

Github link: <https://github.com/Anuvid17/Gen-AI-Project>

## 1. Goal of the Project

### Objective

The goal of this project is to design and develop a **full-stack Generative AI-powered web application** that transforms large volumes of unstructured user feedback into **structured, prioritized, and explainable product roadmap decisions**.

### Problem Statement

Modern software products receive user feedback from multiple sources such as app reviews, support tickets, and surveys. This feedback is:

- unstructured,
- high in volume,
- and difficult to analyze manually.

As a result, product teams struggle to:

- identify genuine feature priorities,
- distinguish critical issues from noise,
- and justify roadmap decisions with data.

### Proposed Solution

The proposed system uses **Generative AI and backend intelligence** to:

- ingest and store raw user feedback,
- extract feature requests and problem statements,
- group similar feedback into product features,
- assign priority scores based on defined logic,
- and generate a draft product roadmap with clear AI-generated justifications.

### Expected Outcome

- Reduced manual effort in feedback analysis
- Data-driven and explainable roadmap prioritization
- Improved alignment between customer needs and product decisions

## 2. Design Patterns Used

### 2.1 Model–View–Controller (MVC)

The backend follows the MVC pattern:

- **Model:** Django ORM models (Feedback, Feature, Roadmap)
- **View:** REST API endpoints (Django REST Framework)
- **Controller:** Business logic and request handling

This pattern ensures clean organization of backend responsibilities.

### 2.2 Pipeline Pattern (AI Processing)

The AI logic is implemented as a **processing pipeline**, where feedback flows through defined stages:

1. Feedback ingestion
2. Text preprocessing
3. Feature and intent extraction using LLM
4. Feedback clustering
5. Priority scoring
6. Roadmap generation

This makes the AI system modular and extensible.

## 3. Technology Used

### Frontend

- **Framework:** React
- **Purpose:** User interface, dashboards, and data visualization
- **Features:** Feedback views, feature priority display, roadmap visualization

### Backend

- **Framework:** Django

- **API Framework:** Django REST Framework (DRF)
- **Purpose:**
  - Handle API requests
  - Execute business logic
  - Coordinate AI processing
  - Persist data

### Generative AI

- **LLM Provider:** OpenAI API (or equivalent)
- **Usage:**
  - Extract features from feedback
  - Summarize user intent
  - Generate roadmap justifications

### Database

- **Type:** Relational Database
- **Options:** PostgreSQL
- **Purpose:** Structured storage of feedback, features, mappings, and roadmap data

### Tools

- Git & GitHub (version control)
- Postman (API testing)
- Django Admin (data inspection and management)

## 4. Database Schema Description

### users

Stores information about system users such as product managers or administrators who access feedback analysis and roadmap insights.

## feedback

Stores raw, unstructured user feedback collected from various sources along with basic metadata such as source and sentiment.

## features

Stores structured product features or improvement themes identified by the AI after analyzing and clustering user feedback, including priority scores.

## feature\_feedback\_map

A junction table that maintains the many-to-many relationship between feedback and features, enabling traceability and explainability of AI decisions.

## roadmap

Stores AI-generated product roadmap entries, linking prioritized features to specific time periods along with justification text explaining why each feature was selected.

