Project Step 7 Portfolio Assignment

Jaclyn Sabo
Sam Cain

Group 12 Coding Kai

HTML pages URL:

http://flip3.engr.oregonstate.edu:9278/index.html

Fixes based on Feedback CRUD
Based on the peer feedback and our own testing, we updated Classes with a pre-populated drop-down menu for the classroomID and teacherID fields. We also updated the Student Assigned Classes with a pre-populated drop-down menu of current studentIDs and classIDs. Therefore, a user can easily add a class or assign a student to a class without having to look up the IDs. For Students and Teachers, the Address2 input was updated so that the user can enter numbers and letters (i.e, unit 123). The phone number and postal code fields were also updated to be required fields. Delete functionality was added to Students as well as the ability to delete a student from Student Assigned Classes.

Fixes based on Feedback DML and DDQ Queries
Based on feedback from our peers, typos were updated including adding semi-colons to the end of queries. To show the M:M relationship, we also added a query that utilizes an INNER JOIN between Students and Classes. In addition, studentID was omitted from the Update Students UI as well as the search Students UI.

Actions based on the feedback on Design HTML
Based on the feedback, we decided to update our index.html UI to have a cleaner look with descriptions. We updated our search feature under Students to express a 'filter' to convey a clearer understanding that the user can filter the database based on student ID, last name, or gender. We did not add an INSERT to Student Assigned Classes since it's sole purpose is to express and display the M:M relationship among students and classes. We decided to change 'search by first and last name' to only include a filter by last name or search by gender so a user can dynamically filter students. We also added an insert to our intersection table based on advice for Ed Discussions.

Fixes to schema and database based on Feedback
Based on the feedback, we broke down the address into more specific attributes.
We updated the classroomID FK in Classes to be an optional NULL in order to meet the following project requirement: *"In a one-to-many relationship (like bsg_people to bsg_planets), you should be able to set the foreign key value to NULL (such as on a person in bsg_people), that removes the relationship."* In addition, it is possible for a class to exist without a classroom (i.e., online classes).

Project Outline and Database Outline, ERD and Schema

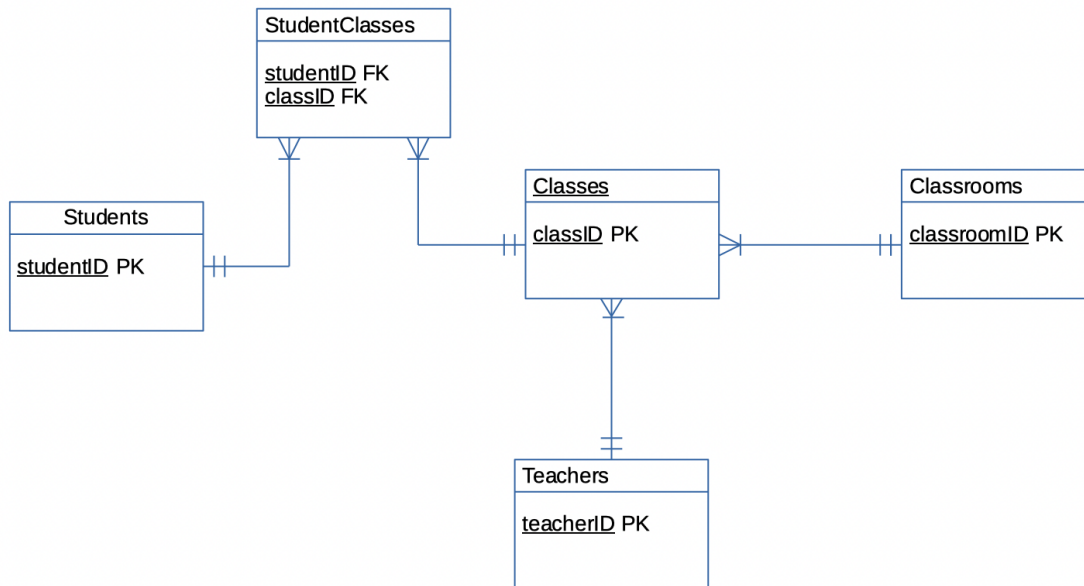**Project Title:** Mooncake High School Class Management Database

Overview

Mooncake High School has 90 classrooms with a student body of roughly 2000 students and a faculty of 90 teachers every year. There are approximately 240 classes available throughout the school year. Our database-driven website will record which/how many students are in each class, as well as which classroom is associated with each class. This can be used to make sure each teacher has a roughly equal number of students and that each classroom isn't double-booked.

Database Outline

- **Students:** records the details of Students enrolled within the school
  - studentID: int, auto_increment, unique, not NULL, PK
  - firstName: varchar, not NULL
  - lastName: varchar, not NULL
  - dateOfBirth: date, not NULL
  - gender: varchar, not NULL
  - streetAddressLine1: varchar, not NULL
  - streetAddressLine2: varchar
  - city: varchar, not NULL
  - state: varchar, not NULL
  - postalCode: int, not NULL
  - phoneNumber: int, not NULL
  - Relationship: a 1:M relationship between Students and Student_Classes is implemented with studentID as a FK inside of Student_Classes; *this conveys the M:M relationship with Students and Classes*

- **Classes**: records the details of Classes available at the school
  - classID: int, auto_increment, unique, not NULL, PK
  - className: varchar, not NULL
  - classroomID: int, FK
  - teacherID: int, not NULL, FK
  - Relationship: a 1:M relationship between Classes and Student_Classes is implemented with classID as a FK inside of Student_Classes; *this conveys the M:M relationship with Classes and Students*
  - Relationship: a M:1 relationship between Classes and Teachers
  - Relationship: a M:1 relationship between Classes and Classrooms

- **StudentClasses:** *intersection table to form M:M relationship between Classes and Students*
  - classID: int, not NULL, FK
  - studentID: int, not NULL, FK
  - Relationship: a M:1 relationship between Student_Classes and Classes

- ○ Relationship: a M:1 relationship between Student_Classes and Students

- ● **Teachers:** records the details of Teachers employed by the school
  - ○ teacherID: int, auto_increment, unique, not NULL, PK
  - ○ firstName: varchar, not NULL
  - ○ lastName: varchar, not NULL
  - ○ dateOfBirth: date, not NULL
  - ○ gender: varchar, not NULL
  - ○ streetAddressLine1: varchar, not NULL
  - ○ streetAddressLine2: varchar
  - ○ city: varchar, not NULL
  - ○ state: varchar, not NULL
  - ○ postalCode: int, not NULL
  - ○ phoneNumber: int, not NULL
  - ○ Relationship: a 1:M relationship between Teachers and Classes is implemented with teacherID as a FK inside of Classes

- ● **Classrooms**: records the details of available classrooms at the school
  - ○ classroomID: int, auto_increment, unique, not NULL, PK
  - ○ classroomNum, int, not NULL
  - ○ maxCapacity: int, not NULL
  - ○ Relationship: a 1:M relationship between Classrooms and Classes is implemented with classroomID as a FK inside of Classes
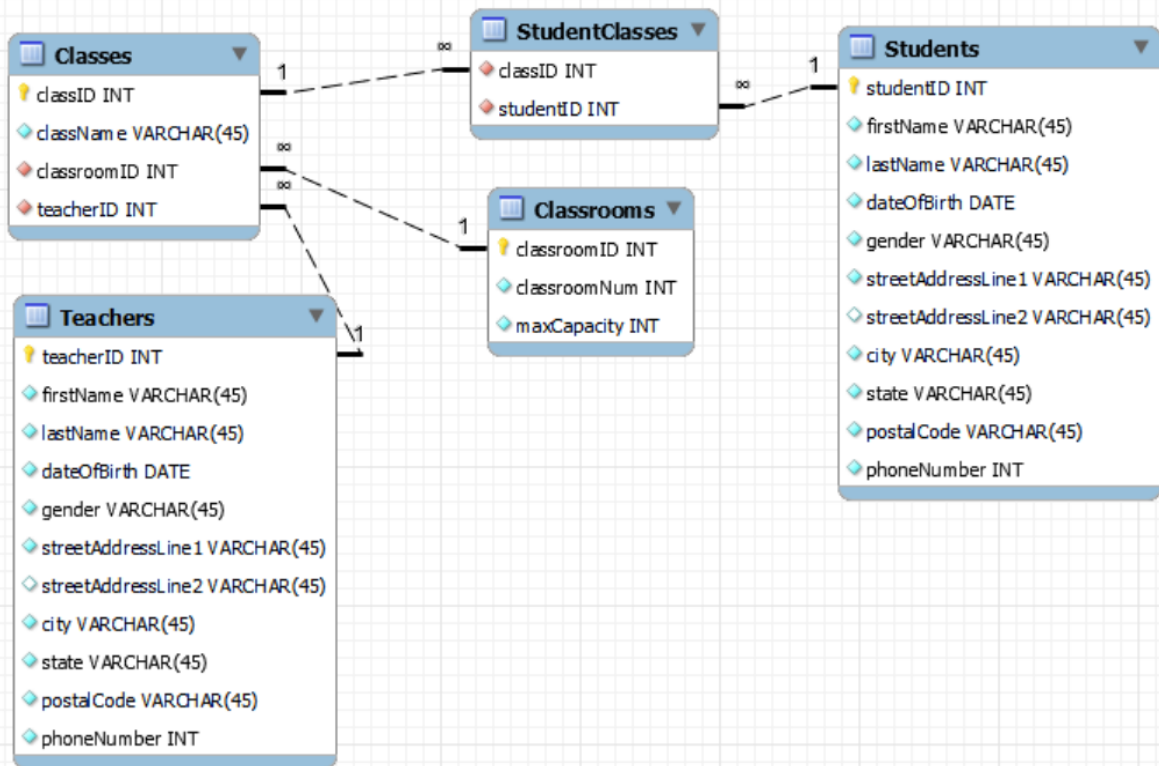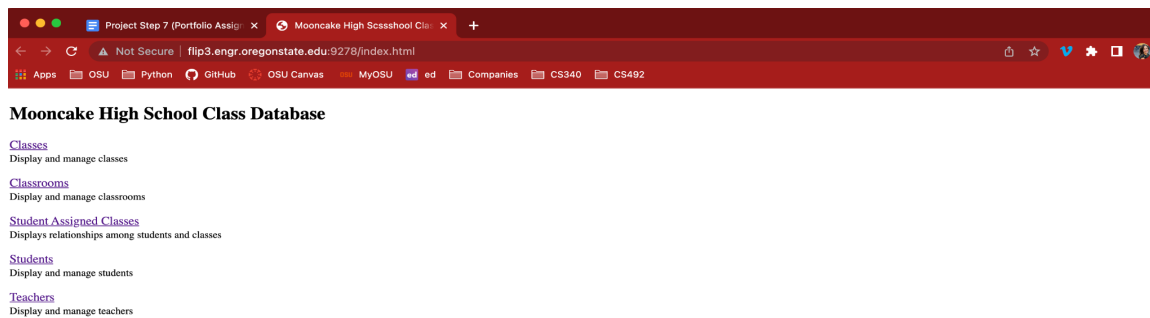
c) Entity-Relationship Diagram:

```
                    ┌─────────────────┐
                    │ StudentClasses  │
                    ├─────────────────┤
                    │ studentID FK    │
                    │ classID FK      │
                    └─────────────────┘
                         ∨        ∨
   ┌──────────────┐                      ┌──────────────┐              ┌──────────────┐
   │  Students    │                      │  Classes     │              │ Classrooms   │
   ├──────────────┤                      ├──────────────┤              ├──────────────┤
   │ studentID PK │────┤┤           ├├──│ classID PK   │──────────────│ classroomID PK│
   └──────────────┘                      └──────────────┘              └──────────────┘
                                              ∨
                                              │
                                         ┌──────────────┐
                                         │  Teachers    │
                                         ├──────────────┤
                                         │ teacherID PK │
                                         └──────────────┘
```

d) Schema:

Red = Foriegn Key
Lightbulb = Primary Key

Classes
- classID INT
- className VARCHAR(45)
- classroomID INT
- teacherID INT

StudentClasses
- classID INT
- studentID INT

Students
- studentID INT
- firstName VARCHAR(45)
- lastName VARCHAR(45)
- dateOfBirth DATE
- gender VARCHAR(45)
- streetAddressLine1 VARCHAR(45)
- streetAddressLine2 VARCHAR(45)
- city VARCHAR(45)
- state VARCHAR(45)
- postalCode VARCHAR(45)
- phoneNumber INT

Classrooms
- classroomID INT
- classroomNum INT
- maxCapacity INT

Teachers
- teacherID INT
- firstName VARCHAR(45)
- lastName VARCHAR(45)
- dateOfBirth DATE
- gender VARCHAR(45)
- streetAddressLine1 VARCHAR(45)
- streetAddressLine2 VARCHAR(45)
- city VARCHAR(45)
- state VARCHAR(45)
- postalCode VARCHAR(45)
- phoneNumber INT

Home index.html page

READ/CREATE Classes page
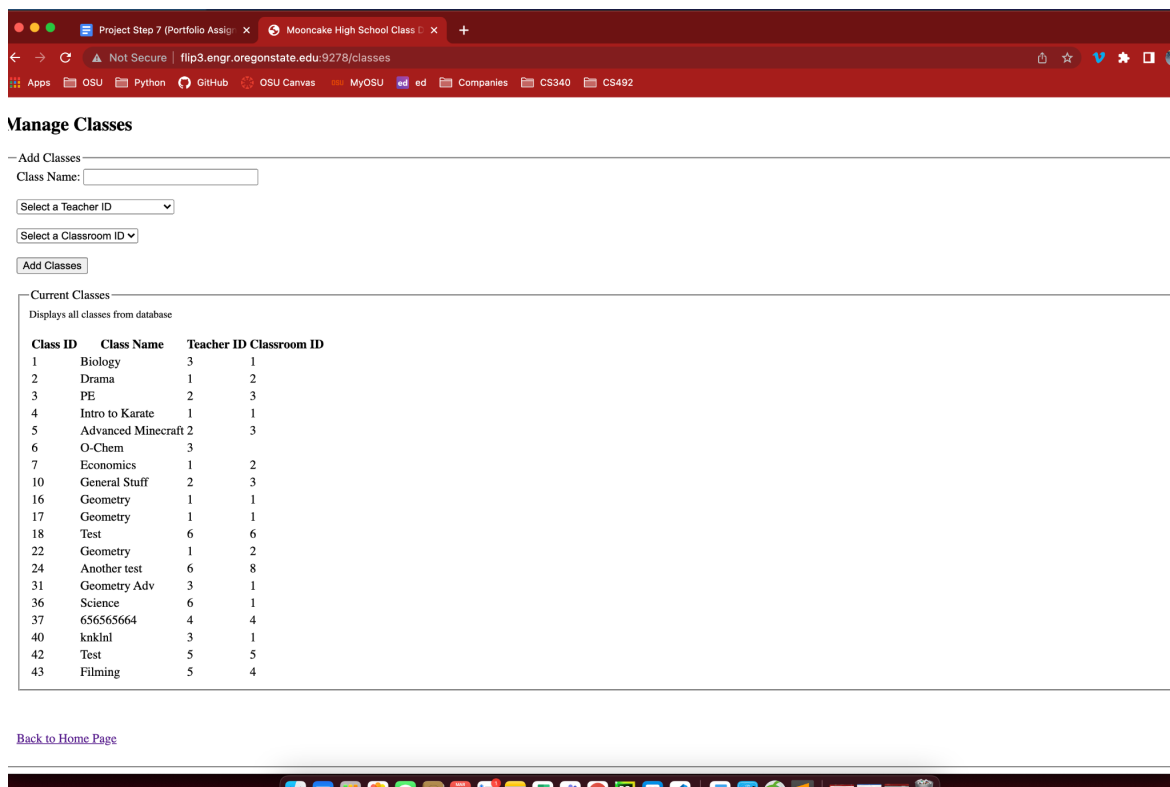
READ/CREATE Classrooms page

Manage Classrooms

Add Classroom

Classroom Number: [_____]

Max Capcity: [_____]

[Add Classroom]

Current Classrooms

Displays all classrooms from database

Classroom ID	Classroom Number	Max Capacity
1	111	35
2	112	20
3	113	55
4	114	55
5	115	34
6	123	32
7	144	23
8	199	9
9	130	3
10	212	800
11	201	200

Back to Home Page

READ/CREATE/DELETE Student Assigned Classes page

Manage Student Assigned Classes

Assign a Class to a Student

[Select a Class ID ▼]

[Select a Student ID ▼]

[Assign Class]

Current Student Assigned Classes

Displays all students and corresponding classes from database with option to delete a student from the class

Student ID	First Name	Last Name	Class ID	Class Name	
3	Dogboy	Boydog	1	Biology	[Delete Student]
2	Regina	Renegade	2	Drama	[Delete Student]
3	Dogboy	Boydog	2	Drama	[Delete Student]

Back to Home Page

READ/CREATE/DELETE/SEARCH/UPDATE Students page

UPDATE Student page/form

READ/CREATE Teachers page

Manage Teachers
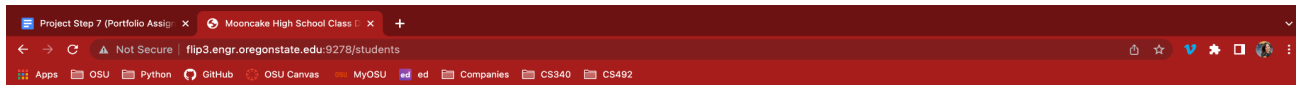
┌─ Add Teachers ───┐

First Name: [   ] Last Name: [   ]

Birthdate: [mm/dd/yyyy 📅] Gender: [▾]

Address: [   ] [Apt, suite, unit, building, floor, etc.]

City: [   ] State : [Alabama ▾] Zip Code: [   ]

Phone Number: [   ] Format: 1234567890

[Add Teacher]

└──┘

┌─ Current Teachers ───┐

Displays all teachers from database

Teacher ID	First Name	Last Name	Birthdate	Gender	Address1	Address2	City	State	Zip Code	Phone Number
1	Bonnie	Buckets	1989-07-11	Female	1758 SE Nebula Rd.		Damascus	OR	97089	9712238095
2	Dean	Domino	1979-11-07	Male	3465 SE Cabriders Dr.	apt. 34	Clackamas	OR	97254	9713448873
3	Karl	LaFrug	1978-11-27	Male	2435 NE Red Rider Rd.		Portland	OR	97225	5038023324
4	Sam	Cain	1992-09-06	male	1433 SE Wendle Ave.	NULL	Portland	OR	97202	503
5	Test	Guy	1988-03-14	Non-Binary	1914 W Henderson St.		Portland	OR	97837	5039988888
6	Jack	Pump Sr.	2022-03-10	Female	24 Smallman Street		Pittsburgh	PA	15222	4121111111
7	Hello	Kitty	1976-04-23	Female	1111 IDK Ave		Tokyo	GA	99999	9998887777
8	Teacher	Mansen	1998-12-12	Female	454 Flower Rd		Portland	OR	45455	4445557777
9	Faketeacher	Real	1984-08-09	Female	45454 Five Rd	3333	Matrix	AZ	74545	3652556645
10	John	Smith	2022-03-09	none	adress	444	portland	AL	405	123456789
11	adam	joe	2022-01-31	none	adress		city	AL	15666	0
12	Fake	Nametwo	2022-03-09	Female	123 Tall Street	Unit 1	Pittsburgh	PA	15222	4121112222

└──┘

Back to Home Page