COURSE 2 · NO-CODE PRODUCT BUILDING & PROTOTYPING

BUILD SOMETHING YOU WOULD USE.

Turn your business website into a working, testable MVP.

THE JOURNEY

WEBSITE → PRODUCT

Business idea → **Useful action** → **Working MVP** → **Real evidence**

You already built the front door. Now we build what happens after someone enters.

TODAY

Three parts. One outcome.

01

Choose the tools

Understand the product stack and make a reasoned choice.

02

Review the websites

Find the clearest next step for every business idea.

03

Design the product

Create the user flow and improve one critical screen.

PART 1

TOOLS THAT TURN IDEAS INTO PRODUCTS

A website explains.
A product does something.

WEBSITE

"Here is our idea."

Pages, information, brand, contact.

PRODUCT

"Complete this useful action."

Input, logic, data, result, feedback.

MVP

The smallest version that can produce real learning.

One user	+	One problem	+	One useful flow	+	One signal

A signup, request, booking, generated result, saved record, payment, or completed task.

AI BUILDING TOOLS · OUR COURSE FOCUS

Lots of tools.
Two state-of-the-art options.

OpenAI Codex

Plan, build, debug and improve your project with an AI coding agent.

Anthropic Claude

Develop ideas and product logic; use Claude Code for hands-on coding.

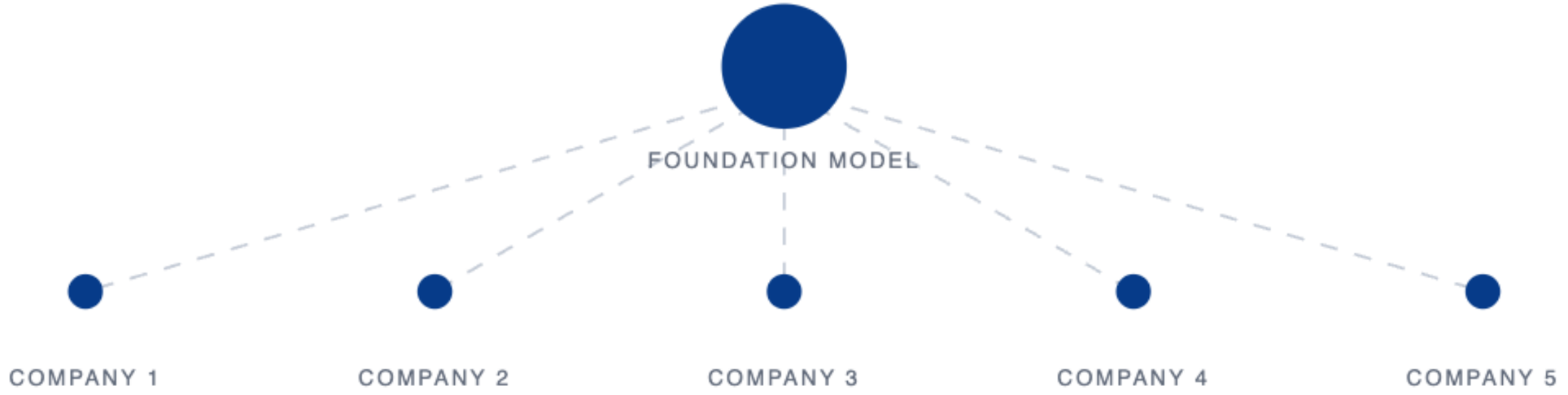
SOTA = State of the art.

Our focus for this course. The best fit depends on the task, model version and workflow.

THE BUSINESS LESSON

Everyone can now access intelligence.

The same foundational AI capabilities are available to your competitors.



Shared model access is the starting point for this strategic argument.

WHAT MAKES YOUR BUSINESS DIFFERENT?

Intelligence alone is no longer the moat.

When everyone can access it, access itself does not set you apart.



A moat is a competitive advantage that others find difficult to copy.

SO WHERE DOES THE ADVANTAGE GO?

Into deployment.

The advantage is where, how, and why you put intelligence to work.



Apply AI to a valuable problem, inside a useful workflow. Deployment here means real-world application. Diagram is illustrative, not measured results.

PROTOTYPE ARCHITECTURE

Most early products need only a few layers.



Webflow
polished interface

Glide
fast data apps

Bubble
custom workflows

AI code
flexible prototypes

CHOOSE YOUR BUILDING APPROACH

Match the tool to the task.

Bubble	Custom app workflows · check complexity and usage costs
Glide	Data-driven app prototype · check layout and data limits
Webflow	Website and content experience · plan app logic separately
Airtable	Structured records and lightweight workflows

Already using AI-generated code? Keep GitHub + Vercel and add only the missing capability.

Pick one tool for each job.

Build	Bubble · Glide · Webflow · AI coding tools
Data	Supabase · Airtable
Users	Clerk · Supabase Auth
Hosting	Vercel · Cloudflare
Payments	Stripe · RevenueCat

Analytics	PostHog · Google Analytics
Email	Resend
Errors	Sentry
Behaviour tracking	Hotjar · Microsoft Clarity
Collaboration	GitHub

PRODUCT TOOLS · FREE ALLOWANCES CHECKED 15 SEPTEMBER 2026

Build, measure, and learn.

Build & operate

Clerk

50,000 retained users/month

Supabase

50,000 active users/month

Cloudflare

Workers: 100,000 requests/day

Analytics

PostHog

1 million analytics events/month

Google Analytics

GA4 Standard: free; limits apply

Sentry

5,000 errors/month · 1 user

Behaviour tracking

Hotjar

Now routes to Contentsquare:
200,000 sessions/month · 1 project
Replay capture has separate limits.

Microsoft Clarity

Free · unlimited heatmaps,
websites and team members

Selected allowances, not all plan limits. Official pricing links are in the reference slide.

TWO MORE USEFUL SERVICES

Send updates. Manage subscriptions.

Resend

3,000 emails/month

Free transactional email allowance.
Daily cap: 100 emails.

RevenueCat

Free below $2,500 MTR

MTR = monthly tracked revenue.
Above the threshold: 1% of MTR.

Choose the services your product needs. You do not need every tool to launch a useful prototype.

Checked 15 September 2026 · resend.com/pricing · revenuecat.com/docs/welcome/set-up-revenuecat/account-management

WORKED EXAMPLE · A TUTORING BUSINESS

Build one booking request.

Choose subject → **Request session** → **See confirmation**

Keep

GitHub + Vercel
Existing website

Add

Supabase
Save requests

Optional

Resend
Email confirmation

A SECOND BUILDER PRINCIPLE

Use a service to move quickly. Build your own when the reason is strong.

USE A SERVICE

- Fast validation
- Reliable infrastructure
- Low early usage
- Small team

BUILD YOUR OWN

- Cost is now material
- The workflow is narrow
- It creates an advantage
- You can maintain it

Industry example: @levelsio on X · https://x.com/levelsio/status/2097692685775565031

EXAMPLES FROM THE POST

Small replacements for specific jobs.

Media

Image resizing with Sharp; screenshots; self-hosted streaming.

Operations

Error alerts, uptime monitoring, fleet management, backups.

Intelligence

AI moderation, NSFW detection, weather data, scraping.

Owning software also means owning maintenance.

Examples reported by @levelsio. Reported savings are personal claims; hosting, maintenance and model usage still cost time or money.

YOUR DECISION

Choose tools you can explain.

01 What job does this tool perform?

02 Why is it suitable for this MVP?

03 What alternative did you consider?

04 What limit, cost, or dependency could matter later?

PAUSE

COME BACK WITH YOUR WEBSITE OPEN.

Website · GitHub repository · business idea

PART 2

WEBSITE REVIEW LAB

FIVE MINUTES PER WEBSITE

We are looking for the next useful action.

01 What is the business idea?

02 Who is the user, and what problem do they have?

03 What can the visitor do today?

04 What single feature would turn this into an MVP?

05 What evidence should the feature collect?

THE CTA TEST

Every website needs one action with a real benefit.

For the user

A result, answer, booking, download, introduction, or saved task.

For the business

A lead, sale, interview, signup, signal of demand, or useful feedback.

For learning

Evidence that supports or challenges an assumption.

No “Learn more” without a meaningful next step.

REVIEW OUTPUT

Each team leaves with one sentence.

For [user], we will add [feature] to help them [benefit]. Our CTA is [action]. We will test [assumption] by observing [signal].

Write it down before moving to the next website.

PART 3

DESIGN THE USEFUL MOMENT

DESIGN PRINCIPLE

Design begins with a decision, then becomes a screen.

Goal → **Action** → **Feedback** → **Next step**

Before styling anything, draw the shortest path to value.

USER FLOW EXERCISE

One user. One task. Five boxes maximum.

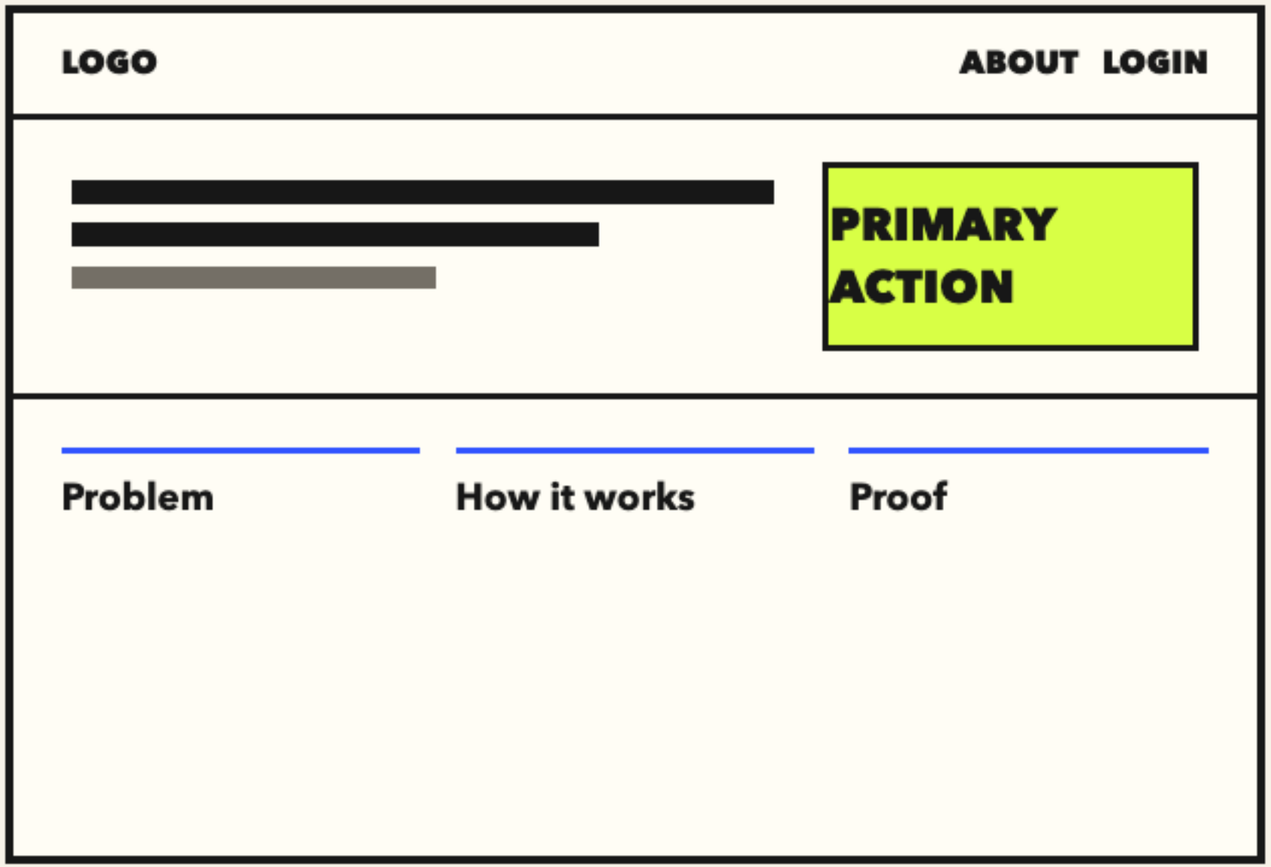
Arrives	Understands	Acts	Gets value	Returns / shares

Remove every step that does not help the user.

LANDING PAGE ANATOMY

Make the next step obvious.

- Who is this for?
- What useful result is promised?
- What should I do now?
- Why should I trust it?



DESIGN CLINIC · TUTORING WEBSITE

Replace vague promises with a useful action.

BEFORE

Welcome to TutorSpace

Innovative learning solutions for everyone.

Learn more

Who is it for? What happens next?

AFTER

Get help with your next maths exam.

Tell us your topic and availability. We will respond within two working days.

Request a tutor

Specific user, clear action, honest expectation.

VISUAL HIERARCHY

If everything is loud, nothing is clear.

1

Primary

The user's goal and action.

2

Supporting

How it works and why it helps.

3

Quiet

Details, navigation, legal, secondary actions.

USER-FLOW ARCHITECTURE · NO DEAD ENDS

Every screen offers a next step.

Home / browse → **Details** → **Request form** → **Confirmation**

Change your mind

Details → Back to results
Form → Edit selection

Something fails

Error → Fix input → Retry
No results → Adjust filters

Task completed

Confirmation → View request
View request → Home

Keep navigation visible, preserve form input, and offer Help → Contact support → Return to the task.

What happens after the click?

Ready

Label required fields. Explain what will happen.

Request a tutor

Working / error

Show progress. Preserve input if submission fails.

“Enter an email address so we can reply.”

Success

Confirm the request and set a next-step expectation.

“Request received. We will reply within two working days.”

GROUP DESIGN CHALLENGE · 45 MINUTES

Redesign one important flow together.

10

Define

User, problem, assumption, success signal.

15

Flow

Draw the path from arrival to useful result.

20

Screen

Design the critical screen and CTA.

End with a clickable or build-ready prototype.

PEER TEST · 10 MINUTES

Watch someone use your design.

1. Give a task: “Request help with a maths exam.”
2. Stay quiet. Observe hesitation, errors and completion.
3. Ask: “What did you expect to happen?”
4. Change one thing based on what you observed.

Peer testing checks usability. Later, test the business assumption with relevant potential users.

TWO BUILD SPRINTS

Build → test → improve.

01
Functional MVP

- Choose a key assumption
- Build the useful flow
- Deploy it
- Prepare a test

02
Evidence-led iteration

- Test with users
- Capture observations
- Prioritize changes
- Ship the next version

BEFORE YOU LEAVE

Save proof of today's work.

- One user, problem and testable assumption
- One selected tool and a reason for choosing it
- One improved flow and screen with a real CTA
- One peer observation and the change it prompted
- One next task in GitHub and a public course note

REFERENCE · CHECKED 15 SEPTEMBER 2026

Free-tier sources

- Clerk ↗
- Supabase ↗
- PostHog ↗
- Google Analytics ↗
- Cloudflare Workers ↗
- Sentry ↗
- Hotjar → Contentsquare ↗
- Replay limits ↗
- Microsoft Clarity ↗
- Resend ↗
- RevenueCat ↗

Pricing and allowances can change. Check the linked plan before choosing a service.

YOUR NEXT MOVE

CHOOSE ONE USEFUL ACTION. BUILD IT.

Business idea → MVP → user evidence → iteration