Here is the complete **Software Requirements Specification (SRS)** for **VibeScale AOS v1.0**, formatted as a formal technical document. You can copy-paste this into Google Docs or Microsoft Word to serve as your project bible.

---

# Software Requirements Specification (SRS)

## Project: VibeScale Agency Operating System (AOS)

Version: 1.0

Date: December 28, 2025

Lead Developer: [Your Name]

Stakeholders: Founding Team (3 Partners)

---

## 1. Introduction

### 1.1 Purpose

The purpose of the VibeScale AOS is to automate the production of high-retention social media assets for small business clients. The system transforms raw client data into two specific output formats:

1. **Viral "Brainrot" Reels:** Vertical videos featuring Hinglish audio, split-screen visuals, and kinetic subtitles.
2. **Static Ad Creatives:** High-definition product images with perfectly rendered marketing text.

### 1.2 Scope

The system acts as a "Hybrid Creative Engine." It creates a web-based dashboard (Streamlit) hosted on a cloud GPU server (RunPod). This dashboard allows non-technical users (the agency founders) to input client details, review AI-generated scripts, and trigger the rendering of video and image assets. The system leverages Google Drive for asset storage and transfer.

### 1.3 Definitions & Acronyms

- **AOS:** Agency Operating System.
- **Hinglish:** Hindi language written in Roman (English) script.
- **Brainrot Content:** High-stimulation video style featuring split screens (gameplay + content) and fast pacing.
- **Nano Banana:** Internal codename for Google Gemini 2.5 Flash Image generation model.
- **HITL:** Human-in-the-Loop (The manual review process).

# 2. Overall Description

## 2.1 Product Perspective

VibeScale AOS is a cloud-native application. It runs on a leased GPU instance (RunPod) but interfaces with Google Drive for persistent storage. It utilizes external APIs (Groq, Google GenAI) for intelligence and local libraries (FFmpeg, EdgeTTS) for media processing.

## 2.2 User Classes and Characteristics

- **Admin (Developer):** Full access to the Python codebase, RunPod terminal, and API configurations.
- **Operator (Agency Founders):** Access via the Streamlit Web Interface only. Responsibilities include inputting client data, editing scripts, and downloading final files.

## 2.3 Operating Environment

- **Server:** RunPod Community Cloud (Linux / Ubuntu).
- **Hardware:** NVIDIA RTX 3090 (24GB VRAM) or equivalent.
- **Storage:** Google Drive (2TB Plan) mounted via Rclone.
- **Client Side:** Any web browser (Chrome/Edge) to access the Dashboard.

# 3. System Features (Functional Requirements)

## 3.1 Module I: The "Brainrot" Video Engine

**Description:** Generates 30-60 second vertical videos (`.mp4`) optimized for Instagram Reels/YouTube Shorts.

- **FR-V1 (Script Generation):** The system shall utilize Llama-3 (via Groq API) to generate scripts based on a provided topic.
  - *Constraint:* Output must be in **Hinglish** (Roman script).
  - *Tone:* Must use Delhi/Mumbai slang (e.g., "Bhai," "Scene," "Jugaad").
- **FR-V2 (Audio Synthesis):** The system shall convert text to speech using Microsoft EdgeTTS.
  - *Constraint:* Must use **`en-IN` (Indian English)** voices (e.g., `en-IN-PrabhatNeural`) to ensure correct pronunciation of Hinglish terms.
- **FR-V3 (Visual Compositing):** The system shall use FFmpeg/MoviePy to create a vertical (9:16) composition.
  - *Top Half:* Randomly selected B-roll from the specific client's folder.
  - *Bottom Half:* Randomly selected "Satisfying" loop (e.g., Subway Surfers) from the Asset Library.
- **FR-V4 (Dynamic Captioning):** The system must burn subtitles onto the video.
  - *Style:* Karaoke-style (word-level highlighting or fast pop-up).
  - *Font:* High-contrast sans-serif (e.g., The Bold Font or Impact) with black stroke.

### 3.2 Module II: The "Nano Banana" Image Engine

**Description:** Generates high-quality product advertisements (.jpg/.png).

- **FR-I1 (Visual Generation):** The system shall utilize Google Gemini 2.5 Flash (via google-genai library) to generate photorealistic images.
- **FR-I2 (Text Rendering):** The system must be capable of rendering legible, correctly spelled text (e.g., "SALE", "OFFER") directly into the image (e.g., on neon signs, chalkboards).
- **FR-I3 (Ad Logic):** The prompt engineering module must enforce "Ad Composition" rules (Product centered, negative space for logos).

### 3.3 Module III: The Dashboard (Human-in-the-Loop)

**Description:** A Streamlit-based User Interface for controlling the generation process.

- **FR-D1 (Client Configuration):** A form input to capture:
    - Business Name.
    - USP / Offer Details.
    - Target Vibe (e.g., "Aggressive," "Professional," "Meme").
- **FR-D2 (The Edit Loop):**
    - The system shall display 3 AI-generated script variations.
    - The system shall provide an editable text box for the user to modify the chosen script **before** audio generation.
- **FR-D3 (Preview & Download):** The dashboard must allow users to view generated images/videos and provide a direct download link or save to the "Processed" folder in Google Drive.

---

## 4. External Interface Requirements

### 4.1 Hardware Interfaces

- **GPU:** The system requires a CUDA-enabled GPU with minimum 16GB VRAM (Recommended: 24GB RTX 3090) to handle batch rendering and local AI models if migrated from APIs.

### 4.2 Software Interfaces

- **Operating System:** Ubuntu 20.04 or later (RunPod Template).
- **Language:** Python 3.10+.
- **APIs:**
    - **Groq API:** For Llama-3-70B text generation.
    - **Google GenAI API:** For Gemini Image generation.
- **Libraries:** streamlit, moviepy, ffmpeg-python, edge-tts, google-genai, rclone.

### 4.3 Communication Interfaces

- **Rclone:** Uses HTTPS to synchronize the local /workspace/assets folder with the remote Google Drive.

- **Streamlit:** Uses a localized web server port (default 8501) exposed via public URL (ngrok or RunPod proxy) for user access.

---

# 5. Non-Functional Requirements

## 5.1 Performance

- **Rendering Speed:** A 60-second video must render in under **60 seconds** utilizing GPU acceleration (NVENC).
- **Latency:** Script generation (Text) must complete in under **5 seconds**.

## 5.2 Reliability & Persistence

- **Storage Safety:** All source assets and final outputs must be stored on **Google Drive**. The RunPod instance is considered ephemeral (temporary) and may be terminated at any time without data loss.

## 5.3 Scalability

- **Batch Processing:** The system must be capable of queuing a "Batch Job" (e.g., "Generate 20 variations of this script") without crashing due to memory leaks.

---

# 6. Architecture Diagram (Textual Representation)

Plaintext

```
[ USER (Cousins) ]
      |
      v
[ STREAMLIT DASHBOARD (UI) ] <--- hosted on ---> [ RUNPOD GPU SERVER (RTX 3090) ]
      |                                  |
      |-- 1. Input Client Data                   |-- 5. Render Video (FFmpeg GPU)
      |-- 2. Edit/Approve Script                  |-- 6. Generate Image (Gemini)
      |                                  |
      v                                  v
[ PYTHON LOGIC CORE ]                      [ GOOGLE DRIVE (Storage) ]
      |                                  ^
      |-- A. Text Gen (Groq API)                  |
      |-- B. Audio Gen (EdgeTTS)                   | (Mounted via Rclone)
      |-- C. Image Gen (Google GenAI)                |
                                         |
                                  [ ASSET LIBRARY ]
                                  (Subway Surfers, Music, Fonts)
```