

# Building AI-Generated Apps with ChatGPT

(aka "Fun with ChatGPT")

## Overview

This workshop provides an opportunity for you to experiment with using ChatGPT to create an application. The applications will rely on Azure Storage static site's capability to host the relevant HTML, JavaScript, and CSS files. **The activity is designed to work through with a group of 5-10 people as a team, however you can also complete the activity independently.**

In this activity you will:

- Use ChatGPT to produce product ideas
- Use ChatGPT to produce product names for your ideas
- Use ChatGPT to help you setup an Azure storage account to host a static web app
- Use ChatGPT to help you author a marketing/landing page for your app
- Use ChatGPT to help you author your static web app
  - Optionally use ChatGPT to help you make calls to free public APIs to add rich functionality to your app
- Use ChatGPT to help you write a commercial jingle for your app
- Share your human-designed/AI-generated app prototype with others

*NOTE: This activity is not compulsory. If you are not comfortable creating or using your account at OpenAI or providing information by typing prompts into ChatGPT, you are completely free to opt-out of this activity entirely and/or play the role of observer on your team and watch the outcomes of the activity. You can review the [ChatGPT FAQ](#), [Privacy Policy](#), and [Terms of Use](#) on the OpenAI website.*

## Task 0: Use ChatGPT to produce some app ideas

As a team **come up with 5-10 ideas of mini-apps that you might consider building** – whether those are entirely self-contained or will need to make external requests. **You might consider integrating with [free public APIs](#) to add some more capabilities to your apps.**

**Don't bite off more than you can chew.** Make sure that it is a reasonably sized chunk of work to consume no more than 30-45 mins of AI-assisted effort. Examples might be a simple in-memory to-do list, timer app, or even an app that tells you what time you need to leave home for a given

flight based on browser location, flight data, and traffic data. The prompts displayed in the steps below produce [this app](#) – an app to track how much money you are making as time passes. You can see more sample apps on [this page](#). **The goal of this workshop is to be able to have a working prototype or at the very least a marketing page at the end.**

If you are having trouble coming up with ideas, choose one person on your team to head over to <https://chat.openai.com/chat>, and login with your personal Microsoft account (e.g., outlook.com, hotmail.com, live.com).

**Ask ChatGPT for ideas for the app.** You might consider providing very clear parameters (e.g., limiting the complexity of the app to something you feel that your team *could* put together within an hour even without the assistance of AI).



I'm trying to build a small application that will be hosted as a static html page with supporting css and javascript alongside. I don't want to write any server-side code. I am willing to call into third-party http-based APIs if they provide useful data or functionality that might be integrated into my app (e.g., weather data, traffic data). Can you provide a list of 5-15 creative app ideas that could be run directly from the browser with no server-side data storage requirement, that might combine the power of multiple 3rd party http-based APIs to provide enhanced capabilities? For each provide a sample name of the app and a brief description of its functionality. The ideas must be low complexity and something that could be built within 30 minutes.



Sure, here are 7 creative app ideas that could be built within 30 minutes, combining the power of multiple third-party APIs:

1. News Aggregator: An app that pulls the latest news articles from multiple sources (e.g. BBC, CNN, Reuters, NPR) and displays them in a single feed. Name: *Headline Hound*.
2. Currency Converter: An app that uses the latest currency conversion rates from a third-party API (e.g. Open Exchange Rates) to convert between different currencies. Name: *Currency Cruncher*.
3. Weather Tracker: An app that displays the current weather conditions and forecast for a specific location using a weather API (e.g. OpenWeatherMap). Name: *Weather Whiz*.
4. Recipe Finder: An app that allows users to search for recipes by ingredients, cuisine, and meal type, using a recipe API (e.g. Edamam). Name: *Recipe Rover*.
5. Movie Recommendation: An app that suggests movies to watch based on a user's preferences (e.g. genre, actor, director) using a movie API (e.g. IMDb). Name: *Movie Maven*.
6. Book Search: An app that allows users to search for books by author, title, or keyword using a book API (e.g. Google Books). Name: *Book Browser*.
7. Stock Tracker: An app that displays real-time stock prices and historical charts for a user's favorite stocks using a stock market API (e.g. Alpha Vantage). Name: *Stock Sleuth*.

If the ideas are too complex, or boring, just say that to ChatGPT – e.g., “Those are too complex and boring. Give me more creative things like an app that shows an animated fire or rainstorm to help me relax.”

Once you have settled on an idea, ask ChatGPT for a list of features that might need to be built for the app. Choose 1-2 of those to deliver as part of the activity.

## Task 1: Setup a place to host your app, and write a script to publish your app

Once your team has decided on a relatively low-complexity idea to pursue:

- **Choose one member of your team** to setup a place to host the app. Alternatively, if you'd like to pursue multiple ideas as a team and then choose 1-2 at the end to share with the group, feel free to have multiple members of your team setup a place to host the app.
  - **Have the rest of the team start ideating on product names as described in the next section.**
1. Create a new Azure Storage account.
  2. Follow the instructions at <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blob-static-website-how-to?tabs=azure-portal#enable-static-website-hosting> to enable static website for the storage account
  3. When your team is ready, follow the instructions at <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blob-static-website-how-to?tabs=azure-portal#upload-files> to upload files (HTML, Javascript, CSS, etc...) to the storage account
  4. You can then find the address to access your site by following the instructions at: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blob-static-website-how-to?tabs=azure-portal#find-the-website-url>

## Task 2: Use ChatGPT to produce some product names

Describe your app(s) to ChatGPT and have it produce some relevant product names. Pick a name as a team and massage the marketing copy. Make sure these are communicated to the folks that have been setting up a storage account for the content.



I want to build an application that will be a twist on a subset of the idea you gave for a Daily Planner. It will help folks track how much money they earn while completing certain tasks. It will serve dual purposes of encouraging folks to continue working on painful tasks with knowledge they're being paid to complete them, and will also help them if they later automate the same tasks to express the cost savings provided to the business through their automation efforts. It won't actually help anyone PLAN anything per se, but it will help them make it through their days. Can you give me 10-15 ideas of a name I could use for this app and then write a paragraph of marketing copy that sounds incredibly compelling?



Sure, here are some names for your app:



1. Earn Track
2. Paycheck Planner
3. WorkWise
4. Task Tracker Pro
5. Money Motivator
6. Labor Ledger
7. Time and Treasure
8. Paycheck Progress
9. Productive Pay
10. Task Tally
11. Effortless Earnings
12. Daily Dividends
13. Money Map
14. Labor Log
15. Time and Treasure Tracker

Marketing copy:

"Unlock your earning potential with [insert app name here], the innovative app that helps you track and quantify the financial value of every task you complete. Say goodbye to the

### Task 3: Use ChatGPT to write the relevant html, JavaScript, and CSS

Have some members of your team start with the landing page, and have the rest move on to the next step. Ask ChatGPT to help you create a landing page for the app that will link to the actual app. If the code does not work, or if corrections are needed. Prompt ChatGPT to correct the code.

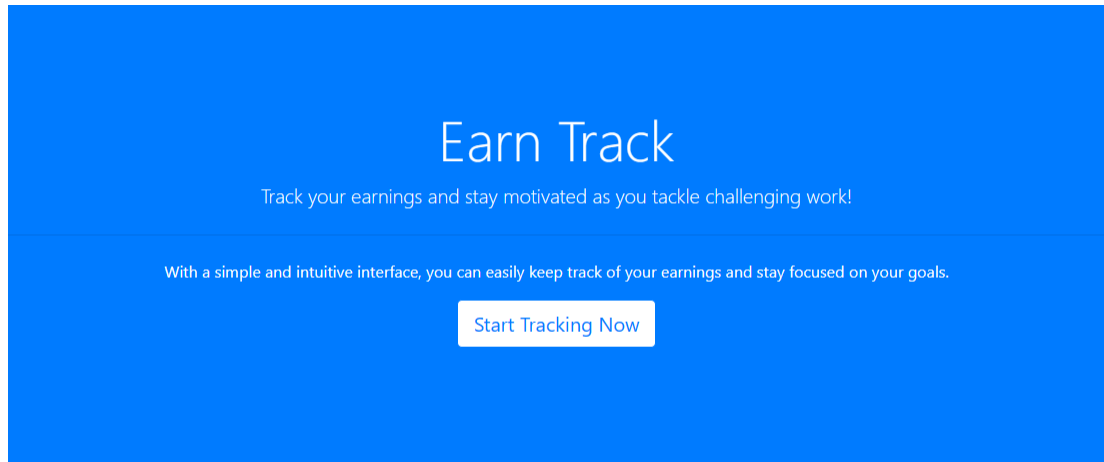


Can you provide me html + css for a compelling landing page for the app. I will be storing it in a file called index.html. Have it link to app.html which will have the actual app itself. Use Bootstrap for the style, and include a giant call to action button that says "Start Tracking Now" which links to the app.html page. Don't include any links to any other pages -- I don't want to commit to having to build those as well.

Sure, here's the HTML + CSS code for the landing page:

```
html
Copy code
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<meta http-equiv="X-UA-Compatible" content="ie=edge" />
<link
rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWVTH0E263XmFcJ1SAwiGgFAM/dA3"
crossorigin="anonymous"
/>
<title>Earn Track</title>
<style>
.jumbotron {
background-color: #007bff;
color: white;
text-align: center;
padding: 20px 0 40px 0;
}
```

Store the code in index.html and then run the steps to publish it to the storage account. Verify the published result.



Ask ChatGPT to write the application itself (separating out the markup, styles, and JavaScript). You might include in your prompt that you will ask for one file at a time by name, and then proceed to do so – asking for each file as needed rather than all at once. Sample prompt is included below for building the whole app at once.

I'm trying to build an application that shows a title, labeled text box, two buttons, and giant section labeled "Total Earned".

I want the text box labeled "Hourly Pay" with a default value of 10.  
I want the first button to be labeled "Start Tracking".  
I want the second button to be labeled "Stop Tracking"

When I click Start Tracking, I want to start a timer. While the timer is running, the Total Earned section should be updated with the amount of money earned each second assuming that the hourly pay box value contains the amount of money earned over the course of an hour.

When I click Stop Tracking, I want the timer to stop, and the total earned section to turn green.

I want the application to be colorful and use Bootstrap for styles as much as possible.

I need you to help provide HTML, JS, and CSS to create the following files:  
- a file named app.html with application markup  
- a file named app.js with application code  
- a file named app.css with any needed styles beyond whatever might exist in Bootstrap.



I'm trying to build an application that shows a title, labeled text box, two buttons, and giant section labeled "Total Earned".

I want the text box labeled "Hourly Pay" with a default value of 10.

I want the first button to be labeled "Start Tracking".

I want the second button to be labeled "Stop Tracking"

When I click Start Tracking, I want to start a timer. While the timer is running, the Total Earned section should updated with the amount of money earned each second assuming that the hourly pay box value contains the amount of money earned over the course of an hour.

When I click Stop Tracking, I want the timer to stop, and the total earned section to turn green.

I want the application to be colorful, and use Bootstrap for styles as much as possible.

I need you to help provide HTML, JS, and CSS to create the following files:

- a file named app.html with application markup

- a file named app.js with application code

- a file named app.css with any needed styles beyond whatever might exist in Bootstrap.



Sure, here is the HTML, JS and CSS code for the app:



app.html:

```
html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
```

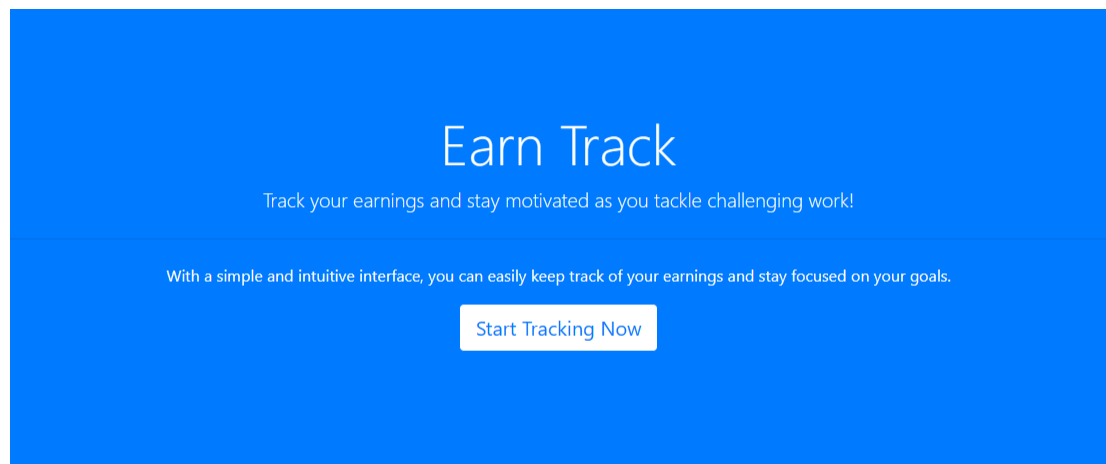
ChatGPT Jan 30 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

## Tips:

- If the code is cut-off at any point, you can simply prompt with **continue** to continue the code listing.
- If you're calling a 3<sup>rd</sup>-party API, include a recent page of documentation from that API inline with your request to generate code. The API could have changed since the training of the model was done in 2021, and ChatGPT is capable of learning in-context how to complete a given task from the prompts themselves.
- If you ask for corrections, specify the specific element on the page to output (e.g., please output only the modified style tag contents)

- Watch for mismatches in element names, this is a common mistake when asking only for partial output after an especially long conversation that goes beyond the limits of ChatGPT (e.g., a div previously named “marketing-blurb” in one output might be referenced in a later output as “marketing-copy”)
- If the page errors out entirely and is no longer responsive, copy your last prompt, reload the page, and then send the prompt again.

Store the code in the relevant files, and then run the steps to publish it to the storage account. Verify the published result. If it doesn't work, share the behavior you observe with ChatGPT. If you have any modifications, ask ChatGPT for them (e.g., change the color, reposition elements, change the behavior). Also be prepared to make changes manually to adjust it to the desired output.



## Earn Track

Hourly Pay:

**Total Earned: \$3056.40**

### Task 4: Use ChatGPT to write a commercial jingle for your app

Once your team has completed a number of apps (even if that number is 1), ask ChatGPT to write a commercial jingle for your favorite app. Then celebrate the trillions of dollars of revenue that could result from your team's efforts.



Can you write me a short catchy commercial jingle for this app?



"Track your pay, watch it grow, with the app that helps you know. Earn more every hour, with just a click of a button, the app that makes your pay grow, it's time to start tracking now!"

Share your commercial jingle and URL to your team's favorite app(s) so that all can enjoy! If you happen to share on social media, use the hashtag #AppByChatGPT so that any folks who have been through this workshop can find it.

## Task 5: Use another instance of ChatGPT, or another SLM/LLM to review the code

Once you have any portion of code completed, have some members of your team run that code through a new instance of ChatGPT or another SLM/LLM for review. Prompt it to review the code with an eye towards maximizing quality, performance, security, etc., share the findings with the team, and if time permits, pursue fixes for the same.

## Summary

Hopefully, you have at least a landing page and jingle for your app so that customers can start getting excited about your concept. Maybe you've made it further and have a working prototype ready to go. If so, congratulations! Either way, you have some more experience at this point with this technology, have seen how the prompts that you provide still must be engineered in a way to optimize the outputs. You've seen the conversational abilities of ChatGPT and ability to be corrected and adapt, alongside its ability to multitask to completely different areas (e.g., generating songs vs. generating code).

Thanks for joining this activity, take care, and go build great things!