

## Fynd AI Intern – Take Home Assessment Report

Candidate: *jateen Mourya*

Project: AI-Driven Feedback System with Dual Dashboards

### 1. Overall Approach

This project was implemented in two parts as defined in the assessment:

- **Task 1** focuses on rating prediction using prompt engineering.
- **Task 2** focuses on building a production-style web application with two dashboards powered by an LLM.

For Task 2, the goal was to design a **real-world feedback system** where:

- Users can submit feedback and immediately receive an AI-generated response.
- Admins can monitor all feedback submissions, view AI summaries, and take recommended actions.

The system follows a **client–server architecture** with:

- A React-based frontend
- A Node.js + Express backend
- MongoDB for persistent storage
- LLM calls handled strictly on the server side

This approach ensures scalability, security, and compliance with all technical constraints mentioned in the assessment.

### Key Design Decisions

- **Server-side LLM usage only**  
All AI calls (summarization, action recommendation, user response) are made from the backend to ensure API key security.
- **Single persistent data source**  
Both dashboards read from and write to the same MongoDB database, ensuring data consistency.
- **Separated dashboards via routes**
  - User Dashboard
  - admin → Admin Dashboard

This clearly satisfies the “two dashboards” requirement while keeping the application simple.

- **Production-style UI**

Modern UI elements (cards, live stats, star rating, feedback stream) were used to mimic a real SaaS dashboard experience.

## 2. Task 1 – Prompting Strategy (Summary)

Three prompting approaches were implemented for Yelp review rating prediction:

1. **Direct Prompting**

Simple instruction to classify the review into a star rating.

2. **Structured Prompting**

Explicit instructions to return strictly valid JSON with reasoning.

3. **Context-Enhanced Prompting**

Added examples and constraints to improve consistency and JSON validity.

## Improvements Across Iterations

- Increased JSON validity
- Reduced ambiguous outputs
- Improved rating consistency for borderline reviews

## 3. Evaluation Methodology & Results (Task 1)

### Dataset

- Yelp Reviews Dataset (Kaggle)
- Sample size: ~200 reviews

### Metrics Used

- Accuracy (Actual vs Predicted rating)
- ±1 Star Accuracy
- JSON validity rate
- Consistency across runs

### Observations

- Structured prompts significantly improved JSON validity.
- Context-enhanced prompts improved rating consistency but slightly increased latency.

- Trade-off observed between strict formatting and flexibility.

## 4. Task 2 – System Behaviour

### User Dashboard (Public-Facing)

Users can:

- Select a star rating (1–5)
- Write a short review
- Submit feedback

On submission:

- The feedback is stored in MongoDB.
- The backend calls an LLM to generate:
  - A user-facing AI response
  - An internal summary
  - A recommended action
- The AI-generated response is immediately displayed to the user.
- Clear success and error states are shown.

### Admin Dashboard (Internal-Facing)

Admins can:

- View a live-updating list of all submissions.
- See:
  - User rating
  - User review
  - AI summary
  - AI-suggested action
- Monitor system analytics:
  - Total feedback count
  - Average customer satisfaction score
  - Live system status

## 5. Trade-offs and Limitations

### Trade-offs

- Frontend analytics (average rating, counts) are computed client-side to reduce backend complexity.
- Auto-refresh polling was used instead of WebSockets for simplicity and reliability.

### Limitations

- No authentication layer for admin access (out of scope).
- LLM responses may vary depending on prompt and model behavior.
- System currently optimized for moderate traffic, not high concurrency.

## 6. Deployment & Reliability

- The application is designed to be deployed on platforms such as **Vercel (frontend)** and **Render (backend)**.
- Data persists across refreshes using MongoDB Atlas.
- The system functions without manual intervention after deployment.

## 7. Submission Links

- **GitHub Repository:** [Backend link : [JATINMOUR/backend](#)] [Frontend link: [JATINMOUR/frontend-user](#)]
- **User Dashboard URL:** [\[frontend-user\]](#)
- **Admin Dashboard URL:** [\[frontend-user\]](#)
- **Report PDF Link:** []