The theme for this project is intelligent browsing. I (Ishan Biswas, ibiswas2@illinois.edu) am the only member of my group; therefore I am also the captain. The specific topic that I have chosen is an AI assisted autocheckout. This is related to a problem because these days, almost every popular product is always sold out, even at release time. Thus, I wanted to create a solution to this issue, so that people are able to secure their popular or limited goods without problems. It relates to the theme as the extension aims to leverage the properties of Large Language Models and how they are naturally geared to interact with DOMs of various websites. These models are pre-trained on very large datasets to function on a general level, and this is something I will be taking advantage of by fine tuning via ensembling prompts to reach a desired outcome. To demonstrate functionality, I will provide a live demo of my chrome extensions on various websites, each with differing and unique DOM + class structures. This will be done in such a way that edge cases that are not addressed by other autofill extensions will no longer be an issue with this extension. I plan to do both the frontend and backend in Javascript, and the workload for this will far exceed 20 hours, as shown by the following steps:

- Research the necessary libraries and LLM implementations needed to complete this project (2 hours)

-Proof of concept a generic script adapting an LLM to identify fields in the DOM of pages, one at a time, that resemble autofill targets like payment info or user names (4 hours)

-Debug said script (2 hours)

-Train the LLM, or pick better starting parameters to better identify fields and test it against multiple pages (4 hours)

-Create a program to fill info loaded from preset configurations into the fields identified (2 hours)

-Illustrate a very basic frontend to allow the end user to plug in their desired credentials to be autofilled and toggle features (3 hours)

-Test, debug, and package this front end so that it can be loaded into chrome on different machines (2 hours)

-Prepare a live demo with target sites outside of the ones used to train the project (1 hour)

-Refine the process as needed and fix bugs (2 hours)

Thus the time estimate for this project is roughly 22 hours in the most ideal case scenario.