# Product Requirements Document (PRD)

Product name: YouTube with smart chatbot

Prepared by: Jual Sunil

Date: 08-04-2025

## 1. Objective

Create a video-sharing platform replicating all core functionalities of YouTube, with the addition of a chatbot powered by a Large Language Model (LLM). The chatbot will assist users by recommending videos, summarizing content, and enabling content discovery.

## 2. Key Features

### A. Core YouTube Clone Features

#### 1. User Authentication & Profile

• Email/password signup & login

• Google OAuth

• Profile customization (avatar, bio)

• User dashboard with uploads, playlists, and history

#### 2. Video Upload & Processing

• Upload with progress bar

• Thumbnail generation/selection

• Video transcoding (resolutions)

• Title, description, tags, category inputs

• Content moderation system

#### 3. Video Playback

• Playback speed & resolution controls

* Responsive Video.js player for external sources

• Closed captions support

• Likes/dislikes, view count tracking

#### 4. Video Embedding via Scraping (using Video.js)

* Add videos via external URLs (YouTube, Vimeo, etc.)
* Video metadata (title, description, tags, category)
* Video.js player integration for seamless playback
* URL validation and safety checks

#### 5. Search & Discovery

• Keyword search

• Trending, categories, and tags

• Filters

#### 6. Comments

• Basic comment writing

#### 7. Subscriptions & Notifications

• Subscribe to channels

• Bell notification system

#### 8. Playlists & Watch Later

• Create/manage playlists

• Add to “Watch Later”

• Shareable playlists

### B. LLM Chatbot Assistant Features

#### 1. Natural Language Chat Interface

• Persistent chat UI

• Contextual awareness (e.g., understanding what the user is currently watching)

#### 2. Personalized Recommendations

• Based on user mood, interests, and history

• Example: “Show me relaxing ambient music”

#### 3. Content Navigation & Discovery

• Assist with search and channel discovery

• Example: “Find videos by VTU padhai about 21CS72”

#### 5. Learning Companion Mode

• Explains educational content

• Answers follow-up questions

• Provides additional links/resources

#### 7. Safety & Moderation

• Filter inappropriate questions

• Enforce safe and accurate responses

• User data protection

## 3. Technical Stack

|  |  |
| --- | --- |
| **Layer** | **Technology / Tool** |
| **Frontend** | HTML, TailwindCSS (or Bootstrap), Jinja2 templates (or React) |
| **Video Player** | [Video.js](https://videojs.com) — handles both user-uploaded and scraped videos |
| **Backend API** | FastAPI |
| **Database** | SQLite (for local/dev) → PostgreSQL (for production) |
| |  | | --- | | **Authentication** | | FastAPI + JWT (using Authlib) |
| |  | | --- | | **ORM** | | SQLAlchemy |
| |  | | --- | | **Chatbot (LLM)** | | OpenAI GPT API (or alternatives like Groq or Claude) |
| |  |  | | --- | --- | | **Video upload** |  | | Upload via HTML form → store locally or in cloud |
| |  |  | | --- | --- | | **Video scraping** |  | | requests, BeautifulSoup, or Playwright/Selenium to scrape video links + metadata |
| **Video embedding** | Embed scraped video URLs using Video.js (YouTube/Vimeo plugin) |
| **Metadata storage** | Title, description, thumbnail, source, video URL saved in your DB |
| |  | | --- | | **Search** | | SQL LIKE queries initially → full-text search (Postgres FTS or Meilisearch) later |
| |  | | --- | | **File Handling** | | Python aiofiles or shutil for saving and serving uploaded videos |
| |  |  | | --- | --- | | **Hosting** |  | | Render, Railway, Fly.io, or localhost (for development) |