**FEEDBACKHUB**

Dissertation

Submitted to

**The Gandhigram Rural Institute – Deemed to be University**

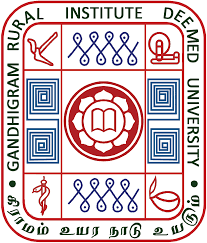
In partial fulfillment of the requirements for the Award of the Degree of

**MASTER OF COMPUTER APPLICATION**

By

**SANGEETHA K**

( Register No : 22322047 )



DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS

The Gandhigram Rural Institute (Deemed To Be University)

Gandhigram - 624 302

Tamil Nadu

India

March 2024

**BONAFIDE CERTIFICATE**

This is to certify that the project titled **“FEEDBACKHUB”** is a bonafide record of work carried out by **SANGEETHA K (22322047)**submitted for Dissertation in partial fulfillment of the requirements for the award of **Master of Computer Applications(MCA)** in The Gandhigram Rural Institute (Deemed to be University), Gandhigram, during the period December 2023 to April 2024.

|  |  |
| --- | --- |
| Dr.N.Senthilkumaran,MCA.,M.Phil.,Ph.D., | Dr.P.Kalavathi,MCA.,M.Phil.,Ph.D., |
| **Project Guide** | **Head of the Department** |

Submitted for the Viva-voce examination held on \_\_\_\_\_\_\_\_\_\_.

|  |  |
| --- | --- |
| **Internal Examiner** | **External Examiner** |

**DECLARATION**

I hereby declare that this project work titled **“FEEDBACKHUB”** is a record of original work done by me under the supervision and guidance of **Dr.N.Senthilkumaran,MCA.,M.Phil.,Ph.D.,** and that this project work has not formed the basis for the award of any Degree / Diploma / Associateship / Fellowship or similar title to any candidate of any other University.

|  |  |
| --- | --- |
| Place : Gandhigram |  |
| Date : 22.03.2024 | **( SANGEETHA K )** |

**ACKNOWLEDGEMENT**

I thank the almighty for giving me divine strength and enabling me to complete this project work successfully.

I heartily thank **Dr.P.Kalavathi.,MCA.,M.Phil.,Ph.D.,** Professor and Head of the Department of Computer Science and Applications, for the facilities given to me for accomplishing this task.

Also I whole heartedly thank my supervisor and guru **Dr.N.Senthilkumaran,MCA.,M.Phil.,Ph.D.,**Assistant Professor, Department of Computer Science and Applications, for giving me a moral support and excellence guidance through the course of the development of this project.

I express my deep appreciation for the help and warm encouragement that I have received from family and my dear friends, because without their support I could not have dreamt of completing this project successfully.

Finally, I would like to thank all those who directly or indirectly helped me in the successful completion of this project work.

**SANGEETHA K**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **NO** | **CONTENTS** | **PAGE.NO** |
| 1. | **INTRODUCTION** | 6 |
|  | * 1. Abstract | 9 |
|  | * 1. Organization Profile | 10 |
| 2. | **SYSTEM ANALYSIS** | 11 |
|  | 2.1. Existing System | 12 |
|  | 2.2. Proposed System | 12 |
|  | 2.3. Feasibility Study | 12 |
| 3. | **SYSTEM REQUIREMENT** | 14 |
|  | 3.1. Hardware Requirement | 15 |
|  | 3.2. Software Requirement | 15 |
|  | 3.3. Software Description | 16 |
| 4. | **SYSTEM DESIGN** | 21 |
|  | 4.1. Flow Diagram | 22 |
| 5. | **SYSTEM DEVELOPMENT** | 23 |
|  | 5.1. Project Description | 24 |
|  | 5.2. Module Description | 25 |
| 6. | **SYSTEM TESTING** | 27 |
| 7. | **SYSTEM IMPLEMENTATION**  **AND MAINTENANCE** | 30 |
| 8. | **CONCLUSION** | 33 |
| 9. | **FUTURE ENHANCEMENTS** | 35 |
|  | **BIBILOGRAPHY** | 37 |
|  | **APPENDIX** | 39 |
|  | Appendix A-Sample Screen shot | 40 |
|  | Appendix B-Sample Reports | 43 |

**CHAPTER-I**

**INTRODUCTION**

1. **INTRODUCTION**

Welcome to the "FEEDBACKHUB" project! The primary goal of this module is to facilitate the performance appraisal process and gather feedback on employees’ work. It aims to provide a comprehensive system for assessing individual performance, setting goals, and fostering continuous improvement.

**Key Features:**

**Performance Reviews:** Enable managers to conduct regular performance reviews for their team members, define customizable performance criteria and rating scales for different job roles, Schedule and automate periodic performance evaluation cycles.

**Goal Setting:** Allow employees and managers to collaboratively set and track performance goals, align individual goals with the overall objectives of the team and the organization.

**Performance Metrics and Analytics:** Capture and analyze key performance indicators (KPIs) to assess individual and team performance.Generate reports and dashboards to visualize performance trends and identify areas for improvement.

**Development Plans:** Facilitate the creation of personalized development plans based on performance reviews and feedback. Recommend training programs or learning resources to address identified skill gaps.

**Feedback Surveys:** Conduct surveys to gather feedback from employees on various aspects of the workplace, team collaboration, and organizational culture. Use survey results to identify areas for improvement and enhance employee engagement.

**Security and Confidentiality:** Ensure data security and confidentiality of performance-related information. Define access controls to restrict information based on roles and responsibilities.

- Encourages continuous feedback and open communication. Provides a structured approach to performance appraisals. Facilitates employee development and career growth. By focusing on these features, the "FeedbackHub" module aims to create a robust and user-friendly system that supports effective performance management within the organization. The module contributes to the overall success of the Employee Management System by ensuring that employees receive constructive feedback, have clear development paths, and are recognized for their contributions.

**1.1 ABSTRACT**

The importance of employee performance appraisal in the context of strategic human resources management, outlining the role of organizational context. Accordingly, the paper presents an employee performance appraisal conceptual framework including the role of contextual factors, where strategic human resources management plays a key role. Moreover, the model includes two sets of procedural and representative basic performance criteria, as well as a new category of profile performance criteria as the foundation for developing flexible, customized and meaningful employee performance appraisal systems and processes aimed at improving individual and organizational overall performance. The primary goal of the Feedbackhub is to facilitate the performance appraisal process and gather feedback on employees’ work. It aims to provide a comprehensive system for assessing individual performance, setting goals, and fostering continuous feedback and open communication, facilitates employee development and career growth and provides a structured approach to performance apprasials, promotes a positive work culture through recognition and rewards.

1.2 **ORGANIZATION PROFILE**

**ZOLABZ TECHNOLOGIES**

**Overview**

Zolabz technologies is a leading IT solutions provider committed to delivering cutting-edge technology services to businesses across various industries. With a passion for technology and a relentless drive for excellence, they’ve been at the forefront of the IT sector for 10 plus years.

Zolabz Tech isa established software company dedicated to providing innovative solutions for businesses. Their mission is to help our clients achieve their goals through technology-driven solutions. Zolabz tech offer custom software development services to meet the unique needs of their clients. Their team is comprised of skilled professionals with extensive experience in software development. They use cutting-edge technologies and agile development methodologies to deliver cost-effective and timely solutions. They specialize in web and mobile app development, as well as enterprise software solutions.

**Mission**

Zolabz Tech mission is to empower businesses with the latest technological advancements, helping them thrive in the digital age. They believes that technology should be an enabler, not a barrier, and we are dedicated to simplifying complex IT challenges to drive growth and success for our clients. Their focus is on providing solutions that enable businesses to operate more efficiently and effectively.

**Innovation**

Zolabz stay ahead of the curve by constantly exploring emerging technologies and industry trends. Our innovative solutions are designed to futureproof your business and give you a competitive edge.

**Services**

* Software Development
* Cybersecurity
* Cloud solutions

**CHAPTER-II**

**SYSTEM ANALYSIS**

1. **SYSTEM ANALYSIS**

**2.1 EXISTING SYSTEM**

Several existing systems and frameworks in Node JS offer solutions for employee management, catering to the diverse needs of employees. Similarly, Profile to build comprehensive employee and admin profiles with fields and permissions. For those seeking custom development, frameworks like React provide robust MVC architectures and packages for efficient Employee profile management system implementation. Providing packages and extensions for managing employee profiles alongside other scholarly activities and resources. These existing systems offer a range of options for management to choose from based on their requirements and technical capabilities, ensuring efficient employee profile management in environments.

**2.2 PROPOSED SYSTEM**

The performance appraisal process refers to the opportunity to exchange information, where the quality of rater – appraised employee relationships is essential for the efficacy of this process. Moreover, Fletcher (2001) states that performance appraisal processes enable companies to develop a variety of activities that allow the development of employee competencies in a quest to

Improve performance.

The “Employee Feedback and performance appraisal” of the organization is developed to overcome the most of the problems occurring in the manual system by computerizing the existing system. The features of the newly proposed computerized system are described in brief as below:After computerizing the system, the owner of the organization or the user of the system canfinish their work in least amount of time and efforts. The computerized systems have many gainsand efforts which the manual system can’t give in any type of situations.In any manual system if we take, the main problem arising is to maintain the number of recordsand finding a particular record

.

* 1. **FEASIBILITY STUDY**

**Technical Feasibility:**

**Economic Feasibility:**

**Cost of Development:** The project utilizes open-source libraries and APIs, reducing development costs. However, there may be expenses related to hosting APIs, email services, and potential hardware requirements.

**Return on Investment (ROI):** The project's ROI can be measured in terms of user satisfaction, efficiency gains, and potential commercialization opportunities if the system proves successful and gains popularity.

**Operational Feasibility:**

**User Acceptance:** The success of the project depends on user acceptance and adoption of voice-controlled personal assistant technology. Conducting user surveys and testing can help gauge user interest and acceptance.

**Maintenance and Support:** The project requires ongoing maintenance and support to address issues, update libraries, and enhance functionalities based on user feedback. Ensuring a sustainable support mechanism is crucial for operational feasibility.

**Schedule Feasibility:**

**Time Constraints:** The project's timeline depends on factors such as the complexity of features, availability of resources, and development expertise. Creating a detailed project plan with milestones and deadlines can help manage schedule feasibility.

**Iterative Development:** Adopting an iterative development approach allows for incremental progress and flexibility in adapting to changing requirements, enhancing schedule feasibility.

**CHAPTER-III**

**SYSTEM REQUIREMENT**

1. **SYSTEM REQUIREMENT**

**3.1 HARDWARE REQUIREMENT**

* System : Pentium IV 2.4 GHz
* Hard Disk : 200 GB
* Mouse : Logitech.
* Keyboard : 110 keys enhanced
* Ram : 4GB

**3.2 SOFTWARE REQUIREMENT**

* O/S : Windows 10.
* Language : Node js
* Front End : REACT JS

**3.3 SOFTWARE DESCRIPTION**

**3.3.1 Flask**

This Flask Tutorial is the latest and comprehensive guide designed for beginners and professionals to learn Python Flask framework, which is one of the most popular Python-based web frameworks. Whether you are a beginner or an experienced developer, this tutorial is specially designed to help you learn and master Flask and build your own real-world web applications. This Flask Tutorial covers a wide range of topics from basic concepts such as setup and installation to advanced concepts like user authentication, database integration, and deployment.

**Advantages of Flask:**

1.Flask is a lightweight backend framework with minimal dependencies.

2.Flask is easy to learn because its simple and intuitive API makes it easy to learn and use for beginners.

3.Flask is a flexible Framework because it allows you to customize and extend the framework to suit your needs easily.

4.Flask can be used with any database like:- SQL and NoSQL and with any Frontend Technology such as React or Angular.

5.Flask is great for small to medium projects that do not require the complexity of a large framework.

**Working principle:**

Flask operates on the principles of the Model-View-Controller (MVC) architecture, although it is often categorized as a micro-framework due to its minimalistic design. The working principle of Flask involves several key components:

**Routing and View Functions:** Flask uses a routing mechanism to map URLs to view functions, which are Python functions that handle HTTP requests and return responses. Developers define routes using decorators such as @app.route('/'), where '/' represents the URL pattern. When a request is made to a specific URL, Flask matches the URL to the corresponding route and invokes the associated view function.

**Request and Response Cycle:** When a client sends an HTTP request to a Flask application, Flask creates a request object that contains information about the request, such as headers, parameters, and form data. The view function processes the request and generates a response object, which contains the data to be sent back to the client. Flask then converts the response object into an HTTP response and sends it back to the client.

**Template Rendering:** Flask integrates with the Jinja2 template engine, allowing developers to create dynamic HTML pages by embedding Python code within HTML templates. Views can render templates using the render\_template() function, passing data to the template for dynamic content generation. Jinja2 templates support template inheritance, filters, and macros, making it easy to build complex web pages.

**Extension Integration:** Flask's flexibility is further enhanced by its extensive ecosystem of extensions, which provide additional functionality such as database integration, user authentication, and API development. Developers can easily integrate extensions into their Flask applications to extend the framework's capabilities and streamline development.

**Middleware and Hooks:** Flask supports the use of middleware and hooks to intercept and modify the request and response objects during the request-response cycle. Middleware functions can perform tasks such as request preprocessing, authentication, logging, and error handling, allowing developers to modularize and customize the application's behavior.

**3.3.2 Python**

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. Created by Guido van Rossum and first released in 1991, Python has grown to become one of the most popular programming languages worldwide. It features a clean and concise syntax that emphasizes code readability and encourages developers to write maintainable and expressive code. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming, making it suitable for a wide range of applications, from web development and data analysis to artificial intelligence and scientific computing. Its extensive standard library and thriving ecosystem of third-party packages contribute to Python's popularity and usability.

**Advantages of Python:**

Simplicity and Readability: Python's clean and concise syntax makes it easy to learn and understand, even for beginners. Its emphasis on readability encourages developers to write clear and expressive code, leading to improved maintainability and collaboration.

Versatility and Flexibility: Python is a versatile language that can be used for a wide variety of tasks, including web development, data analysis, machine learning, artificial intelligence, scientific computing, automation, and more. Its flexibility allows developers to switch between different programming paradigms and use Python for diverse projects.

Large Standard Library: Python comes with a comprehensive standard library that provides ready-to-use modules and functions for common tasks such as file I/O, networking, threading, and data processing. This extensive library reduces the need for external dependencies and accelerates development.

**Thriving Ecosystem:** Python has a vibrant ecosystem of third-party packages and frameworks developed by the open-source community. These packages extend Python's functionality and address specific use cases, allowing developers to leverage pre-built solutions and accelerate development.

**Community Support:** Python has a large and active community of developers, educators, and enthusiasts who contribute to its development, documentation, and support. The Python community is known for its inclusivity, collaboration, and willingness to help newcomers, making it easy to find resources, tutorials, and assistance online.

**Working Principle of Python:**

Python follows a sequential and imperative programming model, where instructions are executed one after another in the order they appear in the source code. The working principle of Python involves several key concepts:

**Interpretation:** Python is an interpreted language, meaning that source code is executed line by line by the Python interpreter. This differs from compiled languages, where source code is translated into machine code before execution.

**Dynamic Typing:** Python is dynamically typed, allowing variables to be assigned values of different types without specifying their data types explicitly. This dynamic typing simplifies coding and promotes code flexibility but requires careful attention to type handling to avoid runtime errors.

**Indentation:** Python uses indentation to denote blocks of code, such as loops, conditional statements, and function definitions. Indentation is critical for code readability and structure in Python and serves as an alternative to explicit block delimiters like braces {} in other languages.

**Object-Oriented Programming:** Python supports object-oriented programming (OOP) principles such as encapsulation, inheritance, and polymorphism. Classes and objects are fundamental concepts in Python, allowing developers to model real-world entities and build modular, reusable code.

**Exception Handling:** Python provides robust support for exception handling, allowing developers to anticipate and handle errors gracefully. Exception handling helps improve code reliability and resilience by handling unexpected situations and preventing program crashes.

**Python Libraries:**

**speech\_recognition:**

This library provides functions for performing speech recognition, allowing the program to transcribe spoken words into text. It supports various speech recognition engines and APIs, making it versatile for different applications.

**pyttsx3:**

pyttsx3 is a text-to-speech conversion library. It allows the program to convert text into synthesized speech, enabling the assistant to respond to user commands and queries audibly.

**pywhatkit:**

pywhatkit is a Python library that provides various functionalities related to web scraping and automation. In the SARA project, it's used for tasks like playing music on YouTube in response to user commands.

**datetime:**

The datetime library provides functions for working with dates and times. It allows the program to retrieve and manipulate current date and time information, enabling functionalities such as displaying the current time.

**wikipedia:**

This library provides functions for interacting with the Wikipedia API. It allows the program to fetch information from Wikipedia based on user queries, providing informative responses to user questions.

**pyjokes:**

pyjokes is a library that provides functions for generating random jokes. It adds a humorous element to the assistant's responses, enhancing the user experience with entertainment value.

**requests:**

requests is a popular library for making HTTP requests in Python. In the SARA project, it's used for interacting with external APIs, such as fetching weather data from the OpenWeatherMap API.

**json:**

The json library provides functions for working with JSON data. It allows the program to parse and manipulate JSON-formatted responses from external APIs, extracting relevant information for further processing.

**sys:**

sys is a built-in Python module that provides functions and variables related to the Python interpreter and system environment. In the SARA project, it's used for system-specific functionalities, such as exiting the program.

**TextBlob:**

TextBlob is a natural language processing library that provides functions for text processing tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, and translation. In the SARA project, it's used for sentiment analysis to assess the emotional tone of user commands.

**smtplib and ssl:**

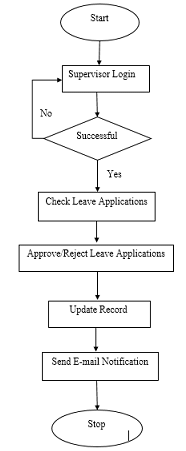
These libraries are used for sending email notifications. smtplib provides functions for sending emails via the Simple Mail Transfer Protocol (SMTP), while ssl is used for creating secure connections. Together, they enable the program to notify parents about their child's emotional state via email.

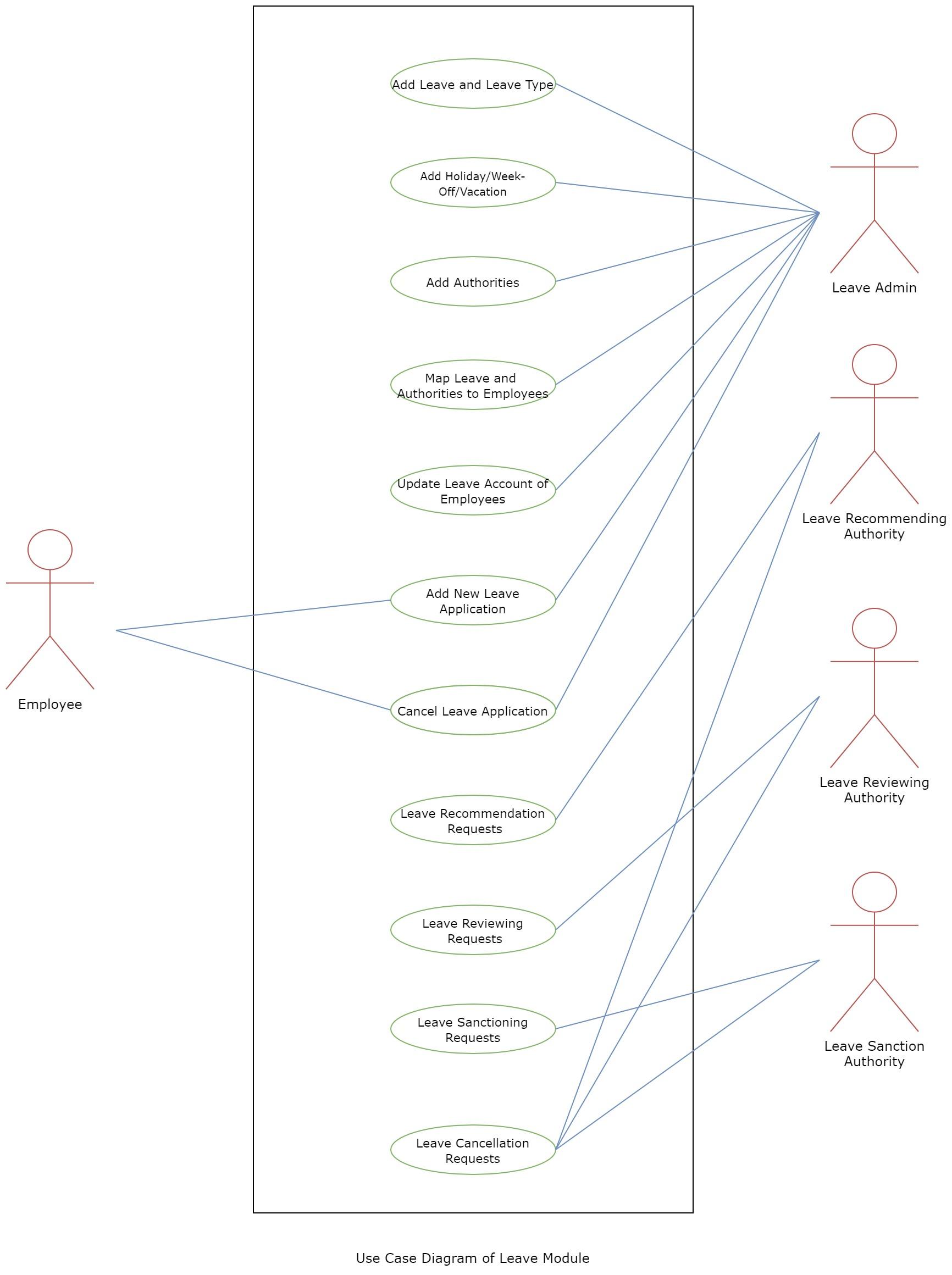
**CHAPTER-IV**

**SYSTEM DESIGN**

**4. SYSTEM DESIGN**

**4.1 FLOW DIAGRAM**





**CHAPTER-V**

**SYSTEM DEVELOPMENT**

**5. SYSTEM DEVELOPMENT**

**5.1 PROJECT DISCRIPTION**

The Speech Analysis and Recognition Assistant (SARA) is a voice-controlled personal assistant implemented in Python. SARA offers a range of functionalities, including speech recognition, natural language understanding, task execution, sentiment analysis, and email notifications. Users interact with SARA by speaking commands aloud, which are then transcribed and processed to perform various tasks such as playing music, fetching information from Wikipedia, providing weather updates, and generating jokes. Additionally, SARA analyzes the sentiment of user commands to assess emotional tone and sends email notifications to parents about their child's emotional state based on the sentiment analysis results. Leveraging libraries and APIs such as speech\_recognition, pyttsx3, pywhatkit, datetime, wikipedia, pyjokes, requests, TextBlob, smtplib, and ssl, SARA offers a versatile and intuitive interaction experience, enhancing productivity, accessibility, and communication through voice-enabled interactions.

**Features:**

**Voice-Controlled Interaction:** Users can interact with SARA through voice commands, providing a hands-free and intuitive user experience.

**Speech Recognition:** SARA transcribes spoken commands into text using the speech recognition library, enabling seamless communication with the system.

**Natural Language Understanding:** The system processes user commands to understand their intent and execute corresponding tasks, such as playing music, fetching information, or providing weather updates.

**Task Execution:** SARA performs various tasks based on user commands, including playing music on YouTube, fetching information from Wikipedia, telling jokes, and providing weather updates.

**Sentiment Analysis:** The system analyzes the sentiment of user commands to assess emotional tone, allowing for personalized responses and notifications.

**Email Notifications:** SARA sends email notifications to parents about their child's emotional state, helping them stay informed and connected.

**External API Integration:** The system integrates with external APIs such as Wikipedia and OpenWeatherMap to fetch information and provide real-time updates, enhancing its capabilities and usefulness.

**Text-to-Speech Output:** SARA responds to users with synthesized speech using the pyttsx3 library, providing audible feedback and enhancing the interaction experience.

**Advantages:**

**Convenience:** SARA offers a convenient way for users to access information, entertainment, and assistance through voice-enabled interactions, reducing the need for manual input methods.

**Efficiency:** By leveraging speech recognition and natural language understanding, SARA streamlines tasks and processes, improving overall efficiency and productivity.

**Accessibility:** The voice-controlled nature of SARA makes it accessible to users with disabilities or special needs, providing an inclusive interaction experience.

**Personalization:** SARA's sentiment analysis feature allows for personalized responses and notifications based on the emotional tone of user commands, enhancing the user experience.

**Connectivity:** With email notifications to parents, SARA promotes communication and connectivity, enabling parents to stay informed about their child's emotional state.

**Versatility:** The integration of various libraries and APIs in SARA enables it to perform a wide range of tasks, making it a versatile and adaptable solution for different use cases.

**5.2 MODULE DESCRIPTION**

**Speech Recognition Module:**

Responsible for transcribing spoken commands into text using the speech\_recognition library.

Converts audio input from the user's microphone into textual form for further processing.

**Natural Language Understanding Module:**

Processes the transcribed text to understand the user's intent and extract relevant information.

Uses techniques such as parsing, keyword extraction, and context analysis to interpret user commands accurately.

**Task Execution Module:**

Executes various tasks based on user commands, such as playing music, fetching information, providing weather updates, and generating jokes.

Utilizes libraries and APIs such as pywhatkit, datetime, wikipedia, pyjokes, and requests for task execution.

**Sentiment Analysis Module:**

Analyzes the sentiment of user commands to assess emotional tone and mood.

Utilizes the TextBlob library for sentiment analysis, providing insights into the emotional state conveyed by user input.

**Email Notification Module:**

Sends email notifications to parents about their child's emotional state based on sentiment analysis results.

Utilizes the smtplib and ssl libraries for sending email notifications securely.

**Text-to-Speech Module:**

Converts textual responses into synthesized speech using the pyttsx3 library.

Provides audible feedback to users, enhancing the interaction experience with the assistant.

**External API Integration Module:**

Integrates with external APIs such as Wikipedia and OpenWeatherMap for fetching information and providing real-time updates.

Utilizes the requests library for making HTTP requests and processing API responses in JSON format.

**Error Handling Module:**

Implements robust error handling mechanisms to address potential issues such as network failures, API errors, and speech recognition inaccuracies.

Ensures system reliability and resilience by handling exceptions and errors gracefully.

**User Interface Module:**

Provides a user-friendly interface for interacting with the assistant, facilitating seamless communication and task execution.

Enables users to input commands through voice interactions and receive feedback in both textual and audible forms.

**Main Loop Module:**

Orchestrates the overall flow of the program by continuously running the assistant and handling user interactions.

Drives the interaction loop, prompting users for commands, executing tasks, and providing feedback or notifications as needed.

**CHAPTER-VI**

**SYSTEM TESTING**

1. **SYSTEM TESTING**

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system.  System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved.  The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test.  A series of tests are performed before the system is ready for the user acceptance testing.  Any engineered product can be tested in one of the following ways.  Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational.  Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

**1.1 Unit Testing:**

Unit testing is the testing of each module and the integration of the overall system is done.  Unit testing becomes verification efforts on the smallest unit of software design in the module.  This is also known as ‘module testing’.  The modules of the system are tested separately.  This testing is carried out during the programming itself.  In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module.  There are some validation checks for the fields.

**1.2Integration Testing:**

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function.  Integrated testing is systematic testing that can be done with sample data.  The need for the integrated test is to find the overall system performance.

* 1. **White Box Testing:**

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases.  Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

* 1. **Black Box Testing:**

Black box testing is done to find incorrect or missing function

* Interface error Errors in external database access
* Performance errors
* Initialization and termination errors

In ‘functional testing’, is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called ‘black box testing’.  It tests the external behavior of the system.  Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

**1.5 Validation Testing:**

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

* 1. **User Acceptance Testing:**

User acceptance of the system is the key factor for the success of the system.  The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

* 1. **Output Testing**

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format.  The output displayed or generated by the system under consideration.  Here the output format is considered in two ways.  One is screen and the other is printed format.  The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs.  For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

**CHAPTER-VII**

**SYSTEM IMPLEMENTATION & MAINTENANCE**

**7. SYSTEM IMPLEMENTATION & MAINTENANCE**

**7.1IMPLEMENTATION:**

During the implementation phase, the focus is on building the SARA project according

to predefined requirements and objectives.

**Planning:** Define project requirements, objectives, and milestones. Determine the scope of functionalities and identify necessary resources.

**Development:** Write code to implement various modules and functionalities described in the project. Ensure proper integration of libraries and APIs for speech recognition, natural language understanding, task execution, etc.

**Testing:** Conduct thorough testing of implemented functionalities to identify and fix bugs, ensure proper functionality, and validate user interactions.

**Deployment:** Deploy the SARA project on appropriate platforms or devices. Ensure compatibility with different operating systems and environments.

**User Training:** Conduct user training sessions to familiarize users with features and usage of the SARA assistant.

**7.2 MAINTENANCE**

After deployment, the focus shifts to maintaining the SARA project to ensure its continued functionality and effectiveness.

**Monitoring:** Continuously monitor the performance and behavior of the SARA assistant. Identify and address any issues or anomalies.

**Updates and Enhancements:** Regularly update and enhance the SARA project to incorporate new features, improve functionality, and address user feedback.

**Security Management:** Implement security measures to protect user data and ensure the integrity of the system.

**Scalability:** Ensure the scalability of the SARA project to accommodate growing user demands and evolving requirements.

**Backup and Recovery:** Implement backup and recovery mechanisms to safeguard data and ensure business continuity.

**User Support:** Provide ongoing user support and assistance to address inquiries, issues, and concerns.

**Documentation Maintenance:** Keep project documentation up-to-date with the latest changes, updates, and enhancements.

**CHAPTER-VIII**

**CONCLUSION**

**8.CONCLUSION**

The software will manage the working of the modules efficiently. The interconnectivity among modules will reduce the time taken to perform different operational task. The software help to gather the basic information of employee automatically, it helps both the employee and the management. The system is capable of storing all the details of from the time employee and manager have joined the department and also maintain their details in a dynamic order thereby eliminating the paper work and reducing the staff required.

**CHAPTER-IX**

**FUTURE ENHANCEMENT**

**9. FUTURE ENHANCEMENT**

This system being web based and an undertaking of cyber security division needs to be thoroughly tested to find out any security gaps. A console for the data centre may be made available to allow the personnel to monitor on the site which cleared for hosting during a particular period. Moreover, it is just a beginning, further the system may be utilized in various other types of auditing operation viz network auditing or similar process workflow based application.

**CHAPTER-X**

**BIBLIOGRAPHY**

**10. BIBLIOGRAPHY**

**REFERENCE WEBSITE**

[1] R. D. Stanciu, “Performance management – a strategic tool”,FAIMA Business & Management Journal, vol. 3, no. 2, pp. 5-12,

2015.

[2] M.-H. Budworth, and S. L. Mann, “Performance management: Where do we go from here?”, Human resources management Review, vol. 22, no. 2, pp. 81-84, 2011.

[3] P. E. Levy, and J. R. Williams, “The social context of performance appraisal: a review and framework for the future”, Journal of Management, vol. 30, no. 6, pp. 881-905, 2004.

[4] D. L. Caruth, and J. H. Humphreys, “Performance appraisal: essential characteristics for strategic control”, Measuring Business Excellence, vol. 12, no.3, pp. 24-32, 2008.

[5] E. D. Pulakos, Performance Management: A New Approach for Driving Business Results. Hong-Kong: Wiley-Blackwell, 2009. ANNALS OF THE UNIVERSITY OF ORADEA Fascicle of Management and Technological Engineering ISSUE #2, AUGUST 2016, <http://www.imtuoradea.ro/auo.fmte/>57.

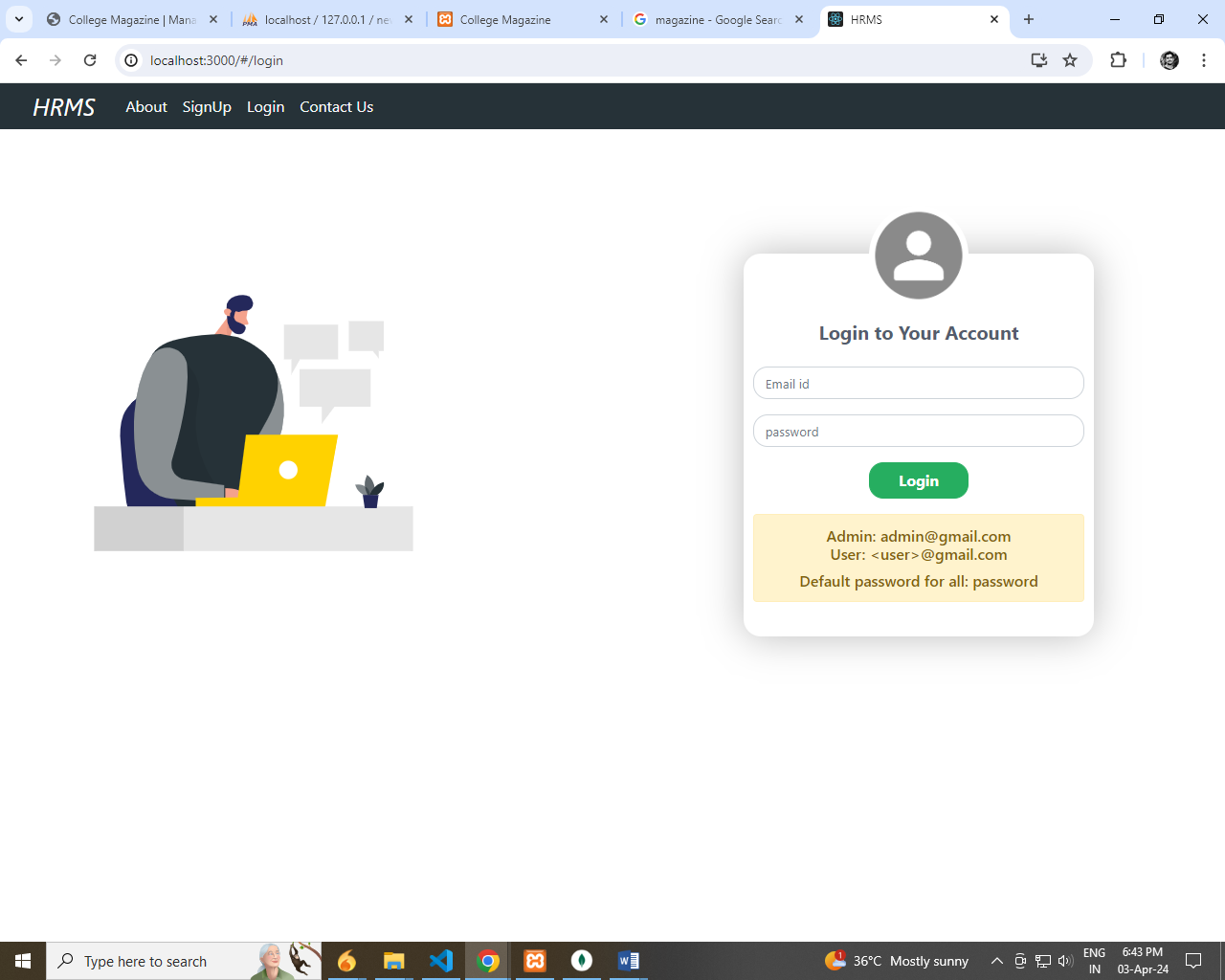
[6] E. D. Pulakos, Elaine D.; R. Mueller Hanson, S. Arad, and N.Moye, “Performance management can be fixed: An on-the-job experiential learning approach for complex behavior change”, Industrial and Organizational Psychology, vol. 8, no. 1, pp. 51 – 76, 2015.

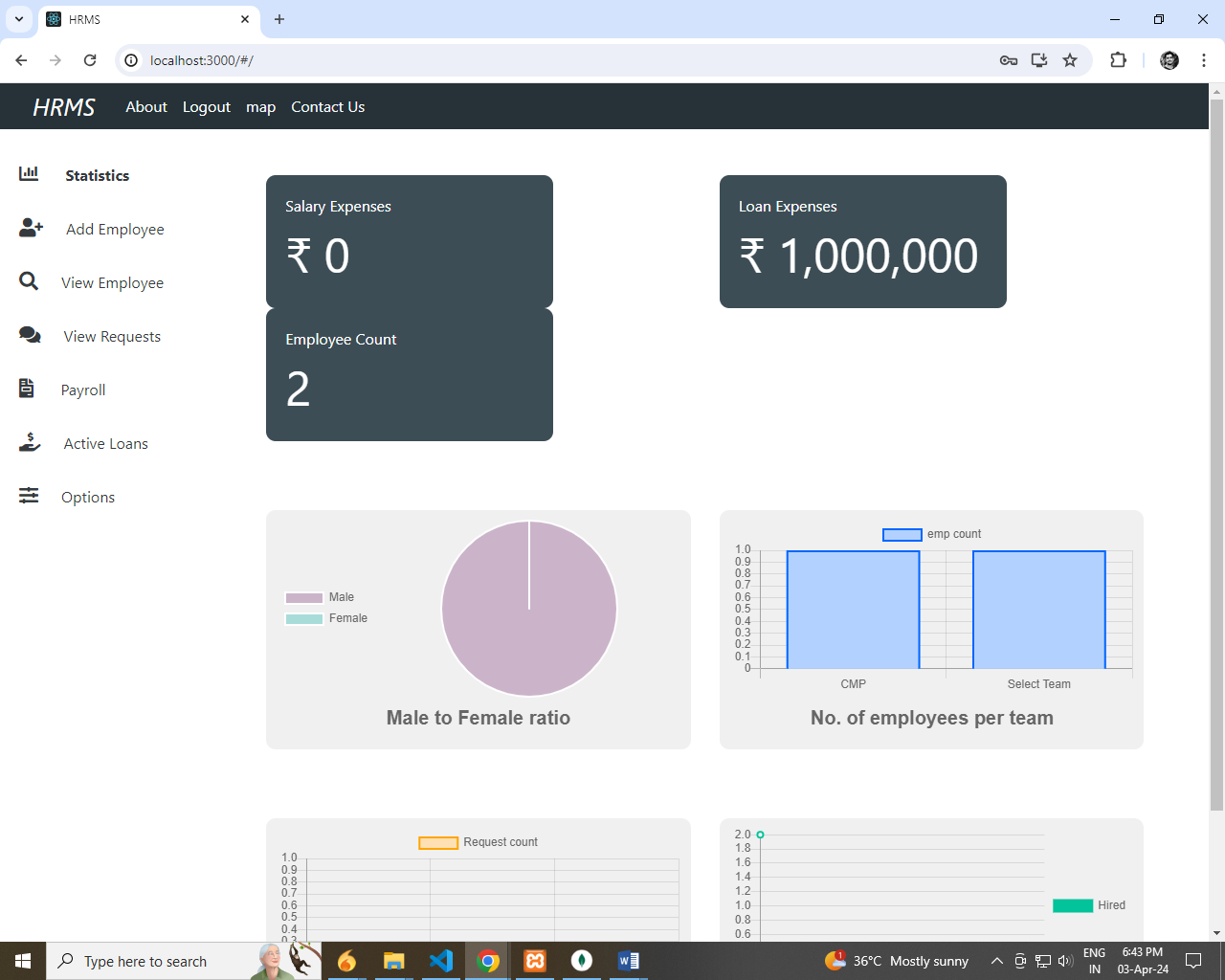
[7] P. E. Levy, S. B. Silverman, and C. M. Cavanaugh, “The performance management fix is in: how practice can build on the research”, Industrial and Organizational Psychology, vol. 8, no.1, pp. 80-85, 2015.

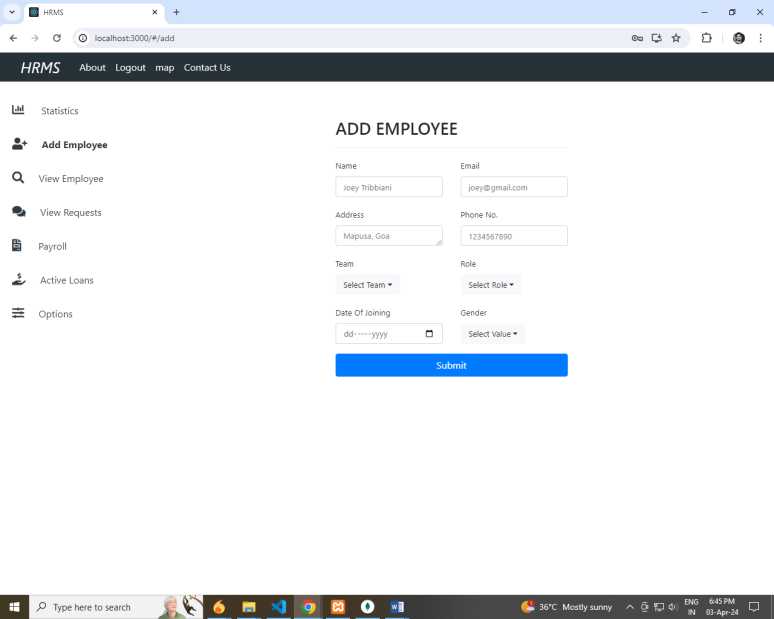
[8] D. Skinner and R. H. Searle, “Trust in the context of performance appraisal”, in Trust and Human resources management, R.H. Searle and

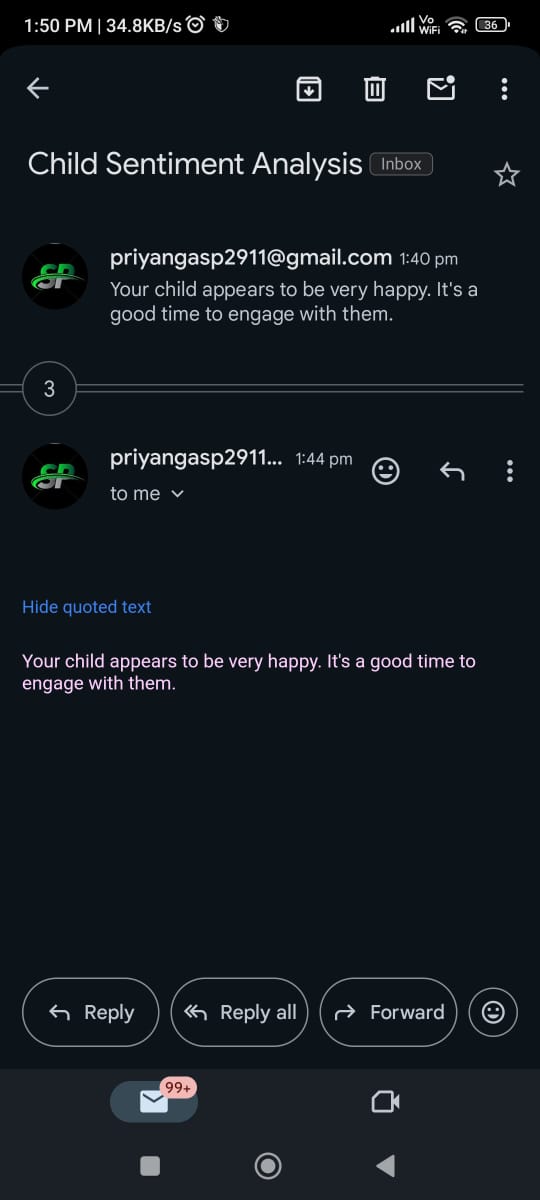
**APPENDIX**

**APPENDIX A - SAMPLE SCREENSHOT**

**LOGIN PAGE**



****



**Happy conversation**

**APPENDIX B-SOURCE CODE**

**App.js**

import React, { Component } from "react";

import { Provider } from "./context";

import { HashRouter as Router, Route, Switch } from "react-router-dom";

import Login from "./components/auth/Login";

import Signup from "./components/auth/Signup";

import ContactUs from "./components/layouts/ContactUs";

import PageNotFound from "./components/layouts/PageNotFound";

import About from "./components/layouts/About";

import "./App.css";

import Header from "./components/layouts/Header";

import Profile from "./components/layouts/Profile";

import EmpDashboard from "./components/layouts/EmpDashboard";

import Attendence from "./components/layouts/Employee/Attendence";

import ViewRequests from "./components/layouts/Admin/View Requests/ViewRequests";

import MyRequests from "./components/layouts/Employee/MyRequests";

import OtherRequests from "./components/layouts/Employee/OtherRequests";

import ViewEmployees from "./components/layouts/Admin/ViewEmployees";

import EditEmpProfile from "./components/layouts/Admin/EditEmpProfile";

import AddEmployee from "./components/layouts/AddEmployee";

import MySalDetails from "./components/layouts/Employee/MySalDetails";

import Payroll from "./components/layouts/Admin/Payroll";

import Statistics from "./components/layouts/Admin/Stats/Statistics";

import Options from "./components/layouts/Admin/Options";

import ViewSingleRequest from "./components/layouts/Employee/ViewSingleRequest";

import CompanyInfo from "./components/layouts/Employee/CompanyInfo";

import ActiveLoans from "./components/layouts/Admin/ActiveLoans";

export default class App extends Component {

render() {

return (

<Provider>

<Router>

<div>

<Header branding="HRMS" />

<Switch>

{/\* general \*/}

<Route exact path="/login" component={Login} />

<Route exact path="/signup" component={Signup} />

<Route exact path="/contactus" component={ContactUs} />

<Route exact path="/about" component={About} /> {/\* emp related \*/}

<Route exact path="/profile" component={Profile} />

<Route exact path="/attendence" component={Attendence} />

<Route exact path="/myRequests" component={MyRequests} />

<Route exact path="/empDashboard" component={EmpDashboard} />

<Route exact path="/otherRequest" component={OtherRequests} />

<Route exact path="/mySalDetails" component={MySalDetails} />

<Route exact path="/companyInfo" component={CompanyInfo} />

<Route

exact

path="/viewSingleRequest/:title/:reqId"

component={ViewSingleRequest />

{/\* admin related \*/}

<Route exact path="/" component={Statistics} />

<Route exact path="/add" component={AddEmployee} />

<Route exact path="/viewRequests" component={ViewRequests} />

<Route exact path="/statistics" component={Statistics} />

<Route exact path="/options" component={Options} />

<Route exact path="/payroll" component={Payroll} />

<Route exact path="/viewEmployees" component={ViewEmployees} />

<Route exact path="/activeLoans" component={ActiveLoans} />

<Route

exact

path="/editEmpProfile/:id"

component={EditEmpProfile}

/>

<Route component={PageNotFound} />

</Switch>

</div>

</Router>

</Provider**> );**

**}**

**}**

Addemployee.js

import React, { Component } from "react";

import { Consumer } from "../../context";

import { Redirect } from "react-router-dom";

import axios from "axios";

import { Spring } from "react-spring/renderprops";

import "../../assets/add-emp/addEmp.css";

import AdminSidePanel from "./Admin/AdminSidePanel";

import toast from "toasted-notes";

import "toasted-notes/src/styles.css";

class AddEmployee extends Component {

constructor() {

super();

this.state = {

email: "",

name: "",

address: "",

phoneNo: "",

role: "Select Role",

team: "Select Team",

gender: "Select Value",

doj: "",

disabled: false,

// error

error: "",

// teams and roels

teamList: [],

roleList: [],

};

}

componentDidMount = async () => {

const teamAndRoleList = await axios.get("/api/admin/getTeamsAndRoles");

console.log(teamAndRoleList.data[0]);

this.setState({

teamList: teamAndRoleList.data[0].teamnames,

roleList: teamAndRoleList.data[0].rolenames,

});

};

onSelectGender = (gender) => this.setState({ gender });

onTeamSelect = (team) => this.setState({ team });

onRoleSelect = (role) => this.setState({ role });

onSubmit = async (dispatch, e) => {

e.preventDefault();

// disable signup btn

this.setState({

disabled: true,

});

const {

email,

name,

address,

phoneNo,

role,

team,

doj,

gender,

} = this.state;

try {

const newUser = await axios.post("/api/admin/addEmployee", {

email,

name,

address,

gender,

phoneNo,

role,

team,

doj,

});

toast.notify("Added new employee", {

position: "top-right",

});

console.log("created acc successfully: ", newUser.data);

this.props.history.push(`/editEmpProfile/${newUser.data.\_id}`);

} catch (err) {

// enable signup btn

this.setState({

disabled: false,

});

console.log("ERROR: ", err.response.data.msg);

this.setState({ error: err.response.data.msg });

}

};

onChange = (e) => this.setState({ [e.target.name]: e.target.value });

render() {

return (

<Consumer>

{(value) => {

let { user, dispatch, token } = value;

if (token === undefined) token = "";

if (token) {

if (user.role !== "admin") return <Redirect to="/" />;

return (

<Spring

// from={{ opacity: 0 }}

// to={{ opacity: 1 }}

// config={{ duration: 300 }}

from={{

transform: "translate3d(1000px,0,0) ",

}}

to={{

transform: "translate3d(0px,0,0) ",

}}

config={{ friction: 20 }}

>

{(props) => (

<>

<div className="row m-0">

{/\* left part \*/}

<div className="col-2 p-0 leftPart">

<AdminSidePanel />

</div>

{/\* right part \*/}

<div

className="col"

style={{

display: "flex ",

flexDirection: "row",

justifyContent: "center",

}}

>

<div style={props}>

{this.state.error ? (

<div className="alert alert-danger my-3">

{this.state.error}

</div>

) : null}

<form

className="addEmpForm"

onSubmit={this.onSubmit.bind(this, dispatch)}

>

<h3 className="">ADD EMPLOYEE</h3>

<hr />

<div className="row">

<div className="col">

{/\* name \*/}

<label htmlFor="name">Name</label>

<input

type="text"

name="name"

className="form-control"

placeholder="Joey Tribbiani"

onChange={this.onChange}

required

/>

</div>

<div className="col">

{/\* email \*/}

<label htmlFor="email">Email</label>

<input

type="email"

name="email"

className="form-control mb-3 "

placeholder="joey@gmail.com"

onChange={this.onChange}

required

/>

</div>

</div>

<div className="row">

<div className="col">

{/\* address \*/}

<label htmlFor="address">Address</label>

<textarea

name="address"

id="address"

// cols="20"

rows="1"

className="form-control mb-3 "

placeholder="Mapusa, Goa"

onChange={this.onChange}

required

/>

</div>

<div className="col">

{/\* phone no \*/}

<label htmlFor="phoneNo">Phone No.</label>

<input

type="number"

name="phoneNo"

className="form-control mb-3 "

placeholder="1234567890"

onChange={this.onChange}

required

/>

</div>

</div>

<div className="row">

{/\* team \*/}

<div className="col">

<label htmlFor="team">Team</label>

<div className="dropdown">

<button

className="btn btn-light dropdown-toggle"

type="button"

id="dropdownMenuButton"

data-toggle="dropdown"

aria-haspopup="true"

aria-expanded="false"

>

{this.state.team}

</button>

<div className="dropdown-menu “aria-labelledby="dropdownMenuButton">

{this.state.teamList.map((teamname) => (

<li style={{ cursor: "pointer" }} key={teamname}

className="dropdown-item"

onClick={() => this.onTeamSelect(teamname)

}>

{teamname}

</li>

))}

</div>

</div>

</div>

{/\* role \*/}

<div className="col">

<label htmlFor="role">Role</label>

<div className="dropdown mb-3">

<button

className="btn btn-light dropdown-toggle" type="button"

id="dropdownMenuButton” data-toggle="dropdown"

aria-haspopup="true” aria-expanded="false">

{this.state.role}

</button>

<div className="dropdown-menu” aria-labelledby="dropdownMenuButton>

{this.state.roleList.map((rolename) => (

<li style={{ cursor: "pointer" }}

key={rolename}

className="dropdown-item"

onClick={() =>

this.onRoleSelect(rolename)>

{rolename>))}

</div>

</div>

</div>

</div>

<div className="row">

{/\* doj \*/}

<div className="col">

<label htmlFor="doj">Date Of Joining</label>

<input type="date" name="doj"className="form-control mb-3 "

placeholder="doj" onChange={this.onChange}

required

{/\* gender \*/}

<div className="col">

<label>Gender</label>

<div className="dropdown">

<button className="btn btn-light dropdown-top type="button"

id="dropdownMenu” data-toggle="dropdown"

aria-haspopup="true" aria-expanded="false">

{this.state.gender}

</button>

<div

className="dropdown-menu"

aria-labelledby="dropdownMenuButton”>

</Spring>

);

} else {

return <Redirect to="/login" />;

}

}}

</Consumer>

);

}

}

export default AddEmployee;