All-in-one

Introduction

I was assigned to a campaign wherein large number of web applications and legacy tools needed to be used. The information on how to access and use these tools was scattered across different files and sometimes hard to find. The campaign was run on Live Chat medium and responding to customers in a timely manner was essential. However, agents face difficulties due to scattered information.

Observations

Observations gathered on agents' day to day usage:

- Browser bookmarks for web applications used to get wiped due to security clean up on regular intervals. So, some tenured agents used to keep links saved in email draft or on a word/excel document. However, new agents often used to miss out on this and had to depend on help from tenured agents.
- Many of the agents used to use Timer from open-source websites to keep track of their login time.
- Some agents preferred dark theme over light
- Multiple versions of lists with steps to access legacy tools. Some of them were either outdated or with hard-to-follow instructions.
- Agents needed to 1) identify NPA to determine customer region 2) a separate short list to know which App is to be used and its App ID
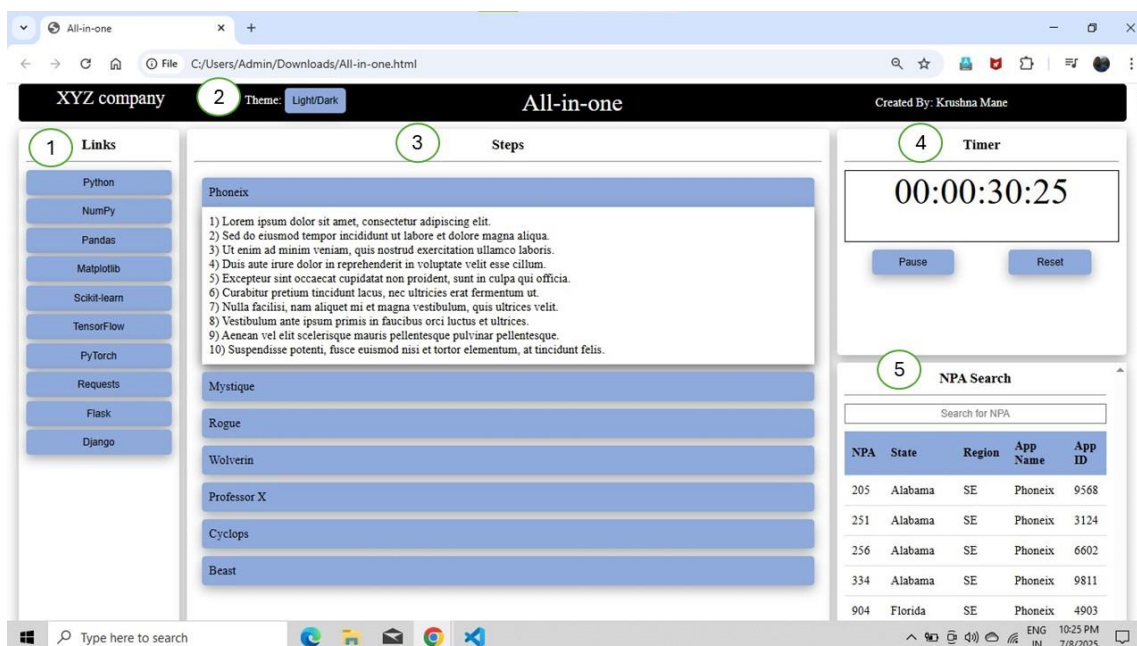
The Challenges

- Updated links need to be gathered in one place for ease of access. Bookmarking could achieve this, but the security clean-up used to wipe them on regular intervals.
- Since the information on steps to use legacy tools was scattered across different files, it wasn't user-friendly and time-consuming.
- External online timers were used, which aren't always safe and trust-worthy.

The Solution

Using HTML/CSS I created a **unified agent assist** file called **All-in-one**. It included:

1) Links of frequently used web applications
2) Option to switch to dark or light theme
3) Space saving collapsible list of steps on how to use legacy tool.
4) In-built Timer running on browser to keep track of their login hours
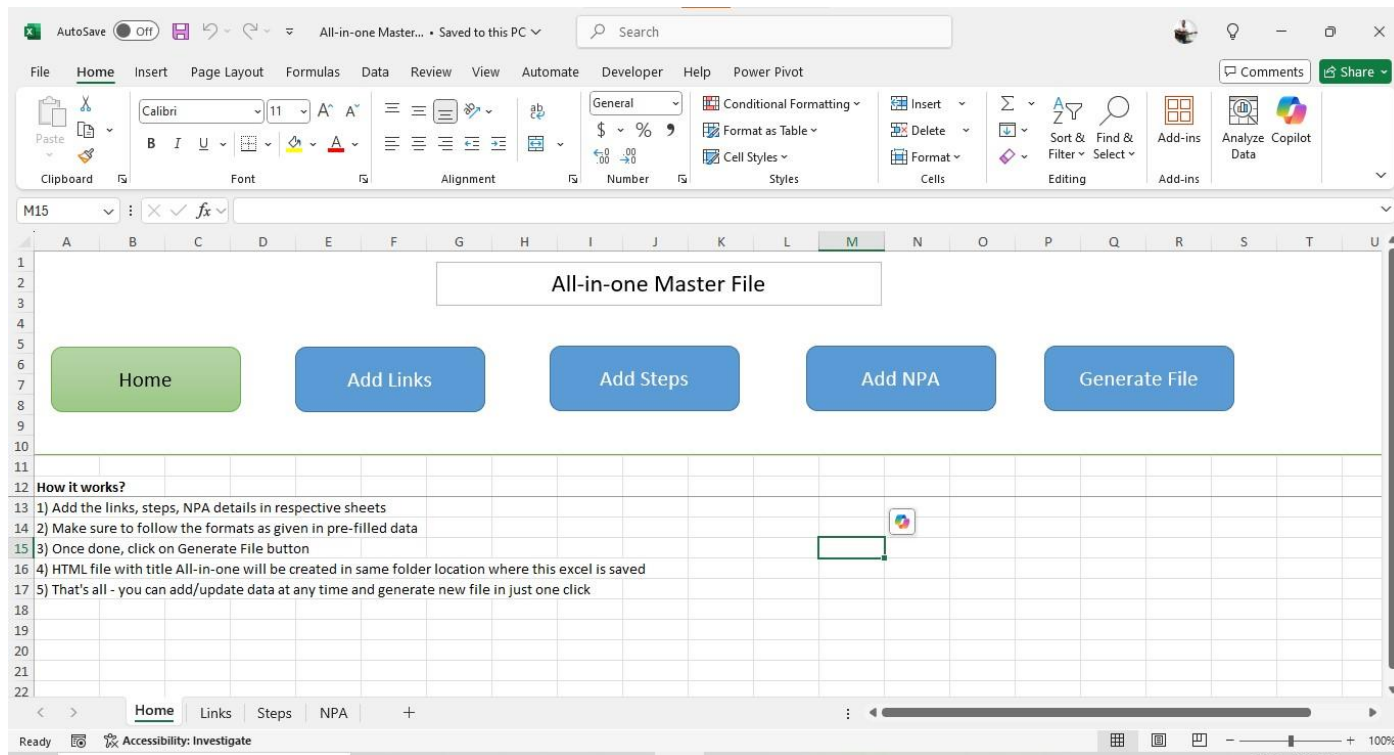5) Searchable list of NPA alongside region, App name, App ID to be used

Snapshot:

Update provision

All-in-one file was well received by agents and proved to be successful as intended. However, the next challenge was what if the file needed an update in the long run. So, as a progression to first file, I created a Master excel file with VBA coding capable of writing a new HTML file with updated information. This file was then handed to Team Leaders, so they can release updated file whenever needed.
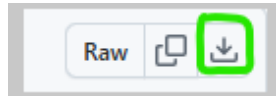
Snapshot:

Download Instructions

All-in-one HTML file:

- Go to https://github.com/krusmane/my_projects/tree/main/Source
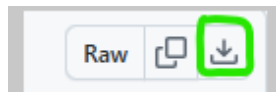- Click "All-in-one.html" file

- Click on "View Raw" link or Download icon

All-in-one Master File:

- Go to https://github.com/krusmane/my_projects/tree/main/Source
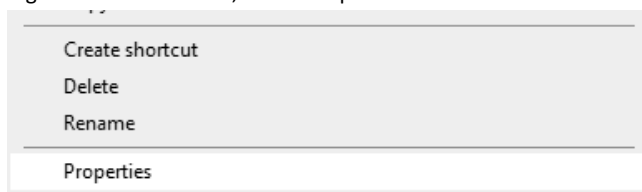- Click "All-in-one Master File.xlsm" file

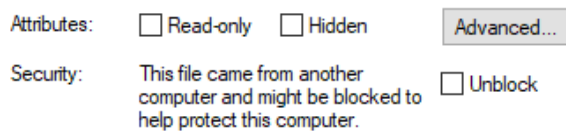- Click on "View Raw" link or Download icon

*Note 1: Since this is a macro-based file, by default your system will apply a security block when you download and use it for the first time. Don't worry though, you just need to **Unblock** that security block and **Enable Content***

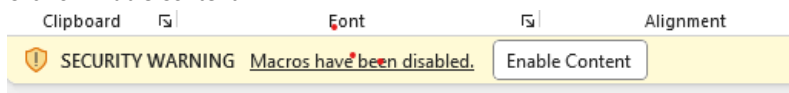**Here are the steps:**

1. Right click on file name, select Properties

2. Look for Security info at the bottom, check on Unblock

3. Click on Enable Content

Note 2: This file uses VBA function to write new file using local system path. So, the folder you save the file on must be a localized folder and not a one backed up on cloud. Example: folder backed up using one-drive