URL: http://flip3.engr.oregonstate.edu:8560/

Zoe Gershuny
CS 340 – Intro to Databases
Portfolio Assignment

EXECUTIVE SUMMARY

There were many feedback and changes made to this project since the beginning of the term in March. The first major change was switching the project when I decided to work alone, which was changing the topic from Pokemon to Baseball cards. Due to the time left to work on the project from scratch, the project is scaled down to make it easier for a team of one to accomplish most if not all minimum requirements.

The Project Overview, Outline, diagrams, Data Manipulation, data dump, and Search and Display View pages were completed for Step 5 Draft submission on Piazza. Based on the feedback, I incorporated strategies to debug. Strategies are:

- Conditional display to confirm that the action was completed
- Conditional coding outputted in the console to pinpoint me where in the code was the error
- Chrome Developer Tools display error status so I have an idea what to look for: 404 vs 500.

For the Step 6 Draft submission, everything was completed except for INSERT to add the corresponding FK attributes. The feedback suggested areas for UI improvements, such as moving all INSERTS to single page and redirecting instead of going to the confirmation page. The changes made were making a single page for each table that will populate a list utilizing all CRUD functionalities. If not found, the user will be asked to add. The Full View under Display View has been modified so that the user can change associations. For example, the user can change the card's information to a different player or team. In addition, a form has been implemented so that the user can add all information at once for a new card.

OVERVIEW

The database is for the user's collection of baseball cards. The user will be able to see what they currently have in their collection from one to over thousands of cards.

It is possible to keep track of the collection on an Excel spreadsheet but as the collection grows it becomes difficult to manage over 4,000 rows. There are many collectors out there who have over 40,000 cards. The database will help to manage the size and speed and in addition to better search for exactly what you have in your collection. The database should be easily expanded to include a wide range of options, such as the oldest known baseball card is from 1860, multiple brands and series (similar to there are car companies who makes different model and series), and teams and players from all over the world at various organizations from college to international.

Each card is unique in that it has its own grade, type, condition, and other features. It is possible that two cards can be the identical in player, team, and grade, but the cards are rarely identical because of the condition. There are different factors that could determine a grade. For example, one card could have a soft corner, but another card has a scratched surface and be graded the same.

Not only the user will be able to see their collection, but also the list of teams and players in their collection. In addition, the user will be able to see the list of teams that the player played for and the list of players played for that team. This is important for collectors because there are many types of

collectors. Some collect certain players regardless of their team affiliations and others collect any players on one team. There are also other collectors that build sets. They will collect all cards from a certain brand's series. This information will help to facilitate the conversation so that the person would not have to physically look through their cards.

DATABASE OUTLINE

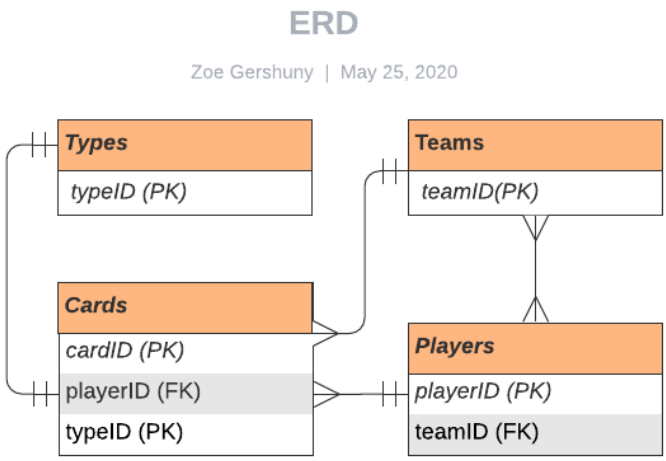
(I am responsible for the entire project)

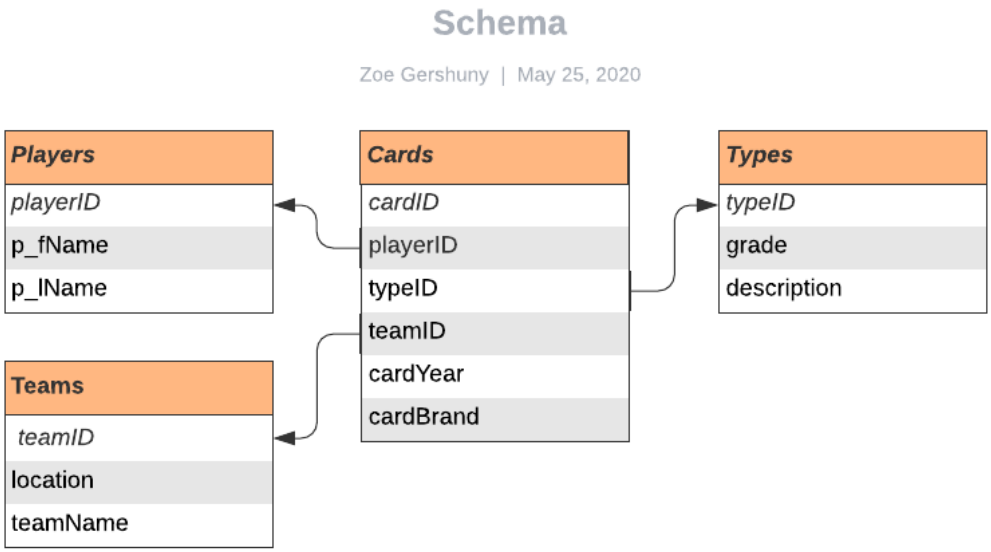
- **Players:** contains the list of all baseball players' first and last name that are on baseball card(s)
  - Attributes
    - *playerID*: int(11), auto_increment, not NULL, PK
    - *p_fName*: varchar(255), not null
    - *p_lName*: varchar(255), not null
  - Relationships
    - 1:M relationship between Players and Cards implemented with a playerID as a FK inside Cards
    - M:M relationship between Players and Teams implemented with a playerID and Teams' teamID as FKs inside Cards
- **Teams:** contains the list of all baseball teams' location and name that the players play for
  - Attributes
    - *teamID*: int(11), auto_increment, not NULL, PK
    - *location*: varchar(255), not null
    - *teamName*: varchar(255), not null
  - Relationships
    - 1:M relationship between Teams and Cards implemented with a teamID as a FK inside Cards
    - M:M relationship between Teams and Players implemented with a teamID and Players' playerID as FKs inside Cards
- **Cards:** contains the information about to the cards – year produced, brand that made the card, grade rating, description, and player and team on the card. Also serves as an associate table to make Players and Teams M:M
  - Attributes
    - *cardID:* int(11), auto_increment, not null, PK
    - *playerID*: int(11), nullable, FK
    - *teamID*: int(11), nullable, FK
    - *typeID*: int(11), nullable, FK
    - *cardYear*: int(11), not null
    - *cardBrand*: varchar(255), not null
  - Relationships
    - M:1 between Cards and Players is implemented with a playerID as a FK inside Cards
    - M:1 between Cards and Teams is implemented with a teamID as a FK inside Cards
    - 1:1 between Cards and Types is implemented with a typeID as a FK inside Cards
      - NOTE: This can be a M:M relationship as there are many types of cards. Various types include Rookie Card, Prospect Card, refractor, parallel, numbered, autographed, insert. It is also possible that one card can have multiple of these features. *However, due to the project specifications, one M:M is required to be implemented, which will be Players and Teams with Cards as the Associate Table.* Types will be designed so that it is 1:1 with Cards and meets the specifications.

URL: http://flip3.engr.oregonstate.edu:8560/

- **Types:** contains the grade and description for each card
  - Attributes
    - *typeID*: int(11), auto_increment, not null, PK
    - *grade*: float, not null
    - *description*: varchar(255), nullable
  - Relationships
    - 1:1 between Cards and Types is implemented with a typeID as a FK inside Cards
      - NOTE: This can be a M:M relationship as there are many types of cards. Various types include Rookie Card, Prospect Card, refractor, parallel, numbered, autographed, insert. It is also possible that one card can have multiple of these features. *However, due to the project specifications, one M:M is required to be implemented, which will be Players and Teams with Cards as the Associate Table.* Types will be designed so that it is 1:1 with Cards and meets the specifications.

DIAGRAMS

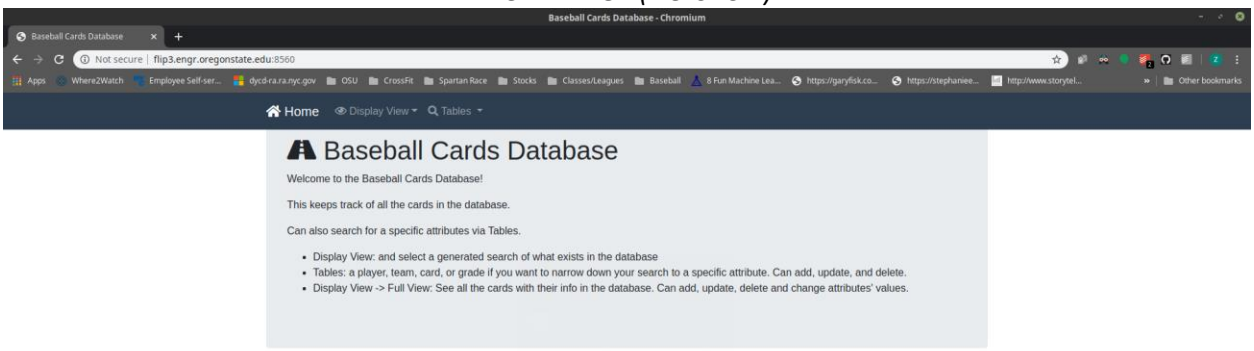




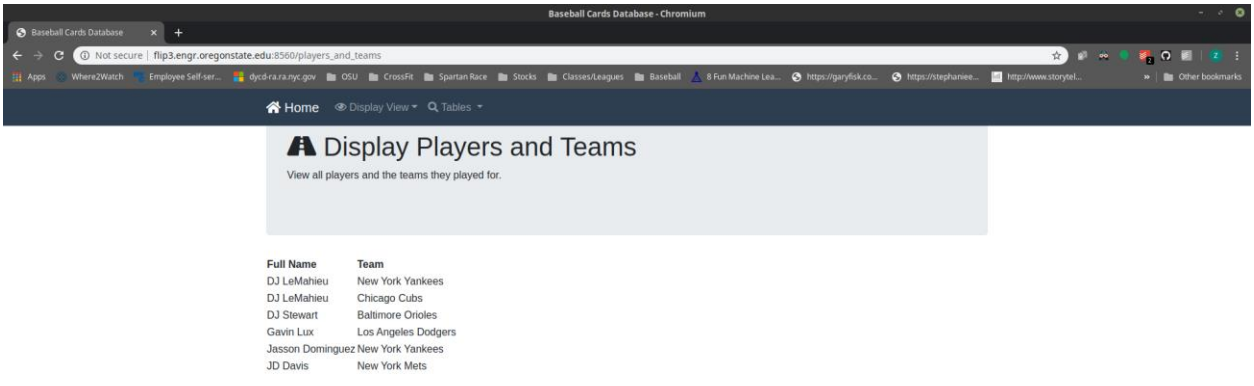
URL: http://flip3.engr.oregonstate.edu:8560/

UI SCREENSHOTS WITH INFORMATIVE TITLES

HOME PAGE (NO CRUD)

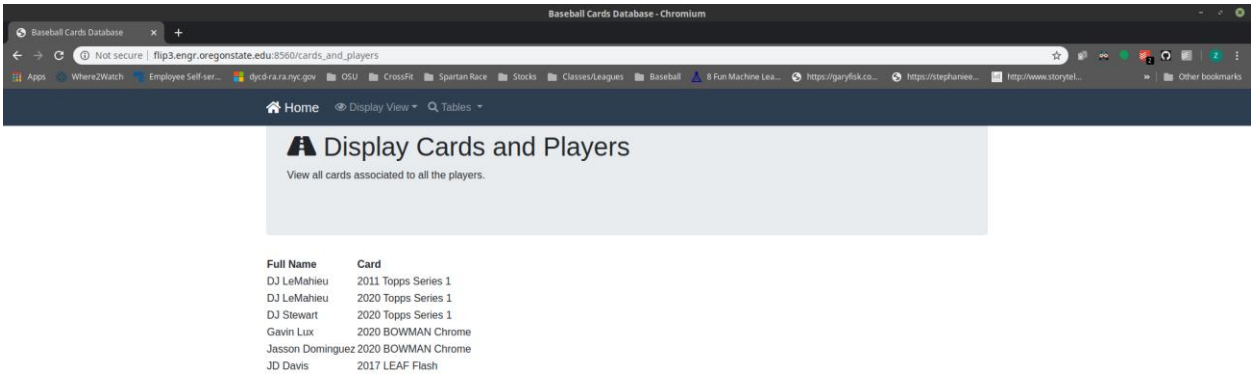


READ PLAYERS AND TEAMS

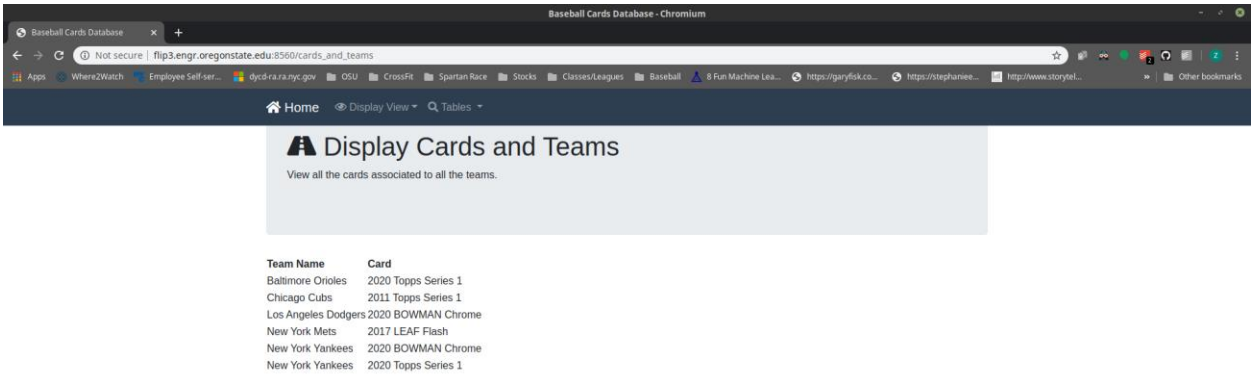


URL: http://flip3.engr.oregonstate.edu:8560/

READ CARDS AND PLAYERS

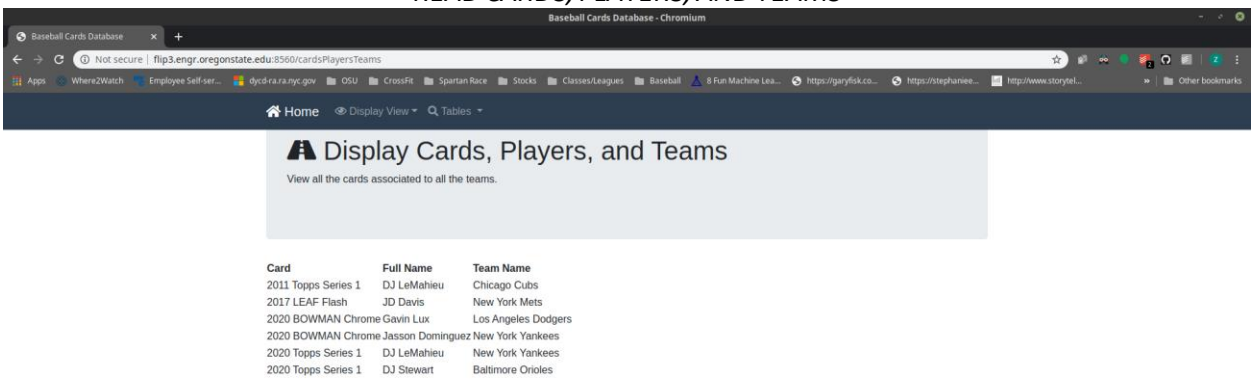


READ CARDS AND TEAMS

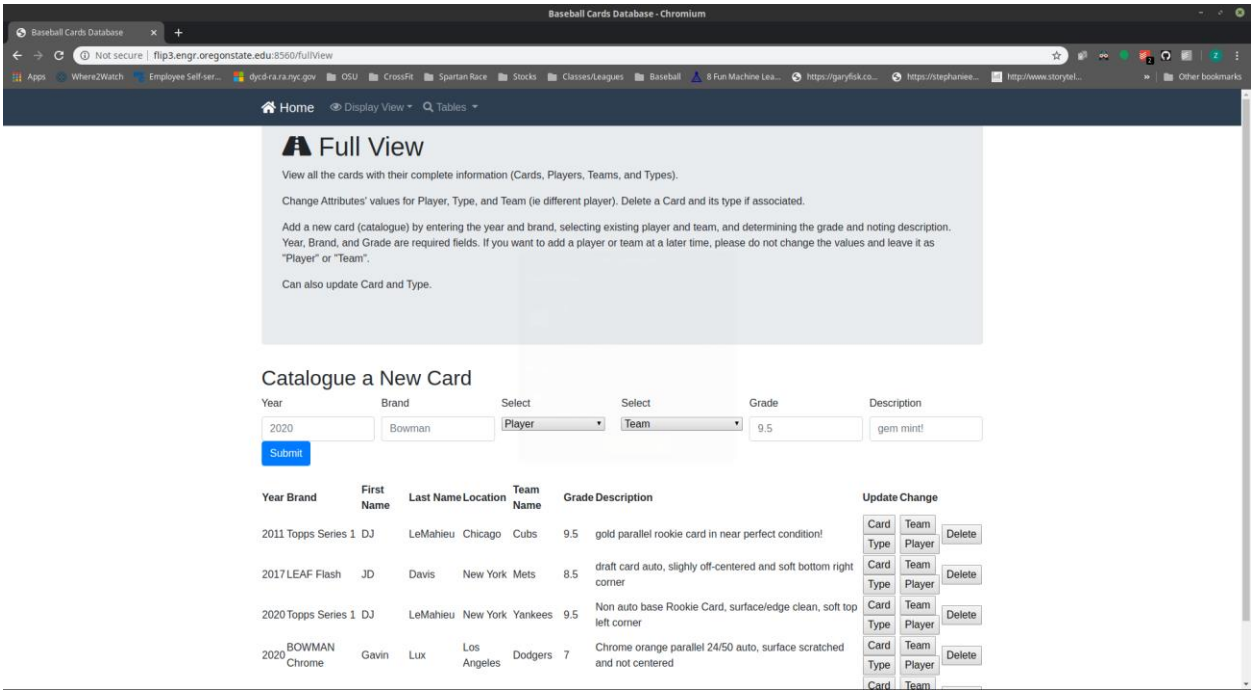


URL: http://flip3.engr.oregonstate.edu:8560/

READ CARDS, PLAYERS, AND TEAMS

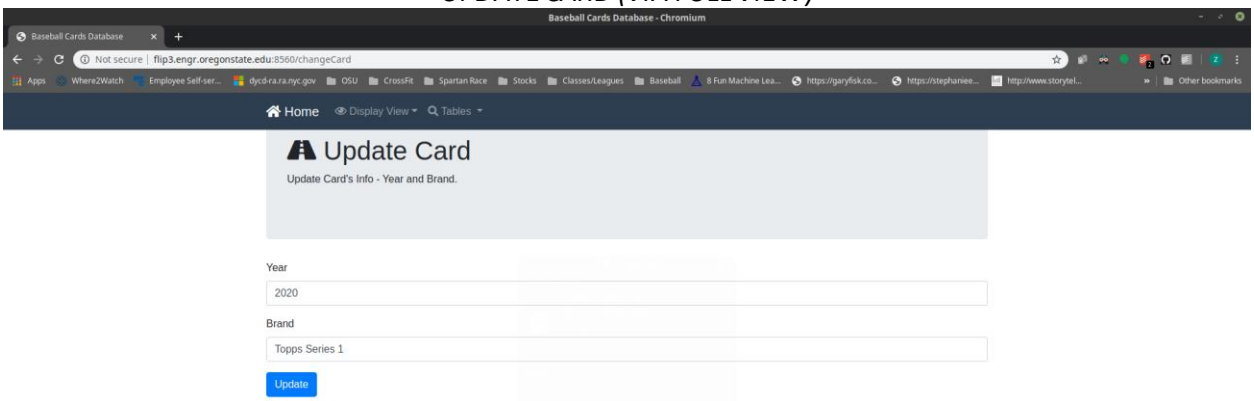


CREATE, READ, UPDATE, DELETE CARDS
READ, SELECT (AND CHANGE ATTRIBUTE VALUES) PLAYERS
READ, SELECT (AND CHANGE ATTRIBUTE VALUES) TEAMS
CREATE, READ, UPDATE, DELETE (IF ASSOCIATED WITH CARDS) TYPES
CREATE CARDS AND ADD THE CORRESPONDING FK ATTRIBUTES
PREPOPULATED DROPDOWN FOR PLAYERS AND TEAMS

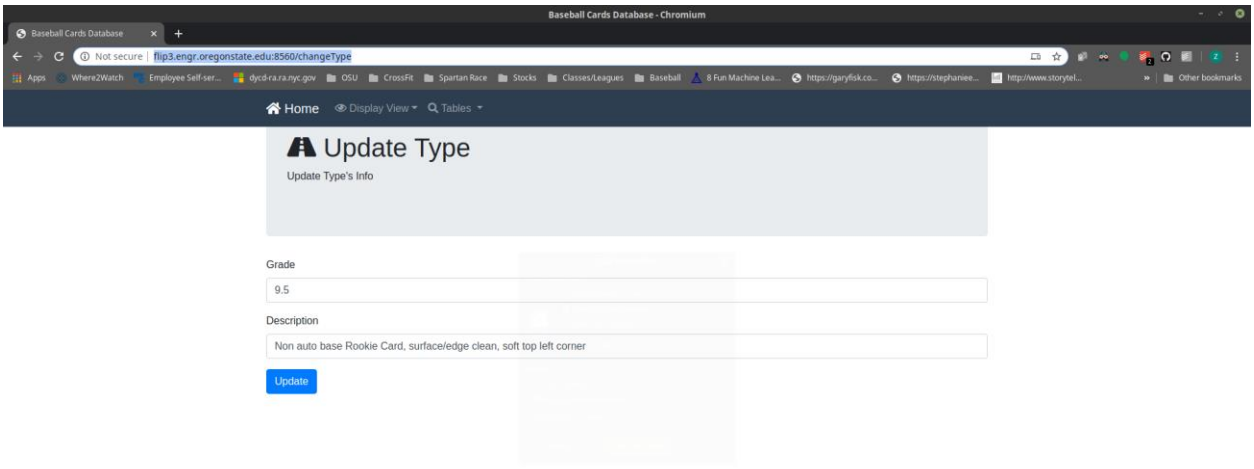


URL: http://flip3.engr.oregonstate.edu:8560/

UPDATE CARD (VIA FULL VIEW)

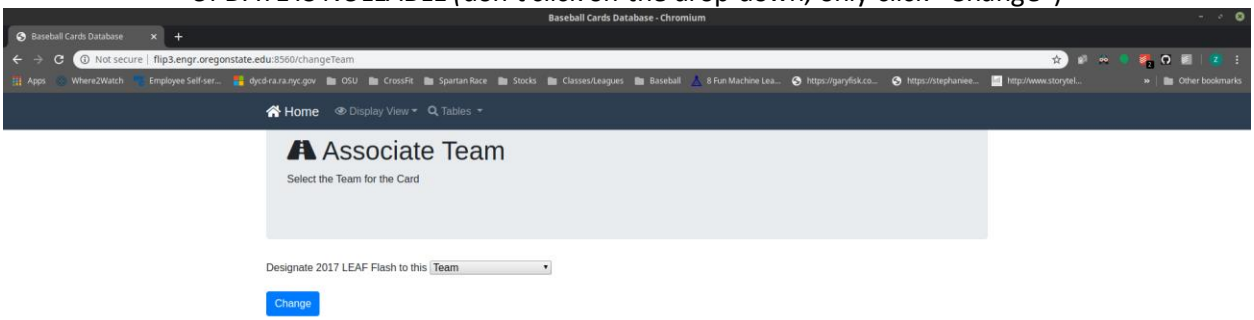


UPDATE TYPE (VIA FULL VIEW)
DESCRIPTION IS NULLABLE (leave the box blank)

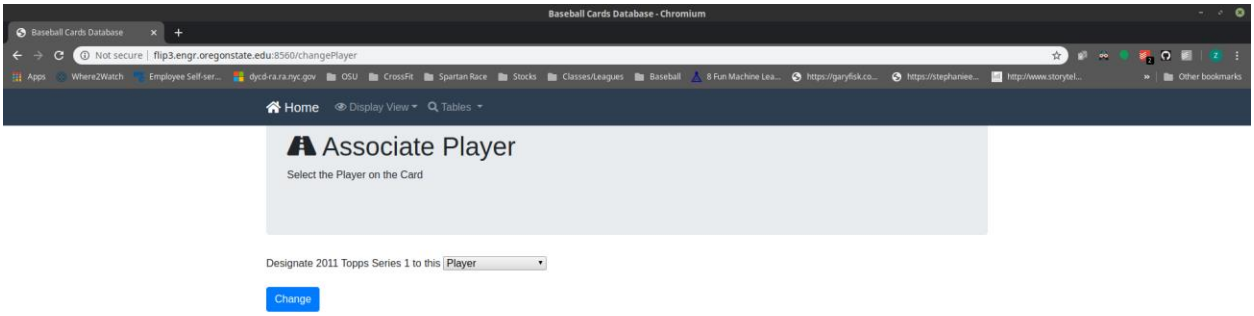


URL: http://flip3.engr.oregonstate.edu:8560/

UPDATE TEAMS (VIA FULL VIEW)
*UPDATE IS NULLABLE (*don't click on the drop-down, only click "Change")

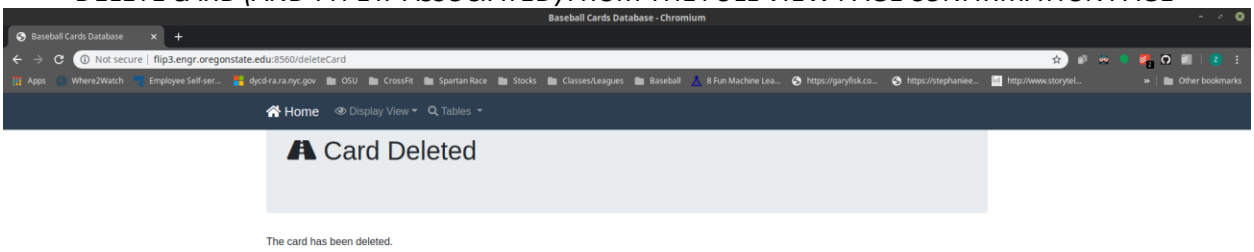


UPDATE PLAYERS (VIA FULL VIEW)
*UPDATE IS NULLABLE (*don't click on the drop-down, only click "Change")



URL: http://flip3.engr.oregonstate.edu:8560/

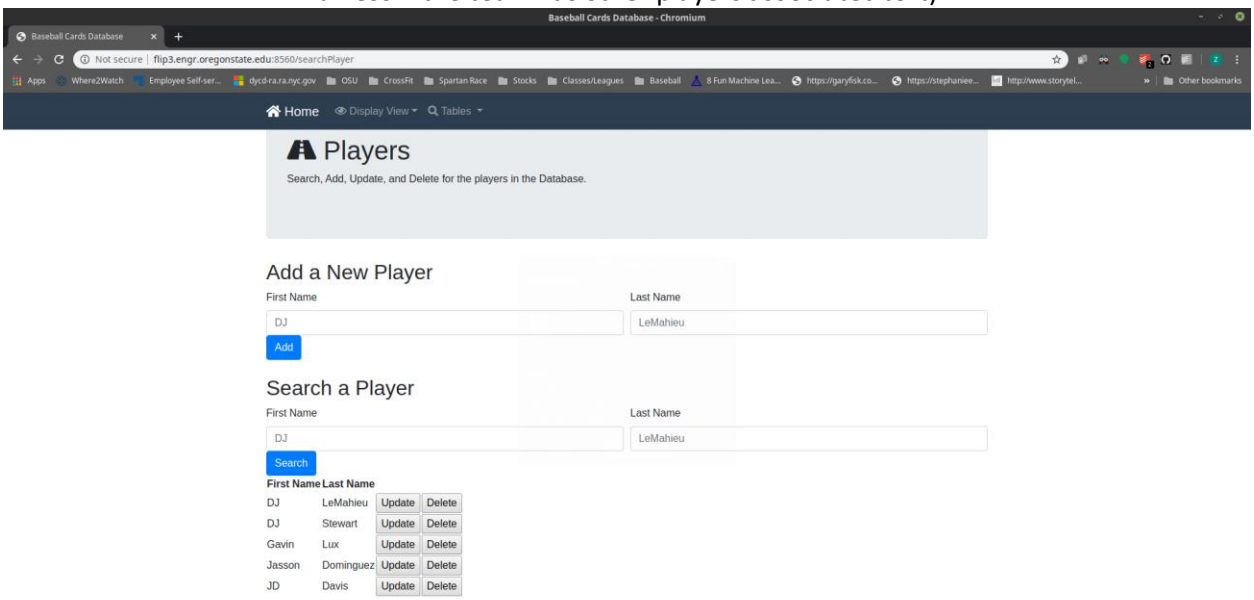
DELETE CARD (AND TYPE IF ASSOCIATED) FROM THE FULL VIEW PAGE CONFIRMATION PAGE



CREATE, READ, UPDATE, DELETE A PLAYER FOR THE PLAYERS' TABLE

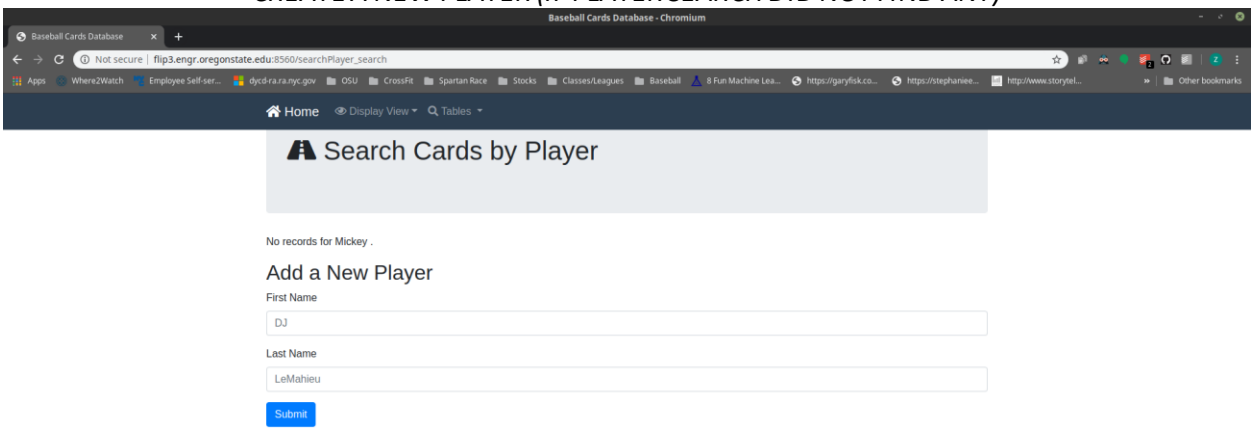
SEARCH FOR A PLAYER BY FIRST, LAST, OR FULL NAMES

If PLAYER deleted, will not affect cards or teams (the player will not show up on Players and Teams unless if the team has other players associated to it)



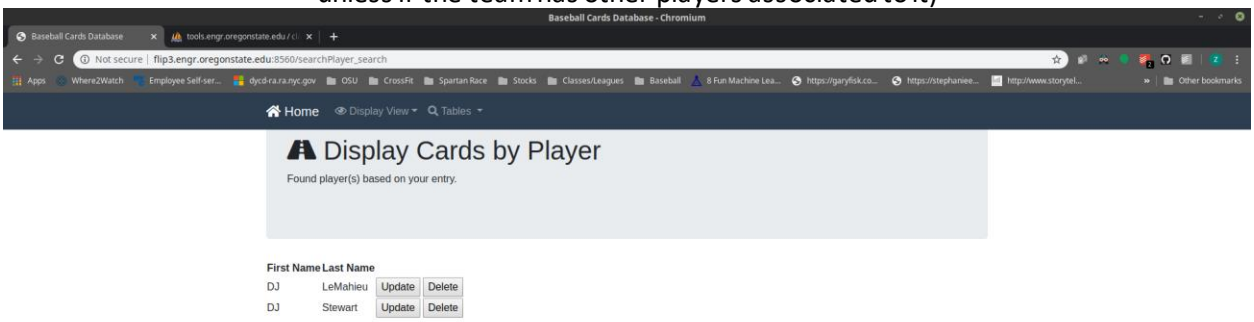
URL: http://flip3.engr.oregonstate.edu:8560/

CREATE A NEW PLAYER (IF PLAYER SEARCH DID NOT FIND ANY)



UPDATE AND DELETE PLAYER AFTER SUCCESSFULLY SEARCHING A PLAYER

If PLAYER deleted, will not affect cards or teams (the player will not show up on Players and Teams unless if the team has other players associated to it)

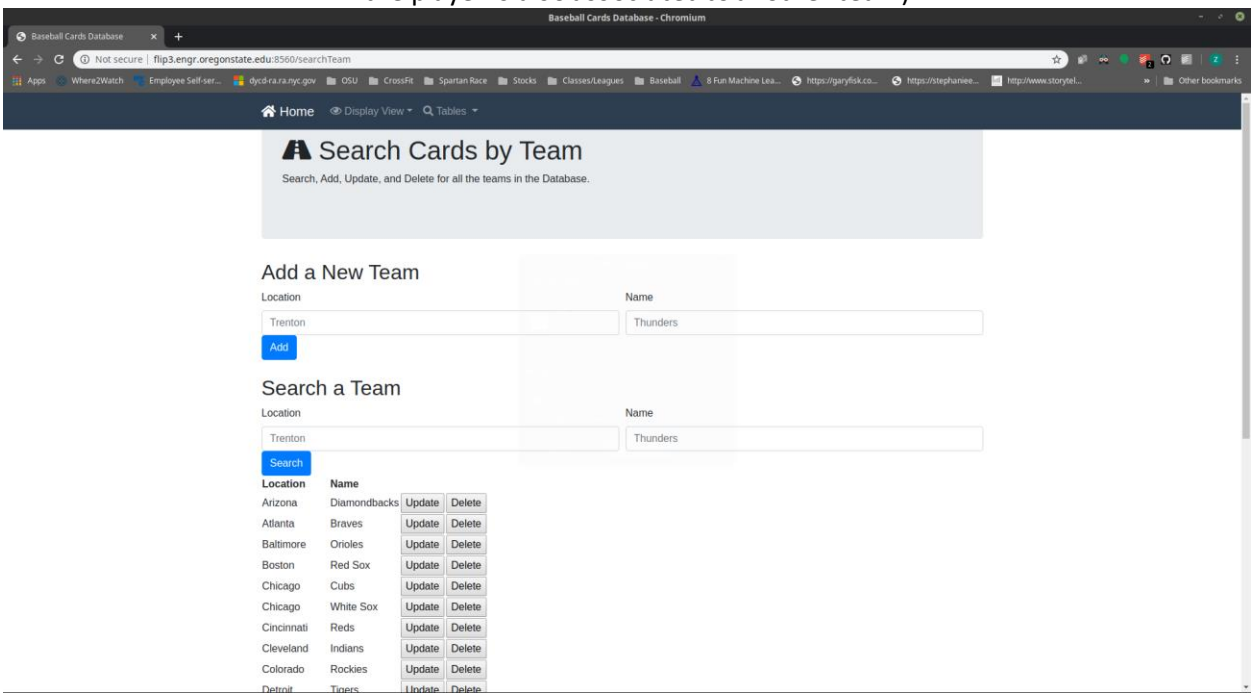


URL: http://flip3.engr.oregonstate.edu:8560/

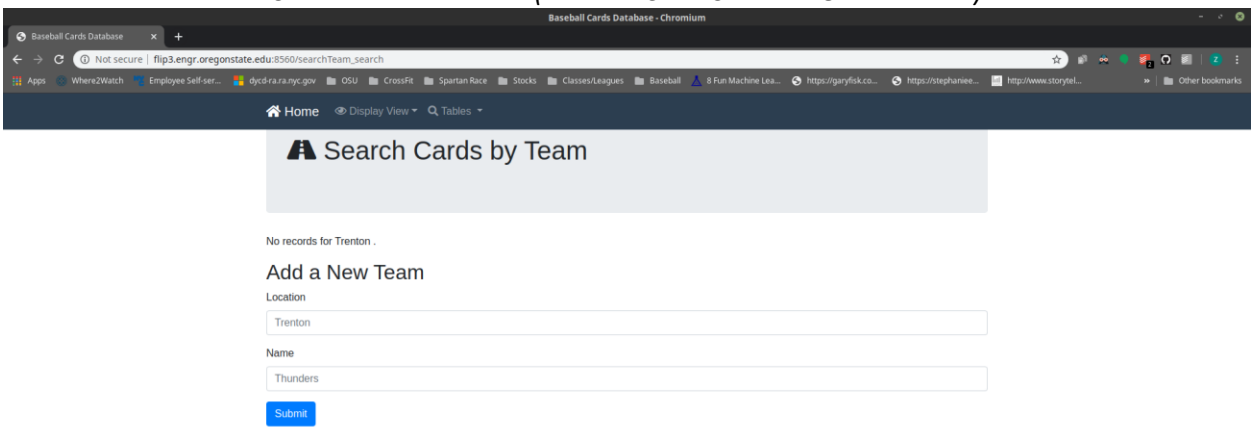
CREATE, READ, UPDATE, DELETE A TEAM FOR THE TEAMS' TABLE

SEARCH FOR A TEAM BY LOCATION, NAME, OR FULL TEAM NAME

If team deleted, will not affect cards or players (the team will not show up on Players and Teams unless if the player is also associated to another team)



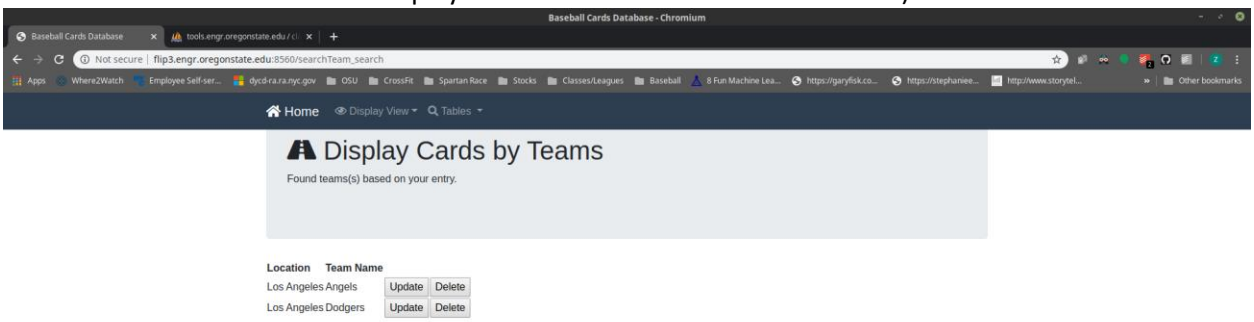
CREATE A NEW TEAM (IF TEAM SEARCH DID NOT FIND ANY)



URL: http://flip3.engr.oregonstate.edu:8560/

UPDATE AND DELETE TEAM AFTER SUCCESSFULLY SEARCHING A TEAM

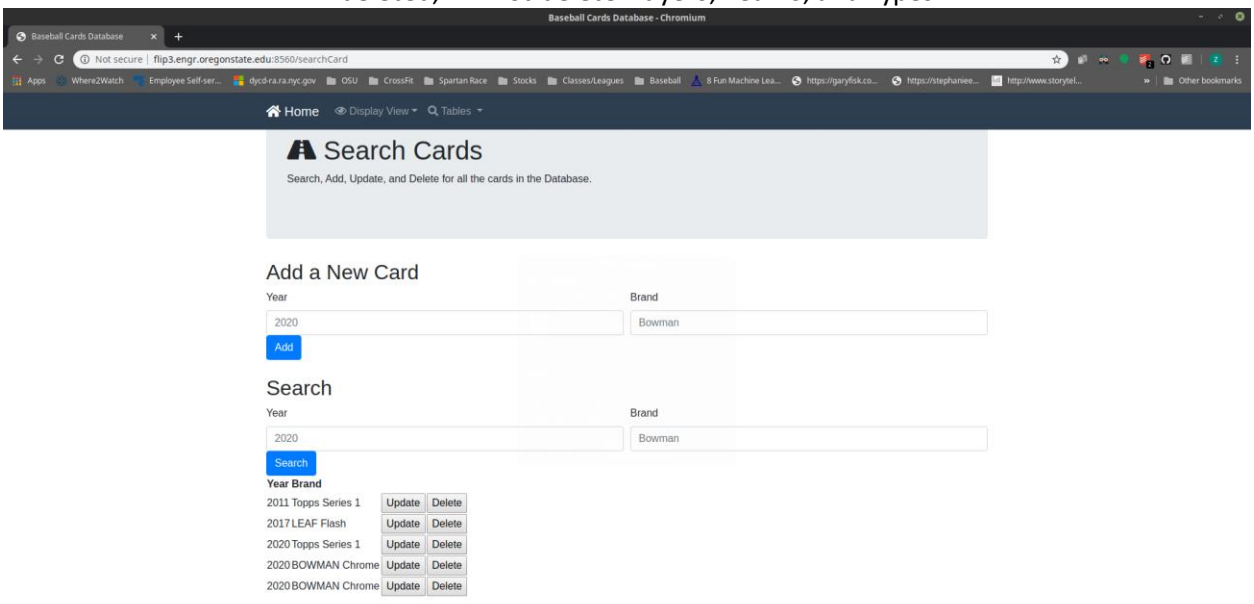
If team deleted, will not affect cards or players (the team will not show up on Players and Teams unless if the player is also associated to another team)



CREATE, READ, UPDATE, DELETE A PLAYER FOR THE CARDS' TABLE

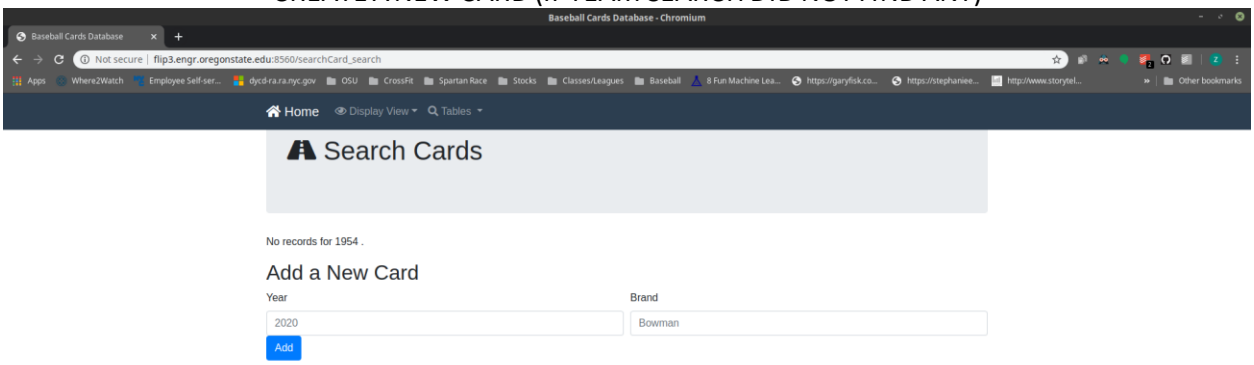
SEARCH FOR A CARD BY YEAR, BRAND, OR BOTH

If deleted, will not delete Players, Teams, and Types

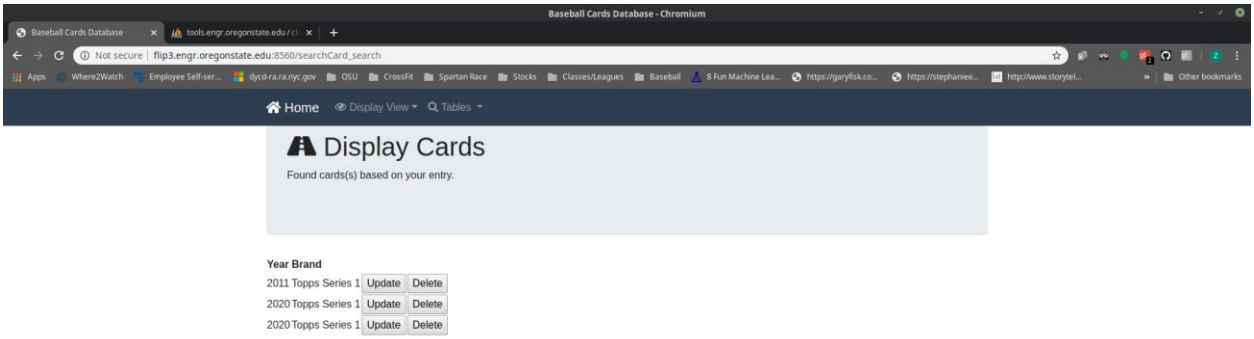


URL: http://flip3.engr.oregonstate.edu:8560/

CREATE A NEW CARD (IF TEAM SEARCH DID NOT FIND ANY)

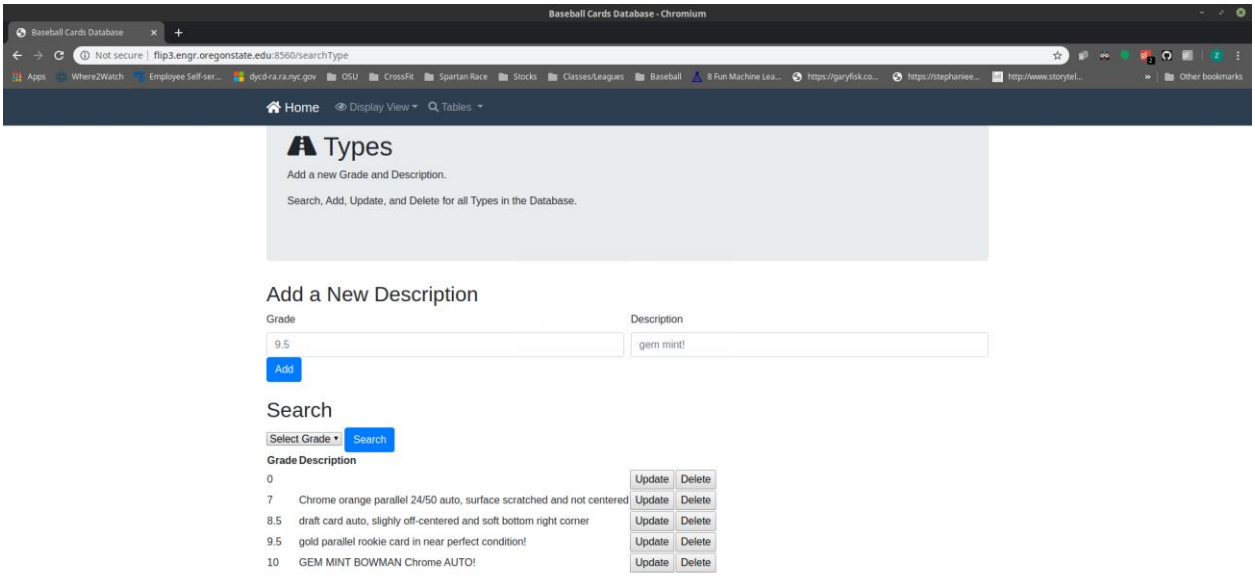


UPDATE AND DELETE CARD AFTER SEARCHING A CARD



URL: http://flip3.engr.oregonstate.edu:8560/

CREATE, READ, UPDATE, DELETE A TYPE FOR THE TYPES' TABLE
SEARCH FOR A TYPE BY GRADE VIA THE PREPOPULATED DROPDOWN LIST
Delete will not affect one on one relationship with Cards



UPDATE AND DELETE TYPE AFTER SEARCHING A TYPE