Table of Contents

Introduction

Overview of the project - Employee Management System

In today's fast-paced business environment, efficient management of employee data and organizational structure is crucial for success. To address these needs, we present a robust and scalable Employee Management System designed for EPI-USE Africa. This cloud-hosted application offers a comprehensive solution to manage the organization's employee hierarchy with ease and precision.

The Employee Management System is built using a combination of cutting-edge technologies to deliver a seamless user experience. It leverages Firebase for authentication and Firestore for real-time database management, ensuring secure and reliable access to employee data. The frontend is crafted with Vue.js, providing a responsive and intuitive interface for users.

Key Features

- **Dynamic Employee Management:** Users can create, read, update, and delete employee records with ease. The system accommodates various data fields including name, surname, birth date, employee number, salary, role/position, and reporting line manager.
- **Visual Hierarchy Representation:** The application includes a visual representation of the organizational structure, allowing users to view and navigate the employee hierarchy
- **Search and Filter Capabilities:** Users can efficiently search and sort employee data based on multiple criteria, facilitating quick access to relevant information.
- **Gravatar Integration:** Employee profile pictures are fetched from Gravatar, providing a professional touch to each employee's profile.
- **Cloud Deployment:** The solution is deployed to Firebase Hosting, ensuring high availability and scalability. Users can access the application through a provided URL, with all data stored and managed securely in the cloud.

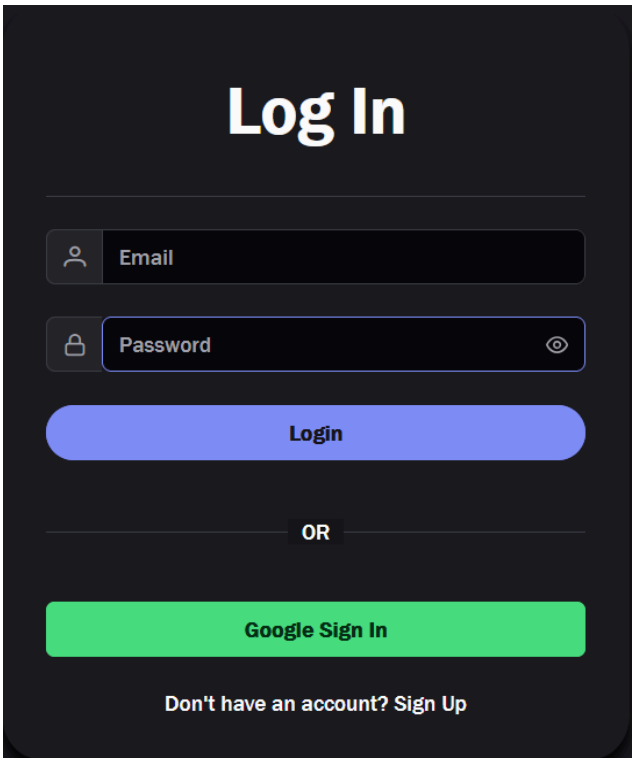
Purpose and Benefits

This system aims to streamline the management of employee data and enhance organizational transparency. By integrating a visual hierarchy and offering powerful search and filtering tools, the application empowers users to efficiently handle employee information and maintain an up-to-date organizational structure. The Gravatar integration adds a personalized element, while the cloud deployment ensures that the system remains accessible and scalable as the organization grows.

This document provides a detailed overview of the application's features, architecture, and design choices, along with instructions for users and technical insights for developers. We hope this system significantly improves the management of EPI-USE Africa's employee data and contributes to overall organizational efficiency.

User Manual

Logging In



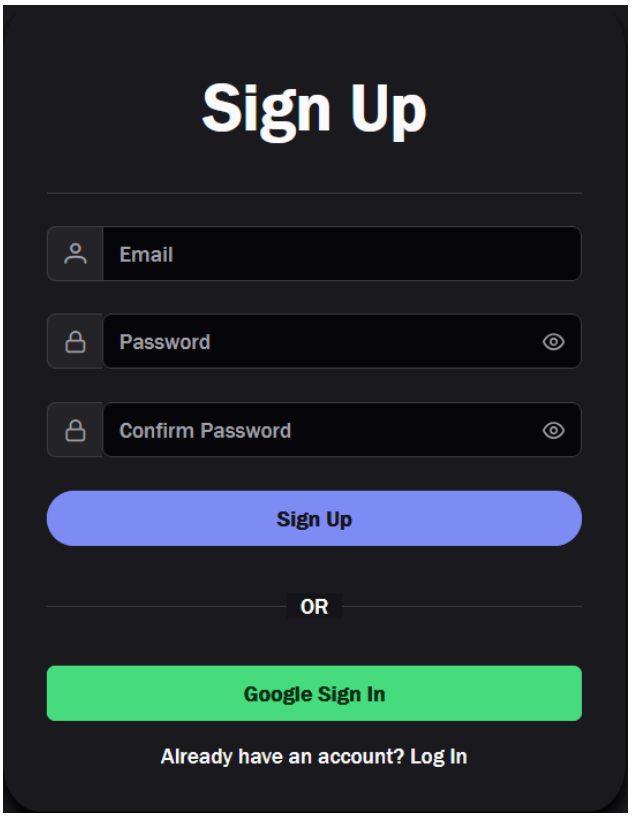
Upon launching the application, you will be greeted with the login screen as shown below:

To access your account, you have the following options:

1. **Email Address and Password:**
    - Enter your registered email address in the "Email Address" field.
    - Enter your password in the "Password" field.
    - Click the "Login" button to proceed.
2. **Google OAuth:**
    - You can sign in using your Google account by clicking the "Google Sign in" button.

If you do not have an account, click on the "Sign up" link at the bottom to create a new account.

Signing Up



If you do not have an account, you can create one by following the steps below:

1. **Email Address:**
    - Enter the email address you wish to use for your account.
2. **Password:**
    - Enter a password that meets the following criteria:
        - At least 8 characters long
        - Contains at least one number
        - Contains at least one uppercase letter
        - Contains at least one lowercase letter
3. **Confirm Password:**
    - Re-enter your password to confirm it matches.
4. **Sign Up:**
    - Click the "Sign up" button to create your account.

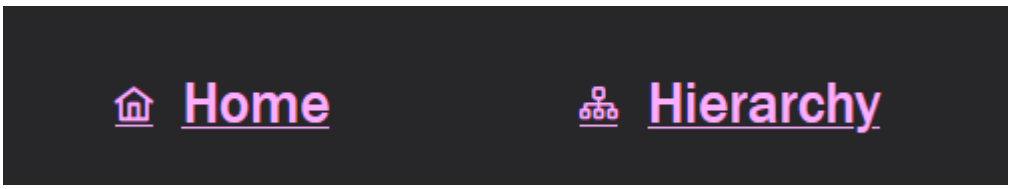
Alternatively, you can sign up using your Google account by clicking the "G" button.

If you already have an account, click the "Login" link at the bottom to return to the login page.

Navigation Bar

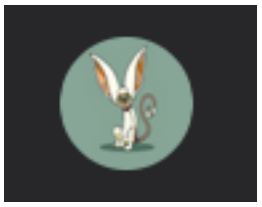
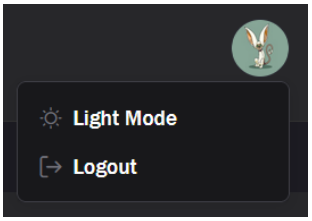
The navigation bar is present on the top of all pages within the application, providing quick access to essential functions and settings. The layout of the navigation bar is as follows:

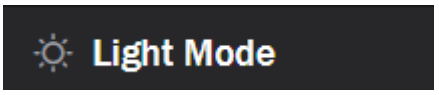
1. **Top Left:**
    - In the top left of the navbar you will see 2 links, these are links to the different pages of the application. Clicking on the name, such as ‘Home’ takes you to the Home page and ‘Hierarchy’ takes you to the Hierarchy page.



2. **Top Right:**
    - In the top right of the navbar you will see an image. This is the user’s profile picture/image. Users can add their own profile

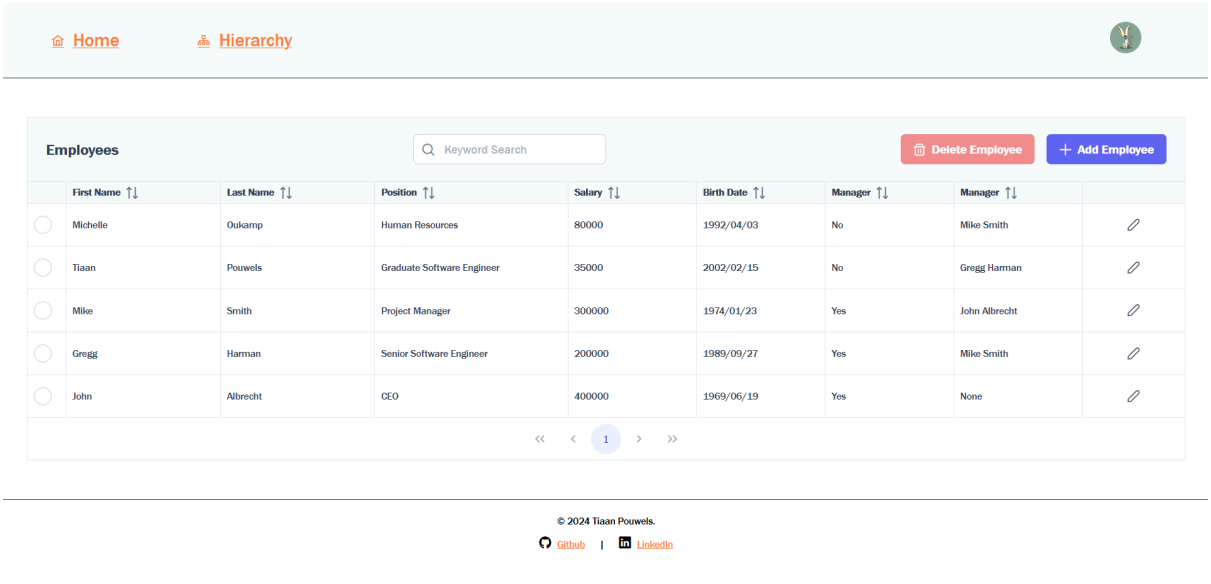
pictures/images on the Gravatar website. Firstly they have to create an account, and can then add their own image which is then linked to their email account and will be displayed in the app. If the user doesn't have a Gravatar account, then Gravatar will give them a default profile picture.

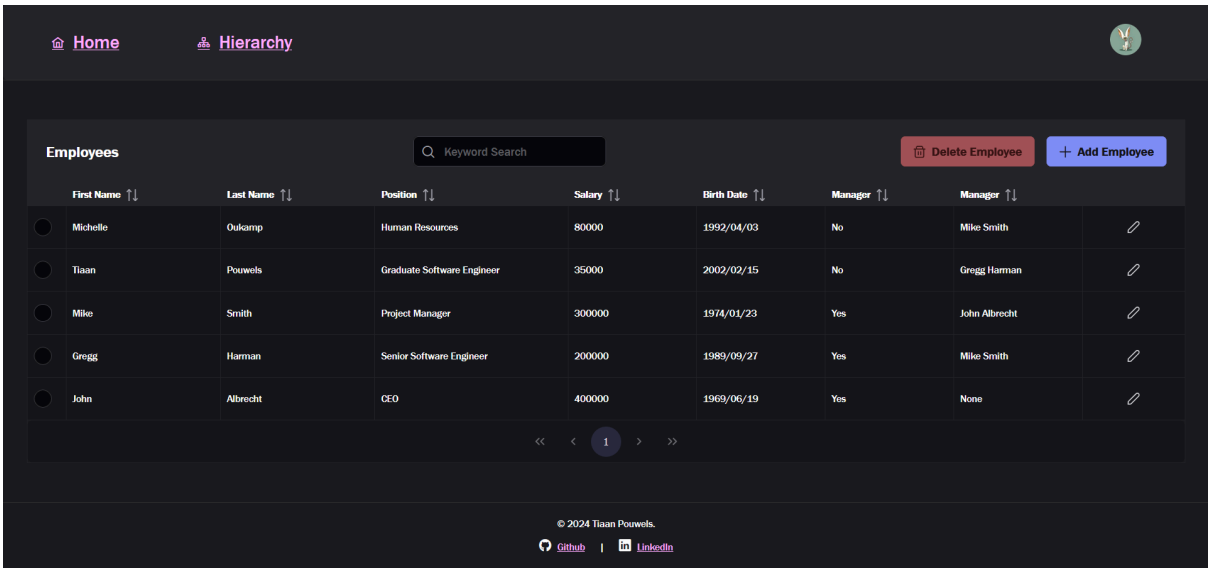
- 
- When clicking on the profile picture/image you will see a dropdown menu open containing a list of 2 options:
  - 

  - **Theme change:** If you click the theme change option then the entire app will change theme (from light to dark and vice-versa)
    1. 

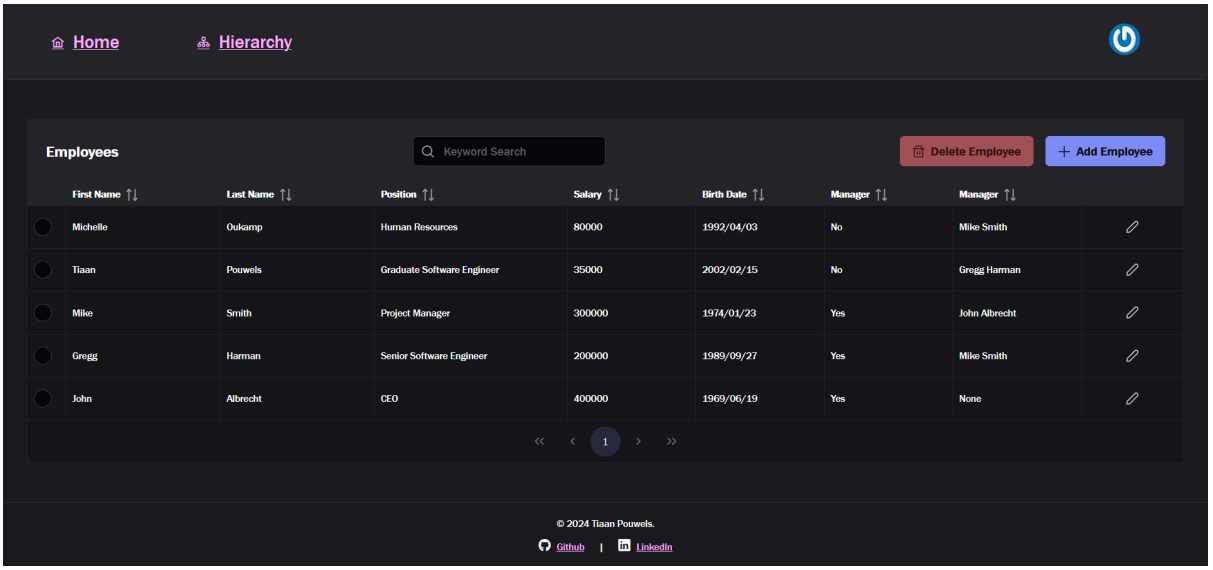
  - **Logout:** If you click the Logout option then your account will be logged out and you will be returned to the login screen.
    1. 

Light vs Dark Theme:


Home
Hierarchy
Employees
Keyword Search
Delete Employee
Add Employee
First Name
Last Name
Position
Salary
Birth Date
Manager
Manager
Michelle Oukamp Human Resources 80000 1992/04/03 No Mike Smith
Tiaan Pouwels Graduate Software Engineer 35000 2002/02/15 No Gregg Harman
Mike Smith Project Manager 300000 1974/01/23 Yes John Albrecht
Gregg Harman Senior Software Engineer 200000 1989/09/27 Yes Mike Smith
John Albrecht CEO 400000 1969/06/19 Yes None
1
© 2024 Tiaan Pouwels.
Github | LinkedIn


Home
Hierarchy
Employees
Keyword Search
Delete Employee
Add Employee
First Name
Last Name
Position
Salary
Birth Date
Manager
Manager
Michelle Oukamp Human Resources 80000 1992/04/03 No Mike Smith
Tiaan Pouwels Graduate Software Engineer 35000 2002/02/15 No Gregg Harman
Mike Smith Project Manager 300000 1974/01/23 Yes John Albrecht
Gregg Harman Senior Software Engineer 200000 1989/09/27 Yes Mike Smith
John Albrecht CEO 400000 1969/06/19 Yes None
1
© 2024 Tiaan Pouwels.
Github | LinkedIn

Home Screen

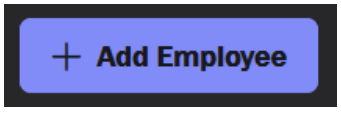


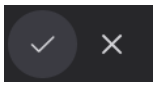
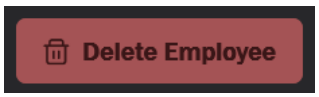
The Home screen is the central hub of the application, offering a comprehensive view of all employees within the company. It provides various functionalities to manage employee data efficiently, including adding, updating, deleting, searching, and sorting employees.

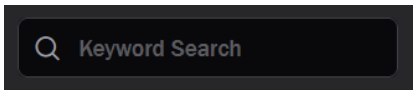
Features and Functionality

- **Add Employees**: Easily add new employees to the system by filling out a form with relevant details.
- **Update Employee Information**: Modify existing employee records to ensure the data remains accurate and up-to-date.
- **Delete Employees**: Remove employees from the system when necessary.
- **Search Employees**: Quickly locate specific employees using the search functionality.
- **Sort Employees**: Organise the employee list based on different criteria for better visibility.

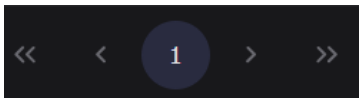
Using the Home Screen

- **Adding Employees**: Click on the "Add Employee" button and complete the required fields. Once submitted, the new employee will be added to the list.
  - 

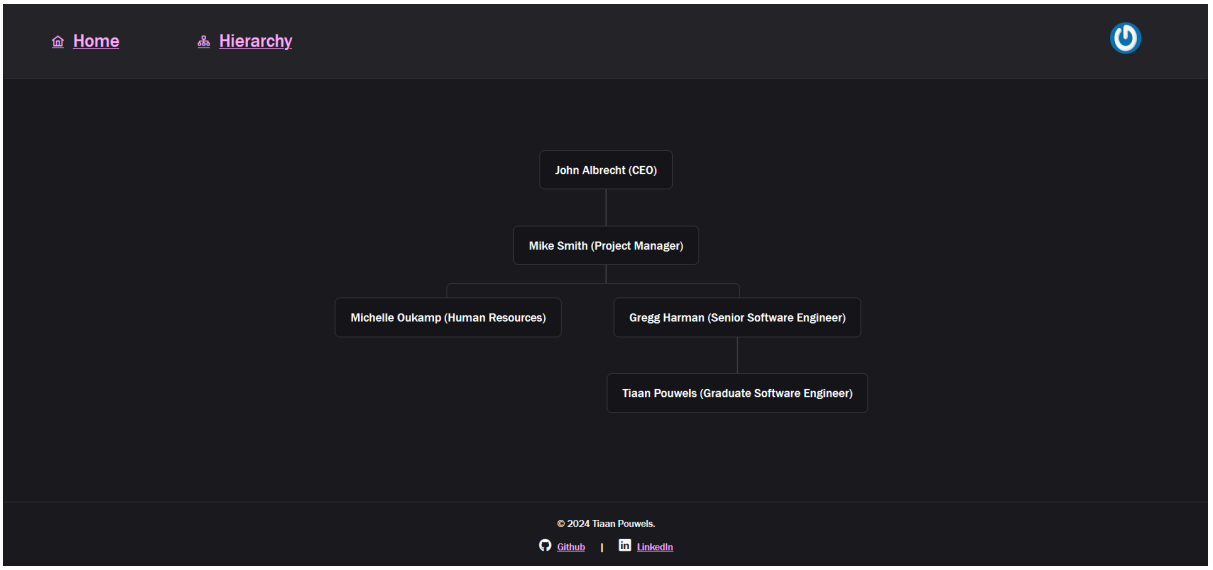
- **Updating Employee Information**: To edit an employee's details, click the pencil icon next to the employee's name. This action will convert the employee's information into editable text boxes and selection fields. After making the necessary changes, click the checkmark icon to save and update the record, or click the X icon to cancel and discard your edits.
  - 
  - 
- **Deleting Employees**: To remove an employee, click the button next to their name, then select the "Delete" option to confirm the action.
  - 
  - 

- **Searching**: Use the search bar at the top of the screen to enter keywords or employee details. The search bar will filter in real-time based on your input.
  - 

- **Sorting**: Click on the column headers in the employee list to sort by different attributes, such as name or position.
  - 

- **Paging:** At the bottom of the table there is a button you can press to see the next set of rows. Or you can skip to the start or end of records via the outer buttons.
  - 

Hierarchy Screen



The Hierarchy screen provides a visual representation of the company's organisational structure. It starts at the top with the CEO and displays the hierarchy down through the various levels of management and staff. This screen allows you to easily see reporting relationships and understand the company's structure at a glance.

Technical Document

Architecture

For this project, I chose to use a client-server architecture as I believe it was the best architecture to efficiently manage the implementation within a limited timeframe. This architecture was selected for the following reasons:

- **Client-Server Model**: This architecture separates the client (frontend) and server (backend) components, allowing each to be developed and optimised independently. This separation simplifies development, deployment and maintenance.
- **Time Constraints**: Given the short period available for implementation, the client-server model was advantageous as it allows for rapid development and deployment. A client-server architecture using firebase as a backend

also significantly simplified the project as the need for developing a custom API was eliminated, which saved a lot of time.
- **Efficiency**: Firebase provided a bunch of pre-built solutions for authentication, database management, and hosting which reduced the complexity of backend development a lot. This allowed me to focus more on the development of the frontend and client-side functionality.

Overall, the client-server architecture was chosen to maximise efficiency and meet the project's timeline requirements effectively.

Design Patterns

For this project, I mainly used two key design patterns, those were the Singleton and Observer patterns. These patterns were selected to enhance simplicity and efficiency while focusing on core functionality.

- **Singleton**: The Singleton pattern was applied to user management. Since only one user can be actively logged in at a time, the Singleton pattern ensured that only one instance of the user was maintained. The singleton's main use was simplifying user state management.
- **Observer**: The Observer pattern was utilised to handle real-time updates for CRUD operations. The Observer pattern observes the database for modifications, thus ensuring that the user interface remains up-to-date with the latest data.

These design patterns were chosen to keep the system straightforward and efficient, ensuring effective management of user state and real-time data synchronisation.

Technologies Used

For this project, the following technologies were utilised:

- **Firebase**: Used as the backend solution. Firebase provides services for everything you could need for your backend, such as authentication, real-time databases, and hosting. Firebase was chosen for its ease of use and ability to effectively handle backend development. I used Firebase for user authentication, such as login and signup, for its data storage, such as firestore db, and for its deployment.

- **Vue(js)**: Used for the frontend. Vue.js offers a reactive and component-based architecture that enhances the development of interactive user interfaces. I used Vue for its ease of use, speed and powerful libraries.
    - **PrimeVue**: It Integrates with Vue.js, providing a bunch t of UI components that enhances the applications user interface. It simplifies the development of consistent and visually appealing components, such as buttons and forms, thus allowing me to focus more on functionality.