Faculty of Engineering and Applied Science

SOFE 3700U: Data Management Systems

Final Report

Date: November 29, 2022

Group: 21

Group Members:
Alexander Mahabir - 100787806
Awais Ahmad - 100785259
Akshat Kapoor-100781511
Mithusan Arulampalam - 100787123
Mohammad Mohammadkhani - 100798165

Abstract

This report aims to define how to modify the database system of *Hustler's university* so that more new courses and professors' information can be added and students can register for their courses more quickly. This report shows what steps and modifications we did to our database system to create a user-friendly experience, and it can store more information. We plan to accomplish this by creating deadlines to ensure our progress is on track, and dividing our progress into sections so that all members can work on different sections equally. We also had to consider how many classes would be running, then create a database that allows students to register for the classes of their choice. We had to find out the timings of classes and consider if multiple classes are running at the same time. Once all this information was acquired, we counted the university students and created unique IDs and passwords, to allow them to log into the website and register for classes. In addition, this report contains the schema diagrams and viewpoints we made to design our database.

Introduction

The “Hustler’s *university”* is one of the most credible universities in the world. To maintain its position, our university decided to attract more students by adding new courses and new members of faculty to teach new courses that will be available soon. In order to do that, our team’s goal is to design a database system with multiple views that different users of the database system, like students and professors, would find helpful and convenient. Using rest API, our team was able to get information from the database that was requested by students. Afterward, the infomation will then be displayed back to the student. Furthermore, our team’s goal is to create webpages with interactive and valuable features that will execute different queries and show the results. In addition, using these features, we demonstrate our views. In conclusion, to make our database, our team was tasked to design an Entity-relationship diagram.

Finally, our team provided a report explaining how to install and run the website, how the tables in our database interact, and how we can grab data from those tables in the database using the available programming language.

Goals of the Project

Our goal is to design an interactive database that is easy to use and can store additional information that we are required to use by the dean, such as new courses and faculties and new professors, and permit a higher number of students to register. This database will include students' information, program and course information, and professor information to be viewed by either the public or an existing user.

Relation to Other Work and the Course

Currently, there exist many universities in the world that use third-party websites such as MyOntarioTech, LORIS, Visual Schedule Builder, and ACORN, among others. These sites allow students

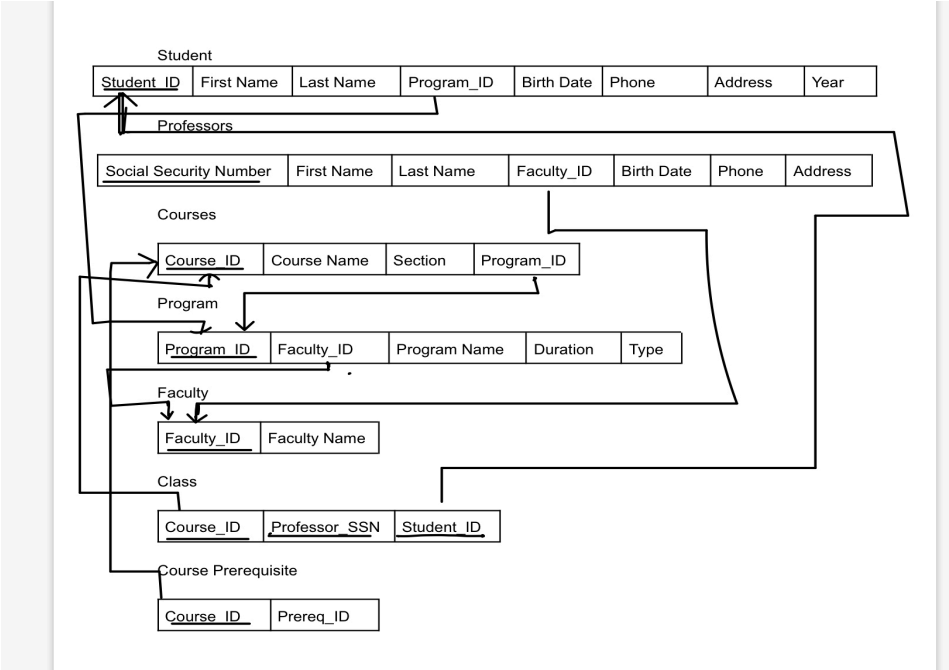
to view, plan, and alter their course schedules for the following semester or school year. It displays a list of available courses and allows the students to add or remove themselves from the course. They also display relevant information about the course, such as the professor teaching it, the faculty that runs it, the degree it is part of, and the different sections available for the course. To add themselves to a course, a student must first be a university student. Once they are a student, A unique ID will be assigned to identify them among other students. Next, students must log in with their unique student ID to register for a course. After registration, a page where students can select the courses that applied to their program. Secondly, our program will add them to those courses in our database. Our process is similar to other third-party services that many universities use. A student logs in with their student ID, registers for a course, and the website adds their information to the database.

In addition, this course management system helps us learn how to use queries and access databases through general programming languages such as PHP and RestAPI. This course teaches us the same thing, and doing this project gives us practical experience and teaches us how to use databases with existing websites.

Hustler University Course Management System:

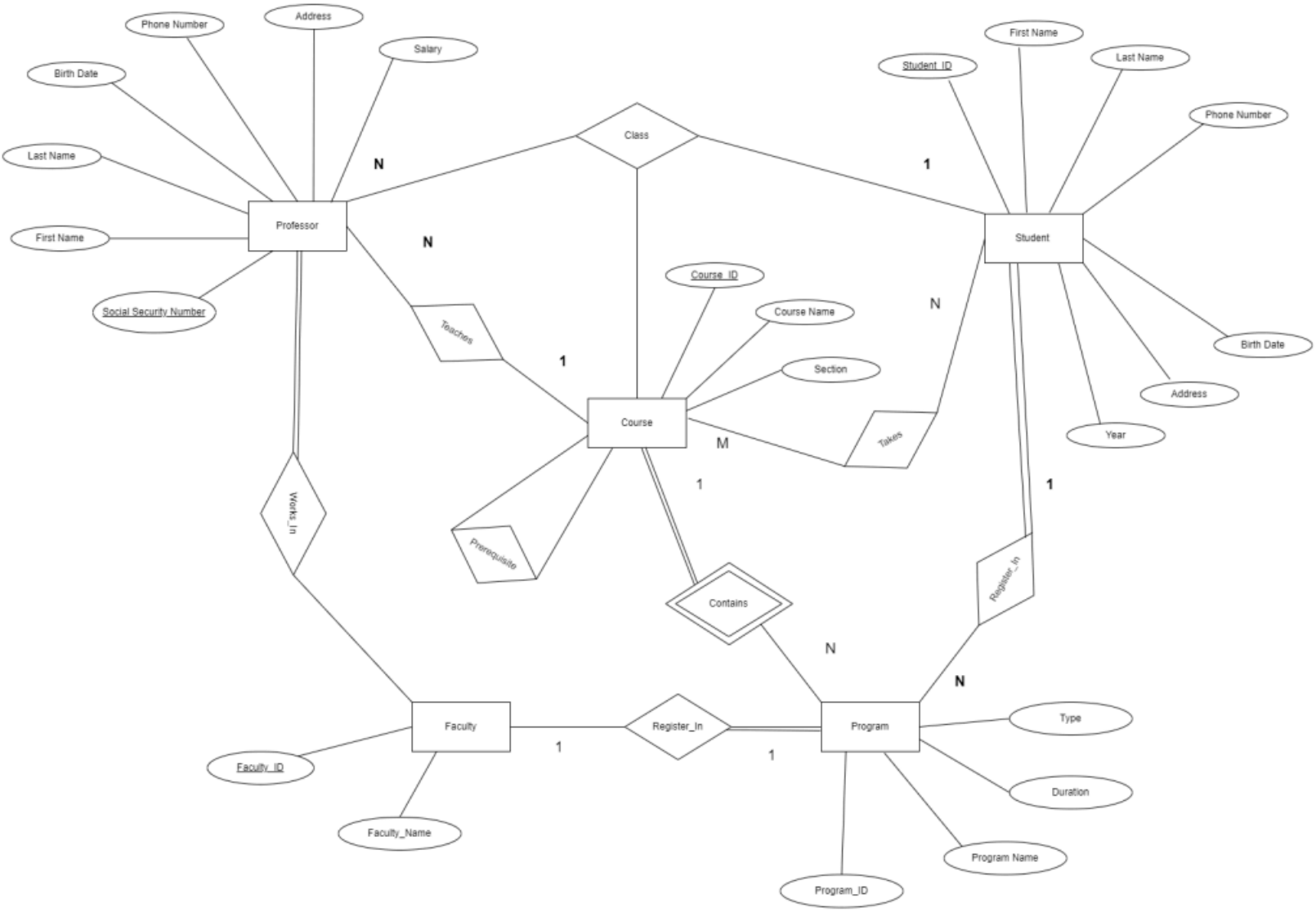
Our new course management system will allow students to view old or new courses and register for them. In addition to that, we made our systems easy to use and navigate. The students can log in and sign up with just one click; they do not have to go to different websites to log in/ sign in or see their courses. Furthermore, after clicking, the database will automatically store the information in the tables we made. Using rest API, we were able to retrieve information from the database requested by user and then outputed back in JSON format, that was available for users to view. Once students log in to the webpage, registered classes will be displayed for a student to view. This feature will help students keep track of their courses. In addition to that, students can view their professor's name.

Diagrams



This is the schema diagram of our relational database, it represents how each Schema/table is related to each other and how they interact with each other.

ER diagram



This is the ER relational diagram of our database,which shows the attributes of the each table and how they interact with each other and the relationship between them.

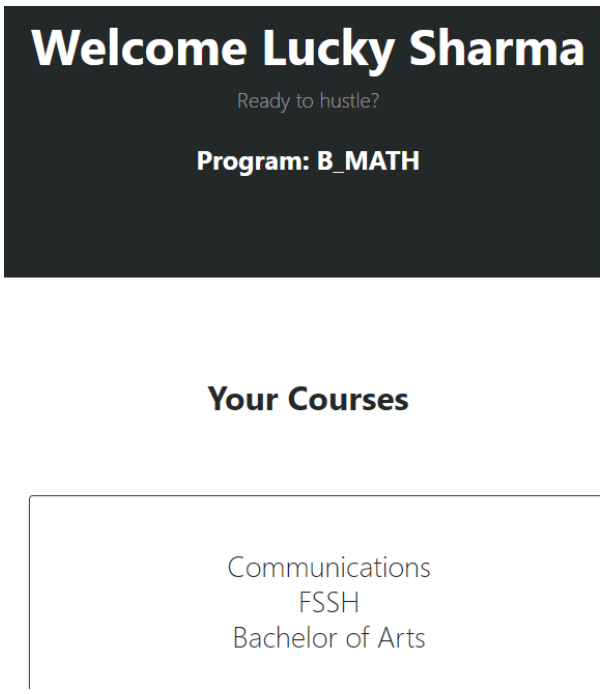
View points:

View 1: Join 3 tables

```
1  CREATE VIEW view1
2  AS SELECT p.FirstName, p.LastName, c.Course_ID, cs.Section
3  FROM class as c, courses as cs, professors_information as p
4  WHERE c.PSsn=p.Ssn AND c.Course_ID=cs.ID
5  GROUP BY p.FirstName;
6
7  SELECT * FROM view1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FirstName	LastName	Course_ID	Section
John	Loki	COMM1050	04
Lorence	Loki	MATH1010	01
Raj	Sharma	BUSI2040	02

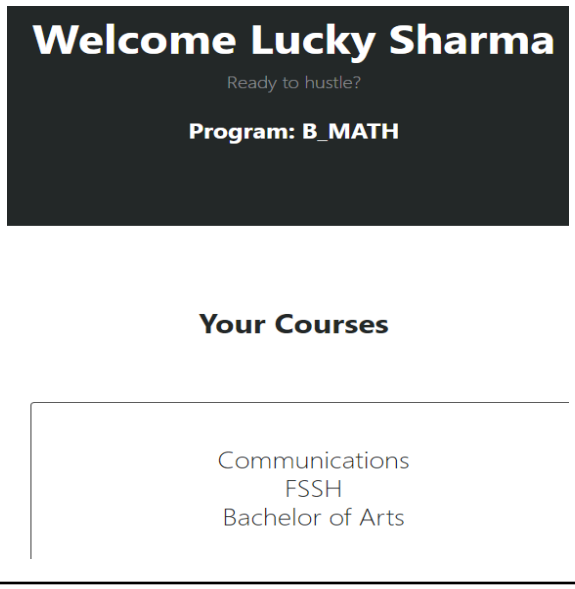


View 2: Nested query

```
1  CREATE VIEW view2
2  AS SELECT s.FirstName, s.LastName, c.Course_ID
3  FROM class as c, student_information as s
4  WHERE c.StudentID = ANY (SELECT Student_ID
5                           FROM student_information
6                           WHERE s.Student_ID = Student_ID)
7  GROUP BY s.FirstName;
8
9  SELECT * FROM view2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FirstName	LastName	Course_ID
Lucky	Sharma	COMM1050
Raj	Kapoor	MATH1010
Jacky	Lous	BUSI2040



View 3: Correlated nested query

```
1  CREATE VIEW view3
2  AS SELECT p.FirstName, p.LastName, p.Salary
3  FROM professors_information as p
4  WHERE p.Salary >        (SELECT AVG(Salary)
5                          FROM professors_information);
6
7  SELECT * FROM view3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FirstName	LastName	Salary
Arnav	Garg	170000
Raj	Sharma	180000

Welcome Lucky Sharma

Ready to hustle?

Program: B_MATH

Your Courses

Communications
FSSH
Bachelor of Arts

View 4: Full join statement

```
1  CREATE VIEW view4
2  AS SELECT s.FirstName, s.Program_id
3  FROM student_information as s
4  JOIN courses
5  ON s.Program_id=courses.Program_ID;
6
7  SELECT * FROM view4;
```

Result Grid | Filter Rows: | Export: | Wra

FirstName	Program_id
Lucky	B_MATH
Lucky	B_MATH
Raj	B_COMM
Jacky	B_COMM

Welcome Lucky Sharma

Ready to hustle?

Program: B_MATH

View 5: Uses union statement

```
1   CREATE VIEW view5
2   AS SELECT s.FirstName
3   FROM student_information as s
4   UNION
5   SELECT p.FirstName
6   FROM professors_information as p;
7
8 • SELECT * FROM view5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

FirstName
Lucky
Raj
Mohan
Jacky
John
Lorence
Arnav

Hustlers Professors	First Name	Last Name
	John	Loki
	Lorence	Loki
	Arnav	Garg
	Raj	Sharma

Hustlers Students	First Name	Last Name
	Lucky	Sharma
	Raj	Kapoor
	Mohan	Sharm
	Jacky	Lous
	aw	ah
	aw	ah

View 6: All first year students and their program

```
1 • CREATE VIEW view6
2   AS SELECT s.FirstName, s.LastName, p.Program_Name
3   FROM student_information as s, program_information as p
4   WHERE s.Program_ID=p.Program_ID AND s.Year=1;
5
6 • SELECT * FROM view6;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FirstName	LastName	Program_Name
Lucky	Sharma	Applied and Industrial Mathematics
Mohan	Sharm	Software Engineering
Jacky	Lous	Communication and Digital Media Studies

First Name	Last Name	Birth date	Phone Number	Year	Program
Lucky	Sharma	2001-10-09	3658851044	1	Applied and Industrial Mathematics
Raj	Kapoor	2003-09-10	4164451022	3	Communication and Digital Media Studies
Mohan	Sharm	2001-08-12	4164451023	1	Software Engineering
Jacky	Lous	2000-07-04	4163351023	1	Communication and Digital Media Studies

View 7: All courses and which degree they're apart of

```
1 •  CREATE VIEW view7
2    AS SELECT c.ID, p.type
3    FROM courses as c, program_information as p
4    WHERE p.Program_ID=c.Program_ID;
5
6 •  SELECT * FROM view7;
```

esult Grid | Filter Rows: | Export: | Wr

ID	type
COMM1050	Bachelor of Arts
ELEE2010	Bachelors of Engineering
MATH1010	Bachelors of Science
BUSI2040	Bachelors of Commerce
MATH1020	Bachelors of Science

Your Courses

Information systems
FBIT
Bachelors of
Commerce

View 8: Each course information and what faculty they're part of

```
1 •  CREATE VIEW view8
2    AS SELECT p.Faculty_ID, c.ID, c.CourseName
3    FROM courses as c, program_information as p
4    WHERE c.Program_ID=p.Program_ID;
5
6 •  SELECT * FROM view8;
```

Result Grid | Filter Rows: | Export: | Wra

Faculty_ID	ID	CourseName
FSSH	COMM1050	Communications
FEAS	ELEE2010	Digital Systems
FSc	MATH1010	Calculus
FBIT	BUSI2040	Information systems
FSc	MATH1020	Calculus2

Your Courses

Information systems
FBIT
Bachelors of
Commerce

View 9: Full faculty, program, and course information

```
1  CREATE VIEW view9
2  AS SELECT f.Name, p.Program_ID, p.type, c.ID, c.CourseName
3  FROM courses as c, program_information as p, faculty as f
4  WHERE c.Program_ID=p.Program_ID AND f.FacultyID=p.Faculty_ID;
5
6  SELECT * FROM view9;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Name	Program_ID	type	ID	CourseName
Faculty of Social Science and Humanities	B_COMM	Bachelor of Arts	COMM1050	Communications
Faculty of Engineering and Applied Science	B_ELEE	Bachelors of Engineering	ELEE2010	Digital Systems
Faculty of Science	B_MATH	Bachelors of Science	MATH1010	Calculus
Faculty of Business and Information Technology	B_BUSI	Bachelors of Commerce	BUSI2040	Information systems
Faculty of Science	B_MATH	Bachelors of Science	MATH1020	Calculus2

Course Name	Course ID	Program Name	Program ID	Faculty Name
Communications	COMM1050	Bachelor of Arts	B_COMM	Faculty of Social Science and Humanities
Digital Systems	ELEE2010	Bachelors of Engineering	B_ELEE	Faculty of Engineering and Applied Science
Calculus	MATH1010	Bachelors of Science	B_MATH	Faculty of Science
Information systems	BUSI2040	Bachelors of Commerce	B_BUSI	Faculty of Business and Information Technology
Calculus2	MATH1020	Bachelors of Science	B_MATH	Faculty of Science

View 10: Each professor and what faculty they're part of

```
1  CREATE VIEW view10
2  AS SELECT p.FirstName, p.LastName, f.Name
3  FROM professors_information as p, faculty as f
4  WHERE p.Faculty_ID=f.FacultyID;
5
6  SELECT * FROM view10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Con

FirstName	LastName	Name
John	Loki	Faculty of Business and Information Technology
Lorence	Loki	Faculty of Social Science and Humanities
Arnav	Garg	Faculty of Engineering and Applied Science
Raj	Sharma	Faculty of Science

Hustlers Professors	First Name	Last Name	Birth date	Phone Number	Faculty
	John	Loki	1972-04-01	346627778	Faculty of Business and Information Technology
	Lorence	Loki	1971-05-02	346887109	Faculty of Social Science and Humanities
	Arnav	Garg	1970-05-21	556991107	Faculty of Engineering and Applied Science
	Raj	Sharma	1969-05-12	413445789	Faculty of Science

Future Work

We plan to expand and add more features at Hustler University to improve our course management system. We expect an increase in students with the popularity and reputation that we are receiving. As a result, we have plans to implement more courses offered at our university.

Implementing new courses will require creating web pages for each class to display their needed information. In addition, due to the increase in students, Hustler University plans to expand its faculty. Therefore, we will require updating our database to include new faculty members and the new courses offered in the future. More importantly, features to help the students are also in our plans. For example, a grade calculator will be implemented in every class webpage to inform our students about their academic standing.

Hustlers University would also like to create the ability for online lectures. During inconvenient times, we want to ensure that the entire course material will be available to our students. Our goal is to create an online classroom environment so students can continue learning during inconvenient times. For example, the course would be taught online during severe weather conditions when it is very difficult for students and faculty to commute to the school.

Conclusion

In conclusion, the "Hustler's *university"* is one of the most credible universities in the world. To maintain its position, our university decided to attract more students by adding new courses and new members of faculty to teach new courses that will be available soon. Furthermore, our team's goal is to create webpages with interactive and valuable features that will execute different queries and show the results. Finally, we demonstrate our views using these features. Our team's goal was to design an Entity-relationship diagram. Afterward, the data is grabbed from tables in the database to display the data that students requested. Students are able to see the courses and register them with one click, and students

can also log in/signup. The database will store every piece of information. Lastly, our database created a more user-friendly and efficient experience.

References

[1] *HTML & CSS*. GeeksforGeeks. Retrieved November 29, 2022, from https://www.geeksforgeeks.org/web-technology/html-css/

[2] *SQL database*. SQL DATABASE. Retrieved November 29, 2022, from https://www.w3schools.com/sql/sql_ref_database.asp

[3]Hristozov, K. (2019, March 8). *Build a simple REST API in PHP*. Okta Developer. Retrieved November 29, 2022, from https://developer.okta.com/blog/2019/03/08/simple-rest-api-php