AI-GENERATED PHYSICS SIMULATIONS
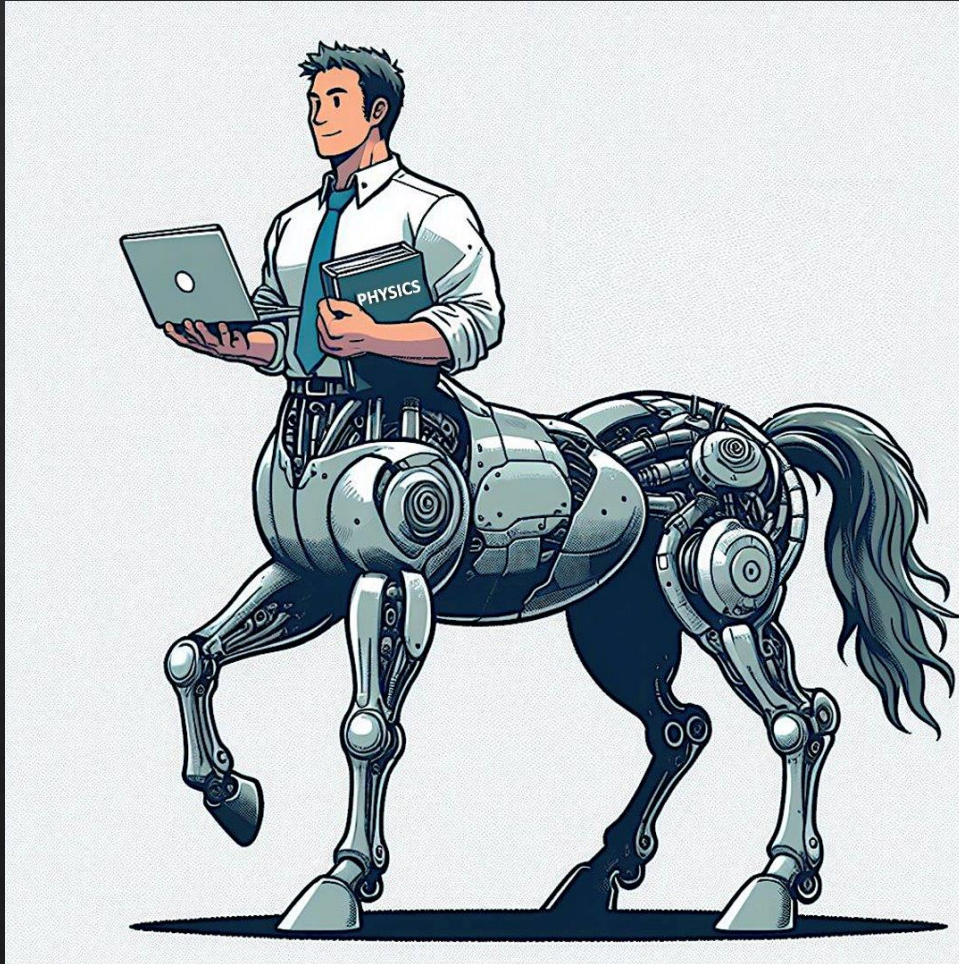
EAST ZONE PHYSICS CHAPTER EDTECH WORKSHOP

tan_seng_kwang@moe.edu.sg

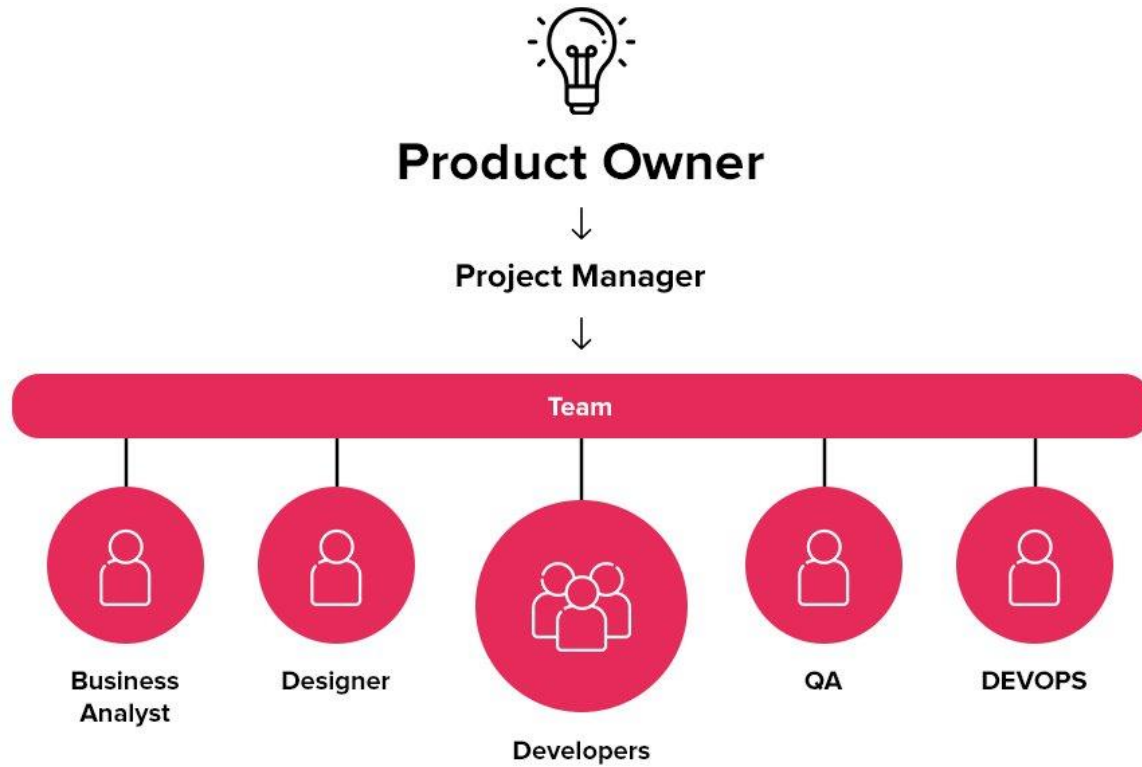PhysicsLens.com

TEMASEK
JUNIOR COLLEGE

BECOME A CENTAUR

While experts will always be in demand, more intelligent machines are continually lowering the bar to creating with new technology.
Kasparov, 2017

Lower barrier of entry for app development; allow for focus on higher levels of problem solving
Bull and Kharrufa, 2023

Replacing the Programmer?

No more coding classes? Nvidia's CEO believes AI is enough (Source – Shutterstock)

AI-Generated Apps for Education

Repository:

for.edu.sg/apps

https://for.edu.sg/apps

About the Repository

- Ready for Use

  - Can use directly on browser (online or offline) or

  - Upload the zipped file and embed into SLS:

  https://www.learning.moe.edu.sg/teacher-user-guide/author/html5-content-development/#uploading-a-html5-zip-file-in-sls

- Prompts used are shown for each app

Examples: Use of Sensors for Experiments

Sound Level Meter
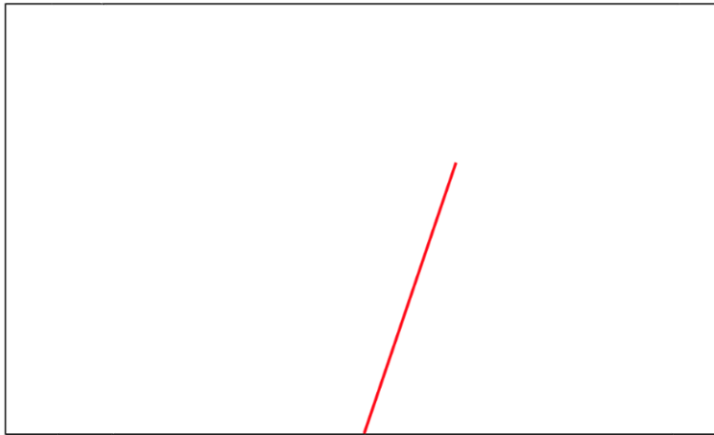
Range:

Sound Frequency Analyser
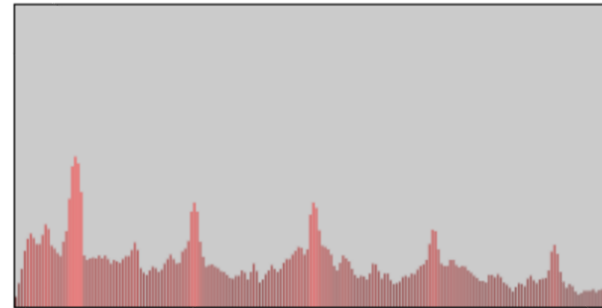
Dominant Frequency: 430.66 Hz

Acoustic Stopwatch

Threshold: 150

Minimum Delay (ms): 1000

Start

Time: 1.502 seconds

Camera Brightness Recorder

Start Sampling   Stop Sampling

Sampling Rate (seconds): 1 seconds

Relative Brightness: 0.30

Download CSV   Reset Data

- 21:29:16: 87.00
- 21:29:17: 94.40
- 21:29:18: 75.52
- 21:29:19: 89.34

An example in the SLS Community Gallery

https://vle.learning.moe.edu.sg/community-gallery/lesson/view/f9c45429-0261-4999-aaeb-f8ded90fe8eb/page/73124376

Examples: Interactive Simulations

An example in the SLS Community Gallery
https://vle.learning.moe.edu.sg/community-gallery/lesson/view/b582192d-1d30-488a-a846-93c34d69da85/cover
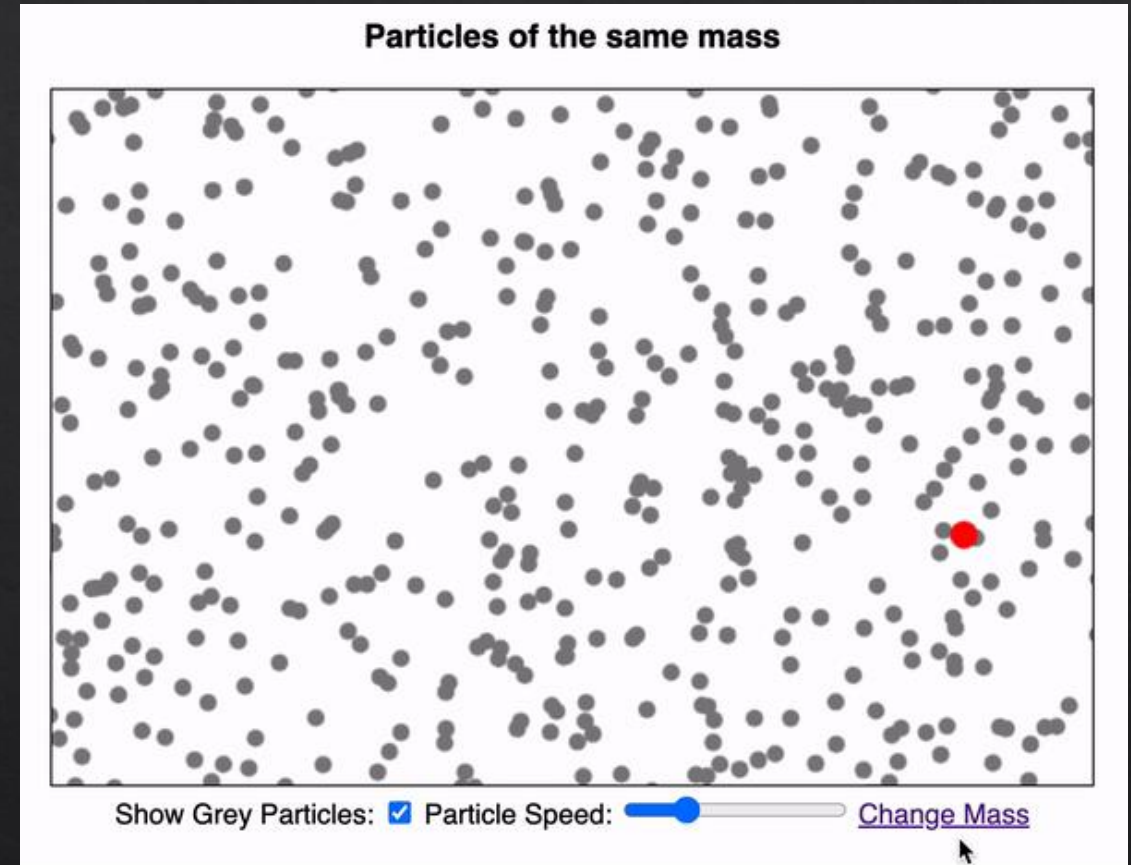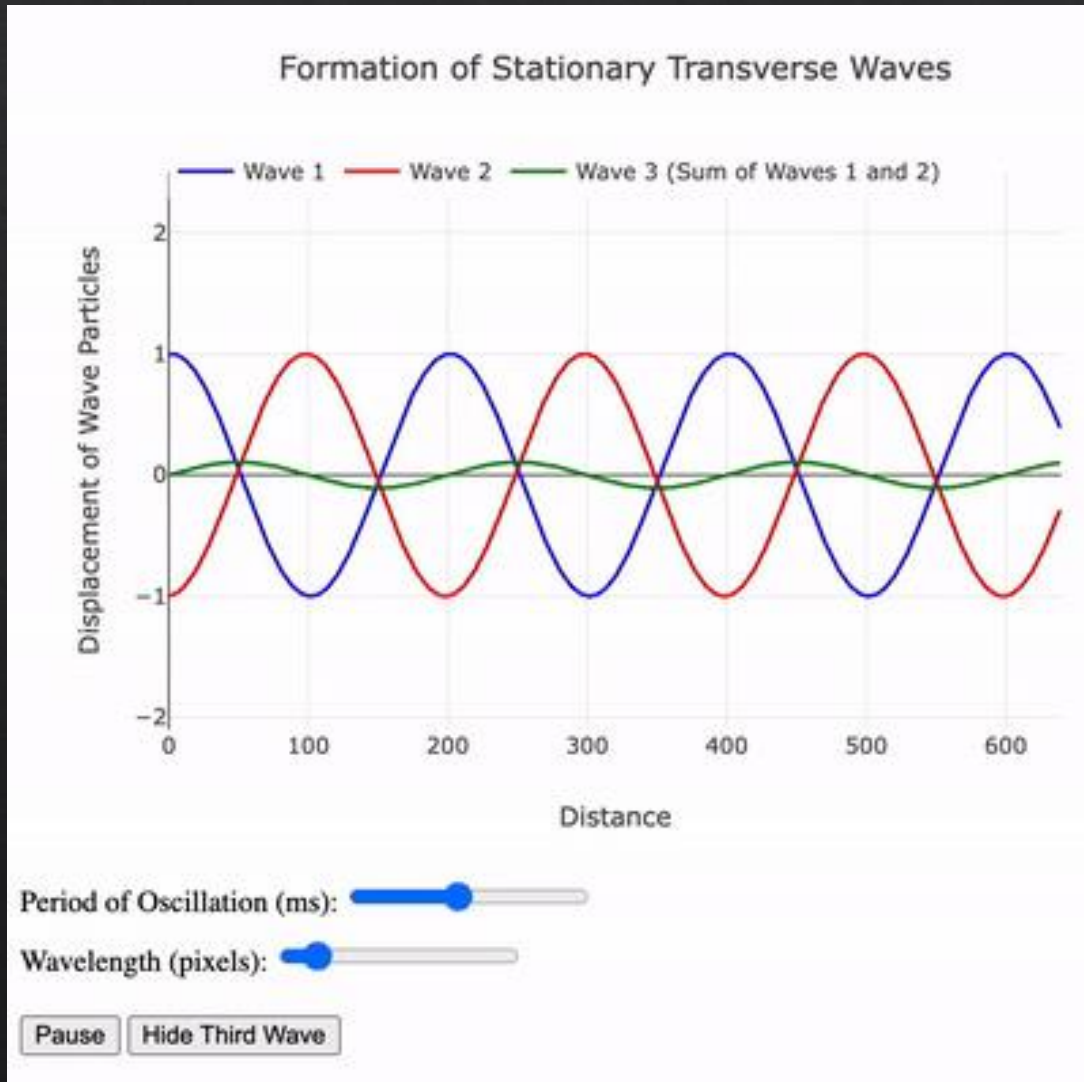
Examples: Interactive Simulations
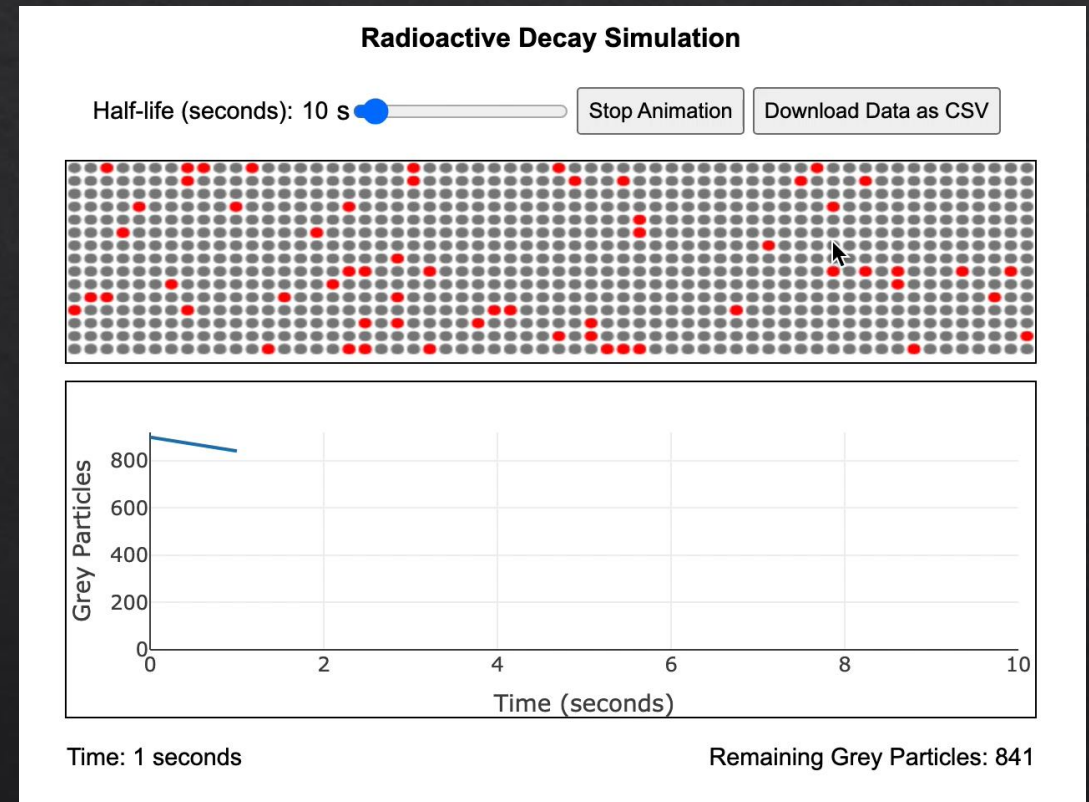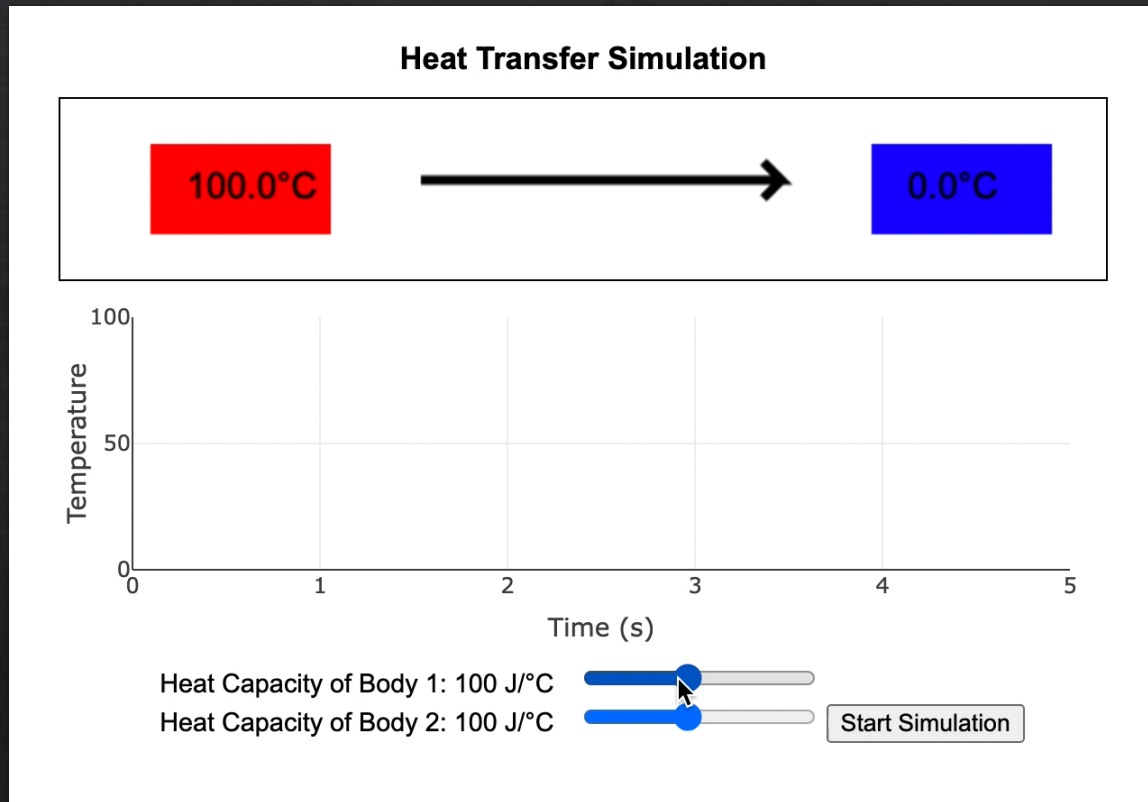
Workflow
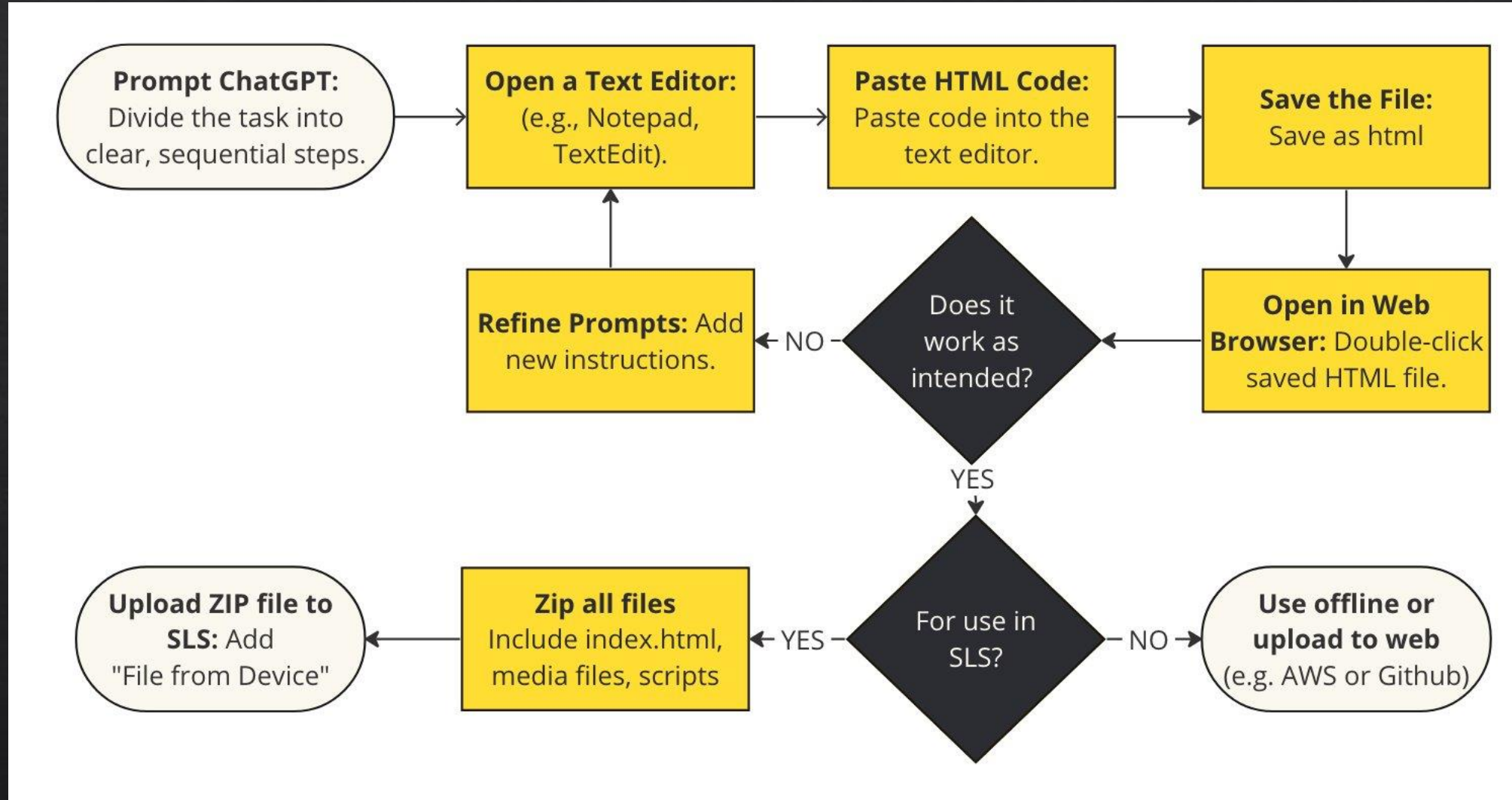
Basics of Web Apps/Sites

- Three languages in use (html, css. javascript)
  - For large sites, these codes are usually placed in separate files for universal use
  - To simplify workflow, we will put all 3 sets of codes in one file.
- Alternative is to code in python for https://www.glowscript.org
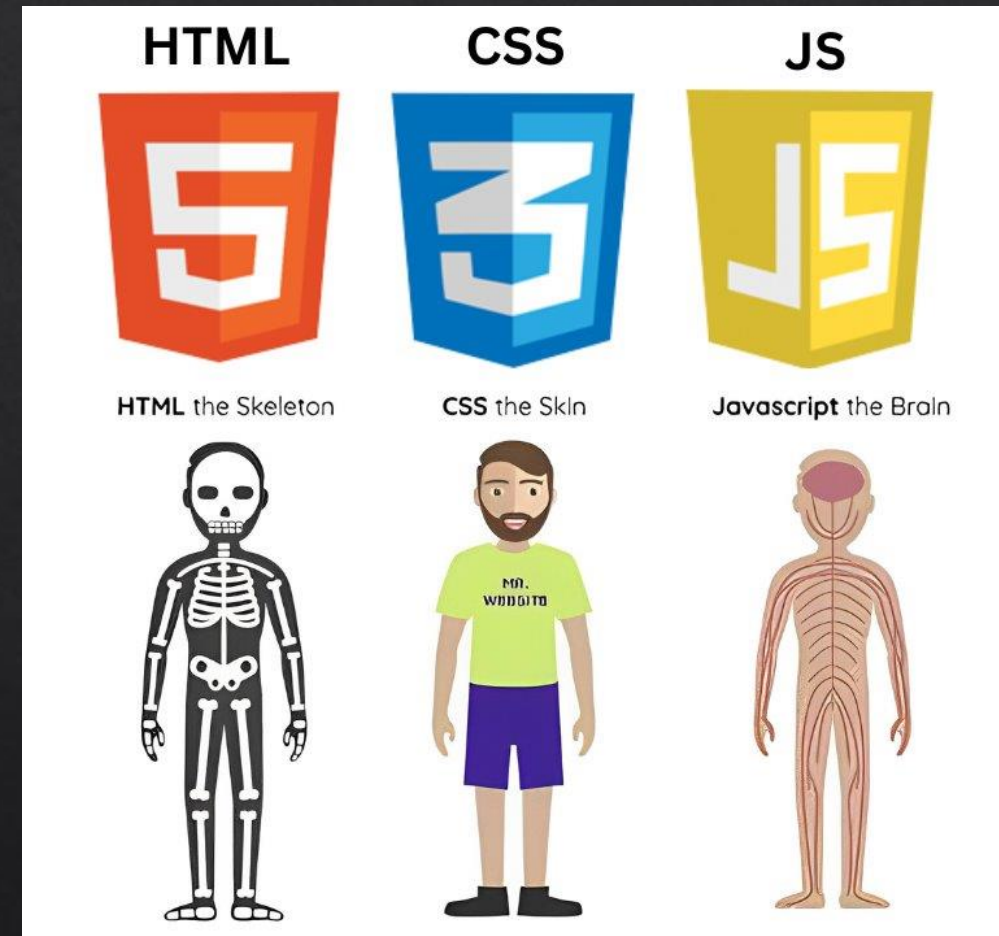
Limits of standalone apps in SLS

1. No connection to online databases. However, data can be stored in browser.

2. No transfer of information between SLS and app.
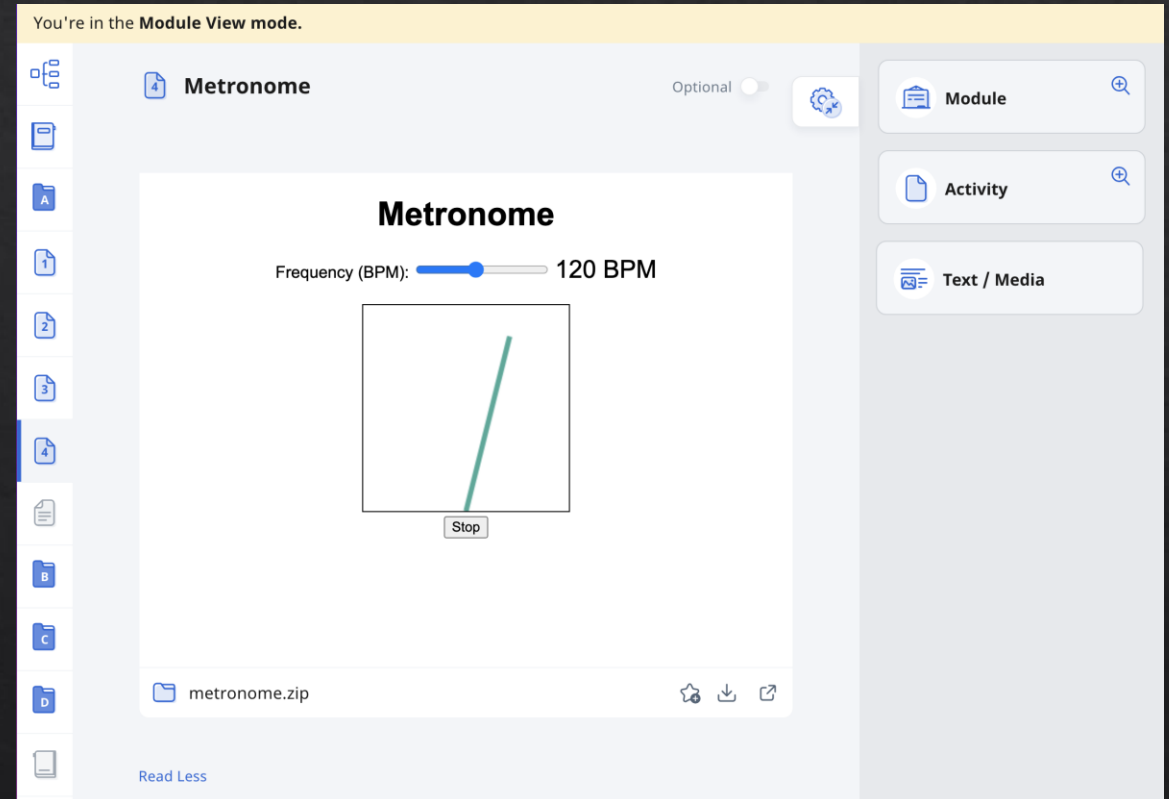
Prompt Engineering Tips

1. Start by asking for the html and javascript codes in a single file.

2. Describe what the app is supposed to show. Do you need a canvas (for graphics) or a graph, or both?

3. Describe the interactions in the app. Use elements such as buttons, sliders or input boxes

4. (If necessary) Give the names of image/video files to be used and put them in the same directory as the html file.

5. (If optimizing for SLS), max width: 580px, height: 460px

6. Finally, change the stylesheet to make the UI more appealing

7. Ask AI to comment or explain the codes if it did not do so

Hands-on: Our Aim
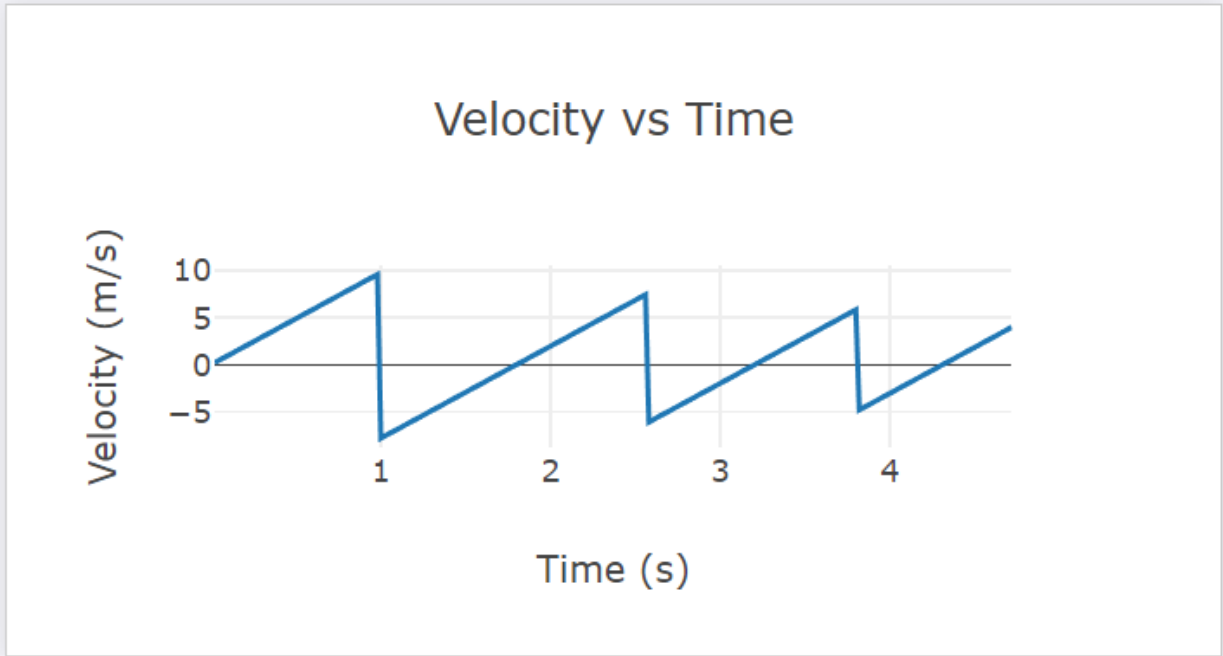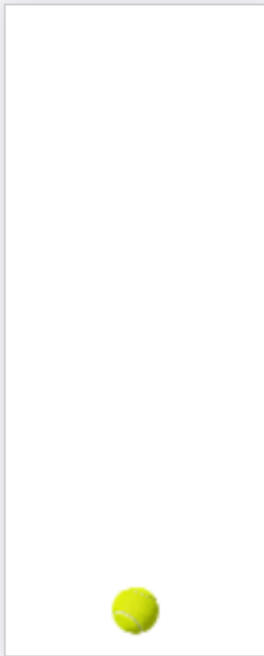
Hands-on: Prompt Sample

STEP 1:

Open ChatGPT or any other GAI (e.g. Claude, CoPilot, Gemini)

STEP 2 (PASTE THIS INTO THE AI):

"Put all the codes in one page.

Create a canvas showing a ball dropped from rest from a height and bouncing off the ground using javascript.

Using the plotly library, plot the graph of velocity versus time for the ball. The time of contact with the ground is negligible.

Create an input box that allows the user to key in the initial height in metres.

Create a slider that changes the percentage energy loss after every collision with the ground

Create a dropdown menu that changes the vertical axis to velocity or displacement.

Initialise the animation and graph upon loading. Use a button to start and stop the animation."

Hands-on: Testing

STEP 3:

Copy the generated code

STEP 4:

Paste into editor at https://for.edu.sg/html

"Run in New Tab" to view

https://for.edu.sg/html

Hands-on: Debugging

STEP 5:

Be prepared to generate 10 or more versions! Repeat STEPS 3-4.

Options:

1. Change the code manually yourself
2. Describe any unexpected behaviour / missing component to AI
3. Ask AI to try a new approach (usually after a few failed iterations)
4. To save time, just ask AI to generate the codes that need to be changed. It will tell you where to update

Hands-on: Make the App Look Pretty!

STEP 6:

1. Optimise for SLS by asking AI to "limit the entire page to a size of max width 580px and max height: 460px"

2. Ask AI to beautify the app with styling

3. Ask AI to add image / video / audio files into the code, giving it the filenames. Save the media files in the same root directory as the index.html file.

Hands-on: Using Code Libraries

To embed into SLS, you will need to copy the code libraries that are used (if any) in a new .js file. The code library is a collection of pre-written code that you can use to perform specific tasks, e.g.

- plotly.js for continously changing graphs

- chart.js for static charts

- papaparse.js for processing csv data

7.1 Paste the link to the script on browser e.g. https://cdn.plot.ly/plotly-latest.min.js

7.2 Save the page using the "Save as" in the same directory as the html file.

7.3 Rename the path to the file:

Change

```
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
<style>
```

to

```
<script src="plotly-latest.min.js"></script>
<style>
```

Hands-on: Packaging for SLS

STEP 8:

Download as HTML.
Save file as index.html.

STEP 9:

Zip all the files and upload the zip file into SLS component.

Share your work with the community!

⬥ Host it online, e.g. AWS, Github, or

⬥ Include in SLS package and submit to Community Gallery or

⬥ Share with me via https://forms.gle/a4XuvrdhsP1WNvyP6