**EASTERN INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY**

**DEPARTMENT OF SOFTWARE ENGINEERING**

**A blue and brown circular logo

Description automatically generated**

**PROJECT 1 REPORT**

**STAFF MANAGEMENT**

**Students**

Văn Minh Thức – 2031200067

Thái Phan Anh Hiểu – 2231200066

**Supervisor**

Nguyễn Xuân Cường

**Binh Duong, December, 2024**

# PROJECT EVALUATION FORM (SUPERVISOR)

1. **General information**

– Project TITLE: Staff Management

– Student name and ID:

* Văn Minh Thức – 2031200067
* Thái Phan Anh Hiểu – 2231200066

– Supervisor name: Nguyễn Xuân Cường

1. **Descriptive comments** 
   1. Presentation (layout, format, style, confidence)
   2. Content (Originality and contribution)
   3. Outcome (Creativity and practice)
   4. Performance during questions and answers
2. **Suggestion for improvement (Optional)**

**Total Score: Binh Duong, Date:**

**Nguyễn Xuân Cường**

# ABSTRACT

Managing human resources effectively is a cornerstone of operational success for small and medium-sized enterprises (SMEs). However, many SMEs struggle with outdated and inefficient manual processes that hinder productivity and escalate administrative costs. **Staff Management** is a comprehensive application designed to transform how SMEs handle human resource management by introducing an organized, automated, and cost-effective solution.

The application centralizes and systematizes HR functions such as leave management, employee transfers, business trip approvals, overtime tracking, and recruitment. By implementing a role-based permissions model, it ensures transparency, accountability, and operational clarity across all hierarchical levels, from employees to directors. Built on modern technologies like React.js, Bootstrap for the frontend, Node.js for API development, and Microsoft SQL for data management, the application is both scalable and user-friendly.

Targeted at SMEs, **Staff Management** not only streamlines HR workflows but also significantly reduces the time and cost associated with traditional management methods. With a commitment to simplifying complex HR processes, **Staff Management** serves as a tool that empowers businesses to focus on growth and efficiency.

# ACKNOWLEDGEMENT

In the process of completing this project, we have received a great deal of help, guidance and encouragement from our teacher and many programmers in many places in the world.

First of all, we would like to express my deepest thanks to our supervisor, Mr. Nguyen Xuan Cuong who gave us proper suggestions on how to shape the study and always been most willing and ready to give us valuable advices, helpful comments as well as correction of our research paper.

Next, we would like to express our gratitude to all the programmers who made several videos and gave advice on the Internet who helped us much in gaining knowledge, detecting, fixing errors and completing this project.

Their assistance has been vital in helping us improve our skills and meet the project's objectives, and we sincerely appreciate it.

# TABLE OF CONTENTS

[PROJECT EVALUATION FORM (SUPERVISOR) 1](#_Toc186726214)

[ABSTRACT 2](#_Toc186726215)

[ACKNOWLEDGEMENT 3](#_Toc186726216)

[TABLE OF CONTENTS 4](#_Toc186726217)

[CHAPTER I: INTRODUCTION 6](#_Toc186726218)

[1.1 Motivation 6](#_Toc186726219)

[1.2. Problems Addressed 7](#_Toc186726220)

[1.3. Objectives 7](#_Toc186726221)

[1.4. Unique Features 8](#_Toc186726222)

[1.5. Related Work 8](#_Toc186726223)

[1.5.1. Introduction 8](#_Toc186726224)

[1.5.2. Overview of Existing Solutions 9](#_Toc186726225)

[1.5.3. Limitations of Existing Systems 9](#_Toc186726226)

[1.5.4. Comparison with Staff Management 10](#_Toc186726227)

[1.5.5. Conclusion 11](#_Toc186726228)

[CHAPTER II: TECHNOLOGY 12](#_Toc186726229)

[2.1. React.js 12](#_Toc186726230)

[2.1.1. React.js Introduction 12](#_Toc186726231)

[2.1.2. React.js Advantages [3] 12](#_Toc186726232)

[2.2. MS SQL 14](#_Toc186726233)

[2.2.1. MS SQL Introduction 14](#_Toc186726234)

[2.2.2. MS SQL Advantages [5] 14](#_Toc186726235)

[2.3. Node.js 16](#_Toc186726236)

[2.3.1. Node.js Introduction 16](#_Toc186726237)

[2.3.2. Node.js Advantages [8] [9] 16](#_Toc186726238)

[2.4. HTML 18](#_Toc186726239)

[2.4.1. HTML Introduction 18](#_Toc186726240)

[2.4.2. HTML Advantages 18](#_Toc186726241)

[2.5. CSS 19](#_Toc186726242)

[2.5.1. CSS Introduction 19](#_Toc186726243)

[2.5.2. CSS Advantages 19](#_Toc186726244)

[2.6. Express.js 20](#_Toc186726245)

[2.7. JSON Web Token (JWT) 20](#_Toc186726246)

[CHAPTER III: SYSTEM ANALYSIS 22](#_Toc186726247)

[3.1. Use Case 22](#_Toc186726248)

[3.1.1. Overview 23](#_Toc186726249)

[3.1.2. Key Actors and Permissions 23](#_Toc186726250)

[3.2. Database Diagram 25](#_Toc186726251)

[3.3. Activity Diagram 27](#_Toc186726252)

[3.3.1. Appointment and Transfer Staff 27](#_Toc186726253)

[3.3.2. Overtime Management 28](#_Toc186726254)

[3.3.3. Probationary Management 29](#_Toc186726255)

[3.3.4. Day-off Management 30](#_Toc186726256)

[3.3.5. Resignation Management 31](#_Toc186726257)

[3.3.6. Recruitment Management 32](#_Toc186726258)

[3.3.7. Business Trip Management 33](#_Toc186726259)

[3.4. Implementation 34](#_Toc186726260)

[CHAPTER IV: RESULT AND FUTURE WORK 38](#_Toc186726261)

[4.1. Achievements 38](#_Toc186726262)

[4.1.1. Technical Milestones: 38](#_Toc186726263)

[4.1.2. User Role Management: 38](#_Toc186726264)

[4.1.3. Testing and Optimization: 38](#_Toc186726265)

[4.1.4. Real-World Applicability: 38](#_Toc186726266)

[4.2. Future Work 38](#_Toc186726267)

[4.2.1. Advanced Security Measures: 38](#_Toc186726268)

[4.2.2. Additional Features: 38](#_Toc186726269)

[4.2.3. Scalability Enhancements: 39](#_Toc186726270)

[4.2.4. Localization and Multi-Language Support: 39](#_Toc186726271)

[REFERENCES 40](#_Toc186726272)

# CHAPTER I: INTRODUCTION

## 1.1 Motivation

The idea for the **Staff Management** application emerged from an in-depth survey of small and medium-sized enterprises (SMEs), revealing a pressing need for improved human resource management tools. SMEs, unlike larger corporations, often operate with limited budgets and resources, which restricts their ability to implement advanced HR systems. As a result, these businesses rely heavily on manual processes to handle critical HR functions such as leave requests, employee transfers, business trip authorizations, and recruitment. While such manual systems may suffice for smaller teams, they quickly become cumbersome and inefficient as the organization grows.

The challenges inherent in these manual processes are numerous. Managing leave requests, for instance, often involves a lengthy approval chain, with information passing through emails or spreadsheets, increasing the likelihood of delays or errors. Similarly, tracking employee transfers or recruitment without a structured system can lead to inconsistencies, miscommunications, and data loss. Such inefficiencies not only consume valuable time but also distract from the organization's primary objectives, such as growth and operational excellence.

Through these surveys, it became increasingly clear that SMEs urgently need a tailored digital solution to optimize HR management. The absence of a centralized platform for HR operations emerged as a key pain point, resulting in scattered data that is difficult to retrieve or analyze, inefficient communication between departments, and unclear role-based permissions that undermine accountability and transparency.

The fragmented nature of existing processes also limits scalability. SMEs that aspire to grow find themselves burdened by the administrative complexity of managing a larger workforce with manual methods. This can lead to missed opportunities, lower employee satisfaction, and ultimately, reduced competitiveness in the market.

Addressing these critical pain points became the driving force behind the development of **Staff Management**. The application is designed not just to replace manual processes but to transform them into streamlined, automated workflows that save time and reduce errors. By centralizing HR data and offering role-based permissions, **Staff Management** ensures that information is easily accessible and securely managed. This enables SMEs to refocus their energy on core business activities, driving productivity and growth while maintaining a robust and efficient HR framework.

## 1.2. Problems Addressed

**Staff Management** tackles several fundamental issues faced by SMEs:

* **Inefficient Manual Processes**: Manual methods of HR management consume a significant amount of time and are prone to human error.
* **Unorganized Data Management**: Without a unified system, HR data is often scattered across different platforms, making it difficult to access, analyze, or store effectively.
* **Lack of Role-Based Permissions**: The absence of a clear and structured permission system often leads to operational inefficiencies and accountability gaps.
* **Limited Scalability**: SMEs often find it challenging to scale manual HR processes as their workforce grows, which hampers business expansion.

By addressing these issues, the application provides SMEs with a modern, scalable, and efficient alternative to traditional HR management.

## 1.3. Objectives

The primary objective of **Staff Management** is to deliver a tool that makes human resource management:

* **Efficient**: Automating repetitive tasks and streamlining workflows to save time.
* **Organized**: Centralizing all HR operations in a unified platform.
* **Cost-Effective**: Reducing HR management expenses compared to traditional manual processes.

Additionally, the application aims to empower SMEs to adopt a structured approach to HR management, enabling faster decision-making and reducing administrative overhead.

## 1.4. Unique Features

**Staff Management** distinguishes itself through:

* **Streamlined Workflow Automation**: Automating processes like leave approvals, business trip authorizations, and recruitment significantly speeds up decision-making.
* **Centralized Data Management**: All employee information, including requests, approvals, and feedback, is stored in a single, easily accessible database.
* **Role-Based Access Control (RBAC)**: Ensuring only authorized individuals can access or modify specific data, improving operational transparency.
* **Targeted for SMEs(Small and Medium-sized Enterprises)**: Designed to be scalable yet affordable, catering specifically to the needs and limitations of small and medium-sized businesses.
* **Modern Technology Stack**: Leveraging React.js for a responsive user interface and Node.js for efficient backend operations ensures a seamless and robust experience.

## 1.5. Related Work

### 1.5.1. Introduction

As developers of the **Staff Management** application, we recognize the critical role Human Resource (HR) management systems play in modern organizations. These tools streamline administrative tasks, enhance employee engagement, and boost overall efficiency. However, during our research, we observed that small and medium-sized enterprises (SMEs) often struggle to adopt these systems due to their cost and complexity. This realization motivated us to create a solution specifically tailored to address the needs of SMEs(Small and Medium-sized Enterprises).

### 1.5.2. Overview of Existing Solutions

During our research, we studied several well-known HR management systems to understand their strengths and limitations:

* + **BambooHR**:
    - BambooHR is designed for small to medium-sized businesses and offers features such as employee record management, leave tracking, and performance reporting.
    - While it caters to SMEs, its pricing model can be restrictive for smaller organizations, limiting its accessibility.
  + **Workday**:
    - Workday provides a comprehensive platform with advanced functionalities, including payroll management, workforce analytics, and financial planning.
    - However, we found that it is primarily geared toward large organizations, requiring significant training and investment, making it less feasible for SMEs.
  + **SAP SuccessFactors**:
    - SAP SuccessFactors is focused on enterprise-scale operations and offers robust features like talent management and succession planning.
    - From our analysis, these extensive features often exceed the needs of SMEs, leading to underutilization and inefficiency for smaller businesses.

### 1.5.3. Limitations of Existing Systems

Through our evaluation, we identified key challenges that SMEs face when adopting existing HR systems:

* + **High Costs**: Subscription fees and implementation costs are prohibitive, particularly for smaller businesses with limited budgets.
  + **Complex Interfaces**: These systems often have steep learning curves, requiring extensive training and dedicated resources.
  + **Enterprise-Focused Features**: Many functionalities are designed for large-scale operations, leaving SMEs overwhelmed and unable to utilize the full potential of the systems.

### 1.5.4. Comparison with Staff Management

In developing **Staff Management**, we addressed the limitations of existing systems by focusing on the unique needs of SMEs.

* + **Affordability**:
    - We designed our pricing model to ensure accessibility for SMEs, allowing even the smallest businesses to adopt our system without financial strain.
  + **Simplified Core Functionalities**:
    - Our application emphasizes essential features such as leave management, employee transfers, recruitment, and overtime tracking.
    - By focusing on these core functions, we provide SMEs with a streamlined and user-friendly experience.
  + **User-Friendly Interface**:
    - Using React.js and Bootstrap, we created a responsive and intuitive interface that minimizes the need for training.
    - Our design ensures that users can quickly navigate the system and perform tasks efficiently.
  + **Tailored for SMEs**:
    - Unlike enterprise-focused solutions, **Staff Management** is built specifically for SMEs. We prioritized relevance and usability to ensure our application meets the needs of smaller businesses effectively.

### 1.5.5. Conclusion

While systems like BambooHR, Workday, and SAP SuccessFactors are powerful tools for larger organizations, their limitations often exclude SMEs from fully benefiting. With **Staff Management**, we aim to fill this gap by delivering an affordable, focused, and easy-to-use solution. By addressing the unique challenges faced by SMEs, we believe our application provides an effective alternative to complex, high-cost HR systems, empowering smaller businesses to optimize their HR processes without unnecessary burdens.

# CHAPTER II: TECHNOLOGY

## 2.1. React.js

### 2.1.1. React.js Introduction

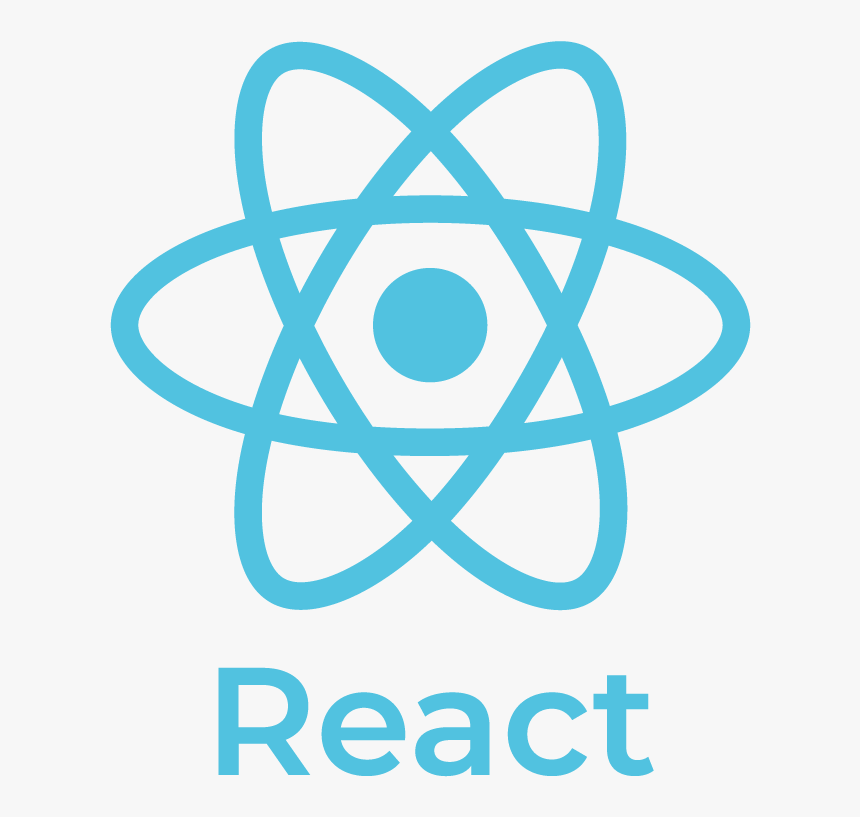


Figure 1. React Logo

React.js is an open-source JavaScript library used to build user interfaces, particularly single-page applications. It enables developers to create reusable UI components, manage the application's state efficiently, and deliver interactive web experiences. Developed and maintained by Facebook, React.js has become one of the most widely adopted libraries for modern frontend development due to its flexibility, performance, and active ecosystem [1] [2].

### **2.1.2. React.js Advantages** [3]

* **Component-Based Architecture**  
  React.js adopts a component-based structure, allowing developers to break down the UI into reusable and independent pieces. This modularity improves code maintainability and promotes reusability across the application.
* **Virtual DOM**  
  React.js uses a Virtual DOM for rendering, which minimizes direct manipulation of the real DOM. This approach improves performance by efficiently updating only the parts of the DOM that change, leading to faster and smoother applications.
* **Declarative Syntax**  
  With React.js, developers can describe what the UI should look like at any point in time using a declarative syntax. This makes code more predictable, easier to debug, and easier to maintain.
* **One-Way Data Flow**  
  React.js enforces a unidirectional data flow, which simplifies application architecture and makes it easier to trace and manage changes in the application's state.
* **JSX (JavaScript XML)**  
  React introduces JSX, a syntax extension that allows developers to write HTML-like code directly in JavaScript. This improves readability and simplifies the process of writing and visualizing components.
* **Ecosystem and Community**  
  React.js has a large and active community, providing a vast ecosystem of libraries, tools, and third-party integrations. This ecosystem includes popular tools like React Router for routing and Redux for state management, enabling developers to extend React's functionality easily.
* **SEO Optimization with SSR**  
  React.js supports Server-Side Rendering (SSR) through libraries like Next.js, enhancing SEO performance by pre-rendering content on the server and delivering faster page loads.
* **Cross-Platform Development**  
  React.js is the foundation for **React Native**, a framework for building mobile applications using the same React principles. This enables developers to share code and expertise between web and mobile platforms.
* **Community-Driven Innovation**  
  React.js is continuously evolving with contributions from Facebook and the global developer community. Its documentation is extensive, and the community is highly active in creating tutorials, tools, and support for developers at all levels.

## 2.2. MS SQL

### 2.2.1. MS SQL Introduction

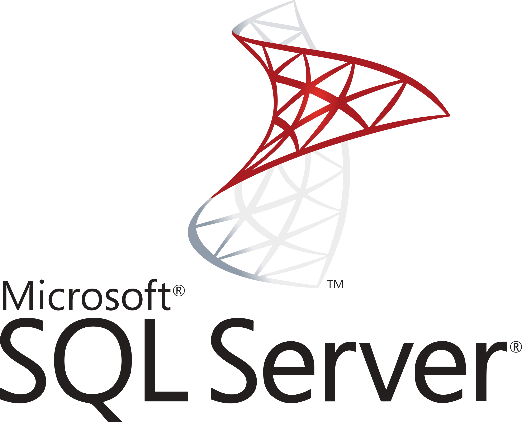


Figure 2. MSSQL Logo

Microsoft SQL Server (MS SQL) is a relational database management system (RDBMS) developed by Microsoft. It is designed to store, retrieve, and manage data in structured formats using SQL (Structured Query Language). MS SQL is widely used in enterprise environments due to its scalability, security features, and seamless integration with other Microsoft products such as Azure and .NET Framework [4].

### **2.2.2. MS SQL Advantages** [5]

* **Robust Data Management**  
  MS SQL supports advanced data management features such as transactions, indexing, and query optimization. It ensures data consistency, accuracy, and high performance for both small-scale and large-scale applications.
* **High Performance and Scalability**  
  With built-in support for indexing, partitioning, and in-memory technology, MS SQL delivers excellent performance. Its scalability enables handling massive datasets and supports high transaction loads, making it suitable for enterprises of all sizes.
* **Advanced Security Features**  
  MS SQL provides robust security through role-based access control, encryption (both at rest and in transit), and auditing tools. It ensures that sensitive data is protected against unauthorized access and complies with regulatory requirements.
* **Integration with Microsoft Ecosystem**  
  MS SQL integrates seamlessly with other Microsoft tools and platforms, such as Azure, Power BI, and .NET. This integration enhances productivity by streamlining workflows across different systems.
* **Comprehensive Toolset**  
  The SQL Server Management Studio (SSMS) offers a powerful interface for managing databases, writing queries, and monitoring performance. Additionally, tools like SQL Server Data Tools (SSDT) provide an integrated development environment for building database solutions.
* **Support for Modern Features**  
  MS SQL supports a wide range of modern database features, including:
  + **JSON and XML Processing:** For handling semi-structured data.
  + **Temporal Tables:** For managing historical data.
  + **Graph Data Modeling:** For working with complex relationships in data.
  + **Machine Learning Services:** For embedding Python and R-based models directly into the database.
* **Availability and Disaster Recovery**  
  MS SQL ensures high availability through features like Always On Availability Groups, replication, and log shipping. These features reduce downtime and provide robust disaster recovery solutions for critical applications.
* **Extensive Documentation and Community Support**  
  MS SQL benefits from comprehensive documentation provided by Microsoft and a vibrant community. Developers and administrators can access tutorials, best practices, and forums to solve problems and enhance their skills.

## 2.3. Node.js

### 2.3.1. Node.js Introduction

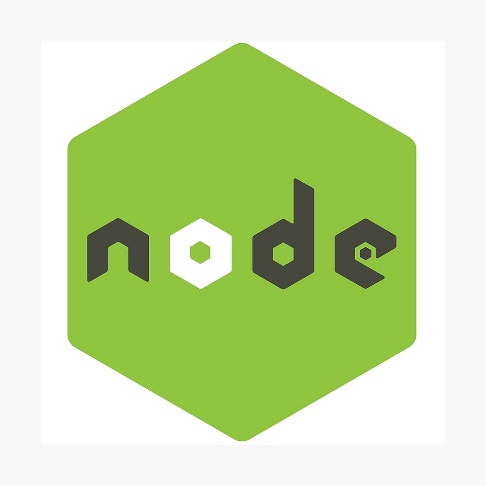


Figure 3. Node Logo

Node.js is an open-source, cross-platform runtime environment that allows developers to execute JavaScript code outside of a browser. Built on Google Chrome's V8 JavaScript engine, Node.js enables the development of fast, scalable, and efficient server-side and networking applications. Its non-blocking, event-driven architecture makes it an ideal choice for handling concurrent requests and building real-time applications like chat applications, APIs, and streaming services [6] [7].

### **2.3.2. Node.js Advantages** [8] [9]

* **Non-Blocking I/O**  
  Node.js uses an event-driven, non-blocking I/O model that enables it to handle multiple requests simultaneously without being resource-intensive. This makes it highly efficient for real-time applications and systems with high I/O operations.
* **Single Programming Language**  
  Node.js allows developers to use JavaScript for both frontend and backend development, streamlining the development process and reducing the need to learn multiple programming languages.
* **NPM Ecosystem**  
  Node.js includes a package manager called NPM (Node Package Manager), which is the world's largest ecosystem of open-source libraries and tools. It simplifies application development by providing prebuilt solutions for a wide range of use cases.
* **High Performance**  
  Powered by the V8 JavaScript engine, Node.js compiles JavaScript code into machine code, resulting in fast execution and high performance.
* **Scalability**  
  Node.js supports scalability through its asynchronous architecture and clustering capabilities, which enable applications to handle large numbers of connections and scale horizontally.
* **Cross-Platform Development**  
  Node.js is cross-platform, allowing developers to build applications that run on Windows, Linux, and macOS with minimal modifications. This flexibility is especially useful for teams working in diverse environments.
* **Real-Time Applications**  
  Node.js is widely used for developing real-time applications such as chat platforms, online games, and live-streaming services. Its WebSocket support facilitates bidirectional communication between the server and the client.
* **Active Community and Ecosystem**  
  Node.js has a vibrant and active community that contributes to its continuous development. Extensive documentation, tutorials, and third-party modules ensure that developers have the resources needed to build robust applications.
* **Microservices-Friendly**  
  Node.js is well-suited for building microservices architectures, allowing developers to break down large applications into smaller, manageable services that communicate efficiently.

## 2.4. HTML

### 2.4.1. HTML Introduction



Figure 4. HTML Logo

HTML (HyperText Markup Language) is the standard markup language used to structure and present content on the web. It defines elements such as headings, paragraphs, links, images, and tables, serving as the foundation for all web applications. HTML ensures that browsers can render content accurately and consistently across devices.

### 2.4.2. HTML Advantages

* **Content Structuring**  
  HTML helps organize and define the structure of web documents, creating an interface that is user-friendly and easily understood by search engines.
* **Ease of Learning and Usage**  
  With its simple and clear syntax, HTML is accessible even for beginners in programming.
* **Extensibility**  
  HTML integrates seamlessly with languages like CSS and JavaScript to enhance website features and styles.
* **SEO Support**  
  A clean and standard HTML structure improves search engine visibility and enhances SEO performance for websites.
* **Cross-Platform Compatibility**  
  HTML works on all modern browsers and operating systems, ensuring websites are easily accessible from any device.

## 2.5. CSS

### 2.5.1. CSS Introduction



Figure 5. CSS Logo

CSS (Cascading Style Sheets) is the language used to style and layout HTML documents. It separates content from presentation, allowing developers to customize the appearance without altering the structure of the web page. CSS supports creating visually appealing, consistent, and user-optimized interfaces

### 2.5.2. CSS Advantages

* **Separation of Style and Content**  
  CSS enables the separation of design and content, making it easier to maintain and update a website's appearance.
* **Customizable Layouts and Styles**  
  CSS offers powerful tools to style elements (e.g., colors, fonts, animations) and create complex layouts using techniques like flexbox and grid.
* **Responsive Design Support**  
  CSS facilitates responsive web design, ensuring that websites are optimized for various screen sizes and devices.
* **Improved Page Load Performance**  
  Efficient CSS usage reduces HTML file size, resulting in faster load times and overall better performance.
* **Reusability**  
  CSS classes can be reused throughout a project, saving developers time and effort.

## 2.6. Express.js

**Express.js** is a lightweight and flexible web application framework for **Node.js** that simplifies the development of robust web APIs. In the **Staff Management** project, it provides the following benefits:

* **Routing Management**: Express.js makes it easier to handle routes for multiple functionalities, such as:
  + /login for user authentication.
  + /requests for handling leave or business trip requests.
  + /approvals for managing approval workflows.
* **Middleware Integration**: You can use middleware to handle:
  + Authentication (e.g., checking if a user has valid access rights).
  + Logging and debugging.
  + Error handling.
* **RESTful API**: Express.js allows you to create clean, scalable RESTful APIs for the frontend to communicate with the backend seamlessly.
* **Fast Development**: Its simplicity speeds up the development process, especially for projects like **Staff Management**, which involve multiple interconnected modules.

## 2.7. JSON Web Token (JWT)

**JWT** is a secure, compact token format used for transmitting data between parties as a JSON object. It is particularly useful for **authentication** and **authorization** in the **Staff Management** system. Here's how it helps:

* **Authentication**:
  + After a user logs in (e.g., an Employee, HR, or Director), the server generates a JWT that includes the user's role and permissions.
  + The frontend stores this token (usually in local storage or cookies) and sends it with every subsequent request to verify the user.
* **Authorization**:
  + Based on the user's role (e.g., Employee, HR, or Director), the system can restrict access to specific endpoints. For example:
    - An **Employee** can only view their requests.
    - **HR** can approve or reject business trip requests.
    - The **Director** has full access to all requests.
* **Stateless Sessions**:
  + JWT eliminates the need for server-side session storage, making the system lightweight and scalable.

# CHAPTER III: SYSTEM ANALYSIS

## 3.1. Use Case

Use cases play a crucial role in software development by documenting user requirements, clarifying system behavior, facilitating communication among stakeholders, guiding system design, supporting testing and validation, enabling requirement traceability, and serving as a basis for Agile development practices. They help ensure that the resulting system meets user needs and expectations effectively.

A screen shot of a cellphone

Description automatically generated

Figure 6. Usecase Diagram

### 3.1.1. Overview

The **Staff Management** application is designed to streamline HR processes for small and medium-sized enterprises by automating workflows and defining role-based permissions. The application’s functionality revolves around various actors who interact with the system to perform tasks related to leave requests, employee transfers, business trip approvals, overtime management, and recruitment. Each actor has distinct permissions and responsibilities, ensuring efficient operations and clear accountability.

### 3.1.2. Key Actors and Permissions

* **Director**
  + **Role Description**: The Director has the highest level of authority in the system and is responsible for the final approval of requests forwarded by HR or Head of Department.
  + **Responsibilities**:
    - Review and approve/reject requests such as leave applications, transfers, or business trips.
    - Oversee overall HR operations and ensure compliance with organizational policies.
  + **Permissions**:
    - Access to all employee-related requests.
    - Ability to approve or reject any request.
* **Human Resources (HR)**
  + **Role Description**: HR acts as the intermediary manager, responsible for processing and managing requests initiated by employees and Heads of Department.
  + **Responsibilities**:
    - Manage and approve leave requests, recruitment processes, and transfer requests.
    - Forward requests requiring higher authority to the Director for final approval.
    - Maintain employee records and ensure data accuracy.
  + **Permissions**:
    - Create and modify employee records.
    - Approve requests initiated by employees and Heads of Department.
    - Forward selected requests to the Director for final decision.
* **Head of Department**
  + **Role Description**: Heads of Department manage requests and activities specific to their respective departments.
  + **Responsibilities**:
    - Initiate and approve departmental requests such as employee transfers or overtime.
    - Act as the first level of approval for leave requests submitted by employees.
    - Coordinate with HR for broader organizational tasks.
  + **Permissions**:
    - Initiate requests (e.g., transfer requests, overtime approvals).
    - Approve leave requests from employees within their department.
    - View departmental employee details.
* **Employee**
  + **Role Description**: Employees interact with the system primarily to submit requests or view their data.
  + **Responsibilities**:
    - Submit leave requests, overtime requests, and business trip applications.
    - Access personal records and request status updates.
  + **Permissions**:
    - Submit and view requests.
    - Edit personal information (limited fields).
* **Summary**

|  |  |
| --- | --- |
| **Actor** | **Permissions** |
| Director | Approve/reject all requests; View all records; Oversee HR operations. |
| HR | Approve/reject leave, recruitment, and transfer requests; Manage employee records. |
| Head of Department | Approve/reject leave and overtime requests; Initiate transfer requests. |
| Employee | Submit requests (leave, overtime, business trips); View personal information. |

## 3.2. Database Diagram

A database diagram, also known as an entity-relationship diagram (ER diagram), is a visual representation of the structure of a database. It illustrates the relationships between different entities (tables) in the database and the attributes (columns) associated with each entity.

**A screenshot of a computer screen

Description automatically generatedA screenshot of a computer screen

Description automatically generated**

Figure 7. Database Diagram

## 3.3. Activity Diagram

### 3.3.1. Appointment and Transfer Staff

A diagram of a work flow

Description automatically generated

Figure 8. Appointment and Transfer Staff Diagram

The Appointment and Transfer Staffs process starts with the Head Department creating a transfer ticket, which is sent to HR for verification. If information is incomplete, HR requests updates from the Head Department. Once complete, the request is sent to the Director for approval. Approved requests are saved to the system database, while rejections end the process. The Head Department can also cancel the request at any stage before approval, ensuring a structured and efficient workflow.

### 3.3.2. Overtime Management

A diagram of a software project

Description automatically generated

Figure 9. Overtime Management Diagram

The Overtime Manage process starts with the Head Department creating an overtime request for staff and sending it to HR for review. HR verifies the request and checks if all necessary information is provided. If information is incomplete, HR requests additional details from the Head Department, who can revise and resubmit the request. Once complete, HR sends it to the Director for approval. Approved requests proceed to employee notification and are saved to the database, while rejected requests terminate the process. The Head Department can also cancel the request anytime before final approval, ensuring flexibility and proper oversight.

### 3.3.3. Probationary Management

A diagram of a work flow

Description automatically generated

Figure 10. Probationary Management Diagram

The Probationary Manage process begins with HR creating a probationary request form and sending it to the Head Department. The Head Department selects a specific job position and monitors the probationary process. Once the probation period ends, the Head Department evaluates the employee's work performance. If the evaluation is positive, the system notifies that the employee has passed the probationary period, and the data is saved to the database. If the evaluation is negative, a notification is sent indicating the employee failed the probationary period. The process ends after the notification is sent.

### 3.3.4. Day-off Management

A diagram of a work flow

Description automatically generated

Figure 11. Day-off Management Diagram

The Day-Off Management process begins with employee creating a day-off request form and sending it to the Head Department. HR and Head of Department verifies the request and checks if all necessary information is provided. If information is incomplete, HR and Head of Department requests additional details from the employee. Once complete, HR approved requests proceed to employee notification and are saved to the database, while rejected requests terminate the process. The employee can also cancel the request anytime before final approval.

### 3.3.5. Resignation Management

A diagram of a company

Description automatically generated

Figure 12. Resignation Management Diagram

The Resignation Request starts when the employee creating a ticket. This request will be sent to the head of Department for the approval. After that, the request will need to be approved by HR. Then the employee will need to arrange handover work. This process is checked by Head Department. Finally, the process ends and information is saved to database.

### 3.3.6. Recruitment Management

A diagram of a company

Description automatically generated

Figure 13. Recruitment Management

The Recruitment Management process starts with the Head Department creating an recruitment request and sending it to HR for review. HR verifies the request and checks if all necessary information is provided. If information is incomplete, HR requests additional details from the Head Department. Once complete, HR sends it to the Director for approval. Approved requests proceed to employee notification and are saved to the database, while rejected requests terminate the process. The Head Department can also cancel the request anytime before final approval.

### 3.3.7. Business Trip Management

A diagram of a business trip

Description automatically generated

Figure 14. Business Trip Management Diagram

The Business Trip Management process starts with the Head Department creating and submitting a request to HR and the Director for approval. If additional information is needed, the Head updates and resubmits the request. Upon approval, the system sends a notification and saves the data. The Head can cancel the request at any point. This ensures transparency and efficiency.

## 3.4. Implementation

A screenshot of a login form

Description automatically generated

Figure 15. Login Interface

Upon accessing the Staff Management application, users are greeted with a login screen requiring their username and password. This login system ensures that only authorized personnel can access the platform, with their role (Employee, HR, or Director) determining the functionalities they can use.

* The user enters their credentials (username and password) into the login form.
* The frontend sends the login request to the backend API endpoint (/api/auth/login) using a POST method.

The backend performs the following tasks:

* Verification: Checks the provided credentials against the database records in the Users or Employees table.
* Role Identification: Retrieves the user\u2019s role (e.g., Employee, HR, or Director) and permissions.

Once the user is authenticated, the frontend stores the token (e.g., in localStorage or as an HTTP-only cookie).

Based on the role, the system redirects the user to their respective dashboard:

* + Employee Dashboard: Submit requests, view statuses.
  + HR Dashboard: Manage employee records, approve/reject requests.
  + Director Dashboard: Review and approve final-level requests.

A screenshot of a computer

Description automatically generated

Figure 16. Employee Interface

The main function of employee interface is allowed user to see notifications. Side functions is to create day-off request, business trip request, Report Issues in company or Resign

A screenshot of a computer

Description automatically generated

Figure 17. HR Interface

HR in the application can do:

* **Leave Management**

HR plays a critical role in overseeing employee leave requests. This function includes:

* + **Approval Workflow**: Reviewing leave requests submitted by employees or approved by Heads of Department. HR can approve, reject, or request modifications.
  + **Leave Records**: Maintaining a centralized log of leave history for each employee, categorized by type (e.g., sick leave, annual leave).
* **Resignation Management**

Managing employee resignations is an essential HR function. This module includes:

* + **Resignation Requests**: Reviewing resignation applications submitted by employees.
  + **Final Approval**: Forwarding resignation cases to the Director for final approval, if necessary.
* **Internship Management**

This feature helps HR handle interns efficiently, ensuring compliance and seamless integration into the company. Key aspects include:

* + **Intern Performance Tracking**: Recording evaluations and progress throughout the internship period.
  + **Internship Completion**: Issuing certificates or feedback upon successful completion of the internship.
* **Overtime Request Management**

HR ensures proper tracking and approval of overtime requests. This involves:

* + **Request Review**: Verifying overtime applications submitted by employees or approved by Heads of Department.
* **Recruitment Management**

Recruitment is a key responsibility of HR, and this module streamlines the process. It includes:

* + **Job Postings**: Creating and publishing job openings on the platform.
  + **Director Approval**: Forwarding shortlisted candidates to the Director for final review and decision-making.
* **Employee Information Management**

HR ensures accurate and up-to-date records for all employees. This function includes:

* + **Profile Updates**: Adding, editing, and managing employee profiles, including contact details, job roles, and department assignments.
  + **Performance Records**: Maintaining performance evaluations and feedback.

# CHAPTER IV: RESULT AND FUTURE WORK

## 4.1. Achievements

### 4.1.1. Technical Milestones:

Designed a robust database schema using Microsoft SQL, covering all HR functionalities, including leave requests, employee management, and approval workflows.

Implemented the Model-View-Controller (MVC) architecture, ensuring a clean separation of concerns and easy maintainability.

### 4.1.2. User Role Management:

Successfully implemented role-based access control, ensuring Director, HR, Head of Department, and Employee roles have distinct permissions.

### 4.1.3. Testing and Optimization:

Extensive testing using Postman for API endpoints ensured minimal errors and rapid troubleshooting. Optimized data queries to handle large datasets efficiently, ensuring system scalability.

### 4.1.4. Real-World Applicability:

Tailored the application to meet the needs of SMEs, providing a cost-effective alternative to existing HR management solutions.

## 4.2. Future Work

### 4.2.1. Advanced Security Measures:

Encrypt sensitive employee data, such as personal and salary information, to meet data protection standards.

### 4.2.2. Additional Features:

Integrate payroll management to automate salary calculations and payments.

Add an analytics dashboard to visualize employee performance, leave trends, and HR metrics.

### 4.2.3. Scalability Enhancements:

Support integration with third-party tools like Slack and Google Calendar for notifications and scheduling.

Develop a mobile application for on-the-go access to HR functionalities.

### 4.2.4. Localization and Multi-Language Support:

Add multi-language support to cater to a diverse user base, making the application accessible globally.

# REFERENCES

[1] *Dan Abramov, React Documentation, 2021.*[2] *Facebook Open Source, React Overview, 2023.*[3] *Robin Wieruch, The Road to React, 2022.*

[4] *Bob Ward, SQL Server 2019 Revealed: Including Big Data Clusters and Machine Learning, 2019.*[5] *Microsoft Documentation, SQL Server Overview, 2023.*

[6] *David Herron, Node.js Web Development, 4th Edition, 2018.*[7] *Ryan Dahl, Node.js: The Event-Driven Paradigm, 2009.*[8] *RisingStack Engineering, Node.js at Scale: Best Practices for Building Large Applications, 2021.*[9] *Node.js Documentation, Node.js Overview, 2023.*