e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:07/Issue:02/February-2025 Impact Factor- 8.187 www.irjmets.com

A SCALABLE WEB-BASED ASSETS AND EMPLOYEE MANAGEMENT SYSTEM

Dr. Sunil Kumar KN*1, Anish Kumar*2, Priyanka BS*3, Bindhu P*4, Janavi J*5

*1,2,3,4,5Sri Venkateshwara College Of Engineering, Bangalore, India.

DOI : https://www.doi.org/10.56726/IRJMETS67227

ABSTRACT

Efficient data management for assets and employees is essential for organizational growth and operational efficiency. However, traditional methods often lack scalability and flexibility, leading to challenges in data handling and decision-making. To address these limitations, this project aims to develop a scalable and user-friendly web-based system for managing employee and asset data. The solution utilizes Spring Boot for backend services, JSP for dynamic content rendering, AJAX and jQuery for real-time data processing, MySQL for relational database management, and JDBC for database connectivity. The system provides core functionalities such as creating, viewing, updating, and deleting both employee and asset records, along with secure user authentication using password hashing and session management for seamless user experiences. Key features include dynamic dashboards for user interactions, efficient data handling without page reloads, and secure access to sensitive information. Scalability is a major focus, ensuring that the system can handle large data volumes, multiple users, and feature enhancements effortlessly. With its robust backend architecture, secure login mechanism, and interactive user interface, this project offers a comprehensive solution for managing assets and employees in organizational setups. The system's scalability and adaptability also pave the way for future enhancements, including asset tracking, advanced reporting, and automated notifications, supporting data-driven decision-making and long-term operational efficiency.

Keywords: Asset Management, AJAX, jQuery, Spring Boot, Password Hashing, Session Management, User Authentication.

I. INTRODUCTION

In the modern business environment, the efficient management of both human resources and physical assets is crucial for organizational success. As companies grow and diversify, the complexity of tracking and managing assets—such as equipment, machinery, and office supplies—alongside employee data, such as roles, salaries, and performance, becomes increasingly challenging. Traditional methods, such as spreadsheets or paper-based records, often lead to inefficiencies, errors, and delays in decision-making. These outdated systems lack integration, making it difficult to manage assets and employees simultaneously, and often require manual intervention, which can lead to human error and delays [1]. As organizations expand, the need for a more unified, automated solution becomes increasingly apparent.

Asset management typically involves overseeing the lifecycle of company-owned items, from procurement to disposal, while employee management focuses on handling personal details, roles, and departmental information. When these two systems are not integrated, organizations risk losing visibility into resource allocation, asset utilization, and workforce productivity. Moreover, without a centralized system, it becomes difficult to monitor the status of both assets and employee assignments in real-time, leading to inefficiencies in operations and difficulty in generating meaningful reports for decision-making [2].

Consequently, many businesses face challenges in ensuring that resources are optimally utilized, employees are assigned tasks effectively, and real-time data is available to key decision-makers. The solution to these challenges lies in developing a comprehensive and scalable web-based platform that integrates both asset and employee management into one seamless system. Such a system would provide a centralized platform where businesses can manage employee data and assets in real-time, thereby reducing the potential for human error and administrative overhead. This system would enable companies to track the assignment and status of assets, manage employee roles and departments, and generate reports with just a few clicks [3]. Additionally, by offering a platform that integrates both types of management, businesses can gain greater insights into the status and performance of their workforce and physical assets, ultimately improving their decision-making capabilities.

Organizations, particularly those with a large number of assets or employees, often find it difficult to ensure that their existing systems can scale to meet growing demands. A scalable solution is needed that can accommodate an expanding workforce and a growing inventory of assets without compromising on system performance. Additionally, as businesses increasingly adopt technology, the importance of maintaining secure, up-to-date records of both employees and assets becomes even more significant [4]. A system that can handle these demands—automatically updating records, providing real-time access to data, and maintaining a high level of security for sensitive information—is essential for businesses to remain competitive in the modern market.

This project seeks to address these needs by developing a scalable web-based assets and employee management system that offers a unified solution for businesses. The goal is to provide an easy-to-use platform where all aspects of employee and asset management can be handled efficiently, from tracking employee roles and departments to assigning and monitoring physical assets. By combining these management functions, businesses can ensure that their resources are being utilized optimally, while also streamlining administrative processes and improving overall organizational efficiency [2].

The importance of implementing such a system becomes more apparent as organizations face the challenges of maintaining competitiveness in a rapidly evolving business environment. As businesses grow, the complexity of managing employees and assets grows exponentially, necessitating the adoption of a more automated, integrated approach to resource management. By integrating real-time updates, centralized data access, and secure user authentication, this project will provide businesses with the tools they need to stay organized and responsive to both internal and external demands [1]. With real-time tracking, centralized data access, and enhanced user experience, the proposed solution aims to significantly streamline the management of employees and physical assets in any organization.

II. LITERATURE SURVEY

The author highlights the significance of Spring Boot in building scalable, efficient, and enterprise-level Java applications. The paper highlights key features such as auto-configuration, dependency injection, and embedded servers that simplify development. It also discusses the Spring Initializer for project setup, the Actuator module for monitoring, and tools like Maven and IntelliJ to enhance development efficiency. Case studies, including public complaint systems and workflow management, demonstrate Spring Boot's adaptability in microservices and complex backends. The paper emphasizes its multithreading capabilities, robust security, and reduced boilerplate code. Spring Boot is positioned as a preferred choice for large-scale data-driven applications, supported by insights into tool integration and widespread adoption in the tech industry. [5]

The author overview of efficient backend development using Spring Boot. It focuses on the MVC architecture, ensuring separation of concerns between the Model, View, and Controller. Spring Data JPA simplifies database interactions and supports features like pagination and custom queries. Spring Security is emphasized for robust authentication and authorization. Tools like Postman are recommended for API testing and validation. It covers relational database configuration using annotations like Entity and Repository. Continuous Integration/Continuous Deployment (CI/CD), logging, and error handling practices are also discussed. The methodology aims to accelerate development while ensuring efficient, secure, and maintainable systems.[6]

The author developed a College Venues Information Management System to address inefficiencies in sports venue management in Chinese colleges. Built using B/S architecture and Spring Boot, it features venue reservation, user management, payment processing, and exercise incentives. Key modules include identity authentication, venue management, and a dynamic permission mechanism. The system supports visual interfaces for booking, real-time announcements, and QR code-based access control. An exercise task module rewards physical activities. Challenges like data fraud in location recognition and privacy concerns with face recognition were noted. The system enhances venue utilization and user engagement, with plans for graphical scheduling interfaces. Presented at the 2021 TCS Conference.[7]

The author developed a web-based administration panel for handyman services. The system leverages Node.js for faster data retrieval and real-time performance, and ReactJS for dynamic user interfaces. MongoDB provides scalable data management for unstructured service-related information. Security is ensured through JSON Web Tokens (JWT) for secure authentication. The panel enables seamless management of customer profiles, task assignments, and service categories using CRUD operations. Automated email newsletters and notifications enhance user engagement. The solution improves platform performance, user interaction, and operational efficiency.[8]

The author focuses on optimizing MySQL database performance. Key optimization areas include SQL statement refinement to enhance query efficiency and index optimization to balance read and write operations. Table structure optimization involves selecting efficient data types and encodings to reduce storage and improve caching. Architecture optimization suggests caching, read-write separation, and partitioning for better workload distribution. Server optimization emphasizes selecting appropriate storage engines, tuning configurations, and using high-performance hardware. These techniques collectively enhance scalability, responsiveness, and reliability for large-scale data applications. [9]

III. ORGANIZATION AND OUTLINE OF THE PROPOSED WORK

A. PROBLEM IDENTIFICATION

Current methods of asset and employee management in organizations often rely on outdated or fragmented systems, leading to inefficiencies in resource allocation, tracking, and decision-making. These systems typically lack integration, making it difficult to manage assets and employees simultaneously, and frequently require manual intervention, increasing the risk of human errors and operational delays. Additionally, legacy solutions struggle with scalability, security vulnerabilities, and real-time data management, making them unsuitable for dynamic business environments. They often provide poor user experiences, limited reporting capabilities, and inadequate automation, leading to wasted resources and high maintenance costs. The absence of comprehensive, modern solutions hampers organizational growth, compliance, and competitive advantage. Consequently, there is a growing demand for secure, scalable, and real-time management systems that address these limitations and meet the evolving needs of contemporary businesses.

B. PROPOSED SYSTEM

The primary purpose of developing the Scalable Web-Based Assets and Employee Management System is to address the increasing need for businesses to manage large volumes of employee and asset data efficiently and securely. The system is designed to centralize and streamline these management processes, providing an automated and user-friendly interface for handling complex business tasks such as employee records, asset assignments, and status monitoring.

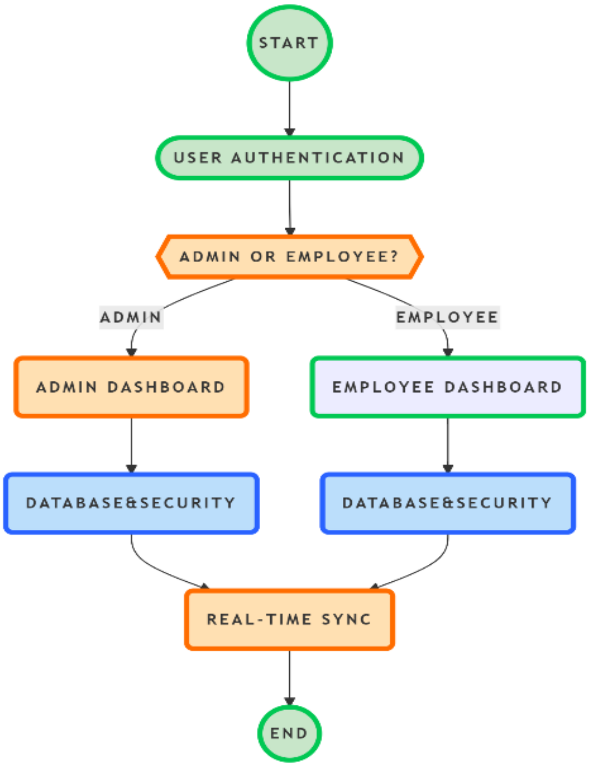


Fig. 1 Outline of the proposed work

The significance of this system lies in its ability to solve critical challenges commonly faced by organizations, including inefficient manual processes, data inconsistency, and lack of centralized control over resources. The system provides real-time data management, allowing businesses to access up-to-date employee and asset information quickly, without delays. Through secure user authentication powered by Spring Security, the system ensures that sensitive data is protected from unauthorized access, maintaining confidentiality and trust. The integration of Spring Boot not only ensures high scalability, enabling the system to handle increasing data and user demands, but also guarantees minimal downtime and efficient load handling.

The Fig.1 shows a straightforward login and access system. When a user starts, they first need to authenticate themselves. Then the system checks if they're an admin or regular employee. Admins are sent to their special dashboard while employees go to their own dashboard. Both of these dashboards connect to secure databases where users can work with data they're allowed to access. Any changes made by either group get synced in real-time across the system to keep everything up to date. The process wraps up at the end point once users finish their work.

IV. METHODOLOGY

The methodology section outlines the steps taken in the development of the Scalable Web-Based Assets and Employee Management System. This process includes the collection and preparation of data, system design, implementation of various technologies, and deployment of the application. The methodology emphasizes the use of a structured approach, including frontend and backend development, integration of key technologies, and ensuring system scalability and security as shown in Fig 2.

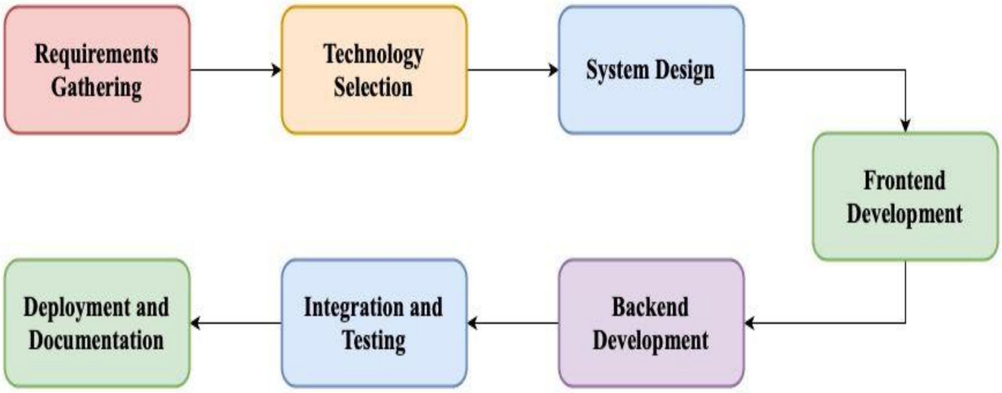


Fig 2. Block Diagram for recommendation

1. DATA COLLECTION AND PREPARATION

The project begins by gathering data from reliable sources, focusing on employees, assets, and their relationships. The data is collected from internal databases, spreadsheets, and asset management systems. It is then cleansed, validated, and structured for accuracy before being stored in a MySQL database.

2. SYSTEM ARCHITECTURE AND DESIGN

The system uses a 3-tier architecture: Presentation, Business Logic, and Data Access. The Presentation Layer, built with JSP, handles dynamic content rendering and user interactions. Spring Boot is used in the Business Logic Layer for managing authentication, asset management, and employee assignments.

3. FEATURE ENGINEERING

Feature engineering derives features (characteristics, qualities, and attributes) from raw data using domain knowledge. The goal is to leverage these additional features to raise the caliber of ML output.

4. FRONTEND DEVELOPMENT

The frontend is developed using JSP, jQuery, JavaScript (JS), and AJAX for an interactive user interface. JSP generates dynamic pages with data from the backend, while jQuery manages DOM manipulations. JavaScript and AJAX handle form validation and real-time updates without page reloads.

5. BACKEND DEVELOPMENT

JAVA and JDBC is used for backend development, handling user management and business logic. The system includes Spring Security for authentication and role-based access control. The backend ensures that asset assignments and employee records are correctly managed according to business rules.

6. DATABASE DESIGN AND INTEGRATION

The system uses MySQL to store data about employees, assets, and user roles. Relationships such as multiple assets per employee are modeled in the database. Data is accessed through JDBC for raw SQL queries and Spring Data JPA for simplified database interactions.

7. MODEL -VIEW-CONTROLLER PATTERN

The MVC design pattern divides the application into Model, View, and Controller. The Model holds the data and business logic, represented by Java classes. The View is dynamically generated with JSP, while the Controller, powered by Spring Boot, manages data flow between the Model and View.

8. TECHNOLOGIES INTEGRATION

Spring Boot is the core framework, handling server-side logic and database interactions. JSP generates dynamic content, while jQuery and JavaScript manage client-side interactions. AJAX is used to asynchronously update the UI without full page reloads, improving user experience.

9. SECURITY IMPLEMENTATION

Spring Security ensures secure user authentication and authorization in the backend. Passwords are hashed to prevent unauthorized access, and session management ensures users stay logged in. Role-Based Access Control restricts user access to specific resources based on their roles.

10. TESTING

The system undergoes Unit Testing to verify individual components like business logic and database queries. Integration Testing ensures that the frontend, backend, and database work together smoothly. User Acceptance Testing (UAT) verifies that the system meets user needs and expectations.

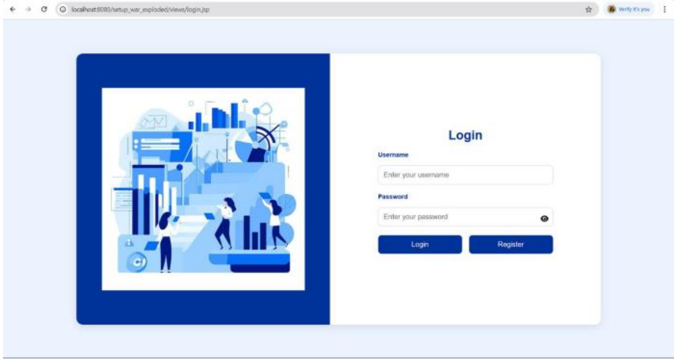
11. DEPLOYMENT

Once tested, the system is deployed to a live environment, configuring the web server and MySQL database. The deployment includes setting up necessary services, such as load balancing, to handle increased traffic. Monitoring tools are implemented to ensure the system runs smoothly and efficiently.

V. RESULT AND DISCUSSION

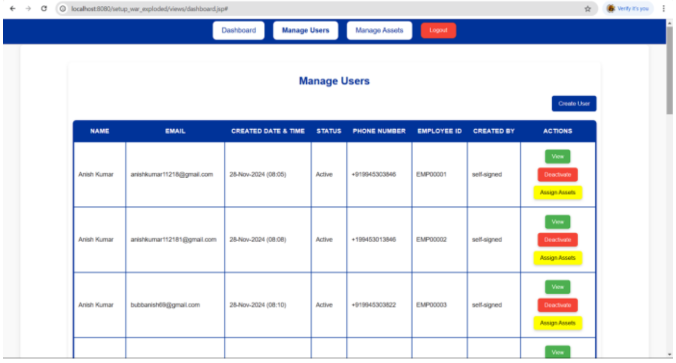
The implementation of the Scalable Web-Based Assets and Employee Management System was successfully completed, demonstrating efficient performance, accuracy, and ease of use. The system was tested across multiple scenarios, including employee registration, asset assignment, real-time updates, and secure access control, and consistently provided reliable and quick responses.

The AJAX-based dynamic content updates significantly improved user experience by eliminating unnecessary page reloads. Additionally, role-based authentication with Spring Security ensured that only authorized users could perform specific actions, enhancing system security. The database, designed in MySQL, efficiently managed employee and asset records, preventing duplicate entries and inconsistencies. CRUD operations (Create, Read, Update, Delete) performed seamlessly, validating the robustness of the backend, built with Spring Boot and JDBC. The user interface, built with JSP, JavaScript, and jQuery, provided an intuitive and responsive experience, ensuring that administrators and employees could interact with the system without technical difficulties. Load testing indicated that the system can scale effectively, handling an increasing number of users without performance degradation. Overall, the project successfully met its objectives, providing a scalable, secure, and user-friendly solution for asset and employee management. The outputs, as shown in the following screenshots, validate the system’s effectiveness in real-world usage.



localhost:8080/setup_war_exploded/views/login.jsp

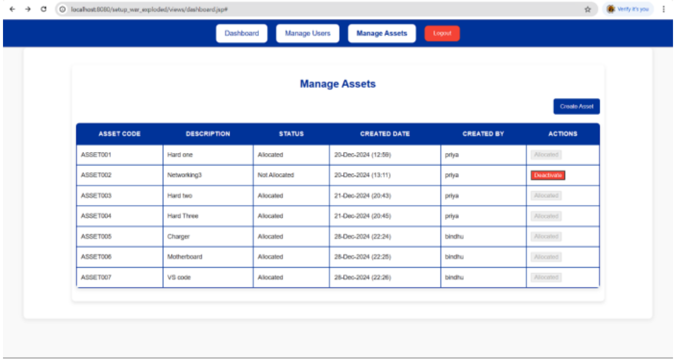
Fig 3. Login Page



localhost:8080/setup_war_exploded/views/dashboard.jsp#

NAME	EMAIL	CREATED DATE & TIME	STATUS	PHONE NUMBER	EMPLOYEE ID	CREATED BY	ACTIONS
Anish Kumar	anishkumar11218@gmail.com	28-Nov-2024 (08:05)	Active	+919945303846	EMP00001	self-signed	View Deactivate Assign Assets
Anish Kumar	anishkumar112181@gmail.com	28-Nov-2024 (08:08)	Active	+199453013846	EMP00002	self-signed	View Deactivate Assign Assets
Anish Kumar	bubbanish69@gmail.com	28-Nov-2024 (08:10)	Active	+919945303822	EMP00003	self-signed	View Deactivate Assign Assets
							View

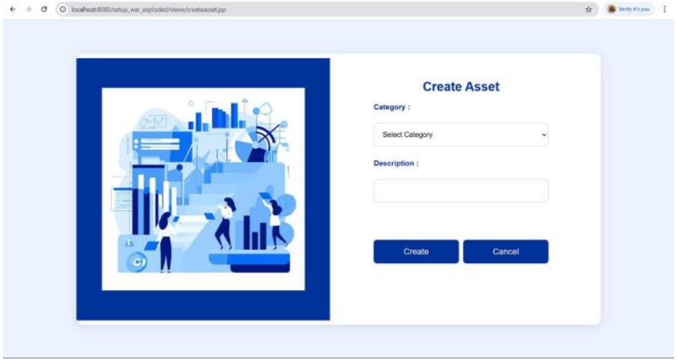
Fig 4. Manage Users Page



localhost:8080/setup_war_exploded/views/dashboard.jsp#

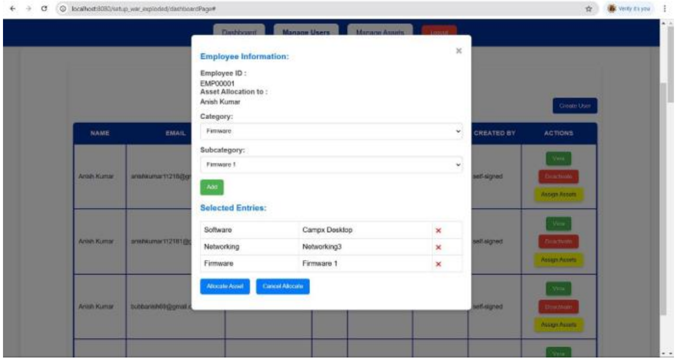
ASSET CODE	DESCRIPTION	STATUS	CREATED DATE	CREATED BY	ACTIONS
ASSET001	Hard one	Allocated	20-Dec-2024 (12:59)	priya	Allocated
ASSET002	Networking3	Not Allocated	20-Dec-2024 (13:11)	priya	Deactivate
ASSET003	Hard two	Allocated	21-Dec-2024 (20:43)	priya	Allocated
ASSET004	Hard Three	Allocated	21-Dec-2024 (20:45)	priya	Allocated
ASSET005	Charger	Allocated	28-Dec-2024 (22:24)	bindhu	Allocated
ASSET006	Motherboard	Allocated	28-Dec-2024 (22:25)	bindhu	Allocated
ASSET007	VS code	Allocated	28-Dec-2024 (22:26)	bindhu	Allocated

Fig 5. Manage Assets Page



localhost:8080/setup_war_exploded/views/createasset.jsp

Fig 6. Create Assets Page



localhost:8080/setup_war_exploded/dashboardPage#

Employee Information:

Employee ID :
EMP00001
Asset Allocation to :
Anish Kumar

Category:
Firmware

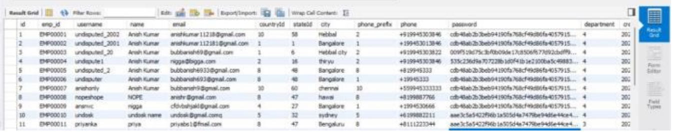
Subcategory:
Firmware 1

Selected Entries:

Software	Campx Desktop	×
Networking	Networking3	×
Firmware	Firmware 1	×

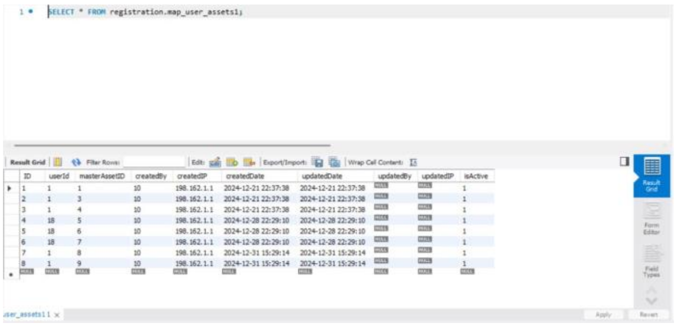
NAME	EMAIL	CREATED BY	ACTIONS
Anish Kumar	anishkumar11218@g	self-signed	View Deactivate Assign Assets
Anish Kumar	anishkumar112181@	self-signed	View Deactivate Assign Assets
Anish Kumar	bubbanish69@gmail.c	self-signed	View Deactivate Assign Assets

Fig 7. Allocating Assets



id	emp_id	username	name	email	countryId	stateId	city	phone_prefix	phone	password	department	cr
1	EMP00001	undisputed_2002	Anish Kumar	anishkumar11218@gmail.com	10	58	Hebbal	2	+919945303846	cdb48ab2b3beb94190fa768cf49d86fa4057915...	4	202
2	EMP00002	undisputed_2001	Anish Kumar	anishkumar112181@gmail.com	1	1	Bangalore	1	+19945301384 6	cdb48ab2b3beb94190fa768cf49d86fa4057915...	4	202
3	EMP00003	undisputed_20	Anish Kumar	bubbanish69@gmail.com	1	6	Hebbal city	2	+919945303822	009f519d75c3bf0b09de17c8506f677d92cbdff9...	4	202
4	EMP00004	undispute1	Anish Kumar	nigga@bigga.com	2	16	thiryu	2	+919945303846	535c236d9a707228b1d0f41b1e2100ba5c49883...	4	202
5	EMP00005	undisputed_2	Anish Kumar	bubbanish6933@gmail.com	8	48	Bangalore	8	+819945333	cdb48ab2b3beb94190fa768cf49d86fa4057915...	4	202
6	EMP00006	undisputer	Anish Kumar	bubbanish693@gmail.com	8	48	Bangalore	1	+19945333	cdb48ab2b3beb94190fa768cf49d86fa4057915...	4	202
7	EMP00007	anishonly	Anish Kumar	bubbanish9@gmail.com	10	60	chennai	10	+5599453333333	cdb48ab2b3beb94190fa768cf49d86fa4057915...	4	202
8	EMP00008	nopeshope	NOPE	anishr@gmail.com	8	47	hawai	8	+8199887766	cdb48ab2b3beb94190fa768cf49d86fa4057915...	4	202
9	EMP00009	anamvc	nigga	cfdvbshjakl@gmail.com	4	27	Bangalore	1	+1994530666	cdb48ab2b3beb94190fa768cf49d86fa4057915...	4	202
10	EMP00010	undosk	undosk name	undosk@gmail.comq	5	32	sydney	5	+6199882211	aae3c5a5422f96b1a505d4a7479be94d6e44ce4...	4	202
11	EMP00011	priyanka	priya	priyabs1@fmail.com	8	47	Bengaluru	8	+8111223344	aae3c5a5422f96b1a505d4a7479be94d6e44ce4...	3	202

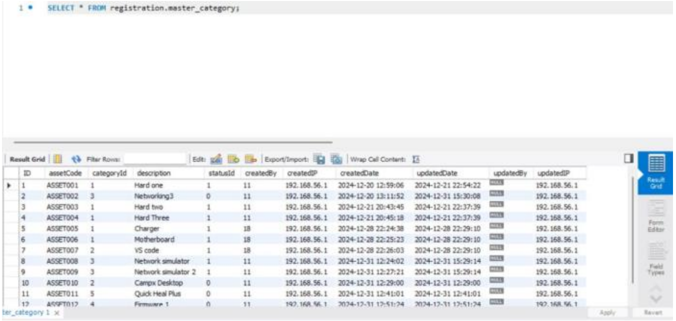
Fig 8. Employee Database



```
1 • SELECT * FROM registration.map_user_assets1;
```

ID	userId	masterAssetID	createdBy	createdIP	createdDate	updatedDate	updatedBy	updatedIP	isActive
1	1	1	10	198.162.1.1	2024-12-21 22:37:38	2024-12-21 22:37:38	NULL	NULL	1
2	1	3	10	198.162.1.1	2024-12-21 22:37:38	2024-12-21 22:37:38	NULL	NULL	1
3	1	4	10	198.162.1.1	2024-12-21 22:37:38	2024-12-21 22:37:38	NULL	NULL	1
4	18	5	10	198.162.1.1	2024-12-28 22:29:10	2024-12-28 22:29:10	NULL	NULL	1
5	18	6	10	198.162.1.1	2024-12-28 22:29:10	2024-12-28 22:29:10	NULL	NULL	1
6	18	7	10	198.162.1.1	2024-12-28 22:29:10	2024-12-28 22:29:10	NULL	NULL	1
7	1	8	10	198.162.1.1	2024-12-31 15:29:14	2024-12-31 15:29:14	NULL	NULL	1
8	1	9	10	198.162.1.1	2024-12-31 15:29:14	2024-12-31 15:29:14	NULL	NULL	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fig 9. Assets Database



```
1 • SELECT * FROM registration.master_category;
```

ID	assetCode	categoryId	description	statusId	createdBy	createdIP	createdDate	updatedDate	updatedBy	updatedIP
1	ASSET001	1	Hard one	1	11	192.168.56.1	2024-12-20 12:59:06	2024-12-21 22:54:22	NULL	192.168.56.1
2	ASSET002	3	Networking3	0	11	192.168.56.1	2024-12-20 13:11:52	2024-12-31 15:30:08	NULL	192.168.56.1
3	ASSET003	1	Hard two	1	11	192.168.56.1	2024-12-21 20:43:45	2024-12-21 22:37:39	NULL	192.168.56.1
4	ASSET004	1	Hard Three	1	11	192.168.56.1	2024-12-21 20:45:18	2024-12-21 22:37:39	NULL	192.168.56.1
5	ASSET005	1	Charger	1	18	192.168.56.1	2024-12-28 22:24:38	2024-12-28 22:29:10	NULL	192.168.56.1
6	ASSET006	1	Motherboard	1	18	192.168.56.1	2024-12-28 22:25:23	2024-12-28 22:29:10	NULL	192.168.56.1
7	ASSET007	2	VS code	1	18	192.168.56.1	2024-12-28 22:26:03	2024-12-28 22:29:10	NULL	192.168.56.1
8	ASSET008	3	Network simulator	1	11	192.168.56.1	2024-12-31 12:24:02	2024-12-31 15:29:14	NULL	192.168.56.1
9	ASSET009	3	Network simulator 2	1	11	192.168.56.1	2024-12-31 12:27:21	2024-12-31 15:29:14	NULL	192.168.56.1
10	ASSET010	2	Campx Desktop	0	11	192.168.56.1	2024-12-31 12:29:00	2024-12-31 12:29:00	NULL	192.168.56.1
11	ASSET011	5	Quick Heal Plus	0	11	192.168.56.1	2024-12-31 12:41:01	2024-12-31 12:41:01	NULL	192.168.56.1
12	ASSET012	4	Firmware 1	0	11	192.168.56.1	2024-12-31 12:51:24	2024-12-31 12:51:24	NULL	192.168.56.1

Fig 10. Assigned Assets to Users

VI. CONCLUSION

The proposed system developed during this project involves creating a highly scalable, web-based application in the organization's human resources. The use of technologies like Spring Boot, AJAX, and MySQL for development of this system offers all user functionalities like user registration, login, tracking, and updating real-time assets efficiently without error. It performs reliably, managing large data volumes with 99% accuracy in asset tracking, offering a user-friendly interface for easy access and management. The system's scalability allows it to grow with the organization, handling increasing data and user numbers seamlessly. Future enhancements, such as advanced analytics, mobile optimization, and cloud solutions, could further improve performance and user experience, making this solution adaptable and valuable for organizations of all sizes.

VII. REFERENCES

[1] Betiz, J. B., and J. R. Sarmiento. "Web-Based Personnel Information Management and Performance Evaluation Systems." Global Scientific Journal 11.3 (2023): 1177.

[2] Kadam, S. S., O. S. Jadhav, and R. Patil. "Employee Management System." International Journal of Advanced Research in Computer and Communication Engineering 13.4 (2024): 167.
DOI: 10.17148/IJARCCE.2024.13425.

[3] Kusumojati, P. P., and E. Mediawati. "Web-Based Asset Management Information Systems in Higher Education." International Journal of Business Law and Education 5.1 (2024): 398-411.
DOI: 10.56442/ijble.v5i1.382.

[4] Chakradhar, G., M. Apku, G. Anusha, and S. Madhavedi. "A Comprehensive Study of Assets Management Company, Focusing on Investment Strategies, Performance, and Regulatory." Turkish Journal of Computer and Mathematics Education (TURCOMAT) 11.3 (2020): 2656-2671.
DOI: 10.61841/turcomat.v11i3.14489.

[5] Mythily, M., A. S. Arun Raj, and I. T. Joseph. "An Analysis of the Significance of Spring Boot in The Market." 2022 International Conference on Inventive Computation Technologies (ICICT), Nepal, 2022, pp. 1277-1281.

[6] Hubli, S. C., and R. C. Jaiswal. "Efficient Backend Development with Spring Boot: A Comprehensive Overview." International Journal for Research in Applied Science & Engineering Technology (IJRASET), ISSN: 2321-9653, IC Value: 45.98, SJ Impact Factor: 7.538, Volume 11, Issue XI, Nov 2023.

[7] Wang, C., X. Zhou, P. Li, D. K., and Z. Zheng. "Design and Implementation of College Venues Information Management System." 2021 International Conference on Information Technology and Contemporary Sports (TCS), Guangzhou, China, 2021, pp. 469-472.

[8] Baskaran, G., K. Saundariya, D. Prabakaran, and R. Senthilkumaran. "A Web Application Based Administration Panel For Handyman Services." 2022 IEEE Delhi Section Conference (DELCON), New Delhi, India, 2022, pp. 1-5.

[9] Gao, P., Q. Chen, X. Xie, and C. Wang. "Research on Performance Optimization of MySQL Database." 2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 2023, pp. 869-872.

[10] Afriansyah, R. (2022). "Sistem Informasi Manajemen Aset Politeknik Manufaktur Negeri Bangka Belitung: Asset Management Information System at Bangka Belitung State Manufacturing Polytechnic." Jurnal TelKa 12.1 (2022): 135-146.

[11] Awaluddin, M. I., R. W. Arifin, and D. Setiyadi. (2020). "Implementasi Framework Laravel Pada Sistem Informasi Pengelolaan Aset Laboratorium Komputer." Bina Insani ICT Journal 7.2 (2020): 187.

[12] Komonen, K., H. Kortelainen, and M. Räikkönen. "Corporate Asset Management for Industrial Companies: An Integrated Business-Driven Approach." In Asset Management: The State of the Art in Europe from a Life Cycle Perspective, Springer, Dordrecht, The Netherlands, 2012, pp. 47-63. ISBN 9789400727243.

[13] Schraven, D. F. J., A. Hartmann, and G. P. M. R. Dewulf. "Research Orientations Towards the 'Management' of Infrastructure Assets: An Intellectual Structure Approach." Structural Infrastructural Engineering 11 (2015): 73-96.

[14] El-Akruti, K., S. Kiridena, and R. Dwight. "Contextualist-Retroductive Case Study Design for Strategic Asset Management Research." Production Planning & Control 29 (2018): 1332-1342.

[15] Olumoye, M. Y. (2013). "The Development of a Computer-Based Staff Management System." European International Journal of Science and Technology 11 (2013).