The Goal

The goal of the benchmark is to collectively analyze how the web agent executes actions based on the User Query. That being said, we made a good list of examples that will be used to test the web agent on how it makes decisions for each action it takes or thinks it should take by comparing the actions of the web agent's predicted actions to the correct ones in the list of good examples.

Results Format

Each time you run the **`benchmark_agent_node.py`**, the results are consistently returned in the code format below, regardless of the action performed. Note that for certain actions like **`click`**, variables such as **`correct_direction`** will always be **`false`**. That being said, the expected return values differ depending on the specific action. We did the code this way so when put into the Json file it can easily be pushed to a pandas dataframe for analysis.

```
"action_matched": action_matched,
 "correct_action": correct_action,
 "predicted_action": end_state['prediction']['action'],
 "normalized_error": normalized_error,
 "time_taken": end_time - start_time,
 "target_window_matched": target_matched,
 "direction_matched": direction_matched,
 "cosine_similarity": cosine_similarity,
 "text_matched": text_matched
```

Below is our previous benchmark implementation which was changed to the final result we showed you above.

Actions	Outputs
Click	Correct bounding box, Predicted bounding box number
Type	Correct bounding box number and correct text Predicted bounding box number, predicted text
Scroll	**Open to suggestions.** Correct Window, Correct Direction Predicted Window, Predicted Direction
Wait	We can write a time function to match the time waited on the page to the correct amount of time
GoBack	URL match the Correct URL(Captured from the previous step) with the predicted URL
Google	URL match https://www.google.com/ with the predicted URL