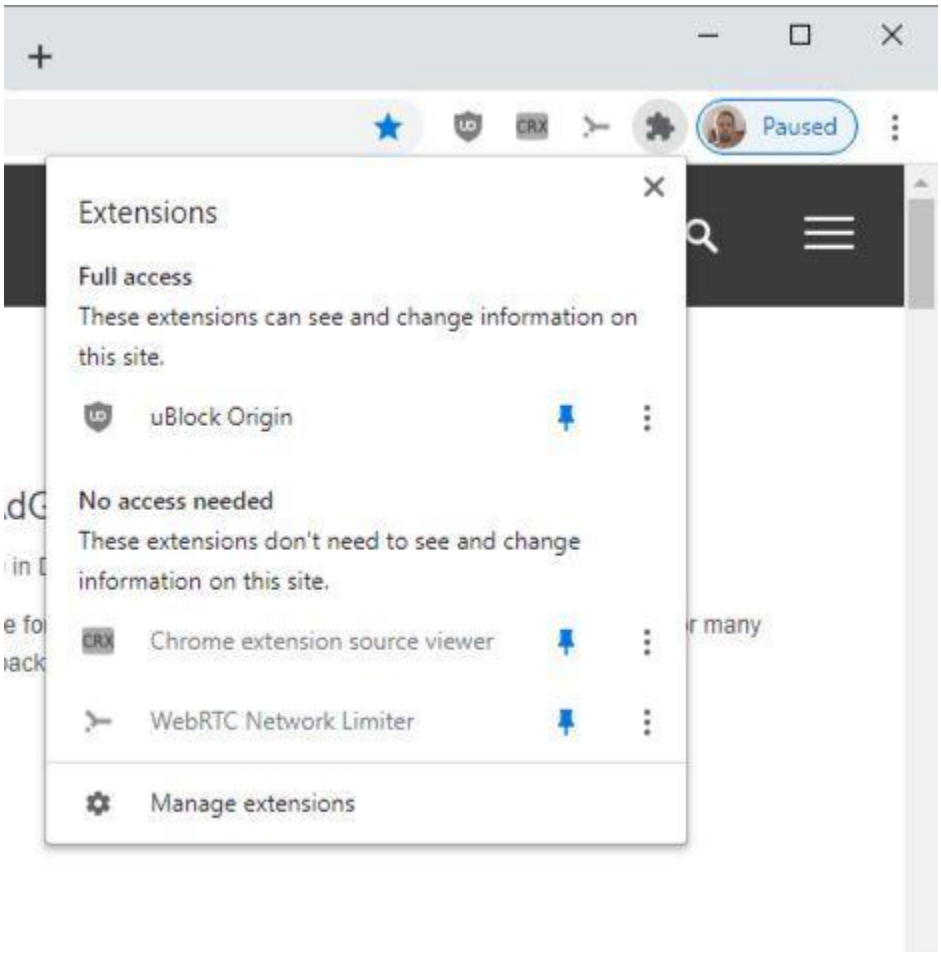
Instructions - SETUP

1. Download this Chrome extension

2. Unzip it in a folder of your preference

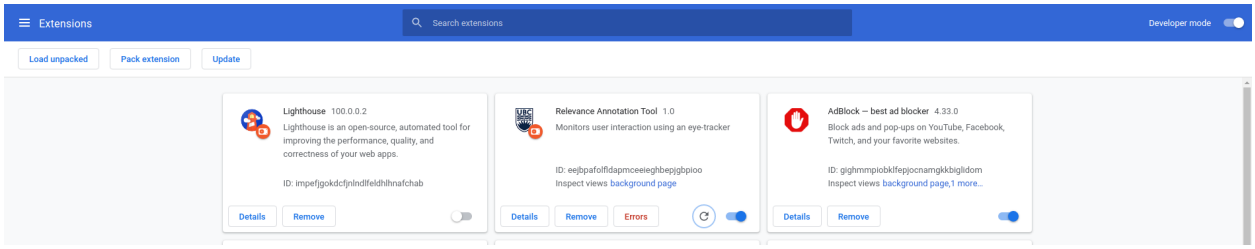
3. Download and install the Google Chrome browser

4. Once the browser has been installed, locate the browser extension panel



5. Enable developer mode (right-hand side)

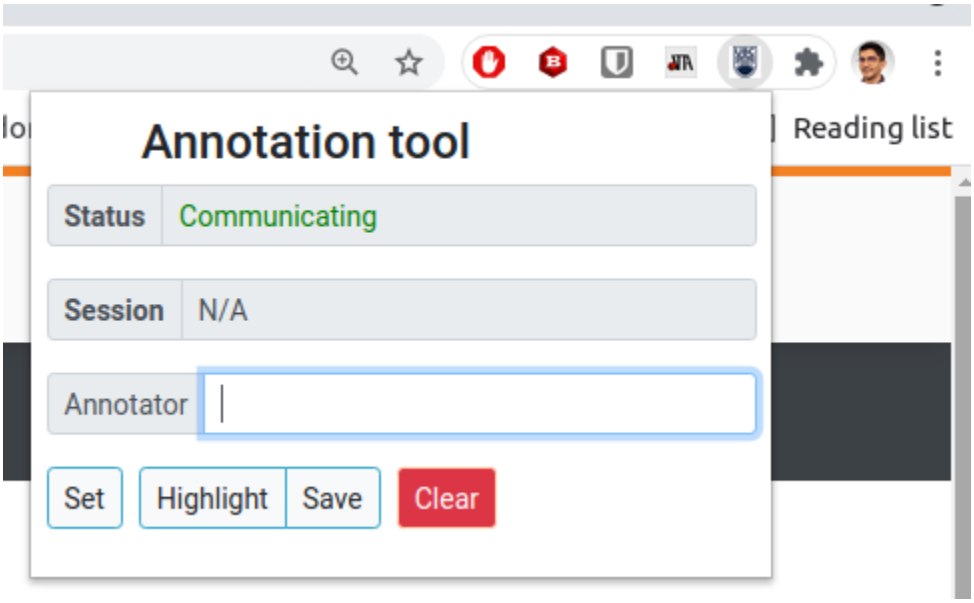
6. Load the unpacked extension (left-hand side)



6. If loaded successfully, you should see the **annotation tool extension** (the one with the UBC icon)

Instructions - USAGE

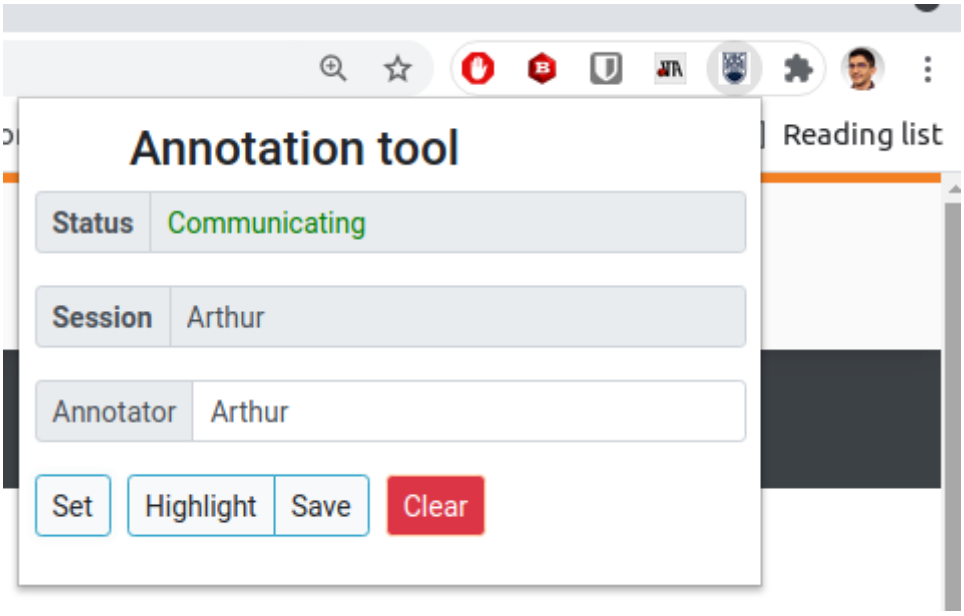
1. Click the extension icon on the top right corner



tip: status says "communicating" but **all data is stored locally**. This is reminiscent of an older version of the tool that used a Tobbi eye-tracker plugin

2. Input a name identifier (1 word) and click set

It's OK to use your first or last name, this information will not be publicly available



3. Session should now show your name. In case of mistakes, you can click clear to start over

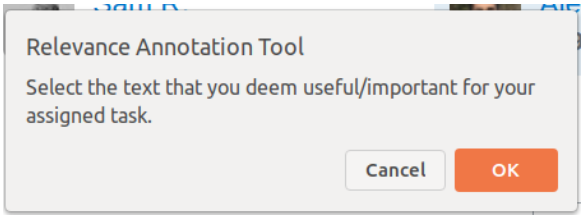
4. Open the links shown on a task in new tabs

> **IMPORTANT:** you can swap the order of steps 5 and 6 if that better suits your workflow, i.e., highlight first and write an answer later

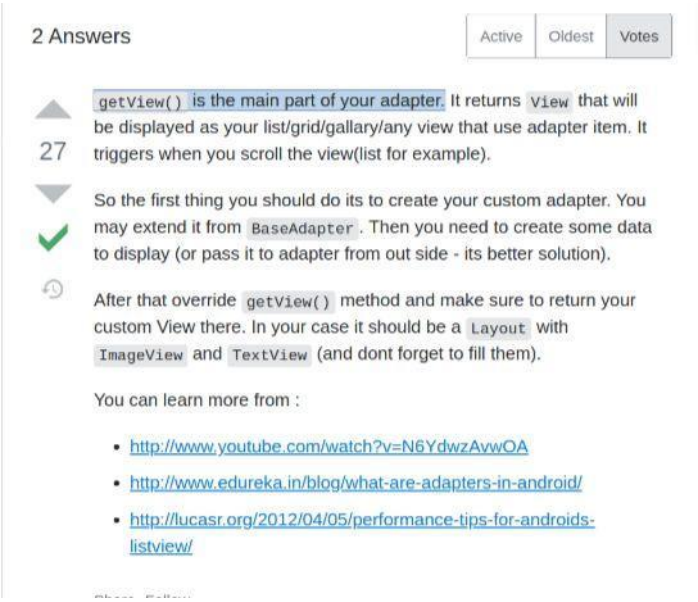
5. Read the content of each link and write a small answer (less than 250 words) on how to complete the task.

> **TIP:** A single short sentence is preferred
>
> E.g.: this is a config issue, check `setPriority`

6. Once you wrote your answer, revisit each link and click **highlight** on the extension menu. An alert will remind you of instructions of what you should highlight



7. Hovering over a sentence will display it in **blue**.



Click on the sentence to highlight it. Highlighted sentences are shown in **orange**.

`getView()` is the main part of your adapter. It returns `View` that will be displayed as your list/grid/gallary/any view that use adapter item. It triggers when you scroll the view(list for example).

So the first thing you should do its to create your custom adapter. You may extend it from `BaseAdapter`. Then you need to create some data to display (or pass it to adapter from out side - its better solution).

After that override `getView()` method and make sure to return your custom View there. In your case it should be a `Layout` with `ImageView` and `TextView` (and dont forget to fill them).

Click on any highlighted sentence to deselect it.

> **IMPORTANT:** while in line codeblocks are OK, please ignore code blocks such as the one below
>
> **e.g.:** on the figure, it is OK to highlight "here is my code ..." but you don't need to highlight the code block itself

Here is My Code Sample For TouchImageView.

```
public class TouchImageViewSample extends ImageView {

private Paint borderPaint = null;
private Paint backgroundPaint = null;

private float mPosX = 0f;
private float mPosY = 0f;

private float mLastTouchX;
private float mLastTouchY;
private static final int INVALID_POINTER_ID = -1;
private static final String LOG_TAG = "TouchImageView";

// The ‘active pointer’ is the one currently moving our object.
private int mActivePointerId = INVALID_POINTER_ID;

public TouchImageViewSample(Context context) {
    this(context, null, 0);
}

public TouchImageViewSample(Context context, AttributeSet attrs) {
    this(context, attrs, 0);
}

private ScaleGestureDetector mScaleDetector;
private float mScaleFactor = 1.f;

// Existing code ...
public TouchImageViewSample(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
    // Create our ScaleGestureDetector
    mScaleDetector = new ScaleGestureDetector(context, new ScaleListener());

    borderPaint = new Paint();
```

IMPORTANT: some times, the tool will have a highlight block with more than one sentence. This is a known bug, try to hoove closer to the sentence text and it will show individual highlights for each sentence

IMPORTANT: Links are not clickable. Stick to the content of the pages provided for each task.

If you believe that an external source may be useful as when someone directs a reader to a resource `"you should use a RecyclerView instead of ListView."` you can still mark the sentence if you believe that the sentence is useful.

8. Once you are done, click the **save** button on the extension menu and confirm that you have finished highlighting. This will **download a json** with your answers.

Once you have highlighted all the tasks, zip your jsons and either:

- email your answers to **msarthur@cs.ubc.ca**
- upload your zip to Qualtrics at the end of the survey