**What We Have So Far - Quick Summary**

**✅ Completed & Working**

**1. Database Schema (Complete!)**

* ✅ **File:** schema.sql artifact
* ✅ All tables designed and created
* ✅ Relationships, indexes, constraints
* ✅ Seed data with default admin user
* ✅ Views and triggers for automation
* ✅ Ready to use!

**Tables include:**

* training\_company - Company details
* users - Admins & Instructors (single table with boolean flags)
* session\_types - CBT, Module 1, Module 2, DAS, A2
* task\_configuration - Configurable tasks per session type
* training\_sessions - Training session records
* students - Student information and verification
* student\_tasks - Task completion tracking
* certificate\_batches - Certificate inventory (batches of 25)
* certificates - Individual certificate tracking
* certificate\_emails - Email delivery tracking
* audit\_log - Complete audit trail

**2. Flutter App (Working on Chrome)**

* ✅ **File:** main.dart artifact
* ✅ Student list screen
* ✅ Photo upload (student + license)
* ✅ Verification interface with match scores
* ✅ Student details form
* ✅ Task checklist (9 CBT tasks)
* ✅ Progress tracking
* ✅ Web-compatible (uses memory images, not file paths)

**Current workflow:**

1. Add new student
2. Upload photos
3. Verify identity (shows match %)
4. Enter student details
5. Complete task checklist
6. Save assessment

**3. Python Backend (Basic)**

* ✅ **File:** main.py artifact (original)
* ✅ Face verification endpoint
* ✅ Assessment save endpoint
* ✅ Statistics endpoint
* ✅ Basic SQLite database connection

**Runs on:** http://localhost:8000

**🔨 Ready to Build Next**

**Priority 1: User Authentication**

**What:** Login system for admins and instructors

**Needs:**

* Registration screen (Flutter)
* Login screen (Flutter)
* JWT authentication (Python backend)
* Password hashing (bcrypt)
* Protected routes
* Role-based access

**Priority 2: Certificate Batch Management**

**What:** Admin adds certificate batches

**Needs:**

* Admin dashboard (Flutter)
* Add certificate batch form
* Enter first number → auto-generate 25 certificates
* Display inventory
* "X certificates remaining" alerts

**Priority 3: Session Management**

**What:** Instructor creates sessions and adds multiple students

**Needs:**

* Quick session start screen
* Dropdown for session type
* Add multiple students to session
* Session-level task completion
* Individual student override tab

**Priority 4: Group Task Completion**

**What:** Check one task, applies to all students in session

**Needs:**

* Session-level checklist
* Student override table/tab
* Timestamp management
* Individual exception handling

**📂 Files You Have**

**Artifacts (copy from Claude):**

1. **schema.sql** - Complete database schema
2. **main.dart** - Flutter app (web-compatible)
3. **main.py** - Python backend (basic version)
4. **README.md** - This complete documentation

**Files You Need to Create:**

**Backend folder structure:**

backend/

├── main.py (copy from artifact)

├── schema.sql (copy from artifact)

├── requirements.txt (create - listed in README)

└── training.db (will be created by sqlite3)

**Flutter folder structure:**

training\_app/

├── lib/

│ └── main.dart (copy from artifact)

├── pubspec.yaml (modify - add dependencies)

└── (other Flutter files auto-generated)

**🎯 Your Design Decisions (Confirmed)**

1. ✅ **Single users table** with is\_admin and is\_instructor booleans
2. ✅ **Certificate batches** of 25, sequential numbering
3. ✅ **Session-level task completion** by default, with individual override
4. ✅ **Quick session start** - dropdown + Start button
5. ✅ **Admin manages:** Company details, certificate batches, instructors
6. ✅ **Instructors see:** Next certificate number + how many remaining
7. ✅ **Multiple sessions per day** allowed (same type even)
8. ✅ **Task configuration** per session type (9 tasks for CBT)
9. ✅ **Group timestamps** - all students get same time when task completed
10. ✅ **Override capability** - can mark individual students as failed/incomplete

**🚦 Current Status**

**What Works Right Now:**

* ✅ Chrome web app opens and runs
* ✅ Can upload photos (file picker on web)
* ✅ Verification shows match percentage (demo mode without backend)
* ✅ Can navigate through workflow
* ✅ Tasks can be checked off
* ✅ Database schema is complete and ready

**What Doesn't Work Yet:**

* ❌ No user login (everyone is anonymous)
* ❌ No real face recognition (backend not connected)
* ❌ No certificate management
* ❌ No session management
* ❌ Can only add one student at a time
* ❌ No group task completion
* ❌ No mobile deployment (Android/iOS)

**Known Issues:**

* Web version uses file picker, not camera (expected behavior)
* Demo face matching is random (need backend running for real AI)
* Data doesn't persist between sessions (no login/sessions yet)

**🎬 Next Steps When You Return**

1. **Review the complete README** - all documentation updated
2. **Test the current Chrome app** - make sure it still works
3. **Decide priority** - which feature to build first:
   * Authentication system?
   * Certificate management?
   * Session management?
4. **Build incrementally** - one feature at a time, test as we go
5. **Iterate and refine** - adjust based on real-world testing

**💡 Quick Start Commands**

**To run what you have:**

Terminal 1 - Backend (optional, for real face recognition):

cd backend

python main.py

Terminal 2 - Flutter app:

cd training\_app

flutter run -d chrome

**To reset/rebuild:**

# Flutter clean build

cd training\_app

flutter clean

flutter pub get

flutter run -d chrome

# Database reset

cd backend

rm training.db

sqlite3 training.db < schema.sql

**📞 Questions to Answer When You Return**

1. **Did the Chrome app work after restarting?**
2. **Any errors or issues to fix?**
3. **Which priority feature do you want to build first?**
4. **Any changes to the database design?**
5. **Ready to start on authentication, or something else?**

**Everything is documented and ready for you to continue!** 🚀

The foundation is solid:

* Database schema is complete
* Flutter app works on web
* Basic backend structure exists
* Clear roadmap ahead

When you return, we'll continue building incrementally. No rush - the design is documented and we can pick up right where we left off!

**Happy to continue when you're ready!** 👍