Phase 2 Report:

DB Design:

- Our SQL schema meets the 3rd Normal Form (3NF) with some violations. Here's the justification for each normal form:
  1st Normal Form (1NF):
  - All tables have a primary key.
  - All columns have atomic values (no repeating groups or arrays).

  2nd Normal Form (2NF):
  - All tables meet 1NF.
  - All non-prime attributes (attributes that are not part of the primary key) are fully dependent on the primary key.

  3rd Normal Form (3NF):
  - All tables meet 2NF.
  - All non-prime attributes are non-transitively dependent on the primary key.

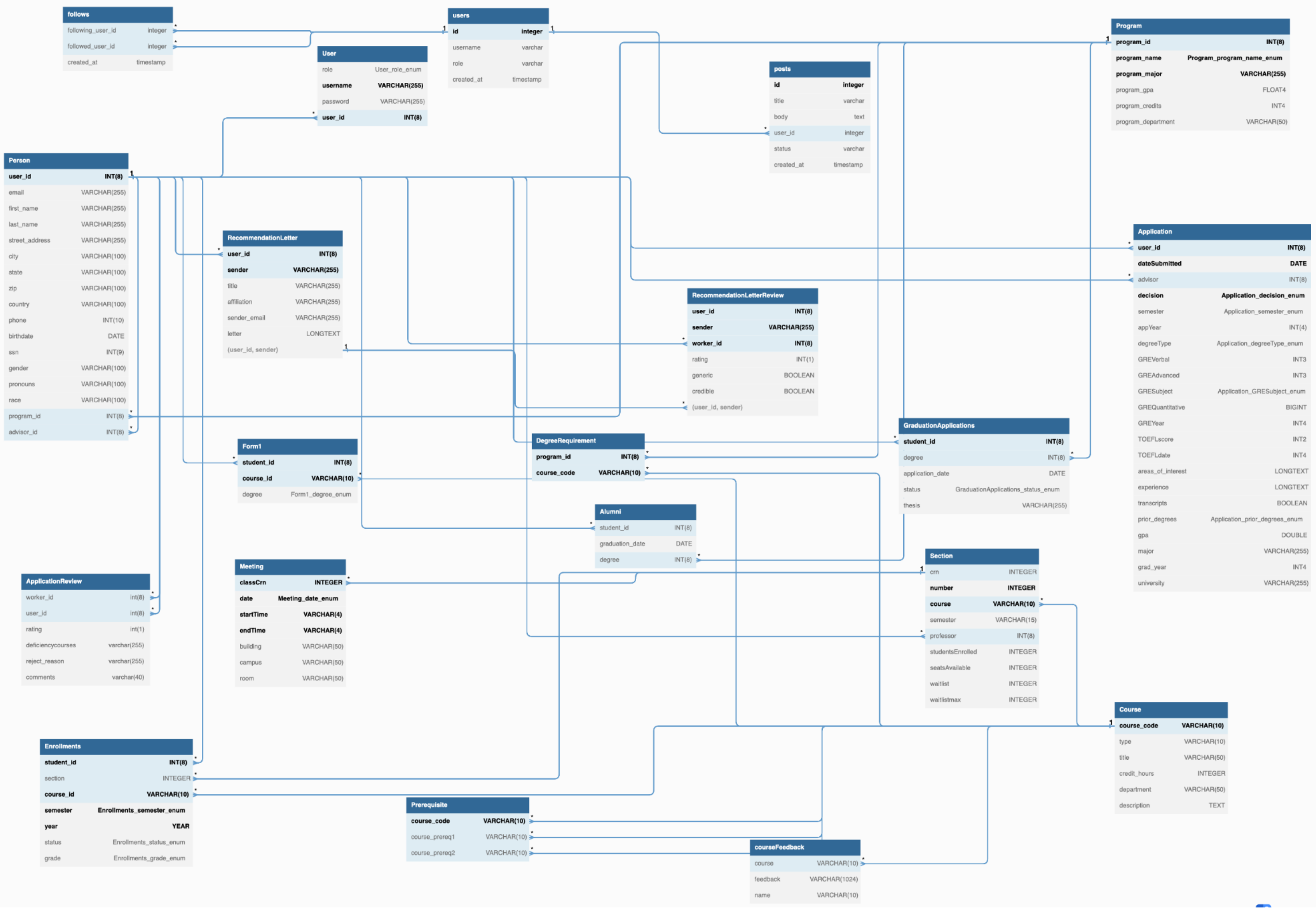
  There are some violations in the schema where the tables don't completely adhere to the 3NF:
  1. In the 'Person' table, the columns 'street_address', 'city', 'state', 'zip', and 'country' are all attributes of an address, which is not directly related to the primary key 'user_id'. A separate 'Address' table with a one-to-many relationship to the 'Person' table can be created to normalize this table.
  2. In the 'Course' table, the 'department' attribute is not directly related to the primary key 'course_code'. To resolve this, a separate 'Department' table can be created, and the 'Course' table can have a foreign key reference to the 'Department' table.

  By addressing these violations, the schema can be further normalized. However, in its current form, the schema mostly meets the 3rd Normal Form.

Design Justification:

- We designed our database schema so that the users and their loggins are separate from the user's data in the database. Our Person table maintains the primary key for user_id that is used throughout the database and then the User table has the user_id as a foreign key and this table tracks what type of user they are and their loggins. This allows us to have multiple loggins for the same user's set of data. This reason for this is so that when an applicant becomes a student they will be given a new username and password generated by the university to log into their student account and they can still access their application data and status by logging in with their original account they applied to the university with.

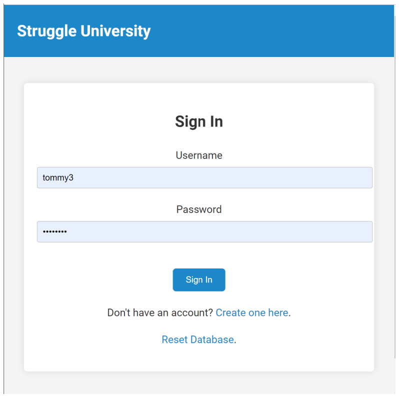
ER Diagram:


follows
following_user_id integer
followed_user_id integer
created_at timestamp
User
role User_role_enum
username VARCHAR(255)
password VARCHAR(255)
user_id INT(8)
users
id integer
username varchar
role varchar
created_at timestamp
posts
id integer
title varchar
body text
user_id integer
status varchar
created_at timestamp
Program
program_id INT(8)
program_name Program_program_name_enum
program_major VARCHAR(255)
program_gpa FLOAT4
program_credits INT4
program_department VARCHAR(50)
Person
user_id INT(8)
email VARCHAR(255)
first_name VARCHAR(255)
last_name VARCHAR(255)
street_address VARCHAR(255)
city VARCHAR(100)
state VARCHAR(100)
zip VARCHAR(100)
country VARCHAR(100)
phone INT(10)
birthdate DATE
ssn INT(9)
gender VARCHAR(100)
pronouns VARCHAR(100)
race VARCHAR(100)
program_id INT(8)
advisor_id INT(8)
RecommendationLetter
user_id INT(8)
sender VARCHAR(255)
title VARCHAR(255)
affiliation VARCHAR(255)
sender_email VARCHAR(255)
letter LONGTEXT
(user_id, sender)
Application
user_id INT(8)
dateSubmitted DATE
advisor INT(8)
decision Application_decision_enum
semester Application_semester_enum
appYear INT(4)
degreeType Application_degreeType_enum
GREVerbal INT3
GREAdvanced INT3
GRESubject Application_GRESubject_enum
GREQuantitative BIGINT
GREYear INT4
TOEFLscore INT2
TOEFLdate INT4
areas_of_interest LONGTEXT
experience LONGTEXT
transcripts BOOLEAN
prior_degrees Application_prior_degrees_enum
gpa DOUBLE
major VARCHAR(255)
grad_year INT4
university VARCHAR(255)
RecommendationLetterReview
user_id INT(8)
sender VARCHAR(255)
worker_id INT(8)
rating INT(1)
generic BOOLEAN
credible BOOLEAN
(user_id, sender)
GraduationApplications
student_id INT(8)
degree INT(8)
application_date DATE
status GraduationApplications_status_enum
thesis VARCHAR(255)
Form1
student_id INT(8)
course_id VARCHAR(10)
degree Form1_degree_enum
DegreeRequirement
program_id INT(8)
course_code VARCHAR(10)
Alumni
student_id INT(8)
graduation_date DATE
degree INT(8)
Section
crn INTEGER
number INTEGER
course VARCHAR(10)
semester VARCHAR(15)
professor INT(8)
studentsEnrolled INTEGER
seatsAvailable INTEGER
waitlist INTEGER
waitlistmax INTEGER
ApplicationReview
worker_id int(8)
user_id int(8)
rating int(1)
deficiencycourses varchar(255)
reject_reason varchar(255)
comments varchar(40)
Meeting
classCrn INTEGER
date Meeting_date_enum
startTime VARCHAR(4)
endTime VARCHAR(4)
building VARCHAR(50)
campus VARCHAR(50)
room VARCHAR(50)
Course
course_code VARCHAR(10)
type VARCHAR(10)
title VARCHAR(50)
credit_hours INTEGER
department VARCHAR(50)
description TEXT
Enrollments
student_id INT(8)
section INTEGER
course_id VARCHAR(10)
semester Enrollments_semester_enum
year YEAR
status Enrollments_status_enum
grade Enrollments_grade_enum
Prerequisite
course_code VARCHAR(10)
course_prereq1 VARCHAR(10)
course_prereq2 VARCHAR(10)
courseFeedback
course VARCHAR(10)
feedback VARCHAR(1024)
name VARCHAR(10)

Visual Overview:

Apps Visual overview

First an applicant can create an account then log in (login shown)



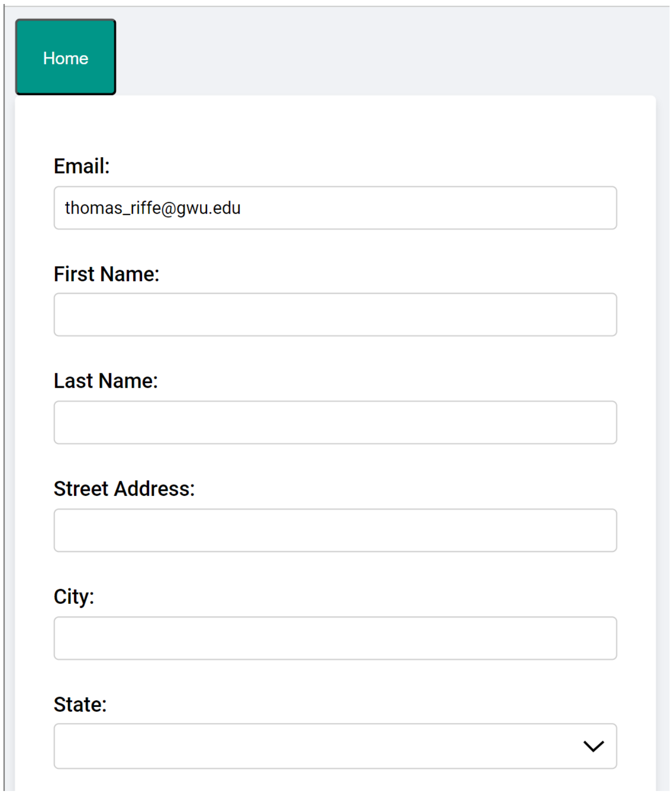
Username: tommy3

Password: ········

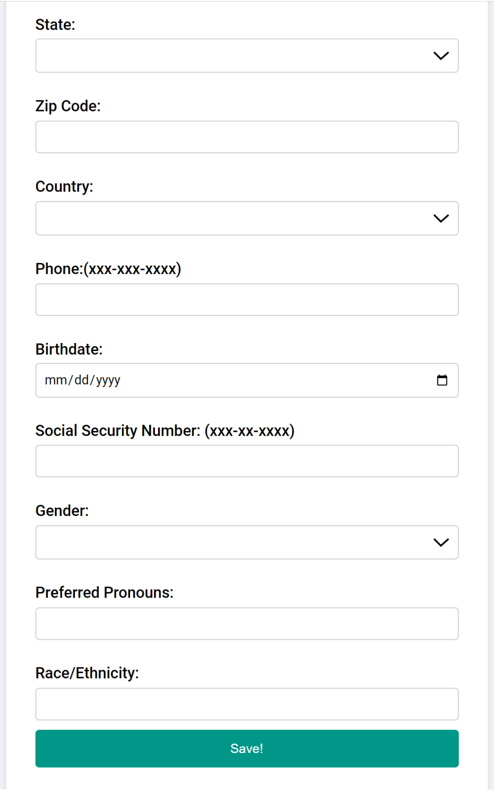
Don't have an account? Create one here.

Reset Database.

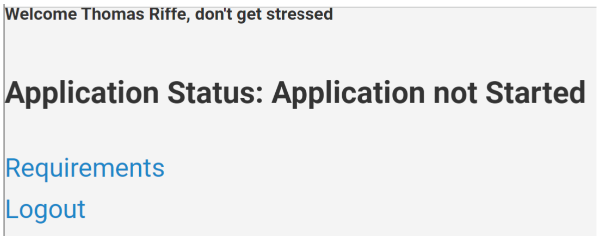
The applicant is then redirected to the personal information page which they are forced to complete before using any other page in the database. These pictures show the



Email: thomas_riffe@gwu.edu



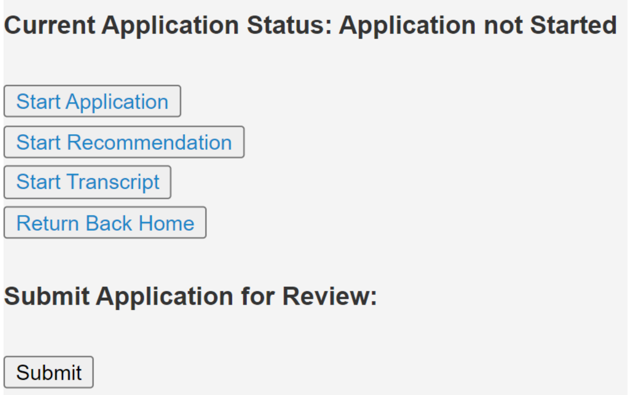
After this the applicant can then access their home page which allows them to access the requirements page as well. Both shown below. When the student receives the decision they will be shown the decision along with their advisor on the home page.



Welcome Thomas Riffe, don't get stressed

Application Status: Application not Started

Requirements
Logout



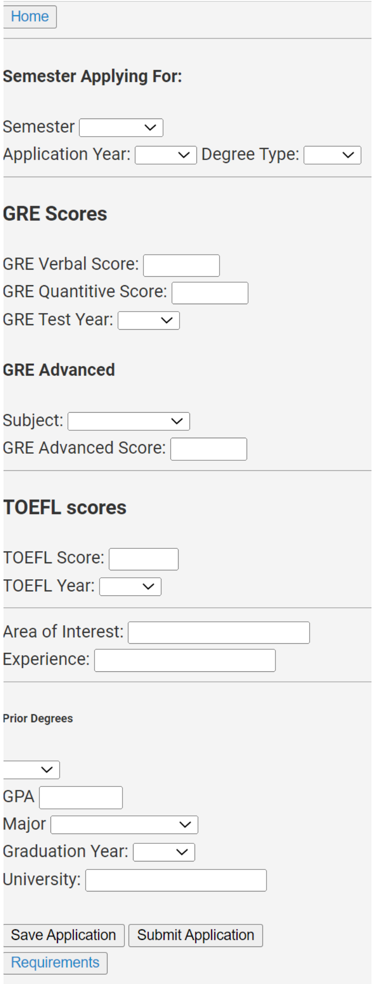
Current Application Status: Application not Started

Start Application
Start Recommendation
Start Transcript
Return Back Home

Submit Application for Review:

Submit

Then From the requirements page they can go to the application page. The application page supports the user saving for later and when they return it will repopulate their previous responses. When they click submit it will lock the form so that they can only view their response and they can return to see their previous response at any time.



Home

Semester Applying For:

Semester
Application Year: Degree Type:

GRE Scores

GRE Verbal Score:
GRE Quantitive Score:
GRE Test Year:

GRE Advanced

Subject:
GRE Advanced Score:

TOEFL scores

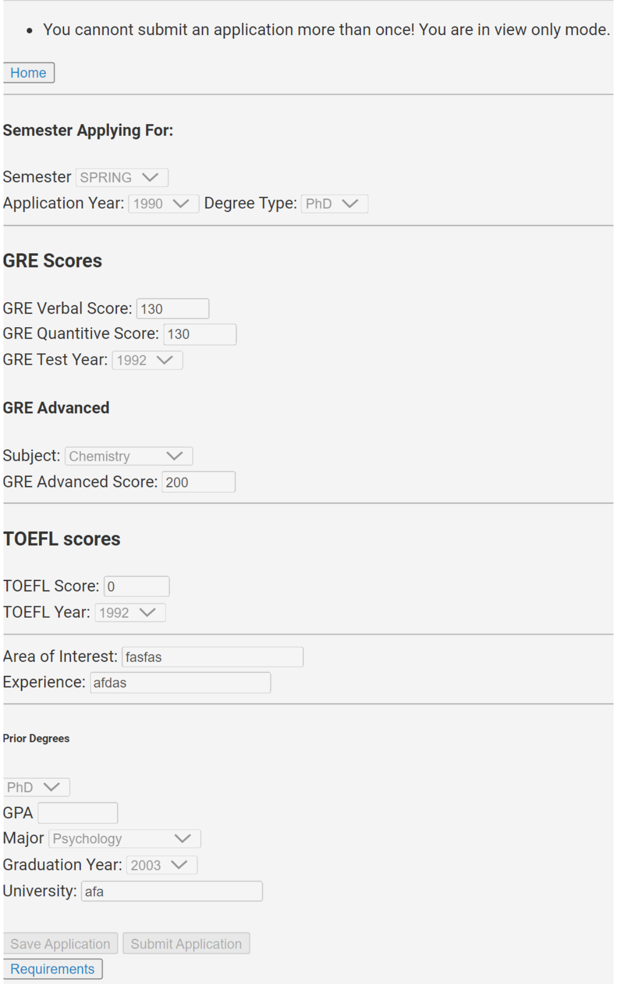
TOEFL Score:
TOEFL Year:

Area of Interest:
Experience:

Prior Degrees

GPA
Major
Graduation Year:
University:

Save Application Submit Application
Requirements



- You cannot submit an application more than once! You are in view only mode.

Home

Semester Applying For:

Semester SPRING
Application Year: 1990 Degree Type: PhD

GRE Scores

GRE Verbal Score: 130
GRE Quantitive Score: 130
GRE Test Year: 1992

GRE Advanced

Subject: Chemistry
GRE Advanced Score: 200

TOEFL scores

TOEFL Score: 0
TOEFL Year: 1992

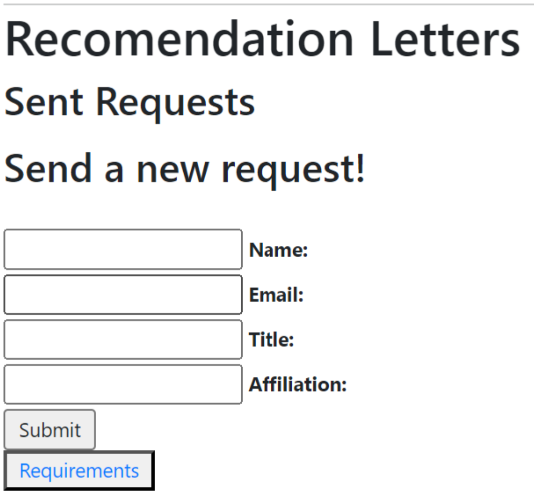
Area of Interest: fasfas
Experience: afdas

Prior Degrees

PhD
GPA
Major Psychology
Graduation Year: 2003
University: afa

Save Application Submit Application
Requirements

After an application is submitted the user can send out their recommendation letters from the requirements page. The applicant can then type in the Recomenders name to sit



Recomendation Letters

Sent Requests

Send a new request!

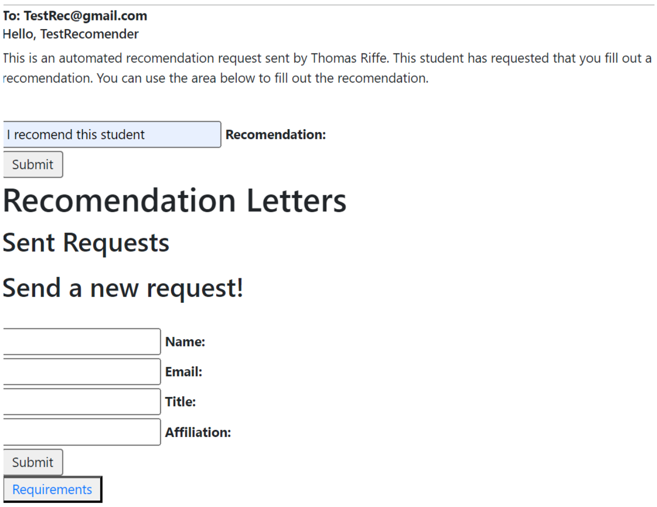
Name:

Email:

Title:

Affiliation:

then the applicant can enter the recommender's information and it will ‘send’ and email to the recommender which just displays a message to the recommender and allows them to enter a recommendation to the user.



To: TestRec@gmail.com

Hello, TestRecomender

This is an automated recomendation request sent by Thomas Riffe. This student has requested that you fill out a recomendation. You can use the area below to fill out the recomendation.

I recomend this student **Recomendation:**

Recomendation Letters

Sent Requests

Send a new request!

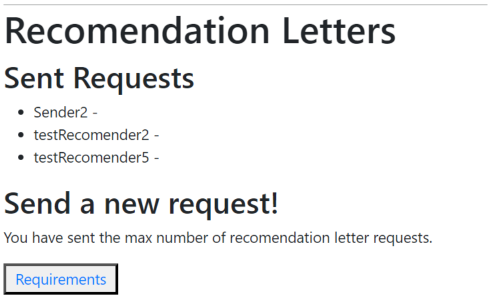
Name:

Email:

Title:

Affiliation:

The applicant can see the list of people they sent their previous letters to and when they have sent their limit of 3 letters they are no longer able to send anymore.



Recomendation Letters

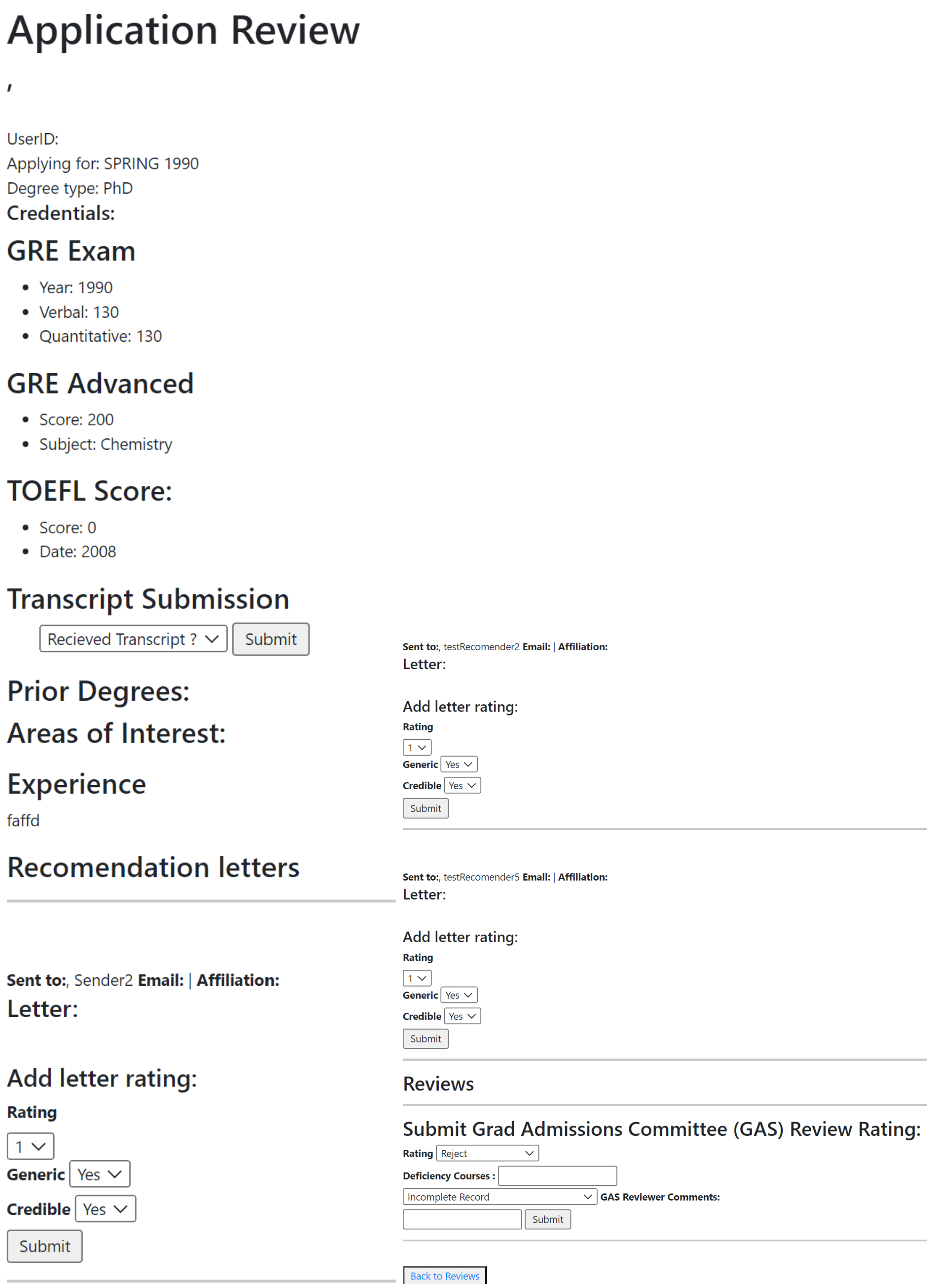
Sent Requests

- Sender2 -
- testRecomender2 -
- testRecomender5 -

Send a new request!

You have sent the max number of recomendation letter requests.

The Application review page changes dynamically based on who is viewing the form so for the reviewer view below which they can only submit reviews of the the recommendation letters and the applicant as a whole.



Application Review

,

UserID:
Applying for: SPRING 1990
Degree type: PhD
Credentials:

GRE Exam

- Year: 1990
- Verbal: 130
- Quantitative: 130

GRE Advanced

- Score: 200
- Subject: Chemistry

TOEFL Score:

- Score: 0
- Date: 2008

Transcript Submission

Recieved Transcript ? ▾ | Submit

Prior Degrees:

Areas of Interest:

Experience

faffd

Recomendation letters

Sent to:, Sender2 **Email:** | **Affiliation:**

Letter:

Add letter rating:

Rating
1 ▾
Generic Yes ▾
Credible Yes ▾
Submit

Sent to:, testRecomender2 **Email:** | **Affiliation:**

Letter:

Add letter rating:

Rating
1 ▾
Generic Yes ▾
Credible Yes ▾
Submit

Sent to:, testRecomender5 **Email:** | **Affiliation:**

Letter:

Add letter rating:

Rating
1 ▾
Generic Yes ▾
Credible Yes ▾
Submit

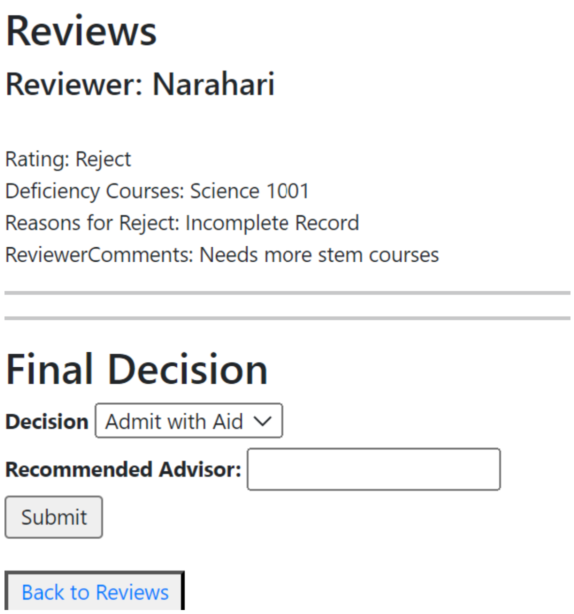
Reviews

Submit Grad Admissions Committee (GAS) Review Rating:

Rating Reject ▾
Deficiency Courses :
Incomplete Record ▾ **GAS Reviewer Comments:**
Submit

Back to Reviews

The GS view shows the reviews and the ability to update the final decision



Reviews

Reviewer: Narahari

Rating: Reject
Deficiency Courses: Science 1001
Reasons for Reject: Incomplete Record
ReviewerComments: Needs more stem courses

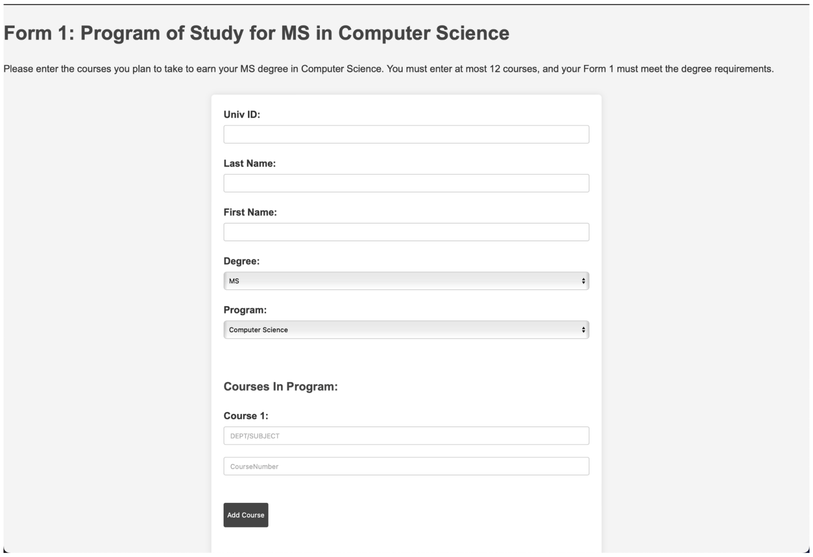
Final Decision

Decision Admit with Aid

Recommended Advisor:

Advising Visual Overview:

- Filling out form 1



Form 1: Program of Study for MS in Computer Science

Please enter the courses you plan to take to earn your MS degree in Computer Science. You must enter at most 12 courses, and your Form 1 must meet the degree requirements.

Univ ID:

Last Name:

First Name:

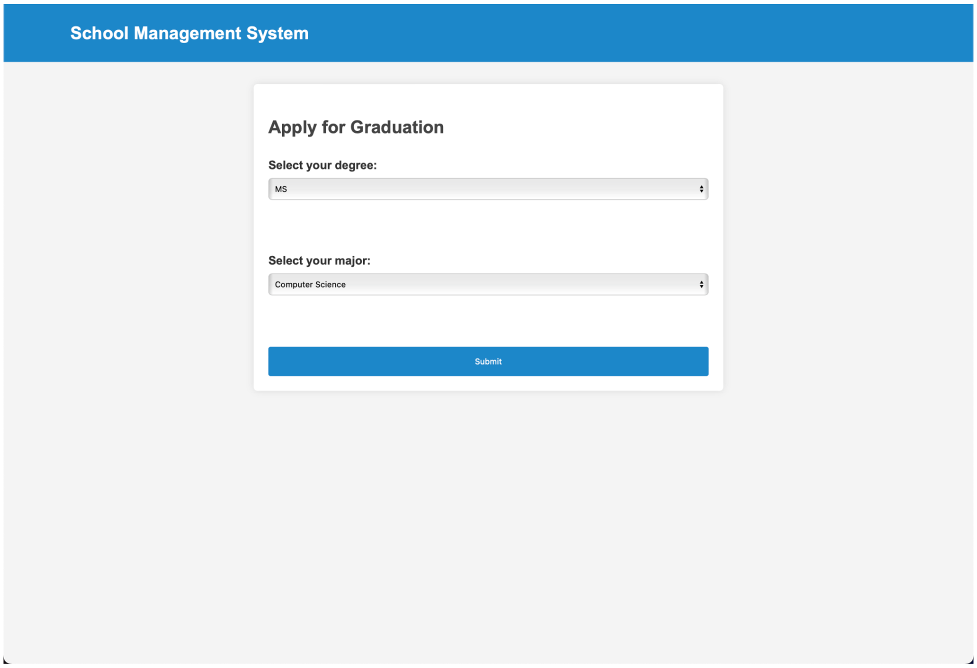
Degree: MS

Program: Computer Science

Courses In Program:

Course 1:

- Applying to graduate


School Management System

Apply for Graduation

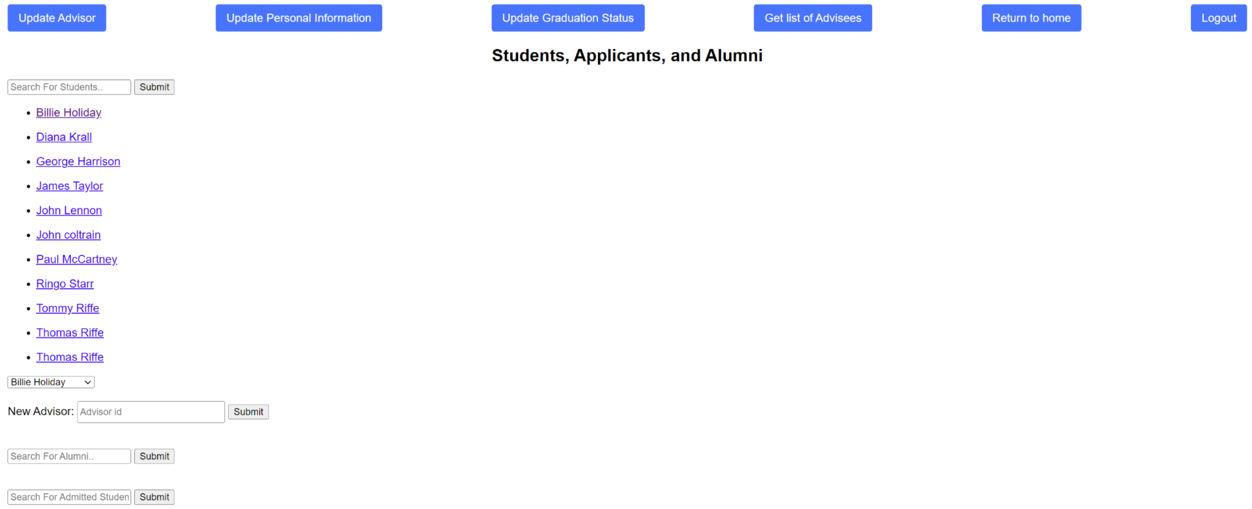
Select your degree:

MS

Select your major:

Computer Science

- GS Home



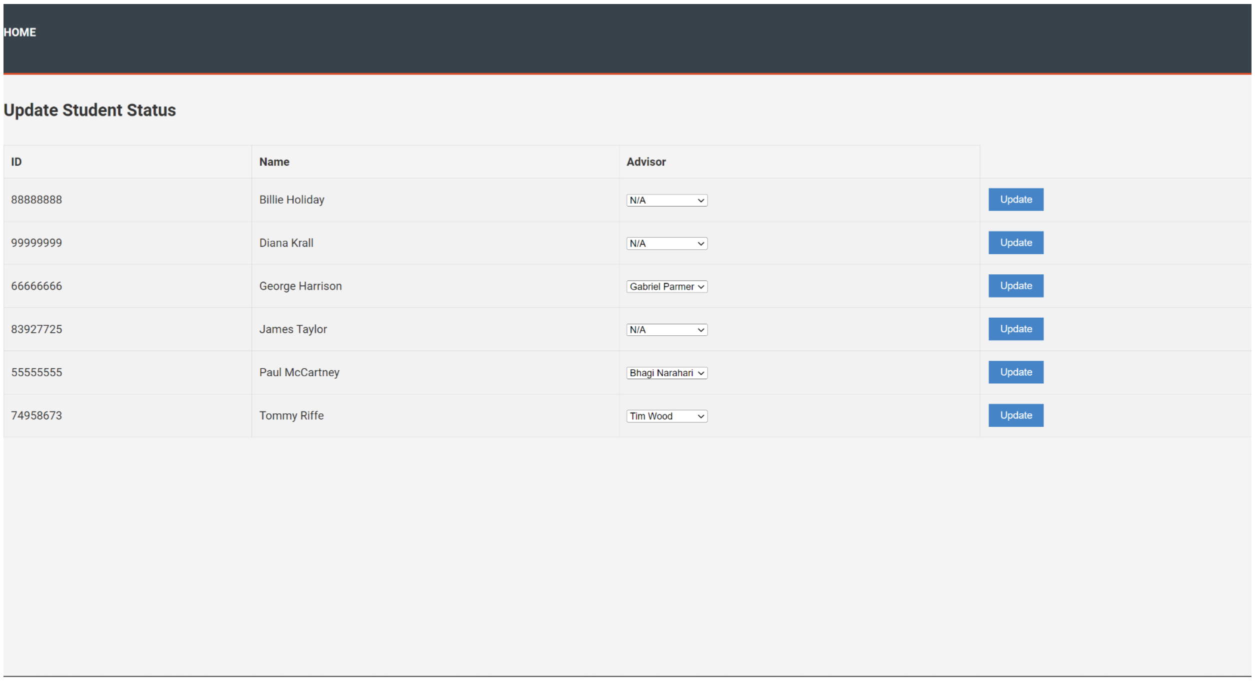
Students, Applicants, and Alumni

- Billie Holiday
- Diana Krall
- George Harrison
- James Taylor
- John Lennon
- John coltrain
- Paul McCartney
- Ringo Starr
- Tommy Riffe
- Thomas Riffe
- Thomas Riffe

Billie Holiday

New Advisor: Advisor id

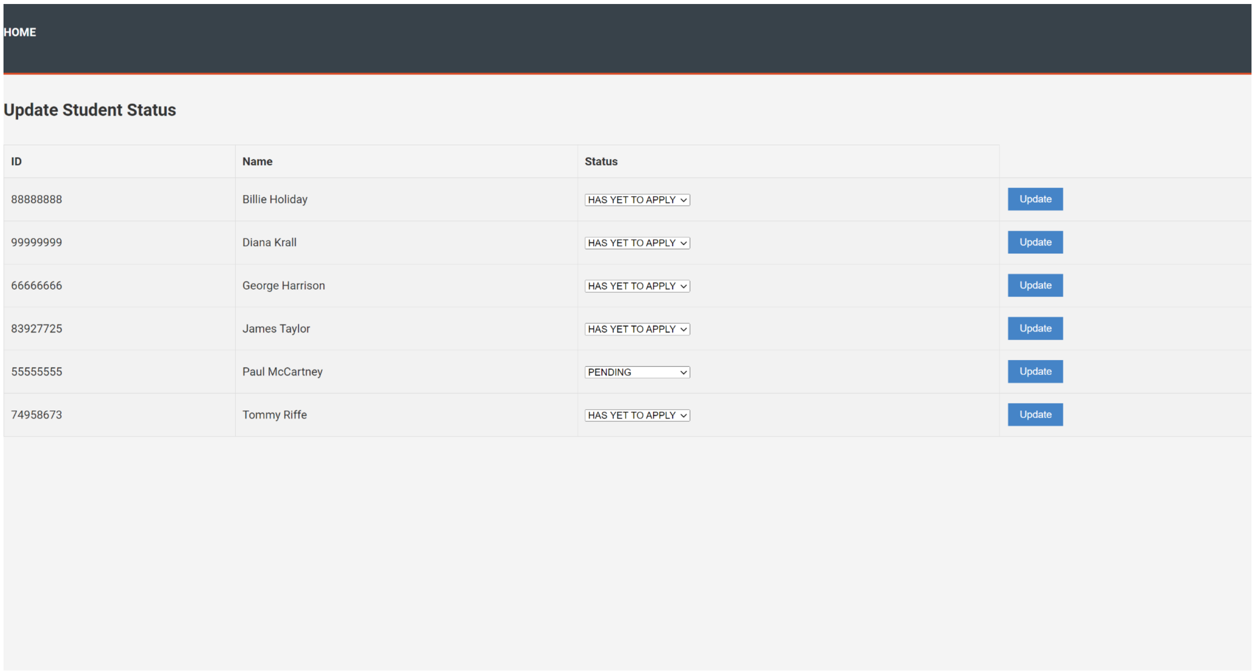
- Assigning an advisor



Update Student Status

ID	Name	Advisor	
88888888	Billie Holiday	N/A	Update
99999999	Diana Krall	N/A	Update
66666666	George Harrison	Gabriel Parmer	Update
83927725	James Taylor	N/A	Update
55555555	Paul McCartney	Bhagi Narahari	Update
74958673	Tommy Riffe	Tim Wood	Update

- Approving a graduation request

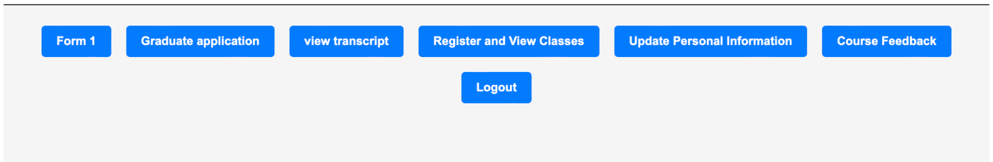


Update Student Status

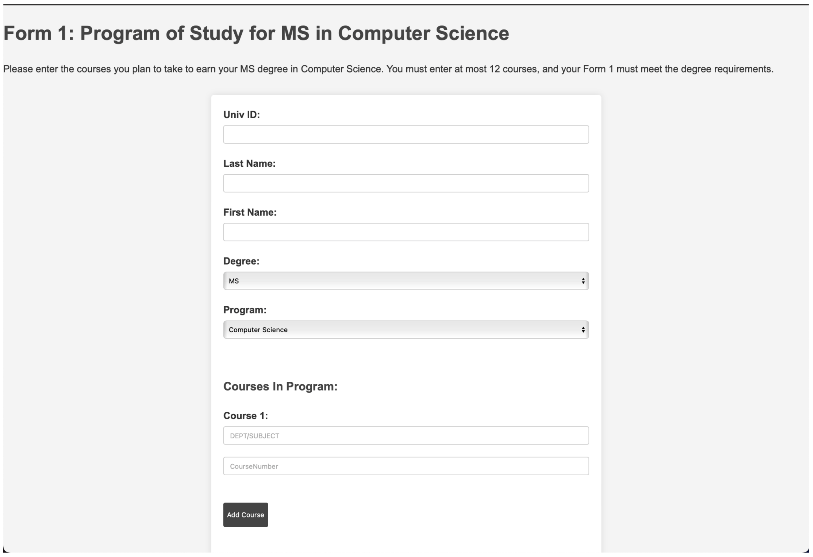
ID	Name	Status	
88888888	Billie Holiday	HAS YET TO APPLY	Update
99999999	Diana Krall	HAS YET TO APPLY	Update
66666666	George Harrison	HAS YET TO APPLY	Update
83927725	James Taylor	HAS YET TO APPLY	Update
55555555	Paul McCartney	PENDING	Update
74958673	Tommy Riffe	HAS YET TO APPLY	Update

Registration Visual Overview:

- Once a user gets the status of student they can get access the new student dashboard



- Once in this new dashboard a student can:
    - Fill out Form 1



Form 1: Program of Study for MS in Computer Science

Please enter the courses you plan to take to earn your MS degree in Computer Science. You must enter at most 12 courses, and your Form 1 must meet the degree requirements.

Univ ID:

Last Name:

First Name:

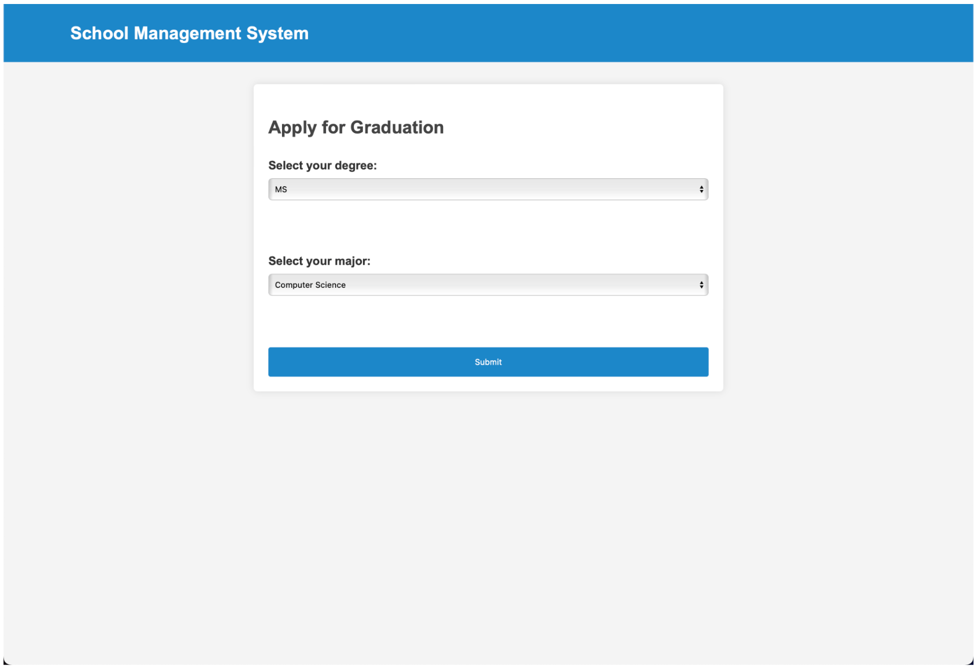
Degree: MS

Program: Computer Science

Courses In Program:

Course 1:

    - Apply to graduate



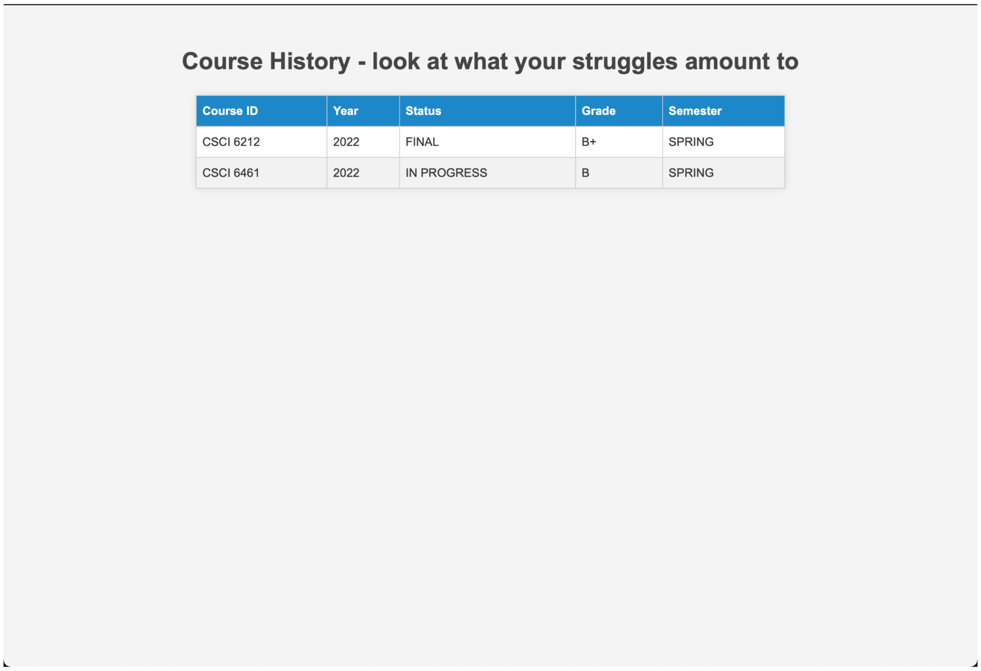
School Management System

Apply for Graduation

Select your degree: MS

Select your major: Computer Science

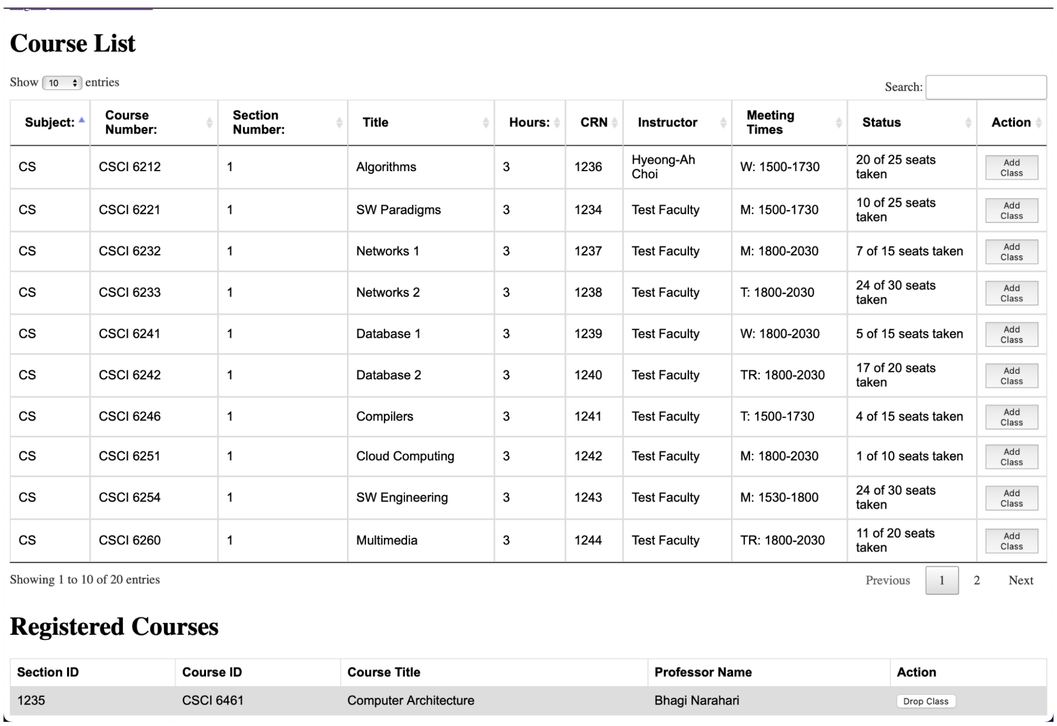
- View transcript



Course History - look at what your struggles amount to

Course ID	Year	Status	Grade	Semester
CSCI 6212	2022	FINAL	B+	SPRING
CSCI 6461	2022	IN PROGRESS	B	SPRING

- register/drop courses



Course List

Show 10 entries

Search:

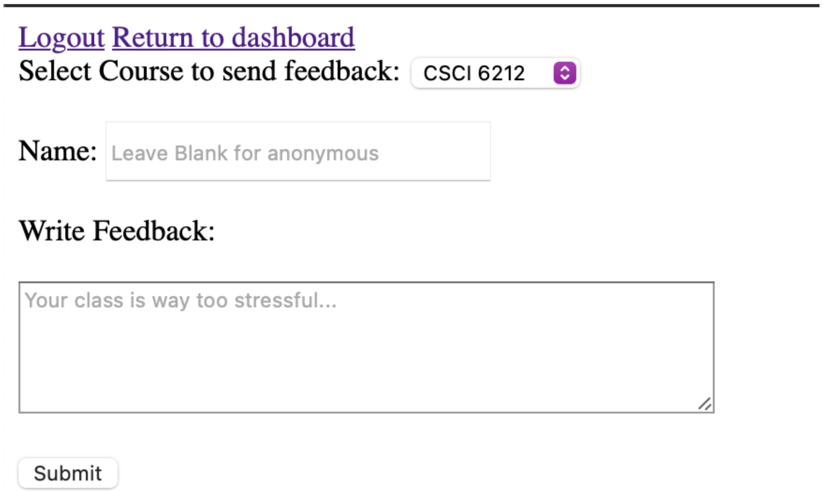
Subject:	Course Number:	Section Number:	Title	Hours:	CRN	Instructor	Meeting Times	Status	Action
CS	CSCI 6212	1	Algorithms	3	1236	Hyeong-Ah Choi	W: 1500-1730	20 of 25 seats taken	Add Class
CS	CSCI 6221	1	SW Paradigms	3	1234	Test Faculty	M: 1500-1730	10 of 25 seats taken	Add Class
CS	CSCI 6232	1	Networks 1	3	1237	Test Faculty	M: 1800-2030	7 of 15 seats taken	Add Class
CS	CSCI 6233	1	Networks 2	3	1238	Test Faculty	T: 1800-2030	24 of 30 seats taken	Add Class
CS	CSCI 6241	1	Database 1	3	1239	Test Faculty	W: 1800-2030	5 of 15 seats taken	Add Class
CS	CSCI 6242	1	Database 2	3	1240	Test Faculty	TR: 1800-2030	17 of 20 seats taken	Add Class
CS	CSCI 6246	1	Compilers	3	1241	Test Faculty	T: 1500-1730	4 of 15 seats taken	Add Class
CS	CSCI 6251	1	Cloud Computing	3	1242	Test Faculty	M: 1800-2030	1 of 10 seats taken	Add Class
CS	CSCI 6254	1	SW Engineering	3	1243	Test Faculty	M: 1530-1800	24 of 30 seats taken	Add Class
CS	CSCI 6260	1	Multimedia	3	1244	Test Faculty	TR: 1800-2030	11 of 20 seats taken	Add Class

Showing 1 to 10 of 20 entries

Registered Courses

Section ID	Course ID	Course Title	Professor Name	Action
1235	CSCI 6461	Computer Architecture	Bhagi Narahari	Drop Class

- Submit course feedback



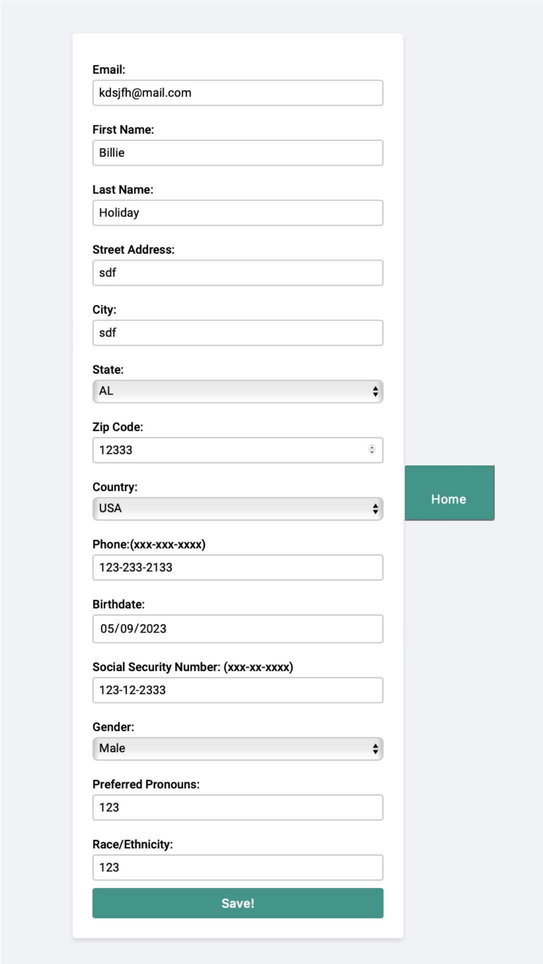
Select Course to send feedback: CSCI 6212

Name: Leave Blank for anonymous

Write Feedback:

Your class is way too stressful...

- Edit personal information


Email: kdsjfh@mail.com

First Name: Billie

Last Name: Holiday

Street Address: sdf

City: sdf

State: AL

Zip Code: 12333

Country: USA

Phone:(xxx-xxx-xxxx) 123-233-2133

Birthdate: 05/09/2023

Social Security Number: (xxx-xx-xxxx) 123-12-2333

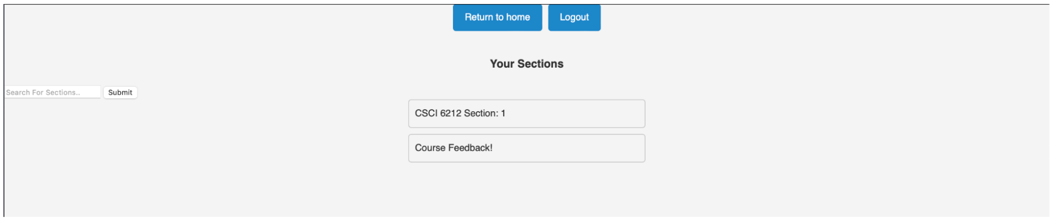
Gender: Male

Preferred Pronouns: 123

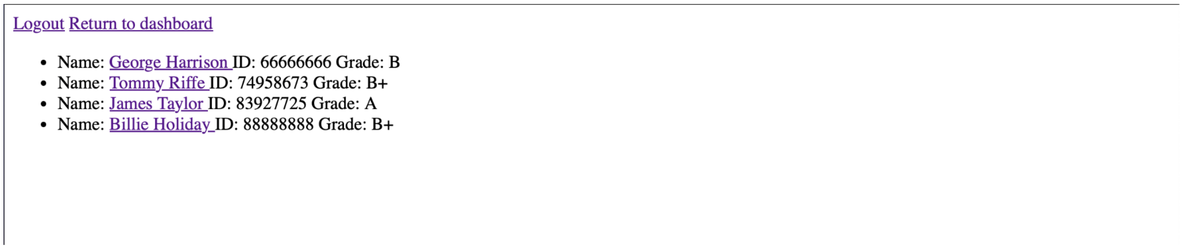
Race/Ethnicity: 123

The view for a faculty in regards to the registration portion is a bit different.

- An instructor can view their sections and the feed back for that section

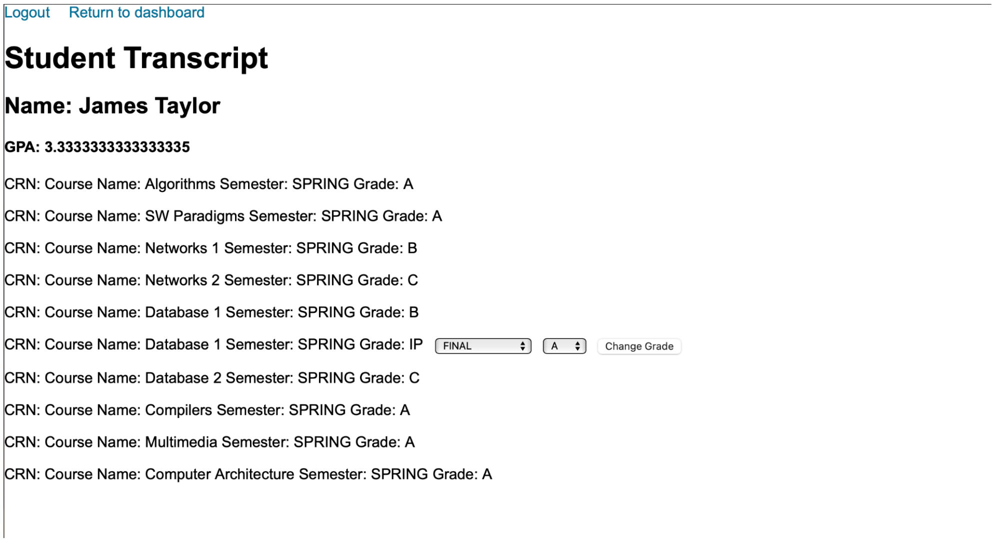

Your Sections

- Once they click on their section they can view all students in that section and their current grade


Logout Return to dashboard

- Name: George Harrison ID: 66666666 Grade: B
- Name: Tommy Riffe ID: 74958673 Grade: B+
- Name: James Taylor ID: 83927725 Grade: A
- Name: Billie Holiday ID: 88888888 Grade: B+

- When they click on a student they can update the students grades that are not final.



Student Transcript

Name: James Taylor

GPA: 3.3333333333333335

CRN: Course Name: Algorithms Semester: SPRING Grade: A

CRN: Course Name: SW Paradigms Semester: SPRING Grade: A

CRN: Course Name: Networks 1 Semester: SPRING Grade: B

CRN: Course Name: Networks 2 Semester: SPRING Grade: C

CRN: Course Name: Database 1 Semester: SPRING Grade: B

CRN: Course Name: Database 1 Semester: SPRING Grade: IP

CRN: Course Name: Database 2 Semester: SPRING Grade: C

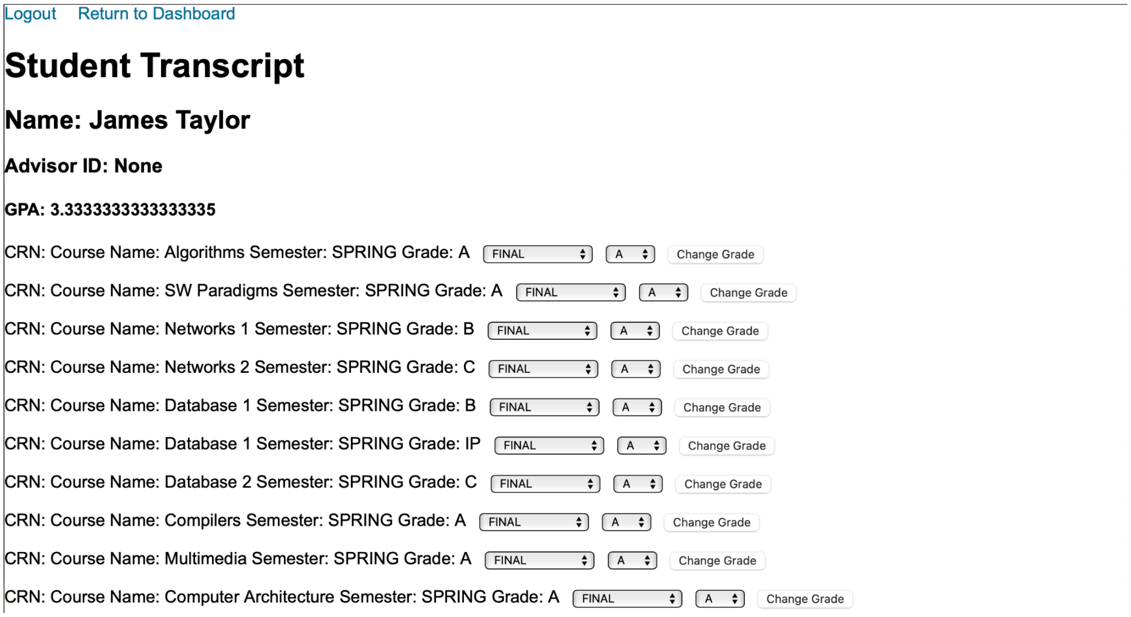
CRN: Course Name: Compilers Semester: SPRING Grade: A

CRN: Course Name: Multimedia Semester: SPRING Grade: A

CRN: Course Name: Computer Architecture Semester: SPRING Grade: A

A GS and admin have similar views, but they can change a student's grades for any class even after it has been marked as final.

- gs/admin grade change view



Student Transcript

Name: James Taylor

Advisor ID: None

GPA: 3.3333333333333335

CRN: Course Name: Algorithms Semester: SPRING Grade: A

CRN: Course Name: SW Paradigms Semester: SPRING Grade: A

CRN: Course Name: Networks 1 Semester: SPRING Grade: B

CRN: Course Name: Networks 2 Semester: SPRING Grade: C

CRN: Course Name: Database 1 Semester: SPRING Grade: B

CRN: Course Name: Database 1 Semester: SPRING Grade: IP

CRN: Course Name: Database 2 Semester: SPRING Grade: C

CRN: Course Name: Compilers Semester: SPRING Grade: A

CRN: Course Name: Multimedia Semester: SPRING Grade: A

CRN: Course Name: Computer Architecture Semester: SPRING Grade: A

Special Features:

- For the registration page the table of classes can be sorted by each column(course title, section, crn, etc.), and a search bar allows you to search any attribute of a class and the results are updated live in the table without having to reload the page.
- A course feedback feature allows all students to submit feedback to a specific course, then the respective instructor can sign in and view all the feedback for their class. Feedback can be submitted anonymously or not.
- Application forms can be saved mid-way and edited at a later time with all the information saved. After an application is submitted a user can only access the application in view mode.

Work Breakdown:

- In the beginning each member fixed any unresolved issues from phase 1.
- The schema was done by all three members with each member giving their input and required tables.
- Each member worked on their own part from phase 1, and when integration between parts were necessary the respective members would collaborate and split the work between them.
- Applications html pages were done by Tommy, advising pages done by Brandon, and Registration pages done by Seeam. Dashboards were a collaborative effort since it required links from different parts.
- Styling was done by Brandon.
- The extra features were done by Seeam.
- And helper functions (in sql_connector.py) were done by Tommy and Brandon