The main new piece I added was the logic for viewing a log entry in an overlay when the user clicks a table row. I split that into two responsibilities:

1. **Row click handler (event delegation on the table)**
    - This code listens for clicks on the table, figures out which <tr> was clicked (and ignores the header), grabs the <td> values, and then passes that data into another function.
    - This part doesn't care how the data is displayed, it just collects it.
2. **viewLogDetails(date, location, firearms, ammo, weather, notes, rating, image)**
    - This function is responsible for actually building and showing the overlay in the middle of the screen.
    - It takes the values from the row as arguments (date, location, etc.) and uses them to build the UI.
    - It also shows the uploaded/linked target image, which was the main reason for adding this feature.
    - It doesn't need to return anything because it updates the DOM directly.

So the pattern is: one function to get the data , another function to display the data. That keeps things consistent with how the rest of the page was already modularized.

2. Why I converted this into functions

I kept this new feature modular for the same reasons as the rest of the page:

- Separation of concerns: the click handler only worries about "what row was clicked and what's in it?", and viewLogDetails(...) only worries about "how should this look on screen?"
- Readability: when someone reads the code later, seeing viewLogDetails(...) with all the arguments makes it obvious what's happening.
- Easy changes later: if I want to change the layout of the overview (for example, add "time spent" or change how the image is sized), I only edit the display function, not the click logic.

- Reusability: later on, if I add a “View” button somewhere else (not in the table), I can still call viewLogDetails(...) with the same parameters.

Basically, this feature is an enhancement to what the page already did — before you could *read* the log in the table, now you can view it in a nicer, focused way, especially to actually see the target photo.

3. Possible small follow-up functions (within this same feature)

I could break the display part down even more if needed, for example:

- buildLogOverlay() / closeLogOverlay()
    - Purpose: keep the overlay creation and closing logic in their own functions instead of mixing it into viewLogDetails(...).
    - Params: buildLogOverlay(...) would take the same info as now; closeLogOverlay() wouldn’t need params.
    - This would make it easier to change styling or add transitions without touching the data part.

These aren’t new features, they’re just further cleanup of the feature I just added.