satisfied with the solutions provided by Armtronix IoT Pvt. Ltd. The Production and efficiency monitoring system, along with the machine idle time monitoring system, have greatly improved our manufacturing processes. Real-time monitoring, operator tracking, and machine idle time insights have enhanced decision-making and productivity. Features like real-time data processing and intuitive dashboards have optimized operations. Alerts for idle times, while historical reports help track performance and identify improvement areas. Overall, Production and efficiency monitoring system, have increased productivity, reduced downtime, and saved costs. We appreciate Armtronix's support in delivering these effective solutions."

- R D Khobare (Director)
Fine Blanking India Pvt Ltd

[Home](#) [Products](#) [Solutions](#) [About Us](#) [Contact](#)

© 2025 Armtronix. All rights reserved.

Figure 6.2: Home Page

Products Page

Explore our wide array of components and discover the potential for excellence in every detail. Together, we build a future of precision and reliability.

[Contact Us](#)

Some of Our Products

Master Control Unit The Master Control Unit (MCU) is a versatile system capable of simultaneously receiving data from multiple data acquisition and edge control devices. Read More	Modular 4 DIO Modular 4 DIO is designed for adding digital inputs and outputs and expanding additional programmable I/O points. Read More	Retro to IoT Retro to IoT is developed for industrial automation applications to reduce the problem of laying cables for remote monitoring and control. Read More
IndusPi IndusPi is an Industrial IoT device for automation, collecting and processing data from sensors and control devices. Read More	Improvar An Image Processing Variant system captures and analyzes visual data to extract information, make decisions, or control other systems. Read More	Presenter This device is developed to interact with peripherals like barcode scanners, RFID readers, displays like TV or monitor. Read More

Home Products Solutions About Us Contact

 © 2023 Armtronix. All rights reserved.

Figure 6.3: Products Page

The Products section displays a list of available products and services provided by Armtronix. Users can browse through categories, view product descriptions, and access additional details by clicking on individual items.

The Product Page provides a detailed view of a selected product, including specifications, features, and pricing. It also contains a contact form for users to make inquiries or request more information.

ARMTRONIX IoT
DATA DRIVEN CONTROL

Home Products Solutions About Contact

IndusPi

Leading the way in IoT automation!

IndusPi

The IndusPi is an Industrial IoT/Industry 4.0 based automation field programmable device. In addition to accepting data from sensors, user inputs, and pre-programmed processes, it may concurrently receive data from numerous data acquisition and edge control devices. After receiving the data, it will calculate it and store it in its local database. Anytime they choose, users can access and analyse that data. In order to monitor and manage the end devices, the device can also transmit commands to data acquisition and other edge control devices. It has the ability to transfer data between servers on the local network and the cloud server. Additionally, it can translate the languages or protocols used by local devices into those used by the cloud, enabling even more remote access and analysis. In addition to above it come with suitable ports for connecting displays like monitors, Smart TVs, touch screen etc as (Human Machine Interface) HMI like applications.

IndusPi is compatible with Node-RED, which is highly versatile. Node-RED is a programming tool for wiring together hardware devices and APIs in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

Few of the benefits of this device are Reduced Latency, Enhanced Security and Reliability.



Lateral Perspective

Overhead Perspective

Drag to spin

Specifications:

- ✓ 2 x Ethernet
- ✓ 2 x USB 2.0
- ✓ 1 x USB Type C
- ✓ 4 x Digital Output
- ✓ 4 x Digital Input
- ✓ 1 x WiFi (Optional)
- ✓ 1 x Micro SD card slot
- ✓ 1 x HDMI

[Download Product Info \(PDF\)](#)

Home Products Solutions About Us Contact

YouTube X Facebook LinkedIn

© 2025 Armtronix. All rights reserved.

Figure 6.4: Product Page

Solutions Page

Production Monitoring System

Machine monitoring in factories is the method of collecting and analyzing data by gathering information from machines with the help of advanced technology. This process includes collecting data from machines, transferring that data to local or cloud servers for analysis to improve performance. The machine monitoring system helps manufacturers predict abnormal behavior of machines, reduce downtime, and identify areas for improvement.

Key Features:

- ✓ Real-time automated data logging
- ✓ Data transparency and data manipulation
- ✓ Error-free and on-time reporting
- ✓ Eliminate manual reporting

Figure 6.5: Solutions Page

The Solutions Page highlights various technological solutions offered by Armtronix, showcasing expertise in IoT, AI, and Cloud Computing.

About Page

WELCOME TO

Armtronix IoT

DATA DRIVEN CONTROL

Armtronix – (AutomationRoboticsMechaTronix) IoT Pvt.Ltd was founded in the year 2021 and it was selected by the Ministry of Electronics and Information Technology (MeitY), Government of India for its Technology Incubation and Development of Entrepreneurs (TIDE 2.0) funding scheme which is incubated in KLE Technological University, Vidyanager, Hubballi.

VISION
Optimal management of techno systems in industry using automation, robotics, and mechatronics.

MISSION
Developing cost-effective products for smooth transition to data-driven control by adopting newer technologies in industry.

OBJECTIVE
To become one of the global leaders in industrial automation solution providers.

Our Achievements

- Technology Incubation**
Selected by the Ministry of Electronics and Information Technology (MeitY), Government of India for the TIDE 2.0 funding scheme.
- Global Recognition**
Recognized for innovation in industrial automation and IoT solutions on a global platform. Recognized by Startup India.
- Successful Product Launches**
Successfully launched a range of cost-effective industrial automation products that revolutionized the market.
- Green Energy, Clean Energy Competition**
Participated in the Tata motors Green Energy, Clean Energy Program.

Our Values

- Innovation**
We value creativity and continuously strive to innovate in the field of automation, robotics, and IoT technologies.
- Integrity**
We uphold the highest standards of integrity and honesty in our business dealings and relationships with clients and partners.
- Excellence**
We are committed to delivering high-quality products and services that exceed customer expectations.
- Collaboration**
We believe in teamwork and collaboration, both within our company and with our clients, to achieve common goals.

Figure 6.6: About Page

This section provides an overview of Armtronix's mission, vision, and company history, as well as its commitment to technological innovation.

Contact Page

Find Us Here

Armtronix IoT Private Limited
94C/F-C9H, Vidyavihar, Hubballi,
Karnataka 580031
[View larger map](#)

Our Address

Location
Armtronix IoT Private Limited, First Floor, KLE Tech Park Building, KLE Technological University, Vidyavihar, Hubballi, Karnataka-580031

Email & Phone
info@armtronix.in
+91 98803 10042

Business Hours
Monday - Saturday: 9:30 AM - 6:30 PM

Home Products Solutions About Us Contact

© 2025 Armtronix. All rights reserved.

Figure 6.7: Contact Page

The Contact Page allows users to reach out to Armtronix for business inquiries, support requests, or partnership opportunities. It includes a contact form, address, and phone number for quick communication.

6.2 Employee Portal

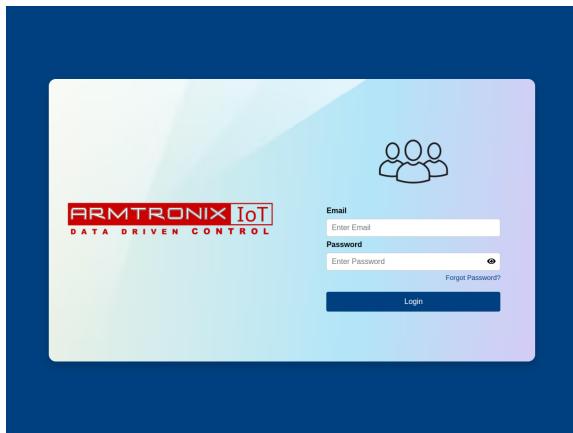


Figure 6.8: Login Page

Login Page

The login page serves as the entry point for authenticated access to the Armtronix application. Users, employees, and admins enter their credentials, which are verified against the MongoDB database using backend logic in Node.js and Express. Secure authentication mechanisms like **JWT (JSON Web Tokens)** are employed to manage sessions and prevent unauthorized access.

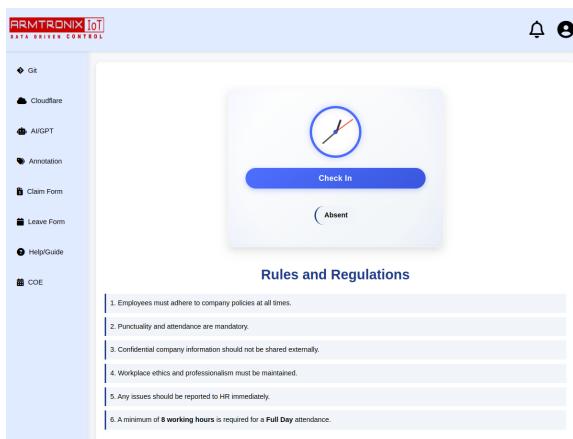


Figure 6.9: Employee Dashboard

Employee Dashboard

The Employee Dashboard provides a quick overview of the employee's tasks, notifications, and important updates. It displays relevant information such as pending tasks, recent claims, and leave status. Employees can easily navigate to different portals like Git, Cloudflare, AI/GPT, Annotation, Claim Form, and Leave Form from this dashboard.

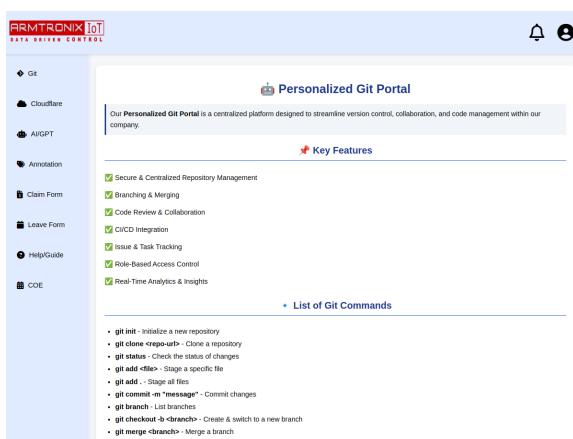


Figure 6.10: Git Portal

Git Portal

The Git Portal presents a centralized interface for version control management. Employees can view Git documentation, access repositories, and monitor CI/CD (Continuous Integration/Continuous Deployment) processes. The interface also allows for secure repository management and tracking of code changes.

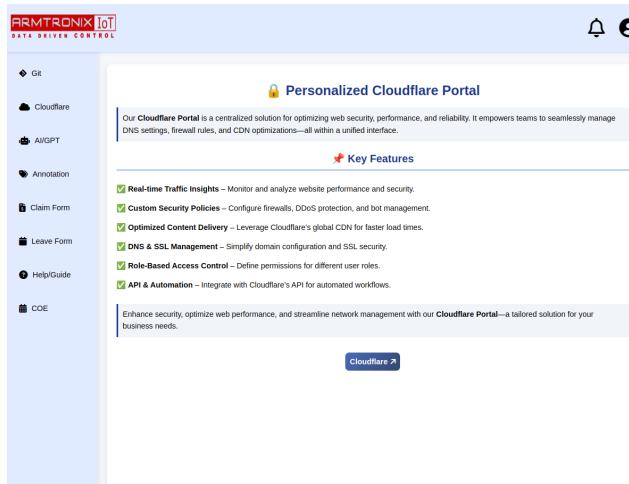


Figure 6.11: Cloudflare portal

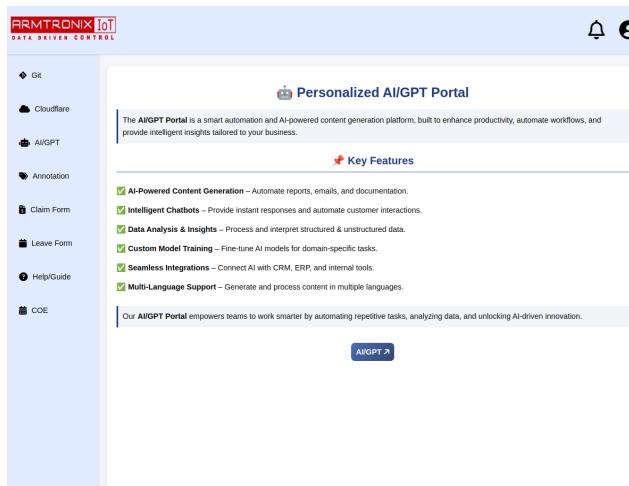


Figure 6.12: AI/GPT portal

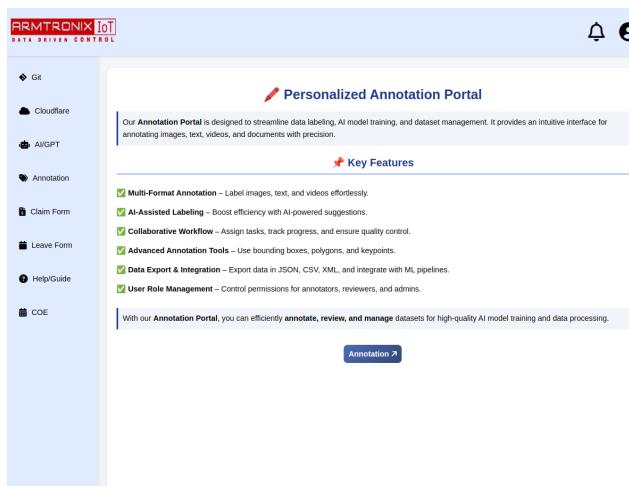


Figure 6.13: Annotation portal

Cloudflare Portal

This portal provides real-time insights into web traffic and security status. Admins and authorized employees can configure DNS settings, SSL certificates, and custom security policies. It also offers DDoS protection and advanced traffic analytics.

AI/GPT Portal

The AI/GPT Portal integrates AI-driven functionalities for document and report generation, chatbot communication, and smart automation tasks. Users can leverage AI models for predictive analytics and content creation within the organization.

Annotation portal

This interface allows employees to perform multi-format annotations on datasets for AI training. Supported formats include text, image, and video annotation. The portal facilitates AI-assisted labeling and provides options to export the annotations in JSON, CSV, and XML formats.

The screenshot shows the 'Expense Claim Form' page within the Armtronix Management System. On the left, there is a sidebar with various icons and links: Git, Cloudflare, AI/GPT, Annotation, Claim Form, Leave Form, Help/Guide, and COE. The main form area has the title 'Expense Claim Form'. It includes fields for 'Claim #' (CF/25-26/06), 'Date*' (14/05/2025), 'Employee Name*' (Armtronix), 'Employee ID' (4), and 'Location' (Hubli). Below these are ten rows for expense items, each with columns for 'Sl. No.', 'Purpose', 'Qty', 'Unit Price', and 'Amount'. At the bottom of the item table, there are three input fields: 'Advance Received (INR)' (0), 'Adjustments with Advance (INR)' (0), and 'Cash to be Returned to Office (INR)' (0). A summary section displays 'Total Expense (INR): 0' and 'Total to be Received (INR): 0.00'. Below this, there are sections for 'Submitted By' (Armtronix) and 'Approved By' (Naren Nayak), each with a signature placeholder. There is also a 'Receiver Signature' placeholder. At the bottom right are 'Submit' and 'Reset' buttons. A file upload section titled 'Upload Bills' with a 'Choose files' button is also present.

Figure 6.14: Claim Form

Claim Form

The Claim Form interface allows employees to submit reimbursement claims for official expenses. Each claim is automatically assigned a unique claim number and can be downloaded as a PDF. UTR verification is also available for tracking payment status securely.

The screenshot shows the 'Leave Form' page. At the top, it displays 'Remaining Leaves: 39'. Below that, there are input fields for 'Employee Name' (Armtronix) and 'Email' (armtronix@gmail.com). Under 'Designation', it says 'Senior Manager' and 'Type of Leave' with a dropdown menu set to 'Select Leave Type'. There are date pickers for 'Start Date' and 'End Date', both labeled 'dd/mm/yyyy'. A text input field for 'Total Leave Days' is present. Below these, a large text area for 'Reason for Leave' is followed by a 'Submit' button. At the bottom of the form, a note says 'Track your leave request status in the [My Leave Status](#) page.'

Figure 6.15: Leave Form

The screenshot shows the 'My Leave Requests' page. It lists a single 'Sick Leave' entry. The details are: Submitted: 17/04/2025, Start Date: 17/04/2025, End Date: 17/04/2025, Total Leave Days: 1, and Reason: ill health. To the right of the details, the word 'APPROVED' is written in green capital letters. On the far left, a sidebar lists various icons corresponding to different management functions.

Figure 6.16: Leave Status

The screenshot shows the 'Calendar of Events (COE)' page. It features a calendar for May 2025 with days from Sunday to Saturday. Specific dates like 12, 14, 23, 25, 26, 27, 28, 29, 30, and 31 are highlighted in various colors (orange, green, blue). To the right of the calendar, a sidebar titled 'Events & Holidays in May 2025' lists the following events with their respective dates and descriptions:

- 09/05/2025: pleadies-holiday
- 10/05/2025: pleadies-holiday
- 23/05/2025: webinar
- 12/05/2025: workshop

On the far left, a sidebar lists various icons corresponding to different management functions.

Figure 6.17: Calendar of Events(COE)

Leave Form

The Leave Form allows employees to request leave with real-time tracking. Upon submission, the request is routed to the admin for approval or rejection. Employees can monitor the status updates directly from their dashboard.

Leave Status

This interface provides an overview of all leave requests made by the employee. It displays the status (Pending, Approved, or Rejected) and allows employees to track the progress of their applications.

Calendar of Events(COE)

The COE interface displays company events, public holidays, and scheduled meetings. Admins have the capability to update events, which are reflected in real-time across employee dashboards.

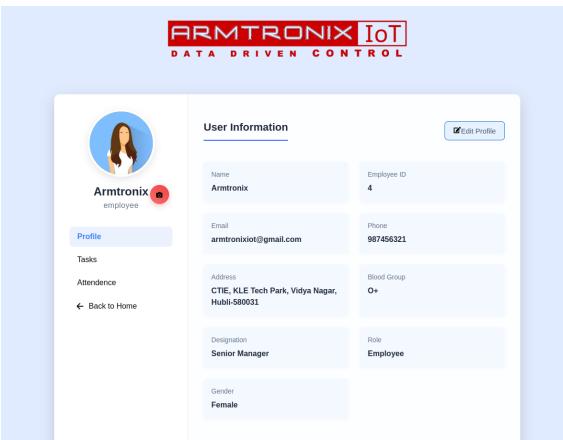


Figure 6.18: Employee Profile

Employee Profile

Employees can view and update their personal information, including contact details, address, and emergency contact. The profile management is secured with role-based access to prevent unauthorized modifications.

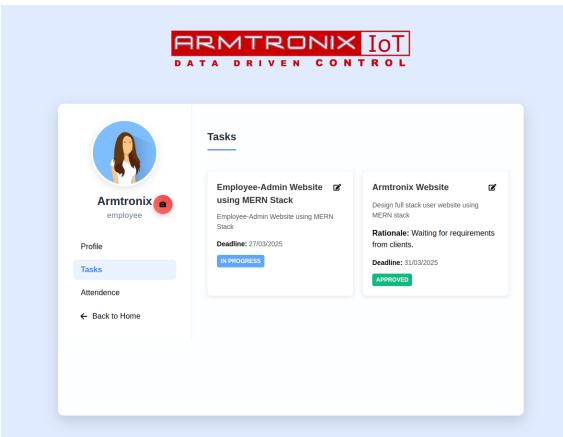


Figure 6.19: Tasks Assigned

Tasks Assigned

This page lists all tasks assigned to the employee by the admin. It includes task descriptions, deadlines, and status updates, allowing employees to keep track of their responsibilities efficiently.

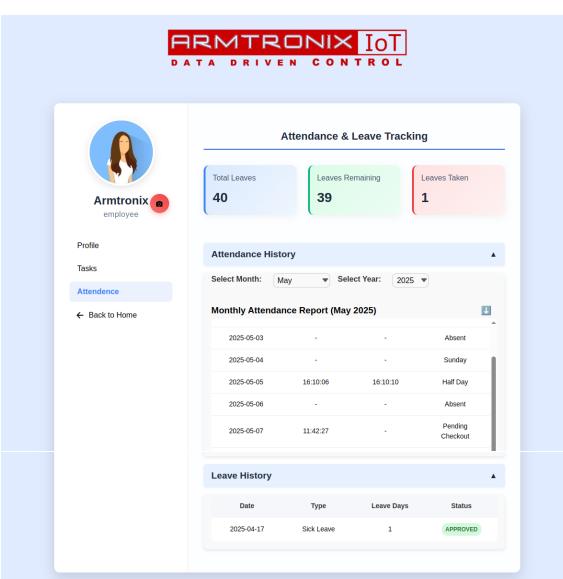


Figure 6.20: Attendance and Leave Tracking

Attendance and Leave Tracking

This section displays the attendance history and leave records of the employee. The system automatically logs check-in and check-out times, ensuring accurate attendance tracking and compliance with company policies.

6.3 Admin Portal

Admin portal contains all the feature available in the employee portal and additionally have the following features:

Expense Claim Management

Expense Claim Form

Sl. No.	Purpose	Qty	Unit Price	Amount
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Advance Received (INR):

Adjustments with Advance (INR):

Cash to be Returned to Office (INR):

Total Expense (INR): 0

Total to be Received (INR): 0.00

Submitted By:

Approved By:

Receiver Signature:

Approver Signature:

Upload Bills

Choose files | No file chosen

Buttons: Submit, Reset

Figure 6.21: Admin ClaimForm

Claim Form

This page allows the admin to view all claims submitted by employees. Admins can review, approve, or reject claims directly from this interface, and update the status for employee visibility.

The screenshot shows the 'Expense Claim Management' interface. On the left is a sidebar with icons for Git, Cloudflare, AllGPT, Annotation, Claim Form, Leave Form, RegisterEmp, Emp Details, Set Leave, HelpGuide, and COE. The main area has tabs for 'Claim Form', 'Payment Details' (which is selected), and 'Reports'. Below the tabs is a section titled 'Submitted Claims' with a table:

Claim #	Employee Name	Total Expense	Date	Payment Status	Action	Bills	Delete
CF05-2605	Armtronix	8000.00	2025-04-07	Paid	<input checked="" type="checkbox"/>	No bills	
CF05-2904	Armtronix	800.00	2025-04-07	Paid	<input checked="" type="checkbox"/>		
CF04-2510	Samarth	4999.00	2025-03-21	Paid	<input checked="" type="checkbox"/>	No bills	
CF02-2302	Samarth	1500.00	2025-03-27	Paid	<input checked="" type="checkbox"/>	No bills	

At the bottom is a pagination bar: 'Page 1 of 1' with arrows.

Figure 6.22: Payment Details

The screenshot shows the 'Expense Claim Management' interface. The sidebar and tabs are identical to Figure 6.22. The main area has a tab for 'Reports' which is selected. Below the tabs is a section titled 'Claims and Payments Report' with a button 'Previous Year Report' and a table:

Current Financial Year Claim Details								
Claim#	Date	Employee	ID	Total Expense	Payment Status	Payment Type	Payment Date	Bills
CF05-2604	2025-04-07	Armtronix	1	800.00	Paid	cash	08/05/2025	
CF05-2605	2025-04-07	Armtronix		8000.00	Paid	cash	08/05/2025	No bills

Figure 6.23: Reports

Payment Details

Displays detailed payment information for approved claims. Admins can verify payment status, check UTR (Unique Transaction Reference) numbers, and confirm successful transactions.

Reports

The Reports interface allows admins to generate analytical reports on employee activities, attendance, claim processing, and leave statuses. These reports help in data-driven decision-making.

The screenshot shows the 'Register' page. It features a logo for 'ARMTRONIX IoT DATA DRIVEN CONTROL'. The form fields include:

- Role: Employee (dropdown)
- Name: Enter your name (text input)
- Employee ID: Enter Employee ID (text input)
- Email: Enter your email (text input)
- Password: Enter Password (password input)
- Confirm Password: Confirm Password (password input)

A 'Register' button is at the bottom.

Figure 6.24: Register page

Register page

Admins can use the Register Page to onboard new employees, assign roles, and create user profiles. This page is essential for maintaining user records securely and efficiently.

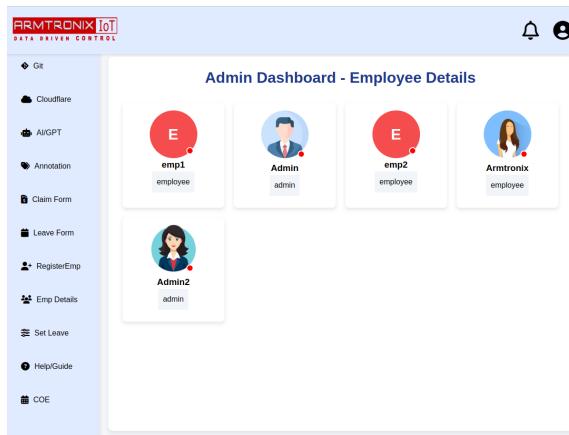


Figure 6.25: User Info

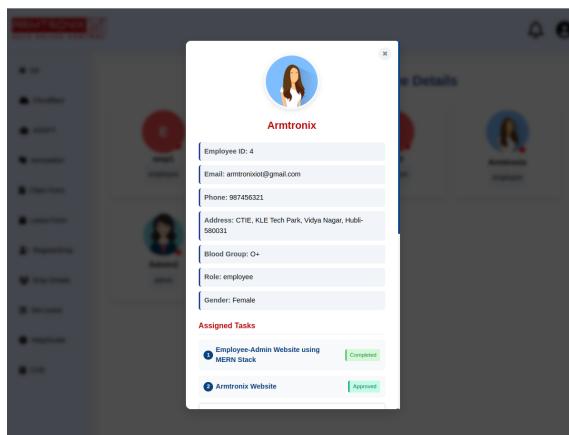


Figure 6.26: Employee Details

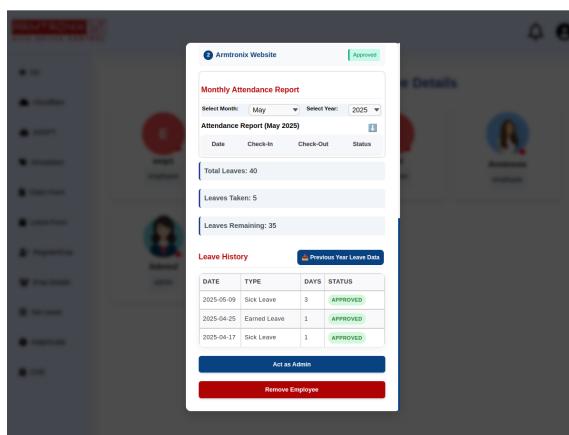


Figure 6.27: Employee Details

User Info

This interface provides detailed information about all users in the system. Admins can view user profiles, edit permissions, and manage account details with ease.

Employee Details

Displays all registered employees along with their roles, tasks assigned, and contact information. Admins can update or remove employees as necessary.

Employee Details

This interface provides a more detailed view of each employee's profile, including their attendance records, leave status, and task completion metrics.

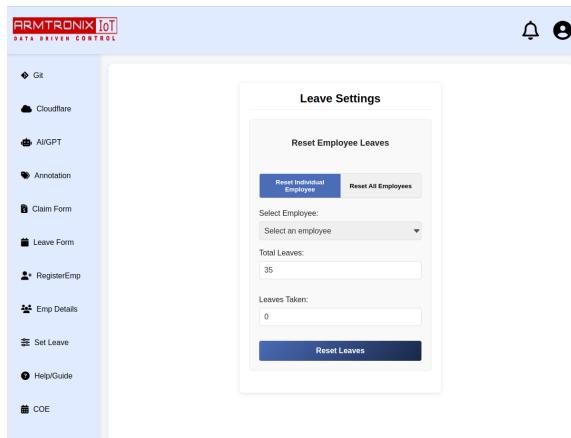


Figure 6.28: Employee Leave settings

Employee Leave settings

Allows admins to configure leave policies for employees, set limits, and adjust leave types. This helps maintain consistency and compliance with company policies.

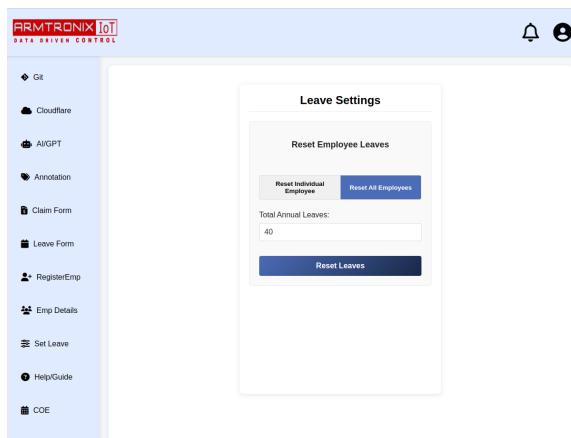


Figure 6.29: Leave settings

Leave settings

Admins can modify general leave settings for the entire organization, including holiday schedules and carry-forward policies.

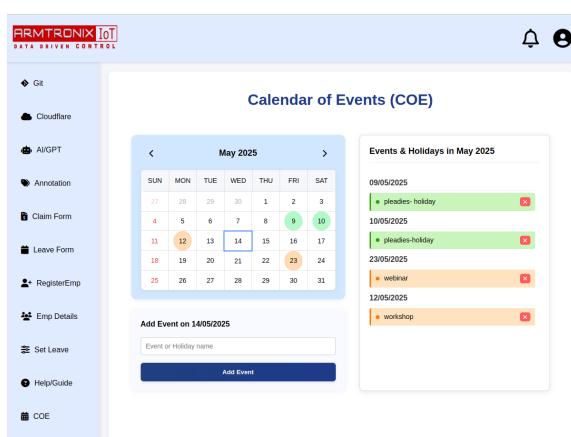


Figure 6.30: Calendar of Events(COE)

Calendar of Events(COE)

Admins can update the Calendar of Events to reflect company events, public holidays, and special announcements. Changes are synchronized across all employee dashboards in real-time.

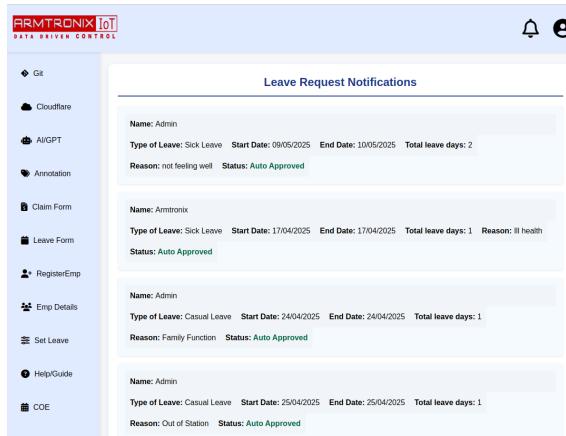


Figure 6.31: Leave Notifications

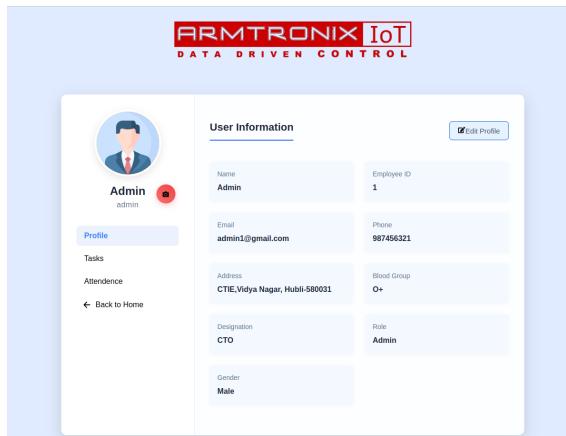


Figure 6.32: Admin Profile

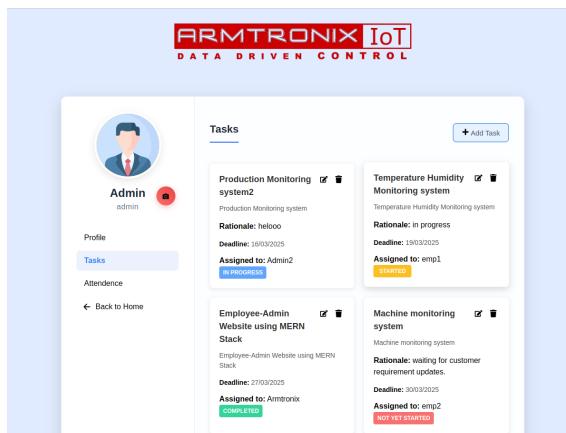


Figure 6.33: Tasks Assignment

Leave Notifications

Displays real-time updates regarding leave status, approvals, and rejections. Employees are notified instantly when their leave requests are processed.

Admin Profile

Admins can update their personal and contact information securely. Role-based access ensures that only authorized personnel can make critical changes.

Tasks Assignment

Admins can assign tasks to employees, set deadlines, and monitor completion status from this interface.

Tasks

Add New Task

Project Title:

Description:

Deadline: dd/mm/yyyy

Assign To:

- emp1 (emp1@gmail.com)
- emp2 (emp2@gmail.com)
- Armtronix (armtronixiot@gmail.com)
- Admin2 (admin2@gmail.com)

Status: Not Yet Started

Rationale: Provide reasoning or context for this task...

Create Task

Figure 6.34: Add Tasks

ARMTRONIX IoT
DATA DRIVEN CONTROL

Attendance & Leave Tracking

Total Leaves: 40 | Leaves Remaining: 36 | Leaves Taken: 4

Attendance History

Date	Check-In	Check-Out	Status
2025-05-06	10:38:53	-	Pending Checkout
2025-05-07	10:23:56	-	Pending Checkout
2025-05-08	12:57:30	-	Pending Checkout
2025-05-09	-	-	Holiday
2025-05-10	-	-	Holiday
2025-05-11	-	-	Sunday
2025-05-12	10:29:22	-	Pending

Leave History

Date	Type	Leave Days	Status
2025-05-09	Sick Leave	2	APPROVED
2025-04-25	Casual Leave	1	APPROVED
2025-04-24	Casual Leave	1	APPROVED

Figure 6.35: Attendance and Leave Tracking

Add Tasks

A structured form for admins to add new tasks to the system, complete with descriptions, deadlines, and employee assignments.

Attendance and Leave Tracking

This module allows admins to monitor and update attendance logs, check-in/check-out times, and leave records. It provides a holistic view of employee attendance for efficient management.

Chapter 7

CONCLUSION AND FUTURE SCOPE OF THE WORK

7.1 Conclusion

The Full Stack Web Development of the Armtronix Management System using the MERN stack successfully established a robust, scalable, and user-friendly platform. The integration of the User Portal, Employee Portal, and Admin Portal facilitated streamlined operations, enhanced data accessibility, and real-time communication across the organization.

7.2 Future Scope:

- Advanced Analytics Dashboards: Implementing AI-driven analytics for predictive maintenance and business insights.
- Mobile Application Support: Developing Android and iOS applications for seamless mobile access.
- Chatbot Integration: Enabling AI-powered chat support for users and employees for quick query resolutions.
- Role-Based Custom Dashboards: Customizable dashboards for different employee roles based on access levels.
- Integration with IoT Devices: Synchronizing the platform with IoT-based monitoring systems for real-time data analytics.
- Enhanced Reporting Mechanisms: Generating dynamic and customizable reports for deeper business insights.

REFERENCES

- [1] C. Sravani, P. Kumar, S. Priya, S. K. Yadav, M. J. Rao, and U. D. Prasan. Constructing a study buddy using MERN (MongoDB, Express.js, React, Node.js) stack technologies. *Engineering Proceedings*, 66(1):27, July 2024.
- [2] V. Sahni, A. Chopde, M. Goswami, and A. Kumar. MERN stack web-based themefied education platform for placement preparation. *Educational Administration: Theory and Practice*, 30(5):1918–1928, 2024.
- [3] A. Kulkarni, H. Jain, and V. Sharma. Navigating the e-learning platform with MongoDB, Express, React, and Node. *International Journal of Innovative Science and Research Technology*, 9(3), March 2024.
- [4] H. Prakash and M. Pathak. Research paper on social media application using MERN stack. *International Journal for Research in Applied Science and Engineering Technology*, 12(4):61983, April 2024.
- [5] V. Subramanian. *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node*. Apress, Berkeley, CA, 1 edition, 2017.
- [6] P. Porter, S. Yang, and X. Xi. The design and implementation of a RESTful IoT service using the MERN stack. In *Proceedings of the 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops*, pages 140–145, Monterey, CA, USA, November 2019.
- [7] D. Vasanthi, T. Sivasakthi, V. Abarna, and R. Arthi. Design and development of car rental website using MERN stack. In *Proceedings of the 2021 International Conference on Computing, Communication and Green Engineering*, pages 1–5, Pune, India, September 2021.
- [8] P. Hemalatha, M. Dhavarshini, and N. Dhivya. Development of flexible, sharing and leasing a car using MERN stack. In *Proceedings of the 2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering*, pages 1–5, Bangalore, India, December 2022.
- [9] C. Vinothini, C. Tarun, K. A. Bhavana, G. N. Krishna, and A. Praveen. NBA web portal: A comprehensive survey on NBA accreditation and MERN stack for the purpose of implementing a portal. In *Proceedings of the 2022 International Conference on Inventive Computation Technologies*, pages 1245–1251, Lalitpur, Nepal, July 2022.

- [10] P. Kanchanamala, B. H. Sai, B. Balaji, A. Panigrahi, K. Srinivas, and A. V. Vardhan. Automated programming evaluation using MERN. In *Proceedings of the 2023 2nd International Conference on Applied Artificial Intelligence and Computing*, pages 706–712, Salem, India, May 2023.
- [11] A. S. Chauhan, S. Bhardwaj, R. Shaikh, A. Mishra, and S. Nandgave. Food ordering website cooked with care developed using MERN stack. In *Proceedings of the 2022 6th International Conference on Intelligent Computing and Control Systems*, pages 1690–1695, Madurai, India, May 2022.