Stotradhwani — Complete Project Overview

> **Who this document is for:** The Program Manager joining the team, and the client (teacher) who commissioned the website. No coding knowledge is assumed. Technical terms are explained in plain English wherever they appear.

1. What Is Stotradhwani?

Stotradhwani is a private online learning hub built exclusively for a Sanskrit/Sloka teacher and their students. Think of it as a **private YouTube + Google Classroom, but owned entirely by the school** — no distractions, no algorithm, no irrelevant content. Everything the teacher posts, only enrolled students see.

The problem it solves: class recordings were scattered across YouTube (hard to find), study materials were on WhatsApp (messy), and there was no single place for announcements. Stotradhwani puts all three — recordings, materials, and announcements — in one organised, mobile-friendly website.

URL (planned): `stotradhwani.com`

2. Who Uses It?

There are two types of users, each with their own separate area of the website:

Role	What they see	Access point
Student	Home, courses, class videos, PDFs, announcements, profile	`stotradhwani.com/home`
Admin (Teacher)	Dashboard, manage courses/classes, invite students, write announcements	`stotradhwani.com/admin`

Key design constraint: The student base skews older (55+). Every button is extra large (minimum 44px tap target), the font can be made bigger with an A+ toggle, and navigation is as simple as possible — bottom bar on mobile, like WhatsApp.

3. What Can Students Do? (The Student Experience)

Once a student logs in, they land on their personal home screen. The five sections of the website are:

1. **Home** — Welcome screen, latest updates at a glance
2. **Courses** — Browse all courses the student is enrolled in (e.g. "Vishnu Sahasranamam", "Bhagavad Gita Chapter 1")
3. **Materials** — Download PDFs or audio files uploaded by the teacher

4. **Announcements** — Notices from the teacher ("Class postponed this Sunday", "New batch starting next month")
5. **Profile** — Student's own account details, font-size preference saved here

When a student opens a course, they see a list of recorded classes in order. Each class has an embedded YouTube video (it plays inside the website — the student never has to go to YouTube separately) and an optional PDF handout.

4. What Can the Teacher Do? (The Admin Experience)

The teacher logs into `/admin` and gets a control panel with five management sections:

1. **Dashboard** — A quick overview (number of students, recent activity)
2. **Courses** — Create and manage course series; set them as Draft or Published
3. **Classes** — Add individual class recordings to a course (paste YouTube link, set class number, attach PDF)
4. **Students** — See all enrolled students; invite new ones by phone/email; bulk-invite via file upload
5. **Announcements** — Write and publish notices to all students or just one course's students

Adding a class — simplified steps:

1. Log in to `/admin`
2. Click Courses → pick the course
3. Click "Add New Class"
4. Paste the YouTube URL, fill in title, class number, and date
5. Optionally upload a PDF
6. Click Save — students see it immediately

5. How Students Log In (Authentication)

Students don't use passwords. They log in using a **one-time code (OTP)** sent to their phone or email — the same way most banking apps work in India. This is handled by a service called **Clerk**.

- No password to forget
- Works on any device (mobile browser, laptop)
- New students receive an invite from the teacher first; they can't self-register

Known issue resolved: Indian phone numbers initially had SMS delivery issues. This was fixed by switching to email-only OTP where needed.

6. The Tech Stack — Plain English Explanation

The platform is built using six external services, each handling one specific job. All accounts are owned by the client (teacher); the developer (Sanjey) has access only as a team member.

Service	What it does	Simple analogy
Vercel	Hosts and serves the website	The building that houses the shop
Supabase	Stores all data (students, courses, classes, etc.)	The filing cabinet
Cloudflare R2	Stores uploaded files (PDFs, audio)	The storage room for physical files
Clerk	Handles student login (OTP)	The security guard at the door
Sentry	Catches and reports website errors automatically	The CCTV that alerts when something breaks
BetterStack	Checks every 3 minutes if the site is online	The security guard who calls you if the shop shuts unexpectedly
Resend	Sends welcome and invite emails	The post office
cron-job.org	Pings the database every 6 hours so it doesn't go to sleep	Setting an alarm so the database stays awake

The website itself is built with **Next.js** (a popular web framework) and **TypeScript** (a programming language). The visual design uses **Tailwind CSS** — a tool for making the interface look consistent without writing styling from scratch.

7. The Data Structure — What Gets Stored

The database has **nine tables** (think of each as a spreadsheet tab):

Table	What it stores
users	Every student and teacher account
courses	Course catalogue (e.g. "Vishnu Sahasranamam")
classes	Individual recorded sessions within a course
materials	PDFs and audio files linked to courses/classes
enrollments	Which student is enrolled in which course
announcements	Notices published by the teacher
class_views	Analytics — who watched which class and for how long
invites	Pending invitations for students not yet signed up
audit_log	A tamper-proof record of all admin actions

Privacy by design: Students can only see their own data. A student cannot see another student's enrollment or history. The teacher can see data for all their enrolled students only. This is enforced at the database level — it cannot be bypassed even if someone tries to hack the URL.

8. Security — How Access Is Controlled

Access control works in two layers:

Layer 1 — Clerk (the door): Every page in the student and admin areas requires a valid login. If someone isn't logged in, they are redirected to the sign-in page automatically. Anonymous visitors can only see the public-facing pages.

Layer 2 — Database rules (the safe): Even after logging in, each user can only read or write data they are allowed to. Examples:

- A student can only see courses they are enrolled in
- A teacher can only manage their own courses, not another teacher's
- No one (except the system itself) can delete audit logs

9. Accessibility Features (Important for Older Students)

Because the student base is predominantly 55+, these features were built in from the start:

- **A+ Font Toggle:** A button that increases all text size uniformly across the whole website. The preference is remembered — so if a student sets large text, it stays large every time they visit
- **Big tap targets:** Every button is at least 44px tall (the minimum Apple recommends for touchscreens)
- **Bottom navigation bar:** Like WhatsApp's tab bar — familiar and thumb-friendly on mobile
- **No password required:** OTP-only login avoids the frustration of forgotten passwords

10. Cost Structure — What Does This Cost?

The platform is designed to be **nearly free for the first 2 years** for a small school (under 100 students):

Year 1 — ~₹2,000/year (domain name only)

Service	Monthly cost
Vercel (hosting)	₹0
Supabase (database)	₹0
Clerk (login)	₹0

Service	Monthly cost
Cloudflare R2 (files)	₹0
Resend (emails)	₹0
Sentry (errors)	₹0
BetterStack (uptime)	₹0
Domain name	~₹1,000–₹2,000/year

When costs start appearing

Trigger	Approximate cost
Database exceeds 500 MB (after ~500 classes)	~₹2,100/month
Bandwidth exceeds 100 GB (~300+ active students/month)	~₹1,700/month
Files exceed 10 GB storage	₹1.25/GB/month

Bottom line: Costs only appear with growth. Growth is a good problem to have. Sanjey will flag approaching limits before charges kick in.

11. Current Project Status

What is built (Phase 1 — Complete)

- Full project structure and codebase set up
- Student navigation (Home, Courses, Materials, Announcements, Profile sections created)
- Login/authentication wired up via Clerk
- Design system (colours, fonts, brand tokens) configured
- A+ font size toggle — fully working, persists across visits
- Mobile navigation (bottom bar + hamburger drawer) — fully working with accessibility
- Database schema designed and migration scripts written (9 tables, all security rules in place)
- Error monitoring (Sentry) integrated
- Admin area structure created (Dashboard, Courses, Classes, Students, Announcements)
- Launch checklist and operational runbooks written

What is deferred (not yet built)

- Admin page content — the pages exist but have no content/forms yet
- Clerk webhook — the bridge that syncs a new login to the database (critical before launch)
- File upload/streaming utilities for Cloudflare R2

- Email templates in Resend
- Student progress tracking (marking classes as watched)
- Push notifications

12. The Launch Checklist (What Needs to Happen Before Go-Live)

This is the sequence of steps required to take the site from development to live:

24 hours before launch:

1. Run environment check script (verifies all settings are production-ready)
2. Confirm all service keys are switched from test mode to live mode
3. Verify Supabase daily backups are enabled
4. Confirm Cloudflare R2 bucket is the production bucket
5. Run a full smoke test (14+ automated tests) against the staging site

On launch day:

1. Add the domain name (`stotradhwani.com`) to Vercel
2. Update DNS at the domain registrar (two records need to be set)
3. Wait 5–30 minutes for the domain to propagate globally
4. Verify SSL certificate (padlock in browser)
5. Run final smoke test against the live domain
6. Manually walk through the golden path: sign in as student, open a course, watch a video, download a PDF, check announcements, sign in as admin, post an announcement
7. Enable BetterStack monitoring and cron-job.org keep-alive

After launch:

- Notify students via WhatsApp group with the sign-up link
- Pin `stotradhwani.com/sign-up` for new student onboarding

13. Post-Launch Backlog (Future Work)

Prioritised list of improvements planned after the initial launch:

High Priority (first 3 months)

- **PWA install icons** — allows Android users to add the site to their home screen like an app
- **Push notifications** — alert students when a new class or announcement is posted
- **Clerk production phone configuration** — verify Indian numbers (+91) work in live mode

Medium Priority (3–6 months)

- **Student progress tracking** — let students mark classes as "watched"
- **Analytics dashboard** — show teacher which classes are most watched
- **WhatsApp broadcast** — auto-send announcements to WhatsApp group

Future Enhancements

- **Offline caching** — let students access materials with poor connectivity
- **Admin UI in regional language** — Kannada or Tamil option to reduce dependence on developer for day-to-day operations
- **Per-course enrollment** — students only see courses they're specifically registered for (supports multiple batches)

14. What to Do When Something Breaks

The teacher can handle most common problems independently:

Symptom	Most likely cause	First action
Students can't sign in	Clerk is down or SMS failed	Check status.clerk.com
Site completely unreachable	Vercel outage	Check vercel-status.com
Blank pages / database error	Supabase has paused (free tier sleeps after 7 days of no activity)	Go to app.supabase.com → click "Restore project" → wait 60 seconds
Videos don't play	YouTube video is deleted or set to Private	Check the YouTube video in YouTube Studio
PDFs don't open	Cloudflare R2 issue	Contact Sanjey
Students see old content	Browser cache	Ask them to hold Shift + reload the page

The cron-job.org setup (once active) prevents the Supabase sleep issue entirely by pinging the site every 6 hours.

Sanjey's contact: sanjey8105@gmail.com — responds within 24 hours on business days

15. Key Decisions That Were Made (For PM Reference)

These are the technical choices locked in, and the reasons behind them — useful context for any future team discussion:

Decision	What it means	Why
Next.js 14 (not 15)	The website framework version	Next.js 15 has breaking changes; staying on 14 keeps things stable

Decision	What it means	Why
Clerk v5.7.6 pinned	The login service version	Latest Clerk requires Next.js 15; this version works with 14
Email-only OTP option	Indian phone SMS issues	Clerk's SMS had delivery problems for Indian numbers; email OTP is the reliable fallback
Relational database (Supabase/Postgres)	Database type	Structured data (courses, classes, enrollments) suits a table-based database better than a flexible document store
All accounts owned by client	Who holds the contracts	Billing and control stay with the school owner, not the developer
Service-role database access	How the app talks to the database	Simpler for a small school; avoids complex configuration while maintaining full security via code

16. Glossary (Terms You May Hear in Team Discussions)

Term	Plain-English meaning
OTP	One-time password — the code sent by SMS or email to log in
Next.js	The framework (toolkit) the website is built with
TypeScript	The programming language used
Supabase	The database service (where all data lives)
RLS (Row-Level Security)	Database rules that control exactly who can see which rows of data
Clerk	The login/authentication service
Cloudflare R2	The file storage service (for PDFs, audio files)
Vercel	The hosting platform (where the website runs)
Sentry	The error monitoring tool — catches bugs automatically
Migration	A script that sets up or changes the database structure
Webhook	An automatic notification sent between two services when something happens (e.g. "new user signed up")
Middleware	Code that runs before a page loads to check if the user is allowed in
PWA	Progressive Web App — a website that can behave like an app on a phone's home screen
Free tier	The usage limits below which a service charges nothing
Smoke test	A quick automated test that checks the most critical parts of the site are working
DNS	Domain Name System — the internet's address book; connects "stotradhwani.com" to the actual server
SSL/HTTPS	The padlock in the browser — means the connection is encrypted and secure

Term	Plain-English meaning
Seed data	Fake test data loaded into the database during development so developers can test the site