A Major Project Synopsis on

Issue Tracker: Advanced Issue management and Resolution System

Submitted to Manipal University,

Jaipur

Towards the partial fulfillment for the Award of the

Degree of

MASTER OF COMPUTER APPLICATIONS

2023-2025

by

Pawani Awasthi

23FS20MCA00004

MANIPAL UNIVERSITY
JAIPUR

Under the guidance of

Dr. Avichandra Singh

Department of Computer Applications

School of AIML, IoT&IS, CCE, DS and Computer Applications

Faculty of Science, Technology and Architecture

Manipal University Jaipur

Jaipur, Rajasthan

2025

External Internship

Company Name: WENDOR

Role: Frontend Developer Intern

Location: Gurugram, Haryana, India

Company Website: Wendor

I. Introduction

In every organization, employees encounter various workplace issues, such as malfunctioning office equipment, non-operational lifts, faulty air conditioning, or laptop-related problems. To streamline the resolution of these concerns, we are developing an office issue-tracking application.

This application features three key user roles: **Employee, Agent, and Admin**. Employees can generate tickets for their issues and track their status through a dedicated dashboard. Agents are responsible for addressing these concerns and have access to all employee-generated tickets. Meanwhile, Admins oversee the entire process, monitoring all tickets and tracking which agent is assigned to which issue. This ensures better management, faster resolutions, and improved workplace efficiency. By offering a structured and transparent issue-resolution system, this application aims to enhance workplace productivity and provide employees with a seamless way to report and resolve their problems efficiently.

II. Motivation

1. Efficient Issue Resolution: Streamlines problem reporting and ensures quick resolution.
2. Transparency and Accountability**:** Enables employees to track issues and hold agents accountable.

3. Improved Workplace Productivity: Reduces disruptions by resolving office issues efficiently.
4. Effective Management Oversight: Allows admins to monitor and manage all reported issues.
5. User-Friendly and Scalable Solution**:** Easy to use and adaptable for organizations of all sizes.

III. Problem Statement

- **Key challenges include:**
  1. Employees face difficulties in reporting workplace problems, leading to delays in resolution.
  2. There is no centralized system for employees to track the status of their reported issues, causing frustration and inefficiency.
  3. Admins struggle to monitor issue resolution, and agents lack a streamlined system to manage and prioritize tickets.

IV. Methodology/ Planning of Work

1. **Requirement Analysis:**
   - Identify key features and functionalities required for the application.
   - Define user roles (Employee, Agent, Admin) and their respective permissions.
2. **System Design & Architecture:**
   - Design the database schema to store tickets, user information, and issue statuses.
   - Create wireframes and UI mockups for user-friendly navigation.

3. **Frontend & Backend Development:**
    - Develop the frontend using React.js for an interactive user experience.
    - Build the backend with Node.js and Express.js, ensuring seamless API integration.
    - Implement a MongoDB database for efficient data storage and retrieval.
4. **Implementation of Features:**
    - Employee: Ticket creation, tracking, and status updates.
    - Agent: Issue resolution, viewing assigned tickets, and updating status.
    - Admin: Monitoring all tickets, assigning agents, and managing system workflow.
5. **Testing & Debugging:**
    - Perform unit testing and integration testing to ensure functionality and performance.
    - Conduct user testing to refine the user experience and fix bugs.
6. **Deployment & Maintenance:**
    - Deploy the application on a cloud-based platform for accessibility.
    - Continuously monitor and update the system for security and performance improvements.

V. Requirements for Proposed Work

- **Software Requirements:**

  a. Operating System: Windows, Linux

  b. Frontend: React JS, HTML, CSS

  c. Database: MongoDB

  d. Backend: Node.js with Express.js

- **Hardware Requirement:**

  a. Hardware: Pentium based systems with a minimum of P4

  b. RAM: Minimum 256MB (Recommended: 4GB for optimal performance)

  c. Hard Disk: 10 GB Hard Disk Space

VI. References

1. Bcrypt for Secure Password Hashing - *https://www.npmjs.com/package/bcrypt*
2. MongoDB Documentation - *https://www.mongodb.com/docs/*
3. React.js Official Documentation - *https://react.dev/*
4. Node.js & Express.js Official Guide - *https://expressjs.com/*