Multitenant Leave System

Akshat Dhanuka[1], Aayush Bhaskarwar[2], Ziyaad Parkar[3], S.P. Shintre[4]
UG Scholar, Dept. of Computer Engineering, Pune Institute of Computer Technology, Pune, India[1-3]
Asst. Professor, Dept. of Computer Engineering, Pune Institute of Computer Technology, Pune, India[4]

Abstract: ***Efficient leave management is critical for maintaining team productivity and meeting project deadlines in modern organizations, particularly when multiple teams work collaboratively. This paper proposes a dynamic, AI-driven team leave management system designed to streamline leave requests and approvals while minimizing disruptions to workflow. The system integrates a chatbot, utilizing natural language processing (NLP) to convert employee inputs—such as "Who is on leave today?" or "When can I next take leave?"—into SQL queries, providing instant responses and actionable insights. By incorporating cross-team dependency management, the system ensures that leave scheduling is optimized across interdependent teams, preventing bottlenecks and missed deadlines. Furthermore, the system tracks utilized and non-utilized leaves, offering data-driven suggestions for optimal leave timing, based on usage trends throughout the year. This paper also discussesthe technical architecture, the integration of the GPT API for NLP, and the role of a backend database in automating leave management. The proposed solution aims to enhance team efficiency by offering a seamless, intelligent interface for managing leaves without affecting overall project timelines.***

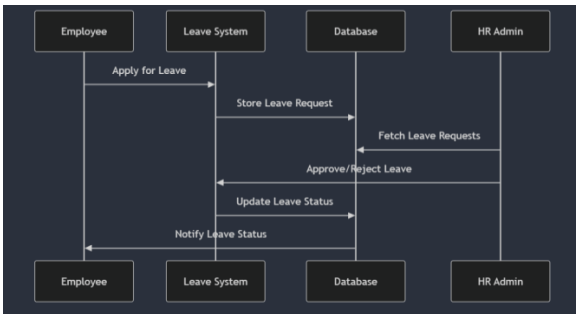
Keywords: leave management system, team collaboration, chatbot integration, NLP, GPT API, SQL query automation, cross-team dependencies, employee productivity, workflow optimization, data-driven leave suggestions, HR automation, interdependent teams, organizational efficiency.

I. INTRODUCTION

In today's fast-paced corporate environment, efficient management of employee leave is critical for ensuring consistent team productivity and meeting project deadlines. However, traditional leave management systems often overlook the complexities of modern, interdependent teams, where the absence of one or more key members can have a cascading effect on workflows. As organizations increasingly rely on team-based structures, the need for a robust, intelligent system that manages leave without disrupting productivity has become essential.

This paper proposes a next-generation team leave managementsystem, specifically designed to address the intricate challenges posed by team dependencies and collaborative projects. By integrating advanced artificial intelligence and natural language processing (NLP) through the GPT API, the system allows employees to easily manage leave requests and gain real-time insights into team availability. SQL queries are processed and the system can also efficiently analyze patterns across the organization, providing data-driven suggestions for optimal leave periods, reducing overlap in critical teams. Furthermore, it handles complex cross-team dependencies, ensuring that when multiple teams collaborate, leave schedules are adjusted to prevent workflow bottlenecks. By automating leave approval processes and ensuring data consistency, the proposed system enhances organizational efficiency and promotes balanced team workload distribution.

Through this research, we explore the development, implementation, and potential impacts of such a system on team management, focusing on improving productivity, employee satisfaction, and operational resilience. The solution highlights the importance of innovative HR technologies in adapting to the evolving needs of modern workplaces.



1.Diagram of structure of leave management system

A. Existing Systems:

Modern leave management systems leverage a range of advanced technologies and algorithms to streamline leave scheduling, optimize team availability, and enhance workforce efficiency. These

systems utilize dependency management algorithms to ensure that leave approvals account for project requirements and team dependencies, preventing workflow disruptions. Machine learning algorithms analyze historical leave data to predict peak leave periods, offering proactive suggestions to employees for optimal leave planning. Additionally, leave balancing algorithms ensure that critical team members are not absent simultaneously, preserving team productivity. Natural language processing (NLP) facilitates chatbot integration, allowing employees to request leave or inquire about team availability through conversational inputs.

Some of the most widely used techniques in this field are –

1. Leave Tracking and Automation System:

Most modern leave management systems implement leave tracking and automation tools that digitize the process of submitting, reviewing, and approving leave requests. These systems often feature dashboards for managers and employees to monitor leave balances, track upcoming leaves, and ensure that leave approvals are processed efficiently. Automation ensures that leave requests are routed to the appropriate supervisors without manual intervention, speeding up approval times and minimizing human error.

2. Natural Language Processing (NLP):

NLP technologies are increasingly integrated into leave management systems to enhance user interaction. By enabling chatbots, employees can make leave requests or inquire about their leave balance through simple conversational interfaces. The chatbot interprets these inputs using NLP and interacts with the underlying database to process leave requests or provide responses in real-time, improving accessibility and reducing the need for manual data entry.

3. Predictive Analysis for Work-Force Planning:

Predictive analytics plays a critical role in optimizing leave management. These systems use historical leave data and patterns to forecast peak leave periods and make recommendations to employees and managers for better leave scheduling. By analyzing trends, the system ensures that there is always sufficient staff coverage to maintain productivity during critical business periods, thereby reducing the risk of disruption due to unplanned leaves.

B. Proposed System

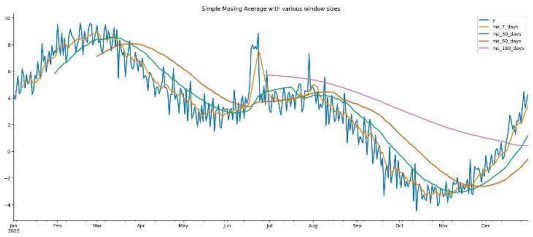
The proposed leave management system aims to enhance organizational efficiency and employee satisfaction through the integration of advanced technologies and data-driven insights. The system is designed to facilitate seamless leave applications, approvals, and management processes while ensuring transparency and accountability across various departments. By incorporating an AI chatbot, the system will allow employees to interact with the database using natural language, converting user queries into SQL commands. This feature will not only streamline the leave application process but also provide instant responses to employee inquiries regarding leave policies, balances, and application status, thereby fostering a more responsive and engaging work environment.

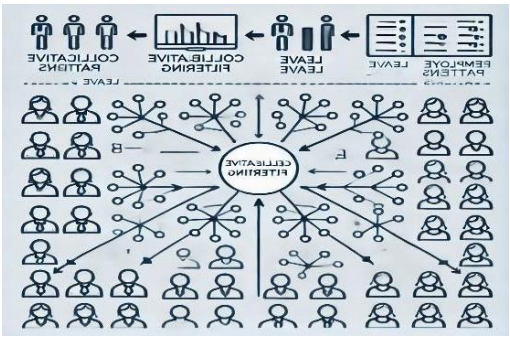
Additionally, the leave management system will include a robust cross-team management module, enabling multiple teams to coordinate their leave schedules effectively. This module is essential for organizations where projects often involve collaborative efforts from various departments. By providing a centralized platform for tracking leave requests and approvals, the system will help minimize disruptions and ensure that critical tasks are managed efficiently. The integration of a calendar view will allow team leaders to visualize team availability, enabling better planning and resource allocation. Consequently, this feature will enhance project continuity and reduce the likelihood of workflow interruptions due to unplanned absences.

To further empower management decisions, the system will leverage data-driven insights through an analytics module. By analyzing historical leave data, the system can identify trends and patterns, such as peak leave periods and individual employee leave frequencies. This information will enable managers to anticipate potential staffing challenges and proactively address them by suggesting optimal leave dates to team members. Such predictive capabilities will not only enhance operational efficiency but also contribute to a positive work culture by allowing employees to plan their leave in a manner that minimizes disruptions to team productivity.

The system will also incorporate advanced algorithms, including Time Series Analysis using

Moving Averages and Collaborative Filtering, to improve its performance and usability. By employing Natural Language Processing (NLP) techniques, such as Transformer Models (BERT/GPT-based), the chatbot will become increasingly adept at understanding user queries and providing relevant information. Additionally, Linear Regression will be utilized to analyze leave frequency as an independent variable, allowing the system to generate personalized insights for individual employees based on their historical leave patterns. This comprehensive approachensures that the proposed leave management system not only addresses current organizational needs but also evolves to meet future challenges, ultimately promoting a harmonious balance between employee well-being and business objectives.





2.Diagrams depicting primary used algorithms

Moving Averages for Time Series Analysis: Analyze leave trends over time by smoothing fluctuations in daily, weekly or monthly leaves requested. It will help HR team to predict leave pattern in the future and use resource more effectively.

Collaborative Filtering Algorithm: Collaborative filtering algorithm will help us to predict that how a particular employee can leave based on the fact of similar behaving others. This will enable us to crosscheck leave approvals sincethe system can alert if two colleagues are requesting a break atsame time.

II. METHODOLOGY

A. Data Sets:

Datasets form the core of the leave management portal, enabling analysis of employee leaves, behavior patterns, and productivity metrics. The dataset consists of structured data collected from user interactions with the portal, such as employee information, leave types, leave balances, and application statuses.

For this system, the dataset is centered around the entity called 'LeaveRequest,' which includes relevant fields such as employee ID, leave type, leave duration, leave reason, and leave status (approved, pending, or rejected). Additionally, the dataset captures time stamps of leave applications, approvals, and modifications, which form the basis for analyzing trends in leave requests and the impact on employee productivity.

The importance of having accurate, structured data cannot be overstated in providing valuable insights. This dataset provides the essential information for analysis, chatbot-driven queries, and recommendation generation, ensuring that the system functions in real time and meets user needs effectively.

B. Prediction and Classification:

In the context of leave management, predictive analysis and classification techniques are essential for optimizing leave allocations and understanding employee patterns. These techniques leverage machine learning algorithms to identify patterns in historical leave data, helping to anticipate peak leave periods and provide HR with actionable insights.

The system employs classification methods to group employees based on their leave-taking behavior (e.g., frequent, infrequent, or seasonal leave takers). This helps administrators plan resources and forecast potential workforce shortages during critical times. Machine learning models, such as decision trees, are utilized to predict leave trends based on historical data, allowing the system to offer recommendations on staffing needs and productivity adjustments during high-demand periods.

Furthermore, classification algorithms assist in generating personalized leave suggestions based on each employee's past leave patterns and productivity metrics. By analyzing leave history, the system provides recommendations for optimal leave timing, balancing both employee needs and company resource management.

III. PROPOSED WORK

The leave management portal incorporates innovative features such as a chatbot-driven query system and a recommendation engine. The chatbot interprets natural language inputs from users (such as "How many leaves do I have left?" or "Can I apply for sick leave next week?") and converts these into structured SQL queries. This interaction simplifies the user experience, enabling employees and administrators to obtain the information they need without having to interact directly with the backend database.

Chatbot SQL Query Generation:

The chatbot uses natural language processing (NLP) to understand user queries, and transforms them into appropriate SQL commands. For example, if an employee asks, "Show my pending leave applications," the chatbot generates a query like:

```
SELECT * FROM leave_requests WHERE
employee_id
= 'X' AND status = 'pending';
```

The system retrieves the relevant data and presents it back to the user in a human-readable format. This process is automated and seamless, enhancing the efficiency of the leave management process.

Algorithms involved:

1. *Support Vector Machine (SVM):*

Support Vector Machines (SVM) are a popular supervised learning algorithm used for classification tasks. In our leave management system, SVM plays a critical role in intent classification, where the system identifies the purpose of a user's natural language query— such as retrieving data, updating records, or performing analysis. SVM operates by finding the optimal hyperplane that best separates the data points belonging to different classes, maximizing the margin between them. In our case, the input features for SVM are derived from the processed natural language query after pre-processing steps like tokenization, lemmatization, and stop-word removal.

- Tokenization: This step involves breaking down the user'sinput into individual words or "tokens" to simplify analysis.
- Lemmatization: Here, words are reduced to their base or root form (e.g., "running" becomes "run"), allowing for a more consistent representation of similar terms.
- Stop-word Removal: Commonly used words that do not contribute much meaning, like "the" or "is," are filtered out to focus on significant terms.

These steps convert the user query into a structured input suitable for SVM, which then categorizes the query intent based on a set of pre- defined categories. The trained SVM model ensures high accuracy in detecting user intents, even with varying query phrasing.

2. *Cosine Similarity:*

Cosine Similarity is a measure used to determine the similarity between two non-zero vectors by calculating the cosine of the angle between them. In our system, it is leveraged during the entity recognition and matching phase. After identifying the user's intent, the query terms are compared with the database schema components, such as table names, column fields, and predefined values.

- The text is transformed into numerical vectors using TF- IDF (Term Frequency-Inverse Document Frequency) or word embedding techniques like Word2Vec or BERT.
- Cosine Similarity then measures the closeness between the query terms and the schema terms, assisting in accurately mapping natural language expressions to database fields.

By utilizing Cosine Similarity, our system can handle synonymous terms and approximate matches, ensuring a higher degree offlexibility in user queries.

3. *Natural Language Processing (NLP) Techniques:*

To accurately interpret user queries and convert them into SQLdatabase queries, several NLP techniques are employed:

- Tokenization: This process splits the user's query into manageable tokens (words or phrases), facilitating deeper analysis.
- Lemmatization and Stemming: Words are standardized to their root forms, aiding in accurate intent recognition and entity extraction.
- Named Entity Recognition (NER): This technique identifies and categorizes entities in the query, such as names, dates, or specific column fields, improving database mapping.

- Dependency Parsing: The grammatical structure of the query is analyzed to determine the relationship between words, assisting in complex query constructions.

By combining these NLP techniques with SVM for intent recognition and Cosine Similarity for entity matching, our system achieves robust and precise natural language understanding, enabling efficient conversion from user queries to SQL database commands.

Data Analysis and Recommendations:

The system uses predictive models and recommendation algorithms to offer suggestions regarding leave requests. For example, by analyzing historical data, the system might suggest that employees avoid applying for leave during busy periods or recommend optimal times based on past behavior.

Moreover, the system monitors employee productivity alongside their leave patterns. By tracking the number of leaves taken and correlating them with productivity data (e.g., performance reviews, task completion rates), the system can provide HR insights into how leave frequency affects overall work output.

Algorithms Involved:

1. Linear Regression:

Linear Regression is a fundamental statistical method used for modeling the relationship between a dependent variable and one or more independent variables. It assumes a linear relationship, which means that the change in the dependent variable is proportional to the change in the independent variables. In the context of the proposed leave management system, Linear Regression can be employed to quantify the impact of various leave patterns on employee productivity.

The general equation of a linear regression model can be represented as:

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \epsilon$ Where:

- Y is the dependent variable (e.g., productivity score).
- $X_1, X_2, \ldots, X_n$ are the independent variables (e.g., frequency of leave, duration of leave).
- β_0 is the intercept, and $\beta_1, \ldots, \beta_n$ are the coefficients representing the weight of each independent variable. ε is the error term.

The advantages of Linear Regression include its simplicity, interpretability, and ease of implementation. It allows for the identification of significant predictors of productivity, which can guide HR decisions regarding leave policies. However, it is crucial to note that Linear Regression may not effectively capture complex, non-linear relationships in the data, which can limit its applicability in certain scenarios.

2. Random Forest Regression:

Random Forest Regression is an ensemble learning technique that combines multiple decision trees to improve predictive accuracy and control over-fitting. Unlike Linear Regression, which assumes a linear relationship, Random Forest can model complex interactions and non-linear relationships among variables, making it particularly suitable for analyzing employee leave patterns and their effects on productivity.

In a Random Forest model, a multitude of decision trees is constructed using random subsets of the data and features. Each tree contributes to the final prediction by averaging the outputs from all trees in the ensemble. This approach reduces the risk of overfitting and enhances model robustness.

The primary steps involved in Random Forest Regression include:

1. Bootstrap Aggregating: Randomly sampling data with replacement to create multiple training datasets.
2. Tree Construction: Building a decision tree for each dataset, selecting a random subset of features at each split.
3. Prediction: Aggregating the predictions from all trees to produce the final output.

The advantages of Random Forest Regression lie in its ability to handle large datasets with high dimensionality and its resilience to noise and outliers. It also provides valuable insights into feature importance, allowing HR managers to understand which leave patterns most significantly impact productivity. However, Random Forest models can be computationally intensive and less interpretable than simpler models like Linear Regression.

Relevance to Employee Productivity:

In the proposed system, employee leave patterns and productivity are analyzed through a framework of weighted metrics rather than traditional distance

measures. By tracking factors like leave frequency, leave duration, and their correlation with productivity metrics, the system calculates a weighted score for each employee.

This score represents the cumulative effect of leave-taking behavior on overall productivity. For example, an employee taking frequent but short leaves might have a different impact on productivity compared to someone taking longer, infrequent leaves. The system considers these various factors, assigning weights to each metric and calculating an overall productivity impact score.

Algorithms Involved:

1. Decision Trees:

Decision Trees are a popular machine learning algorithm used for both classification and regression tasks. They represent decisions and their consequences in a tree-like structure, where each node corresponds to a feature, each branch represents a decision rule, and each leaf node signifies the final output.

The Decision Tree algorithm employs a method called recursive partitioning. It evaluates various attributes to identify the optimal split that minimizes impurity, commonly measured using:

- Gini Index: Assesses the impurity of a dataset; lowervalues indicate better splits.
- Entropy: Measures disorder; splits that maximizeinformation gain are preferred.
- Mean Squared Error (MSE): Used for regression treesto minimize variance in target values.

The tree-building process continues until a stopping criterion is met, such as a predefined depth or minimum sample size at leaf nodes.

Advantages:

- Interpretability: Decision Trees are easy to visualize, making the decision-making process transparent.
- Handling Mixed Data Types: They can process both categorical and numerical data without requiring normalization.

Limitations:

- Overfitting: They are prone to overfitting, which can beaddressed through pruning techniques.
- Instability: Minor changes in the data can significantlyalter the tree structure.

Higher Order Structure of Database:

1. employees Table

Stores details of all employees, including their roles and leavebalances.

Column	Type	Description
employee_id	INT	Unique identifier f oreach employee
name	VARCHAR(255)	Employee's full name
role	VARCHAR(255)	Job role (e.g., SoftwareEngineer, Manager, etc.)
paid_leaves_remaining	INT	Number of paid leavesleft for the employee

2. employee_leaves Table

Tracks leave records for employees, capturing the type and date ofleaves.

Column	Type	Description
employee_id	INT	Reference to the employee's ID
leave_type	VARCHAR(255)	Type of leave (e.g., Paid, Unpaid,Sick, etc.)
leave_date	DATE	Date on which the leave was taken

IV. RESULT AND ANALYSIS

In the test case, the system tracks multiple employees' leave records, including the type of leave, remaining quotas, leave dates, and approval statuses. When an employee submits a leave request, the system dynamically updates the data, ensuring accurate records whileadhering to organizational policies. The data is stored in a structured format, allowing efficient queries. Employees can check their leave balances or submit requests, while managers can view their team's leave status. This ensures real-time leave management and smooth workforce planning.

1. User Query:

"How many leaves from my paid leaves are left?"
Query Conversion into SQL:
SELECT paid_leaves_remainingFROM leave_records
WHERE employee_id = 'USER_ID';

Query Result from Sample Table:
Query returns 5 paid leaves remaining.

Overview of Conversion

The user query is first parsed to identify the type of leave (paid leave) and the specific data requested (remaining leaves). The system translates this into an SQL query by selecting the paid_leaves_remaining column from the leave_records table, filtering the results by the employee's unique identifier (employee_id)s Using the proposed system by implementing a chatbot API integration, we successfully convert a user's query into a valid database query by analysing the keyword and returning a query based of the ranking done by the algorithm.

2. Linear Regression:

Scenario:
The system uses historical data to analyze the relationship between employee leave frequency and their productivity score. In this case, productivity is measured through metrics like task completion rate, performance reviews, and overall team contribution.

User Query:

"How does my leave frequency affect my productivity score?"Query Conversion and Regression Analysis:

1. SQL Query:

```
SELECT leave_frequency, productivity_score
FROM employee_performance
WHERE employee_id = 'USER_ID';
```

Query Result from Sample Table:

leave_frequency	productivity_score
10	85
15	80
5	90
8	88

2. Overview of Linear Regression Analysis:
   - The system retrieves leave_frequency and productivity_score data for the employee.
   - It uses the regression equation Y=β0+β1X+εY = $\beta_0 + \beta_1 X + \varepsilon$Y=β0+β1X+εwhere:
     - YYY = Productivity Score
     - XXX = Leave Frequency
     - β0β₀β0 and β1β₁β1 are coefficients derived from historical data.

3. Result: The Linear Regression model predicts a slight negative correlation between the frequency of leaves and the productivity score, with a coefficient indicating that productivity drops by 2 points for every 5 additional leaves taken.

3. Random Forest Regression:

The system analyzes complex patterns between various factors—such as the duration of leave, leave type, and employee productivity score—to predict the impact of leave duration on future performance.

User Query:

"What's the expected impact on my performance if I take a 10-day leave?"

Query Conversion and Random Forest Analysis:

1. SQL Query:

```
SELECT leave_duration, leave_type,
productivity_scoreFROM employee_leaves
JOIN employee_performance

ON employee_leaves.employee_id =
employee_performance.employee_id

WHERE employee_id = 'USER_ID';
```

Query Result from Sample Table:

leave_duration	leave_type	productivity_score
5	Paid	87
12	Unpaid	75
7	Sick	80
3	Paid	90

2. Overview of Random Forest Analysis:
   - The system retrieves leave_duration, leave_type, and productivity_score data.
   - A Random Forest model is applied, using historical data to predict the impact of a future 10-day leave.
   - Decision Trees within the Random Forest are created based on factors like leave_duration and leave_type, which determine how different types and durations affect productivity_score.

3. Result: The model predicts a moderate decline in productivity if the user takes a 10-day paid leave,

estimating a 5% decrease in productivity due to the length of absence, considering the specific pattern of past leave behavior.

V. CONCLUSION

The AI-driven leave management system presented in this study provides a streamlined solution to managing employee leave while aligning with organizational goals. By leveraging historical data and predictive analytics, the system enables smarter leave approvals that mitigate the risks of understaffing. It enhances decision-making processes for HR teams through features like chatbot-driven user queries and machine learning models, which translate complex data into actionable insights. The system's ability to integrate both NLP and predictive analytics supports efficient workforce planning, reduces the reliance on manual processes, and promotes a more transparent and well-balanced work environment. Ultimately, it fosters better employee satisfaction while maintaining organizational productivity, offering a scalable solution adaptable to various business needs.

VI. REFERENCES

[1] Samuel Mayowa, Samuel and Temitope John, "Design and Implementation of a Web Based Leave Management System" in International Journal of Computer Applications Technology and Research Volume 11–Issue 04, 123-144, 2022

[2] Ms. Snehal Choudhari and Prof. Tarun Yengantiwar,"Review on Leave Management Systems " in International Journal of Advanced Research in Computer and Communication Engineering.

[3] Rajan Deshmukh, Saurav Avhad, Adarsh Maid and Abid Dhankwala, "Leave Management System," in Journal of Emerging Technologies and Innovative Research 2014.

[4] Asikhia Olalekan U.and Omojola, Olugbenga I., "Management Planning and Employee Engagement: A Review of Literature," in The International Journalff Business& Management.

[5] Mary Safowah Akom, Charles Obeng-Sarpong, Florence Enyonam Aflakpui, Smart A. Sarpong, "Exploring Leave Management Practices and Relationship with Performance of Administrative Staff: Evidence from a Tertiary Institution in Ghana," in Universal Journal of Management 2021.

[6] Adamu A., "Employee Leave Management System," in FUDMAJournal of Sciences (FJS).

[7] A. MohanaPriya, G. Shyamala, R. Dharshini, “Mobile HRM for Online Leave Management System”, in International Journal of Computer Science and Mobile Computing 2017.

[8] Peter Zupancic and Pance Panov, "Predicting Employee Absence from Historical Absence Profiles with Machine Learning," in Applied Sciences (Appl. Sci)

[9] Sowjna. K. Shetty., Pruthvi R. Raju., Garima Jha and Sai Sagar K.R, “Cloud Based Employee Leave Management System” in International Journal of Innovative Science and Research Technology 2017

[10] Lavish Sahu, Deepak Jingar, Jay Sahu, Rakshit Kothari, “Chatbot Based Helpdesk for Government Employee and Department” in International Advanced Research Journal in Science, Engineering and Technology 2024