IT Ticketing System - Software Requirements Document (SRD)

# 1. Introduction

This document defines the software requirements for the IT Ticketing System, which will be developed to streamline IT support operations. The system allows employees to raise tickets, IT agents to manage them, and administrators to oversee the entire workflow. The solution includes role-based access, SLA management, notifications, reporting, and analytics.

# 2. System Overview

The IT Ticketing System is a web-based platform with role-based dashboards, ticket management features, SLA monitoring, escalation policies, audit tracking, and reporting capabilities. It will support future integration with mobile applications via APIs.

# 3. User Roles & Permissions

**The system will have the following roles:**

- \*\*Admin\*\*: Full control, manage users, assign/merge tickets, configure SLA policies, view reports.  
- \*\*IT Support Agent\*\*: Manage assigned tickets, update status, comment, attach files, view SLA.  
- \*\*Employee (Requester)\*\*: Create/view own tickets, comment, attach files.  
- \*\*(Optional) Department Manager\*\*: View all tickets within their department.

# 4. Functional Requirements

## 4.1 Authentication & User Management

- Login/Logout with username and password  
- Passwords stored using hashing (bcrypt/sha256)  
- Admin can add/remove/reset users  
- Input validation & sanitization

## 4.2 Ticket Management

- Create tickets with fields: Employee ID, Name, Mobile, Email, Department, Title, Category, Priority, Description, Attachments  
- Attachments allowed: PDF, PNG, JPG, DOCX (max 5MB)  
- Auto-generate unique Ticket ID  
- Status options: Open, In Progress, On Hold, Resolved, Closed  
- Assign tickets manually (Admin) or auto-assign (round robin)  
- Urgent notifications sent via email when new ticket is created

## 4.3 SLA & Escalation Management

- SLA policies configurable (e.g., High Priority = 4 hours)  
- SLA countdown timer visible on ticket  
- Auto-escalation to Admin when SLA breached

## 4.4 Dashboard Views

- Admin: All tickets with filters (status, priority, department, agent)  
- Agent: Assigned & pending tickets  
- Employee: Own tickets  
- Manager: Tickets in their department

## 4.5 Notifications

- Email notifications via SMTP  
- In-app notifications for status updates  
- Daily/weekly summary report to Admin

## 4.6 Search & Filter

- Search by Ticket ID, title, employee name, category  
- Filter by status, date, department, priority

## 4.7 Reports & Analytics

- Export tickets to CSV/Excel  
- Metrics: total tickets, open, resolved, SLA breaches  
- Graphs: tickets by category, department, agent performance

## 4.8 Audit Log & History Tracking

- Track all changes: status updates, comments, assignments  
- Maintain accountability with timestamp and user info

## 4.9 Ticket Linking & Merging

- Merge duplicate tickets  
- Link related tickets together

# 5. Security Requirements

- Password hashing (bcrypt/sha256)  
- File upload restrictions (max 5 MB, PDF/PNG/JPG/DOCX only)  
- Input validation to prevent SQL injection & XSS  
- Rate limiting for ticket creation (anti-spam)

# 6. Technical Requirements

- Backend: Python (Flask/Django) or Node.js  
- Frontend: React.js / Vue.js  
- Database: PostgreSQL/MySQL  
- Deployment: Local setup with sample data  
- Future API integration for mobile app

# 7. Deployment & Setup

Steps for running locally:  
1. Install dependencies (Python/Node, database).  
2. Import sample data (users + tickets).  
3. Start backend and frontend servers.  
4. Login with default Admin credentials:  
 - Username: admin  
 - Password: Admin@123

# 8. Sample Data

- Preloaded users (Admin, Agents, Employees)  
- Example tickets for testing