

# Publish, Prompt, And Refine

Publish the Base44 app, then deepen the program through contextual engineering

## Where We Are Today

The last five class days have been one connected build story.

Teams started with real city budgets, found department priorities, researched programs, mapped Chamber partners, designed win-win-win program concepts, created software ideas, moved into Stitch, and then used Base44 to turn those ideas into working app prototypes.

Some parts moved fast. Some parts may have felt semi-blind in the moment. That is because we were walking through a full pipeline before stopping to name every piece of it.

Today we go deeper.

**Today we name the full story, publish the app, and refine the program behind the software.**

The first part of class is technical. We will use Base44's Publish button to put the app live on the internet.

The second part is product work. We will review the program each team created, improve the activities, strengthen the flow, connect the app more tightly to the speakers and concepts being taught, and use research to make the experience more specific.

### Today's Flow



## The Tutorial Narrative

Today's tutorial is not only about publishing.

The tutorial is the full story of how a local opportunity becomes a live software offer:

## From Opportunity To Software Offer



Here is the narrative:

1. A city budget shows what problems the city already spends money on.
2. Departments show who owns those problems and what programs already exist.
3. Program research shows what has been tried, what gaps still exist, and what pain points matter.
4. Chamber directories show businesses, nonprofits, and service providers that can help deliver a solution.
5. The team becomes the broker by designing a win-win-win program around the department, partners, and participants.
6. The software tool makes the program easier to run, easier to track, and easier to pitch.
7. Publishing turns the prototype into a live website that stakeholders can actually open.
8. Contextual engineering helps the team refine the app until it feels valid, specific, and attractive.

### What students have really been building

You are not just building an app. You are building evidence that your team can understand local demand, organize collaborators, design a program, and provide the software system that helps the program run.

## How We Got Here

Some of the work across the last five days may have felt fast or semi-blind.

That is normal. The class was moving through a full opportunity-to-software pipeline:

### The Full Opportunity Pipeline



The prompts were not random. They were guiding teams through the way real consultants, product strategists, and software builders find demand, map stakeholders, create a program offer, and then build a tool that makes the offer more valuable.

| Prompt Stage                       | What It Was Really Doing                                                                                                        |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Budget upload                      | Turning the city fiscal year plan into a demand map: departments, funding, priorities, and issues the city already cares about. |
| Department table                   | Finding department directors, staff, budgets, programs, plans, and pain points so the team could speak to real decision-makers. |
| Five-year program research         | Studying what the department has already tried so the new idea does not feel disconnected from local reality.                   |
| Chamber partner search             | Finding private-sector or nonprofit supply that could help deliver the program while creating partner ROI.                      |
| Program mechanics                  | Turning an idea into a structured program with roles, flows, lessons, activities, deliverables, and operating logic.            |
| Software tool table                | Identifying where the student company can enter as the technology partner and create value.                                     |
| Three-day transformation blueprint | Defining the audience, promise, lessons, activities, data, partner roles, metrics, and final showcase.                          |
| Product architecture               | Translating the program into app users, dashboards, forms, data flows, tracking, and impact reports.                            |
| Stitch design                      | Creating the visual portal structure so the app had a clear user experience before coding.                                      |
| Base44 build                       | Turning the blueprint and design into a working software prototype with auth, dashboards, activities, and sample data.          |

**Today we make the software feel less like a demo and more like a serious offer.**

That means each team should refine for two audiences at the same time:

- Demand managers: department leaders, program managers, city staff, school staff, or community leaders who own the problem.
- Collaborators: Chamber businesses, nonprofits, service providers, sponsors, mentors, venues, and other partners who can help deliver the program.

Your app should make both groups think:

#### **The credibility test**

This team understands the problem, knows the stakeholders, has a practical program, can track outcomes, and built a tool that would make the program easier to run.

## **Why We Are Publishing**

Base44 helped teams build quickly. Now the app needs a real public home.

The better path for today is to use Base44's built-in Publish button. Base44 can host the application for us, which means students do not need extra hosting or deployment setup.

## The publishing goal

Each team should leave with a live Base44 website link they can open, test, share, screenshot, and use inside the final opportunity proposal report.

That means students get a practical product loop:

- Build in Base44
- Preview the app
- Publish the app
- Open the live website
- Test the user flow
- Ask ChatGPT what to improve
- Ask Base44 to update the app
- Publish again
- See the website improve

## Base44 Publish Tutorial

We will walk through this together in class.

The screenshots below show the exact publishing flow.

### Step 1: Preview The App

Start inside Base44 in Preview mode.

Click through the app before publishing. Make sure the landing page, participant dashboard, activity pages, partner pages, organizer views, forms, and sample data are good enough to share.

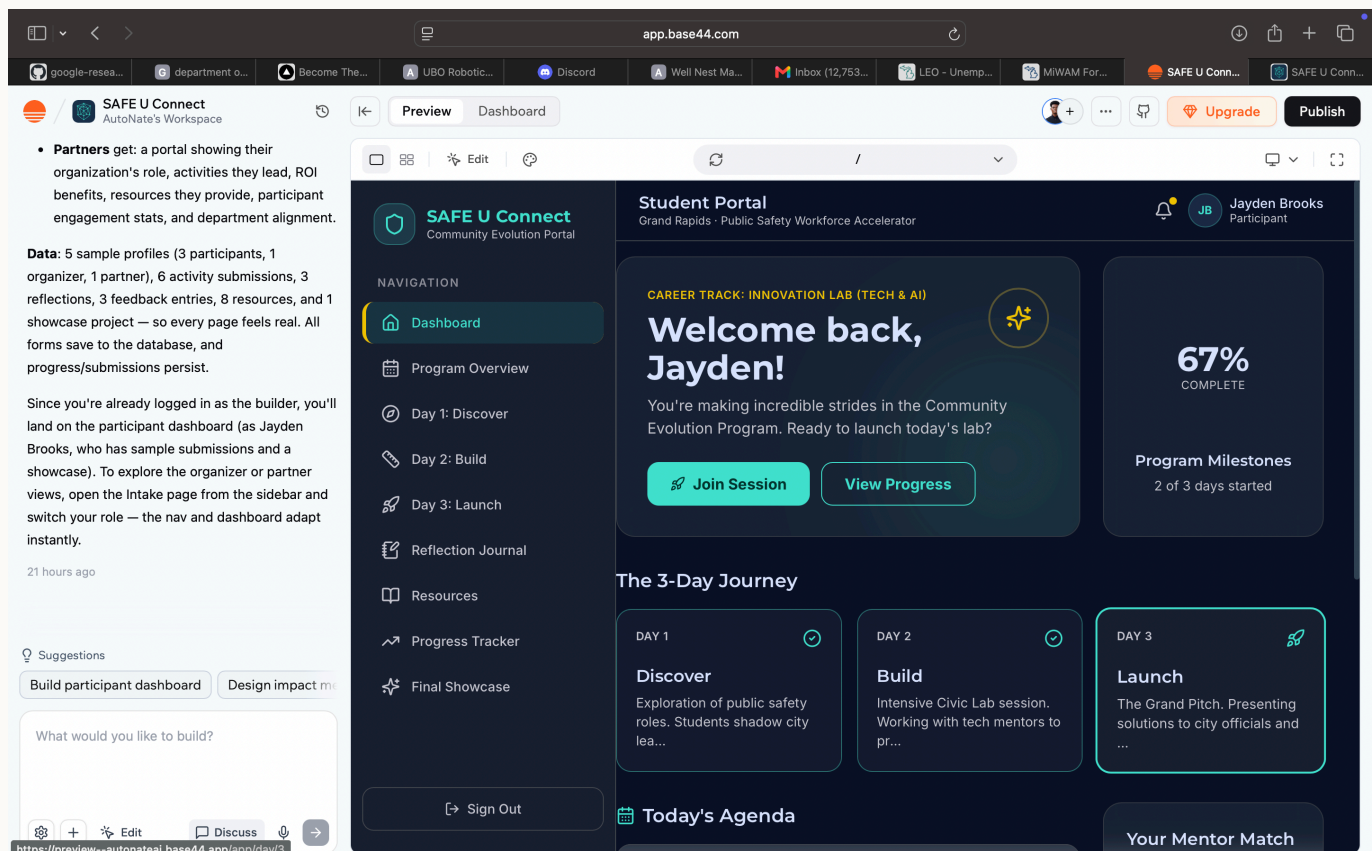


Figure 1: Base44 Preview shows the app before it is published. Use this screen to test the dashboard and navigation.

☐ The app has a clear landing page

☐ The dashboard loads

☐ The 3-day activity flow is visible

☐ Buttons and navigation work

☐ Forms are present where users need to submit information

☐ The app has sample data so it feels real

☐ The partner and organizer value is visible

☐ The app is ready for someone outside the team to open

## Step 2: Click Publish

When the app is ready, click the **Publish** button in the top-right corner of Base44.

Base44 will package the app and host it on a Base44 web address.

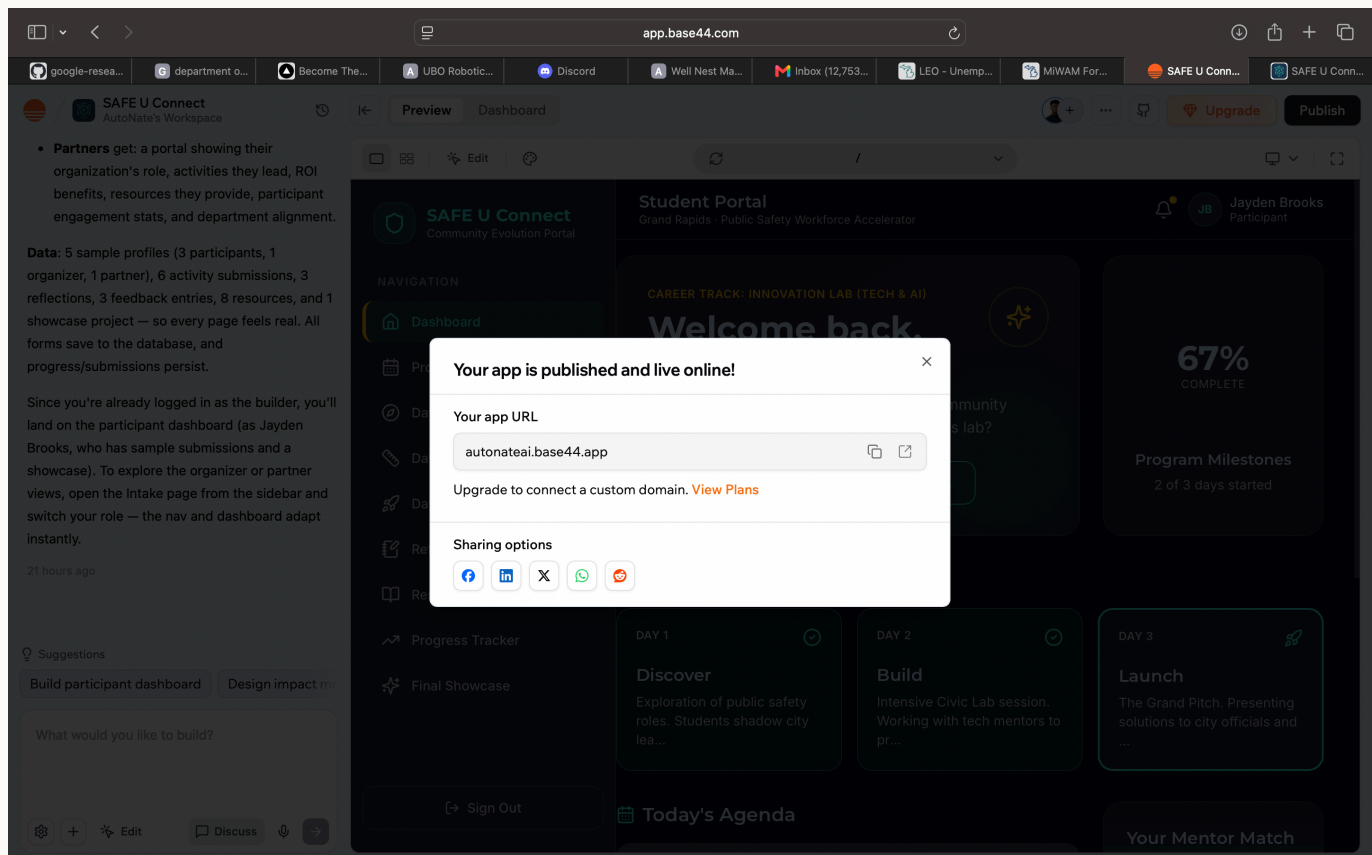


Figure 2: After clicking Publish, Base44 shows the live app URL. Copy this URL into the team's Google Doc.

## Step 3: Copy The Live URL

After publishing, Base44 will show a message that the app is live online.

Copy the app URL and save it in the Google Doc.

The app URL will look similar to:

**your-subdomain.base44.app**

This means Base44 is hosting the app. The website lives on Base44's servers, and students can share the link without setting up extra hosting tools.

### Step 4: Change The Subdomain Name

The part before `.base44.app` is the subdomain.

For example:

| URL Part                                  | Meaning                                                              |
|-------------------------------------------|----------------------------------------------------------------------|
| <code>safe-u-connect.base44.app</code>    | A Base44-hosted app using the <code>safe-u-connect</code> subdomain. |
| <code>career-night-gr.base44.app</code>   | A Base44-hosted app using a different team/program subdomain.        |
| <code>your-program-name.base44.app</code> | A placeholder showing that the team can choose a clearer name.       |

Teams should choose a subdomain that is short, readable, and connected to the program.

Good subdomain names:

- `safe-u-connect`
- `career-night-gr`
- `youth-career-launch`
- `public-safety-lab`
- `well-nest-detroit`

**Custom domain note**

Base44 hosts the app on a `.base44.app` address. If a team wanted the Base44 part removed and wanted a fully custom domain like `safeuconnect.org`, that would usually require a paid/custom domain setup. That is outside the scope of this course. For today, the Base44 subdomain is enough.

### Step 5: Open The Live Site

Open the published URL in a new tab.

This is the public version of the application. It is the link students can use for testing, screenshots, proposals, and partner conversations.



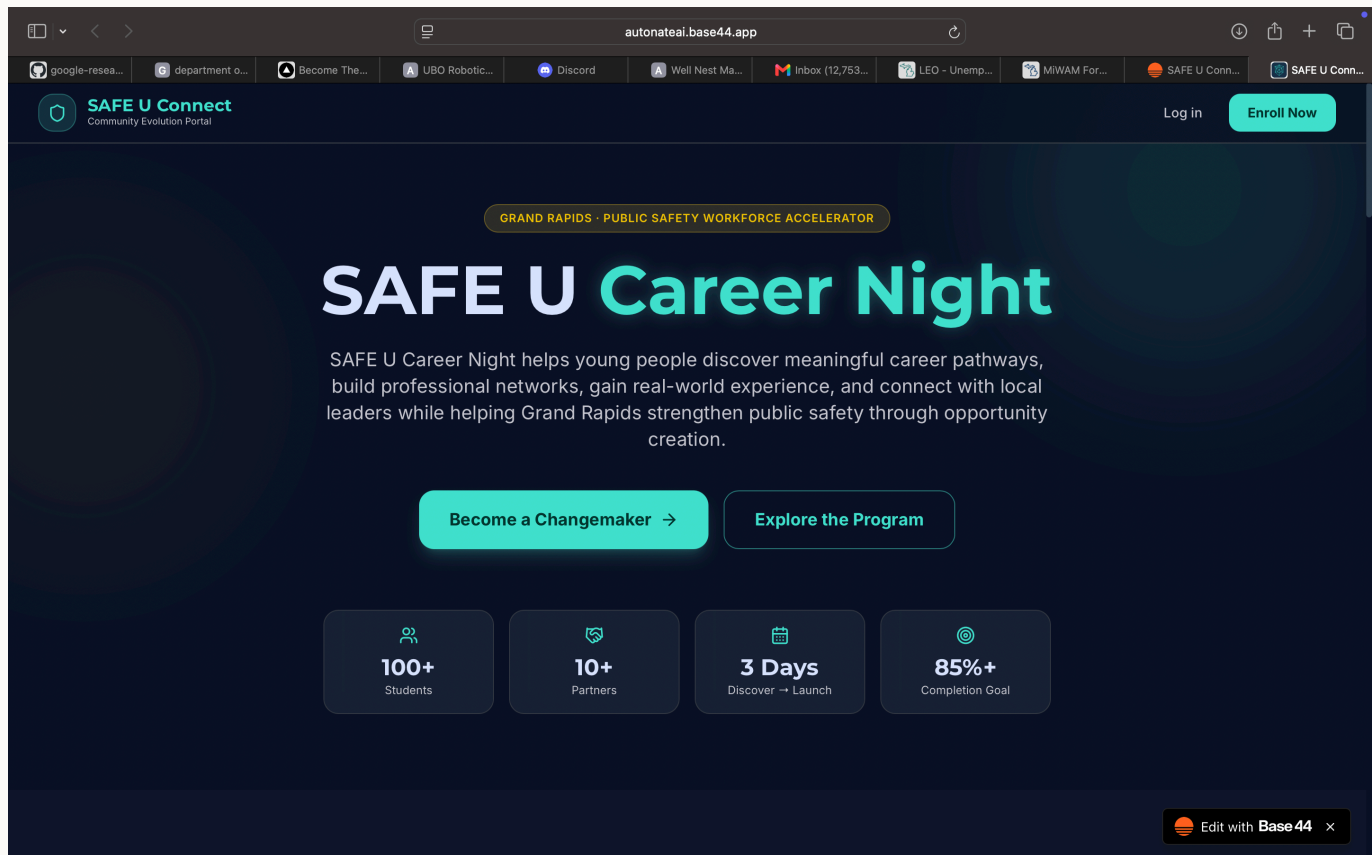


Figure 3: The live Base44 site is what partners, funders, department staff, and participants can open.

## Step 6: Test The Live Site

Testing the published site matters more than seeing the publish message.

- ☐ The landing page loads
- ☐ Images, styles, and icons appear
- ☐ Login or sign-up works if the app has auth
- ☐ Activity pages make sense
- ☐ Partner value is visible
- ☐ The app works on a phone-sized screen
- ☐ The app name and program promise are clear
- ☐ Navigation works
- ☐ Participant dashboard loads
- ☐ Forms and submissions work
- ☐ Organizer or impact views show useful data

## What Happens After Publishing

Once the live site works, the team has a real product loop.

From this point forward, Base44 can keep improving the app. After each meaningful update, publish again and test the live website.

**The live website becomes the testing ground for the program.**

Every team should save:

- Live Base44 app link
- Base44 project link
- Chosen subdomain name
- Screenshots of the app flow
- Notes on what works
- Known issues or confusing screens
- Next refinement prompts

# Prompt Engineering Gets Deeper Today

Across the first part of the program, we used prompt engineering to move through research, strategy, design, and app building.

Today we practice contextual engineering.

Prompt engineering is asking AI for the right output.

Contextual engineering is building the full context around the AI so it can help improve the whole system.

| Prompt Engineering   | Contextual Engineering                                                                                    |
|----------------------|-----------------------------------------------------------------------------------------------------------|
| Ask for a page       | Give the program goal, audience, research, data, partners, and desired transformation.                    |
| Ask for an activity  | Explain what the speaker taught, what students should practice, and what evidence the app should collect. |
| Ask for a dashboard  | Explain who uses it, what decisions they need to make, and what success metrics matter.                   |
| Ask for improvements | Compare the app against the department priority, award-winning examples, partner ROI, and user journey.   |

**Today's deeper skill**

We are not just prompting for screens. We are giving AI the research, goals, constraints, examples, and feedback it needs to help us refine the program.

# What To Fine Tune Today

The app already has a first version. Today, each team should decide which specific part of the flow needs the most improvement.

Do not try to improve everything at once. Pick the part that will make the program feel more valid, more useful, or more attractive to a real stakeholder.



| Flow Area              | What To Improve                                                                                                                              |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Demand story           | Make the landing page connect directly to the city department, budget priority, local pain point, and audience need.                         |
| Program promise        | Make the one-sentence promise clear enough that a department lead or partner understands the value in 10 seconds.                            |
| Partner ROI            | Show why each collaborator benefits: visibility, referrals, customers, workforce pipeline, community impact, data, or relationship building. |
| Participant journey    | Make the three-day flow feel like a transformation instead of three disconnected tasks.                                                      |
| Interactive activities | Turn passive reading into forms, choices, challenges, reflections, submissions, resource matches, or final artifacts.                        |
| Data collection        | Collect evidence that proves progress, participation, learning, behavior change, partner engagement, or program outcomes.                    |
| Organizer dashboard    | Give demand managers a clear view of signups, completion, blockers, submissions, partner usage, and impact metrics.                          |
| Final showcase         | Make the final deliverable something the participant can show and the department or partner can understand quickly.                          |

### Flow Selection Prompt

“Review our current app and program.

Context:

- City: [Grand Rapids or Detroit]
- Department: [insert department]
- Department priority or program: [insert priority]
- Target audience: [insert audience]
- Main partners: [insert partners]
- Our company role: We are the technology partner helping run, track, and prove the program.

Identify the three weakest parts of the current flow.

Rank them by which improvement would create the most validity and attraction for:

1. The department or demand manager
2. The Chamber partners or collaborators
3. The participants
4. Our student company as the software partner

For each weak spot, recommend a specific Base44 prompt to improve the app.“

## Refining The Application

After publishing, teams will review the program they created and improve the app around it.

The app should not feel like a random set of pages. It should flow with the concepts being taught by the speakers and the transformation the program promises.

This is where teams go back to the ChatGPT conversation and the Google Doc.

The ChatGPT thread should become the strategy room. Base44 should become the build room.

## ChatGPT helps you understand what to improve. Base44 helps you make the improvement real in the app.

Start by asking ChatGPT to explain the program back to you in simple language.

### Simple Program Explanation Prompt

“Using our full conversation, explain our program in simple language.

Answer these questions:

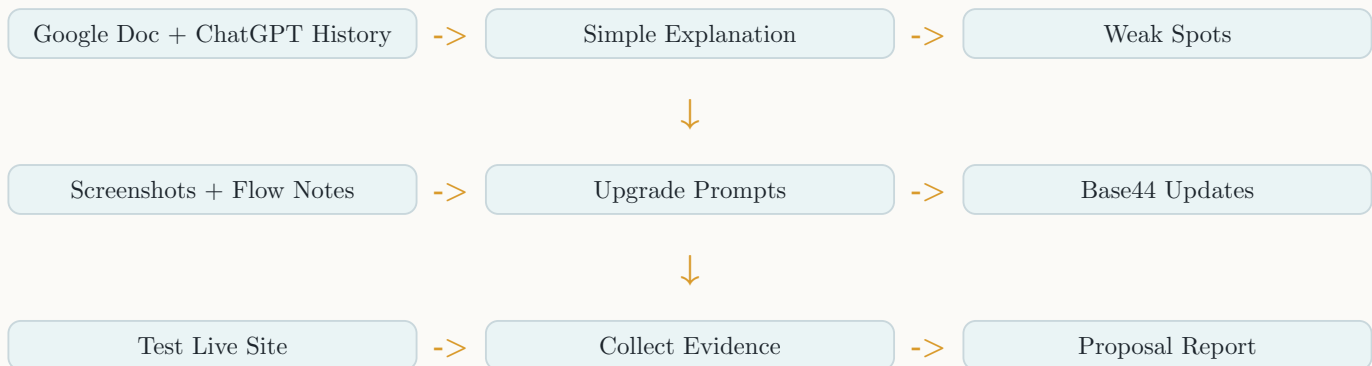
1. What is this program about?
2. Who is it for?
3. What problem is it solving?
4. What transformation is the program trying to create?
5. Why would the city department or funding people care?
6. Why would each partner or collaborator care?
7. What role does our app play in making the program work?
8. What is confusing, weak, or not specific enough yet?

Explain it like we are preparing to pitch this to a city department, funder, school, or Chamber partner.“

Then use that explanation to decide what to update.

Teams should move through this loop:

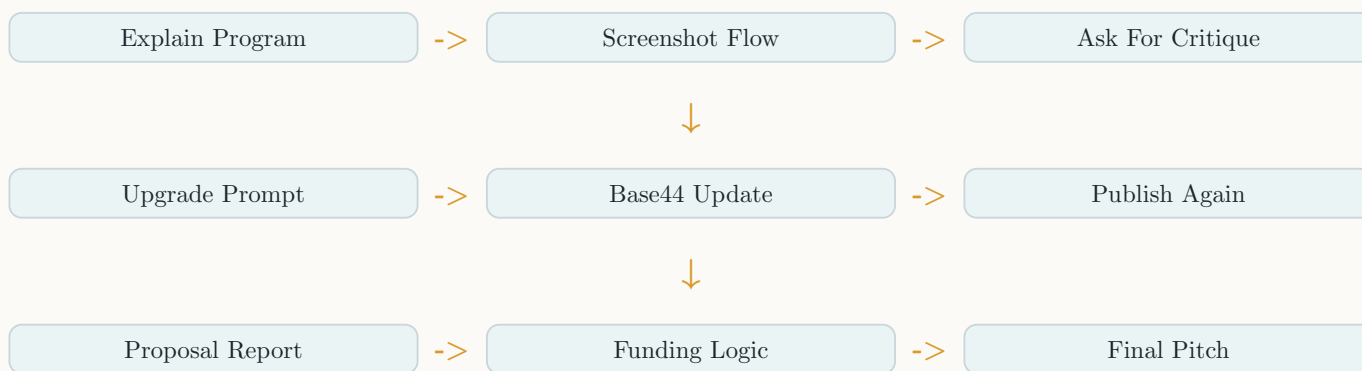
### ChatGPT To Base44 Iteration Loop



Ask:

- What does the participant learn first?
- What should they practice next?
- What should they create?
- What should they submit?
- What should the app track?
- What should the team, department, or partner be able to see?
- What evidence proves the program worked?
- Which app screen best proves that the program creates the promised transformation?
- Which app screen would make a funder, city leader, or partner more interested?

## Refinement Loop



## Use Screenshots To Upgrade The Flow

Do not only describe the app to ChatGPT. Show it.

Take screenshots of the important parts of the app:

- Landing page
- Program overview
- Day 1 activity
- Day 2 activity
- Day 3 activity
- Intake or reflection form
- Progress tracker
- Partner/resource page
- Organizer dashboard
- Impact metrics
- Final showcase/report

Upload the screenshots to ChatGPT with your Google Doc research and ask for upgrades.

### Screenshot Flow Critique Prompt

“I am uploading screenshots of our current app flow plus our program research.

Act as a program strategist, funder reviewer, instructional designer, and software product architect.

Review the screenshots and tell us:

1. What story does the app currently tell?
2. Where does the flow feel confusing or weak?
3. Which screens do not clearly connect to the program transformation?
4. Which activities feel too passive?
5. Which screens would fail to impress a city department, funder, or partner?
6. What data are we missing to prove the program works?
7. What should we ask Base44 to upgrade first?

Give us a prioritized upgrade list and write Base44-ready prompts for the top five improvements.”

This makes the work visual. Students can point to a real screen and ask:

### Screen-by-screen question

What would make this screen more useful, more interactive, more convincing, or more connected to the program transformation?

# Make Activities More Interactive

Today, teams should improve the app by adding stronger activities.

An activity is not just text on a page. A strong activity asks the user to do something, submit something, reflect on something, or make a decision.

| Weak Activity          | Stronger Interactive Activity                              |
|------------------------|------------------------------------------------------------|
| Read about the program | Answer an intake question that personalizes the next step. |
| Watch a lesson         | Complete a short challenge connected to the lesson.        |
| Write a reflection     | Use guided prompts with saved progress and feedback.       |
| Look at resources      | Choose a partner resource and explain how it helps.        |
| Finish the program     | Submit a final artifact, plan, pitch, or showcase item.    |

Teams can also ask ChatGPT what JavaScript packages could make the activities feel more exciting and useful. This does not mean adding random effects. The package should support the learning goal.

| Tool Idea         | How It Could Support The Activity                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------------|
| Three.js          | Create 3D scenes, interactive maps, simulations, product models, journey worlds, or visual metaphors for the transformation. |
| React Flow        | Build drag-and-drop process maps, partner ecosystems, program flows, or decision trees.                                      |
| D3.js or Recharts | Show impact metrics, budget comparisons, participant progress, partner ROI, and dashboard data.                              |
| Framer Motion     | Make activity transitions, progress feedback, and completed states feel polished and responsive.                             |
| Map libraries     | Show local resources, partner locations, service areas, or neighborhood opportunity maps.                                    |
| Canvas tools      | Let participants sketch, annotate, build boards, or create visual plans inside the app.                                      |

### Activity Refinement Prompt

“Review our current app activities and make them more interactive.

Context:

- Program name: [insert program]
- Target audience: [insert audience]
- Main transformation: [starting point -> ending point]
- Concepts taught by the speakers: [insert concepts]
- Partners/resources available: [insert partners]

Improve the app so each activity includes:

- A clear purpose
- A user action
- A saved submission
- A reflection question
- A progress state
- A connection to the program transformation
- Data that helps organizers understand impact“

### Three.js And JS Package Upgrade Prompt

“Review our program and app activities.

We want to make the activities more exciting, interactive, and useful without adding random decoration.

Suggest ways we could use Three.js or other JavaScript packages to improve the learning experience.

For each idea, explain:

- Which activity it improves
- What the participant does
- Why it supports the transformation
- What data the app should collect
- Which package or technique could help
- Whether this is MVP, nice-to-have, or too advanced for now

Then write a Base44-ready prompt for the best interactive upgrade.“

## ChatGPT First, Base44 Second

The workflow today should be:

1. Bring research and screenshots into ChatGPT.
2. Ask ChatGPT to explain the program simply.
3. Ask ChatGPT to identify weak spots in the flow.
4. Ask ChatGPT for upgrades that improve transformation, funding logic, partner ROI, and app interaction.
5. Copy the best upgrade prompt into Base44.
6. Let Base44 update the app.
7. Test the live site again after publishing.
8. Save the before/after notes in the Google Doc.

### Why this order matters

If you ask Base44 to update the app before ChatGPT helps you understand the program logic, you may get prettier screens that still do not prove the program works. Think first, then build.

# Final Opportunity Proposal Report

The final deliverable is not only the app.

Teams should use their Google Doc, ChatGPT conversation, screenshots, research findings, and published app to create a final opportunity proposal report.

This report should explain why the opportunity is real, why the program is practical, and how the application drives the activities and transformations the program is designed to create.

| Report Section             | What It Should Prove                                                                                 |
|----------------------------|------------------------------------------------------------------------------------------------------|
| Opportunity overview       | The team found a real department priority, budget signal, or community problem.                      |
| Department and funding fit | The program connects to what the city, school, funder, or demand manager already cares about.        |
| Target audience            | The team knows who the program serves and what starting problem they face.                           |
| Program transformation     | The three-day experience moves participants from a clear starting point to a clear ending outcome.   |
| Partner model              | The collaborators each have a role, contribution, and reason to participate.                         |
| Application role           | The app is necessary because it guides activities, collects data, tracks progress, and shows impact. |
| Activity flow              | Each activity is interactive and connected to the transformation.                                    |
| Impact metrics             | The team can explain what data proves the program is working.                                        |
| Screenshots and live link  | The proposal shows the actual app, not just an idea.                                                 |
| Next iteration             | The team knows what to improve after feedback from participants, partners, or funding people.        |

## Opportunity Proposal Report Prompt

“Using our Google Doc, our ChatGPT conversation, our app screenshots, and our published app link, help us create a final opportunity proposal report.

The audience is a city department, funding group, school leader, Chamber partner, or program manager.

The report should explain:

1. The local opportunity we found
2. The department priority or funding logic behind it
3. The target audience and starting problem
4. The transformation our three-day program creates
5. The partner/collaborator model
6. Why each partner would care
7. How the application drives the activities
8. How the app tracks progress and impact
9. What screenshots prove the app is real
10. What data would show the program worked
11. What we would improve next with more time

Make the report clear, professional, practical, and appropriate for funding people.“

# Refinement Prompt Stack

Use these prompts after the app is published and the team has chosen which part of the flow to improve first.

## Demand Manager Validity Prompt

“Improve our app so it feels credible to a department lead, city staff member, program manager, or other demand manager.

Use this context:

- Department: [insert department]
- Budget priority or pain point: [insert priority]
- Existing programs we researched: [insert examples]
- Target audience: [insert audience]
- Program promise: [insert promise]

Improve the landing page, program overview, organizer dashboard, and impact metrics so a decision-maker can quickly see:

- Why this program matches their real priority
- Who it serves
- What happens during the three days
- What partners are involved
- What data is collected
- What outcomes can be reported
- Why our software makes the program easier to run“

## Collaborator Attraction Prompt

“Improve the app so collaborators and Chamber partners can clearly see why they should join this program.

Partners: [insert partner list]

For each partner, improve the app so it explains:

- What the partner contributes
- Which activity they support
- What audience they reach
- What business, community, recruiting, visibility, or impact value they receive
- What data proves their role mattered
- What follow-up opportunity the program creates

Add partner cards, ROI sections, partner dashboard details, and any missing resource links.“



### Program Flow Tightening Prompt

“Tighten the three-day program flow inside the app.

Make the program feel like a clear transformation:

- Day 1 should diagnose the starting point and introduce the core concept.
- Day 2 should help participants practice with partners, resources, or service providers.
- Day 3 should help participants create a final artifact, plan, pitch, showcase, or submission.

For each day, improve:

- Lesson title
- Activity instructions
- User action
- Submission form
- Reflection prompt
- Progress state
- Partner connection
- Data collected
- How it prepares the participant for the final showcase“

### Software Offer Prompt

“Improve the app so our student company looks like a serious software partner.

The app should show that we can:

- Help participants complete the program
- Help organizers manage the program
- Help partners see ROI
- Help the department understand impact
- Collect useful data
- Generate a final report or showcase

Add or improve features that make our software offer more valuable, pitchable, and practical.“

## Research Before You Refine

Some teams may go back and research programs that have already won awards, earned recognition, or produced strong results in their department area.

That research can make the app sharper.

For example, if your program is connected to parks, public health, youth employment, small business, housing, public safety, sustainability, or arts and culture, look for:

- Award-winning city programs
- Department reports
- Case studies
- Program dashboards
- Community partner examples
- Metrics used to prove success
- Activities that participants actually complete

### Research rule

Do not copy another program. Study what works, then adapt the pattern to your audience, city, partners, and 3-day transformation.

### Award-Winning Program Research Prompt

“Help us research strong examples for our program area.

Program area: [parks / health / youth employment / small business / housing / public safety / sustainability / arts / other] City or department priority: [insert priority] Target audience: [insert audience]

Find patterns from award-winning or high-performing programs:

- What made the program effective?
- What activities did participants complete?
- What partners were involved?
- What data did the program track?
- What outcomes did it report?
- What could we adapt into our 3-day program and app?

Turn the findings into 10 specific improvements for our Base44 app.“

## What I Will Be Doing In Class

---

After the publishing tutorial, I will come around to each group.

We will look at your live app, your program concept, your activities, your partner model, and your tracking. The goal is to help each group make the tool more refined for what they are trying to build.

Be ready to show:

☐ Your live Base44 app link

☐ Your Base44 subdomain name

☐ Your Base44 project

☐ Screenshots of the main app flow

☐ Your program blueprint

☐ Your target audience

☐ Your three-day activity flow

☐ Your partner list

☐ What data your app collects

☐ What feels weak or unfinished

☐ What you want the app to become next

## Today's Deliverable

---

By the end of class, each team should have:

- A published Base44 app
- A live website link saved in the Google Doc
- At least one publish-test-refine cycle completed
- More interactive activities
- Research notes that support the next iteration
- A draft opportunity proposal report
- A clear `.base44.app`` subdomain
- Screenshots of the app flow
- A refined program flow
- A clearer tracking or impact plan
- A list of the next Base44 prompts to run
- A stronger app that matches the program vision

## The Big Idea

---

Today connects two worlds.

First, we learn how builders publish software so other people can use it.

Then we learn how builders refine software so it becomes more useful, more specific, and more connected to a real human outcome.

**Publishing makes the app real. Contextual engineering makes the app worth using.**