

HBS AI Workshop: Building with AI

Harvard Business School - Launch Technology Ventures - Spring 2025

Contents

| | |
|---|-----------|
| Overview | 2 |
| Goal | 2 |
| Objectives | 2 |
| Lesson Plan | 2 |
| Skills & Tools Covered | 2 |
| Pre-Work (Homework to be completed before class) | 3 |
| Level 0: MVP / MVT Product Specification (Group) | 3 |
| Level 1: Basic Website with ChatGPT (Individual) | 4 |
| Step 0: Troubleshooting: Leveraging LLMs to Debug and Learn | 4 |
| Step 1: Set up your initial project | 5 |
| Step 2: Deploy your project to the public internet | 8 |
| <i>Submission: Level 1: Basic Website Responses</i> | 12 |
| Level 2: Enhancing Your Webpage (Group) | 12 |
| Improving Your Website | 12 |
| Hypothesis Testing & AI Demos | 12 |
| <i>Submission: Level 2: Website Enhancement Responses</i> | 13 |
| Level 3: An Interactive App | 13 |
| Leveraging AI App Builders | 13 |
| App Builders | 14 |
| Suggested Path to an MVP / MVT | 14 |
| Step 0: Clarifying the outcome | 14 |
| Step 1: Defining product requirements | 14 |
| Step 2: Creating an Initial App Builder Prompt | 15 |
| Step 3: Iterate! | 16 |
| <i>Submission: Level 3: Interactive App Responses</i> | 16 |
| Office Hours | 16 |
| Appendix | 17 |
| Example Level 3 Projects | 17 |
| Example Level 2 Website Enhancements | 18 |
| Basic | 18 |
| Intermediate / Semi-Technical | 19 |
| Technical | 19 |
| Class Session Plan | 19 |
| Credits | 20 |

Overview

Goal

At the end of this exercise, you'll have gained hands-on experience building using AI-powered tools that you'll be able to use to further test business ventures.

Objectives

- Help students gain practical hands-on experience building web apps using AI tools.
- Empower students across varying levels of sophistication (beginner, intermediate, advanced).
- Encourage peer-to-peer learning and demonstrate varied approaches.
- Provide students with practical tools, tips, and resources to effectively use AI tools.

With a range of experience in the class, we'll be looking to the more technical students to serve as peer experts. Those with the least technical experience should level up their technical abilities, and those with the most experience should level up how they can empower others to use the latest AI tools.

Lesson Plan

Students will organize into groups of 2-3 students.

The groups will **complete homework and submit before class** - building several progressive levels of [minimum viable products](#) (MVPs) / [minimum viable tests](#) (MVTs).

The class session will select several groups to present demos of their work and discuss reflections, takeaways, and next steps to incorporating the experience into future work.

Skills & Tools Covered

- Effective prompting techniques
- Basic Front-end (FE) design
- Introductory Back-end (BE) integration
- Database (DB) basics
- Product management mindset: defining requirements, constraints, and hypothesis-driven approach
- AI-assisted HTML generation (e.g. ChatGPT)
- Visual Studio Code setup and use
- App generator tools (e.g. v0.dev, bolt.new, loveable.dev, replit.com)
- AI-Media generation tools (e.g. HeyGen, Suno.ai, Runway)

Pre-Work (Homework to be completed before class)

Form groups of 2-3.

Make sure to skim through the full assignment before starting, and start early to **complete and submit before class!**

The exercise is built to several progressive levels:

- **Level 0:** Specify an MVP / MVT (Group)
- **Level 1:** Create and launch a basic HTML website using ChatGPT (Individual)
- **Level 2:** Enhance and polish interactive pages with advanced AI prompting and creative tools (Group)
- **Level 3:** Develop simple AI-powered apps for hypothesis testing (Front-end, Back-end, Database) (Group)

Level 0: MVP / MVT Product Specification (Group)

Choose a product idea, what you will test, and how you will test it with an MVP ([minimum viable product](#)) / MVT ([minimum viable test](#)).¹

Read through Levels 1-3 of the exercise to get an understanding of the progressive levels of building:

- Level 1: A Basic Website with ChatGPT
- Level 2: Additional Polish on that Website
- Level 3: An Interactive Application

We recommend that this be an idea that one of you in the group is working on for Level 1 and Level 2. For Level 3, we encourage you to come up with an MVT you can build within the time for class preparation.

We've also provided some example apps that you can choose for Level 3 if you're having trouble scoping:

See [Appendix: Example Level 3 Projects](#)

Make sure to explicitly state in the submission below what hypothesis you plan to test with each level of MVP / MVT. It can be as simple as testing positioning with a customer or as complicated as the actual functionality of an application.

¹ Note: Links in this document made with Google Docs may take a few seconds to redirect.

Write these down and share them with your teammate(s).

Discussion: Level 0: Product Specification Responses

- What is your product idea?
 - What are the hypotheses you are going to test?
 - How do you plan to test them with Levels 1-3?
 - What is your rationale behind choosing these tests?
-

Level 1: Basic Website with ChatGPT (Individual)

Creating Your Own Landing Page with ChatGPT, VS Code, and GitHub

Note 1: *If you're experienced with web development, you can move quickly through this and Level 2 to put a site on GitHub Pages and move to [Level 3](#).*

Note 2: *We recommend using GPT 4.5 if you have access to that model. As the time of this writing, HBS ChatGPT Edu accounts may only have o3-mini-high or 4o. These are sufficient for the exercise as well.*

In Level 1, you're going to build, edit, and deploy your own code to the internet with ChatGPT using VS Code and Github Pages.²

For non-technical students, our goal is to empower you to use ChatGPT to leverage tools used by professional developers today - even if you have zero experience with software programming.

We'll provide two initial paths: either using ChatGPT or following the manual instructions.

Each student should do this individually.

Step 0: Troubleshooting: Leveraging LLMs to Debug and Learn³

One of the best use cases for LLMs is to ask them for help when you run into errors (i.e. "bugs") and need to "debug" your program or issue. Here are some tips from ChatGPT to get you started:

² We're using VSCode because it is a free tool widely used by software professionals to develop code. Github is also a widely used professional tool with free features, including Github pages.

ChatGPT: [What is a text editor?](#)

³ ChatGPT: [Why would a computer engineer start their instructions with "Step 0"?](#)

ChatGPT: [Debugging Strategies with ChatGPT](#)⁴

How can I use ChatGPT to debug building a basic webpage with VS Code, basic HTML, CSS, JavaScript, and GitHub Pages when I run into errors?

What are some suggestions on how to input information, errors, or other challenges when I sometimes have trouble understanding how to articulate the problem?

Assume I do not know how to code.

If you get stuck, keep trying different questions and screenshots. You can also come to [office hours](#).

Step 1: Set up your initial project

We're going to install Visual Studio Code (i.e. "VS Code") a text editor used by software engineers to edit code, create a project folder, create our first HTML file, and test viewing it.

Here's an example prompt to get you started:

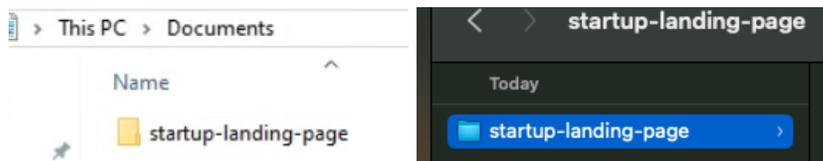
ChatGPT: [VS Code Project Setup](#)

Assume that I am an MBA student who does not know how to code and you're an expert, patient front end-engineer.

Can you walk me through the steps to install a new version of VS Code, create a new project folder, and create an index.html file for a basic landing page website?

Alternatively, you can follow these manual instructions:

- 1) Install Visual Studio Code (i.e. "VS Code"): [Visual Studio Code Download Link](#)⁵
- 2) Create a new folder⁶ in Windows Explorer (Windows) or Finder (Mac). You can name it anything you want. We called it "startup landing page" below.



⁴ OpenAI also has a more [technical guide on "prompt engineering"](#) and [Prompt Engineering Guide](#) goes in further depth.

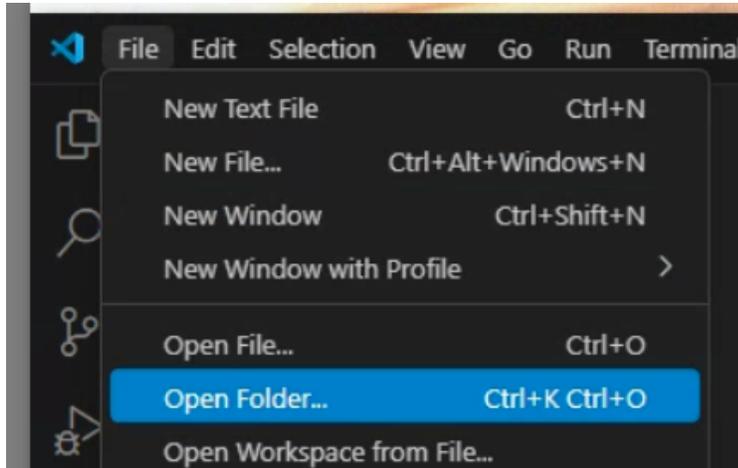
⁵ ChatGPT: [What is a text editor?](#)

⁶ Folders are also sometimes called "[directories](#)."

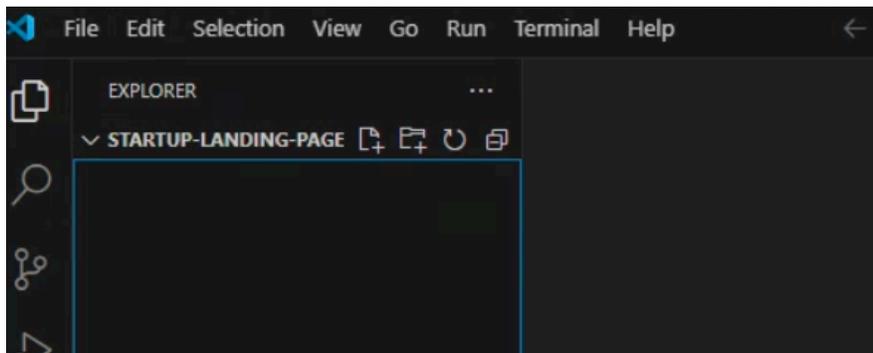
Windows Explorer.

Mac Finder.

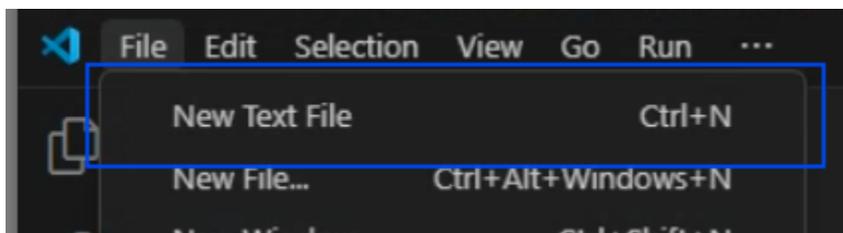
- 3) Open your folder in VS Code:
 - a) Open VS Code
 - b) File → Open Folder → Navigate to your recently created folder



- c) You should see the name of your folder and a blank section on the left-hand side of your screen.



- 4) Create the main page for your website.
Create a new file. VS Code → File → New Text File



- 5) Save the file with the name **index.html**.⁷ File → Save or Ctrl+S (Windows) or Cmd+S (Mac)
Make sure to save the file inside of the folder you just created.

⁷ ChatGPT: [What is an index.html file?](#)

- 6) Generate your first HTML code⁸ using ChatGPT.
Copy and paste the code generated into the index.html file and save it with Ctrl+S (Windows) or Cmd+S (Mac).

Here is an example prompt:

ChatGPT: [Create a Startup Landing Page](#)

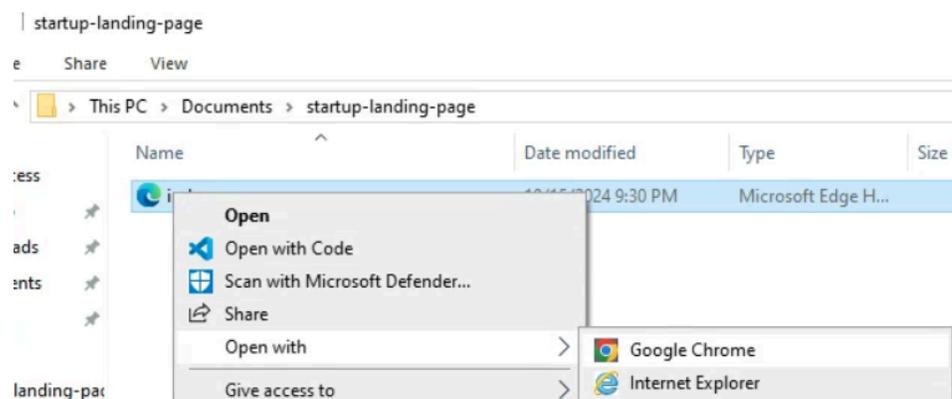
Assume you're an expert frontend software engineer and you're developing a landing page for a new startup run by a non-technical MBA.

For a new startup <enter your startup name here>

Can you generate a landing page that I can test?

Please give me the code all as one file.

- 7) Test your landing page.
Navigate to the file in Windows Explorer (Windows) or Finder (Mac)
Right-click the file and “Open With” your preferred web browser (e.g. Chrome)



You should see your initial landing page up and running.

Checkpoint: You’ve got a working webpage running locally on your computer!

Take a closer look at the code that you copied into the index.html file. Notice that this is human-readable. You should be able to see a `<title>` tag with the title of the webpage, a

⁸ ChatGPT: [What is HTML, CSS, and JavaScript?](#)

`<body>` tag with the main content of the web page, and potentially some code inside of a `<style>` tag that is referring to other elements in the page and adjusting their appearance.

We're going to use ChatGPT to evolve this, but one of the best ways to gain leverage with ChatGPT is by using it to learn more about the tasks you're instructing it to accomplish.

The more you know about the actual thing that ChatGPT needs to do, the better you can instruct it on what to do.

What to learn more about HTML and CSS?

Here's an example prompt:

ChatGPT: [Basics of HTML and CSS for Non-technical MBAs](#)

Can you explain the basics of HTML, CSS, and JavaScript as they pertain to making a startup landing page.

Please give an overview and detail that would be helpful for a Harvard Business School MBA student who is learning to code for the first time and working on a project to make a startup landing page with VS Code and GitHub Pages.

Assume they are recopying and pasting the files for each change.

Step 2: Deploy your project to the public internet

Next, we're going to deploy your web page to the public internet using GitHub Pages.

Here's an example prompt to get you started:

ChatGPT: [How do I deploy a web page to GitHub Pages without knowing how to code?](#)

I'm a non-technical MBA student at Harvard Business School and don't know how to code.

I've set up VS Code, and created an index.html landing page for a startup in a folder on my local computer.

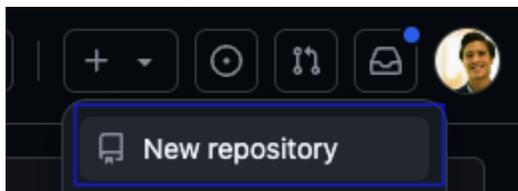
Can you walk me through how to set up an account on Github.com and deploy my page to GitHub Pages?

Note: If ChatGPT asks you to do anything from the command line, terminal, or “Git” (not to be confused with “Github”) and you’re not familiar with those tools, just tell it so and ask it to walk you through how to do it through the web browser or “GUI.”⁹

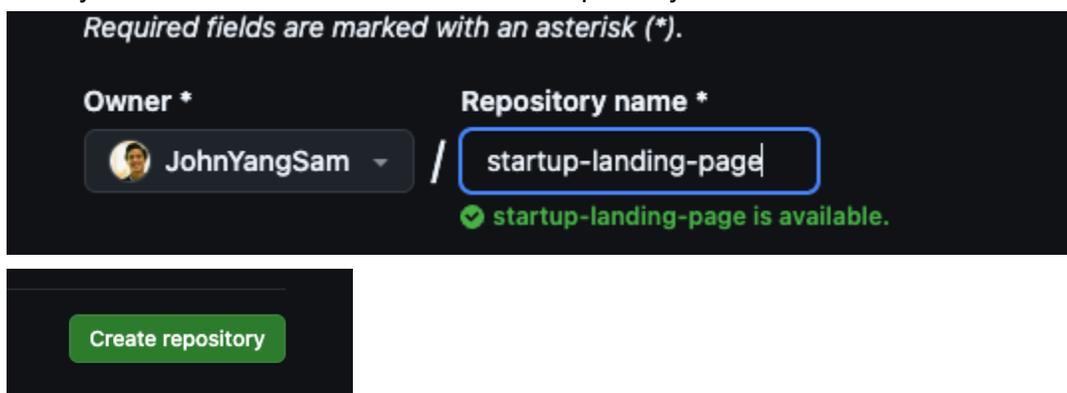
Alternatively, you can follow these manual instructions:

- 1) Sign up for a free account on Github.com.
- 2) Create a new repository. A repository is the top-level folder or “directory” where you will store your code on GitHub.

Your landing page will be `http<your GitHub username>.github.io/<your repository name>`
Click the “+” sign in at the top right of the page near your profile picture.
Click “New Repository”



Fill in your desired name and click “Create repository”

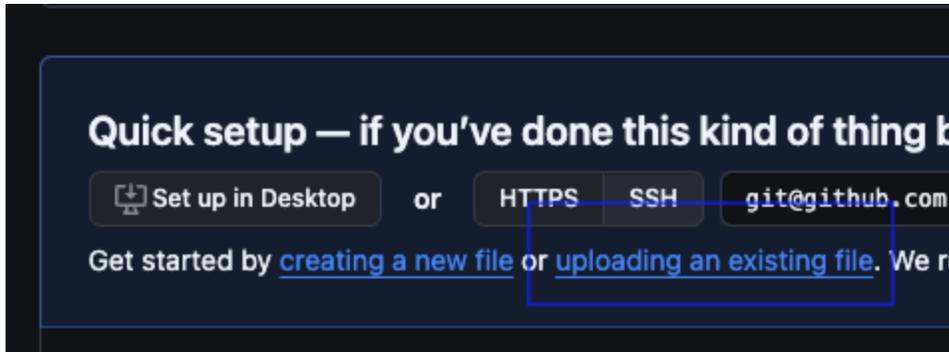
A screenshot of the GitHub "Create repository" form. At the top, it says "Required fields are marked with an asterisk (*)." Below this, there are two main fields: "Owner *" and "Repository name *". The "Owner" field is a dropdown menu showing "JohnYangSam" with a profile picture. The "Repository name" field is a text input box containing "startup-landing-page". Below the "Repository name" field, there is a green checkmark icon followed by the text "startup-landing-page is available." At the bottom of the form, there is a green button with the text "Create repository".

You should be taken to the repository page: <https://github.com/<your username>/<your repository name>>

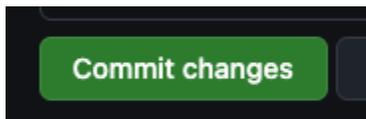
- 3) Upload your index.html file to your repository.

Click the “upload an existing file” link in the blue bar in the middle of the page.

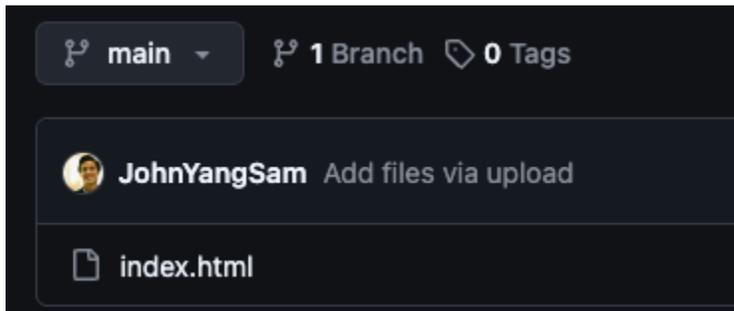
⁹ ChatGPT: [What is a GUI?](#)



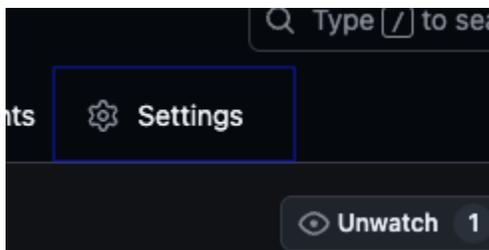
Drag your index.html file and click “Commit changes” to save it.



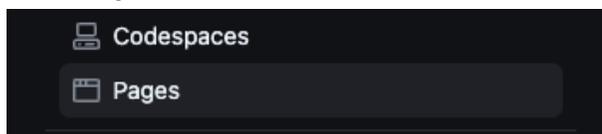
You should see your file in your repository



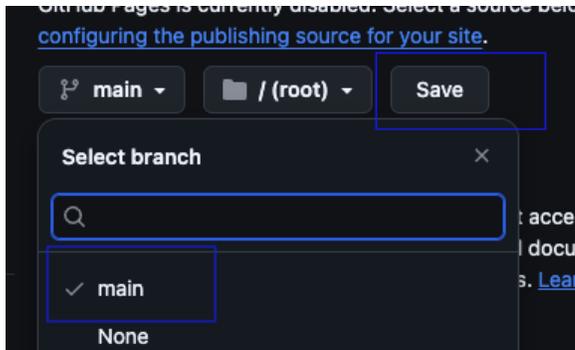
- 4) To turn on GitHub Pages to display your web page, click the repository “Settings” next to “Insights”



- 5) Click “Pages” in the left hand sidebar under the “Code and Automation” Section

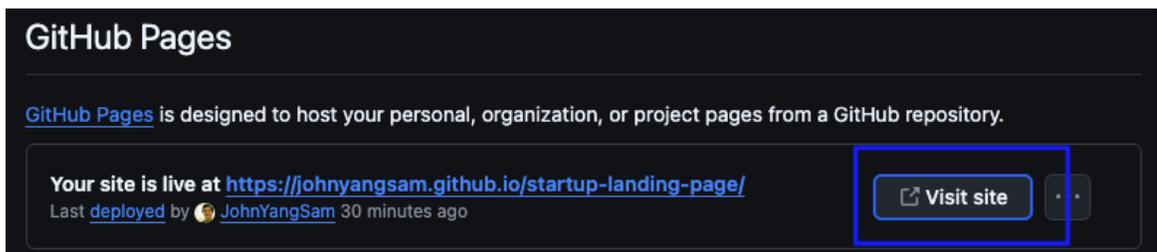


- 6) Make sure the following settings are set:
Build: “Deploy from a branch” (should be the default)
Branch: “main” and “/ (root)” and click “Save” next to those dropdowns (see screenshot):



Note: This only needs to be done once as a setting. Afterward, you should be able to see your updates once the files have been committed (though it may take several seconds or require you to [clear your browser cache](#)).

- 7) Click “Visit site” in the top right to visit your live website.



- 8) You can test this website to make sure it’s working by sending it to your phone or opening it in Incognito mode.
- 9) Experiment with ChatGPT to make edits to the page to make the content consistent with your startup idea. Don’t worry about getting something perfect; just get something that distinguishes your page as belonging to your startup.

Here’s a ChatGPT prompt and conversation to get you started.

ChatGPT: [Iterating on a startup landing page](#)

I’m building a startup landing page for a company called <name> that will be <describe your startup>.

Can you help me create the landing page?

Hints:

Hint 1: If you are updating your index.html file on GitHub, but the updates aren’t showing up in the browser, the answer is early in the conversation.

Hint 2: If your chat is getting unruly, try starting a new chat and explaining, copying, and pasting the relevant context in the new chat.

Hint 2: If you're lost, ask ChatGPT to explain, start smaller, and ask it for "all the code in one file."

Checkpoint: If this is your first time writing code - congratulations!

You've just built and deployed a fully working webpage to the public internet!

Submit your ChatGPT thread and public website here:

[Level 1: Basic Website Submission](#)

Submission: Level 1: Basic Website Responses

- Link to your landing page
-

Level 2: Enhancing Your Webpage (Group)

Enhance and polish interactive pages with advanced AI prompting and creative tools

Improving Your Website

Now that you have a full landing page accessible to potential customers, it's time to further develop your website.

If this is your first time coding, don't worry about getting something perfect; the goal is to get more practice using the tools and exploring what is possible.

Choose at least 2 to 3 enhancements for your website. We've provided a list of example enhancements in the appendix, but feel free to choose your own improvements:

See [Appendix: Example Level 2 Enhancements](#)

Hypothesis Testing & AI Demos

The best improvements will be one of two categories:

- Hypothesis Testing:** An enhancement that can move your experimentation with your idea forward (e.g., further testing a hypothesis about your customers, even if it is as simple as creating the fidelity or apparent credibility of your idea).
- AI Demo:** Something that you believe will add to the discussion in class about that "art of the possible" with generative AI.

You can do this as a group or individually.

Checkpoint: Submit brief answers to the discussion questions below before continuing:

[Level 2: Website Enhancement Submission](#)

Submission: Level 2: Website Enhancement Responses

- Link to your landing page
 - What enhancements did you make to your website?
 - How do those enhancements move your hypothesis testing forward or demonstrate the “art of possible” with generative AI?
 - What surprised you?
 - What worked well? What didn’t work well?
-

Level 3: An Interactive App

Leveraging AI App Builders

Leveraging AI App Builders

In our final level, we’ll use AI app builder(s) to create an interactive version of your MVP / MVT.

Your objective is to use one of the latest AI app builders to create an interactive application for your concept to understand the possibilities and limits of AI app builders today.

Make sure to read the questions at the end of this section before you start, then submit a link to your live application:

[Submission: Level 3: Interactive App Responses](#)

You are free to choose whichever AI app builders and tools you prefer.

We’ve provided more detailed instructions and guidance below for those looking for direction to get started.

You can do this as a group. We highly recommend working together in person to do this. Students have learned a lot from seeing how peers approach the same prompting tasks differently in the past.

App Builders

We recommend trying one or more of the following app builders:

- v0.dev
- bolt.new
- loveable.dev
- replit.com
- bubble.io

Students with more technical experience can also use other tools, including [Cursor](#), [Windsurf](#) by Codeium, [Copilot](#)¹⁰ for VSCode.

Suggested Path to an MVP / MVT

Step 0: Clarifying the outcome

Clearly define the hypothesis you want to test and the outcome you want to achieve.

This can either be an idea that one of the group members is working on or one of the example apps ([Appendix: Example Level 3 Projects](#)).

The goal here is to explore the possibilities and limitations of app builders available today. The gap between your stated and eventual outcome is less important. Your hypothesis can be as simple as testing the feasibility of building part of your idea.

Step 1: Defining product requirements

In this step, we'll develop the initial prompts to guide the app builder on the requirements for our application.

Web applications are often broken down into three separate parts:

- a) **Frontend** - The HTML, CSS, and JavaScript¹¹ code that runs in the user's browser.
- b) **Backend** - The code that runs on a server to communicate between the browser code and database.
- c) **Database** - The part of the application running on the server that stores data that persists in the app.

We'll break down our application into these parts for the app builder. We recommend creating a shared Google Doc with your group to keep track of these prompts:

¹⁰ You can sign up for [free Copilot access via Github Edu](#).

¹¹ Traditionally, HTML provides content, CSS provides styling, and JavaScript provides interactivity.

- 1) **Frontend** - Describe the main pages the user will see and the key elements on each page.
- 2) **Backend** - For each front-end page, identify the actions a user can take. (e.g. "recipe: type in ingredients and then get a response").
 - a) For each action, define:
 - i) **Description:** What happens when the action is taken?
 - ii) **Input:** What data is provided by the user or the system to initiate the action?
 - iii) **Output:** What is the result of the action (if any)?
- 3) **Database:** Consider what data needs to be stored so that if any page were reloaded, the relevant information could be displayed again.

List the data fields that would need to be stored.

Hints:

Hint 1: Try using ChatGPT to help you improve your product specifications above.

Hint 2: You can look at [Appendix: Example Level 3 Projects](#) for some examples of initial, high-level prompts to get started, but these should be fleshed out further.

Step 2: Creating an Initial App Builder Prompt

Ask ChatGPT to generate a series of prompts to give your chosen app builder based on the information above.

Here's an example prompt to get you started, but feel free to adjust it or use a completely different one:

We are using [v0.dev/bolt.new/Lovable.dev/replit.com], an AI-native developer tool that takes a prompt and turns it into code, in order to build our product. Your goal is to output a series of prompts that, when given to this tool, create the product. Remember, the tool uses AI to create the product, so sequence your prompts accordingly.

The product idea is as follows: [YOUR IDEA]. Here is a non-exhaustive description of the specific pages, backend actions, and data stored in our database to build the application. If there is anything else needed to build the product, include it. Specific pages include [YOUR FRONTEND SPEC], Backend Actions [YOUR BACKEND SPEC], Data stored in our database [YOUR DATABASE SPEC].

Step 3: Iterate!

Using your initial code, prompt your AI app builder to improve your application.

The sky is the limit. Based on our run-through of this assignment, we suggest doing at least several iterations and stretching yourself to push the limits of the tools.

Make sure you have at least one live app by the end of Step 3, but feel free to experiment with multiple tools and submit multiple URLs.

Make sure to submit a link to your application(s) along with answers to the response questions below.

Hint: For inspiration and some techniques, see this example on YouTube: [Riley Brown Builds 6 Versions of The Same App \(No Coding Required\)](#).

Checkpoint: Submit brief answers to the discussion questions below:

[Level 3: An Interactive App](#)

Submission: Level 3: Interactive App Responses

- Link(s) to your web app(s)
- What hypothesis did you test and/or outcome did you set out to achieve?
- What app builder(s) did you choose?
- What strateg(ies) did you use to prompt the app builder(s)? Give some example prompts.
- What surprised you?
- What worked well? What didn't work well?
- How does this move forward your experimentation/hypothesis testing?

Office Hours

We'll hold office hours for those looking for tips, tricks, help, or who are generally interested in other ways to leverage ChatGPT and other AI tools to get things done.

[John YS](#) (+ TBD) will host an open Google Meet Office Hours.

Check the Google spreadsheet for updates. Tentative times are below, but we're likely to update them depending on demand. If no one is signed up for a slot, 1 hour before, then it will be canceled.

[2024-10 HBS LTV AI Workshop Office Hours Sign Up](#)

Appendix

Example Level 3 Projects

Note: We *HIGHLY* recommend coming up with your own concept to test and just using these as inspiration.

An AI Chat App: This app could allow users to chat with a bot similar to ChatGPT's web interface.

- The "Frontend" would provide input to converse with the bot and display responses.
- The "Backend" would handle connecting with an LLM API.
- The "Database" would store the context of conversations and user preferences.

A Recipe Recommendation App: This app could allow users to input ingredients they have on hand and receive recipe suggestions.

- The "Frontend" would provide input forms for ingredients and display the recipes.
- The "Backend" would handle the recipe lookups or generation.
- The "Database" would store recipe information (even a small, curated set for an MVP).

An LTV Flashcard App: Users could create or select sets of flashcards for learning LTV terms and acronyms.

- The "Frontend" would display the cards and handle user interaction (e.g., flipping the card, marking as correct/incorrect).
- The "Backend" would manage the sets and potentially the review order.
- The "Database" would store the flashcard content and user progress (if tracking is included).

A Simple Event RSVP Tool: Users could create a basic event and invite others to RSVP.

- The "Frontend" would have forms for event creation and a way for invitees to RSVP.
- The "Backend" would handle the event creation, invitation management, and response recording.
- The "Database" would store event details and attendee responses.

A Minimalist Budget Tracker: This app would allow users to input their income and expenses to get a basic overview of their spending.

- The "Frontend" would have input forms and display the budget summary.
- The "Backend" would perform the calculations.
- The "Database" would store the income and expense data for each user.

A Media Player App: This app would allow users to play, list, and save videos or songs from a service such as YouTube, Spotify, or SoundCloud.

- The "Frontend" would have input to search for and display media.
 - The "Backend" would connect to the YouTube, Spotify, or SoundCloud API.
 - The "Database" would store lists and preferences of the user.
-

Example Level 2 Website Enhancements

Here are a few ideas on how to get started improving your landing page.

Basic

Text Content: Refine the content or sections of your site by asking ChatGPT to edit the HTML or asking it how to do it manually yourself in tandem.

Styling in CSS and JavaScript: Try adding more sophisticated styling to your website or some interactivity with JavaScript animations. Ask ChatGPT to rearrange and modify elements on the page to your liking. Consider providing a screenshot or a link to a site you want it to emulate.

Links: Ensure your links function correctly, directing users to external websites or internal pages within your site. (see: "Multiple Pages" below).

Embed a Form: Try embedding a form (e.g. Google Forms, Typeform, or another service) to collect leads.

Images and Video: Use ChatGPT to learn how to link to asset files like images and video. Create some new images to use on the site by asking ChatGPT.¹²

Favicon: Change the small image you see on the browser tab by asking ChatGPT to generate and walk you through adding a favicon.

Multiple Pages: Add multiple pages to your website by asking ChatGPT how to create the pages, upload them to GitHub's web interface, and link them properly in index.html.

Different Personas: See how ChatGPT responds differently when you change the persona: e.g. "You are a... frontend web developer / expert designer / business person" and ask it the same prompts.

¹² ChatGPT: [How can I copy the raw link for a video I've uploaded to github?](#)

Add Analytics: Create a Free Google Analytics and/or FullyStory account and have ChatGPT walk you through setting up real-time tracking and user session recording.

Intermediate / Semi-Technical

Make a Promotional Video: Create a promotional video for your startup using a tool like [HeyGen](#), [Hedra](#), or [Runway](#). You can build the script in ChatGPT.¹³ Include it on your site.¹⁴

Make a Jingle: Create a “jingle” song about your startup with a tool like [Suno](#). Hint: You can use ChatGPT to help write the script. Add it to a section of your site.

“Great Artists Steal” - Experiment with how quickly you can iterate on your website by pulling example code from screenshots or inspect element on other website references.

Learn Git with ChatGPT - Use ChatGPT to learn how to use the command and Git to manage code like a professional software engineer.

Try Out Another AI Tool - Try using another LLM like [Anthropic's Claude AI](#), web development tools like [Create.xyz](#), or an LLM business model generator like [ChatAgency](#) and be prepared to share the results with the class.

Technical

Go to Town on Your Landing Page - See how polished and valuable you can make your landing page to advance your testing. Try out using/learning a new technology with ChatGPT.

Use Another AI Tool - Try using another AI tool like [Aider](#) to develop your submission.

Build a Utility Tool - Build something to help move forward your workflow or product market fit. E.g. a scraper for public data, a command line program to download videos, transcribe, and summarize content in an area you're researching, etc. Bash scripts are a great use case. Record a video and link it on your site.

Class Session Plan

- **Intro (10 mins):**
 - Brief overview and framing of class objectives.
 - Explain expectations for demos.
- **Student Demos/Showcase (50 mins)**
 - Several groups to demo (~5 mins per group).

¹³ Tip: You can generate a script with different tones by referencing an ad studio, director, or movie. E.g. Write a pitch script like Aaron Sorkin, Stephen Spielberg, or [Sandwich Video](#).

¹⁴ ChatGPT: [How can I copy the raw link for a video I've uploaded to github?](#)

- Prompts for demos:
 - What did you build?
 - What were you aiming to achieve?
 - What surprised you?
 - What worked well? What didn't?
 - Key learnings?
- **Group Q&A / Discussion (~15 mins)**
 - Facilitate questions, insights, and peer feedback.
- **Wrap-Up (5-10 mins)**
 - Key takeaways and student reflections

Additional Resources

Prompting Resources

- [Lovable's Prompting Guide](#)
- [DeepLearning: Prompt Engineering for Developers](#)

More About LLMs

- [Deep Dive into LLMs like ChatGPT by Andrej Karpathy](#) - Great layperson's deep dive into how LLMs work. Worth going through the 3 hours to build an intuition for what is possible with the current state of technology today.
- [The Identity Crisis of Software Engineers \(In the Age of AI\)](#) - Software engineering is becoming more like management (and maybe vice versa).

Inspiration from Last Semester

- [Recap of 1st HBS AI Workshop](#)
- [Student Showcase from Fall 2024](#)

Credits

Authored for [Professor Jeff Busgang's Launching Technology Ventures \(LTV\) Spring 2025 class](#).

Thank you to [Ali Ahmed](#), [Irtaza Ali](#), [Nina Prabhu](#), [Eric Schwartz](#), and [Annabel Strauss](#) for their contributions to the course content.