CHAPTER I: INTRODUCTION

1.1 Background

In today’s fast-changing business environment, organizations face growing challenges in managing their workforce effectively. Human Resource (HR) teams are responsible not only for maintaining employee records but also for supporting decisions that affect hiring, retention, and overall employee satisfaction. Traditional HR systems often handle only administrative tasks and miss the opportunity to use data to predict important workforce trends.

One key concern for many companies is employee churn, which refers to the rate at which employees leave an organization. High churn can lead to increased recruitment costs, reduced productivity, and the loss of experienced talent. Identifying patterns behind employee departures and predicting who might leave allows organizations to take timely action to improve retention and maintain team stability.

To meet modern HR demands, there is a growing need for systems that combine employee data management with predictive analytics, role-based access, and real-time insights. Advanced analytics in HR, although still in its early stages, has shown significant potential in improving attrition predictions and identifying key retention factors (Dutta et al., 2024). This approach helps businesses make better decisions, reduce turnover, and boost overall performance.

1.2 Problem Statement

Many organizations find it hard to manage employee data and adapt to workforce changes. HR teams often lack sophisticated tools for quick data analysis, and the inability to predict employee turnover leads to increased costs and disruptions (Alabi et al., 2024). Manual leave and record management systems also contribute to delays and errors (Gaur, 2020).

This project addresses three key challenges:

1. HR teams struggle to analyze and visualize employee data.
2. Leave and employee records are managed manually.
3. There is no system to predict which employees may leave.

1.3 Objectives

This project aims to address key challenges faced by HR departments through clear and focused objectives. Specifically, the objectives are:

1. To develop an integrated HRM system for managing employee information.
2. To create a machine learning system to predict employee churn.
3. To automate leave management through a web-based application.

1.4 Literature Review

The Literature Review section of this report examines existing research, tools, and systems related to HR management and employee attrition prediction. Its objective is to understand current methodologies, identify gaps in traditional systems, and justify the use of machine learning for proactive churn management. This background helps position the developed system within the context of ongoing technological advancements in HR analytics.

1.4.1 Overview of Existing Systems

Employee Analytics and Churn Prediction Systems are important tools for modern human resource management. Popular software like OrangeHRM and BambooHR help organizations track employee data and show basic trends in staff turnover. However, these systems usually do not provide advanced prediction of which employees might leave.

Larger platforms such as Workday and SAP SuccessFactors use more advanced methods, including some machine learning, to estimate the risk of employees leaving. Still, these features often need extra setup and are not always easy to use for every organization.

Most of these systems mainly use basic HR data, such as employee age or years of service, and do not include information from emails or other unstructured sources. They also do not give real-time predictions and sometimes make it hard for managers to understand why an employee is at risk of leaving.

In summary, while current systems help organizations watch employee turnover, their ability to accurately predict and explain employee churn is still limited. More advanced methods from academic research are not yet widely used in these commercial tools.

1.4.2 Technologies used in Similar Systems

Similar HR systems use a mix of cloud-based architectures, REST APIs, and technologies like PHP, Java, JavaScript, and proprietary platforms to enable modular, scalable, and mobile-friendly HR solutions.

Table 1.1: Technologies Used in Similar Systems

Product	Main Technologies Used	Notes
OrangeHRM	PHP, MySQL, Apache, Vue.js	Open-source, web-based
BambooHR	JavaScript (React), Python, AWS	Cloud-based, modern web stack
Workday	Proprietary cloud tech, APIs, AI tools	Enterprise, cloud-only
SAP SuccessFactors	SAP Cloud, SAP HANA, APIs, AI/ML	Enterprise, SAP ecosystem

1.4.3 Theoretical Background

In academic and technical writing, the theoretical framework is essentially the set of ideas, models, and technologies that underpin a project. It provides the intellectual scaffolding that explains why certain tools or approaches are chosen and how they fit together to achieve the project's goals. For a system like the Employee Analytics and Churn Prediction System, the theoretical framework is rooted in both web application development and machine learning principles.

At the heart of the web application is Flask, a lightweight Python web framework. Flask is known for its simplicity and flexibility, making it ideal for rapid development of web-based systems. It follows a minimalistic approach, allowing developers to add only the components they need, which results in a streamlined and maintainable codebase. Flask handles routing, request processing, and integrates easily with databases and other backend services.

Data storage and management in the system rely on MySQL, a relational database management system. MySQL organizes data into structured tables, enabling efficient querying, updating, and retrieval of employee records, leave requests, and analytics data. The use of SQL for data manipulation ensures data integrity and supports complex queries necessary for HR analytics.

The user interface is crafted using HTML for structure, Tailwind CSS for responsive and utility-first styling, and JavaScript for dynamic interactions. Tailwind CSS allows for rapid UI development by providing a rich set of utility classes, reducing the need for custom CSS. JavaScript is employed to enhance interactivity, manage form submissions, and facilitate real-time updates on the client side.

For data visualization, Chart.js is integrated into the system. Chart.js is a JavaScript library that renders interactive and visually appealing charts, such as bar graphs and pie charts, directly in the browser. This enables HR users to quickly interpret workforce trends, departmental distributions, and attrition rates through intuitive graphical representations.

The predictive component of the system is powered by a machine learning model developed using scikit-learn, a robust Python library for data mining and analysis. Scikit-learn provides tools for preprocessing data, training models, and evaluating their performance. In this project, a RandomForestClassifier is used to predict employee churn based on features like job satisfaction and overtime. The model is trained and validated using a standard 80/20 train-test split, ensuring reliable performance before integration into the live system.

Together, these technologies form a cohesive foundation for the Employee Analytics and Churn Prediction System, enabling secure data management, insightful analytics, and intelligent prediction capabilities within a modern web application framework.

1.4.4 Comparative Studyshap

When compared to established HR platforms like OrangeHRM, BambooHR, SAP, and Workday, the Employee Analytics and Churn Prediction System offers a distinct advantage in predictive analytics and customization. OrangeHRM and BambooHR are excellent for core HR functions and user-friendly workflows, but their analytics are mostly descriptive and lack integrated machine learning for churn prediction. SAP and Workday, while powerful and feature-rich, are designed for large enterprises and can be complex and costly, often requiring significant resources to deploy and customize.

In contrast, this project stands out by embedding a machine learning model—trained specifically for employee churn prediction—directly into the HR workflow. Leveraging scikit-learn and a Random Forest Classifier, the system delivers highly accurate, actionable insights tailored to real organizational data. Its modular, open-source architecture built with Flask and modern web technologies allows for easy adaptation and transparency, making it particularly suitable for organizations seeking data-driven HR solutions without the overhead of enterprise platforms.

Ultimately, while mainstream HR systems excel in breadth and scalability, the Employee Analytics and Churn Prediction System subtly outperforms them in delivering focused, proactive retention strategies through advanced predictive analytics.

1.4.5 Research Papers and Related Work

Employee Analytics and Churn Prediction Systems have become central to modern human resource management, with both commercial platforms and academic models advancing the field in recent years. These systems aim to help organizations understand workforce dynamics, monitor employee engagement, and proactively address turnover risks.

Open-source platforms such as OrangeHRM are widely adopted by small and medium-sized organizations for basic HR management and analytics. These systems typically provide descriptive statistics and historical trend analysis related to employee turnover, but lack advanced predictive capabilities and integration with external data sources (Ekawati, 2019).

Cloud-based solutions like BambooHR extend traditional HR functions by offering dashboards that visualize turnover rates and highlight possible risk factors. BambooHR primarily uses historical data and simple regression models to identify at-risk employees. However, its predictive functionality remains limited, focusing more on trend visualization than on individualized, real-time churn risk scoring (Fallucchi et al., 2020).

Kannimuthu and Premalatha (2024) compared multiple models, finding that deep learning algorithms outperformed traditional methods in predicting turnover, emphasizing the importance of identifying key factors influencing attrition. SAP

SuccessFactors, another enterprise-level solution, provides predefined metrics and dashboards for turnover analysis. Yet, its predictive capabilities are not real-time and often rely on manual interpretation of data (Gabrani & Kwatra, 2018).

The integration of diverse data sources, including internal communications and external labor market trends, is crucial for enhancing sustainable HR strategies. Advanced technologies such as machine learning and natural language processing are being employed to analyze unstructured data, improving efficiency and promoting ecological HR practices. Balancing data-driven strategies with the human element remains essential in HR practices (Asfahani, 2024). Maan and Maan (2023) emphasized the importance of model interpretability in predicting customer churn by utilizing Shapley values to explain feature importance. These methods aim to enhance transparency and support the development of adaptive risk thresholds, thereby enabling human resource professionals to implement targeted retention strategies based on the specific factors identified by the models (Mohiuddin et al., 2023; Makanga et al., 2024)

Several studies have investigated machine learning models for predicting employee churn. Random Forest consistently emerged as the top-performing algorithm across multiple studies, achieving high accuracy rates of 97.5% (Alamsyah & Salma, 2018; Musanga et al., 2022). Bandyopadhyay & Jadhav (2021) also found Random Forest to be the best model when compared to SVM and Naive Bayes. Other effective models included Logistic Regression, Decision Trees, and Gradient Boosting Machines (Musanga et al., 2022). The implementation of BI-driven HR practices has shown positive effects on employee retention and organizational performance (Ara, 2025).

In summary, Employee Analytics and Churn Prediction Systems are helping organizations move from simply tracking turnover to actively preventing it. As both research and commercial tools improve, HR teams are better equipped to understand why employees leave and to take action before problems arise. The real benefit comes from using these insights to create a more supportive and engaging workplace for everyone.

1.5 Development Methodology

The Development Methodology describes the overall approach used to guide the project from concept to completion. It includes the Research and Development Process, which involved exploring existing solutions and planning system features. The Project Framework ensured an organized and iterative workflow. Data Collection Methods refer to how relevant information was gathered to support system functionality and analysis.

1.5.1 Research and Development Process

The project began by thoroughly examining current HR practices and challenges, especially in mid-sized to large organizations. This initial research involved reviewing existing employee management systems and churn prediction studies to identify common features, gaps, and opportunities for improvement. The focus was on developing a system that is both practical and data-driven. Recent research has emphasized data-driven approaches to predict and reduce employee attrition, with machine learning techniques such as deep neural networks, random forests, and gradient boosting demonstrating promising results in forecasting employee churn (Pachghare et al., 2024; Srivastava & Eachempati, 2021). To support this, appropriate datasets were carefully selected and prepared to train the machine learning model accurately for churn prediction. With these insights and resources, the project progressed into the development phase, emphasizing the creation of user-friendly interfaces, robust backend functionality, and seamless integration of the predictive model within the system workflow.

1.5.2 Project Framework

The project followed the Spiral Model, which is a step-by-step method that repeats in cycles to make the system better over time. Each cycle includes planning, risk checking, development, and testing. Here's how it was used in this project:

1. **Planning Phase**: The project began by understanding what the system needed to do like managing employee data, leave requests, and predicting employee churn. Features were listed, and the scope was set based on what could be done within time and resources.
2. **Risk Analysis Phase**: Possible issues were identified such as incorrect churn predictions, user login problems, or database errors. Solutions were planned early, like using hashed passwords for better security and checking the model performance carefully.
3. **Engineering Phase**: The system was built in parts using Flask, MySQL, and machine learning. The login page, employee forms, leave system, and churn prediction feature were created one by one in small steps.
4. **Evaluation Phase**: Each part was tested to make sure it worked well. Changes were made based on what was found. For example, the churn prediction was adjusted to improve accuracy and errors in user access were fixed.

This process was repeated in cycles (spirals), where new features were added and existing ones improved with each iteration. The spiral model helped to catch problems early and improve the project step by step.

1.5.3 Data Collection Methods

To build the churn prediction model, the IBM HR Analytics Employee Attrition and Performance dataset from Kaggle was used. The original dataset has 1,470 records and 35 columns with details like age, job role, and attrition status. Some columns are removed during preprocessing to keep only the most important features for predicting employee churn.

To improve the model's performance, additional synthetic data was added, increasing the dataset to around 4,680 records. Since the data had fewer cases of employees who left, a balancing method was used to make both classes equal. This helped the model learn more fairly and avoid bias. The data was then split into 80% for training and 20% for testing. For real-world simulation within the system, a subset of 20 employee records was extracted from the dataset.

1.6 Scope and Limitations

The Scope and Limitations section defines what the system is designed to achieve and the boundaries within which it operates. It outlines the key features and functions included in the project while also acknowledging any constraints, such as features that were excluded, technical limitations, or areas for future improvement. This helps set clear expectations for users and stakeholders.

1.6.1 Project Scope

The project includes the following key features and functionalities:

1. It allows HR to perform create, read, update and delete (CRUD) operations on employee information.
2. A job application feature lets applicants apply, and HR can hire them.
3. Employees can submit leave requests, and HR can approve or reject them.
4. HR users can see visual dashboards showing employee trends and important data.
5. The system gives different access to HR and employees based on their roles.
6. It uses a machine learning model to predict which employees are at risk of leaving.

1.6.2 Project Limitations

The project excludes certain features and functionalities that fall outside its current scope, such as:

1. It does not track employee attendance, as this is usually done with special IoT devices.
2. It does not handle salary calculations or payroll management, which are managed by financial tools.
3. It does not include detailed employee performance evaluations, which require more advanced tools.
4. It does not support recruitment processes beyond basic job applications, such as full applicant tracking systems (ATS).

1.7 Report Organization

The Report Structure and Organization section provides an overview of how the document is arranged for clarity and coherence. It outlines the sequence of topics covered, beginning with the project overview, followed by methodology, system features, implementation details, testing, and evaluation.

Table 1.2: Report Structure

Chapter No.	Chapter Title	Description
CHAPTER I	**INTRODUCTON**	This section provides an overview of the project's background, objectives, related works, development approach, scope, limitations, and report structure.
CHAPTER II	**SYSTEM DEVELOPMENT PROCESS**	
	Analysis	Covers requirement analysis, feasibility, and system modeling.
	Design	Presents the system's user interface and architectural design.
	Implementation	Describes the tools, modules, and coding processes used.
	Testing	Discusses the testing strategies and results.
CHAPTER III	**CONCLUSION AND RECOMMENDATIONS**	This section summarizes the project results, presents key conclusions drawn from the findings, and suggests possible improvements or future enhancements.
REFERENCES		Lists all sources and materials referenced in the report.
APPENDICES		Contains supplementary materials such as code snippets, data samples, and additional documentation.

CHAPTER II: SYSTEM DEVELOPMENT PROCESS

2.1 Analysis

The Analysis phase helps to understand system requirements and user needs. By examining existing processes, important information is gathered to guide development and ensure the system meets its goals.

2.1.1 Requirement Analysis

The Requirement Analysis section identifies and defines the essential needs and functionalities of the system. It involves gathering input from potential users and stakeholders to understand what features are necessary, the system's performance expectations, and any constraints. This analysis guides the design and development process to ensure the final product meets user needs effectively.

2.1.1.1 Functional Requirements

The functional requirements (FR) define what the system should do and how it should behave under specific conditions. These requirements are based on the needs of the end users—HR personnel and employees—and cover features that ensure the system performs key tasks like managing employee data, handling leave requests, and predicting employee churn. Clearly defining these functionalities helps guide the development process and ensures the final product meets user expectations. The role-based functional requirements of Employee Analytics and Churn Prediction System includes:

Table 2.1: HR and Employee Functions

Requirement ID	Requirement Description	Input	Output
FR-01	User Login with Role-Based Access	Username and password	Access to system dashboard based on user role (HR or Employee)

Table 2.2: Employee Functions

Requirement ID	Requirement Description	Input	Output
FR-02	View and Edit Employee Profile	Profile update form	Updated profile details
FR-03	Submit Leave Request	Leave request form	Leave request entry under employee profile

Table 2.3: HR Functions

Requirement ID	Requirement Description	Input	Output
FR-04	View Analytical Dashboards	User login (HR role)	Charts showing attrition trends, department stats
FR-05	Review and Manage Job Applicants	List of submitted applications	Applicant status updated to Hired or Rejected; user account created if hired
FR-06	HR adds or edits employee data	Employee details form (name, role, salary, etc.)	New or updated employee record in the database
FR-07	Approve or Reject Leave Request	Leave request with decision	Updated leave request status and email notification
FR-08	Predict Employee Churn	Employee data (internal from DB)	Churn prediction label (Yes/No)

Table 2.4: Applicant Functions

Requirement ID	Requirement Description	Input	Output
FR-09	Submit Job Application	Job application form	New applicant record created
FR-10	Receive Hiring Notification Email	Hiring decision trigger	Email sent to applicant if hired

2.1.1.2 Use Case Diagram

A Use Case Diagram visually represents the interactions between users and the system to illustrate its functional requirements.

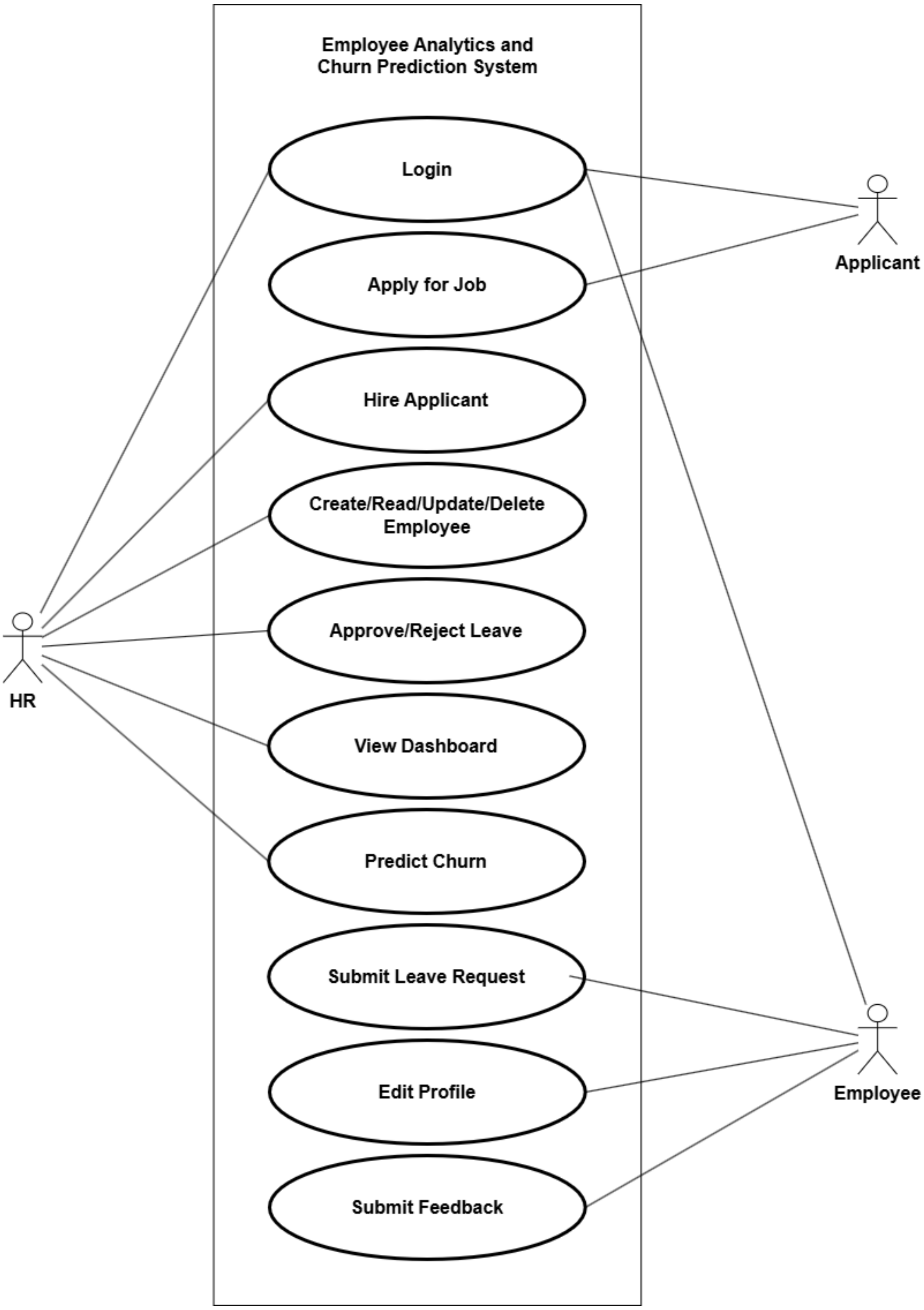


Figure 2.1: Use-Case Diagram

2.1.1.3 Use Case Scenario

Use Case (UC) Scenarios describe specific sequences of actions between users and the system to achieve particular goals, helping to clarify functional requirements and user interactions.

Table 2.5: UC-01 User Login with Role-Based Access

Field	Description
Use Case Identifier	UC-01
Primary Actor	User (HR or Employee)
Secondary Actor	System
Description	The user logs into the system using valid credentials. The system authenticates the user and grants access to the appropriate dashboard based on their role.
Precondition	The user is registered with valid credentials and assigned a valid role.
Post-condition	The user is logged in and redirected to a dashboard that matches their role.
Success Scenario	User enters valid username and password → system verifies credentials → role is checked → session starts → user is redirected to HR or Employee dashboard.
Failure Scenario	Invalid credentials or role → access denied → appropriate error message is displayed.

Table 2.6: UC-03 View and Edit Employee Profile

Field	Description
Use Case Identifier	UC-03
Primary Actor	Employee
Secondary Actor	System
Description	The employee views and updates their personal profile details such as name, contact info, etc.
Precondition	Employee must be logged in successfully.
Post-condition	Profile is updated in the system and reflects the new information.
Success Scenario	Employee logs in → navigates to profile → edits information → submits → system updates record.
Failure Scenario	Invalid input or system error during submission → update fails → error message is shown.

Table 2.7: UC-06 Submit Leave Request

Field	Description
Use Case Identifier	UC-04
Primary Actor	Employee
Secondary Actor	System
Description	Employee fills out a leave form and submits a request for approval.
Precondition	Employee must be logged in and have access to the leave request page.
Post-condition	Leave request is saved in the system and awaits HR approval.
Success Scenario	Employee logs in → navigates to leave request form → fills and submits → request saved in DB.
Failure Scenario	Form incomplete or invalid data → system shows error → leave request not submitted.

Table 2.8: UC-06 View Analytical Dashboards

Field	Description
Use Case Identifier	UC-06
Primary Actor	HR
Secondary Actor	System
Description	HR views interactive dashboards showing employee analytics, attrition trends, and department stats.
Precondition	HR user must be logged in and have dashboard access rights.
Post-condition	Relevant charts and reports are displayed based on current employee data.
Success Scenario	HR logs in → navigates to dashboard → system retrieves and displays updated analytics charts.
Failure Scenario	Data retrieval failure or system error → charts fail to load → error message shown.

Table 2.9: UC-07 Review and Manage Job Applicants

Field	Description
Use Case Identifier	UC-02
Primary Actor	HR User
Secondary Actor	System

Description	The HR views job applications and decides whether to hire or reject each applicant. If hired, the system creates a user account and transfers applicant data to the employee database.
Precondition	The applicant has submitted a complete job application and the HR user is logged in.
Post-condition	The applicant's status is updated; if hired, their data is added to the employee records and a system account is generated.
Success Scenario	HR logs in → views list of applicants → selects an applicant → chooses "Hire" → system adds data to employee table and creates a user account.
Failure Scenario	HR attempts to hire without valid application data → error shown; or hiring action fails → status not updated.

Table 2.10: UC-07 HR Adds or Edits Employee Data

Field	Description
Use Case Identifier	UC-07
Primary Actor	HR
Secondary Actor	System
Description	HR staff adds new employee records or updates existing employee information in the system.
Precondition	HR user must be logged in with appropriate permissions.
Post-condition	Employee data is saved or updated successfully in the database.
Success Scenario	HR logs in → navigates to employee management → fills or edits form → submits → data saved.
Failure Scenario	Invalid input or database error → data not saved → error message displayed.

Table 2.11: UC-08 Approve or Reject Leave Requests

Field	Description
Use Case Identifier	UC-08
Primary Actor	HR
Secondary Actor	System, Employee
Description	HR reviews pending leave requests and approves or rejects them, triggering notifications to employees.

Precondition	HR must be logged in and have access to leave request management.
Post-condition	Leave request status is updated; if approved, employee receives email notification.
Success Scenario	HR logs in → reviews leave requests → approves or rejects → status updated → notification sent.
Failure Scenario	System error or invalid operation → status not updated → error message shown.

Table 2.12: UC-09 Predict Employee Churn

Field	Description
Use Case Identifier	UC-09
Primary Actor	HR
Secondary Actor	System
Description	The system predicts the likelihood of employee attrition using the trained machine learning model.
Precondition	HR must be logged in; employee data must be up-to-date and available for prediction.
Post-condition	Churn prediction results are generated and displayed on the dashboard.
Success Scenario	HR requests prediction → system processes data → displays prediction (Yes/No) for each employee.
Failure Scenario	Data missing or model error → prediction fails → error message displayed.

Table 2.13: UC-09 Submit Job Application

Field	Description
Use Case Identifier	UC-10
Primary Actor	Applicant
Secondary Actor	System
Description	The applicant fills out and submits a job application form through the system.
Precondition	The job application form is accessible, and the applicant provides required details.
Post-condition	A new applicant record is stored in the database.
Success Scenario	Applicant opens form → fills in details → submits form → system saves application.

Failure Scenario	Required fields are missing → system shows error → form not submitted.

Table 2.14: UC-10 Receive Hiring Notification Email

Field	Description
Use Case Identifier	UC-11
Primary Actor	System
Secondary Actor	Applicant
Description	The system sends an email to the applicant after the HR marks them as hired.
Precondition	HR must update the applicant status to “Hired.”
Post-condition	The applicant receives a notification email with hiring details.
Success Scenario	HR hires applicant → system triggers email → applicant receives confirmation.
Failure Scenario	Email service is down or incorrect email provided → applicant does not receive notification.

2.1.1.4 Non-functional Requirements

Non-functional requirements define how the system should perform, focusing on quality, security, and usability. Some non-functional requirements of Employee Analytics and Churn Prediction System include:

Table 2.15: Non-functional Requirements

Requirement ID	Category	Description
NFR-01	Performance	The system should load key pages (dashboard, profile) within 3 seconds under normal conditions.
NFR-02	Usability	The user interface must be simple, intuitive, and consistent for both HR and Employees.
NFR-03	Security	Passwords must be hashed using bcrypt; sessions and user roles must be securely managed.
NFR-04	Compatibility	The application should work across major web browsers.
NFR-05	Accuracy	The churn prediction model should maintain at least 90% accuracy on test data.

2.1.2 Feasibility Study

An evaluation covering Technical, Economic, Legal, Operational, and Schedule aspects was performed to determine the project's viability. This helped ensure the system could be developed successfully while meeting all necessary requirements and constraints.

2.1.2.1 Technical Feasibility

The project is technically feasible as it uses well-supported technologies like Flask (Python), MySQL, HTML, Tailwind CSS, JavaScript, and scikit-learn for machine learning. These tools are open-source, have strong community support, and are suitable for building both the analytical and predictive features of the system. Having experience with these tools and all resources being freely available, technical implementation was achievable without external dependencies.

2.1.2.2 Economic Feasibility

The system is cost-effective since it primarily relies on free and open-source technologies. No paid tools or licenses are required for development or local testing. Since the project is not intended for deployment in this phase, server or hosting costs are not applicable. Long-term economic benefits include reduced HR workload, better retention planning, and improved decision-making through predictive analytics.

2.1.2.3 Legal Feasibility

The system uses publicly available datasets (IBM HR Dataset from Kaggle) that are cleared for academic and non-commercial use. It handles sensitive employee data with proper login authentication and hashed passwords, following good data handling practices. If deployed commercially, additional data protection policies may need to be considered.

2.1.2.4 Operational Feasibility

The system meets key operational needs of HR and employees, such as managing employee data, processing leave requests, and predicting churn. The interfaces are role-specific, user-friendly, and meet the day-to-day requirements of the intended users. The system has been tested to ensure functionality and ease of use.

2.1.2.5 Schedule Feasibility

The project was planned and developed using the Spiral methodology, with clearly defined tasks and iterative updates. The scope was well managed by focusing only on core features. The work was completed within the expected academic timeline, making it feasible in terms of scheduling.

2.1.3 Structured Modelling

The Employee Analytics and Churn Prediction System is designed using a structured modeling approach. This method focuses on breaking down the system into clearly defined processes and data flows, making it ideal for applications with role-based features and CRUD operations.

Structured modeling helps visualize how data moves through different modules such as login, employee management, leave handling, and churn prediction. It typically involves tools like data flow diagrams (DFDs) and entity-relationship diagrams (ERDs) to represent how inputs are processed and stored.

By following a top-down design approach, the system remains organized, easy to develop, and straightforward to test. Given the predictable workflows and well-defined user roles, structured modeling is a suitable and effective choice for this project.

The following diagrams were central to this process:

2.1.3.1 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) shows how data moves within the system, describing the interaction between users, system processes, and the database. In this project, the DFD includes key processes like user login, employee data management, leave requests, churn prediction, and the job application and hiring process.

External entities such as HR, Employees, and Applicants interact with processes that handle authentication, CRUD operations, machine learning predictions, and job applications. These processes exchange data with internal data stores like the users, employees, applicants, and leave requests tables in the database.

The DFD helps to clearly show how all modules are connected and how data flows through the system step-by-step, making the design easier to understand and manage.

Level 0 DFD (Context Diagram)

The Level 0 DFD, also known as the context diagram, provides a high-level overview of the entire system as a single process. It shows the system’s interactions with external entities such as users or other systems, emphasizing what data enters and exits the system without detailing internal processes. This diagram sets the foundation for deeper analysis by identifying the main data flows and external actors involved.

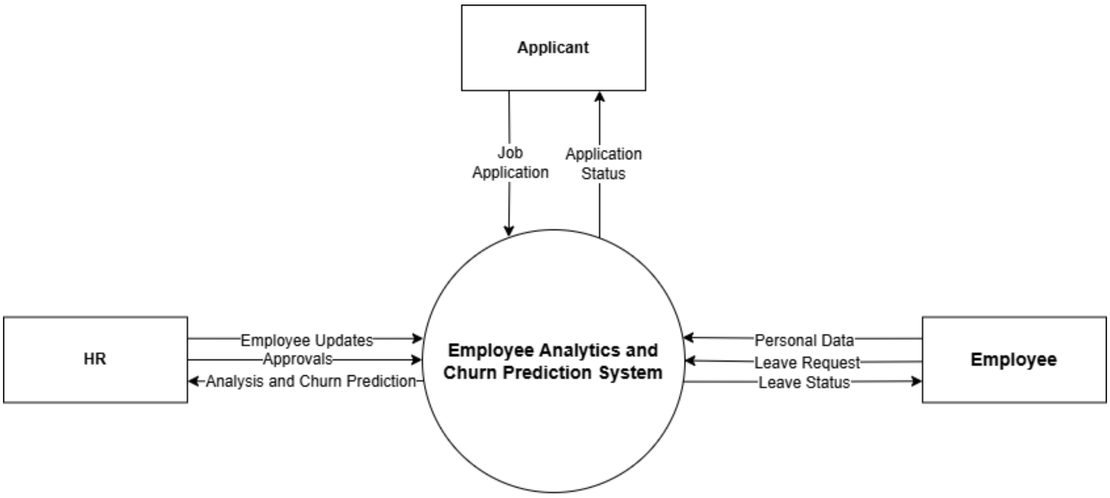


Figure 2.2: DFD Level 0 (Context Diagram)

Level 1 DFD

A Level 1 DFD decomposes the single process from the Level 0 diagram into multiple sub-processes. It reveals major functional areas within the system and shows how data moves between them, external entities, and data stores. This level provides more detail, offering insight into how the system handles different operations and the logic behind each process.

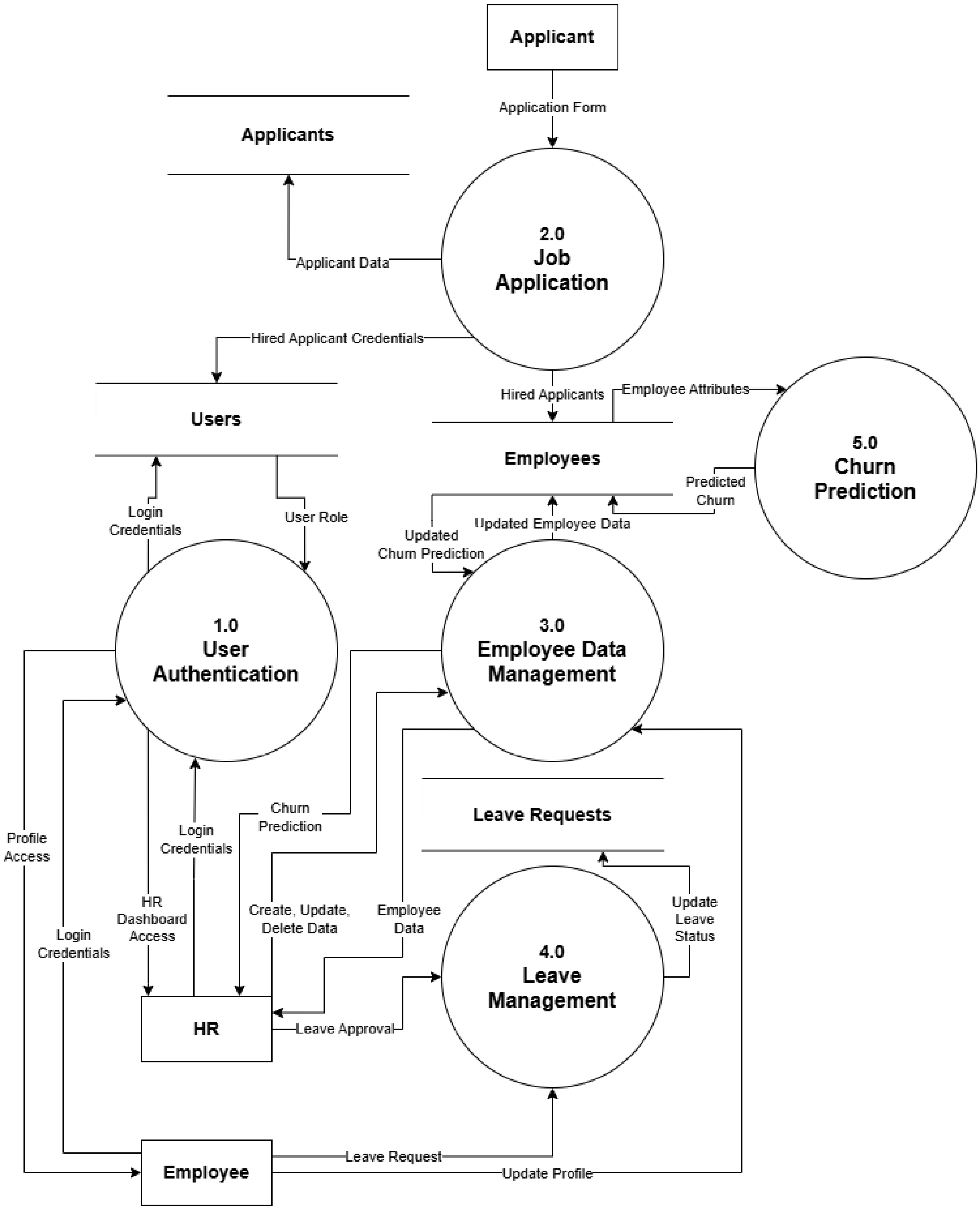


Figure 2.3: DFD Level 1

Level 2 DFD

The Level 2 DFD further breaks down individual processes from the Level 1 diagram into more specific sub-processes. It gives a detailed look into each functional component, showing precise data inputs, processing steps, and outputs. This level is often used by developers and system analysts to understand intricate workflows and design efficient system components.

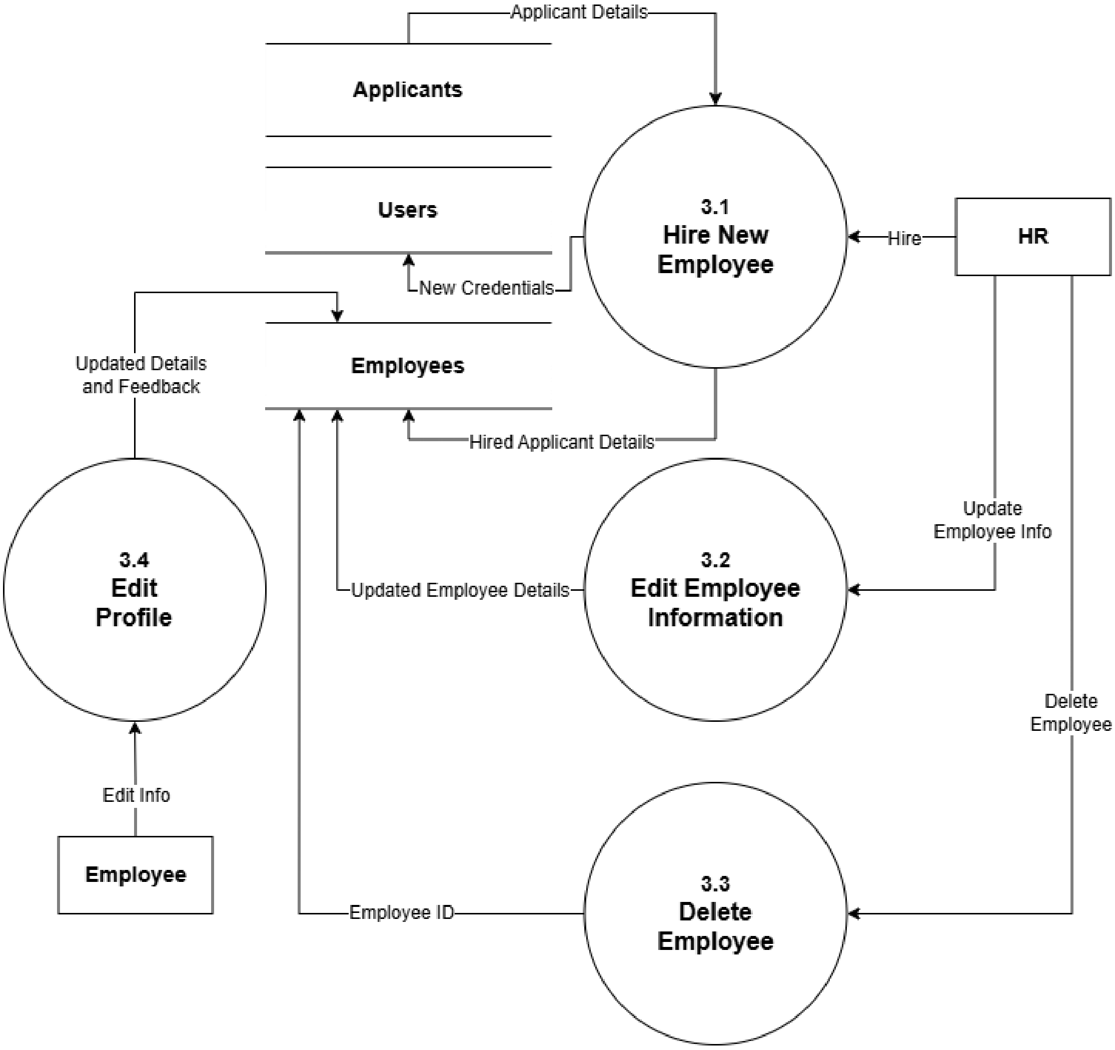


Figure 2.4: DFD Level 2

2.1.3.2 Entity-Relationship Diagram (ERD)

The Entity-Relationship Diagram (ERD) illustrates the structure of the database by defining the entities, their attributes, and the relationships between them. In this system, the key entities are Users, Employees, and Leave Requests.

Each user has a role (either HR or Employee), and is linked to an employee record. Employees can submit multiple leave requests, which are tracked and managed by HR. The ERD helps ensure that the database is well-organized, supports role-based access, and maintains data integrity.

By mapping out these relationships, the ERD provides a clear foundation for designing efficient queries and maintaining consistency across the system.

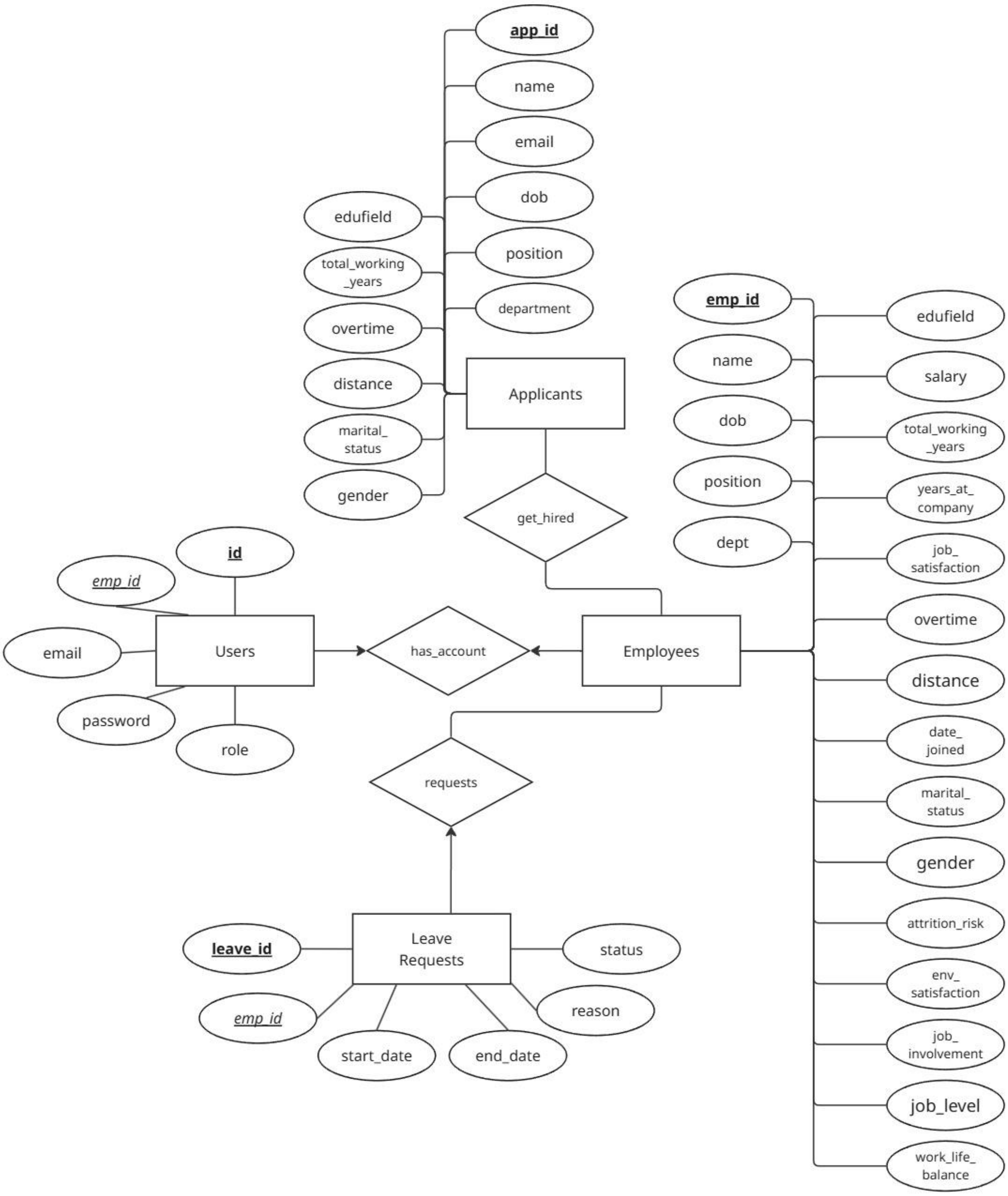


Figure 2.5: ER Diagram

2.1.4 System Requirements Specification (SRS)

The System Requirements Specification (SRS) lists the hardware and software needed for the system to work.

Table 2.16: Hardware Requirements

Category	**Requirements (Minimum / Recommended)**	**Purpose**
Processor	Minimum: Dual-core 2.0 GHz / Recommended: Quad-core 2.5 GHz+	To run the application and ML models smoothly
RAM	Minimum: 4 GB / Recommended: 8 GB or more	To support multitasking and data processing
Storage	Minimum: 100 GB HDD/SSD / Recommended: 256 GB SSD+	To store project files, database, and software
Display	Minimum: 13-inch HD / Recommended: 14-inch Full HD+	To provide clear visualization of UI and dashboards

Table 2.17: Software Requirements

Category	**Requirements (Minimum / Recommended)**	**Purpose**
Operating System	Minimum: Windows 10 64-bit or Linux / Recommended: Latest versions	To provide a stable environment for development and testing
Programming Language and Framework	Python 3.8+ / Flask Framework latest stable	To develop backend, APIs, and machine learning components
Database	MySQL Server latest stable	To store and manage employee and leave data securely
Web Browser	Modern browsers like Chrome, Firefox, Edge latest versions	To access and test the web application interface
Development Tools	Text Editor (minimum) / IDE like VS Code or PyCharm (recommended)	To write, debug, and maintain code efficiently

2.2 Design

Planning the system's structure and user interface is essential to create a clear and organized solution. This involves deciding how components connect and how users will interact with the system to meet all requirements effectively.

2.2.1 System Architecture

The Employee Analytics and Churn Prediction System is structured using a three-tier architecture, which separates the system into three logical layers: the Presentation Tier, Application Tier, and Data Tier. This design enhances modularity, scalability, and maintainability.

1. **Presentation Tier (Frontend)**: The presentation layer is the user interface of the system, accessible via a web browser. It is developed using HTML, Tailwind CSS, JavaScript, and Chart.js. It allows HR and Employee users to interact with the system, including logging in, managing records, requesting leave, and viewing analytics dashboards. This tier handles all user input and displays visual feedback from backend responses.

2. **Application Tier (Backend/Logic Layer)**: The application logic is handled by the Flask framework. It processes user requests, performs authentication, manages sessions, handles CRUD operations for employee and leave data, and serves as the communication bridge between the frontend and backend. It also integrates the churn prediction module, where a Random Forest model built using scikit-learn is used to evaluate attrition risk based on employee features.

3. **Data Tier (Database)**: The data storage layer uses a MySQL database. It holds all persistent data including user credentials, employee records, leave applications, and churn prediction results. This tier ensures data integrity, supports queries for data retrieval, and serves as the foundation for analytics and model inputs.

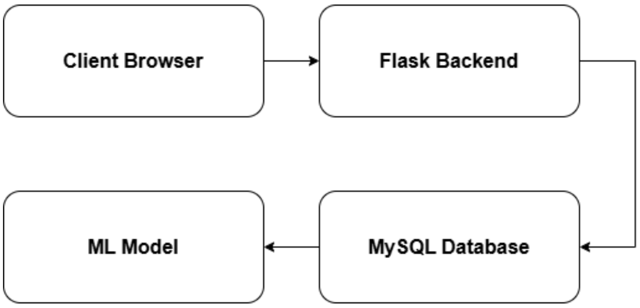


Figure 2.6: System Architecture Diagram

2.2.2 Database Design

A database schema is the structural design that shows how data is organized in a database, including tables, fields, relationships, and rules. It works as a blueprint that defines how data is stored, linked, and accessed inside the system.

In the Employee Analytics and Churn Prediction System, the database schema includes four main tables: employees, users, leave requests, and applicants. The employees table stores detailed information about each staff member, such as department, salary, experience, and other fields used in analytics and churn prediction. The users table manages system login and access, storing credentials and assigning roles (HR or employee), and is linked to each employee through a foreign key. The leave requests table records leave applications submitted by employees, including type, duration, and approval status. The applicants table saves the job application details submitted by users applying for positions, and once an applicant is hired, their data is moved to the employees table and a user account is created.

These tables work together to ensure smooth system operation and data consistency, supporting key features like authentication, leave handling, analytics, and the hiring process.

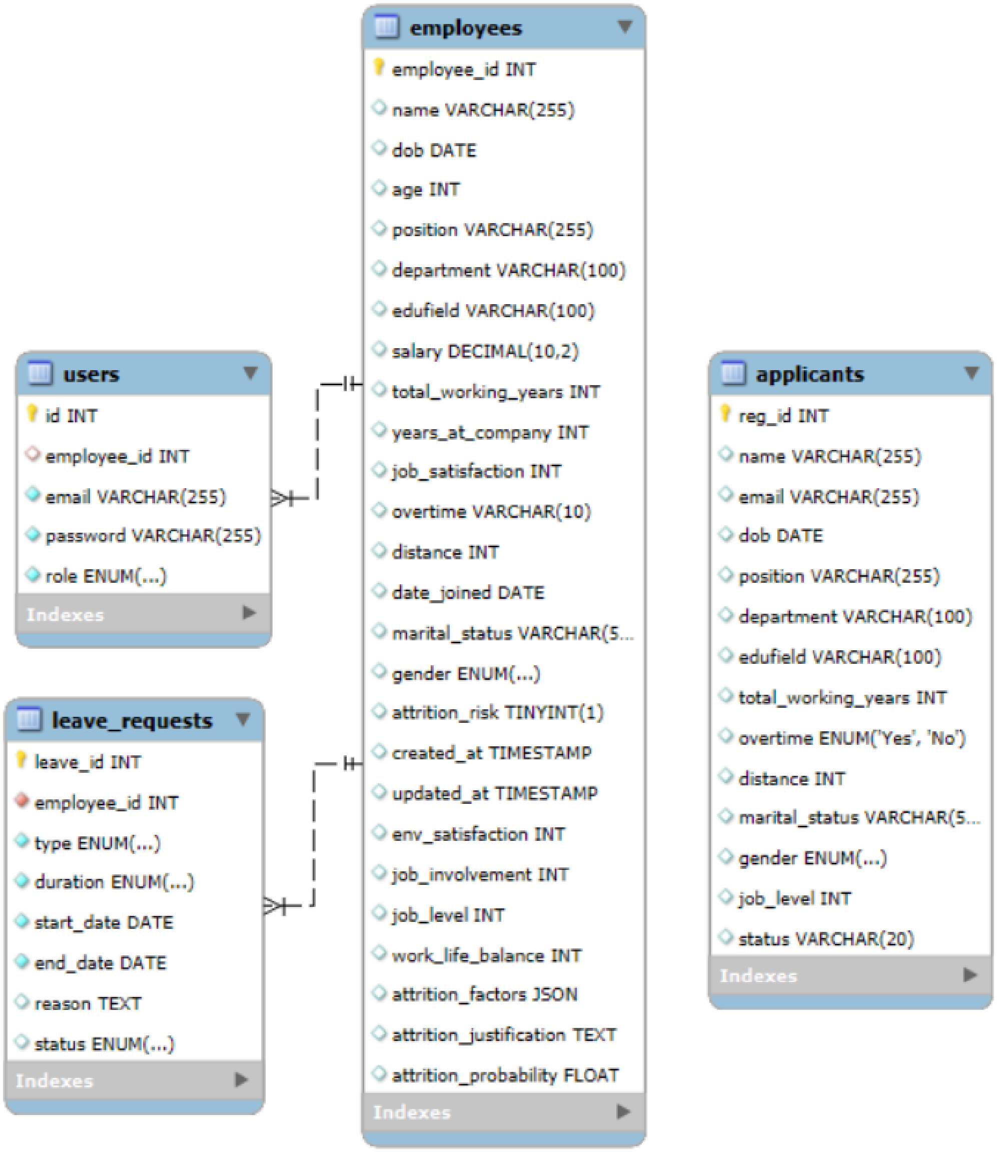


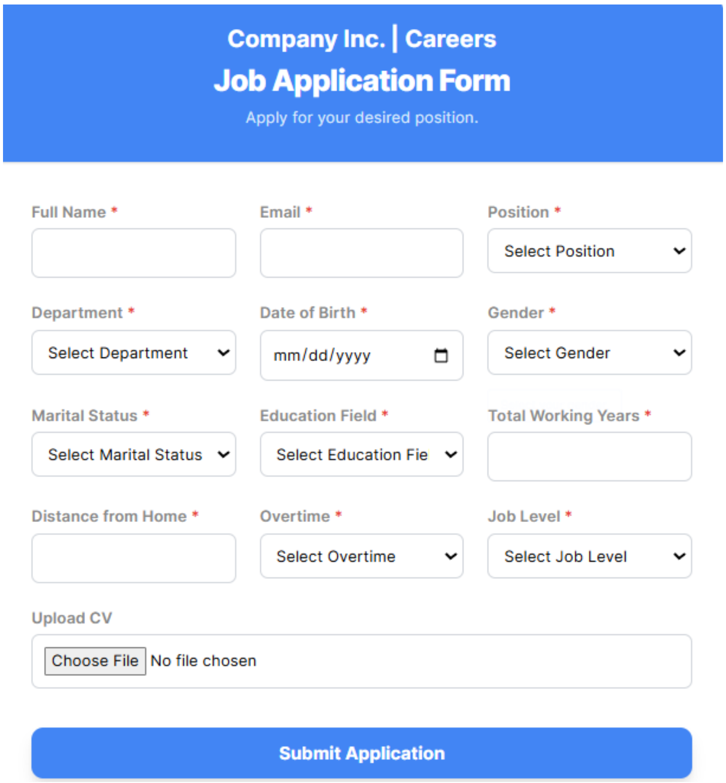
Figure 2.7: Database Schema Diagram

2.2.3 User Interface Design

A user interface (UI) diagram shows how the screens or pages are arranged in a system and how users move between them. It helps visualize the navigation flow and where key elements like forms, buttons, and dashboards are placed.

In the Employee Analytics and Churn Prediction System, the UI diagram includes three main roles, HR, Employee, and Applicant. For HR, the interface has analytics dashboards, employee management pages (add, edit, view, delete), leave approval sections, churn prediction results, and views to manage job applications. Employees have access to a personal profile page, a leave request form, leave history, and feedback options. Applicants can fill out and submit job application forms through a dedicated view.

This layout helps keep the system simple and user-friendly, making it easier for all users to access features relevant to their role.



Company Inc. | Careers

Job Application Form

Apply for your desired position.

Full Name *

Email *

Position *

Select Position

Department *

Select Department

Date of Birth *

mm/dd/yyyy

Gender *

Select Gender

Marital Status *

Select Marital Status

Education Field *

Select Education Fie

Total Working Years *

Distance from Home *

Overtime *

Select Overtime

Job Level *

Select Job Level

Upload CV

Choose File No file chosen

Submit Application

Figure 2.8: Job Application UI

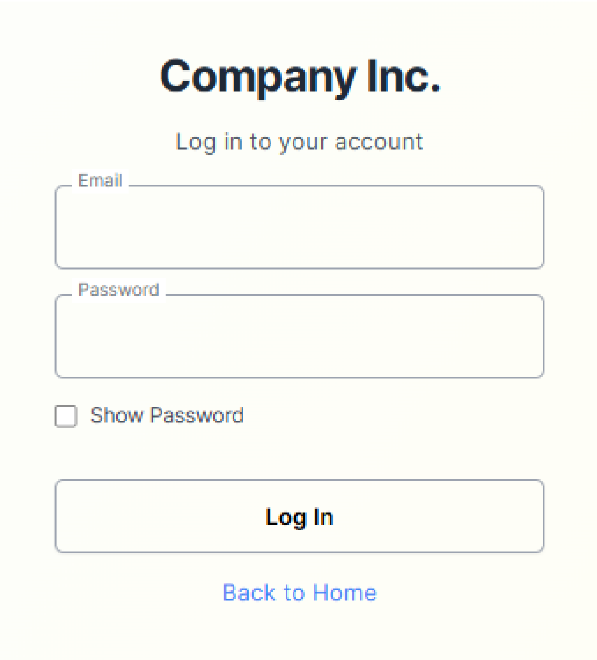
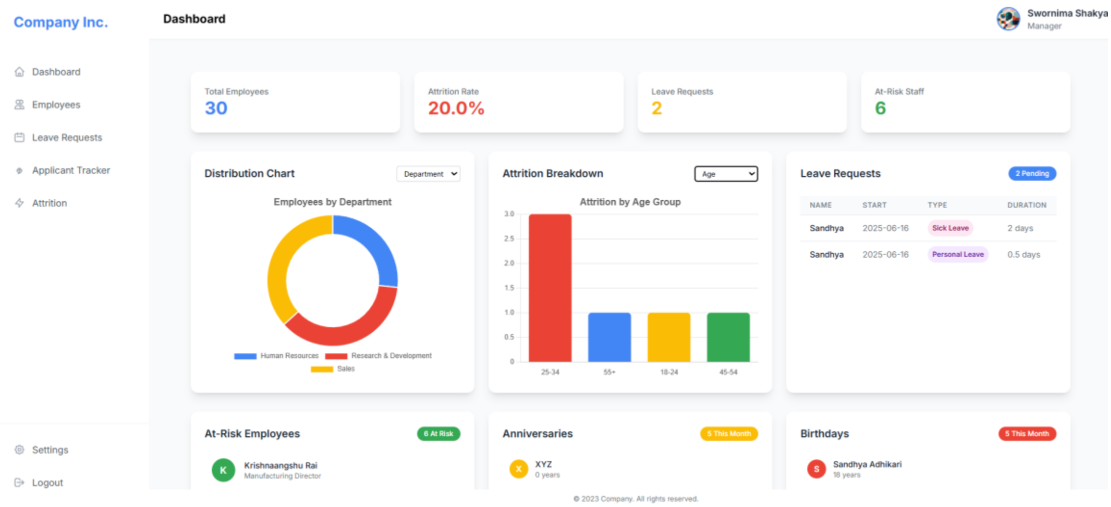


Figure 2.9: Login UI



Total Employees
30

Attrition Rate
20.0%

Leave Requests
2

At-Risk Staff
6

NAME	START	TYPE	DURATION
Sandhya	2025-06-16	Sick Leave	2 days
Sandhya	2025-06-16	Personal Leave	0.5 days

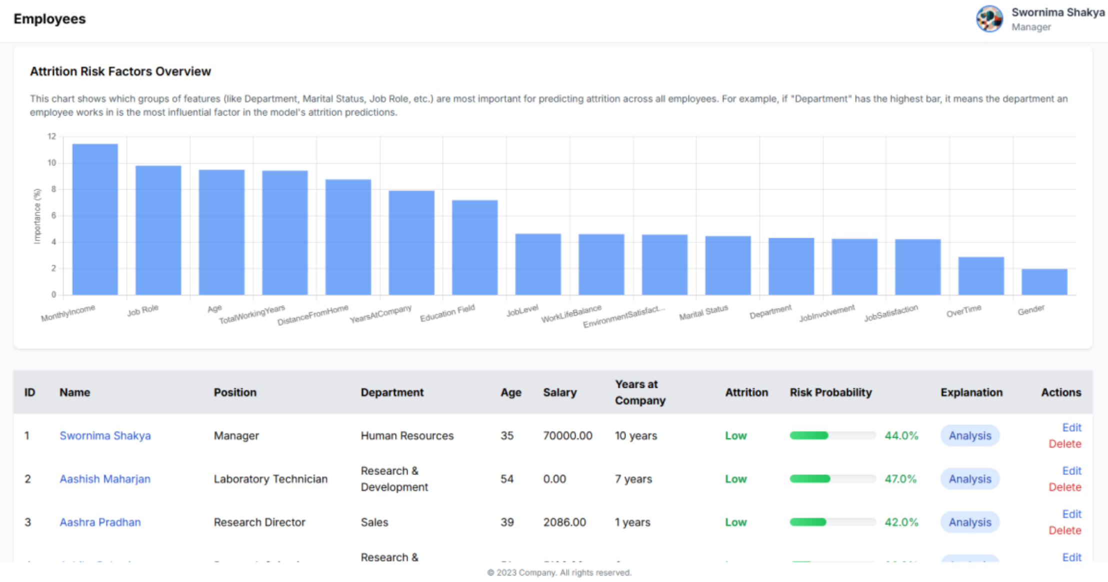
K Krishnaangshu Rai
Manufacturing Director

X XYZ
0 years

S Sandhya Adhikari
18 years

© 2023 Company. All rights reserved.

Figure 2.10: Dashboard View (HR)



Employees

Swornima Shakya
Manager

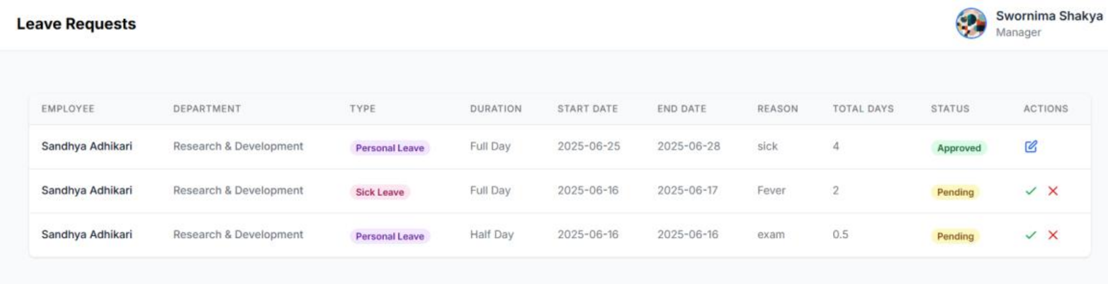
Attrition Risk Factors Overview

This chart shows which groups of features (like Department, Marital Status, Job Role, etc.) are most important for predicting attrition across all employees. For example, if "Department" has the highest bar, it means the department an employee works in is the most influential factor in the model's attrition predictions.

ID	Name	Position	Department	Age	Salary	Years at Company	Attrition	Risk Probability	Explanation	Actions
1	Swornima Shakya	Manager	Human Resources	35	70000.00	10 years	Low	44.0%	Analysis	Edit Delete
2	Aashish Maharjan	Laboratory Technician	Research & Development	54	0.00	7 years	Low	47.0%	Analysis	Edit Delete
3	Aashra Pradhan	Research Director	Sales	39	2086.00	1 years	Low	42.0%	Analysis	Edit Delete
			Research &							Edit

© 2023 Company. All rights reserved.

Figure 2.11: Employee Details and Churn Risk View (HR)

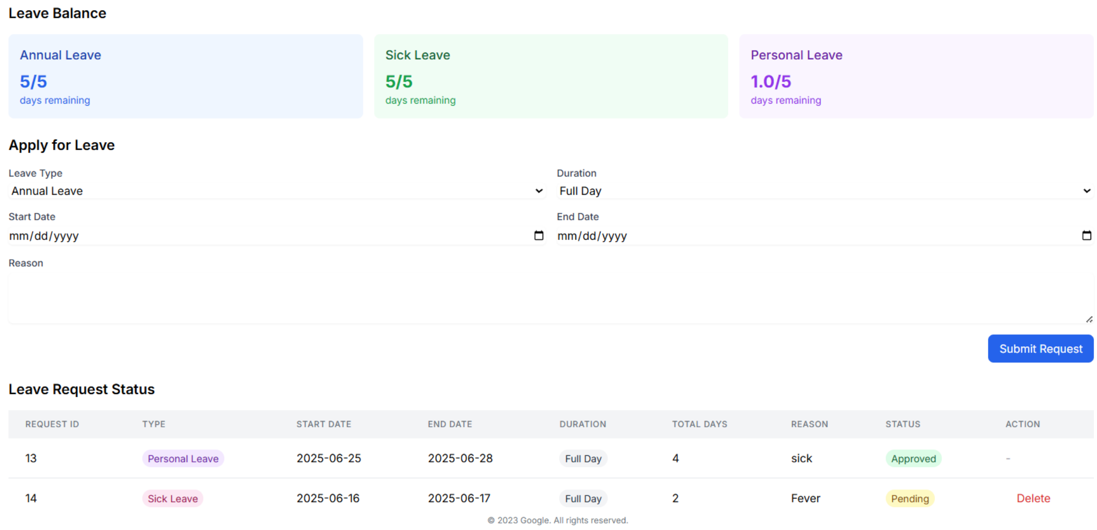


Leave Requests

Swornima Shakya
Manager

EMPLOYEE	DEPARTMENT	TYPE	DURATION	START DATE	END DATE	REASON	TOTAL DAYS	STATUS	ACTIONS
Sandhya Adhikari	Research & Development	Personal Leave	Full Day	2025-06-25	2025-06-28	sick	4	Approved	✎
Sandhya Adhikari	Research & Development	Sick Leave	Full Day	2025-06-16	2025-06-17	Fever	2	Pending	✓ ✗
Sandhya Adhikari	Research & Development	Personal Leave	Half Day	2025-06-16	2025-06-16	exam	0.5	Pending	✓ ✗

Figure 2.12: Leave Requests View (HR)



Leave Balance

Annual Leave
5/5
days remaining

Sick Leave
5/5
days remaining

Personal Leave
1.0/5
days remaining

Apply for Leave

Leave Type
Annual Leave ⌄

Duration
Full Day ⌄

Start Date
mm/dd/yyyy

End Date
mm/dd/yyyy

Reason

Leave Request Status

REQUEST ID	TYPE	START DATE	END DATE	DURATION	TOTAL DAYS	REASON	STATUS	ACTION
13	Personal Leave	2025-06-25	2025-06-28	Full Day	4	sick	Approved	-
14	Sick Leave	2025-06-16	2025-06-17	Full Day	2	Fever	Pending	Delete

© 2023 Google. All rights reserved.

Figure 2.13: Leave Request View (Employee)

2.2.4 Algorithm Design

Algorithm design is the process of creating a clear and logical set of steps to solve problems or complete tasks in a system. It helps define how the system should process inputs to reach the correct output.

In the Employee Analytics and Churn Prediction System, algorithm design is mainly used in the machine learning part. A Random Forest Classifier was used to predict employee attrition. The model was trained on the IBM HR dataset after cleaning the data by removing unnecessary fields, converting categories to numbers, and normalizing values. It uses factors like job satisfaction, years at the company, and salary to predict if an employee might leave. Additionally, simple logic is used in the hiring process to move applicant data to the employee table once hired.

This logical structure helps make accurate predictions and improves HR decision-making to lower churn and manage employee data more effectively.

2.2.4.1 User Login and Authentication Module

The User Login and Authentication module is responsible for securely verifying user identities before granting access to the system.

Pseudocode

```
FUNCTION login:
    IF form is submitted:
        Get email and password from form

        Connect to database
        Look for user with given email

        IF user not found:
            Show error and go back to login

        Check if stored password is hashed
        Compare entered password with stored one

        IF password is incorrect:
            Show error and go back to login

        Save user info in session

        Get employee name and position
        Save in session

        IF role is HR:
            Go to HR dashboard
        ELSE IF role is Employee:
            Go to Employee view
        ELSE:
            Show error for invalid role
            Go back to login
    ELSE:
        Show login form
```

2.2.4.2 Job Application Submission Module

The Job Application Submission module allows applicants to fill in and submit their job applications. The system stores the application details in the database for HR review.

Pseudocode

```
FUNCTION job_apply:
    IF form is submitted:
        Read all form fields like name, email, position, etc.

        Connect to the database

        TRY:
            Open database cursor
            Insert the form data into applicants table
            Save the changes
            Close the cursor

        FINALLY:
            Close the database connection

        Show message: "Application submitted successfully"
        Go back to application page

    ELSE:
        Show the job application form
```

2.2.4.3 Employee Data Management Module

This module allows HR users to view, add, edit, and delete employee records. It also supports automatic user account creation and churn prediction for each employee.

Pseudocode

```
FUNCTION show_employees:
    Check if user is logged in and is HR
    Get all employee data from database
    Show the list on the HR page

FUNCTION add_employee:
    IF form is submitted:
        Get employee details from the form
        Calculate how many years they've worked

        Add employee to the database
        Create login account for the employee
        Check if the employee is likely to leave

        Show success message and go back to list
    ELSE:
        Show the form to add a new employee

FUNCTION edit_employee(employee_id):
    IF form is submitted:
        Get updated details from form
        Recalculate years at company

        Update employee info in database
        Run prediction again

        Show success message and go back
    ELSE:
        Load employee data and show in form

FUNCTION delete_employee(employee_id):
    Remove user account for employee
```

```
          Delete employee from database
          Show success message and go back to list
```

2.2.4.4 Leave Management Module

The Leave Management module allows employees to submit leave requests, view their leave history, and enables HR to approve or reject leave applications while sending email notifications upon approval.

Pseudocode

```
      FUNCTION leave_requests:
          Get all leave requests for HR
          Show the leave requests page

      FUNCTION emp_leave:
          IF form is submitted:
              Read leave details from form
              Calculate total leave days
              Save new leave request with 'Pending' status
              Redirect to leave page

          ELSE:
              Get leave records for current employee
              Calculate total leave days for each
              Show the leave history page

      FUNCTION update_leave_status(request_id, status):
          IF user not logged in OR not HR:
              Redirect to login

          IF status is invalid:
              Redirect to leave requests page

          Update status of the leave request

          IF status is 'Approved':
              Get employee info
              Send approval email

          Redirect to leave requests page
```

2.2.4.5 Churn Prediction Module

The Churn Prediction module evaluates the likelihood of employee attrition using machine learning and updates the database, helping HR proactively identify at-risk employees.

Pseudocode

```
      FUNCTION preprocess_employee_data(raw_data):
          Convert data to DataFrame
          Standardize key names to lowercase
          Encode binary fields like gender and overtime
          One-hot   encode   categorical   fields   like   department,
education, etc.
          Ensure all expected columns exist
          Reorder columns to match model input
          Apply saved scaler to normalize data
          RETURN processed data
```

```
FUNCTION predict_attrition_for_all:
    Get all employee records from database
    FOR each employee:
        Preprocess data
        Predict attrition risk with model
        Update prediction result in database
    Save and close database connection

ROUTE /predict-attrition:
    Check user is logged in and is HR
    Run predictions for all employees
    Get employees marked high-risk
    Show results on prediction page
```

2.2.5 System Flowcharts

A system flowchart is a visual diagram that shows the flow of data and processes within a system, illustrating how different components interact. It helps simplify complex systems by making it easier to understand, analyze, and communicate the workflow. System flowcharts are closely related to algorithm design because they both outline step-by-step procedures, with flowcharts providing a clear, visual representation of the algorithm’s logic. This connection aids in planning and debugging system processes effectively.

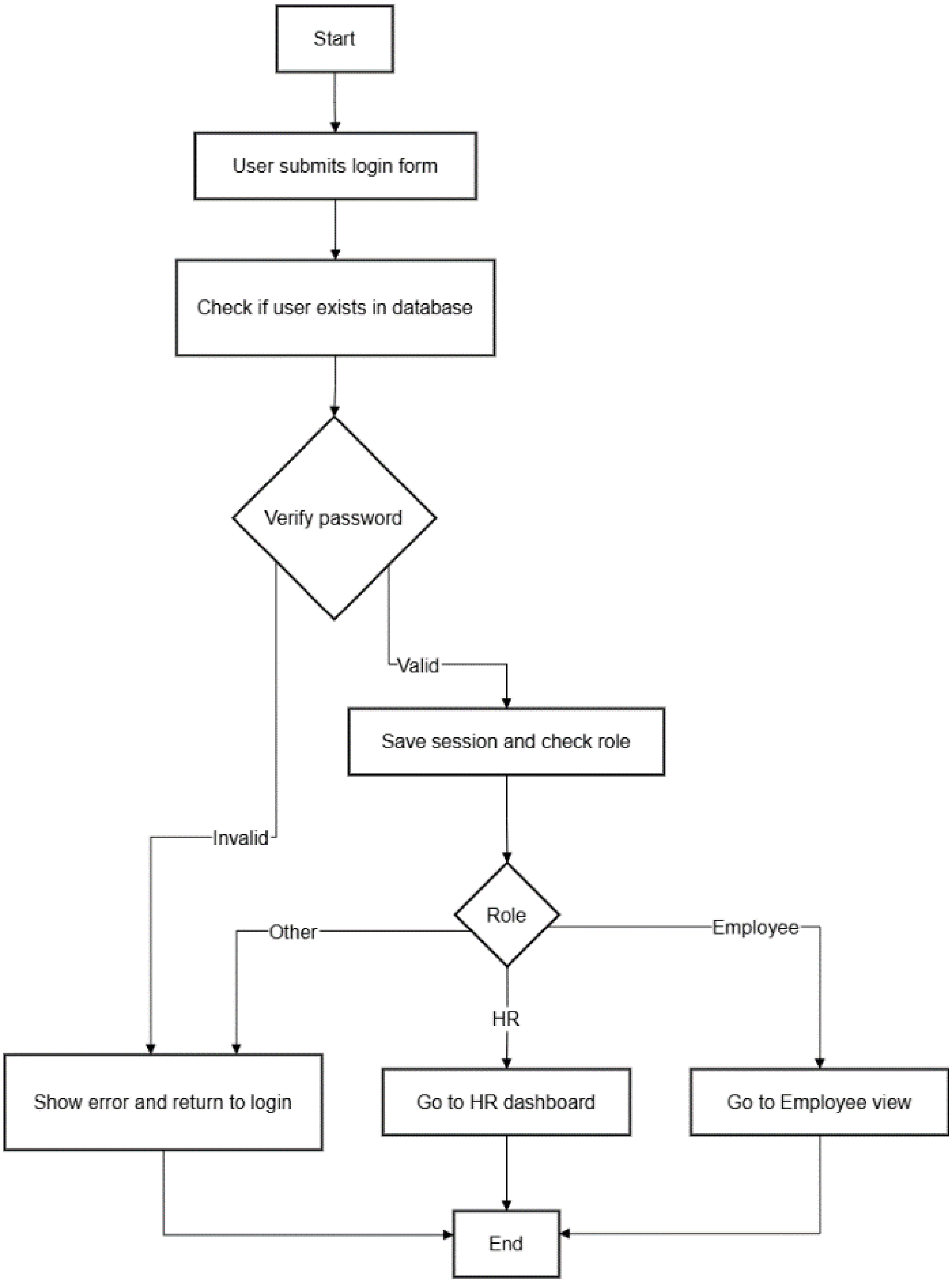


Figure 2.14: User Login and Authentication Algorithm Flowchart

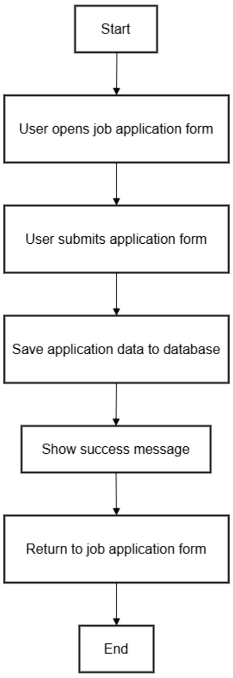

Start
User opens job application form
User submits application form
Save application data to database
Show success message
Return to job application form
End

Figure 2.15: Job Application Module Flowchart

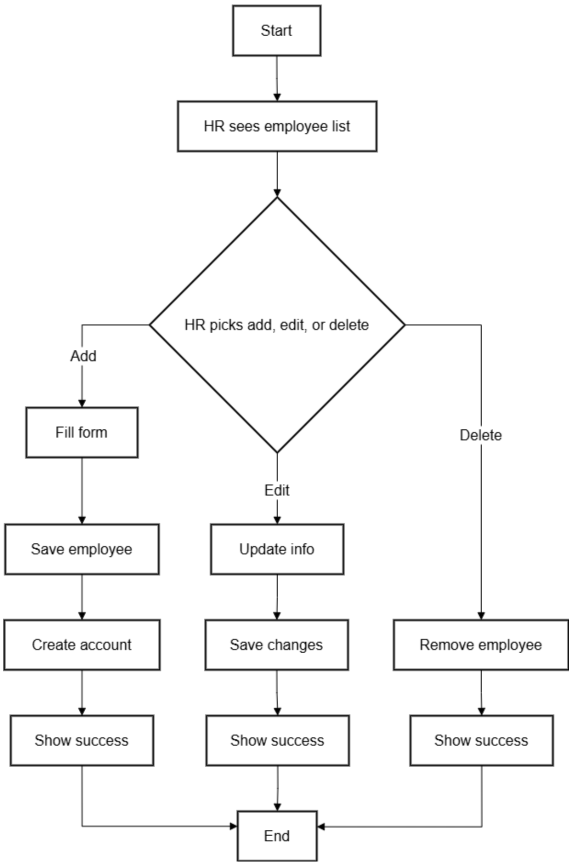


Figure 2.16: Employee Data Management Module Flowchart

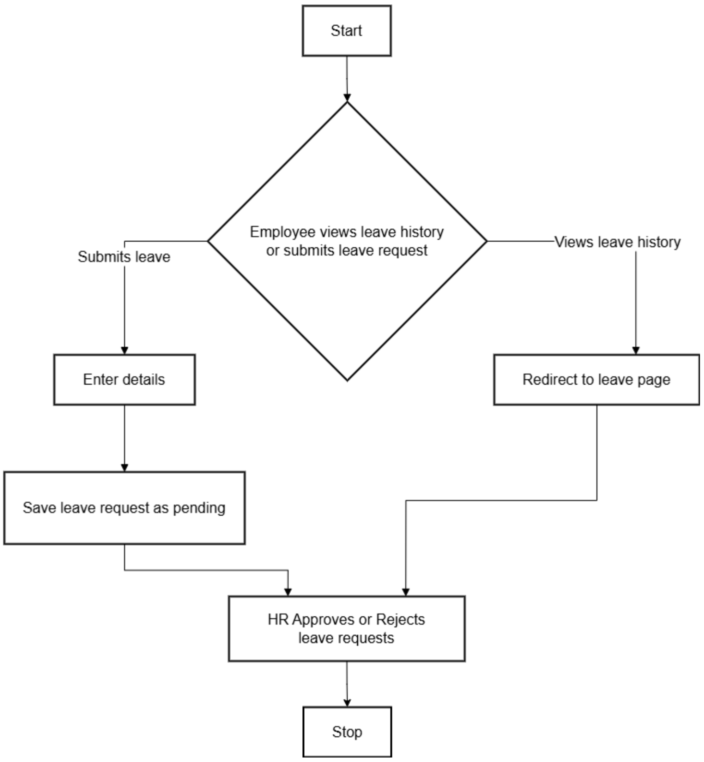

Start
Employee views leave history or submits leave request
Submits leave
Views leave history
Enter details
Redirect to leave page
Save leave request as pending
HR Approves or Rejects leave requests
Stop

Figure 2.17: Leave Management Algorithm Flowchart

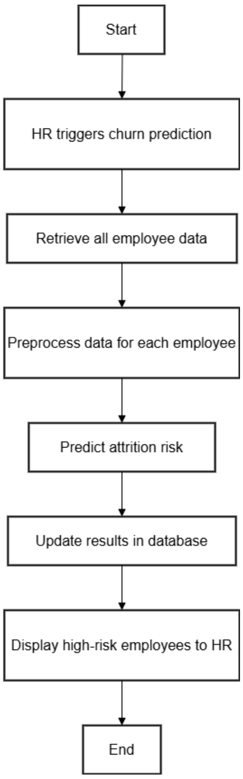

Start
HR triggers churn prediction
Retrieve all employee data
Preprocess data for each employee
Predict attrition risk
Update results in database
Display high-risk employees to HR
End

Figure 2.18: Churn Prediction Module Algorithm Flowchart

2.3 Implementation

Putting the planned designs into action by building the system’s features and integrating all components is a key step. This process turns concepts into a working application that performs the intended tasks and functions.

2.3.1 Tools Used

The Employee Analytics and Churn Prediction System uses a modern set of tools to provide a smooth and efficient user experience. Below is a summary of the main technologies used:

2.3.1.1 Frontend Development

The frontend was made using HTML to build the main parts of the web pages, like forms and tables. Tailwind CSS helped create designs that work well on different devices and look good. JavaScript was used to add interactive parts and make the pages more dynamic. For showing data in charts, Chart.js was used because it makes easy-to-understand graphs.

2.3.1.2 Backend Development

The backend used Flask to handle website routes, user sessions, and APIs. The frontend and backend talk to each other using RESTful APIs to share data smoothly. For security, user login is protected with flask-login and passwords are safely stored using bcrypt hashing.

2.3.1.3 Machine Learning

The prediction model was built with scikit-learn, a Python tool for machine learning. The Random Forest Classifier model was used to predict which employees might leave the company.

2.3.1.4 Database

MySQL was used to store important data like employee details, user accounts, and leave requests. This keeps all information safe and easy to access.

2.3.1.5 Development Tools

The development of the project utilized several essential tools to streamline the workflow. Visual Studio Code (VS Code) served as the primary integrated development environment (IDE) for writing code, managing project files, and debugging, with helpful extensions for Python and Flask. Postman was used to test API endpoints, ensuring proper communication between the frontend and backend components. Git functioned as the version control tool, allowing for efficient tracking of changes and collaboration during development. Additionally, the Command Line Interface (CLI) was used frequently to run Flask servers, manage virtual environments, and execute Python scripts.

2.3.2 Module Description

Modules in code are separate components that group related functions, routes, or logic into organized sections. They help in breaking down a large application into manageable parts, making the code easier to understand, test, and maintain. Using modules improves reusability and allows different team members to work on separate features without conflicts. This structure enhances scalability and supports a clean development workflow.

2.3.2.1 User Login and Authentication Module

This module handles secure access control for both HR and employee users. Users log in through a form where their credentials are verified against stored data in the database. Passwords are hashed using bcrypt to ensure security. After authentication, session data is initialized, including user role and employee ID. Depending on their role, users are redirected to different dashboards. HR users access administrative tools, while employees access their personal dashboard. Invalid credentials or roles trigger appropriate error messages.

2.3.2.2 Job Application Submission Module

This module allows external users (job applicants) to submit applications for open roles. The form collects essential fields such as name, email, date of birth, position applied for, department, education field, work experience, marital status, and other demographics. On submission, the data is stored in the applicants table in the database for HR review. A success message confirms the submission, and applicants are redirected to the application page. This module ensures all applications are organized and ready for future processing or selection.

2.3.2.3 Employee Data Management Module

Designed for HR users, this module supports complete CRUD operations (Create, Read, Update, Delete) on employee records. HR can add new employees by entering details such as name, department, salary, position, and date of joining. When a new employee is added, a login account is automatically generated with a unique email and default password. The system also calculates years at company based on the joining date. Updates to existing employee records trigger a re-evaluation of attrition risk using the churn prediction model. Deleting an employee also removes their user account to maintain referential integrity.

2.3.2.4 Leave Management Module

This module handles leave requests and approvals. Employees can apply for leave through a form, providing start and end dates along with the reason. Leave requests are marked as 'Pending' by default and saved in the database. HR users can view all leave requests and change their status to 'Approved' or 'Rejected'. Upon approval, the system sends an automatic email notification to the employee. Employees can also view their own leave history, while HR maintains oversight of all applications for decision-making and record-keeping.

2.3.2.5 Churn Prediction Module

The Churn Prediction Module is designed to help HR identify employees who are likely to leave the company, allowing for timely interventions. This is achieved using machine learning models trained on the IBM HR Analytics dataset. The five different classification algorithms that were evaluated are: Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes, and Random Forest.

The models were assessed using four primary performance metrics:

1. **Accuracy**: The percentage of total predictions that were correct.
2. **Precision**: The proportion of predicted positives that were actually positive.
3. **Recall**: The proportion of actual positives that were correctly identified.
4. **F1-Score**: The harmonic mean of precision and recall; useful when classes are imbalanced.

Model Performance Comparison:

Table 2.18: Model Performance Comparison

Model	Class	Precision	Recall	F1-Score	Accuracy
Random Forest	0	0.95	0.92	0.94	**0.94**
	1	0.93	0.95	0.94	
Logistic Regression	0	0.62	0.66	0.64	**0.64**
	1	0.66	0.62	0.64	
K-Nearest Neighbors	0	0.73	0.69	0.71	**0.73**
	1	0.72	0.76	0.74	
Support Vector Machine	0	0.60	0.67	0.63	**0.63**
	1	0.65	0.59	0.62	
Naive Bayes	0	0.59	0.73	0.65	**0.62**
	1	0.67	0.52	0.59	

Note: Class 0 indicates no attrition (employee stays), while class 1 indicates attrition (employee likely to leave).

Among all models tested, the Random Forest Classifier achieved the highest performance across all metrics, making it the most reliable model for predicting employee attrition. It balances accuracy, precision, and recall, which is essential for HR decisions where both false positives and false negatives carry consequences.

The Random Forest model considers various features such as job satisfaction, years at company, salary, department, and distance from home. When a new employee is added or an existing record is updated, the system automatically processes the data, makes a churn prediction, and updates the database with the risk status.

These insights are displayed on an interactive dashboard accessible to HR, helping them prioritize retention strategies for high-risk employees.

2.3.3 Coding Standards and Conventions

The project uses simple and clear coding practices to keep the code easy to read and manage. Comments are added in the code to explain what each part does, making it easier for others to understand or update later. Variables and functions have meaningful names, and the code is split into smaller parts (modules) to stay organized. Errors are handled properly to prevent crashes, and important data like passwords is kept safe using encryption. These basic practices help keep the software clean, reliable, and easy to improve.

2.3.4 Security Considerations

Security is a key focus in the design of the system to protect sensitive employee and company information. Passwords are securely stored using bcrypt hashing, which makes it difficult for unauthorized users to access login credentials. The system uses role-based access control to ensure that users only see and interact with data and features appropriate to their role, reducing the risk of unauthorized access. User sessions are carefully managed and ended when users log out.

Additionally, the system validates and cleans all user inputs to protect against common attacks like SQL injection. Data sent between the user's browser and the server is handled securely to prevent interception or alteration. Automated emails, such as leave approval notifications, are sent using secure methods to keep communication safe. These steps help maintain the privacy, accuracy, and availability of the system and its data.

2.3.5 Testing

Testing is the process of evaluating a software system or its components to identify defects and ensure it meets specified requirements. The levels of testing are:

1. **Unit Testing**: Unit testing was conducted to verify the correctness of individual modules, such as user authentication, leave submission, and churn prediction logic. Each function was tested in isolation with expected input and output scenarios using dummy data and mock sessions to ensure accuracy and reliability.
2. **Integration Testing**: Integration testing was carried out to ensure that different components of the system interact correctly. This included testing how the login module integrates with the session management and database, and how the leave request form processes and displays data from both the employee and leave tables.
3. **System Testing**: System testing was performed to validate the complete functionality of the application as a whole. All modules, including authentication, role-based access, dashboards, and database updates, were tested end-to-end to confirm that the system meets the specified requirements.
4. **User Acceptance Testing (UAT)**: User acceptance testing involved testing the system from the end-user's perspective, simulating real-world HR and employee

usage scenarios. Feedback was gathered on usability, role-based access, and workflow correctness to ensure the system aligns with user expectations and business needs.

The following test cases were designed and executed based on the testing approaches outlined above, ensuring each module functions correctly and meets the project's functional requirements.

2.3.5.1 Test Cases

Test cases are detailed steps to follow during testing, showing what to do, what to enter, and what result to expect to make sure everything works right.

Test Case ID	Test Scenario	Test Steps	Expected Output	Actual Output	Status
TC01	User Login with Valid Credentials	1. Open login page 2. Enter valid email and password 3. Click login	User is redirected to dashboard or employee view based on role	User redirected to dashboard	Pass
TC02	User Login with Invalid Password	1. Open login page 2. Enter valid email and invalid password 3. Click login	Error message "Invalid email or password" is displayed	Error message displayed	Pass
TC03	Job Application Submission	1. Open job application form 2. Enter all required details 3. Submit application	Application is submitted and confirmation message is shown	Confirmation message displayed	Pass
TC04	Edit Employee Details	1. Login as HR 2. Go to Employees 3. Edit an employee's details 4. Save changes	Employee details updated and attrition prediction re-calculated	Details updated, prediction re-calculated	Pass
TC05	Delete Employee	1. Login as HR 2. Go to Employees	Employee and their user account are	Employee deleted	Pass

		3. Delete an employee	removed from the database		
TC06	Employee Applies for Leave	1. Login as Employee 2. Go to Leave Request 3. Fill leave form 4. Submit	Leave request is saved and appears in leave request list with status "Pending"	Leave request status "Pending"	Pass
TC07	HR Approves Leave Request	1. Login as HR 2. Go to Leave Requests 3. Approve a pending leave	Leave request status updated to "Approved", employee receives approval email	Status "Approved", email sent	Pass
TC08	Employee Updates Profile Details	1. Login as Employee 2. Go to Edit Profile 3. Update details 4. Save	Profile is updated, attrition prediction re-calculated, success message displayed	Profile updated, prediction re-calculated	Pass
TC09	Employee Submits Feedback (After 1 Year)	1. Login as Employee (>1 year at company) 2. Go to Feedback Portal 3. Submit feedback	Feedback saved, attrition prediction re-calculated, success message displayed	Feedback saved, prediction re-calculated	Pass
TC10	Predict Attrition for All Employees (HR)	1. Login as HR 2. Go to Predict Attrition 3. Trigger prediction	All employees with high attrition risk are listed on the page	High-risk employees listed	Pass

2.3.6 Deployment

This project was run and tested locally on a Lenovo IdeaPad3 laptop using Python and Flask as the main tools. The MySQL database was set up on the same machine to store all data. The application runs through Flask’s built-in development server and is accessed via a web browser on the local computer.

No live or cloud deployment was performed; the system is currently for local use only. For future use, deployment on a web server or cloud platform is recommended to allow access from multiple devices and improve scalability.

CHAPTER III: CONCLUSION AND RECOMMENDATION

3.1 Summary

This project created a web-based system to help manage employee data and leave requests while providing insights through data analysis. It includes a machine learning model to predict which employees may leave, supporting better HR decisions. Overall, the system aims to improve workforce management and reduce employee turnover.

3.1.1 Findings

The Employee Analytics and Churn Prediction System proved to be an effective and practical solution for streamlining HR operations while integrating machine learning for data-driven decision-making. Among the machine learning models tested—Logistic Regression, K-Nearest Neighbors, Support Vector Machine, and Naive Bayes—the Random Forest model delivered the best results. It achieved a high accuracy of 94%, with well-balanced precision and recall, making it the most reliable option for predicting employee attrition. The system analyzes key factors such as job satisfaction, department, years at the company, and salary to determine attrition risk.

The confusion matrix below summarize the performance of the Random Forest model in distinguishing between employees likely to stay (class 0) and those at risk of leaving (class 1):

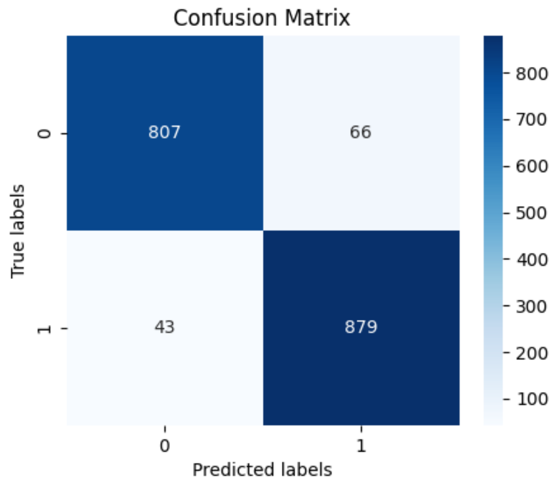


Figure 3.1: Confusion Matrix of Random Forest Model

Beyond prediction, the system includes secure login with role-based access, a job application module, employee record management, and a leave management system with automated email alerts. Visual dashboards offer insights into employee trends, helping HR make informed decisions. This project shows that even with a lightweight tech stack like Flask and MySQL, it's possible to create a functional, scalable HR tool that boosts efficiency and supports strategic planning.

3.1.2 Limitations

While the system fulfills its main objectives, a few limitations were identified during its development and testing. The most important limitations are:

1. The system is currently hosted and tested only in a local environment, which limits its accessibility and scalability without proper deployment on a cloud or production server.
2. The machine learning model, while accurate, is trained on a small dataset, so it may not perform well in larger or more varied organizational settings.
3. Key HR features such as attendance tracking, payroll processing, performance evaluations, and recruitment support are not included in this version of the system.
4. Email notifications use a basic SMTP setup, which may not be efficient or reliable for handling larger volumes or more complex communication needs.
5. The system does not include mobile compatibility, which can affect accessibility for users who prefer managing HR tasks on smartphones or tablets.

These limitations provide areas of improvement that can be addressed in future versions of the system.

3.1.3 Future Enhancements

To enhance the system's functionality and make it more suitable for real-world organizational use, several future improvements are suggested. These upgrades aim to overcome current limitations and extend the system's reach and efficiency:

1. Retrain the churn prediction model using a larger and more diverse dataset to ensure better accuracy and adaptability across various industries and employee roles.
2. Add attendance tracking capabilities by integrating with biometric or IoT-based devices commonly used in modern workplaces.
3. Deploy the system on cloud platforms like Heroku or AWS to enable remote access and support scalability for larger organizations.
4. Create a mobile-responsive version of the system, particularly for employees, to make tasks like checking profiles and applying for leave more convenient.
5. Expand the system to include payroll processing, attendance monitoring, advanced performance evaluation tools, and an advanced applicant tracking system (ATS).

These enhancements will help the system evolve into a more comprehensive HR solution suitable for a wide range of organizational needs.

3.1.4 Lessons Learned

The development of the Employee Analytics and Churn Prediction System provided valuable insights into both the technical and practical aspects of software creation. One important lesson was the need to understand the different requirements of HR personnel and employees, and to design the system in a way that offers each group appropriate and efficient tools.

The project also demonstrated the critical role of clean and well-prepared data in building effective machine learning models. Careful feature selection and preprocessing contributed significantly to achieving strong prediction accuracy.

Integrating various components such as databases, APIs, and machine learning modules highlighted the importance of modular and maintainable code design.

Experience with Flask proved essential, particularly in understanding core features like routing, session management, and API development, which are fundamental for building secure and scalable web applications.

Overall, the project underscored the importance of balancing functionality with usability. Developing simple, intuitive interfaces alongside reliable back-end operations was key to creating a practical and efficient system. Furthermore, maintaining clear documentation, adhering to consistent coding standards, and thorough planning were crucial factors in ensuring smooth development and facilitating future scalability.

3.2 Conclusion

The Employee Analytics and Churn Prediction System successfully demonstrates how combining data analytics with HR management can enhance organizational efficiency and decision-making. By providing tools for employee data management, leave handling, and predictive insights into employee attrition, the system supports HR teams in addressing workforce challenges proactively. The integration of machine learning for churn prediction, alongside intuitive dashboards and role-based access, ensures that both HR personnel and employees benefit from a streamlined and secure experience.

Despite some limitations, such as dataset constraints and the absence of certain features like payroll management and attendance integration, the project lays a strong foundation for future enhancements. Overall, the system exemplifies how modern technology can be leveraged to improve productivity, reduce costs, and foster data-driven human resource practices in organizations of varying sizes.

From an academic perspective, this project contributes to knowledge in computer science by applying machine learning techniques within a real-world web application framework, demonstrating effective integration of backend development, database management, and data science. In terms of information management, it highlights the importance of organized, role-based data access and analytics in improving business processes, offering practical insights into how data-driven systems can support decision-making and operational efficiency in human resource management.

3.3 Recommendation

The Employee Analytics and Churn Prediction System has strong potential to improve HR processes through data insights. To enhance its value and usability, the following concise recommendations are proposed:

1. Use a larger, more diverse dataset to improve churn prediction accuracy.
2. Add payroll and attendance tracking for a more complete HR tool.

3. Deploy on cloud platforms for better scalability and remote access.
4. Create a mobile-friendly interface for easier employee access.
5. Maintain clear documentation and clean code to support future updates.
6. Include advanced analytics and customizable reports for deeper insights.

Applying these improvements will make the system more powerful, adaptable, and suitable for real organizational needs, helping HR teams work smarter and keep employees engaged.