

# Fynd AI Intern – Take Home Assessment

## Deliverables

You must submit:

### 1. A GitHub Repository (mandatory)

The repository must contain:

- The Python notebook for **Task 1**
- Application code for **Task 2**
- Any supporting files
- **Deployment links for both dashboards (mandatory)**

### 2. A Short Report

Summarising your approach, design decisions, prompt iterations, evaluation, and system behavior for both tasks.

### 3. Deployed Dashboards

Both User & Admin dashboards **must be deployed** and accessible via public URLs.

**Note:** Faster completion is viewed positively.

---

## LLM Usage

You may use **any LLM** you prefer.

Free options include:

- Gemini API (free tier)
- OpenRouter (various free open-source LLMs)

Choose whichever works best for your experiments.

---

## TASK 1 - Rating Prediction via Prompting

Design prompts that classify Yelp reviews into **1–5 stars**, returning structured JSON.

### Dataset

Use the Yelp Reviews dataset from Kaggle: <https://www.kaggle.com/datasets/omkarsabnis/yelp-reviews-dataset>

You may sample a subset for efficiency.

---

## Output Format

```
{
  "predicted_stars": 4,
  "explanation": "Brief reasoning for the assigned rating."
}
```

---

## Requirements

- Implement **at least 3 different prompting approaches** (your own design).
- Evaluate how each approach affects:
  - Accuracy (Actual vs Predicted)
  - JSON validity rate
  - Reliability and consistency

You must:

- Show each prompt version clearly
- Explain why you changed or improved it
- Evaluate on a sampled set (~200 rows recommended)
- Provide a comparison table + short discussion

---

## TASK 2 - Two-Dashboard AI Feedback System (Web-Based)

Build a simple **web application** with two dashboards.

**Both dashboards must be fully deployed.**

---

### A. User Dashboard (Public-Facing)

Users should be able to:

- Select a star rating
- Write a short review
- Submit it

On submission:

- An AI-generated response should be returned
  - The data should be stored (CSV, JSON file, or lightweight DB)
-

## B. Admin Dashboard (Internal-Facing)

The admin dashboard must display a **live-updating list** of all submissions, including:

- User rating
- User review
- AI-generated summary
- AI-suggested recommended actions

Additionally:

- Admin Dashboard may include **any analytics that you deem useful**

Both dashboards must read/write from the same stored data source.

---

### Requirements

- Dashboards must be **web-based**  
(Any stack: Vercel, Render, HuggingFace Spaces, etc.)
  - LLMs must be used for:
    - Review summarisation
    - Recommended next actions
    - User-facing responses
  - **Deployments are compulsory**  
(Both dashboards must have working online deployments)
- 

## Final Submission Format

Submit the following:

GitHub Repository (must include the notebook):  
Report PDF Link:  
User Dashboard URL:  
Admin Dashboard URL: