

Core API Endpoints and Data Schemas for AI Site Visit System

✓ Core API Endpoints

POST /visits

Description: Create a new site visit entry.

Payload / Response Example:

```
{
  "project_id": "alpha123",
  "engineer_id": "eng456",
  "client_id": "client789",
  "date": "2025-03-26",
  "location": "XYZ Site",
  "notes": "Initial observations on-site."
}
```

Response: 201 Created

POST /upload/audio

Description: Upload raw site audio to S3 and trigger transcription pipeline.

Payload / Response Example:

- visit_id
- audio_file (Multipart)

Response: 202 Accepted

POST /upload/floorplan

Description: Upload floor plan file and associate with visit.

Payload / Response Example:

- visit_id
- floor_plan_file (Multipart)

Response: 200 OK

GET /visits/{visit_id}/report

Description: Get full report including AI summary, floor plan overlay, metadata.

Payload / Response Example:

```
{
  "visit_id": "visit123",
  "project": "Project Alpha",
  "date": "2025-03-26",
  "engineer": "Engineer A",
  "client": "Client C",
  "location": "XYZ Site",
  "audio_transcription": "...",
  "ai_summary": [
    "Issue in Living Room (Wall crack)",
    "Paint misalignment in Hallway"
  ],
  "action_items": [
    {
      "room": "Kitchen",
      "issue": "Plumbing issue",
      "assigned_to": "Vendor X"
    }
  ]
}
```

Response: 200 OK

PATCH /visits/{visit_id}/actions

Description: Assign or update action items.

Payload / Response Example:

```
{
  "actions": [
    {
      "room": "Hallway",
      "issue": "Paint mismatch",
      "assigned_to": "Vendor Y"
    }
  ]
}
```

Response: 200 OK

GET /projects

Description: Fetch all available projects and related site visits.

Payload / Response Example:

```
[
  {
    "project_id": "alpha123",
    "name": "Project Alpha",
    "visit_count": 2
  }
]
```

Response: 200 OK

GET /status/{visit_id}

Description: Check transcription and processing status.

Payload / Response Example:

```
{
  "status": "Processing | Completed | Error",
  "message": "Whisper timeout on audio file"
}
```

Response: 200 OK

📖 Data Schemas (PostgreSQL)

Table: projects

Column	Type	Description
id	UUID PK	Project unique ID
name	TEXT	Project name
created_at	TIMESTAMP	Auto-generated

Table: visits

Column	Type	Description
id	UUID PK	Visit ID
project_id	UUID FK	Linked to projects.id
engineer_id	UUID FK	Linked to users.id
client_id	UUID FK	Linked to clients.id
location	TEXT	Site location
date	DATE	Date of visit
status	TEXT	Processing, Completed...
notes	TEXT	Optional engineer notes

Table: audio_files

Column	Type	Description
id	UUID PK	Audio file ID
visit_id	UUID FK	Linked to visits.id
s3_url	TEXT	S3 storage link
uploaded_at	TIMESTAMP	Upload time

Table: floor_plans

Column	Type	Description
id	UUID PK	Floor plan ID
visit_id	UUID FK	Linked to visits.id
file_url	TEXT	S3 or storage location

Table: ai_insights

Column	Type	Description
id	UUID PK	AI Insight ID

visit_id	UUID FK	Linked to visits.id
summary_json	JSONB	Parsed AI output
generated_at	TIMESTAMP	When insight created

Table: action_items

Column	Type	Description
id	UUID PK	Action ID
visit_id	UUID FK	Linked to visits.id
room	TEXT	Room/zone in project
issue	TEXT	Problem description
assigned_to	TEXT	Vendor name or staff ID
status	TEXT	Pending, Done, Escalated
