Our new project is a type of dating application that takes your information to a database to create an account. Once your account is created you submit information about a person you want to connect with, for example their name, location of encounter, or date of encounter. This information is stored in a database and associated with your name. If you entered someone's name who also entered your name, then you are flagged as a match, and put in contact with each other. The users will be notified either by text or email, and if possible we would like to implement a messaging system within the application.

Features:

Hosting a Server:
We plan on hosting the server locally using, this means that the computer we are using for the presentation, must have MySQL and all of our HTML files on it in an organized manner. We plan to use SimpleHTTPServer, since we have exposure from class.

Web Pages:
Our app will have the user interact with the application through a series of web pages. These need to be designed with a user centered focus, and need to link together based on what the options the user selects.For example, create an account has a unique page, submit a match has a unique page and the main page needs to link to these separate page. This way, the user can follow a series of guided steps, and the app will be more user friendly.

Take User Input From a Webpage:
Our app will need to have a working HTML input form. In order for this to be successful, it must have a submit button, that links to a PHP script. The PHP file will link to the mySQL database and add the data using INSERT.

Pre-Check a Users Input:
Our HTML input form must be able to check a users information to see if it matches a desired format before it is sent to the SQL database. This includes checking that the email ends in "@colorado.edu", the first and last names match what the colorado email address should be. An Instagram username begins with "@". If the information does not match this format, the user should be prompted again for valid information.

Hosting the Database:
Our app requires a database of user answers and connections. To do this we plan on using mySQL to host the database. We will need three tables for the project, one table to hold all of our users input values for finding a match, another table holding all the users identifying information (for others to match with), and a table of found matches.

Searching Through the Database:
We need to have a SQL program that runs after each user input, which uses the SQL LIKE operator to compare the new information with all the old information searching for a match. If one is found, the SQL program needs to add the users to the match database and activate a program meant to contact users.

Require Multiple Inputs for Data:
Our app will need to accept multiple ways to connect people in the event that a person searching for a match doesn't know the first and last name or there are multiple users with the same name. The program that searches for the match will need to disregard certain bits of information or only pay attention to the data that has been entered correctly in order for this feature to work.

Notifying Users of a Match:
Our app needs to have a feature that activates a script (most likely in Python) that emails two users that a match was found. This program needs to get the contact information from the match table in the SQL database and email them both the others information. The stretch goal for this feature is to add a messaging system inside the app which users can opt into. The messaging system will automatically connect the match and the emails will be skipped. This avoids sharing excessive information and forces the pair into quicker contact.

HTML Web Pages Be Mobile Friendly:
Our web pages need to have a mobile friendly feature. In order to do this, our web page needs to be implemented with a “Responsive Design.” This means implementing features such as text, images, and spacing, to change dynamically with the screen size. This can be done with the HTML “vw” feature, as well as checking the media type in HTML.