

Skills Bridge

Capstone 13 · EdTech · Talodabi Academy

1 · Project overview

You are designing **one vocational path** on a low-end phone with offline lessons. Certificate and progress matter more than a huge course library.

Deliverable	Full mobile MVP in Figma (not code)
Platform	Mobile app · Android-first · portrait
Screens	11 screens + 1 unhappy path (12 frames)
Features	3 core features
Bottom navigation	Bottom navigation with 4 items: Learn · My Courses · Progress · Profile
Prototype	4 linked flows

Build order: Auth (screens 1–4) → nav tabs (5–8) → core flow (9–11) → unhappy path → link prototypes.

2 · Problem & primary user

Millions of Nigerian youth aged 18 to 25 are unemployed or underemployed. Formal education does not teach practical income-generating skills like phone repair, tailoring, plumbing, or baking. Vocational training centres exist but are expensive or too far away. Practical knowledge is inaccessible.

Primary user

"Chisom, a 21-year-old with an SSCE certificate living in Aba who wants to learn phone repair but cannot afford trade school and has nowhere to start."

Geographic scope: National-level

3 · Goals

User goal: Learn a practical skill through short video lessons he can access and watch offline on his phone.

Business goal: Build Nigeria's largest vocational learning platform and connect certified graduates to gig job opportunities.

Success: In a quick test, someone **enrols**, watches or downloads **one lesson**, and finds their **progress / certificate** screen.

4 • Market context

QUESTION	ANSWER
Who pays	Chisom pays if certificate leads to income; NGOs, state skills programs, or employers may subsidize (assumption).
Why not WhatsApp	YouTube teaches skills but has no progress, offline lesson, or employable certificate in one path.
Design focus	Pick one trade (phone repair). Design offline download + certificate screen for low-end Android.

Competitors to check:

- **YouTube, TikTok** (free skill videos; no certificate path)
 - **AltSchool, Decagon** (paid tech bootcamps, not phone repair)
 - **State / NGO vocational programmes** (in-person; episodic)
 - **TVET colleges** (e.g. GTC-style GSM repair; far and long)
-

5 • Validate before you design

Interview 3–5 people who match your primary user. Update this brief if interviews contradict it. Do not start hi-fi until you complete the confirmation below.

FIELD	YOUR ANSWER
Validated user (name, age, city)	
Problem in one sentence	
One interview quote	
Success metric for your test	
Primary colour and tone	
One competitor you checked	

Interview prompts

- "Walk me through the last time this happened."
- "What did you use instead? WhatsApp, paper, or nothing?"
- "What would make you trust this on a cheap Android phone?"
- "What would stop you from using this every week?"

6 • Features & scope

v1 • three features

1. Short video-based skill courses (phone repair, tailoring, baking, plumbing) with offline download
2. Progress tracker showing lessons completed and percentage done
3. A completion certificate that can be downloaded and shared

Out of scope

- Job placement guarantee
- Ten trades in v1
- Live classes or coding bootcamp content

7 • Screens

Design every row in wireframe, high-fidelity UI, and prototype (where the flow applies).

#	LAYER	SCREEN	WHAT TO SHOW
1	Auth	Splash Screen	App name and a bold illustration of a person learning a trade.
2	Auth	Onboarding (2 screens)	Screen 1: 'Learn a skill, earn a living. Practical courses for real jobs.' Screen 2: Pick up to 3 skills you are interested in.
3	Auth	Sign Up Screen	Name, phone number, state, and PIN.
4	Auth	Sign In Screen	Phone number and PIN.
5	App shell	Learn (tab)	Grid of skill cards (icon, name, number of lessons, duration). A 'Continue Learning' card at the top if the user has an active course.
6	App shell	My Courses (tab)	Course title, instructor name, total lessons, estimated time, a brief description, and an 'Enrol' or 'Continue' button. Show a progress bar if already started.
7	App shell	Progress (tab)	Video player taking up the top 60% of screen. Lesson title and notes below. 'Next Lesson' button at the bottom. Offline download toggle.
8	App shell	Profile (tab)	A styled certificate with the user's name, skill earned, and completion date. 'Download' and 'Share' buttons.
9	Core flow	Course detail	Lessons count, Enrol/Continue button. Show bottom nav on post-sign-in screens.

10	Core flow	Lesson player	Video area, notes, Next lesson. Show bottom nav on post-sign-in screens.
11	Core flow	Certificate ready	Download/share certificate after completion. Show bottom nav on post-sign-in screens.

Layers: Auth = 1–4 · App shell = 5–8 · Core flow = 9–11

8 · Prototype flows

Link these in Figma. Examiners should click through without your voice-over.

Flow 1 · App entry

Splash → Onboarding → Sign Up or Sign In → first main tab

Flow 2 · Core task (all three features)

Sign In → Learn tab → Course detail → Lesson player → Certificate

Core flow screens: Course detail · Lesson player · Certificate ready

Flow 3 · Navigation

Link every nav tab so examiners can tap each bottom nav item. Tabs: Bottom navigation with 4 items: Learn · My Courses · Progress · Profile

Flow 4 · Unhappy path

Trigger the unhappy path edge case and link to that screen.

9 · Unhappy path

Design this as its own screen state (frame 12):

User's phone storage is full and they cannot download a lesson. Show: 'Not enough storage for this download. You can watch it online instead.' Offer a streaming button.

Link in **Flow 4**.

10 · Design system

RULE	DETAIL
Talodabi Academy Colour	You choose. Keep it simple and readable.

Type	You choose. Keep body text easy to read on a phone.
Tone	Design for your user, not for a tech audience.

Reuse on every screen:

- Primary button (active and disabled states)
- Input field (empty, filled, and error states)
- Card or list item used across the app
- Navigation bar with correct labels and icons
- Empty state: what the screen looks like with no data

11 · Submit & defense

Deliverables

#	DELIVERABLE	WHAT GOOD LOOKS LIKE
1	Wireframes	Every screen in this brief, plus the unhappy path. Layout and labels clear.
2	High-fidelity UI	Same frames polished. You choose colour and type. Bottom nav on every post-sign-in screen.
3	Prototype	Four linked flows (entry, core task, navigation, unhappy path). Examiners can use the app without your voice-over.
4	AI prototype	Figma Make: at least 2 screens. Share link.
5	Case study (1 page)	Problem · User · Who pays · Competitors · Solution · One design decision
6	Presentation (6–8 slides)	Name · Problem · User · Solution · Key screens · Flow demo · One decision

Before you present

- ☐ Every screen designed in wireframe and hi-fi
- ☐ Unhappy path screen designed and linked in the prototype
- ☐ Four prototype flows linked end to end
- ☐ At least one empty state on a list tab
- ☐ Shared components reused: button, input, card or row, bottom nav
- ☐ Case study names who pays and one competitor you checked
- ☐ You can explain one design decision in critique (why, not what)
- ☐ Completed 5 user interviews or 1 usability test
- ☐ Checked at least one competitor listed under Market context

