**Project Title**: YouTube Video Analyzer (Frontend using HTML, CSS, JS)

**Prepared By**: Sadew Deshan  
**Date**: 2025/08/04

**1. Introduction**

**1.1 Purpose**

The purpose of this project is to create a simple frontend web application that analyzes a specific YouTube video using the YouTube Data API. The application will fetch and display data such as video title, description, views, likes, and top comments in a user-friendly layout.

**1.2 Scope**

This application will:

* Allow the user to input a YouTube video URL or ID.
* Use JavaScript to call the YouTube Data API and fetch relevant information.
* Display fetched data using clean HTML and styled with CSS.
* Be responsive and lightweight, working on both desktop and mobile browsers.

**1.3 Definitions, Acronyms, and Abbreviations**

* **API**: Application Programming Interface
* **HTML**: HyperText Markup Language
* **CSS**: Cascading Style Sheets
* **JS**: JavaScript
* **YouTube Data API**: A Google API used to access YouTube content and statistics

**2. Overall Description**

**2.1 Product Perspective**

This is a standalone frontend application with no server-side backend. It communicates directly with the YouTube Data API.

**2.2 Product Functions**

* Input field for YouTube video ID or URL
* Fetch button to trigger API request
* Ability to watch the video
* Display section for video:
  + Title
  + Channel Name
  + Description
  + View count
  + Like count
  + Top 3 comments

**2.3 User Characteristics**

The users are general web users, YouTubers, or marketers interested in quick video insights.

**2.4 Constraints**

* Requires an API key from Google Developers Console
* Limited by YouTube API quotas
* Only public videos can be analyzed

**2.5 Assumptions and Dependencies**

* Users will input valid YouTube video URLs or IDs
* The internet connection is available
* Google’s YouTube API remains operational

**3. Specific Requirements**

**3.1 Functional Requirements**

* **FR1**: User can enter a YouTube video URL or ID.
* **FR2**: The app extracts the video ID if a full URL is given.
* **FR3**: On clicking "Analyze", JS sends a request to YouTube API.
* **FR4**: The app parses and displays the video information.
* **FR5**: If data retrieval fails, show an error message.

**3.2 Non-Functional Requirements**

* **NFR1**: The UI must be responsive and mobile-friendly.
* **NFR2**: API requests should be asynchronous (using fetch or async/await).
* **NFR3**: The system should display results within 2 seconds (on a good connection).
* **NFR4**: Use semantic HTML and accessible CSS.

**3.3 External Interface Requirements**

* **YouTube Data API v3**
  + Endpoint: https://www.googleapis.com/youtube/v3/videos

Parameters: part=snippet,statistics, id=VIDEO\_ID, key=AIzaSyBocojCJEATjPwb7Jv53GHCjBs5zq4TjZg