

# Base44 Build And Iterate

Turn the Stitch design into a working app, then refine it like a product team

## Where We Are

Yesterday, your team made it to the design layer.

You had already:

- Used the city budget to understand departments and current programs
- Used ChatGPT to extract useful department and program information
- Added Chamber of Commerce partners so your idea had real supply
- Built a win-win-win program concept where the department, partners, and participants all get value
- Expanded that program into a multi-day transformation
- Used Google Stitch to create the first version of the portal design

Today, we go all the way into Base44.

**Day 5 is where the design becomes a system people can click, test, and improve.**

The goal is not perfection on the first try.

The goal is to get a working version up fast, test the experience, and refine it based on the transformation you want to create for the audience.

### Day 5 Build Flow



## The Big Idea

Yesterday you designed what the program should look like.

Today you learn how software builders think:

| Builder Question                     | What It Means                                                                                          |
|--------------------------------------|--------------------------------------------------------------------------------------------------------|
| Who is this for?                     | The app should serve a real audience with a real problem.                                              |
| What transformation are we creating? | The app should help users move from their starting point to a better outcome.                          |
| What actions must users take?        | Every page should help someone complete a task, submit something, learn something, or make a decision. |
| What data do we need?                | The system should collect the information that proves the program is working.                          |
| Who needs to see what?               | Participants, organizers, partners, and admins may need different views.                               |
| What should we refine?               | Testing shows what is confusing, missing, weak, or too generic.                                        |

You are not just building a website.

You are building a program operating system.

## Today's Toolchain

| Tool            | What We Use It For Today                                                                             |
|-----------------|------------------------------------------------------------------------------------------------------|
| Google Stitch   | Export the design code/text from yesterday's portal concept.                                         |
| Base44          | Build the web app with authentication, landing page, auth pages, dashboards, forms, and sample data. |
| ChatGPT         | Refine the program, app requirements, test cases, and improvement prompts.                           |
| Google Docs     | Save the Stitch export, Base44 prompt, app link, feedback, and next iteration notes.                 |
| Team Discussion | Decide what the user should experience and what the audience transformation should feel like.        |

Base44 is the developer tool today.

Stitch gave you the design direction.

Base44 turns that direction into a working system.

## Prompt Cheat Codes

Yesterday, your team copied a sequence of prompts from ChatGPT into Google Stitch and got to a real design faster than most people expect.

Today is similar.

The prompt is not just a question.

It is a cheat code for getting AI to move quickly.

### Why prompts matter

A good contextual prompt gives the AI the setup, the goal, the audience, the constraints, and the expected output. When that context is strong, AI can help you move faster than someone using weak prompts, and much faster than someone not using AI at all.

We will study prompting more deeply tomorrow and during the refinement periods.

For today, remember the simple rule:

**Better context creates better output.**

Then use the prompts to build the web app, test it, refine it, and get it ready to share with other people.

## Step 1: Export From Stitch

In Google Stitch, open the design your team created.

Use the export or copy-to-code/text button.

Copy the design output to your clipboard.

Then paste it into your Google Doc before going to Base44.

### Why save the export first

Your Google Doc is the team command center. If the browser refreshes, Base44 changes the app, or someone needs to rebuild later, the team still has the design text and can reuse it.

| Google Doc Tab       | What Goes There                                                                 |
|----------------------|---------------------------------------------------------------------------------|
| Program Blueprint    | The 3-day transformation program and partner model.                             |
| Research             | City budget notes, department tables, Chamber partner options, and AI research. |
| Stitch Design Export | The copied text/code from Stitch.                                               |
| Base44 Prompt        | The exact prompt your developer pasted into Base44.                             |
| Build Link           | The app link, notes on what works, and notes on what needs fixing.              |
| Testing Notes        | Team feedback, confusing screens, missing features, and next prompts.           |

## Step 2: Start Base44 With The Right Frame

When you go into Base44, do not only say:

**Build me an app from this design.**

That is too weak.

You need to tell Base44:

- What the program is
- Who the users are

- What transformation the program creates
- What the app must help people do
- What pages you need
- What authentication should exist
- What data the system should save
- What roles or views are needed
- What design export it should follow

### Base44 Prompting Rule

Give Base44 the product context and the system requirements before you paste the design export. The design tells Base44 what it should look like. The requirements tell Base44 what it should do.

## Master Base44 Prompt

Use this as the first big prompt in Base44.

### Base44 App Builder Prompt

“Build a full web app for our community program.

Context: We are a student company building a 3-day transformation program from real city budget research, department priorities, and Chamber of Commerce partner matching.

Program:

- Program name: [insert program name]
- City: [Grand Rapids or Detroit]
- Department / priority: [insert department and program priority]
- Target audience: [insert audience]
- Transformation we want to create: [insert starting point → ending point]
- Partners: [insert at least 3 partner organizations]
- Our company role: We provide the digital technology that helps participants complete activities, organizers run the program, partners see ROI, and the department understand impact.

Build requirements:

1. Create a polished landing page that explains the program, audience, transformation, partners, and call to action.
2. Add authentication with sign up, login, and logout.
3. Create clean auth pages that match the landing page branding.
4. After login, users should see a dashboard.
5. Include these user types: participant, organizer/admin, and partner.
6. Participants need pages for:
  - Program overview
  - Day 1 activity
  - Day 2 activity
  - Day 3 activity
  - Reflection journal
  - Resource recommendations
  - Progress tracker
  - Final showcase/submission
7. Organizers/admins need pages for:
  - Participant list
  - Activity submissions
  - Progress overview
  - Feedback review
  - Impact metrics
8. Partners need pages or sections for:
  - Their role in the program

- Resources they provide
  - Participant engagement
  - ROI or impact signals
9. Create sample data so the app feels real.
  10. Create forms for intake, activity submissions, reflections, and feedback.
  11. Store user progress and submissions.
  12. Make the interface clean, modern, student-friendly, and pitch-ready for a city department or partner organization.
  13. Use the Stitch design export below as the visual direction. Keep the layout and style close to this design, but improve anything needed to make the app usable.

Stitch design export: [paste the full Stitch export here]

Before building, briefly summarize the app structure you are going to create. Then build the first working version.“

## Authentication Requirements

Authentication matters because the program is not just a public page.

Participants need progress saved.

Organizers need to review submissions.

Partners need to see the value they are creating.

That means the app needs users.

| User Type       | What They Should See                                                             |
|-----------------|----------------------------------------------------------------------------------|
| Public Visitor  | Landing page, program story, partner list, and sign-up call to action.           |
| Participant     | Dashboard, activities, reflections, progress, resources, final submission.       |
| Organizer/Admin | Participant records, submissions, progress data, feedback, impact summary.       |
| Partner         | Partner role, resources, engagement signals, participant outcomes, ROI evidence. |

### Auth Follow-Up Prompt

“Add or improve authentication.

Requirements:

- Public users should only see the landing page and auth pages.
- Users can sign up and log in.
- After login, users land on the correct dashboard.
- Include participant, organizer/admin, and partner views.
- Make the login and sign-up pages match the landing page branding.
- Make sure user progress, reflections, and submissions are saved to the correct user.
- Add helpful empty states for new users who have not submitted anything yet.“

## Step 3: Test Like A User

After Base44 builds the first version, do not just admire the screen.

Test it.

Move through the system like a participant.

☐ Can a public visitor understand the program from the landing page?

☐ Can a participant sign up or log in?

☐ Does the dashboard tell the participant what to do next?

☐ Are the 3 days of activities clear?

☐ Can users submit reflections or activity work?

☐ Does progress update after a user completes something?

☐ Can the organizer see participant progress?

☐ Can partners understand their role and ROI?

☐ Does the app collect data that proves the program worked?

☐ Does the app feel connected to the actual research and partner model?

## The Second Half: Go Back In Time

Once the app is working, we go backward.

Not backward because we failed.

Backward because real builders iterate.

The app will expose gaps in the program.

You may realize:

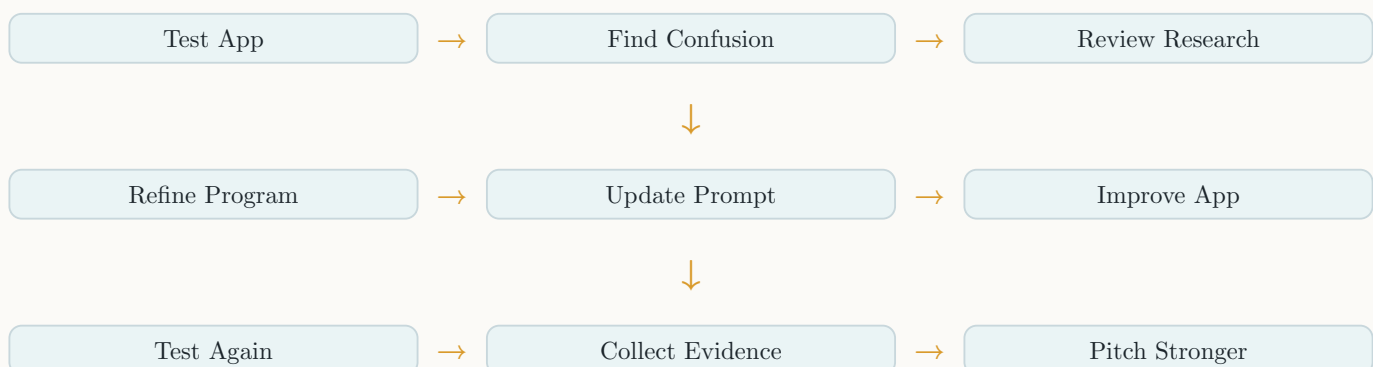
- The audience is too broad
- The activities are not specific enough
- The partner roles are weak
- The data being collected does not prove impact
- The dashboard does not guide the user
- The landing page does not explain the transformation
- The app is pretty but not useful yet

That is normal.

**The first build shows you what your idea is missing.**

Now we refine the program and the software at the same time.

### Iteration Loop



# Transformation First

Every review should start with the audience transformation.

Ask:

**Transformation Question**  
  
If a participant uses this program and app for 3 days, what should they understand, believe, practice, create, or be able to do by the end?

If the app does not help that transformation happen, refine it.

| If You Notice             | Ask The AI To Improve                                                                |
|---------------------------|--------------------------------------------------------------------------------------|
| The landing page is vague | Make the promise, audience, problem, and transformation clearer.                     |
| The dashboard feels empty | Add next steps, progress, deadlines, and activity cards.                             |
| Activities feel generic   | Tie each activity to the program goals and service provider lessons.                 |
| Partner value is unclear  | Add partner ROI, recognition, traffic, sales, recruiting, or impact signals.         |
| Data collection is weak   | Add forms and metrics that prove behavior change, learning, completion, or outcomes. |
| Admin view is weak        | Add tables, filters, progress summaries, and exportable impact data.                 |

# Refinement Prompt Stack

Use these after the first app is built.

**Landing Page Refinement Prompt**  
  
“Improve the landing page so it clearly explains:

- Who this program is for
- What problem the audience has
- The transformation participants experience
- Why the city department would care
- Which partners are involved
- What each partner provides
- Why someone should sign up

Make the page feel professional, local, practical, and pitch-ready.”

### Participant Experience Refinement Prompt

“Review the participant experience.

Improve the app so a participant always knows:

- What day they are on
- What lesson they are learning
- What activity they need to complete
- What they need to submit
- How the activity connects to their transformation
- What resources or partners can help them
- How much progress they have made

Add better activity instructions, progress states, and reflection prompts.“

### Organizer/Admin Refinement Prompt

“Improve the organizer/admin dashboard.

The organizer needs to quickly see:

- Who signed up
- Which participants completed each activity
- Which participants are stuck
- What reflections or submissions came in
- Which partners are being used
- What impact data has been collected
- What can be exported or summarized for the city department

Add useful tables, filters, cards, and summary metrics.“

### Partner ROI Refinement Prompt

“Improve the partner sections so every partner can see how they benefit.

For each partner, show:

- What they contribute
- Which activity they support
- What recognition they receive
- What audience they reach
- What data proves their value
- What follow-up opportunity they might get

Make it clear that this is a win-win-win program.“



### Research Alignment Prompt

“Compare this app against our research and program blueprint.

Find what is missing or too generic.

Check:

- Department priority alignment
- Chamber partner fit
- Audience transformation
- Activity quality
- Data collection
- Partner ROI
- Final showcase or deliverable
- Pitch readiness

Give us a prioritized list of the top 10 improvements to make next.”

## What The Developer Does Today

The developer role is not just pressing buttons.

The developer is responsible for turning the team’s idea into a usable system.

| Developer Task          | What Good Looks Like                                       |
|-------------------------|------------------------------------------------------------|
| Paste the Stitch export | The app follows the team’s design direction.               |
| Add app requirements    | Base44 knows the users, pages, data, and auth needs.       |
| Check the build         | The developer clicks through every important path.         |
| Find issues             | The developer names what is broken, confusing, or missing. |
| Prompt refinements      | The developer asks Base44 for specific improvements.       |
| Save links and notes    | The whole team can access the app, prompt, and next steps. |

## What Everyone Else Does Today

While the developer is building, the rest of the team still works.

| Role            | Day 5 Focus                                                                                               |
|-----------------|-----------------------------------------------------------------------------------------------------------|
| Researcher      | Checks whether the app still matches the department, program, Chamber partners, and opportunity research. |
| Project Manager | Keeps Google Docs organized, saves prompts, tracks app links, and records testing feedback.               |
| Designer        | Checks whether the Base44 build still matches the Stitch design and improves layout notes.                |
| Developer       | Builds in Base44, tests flows, and prompts improvements.                                                  |

The team should talk while testing.

Do not let one person silently build while everyone else waits.

# End Of Class Deliverable

By the end of class, each team should have:

□ A saved Stitch design export in Google Docs

□ A working Base44 app link

□ Authentication or auth plan started

□ Three activity pages or flows

□ Organizer/admin view or plan

□ At least three refinement prompts saved

□ A saved Base44 master prompt

□ Landing page with clear program story

□ Participant dashboard

□ At least one form that collects useful program data

□ Testing notes with the top improvements needed

□ A clearer understanding of how the program should work

## What Today Is Really Teaching

Software engineering is not just code. It is turning a real human process into a system, testing that system, finding what does not work yet, and improving it until the system supports the transformation you promised.

# Tomorrow's Bridge

Today we build the first working system.

Tomorrow we keep iterating.

We will use the app to sharpen the program, improve the activities, strengthen the partner story, and make the opportunity report feel more real.

## Day 5 To Day 6

