Emotion Analysis API – Backend Report

# 1. Introduction

This document provides a summary of the backend implementation of an Emotion Analysis API developed using Flask, a lightweight Python web framework. The application simulates emotion detection based on input text and responds with a randomly selected emotion and a confidence score.

# 2. Technologies Used

- Python 3.x

- Flask

- Flask-CORS

# 3. Code Overview

## 3.1 API Endpoint

The API exposes a single POST endpoint '/analyze'. This endpoint accepts a JSON payload containing a 'text' key. The backend then returns a simulated emotion and a confidence score.

## 3.2 Random Emotion Generator

The backend uses a predefined list of emotions: ['Happy', 'Sad', 'Anxious', 'Excited', 'Angry']. It randomly selects one of these emotions and assigns a confidence score between 0.7 and 0.99.

## 3.3 CORS Configuration

CORS (Cross-Origin Resource Sharing) is enabled using Flask-CORS to allow communication between frontend and backend services hosted on different origins.

## 3.4 Running the App

The application is configured to run on port 8000 in debug mode. To start the server, use:  
`python app.py`

# 4. Sample Input and Output

## 4.1 Sample Input (JSON)

{  
 "text": "I have a lot on my mind lately."  
}

## 4.2 Sample Output (JSON)

{  
 "emotion": "Anxious",  
 "confidence": 0.92  
}

# 5. Conclusion

This backend provides a mock implementation of an emotion analysis service. While it currently returns randomized emotions for demonstration purposes, the structure can easily be adapted to integrate real machine learning models in the future.