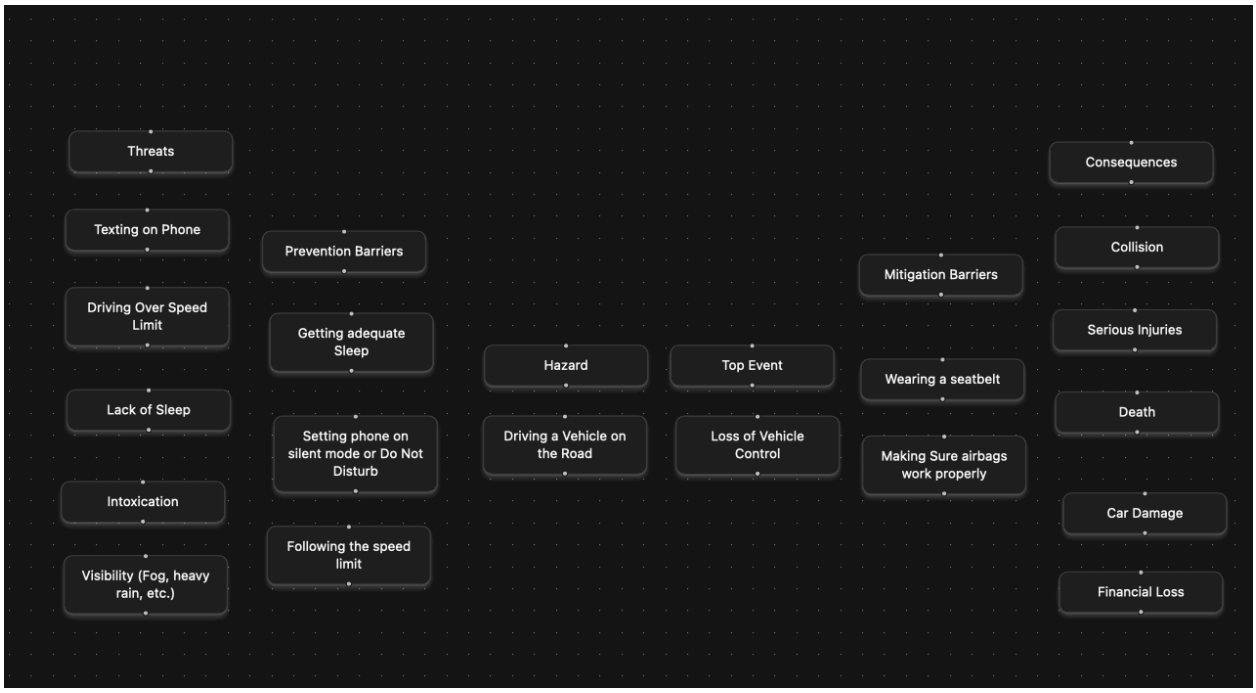
Bowtie Visualization Attempt Log

Reactflow ended up being very difficult to create diagrams with. Initially we thought that we were able to create the entire bowtie visualization in Reactflow Playground but that turned out to not be the case for us. We did some researching after creating our initial nodes and found out that Reactflow Playground has very limited functionality when the user is on the free version, this includes the expansion and collapsing of nodes. Documentation.

Our initial exploration of Reactflow was pretty simple. We were able to create nodes and had a rough layout of what we wanted the final diagram to look like:



We spent a lot of time researching and trying to figure out how to make the nodes interactable, and how to import custom images or change the color of the nodes but were unable to figure out how to do that in Playground even through the use of ChatGPT. ChatGPT seemed to always confuse the layout of the Reactflow Playground and when asked to recreate a simple expand/collapse node in here it responded with code (something that we assumed to be unnecessary for this project) to paste.

Perfect 👍 — the **React Flow Playground** (on reactflow.dev/play ↗) is a great way to prototype this quickly.

Let's go step-by-step so you can copy and paste directly into the **Playground code editor** there.

🪶 Step 1: Open the React Flow Playground

Go to 👉 https://reactflow.dev/play ↗

Then in the **Code** tab, replace everything in the editor with the following:

🎀 Step 2: Paste This "Bow-Tie" Diagram Code

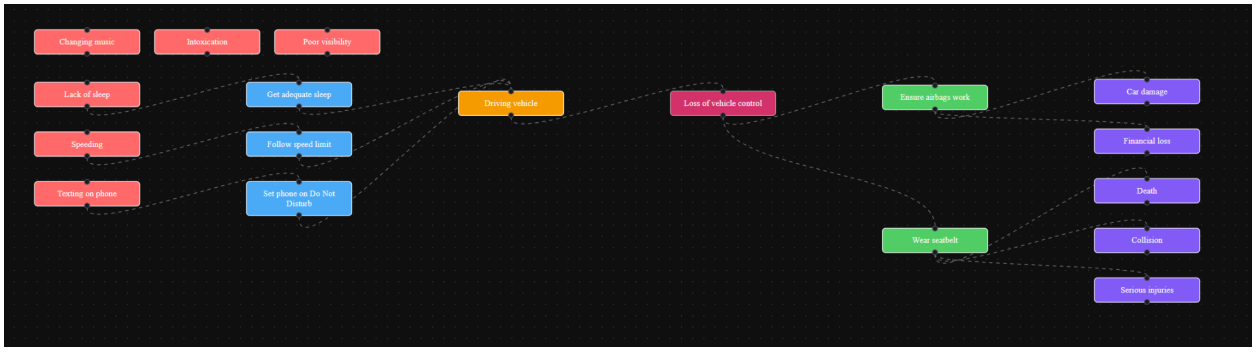
jsx Copy code

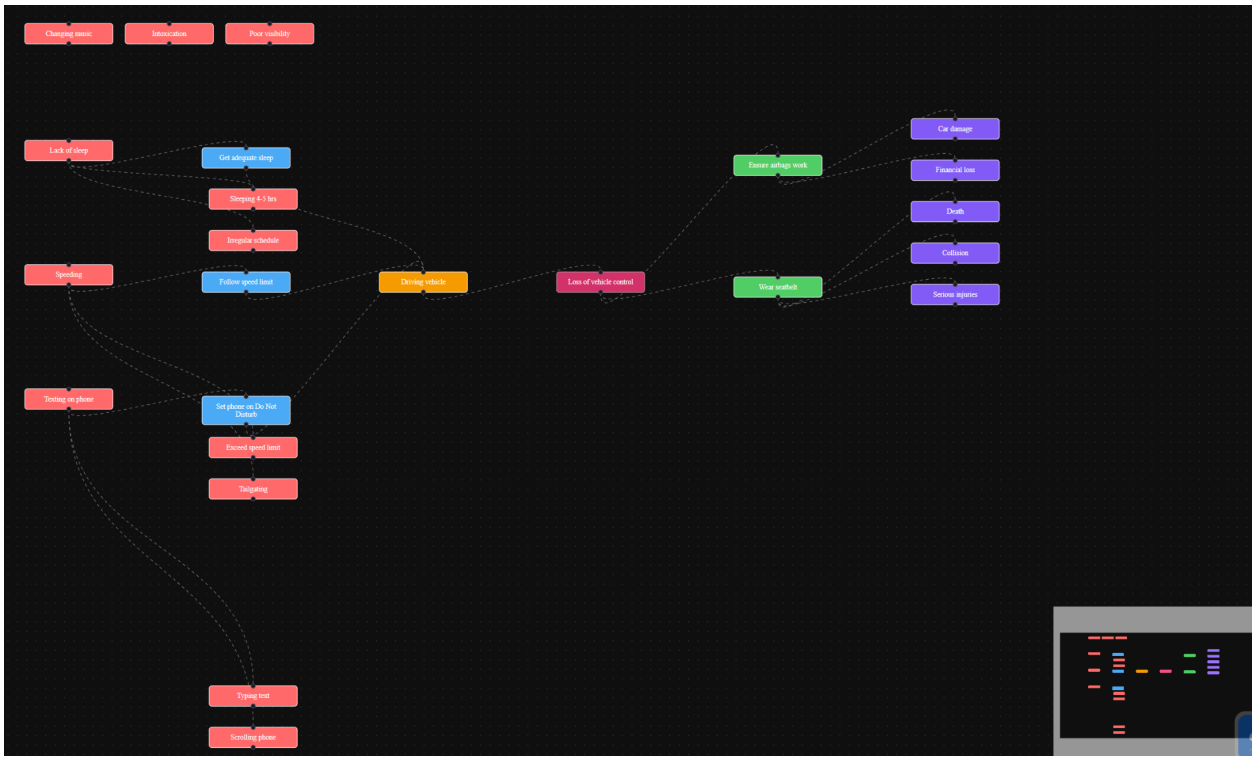
```jsx
import React from 'react';
import ReactFlow, { Background, Controls, MiniMap } from 'reactflow';
import 'reactflow/dist/style.css';

const initialNodes = [
  // Causes (left wing)
```

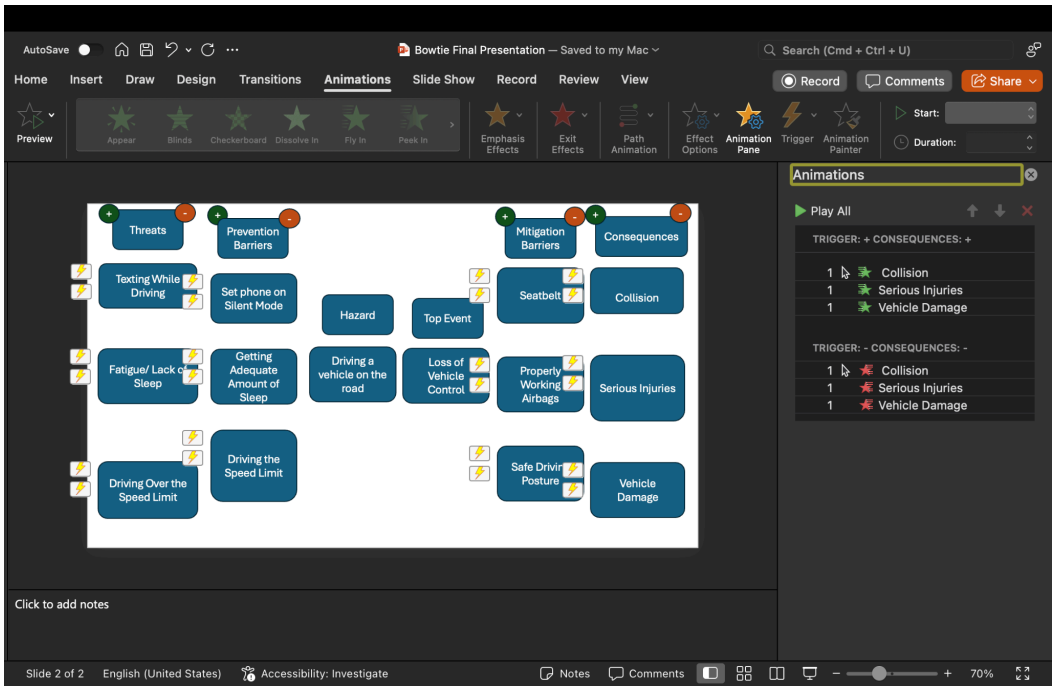
We were never able to find this "Code" tab in Reactflow Playground and after a long back and forth with ChatGPT we concluded that Reactflow Playground was not a potential solution to create the Bowtie Diagram interactability.

Since pro ended up being paywalled, Chris installed Reactflow onto his local machine and tried to code this functionality since it was either pay for pro or code the solution ourselves. The time to learn and code this functionality seemed unrealistic for us as a team to complete before the deadline. Below are some screenshots of work in progress through Reactflow coding on the local machine.





The nodes were expandable and collapsable but it was never consistent in what happened. Since some of us had not coded in a very long time or have very little to no coding experience, we then decided to use powerpoint after the guidelines had been updated to create a prototype instead.



During the creation of our bowtie risk visualization, April asked ChatGPT for guidance on how to structure the layout and make the bowtie interactive such as the alignment, spacing, expand/collapse buttons, trigger based animations and visual clarity. I learned how to duplicate shapes, establish visual hierarchies which improved readability. I also learned about how to organize information – starting with the bowtie structure first, then adding animations, as well as hover tool cards.

One challenge I faced was realizing how complex Powerpoint can actually be when used for more than just basic slides. At first I thought it would just involve placing shapes on a slide and using a few buttons, but there were many tools and features I was not familiar with (like the animation pane, triggers, shape formatting, etc). It was a bit time consuming at first, but once I got the hang of it I remembered where most things were. This process showed me how powerful Powerpoint can be when creating interactive visual tools – not just presentations.

However, we ended up going with Figma, because I was having trouble with learning how to expand/collapse horizontally. I could only do it vertically. But I kept running into small technical challenges where the animation wouldn't work, etc, which started to slow down the design process. At the same time, Chris was building a cleaner prototype in Figma that supported the interactions we wanted. I preferred the clarity and structure of his version, so we decided to use his approach instead. I still gained knowledge from this experience – I had no idea about the tools and techniques you can use in Powerpoint and will try to apply it outside of class in the future.