### **Project Title:**

**"Ethical AI-Powered Review Rating System: Ensuring Fairness, Transparency, and Accountability in AI-Generated Product Reviews"**

## **📌 Project Description**

### **🔹 Overview**

The rapid adoption of **Generative AI** in e-commerce, product recommendations, and review platforms has introduced challenges related to **bias, transparency, and accountability** in AI-generated reviews and ratings. Many AI models tend to favor **certain brands, product categories, or price ranges**, leading to unfair consumer perceptions and potential market distortion.

This project aims to develop an **AI-powered review rating system** that ensures **fair and unbiased product evaluations**. By leveraging **Natural Language Processing (NLP), sentiment analysis, and fairness metrics**, we will extract **structured ratings (1-5 stars) from AI-generated reviews** and apply **ethical AI principles** to mitigate biases.

The system will incorporate **SHAP & LIME** to **explain AI-generated review decisions**, **demographic parity checks** to detect brand favoritism, and **differential privacy** to protect user-generated data. Additionally, **human oversight** will be integrated to verify AI recommendations before publication.

## **📌 Abstract**

As AI-driven product recommendations and review generation systems become mainstream, concerns regarding **bias, lack of transparency, and fairness** in AI-generated ratings have emerged. This project proposes an **Ethical AI-Powered Review Rating System** that ensures **accountable AI decision-making in product reviews**.

The system will use **Generative AI (GPT/BART) to generate reviews**, followed by **sentiment analysis (VADER/BERT) to extract star ratings**. To ensure fairness, the project will integrate **SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-Agnostic Explanations) to explain AI rating decisions**, along with **demographic parity metrics to identify biases**.

To maintain **accountability**, the system will include **human-AI collaboration** for review validation and **differential privacy mechanisms** to prevent AI from memorizing personal data. The project will be deployed as a **Flask API**, making it scalable for integration into **e-commerce, review platforms, and AI-driven recommendation engines**.

By addressing **bias, explainability, and ethical AI governance**, this project will contribute towards a **trustworthy AI ecosystem** that ensures **fair and transparent review recommendations** for consumers and businesses alike.

### **📌 Key Features:**

✔ **AI-generated reviews & ratings** (using GPT/BART)  
✔ **Sentiment analysis-based rating extraction** (VADER/BERT)  
✔ **Bias detection using fairness metrics (Demographic Parity, Equalized Odds)**✔ **Explainable AI with SHAP & LIME** for AI-generated reviews  
✔ **Privacy protection using Differential Privacy techniques**✔ **Human oversight for AI-generated review validation**✔ **Deployable Flask API for real-world applications**

## **✅ 1️⃣ Project Architecture**

### **🔹 High-Level Workflow**

1️⃣ **User Inputs a Product Name** → AI generates a review.  
2️⃣ **AI Extracts a Sentiment Score** → Converts sentiment into a **1-5 star rating**.  
3️⃣ **Bias Detection & Explanation** → Use **SHAP & LIME** to explain rating decisions.  
4️⃣ **Fairness Auditing** → Check AI **bias using demographic parity metrics**.  
5️⃣ **Human-AI Oversight** → Allow human validation for AI-generated reviews.  
6️⃣ **Deployment via Flask API** → Make the system **accessible for integration into e-commerce platforms**.

### **System**

This plan includes **project architecture, dataset integration, and deployment strategy** to build a **scalable, ethical AI-powered review rating system** that ensures **fairness, transparency, and accountability** in AI-generated reviews.

## **✅ 1️⃣ Project Architecture**

### **🔹 High-Level Workflow**

1️⃣ **User Inputs a Product Name** → AI generates a review.  
2️⃣ **AI Extracts a Sentiment Score** → Converts sentiment into a **1-5 star rating**.  
3️⃣ **Bias Detection & Explanation** → Use **SHAP & LIME** to explain rating decisions.  
4️⃣ **Fairness Auditing** → Check AI **bias using demographic parity metrics**.  
5️⃣ **Human-AI Oversight** → Allow human validation for AI-generated reviews.  
6️⃣ **Deployment via Flask API** → Make the system **accessible for integration into e-commerce platforms**.

## **✅ 2️⃣ Dataset Integration**

Since **Generative AI does not have direct labeled ratings**, we will build an **AI-generated dataset** using **NLP & sentiment analysis**.

### **🔹 Step 1: Collect Training Data for AI Model**

We will train AI using **a mix of real & synthetic product reviews**:

* **Amazon Review Dataset** (Kaggle) → Labeled reviews with ratings
* **Yelp Open Dataset** → User-generated product & service reviews
* **AI-Synthesized Reviews** → AI-generated reviews for additional brands

📌 **Example Training Data Structure:**

| **Product** | **Review** | **User Rating** |
| --- | --- | --- |
| Brand X Phone | "Great battery but very expensive." | ⭐⭐⭐ |
| Brand Y Laptop | "Fast and smooth, worth the price." | ⭐⭐⭐⭐⭐ |
| Brand Z Tablet | "Decent screen but slow processor." | ⭐⭐ |

✔ **We use AI to generate additional reviews** from existing datasets.

### **🔹 Step 2: Train AI Model to Generate Reviews**

We use **GPT-4/BART** to generate **balanced reviews**:

from transformers import pipeline

# Load AI Review Generator

review\_generator = pipeline("text-generation", model="facebook/bart-large-cnn")

def generate\_review(product\_name):

prompt = f"Write a fair and objective review of {product\_name}, covering both pros and cons."

review = review\_generator(prompt, max\_length=50, do\_sample=False)[0]["generated\_text"]

return review

✔ **Ensures AI-generated reviews are unbiased** by prompting for both pros & cons.

### **🔹 Step 3: Extract Ratings from AI-Generated Reviews**

We **convert AI-generated text into a numerical rating** (1-5) using **VADER Sentiment Analysis**.

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# Load Sentiment Analyzer

analyzer = SentimentIntensityAnalyzer()

def extract\_rating(review\_text):

sentiment\_score = analyzer.polarity\_scores(review\_text)["compound"] # -1 to 1

return round((sentiment\_score + 1) \* 2.5) # Convert to 1-5 scale

✔ **Maps AI-generated reviews into star ratings.**

### **🔹 Step 4: Store AI Ratings in a Dataset**

We create a structured dataset from AI-generated ratings.

import pandas as pd

# List of Brands for AI Review Generation

brands = ["Brand X", "Brand Y", "Brand Z"]

# Generate AI Reviews and Ratings

data = {"brand": [], "ai\_review": [], "ai\_rating": []}

for brand in brands:

review = generate\_review(brand)

rating = extract\_rating(review)

data["brand"].append(brand)

data["ai\_review"].append(review)

data["ai\_rating"].append(rating)

# Convert to DataFrame

df = pd.DataFrame(data)

df.to\_csv("ai\_generated\_ratings.csv", index=False)

print(df)

✔ **This dataset will be used to train & validate the AI review rating system.**

## **✅ 3️⃣ Bias Detection & Fairness Auditing**

### **🔹 Step 5: Detect Bias Using SHAP**

We use **SHAP to explain AI-generated ratings** and detect bias.

import shap

# SHAP Explanation of AI-generated Review

explainer = shap.Explainer(review\_generator)

shap\_values = explainer(["This smartphone has great battery life but is expensive."])

shap.summary\_plot(shap\_values)

✔ **Identifies words that influence AI rating decisions.**

### **🔹 Step 6: Fairness Auditing Using Demographic Parity**

We check **if AI ratings systematically favor certain brands**.

python

CopyEdit

from fairlearn.metrics import demographic\_parity\_difference

# Check fairness across brands

dpd = demographic\_parity\_difference(df["ai\_rating"], df["brand"], sensitive\_features=df["brand"])

print("Demographic Parity Difference:", dpd)

✔ **If DPD > 0.1, AI ratings are biased and need correction.**

## **✅ 4️⃣ Bias Correction & Human-AI Collaboration**

### **🔹 Step 7: Normalize AI Ratings for Fairness**

We adjust AI ratings if bias is detected.

def normalize\_ratings(brand, predicted\_rating):

bias\_correction = {"Brand X": -0.2, "Brand Y": +0.2} # Adjust detected bias

return max(1, min(5, predicted\_rating + bias\_correction.get(brand, 0)))

df["rating\_corrected"] = df.apply(lambda row: normalize\_ratings(row["brand"], row["ai\_rating"]), axis=1)

print(df)

✔ **Ensures AI-generated ratings remain fair.**

### **🔹 Step 8: Implement Human Review for AI Ratings**

We introduce **human oversight** in AI-generated review validation.

def human\_review\_needed(ai\_rating, brand):

if ai\_rating < 3: # Flag negative reviews for human review

return f"Human review needed for {brand}'s AI rating."

return f"AI rating for {brand} approved."

print(human\_review\_needed(2, "Brand X"))

✔ **Allows human intervention before publishing AI-generated reviews.**

## **✅ 5️⃣ Deployment Strategy**

We deploy this system as a **Flask API**, making it accessible for integration into **e-commerce platforms**.

### **🔹 Step 9: Build Flask API**

from flask import Flask, request, jsonify

app = Flask(\_\_name\_\_)

@app.route('/generate\_review', methods=['POST'])

def generate\_review\_api():

data = request.json

product\_name = data.get("product")

review = generate\_review(product\_name)

rating = extract\_rating(review)

return jsonify({

"product": product\_name,

"ai\_review": review,

"ai\_rating": rating

})

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

✔ **Allows external applications to request AI-generated reviews & ratings.**