

Fynd AI Intern - Take Home Assessment Report  
Candidate - Divanshu

## Overview

The submission shows a complete AI feedback system that runs in Streamlit and uses large language models. Users type feedback plus a rating - the system returns an AI reply on the spot. Administrators see a page that lists every comment, shows a short summary of the batch, lists suggested next steps and displays how many ratings fall into each star level.

### Task 1: Prompt-based Rating Prediction & Analysis

A Jupyter Notebook tests different wordings of prompts that ask a model to guess the star rating from the text of a review. Each wording is scored for how often the guess matches the true rating but also for the clarity of the explanation that the model gives. The notebook is stored in the GitHub folder for Task 1.

### Task 2: AI-powered Feedback Application

1 User Dashboard - The visitor picks a star rating, writes comments and presses Submit. The page sends the text to the model as well as shows the model's reply right away.  
2 Admin Dashboard - The page lists every review, shows a short paragraph that sums up the main points, lists concrete actions the firm could take and shows a bar chart of how many reviews belong to each star level.

### Deployed Application Links

User Dashboard - <https://ai-powered-feedback-system-3upggb8czgyckedssyu7ab.streamlit.app/>  
Admin Dashboard - <https://ai-powered-feedback-system-e4pgzw7bthbb3ackipdfgm.streamlit.app/>

### Technical Stack

- 1 Python 3.11
- 2 Streamlit for the interface or for hosting
- 3 OpenRouter API with a Mistral-based large language model
- 4 Pandas to store and to study the data
- 5 GitHub to track changes

### Conclusion

The project meets every requirement - it uses prompt engineering, calls an LLM, replies to users in real time and gives managers a ready-to-read analytic page. The full system is live, the code reproduces on any machine and new features can be added later.