# Monitoring Migrant Labors by Using Decision Tree

Mrs. Radhika D AP[1], Ms. Ramya S[2], Ms. Rathipriya G[3], Ms. Rithika K[4]
*[1,2,3,4] Department of Computer, Science and Engineering Vivekanandha College of Engineering for Women*
*Namakkal, India*

## Abstract

The monitoring of migrant laborers is a significant challenge in various industries. The lack of structured data collection and performance analysis often leads to inefficiencies, wage disputes, and insecure working conditions. Decision Tree algorithms provide a powerful analytical tool for evaluating workers' performance based on collected data such as work hours and task completion status. This literature survey explores previous research on labor monitoring, workforce tracking, and decision tree applications in labor analytics. It explores various research studies and methodologies related to monitoring migrant laborers using Decision Tree algorithms. It highlights key challenges faced by migrant workers, existing monitoring systems, and how decision tree models can improve workforce management.

## Introduction

Traditional monitoring methods include biometric attendance systems, workforce management tools, and GPS-based tracking. These approaches help track workers but lack real-time performance analysis. Decision Trees are used to predict worker performance based on work hours, task completion rates, and uploaded work status images. It provides graphical insights into efficiency and helps administrators make data-driven decisions. The monitoring of migrant laborers has traditionally relied on manual methods and basic automated systems. Biometric attendance systems, such as fingerprint or facial recognition technologies, have been widely used to track workers' presence. While these systems offer accuracy in attendance management, they fail to provide insights into workers' productivity or efficiency. GPS-based tracking has also been employed, especially in outdoor industries, to monitor workers' locations in real time. Although this method helps ensure that workers are on-site, it raises privacy concerns and does not capture performancerelated data. Additionally, workforce management tools, which integrate various tracking features, are commonly used to monitor work hours, task assignments, and reporting. However, these tools often lack advanced data analysis capabilities to assess worker efficiency comprehensively. SQLite for instance, offers a scalable.
backend for storing worker-related data, including attendance, issue reporting, and location tracking.

Delayed wages, workplace safety issues, and job insecurity are common problems. Many workers lack digital employment records, leading to job instability. Manual record-keeping is prone to errors. Biometric attendance tracks presence but not actual work. Mobile tracking apps lack in-depth performance analysis. The project integrates Decision Tree-based analytics to track work hours, analyze task completion, and visualize worker efficiency through graphs. It also includes real-time issue reporting and an automated system for monitoring work status. The proposed model enhances transparency, ensures real-time tracking, and improves labor management efficiency by using data-driven decisionmaking. This literature survey provides a strong foundation for implementing a Decision Tree-based migrant labor monitoring system to ensure fair work conditions and efficient workforce tracking. The project "Monitoring Migrant Laborers Using Decision Tree" is designed to provide an efficient and structured system for tracking and analyzing migrant workers' activities. The admin registers workers, monitors work hours, and evaluates performance based on work status updates. Workers must enable their mobile location to log in and upload work status images, which are analyzed using a Decision Tree Algorithm . When combined with decision tree algorithms, such platforms enable dynamic assessment of worker performance and immediate identification of inefficiencies or issues. This fusion of mobile data collection and machine learning not only improves transparency but also empowers administrators to respond swiftly to labor concerns, ensuring fair treatment and optimized resource allocation. Decision Tree algorithms are supervised machine learning models used for classification and regression tasks. They operate by splitting data into subsets based on specific features, creating a tree-like structure of decisions and outcomes. The primary components of a Decision Tree include nodes, which represent decision points, branches, which indicate decision rules, and leaves, which represent final outcomes or classifications. These algorithms are valued for their simplicity, interpretability, and ability to handle both numerical and categorical data effectively.

proper safety measures results in frequent workplace accidents, posing serious risks to their health and well-being. Moreover, many workers are unable to track their employment history due to the absence of digital records, leading to job insecurities and

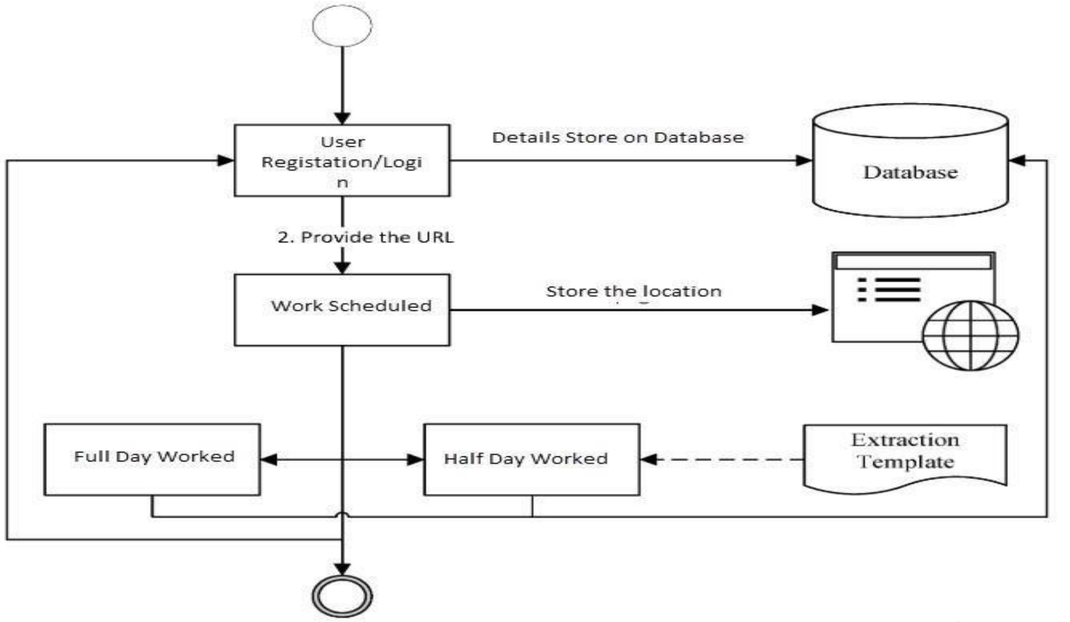


Figure 1. Architecture Diagram

Decision Tree is widely used in performance evaluation due to its ability to classify data and predict trends. Application in Workforce Productivity: Studies have demonstrated that Decision Tree models can effectively analyze work performance based on time tracking and task completion rates. Decision Tree-based models have been utilized in predicting employee performance using historical data, work logs, work status analysing and task completion records. Decision Trees are frequently used to visualize performance analytics in structured formats, enhancing transparency in workforce management. The Decision Tree algorithm analyzes the workers' uploaded work status daily, in the morning and afternoon, to determine their performance trends. The algorithm provides a graphical representation indicating whether a worker's performance is high or low based on task completion and consistency. Many researchers have highlighted the difficulties faced by migrant workers such as "Research indicates that a lack of proper safety measures results in frequent workplace accidents. Many workers are unable to track their employment history due to the absence of digital records, leading to job insecurities". Despite these advancements, migrant workers continue to face significant challenges in the workplace. Research indicates that a lack of difficulties in accessing benefits or legal protections.

Our project aims to address the above limitations by integrating a Decision Tree-based analytical model into a labor monitoring system. The key features include The system ensures real-time work hour monitoring based on mobile GPS data. Workers upload images and descriptions of their tasks in the morning and afternoon, which are analyzed using a Decision Tree model to assess efficiency. The system generates graphical reports indicating whether the worker's performance is high or low. Workers can report grievances related to wage delays or unsafe conditions, and the admin takes appropriate action. The system generates performance graphs to visualize worker efficiency trends. Decision Trees are frequently used to visualize performance analytics in structured.

## Methodology

Mobile applications have been widely used for workforce management in different domains GPS- Enabled Attendance Tracking is Research that
demonstrated a GPS-based mobile application for employee attendance verification, ensuring that employees mark attendance only from their workplace. Mobile-Based Grievance Reporting is developed an Android-based complaint management system where workers could report workplace issues directly from their smartphones.

Real-time Productivity Monitoring Applications is Applications such as use mobile-based tracking systems to analyze workers' daily performance and work engagement. For developing your Android mobile application using Java and SQLite, several tools and libraries are required in Android Studio to implement various functionalities. The primary development tools include Android Studio, which serves as the official IDE for Android app development, Java as the core programming language, and Gradle for managing dependencies and building the project efficiently. Since you are using SQLite tools will be essential, status, salary details, and reported issues, SQLite is to send real-time notifications such as notifying workers when their issues are resolved, and SQLite Storage to handle the storage of images like profile pictures and work status images. For designing the user interface, XML Layouts are used to create app screens.

- The Recycler View component is necessary for displaying dynamic lists, such as the list of employees, work status history, or reported issues.
- To ensure smooth background processing, Work Manager is useful for scheduling background tasks like syncing work.

This section outlines the methodology for developing the Monitoring Migrating Laborers Using Decision-Free Algorithms mobile application. The approach covers the design and implementation of two modules the Admin Module and the User Module, as well as the integration of key technologies such as Android Studio, SQLite, Java, and Decision-Free Algorithms like decision trees. The objective is to create a platform that monitors laborers’ working hours, work performance, location, and issues, while providing an intuitive interface for both admins and users migrant workers.

- For handling image and media processing, the Camera X API enables workers to capture profile pictures and work- related images, while Glide or Picasso are image-loading libraries that efficiently display images stored .
- Regarding security, SQLite Security Rules will be configured to restrict unauthorized access to the SQLite database, ensuring that only verified users can access and modify relevant data. Additionally, various Android permissions will be required, including ACCESS FINE
LOCATION for worker location tracking, CAMERA for capturing profile images, and STORAGE for saving and retrieving images.

By integrating all these tools and libraries effectively, your Android application will efficiently manage and monitor migrant laborers, providing a robust solution for tracking work hours, evaluating performance, addressing issues, and ensuring transparency in salary distribution. The Recycler View component is necessary for displaying dynamic lists, such as the list of employees, work status history, or reported issues.

## Existing system

Integration of Multiple Features in a Single System is Existing works often focus on individual aspects such as attendance tracking, complaint management, or performance evaluation, but not a holistic system integrating all these functionalities. physical infrastructure, which can be costly and impractical for migrant laborers working in remote areas or temporary worksites. The use of technology in managing migrant laborers and ensuring the proper monitoring of their performance has seen significant advancements in recent years. With the rise of mobile applications and cloudbased databases, there has been a marked improvement in how labor is tracked, evaluated, and optimized. Migrant laborers, particularly in sectors like construction, agriculture, and manufacturing, often face challenges such as poor working conditions, delayed wages, and insufficient supervision. Mobile applications provide an effective platform for monitoring these workers, reporting issues, and ensuring better working environments. The existing system for monitoring and managing migrant laborers primarily relies on manual record-keeping, biometric attendance systems, and basic mobile communication methods, which often lead to inefficiencies, inaccuracies, and delays in decisionmaking. Many organizations still use paper- based registers or spreadsheets to log attendance and track working hours, making it difficult to maintain real- time data access, prevent fraud, and ensure accurate wage calculation. In traditional workforce management, biometric attendance systems such as fingerprint scanners or RFID-based check-ins are commonly used however, these methods are not location-aware and can be easily manipulated by proxy attendance, where one worker clocks in for another. Additionally, biometric systems require physical infrastructure, which can be costly and impractical for migrant laborers working in remote areas or temporary worksites.

In terms of worker issue reporting, many organizations depend on verbal complaints, phone calls, or in-person meetings, which are inefficient and often result in delayed responses or lack of proper documentation. Workers facing issues such as wage delays, unsafe working conditions, lack of food and water, or housing problems have limited ways to report these concerns effectively. Due to the lack of a centralized system for grievance handling, many complaints go unnoticed or unresolved, leading to dissatisfaction and potential labor disputes. Furthermore, traditional salary evaluation methods are typically based on manual observations or supervisor reports, which can introduce bias, favoritism, or errors in assessing performance and determining wages.

## Proposed system

The proposed system includes an online payment system where the admin provides salary payments to workers based on their working hours and performance evaluation. The salary calculation is automated using the decision tree algorithm, which considers factors such as total hours worked, work status updates (morning and afternoon), and task completion efficiency to determine the appropriate wages for each worker Once the salary is calculated, the admin can process payments digitally, ensuring a seamless and transparent transaction process. To enable online payments, the system integrates a secure payment gateway such as Google Pay, Paytm, Razor pay, Stripe, or UPIbased transactions, allowing direct fund transfers from the employer to the worker's registered bank account or mobile wallet. All salary details, including worked hours, performance evaluation, deductions (if any), and final payout, are stored securely in Realtime Database or, ensuring accurate payroll management and financial tracking. Workers can also view their payment history through the mobile application, providing complete transparency regarding their earnings. To further enhance efficiency, the admin side includes a payment management dashboard, allowing employers to approve, process, and verify payments before transferring salaries. Additionally, SQLITE can be implemented to notify workers when their salary has been credited, ensuring real-time communication and reducing wage-related disputes. Security measures such as Authentication, encrypted transactions, and multi-factor authentication for admin access are incorporated to prevent unauthorized salary modifications and ensure that only legitimate payments are processed.

This system not only eliminates the hassle of manual salary calculations and cash transactions but also enhances the efficiency, accuracy, and reliability of wage disbursement for migrant laborers, making it a secure, automated, and datadriven payroll solution. Furthermore, the proposed online payment system not only ensures timely salary disbursement but also enhances financial inclusion for migrant laborers, many of whom may not have access to traditional banking services. By integrating digital wallets and UPI-based transactions, workers can conveniently receive their wages without relying on cash payments, reducing the risks associated with handling physical currency, such as theft, loss, or delayed. Additionally, the payment logs and transaction records are securely maintained proposed system focuses on the development of a comprehensive solution for monitoring migrant laborers in real-time, using a mobile-based application that integrates decision-free algorithms, location tracking, and automated issue management. The system will allow both admins (employers) and workers (laborers) to interact through an Android-based platform built with Java programming language and SQLite as the database. This system is designed to ensure efficient monitoring of migrant workers' working hours, work status, and the real-time tracking of their location. By implementing the decisionfree algorithm, the system can automate the decisionmaking process regarding laborer performance, wages, and issue resolution. This paper will discuss the structure and functionality of the proposed system, with a detailed look at both the admin and user modules, the integration of decision-free algorithms, and the role of location-based services. The core objective of the proposed system is to monitor the performance, working hours, and issues faced by migrant laborers in real-time through a mobile application. The system will consist of two primary modules: the Admin Module and the User (Worker) Module. The Admin Module is for employers or supervisors, who will have full access to manage workers, monitor their activities, track their issues, and make decisions based on performance metrics. once the workers log into the application using their mobile devices, the admin can access their performance data, including working hours, work status,

and whether the worker. The User Module is for the workers who will use the system to log their work status, report issues, and have their location tracked.

This data will be analyzed using the decision-free algorithm
to determine whether workers are meeting performance standards and if any intervention is needed. Additionally, workers can report issues such as unsafe working conditions, delayed wages, food or water shortages, or any other problems they may encounter. he experimental setup for the Monitoring Migrant Laborers Using Decision Tree Algorithm project involves rigorous testing and validation of the key system functionalities, including GPS-based attendance tracking.

The first experiment focuses on GPS-based attendance tracking, where workers are required to log in to the system only if their mobile location is enabled. The system verifies whether the worker is physically present at the job site before allowing them to mark attendance. Several test cases are conducted by simulating different geographical locations, including cases where the worker is within the designated work area, outside the work area, or attempting to fake the location. The second experiment involves testing the work status reporting module, where workers upload morning and afternoon work updates. The system verifies whether the updates are submitted within the expected time range and ensures that each worker submits only one status per session to prevent duplicate or fraudulent entries.

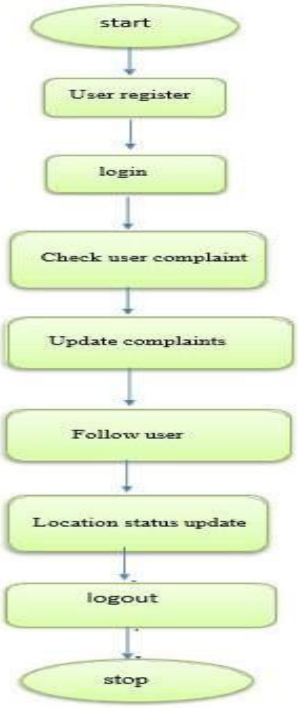


Figure 2. Flow chart.

Experiments

The experimental setup for the Monitoring Migrant Laborers Using Decision Tree Algorithm project involves rigorous testing and validation of the key system functionalities, including GPS-based attendance tracking, work status monitoring, decision tree-based performance evaluation, grievance reporting, and online salary payment processing. The experiments are conducted in a controlled environment where both admin and worker modules are tested for accuracy, efficiency, security, and real-time data processing. The first experiment focuses on GPS-based attendance tracking, where workers are required to log in to the system only if their mobile location is enabled. The system verifies whether the worker is physically present at the job site before allowing them to mark attendance. Several test cases are conducted by simulating different geographical locations, including cases where the worker is within the designated work area, outside the work area, or attempting to fake the location. The second experiment involves testing the work status reporting module, where workers upload morning and afternoon work updates. The system verifies whether the updates are submitted within the expected time range. The primary objective of the experiments conducted for this system is to evaluate the effectiveness of the Android-based application in tracking the working hours, monitoring worker performance, managing issues, and ensuring real-time location tracking. The primary objective of the experiments conducted for this system is to evaluate the effectiveness. The third experiment evaluates the decision tree algorithm for worker performance assessment. The algorithm is tested using multiple worker profiles with varying working hours, attendance patterns, and task completion rates. The decision tree processes these inputs and classifies workers into different salary categories based on their efficiency and consistency. SQLite Security Rules are examined to validate their effectiveness in restricting unauthorized data access. The grievance reporting and resolution system, where workers submit complaints related to delayed wages, workplace safety, food and water shortages, or housing issues. The system is tested for realtime data synchronization with Firebase, ensuring that reported issues appear immediately in the admin dashboard. The experiment also measures the response time for admin intervention, where the admin reviews the complaints, updates the resolution status, and pushes real-time notifications via SQlite to notify workers once their issues are addressed. The effectiveness of the notification system is evaluated by checking whether workers receive alerts without delays and whether multiple issue resolutions can be handled simultaneously.

The decision tree operates by recursively splitting data into branches based on conditions that maximize information gain. By using key attributes such as working hours, work status (full- time, part-time, contract), work completion rate, and performance ratings, the algorithm can classify workers into different salary categories. For example, employees who work more than 40 hours per week and have a high task completion rate may be classified into a high-salary category, whereas those with fewer working hours and a lower completion rate might fall into a lower salary range. This structured approach helps organizations identify performance trends, reward top- performing employees, and optimize workforce management. For example, employees who work more than 40 hours per week and have a high task completion rate may be classified into a high-salary category, whereas those with fewer working hours and a lower completion rate might fall into a lower salary range.

Figure 4. Work Performance Analysis By Using Decision Tree Model.

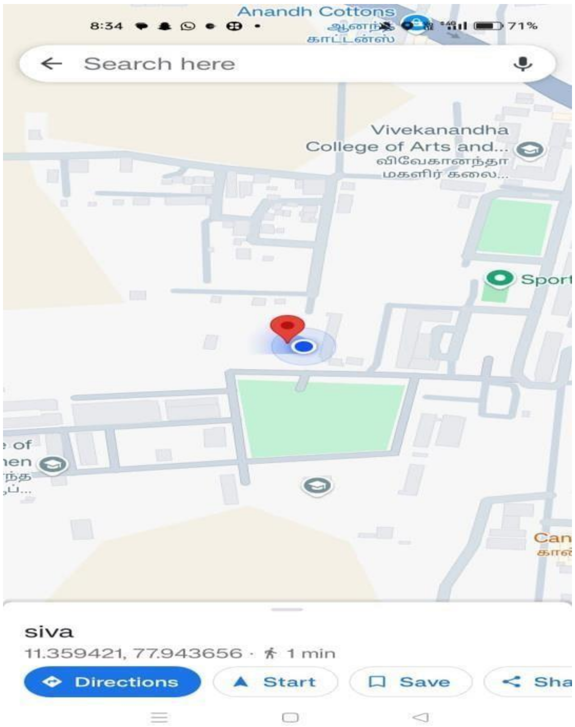
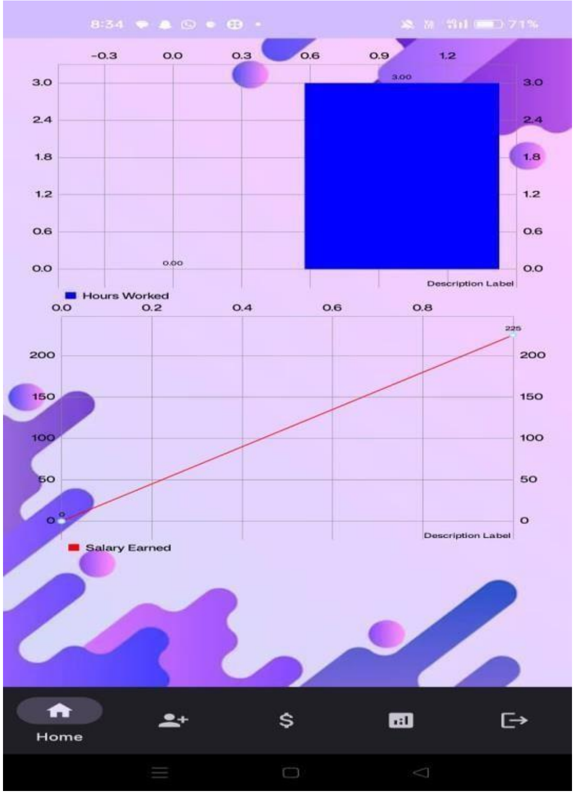


Figure 3 . Tracking Labors Current Location.



online salary payment processing, where the system calculates wages based on working hours and decision tree evaluation and facilitates secure fund transfers using a payment gateway such as Google Pay, Paytm, Razor pay, or UPI-based transactions.

The decision tree algorithm is used to analyze the performance of workers based on their working hours and work status. The decision tree is a machine learning algorithm that helps in decisionmaking by splitting data into branches based on certain conditions. In your case, it is used to determine salary levels and visualize worker performance. The algorithm will create a decision tree structure to classify workers into different categories and determine their salary level. Once the decision tree is executed, the results can be visualized. The decision tree algorithm is a powerful tool for analyzing worker performance and salary determination by systematically breaking down complex decision-making processes into a structured hierarchy of conditions.

This structured approach helps organizations identify performance trends, reward topperforming employees, and optimize workforce management. The decision tree results can be visualized using various graphical tools such as bar charts to compare individual performance metrics, decision tree diagrams to illustrate classification structures, and feature importance charts to highlight the most critical factors influencing salaries. These visualizations enable HR managers and business analysts to make data-driven decisions, ensuring fair compensation strategies and improved workforce efficiency. Additionally, decision trees provide a transparent and interpretable framework, making them ideal for understanding salary distribution patterns without requiring complex statistical knowledge.

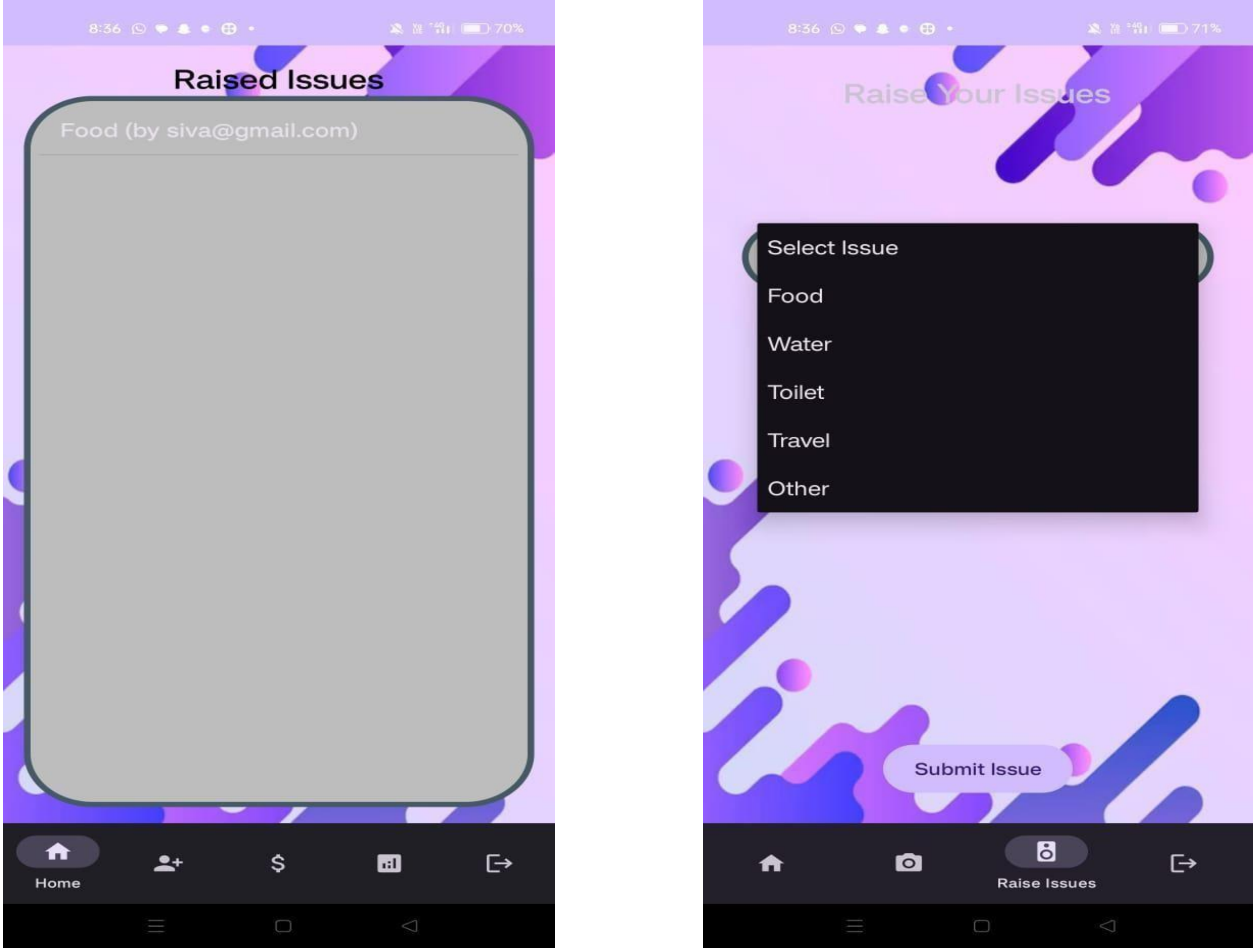


Figure 5. Arising Issues To The Admin.

The Admin Module requires the admin to log in with secure credentials before accessing the system. After a successful login, the admin can add new employees to the database, which involves filling in personal details such as the worker's name, age, job role, work location, contact details, and profile picture. This process ensures that each worker is properly registered in the system and can be monitored effectively. The admin also has the authority to update or remove worker details when necessary. On the User Module side, workers must also log in to access their accounts. However, an additional layer of security is implemented—workers must turn ON their mobile location before logging in. This prevents unauthorized access and ensures that only active workers can use the application. Once logged in, workers can update their work status in the morning and afternoon, upload images related to their work, and report any issues they are facing. The ability to set a profile picture adds a personalized touch to the system, making it easier for the admin to identify workers at a glance. This structured login and worker registration process helps maintain an organized system, ensuring transparency in workforce monitoring and improving overall labor management efficiency. The development of the Monitoring Migrating Labors Android application involves the use of various tools and technologies to ensure efficient tracking, management, and analysis of worker performance. The project is built using Android Studio, the official IDE for Android development, with Java as the primary programming language to implement the backend logic and user interface functionalities. The Monitoring Migrating Labors Android application is developed using Android Studio with Java as the primary programming language and SQLite as the database for data storage and management. SQLite is a lightweight, local database that efficiently stores worker details, work status, reported issues, and salary classifications without requiring an internet connection The admin module allows the administrator to log in, add new employees, and monitor their working hours and job status using structured data storage in SQLite. The database ensures that all information related to workers, including their personal details, work updates, and reported issues, is securely stored and easily retrievable. Online payment salary is a digital method of transferring wages directly to employees' bank accounts or digital wallets through an electronic payment system. This method eliminates the need for physical cash or checks, making salary disbursement more efficient, secure, and convenient for both employers and employees. Typically, online salary payments are processed through payroll management systems integrated with banking services, allowing automatic calculations of wages based on working hours, deductions, and bonuses. In your project, since employee performance is monitored using a decision-tree algorithm, the system can determine salary levels based on work hours and performance metrics, and then process payments accordingly. Employers can use various online payment gateways such as In conclusion, the "Monitoring Migrating Labors by Using DecisionTree" mobile application provides an efficient and technology-driven approach to labor management.

The Monitoring Migrating Labors Android application is developed using Android Studio with Java as the primary programming language and SQLite as the database for data storage and management. SQLite is a lightweight, local database that efficiently stores worker details, work status, reported issues, and salary classifications without requiring an internet connection The admin module allows the administrator to log in, add new employees, and monitor their working hours and job status using structured data storage in SQLite. The database ensures that all information related to workers, including their personal details, work updates, and reported issues, is securely stored and easily retrievable. Online payment salary is a digital method of transferring wages directly to employees' bank accounts or digital wallets through an electronic payment system. This method eliminates the need for physical cash or checks, making salary disbursement more efficient, secure, and convenient for both employers and employees. Typically, online salary payments are processed through payroll management systems integrated with banking services, allowing automatic calculations of wages based on working hours, deductions, and bonuses. In your project, since employee performance is monitored using a decision-tree algorithm, the system can determine salary levels based on work hours and performance metrics, and then process payments accordingly. Employers can use various online payment gateways such as In conclusion, the "Monitoring Migrating Labors by Using Decision-Tree" mobile application provides an efficient and technology-driven approach to labor management, ensuring transparency, accountability, and improved working conditions for migrant workers. , or direct bank transfers to send salaries, ensuring timely payments and reducing errors. Additionally, digital salary payments provide transparency, as workers can view their payment history and earnings through a mobile application. If there are any disputes regarding payments, the system can store records for verification. Moreover, integrating an online payment system can also support automatic tax deductions, generate digital , and provide real-time notifications to employees when their salary is credited. This approach enhances financial security for migrant laborers, ensuring they receive payments on time and reducing the risk of wage theft or payment delays.

I would like to express my sincere gratitude to all the individuals and institutions that have supported and contributed to the successful completion of this project titled "Monitoring Migrating Laborers Using DecisionTree Algorithms." First and foremost, we would like to thank our project guide for their invaluable guidance, unwavering support, and insightful feedback throughout the entire project. By leveraging technology, this project not only optimizes workforce management but also fosters a more structured and fair labor system. we would like to thank our project guide for their invaluable guidance, unwavering support, and insightful feedback throughout the entire project. Their expertise and commitment to our growth as researchers have played a crucial role in shaping the direction of this project.

References

*1* *Patel, R., Sharma, K., & Mehta, P. (2019). Smart attendance tracking using biometric systems.*

*2* Sharma, A., & Kumar, V.
(2021). Cloud-based workforce management solutions.

*3* *Singh, D., Reddy, P., & Banerjee, S. (2020). GPS-based tracking for migrant laborers.*

*4* Mishra, A., Roy, T., & Das, S. (2022). Decision tree applications in workforce productivity

*5* Gupta, S., & Rao, M. (2021). Predictive analysis of employee performance.

*6* Zhang, H., Liu, X., & Chen, Y. (2019). Visualizing workforce analytics using Decision Trees.

*7* *Bose, P., Kumar, A., & Das, R. (2018). Challenges faced by migrant laborers*

*8* Das, P., & Sinha, R. (2020). Workplace safety concerns in labor-intensive industries.

*9* Verma, N., & Yadav, K. (2021). Job insecurity among
migrant workers.

*10* Choudhary, B., & Mehta, D. (2019). Limitations of manual workforce tracking systems.

*11* Roy, S., Gupta, K., & Sharma, N. (2020). Biometric attendance systems: Challenges and opportunities.

*12* Jain, R., & Malhotra, P. (2022). Mobile applications for labor tracking.

*13* Android Developers (n.d.). Android Development Documentation. Retrieved from
https://developer.android.com

*14* SQLite Documentation (n.d.). Official SQLite Guide. Retrieved from https://www.sqlite.org/docs.html

*15* Quinlan, J. R. (1986). Induction of Decision Trees. Machine Learning, 1(1), 81-106

*16* Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques. Elsevier.

*17* Mitchell, T. M. (1997). Machine Learning. McGrawHill Education.