TASK 2:FYND AI FEEDBACK SYSTEM – TECHNICAL REPORT

1. Project Overview

This project implements a production-grade, AI-powered customer feedback system for the Fynd e-commerce platform. The system consists of two interconnected web dashboards designed to collect, process, and analyze customer reviews using artificial intelligence.

The **User Dashboard** acts as the public-facing interface where customers can submit:

- Star ratings (1–5)
- Written feedback

Once submitted, the feedback is processed using **Google Gemini 2.5 Flash**, which generates a personalized AI response that is immediately displayed to the customer.

The **Admin Dashboard** provides an internal interface for the operations team. It displays all submitted reviews along with:

- AI-generated summaries
- Recommended actions
- Analytics such as rating distribution and average ratings

The admin dashboard auto-refreshes every 5 seconds to ensure real-time visibility of new feedback.

---

2. Technology Stack

Backend

- Python Flask framework
- Flask-SQLAlchemy for database operations
- Deployed on Render cloud platform

Frontend

- Two separate React.js applications
- Black-and-white branding aligned with Fynd's design language

- User Dashboard deployed on Vercel
- Admin Dashboard deployed on Netlify

Database

- SQLite for persistent storage of reviews and AI-generated data

AI Integration

- Google Gemini 2.5 Flash model
- Server-side integration for:
    - Review summarization
    - Personalized response generation
    - Action recommendations

---

3. Key Features Implemented

User Dashboard Features

- Interactive 5-star rating selector with visual feedback
- Text area for detailed review submission
- Real-time validation for empty inputs
- Display of AI-generated personalized response after submission
- Clear success and error state messaging
- Fully responsive design consistent with Fynd branding

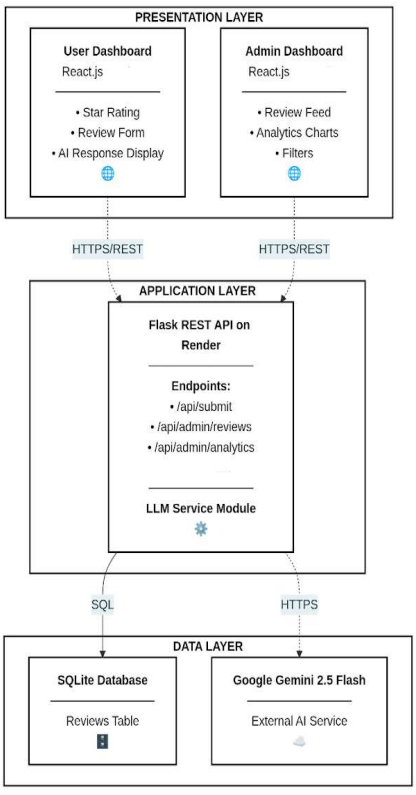
Admin Dashboard Features

- Auto-refreshing review feed (every 5 seconds)
- Display of:
    - Customer rating
    - Review text
    - AI-generated summary
    - AI-recommended action
- Analytics section showing:
    - Total number of reviews
    - Average rating
- Rating distribution bar chart visualization
- Filter functionality based on star ratings
- Color-coded rating indicators:

- Green for positive feedback
- Red for negative feedback

---

4. Technical Architecture



The system follows a **three-tier architecture**:

Presentation Layer

- Two React single-page applications (User and Admin dashboards)
- All UI interactions handled client-side
- Communication with backend via REST APIs using Axios

Application Layer

- Flask-based REST API server
- Exposes four primary endpoints:
    - Submit review
    - Fetch admin reviews
    - Fetch analytics
    - Health check
- All AI processing occurs server-side to ensure API key security

Data Layer

- SQLite database
- Stores the following fields:
    - Star rating
    - Review text
    - AI-generated response
    - AI-generated summary
    - AI-recommended action
    - Timestamp

---

5. API Endpoints

- **POST /api/submit**
  Accepts customer rating and review text, returns AI-generated response
- **GET /api/admin/reviews**
  Returns all submitted reviews for the admin dashboard
- **GET /api/admin/analytics**
  Returns aggregated statistics such as average rating and distribution

---

6. Error Handling

The system includes comprehensive error handling mechanisms:

- Validation for empty reviews and invalid star ratings
- Truncation of reviews exceeding 2000 characters
- Graceful fallback responses if the AI service is unavailable
- Fallback responses are dynamically customized based on the submitted star rating

---

7. Deployment Details

- **Backend API**
  Deployed on Render
  URL: https://fynd-be5l.onrender.com
- **User Dashboard:-** https://fynduser.netlify.app/
- **Admin Dashboard**:- https://fyndadmin.netlify.app/

All deployments are configured with:

- Proper environment variables
- CORS settings to enable secure cross-origin communication between frontend applications and the backend API