

Title: Fynd AI Intern Take-Home Assessment

1) Introduction

Two components: (a) Task 1 notebook for Yelp star rating prediction using prompting, (b) Task 2 two-dashboard web app (User + Admin) with LLM-generated responses, summaries, and recommended actions.

2) Task 1: Rating Prediction

- Dataset: Kaggle Yelp Reviews via kagglehub.
- Approaches: Zero-Shot, Few-Shot, Chain-of-Thought.
- Output: JSON { "predicted_stars": n, "explanation": "..." }.
- Evaluation (sampled ~200 reviews, 1s throttle; indicative):
- Few-Shot: 57% exact, 94% within ± 1 , JSON validity 100%, avg latency ~6.16s (std ~2.81).
- Zero-Shot: 51% exact, 93.5% within ± 1 , JSON validity 100%, avg latency ~5.19s (std ~3.03) — fastest.
- CoT: 51.5% exact, 93% within ± 1 , JSON validity 100%, avg latency ~7.99s (std ~3.03).
- Best: Few-Shot on accuracy; Zero-Shot on speed. Metrics are indicative due to sample size/quota.

3) Task 2: Web Application

- Backend: FastAPI (LLM responses, summaries, recommended actions); API URL: review-feedback-system.onrender.com
- Frontend: React + Vite (User + Admin dashboards); URL: review-feedback-system-4yim.vercel.app
- Data: JSON storage `task2/data/reviews.json` with fields id, timestamp, rating, review, ai_response, ai_summary, recommended_actions.

4) Environment & Deployment

- Env vars: GEMINI_API_KEY, GEMINI_MODEL (e.g., models/gemini-1.5-flash), VITE_API_URL → backend URL.
- Backend: uvicorn main:app --host 0.0.0.0 --port \$PORT (render.yaml)
- Frontend: npm install && npm run build/dev (root: task2/frontend).

5) Conclusion

- Three prompting strategies delivered with evaluation and visuals.
- Fully functional two-dashboard web app with AI responses, summaries, actions, analytics, and persistence.
- Latest metrics are from the ~200-sample run; results are indicative given quota constraints.