Tracks: a PhD progress tracking tool

Prepared for: Department of Computer Science, The University of Texas at El Paso

Prepared by: Team Nanikore (Abhijeet Patil, Alex Zamora, Matt Montoya, and Jorge Felix)

Wednesday, May 6, 2020

Version Number: 2.1.0

# Table of Contents

Table of Contents 2

Table of Figures 6

1. Database Scope 7

Purpose and Intended Audience 7

Domain 7

Key Entities and Relations 7

Document Overview 8

2. Requirements and Assumptions 10

Requirements 10

Assumptions 11

3. Entity-Relationship Model 12

4. Relational Schema 13

5. Normalized Database Schema 14

List of All Functional Dependencies for Each Relation 14

Normalized Schema in 1NF, 2NF, and 3NF 15

Process of Normalization 16

First Normal Form (1NF) 16

Second Normal Form (2NF) 16

Third Normal Form (3NF) 17

List of Final Relations in 3NF with functional dependencies 19

6. Database Schema in MySQL 20

Creating the Table for Student Information 20

Creating the Table for Faculty Information 20

Creating the Table for Admin Information 20

Creating the Table for Student Milestones 21

Creating the Table for Student Courses 21

Creating the Table for Student Conferences 21

Creating the Table for Student Papers 21

Creating the Table for Confirms\_updates Information 22

Creating the Table for Advising Information 22

7. Database Records 23

Student Table 23

Faculty Table 23

Admin Table 23

Student\_milestones Table 23

Student\_courses Table 24

Student\_conferences Table 24

Student\_papers Table 24

Confirms\_updates Table 24

Committee\_member Table 24

8. SQL Queries 25

SQL Queries Required To Satisfy Requirement 30 25

SQL Queries Required To Satisfy Requirement 31 25

SQL Queries Required To Satisfy Requirement 32 26

SQL Queries Required To Satisfy Requirement 33 26

SQL Queries Required To Satisfy Requirement 34 26

SQL Queries Required To Satisfy Requirement 35 26

SQL Queries Required To Satisfy Requirement 36 26

SQL Queries Required To Satisfy Requirement 37 27

SQL Queries Required To Satisfy Requirement 38 27

SQL Queries Required To Satisfy Requirement 39 27

9. Views 28

View Used To Satisfy Requirement 38 28

View Used To Satisfy Requirement 39 28

10. Procedures and Triggers 29

Procedure Used To Satisfy Requirement 36 29

Procedure Used To Satisfy Requirement 37 30

Trigger Used To Satisfy Requirement 30 30

Trigger Used To Satisfy Requirement 31 31

11. Reports 32

SQL Queries Required To Satisfy Requirement 32 32

SQL Queries Required To Satisfy Requirement 33 33

SQL Queries Required To Satisfy Requirement 34 34

SQL Queries Required To Satisfy Requirement 36 35

SQL Queries Required To Satisfy Requirement 37 35

SQL Queries Required To Satisfy Requirement 38 35

SQL Queries Required To Satisfy Requirement 39 35

12. Graphical User Interface 37

Screenshot of login Page 37

Screenshot of Student-welcome Page 38

Screenshot of Student-edit Page 39

Screenshot of student add milestone page 40

Screenshot of faculty welcome Page 40

Screenshot of faculty-search Page 40

Screenshot of Admin-welcome Page 41

15. References in APA Style 43

16. Appendix A. Contribution Information 44

Activities Relating to Database Scope 44

Activities Relating to Requirements 44

Activities Relating to Assumptions 44

Activities Relating to Entity-Relationship Model 44

Activities Relating to Relational Model 44

Activities Relating to MySQL Server 44

Activities Relating to General Document 45

Team Member Signatures 45

# Table of Figures

# 1. Database Scope

/\*

1.3

GROUP RESPONSABILITY

This section describes the purpose of the database system, including who will use it and domain e.g. the “mini-world”, the key entities and how they relate to each other, and information that will be out of the scope.

\*/

## Purpose and Intended Audience

The purpose of the Ph.D. Progress Tracking Tool is to track Ph.D. students' progress at The University of

Texas at El Paso (UTEP) during and after their Ph.D. degree conferral. This tool is proposed by our client,

Dr. Daniel Mejia, from the UTEP Computer Science Department. The intended audience of this tool will be

a set of users, consisting of current UTEP CS faculty and Ph.D. Students, including administrators, advisors,

current students, and alumni. There are two end goals for this project. One goal is to ensure non-graduated

students are progressing towards their degrees in a timely manner. The second goal is to track the graduated

student's professional career post-graduation. The website will follow the basic guidelines set by UTEP.

## Domain

The mini-world for the database is as follows: Computer Science Front Desk, Computer Science Students, Computer Science Advisors, Computer Science Admin.

## Key Entities and Relations

* **Student**: a student may be either a current or former PhD student of UTEP's Department of Computer Science. Each student will have a profile in the system. Their profile will store their login information (username and password) as well as several attributes that work to track their progress in the PhD program. Each student will only be able to view their own information.
* **Advisor**: an advisor any individual that advises students. Each advisor will have a profile in the system. Their profile will store their login information (username and password). Each advisor will have viewing access limited to the students they advise.
* **Admin**: an admin will be any individual given admin privileges by the UTEP Computer Science Department. Each advisor will have a profile in the system. Their profile will store their login information (username and password). Each admin will have viewing access to all advisor and student profiles in the system.

## Document Overview

**Section 1: Database Scope**

Section 1 introduces the Database Scope, including the purpose of the database system, the domain (or mini world), key entities and how each entity relates to other entities within the system domain, as well as any assumptions that don't violate the requirements.

**Section 2: Requirements & Assumptions**

Section 2 details the requirements that the Ph.D. Progress Tracking Tool database system shall satisfy. This includes functional and non-functional requirements that are correct, consistent, complete, feasible, verifiable, and traceable. This section further lists assumptions that do not violate the requirements listed herein. These assumptions are made by Team 7 (PM) to avoid ambiguity in the database design.

**Section 3: Entity-Relationship Models**

Section 3 details the ER model that shall define the implementation of the database system. The ER model visually details the requirements that the system shall satisfy. The Relational Model, derived from the ER diagram, further depicts the requirements that the tool shall satisfy.

**Section 4: Relational Schema**

Section 4 details the Relational Schema that shall define the implementation of the database system. The Relational Schema, derived from the ER diagram, further depicts the requirements that the tool shall satisfy.

**Section 5: Normalized Database Schema**

Section 5 introduces the schema for the Ph.D. Progress Tracking Tool database in normalized form. This includes the Normalized Schema in 3rd Normal Form (3NF), with all functional dependencies for each relation listed.

**Section 6: Database Schema in MySQL**

Section 6 details MySQL Server information, as it pertains to the Ph.D. Progress Tracking Tool database. This includes the Database Schema from Section 4 (Schema) in MySQL, group tables that allow us to create a new account and log into the website, as well as the MySQL queries required to satisfy the functional requirements.

**Section 7: Database Records**

Section 7 details 3 sample database records for each table that are representative of the domain the tool is modeling.

**Section 8: SQL Queries**

Section 8 outlines the MySQL queries that will be used in order to satisfy the functional requirements.

**Section 9: Views**

Section 9 describes two views in MySQL that will be used in the project. This section includes the CREATE VIEW statements corresponding to the views.

**Section 10: Procedures and Triggers**

Section 10 outlines two procedures and two triggers used in the project.

**Section 11: Reports**

Section 11 describes the reports used in the tool. Each report includes a trace back to functional requirements, the list of queries and functions needed to generate the report, and a screenshot of the data returned after running the report in MySQL.

**Section 12: Graphical User Interface**

Section 12 shows the GUI designed by Team 7 (PM). This includes a login page that connected to the database in UTEP's MySQL server, the GUI Menu, Records, and Reports required by the client. The GUI is available at http://cssrvlab01.utep.edu/Classes/cs4342/team7.

**Section 13: Security**

Section 13 details security standards used to ensure program security.

**Section 14: Usability Report**

Section 14 details several usability tests to ensure program usability.

**Section 15: References**

Section 7 lists the references used for this document in APA style.

**Section 16: Appendix A**

Section 8 includes an appendix indicating the tasks that each one of the team members carried out for It lists each activity and which individual fulfilled that activity.

# 2. Requirements and Assumptions

/\*

1.4

GROUP RESPONSABILITY

This section includes a list of functional requirements (requirements should be correct, consistent,

complete, feasible, verifiable and traceable) and a list of assumptions to avoid ambiguity in the database design. Assumptions should not violate your requirements.

\*/

## Requirements

1. The system shall follow the UTEP Style Identity Guide [[1](https://www.utep.edu/university-communications/_Files/docs/UTEP-GRAPHIC-IDENTITY-GUIDE.pdf)].
2. The system shall have an admin user type.
3. The system shall have a faculty user type.
4. The system shall have a student user type.
5. The system shall require a username and password from each user before they can access their account.
6. The system shall allow each user to change their login password.
7. The system shall have Single Sign-On capabilities.
8. The system shall allow an admin user to view every component of the system.
9. The system shall allow every student to have an assigned faculty user.
10. The system shall allow a faculty user to view every attribute of their students.
11. The system shall keep track of every user's username.
12. The system shall keep track of every user's password.
13. The system shall keep track of every user's first name.
14. The system shall keep track of every user's middle initial.
15. The system shall keep track of every user's last name.
16. The system shall keep track of students' UTEP ID.
17. The system shall keep track of students' graduation status.
18. The system shall keep track of students' degree.
19. The system shall keep track of students' overall GPA.
20. The system shall keep track of students' graduation year.
21. The system shall keep track of students' grant information.
22. The system shall keep track of students' assigned faculty user.
23. The system shall keep track of students' committee members.
24. The system shall keep track of students' achieved milestones.
25. The system shall keep track of students' achieved milestones' achievement dates.
26. The system shall keep track of students' current curriculum.
27. The system shall keep track of students' published papers.
28. The system shall keep track of students' published papers' publication dates.
29. The system shall keep track of students' attended conferences.
30. The system shall keep track of students' changes to their record of attended conferences. (Trigger 1)
31. The system shall keep track of students' changes to their record of published papers. (Trigger 2)
32. The system shall allow admin users to view a list of current students by ascending GPA attribute. (Report 1)
33. The system shall allow admin users to view a list of current students by descending GPA attribute. (Report 2)
34. The system shall allow admin users to view a list of degrees and each degree’s average student gpa. (Report 3)
35. The system shall allow admin users to view a list of current students with a graduation year after 2021. (Report 4)
36. The system shall allow admin users to view a list of students from Machine Learning degree who meet the GPA requirements to be eligible for teaching assistantships (3.75). (Report 5) (Procedure 1)
37. The system shall allow admin users to view a list of students who are not qualified for any kind of scholarship support from university. (Report 6) (Procedure 2)
38. The system shall allow admin users to view a list of students that have the NIH grant. (View 1)
39. The system shall allow admin users to view a total count of how many students have received the NSF grant. (View 2)

## Assumptions

1. All users have a fundamental understanding of computer literacy.
2. All users have a fundamental understanding of Microsoft Office 365.
3. All users have a fundamental understanding of the UTEP CS website.
4. Faculty have an understanding of Ph.D. degree plan sequences.
5. Students are registered for a Ph.D. Computer Science degree with the UTEP Computer Science Department.
6. Users have a valid UTEP ID number to be authenticated as a UTEP student.
7. Users have internet access to access the database.
8. All users will have a username that is 10 characters or less
9. All users will have first name that is 50 characters or less
10. All users will have last name that is 50 characters or less

Acronyms

* **GPA**: Grade Point Average
* **IRB**: Institutional Review Board
* **SSO**: Single Sign-On

Definitions

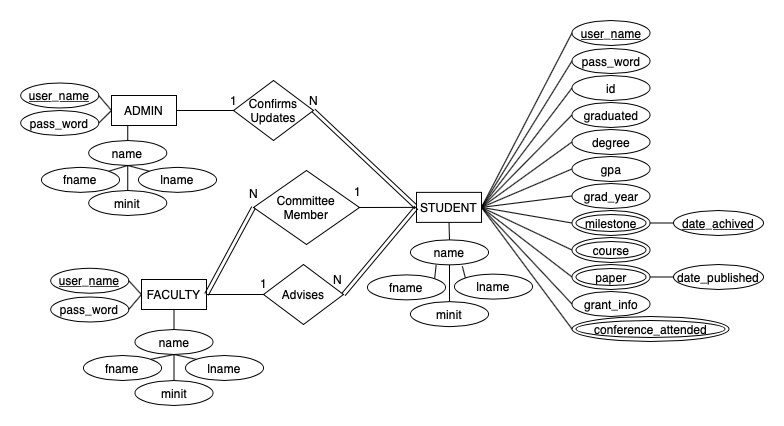
* **Milestones**: An event outlined by the Ph.D. Program Handbook 2.0 [[2](https://www.utep.edu/cs/graduate/PDFs/PhD-Handbook-2019-2020.pdf)].
* **Report**: A CSV file consisting of columns with values corresponding to attributes of selected students.
* **Significant**: Of great importance as defined by the advisor and student.
* **Single** **Sign-On**: A system's ability to allow a user to sign on once and grant access to multiple parts of the system.
* **List of students**: An HTML table with students' username, first name, middle initial, last name, ID, degree, and graduation date attributes as columns, and students as rows.

# 3. Entity-Relationship Model

/\*

1.5

DESIGN RESPONSIBILITY

\*/

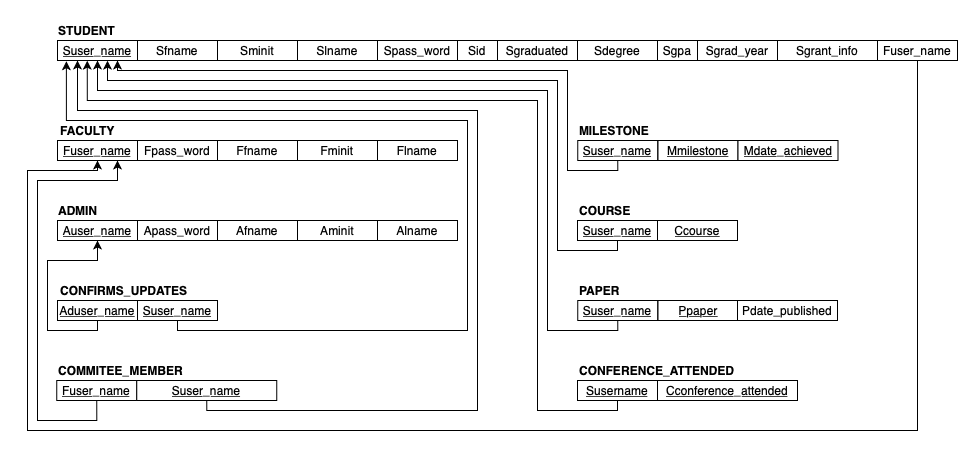
# 4. Relational Schema

/\*

1.6

DESING RESPONSABILITY

\*/



# 5. Normalized Database Schema

/\*

1.7

DESING RESPONSABILITY

\*/

Section 5 introduces the schema for the Ph.D. Progress Tracking Tool database. This includes the Database Records that are representative of the domain being modeled, and the Normalized Schema in 3rd Normal Form (3NF), with all functional dependencies for each relation listed. The normalization process carried out is detailed in each respective subsection of the Normalized Schema.

## List of All Functional Dependencies for Each Relation

## Normalized Schema in 1NF, 2NF, and 3NF

## Process of Normalization

## First Normal Form (1NF)

* + **STUDENT** is in 1NF because all attributes are atomic.
  + **FACULTY** is in 1NF because all attributes are atomic.
  + **ADMIN** is in 1NF because all attributes are atomic.
  + **MILESTONE** is in 1NF given that all attributes are atomic.
  + **COURSE** is in 1NF as all attributes are atomic.
  + **PAPER** is in 1NF as all attributes are atomic.
  + **CONFIRMS\_UPDATES** is in 1NF as all attributes are atomic.
  + **COMMITEE\_MEMBER** is in 1NF as all attributes are atomic.
  + **CONFERENCE\_ATTENDED** is in 1NF as all attributes are atomic.

## Second Normal Form (2NF)

The relation **STUDENT** is in 2NF because

* + It is in 1NF and
  + All its non-prime attributes fully-functionally dependent on the primary key *Suser\_name*.

The relation **FACULTY** is in 2NF because

* + It is in 1NF and
  + All its non-prime attributes fully-functionally dependent on the primary key *Fuser\_name*.

The relation **ADMIN** is in 2NF because

* + It is in 1NF and
  + All its non-prime attributes fully-functionally depend on the primary key Auser\_name.

The relation **MILESTONE** is in 2NF because

* + It is in 1NF and
  + It does not have non-prime attributes that can violate the norm.

The relation **COURSE** is in 2NF because

* + It is in 1NF and
  + It does not have non-prime attributes that can violate the norm.

The relation **PAPER** is in 2NF because

* + It is in 1NF and
  + All its non-prime attributes fully-functionally depend on the primary key S*user\_name*.

The relation **CONFIRMS\_UPDATES** is in 2NF because

* + It is in 1NF and
  + It does not have non-prime attributes that can violate the norm.

The relation **COMMITTEE\_MEMBER** is in 2NF because

* + It is in 1NF and
  + It does not have non-prime attributes that can violate the norm.

The relation **CONFERENCE\_ATTENDED** is in 2NF because

* + It is in 1NF and
  + It does not have non-prime attributes that can violate the norm.

## Third Normal Form (3NF)

The relation **STUDENT** is in 3NF because

* + It is in 2NF and
  + There is no dependency among any of the non-prime attributes.

The relation **FACULTY** is in 3NF because

* + It is in 2NF and
  + No non-prime attributes depend on another non-primary attribute.

The relation **ADMIN** is in 3NF because

* + It is in 2NF and
  + There is no dependency among any of the non-prime attributes.

The relation **MILESTONE** is in 3NF because

* + It is in 2NF and
  + It does not contain non-prime attributes that can violate the norm.

The relation **CONFERENCE\_ATTENDED** is in 3NF because

* + It is in 2NF and
  + It does not have any non-prime attributes that would violate the norm.

The relation **PAPER** is in 3NF because

* + It is in 2NF and
  + It does not have any non-prime attributes that would violate the norm.

The relation **COURSE** is in 3NF because

* + It is in 2NF and
  + It does not have any non-prime attributes that would violate the norm.

The relation **CONFIRMS\_UPDATES** is in 3NF because

* + It is in 2NF and
  + It does not contain non-prime attributes that can violate the norm.

The relation **COMMITTEE\_MEMBER** is in 3NF because

* + It is in 2NF and
  + It does not contain non-prime attributes that can violate the norm.

## List of Final Relations in 3NF with functional dependencies

# 6. Database Schema in MySQL

/\*

1.8

IMPLEMENTATION RESPONSABILITY

\*/

<<<List of CREATE TABLE statements required to create tables in database>>>

## Creating the Table for Student Information

CREATE TABLE Student (

Suser\_name VARCHAR(10) NOT NULL PRIMARY KEY,

Sfname VARCHAR(50) NOT NULL,

Sminit VARCHAR(1),

Slname VARCHAR(50) NOT NULL,

Spass\_word VARCHAR(50) NOT NULL,

Sid INT(8) NOT NULL,

Sgraduated BOOLEAN NOT NULL,

Sdegree VARCHAR(50) NOT NULL,

Sgpa FLOAT(3) NOT NULL,

Sgrad\_year DATE NOT NULL,

Sgrant\_info VARCHAR(50) NOT NULL,

Fuser\_name VARCHAR(10) NOT NULL,

FOREIGN KEY (Fuser\_name) REFERENCES Faculty(Fuser\_name)

);

## Creating the Table for Faculty Information

CREATE TABLE Faculty (

Fuser\_name VARCHAR(10) NOT NULL PRIMARY KEY,

Fpass\_word VARCHAR(50) NOT NULL,

Ffname VARCHAR(50) NOT NULL,

Fminit VARCHAR(1),

Flname VARCHAR(50) NOT NULL

);

## Creating the Table for Admin Information

CREATE TABLE Admin (

Auser\_name VARCHAR(10) NOT NULL PRIMARY KEY,

Apass\_word VARCHAR(50) NOT NULL,

Afname VARCHAR(50) NOT NULL,

Aminit VARCHAR(1),

Alname VARCHAR(50) NOT NULL

);

## Creating the Table for Student Milestones

CREATE TABLE Student\_milestones (

Suser\_name VARCHAR(10) NOT NULL,

Smilestones VARCHAR(50) NOT NULL,

Mdate\_achieved DATE NOT NULL,

PRIMARY KEY(Suser\_name, Smilestones, Mdate\_achieved),

FOREIGN KEY (Suser\_name) REFERENCES Student(Suser\_name)

);

## Creating the Table for Student Courses

CREATE TABLE Student\_courses (

Suser\_name VARCHAR(10) NOT NULL,

Scourses VARCHAR(50) NOT NULL,

PRIMARY KEY(Suser\_name, Scourses),

FOREIGN KEY (Suser\_name) REFERENCES Student(Suser\_name)

);

## Creating the Table for Student Conferences

CREATE TABLE Student\_conferences (

Suser\_name VARCHAR(10) NOT NULL,

Sconferences VARCHAR(50) NOT NULL,

PRIMARY KEY(Suser\_name, Sconferences),

FOREIGN KEY (Suser\_name) REFERENCES Student(Suser\_name)

);

## Creating the Table for Student Papers

CREATE TABLE Student\_papers (

Suser\_name VARCHAR(10) NOT NULL,

Spapers VARCHAR(100) NOT NULL,

Pdate\_published DATE NOT NULL,

PRIMARY KEY(Suser\_name, Spapers),

FOREIGN KEY (Suser\_name) REFERENCES Student(Suser\_name)

);

## Creating the Table for Confirms\_updates Information

CREATE TABLE Confirms\_updates (

Suser\_name VARCHAR(10) NOT NULL,

Auser\_name VARCHAR(10) NOT NULL,

PRIMARY KEY(Suser\_name, Auser\_name),

FOREIGN KEY (Suser\_name) REFERENCES Student(Suser\_name),

FOREIGN KEY (Auser\_name) REFERENCES Admin(Auser\_name)

);

## Creating the Table for Advising Information

CREATE TABLE Advises (

Suser\_name VARCHAR(10) NOT NULL,

Fuser\_name VARCHAR(10) NOT NULL,

PRIMARY KEY(Suser\_name, Fuser\_name),

FOREIGN KEY (Suser\_name) REFERENCES Student(Suser\_name),

FOREIGN KEY (Fuser\_name) REFERENCES Faculty(Fuser\_name)

);

# 7. Database Records

/\*

1.9

IMPLEMENTATION RESPONSABILITY

List of INSERT statements for a sample of database records representative of the domain (3 per table)

\*/

## Student Table

* INSERT INTO Student VALUES ('user', 'Michael', 'I', 'Jordan', 'user1', 80512345, 0, 'Database Managment', 0.45, '2020-12-15', 'Received NSF Grant', 'vvapnik');
* INSERT INTO Student VALUES ('etardos', 'Eva', '', 'Tardos', 'optimization', 80512346, 0, 'Database Managment', 1.45, '2021-12-15', 'Received IEEE Grant', 'jgosling');
* INSERT INTO Student VALUES ('pcousot', 'Patrick', '', 'Cousot', 'abstraction', 80512347, 0, 'Database Managment', 2.45, '2022-06-15', '', 'emccluskey');
* INSERT INTO Student VALUES ('aaho', 'Alfred', '', 'Aho', 'fgrep', 80512348, 0, 'Data Structures', 3.45, '2022-06-15', '', ‘emccluskey');
* INSERT INTO Student VALUES ('rfitch','Robin', 'A', 'Fitch', 'coolpass6', '80512346', 0, 'Cyber Security', '3.75', '2019-12-15', 'NSF', 'jgosling');
* INSERT INTO Student VALUES (‘tdoe', 'Tim', 'C', 'Doe','coolpass7', '80512347', 0, 'Machine Learning', '3.95', '2021-12-15', 'NIH', 'jgosling');
* INSERT INTO Student VALUES ('jdoe', 'Josh', 'D', 'Doe', 'coolpass7', '80512348', 0, 'Information Systems', '3.95', '2021-12-15', 'NIH', 'jgosling');
* INSERT INTO Student VALUES ('aliz', 'Ash', 'H', 'Liz','coolpass8', '80512349', 0, 'Cyber Security', '2.50', '2021-12-15', 'NONE', 'jgosling');
* INSERT INTO Student VALUES ('azou', 'Astor', 'M', 'Zou', 'coolpass9', '80512350', 0,'Information Systems', '3.50', '2021-12-15', 'DOE', 'jgosling');
* INSERT INTO Student VALUES ('jzhao', 'Jeremy', 'S', 'Zhao','coolpass10', '80512351', 0,'Machine Learning', '3.95', '2020-12-15', 'NSF', 'jgosling');

## Faculty Table

* INSERT INTO Faculty VALUES ('vvapnik', 'statistics', 'Vladimir', '', 'Vapnik');
* INSERT INTO Faculty VALUES ('jgosling', 'java', 'James', 'A', 'Gosling');
* INSERT INTO Faculty VALUES ('emccluskey', 'electrical', 'Edward', 'J', ‘McCluskey’);

## Admin Table

* INSERT INTO Admin VALUES ('sgraham', 'performance', 'Susan', 'L', 'Graham');
* INSERT INTO Admin VALUES ('llamport', 'latex', 'Leslie', '', 'Lamport');
* INSERT INTO Admin VALUES ('bliskov', 'oo', 'Barbara', 'H', 'Liskov');

## Student\_milestones Table

* INSERT INTO Student\_milestones VALUES ('etardos', 'Successful completion of qualifying process', '2021-01-05');
* INSERT INTO Student\_milestones VALUES ('pcousot', 'Coursework successfully completed', '2018-10-05');
* INSERT INTO Student\_milestones VALUES ('pcousot', 'Dissertation proposal completed and approved', '2019-05-02');

## Student\_courses Table

* INSERT INTO Student\_courses VALUES (‘etardos’, ‘CS 4311’);
* INSERT INTO Student\_courses VALUES (‘etardos’, ‘CS 5343’);
* INSERT INTO Student\_courses VALUES (‘pcousot’, ‘CS 5678’);

## Student\_conferences Table

INSERT INTO Student\_conferences VALUES ('etardos', 'Defcon 2018');

INSERT INTO Student\_conferences VALUES ('pcousot', 'ICML 2019');

INSERT INTO Student\_conferences VALUES ('pcousot', 'PyCon 2019');

## Student\_papers Table

INSERT INTO Student\_papers VALUES ('etardos', 'Maximizing the spread of influence through a social network’, ‘2003-01-01’);

INSERT INTO Student\_papers VALUES ('etardos', 'Influential nodes in a diffusion model for social networks’, ‘2005-01-01‘);

INSERT INTO Student\_papers VALUES ('pcousot', 'Systematic design of program analysis frameworks’, ‘1979-01-01‘);

## Confirms\_updates Table

INSERT INTO Confirms\_updates VALUES (‘user’, 'sgraham');

INSERT INTO Confirms\_updates VALUES (‘etardos’, 'llamport');

INSERT INTO Confirms\_updates VALUES (‘pcousot’, 'bliskov');

## Committee\_member Table

INSERT INTO Advises VALUES (‘user’, 'vvapnik');

INSERT INTO Advises VALUES (‘etardos’, 'jgosling');

INSERT INTO Advises VALUES (’pcousot’, ’emccluskey');

# 8. SQL Queries

/\*

1.10

IMPLEMENTASTION RESPONSABILITY

List of MySQL queries required to satisfy functional requirements along with the requirements each satisfies and how it satisfies the requirement

\*/

## SQL Queries Required To Satisfy Requirement 30

Requirement 30: The system shall keep track of students' changes to their record of attended conferences (Trigger 1).

CREATE TABLE Stud\_conferences\_update(Stud\_utep\_id INT AUTO\_INCREMENT PRIMARY KEY,

Sconferences char(50), changeDate DATETIME DEFAULT NULL, action

VARCHAR(50) DEFAULT NULL);

CREATE TRIGGER before\_stud\_conferences\_update

BEFORE UPDATE ON Student\_conferences

FOR EACH ROW

INSERT INTO Stud\_conferences\_update

SET action=’update’,

Stud\_username = OLD.Suser\_name,

Sconferences = OLD.Sconferences,

changeDate = NOW();

## SQL Queries Required To Satisfy Requirement 31

Requirement 31: The system shall keep track of students' changes to their record of published papers (Trigger 2).

CREATE TABLE Stud\_publications\_update(Stud\_utep\_id INT AUTO\_INCREMENT

PRIMARY KEY,

Spublications char(50), changeDate DATETIME DEFAULT NULL, action

VARCHAR(50) DEFAULT NULL);

CREATE TRIGGER before\_stud\_publication\_update

BEFORE UPDATE ON Student\_papers

FOR EACH ROW

INSERT INTO Stud\_publications\_update

SET action=’update’,

Stud\_utep\_id = OLD.Suser\_name,

Spublications = OLD.Spapers,

changeDate = NOW();

## SQL Queries Required To Satisfy Requirement 32

Requirement 32: The system shall allow admin users to view a list of current students by ascending GPA attribute (Report 1).

SELECT \* FROM Student ORDER BY Sgpa;

## SQL Queries Required To Satisfy Requirement 33

Requiremtent 33: The system shall allow admin users to view a list of current students by descending GPA attribute (Report 2).

SELECT \* FROM Student ORDER BY Sgpa DESC;

## SQL Queries Required To Satisfy Requirement 34

Requirement 34: The system shall allow admin users to view a list of degrees and each degree’s average student gpa. (Report 3).

SELECT Sdegree, AVG(Sgpa) FROM Student GROUP BY 1;

## SQL Queries Required To Satisfy Requirement 35

Requirement 35: The system shall allow admin users to view a list of current students with a graduation year after 2021 (Report 4).

select sfname, sdegree from student where sgrad\_year >= ‘2021’;

## SQL Queries Required To Satisfy Requirement 36

The system shall allow admin users to view a list of students from Machine Learning degree who meet the GPA requirements to be eligible for teaching assistantships (3.75). (Report 5) (Procedure 1).

DELIMITER //

CREATE PROCEDURE PhD\_ML\_Assistantships(IN req\_GPA float)

BEGIN

SELECT Sutep\_id, Sfname, Slname, Sgrad\_year, Sgpa

FROM Student WHERE Sgpa >= req\_GPA AND Sdegree = 'Machine learning';

END //

DELIMITER ;

CALL PhD\_ML\_Assistantships('3.75');

## SQL Queries Required To Satisfy Requirement 37

Requirement 37: The system shall allow admin users to view a list of students who are not qualified for any kind of scholarship support from university. (Report 6) (Procedure 2).

DELIMITER //

CREATE PROCEDURE PhD\_Unqualified(IN req\_GPA float)

BEGIN

SELECT Sid, Sfname, Slname, Sgrad\_year, Sgpa

FROM Student WHERE Sgpa <= req\_GPA;

END //

DELIMITER ;

CALL PhD\_ML\_Assistantships('2.5');

## SQL Queries Required To Satisfy Requirement 38

Requirement 38: The system shall allow admin users to view a list of students that have the NIH grant. (View 1)

CREATE VIEW student\_grantinfo\_NIH AS SELECT Suser\_name FROM STUDENT WHERE Sgrant\_info='NIH';

## SQL Queries Required To Satisfy Requirement 39

Requirement 39: The system shall allow admin users to view a total count of how many students have received the NSF grant (View 2).

CREATE VIEW Student\_totalNSF AS SELECT COUNT(Sgrant\_info) AS total\_students FROM STUDENT WHERE Sgrant\_info = 'NSF';

# 9. Views

/\*

1.11

IMPLEMENTASTION RESPONSABILITY

Create at least two views in MySQL that are required in your project, e.g., for generating   
reports or display information in the interface.

List the CREATE VIEW statements corresponding to the two views created in your   
document.   
\*/

## View Used To Satisfy Requirement 38

Requirement 38: The system shall allow admin users to view a list of students that have the NIH grant. (View 1)

CREATE VIEW Student\_grant\_info\_NIH AS SELECT Suser\_name, Sfname, Sminit, Slname, Sid, Sdegree, Sgrad\_year FROM STUDENT WHERE Sdegree=‘Data Structures';

## View Used To Satisfy Requirement 39

Requirement 39: The system shall allow admin users to view a total count of how many students have received the NIH grant (View 2).

CREATE VIEW Student\_total\_NSF AS SELECT COUNT(Sgrant\_info) AS total\_students FROM STUDENT WHERE Sgrant\_info = 'NSF';

# 10. Procedures and Triggers

/\*

1.12

REPORTS RESPONSABILITY

Create at least two procedures that are appropriate for your implementation. Include in   
your document the definition of the procedure in the database and the results after running them. If applicable, trace back to the functional requirements they are contributing to.

Create at least two triggers that are appropriate for your implementation. If applicable, trace back to the functional requirements they are contributing to. **Triggers will only be included in your document.**   
\*/

## Procedure Used To Satisfy Requirement 36

The system shall allow admin users to view a list of students from Machine Learning degree who meet the GPA requirements to be eligible for teaching assistantships (3.75). (Report 5) (Procedure 1).

DELIMITER //

CREATE PROCEDURE ML\_Assistantships(IN req\_GPA float)

BEGIN

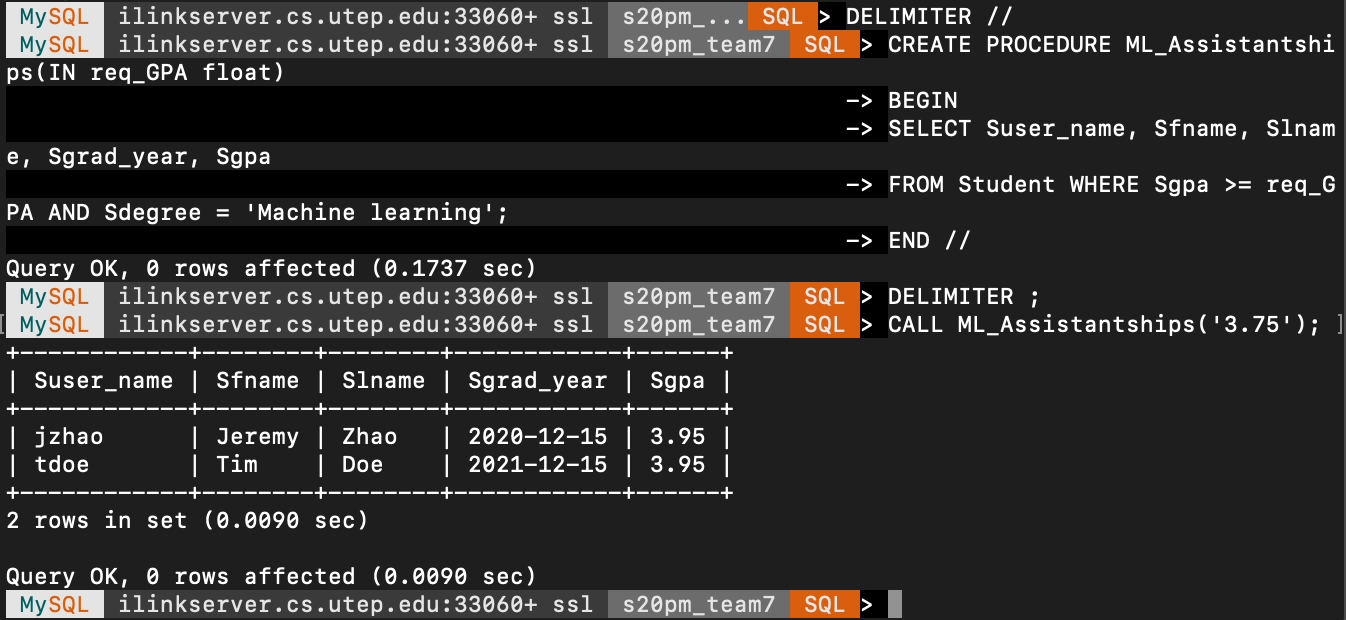
SELECT Suser\_name, Sfname, Slname, Sgrad\_year, Sgpa

FROM Student WHERE Sgpa >= req\_GPA AND Sdegree = 'Machine learning';

END //

DELIMITER ;

CALL ML\_Assistantships(‘3.75');



## Procedure Used To Satisfy Requirement 37

Requirement 37: The system shall allow admin users to view a list of students who are not qualified for any kind of scholarship support from university. (Report 6) (Procedure 2).

DELIMITER //

CREATE PROCEDURE PhD\_Unqualified(IN req\_GPA float)

BEGIN

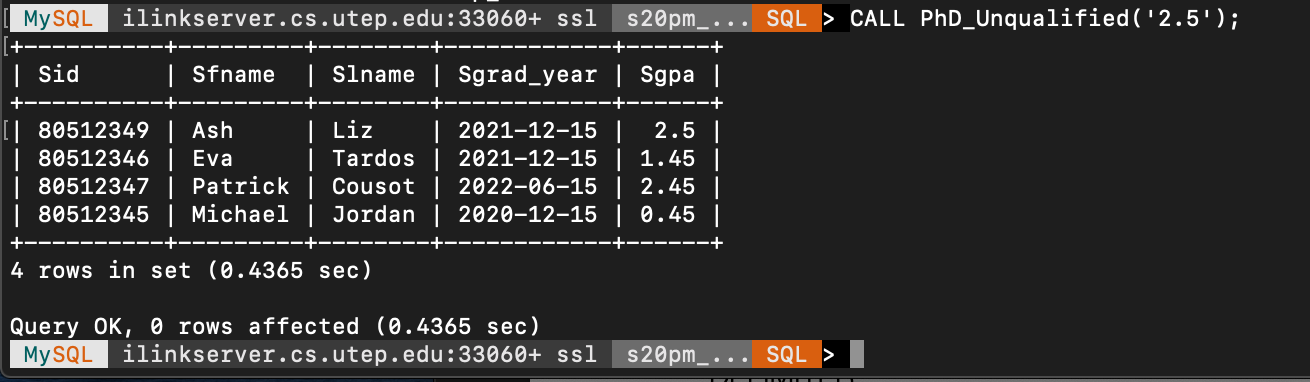
SELECT Sid, Sfname, Slname, Sgrad\_year, Sgpa

FROM Student WHERE Sgpa <= req\_GPA;

END //

DELIMITER ;

CALL PhD\_Unqualified(‘2.5');



## Trigger Used To Satisfy Requirement 30

Requirement 30: The system shall keep track of students' changes to their record of attended conferences (Trigger 1).

CREATE TABLE Stud\_conferences\_update(Stud\_utep\_id INT AUTO\_INCREMENT PRIMARY KEY,

Sconferences char(50), changeDate DATETIME DEFAULT NULL, action

VARCHAR(50) DEFAULT NULL);

CREATE TRIGGER before\_stud\_conferences\_update

BEFORE UPDATE ON Conference\_attended

FOR EACH ROW

INSERT INTO Stud\_conferences\_update

SET action=’update’,

Stud\_username = OLD.Susername,

Sconferences = OLD.Conference\_attended,

changeDate = NOW();

## Trigger Used To Satisfy Requirement 31

Requirement 31: The system shall keep track of students' changes to their record of published papers (Trigger 2).

CREATE TABLE Stud\_publications\_update(Stud\_utep\_id INT AUTO\_INCREMENT

PRIMARY KEY,

Spublications char(50), changeDate DATETIME DEFAULT NULL, action

VARCHAR(50) DEFAULT NULL);

CREATE TRIGGER before\_stud\_publication\_update

BEFORE UPDATE ON Student\_papers

FOR EACH ROW

INSERT INTO Stud\_publications\_update

SET action=’update’,

Stud\_utep\_id = OLD.Sutep\_id,

Spublications = OLD.Student\_papers,

changeDate = NOW();

# 11. Reports

/\*

1.13

REPORTS RESPONSABILITY

For each one of the reports required in your system:

.1.  Trace back to functional requirements.

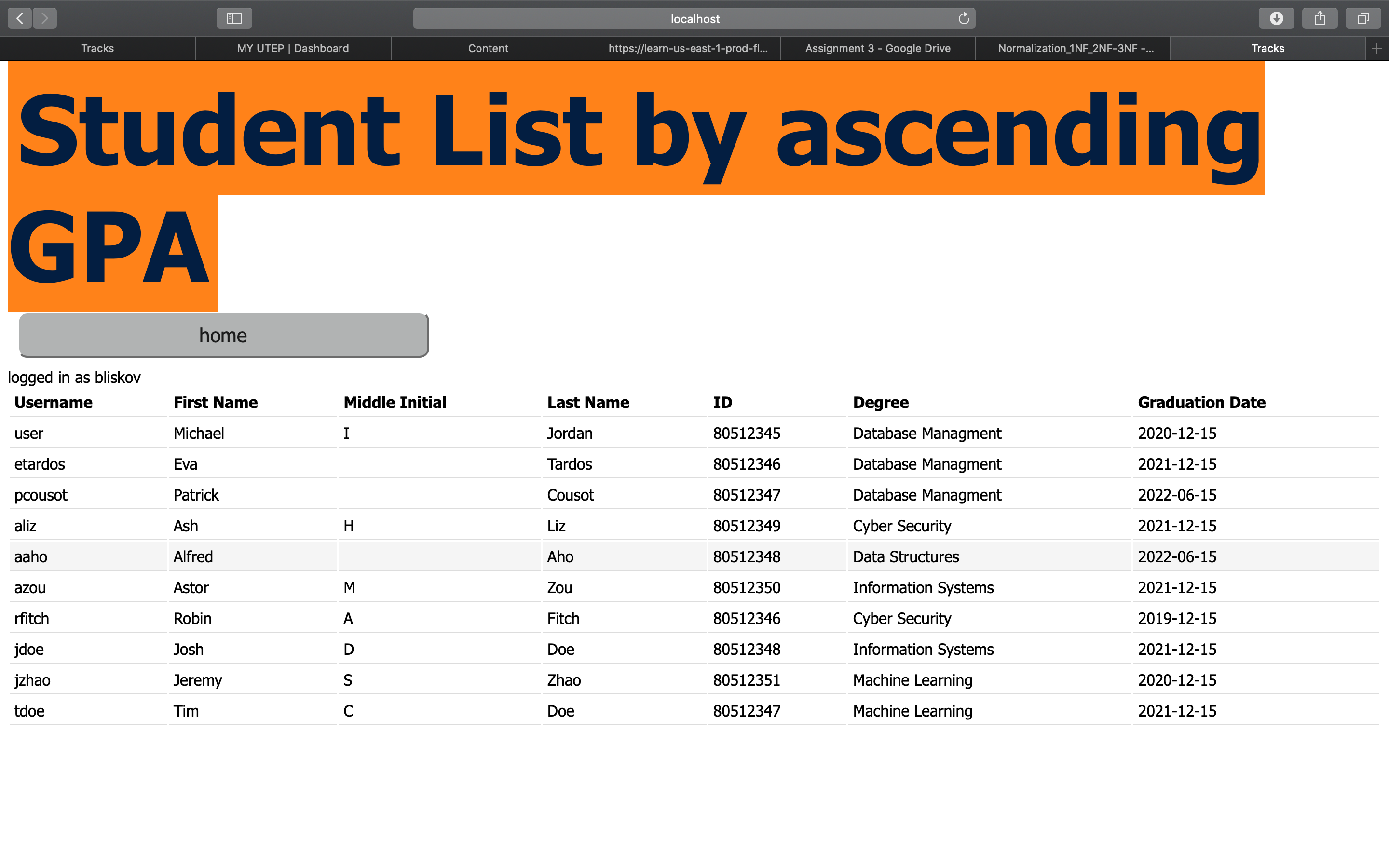
.2.  List the queries, functions, etc. required to generate the report.

.3.  Include a screenshot of the data returned after running your report in MySQL.

\*/

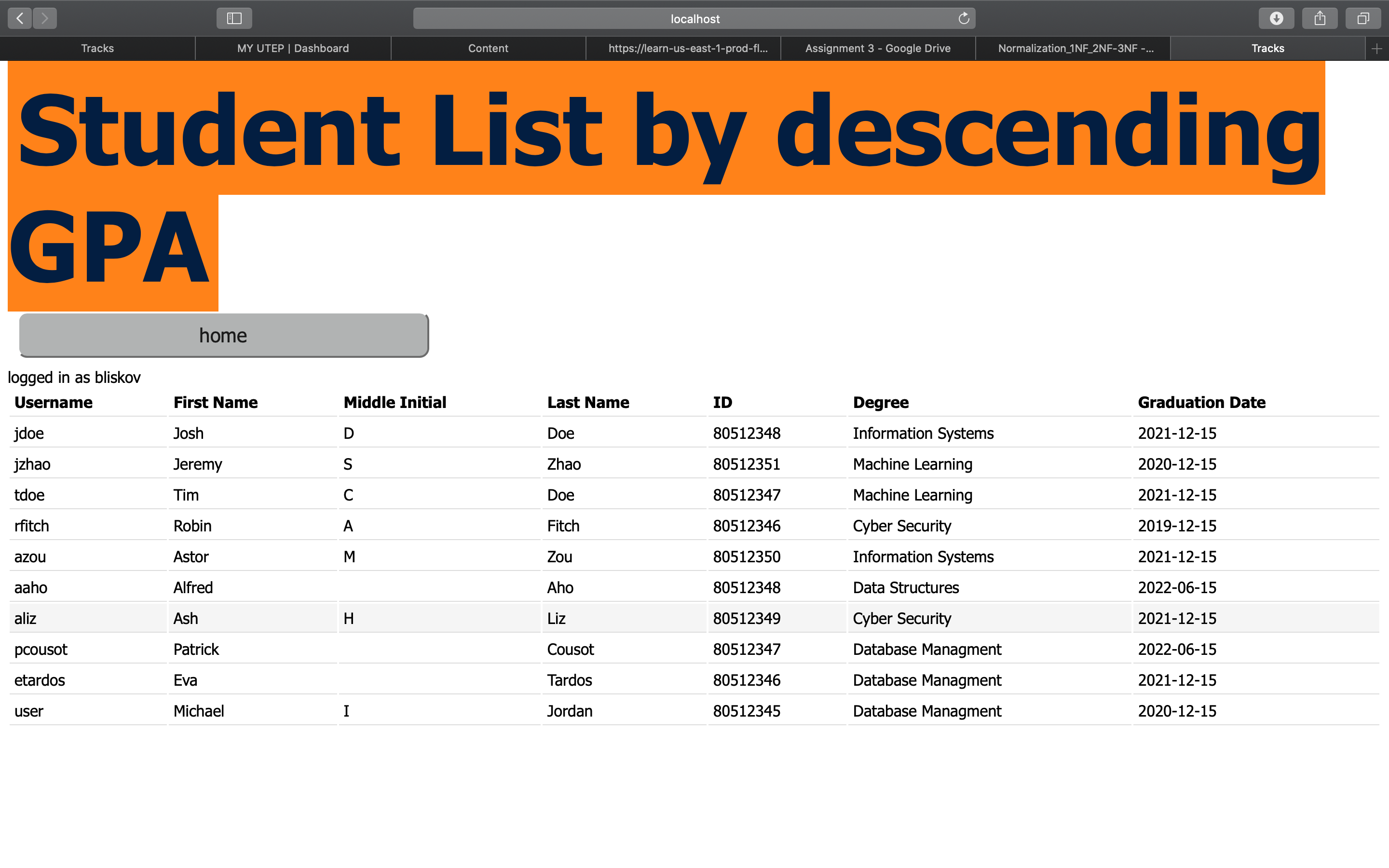
## SQL Queries Required To Satisfy Requirement 32

Requirement 32: The system shall allow admin users to view a list of current students by ascending GPA attribute (Report 1).

SELECT \* FROM Student ORDER BY Sgpa;

## SQL Queries Required To Satisfy Requirement 33

Requiremtent 33: The system shall allow admin users to view a list of current students by descending GPA attribute (Report 2).

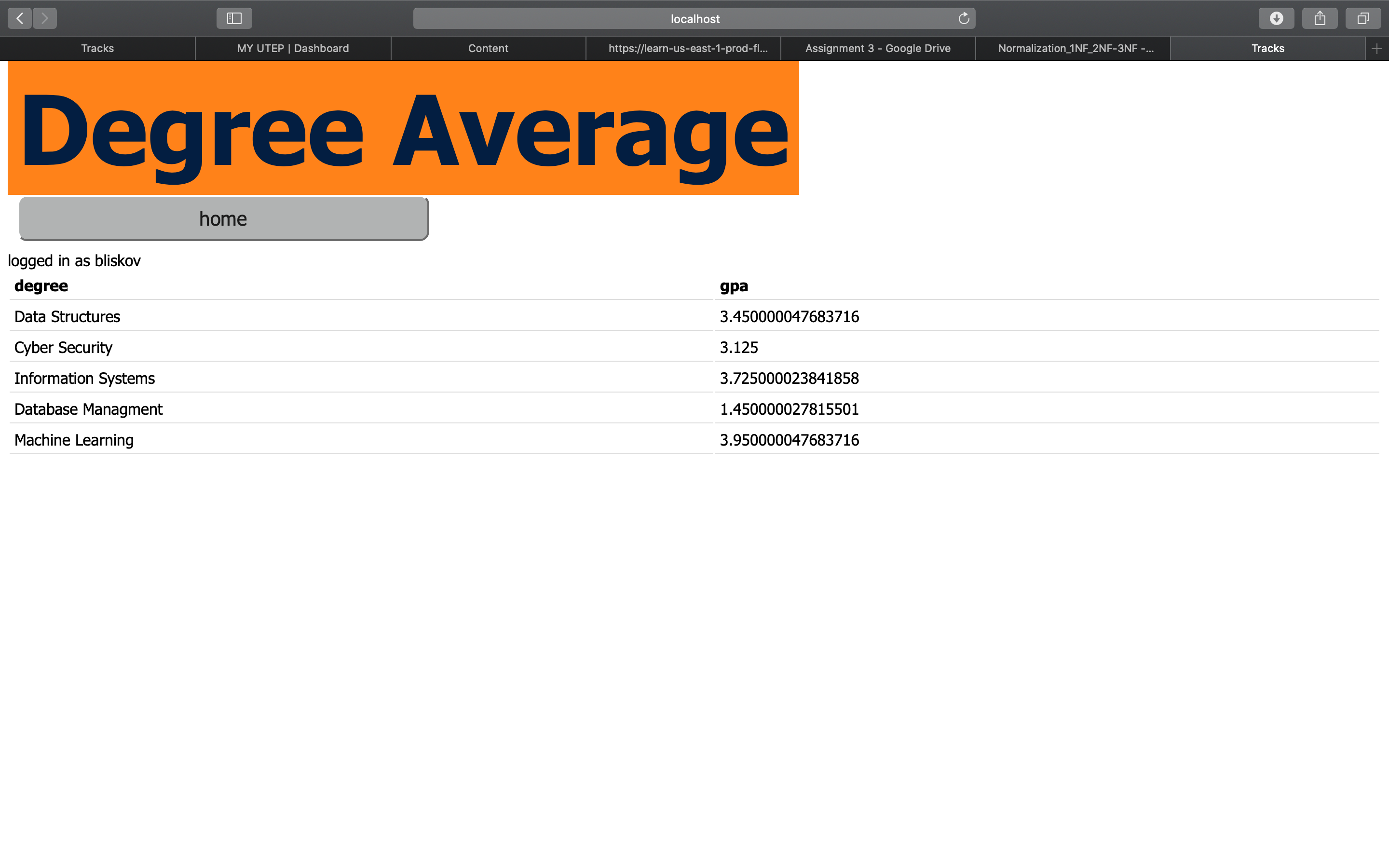
SELECT \* FROM Student ORDER BY Sgpa DESC;

## SQL Queries Required To Satisfy Requirement 34

Requirement 34: The system shall allow admin users to view a list of degrees and each degree’s average student gpa. (Report 3).

SELECT Sdegree, AVG(Sgpa)

FROM Student

GROUP BY 1;

## SQL Queries Required To Satisfy Requirement 36

The system shall allow admin users to view a list of students from Machine Learning degree who meet the GPA requirements to be eligible for teaching assistantships (3.75). (Report 5) (Procedure 1).

DELIMITER //

CREATE PROCEDURE PhD\_ML\_Assistantships(IN req\_GPA float)

BEGIN

SELECT Sutep\_id, Sfname, Slname, Sgrad\_year, Sgpa

FROM Student WHERE Sgpa >= req\_GPA AND Sdegree = 'Machine learning';

END //

DELIMITER ;

CALL PhD\_ML\_Assistantships('3.75');

## SQL Queries Required To Satisfy Requirement 37

Requirement 37: The system shall allow admin users to view a list of students who are not qualified for any kind of scholarship support from university. (Report 6) (Procedure 2).

DELIMITER //

CREATE PROCEDURE PhD\_Unqualified(IN req\_GPA float)

BEGIN

SELECT Sid, Sfname, Slname, Sgrad\_year, Sgpa

FROM Student WHERE Sgpa <= req\_GPA;

END //

DELIMITER ;

CALL PhD\_ML\_Assistantships('2.5');

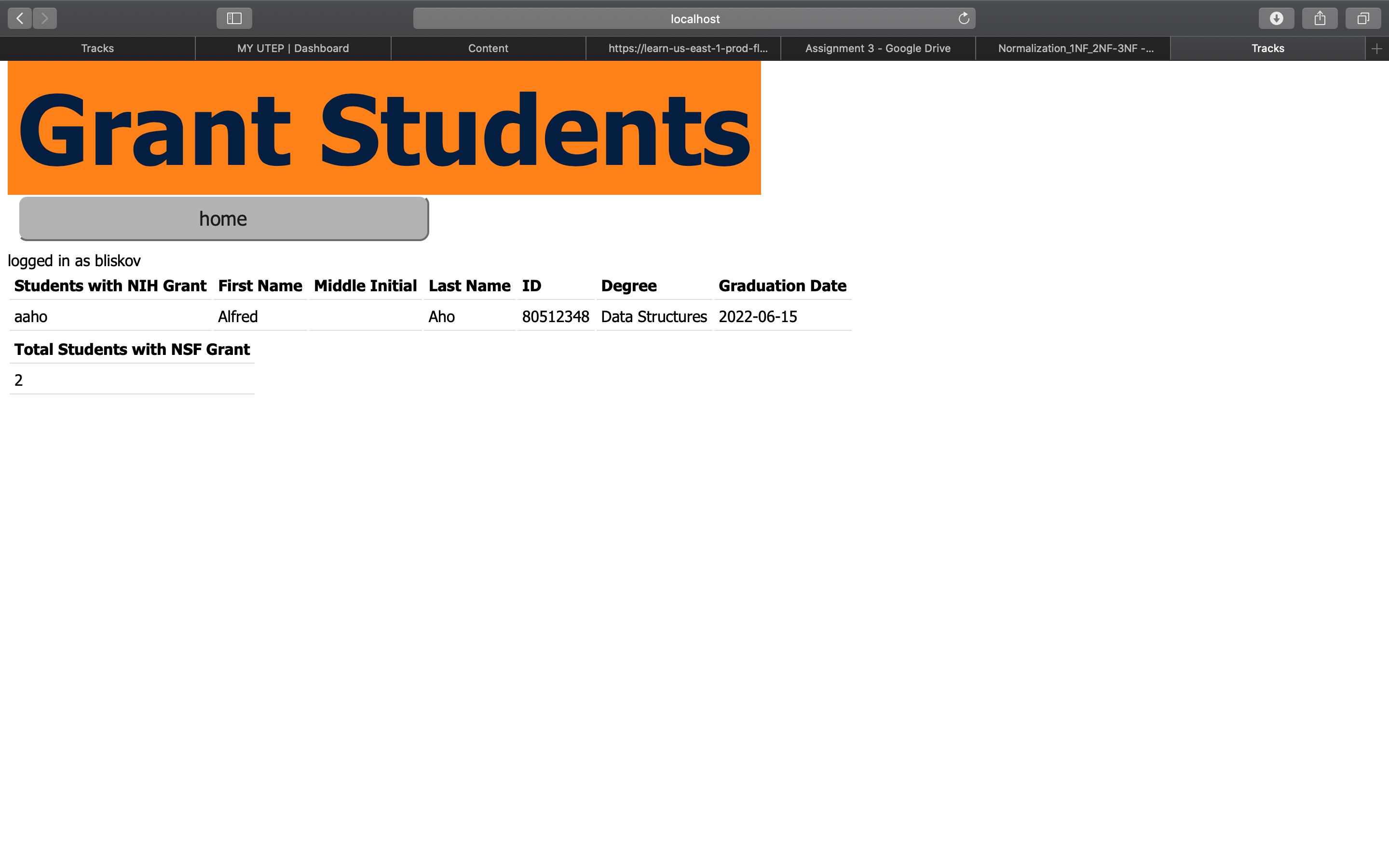
## SQL Queries Required To Satisfy Requirement 38

Requirement 38: The system shall allow admin users to view a list of students that have the NIH grant. (View 1)

CREATE VIEW student\_grantinfo\_NIH AS SELECT Suser\_name FROM STUDENT WHERE Sgrant\_info='NIH';

## SQL Queries Required To Satisfy Requirement 39

Requirement 39: The system shall allow admin users to view a total count of how many students have received the NSF grant (View 2).

CREATE VIEW Student\_totalNSF AS SELECT COUNT(Sgrant\_info) AS total\_students FROM STUDENT WHERE Sgrant\_info = 'NSF';

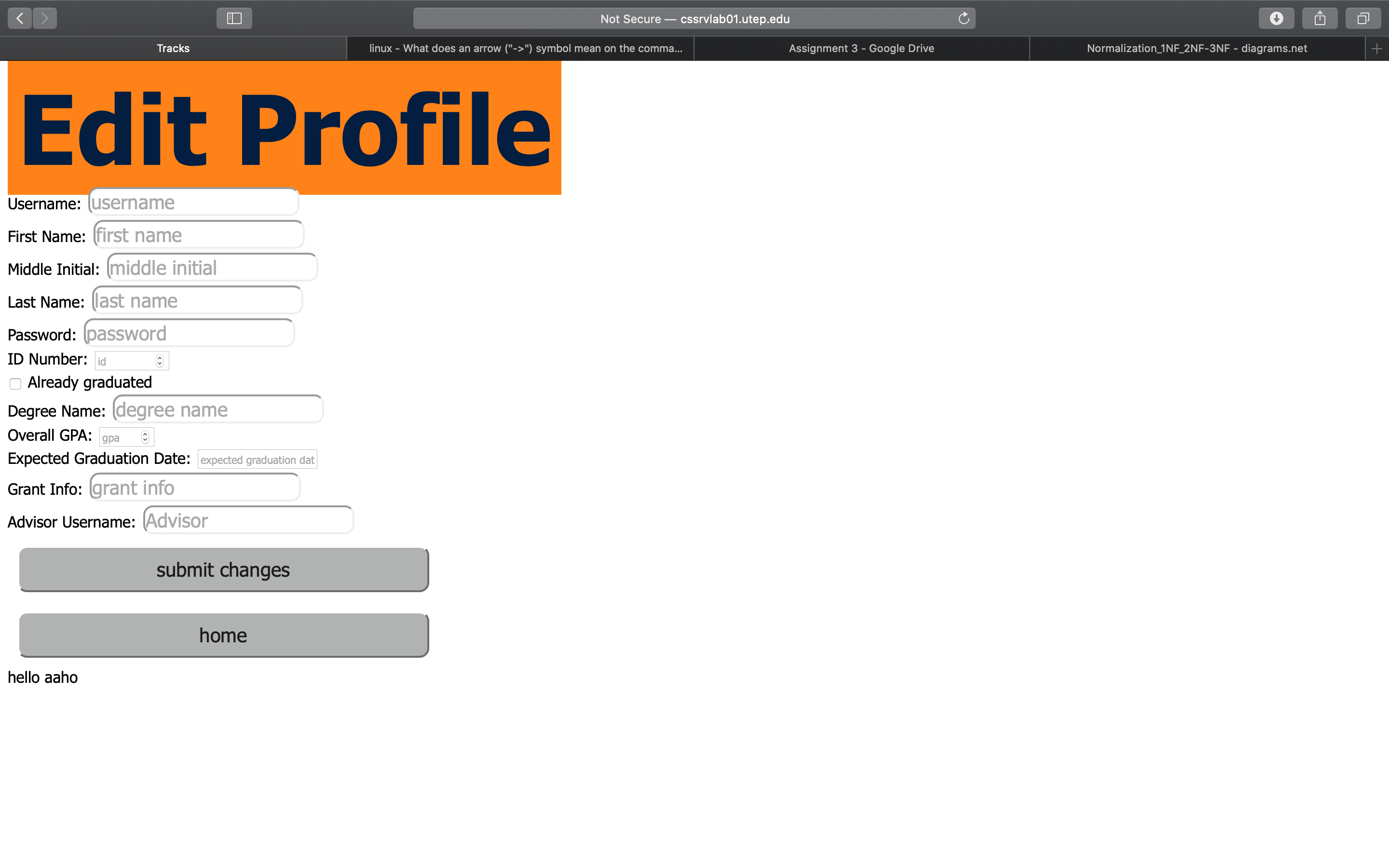
# 12. Graphical User Interface

/\*

1.14

INTERFACE RESPONSABILITY

Include a screenshot of your user interface and describe how you insert, retrieve, modify

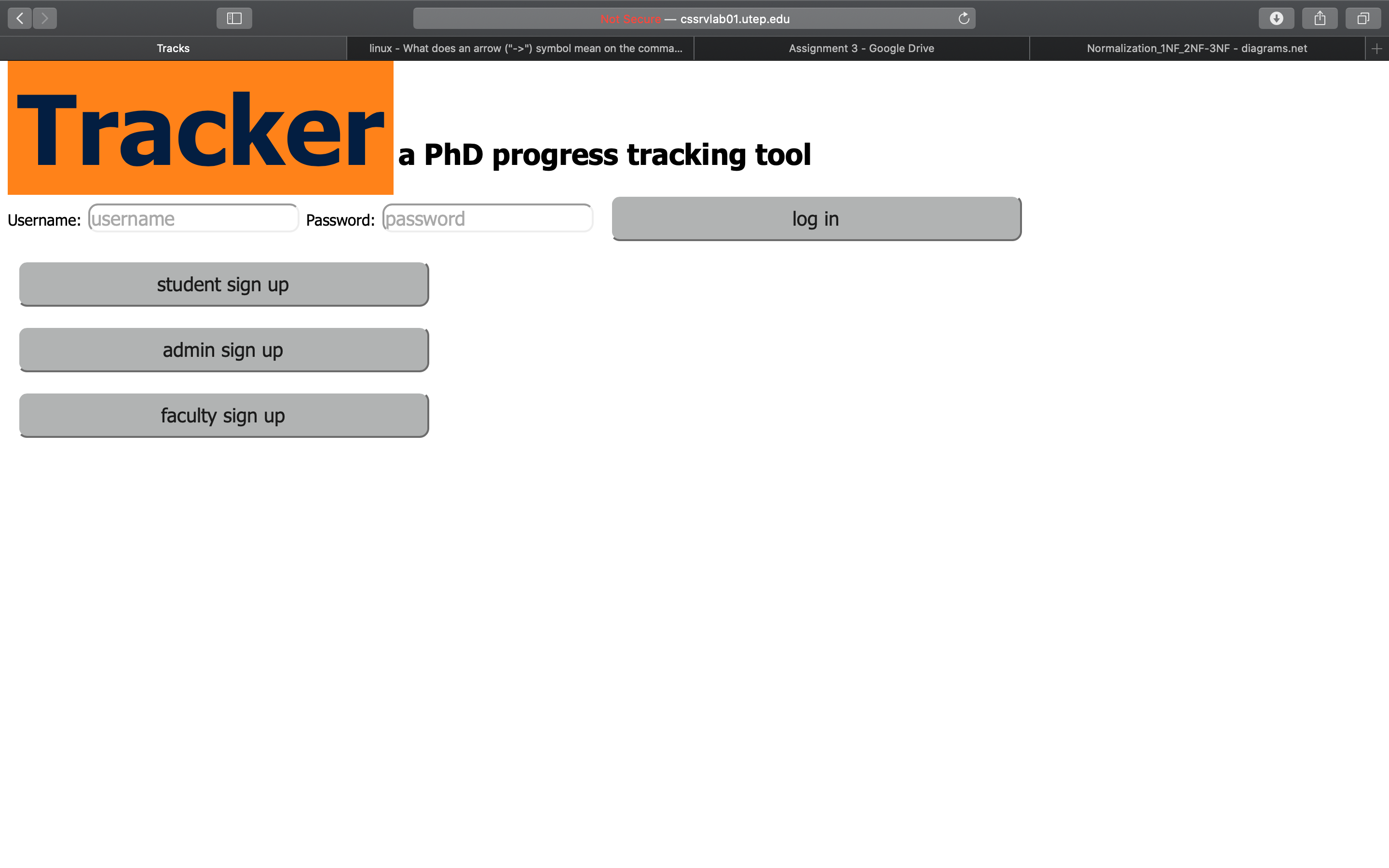
and delete data from your database.

Include a screenshot of your user interface generating the reports required in your

system.

\*/

## Screenshot of login Page

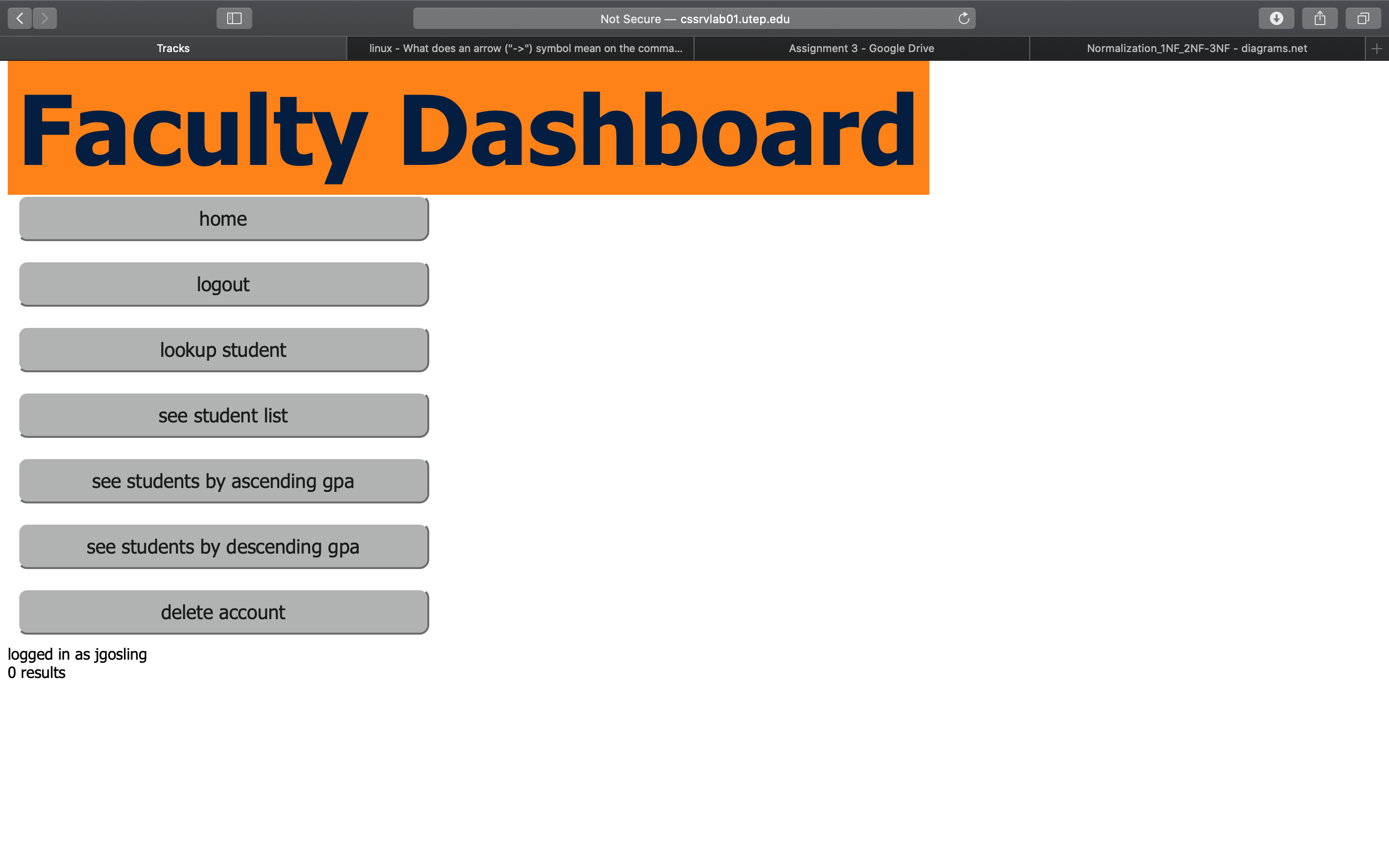


## Screenshot of Student-welcome Page

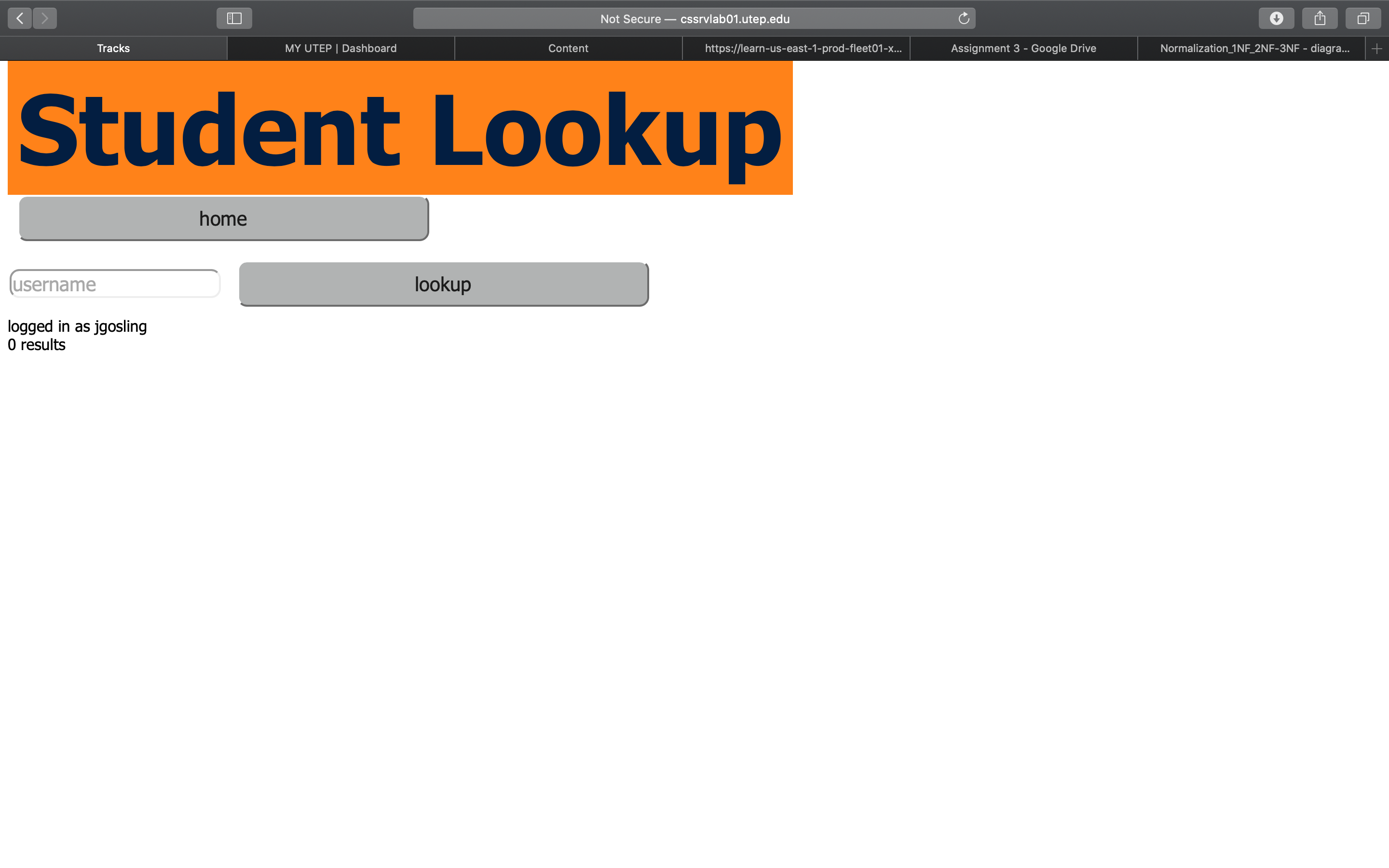
## Screenshot of Student-edit Page

## Screenshot of student add milestone page

## Screenshot of faculty welcome Page



## Screenshot of faculty-search Page



## Screenshot of Admin-welcome Page

## 

# 15. References in APA Style

1. (2020) The University of Texas at El Paso Graphic Identity Guide [PDF]. El Paso, Texas: The University of Texas at El Paso. [www.utep.edu/university-communications/\_Files/docs/UTEP-GRAPHIC-IDENTITY-GUIDE.pdf](http://www.utep.edu/university-communications/_Files/docs/UTEP-GRAPHIC-IDENTITY-GUIDE.pdf)
2. The Graduate Program Committee with contributions from CS Faculty (2019) Ph.D. Program Handbook v2.0 [PDF]. El Paso, Texas: The University of Texas at El Paso. [www.utep.edu/cs/graduate/PDFs/PhD-Handbook-2019-2020.pdf](http://www.utep.edu/cs/graduate/PDFs/PhD-Handbook-2019-2020.pdf)

# 16. Appendix A. Contribution Information

## Activities Relating to Database Scope

* Varius natoque turpis elementum est. Duis montes, tellus lobortis lacus amet arcu et. In vitae vel.
* Wisi at id praesent bibendum libero faucibus porta egestas, quisque praesent ipsum fermentum tempor. Curabitur auctor, erat mollis sed.
* Turpis vivamus a dictumst congue magnis. Aliquam amet ullamcorper dignissim molestie, mollis. Tortor vitae tortor eros wisi facilisis.

## Activities Relating to Requirements

* Varius natoque turpis elementum est. Duis montes, tellus lobortis lacus amet arcu et. In vitae vel.
* Wisi at id praesent bibendum libero faucibus porta egestas, quisque praesent ipsum fermentum tempor. Curabitur auctor, erat mollis sed.
* Turpis vivamus a dictumst congue magnis. Aliquam amet ullamcorper dignissim molestie, mollis. Tortor vitae tortor eros wisi facilisis.

## Activities Relating to Assumptions

* Varius natoque turpis elementum est. Duis montes, tellus lobortis lacus amet arcu et. In vitae vel.
* Wisi at id praesent bibendum libero faucibus porta egestas, quisque praesent ipsum fermentum tempor. Curabitur auctor, erat mollis sed.
* Turpis vivamus a dictumst congue magnis. Aliquam amet ullamcorper dignissim molestie, mollis. Tortor vitae tortor eros wisi facilisis.

## Activities Relating to Entity-Relationship Model

* Varius natoque turpis elementum est. Duis montes, tellus lobortis lacus amet arcu et. In vitae vel.
* Wisi at id praesent bibendum libero faucibus porta egestas, quisque praesent ipsum fermentum tempor. Curabitur auctor, erat mollis sed.
* Turpis vivamus a dictumst congue magnis. Aliquam amet ullamcorper dignissim molestie, mollis. Tortor vitae tortor eros wisi facilisis.

## Activities Relating to Relational Model

* Varius natoque turpis elementum est. Duis montes, tellus lobortis lacus amet arcu et. In vitae vel.
* Wisi at id praesent bibendum libero faucibus porta egestas, quisque praesent ipsum fermentum tempor. Curabitur auctor, erat mollis sed.
* Turpis vivamus a dictumst congue magnis. Aliquam amet ullamcorper dignissim molestie, mollis. Tortor vitae tortor eros wisi facilisis.

## Activities Relating to MySQL Server

* Varius natoque turpis elementum est. Duis montes, tellus lobortis lacus amet arcu et. In vitae vel.
* Wisi at id praesent bibendum libero faucibus porta egestas, quisque praesent ipsum fermentum tempor. Curabitur auctor, erat mollis sed.
* Turpis vivamus a dictumst congue magnis. Aliquam amet ullamcorper dignissim molestie, mollis. Tortor vitae tortor eros wisi facilisis.

## Activities Relating to General Document

* Varius natoque turpis elementum est. Duis montes, tellus lobortis lacus amet arcu et. In vitae vel.
* Wisi at id praesent bibendum libero faucibus porta egestas, quisque praesent ipsum fermentum tempor. Curabitur auctor, erat mollis sed.
* Turpis vivamus a dictumst congue magnis. Aliquam amet ullamcorper dignissim molestie, mollis. Tortor vitae tortor eros wisi facilisis.

## Team Member Signatures

Abhijeet Patil: Date:

Alex Zamora: Date:

Matt Montoya: Date:

Jorge Felix: Date: