Task 3: Avatar Coaching Assistant (Webapp)

📅 Date 🔗

Jul 6, 2025

👥 Participants 🔗

- **@TASNIM MAHDIYA / UPM**

💡 [Task 3 documentation link](#) 🔗

🎵 [Audio to convert speech to text (test)](#)

📌 n8n workflow 🔗

Task3 - Avata... ow.json
06 Jul 2025, 07:39 PM

🧠 Project Overview 🔗

This project implements a **Conversational Avatar Coaching Assistant**, simulating a human-like coach that engages with users via a voice-based chat interface. The assistant receives user voice input, processes it with an LLM, generates a coaching response, creates an avatar video, and is meant to deliver the video to a user-facing UI.

🛠️ Tech Stack 🔗

🧩 Frontend Stack: 🔗

Tech	Purpose
React + MUI	Responsive web interface for viewing avatar
Axios	API call to backend `n8n` to fetch video URL
Vite	Build tool for fast dev environment

🌐 Workflow Architecture 🔗

```
1  User Input (Webhook Trigger)
2      ↳ LLM - DeepSeek
3          ↳ Code (Parse JSON & Format Date)
4              ↳ Generate Avatar Video (HeyGen)
5                  ↳ Store Video ID + Status
6                      ↳ LOOP: Get Video Status (Wait every 20s, max X loops)
7                          └──[IF loop count ended || video Completed?]
```

```
  8                      |↳ Record to Airtable (both)
  9                      └──[IF Completed?]
 10                          |↳ True: Use Generated URL
 11                          └─↳ False: Use Fallback Video URL
 12                              ↳ Respond to Webhook (return final URL)
 13
```

⚙️ Backend Logic (n8n) 🔗

1. Webhook Trigger 🔗

- Method: POST
- Accepts user input to begin avatar coaching session

2. DeepSeek LLM 🔗

- Prompt: Coaching assistant behavior
- JSON Output containing reply text

3. Generate Avatar Video (HeyGen API) 🔗

```
 1  POST https://api.heygen.com/v2/video/generate
 2  Headers:
 3    X-Api-Key: <API_KEY>
 4  Body:
 5  {
 6    "video_inputs": [
 7      {
 8        "character": {
 9          "type": "avatar",
10          "avatar_id": "Lina_Dress_Sitting_Side_public",
11          "avatar_style": "normal"
12        },
13        "voice": {
14          "type": "text",
15          "input_text": "{{ $node['LLM - Coach
    Reply'].json.choices[0].message.content.replace(/\n/g, ' ').replace(/\"/g, '\\\"') }}",
16          "voice_id": "119caed25533477ba63822d5d1552d25",
17          "speed": 1
18        }
19      }
20    ],
21    "dimension": {
22      "width": 1280,
23      "height": 720
24    },
25    "test": true
26  }
27
```

4. Loop: Check Video Status 🔗

- GET from: `https://api.heygen.com/v1/video_status.get?video_id={{ $json.data.id }}`
- Wait Node: 20s between loops
- Loop Max: 30

5. IF Node (Status == completed) 🔗

- True: Store video_url from API
- False: Use fallback video URL (e.g., HeyGen demo)

6. Airtable Record 🔗

- Fields: Prompt, Video URL, Status

7. Respond to Webhook (Send to UI) 🔗

```
1  {
2    "video_url": "{{ $json.video_url ||
   'https://cdn.heygen.com/videos/heygen_default_demo.mp4' }}"
3  }
4
```

🖥️ Frontend: Avatar Viewer UI (MUI) 🔗

❌ Known Frontend Limitation 🔗

- **Backend-Frontend Connection Failed**:
  - Workflow Webhook node (Respond to UI) expects manual trigger or frontend-initiated call
  - Not possible to test automatically in time-limited environment
  - No real-time communication possible due to webhook not being triggered correctly

Libraries Used 🔗

```
1  @mui/material
2  @emotion/react
3  @emotion/styled
4
```

Key File: `App.jsx` 🔗

```
1  useEffect(() => {
2    fetch(BACKEND_URL, { method: 'POST' })
3      .then(res => res.json())
4      .then(data => setVideoUrl(data.video_url))
5      .catch(() => setError(true));
6  }, []);
7
```

🔄 Backend-Frontend Integration 🔗

- **Webhook Path**: `/return-video`
- **Type**: Respond to Webhook
- **Frontend expects**:

```
1  {
2    "video_url": "https://api.heygen.com/..."
3  }
4
```

- No authentication required for internal test

⚠️ Limitations 🔗

Area	Limitation / Workaround
🧠 LLM	DeepSeek can parse and reply properly, but may return date as `"2024"` which was manually fixed to `"2025"` in the code
🧍 Avatar	**HeyGen Free Tier allows 5 runs/day**. After that: `"AVATAR_USAGE_NOT_PERMITTED"` error
🔄 Looping	`loop_count` logic breaks due to re-execution context loss — stays at `1`
🔁 Video Processing	Can take up to **2–3 minutes** or longer; often exceeds real-time demo constraints
🎥 Video URL	Sometimes HeyGen dashboard doesn't show the video even when generation succeeds — URL must be extracted manually
⚠️ Webhook	Workflow Webhook node is not properly connected to frontend — real-time user interaction is not available in final build

📦 Sample Fallback URL Used 🔗

```arduino
1  arduino
```

CopyEdit

```
https://files.heygen.com/fallbacks/motivational-avatar-message.mp4
```

Shown when video generation fails or times out.

📂 Sample Output Sent to Airtable 🔗

Field	Value (Example)
ID	80bee38ea8944f37967dbce7fabf5e24
Audio Transcript	A king ruled the state in early days...
AI Reply	It sounds like you're sharing vivid imagery...

Video URL	[HeyGen generated Video URL](...)...
Status	completed
Notes	Loop #1 (fallback used or not)

Sample proof in Airtable

✅ Features Highlight 🔗

- Full coaching assistant experience
- Humanlike avatar using HeyGen
- LLM-integrated dynamic response
- Multilingual support (Malay + English)
- Airtable logging for all sessions
- Seamless UI integration
- Robust fallback logic with loops