**AI-Powered 3D Character Agent in Camera Kit Web - Workflow Documentation**

---

## Description

Interactive AI-powered 3D character using Snapchat's **Camera Kit Web SDK** and **Lens Studio**, where:

- The user speaks into the mic.
- A **custom API (5DVR APIs)** handles audio-to-text transcription, ChatGPT response and returning the response in audio format.
- The character performs realistic **lipsync** using **rhubarb-lip-sync** extracted from the audio file.

---

## High-Level Workflow

```
1. User speaks into mic
2. Web app sends audio to custom API
3. API returns GPT-based text reply
4. Web app calls custom TTS API to generate MP3
5. Convert MP3 to WAV (using ffmpeg)
6. Run rhubarb-lip-sync on WAV to get viseme/phoneme JSON
7. Web app plays the MP3 via <audio> or Web Audio API
8. Web app sends viseme JSON to Lens Studio
9. Lens Studio animates the 3D character's mouth (blendshapes) in sync with
viseme timing
```

---

## Components Breakdown

| Step | Description | Tools |
|---|---|---|
| 1 | Record user voice | Web Audio API / getUserMedia |
| 2 | Speech-to-text + GPT | 5DVR custom API |
| 3 | Text-to-speech | 5DVR custom API |
| 4 | Audio conversion | `ffmpeg` / `fluent-ffmpeg` (MP3 → WAV) |
| 5 | Viseme extraction | `rhubarb-lip-sync` (open-source) |
| 6 | Audio playback | HTML5 `<audio>` or Web Audio API |
| 7 | Lens connection | `CameraKit.lens.executeScript()` |
| 8 | Blendshape animation | JavaScript inside Lens Studio lens |

---

## Lens Studio Integration

### Receiving viseme JSON:

```
lens.executeScript(`startLipsync(${JSON.stringify(visemeArray)})`);
```

### Blendshape Mapping Table (example):

```
const visemeToBlendshape = {
  "A": { mouthOpen: 0.8 },
  "B": { lipsTogether: 1.0 },
  "C": { mouthWide: 1.0 },
  ...
};
```

### Animation Engine (Lens Studio):

- Use a `ScriptComponent` to animate the blendshapes at correct timestamps.
- Drive animation with `getTime()` + `updateEvent`

---