Link: http://flip3.engr.oregonstate.edu:8530/users

Project Step 5 Draft Version: Implement MOST Remaining Entities/Pages (Group on Ed Discussions)

Names: Woo Pyon, Salim Jalaleddine

Summary Reflection:

There were a lot of changes from our initial post from Project Step 1 to Project Step 6. There were a lot of helpful comments during each step of our project peer reviews. We had to tweak it a little to make it more user-friendly. On Project Step 4, one peer mentioned we should add an alert for deletion to make sure the user isn't deleting it by accident. In order to fix this we added a confirm function in our delete javascript files. Also, we changed our column titles so it looks better than using an attribute naming style. We did this by making capitalization of the attributes consistent by making the first letter upper-case. We didn't realize there were a lot of bugs that users could run into without the help of peer reviews. For example, setting up a limit for GPA so there isn't a negative number or above a 4.0. We fixed it by putting a min and max inside our form in the Handlebars file. In addition, some forms that were required, the user could leave blank, which could be problematic. To counter this, we made forms that are supposed to be not nullable in our schema to be required in our Handlebars forms. Additionally, one thing we had to update constantly was our DDL because we wanted to change some attributes in certain entities such as the internship entity as well as the intersection tables.

On project step 5, we got stuck on getting the update functionality to work for our Projects entity, but this was fixed by adding a conditional in our app.js file that handled null inputs. Another change we also made was making the search functionality dynamic for our intersection table, Users Questions. From Step 5 to the final version, we also added a javascript function that allowed for values to be automatically filled into the update form when a User chooses an existing attribute from the table to update. From Step 5 to the final version we also removed the redundant selection of attributes by the user, such as when selecting a row to update, the user will choose a name from the dropdown instead of an id. The major changes throughout the project culminated in the presented final version which fulfills the requirements of the project.

Project Outline and Database Outline:

A. Overview: A good question that one of the 65,0000 computer science graduates yearly may ask is, am I ready for a job in software engineering? Look no further, as our web based database "JobReady" will solve this problem. JobReady takes into account many different aspects that factor into job readiness and provides users a database to store important materials relevant to job searching. The database can hold up to 1,000 users at once, and will allow users to store information regarding interview Questions (up to 1000), Classes taken, Projects, as well as Internships that have been completed. This gives students the confidence boost when they apply for full-time software engineering positions. JobReady, with its massive database of computer science graduates, can also be utilized by companies to determine whether a student has prepared for interviews through data structure and algorithm practice questions, which can be an added filter to the candidate seeking process. Some companies we partnered with are FlixNet, Rainforest, Red Fruit and many more. These companies are here to help students be more aware if they are suitable for that company.

B. Database Outline, in Words

Users: records the user that will have score calculated

- user_id: int, auto_increment, unique, not NULL, PK -
- full_name: varchar(100), not NULL
- language: varchar(100), not NULL
- university: varchar(100), not NULL
- gpa: decimal(19,2),
- location: varchar(100), not NULL
- linkedin: varchar(100), not NULL
- Relationship:
    - 1:M relationship:
        - Between user and classes implemented with user_id as FK inside of Classes.
        - Between user and projects implemented with user_id as FK inside of Projects
        - Between user and internship implemented with user_id as FK inside of Internships
    - A M:M relationship between users and questions implemented with user_id as FK inside of intersection table User_Questions
    - A M:M relationship between users and classes implemented with user_id as FK inside of intersection table User_Classes

Questions:

- question_id: int, auto_increment, unique, not NULL, PK
- problem: varchar(100), not NULL
- difficulty: varchar(50), not NULL
- q_link: varchar(100), not NULL
- Relationship: M:M relationship between questions and users implemented with user_id as FK inside a intersection table, User_Questions

Users_Questions (Intersection Table)**:**

- users_questions_id: int, NOT NULL, UNIQUE, auto_increment
- user_id: int, not NULL, FK
- question_id: int, not NULL, FK
- Relationship:
    - M:1 relationship between questions to user implemented with user_id as FK inside User_Questions
    - M:1 relationship between users to question implemented with user_id as FK inside User_Questions

Classes:

- class_id: int, auto_increment, unique, not NULL, PK
- name: varchar(100), not NULL
- Relationship: M:M relationship between classes and user implemented with user_id as FK inside a intersection table, User_Classes

Users_Classes:

- users_classes_id: int, not NULL, unique, auto_increment
- user_id: int, not NULL, FK
- class_id: int, not NULL, FK
- Relationship:
    - M:1 relationship between classes and user implemented with user_id as FK inside User_Classes
    - M:1 relationship between users and class with user_id as FK inside User_Classes

Projects:

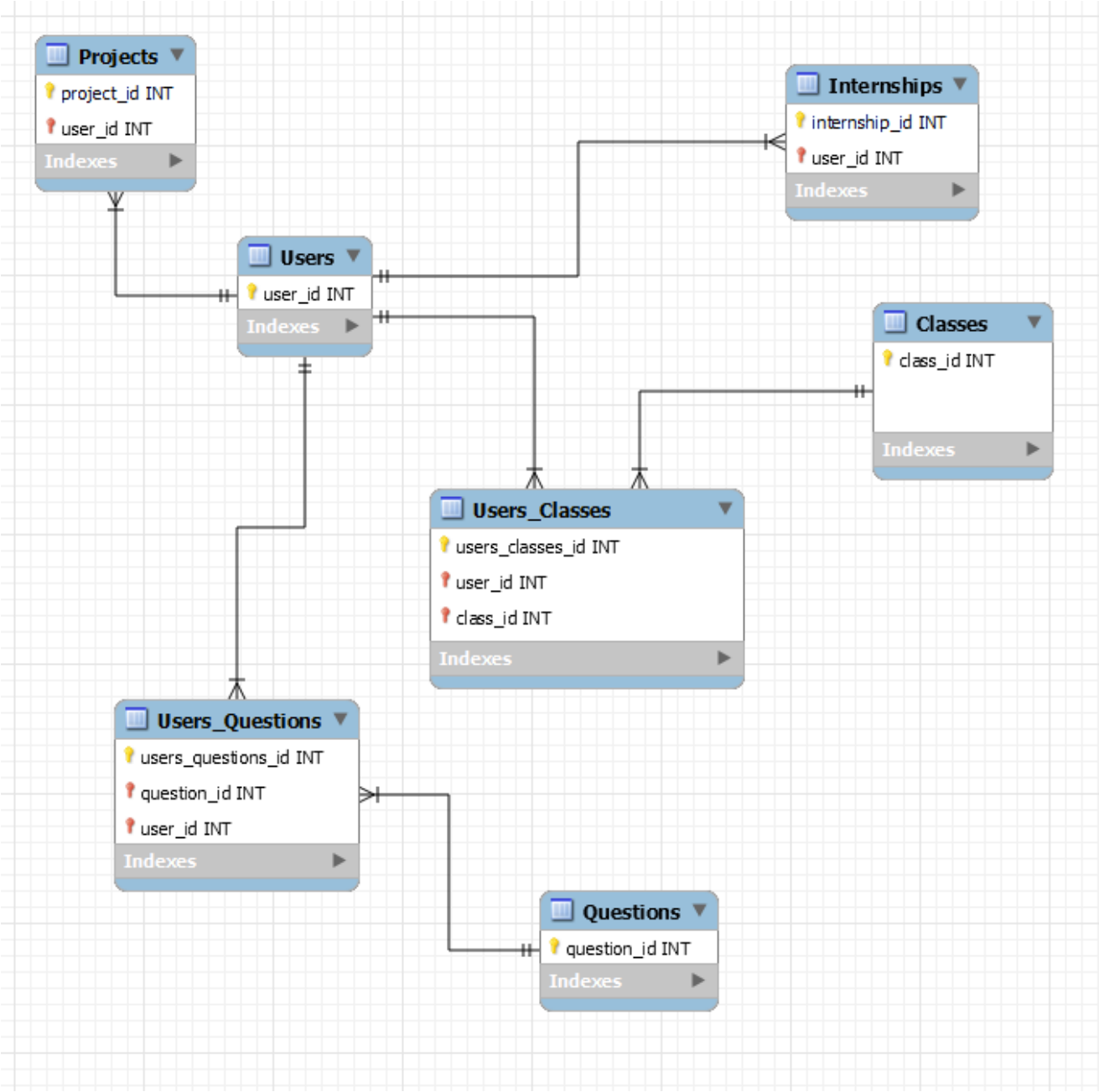
- project_id: int, auto_increment, unique, not NULL, PK
- user_id: int, unique, FK
- description: LONGTEXT, noT NULL
- language: varchar(50), not NULL

- github_link: varchar(100), not NULL
- Relationship: M:1 relationship between projects and user implemented with user_id as FK inside of Projects

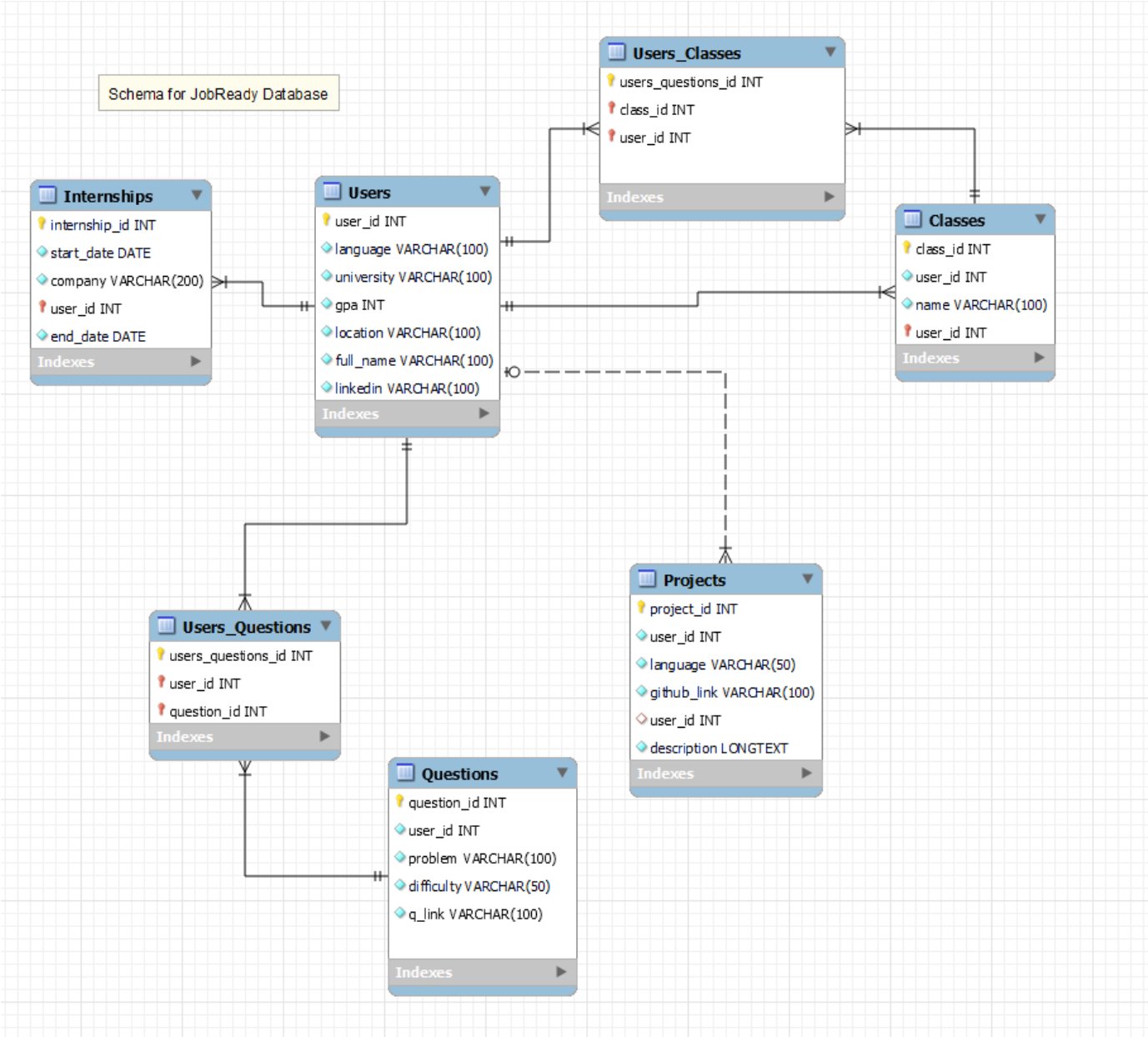
Internships:

- internship_id: int, auto_increment, unique, not NULL, PK
- user_id: int, unique not NULL, FK
- start_date: DATE, not NULL
- end_date: DATE, not NULL
- company: varchar(200), not NULL
- Relationship: M:1 relationship between internships and user implemented with user_id as FK inside of Internships

Entity-Relationship Diagram:



Schema:


Schema for JobReady Database
Users_Classes
users_questions_id INT
class_id INT
user_id INT
Indexes
Internships
internship_id INT
start_date DATE
company VARCHAR(200)
user_id INT
end_date DATE
Indexes
Users
user_id INT
language VARCHAR(100)
university VARCHAR(100)
gpa INT
location VARCHAR(100)
full_name VARCHAR(100)
linkedin VARCHAR(100)
Indexes
Classes
class_id INT
user_id INT
name VARCHAR(100)
user_id INT
Indexes
Projects
project_id INT
user_id INT
language VARCHAR(50)
github_link VARCHAR(100)
user_id INT
description LONGTEXT
Indexes
Users_Questions
users_questions_id INT
user_id INT
question_id INT
Indexes
Questions
question_id INT
user_id INT
problem VARCHAR(100)
difficulty VARCHAR(50)
q_link VARCHAR(100)
Indexes

Sample Data

Users:

user_id	full_name	language	university	gpa	location	linkedin
1	John Doe	Python, Javascript	Harvard	4.0	Irvine, CA	www.linkedin.com/in/john-doe
2	Sammy Smith	C++	MIT	3.9	Cambridge, MA	www.linkedin.com/in/sammy-smith
3	Stacee Lopez	C,Javascript, CSS, HTML	Oregon State	3.5	Seattle, WA	www.linkedin.com/in/stacee-lopez

Questions:

question_id	problem	difficulty	q_link
1	Two Sum	Easy	https://leetcode.com/problems/two-sum/
2	Trapping Rain Water	Hard	https://leetcode.com/problems/trapping-rain-water/
3	Snakes and Ladders	Medium	https://leetcode.com/problems/snakes-and-ladders/

Users_Questions:

user_id	question_id
1	1
2	2
3	2
3	3

Classes:

class_id	name
1	Data Structures
2	Web Development
3	Algorithms

Users_Classes:

user_id	class_id
1	1
2	1
3	2
2	3

Projects:

project_id	user_id	description	language	github_link
1	1	Created a battleship game	Python	https://github.com/doej/battleship
2	2	Created a tetris game	C++	https://github.com/smiths/sammy_tetris
3	3	Developed a full stack website	Javascript, CSS, HTML	https://github.com/lopezs/website

Internships:

internship_id	user_id	start_date	end_date	company
1	1	05/25/2016	09/10/2016	Red Fruit
2	2	10/15/2012	02/10/2012	Rainforest
3	3	05/11/2015	09/11/2015	Checks

UI Screenshots

Users

DISPLAY, CREATE, SEARCH, UPDATE, DELETE Users page

Users | Questions | Classes | Projects | Internships | Users Questions | Users Classes

> > > >

Users

User ID	Full Name	Language	University	GPA	Location	Linkedin	Delete
1	John Doe	Python, Javascript	Harvard	4	Irvine, CA	www.linkedin.com/in/john-doe	Delete
2	Sammy Smith	C++	MIT	3.9	Cambridge, MA	www.linkedin.com/in/sammy-smith	Delete
3	Stacee Lopez	C, Javascript, CSS, HTML	Oregon State	3.5	Seattle, WA	www.linkedin.com/in/stacee-lopez	Delete

Add User

To add a new user, please enter the user's information down below

Full Name*: Language*: University*:

GPA: Location*: Linkedin*: http://linkedin.com Submit

Search for User

Search by full name using the field below. Just the first/last name will work too

Search by Full Name: Submit Reset

Updating A User's Information

To update a specific user's information select their name from the drop-down menu on the left and input the new information in the other boxes

Full name*: Select a User Language*: University*:

GPA: Location*: Linkedin*: http://linkedin.com Submit

Questions

DISPLAY, CREATE, SEARCH, UPDATE, DELETE Questions page

Users | Questions | Classes | Projects | Internships | Users Questions | Users Classes

> > > >

Questions

Question ID	Problem	Difficulty	Link	Delete
1	Two Sum	Easy	https://leetcode.com/problems/two-sum/	Delete
2	Trapping Rain Water	Hard	https://leetcode.com/problems/trapping-rain-water/	Delete
3	Snakes and Ladders	Medium	https://leetcode.com/problems/snakes-and-ladders/	Delete

Add Question

To add a new question, please enter the question's information below

Problem*: Difficulty* Easy Link*: Submit

Search for Question

Search by Problem: Submit Reset

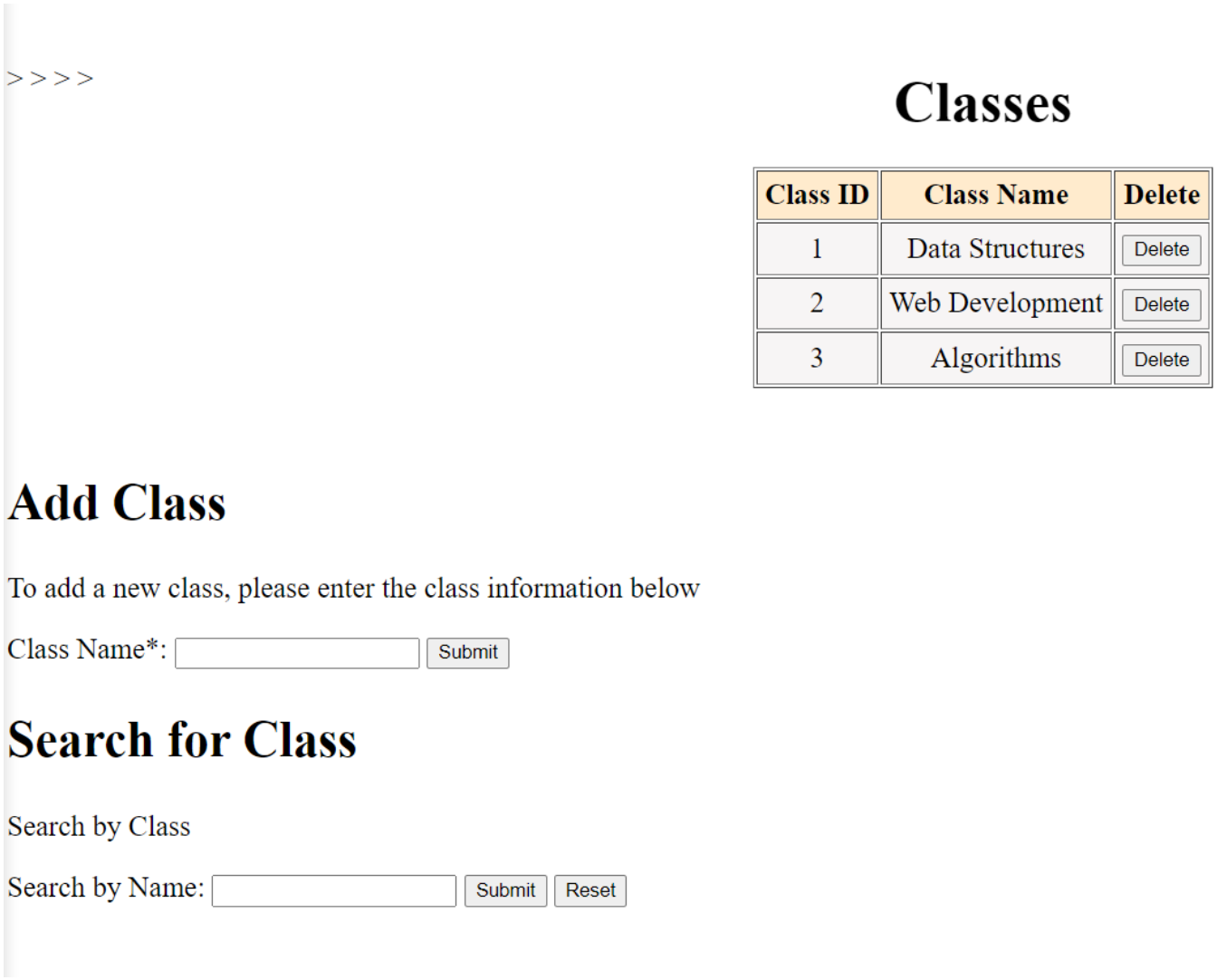
Updating a Question's Information

To update a specific question's information select the problem from the drop-down menu on the left and input the new information in the other

Problem*: Select a problem Difficulty* Easy Link*: Submit

Classes

DISPLAY, CREATE, SEARCH, DELETE Classes page



Projects

DISPLAY, CREATE, SEARCH, UPDATE, DELETE Projects page

> > > >

Projects

Project ID	User Name	Description	Language	Github link	Delete
1	John Doe	Created a battleship game	Python	https://github.com/doej/battleship	Delete
2	Sammy Smith	Created a tetris game	C++	https://github.com/smiths/sammy_tetris	Delete
3	Stacee Lopez	Developed a full stack website	Javascript, CSS, HTML	https://github.com/lopezs/website	Delete

Add Project

To add a new project, please enter the project's information down below

Full name: No User Description*: Language*:

Github*: Submit

Search for Project

Search by language using the field below.

Search by Language: Submit Reset

Updating a Project's Information

To update a specific project's information select the git link from the drop-down menu on the left and input the new information in the other boxes

Github*: Select a project Full name: No User Description*: Language*: Submit

Internships

DISPLAY, CREATE, DELETE Internships page

> > > >

Internships

Internship ID	Full Name	Start Date	End Date	Github link	Delete
1	John Doe	05/25/2016	09/10/2016	Red Fruit	Delete
2	Sammy Smith	10/15/2012	02/10/2012	Rainforest	Delete
3	Stacee Lopez	05/11/2015	09/11/2015	Checks	Delete

Add Internship

To add a new internship, please enter the internship's information down below

Full Name*: Select a User Start Date*: mm/dd/yyyy End Date*: mm/dd/yyyy

Company*: Submit

Users Questions

DISPLAY, CREATE, SEARCH, UPDATE, DELETE Users Questions page

> > > > >

Users Questions

ID	Name	Question	Delete
1	John Doe	Two Sum	Delete
2	Sammy Smith	Trapping Rain Water	Delete
3	Stacee Lopez	Trapping Rain Water	Delete
4	Stacee Lopez	Snakes and Ladders	Delete

Create User and Question pair

To add a new pair, please use boxes below

Full name*: John Doe Question*: Two Sum Submit

Search for User Question Pair

Search by user name using the dropdown below.

Search by Name: Select a User Submit Reset

Updating a User Question Pair

To update a User Question pair select id from the drop-down menu on the left, and input the new question/full name from the two dropdowns

ID: Select a project Full Name: No User Question: No Question Submit

Users Classes

DISPLAY, CREATE, DELETE Users Classes page

> > > > >

Users Classes

ID	Name	Class	Delete
1	John Doe	Data Structures	Delete
2	Sammy Smith	Data Structures	Delete
3	Stacee Lopez	Web Development	Delete
4	Sammy Smith	Algorithms	Delete

Create User and Class pair

To add a new pair, please use boxes below

Full name*: John Doe Class*: Data Structures Submit

Citations: All citations can be found in source code submitted