PHASE 2 PROJECT PLANNING PHASE DOCUMENT

Streamlining Ticket Assignment for Efficient Support Operations

TEAM ID	**NM2025TMID08979**
Team Leader	Yogesh S
Team Member	Selvin Joshva A
Team Member	Naveen P
Team Member	Ravin Akash S

Introduction:

Efficient ticket handling is a cornerstone of high-performing support operations. Manual ticket triage often leads to misrouting, uneven workloads, and delayed responses, reducing customer satisfaction. This project aims to develop an **AI-powered ticket assignment system** that automates categorization and routing of support tickets to the most suitable agents based on expertise, workload, and priority. The planning phase focuses on defining clear goals, scope, milestones, resource allocation, and risk management strategies to ensure structured and timely execution.

Project Objectives:

- Automate ticket categorization and assignment using NLP and machine learning.
- Improve first-response and resolution times.
- Balance workload among agents to enhance productivity.
- Integrate seamlessly with existing ticketing platforms (Jira, Freshdesk, etc.).
- Provide a web-based dashboard for monitoring support operations and analytics.

Scope of the Project:

In-Scope:

- Development of a ticket classification model using NLP.
- Automatic ticket assignment based on skillset and workload.
- Integration with a simulated or real ticketing dataset.
- Interactive dashboard for visualizing ticket distribution and performance metrics.
- Deployment of the solution on a cloud platform for demonstration.

Out of Scope (Future Enhancements):

- Voice-based ticket creation.
- Multilingual ticket support.
- Integration with real enterprise systems (for pilot stage only).

Deliverables:

Deliverable	Description	Expected Output
Requirement Document	Detailed user and system requirements	SRS Document
System Design	Architecture diagram, flowcharts, data model	UML diagrams, ER model
ML Model	NLP-based classification model	Trained AI model
Backend API	RESTful service for ticket handling	Flask/FastAPI endpoints
Frontend Dashboard	Visualization of ticket flow and metrics	React/HTML UI
Final Report & Presentation	Documentation and demo presentation	Project report + slides

Work Breakdown Structure (WBS):

Phase	Task	Responsible Team	Duration
1. Requirement Analysis	Gather requirements, identify ticket flow	All Members	Week 1
2. Design Phase	Architecture, UI mockups, ML pipeline design	Design & ML Team	Week 2
3. Development Phase	Build NLP model, backend APIs	Backend & ML Team	Week 3–4
4. Integration Phase	Connect frontend, backend, and database	Full Stack Team	Week 5
5. Testing Phase	Functional, performance, and integration testing	QA Team	Week 6
6. Deployment & Review	Deploy to cloud and prepare demo	All Members	Week 7

Resource Planning:

Resource Type	Description	Tool / Platform
Hardware	System for development and testing	Laptops, Cloud Server (AWS/Render)
Software	Programming, ML, and integration tools	Python, Flask, React, PostgreSQL
ML Libraries	For NLP and model training	TensorFlow / Scikit-learn / SpaCy

Human Resources	Development and management team	1 Team Lead, 3 Developers, 1 QA
Dataset	Ticket datasets for training & testing	Public datasets (e.g., Kaggle IT Tickets)

Timeline / Milestone Chart:

Milestone	Task	Deliverable	Duration
M1	Requirement Gathering	Requirement Specification	Week 1
M2	Design Completion	System Architecture & Mockups	Week 2
M3	Backend & Model Development	API + ML Model	Week 3–4
M4	Frontend Integration	Dashboard with live data	Week 5
M5	Testing & Bug Fixing	Tested, functional prototype	Week 6
M6	Deployment & Documentation	Final report & demo	Week 7

Risk Analysis and Mitigation:

Potential Risk	Impact	Likelihood	Mitigation Strategy
Delay in model training	Medium	Medium	Use pre-trained NLP models to save time
API integration failure	High	Low	Perform modular testing and maintain version control

Data inconsistency	Medium	High	Use data validation and cleaning scripts
Limited time/resources	High	Medium	Divide tasks and prioritize core features

Quality Assurance Plan:

- Conduct **unit testing** for each module (frontend, backend, ML).
- Perform **integration testing** for smooth communication between services.
- Carry out **performance testing** under simulated ticket load.
- Review and verify accuracy of ticket classification model.
- Maintain version control via GitHub/GitLab for transparency.

Communication and Reporting:

Activity	Frequency	Platform	Responsible
Team Meetings	Twice a week	Google Meet / MS Teams	Team Lead
Progress Updates	Weekly	Shared Document / Slack	All Members
Code Reviews	After each milestone	GitHub Pull Requests	Developers
Final Review	End of Project	Presentation / Demo	Entire Team

Expected Outcome:

- A fully functional **AI-powered ticket assignment system**.
- Improved efficiency in support operations through intelligent automation.

- Measurable reduction in ticket resolution time.
- Interactive dashboard displaying ticket flow and analytics.
- Scalable and adaptable framework for future enterprise integration.

Future Enhancements:

- Incorporate **sentiment analysis** to gauge customer urgency.
- Enable **multi-channel ticket intake** (email, chat, voice).
- Implement **self-learning mechanisms** to improve classification accuracy.
- Add **real-time agent performance analytics**.

Conclusion:

The planning phase establishes a strong foundation for building a scalable and intelligent ticket assignment system. Through AI-driven automation, this project seeks to revolutionize support operations, ensuring faster resolutions, balanced workloads, and improved customer satisfaction.