

TÍNH NĂNG 1 — TỰ QUAY trận đấu & trigger highlight (bằng voice hoặc nút)

Vấn đề anh đang gặp:

- Khó thiết kế UI: vừa quay video toàn trận, vừa bắt highlight bằng voice hoặc nút mà không làm người chơi bị rối.
- Khó xử lý logic: liên tục *cắt clip 30–60 giây trước* mỗi lần có trigger mà không lag, không tốn bộ nhớ.

Tôi đưa ra **luồng giải pháp tối ưu nhất**:

Kiến trúc xử lý video đúng chuẩn (giúp app mượt)

App **không được** cắt clip ngay lúc người dùng bấm hoặc nói “highlight”.

Lý do:

- Cắt video trực tiếp trên mobile → dừng app → CPU nóng → dropout frame → crash sau 20–40 phút quay.

Cách đúng:

Quay suốt trận đấu = buffer 1 video duy nhất (1 file)

Highlight = metadata bookmark, không rờ vào video ngay lúc bấm

📌 Khi user nói keyword “highlight” → event: {timestamp = 17:22:14}

📌 Khi user nhấn nút “Lưu highlight” → event: {timestamp = 24:01:08}

Toàn bộ sự kiện được ghi lại dưới dạng danh sách *highlight events*.

👉 Sau khi trận kết thúc → backend chỉ việc:

- mở file video full
- cắt lại từng đoạn dựa trên timestamps
- 30–60 giây backward mỗi event
- xuất thành các video highlight hoàn chỉnh



→ App luôn mượt

→ Không tiêu tốn tài nguyên lúc đang quay

UI/UX — Giao diện đẹp và dễ dùng

Màn hình quay trận (chế độ camera)

UI chỉ có 2 hành động lớn — không phân tán người dùng

-  Nút “Lưu highlight” (tap một lần)
-  Mic hiển thị trạng thái “voice highlight ON”

Không để nhiều nút, không có thanh công cụ, không cho filter khi đang quay.

Sắc màu

- Nút Highlight = đỏ / nổi bật
- Info nhỏ góc phải: “Highlight đã lưu • 12s” (fade sau 1s)

Tâm lý người chơi

Họ muốn chơi, không muốn “điều khiển app”.

UI phải 100% tập trung vào việc: “**1 chạm hoặc 1 câu nói là xong**”

Màn hình review sau trận

Hiển thị dạng danh sách thẻ video:

Ví dụ:

Highlight 1 – 35 giây (timestamp 07:20)

Highlight 2 - 41 giây (timestamp 13:54)

Highlight 3 - 48 giây (timestamp 23:10)

Mỗi thẻ có:

- Play
- Edit (cắt + thêm text + hiệu ứng)
- Add to compilation (đưa vào video highlight cuối)

Cuối cùng: **1 nút “Tạo video highlight hoàn chỉnh”**

→ Render video tổng → share / download

Công nghệ phù hợp nhất cho tính năng này

- Video quay: **1 file duy nhất** (.mp4 / h.264 / 1080p hoặc 720p)
- Lưu buffer timestamp trong SQLite hoặc local JSON
- Voice trigger: dùng **offline speech keyword detector** → không cần internet → nhanh → không tốn pin
 - Kỹ thuật phù hợp: **Porcupine / Vosk / Whisper tiny offline**
- Sau trận: upload video + metadata → backend cắt highlight bằng **FFmpeg** (nhanh, rẻ, không GPU)

FFmpeg command mẫu

```
ffmpeg -i fullmatch.mp4 -ss (timestamp - 00:00:45) -t 60  
highlight1.mp4
```

→ siêu rẻ, siêu nhanh, không cần AI ở step này.

● TÍNH NĂNG 2 — Quét QR tại sân → camera sân bật → AI highlight tự động

Vấn đề anh kệt:

- Luồng người dùng: thanh toán xong, camera sao tự bật?
- Công nghệ: kết nối giữa app với camera ngoài sân như thế nào?

Tôi cho anh cấu trúc đúng, đã chứng minh là triển khai được:

🧩 Luồng xử lý chính xác (B2B2C)

- 1) Người dùng quét mã QR → app nhận **Court_ID**
- 2) Người dùng chọn gói → thanh toán → backend nhận **OK**
- 3) Backend gửi **command** tới **Camera_agent** tại sân
- 4) Camera_agent bắt đầu stream/record → upload đến storage
- 5) Kết thúc giờ → Camera_agent dừng record
- 6) Video full được gửi sang **Video_processing_service**
- 7) AI highlight tự động → gộp clip → gửi về app người dùng

← người dùng hầu như **không phải thao tác gì** ngoài quét QR và thanh toán.

🔌 Camera agent — chìa khóa để hệ thống chạy được

Camera ở sân **không được điều khiển trực tiếp từ app**.

Cách đúng:

- Mỗi camera chạy 1 dịch vụ nhỏ gọi là **camera agent** (Raspberry Pi / mini PC / Android box)
- Agent kết nối **1 chiều tới backend** (websocket hoặc polling)
- Backend gửi lệnh “start / stop” là đủ

Đây là nơi mọi startup sports-tech đều làm sai:

- ✗ cố điều khiển camera trực tiếp từ app → NAT/IP/firewall vỡ mặt
 - ✓ đúng là: agent cài trong sân, nghe lệnh từ server
-

AI highlight cho chế độ QR tại sân

Ở mô hình quét QR, không có nút/voice trigger → phải có AI tracking.

Luồng gọn nhất và rẻ nhất:

1. Tự động tăng điểm cho các đoạn có:
 - motion cao liên tục (rally)
 - audio peak
 - vị trí bóng & người trong vùng court
2. Chọn top N đoạn
3. Xuất highlight 1–2 phút + 3 video dọc

→ MVP đẹp — chưa cần deep learning phức tạp

→ Không tiêu GPU nặng

Khi scale tăng:

- Thay heuristic bằng model action recognition để tracking tình huống điểm thắng / pha khó / smash

1. Kiến trúc tổng thể hệ thống (ở mức “dev cầm làm được”)

Các khối chính:

- **Client (React)**
 - App quay video (tự quay)

- App quét QR, xem & quản lý highlight
 - **API Server** (Node/Nest/Express gì cũng được)
 - Auth, session, payment, quản lý sân, quản lý camera
 - Tạo/nhiệm vụ “highlight_job”
 - **Media & AI Service** (microservice riêng hoặc module trong backend)
 - Nhận video full
 - Dùng FFmpeg + AI/heuristic để cắt highlight
 - Ghép thành video tổng
 - **Storage**
 - S3 / GCS / Wasabi... lưu:
 - Video full trận
 - Các highlight clip
 - Final highlight
 - **Edge Agent tại sân** (rất quan trọng cho IP cam)
 - Chạy ở sân (mini PC/Raspberry Pi/box)
 - Đọc stream từ IP cam (RTSP/ONVIF)
 - Nhận lệnh từ server (start/stop)
 - Gửi stream hoặc file lên cloud
-

2. TÍNH NĂNG 1 – TỰ QUAY + BẮT HIGHLIGHT (VOICE / NÚT)

2.1. Luồng user (UX flow)


Màn hình 1: Chọn chế độ quay

- Nút: **Start Self Recording**
- Chọn:
 - Môn: Pickleball / ...
 - Thời lượng dự kiến (vd: 60 phút) – dùng để gợi ý, không bắt buộc chính xác

Nhấn “Bắt đầu” → chuyển sang màn hình camera.


Màn hình 2: Màn hình quay trận

Layout đề xuất:

- View full màn hình: preview camera
- Overlay trên video:
 - Góc trái trên: thời gian quay (00:12:35)
 - Góc phải trên: trạng thái mic (icon  on/off)
 - Góc dưới giữa:
 - Nút tròn to “**Lưu Highlight**”
 - Nút nhỏ “Kết thúc trận”

Hành vi:

- Người dùng có thể:
 - Bấm “**Lưu Highlight**” bất kỳ lúc nào
 - Hoặc nói từ khóa “highlight” (nếu bật Voice trigger)
- Khi trigger:

- Hiển thị toast:  Highlight saved (00:13:54) trong 1–2s
 - Khi nhấn “Kết thúc trận”:
 - Dừng quay
 - Hiển thị màn hình “Đang upload video & xử lý highlight...”
-

Màn hình 3: Danh sách highlight sau trận

Sau khi backend xử lý xong:

- Danh sách dạng card:
 - Thumbnail
 - Thời lượng: 42s
 - Label: “Highlight #1 – Rally dài bên trái sân” (anh có thể auto đặt tên hoặc để generic)
- Với mỗi highlight:
 - Xem
 - **Chỉnh sửa** (trim, thêm text, filter đơn giản)
 - **Thêm vào video tổng**

Cuối cùng có nút:

- **Tạo video highlight hoàn chỉnh** → backend ghép các clip được chọn
-

2.2. Cách xử lý trên client (React)

Quan trọng: không được cắt video khi đang quay.

Trên client anh làm:


```
state = {
  isRecording: false,
  recordingStartTime: Date.now(),
  highlightEvents: [ timestampInMs, ... ], // list các mốc highlight
};
```

- Khi bắt đầu quay:
 - Gọi `getUserMedia` (web) hoặc lib camera (nếu dùng React Native), bắt đầu quay 1 file duy nhất.
- Khi user:
 - Bấm nút “Lưu Highlight” → push `Date.now()` vào `highlightEvents`
 - Nói “highlight” → module nhận diện từ khóa, cũng push `timestamp` vào `highlightEvents`
- Khi kết thúc:
 - Dừng quay → có file `full_match.mp4`
 - Upload file này lên storage (qua backend hoặc trực tiếp S3 + lấy `video_url`)
 - Gửi request tới backend:

```
POST /api/sessions/self-record/finish
{
  "session_id": "abc123",
  "video_url": "https://s3.../full_match.mp4",
  "recording_start_time": "2025-11-29T10:00:00Z",
  "highlight_events": [
    "2025-11-29T10:12:34Z",
    "2025-11-29T10:21:05Z",
    ...
  ]
}
```

Backend lo phần còn lại.

2.3. Voice trigger (React)

Nếu là **web app**:

- Dùng: **Web Speech API** (SpeechRecognition)
 - Listen continuous
 - Khi detect “highlight” trong transcript → tạo highlight event.

Nếu là **React Native**:

- Dùng lib như `react-native-voice` hoặc tích hợp `Vosk/Whisper tiny` qua native
 - Hoặc phiên bản V1: **chỉ có nút “Lưu Highlight”**, voice để phase 2 – product vẫn dùng ngon.
-

2.4. Xử lý backend cho Self Record

Model dữ liệu Session

```
Session {
  id: string;
  user_id: string;
  type: "SELF_RECORD" | "COURT_CAMERA";
  start_time: Date;
  end_time: Date;
  video_url: string;          // video full trận
  status: "PROCESSING" | "DONE" | "FAILED";
}
```

Bảng HighlightEvents

```
HighlightEvent {
  id: string;
```

```
    session_id: string;
    event_time: Date;           // timestamp tuyệt đối
}
```

Bảng HighlightClip

```
HighlightClip {
    id: string;
    session_id: string;
    start_offset_sec: number; // giây tính từ đầu video full
    duration_sec: number;
    clip_url: string;
}
```

Job xử lý (worker)

Pseudo logic:

```
for each HighlightEvent in session:
    offset = event_time - session.start_time // giây
    clip_start = max(offset - 45, 0)
    duration = 45 or 60

    ffmpeg(
        input = session.video_url,
        ss = clip_start,
        t = duration,
        output = new_clip_url
    )

    save HighlightClip
```

Sau đó:

- Nếu user yêu cầu “video tổng”:
 - Lấy danh sách clip user chọn

- Tạo file `concat_list.txt` rồi dùng `ffmpeg concat filter`

```
ffmpeg -f concat -safe 0 -i concat_list.txt -c copy  
final_highlight.mp4
```

→ Rất rẻ, chạy được trên server CPU bình thường (EC2 tầm trung là đủ).

3. TÍNH NĂNG 2 – QUÉT QR TẠI SÂN + IP CAM + AI HIGHLIGHT

3.1. Luồng user (UX)

Step 1 – Quét QR

QR dán ở sân:

`my2light.com/court/{COURT_ID}`

User:

- Quét bằng camera → mở app/web my2light
- App gọi `GET /api/courts/{COURT_ID}` để:
 - Hiển thị tên sân, giá gói, thời lượng

Step 2 – Thanh toán

- User chọn:
 - Gói: 60/90 phút
 - Số người share chi phí (optional)

Nhấn “Thanh toán” → redirect qua Momo/VNPay/Stripe.

Khi thanh toán xong:

- Backend nhận webhook `PAYMENT_SUCCESS` cho `session_id`
- Backend:
 - Cập nhật session status = `PAID`
 - Gửi lệnh tới **Edge Agent** của sân: `START_RECORDING(session_id, duration)`

Step 3 – Camera sân tự quay

- Edge Agent:
 - Kết nối với IP cam (RTSP)
 - Hoặc:
 - Push RTMP lên server stream
 - Hoặc ghi file local rồi upload chunk/1 file sau khi kết thúc
- Hết thời lượng:
 - Dừng quay
 - Upload video full
 - Gửi callback `RECORDING_DONE(session_id, video_url)` về backend

Step 4 – Backend xử lý AI highlight

- Nhận `video_url` → tạo job AI highlight:
 - Heuristic: motion + audio
 - Hoặc action recognition khi về sau có data
- Tạo nhiều **HighlightClip** giống như self-record nhưng không cần `HighlightEvent` (AI tự chọn event)

- Khi xong:
 - Gửi push/email/in-app: “Highlight trận của bạn tại sân X đã sẵn sàng”
 - User vào app → xem & tải.
-

3.2. Kiến trúc kỹ thuật cho IP cam + Edge Agent

Do anh dùng **IP cam chuyên dụng**, đa số hỗ trợ:

- RTSP stream (vd: `rtsp://user:pass@ip:port/stream1`)
- ONVIF (để điều khiển)

Edge Agent (ở sân):

- Có thể dùng:
 - 1 con **mini PC (Intel NUC)** hoặc
 - **Raspberry Pi 4** nếu bitrate không quá cao
- Software:
 - Service Node/Python chạy:
 - WebSocket client kết nối backend:
`wss://api.my2light.com/agents/{AGENT_ID}`
 - Module FFmpeg/LibVLC để:
 - start record: `ffmpeg -i rtsp://... -c copy file.mp4`
 - hoặc push RTMP: `ffmpeg -i rtsp://... -c copy -f flv rtmp://stream.my2light.com/live/session_id`

Backend – Agent Protocol (giản lược)

- Khi payment xong:

WS message to agent:

```
{
  "type": "START_RECORDING",
  "session_id": "sess_123",
  "court_id": "court_abc",
  "duration_sec": 3600
}
```

- Agent trả lời:

```
{
  "type": "RECORDING_STARTED",
  "session_id": "sess_123",
  "started_at": "..."
}
```

- Sau khi quay xong & upload:

```
{
  "type": "RECORDING_DONE",
  "session_id": "sess_123",
  "video_url": "https://s3.../sess_123_full.mp4"
}
```

Backend nhận `video_url` → đẩy vào pipeline AI highlight giống self-record (nhưng event do AI xác định).

3.3. AI highlight cho chế độ sân (không có nút/voice)

MVP (chi phí thấp):

- Heuristic:

- Downscale video → 720p
- Tính frame difference / optical flow → chấm `motion_score`
- Tính audio peak → `audio_score`
- Đặt threshold để xác định “khoảng rally”
- Sau đó:
 - Gom các segment liên tiếp có score cao thành 1 highlight
 - Lấy top N segment theo “độ dài rally + audio peak”
 - Cắt thành clip bằng FFmpeg
- Output:
 - 1 video tổng (1–2 phút) + 3–5 video dọc

Phase sau (nâng chất lượng):

- Thu thập data từ người dùng:
 - Hỏi “Highlight này có hay không?” → feedback
- Train model action recognition riêng cho pickleball
- Dần dần thay heuristic bằng model → highlight chính xác hơn

4. Tóm lại cho 2 vấn đề anh đang kẹt

Tự quay + highlight (Voice/Nút)

- Giải UX:
 - Màn hình quay cực đơn giản: 1 nút lưu highlight + trạng thái mic.

- Không edit trong lúc quay, chỉ đặt bookmark.
- **Giải tech:**
 - Quay 1 file duy nhất → upload → backend cắt bằng timestamps + FFmpeg.
 - Voice trigger: Web Speech API / react-native-voice, hoặc phase 2 mới làm.

Quét QR tại sân + IP cam + AI

- **Giải flow:**
 - QR → Chọn gói → Thanh toán → Server dựng **session** → Gửi lệnh cho Edge Agent → Quay → Upload → AI xử lý → Gửi highlight cho user.
- **Giải tech:**
 - IP cam không nói chuyện trực tiếp với app, mà với **Edge Agent** tại sân.
 - Backend là “não”: quản lý session, payment, lệnh start/stop, xử lý AI.

I. Nguyên tắc chung (web app, mobile-first, tối ưu performance)

- **Mobile-first:**
 - Layout 1 cột, hạn chế scroll ngang.
 - Mọi nút hành động chính phải dễ bấm bằng ngón cái (đáy màn hình).
- **Không blocking UI:**
 - Tất cả xử lý nặng (cắt video, AI) chạy backend.
 - Front chỉ upload + gửi metadata → hiển thị trạng thái chờ.
- **Video:**
 - Quay **1 file duy nhất**, không cắt trên client.

- Upload dạng **chunk** (resumable) nếu trận dài.
 - **Trạng thái rõ ràng:**
 - Mỗi bước có state: `idle / recording / uploading / processing / ready / failed`.
 - Người dùng luôn biết “đang ở đâu trong quy trình”.
-

II. UI/UX FLOW – MOBILE WEB

A. Flow 1 – Tự quay + bắt highlight (voice / nút)

1. Màn hình Home / Dashboard

Mục tiêu: cho user chọn 2 chế độ:

- “Tự quay”
- “Quét QR tại sân”

Layout (mobile):

- Header: logo my2light, avatar nhỏ.
- Section chính:

[Card 1]

 Tự quay trận đấu của bạn

Mô tả: Dùng điện thoại quay, bấm/nói "highlight" để lưu pha hay

[Nút: Bắt đầu tự quay]

[Card 2]

 Quét QR tại sân

Mô tả: Quét mã tại sân có my2light, camera sân sẽ tự quay & làm highlight

[Nút: Quét QR ngay]

- Footer: tab nhỏ “Highlight của tôi”, “Tài khoản”.

2. Màn hình “Chuẩn bị quay”

Khi user bấm “Bắt đầu tự quay”:

- Step 1: Check permission:
 - Nếu chưa cho phép camera/mic → popup giải thích ngắn:

“my2light cần quyền sử dụng camera & micro để quay trận của bạn.”

- Step 2: Form ngắn:

Môn: [Pickleball ▼]


Thời lượng dự kiến: [60 phút ▼] (chỉ để hiển thị, không bắt buộc)

Voice trigger: [✓] Bật lệnh "highlight"

[Nút: Bắt đầu quay]

3. Màn hình quay (Self record)

Đây là màn hình quan trọng nhất. Đề xuất layout:

[Video Preview full màn]
(Overlay góc trái trên)
• 00:13:45
(Overlay góc phải trên)
 Voice: ON

[Toast nổi khi lưu highlight]

✓ Highlight saved (00:12:34)

(Thanh đáy màn hình)

[■ Kết thúc] [★ Lưu highlight]

Hành vi:

- ★ Lưu highlight:
 - Gửi event local: `highlightEvents.push(Date.now())`
 - Hiện toast 1–2s.
- Voice trigger (nếu bật):
 - UI nhỏ ở góc trên: "Listening 'highlight'..."
 - Khi detect → xử lý như Lưu highlight.
- ■ Kết thúc:
 - Dừng quay.
 - Chuyển sang màn hình “Xử lý video”.

4. Màn hình “Đang upload & xử lý”

Sau khi user bấm Kết thúc:

Tiêu đề: Đang xử lý trận của bạn...

[Stepper 3 bước]

1. ↑ Upload video trận đấu [xong / đang chạy]
2. ⚙ Tạo các highlight từ mốc bạn đánh dấu
3. 🎬 Chuẩn bị video highlight

[Thanh tiến trình] [-----] 65%

[Nút phụ: Về trang chủ]

(Nhấn người dùng: Khi xong sẽ có thông báo & email)

- Không bắt user chờ mở app → cho phép rời màn hình, thông báo sau.

5. Màn hình danh sách highlight (self-record)

Khi xử lý xong:

- User được dẫn tới:

Tiêu đề: Highlight trận ngày 29/11

[Card Highlight 1]

Thumbnail | Highlight #1 – 42s

| Lúc 00:12:34 trong trận

[Xem] [Chỉnh sửa] [Thêm vào video tổng]

[Card Highlight 2]

...

[Nút lớn đáy màn hình]

[ Tạo video highlight hoàn chỉnh]

6. Màn hình chỉnh sửa & tạo video tổng

- Khi bấm **Chỉnh sửa**:
 - Cho phép trim start/end
 - Thêm caption đơn giản, speed, filter nhẹ.

- Khi bấm **Tạo video highlight** hoàn chỉnh:
 - Popup:
 - Chọn thứ tự highlight
 - Chọn intro/outro (template)
 - Xác nhận → gửi request backend build video final → màn hình trạng thái + notify khi xong.
-

B. Flow 2 – Quét QR tại sân (IP cam + AI highlight)

1. Màn “Quét QR”

Trên mobile web, anh có 2 cách:

- Cách 1: QR mở **URL** dạng
https://my2light.com/court/{COURT_ID}?ref=qrcode
→ trình duyệt mở thẳng tới trang sân → không cần scan trong app.
- Cách 2: App có màn hình camera scan QR (nếu anh làm PWA riêng).

Đơn giản nhất: dùng Cách 1.

Khi QR mở URL:

2. Màn hình “Thông tin sân & gói”



Sân Pickleball ABC - Quận 7

Thời lượng gợi ý: [60 phút ▼]

Giá gói quay & highlight:

- 60 phút: 180.000đ
- 90 phút: 240.000đ

[Chọn 60p] [Chọn 90p]

Thanh toán:

(Tích chọn) Chia đều cho 4 người? (chỉ để gợi ý, không bắt buộc backend)

[Nút lớn: Thanh toán & bắt đầu quay]

3. Màn hình thanh toán

- Redirect hoặc embed: VNPay / Momo / Stripe.
- Sau khi thanh toán xong:
 - Hiện màn hình:

✓ Thanh toán thành công!

Camera tại sân đang được bật và ghi hình trận của bạn.

Trận của bạn:

- Sân: ABC - Court #2
- Thời lượng: 60 phút
- Bắt đầu: 19:30

[Nút: Quay về trang chủ]

[Link: Xem trạng thái highlight]

Backend đồng thời gửi lệnh tới Agent.

4. Màn hình “Trận đang được ghi hình”

Trong tab “Highlight của tôi”, user thấy:

Đang ghi hình...

Trận ngày 29/11 - Sân ABC

Trạng thái: ● Đang quay (19:30 – 20:30)

Sau khi kết thúc, hệ thống sẽ tự tạo highlight và gửi cho bạn.

[[Nút: Xem chi tiết](#)]

5. Màn “Highlight trận sân” (sau khi AI xử lý xong)

Gần giống self-record nhưng chú thích:

Trận sân ABC - 29/11

[Card Highlight 1]

AI chọn – Rally dài 28s

[[Xem](#)] [[Bỏ chọn khỏi video tổng](#)] [[Tải xuống](#)]

[Card Highlight 2]

AI chọn – Pha kết thúc trận 18s

[ [Tạo video highlight hoàn chỉnh](#)]

User có thể bỏ bớt highlight AI đề xuất trước khi build video tổng.

III. API SPEC – CHO DEV BACKEND

Giả sử backend là REST JSON.

1. Auth (rút gọn)

POST /api/auth/login

- Body:

```
{ "provider": "google", "id_token": "..." }
```


- Response:

```
{
  "access_token": "jwt...",
  "user": {
    "id": "u_123",
    "name": "Tom",
    "email": "tom@example.com"
  }
}
```

(Chi tiết auth anh tùy chọn, không phải trọng tâm feature.)

2. Model dữ liệu chính

User

```
User {
  id: string;
  name: string;
  email: string;
  avatar_url?: string;
  created_at: Date;
}
```

Court

```
Court {
  id: string;
  name: string;
  location: string;
  sport_type: "PICKLEBALL" | "TENNIS" | ...;
  agent_id: string;           // agent tại sân
  pricing: {
    "60": number;             // giá 60 phút
    "90": number;             // giá 90 phút
  }
}
```

```
};  
  active: boolean;  
}
```

Agent (Edge Agent tại sân)

```
Agent {  
  id: string;  
  court_id: string;  
  status: "ONLINE" | "OFFLINE";  
  last_seen_at: Date;  
  rtsp_url: string;          // stream IP cam  
}
```

Session

```
Session {  
  id: string;  
  user_id: string;  
  court_id?: string;          // null nếu self-record  
  type: "SELF_RECORD" | "COURT_CAMERA";  
  start_time: Date;  
  end_time?: Date;  
  duration_sec?: number;  
  video_url?: string;         // full video  
  status: "CREATED" | "RECORDING" | "UPLOADING" | "PROCESSING" |  
  "READY" | "FAILED";  
  source: "MOBILE" | "AGENT";  
}
```

HighlightEvent (chỉ self-record)

```
HighlightEvent {  
  id: string;  
  session_id: string;  
  event_time: Date;           // thời điểm thực tế  
  created_at: Date;  
}
```

HighlightClip (cả 2 loại)

```
HighlightClip {
  id: string;
  session_id: string;
  start_offset_sec: number;    // giây từ đầu video
  duration_sec: number;
  clip_url: string;
  selected_for_compilation: boolean;
  origin: "USER_EVENT" | "AI_DETECTED";
}
```

3. Self-record API

3.1. Tạo session self-record

POST /api/sessions/self-record/start

- Auth: user
- Body:

```
{
  "sport_type": "PICKLEBALL",
  "expected_duration_sec": 3600,
  "voice_trigger_enabled": true
}
```

- Response:

```
{
  "session_id": "sess_self_123",
  "upload_url": "https://upload.my2light.com/... (optional, nếu dùng pre-signed URL)"
}
```

3.2. Gửi highlight events (có thể batch hoặc realtime)

Option 1 – realtime (simple):

POST /api/sessions/self-record/:session_id/events

```
{
  "events": [
    { "event_time": "2025-11-29T10:12:34Z" },
    { "event_time": "2025-11-29T10:21:05Z" }
  ]
}
```

Option 2:

Lưu local, gửi 1 lần khi kết thúc – backend để xử lý hơn.

3.3. Kết thúc session self-record

POST /api/sessions/self-record/:session_id/finish

- Body:

```
{
  "video_url": "https://s3.../full_match.mp4",
  "recording_start_time": "2025-11-29T10:00:00Z",
  "highlight_events": [
    "2025-11-29T10:12:34Z",
    "2025-11-29T10:21:05Z"
  ]
}
```

- Response:

```
{
  "session_id": "sess_self_123",
  "status": "PROCESSING"
}
```

Backend sau đó tạo job:

- Tính `offset_sec` = `event_time` – `start_time`
 - Dùng FFmpeg cắt clip.
 - Tạo các `HighlightClip`.
-

3.4. Lấy danh sách highlight clip của session

GET `/api/sessions/:session_id/highlights`

- Response:

```
{
  "session_id": "sess_self_123",
  "status": "READY",
  "clips": [
    {
      "id": "clip_1",
      "start_offset_sec": 540,
      "duration_sec": 45,
      "clip_url": "https://...",
      "origin": "USER_EVENT",
      "selected_for_compilation": true
    },
    ...
  ]
}
```

3.5. Cập nhật lựa chọn clip cho video tổng

PATCH `/api/sessions/:session_id/highlights/:clip_id`

```
{  
  "selected_for_compilation": true  
}
```

3.6. Tạo video highlight tổng hợp

POST /api/sessions/:session_id/compilation

- Body (optional):
 - Thứ tự clip.

```
{  
  "clip_ids": ["clip_3", "clip_1", "clip_2"]  
}
```

- Response:

```
{  
  "session_id": "sess_self_123",  
  "compilation_id": "comp_789",  
  "status": "PROCESSING"  
}
```

Sau khi xong:

GET /api/sessions/:session_id/compilation/:compilation_id

```
{  
  "status": "READY",  
  "video_url": "https://.../final_highlight.mp4"  
}
```

4. Court + QR + Payment + Agent API

4.1. Lấy thông tin sân từ QR URL

GET /api/courts/:court_id

- Response:

```
{
  "id": "court_abc",
  "name": "Sân Pickleball ABC",
  "location": "Quận 7, TP.HCM",
  "sport_type": "PICKLEBALL",
  "pricing": {
    "60": 180000,
    "90": 240000
  },
  "agent_status": "ONLINE"
}
```

4.2. Tạo session court-camera trước khi thanh toán

POST /api/sessions/court-camera/start

```
{
  "court_id": "court_abc",
  "duration_sec": 3600
}
```

- Response:

```
{
  "session_id": "sess_court_123",
  "payment_url": "https://pay.my2light.com/..."
}
```

Frontend redirect user tới `payment_url`.

4.3. Webhook thanh toán thành công (từ Momo/VNPay...)

POST /api/payments/webhook

- Body (tùy cổng):

```
{
  "gateway": "momo",
  "transaction_id": "txn_123",
  "status": "SUCCESS",
  "session_id": "sess_court_123",
  "amount": 180000
}
```

Backend:

- Xác nhận thanh toán.
- Set session status **PAID**.
- Gửi lệnh cho Agent:

```
{
  "type": "START_RECORDING",
  "session_id": "sess_court_123",
  "duration_sec": 3600
}
```

4.4. WebSocket protocol giữa Backend & Agent

Kết nối:

- Agent connect tới: <wss://api.my2light.com/agents>
- Gửi message **AUTH**:


```
{
  "type": "AUTH",
  "agent_id": "agent_abc",
  "secret": "... " // token
}
```

Backend reply **AUTH_OK**, sau đó giữ kết nối.

Command từ Backend tới Agent:

```
{
  "type": "START_RECORDING",
  "session_id": "sess_court_123",
  "rtsp_url": "rtsp://user:pass@ip/stream1",
  "duration_sec": 3600
}
```

Agent trả lời:

- Khi bắt đầu:

```
{
  "type": "RECORDING_STARTED",
  "session_id": "sess_court_123",
  "started_at": "2025-11-29T12:00:00Z"
}
```

- Khi kết thúc & upload xong:

```
{
  "type": "RECORDING_DONE",
  "session_id": "sess_court_123",
  "video_url": "https://s3.../sess_court_123_full.mp4",
  "duration_sec": 3590
}
```

- Nếu lỗi:

```
{
  "type": "RECORDING_FAILED",
  "session_id": "sess_court_123",
  "error": "RTSP_CONNECTION_ERROR"
}
```

Backend nhận `video_url` → tạo job AI highlight tương tự self-record (nhưng tự detect event).

4.5. Lấy danh sách session của user

GET `/api/sessions?user_id=me`

- Response:

```
{
  "sessions": [
    {
      "id": "sess_self_123",
      "type": "SELF_RECORD",
      "status": "READY",
      "created_at": "...",
      "source": "MOBILE"
    },
    {
      "id": "sess_court_123",
      "type": "COURT_CAMERA",
      "status": "PROCESSING",
      "created_at": "...",
      "court": {
        "id": "court_abc",
        "name": "Sân ABC"
      }
    }
  ]
}
```

}

IV. Một vài lưu ý performance & ổn định

- **Self-record:**
 - Giới hạn quality quay: 720p @ 30fps là đủ cho mobile + social.
 - Dùng MediaRecorder với `timeslice` (chunk upload) nếu trận quá dài.
 - Nếu mạng yếu: cho phép “tải lên sau” khi có wifi.
- **Court-camera:**
 - Agent nên ghi file local trước, rồi upload sau, tránh phụ thuộc vào mạng realtime.
 - Nếu dùng RTMP → backend có thể vừa record vừa chuẩn bị cho livestream (phase 2).
- **AI/Highlight:**
 - MVP: heuristic + FFmpeg, không cần GPU.
 - Tối ưu disk I/O: xử lý từng file, xong là xoá tạm.