

# Title: Fynd AI Intern Take-Home Assessment

## 1) Introduction

A complete AI-powered review feedback system with two main components: (1) Task 1 notebook for Yelp star rating prediction using prompting, and (2) Task 2 two-dashboard web app (User + Admin) with LLM-generated responses, summaries, and recommendations.

## 2) Task 1: Rating Prediction

- Approaches: Zero-Shot, Few-Shot, Chain-of-Thought.
- Dataset: Yelp reviews (sampled).
- Output format: JSON { "predicted\_stars": n, "explanation": "..." }.
- Evaluation (reduced sample: 10 reviews, 1s throttle, due to API quota):
- Zero-Shot: Exact 60%, Within  $\pm 1$  100%, JSON validity 100%, Avg latency ~2.5s
- Few-Shot: Exact 60%, Within  $\pm 1$  100%, JSON validity 100%, Avg latency ~3.1s
- CoT: Exact 60%, Within  $\pm 1$  100%, JSON validity 100%, Avg latency ~4.4s
- Best approach (speed): Zero-Shot (same accuracy, lowest latency).
- Note: Metrics are indicative; full evaluation constrained by API quota.

## 3) Task 2: Dashboard System

- Backend: FastAPI; LLM service for user responses, admin summaries, recommended actions; JSON persistence at [task2/data/reviews.json](#).
- Frontend: React + Vite; User Dashboard (submit review, get AI response); Admin Dashboard (table, filters, search, analytics charts).
- Data schema: id, timestamp, rating, review, ai\_response, ai\_summary, recommended\_actions.

## 4) Deployment

- Frontend deployed: review-feedback-system-4yim.vercel.app
- Backend : Render-ready via render.yaml; set GEMINI\_API\_KEY (and optional GEMINI\_MODEL, eg.,models/gemini-1.5-flash).
- Env vars: GEMINI\_API\_KEY, GEMINI\_MODEL, VITE\_API\_URL (frontend → backend).
- Backend start: unicorn main:app --host 0.0.0.0 --port \$PORT
- Frontend build/run: npm install && npm run dev (or npm run build for deploy).

## 5) Conclusion

- Delivered three prompting strategies with evaluation and visualizations.
- Delivered functional two-dashboard web app with LLM responses, summaries, actions, and analytics.
- Noted evaluation on a reduced sample due to quota; Zero-shot preferred for speed with matching accuracy in this sample.

**Optional run instructions :**

- Task 1: pip install -r task1/requirements.txt; set GEMINI\_API\_KEY; run the notebook.
- Task 2 backend: pip install -r task2/backend/requirements.txt; uvicorn main:app --reload.
- Task 2 frontend: npm install && npm run dev; set VITE\_API\_URL to backend.