

A Major Project Synopsis on

# **TICKET ANALYSIS & SUPPORT**

Submitted to Manipal University, Jaipur

Towards the partial fulfillment for the Award of the Degree of

**MASTER OF COMPUTER APPLICATIONS**

2023-2025

by

**AWANI VINIT SATHE**

23FS20MCA00007



**MANIPAL UNIVERSITY  
JAIPUR**

Under the guidance of

**Dr. Kuntal Gaur**

**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**

## **I. Introduction**

In the modern age of customer service, quick and accurate support response is essential. Businesses often receive numerous support tickets from users seeking solutions for their technical issues or inquiries. However, traditional support systems rely heavily on manual processes, which can result in delays, miscommunication, or incorrect ticket assignments.

This project introduces a web-based ticket management solution built using FastAPI and integrated with Meta's LLaMA 3.1 language model. The aim is to automate ticket routing, generate solutions using AI, and notify support agents promptly via email. By automating key parts of the process, this platform enhances efficiency, reduces human error, and ensures timely responses to support queries.

**Objectives of the project include:**

- Automating ticket assignment to the most suitable support agents.
- Predicting ticket priority (High, Medium, Low) using LLMs.
- Notifying assigned agents via automated email notifications.
- Generating LLM-based beginner-friendly solutions for ticket topics and descriptions.
- Secure and scalable storage of all ticket data in a MySQL database.

## **II. Motivation**

The motivation behind this project stems from real-world challenges faced by organizations in managing customer support efficiently. In today's fast-paced digital environment, customers expect quick, reliable, and accurate responses to their queries. Traditional support systems often fall short of these expectations due to their dependence on manual operations.

Key motivating factors include:

- The increasing volume of support tickets and limited staff to manage them.
- Delays in assigning tickets to the appropriate agents.
- Lack of personalization and intelligence in current ticket handling methods.
- The availability and power of LLMs like Meta LLaMA 3.1, which can understand natural language and generate intelligent, human-like responses.
- The opportunity to blend modern technologies like AI, APIs, and cloud-based systems to create a seamless, responsive support ecosystem.

This project aims to combine innovation and practicality to solve a widespread problem in a scalable and impactful manner.

### III. Problem Statement

In many customer service environments, incoming tickets are handled manually, requiring human staff to read, understand, and assign them. This manual approach results in:

- Inefficient response times.
- Incorrect ticket assignments.
- Lack of clarity in communication.
- Increased workload for support managers.

Furthermore, users often have to wait long periods before receiving solutions to their queries. With increasing ticket volumes, the demand for an automated system that can reduce human workload and improve accuracy becomes more significant.

This project seeks to solve these challenges by incorporating an LLM that can understand and analyze the ticket content, suggest the best-fit agent, set appropriate priority levels, and even generate informative email alerts and responses.

### IV. Methodology/ Planning of work:

The methodology involves multiple stages, beginning with user input and ending with data storage and notification.

#### Step-by-step Process:

1. The user submits a ticket via an API form with their name, topic, description, and active status.
2. The backend receives the data via the `/master` endpoint and prepares a prompt for the LLM.
3. The Meta LLaMA model analyzes the ticket and determines:
  - Best-suited agent (name, department, email).
  - Priority level (High, Medium, Low).
4. These details are stored in the MySQL database.
5. Another LLM prompt generates a formal and informative email.
6. The email is sent to the assigned agent using SMTP (Gmail).
7. When a user wants a solution, they access the `/get_solution` endpoint.
8. The LLM returns a simple and helpful explanation or code correction.

This methodology allows for a seamless integration between AI-driven decision-making and traditional backend operations.

## **V. Requirements for proposed work:**

### **Software Requirement:**

The project utilizes a combination of modern software tools and libraries to implement, deploy, and run the platform efficiently.

Software Components:

1. **Python 3.11+** – Core language for development.
2. **FastAPI** – High-performance backend web framework for building APIs.
3. **Uvicorn** – ASGI server used to run the FastAPI app.
4. **MySQL** – Relational database for storing ticket and agent data.
5. **mysql-connector-python** – Connector for Python-MySQL integration.
6. **OpenAI SDK** – Interface to connect with Meta LLaMA 3.1 hosted on NVIDIA Inference.
7. **smtplib** – Standard Python module used to send email notifications.
8. **dotenv** – Secure management of environment variables like API keys and passwords.

### **Hardware Requirement:**

To run this project efficiently, certain minimum hardware resources are required:

Minimum Hardware Configuration:

1. **Processor:** Intel Core i5/i7 or equivalent.
2. **Memory (RAM):** 8 GB or more.
3. **Storage:** At least 250 GB SSD.
4. **Internet Connection:** Required for accessing LLM APIs and sending emails.

This configuration ensures smooth development and execution of the application along with real-time API communication.

