|  |  |
| --- | --- |
|  |  |
|  |  |
| **Vacation Request** |  |
|  |  |
|  | Summer 2024 – 2025 |
|  | Zoulficar kanso – 42330522  Mohammad Fahes – 42330450  Hassan Joumaa – 42330220  Advisor: Dr. Ghinwa Tarhini |

# Abstract

# 

# Abstract

This project focuses on the design and implementation of a web-based **Multi-Level Vacation Request Approval System**, developed using **React** for the frontend and **Node.js with Express.js** for the backend. The primary objective was to create an intuitive and efficient digital platform that streamlines the traditional, often cumbersome, process of submitting and approving employee vacation requests within an organizational hierarchy.

The system facilitates a clear workflow where employees can submit their vacation requests, which then sequentially pass through a series of managerial approvals (Manager, Chief Manager, and HR Manager). Key functionalities include user authentication for different roles, dynamic dashboards tailored to each user's permissions, real-time tracking of request statuses, and automated email notifications upon final HR approval. While the current iteration utilizes an in-memory data store for demonstration purposes, the architecture is designed to be scalable for future database integration.

The successful development of this system demonstrates the practical application of modern web development technologies to address real-world administrative challenges. It significantly enhances efficiency, transparency, and communication within the vacation approval process, providing a robust solution for organizational leave management.

# Table of Contents

[Abstract 2](#_gjdgxs)

[Table of Contents 3](#_1fob9te)

[Introduction 4](#_3znysh7)-5

[Methodology 6-7](#_2et92p0)

[Hardware and Software 8](#_tyjcwt)

[Results 9](#_3dy6vkm)-12

[Conclusion 13](#_1t3h5sf)

[References 14](#_4d34og8)

[Appendix I: Information 15](#_2s8eyo1)

# Introduction

### ****Web Development: Overview and Purpose****

Web development is the process of creating and maintaining websites and web applications that are accessible via the internet . It encompasses several disciplines, including front-end development (the user interface), back-end development (server-side logic), database management, and full-stack development, which combines both front and back ends .

The primary goals of web development are to :

* Deliver interactive and user-friendly digital experiences.
* Provide secure and scalable platforms for communication, e-commerce, education, and enterprise applications.
* Enable fast and responsive design across various devices.
* Ensure maintainability and performance in evolving technical environments.

In today's technology-driven world, web development plays a critical role in enabling digital transformation across industries. Tools and frameworks such as **Node.js** and **React** have become essential for modern developers, offering efficient, scalable, and real-time solutions.

### ****2. Technologies Used: Node.js and React****

**Node.js** is a back-end JavaScript runtime environment built on Chrome's V8 engine. It allows developers to run JavaScript on the server side, making it possible to use a single language throughout the full stack. Node.js is known for:

* Handling concurrent requests efficiently using non-blocking I/O.
* Supporting scalable, real-time applications.
* Having a vast ecosystem of open-source packages via npm.

**React** is a front-end JavaScript library developed by Facebook for building dynamic user interfaces. It is component-based, which means developers can build encapsulated elements that manage their own state, and then compose them to create complex UIs. Key features of React include:

* Virtual DOM for optimized rendering.
* One-way data binding.
* Strong community support and vast libraries.

Combining **React for the front end** and **Node.js for the back end** allows developers to create full-stack JavaScript applications that are fast, interactive, and scalable.

### ****3. Project Context: Vacation Request Form****

The main objective of this project is to develop a **Vacation Request Form** using **React** for the front-end interface and **Node.js** for the back-end server logic.

The workflow of the application is structured as follows:

* The **employee** fills out a vacation request form through a **web interface** built with React.
* Once submitted, the request is processed and passed through a **multi-level approval chain** managed by the Node.js server:
  1. First reviewed by the **Manager**.
  2. Then approved by the **Chief Manager**.
  3. Finally, it goes to the **HR Manager** for the final decision.
* After completing all approval stages, the system sends an **email notification** to the employee, informing them whether the vacation has been **approved or rejected**.

This project serves several key purposes:

* **Streamlines internal HR processes**, reducing paperwork and improving efficiency.
* **Ensures accountability** and tracking across the entire approval process.
* **Enhances communication** by automating email notifications and status updates.
* **Provides hands-on experience** in building a real-world approval system using modern web development tools.

# Methodology

The development process was structured into several key phases:

**a. Planning and Analysis**

* Defined the core requirements: form structure, approval workflow, and notification mechanism.
* Mapped out the user journey: employee submits → managers approve → system notifies.

**b. Front-End Development (React)**

* Designed the user interface using **React**, with components for input fields, submission buttons, and status updates.
* Implemented form validation to ensure that users provide complete and accurate information.

**c. Back-End Development (Node.js)**

* Built a RESTful API using **Node.js** and **Express.js** to handle form submissions, approval status updates, and email notifications.
* Created endpoints for each user role (Employee, Manager, Chief Manager, HR) to access and manage requests.

**e. Email Notification System**

* Integrated **Nodemailer** (Node.js library) to automatically send emails to the employee upon final decision.

**f. Testing and Validation**

* Performed manual and functional testing on ( submission, approval flow, email delivery, input validation).

### ****2. Research and Data Collection Methods****

To ensure the design and logic of the system matched industry standards and user expectations, the following research activities were conducted:

* Reviewed existing HR workflow systems and leave management portals.
* Studied multi-level approval patterns through academic articles and software engineering blogs.
* Analyzed tutorials and documentation for **React**, **Node.js.**
* Watched educational videos and walkthroughs to refine implementation techniques.

All this research helped shape both the technical structure and user interface of the application.

### ****3. Equipment and Technology Used****

The following technologies, libraries, and tools were used during the development of this project:

| **Tool / Technology** | **Purpose** |
| --- | --- |
| **React.js** | Front-end development – building interactive user interfaces |
| **Node.js** | Back-end development – handling business logic and API routes |
| **Express.js** | Web framework for Node.js – building RESTful APIs |
| **Nodemailer** | Email service – sending notifications to users |
| **Postman** | API testing – simulating and debugging HTTP requests |
| **Visual Studio Code** | Code editor – writing and managing source code |

# Hardware and Software

**Hardware (Physical Tools I Used)**

Laptop (Hp laptop 15s , Core i7 , 16GB RAM , generation 12 )

**Software (Programs & Tools I Used)**

**Frontend (What Users See)**

Languages: HTML, CSS, JavaScript

Frameworks/Libraries: React

**Backend (Server-Side Logic)**

Languages: Node.js.

**Other Tools**

Code Editor: VS Code, Sublime Text

APIs: RESTful APIs.

# Results

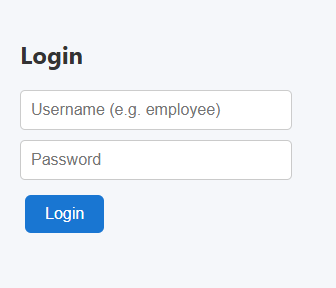
### Implemented Features and Functionality

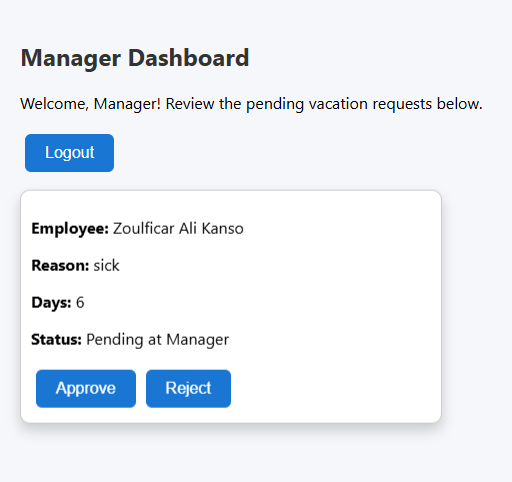
The developed web application provides a robust and intuitive platform for managing employee vacation requests through a multi-level approval workflow. The core functionalities include:

1. **Employee Request Submission**: Employees can easily log in, access a dedicated dashboard, and submit new vacation requests by providing their name, reason for leave, and number of days. The system automatically associates the request with their logged-in employeeId and stores their email for notifications.
2. **Multi-Level Approval Workflow**: Requests progress sequentially through three distinct managerial roles:
   * **Manager**: The first level of approval. Managers can view pending requests and choose to Approve (advancing the request to the Chief Manager) or Reject it.
   * **Chief Manager**: The second level of approval. Chief Managers review requests approved by the Manager and can Approve (advancing to HR) or Reject them.
   * **HR Manager**: The final level of approval. HR Managers review requests approved by both previous levels. Upon Approval, the request status becomes Accepted, and an automated email notification is sent to the employee. If Rejected at any stage, the request status updates accordingly.
3. **Real-time Request Tracking**: Each dashboard displays requests relevant to that role's current approval stage, ensuring clear visibility for all stakeholders. Employees can also see the status of their submitted requests.
4. **Automated Email Notifications**: Upon final acceptance by the HR Manager, the system leverages Nodemailer to send a personalized email to the employee, confirming their vacation approval. This streamlines communication and provides timely updates.

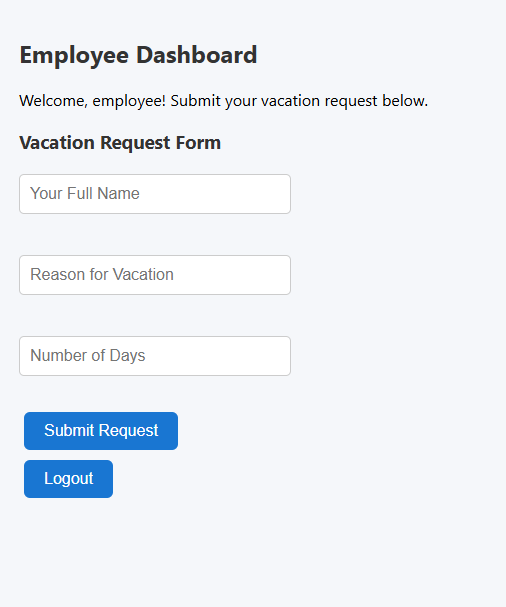
### User Interface and Experience

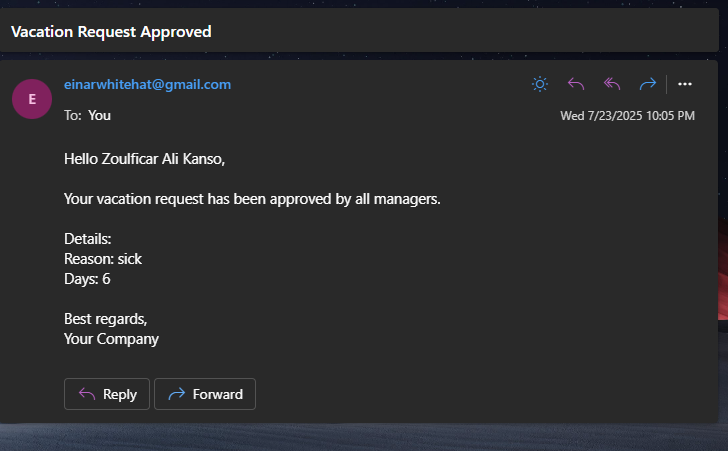
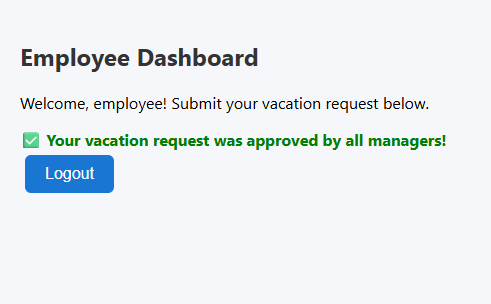
The frontend, built with **React**, offers a responsive and user-friendly interface. Each user role (Employee, Manager, Chief Manager, HR Manager) has a dedicated dashboard, simplifying navigation and access to relevant information. The application ensures a smooth user flow from request submission to final approval. Key visual elements include clear forms, status indicators, and intuitive action buttons.

* **Figure 1: Login Page**
  + 
* **Figure 2: Employee Dashboard with Vacation Request Form**

****

* **Figure 3: Manager Dashboard with Pending Requests**

****

* **Figure 4: HR Manager Dashboard and Request Approval & Email Notification.**
  + 

### Accomplishments and Challenges

**Accomplishments**:

* Successfully implemented a fully functional multi-level approval system from scratch using modern web technologies.
* Established clear communication channels between different organizational levels through sequential approvals and automated email alerts.
* Developed a modular and maintainable codebase, separating frontend and backend concerns.
* Gained practical experience in full-stack development, API design, and integrating third-party services like Nodemailer.

**User Role Permissions Table :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User Role | Can Login | Can Submit Vacation Request | Can View All Requests | Receives Email Notification |
| Employee | Yes | Yes | Only their own requests | Yes (on final approval) |
| Manager | Yes | No | Yes | No |
| Chief Manager | Yes | No | Yes | No |
| HR Manager | Yes | No | Yes | No |

# Conclusion

This project successfully developed a functional web-based vacation request approval system, demonstrating the practical application of **React** for building interactive user interfaces and **Node.js with Express.js** for robust backend logic. The system effectively automates the traditional multi-level approval process, from employee submission through managerial review to final HR acceptance and automated notification.

The implementation addresses key organizational needs by streamlining the vacation approval workflow, enhancing transparency through status tracking, and improving communication efficiency via email alerts. This project has provided valuable insights into full-stack web development, API design principles, and the benefits of adopting modern JavaScript ecosystems for enterprise solutions.

### Future Work

To further enhance the system's capabilities and robustness, the following future improvements are recommended:

* **Database Integration**: Implement a persistent database (e.g., **SQL Server**, MongoDB, PostgreSQL) to store vacation requests, user data, and roles. This would ensure data durability, scalability, and allow for more complex querying and reporting beyond the current in-memory storage.
* **Robust User Authentication**: Develop a more secure and comprehensive user authentication and authorization system (e.g., using JWT tokens,bcrypt for password hashing) to manage user accounts and roles more effectively, moving beyond the current static user definitions.
* **Admin Panel**: Create an administrative interface for HR or IT to manage users, roles, and potentially customize approval workflows or notification templates.
* **Advanced Reporting**: Implement features for generating detailed reports on vacation trends, accumulated leave, or approval times.
* **Calendar Integration**: Integrate with calendar services (e.g., Google Calendar, Outlook Calendar) to automatically add approved vacation dates to an employee's calendar.
* **Enhanced Error Handling and Validation**: Implement more comprehensive server-side input validation and error handling to provide more informative feedback to users and ensure data integrity.

# References

Here are the main sources used for the content in the introduction :

1. **Mozilla Developer Network (MDN Web Docs)** – <https://developer.mozilla.org/>
2. **Node.js Official Documentation** – https://nodejs.org/en/docs
3. **React Official Documentation** – <https://react.dev>
4. **GeeksforGeeks – Node.js and React tutorials** – <https://www.geeksforgeeks.org>
5. **FreeCodeCamp – Web Development Curriculum** – <https://www.freecodecamp.org>
6. **Coursera Course on Full Stack Web Development with React** – <https://www.coursera.org/specializations/full-stack-mobile-app-development>

# Appendix I: Information

**Simplified pseudo-code for Vacation Request Approval**

FUNCTION handleApproval(requestId, approverRole, action)

FIND request BY requestId

IF request NOT FOUND THEN RETURN error

IF approverRole IS "Manager" AND request.status IS "Pending at Manager" THEN

SET request.status TO (action IS "approve" ? "Pending at Chief Manager" : "Rejected by Manager")

ELSE IF approverRole IS "Chief Manager" AND request.status IS "Pending at Chief Manager" THEN

SET request.status TO (action IS "approve" ? "Pending at HR Manager" : "Rejected by Chief Manager")

ELSE IF approverRole IS "HR Manager" AND request.status IS "Pending at HR Manager" THEN

SET request.status TO (action IS "approve" ? "Accepted" : "Rejected by HR Manager")

IF request.status IS "Accepted" THEN SEND EMAIL to employee confirming approval

ELSE

RETURN error (Invalid approval stage or role)

RETURN updated request status

END FUNCTION