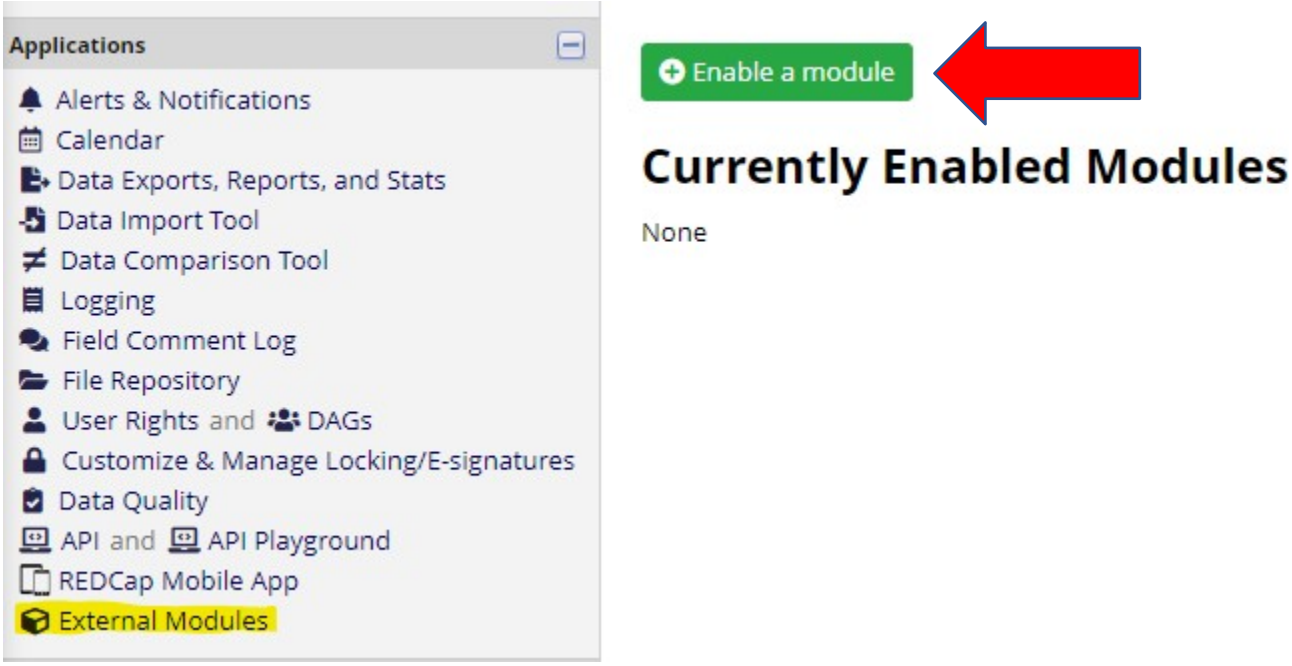
SOPs for REDCap External Module: CheatBlocker

A. Add the CheatBlocker module to your REDCap project:

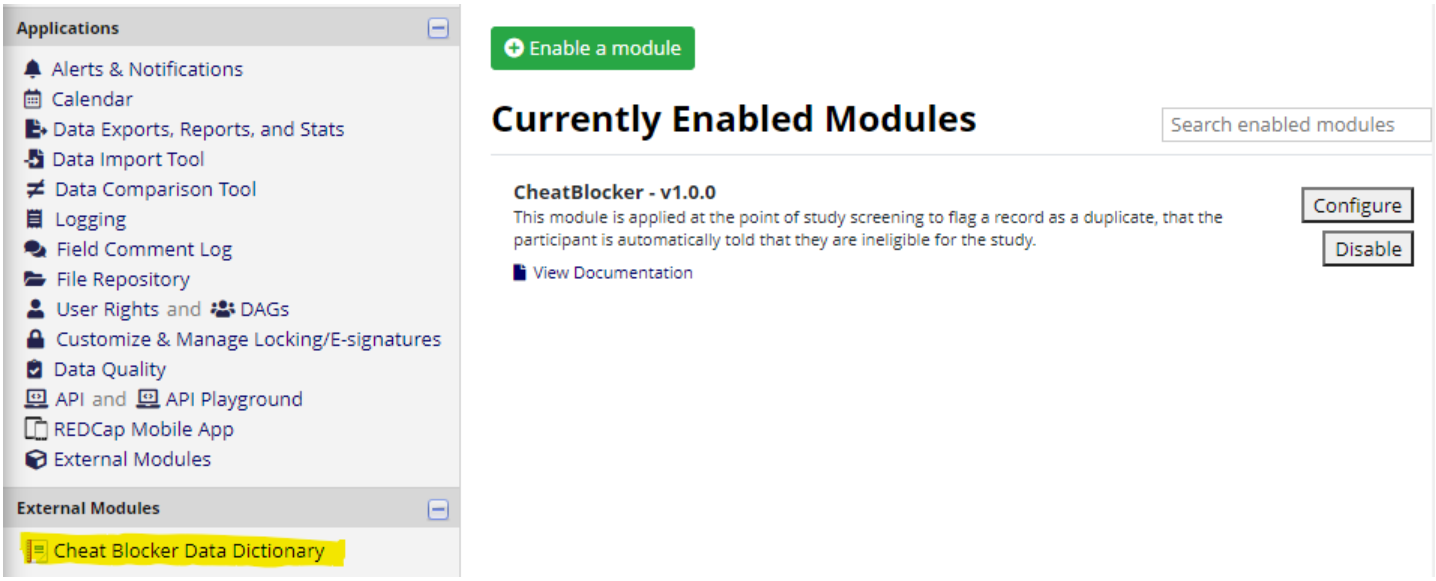
1. Open your REDCap project that you want to use the CheatBlocker module in.
2. In REDCap, underneath the "Applications" header on the left-hand sidebar, click "External Modules."
3. Then, click the green button that says, "Enable a Module."



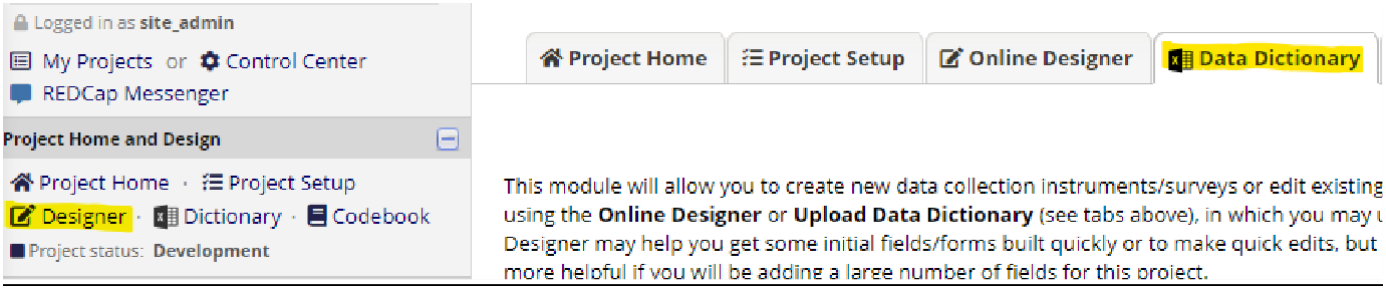
4. Search for "CheatBlocker" and Select the "Enable" button.

B. Add the variables to your project that will be used to customize the CheatBlocker module:

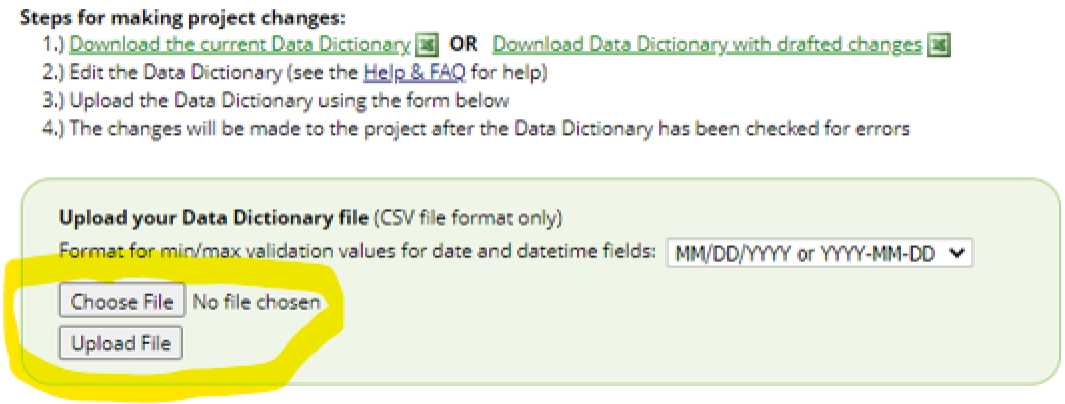
1. On the left -hand sidebar in the External Modules section, click "Cheat Blocker Data Dictionary" to download a .csv file of variables you will need to use this module. Save this "CheatBlocker_data_dictionary.csv" file to your computer.



2. On the left-hand sidebar under the header “Project Home and Design,” click “Designer” and then open the “Data Dictionary” tab.



a. If your project has no forms/variables in it yet, click “Choose File” button near the bottom of the page (shown below), and then upload the .csv file you just saved to be your project’s Data Dictionary.
b. **OR,** if you already have forms/variables in your project,
    i. click the green “Download the current Data Dictionary” link shown below,
    ii. open your downloaded Data Dictionary and the .csv file you saved from above,
    iii. then copy and paste the rows from the new .csv file into your existing Data Dictionary
    iv. Finally, re-upload your Data Dictionary with the cheat blocker variables now added, by selecting the “Choose File” and then the “Upload File” buttons circled below.



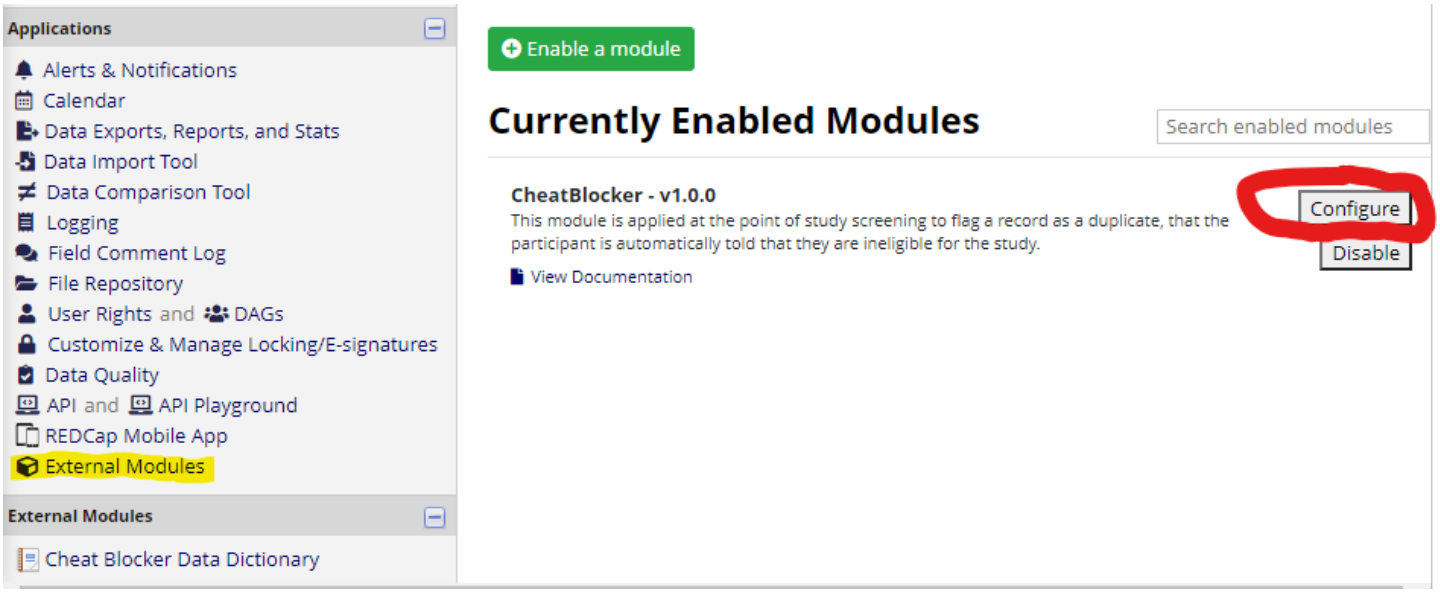
---

c. **Important tips to allow CheatBlocker to work correctly:**
    i. The module must run in the first event in the project. IE, if your REDCap project has various events, the module will only check for cheaters in the FIRST EVENT, so make sure the instrument with the fields for the module is placed in the first event in the project.
    ii. You must add or move any other variables that you wish to use to “catch” duplicates to the “Demographics” form that is part of the CheatBlocker_data_dictionary.csv file; the module cannot use variables found on other forms in the project. It is OK to have demographic information on other forms, if those variables are not needed to identify duplicates.
    iii. Do **not** re-name the form or **any** of the variables from the CheatBlocker_data_dictionary.csv file, as these are used behind the scene by the module, and it will not work correctly if you change these names. You can however *delete* demographic variable rows (DOB, Race, BMI etc.) that you do not need.
    iv. Mark all added variables that will be used as criteria to catch duplicates as “Required” so that blank fields do not show up as matches in the duplicate check.

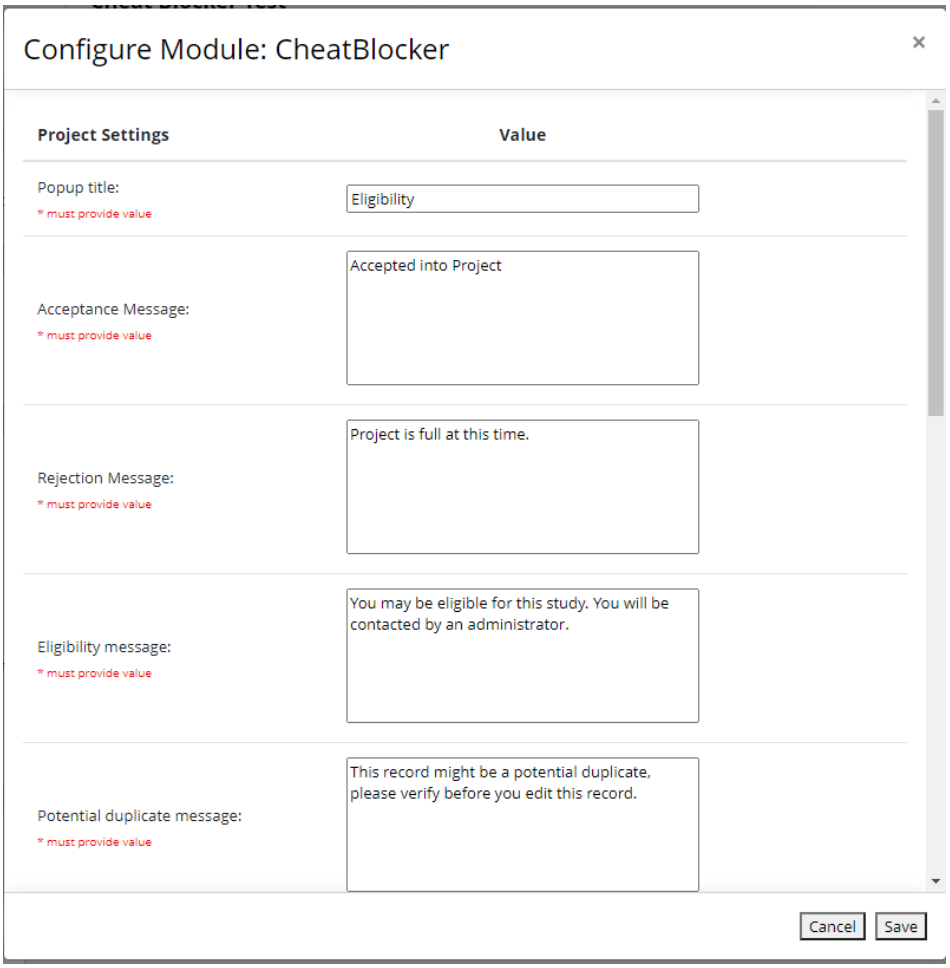
3. After you upload the new data dictionary, you can also make changes to the project by using the Online Designer, also under Project Setup. If you are not familiar with creating instruments and variables in REDCap, here is a short tutorial provided by REDCap.

C. Customize the module for your project:

1. Under “Applications” on the left-hand toolbar, click “External Module” and then click the “Configure” button next to the “CheatBlocker - v1.0.0” module.

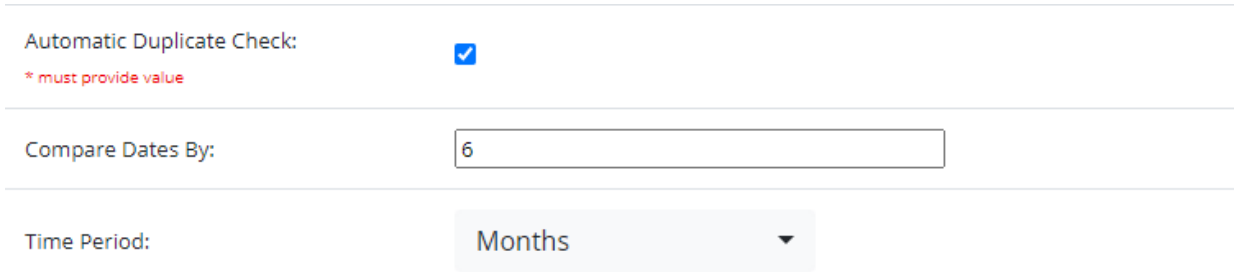


2. The dialogue box below will then open:



3. Customize the messages you want to pop up when someone fills out your survey to let them know if they are eligible or possibly eligible, based on duplicate criteria you will set up below. A screenshot of the default messages is above.

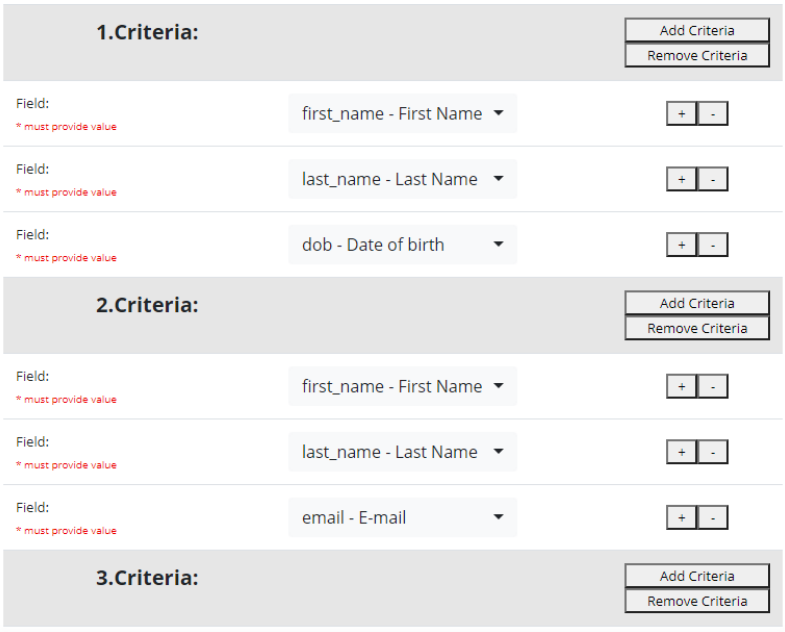
4. Select whether you want the Duplicate Check to occur automatically, or not (screenshot below).
    a. If you **check** the box beside "Automatic Duplicate Check," individuals filling out your survey will receive **either** the customized message that you put in the "Acceptance Message" or the "Rejection Message" boxes above; if someone gets the "Rejection Message" they will not be able to continue with the survey.
    b. If you leave the box **unchecked** beside "Automatic Duplicate Check," **all** individuals filling out the survey will receive your customized "Eligibility Message."
        i. Study staff will then need to open each new record as they come in and look at the "Potential Record IDs" and "Potential Failed Criteria" fields to see if the module identified the record as a potential duplicate with any other existing record(s) and on what criteria.
        ii. If the study staff decides that the record is a duplicate, they should select "yes" for the "Duplicate Check" question; if the study staff decides that the record should not be considered a duplicate, they would select "no" to the "Duplicate Check" question.



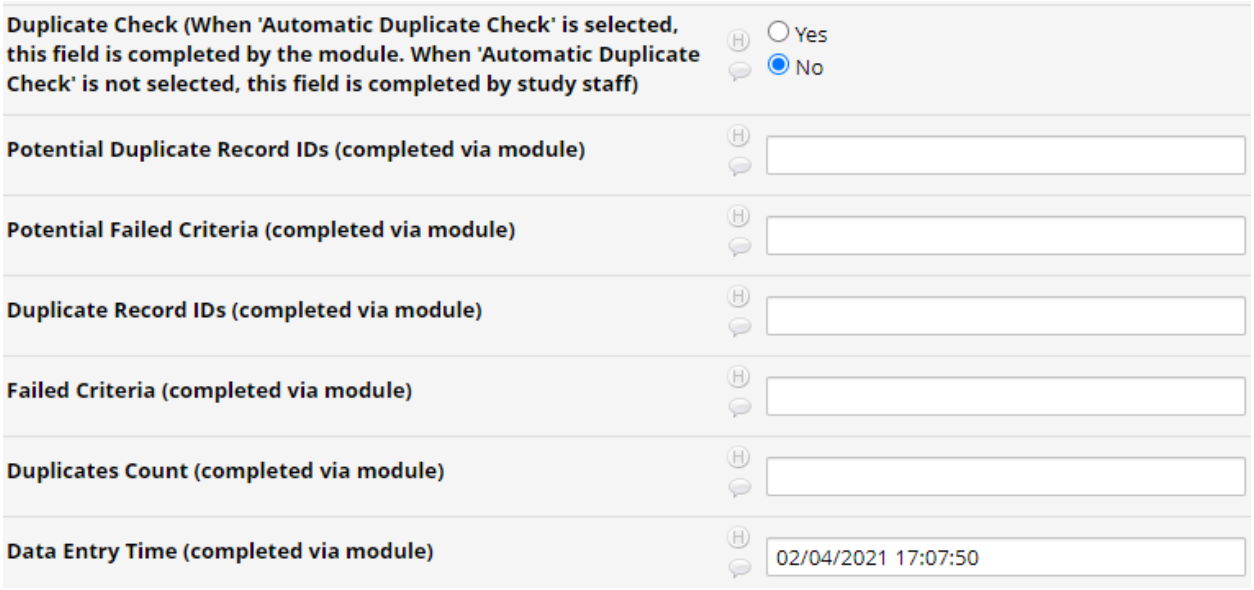
5. In the image above, you can also see the "compare dates by" field, which allows you to control which records are included in the cheat blocker check. In the scenario above, it would only check records that were submitted within 6 months from the day the screener was filled out.
    a. Using the "Compare Dates By" feature is optional and may be beneficial if you want to allow for changes in eligibility criteria over time within an individual (e.g., if a person was feeling fine a year ago, but now is feeling depressed, or if a person had uncontrolled high blood pressure a year ago but it is now under control with medication). If you leave this space blank, all records will be checked.

D. Finally, add your desired cheat blocking criteria to the project:

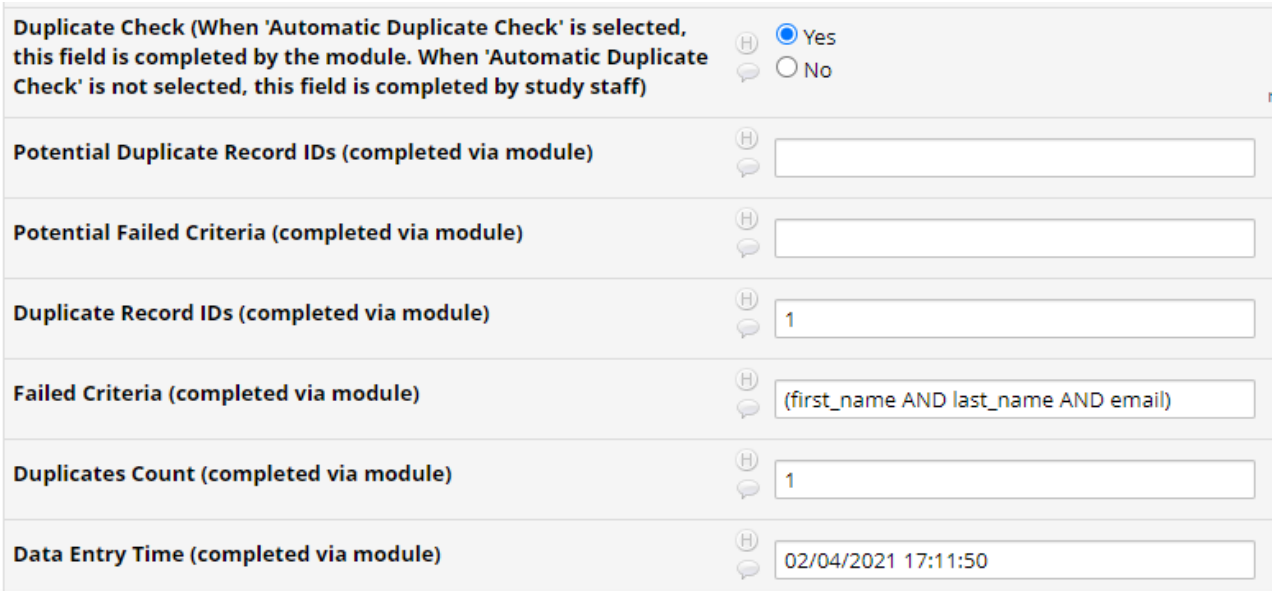
1. At the bottom of the page, you will enter your desired criteria to be used in checking for duplicate/cheat entries, by clicking the "Add Criteria" button on the right.



2. Under each criteria you will add the fields that, if duplicated, will result in a rejection.
    a. Within each set of criteria, click the “+” or “-” buttons to add or remove fields.
    b. The +/- buttons add “AND” statements.
3. To add a second set of criteria (an “OR” statement), you would click the “Add Criteria” button to the right of “1. Criteria:”.
    a. In the example above, it will reject entries with matching first name, last name, and date of birth (Criteria #1), OR those with matching first name, last name, and email (Criteria #2).
4. Be sure to click “Save” to save your criteria before closing out of the dialogue box.
5. You can add as many sets of criteria as needed. This will be the number of different combinations of matching fields that will result in a Cheat Blocker rejection. You can test this using practice records and seeing if the desired pop up message appears.
6. You can also open any record and look at the bottom of the Demographics form, under General Comments, to see if someone passed the duplicate test or not, and if they failed it, what other record(s) it matched
    a. Example of a Passed Duplicate Check:



    b. Example of a Failed Duplicate Check (and why it failed):



c. Example of a Potential Duplicate (when the box next to “Automatic Duplicate Check” is left blank):

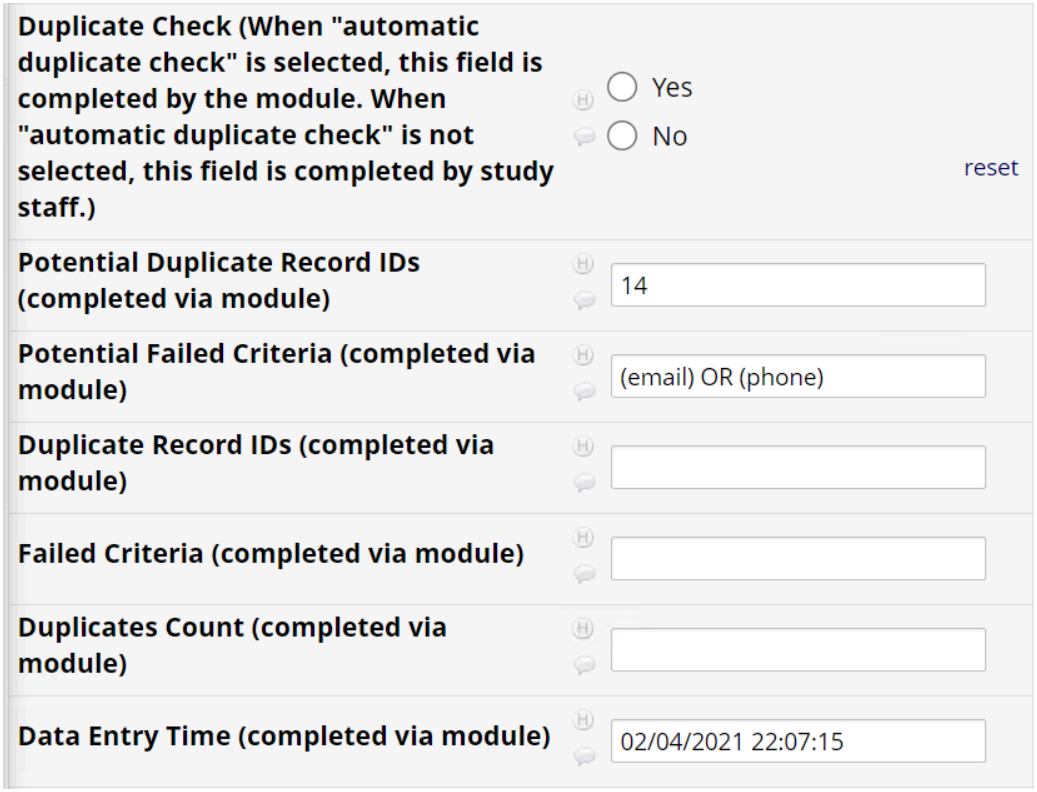


Figure 1: You can see the duplicate check field is blank, the staff will need to fill this out after making the decision if the ppt is a duplicate or not.

Once you complete the duplicate yes/no it will appear like the corresponding photo above (a or b).

Notes:

a. <u>The module must run in the first event in the project.</u> IE, if your REDCap project has various events, the module will only check for quotas or cheaters in the FIRST EVENT.

b. List of data dictionary fields and their uses:

- **i.** duplicate_check “Duplicate Check (When “automatic duplicate check” is selected, this field is completed by the module. When “automatic duplicate check” is not selected, this field is completed by study staff.” – when using delayed enrollment (automatic duplicate check will be left blank) you fill this out after making your determination
- **ii.** failed_criteria “Failed Criteria (completed via module)” – show which fields were duplicates with other records
- **iii.** duplicate_record_ids “Duplicate Record IDs (completed via module)” – show the record ids of the records that had duplicate matches with the current record
- **iv.** pot_duplicate_record_ids “Potential Duplicate Record IDs (completed via module)” – show the record ids of the records that had potentially duplicate matches with the current record
- **v.** duplicates_count “Duplicates Count (completed via module)” – shows the number of records that have duplicate matches with the current record
- **vi.** data_entry_time “Data Entry Time (completed via module)” – shows the time the entry was completed