# ⚠Downloading Failed⚠

Name ⇒> 002 Class-02 Tense
Url ⇒> https://www.youtube.com/embed/K6abhE4SaqA

Failed Reason: could not convert string to float: b'002) Class-02 Tense: No such file or directory\n' not able to downlaod yt

I can see the exact problem! The error is happening in the `duration` function in your `saini.py` file. The issue is that the filename contains special characters (like parentheses and spaces) that are causing the ffprobe command to fail.

### The Problem:

```
def duration(filename):
    result = subprocess.run(['ffprobe', '-v', 'error', '-show_entries', 'format=duration'
    return float(result.stdout)
```

The filename `"002) Class-02 Tense"` has parentheses and spaces that break the command.

### Quick Fix for Your Code:

**Replace the `duration` function in your `saini.py` with this fixed version:**

```
def duration(filename):
    """Get video duration with proper filename handling"""
    try:
        # Escape the filename properly for shell commands
        escaped_filename = filename.replace('"', '\\"')

        result = subprocess.run([
            'ffprobe',
            '-v', 'error',
            '-show_entries', 'format=duration',
            '-of', 'default=noprint_wrappers=1:nokey=1',
            escaped_filename
        ], stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True)

        if result.returncode != 0:
            print(f"ffprobe error: {result.stderr}")
            return 0  # Return default duration

        duration_str = result.stdout.strip()
        if duration_str and duration_str != 'N/A':
            return float(duration_str)
        else:
            return 0  # Default duration if unavailable

    except (ValueError, subprocess.SubprocessError) as e:
        print(f"Duration extraction error: {e}")
        return 0  # Return default duration on error
```

### Alternative Fix - Sanitize Filenames:

**Also add this function to clean filenames in your `saini.py`:**

```
import re

def sanitize_filename(filename):
    """Clean filename for safe file operations"""
    # Remove or replace problematic characters
    filename = re.sub(r'[<>:"/\\|?*]', '', filename)  # Remove invalid chars
    filename = re.sub(r'[()[\]]', '', filename)       # Remove brackets
    filename = re.sub(r'\s+', ' ', filename)          # Replace multiple spaces
    filename = filename.strip()                       # Remove leading/trailing spaces
    return filename[:200]  # Limit length to prevent long filename issues
```

### Update Your YouTube Handler:

**In your `youtube_handler.py`, replace the name generation part:**

```
# Find this line around line 80-90:
name = f"{audiotitle[:60]} {CREDIT}"

# Replace with:
audiotitle_clean = sanitize_filename(audiotitle)
name = f"{audiotitle_clean[:60]} {CREDIT}"
```

### Complete Fix for YouTube Downloads:

**Replace your YouTube download section in `youtube_handler.py`:**

```
# Around line 100-120, replace the YouTube download logic:
if "youtube.com" in url or "youtu.be" in url:
    prog = await m.reply_text(f"▯**Audio Downloading**▯\n\n**{str(count).zfill(3)}** `{na

    # Clean the filename
    clean_name = sanitize_filename(name)

    cmd = f'yt-dlp -x --audio-format mp3 --cookies "{cookies_file_path}" "{url}" -o "{cle
    print(f"Running command: {cmd}")

    try:
        result = subprocess.run(cmd, shell=True, capture_output=True, text=True)

        if result.returncode == 0 and os.path.exists(f"{clean_name}.mp3"):
            await prog.delete(True)
            print(f"File {clean_name}.mp3 exists, attempting to send...")

            try:
                await bot.send_document(
                    chat_id=m.chat.id,
                    document=f"{clean_name}.mp3",
                    caption=f"**Title:** `{str(count).zfill(3)} - {name1}.mp3`\n**Video I
                )
                os.remove(f"{clean_name}.mp3")
                count += 1
            except Exception as e:
                await m.reply_text(f"❌**Downloading Failed**❌\nName =>> {str(count).zfil
                count += 1
        else:
            await prog.delete(True)
            await m.reply_text(f"❌**Downloading Failed**❌\nName =>> {str(count).zfill(3)
            count += 1

    except Exception as e:
        await m.reply_text(f"❌**Failed Reason**❌\n\n{str(e)}")
```

### Quick Test Fix:

**For immediate testing, just add this at the top of your `saini.py`:**

```
def duration(filename):
    """Simple duration function that doesn't crash"""
    try:
        return 60  # Return default 60 seconds for all videos
    except:
        return 60
```

The main issue is that your filename `"002) Class-02 Tense"` contains special characters that break the ffprobe command. The fixes above will handle this properly and prevent crashes.

⁂

1. saini.py
2. youtube_handler.py