**OMEGA – EMPLOYEE MANAGEMENT SYSTEM**

|  |
| --- |
| A Project Report Presented to  CMPE-272  Fall, 2021 |
|  |

|  |
| --- |
| By |
| Swathi Anandram  Aryan Jadon  Harika Nalam  Supreetha M A  Shreya Nimbhorkar |
| 08/12/2021 |

|  |
| --- |
| Copyright © 2021 |
| Swathi Anandram  Aryan Jadon  Harika Nalam  Supreetha M A  Shreya Nimbhorkar |
| ALL RIGHTS RESERVED |

**ABSTRACT**

**OMEGA**

By

Swathi Anandram

Aryan Jadon

Harika Nalam

Supreetha M A

Shreya Nimbhorkar

Employees are the tangible assets of any organization who play a vital role in the successful operation of any company. Omega is a platform for centralized employee management that focuses on easy access to the company portal. The project focuses on designing and implementing an enterprise software application to manage employee-related information.

The project's main objective is to develop a role-based system that consists of different modules such as training for employee management, project management, login module, employee-manager relationship, admin management, and organization-level management.

Omega provides a dashboard-driven design to navigate to their corresponding sections. Employees can log in to perform the following tasks - the list of projects the employee works under and a learning portal that indicates the manager's completed training.

The manager can log in to perform the following tasks - post-training for employees, check the training status and add employees to projects under him.

Employees and managers can log in to perform the following tasks: access an employee search directory, password management, view their respective profiles, a storage tab to store mark sheets and certificates for quick retrieval, and a notification tab informing any upcoming events or news posted by Omega Org.

Omega Org login will perform the following tasks - notify the whole organization about any upcoming events or news. Admin will perform the following tasks: unblock the employees who have locked their accounts, maintain the web page, and add additional tabs on the employee and manager screens.

**Acknowledgements**

We want to show our gratitude and appreciation to everyone who made it possible to finish this project. Our professor, Mr. Andrew Bond, deserves special recognition for his assistance, stimulating suggestions, and encouragement throughout the fabrication process and drafting this report. We also appreciate the time he spent proofreading and correcting our errors. We owe a debt of gratitude to all our classmates, particularly our friends, for volunteering their time to assist and support us in developing our project.

Table of Contents

**Chapter 1 Introduction**

* 1. Project goals and objectives
  2. Problem and motivation
  3. Project application and impact
  4. Project results and deliverables
  5. Market research
  6. Project report structure

**Chapter 2 Project Background and Related Work**

* 1. Background and used technologies

2.2 State-of-the-art technologies

* 1. Literature survey

**Chapter 3 System Requirements and Analysis**

3.1 Domain and business requirements

3.2 Customer-oriented requirements

3.3 System function requirements

3.4 System behavior requirements

3.5 System performance and non-function requirements

3.6 System context and interface requirements

3.7 Technology and resource requirements

**Chapter 4 System Design**

4.1. System architecture design

4.2. System data and database design *(for software project only)*

4.3. System interface and connectivity design

* 1. System user interface design *(for software project only)*
  2. System component API and logic design *(for software project only)*

4.6 System design problems, solutions, and patterns

**Chapter 5 System Implementation**

5.1. System implementation summary

5.2. System implementation issues and resolutions

5.3. Used technologies and tools

**Chapter 6 System Testing and Experiment**

6.1 Testing and experiment scope

6.2 Testing and experiment approaches

6.3. Testing report (or case study results)

**Chapter 7 Conclusion and Future Work**

7.1 Project summary

7.2 Future work

**References**

**List of Figures**

Figure 1. Sample Dashboard Interface of Connecteam 2

Figure 2. Sample Dashboard Interface of Remote 3

Figure 3. Class Diagram 6

Figure 4. Use Case Diagram 8

Figure 5. Component Diagram 8

Figure 6. System Architecture Design 11

Figure 7. Database design 11

Figure 8. System Interface And Connectivity Diagram 13

Figure 9. Sign In page 10

Figure 10. Omega user homepage 10

Figure 11. Admin sign up homepage 11

Figure 12. No access page 11

Figure 13. Manager Homepage 12

Figure 14. Employee Homepage 12

Figure 15. Notifications Visualization 13

Figure 16. Admin Employee Login 13

Figure 17. EMS 14

Figure 18. Add Employee EMS 14

Figure 19. Leave Management System 15

**List of Tables**

Table 1. Customer-oriented requirements details 6

1. **Introduction**
   1. **Project goals and objectives**

To develop an employee management system that focuses on easy access to the company portal. The project focuses on designing and implementing an enterprise software application to manage employee-related information. The project's main objective is to develop a role-based system that consists of different modules. The application facilitates excellent communication or interaction between different modules and roles.

* 1. **Problem and motivation**Consider a scenario in which each piece of essential data relating to the most valuable assets, i.e., employees, is accessible via a centralized database on a single dashboard and is accessible at lightning speed 24 hours a day, seven days a week. A sound Employee Management system acts as a single center for all your leave management, project management, and admin responsibilities.

Omega is a centralized staff management software that prioritizes quick access to the company site. The project's goal is to create and execute an enterprise software application to handle employee data.

**1.3 Project application and impact**

Following are the benefits of an EMS –

* Better Efficiency and Accuracy
* Compliance Risks are Reduced
* Increased Profitability
* There are very few manual errors.
* Increased Productivity
* Lower Costs Due to Higher Motivation

Implementing an EMS framework for employees is akin to a corporation pushing mountains, which is especially true when dealing with a remote workforce.

**1.4 Project results and expected deliverables**

Omega - an employee management system is a distributed system designed to keep track of employee information and the company's workflow process. The application facilitates excellent communication or interaction between different modules and roles.

Following component interactions are expected as results of the systems:

* Employees can log in to perform the following tasks - the list of projects the employee works under and a learning portal that indicates the manager's completed training.
* The manager can log in to perform the following tasks - post-training for employees, check the training status and add employees to projects under him.
* Employees and managers can log in to perform the following tasks: access an employee search directory, password management, view their respective profiles, a storage tab to store mark sheets and certificates for quick retrieval, and a notification tab informing any upcoming events or news posted by Omega Org.
* Omega Org login will perform the following tasks - notify the whole organization about any upcoming events or news.

**1.5** **Market research**

Following are the product profiles of major companies

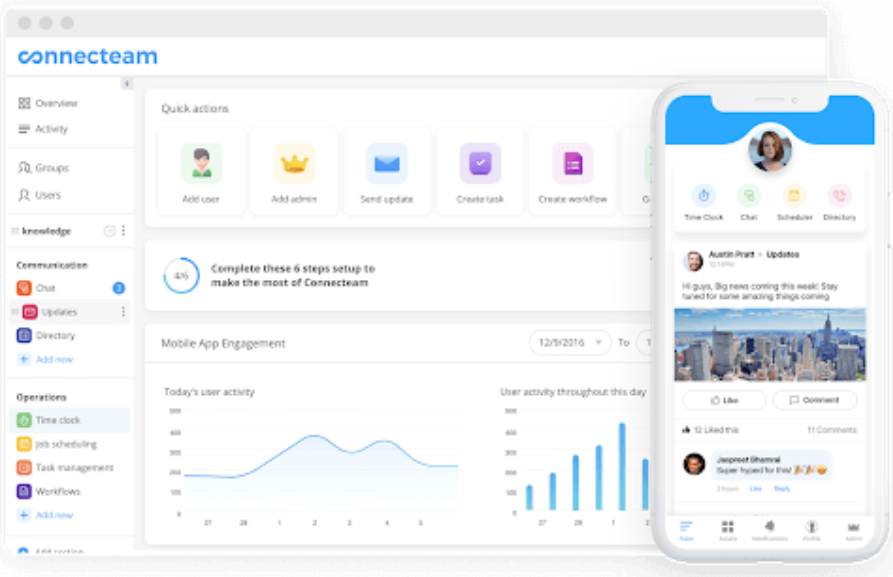
* Connecteam - Connecteam is a dedicated employee communication channel with targeted updates, team engagement features, surveys, polls, a suggestion box to collect employee feedback, a searchable company directory. It also provides organizational charts, training, onboarding capabilities, a company library, time management, digital forms and checklists, quick tasks, and more in Connecteam's features.
* Connecteam is simple to use, customize, and set up. Connecteam can assist increase team communication and employee engagement by automating daily procedures. 

Figure 1: Sample Dashboard Interface of Connecteam

* Remote - Companies of all sizes can use Remote to pay and manage full-time and contract workers worldwide. Every continent has local employment law and HR specialists that assist you in creating culturally sensitive employment packages that help you build trust with your global staff.
* Remote allows you and your team to enroll easily and pay contractors and workers in one contemporary, secure, cloud-based platform, allowing you and your team to focus on more strategic tasks. You will have access to all your most crucial employment paperwork, data, and HR operations in one place, thanks to Remote's personnel management system. Wherever your team works, Remote's team of professionals takes care of your regional payroll, benefits, compliance, IP protection, and taxes needs.

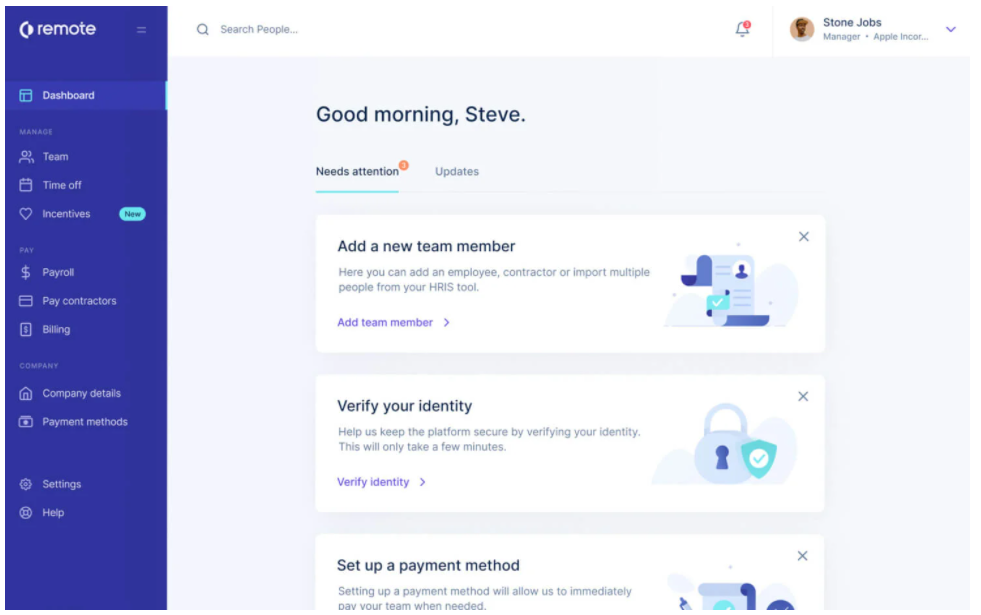


Figure 2: Sample Dashboard Interface of Remote

* 1. **Project Report Structure**

This project report follows the below-mentioned structure

* Introduction
* Project Background and Related Work
* System Requirements and Analysis
* System Design
* System Testing and Experiment
* Conclusion and Future Work

**Chapter 2 Background and Related Work**

* 1. **Background and used technologies**

Employee management is a method of assisting your employees in achieving their full potential and achieving your company's objectives. It is a comprehensive process that encompasses practically all aspects of human resources, including new hire recruitment, payroll management, performance management, and more.

Because personnel is one of the most critical components in its success, organizations must treat them with respect. As a result, you must consider any concerns of your employees and provide them with a better employee experience. Some of the most significant issues in employee management those businesses must be prepared to face are listed below.

* Employee engagement is critical.
* Employee productivity is a challenge.
* It maintains a record of employee attendance.
* It is not easy to track down and record information on employees.

Following are the technologies used in our project –

* Frontend - Angular 8
* Backend - Python
* Framework - Django, RESTAPI
* Authentication - Cognito
* Automation - Jenkins, Zappa, Sonarcloud
* Hosting Platform - Heroku
  1. **State-of-the-art**

Choosing the best employee software among the large array of personnel management apps available might be challenging at times. Following are the two best existing employee management apps found during our literature survey.

1. Connecteam: Managers can now have comprehensive real-time monitoring and control over daily operations and procedures thanks to Connecteam. Employee hours can be tracked, reducing buddy punching and time theft. Managers can cut the time to create a work schedule for their employees in half. With digital checklists and daily activities, you can reduce day-to-day friction and streamline operations. With Connecteam, it is time to eliminate pen and paper manual processes and automate practically every aspect of personnel management.
2. Gusto: Gusto (previously ZenPayroll) is a cloud-based payroll solution among the best and most widely used. Small and medium-sized enterprises can benefit from Gusto's built-in benefits administration at a low cost. Gusto's platform boasts a remarkably user-friendly design and automates many aspects of the payroll process. This software provides an employee onboarding interface that allows new workers to all of their personal information (such as bank account information) into the system immediately.
   1. **Literature survey**

We conducted research and spoke with businesses that have already implemented personnel management software and found that Employee management software solutions are the new modern workplace essential that most businesses are implementing to manage their workforce better. It is a no-brainer if jobs can be automated and completed in real-time without lag, and your staff is happier since everything is going smoothly.

Our survey concluded that to address all the shortcomings of current employee management software. The system will oversee keeping track of employee information, including their profiles. The system will include leave management, from application to approval or denial of leave requests, and all employee projects, with close monitoring of projects from start to finish and training to monitor active and inactive personnel.

We referred following papers for our project implementation –

<https://www.researchgate.net/publication/347615091_Employee_Management_System>

<https://www.researchgate.net/publication/338216184_Challenges_of_Implementing_an_Employee_Management_System_for_Improving_Workplace_Management_Effectiveness>

<https://journals.sagepub.com/doi/10.1177/2158244020969383>

**Chapter 3 System Requirements and Analysis**

**3.1.** **Domain and business Requirements**

Diagram

Description automatically generated

Figure 3: Class Diagram

**3.2. Customer-oriented requirements**

|  |  |
| --- | --- |
| User Group | Use cases |
| Omega org | post events and updates of org |
| Employees | login, perform a task, check the project's updates, training |
| Managers | assign people to projects and post-training |
| Admin | add new employees, handle issues of people accounts |

Table 1: Customer-oriented requirements details

**3.3. System (or component) function requirements**

**High-level functional features:**

* Prominent roles - employee, manager, org, admin, CEO (owner of the company)
* Employee - login, perform a task, check the project's updates, training
* Manager - assign people to projects and post-training
* admin - add new employees, handle issues of people account.
* org - post events and updates of org

The system should comprise of below listed APIs and corresponding behaviors/ functions:

* On a load of any screen:

1. GET ENDPOINT TO GET THE NOTIFICATIONS: Get from the Omega table

* For Manager Screen:

1. GET ENDPOINT FOR NUMBER OF PROJECTS: Count of projects under him -> List of projects id

2. GET ENDPOINT FOR NUMBER OF TRAININGS: Count from Training table based on the managers’ projects

3. GET ENDPOINT FOR NUMBER OF MEMBERS UNDER HIM: Count from Project table under the and get the count of employees

4. POST ENDPOINT TO ADD PROJECTS: Add project Description title and get the manager id from login and select all the employees from the database

5. POST ENDPOINT TO ASSIGN TRAININGS: Name, description, URL, and projects under him

6. GET ENDPOINT FOR Dropdown values to assign employees -> search string search from all employees

7. GET ENDPOINT: Get all the trainings data that is assigned to which project and the status of it - > suppose there are two employees in the project; both need to click on the completed icon in order that the primary IsCompleted Boolean value becomes true

* For Employee Screen:

1. GET ENDPOINT FOR PROJECTS: Get all the project details of projects based on project ids

2. GET ENDPOINT FOR TRAININGS: Get all the training details from the trainings table based on project id

3. POST ENDPOINT FOR COMPLETED TRAININGS:

* For Omega:

1. POST ENDPOINT: to add a notification along with the date

**3.4. System performance and non-function requirements**

* The system requires a good level of performance as it is one of the primary tenets of any application to provide an excellent experience to its user; certain functionalities like serving notifications to all users require delivery timely.
* The system should have a capacity to cater to at least 50 concurrent users at a time.
* Though there can be a rise in the organization's size, availability should be close to 100%.
* With Security being one of the necessities, the system should incorporate authentication and authorization, encrypt sensitive data, and follow all security best practices.
  1. **System behavior requirements**

**Diagram

Description automatically generated**

Figure 4: Use Case Diagram

**Diagram, schematic

Description automatically generated**

Figure 5: Component Diagram

* 1. **Context and interface requirements**

### *Employee Dashboard:*

* **Profile Tab -** basic contact information (Manager information Manager name as a hyperlink - will show manager details).
* **Projects tab -** List of projects he has assigned and billed, learning portal (List of trainings provided by manager or organization)
* **Employee documents -** Storing PDFs and certificates obtained for quick retrieval can only be viewed by organization and employee
* **Employee Search Directory TAB - all** organization **employees’ details**
* **Rewards and Recognition TAB**

### *Manager login Dashboard:*

* Profile tab - basic contact information
* Settings tab - Password Management
* Projects tab - Post trainings (URL, He can see how many employees have completed the trainings (export to excel)), List of projects (Discussion forum, add employees to particular project)
* Standard notification posted by the organization

### *Admin - Super User:*

* Maintains the website, can add new sections to the website.
* When he opens his screen, he can add or remove sections

### *Org:*

* Dashboard: Post standard notifications for throughout org (Vaccine status news, WFH updates, virtual events, company achievements news)
* Employees and managers count

*In summary, Front-end Screen units*:

* Login screens
* Dashboard screen (manager, employee, organization, and admin dashboards)
* Content Upload screen (link to SJSU / professor view and student view)
* Profile
  1. **Technology and resource requirements**
* FrontEnd - Angular 8
* Backend - Python
* Framework - Django, RESTAPI
* Authentication – Cognito
* Automation - Jenkins, Zappa, Sonarcloud
* Hosting Platform - Heroku

**Chapter 4 System Design**

**4.1 System architecture design**

Diagram, schematic

Description automatically generated

Figure 6: System Architecture Design

* 1. **System data and database design**

Diagram

Description automatically generated

Figure 7: Database design

Roles and Functionality

Omega Admin - Only Admins can create employees. Admin first has to register the employee using a username and temporary passwords as inputs. After employee creation, more information about employees with respective fields gets added to the employees' database.

Username, Profile Image, First Name, Last Name, Other names, Birthday, Department, Project, Training, Employment Date, Employee Type, Employee ID, Date Issued.

Omega Admin also has the functionality to Enable and Disable an Employee Account. Employees cannot log in to their respective accounts once an account is disabled. Omega Admins also have access to the leaves requested by the employees, and they can override the leave approvals. Omega Admins can create notifications for all employees.

Omega Company Employees - Employees can log in to the dashboard. They can check their profile information, update their passwords, and request leaves. Once requested, the employee managers get notifications.

Omega Company Managers – Managers, can log in to the dashboard and check for the number of employees assigned for supervision. Managers can check information like training assigned, Project assigned, and Leaves Requested by employees. Managers can approve or reject the employees' request for Leaves.

* 1. **System interface and connectivity design**

Diagram

Description automatically generated

Figure 8: System Interface And Connectivity Diagram

* 1. **System user interface design**

**Graphical user interface, application, Teams

Description automatically generated**

**Figure 9: Sign In page**

Graphical user interface, application, Teams

Description automatically generated

Figure 10: Omega user homepage

Graphical user interface, application, Teams

Description automatically generated

Figure 11: Admin sign up homepage

Graphical user interface, application, Teams

Description automatically generated

Figure 12: No access page

A screenshot of a computer

Description automatically generated with medium confidence

Figure 13: Manager Homepage

Graphical user interface, application, Teams

Description automatically generated

Figure 14: Employee Homepage

Graphical user interface, application, Teams

Description automatically generated

Figure 15: Notifications Visualization

Graphical user interface, table

Description automatically generated

Figure 16: Admin Employee Login

Text

Description automatically generated

Figure 17: EMS

Graphical user interface, application

Description automatically generated

Figure 18: Add Employee EMS

Graphical user interface, text, application, email

Description automatically generated

Figure 19: Leave Management System

* 1. **System component API and logic design***(for software project only)*

Website Backend uses Django Framework; Complete Backend Code is under Backend Repository Folder.

Django Apps created for the backed - accounts, api\_services, Dashboard, documents\_api, employee, leave, notifications.

accounts app - accounts app has the functions and routes of extracting information about the employees.

api\_services app - API services app uses the functions in all other apps and returns the response JSON for front-end calls.

dashboard app - dashboard app contains the functions to map employee leave and update information requests.

documents\_api app- documents app contains the file uploading functionality to amazon s3 buckets (employee Image and documents)

employee app - employee app has functions related to employee onboarding and employee account purpose.

Leave app- leave app have the functionality of leave management and requested created by the employee's

Notifications app- notifications app has the functionality of creating notifications for the employees.

Other Backend Folders and Files -

Media: Media folder contains the files like images uploaded by the employees. This folder is only used in development/testing.

Static - Static Folder contains the files used by the Dashboard. Static files have files like CSS, JavaScript, and Images files.

Profile - contains gunicorn settings for deployment purposes.

* 1. **Design problems, solutions, and patterns**
* The notifications panel developed for the Omega user role had to publish for all the roles. A role-based check achieves this.
* The foreign key link between tables was an obstacle that we overcame by rigorous research.
* The test cases failed for a few APIs and are corrected by continuous research.

**Chapter 5 System Implementation**

**5.1. System implementation summary**

The system implementation uses the tools and technologies mentioned in Section 5.3. The system has been implemented based on the diagrams mentioned in the previous sections and followed the milestones detailed in the project plan report. The system has been implemented for easy access to the company portal to manage Omega's necessary operations. The system follows an iterative approach for the implementation. The system undergoes various stages to achieve the mentioned goals and a successful working application. It consists of developing the application with easy access to the user interface, a clean backend for necessary APIs, and a well-maintained database for proper data storage. The system provides all-time access to the application.

**5.2. System implementation issues and resolutions**

* Continuous discussions and research resolved the issues faced during the database design.
* Issues faced for authentication were resolved using AWS Cognito and Auth Guards on the frontend using Angular.

**5.3. Used technologies and tools**

### FrontEnd - Angular 8

### Backend - Python

### Framework - Django, RESTAPI

### Authentication – Cognito

### Automation - Jenkins, Zappa, Sonarcloud

### Hosting Platform - Heroku

**Chapter 6 System Testing and Experiment**

**6.1 Testing and experiment scope**

As the first step of testing, analyze the requirements which help to identify the items and functions to be tested. We have written unit test cases to verify the functional behavior of functional components. The unit test cases are dedicated to Functional Verification Testing.

**6.2 Testing and experiment approaches**

Once we gather the requirements of projects and identify the scope of testing, then we define the testing approach to achieve the test documents. The unit tests are used to verify the functions of the application. We have built the application using the Django framework and have an automated testing feature to verify the functionality of the components. Application is tested in two phases. All the application features are tested in the local environment and then the features. The application performance is tested in the production environment as a second phase of testing before releasing it.

**6.3 Testing and experiment**

The features of the application tested are:

* The features adding new employees to the organization portal and their basic details to the portal are done by the admin
* When an employee login to the portal he can see his basic information like projects he is handling(working), the training he was assigned, and a notification posted by the organization.
* Organization posting new notification.
* Admin approving or rejecting leave application.

Application developed by handling all the edge cases. All the features went through two phases of testing. There are no failures of application.

The Omega Organization portal is ready for release.

**Chapter 7 Conclusion and Future Work**

**7.1 Project summary**

Omega, a platform for centralized employee management, has successfully proved its ease of use in managing employee-related information. The implementation is complete, with all the modules integrated and functional. It was a good learning experience to achieve the milestones in Omega. We got excellent exposure to the latest technology frameworks and their capabilities. Working with Amazon Cognito, Jenkins, Zappa, Heroku, and SonarCloud was an enlightening experience.

**7.2 Future work**

Possible upcoming features/ enhancements to Omega include:

* Reminders for mandatory training’s employees must complete.
* A full-fledged training experience for employees
* Feature for HR to launch pulse surveys for employees

The project plan report can be found on the following link:

[https://github.com/CMPE-272-5ThePeople/Enterprise-SW-Platforms/blob/main/Reports/Project\_Plan\_document\_v1.0 (1) 5ThePeople.docx](https://github.com/CMPE-272-5ThePeople/Enterprise-SW-Platforms/blob/main/Reports/Project_Plan_document_v1.0%20(1)%205ThePeople.docx)

The project video can be found on the following link:

<https://github.com/CMPE-272-5ThePeople/Enterprise-SW-Platforms/blob/main/Reports/CMPE272_PROJECT_OMEGA.mp4>

**References**

[1] <https://www.researchgate.net/publication/347615091_Employee_Management_System>

[2] <https://www.researchgate.net/publication/338216184_Challenges_of_Implementing_an_Employee_Management_System_for_Improving_Workplace_Management_Effectiveness>

[3] <https://journals.sagepub.com/doi/10.1177/2158244020969383>

[4] <https://devcenter.heroku.com/>

[5] <https://aws.amazon.com/>

[6] <https://sonarcloud.io/>

[7] <https://peoplemanagingpeople.com/tools/employee-management-system/>