

# AmplifiEd — Architecture, Data Model, Routing, and API (Final v1.0)

**Purpose of this document** - Single source of truth for our MVP through Demo Day. - Align the team (and PM bot) on architecture, data model, routing, APIs, and workflows. - Includes field-by-field table descriptions and sample data previews.

---

## 1) Product scope (MVP)

**Users:** - Tutors (independent, subject-specialized: Physics, Chemistry) - Students (20–30 per course/year) - Parents (read-only progress overview) - Admin/Dev (internal ops)

**Core value:** Tutors keep their videos in Google Drive/Dropbox. AmplifiEd consumes the link + metadata, runs local GPU transcription, generates study aids, enables tutor review/approval, delivers student learning tools (notes, flashcards, quizzes, chatbot), and tracks progress.

**Must-have features for Demo Day:** 1. Tutor uploads session link + metadata. 2. Local RTX worker transcribes, chunks, embeds. 3. AI generates summary + flashcards (draft), Tutor reviews & publishes. 4. RAG chatbot answers questions based on transcript + docs. 5. Tutor quiz builder (uses tutor bank + AI); students attempt; topic mastery updates. 6. Student & Parent views for progress.

---

## 2) Final Architecture (locked)

### 2.1 Overview

- **Frontend (Vercel):** Next.js (App Router), TailwindCSS, shadcn/ui.
- **Backend services:**
- **Supabase:** Postgres + Auth + Storage + pgvector.
- **FastAPI** microservice(s) for AI orchestration (summaries/flashcards/quizzes/embeddings) and REST APIs.
- **Worker:** Local RTX 3060 node (Whisper transcription) + polling via DB ( `processing_jobs` ).
- **LLM stack:** Whisper (local) for ASR; GPT-4o/Gemini for text; OpenAI Embeddings (pgvector) for retrieval.
- **Storage:** Tutors retain raw video in Drive/Dropbox; AmplifiEd stores only derived artifacts (transcripts, chunks, materials, quizzes, attempts, analytics) and optionally uploaded docs.

### 2.2 Data Flow (text diagram)

```
Tutor portal (add session URL + metadata)
└─> DB.sessions (status=draft) + processing_jobs(job='transcription')
    └─ Local RTX Worker polls → downloads video → Whisper → transcripts
```

```

    |> transcript_chunks (+ embeddings)
    |> enqueue jobs: summarization / flashcards / doc_ingest /
embedding
    |> job_events telemetry

Study materials (draft) —> Tutor Review (approve/reject) —> published
    |
    |_____ visible in Student portal (notes/flashcards/chatbot/
quizzes)

Quizzes (from tutor bank + AI) —> attempts —> topic_mastery (student progress)
Parent view reads progress only.

```

## 2.3 Environments & deployment

- **Vercel:** Next.js frontend (preview deployments from PRs).
- **Supabase Project:** Postgres, Auth, Storage; pgvector enabled.
- **Render/Fly.io:** FastAPI service(s).
- **Local RTX Worker:** Python process with a `.env` that has Supabase creds and a `WORKER_ID` for telemetry.

## 2.4 Observability

- **processing\_jobs / job\_events:** operational telemetry.
- **retrieval\_hits / message\_feedback:** RAG quality signals.
- **audit\_log:** administrative trace for key actions (publish, approve, etc.).

# 3) Frontend routing (App Router)

## Public

- `/` — Landing (marketing); link to login.
- `/login` — Auth (Supabase).

## Tutor

- `/tutor/home` — Course list; quick actions: *Create Course*, *Add Session by Link*.
- `/tutor/courses/:courseId` — Course overview, enrolled students, quick analytics.
- `/tutor/sessions/:sessionId` — Tabs: **Transcript**, **Study Materials**, **Review & Publish**, **Files**.
- `/tutor/question-bank` — Manage tutor banks, upload/import CSV, tag to topics.
- `/tutor/quiz-builder` — Choose topics/difficulty; assemble quiz (tutor + AI); save as template.
- `/tutor/performance` — Student/topic analytics across courses.

## Student

- `/student/home` — Enrolled courses & latest materials.

- `/student/courses/:courseId` — Course hub; sessions list.
- `/student/sessions/:sessionId` — Recap, Notes, Flashcards, **Chatbot**, Quizzes.
- `/student/performance` — Topic mastery summary.

## Parent

- `/parent/students/:studentId/performance` — Read-only progress dashboard.

## Admin/Ops (internal)

- `/admin/queue` — View jobs, statuses, errors; retry/cancel.
- `/admin/rag` — Index health (chunks, last embed), sampling.

## 4) API (FastAPI) — minimal but complete surface

**Ingestion & pipeline** - `POST /ingest/session` — Create session from URL + metadata (creates job: transcription). - `POST /jobs/queue` — Queue specific job (embedding, summarization, etc.). - `POST /jobs/{id}/status` — (Optional webhook) worker status update.

**Transcripts & materials** - `GET /sessions/{id}/transcript` — Transcript + chunks. - `POST /materials/{sessionId}/{type}` — Generate material: `summary|flashcards|cheatsheet`. - `POST /materials/{id}/submit-review` — Move to `pending_review`. - `POST /materials/{id}/approve` — Approve & publish. - `POST /materials/{id}/reject` — Reject with comment.

**Question banks & quizzes** - `POST /questions/import` — CSV/JSON bank import (returns counts, errors). - `POST /quizzes/build` — Input: topics + difficulty + #Q → returns template id. - `POST /quizzes/{id}/attempt` — Start attempt. - `POST /attempts/{id}/submit` — Submit; auto-grade MCQ; score.

**Chat** - `POST /chat/ask` — Input: sessionId + message; returns answer + citations; logs retrieval\_hits. - `POST /chat/feedback` — Rate answer (1-5) + optional comment.

## 5) Database model (tables, fields, relationships) with samples

We use Supabase Auth ( `auth.users` ) for accounts. Our app profile lives in `profiles`.

### 5.1 Accounts & relationships

**Table: profiles** (one row per user) - `user_id` (uuid, PK) — Supabase auth id. - `role` (enum) — tutor | student | parent | admin. - `full_name` (text) - `email` (text) - `timezone` (text, default Asia/Kolkata) - `created_at` (timestampz)

*Sample rows*

user_id	role	full_name
email	timezone	created_at
6c0b...a1	tutor	Rahul Sharma
rahul@tutors.in	Asia/Kolkata	2025-09-17 10:00
b91f...c3	student	Aditi Mehta
aditi@studentmail.com	Asia/Kolkata	2025-09-18 09:30
7ab2...e9	parent	Mehta (Parent)
parent.mehta@mail.com	Asia/Kolkata	2025-09-18 09:32

**Table: guardianships** (parent ↔ student link) - `parent_id` (uuid → profiles.user\_id) - `student_id` (uuid → profiles.user\_id) - `created_at`

*Sample*

parent_id	student_id	created_at
7ab2...e9	b91f...c3	2025-09-18 09:33

## 5.2 Curriculum

**Table: subjects** - `id`, `name` (e.g., Physics), `board` (CBSE), `grade` (10)

**Table: topics** - `id`, `subject_id`, `name` (e.g., Newton's Laws), `parent_topic_id` (nullable)

*Sample (subjects & topics)*

subjects			
id	name	board	grade
sub-phys	Physics	CBSE	10

  

topics			
id	subject_id	name	parent_topic_id
t-mech	sub-phys	Mechanics	null
t-newton	sub-phys	Newton's Laws	t-mech

### 5.3 Courses, sessions, files

**Table: courses** (one per subject per academic year) - `id`, `tutor_id`, `subject_id`, `title`, `description`, `board`, `grade` - `academic_year` (text, e.g., 2025-2026) - `is_active` (bool) - `created_at`

*Sample*

id	tutor_id	subject_id	title	academic_year
is_active	created_at			
co-001	6c0b...a1	sub-phys	Class 10 Physics (Full Year)	2025-2026
true	2025-09-18 09:40			

**Table: enrollments** (students per course) - `id`, `course_id`, `student_id`, `enrolled_on`, `left_on`

*Sample*

id	course_id	student_id	enrolled_on	left_on
enr-1001	co-001	b91f...c3	2025-06-15	null

**Table: sessions** (recorded classes) - `id`, `course_id`, `title`, `session_date`, `language`, `duration_seconds` - `video_source_url` (Drive/Dropbox share link) - `status` (draft | pending\_review | approved | rejected | published) - `created_at`

*Sample*

id	course_id	title	session_date	language
duration_seconds	video_source_url		status	created_at
ses-001	co-001	Kinematics - Part 1	2025-07-01	en
5400		https://drive.google.com/....vid1	draft	2025-09-18
09:45				

**Table: session\_files** (slides/PDFs) - `id`, `session_id`, `file_type`, `source`, `storage_path`, `external_url`, `created_at`

*Sample*

id	session_id	file_type	source	storage_path	external_url	created_at
file-01	ses-001	pdf	uploaded_doc	sessions/ses-001/notes.pdf		2025-09-18 10:00

## 5.4 Pipeline (jobs & events)

**Table: processing\_jobs** - id, job (transcription|summarization|flashcards|quiz\_gen|embedding|doc\_ingest), session\_id, file\_id, payload - status (queued|running|failed|completed|canceled), assigned\_to, error - queued\_at, started\_at, finished\_at

Sample

id	job	session_id	status	assigned_to	queued_at
job-01	transcription	ses-001	queued	RTX3060-01	2025-09-18 10:01

**Table: job\_events** - id, job\_id, status, message, created\_at

Sample

id	job_id	status	message	created_at
1	job-01	running	Downloaded video (1.2 GB)	2025-09-18 10:05
2	job-01	running	Whisper chunk 1/120 complete	2025-09-18 10:08
3	job-01	completed	Transcription finished	2025-09-18 10:25

## 5.5 Transcripts & chunks (RAG)

**Table: transcripts** - id, session\_id, language, confidence, created\_at

**Table: transcript\_chunks** - id, transcript\_id, start\_ms, end\_ms, text, embedding, created\_at

Sample (chunks)

id	transcript_id	start_ms	end_ms
text			...

chunk-001	tr-001	120000	138000	"Velocity is the rate of change..."
-----------	--------	--------	--------	-------------------------------------

**Table: doc\_chunks** (OCR'd PDFs/slides) - `id`, `file_id`, `page_no`, `text`, `embedding`, `created_at`

## 5.6 Study materials & review

**Table: study\_materials** - `id`, `session_id`, `type` (summary|glossary|flashcards|cheatsheet|exam\_prep), `source` (ai|tutor|uploaded\_doc|transcript) - `content` (jsonb), `status` (draft|pending\_review|approved|rejected|published) - `created_by`, `created_at`, `updated_at`

*Sample*

id	session_id	type	source	status	content
mat-001	ses-001	summary	ai	pending_review	{"bullets":["Key idea 1","Key idea 2"],"length":250}
mat-002	ses-001	flashcards	ai	draft	{"cards":[{"q":"Define velocity","a":"..."}]}

**Table: material\_versions** (history) - `id`, `material_id`, `version_no`, `content`, `created_by`, `created_at`

**Table: reviews** (tutor approval) - `id`, `material_id`, `reviewer_id`, `status`, `comment`, `created_at`

## 5.7 Question banks & quizzes

**Table: question\_banks** - `id`, `owner_id`, `name`, `created_at`

**Table: questions** - `id`, `bank_id`, `subject_id`, `type` (mcq\_single|mcq\_multi|short\_answer|true\_false), `text`, `explanation`, `difficulty`, `source`, `approved`, `created_by`, `created_at`

**Table: choices** (MCQ options) - `id`, `question_id`, `label`, `text`, `is_correct`

**Table: question\_topics** - `question_id`, `topic_id`

*Sample (question + choices)*

```

questions
id          bank_id  subject_id  type          text
difficulty  source  approved
-----
q-101      bank-1  sub-phys   mcq_single    "Unit of acceleration is?"
easy       tutor   true

choices
id          question_id  label  text          is_correct
-----
ch-1       q-101      A      m/s            false
ch-2       q-101      B      m/s^2          true
ch-3       q-101      C      N              false

```

**Table: quiz\_templates** (reusable quizzes) - `id`, `session_id` (optional), `course_id` (optional), `title`, `description`, `status`, `created_by`, `created_at`

**Table: quiz\_questions** (composition) - `quiz_id`, `question_id`, `points`, `position`

**Table: quiz\_attempts** (per student) - `id`, `quiz_id`, `student_id`, `started_at`, `submitted_at`, `score`

**Table: attempt\_answers** (answers inside an attempt) - `id`, `attempt_id`, `question_id`, `answer` (json), `is_correct`, `graded_by`, `graded_at`

**Table: topic\_mastery** (aggregates) - `student_id`, `topic_id`, `attempts`, `correct`, `last_attempt`

*Sample (attempt + mastery)*

```

quiz_attempts
id          quiz_id  student_id  started_at          submitted_at
score
-----
at-2001     qt-5001  b91f...c3  2025-09-20 16:00:00  2025-09-20 16:12:00  8

topic_mastery
student_id  topic_id  attempts  correct  last_attempt
-----
b91f...c3  t-newton  5         4        2025-09-20 16:12:00

```



## 5.8 Chatbot telemetry & feedback

**Table: conversations** — `id`, `user_id`, `session_id`, `created_at`

**Table: messages** — `id`, `conversation_id`, `sender` (user|assistant), `text`, `created_at`

**Table: retrieval\_hits** — `id`, `message_id`, `chunk_id`, `chunk_type` (transcript|doc), `score`, `created_at`

**Table: message\_feedback** — `id`, `message_id`, `user_id`, `rating` (1-5), `comment`, `created_at`

---

## 5.9 Notifications & audit

**Table: notification\_preferences** — `user_id`, `email`, `sms`, `in_app`

**Table: notifications** — `id`, `user_id`, `type`, `payload`, `read_at`, `created_at`

**Table: audit\_log** — `id`, `actor_id`, `action`, `entity_table`, `entity_id`, `meta`, `created_at`

---

## 6) Relationships (summary)

- **Profile (tutor)** 1—N **Courses**; **Course** 1—N **Sessions**.
  - **Profile (student)** N—N **Courses** via **Enrollments**.
  - **Session** 1—N **Transcripts**, 1—N **Session Files**.
  - **Transcript** 1—N **Transcript Chunks**; **Session File** 1—N **Doc Chunks**.
  - **Session** 1—N **Study Materials**; **Study Materials** 1—N **Material Versions & Reviews**.
  - **Question Bank** 1—N **Questions**; **Questions** 1—N **Choices**; **Questions** N—N **Topics** via **Question Topics**.
  - **Quiz Template** N—N **Questions** via **Quiz Questions**; **Quiz Template** 1—N **Quiz Attempts**; **Quiz Attempt** 1—N **Attempt Answers**.
  - **Student** 1—N **Topic Mastery** (per topic).
  - **Parent** ↔ **Student** via **Guardianships** (read-only access to performance).
- 

## 7) Security (RLS) principles for MVP

- **Tutors**: read/write only their courses, sessions, materials, banks.
  - **Students**: read-only to courses they're enrolled in; write their own quiz attempts; read their own mastery.
  - **Parents**: read-only to their child's mastery via guardianships.
  - **Admin**: full read (and limited write where needed).
-

## 8) Operational notes

- **Worker retry policy:** failed jobs can be retried from `/admin/queue`.
- **Idempotency:** job payloads include session/file ids; workers should upsert, not duplicate.
- **Versioning:** every published material keeps versions; rollbacks are possible.
- **RAG quality:** monitor retrieval\_hits and low feedback to improve chunking/embeddings.

## 9) Demo Day script (golden path)

1) Tutor adds a new session with a Drive link → job queued. 2) Admin shows `/admin/queue` as the RTX worker processes transcription. 3) Session's Transcript tab shows chunks + timestamps (jump-to-time). 4) Generate Summary + Flashcards → Tutor edits → Approve & Publish. 5) Student opens session page → reads recap, uses flashcards, chats with the bot (citations shown). 6) Student takes a short quiz → score updates → Topic Mastery shows strengths/weaknesses. 7) Parent view displays the progress snapshot.

## 10) Appendix

### 10.1 ASCII ERD (overview)

```
profiles (user) —1..N— courses —1..N— sessions —1..N— transcripts —1..N—
transcript_chunks
                                |
                                L1..N— session_files —1..N—
doc_chunks

sessions —1..N— study_materials —1..N— material_versions
study_materials —1..N— reviews

profiles(student) —N..N— courses via enrollments

question_banks —1..N— questions —1..N— choices
questions —N..N— topics via question_topics

quiz_templates —N..N— questions via quiz_questions
quiz_templates —1..N— quiz_attempts —1..N— attempt_answers
profiles(student) —1..N— topic_mastery

guardianships: parent ↔ student

conversations —1..N— messages —1..N— retrieval_hits
messages —1..N— message_feedback

notification_preferences (1:1 profile) & notifications (1:N)
```

## 10.2 Ready-to-run SQL DDL

A full Supabase/Postgres schema (including enums, tables, indexes, vector ops, minimal RLS samples) is attached in the project repo. Ask an admin for the latest `amplified_schema.sql` (or import the copy distributed with this doc).

## 10.3 Sample CSV headers

- **Question import:**

`bank_name,subject,topic,type,text,choice_A,choice_B,choice_C,choice_D,correct_label,explanation`

- **Enrollments import:** `course_title,student_email,student_name,enrolled_on`

---

**This document is the locked v1.0 reference.** Any material deviations should be proposed as a change request and reflected here so the PM bot and the team stay aligned.