Spring ‘23 COSC 386 Project Proposal

GULLCODE Website/Database

# GROUP MEMBERS:

Brody Channell

George Fotiou

Jade Pearl

John Schmidt

# PROJECT DESCRIPTION:

GullCode is an annual coding competition where SU students can compete in teams to finish coding challenges. Managing the registration and participating teams can be challenging for the Math & Computer Science club that hosts the event. Thus, there is an opportunity to assist GullCode with the creation of a public website to facilitate the event, hence our motivation to create this project. The purpose of the database will be to store information needed to support this newly created GullCode website. The website will have the following features:

* Allow students to create their own coding profile using their SU email
* Allow students to see the format of previous coding challenges
* Allow students to register for the upcoming GullCode event (as a team)
* Anyone can access: pictures from previous events (if any exist)
* Anyone can access: the history of GullCode
* Anyone can access: upcoming GullCode information
* Anyone can access: basic profile information about previous winners (name, emails)
* Links to Leetcode, Codewars, and other helpful resources

Additionally, the website should have intuitive and attractive controls for both students and admin access. The primary clients for the database (and site) are the administrators, and students are the secondary clients. The admin will be able to upload current-year SU students’ emails and ID numbers in a batch. This should allow students to create a new profile via their ID and email (having a verification email sent to their email address), without the need for admin intervention. Alternatively (and less desirably), each student can email an admin for access directly. Once a student's information is in the database, they may log in to view practice problems and request a team. The administrator will have access to approve or reject requested teams, and can also view and modify winning team statistics. The administrator will also be able to modify the information displayed on the welcome page without needing to modify the PHP code.

# REQUIREMENTS:

* The databases keep track of S.U. **STUDENTS**, keeping their unique S.U. *ID\_Number*, their *firstName*, *lastName*, *SU\_Email*, *personalEmail*, *versionControl* (i.e. GitHub), *website*, and also *shirtSize* (because participants are often given free shirts). Each student will be able to request an entry with a team by including other students’ information and a unique *teamName* that hasn’t been used previously in that year. This will add the basic team information to **TEAMS** as pending, which can be approved by an **ADMINISTRATOR**
* Each **TEAMS** will have a unique *teamName* and *year* in which they are participating. Teams be allowed to duplicate names over various years. This table will contain the current application status for the team (pending, approved, or denied). It will also include the final *placement* of the team following the event.
* The **ADMINISTRATORS** will record the administrators that currently have access to the database. These tuples will contain a unique S.U. *ID\_Number*, *SU\_Email*, *personalEmail*, *versionControl*, and *website*. The **ADMINISTRATORS** will have access to modification of all database components.
* The **JUDGES** will contain an *iD\_Number*, which will be a student or faculty ID, the *firstName* and *lastName* of the judge, their *email* (personal or work/school), and their *title* (student, professor, Dr. etc). This information will be used to display who is judging each entry. It is assumed that there may be judges that are qualified for some submissions but not others, based on possibility of multiple submitted languages, etc. The **JUDGES** will have access to modify the **JUDGE\_GRADES**.
* **TEAM\_PARTICIPANTS** will link the student information (iD\_Number) with the team they are on (*teamName* and *year*). This will allow for the number of students in a team to vary without the need for modificaiton of the schema.
* **CODING\_CHALLENGES** will contain the descriptions and information needed pertaining to the current (and past) challenges. It will include a unique *challenge\_ID* (ramdomly generated is acceptable), a *challengeName* that could briefly describe it, a *description* that will describe it in full, the *year* in which it was used, *languages* permitted for the solution, estimated *difficulty*, and the *rules* particular to this challenge. Note that assigning only a unique ID as the primary key for this table will allow for repeats with variations on the rules or language without the need to change the name.
* **TEAM\_SUBMISSIONS** will serve to link the team with its final submission for consideration by the judges. It will contain the foreign keys necessary for the primary key (*teamName*, *year*, and *challengeID*). It will also hold the *submission* via formatted text, the language to be used for compilation, and the *totalScore* assigned by the judges.
* **JUDGE\_GRADES** will keep track of the individual scores every judge can issue to a **TEAM\_SUBMISSIONS** submission. The *iD\_Number* of the judge will be used to identify which judge is issuing the score, while the *teamName*, *year*, and *challengeID* will be used to identify which submission is being referenced. The *score* will be stored, as well as any *comments* pertaining to the reasoning for that score.
* It is assumed that the database will store primarily textual data for some portion of the student body per year ( < 1000), so storage requirements should be minimal. Less than 1 GB per event year should be sufficient. If pictures are able to be integrated into each profile, more may be required ( ~2GB / year).

# SCHEMA:

students(

iD\_Number: string,

firstName: string,

lastName: string,

SU\_Email: string,

personalEmail: string,

versionControl: string,

website: string

shirtSize: string,

)

teams(

teamName: string,

year: integer,

applicationStatus: string,

placement: string

)

administrators(

iD\_Number: string,

firstName: string,

lastName: string,

SU\_Email: string,

personalEmail: string,

versionControl: string,

website: string

)

judges(

iD\_Number: string

firstName: string,

lastName: string,

email: string,

title: string,

)

team\_participants(

teamName: string,

year: integer,

iD\_Number: string,

foreign key (teamName, year) references teams(teamName, year),

foreign key (iD\_Number) references students(iD\_Number)

)

coding\_Challenges(

challengeID: string,

challengeName: string,

description: string,

year: int,

languages: string,

difficulty: string,

rules: string

)

team\_Submissions(

teamName: string,

year: integer,

challengeID: string,

submission: string,

language: string,

totalScore: float

)

judge\_Grades(

iD\_Number: string,

teamName: string,

year: integer,

challengeID: string,

score: float,

comments: string,

foreign key (iD\_Number) references judges(iD\_Number),

foreign key (teamName, year, challengeID) references teamSubmissions(teamName, year, challengeID)

)

# DATABASE PROTOTYPE DESIGN:

# 

# INTERFACE SKETCH:

# TEAM PLAN:

Meet over Zoom and in person if and when available. There will be regular communication about the project over Google Drive and GroupMe sms app as well.

Draft of project duties:

* Everyone:
  + Proposal drafting and edits
  + Overall edits to the database
  + Feedback and suggestions
* John (group leader):
  + Website mock-up draft (John made this already)
  + Initial creation of the ER chart on lucid chart
  + DESIGNATED PART OF THE DATABASE CREATION PROCESS (i.e. Admin page)
* George:
  + ER Design draft
  + Edits for database
  + DESIGNATED PART OF THE DATABASE CREATION PROCESS (i.e. Admin page)
* Jade:
  + Database Logo Design (possibly AI generated)
  + ER design draft
  + DESIGNATED PART OF THE DATABASE CREATION PROCESS (i.e. Admin page)
* Brody:
  + Edits for database
  + ER design draft
  + DESIGNATED PART OF THE DATABASE CREATION PROCESS (i.e. Admin page)

Note: The selected part of the database that each member will work on has yet to be assigned or decided.