

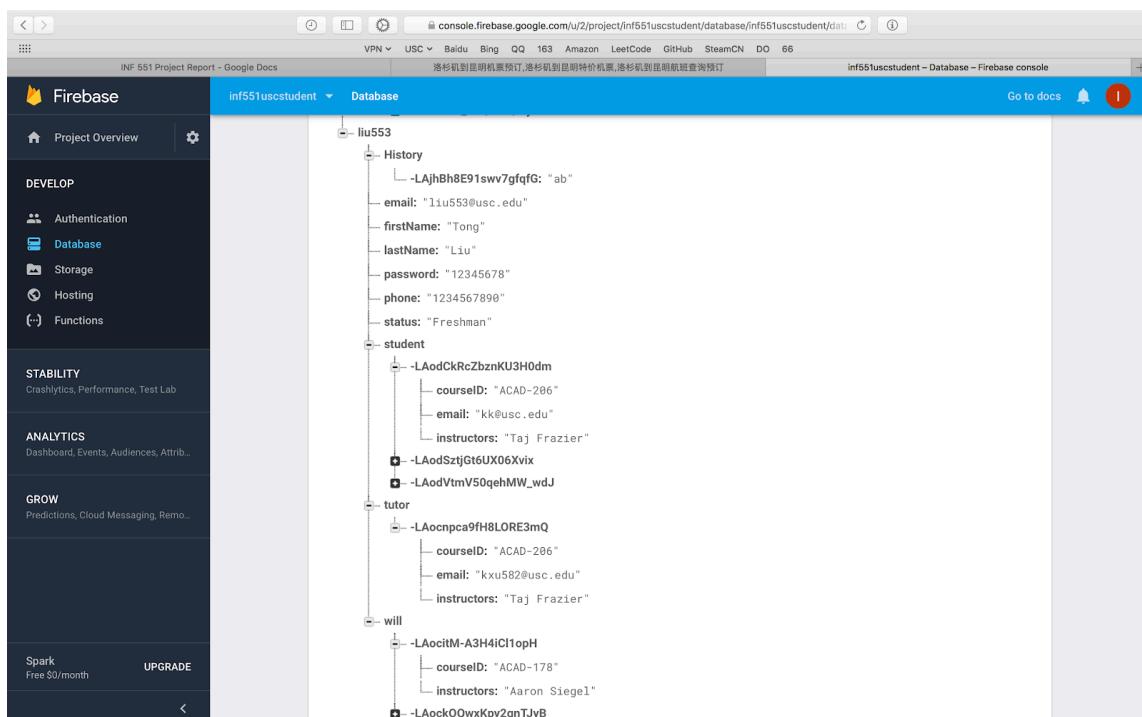
# INF551 Final report -- USC Tutor Finder

Xueping Li (xuepingl@usc.edu), Ke Xu (kxu582@usc.edu)

USC Tutor Finder is a website that aims to provide convenience to students who need immediate help from peers who are able to solve their academic doubts. The intuition comes from the fact that some students, especially freshman, do not know a lot of people from their classes and might have many doubts on the course materials. USC Tutor Finder can solve this situation. We want to produce a user-friendly environment for all USC students that allows them to pair with each other freely as tutor and student. Not only can they make friends, but also solve the problems one student has, which can help freshmen achieve their academic goals.

## 1. Data Structure

Since the website targets to USC students, and one of the functions is to let users search through the classes, we firstly crawled all the courses information from <https://classes.usc.edu/term-20181/>. we got urls step by step, then retrieved information from new urls we've got. In the end, stored them in the format of Json on Google Firebase. In the data retrieving part, we haven't used any framework or existing projects as our crawler's base because of the fewer data amount for this project.



The screenshot shows the Firebase Database interface with the following structure:

```
inf551uscstudent
  - liu553
    - History
      - -LAjhBh8E91swv7gfqfG: "ab"
      - email: "liu553@usc.edu"
      - firstName: "Tong"
      - lastName: "Liu"
      - password: "12345678"
      - phone: "1234567890"
      - status: "Freshman"
    - student
      - -LAodCkRcZbznKU3H0dm
        - courseId: "ACAD-206"
        - email: "kk@usc.edu"
        - instructors: "Taj Frazier"
      - -LAodSz1Gt6UX06Xvkx
      - -LAodVtmV50qehMW_wdJ
    - tutor
      - -LAocnPCA9fH8LORE3mQ
        - courseId: "ACAD-206"
        - email: "kxu582@usc.edu"
        - instructors: "Taj Frazier"
    - will
      - -LAocitM-A3H4iCl1opH
        - courseId: "ACAD-178"
        - email: "will@usc.edu"
        - instructors: "Aaron Siegel"
      - -LAockOQwKpy2gnTjyB
```

We have devised user information structure like the screenshot shown above. We chose to store user email and password in the Firebase. Email had the function of primary key. This tree-like structure also contained the basic information of users' student and tutor courses. Besides, this user database can store other information as well, first name, last name, history, etc.

The screenshot shows the Firebase Realtime Database console. The left sidebar has sections for DEVELOP (Authentication, Database, Storage, Hosting, Functions), STABILITY (Crashlytics, Performance, Test Lab), ANALYTICS (Dashboard, Events, Audiences, Attribution), and GROW (Predictions, Cloud Messaging, Remote Config). The main area is titled 'Database' and shows the 'Realtime Database' tab selected. Below are tabs for DATA, RULES, BACKUPS, and USAGE. The database structure is displayed as a tree:

```

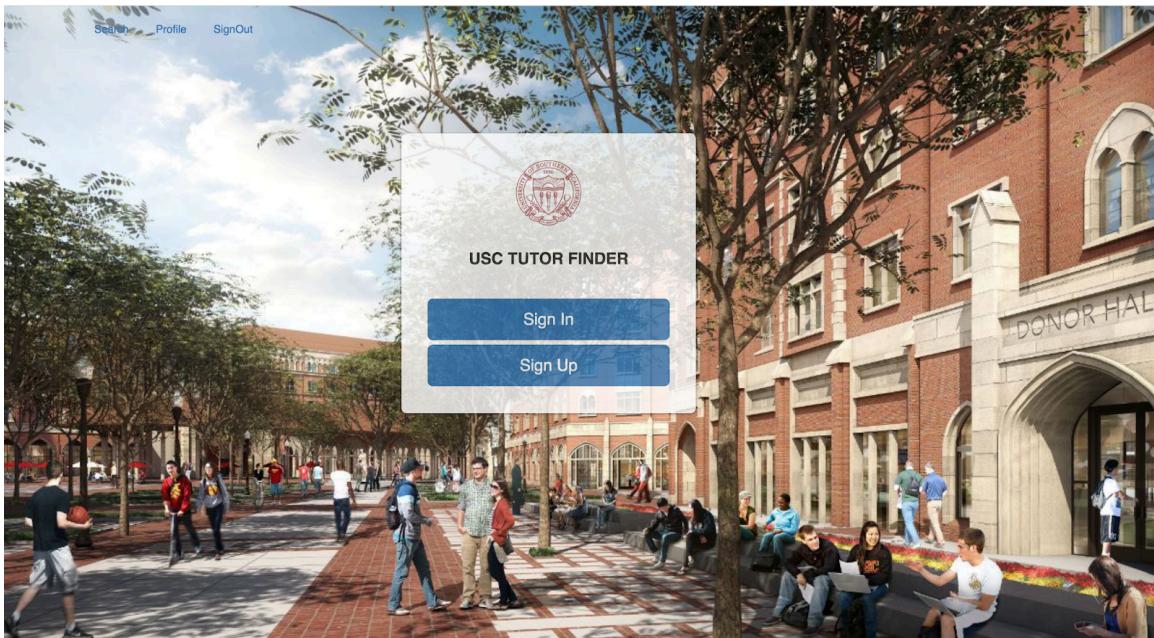
inf551usc-61ddc
  |- 0
    |-- courseId: "ALI-234"
    |-- courseInstructors: "Mary Ann Murphy"
    |-- courseName: "Intermediate Oral Skills"
  |- 1
    |-- courseId: "ALI-234"
    |-- courseInstructors: "Olivia Martinez"
    |-- courseName: "Intermediate Oral Skills"
  |- 2
    |-- courseId: "ALI-235"
    |-- courseInstructors: "Kimberley Briesch Sumner"
    |-- courseName: "Intermediate Writing Skills"
  |- 3
    |-- courseId: "ALI-235"
    |-- courseInstructors: "Juli Kirkpatrick, Richard Jones"
    |-- courseName: "Intermediate Writing Skills"
  |- 4
    ...
  ...

```

The screenshot above shows the structure of course database. We only recorded the course id, instructor names and course name for one particular course. There are some constraints of the course database. For example, USC has classes with different section time but same course id. Thus the combination of course id and instructors serves as a composite primary key. So, later, in the phase of data connecting and retrieval, we pass in both the values of course id and instructors to match record.

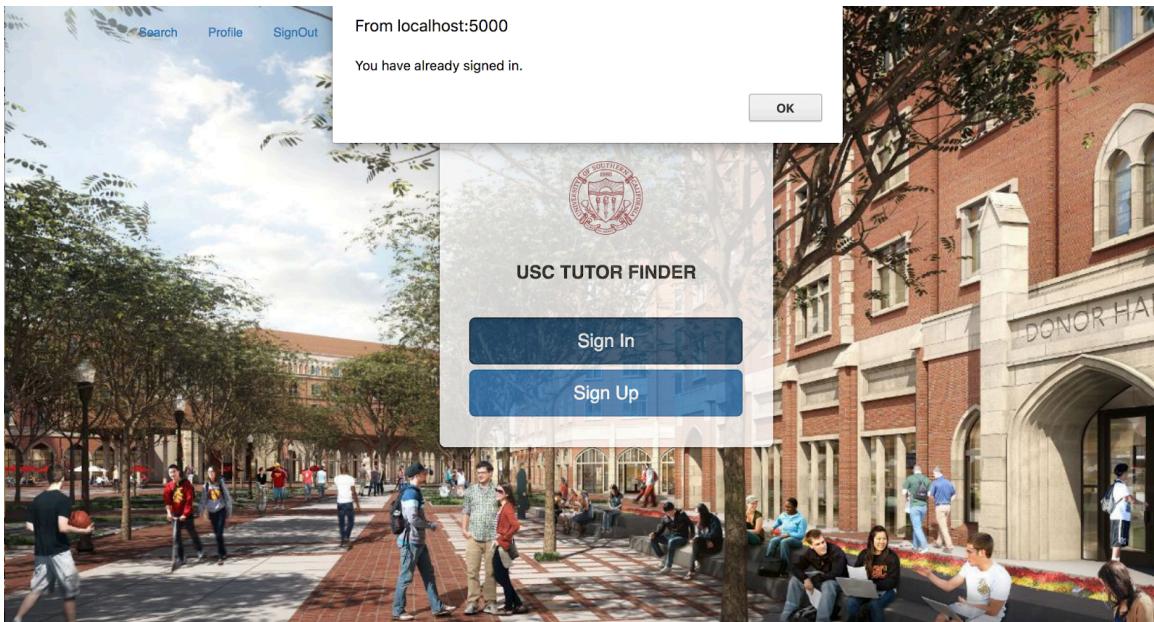
## 2. Website Pages and Backend

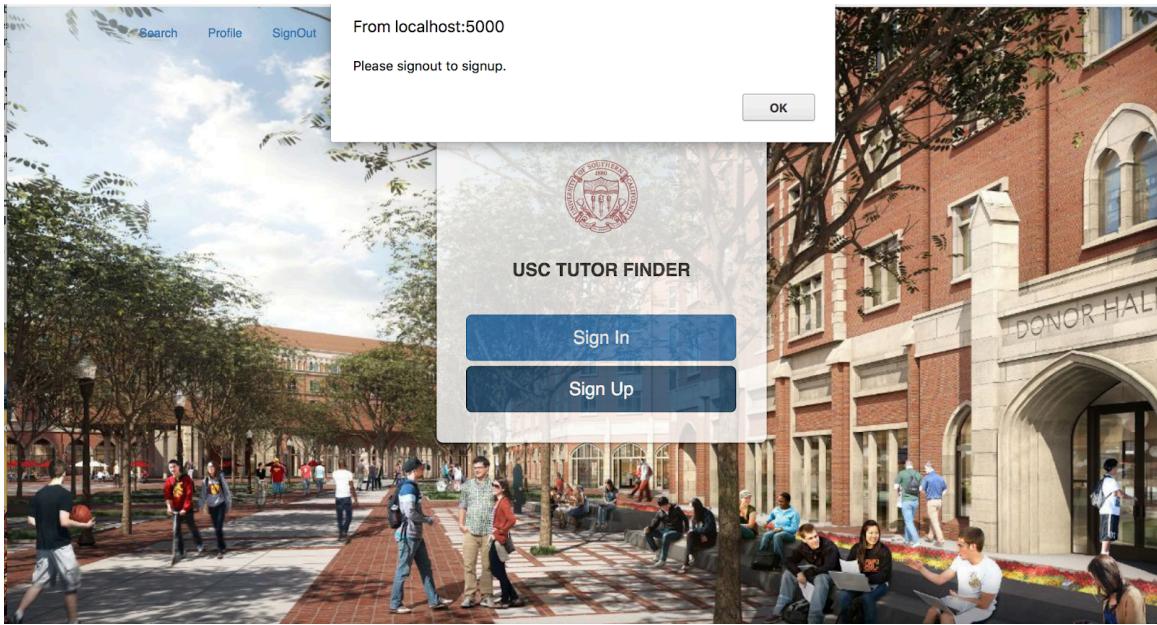
Let's dive into our user interface along with all the functions we have built for this website.



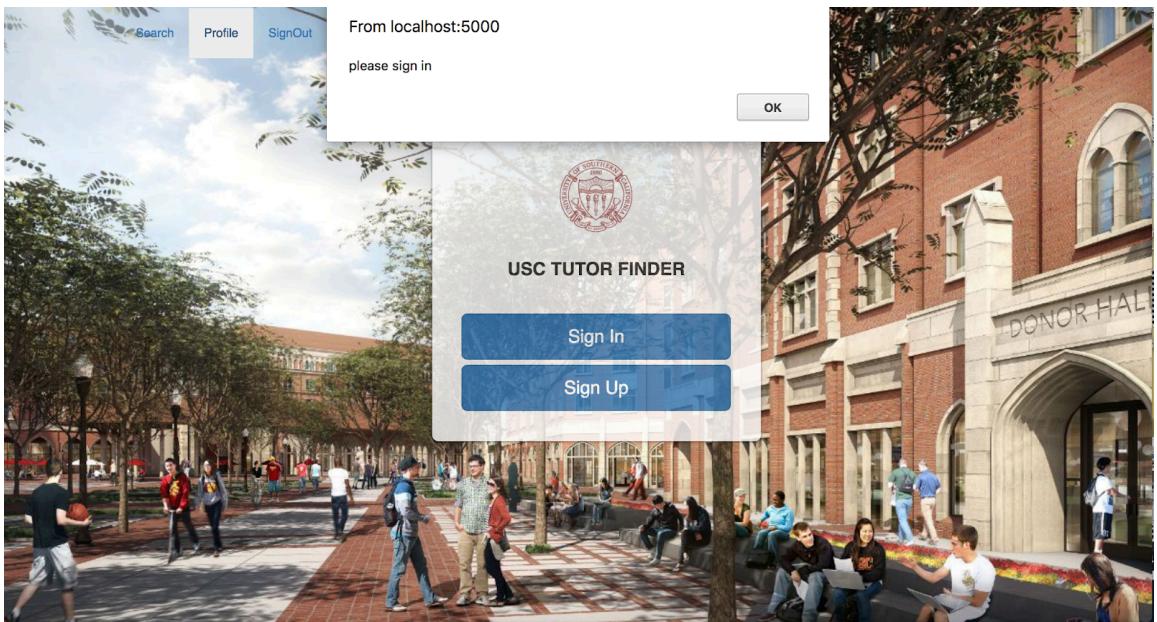
This is the index page. It has a SignIn and a SignUp box in the middle with Search, Profile and SignOut link are placed in the left upper corner. The functions of these pages are described below.

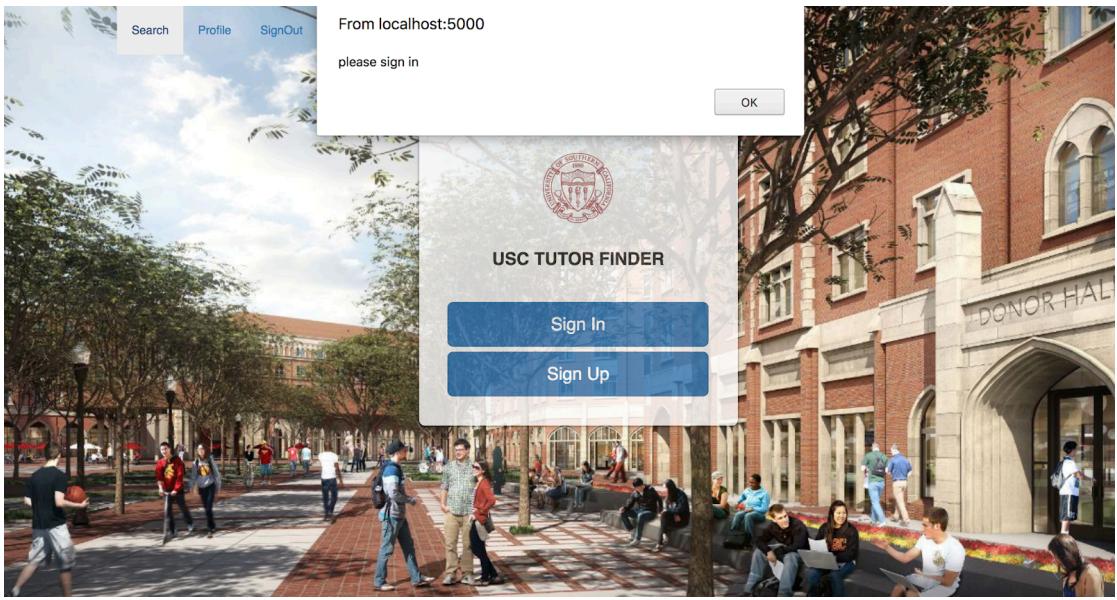
If one user has been signed in, he cannot sign in or sign up.



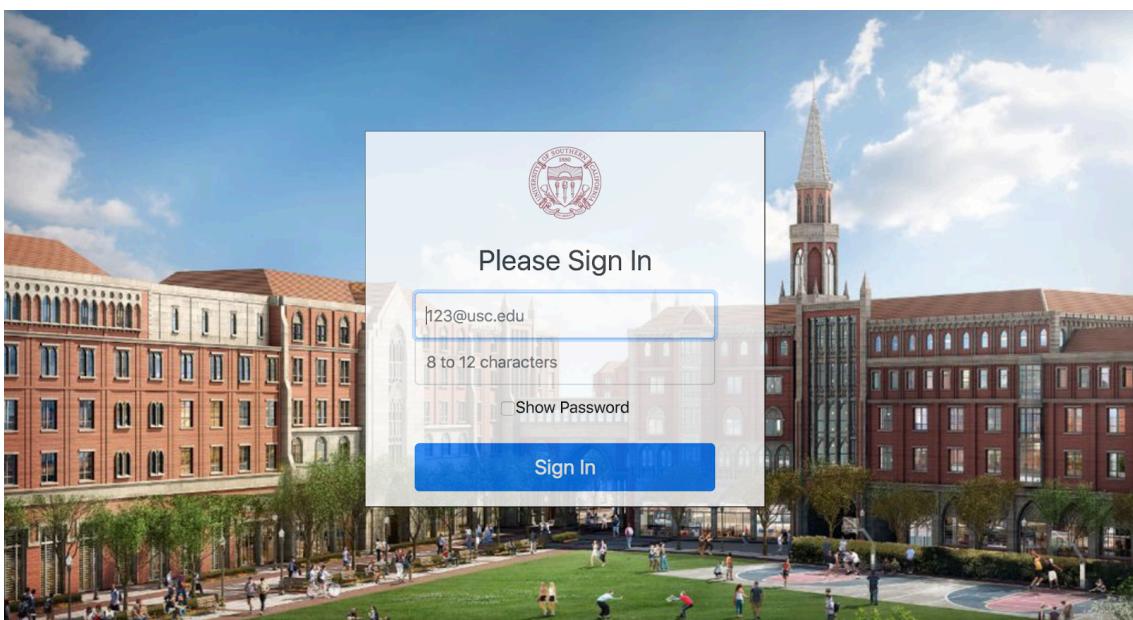


If a user isn't signed in, he can access neither search page nor profile page.

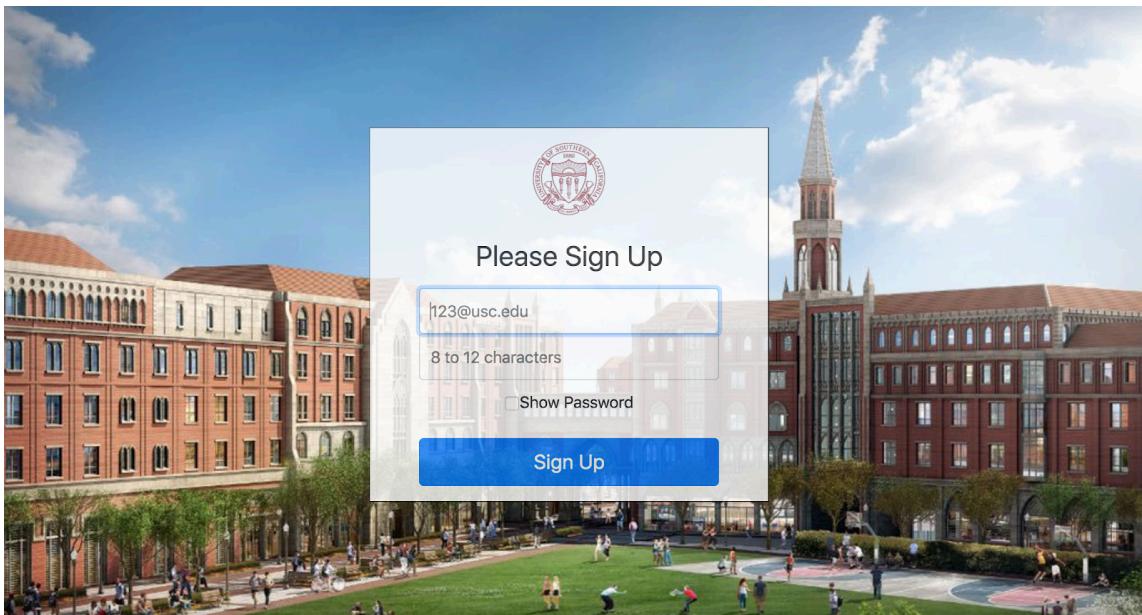




The followings are the SignIn and SignUp page.



After clicking “SignIn” in the Index page or being directed from the signup page, we will be on the SignIn page, which looks like the screenshot above. Users can type in the usc email, password and hit ‘sign in’, in the backend, typed email and password will be matched with corresponding information on Firebase. If the information is matched incorrectly, a warning would pop up and request reinput. If the backend approved the SignIn action, session will remember the user and redirect to search page.



In the “SignUp” page, we have the same layout as SignIn.html. Except that, instead of getting the information from Firebase in the backend, we post user typed information onto the Firebase. During the POST action, we will have a check process for the typed information to see if they go against our data format rules. After user successfully signs up, the page will be redirected to sign in page.

Course ID	Instructor(s)	Course Name	Tutors Available
ACAD-174	Doug Thomas	Innovators Forum	<input type="button" value="Not selected"/> <input type="checkbox"/> Willing to tutor this course?
ACAD-178	Aaron Siegel	Digital Toolbox: Motion Graphics	<input checked="" type="checkbox"/> Not selected <input type="button" value="ilu553@usc.edu"/> course?
ACAD-182	Steve Barth	Case Studies in Innovation	<input type="button" value="Not selected"/> <input type="checkbox"/> Willing to tutor this course?
ACAD-187	Aaron Siegel	Digital Toolbox: 3D Design	<input type="button" value="Not selected"/> <input type="checkbox"/> Willing to tutor this course?
ACAD-203	Bocheng Jin	Statics	<input type="button" value="Not selected"/> <input checked="" type="checkbox"/> Willing to tutor this course?
ACAD-204	Babak Bolouri-Azad	Strength of Materials	<input type="button" value="Not selected"/> <input type="checkbox"/> Willing to tutor this course?
ACAD-206	Taj Frazier	Communication and Culture	<input type="button" value="Not selected"/> <input type="checkbox"/> Willing to tutor this course?
ACAD-230	Alexis Zoto	3D Design: Materials and Tools	<input type="button" value="Not selected"/> <input type="checkbox"/> Willing to tutor this course?
ACAD-231L	Andrea Hodge, Ken-Ichi Nomura	Mechanical Behavior of Materials	<input type="button" value="Not selected"/> <input type="checkbox"/> Willing to tutor this course?

Search page (shown above) contains different methods of searching, including keyword, facet and fuzzy search. It has a table of columns named Course ID, Instructors, Course Name and Tutors Available. If users click on any of the four titles, sorting will be implemented for the columns

accordingly in ascending or descending order. In the Tutors Available column, a checkbox and a list of options are here for users to choose to be a tutor of one specific course and also provide a platform for them to select an available student tutor to help with that course. All operations availability are checked in the Flask and Firebase before operations were executed. At the top-right corner, we showed the email of the current user. Users can reach the profile page by clicking it.

The screenshot displays the USC Tutor Finder application interface with five main sections:

- Profile:** Shows user information: FirstName (Ke), LastName (Xu), Status (freshman), and Phone (1234567890). Includes a "submit" button.
- Search History:** Displays a search history entry: "inf". Includes a "clear all" button.
- Your Tutor:** A table listing tutors with columns: #, Course ID, Instructors, and Email. Entries include:
 

#	Course ID	Instructors	Email
>	ACAD-182	Steve Barth	lu553@usc.edu
>	ACAD-230	Alexis Zolo	kk@usc.edu
>	ACAD-178	Aaron Siegel	lu553@usc.edu
- Your Student:** A table listing students with columns: #, Course ID, Instructor, and Email. Entries include:
 

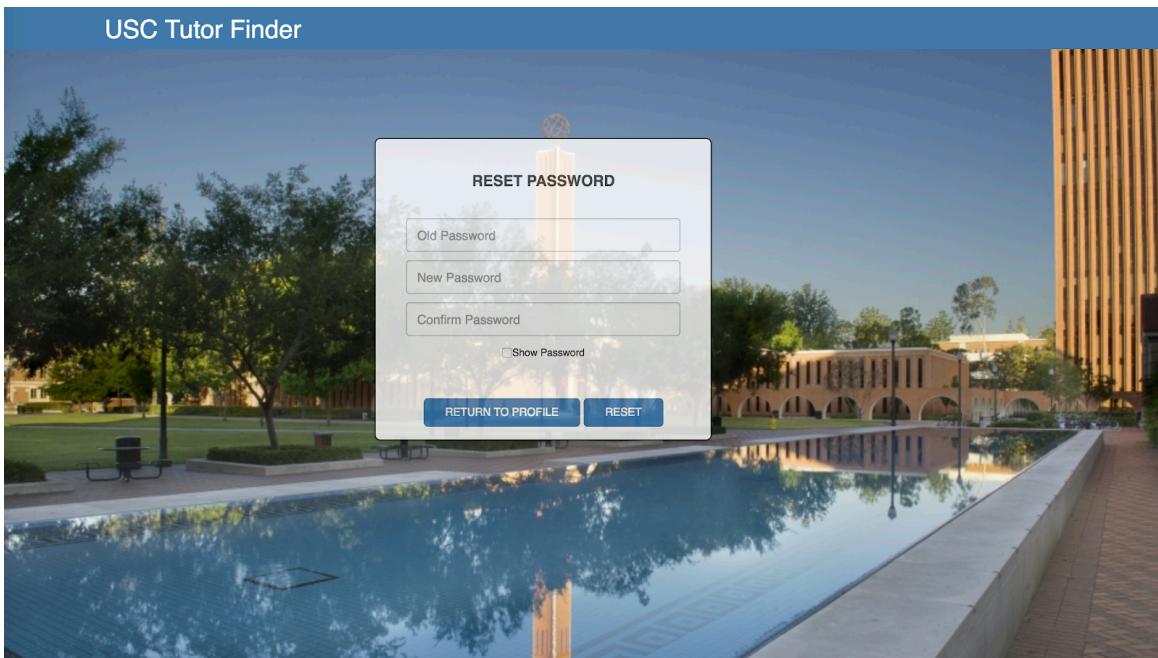
#	Course ID	Instructor	Email
>	ACAD-206	Taj Frazier	lu553@usc.edu
>	ACAD-245	Grant Delgatty	kk@usc.edu
>	ACAD-174	Doug Thomas	kk@usc.edu
- Your Will:** A table listing wills with columns: #, Course ID, Instructor. Entries include:
 

#	Course ID	Instructor
>	ACAD-187	Aaron Siegel
>	ACAD-174	Doug Thomas
>	ACAD-203	Bocheng Jin
>	ACAD-245	Grant Delgatty
>	ACAD-206	Taj Frazier
>	ACAD-354	Andrew Zacharias

The purpose of the profile page is to show users all the currently information they have. Also, each of information sections supports editing by clicking on the setting icon and one user can make

changes to the form by clicking on or typing in the newly appeared textboxes or buttons. After user submits the new changes, Flask will check the input format and then update the Firebase. Besides, this webpage could also show search history, tutor student pair, and the willingness to be a tutor. Users can delete these records by one clicking. And the backend will check these executions and execute these operations in the Firebase accordingly. At the left side of this profile page, it has a column of hyperlinks. These links can direct current user into the pages listed on there. For the SignOut link, our server will clear out the session.

USC Tutor Finder also has a password reset page, which looks like below.



Users are allowed to reset their password. To execute this operation, user must input old password correctly. The server will check the correctness of two inputs of new passwords and then reset the password in the Firebase.

### **3. Individual Work and Conclusions.**

In this project, Ke Xu mainly focused on the backend construction and Xueping Li mainly focused on the frontend implementation and while working on the project, we also did some work of the other person. From this project, we gain some experience in the cloud-based database, especially Google Firebase. However, it does have pros and cons. The pros are that Firebase is easy to use and structured. We can manually add the row since Firebase have a beautifully formatted GUI, The cons of Firebase are that it does not support a lot of query or relational operations, like join or groupby, we need to implement it by ourselves. Besides, this project provides us a deeper understanding on web page development in both frontend and backend. We are now more familiar with Flask, Bootstrap, Javascript, html, CSS and are able to implement them into a real project.

## 4. Code documentation

