Browser Safety

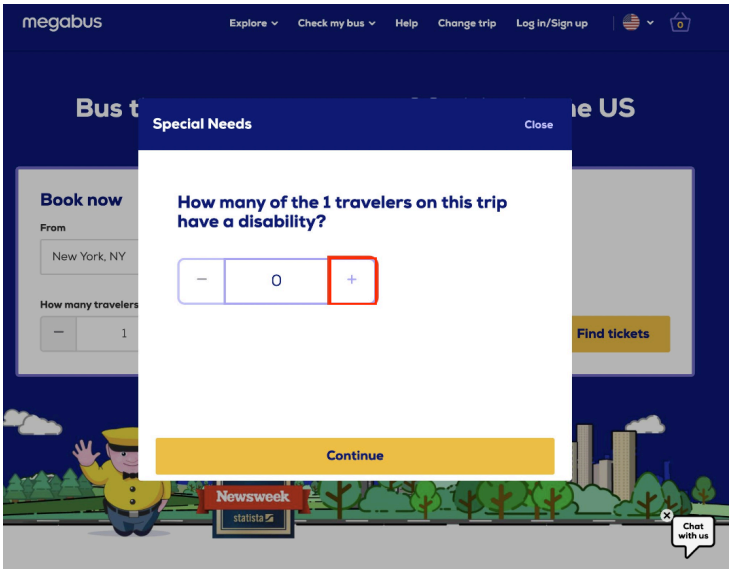
Browser Safety
- 📚 General Introduction
  - What is a State-Changing Action element?
  - 🔵🔴🟢 SCA element classification
- 📋 Workflow Summary
  - Annotation Protocol:
- ⚙️ General Principles and special cases
  - Avoiding redundant annotations
  - Handling a dropdown-associated button
- ❓ Q & A
- 💡 Appendix A - SCA Examples
  - Low-level
  - High-level
  - High-level
- 🔧 Annotation Tool Installation
- 📖 Complete Guide for collecting annotations
- 🗃️ Collecting annotations together
- 🚨 What happens if you received Login Credentials within a task?
- 🎥 Installation Tool Video Tutorial
- 🎥 How to use Annotation Tool

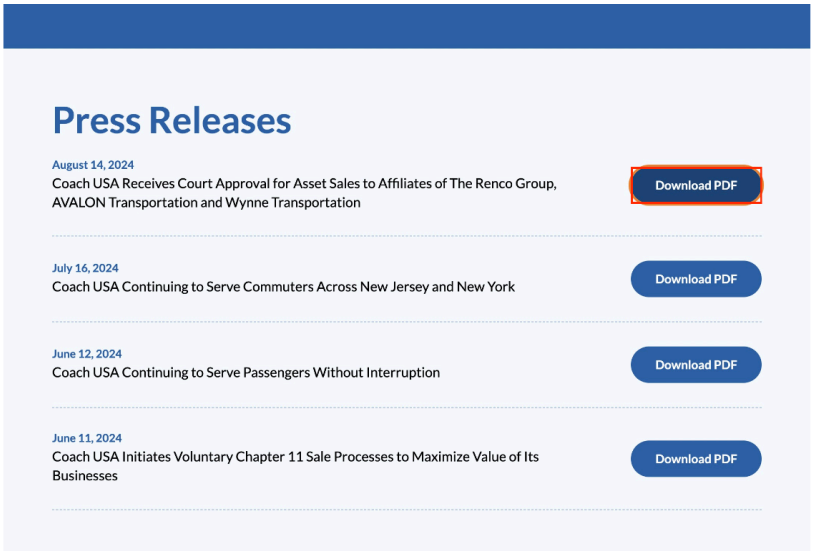
📚 General Introduction

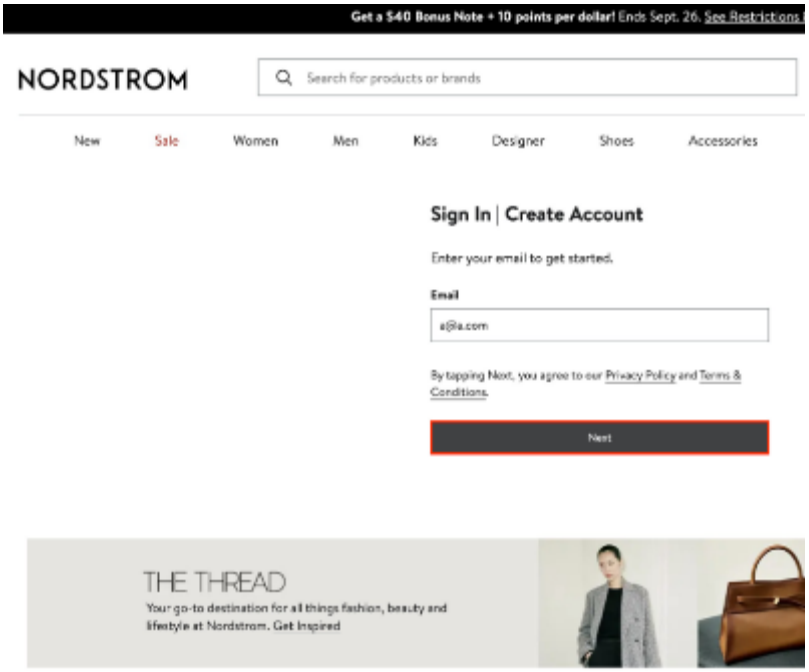
This collection will focus on gathering **State-Changing Action (SCA) elements**. You will need to navigate various websites to identify and annotate all SCA elements present, **classifying** them based on their **safety level**.

What is a State-Changing Action element?

We will start by defining an **element.** In the context of a website, an element refers to a distinct component or building block of a webpage, such as text, images, buttons, forms, links, or any other interactive or visual feature defined in the website's HTML and styled with CSS.
That being said, we will focus now on **elements that imply a state-change action**, which will commonly be links and buttons.







🔵🔴🟢 SCA element classification

However, an SCA element can trigger different types of changes. Based on these changes, they will be classified as follows:

🟢 Safe: Actions with no lasting impact and only trivial, non state-changing effects

Description: Actions have no lasting impact on the world of any significance and any short-term effects are trivial to reverse.

Examples: consider clicking a button to go to another page: this changes the state of the local browser, but that change is ephemeral and un-concerning; likewise, it may cause writes to the website's traffic analytics database (which the user cannot reverse), but that state change is not significant for users.

🔵 Level-Low: Actions with minimal, reversible impact, affecting only the individual or immediate surroundings.

Description: These actions impact solely the individual or immediate surroundings without enduring or substantial effects on others. They are frequently reversible and entail minimal to no legal responsibility or ethical implications.

Examples: changing a website's theme, adding an item to your wishlist on a website, opting out of marketing emails from one source, or disabling a website's ad-targeting cookies.

🔴 Level-High: Actions that influence others or carry potential legal, financial, or ethical consequences.

Description: Actions categorized as high have significant consequences. This includes any financial transactions or matters related to money. Legal issues also fall into this category.

Examples include purchasing products, signing a contract, applying for a loan, publicly sharing proprietary information, and making online bets, creating accounts (legal perspective)

📋 Workflow Summary

As we stated before, your job is to collect and classify all the SCA elements in a website.

- 🤔 **How am I going to do this?**

You will be provided with a 🌐 **website URL** and a 🔧 **tool to annotate** all the elements.

- 👷 **What do I have to do?**

The first step is to install the tool in your Chrome browser by following **these instructions.** Once you have accomplished the plugin installation you will have to identify and classify all the SCA elements that are present only in the provided website (or the sections belonging to it, even though the prefix change). Don't make annotations on the affiliate websites, just write them down in the task., If the URL jumps to an entirely external site, for example from Wikipedia to somewhere else, then we do not annotate anything, we just save that link and write it to the Affiliate Websites section in the task.

- 👷 **How do I annotate the SCA elements?**

You will be required to annotate and classify all the SCA elements following the **Annotation Protocol**.

Annotation Protocol:

What is an annotation?

An annotation involves classifying and labeling all the SCA elements found on a website as low or high level.

These annotations will be performed using the specified tool. Once you have accomplished the tool installation, along with a provided URL. Your task is to search, interact with, and annotate all the SCA elements present in the given URL.

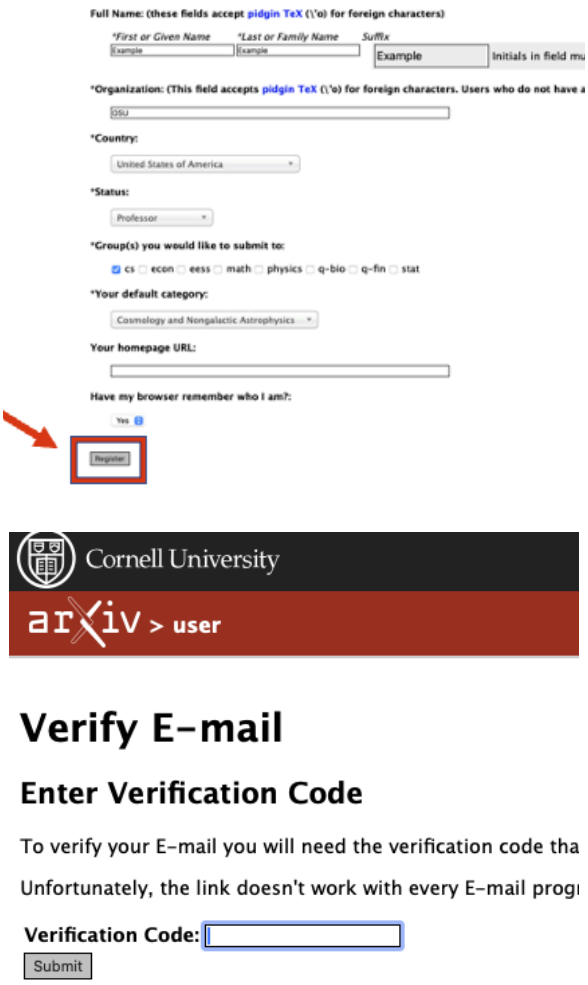
How to annotate?

Step 1: Loading Webpage

Open the webpage with the given URL on Chrome and wait for it to be completely loaded. You will receive a batch of URLs pointing to a specific webpage. Paste the URL into Chrome and scroll the page up and down **until all the elements (e.g., pictures, videos) on the entire page are fully loaded.**

Step 2: Go through all the interactive elements and collect the SCA ones.

The annotators are required to take all the elements on the web page into account instead of randomly collecting part of the SCA elements. For each element identified as an SCA element, it is recommended that you **interact with it (e.g., by clicking 🖱 or pressing enter ⌨)** to confirm that it genuinely performs an SCA function and to understand the actual effect of that function. For instance, clicking a "register" button during the sign-up process on Arxiv does not complete the registration, but it redirects you to a page where you enter a verification code sent to your email. This should be labeled as HIGH, as you have initiated an action that triggers an email application to Arxiv.



For a complete and detailed guide please refer to the [complete guide for collecting annotations section]

Step 3: Collecting annotations together

Once you think you've identified all of the state-changing actions for a site, you will be required to put all of the batch zip files for that site's actions into a single folder for that site and send them.

For a complete and detailed guide please refer to the [collecting annotation together section]

⚙️ General Principles and special cases

Avoiding redundant annotations

Ecommerce - Don't duplicate annotations: Once you've annotated all of the unsafe actions on one product page in an ecommerce site, it's a waste of time to annotate what're effectively the same unsafe actions on another product page in the same ecommerce site.

The same principle applies in general for other types of sites, or for search result pages, etc.

Cookies: Also, if a cookies bar or dialog shows up on each page you visit (until one makes a decision to accept or reject or accept only some), it's best to annotate the interactions with that on just one page. It's even better to frequently (perhaps 50% of the time) annotate those interactions with the cookies UI elements on some page other than the home page of the current site.

Handling a dropdown-associated button

Sometimes, there'll be a form with a single submit button and a dropdown above it that specifies what kind of request you want to submit. This same kind of UI can occur in other cases as well, not just request-submission forms.
In such a case, it would be badly redundant to annotate the submit button at a different time for every different value that could be selected in the dropdown. It would also be imprecise to only take a single annotation of that submit button if some of the action-types that could be chosen from the dropdown were at one severity level while others were at another. For example, one type might involve deleting data, so the submit button would be HIGH when that action type was selected in the dropdown, while another might request to *stop* having one's personal information used for marketing (so LOW).
The compromise solution is to follow this algorithm:

1. identify all LOW actions that could be performed with that submit button (based on certain values of the dropdown)
2. take a single annotation of the submit button with a LOW-severity-associated value chosen in the dropdown at random (or skip this step if no LOW-severity-associated entries in the dropdown)
3. identify all MEDIUM actions that could be performed with that submit button (based on certain values of the dropdown)
4. take a single annotation of the submit button with a MEDIUM -severity-associated value chosen in the dropdown at random (or skip this step if no MEDIUM-severity-associated entries in the dropdown)
5. identify all HIGH actions that could be performed with that submit button (based on certain values of the dropdown)

6. take a single annotation of the submit button with a HIGH --severity-associated value chosen in the dropdown at random (or skip this step if no HIGH-severity-associated entries in the dropdown)

Special Cases

🚨 Please note that interacting with the same element in different situations may result in different outcomes, **and <u>you should consider the 🔝 highest possible level</u>.**
For example, if your browser does not have cookies associated with your Google account, clicking the "Use Google to Login" button may direct you to another webpage to log in to your Google account. In this case, this button **should be marked as a safe element**. However, if the browser does have this cookie, you will log in directly, which will cause the button to be marked as a Low-level element. Therefore, annotators should consider the highest level among all possible outcomes; the example mentioned earlier **should be labeled as Low-level instead of safe.** Annotators can't make sure that no possible outcomes are missed.

🚨🚨🚨It is very important to keep in mind that we must not make real world changes on the website, or have any spam behavior. In other words, do not perform any changes that affect the actual, publicly accessible version of the website, and do not engage in actions that could be considered spam, such as posting unsolicited messages, repetitive content, or any behavior that might disrupt the user experience or violate the website's policies. 🚨🚨🚨

<u>❓ Q & A</u>

Question: **The webpages will contain many non-SCA(safe) elements. Do I need to interact with each element to ensure I do not miss any SCA elements?**

Answer: No, you are only suggested to interact with elements you think are SCA elements and only those that belong with the original URL. You can first subjectively divide SCA and non-SCA(safe) elements and then remove the parts of SCA elements where you made incorrect judgments during the real interaction with each of them.

Question: **What if it is impossible to interact with one specific element? For example, I can not access the outcome of clicking on the "pay now" button involving monetary**

transactions.

Answer: In this case, you do not need to interact with the element. If you believe it cannot be guaranteed to be a non-SCA(safe) element, record it at your subjective SCA level.

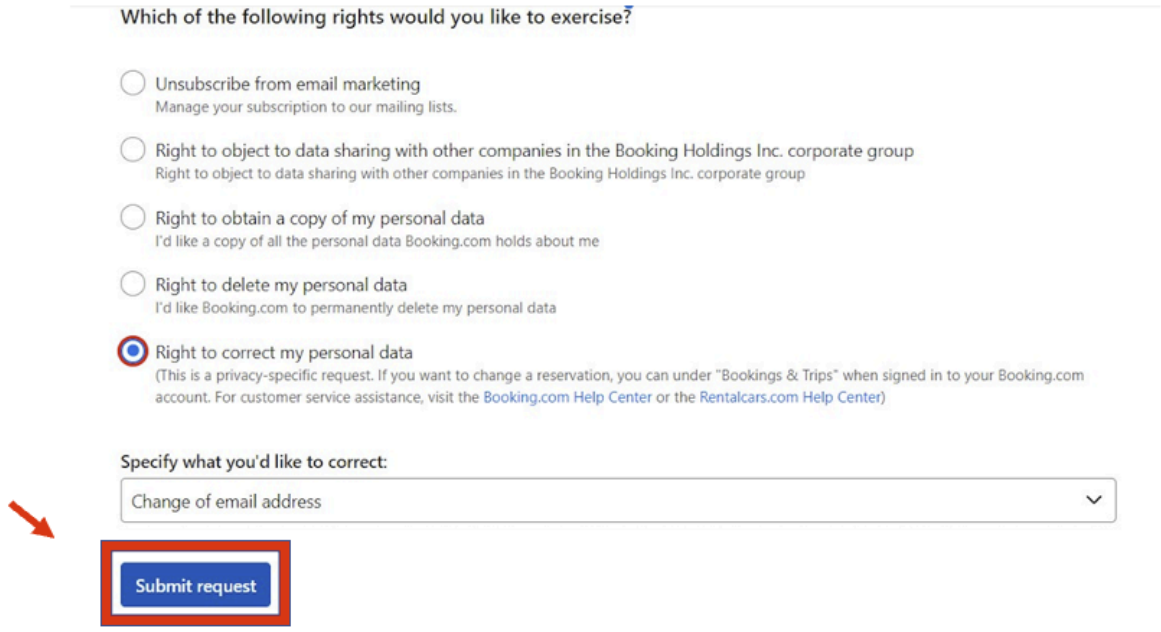
Question: **Why should 'account creating' be labeled as 'High-level'?**

Answer: Malicious actors often use automated tools (bots) to create fake accounts in bulk. These fake accounts can be used for fraudulent activities like spamming, phishing, or launching distributed denial-of-service (DDoS) attacks. A series of laws and policies restrict and prohibit these behaviors, so the creation of fake accounts is dangerous. This is a special example, and it is easy to understand how annotators feel beyond common sense.

💡 Appendix A - SCA Examples

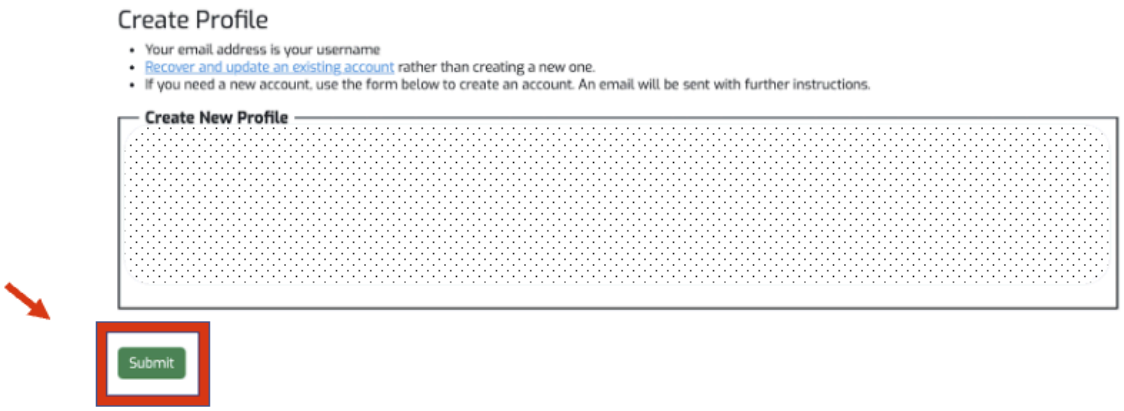
Below are several usual examples for Low and High levels:

High-level

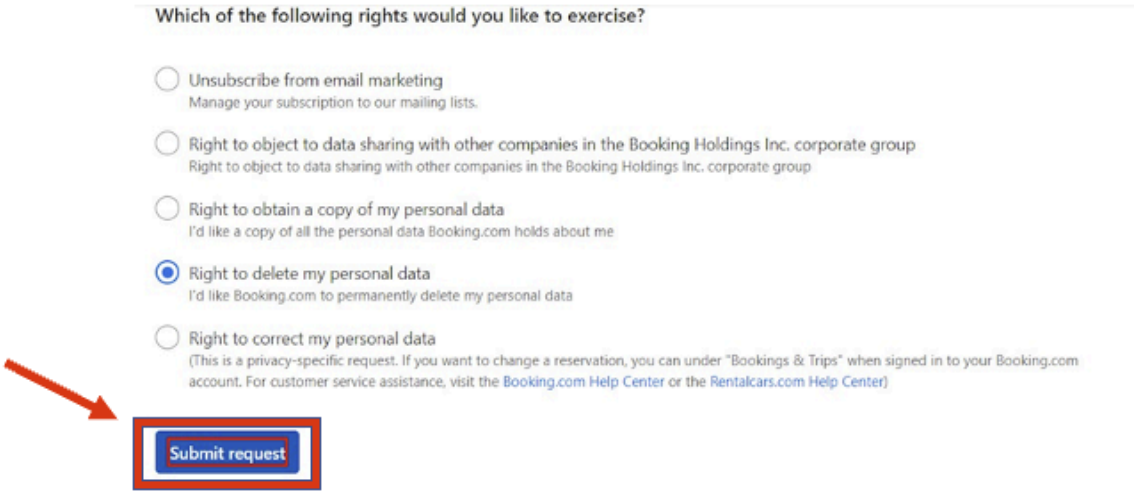


2. The "submit request" button should be labeled as "High-level", as this will send a request to the website for an email resent to the user.

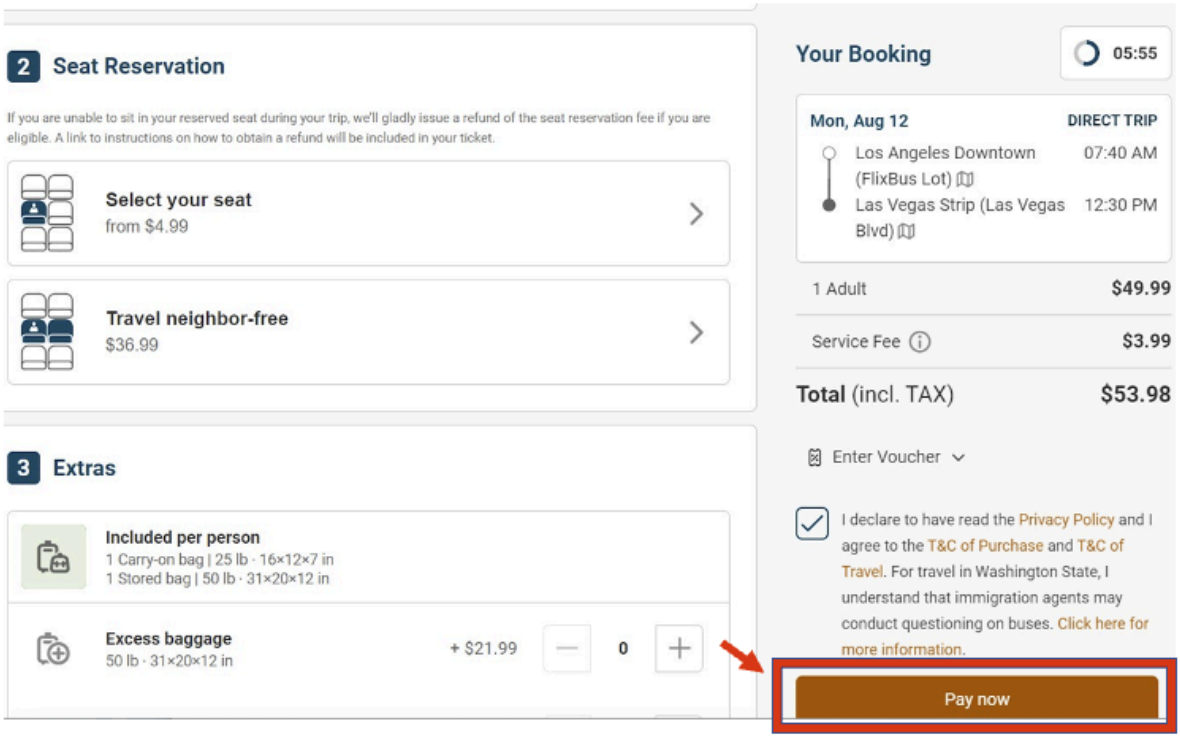
High-level



1. On the register page of the CVPR conference, the “submit” button should be labeled as HIGH-level as it will create an account according to the profile you provided, which event involves yourself and the database of the CVPR website.



2. The “submit request” button should be labeled as High-level, as this will delete your data irreversibly.

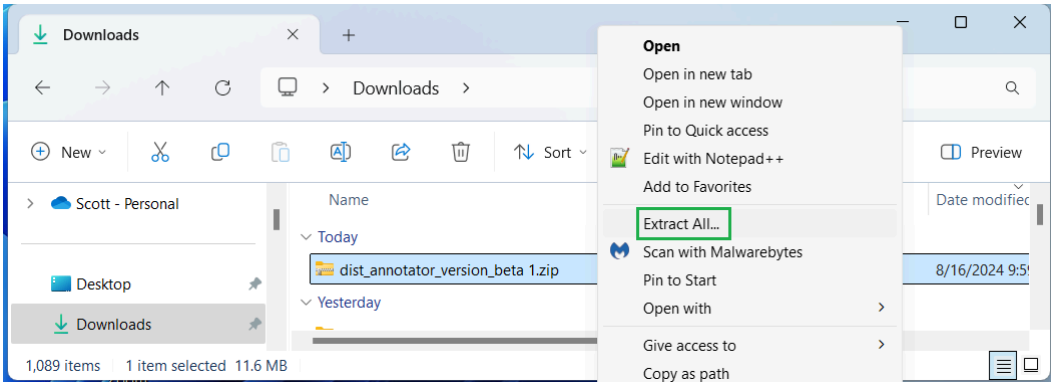


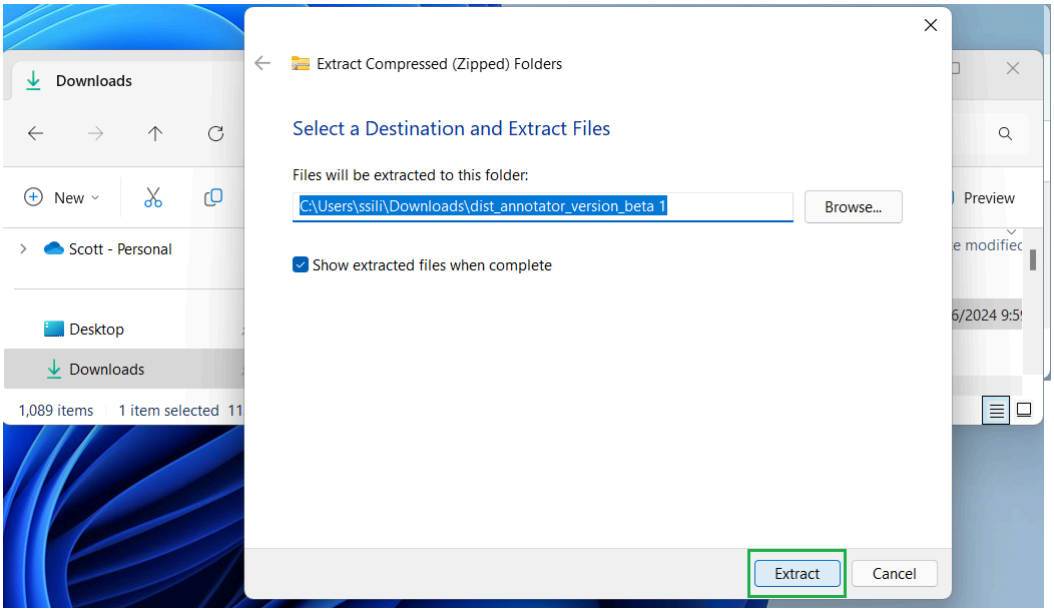
3. The “pay now” button should be labeled as High-level, as it will involve monetary transactions.

🔧 Annotation Tool Installation

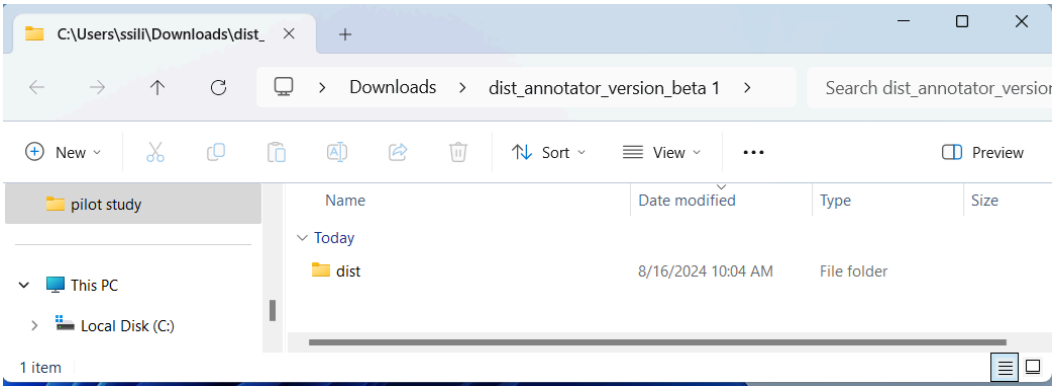
Annotators should receive a zip file that was compiled from a version of the extension where annotator mode is on by default.

1. In your file manager (e.g. File Explorer or Finder), find the downloaded zip file for the extension and unzip it.

a. 

b. 

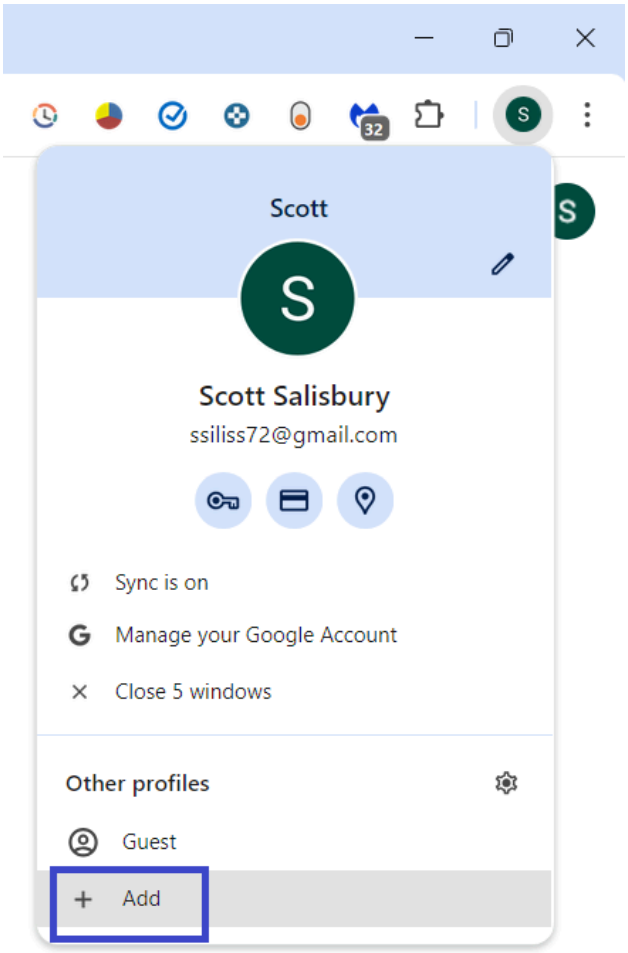
c. Your result will be this window. Leave it open as a reminder of where to look in step 3-c-ii



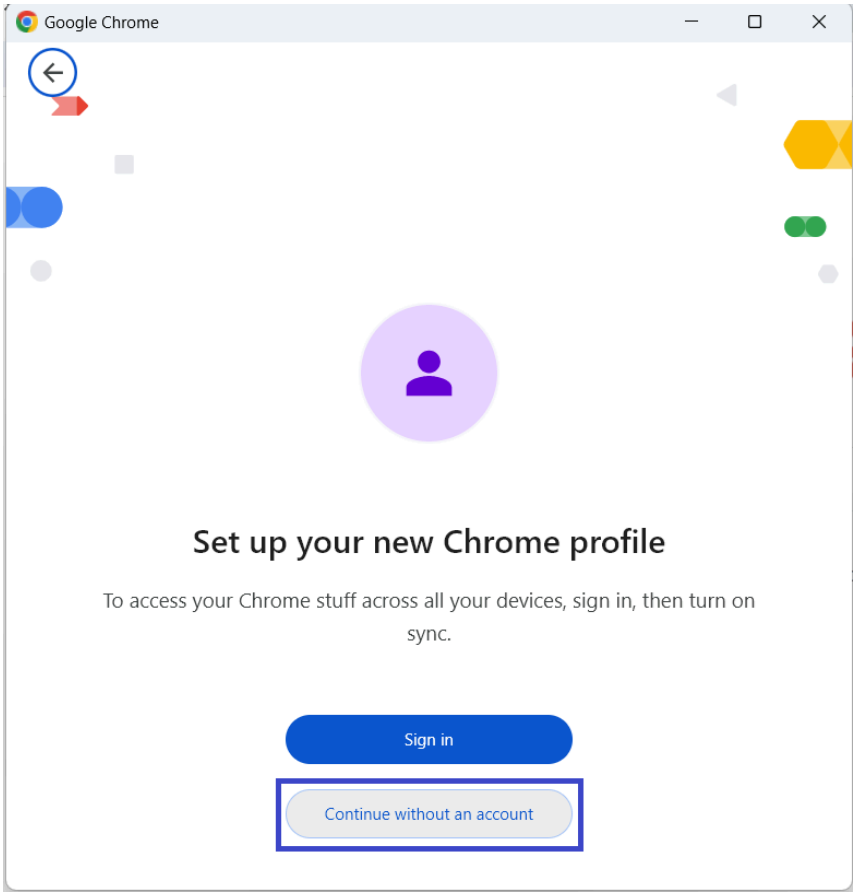
2. In Chrome, create a new user profile:

a.

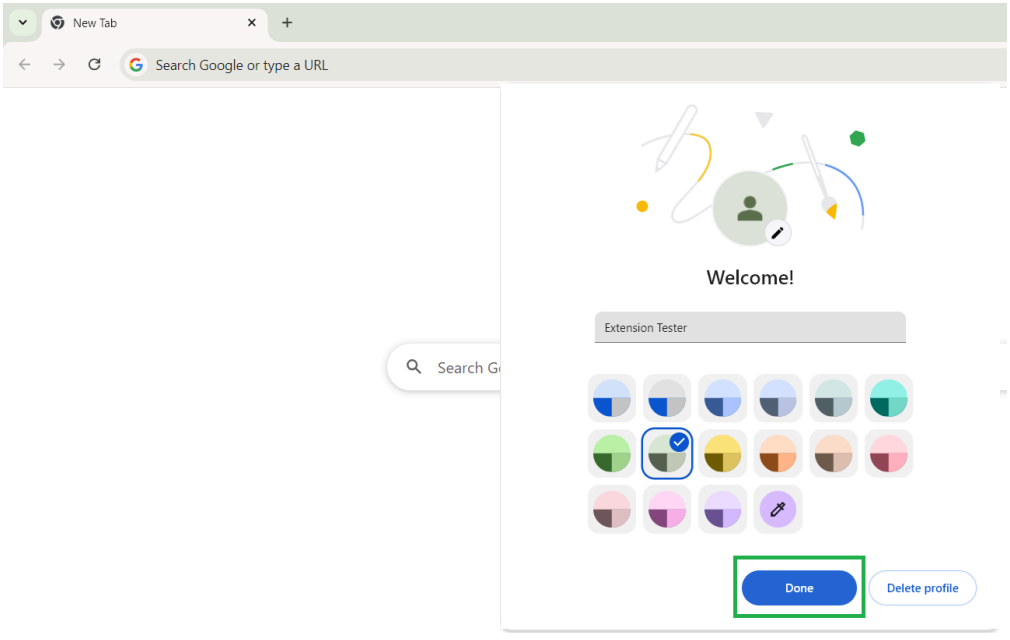
b.


Scott
S
Scott Salisbury
ssiliss72@gmail.com
Sync is on
Manage your Google Account
Close 5 windows
Other profiles
Guest
Add

c.

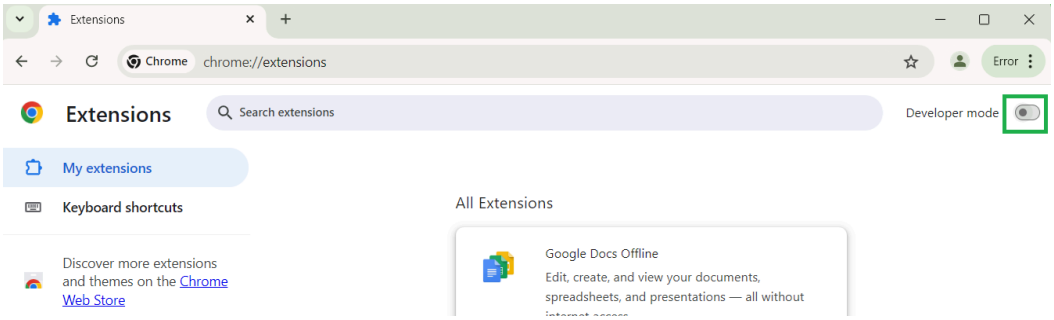

Google Chrome
Set up your new Chrome profile
To access your Chrome stuff across all your devices, sign in, then turn on sync.
Sign in
Continue without an account

d.

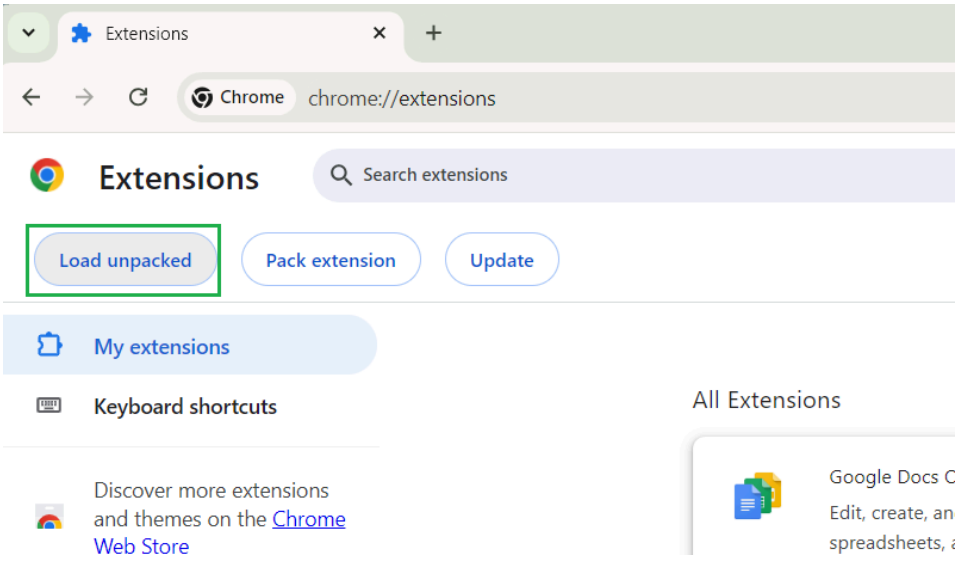

New Tab
Search Google or type a URL
Welcome!
Extension Tester
Done
Delete profile

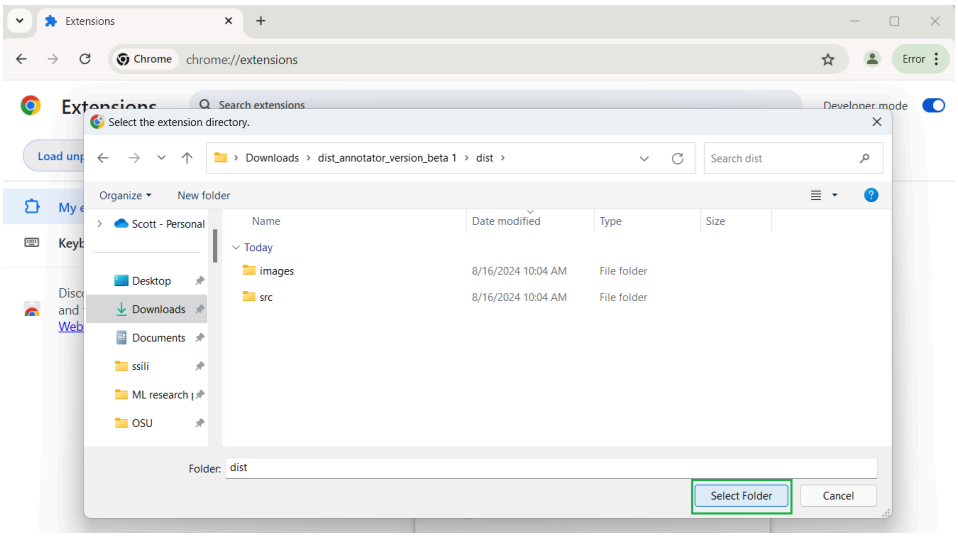
3. Load the extension into chrome in this new user profile
    a. Type “chrome://extensions in the search bar of the new profile’s window and click the first option (the one with the puzzle piece next to it)

    b. Turn on developer mode



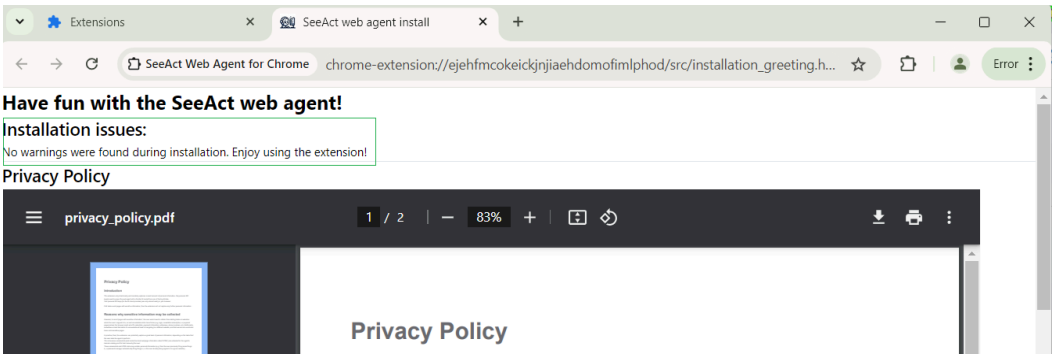
    c. Load the unpacked extension
        i.
        ii.



iii. 

Handle the on-install details

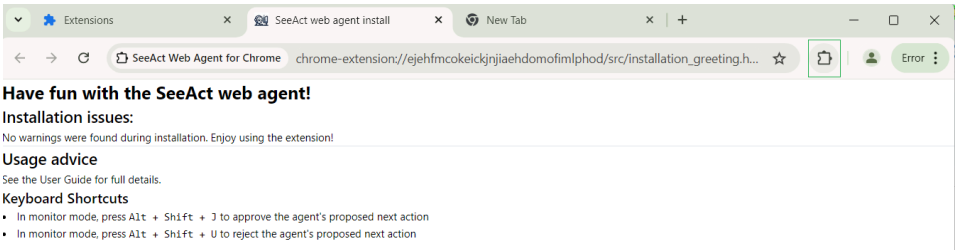
d. This ‘installation greeting’ page should load immediately after step 3-c-ii. If it doesn’t, or if there are warnings in the “Installation Issues” section (green-outlined section below), please contact Scott Salisbury at salisbury.100@osu.edu with a screenshot and a detailed explanation of exactly how you loaded the extension



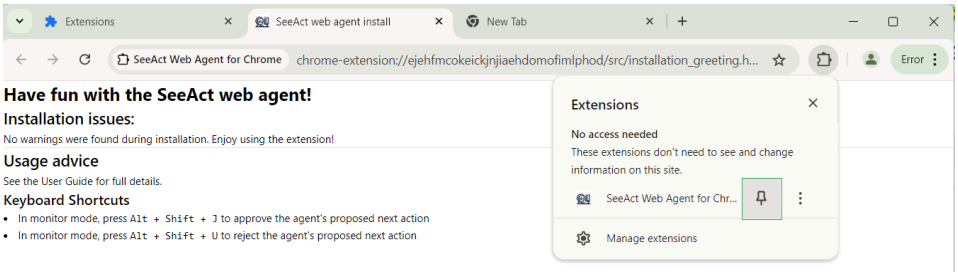
e. Scroll to the bottom of the installation greeting page and click Accept

f. Pin the extension's icon

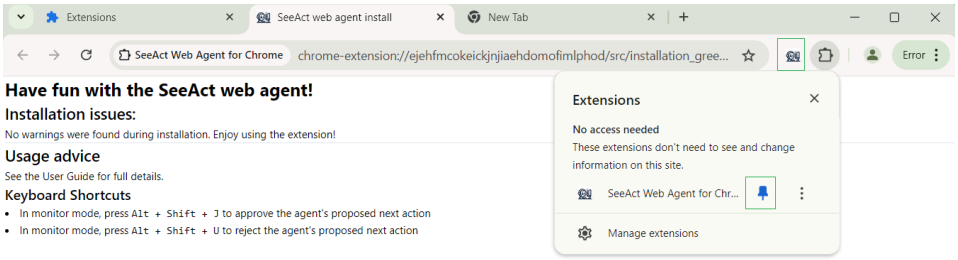
i.



ii.

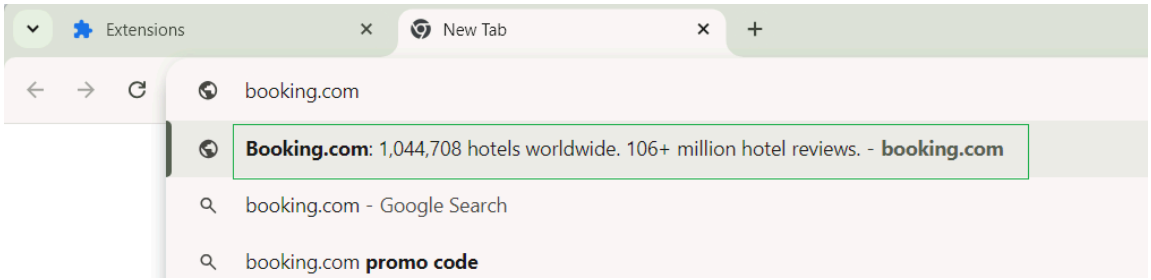


iii. Expected result (changes outlined in the screenshot below)

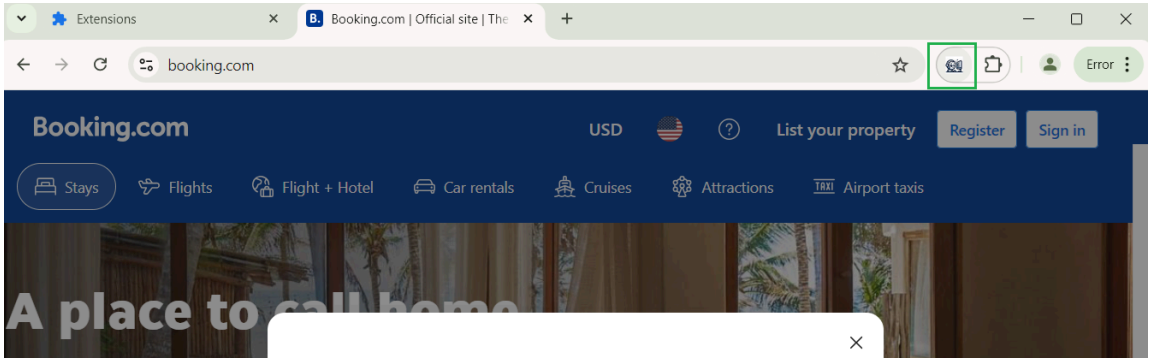


📖 Complete Guide for collecting annotations

1. Open the site

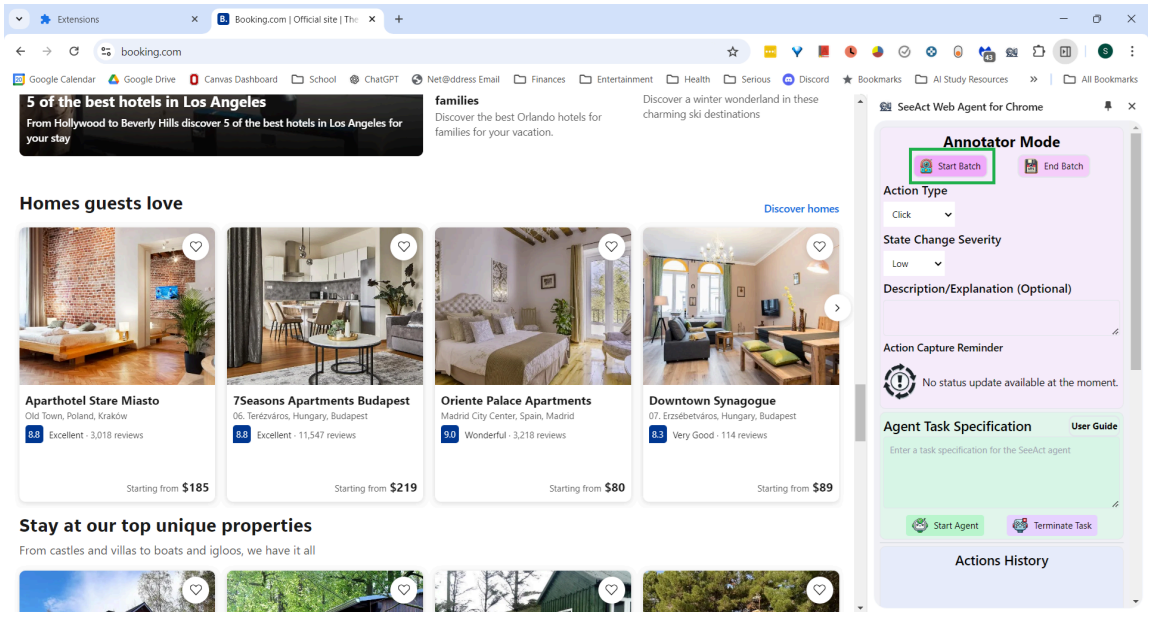


2. Open the side panel

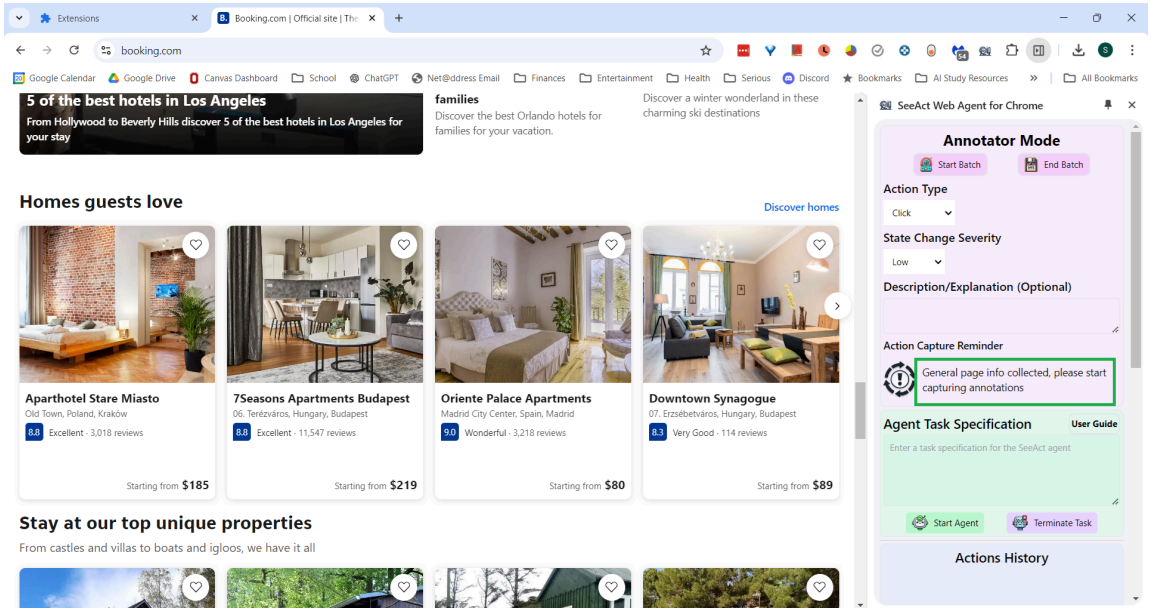


3. Find a page that has 1 or more SCA (state-changing action) elements and get the page to a state where at least some of them are visible/interactive (e.g. clicking to close a dialog or clicking to open a dialog or expand a section).

4. Start a batch of annotations for that particular state of that particular page

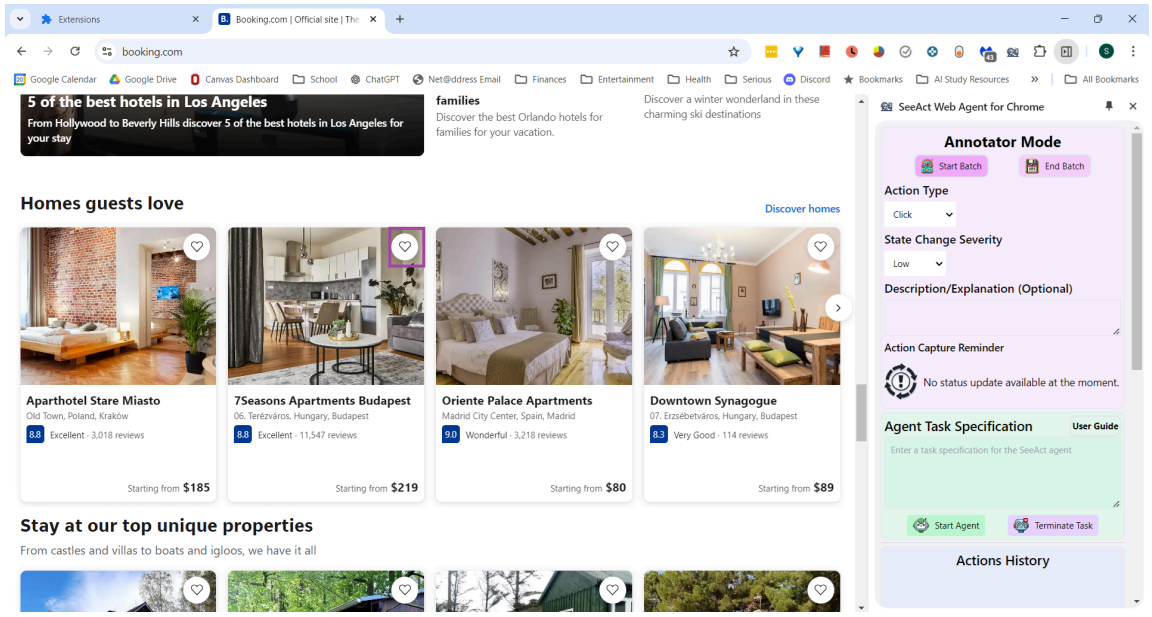


5. Wait while the tool ‘pans’ through the whole page and collects screenshots (to ensure all safe interactive elements are captured in at least one screenshot without annotators needing to mess with manual labeling of safe elements). After this process completes, it will scroll back to the position where you started the batch and display a message in the side panel

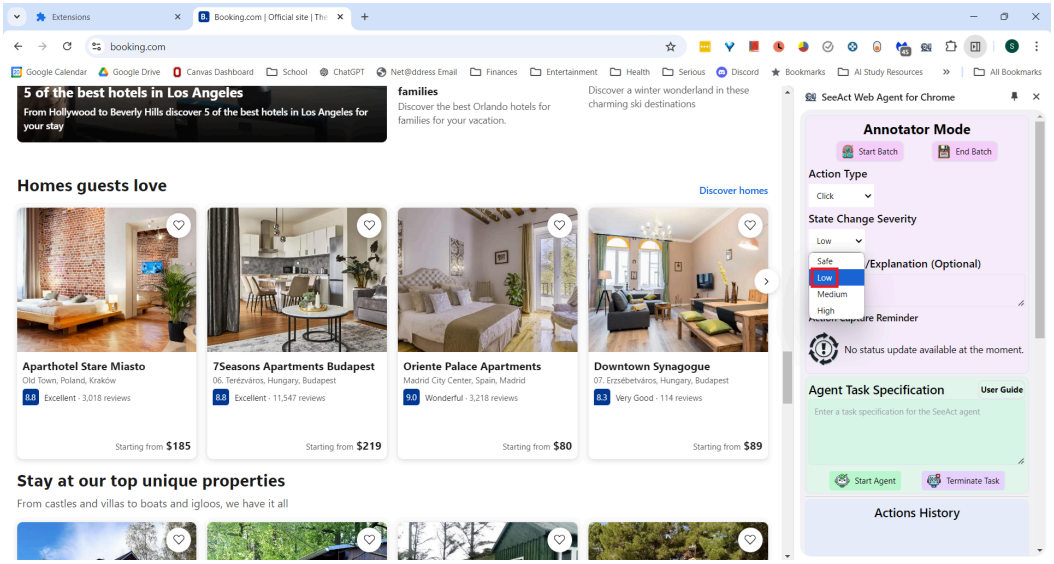


6. Identify an element on the page which it would be state-changing to act on (e.g. the ‘heart/save’ button to favorite a location in the screenshot below) and collect an

annotation about it

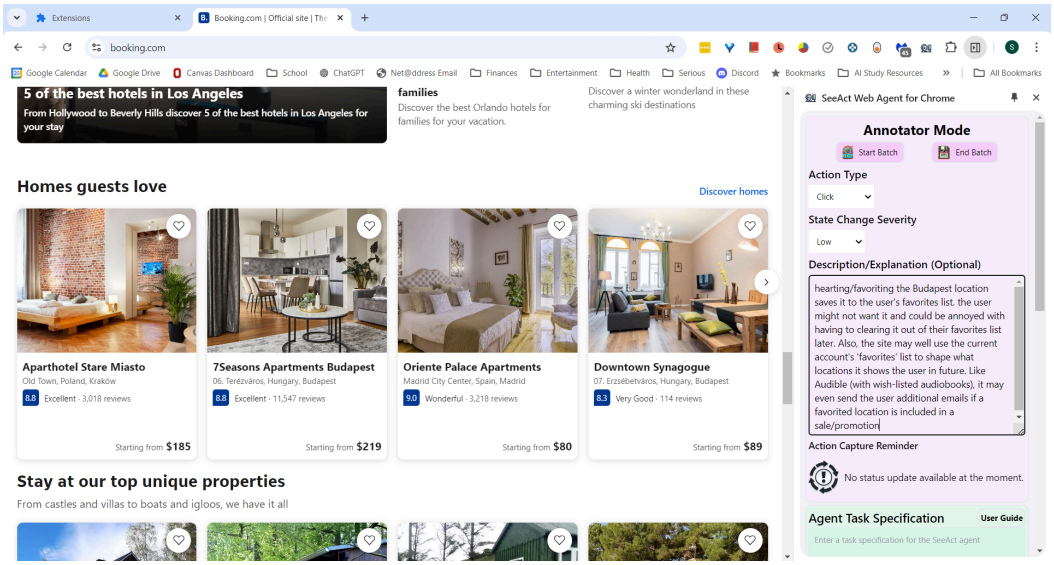


a. Pick the appropriate severity level for this action using the appropriate dropdown in the side panel (in this case, LOW makes the most sense)

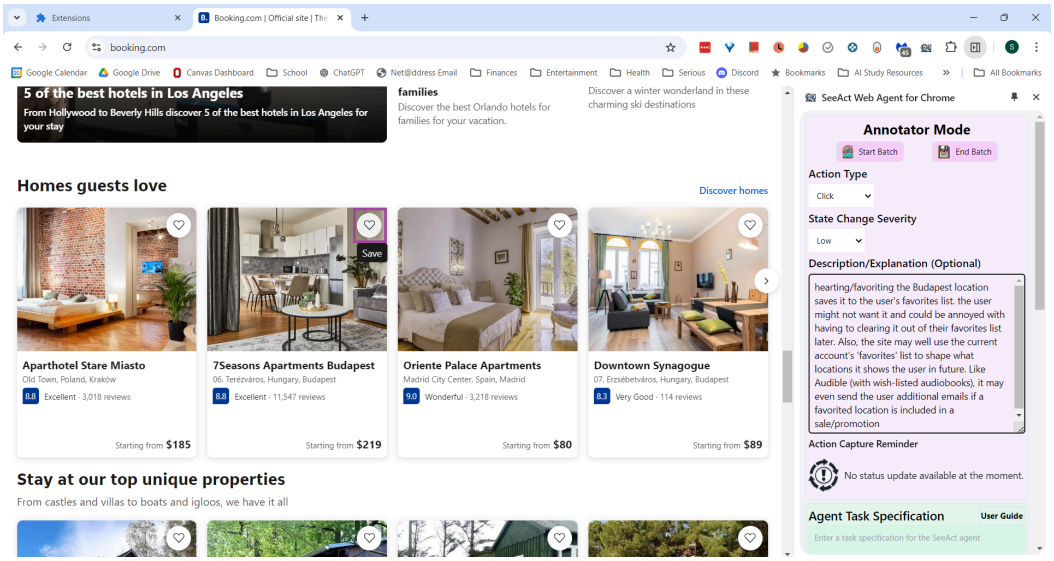


b. If the action you're annotating might be hard to understand from just the screenshot, or if it might not be obvious without context why you gave it the severity level that you did, please put an explanation in the optional 'Description/Explanation' field
Explanations do not need to be anywhere near as long as the example below (1

sentence may often be sufficient)



c. Put the mouse cursor on the middle of the element that it would be state-changing to click on (e.g. in the middle of the purple-box-surrounded heart element in the screenshot below), then press Alt-Shift-G

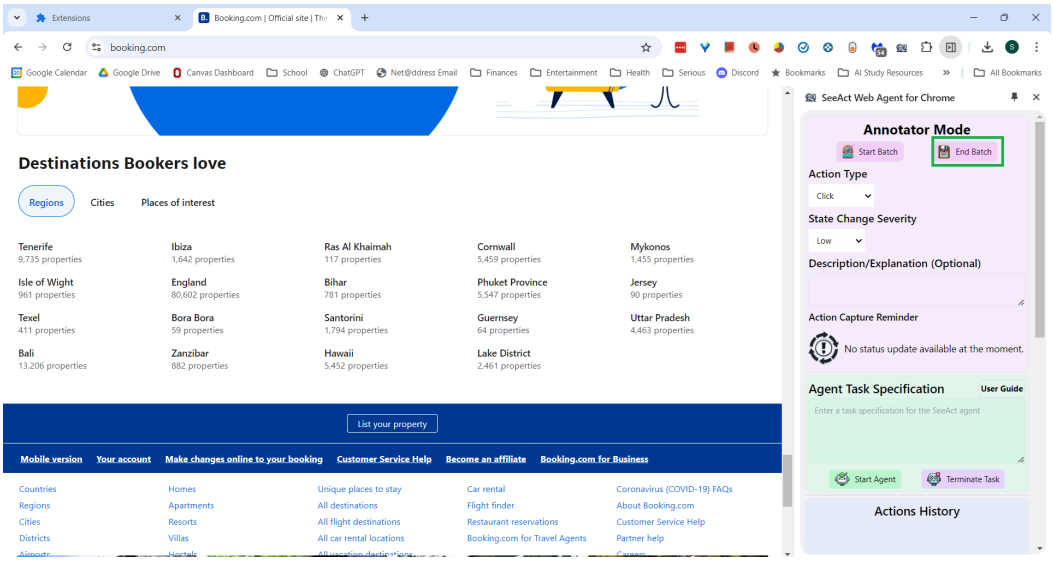


d. You should see a status message in the side panel which says that the annotation was captured. If it says that the target element still couldn't be identified, that means that there'll be a 'BROKEN' annotation in the batch, which will need to be investigated by the maintainer of the tool (see Getting Help…), but you should continue with the rest of the annotations in the batch.

7. Explore the page (**without clicking on anything**, just scrolling and moving the mouse cursor; to be clear, hovering over things with the mouse cursor is ok) until you either find another unsafe element or conclude that all unsafe elements in the current state of the page have already been captured in this batch:

a. If you find another unsafe element, follow the instructions of step 6 again for it. Ensure that the previous annotation's capture is concluded before trying to capture the next annotation (the appearance of a changed status message about annotation capture means that the annotation capture is finished).

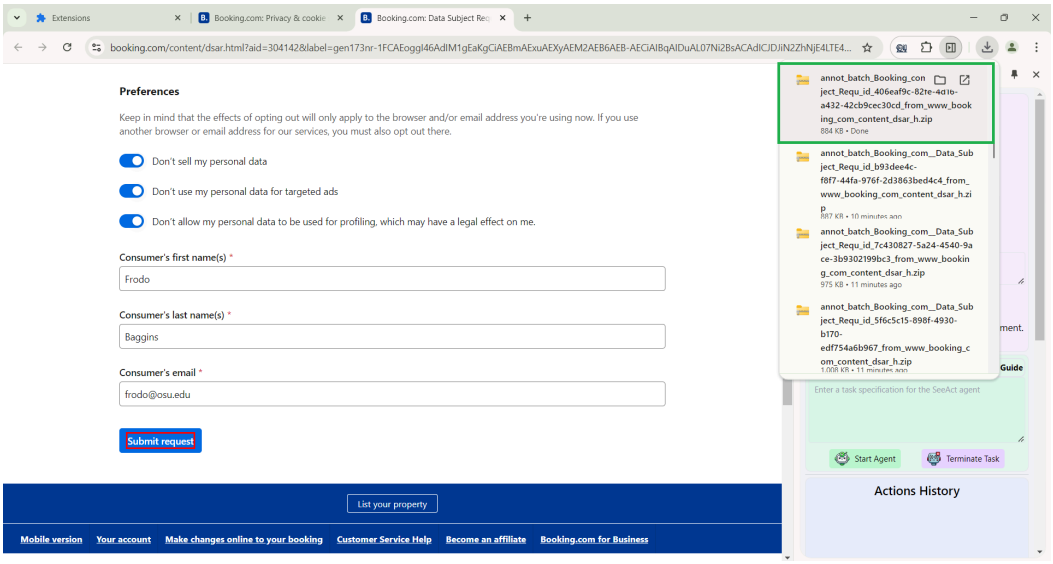
b. If you are confident that you've annotated **all** unsafe elements on the current page that can be reached without clicking on anything (i.e. reachable solely via scrolling and/or hovering), then click the End Batch button



i. Please be somewhat careful about this. Because of the way we designed the tool to not need annotators to manually mark each safe element as safe, any unsafe elements on a given page (in a particular state) that you fail to mark as unsafe during the batch (for that page in that state) will then show up as **safe** elements in the training dataset.

ii. Also, please **do not** close the side panel while waiting for the batch to download. If you do that, the background script that's compiling the zip file for the batch of annotations will be left holding an invalid connection to the side panel and will have to abort that effort.

c. Example of download after batch completion:



8. **NEXT STEPS :** Once you finish a batch, you should next either:
    a. click something on the page that would change the page state in a way that exposes **new unsafe elements** (e.g. opening a cookies-settings dialog)
    Or
    b. If nothing like what a) discusses seems available on the current page, **explore the rest of the site to find another page with unsafe elements**.
    Or
    c. If you can't think of any other areas of the site to explore for pages that might have unsafe elements, bundle together all batches for the current site **(see next section)** and move on to the next website you've been assigned.

In general, if you aren't sure what a UI element does, please put your mouse cursor over it and read the element's hover text (this also applies for the "Action Capture Reminder" static text element).

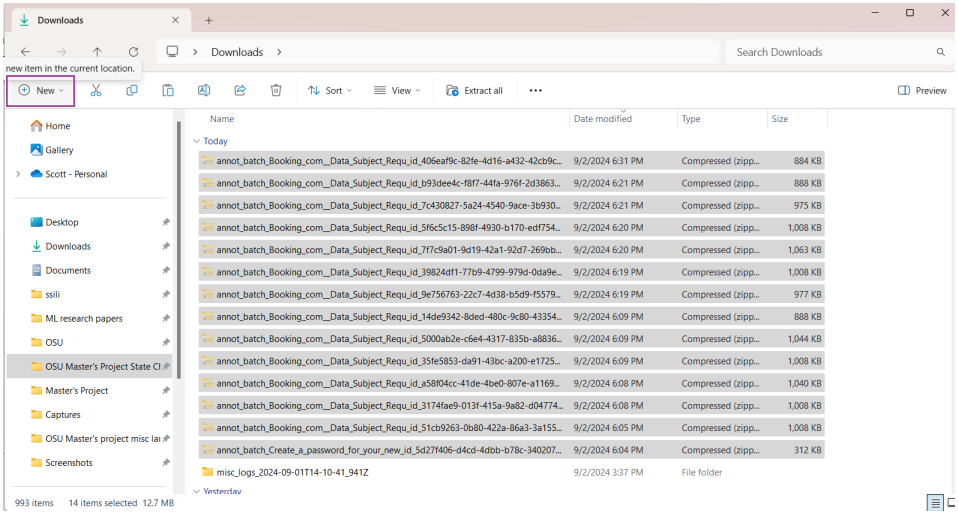
🗂️ Collecting annotations together

Once you think you've identified all of the state-changing actions for a site,

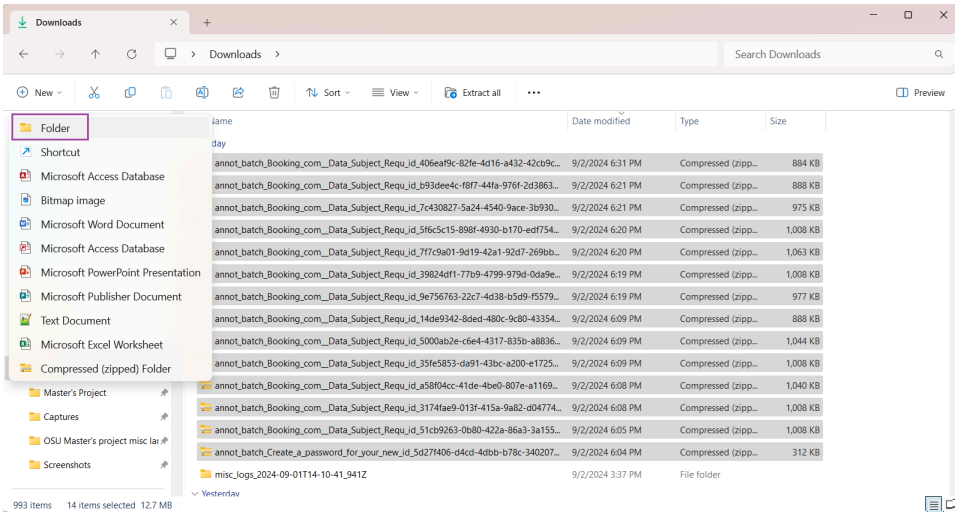
1. Please put all of the batch zip files for that site's actions into a single folder for that site:

    a. Create a new folder

i.

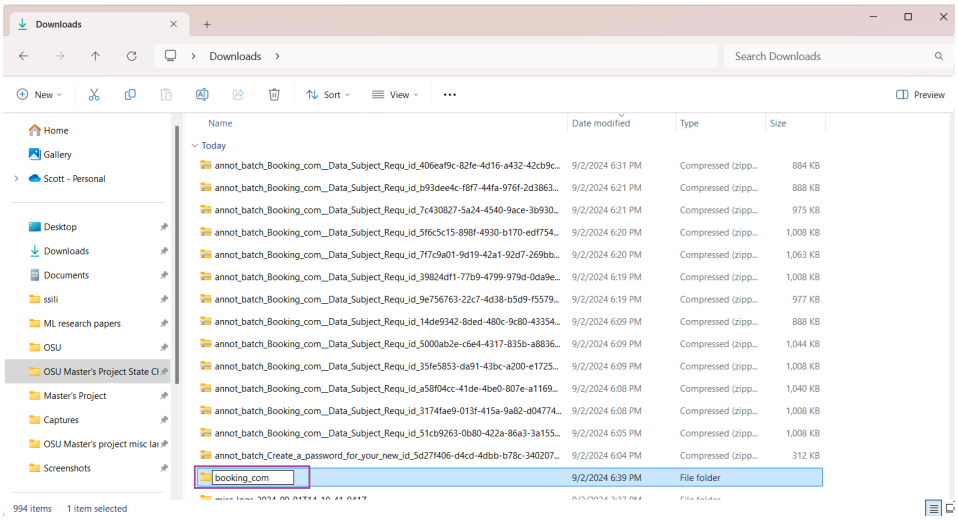


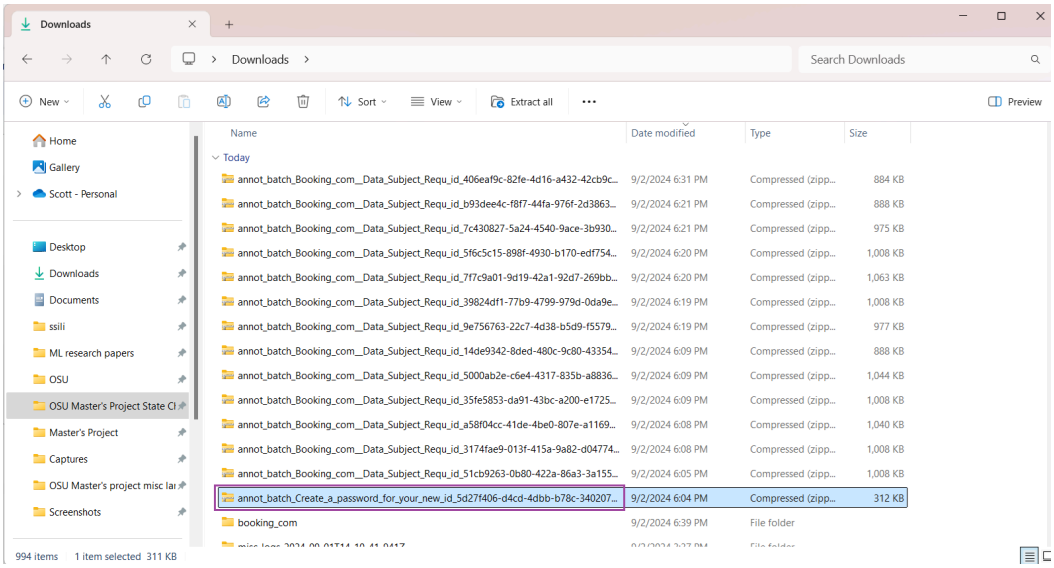
ii.



iii. Type in the site name (please be careful not to type a different name, since this can cause problems at the moment of gathering the data) and then press enter. For example, if you made annotations on the

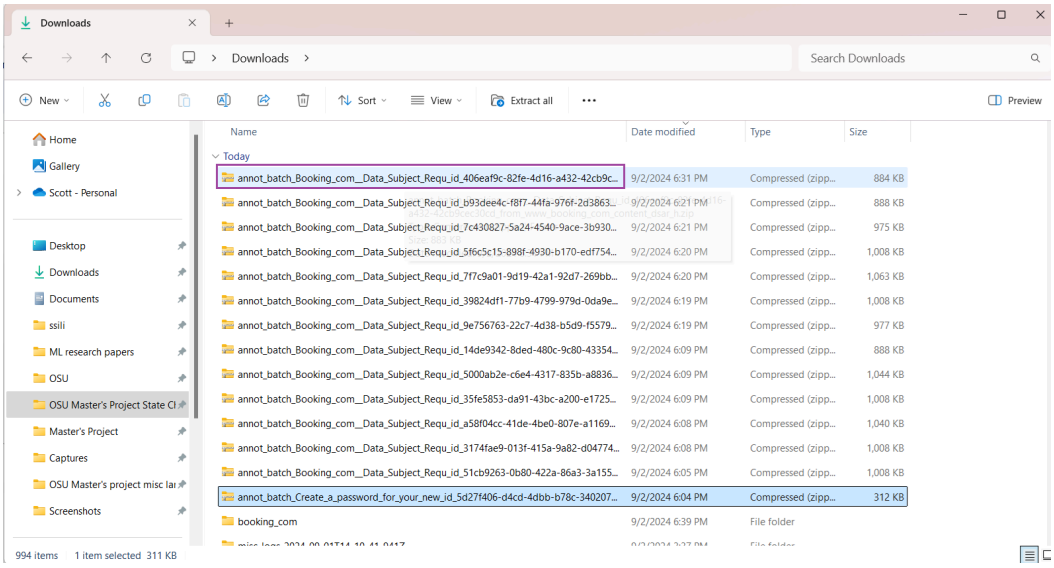
Booking.com website, name the zip as **Booking.zip**.



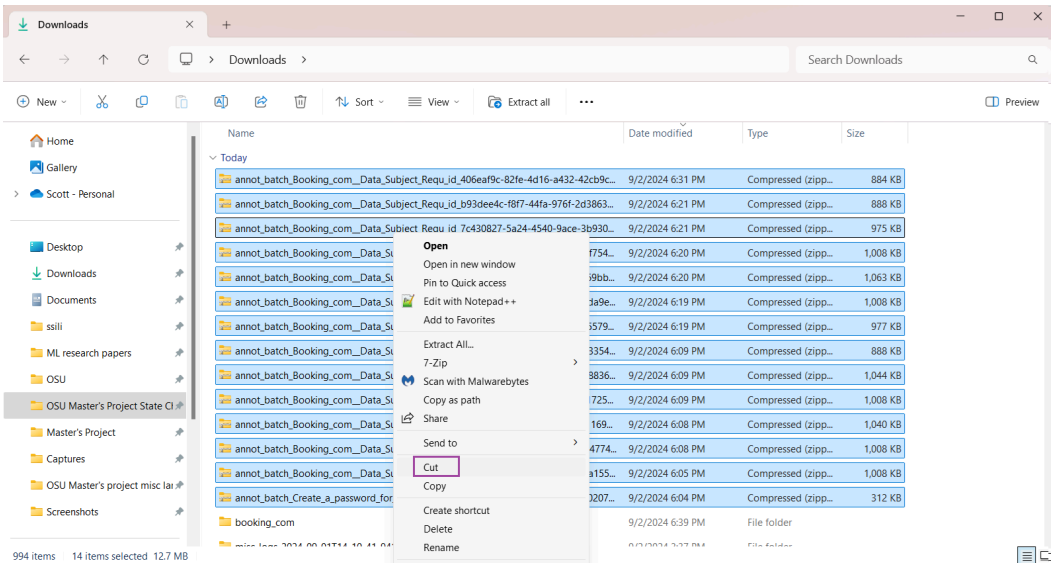
b. 

Select all of the batch zips by left-clicking on the last one (screenshot above),

then holding shift while left clicking on the first one (screenshot below)

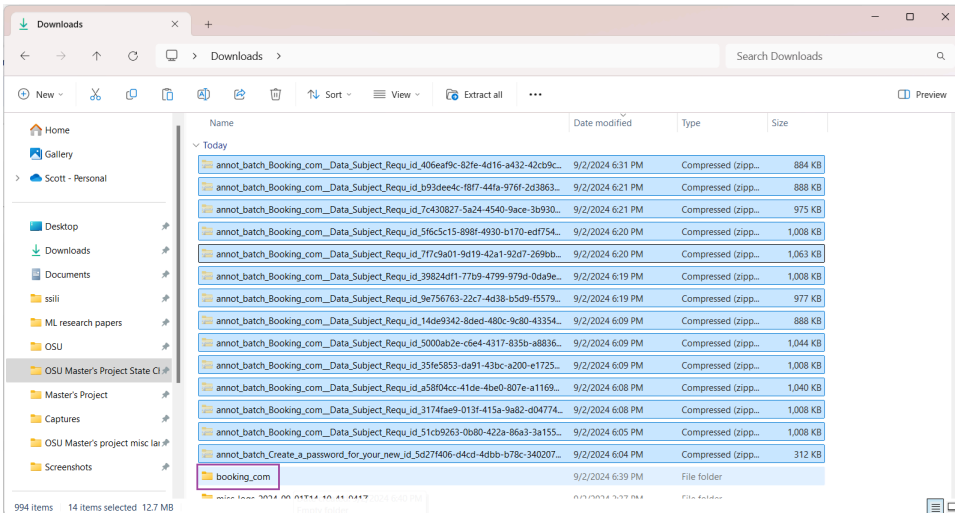


c. 'Cut' all of those selected annotation batch zips so they can be moved to a new folder

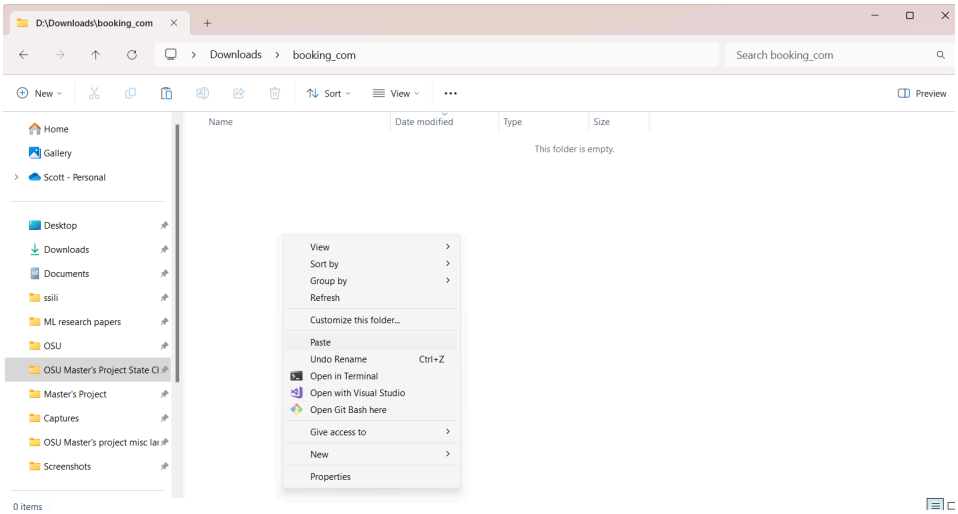


d. Move the selected annotation batch zips into the new folder

i. Click the new folder to go into it

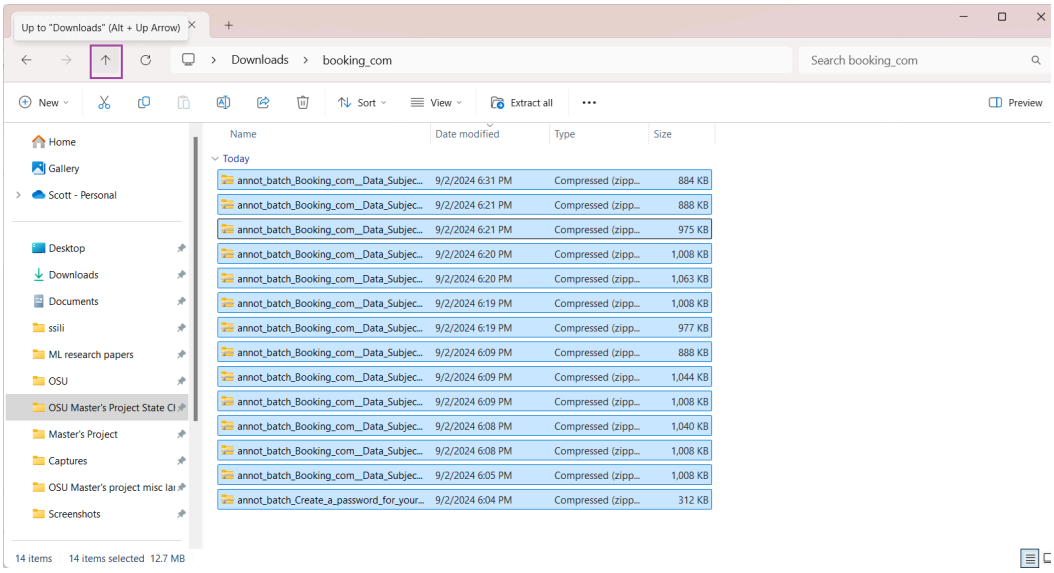


ii. Paste the ‘cut’ batch zips into the folder

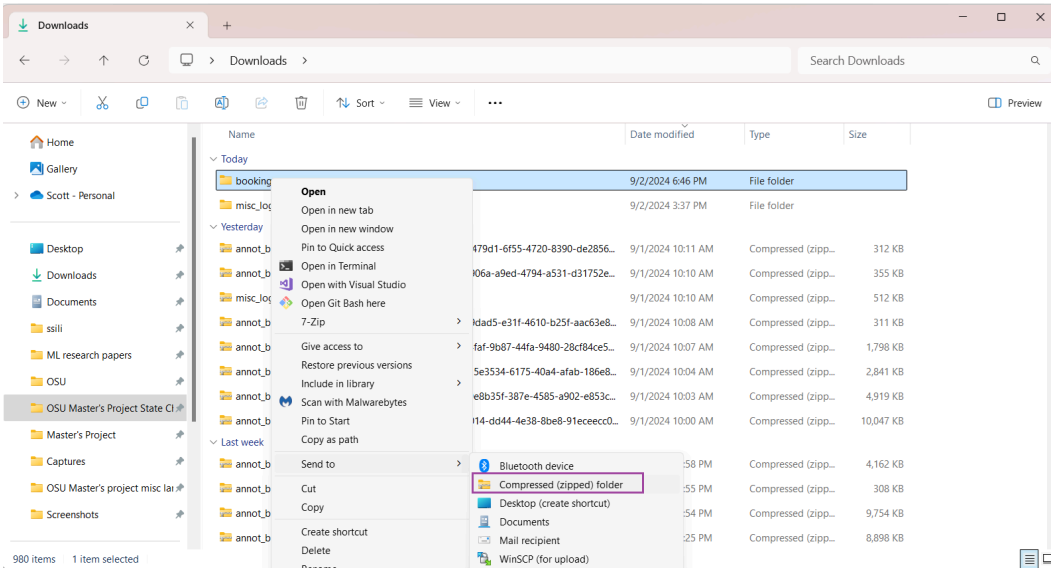


2. Compress that site’s folder into a single large zip archive

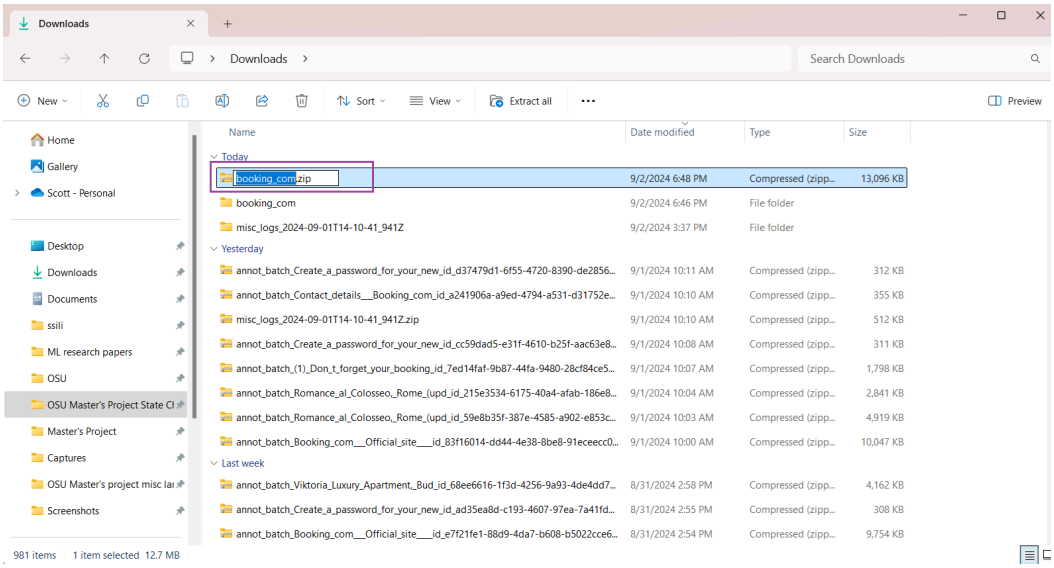
a. Go back up to the Downloads folder



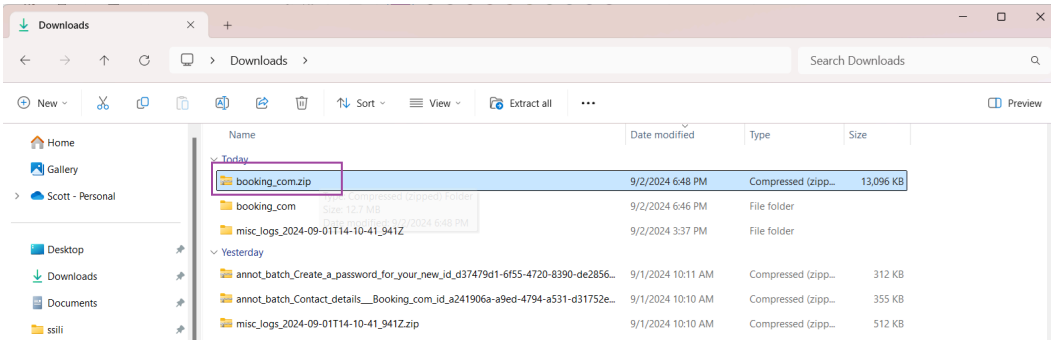
b. Right-click on the folder for this site and choose “Send to” then “Compressed (zipped) folder”



c. Press enter to finalize the name of the zip archive for all of the site's annotations. Put the name of the website while renaming the zip.



d. Expected result



🚨 What happens if you received Login Credentials within a task?

Guidelines for Tasks Involving Login Credentials

When working on tasks, you may encounter different scenarios based on whether login credentials are provided. Here's what you need to know:

1. **Task Categories:**

    - **Guest Mode:** Tasks where no credentials are provided. You are limited to the public areas of the website.

- **Login Required (With Verification Code):** Tasks where credentials are provided and a verification code is necessary. Refer to the Google Sheet for the proper code.
- **Login Required (Without Verification Code):** Tasks where credentials are provided without the need for a verification code.

2. **Using Provided Credentials:**

   - **Access Restricted Areas:** If login credentials (username/password) are supplied, use them to access the website's restricted sections where additional State-Changing Action (SCA) elements might be present.
   - **Annotation:** Once logged in, document and annotate any elements that appear in the environment.
   - **Verification Codes:** Sometimes you'll need a verification code. In these cases, open the designated Google Sheet and find the sheet corresponding to the email address you used to log in. Then, scroll down to the bottom of that sheet to retrieve your verification code.

     Verification Code Access Sheet

   - **Failed Logins:** If the credentials do not work, skip the task rather than attempting multiple logins, as repeated failed attempts may lead to account blocking.

3. **When No Credentials Are Provided:**

   - **Guest Mode Analysis:** Only review and analyze the public-facing elements of the website. Do not attempt to access any restricted sections.

4. **Security and Privacy:**

   - **Strict Compliance:** Under no circumstances should you alter real-world data, submit forms, or make any unauthorized changes while logged in.

- **Recording Risk Levels:** Be aware that website elements might behave differently when a user is logged in (e.g., a “Login” button might automatically sign in or redirect). Always record the highest possible risk level in these scenarios.

🎥 Installation Tool Video Tutorial

This tutorial provides a step-by-step guide on installing the required extension. It walks through downloading, configuring, and setting up the extension for proper functionality. The video ensures users can complete the installation process smoothly and troubleshoot common issues.

🎥 How to use Annotation Tool

This tutorial demonstrates how to utilize the annotation tool effectively. It covers key features, such as adding, editing, and managing annotations, as well as best practices for enhancing collaboration and clarity in your work. The video provides practical examples to help users maximize the tool’s potential.