Mini project report on

University Capstone Projects Management

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

UE21CS351 – DBMS Project

Submitted by:

Darsh Patel	**PES2UG21CS150**
Gautam Panday	**PES2UG21CS176**

Under the guidance of

Prof. Nivedita Kasturi

Assistant Professor

Designation

PES University

AUG - DEC 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India.

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

University Capstones Project Management

is a Bonafide work carried out by

Gautam Panday	**PES2UG21CS176**
Darsh Patel	**PES2UG21CS150**

In partial fulfilment for the completion of fifth semester DBMS Project (UE20CSS301) in the Program of Study Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period AUG. 2023 – DEC. 2023. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 5th semester academic requirements in respect of project work.

Signature
Prof Nivedita
Assistant Professor

DECLARATION

We hereby declare that the DBMS Project entitled **Capstone Project Management** has been carried out by us under the guidance of **Prof. Nivedita , Assistant Professor** and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester AUG – DEC 2023.

Gautam Panday	**PES2UG21CS176**
Darsh Patel	**PES2UG21CS150**

Signature

ACKNOWLEDGEMENT

I would like to express my gratitude to Prof. Nivedita, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE21CS351 - DBMS Project.

I take this opportunity to thank Dr. Sandesh , Professor, ChairPerson , Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department.

I am deeply grateful to Dr. M.R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr.Suryaprasad J, ViceChancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this DBMS Project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

The "Capstone Project Management System" is a Streamlit application designed to create user interactions with a database seamlessly . This project caters to three primary user roles: admin, professor and students, each with distinct functionalities and privileges.

Key Features and Components:

User Authentication:

- The system incorporates user authentication, ensuring that only authorized individuals can access specific features.
- Administrators and regular users have different login credentials, granting them access to their respective dashboards.

Query Processing:

- Users can input their queries using either voice or text inputs. This flexibility allows individuals to choose their preferred mode of interaction.
- Upon receiving a query, the system forwards it to a server for analysis and response generation.

Database Integration:

- A MySQL database lies at the heart of the application, serving as the repository for user data, queries, and the corresponding answers.
- This database integration enables efficient storage, retrieval, and management of information.

User Interface:

- The application boasts a user-friendly interface with intuitive controls. These include buttons for voice recording, text input, and query submission.
- The interface is designed to be accessible and responsive, catering to users of various technical proficiencies.

Application Flow:

1. **User Authentication:**
   - Users log in with their unique credentials.
   - The system verifies the user's role (administrator or regular user).
2. **Query Processing:**
   - Regardless of the input method, the queries are sent to a server for processing.
   - The server analyses the queries and retrieves relevant answers from the database.
3. **Response Display:**

- The system presents the responses to the users in an easy-to-understand format.
- Responses are displayed on the user interface, making the information readily accessible.

5. **Admin Dashboard:**

- Administrators have access to an exclusive dashboard with advanced features for managing and retrieving queries and answers

TABLE OF CONTENTS

1. INTRODUCTION

Welcome to the Capstone Project Management System, a comprehensive platform designed to streamline and enhance the capstone project experience for students, professors, and administrators alike. In this innovative system, we cater to the unique roles of three primary users: students, professors, and administrators.

For students, the system facilitates collaboration by allowing them to form teams with two other students, creating a dynamic environment for project development. However, the journey doesn't end there. To ensure the academic rigor of the projects, students must seek the approval of a designated professor before proceeding. This ensures that the projects align with educational standards and receive valuable guidance from experienced faculty members.

Professors play a crucial role in the Capstone Project Management System, not only guiding students but also assessing their performance through various evaluations. Students undergo three internal assessments named ISA1, ISA2, and ISA3, along with a semester-ending examination, ESA. Furthermore, the collaborative efforts of student teams are subject to evaluation by a panel of professors, ensuring a comprehensive assessment that reflects both individual and group contributions.

At the helm of this sophisticated system is the administrator, who holds a managerial role in handling the database and addressing any discrepancies that may arise. The administrator ensures the smooth functioning of the system, guaranteeing data integrity and resolving issues promptly. This pivotal role contributes to the overall efficiency and reliability of the Capstone Project Management System. The administrator has got all the permissions of root.

In essence, our system provides a structured and transparent framework for managing capstone projects, fostering collaboration, academic excellence, and administrative efficiency. Welcome to a platform where innovation meets education, and where the capstone project journey is elevated to new heights.

2. PROBLEM DEFINITION

Problem Statement:
Every student in the university has to make a capstone project in pre-final year. These projects are extremely important for placements/higher studies. Thus there is a need for a proper digitalized system for managing these projects which if done manually is a very tedious job for the university.

Proposed Solution:

The University Capstone Projects Management System is a software solution aimed at simplifying and enhancing the management of capstone projects within a university setting. This system is intended to streamline the entire lifecycle of capstone projects, from project proposal submission to evaluation and grading. The purpose of this system is to provide an efficient and organized platform for students, faculty advisors, and administrators to collaborate and manage capstone projects effectively.

Benefits of the Proposed Solution:

1. Efficiency and Time Savings: The system automates and digitizes the entire lifecycle of capstone projects, significantly reducing the time and effort required for administrative tasks.
Automated workflows for project proposal submission, approval processes, and evaluation save time for both students and faculty.
Centralized Information Hub:

2.Acts as a centralized repository for all capstone project-related information, making it easily accessible to students, professors, and administrators.
Eliminates the need for manual record-keeping and reduces the risk of data loss or mismanagement.
Enhanced Collaboration:

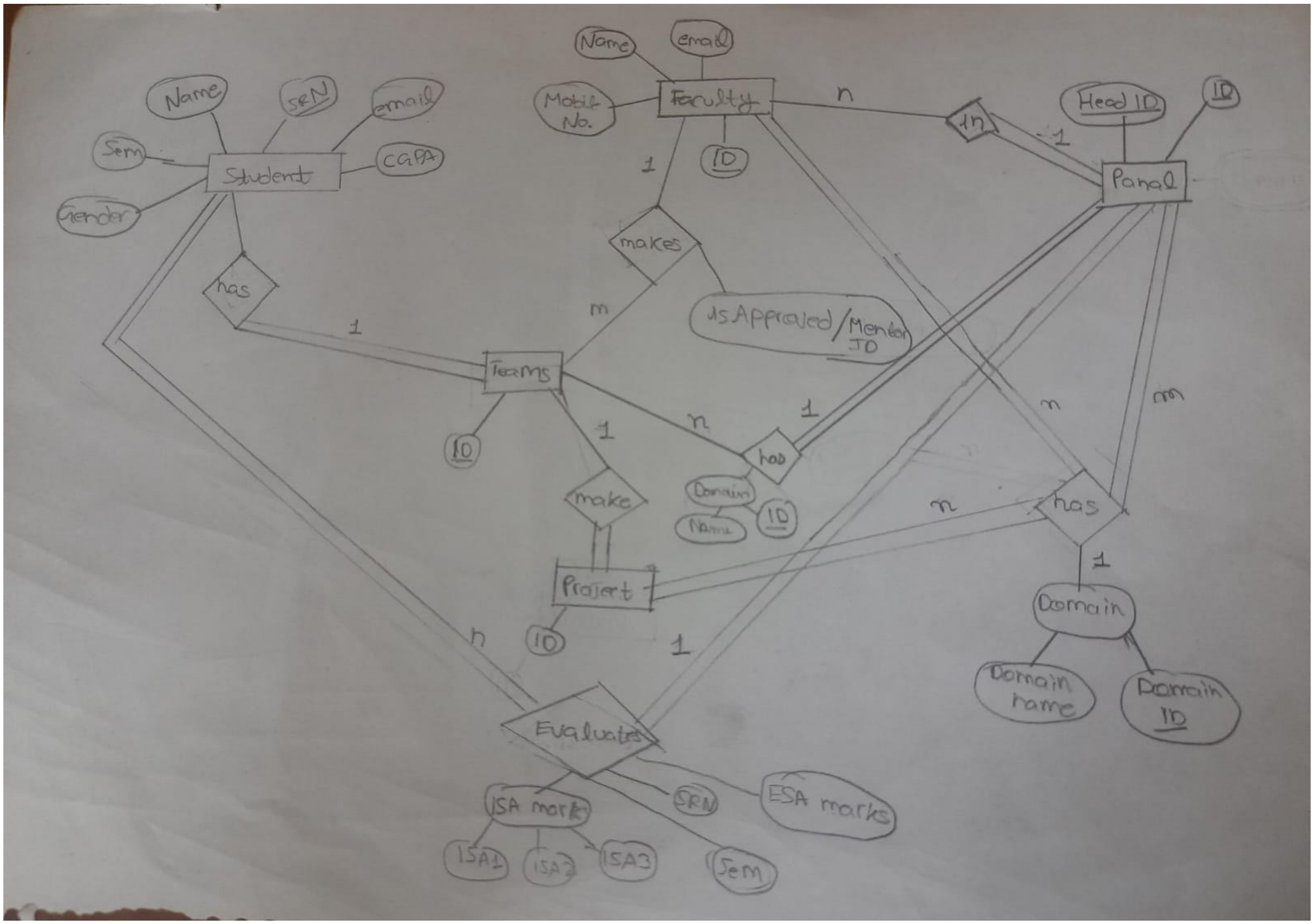
3.Facilitates seamless collaboration among students forming project teams, ensuring effective communication and coordination throughout the project duration.
Enables real-time updates, file sharing, and communication within the system, fostering a collaborative environment.

4.Improved Proposal Management: Simplifies the process of submitting and tracking project proposals, allowing for timely feedback and approval from faculty advisors.

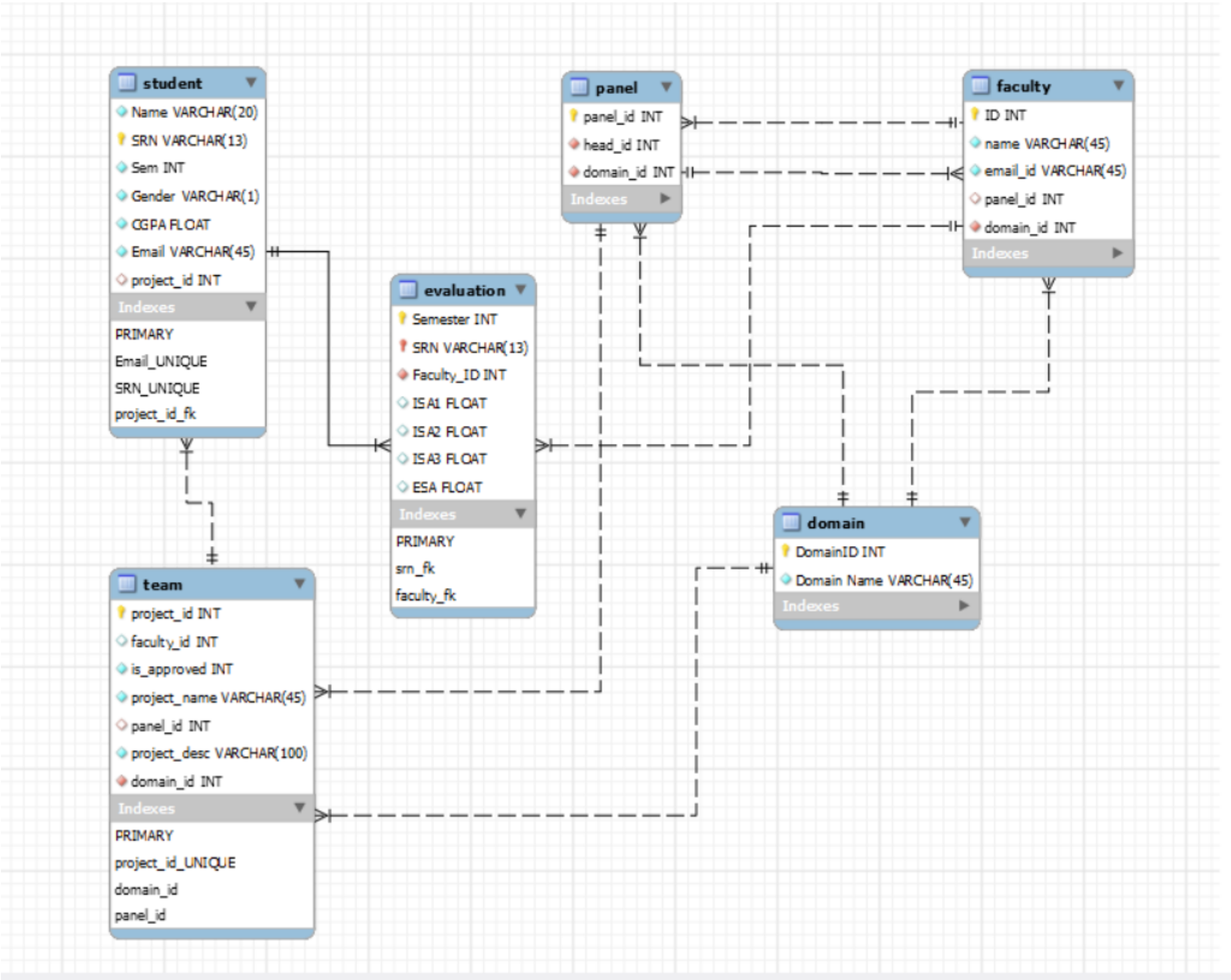
Provides a transparent view of the proposal status, reducing delays in project initiation.

5.Structured Evaluation and Grading: Standardizes the evaluation process with predefined criteria for ISA assessments, ESA examinations, and panel evaluations.
Ensures fair and consistent grading, eliminating potential biases and subjectivity in the assessment of capstone projects.

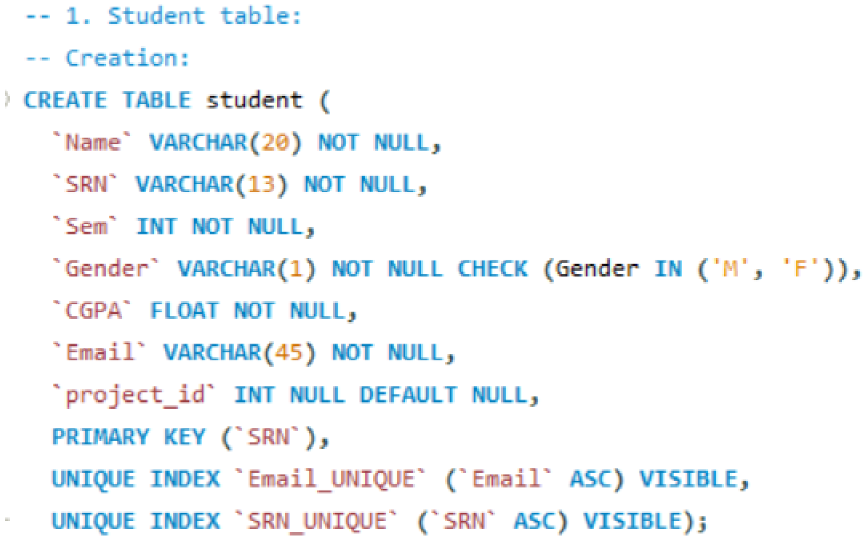
3.ER MODEL


Name
email
Mobile No.
Faculty
ID
n
1n
1
Head ID
ID
Panal
Name
SRN
email
Sem
Student
CGPA
Gender
has
1
1
makes
m
isApproved/Mentor ID
Teams
ID
1
n
1
has
Domain
ID
Name
make
Project
ID
n
m
n
has
1
Domain
Domain name
Domain ID
n
1
Evaluates
ISA marks
SRN
ESA marks
ISA1
ISA2
ISA3
Sem

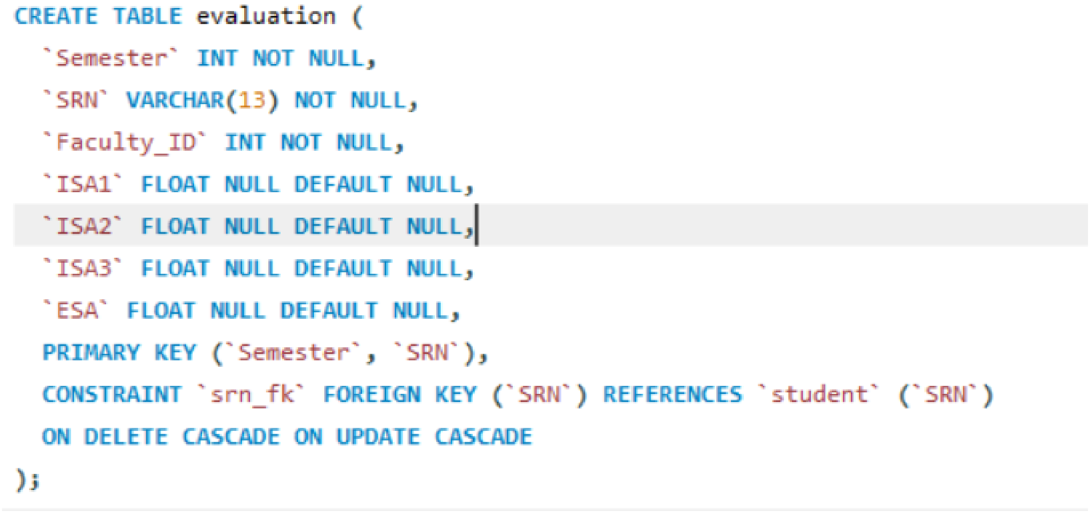
4.ER TO RELATIONAL MAPPING



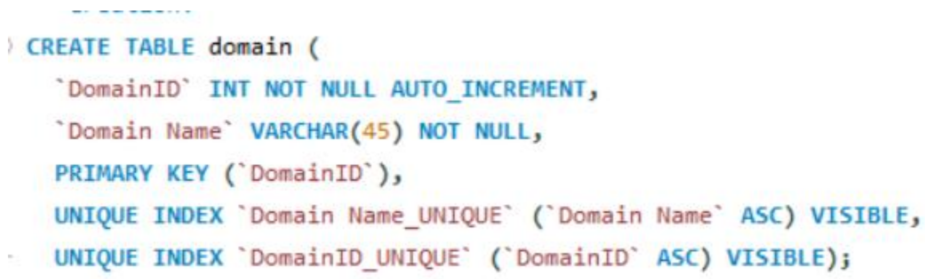
5.DDL STATEMENTS



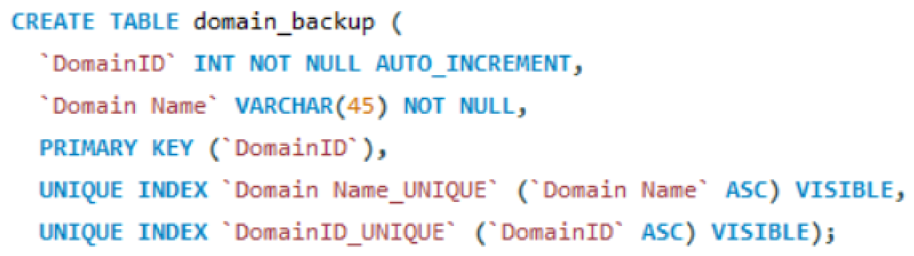
```sql
-- 1. Student table:
-- Creation:
CREATE TABLE student (
  `Name` VARCHAR(20) NOT NULL,
  `SRN` VARCHAR(13) NOT NULL,
  `Sem` INT NOT NULL,
  `Gender` VARCHAR(1) NOT NULL CHECK (Gender IN ('M', 'F')),
  `CGPA` FLOAT NOT NULL,
  `Email` VARCHAR(45) NOT NULL,
  `project_id` INT NULL DEFAULT NULL,
  PRIMARY KEY (`SRN`),
  UNIQUE INDEX `Email_UNIQUE` (`Email` ASC) VISIBLE,
  UNIQUE INDEX `SRN_UNIQUE` (`SRN` ASC) VISIBLE);
```



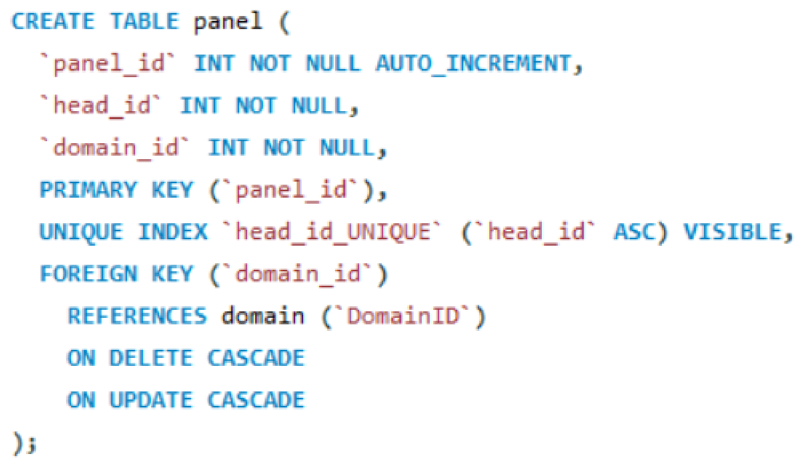
```sql
CREATE TABLE evaluation (
  `Semester` INT NOT NULL,
  `SRN` VARCHAR(13) NOT NULL,
  `Faculty_ID` INT NOT NULL,
  `ISA1` FLOAT NULL DEFAULT NULL,
  `ISA2` FLOAT NULL DEFAULT NULL,
  `ISA3` FLOAT NULL DEFAULT NULL,
  `ESA` FLOAT NULL DEFAULT NULL,
  PRIMARY KEY (`Semester`, `SRN`),
  CONSTRAINT `srn_fk` FOREIGN KEY (`SRN`) REFERENCES `student` (`SRN`)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```



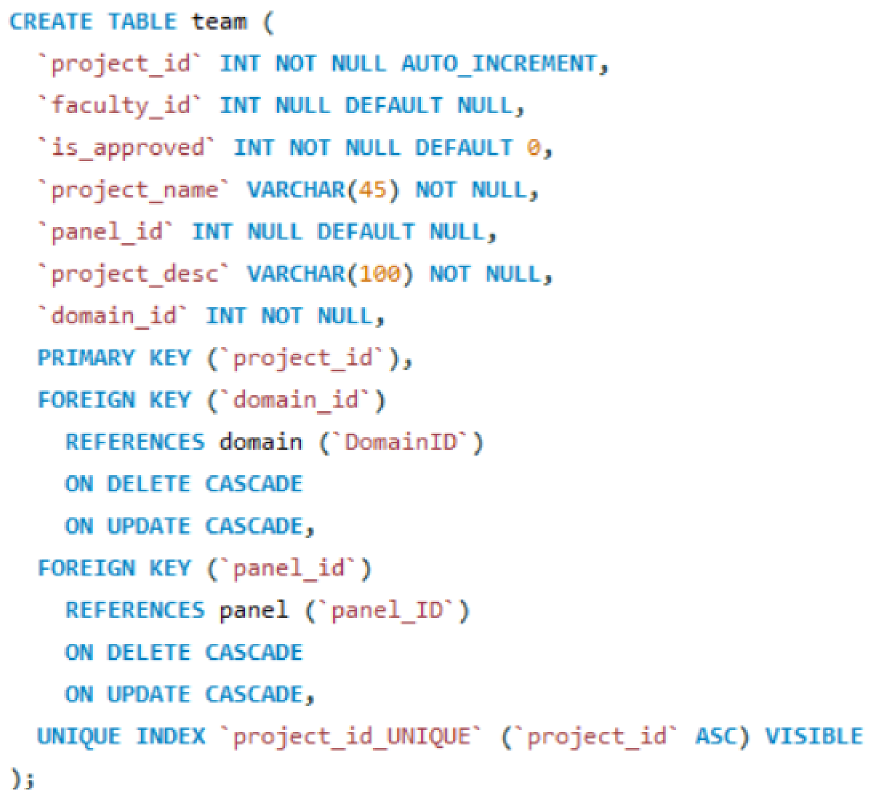
```sql
CREATE TABLE domain (
  `DomainID` INT NOT NULL AUTO_INCREMENT,
  `Domain Name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`DomainID`),
  UNIQUE INDEX `Domain Name_UNIQUE` (`Domain Name` ASC) VISIBLE,
  UNIQUE INDEX `DomainID_UNIQUE` (`DomainID` ASC) VISIBLE);
```



```sql
CREATE TABLE domain_backup (
  `DomainID` INT NOT NULL AUTO_INCREMENT,
  `Domain Name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`DomainID`),
  UNIQUE INDEX `Domain Name_UNIQUE` (`Domain Name` ASC) VISIBLE,
  UNIQUE INDEX `DomainID_UNIQUE` (`DomainID` ASC) VISIBLE);
```

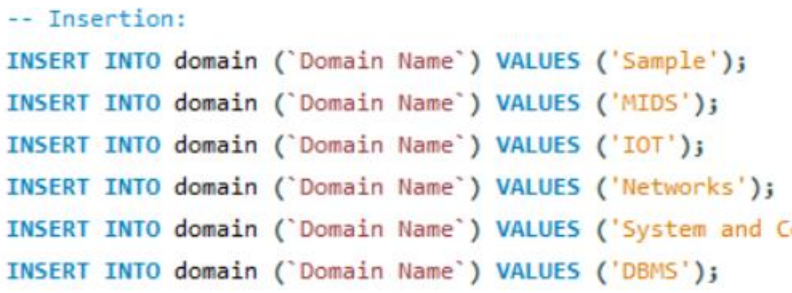


```sql
CREATE TABLE panel (
  `panel_id` INT NOT NULL AUTO_INCREMENT,
  `head_id` INT NOT NULL,
  `domain_id` INT NOT NULL,
  PRIMARY KEY (`panel_id`),
  UNIQUE INDEX `head_id_UNIQUE` (`head_id` ASC) VISIBLE,
  FOREIGN KEY (`domain_id`)
    REFERENCES domain (`DomainID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

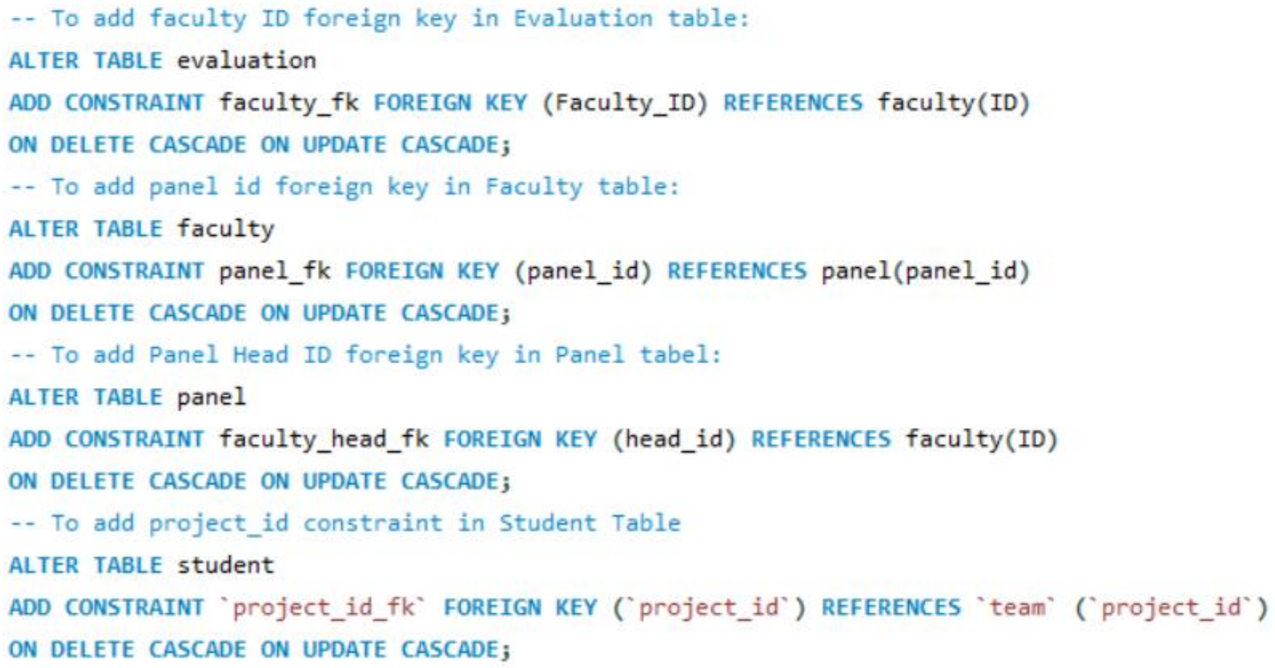


```sql
CREATE TABLE team (
  `project_id` INT NOT NULL AUTO_INCREMENT,
  `faculty_id` INT NULL DEFAULT NULL,
  `is_approved` INT NOT NULL DEFAULT 0,
  `project_name` VARCHAR(45) NOT NULL,
  `panel_id` INT NULL DEFAULT NULL,
  `project_desc` VARCHAR(100) NOT NULL,
  `domain_id` INT NOT NULL,
  PRIMARY KEY (`project_id`),
  FOREIGN KEY (`domain_id`)
    REFERENCES domain (`DomainID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (`panel_id`)
    REFERENCES panel (`panel_ID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  UNIQUE INDEX `project_id_UNIQUE` (`project_id` ASC) VISIBLE
);
```

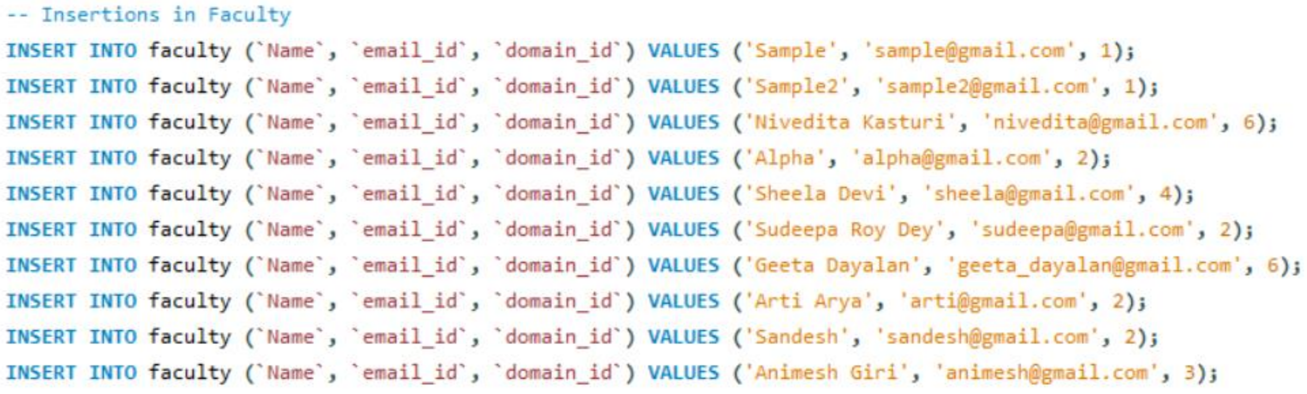
6. DML STATEMENTS



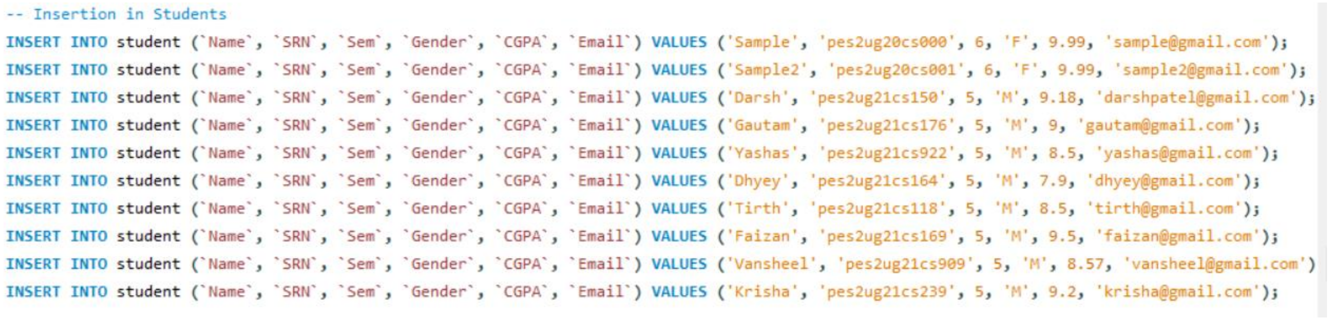
```sql
-- Insertion:
INSERT INTO domain (`Domain Name`) VALUES ('Sample');
INSERT INTO domain (`Domain Name`) VALUES ('MIDS');
INSERT INTO domain (`Domain Name`) VALUES ('IOT');
INSERT INTO domain (`Domain Name`) VALUES ('Networks');
INSERT INTO domain (`Domain Name`) VALUES ('System and Core');
INSERT INTO domain (`Domain Name`) VALUES ('DBMS');
```



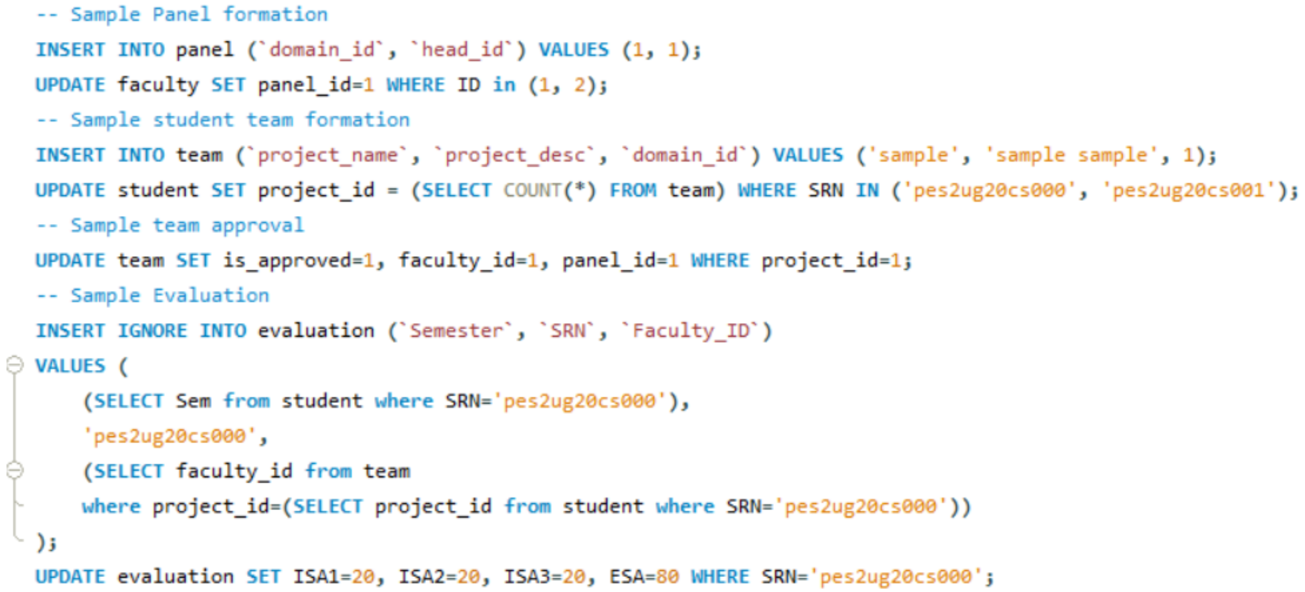
```sql
-- To add faculty ID foreign key in Evaluation table:
ALTER TABLE evaluation
ADD CONSTRAINT faculty_fk FOREIGN KEY (Faculty_ID) REFERENCES faculty(ID)
ON DELETE CASCADE ON UPDATE CASCADE;
-- To add panel id foreign key in Faculty table:
ALTER TABLE faculty
ADD CONSTRAINT panel_fk FOREIGN KEY (panel_id) REFERENCES panel(panel_id)
ON DELETE CASCADE ON UPDATE CASCADE;
-- To add Panel Head ID foreign key in Panel tabel:
ALTER TABLE panel
ADD CONSTRAINT faculty_head_fk FOREIGN KEY (head_id) REFERENCES faculty(ID)
ON DELETE CASCADE ON UPDATE CASCADE;
-- To add project_id constraint in Student Table
ALTER TABLE student
ADD CONSTRAINT `project_id_fk` FOREIGN KEY (`project_id`) REFERENCES `team` (`project_id`)
ON DELETE CASCADE ON UPDATE CASCADE;
```



```sql
-- Insertions in Faculty
INSERT INTO faculty (`Name`, `email_id`, `domain_id`) VALUES ('Sample', 'sample@gmail.com', 1);
INSERT INTO faculty (`Name`, `email_id`, `domain_id`) VALUES ('Sample2', 'sample2@gmail.com', 1);
INSERT INTO faculty (`Name`, `email_id`, `domain_id`) VALUES ('Nivedita Kasturi', 'nivedita@gmail.com', 6);
INSERT INTO faculty (`Name`, `email_id`, `domain_id`) VALUES ('Alpha', 'alpha@gmail.com', 2);
INSERT INTO faculty (`Name`, `email_id`, `domain_id`) VALUES ('Sheela Devi', 'sheela@gmail.com', 4);
INSERT INTO faculty (`Name`, `email_id`, `domain_id`) VALUES ('Sudeepa Roy Dey', 'sudeepa@gmail.com', 2);
INSERT INTO faculty (`Name`, `email_id`, `domain_id`) VALUES ('Geeta Dayalan', 'geeta_dayalan@gmail.com', 6);
INSERT INTO faculty (`Name`, `email_id`, `domain_id`) VALUES ('Arti Arya', 'arti@gmail.com', 2);
INSERT INTO faculty (`Name`, `email_id`, `domain_id`) VALUES ('Sandesh', 'sandesh@gmail.com', 2);
INSERT INTO faculty (`Name`, `email_id`, `domain_id`) VALUES ('Animesh Giri', 'animesh@gmail.com', 3);
```



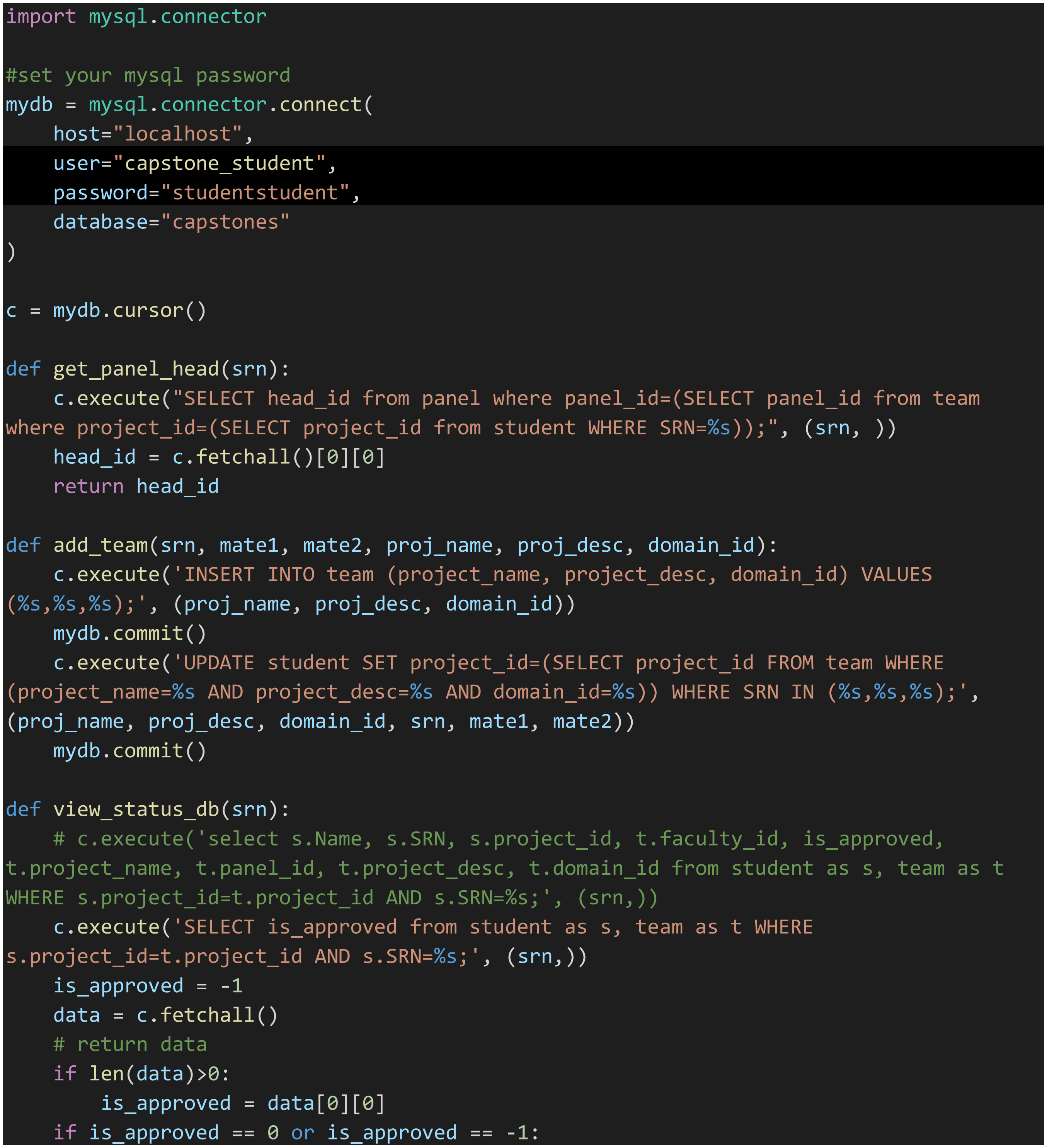
```sql
-- Insertion in Students
INSERT INTO student (`Name`, `SRN`, `Sem`, `Gender`, `CGPA`, `Email`) VALUES ('Sample', 'pes2ug20cs000', 6, 'F', 9.99, 'sample@gmail.com');
INSERT INTO student (`Name`, `SRN`, `Sem`, `Gender`, `CGPA`, `Email`) VALUES ('Sample2', 'pes2ug20cs001', 6, 'F', 9.99, 'sample2@gmail.com');
INSERT INTO student (`Name`, `SRN`, `Sem`, `Gender`, `CGPA`, `Email`) VALUES ('Darsh', 'pes2ug21cs150', 5, 'M', 9.18, 'darshpatel@gmail.com');
INSERT INTO student (`Name`, `SRN`, `Sem`, `Gender`, `CGPA`, `Email`) VALUES ('Gautam', 'pes2ug21cs176', 5, 'M', 9, 'gautam@gmail.com');
INSERT INTO student (`Name`, `SRN`, `Sem`, `Gender`, `CGPA`, `Email`) VALUES ('Yashas', 'pes2ug21cs922', 5, 'M', 8.5, 'yashas@gmail.com');
INSERT INTO student (`Name`, `SRN`, `Sem`, `Gender`, `CGPA`, `Email`) VALUES ('Dhyey', 'pes2ug21cs164', 5, 'M', 7.9, 'dhyey@gmail.com');
INSERT INTO student (`Name`, `SRN`, `Sem`, `Gender`, `CGPA`, `Email`) VALUES ('Tirth', 'pes2ug21cs118', 5, 'M', 8.5, 'tirth@gmail.com');
INSERT INTO student (`Name`, `SRN`, `Sem`, `Gender`, `CGPA`, `Email`) VALUES ('Faizan', 'pes2ug21cs169', 5, 'M', 9.5, 'faizan@gmail.com');
INSERT INTO student (`Name`, `SRN`, `Sem`, `Gender`, `CGPA`, `Email`) VALUES ('Vansheel', 'pes2ug21cs909', 5, 'M', 8.57, 'vansheel@gmail.com')
INSERT INTO student (`Name`, `SRN`, `Sem`, `Gender`, `CGPA`, `Email`) VALUES ('Krisha', 'pes2ug21cs239', 5, 'M', 9.2, 'krisha@gmail.com');
```



```sql
-- Sample Panel formation
INSERT INTO panel (`domain_id`, `head_id`) VALUES (1, 1);
UPDATE faculty SET panel_id=1 WHERE ID in (1, 2);
-- Sample student team formation
INSERT INTO team (`project_name`, `project_desc`, `domain_id`) VALUES ('sample', 'sample sample', 1);
UPDATE student SET project_id = (SELECT COUNT(*) FROM team) WHERE SRN IN ('pes2ug20cs000', 'pes2ug20cs001');
-- Sample team approval
UPDATE team SET is_approved=1, faculty_id=1, panel_id=1 WHERE project_id=1;
-- Sample Evaluation
INSERT IGNORE INTO evaluation (`Semester`, `SRN`, `Faculty_ID`)
VALUES (
    (SELECT Sem from student where SRN='pes2ug20cs000'),
    'pes2ug20cs000',
    (SELECT faculty_id from team
    where project_id=(SELECT project_id from student where SRN='pes2ug20cs000'))
);
UPDATE evaluation SET ISA1=20, ISA2=20, ISA3=20, ESA=80 WHERE SRN='pes2ug20cs000';
```

7. QUERIES

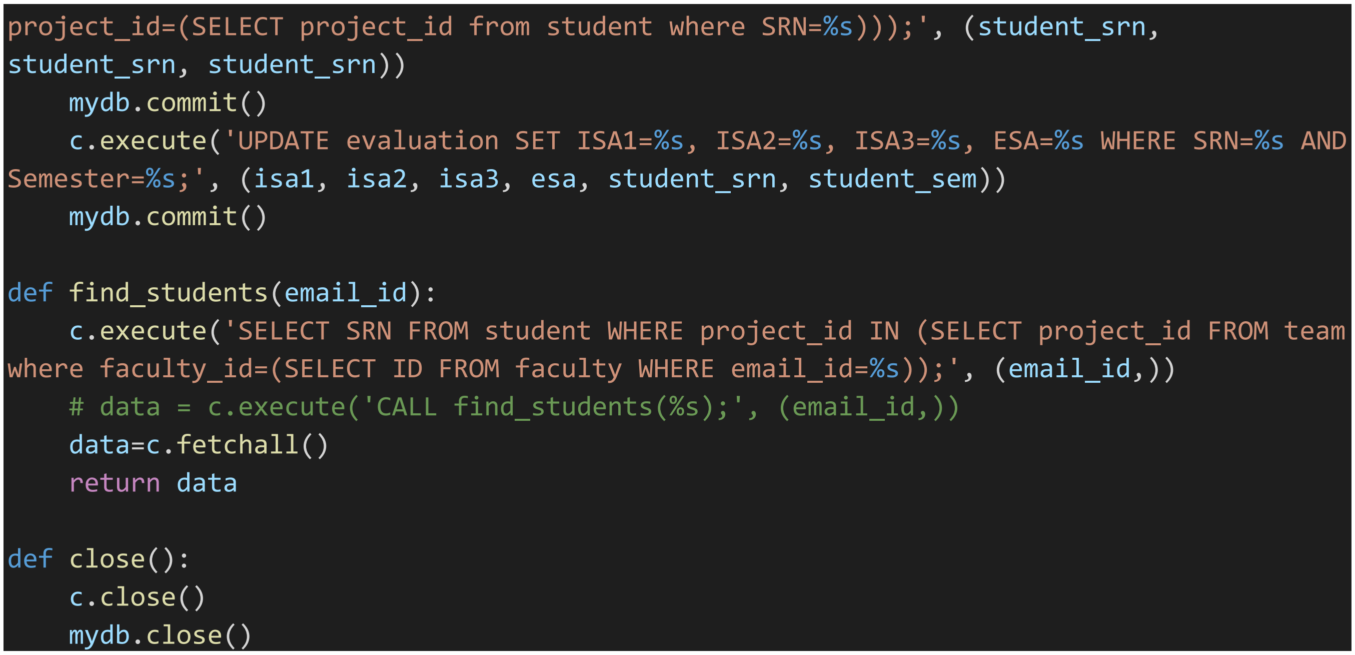
Student Database Queries :



```
import mysql.connector

#set your mysql password
mydb = mysql.connector.connect(
    host="localhost",
    user="capstone_student",
    password="studentstudent",
    database="capstones"
)

c = mydb.cursor()

def get_panel_head(srn):
    c.execute("SELECT head_id from panel where panel_id=(SELECT panel_id from team
where project_id=(SELECT project_id from student WHERE SRN=%s));", (srn, ))
    head_id = c.fetchall()[0][0]
    return head_id

def add_team(srn, mate1, mate2, proj_name, proj_desc, domain_id):
    c.execute('INSERT INTO team (project_name, project_desc, domain_id) VALUES
(%s,%s,%s);', (proj_name, proj_desc, domain_id))
    mydb.commit()
    c.execute('UPDATE student SET project_id=(SELECT project_id FROM team WHERE
(project_name=%s AND project_desc=%s AND domain_id=%s)) WHERE SRN IN (%s,%s,%s);',
(proj_name, proj_desc, domain_id, srn, mate1, mate2))
    mydb.commit()

def view_status_db(srn):
    # c.execute('select s.Name, s.SRN, s.project_id, t.faculty_id, is_approved,
t.project_name, t.panel_id, t.project_desc, t.domain_id from student as s, team as t
WHERE s.project_id=t.project_id AND s.SRN=%s;', (srn,))
    c.execute('SELECT is_approved from student as s, team as t WHERE
s.project_id=t.project_id AND s.SRN=%s;', (srn,))
    is_approved = -1
    data = c.fetchall()
    # return data
    if len(data)>0:
        is_approved = data[0][0]
    if is_approved == 0 or is_approved == -1:
        c.execute('select s.Name, s.SRN, s.project_id, t.faculty_id, is_approved,
t.project_name, t.panel_id, t.project_desc, t.domain_id from student as s, team as t
WHERE s.project_id=t.project_id AND s.SRN=%s;', (srn,))
        ans = c.fetchall()
        return ans
    elif is_approved == 1:
        c.execute('select s.Name, s.SRN, s.project_id, f.name, is_approved,
t.project_name, t.panel_id, t.project_desc, t.domain_id from student as s, team as t,
faculty as f WHERE s.project_id=t.project_id AND t.faculty_id=f.ID AND s.SRN=%s;',
(srn,))
        ans = c.fetchall()
        return ans


def view_domains_db():
    # c.execute('CALL get_all_domains();')
    c.execute('SELECT * FROM domain ORDER BY DomainID ASC;')
    data = c.fetchall()
    return data

def view_evaluation_db(srn):
    c.execute('INSERT IGNORE INTO evaluation (`Semester`, `SRN`, `Faculty_ID`) VALUES
((SELECT Sem from student where SRN=%s),%s, (SELECT faculty_id from team where
project_id=(SELECT project_id from student where SRN=%s)));', (srn, srn, srn))
    mydb.commit()
    c.execute('SELECT * FROM evaluation WHERE SRN=%s;', (srn,))
    data = c.fetchall()
    return data

def view_panel_db(srn):
    c.execute('SELECT ID, name, email_id, d.`Domain Name` FROM faculty as f, domain as
d WHERE panel_id=(SELECT panel_id FROM team where project_id=(SELECT project_id FROM
student WHERE SRN=%s)) AND d.DomainID=f.domain_id;', (srn,))
    data = c.fetchall()
    return data

def close():
    c.close()
    mydb.close()
```

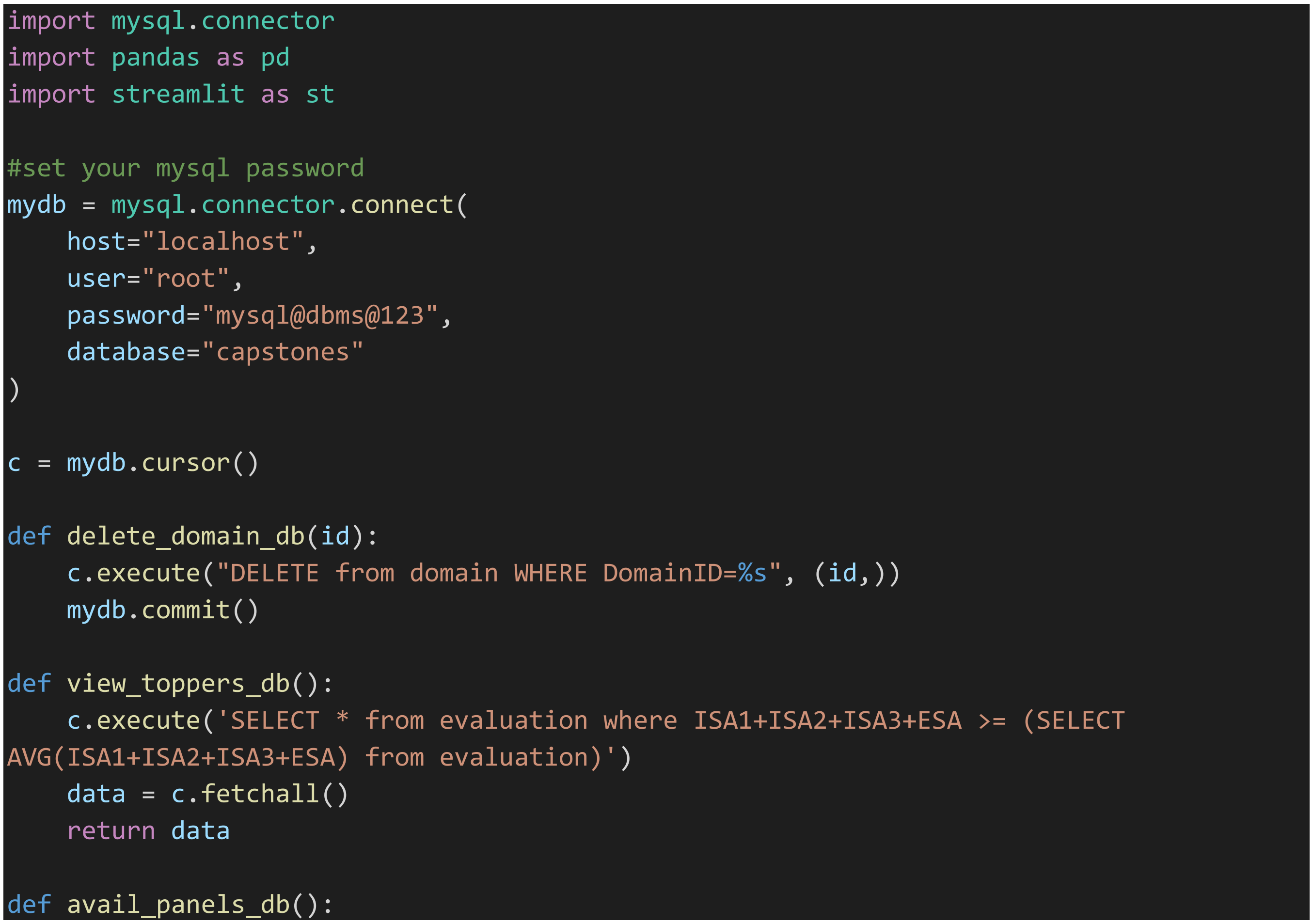
Faculty Database Queries:

```
import mysql.connector

#set your mysql password
mydb = mysql.connector.connect(
    host="localhost",
    user="capstone_faculty",
    password="facultyfaculty",
    database="capstones"
)

c = mydb.cursor()

def your_students_db(email):
    c.execute("SELECT * FROM student where project_id IN (SELECT project_id FROM team WHERE faculty_id=(SELECT ID from faculty where email_id=%s))", (email,))
    data = c.fetchall()
    return data

def view_teams_db(email_id):
    c.execute('SELECT t.project_id, t.project_name, t.project_desc, d.`Domain Name` FROM team as t, domain as d WHERE t.panel_id=(SELECT panel_id FROM faculty where email_id=%s) AND t.domain_id=d.DomainID', (email_id,))
    data = c.fetchall()
    return data

def view_panel(email):
    c.execute('SELECT ID, name, email_id, domain_id from faculty WHERE panel_id=(SELECT panel_id FROM faculty WHERE email_id=%s);', (email,))
    data = c.fetchall()
    return data

def get_panel_head(email):
    c.execute('SELECT head_id from panel where panel_id=(SELECT panel_id FROM faculty WHERE email_id=%s);', (email,))
    data = c.fetchall()[0][0]
    return data

def see_eval(student_srn, student_sem):
    c.execute('INSERT IGNORE INTO evaluation (`Semester`, `SRN`, `Faculty_ID`) VALUES ((SELECT Sem from student where SRN=%s),%s, (SELECT faculty_id from team where project_id=(SELECT project_id from student where SRN=%s)));', (student_srn, student_srn, student_srn))
    mydb.commit()
    c.execute('SELECT * FROM evaluation where SRN=%s AND Semester=%s', (student_srn, student_sem))
    data=c.fetchall()
    return data

def eval_db(isa1, isa2, isa3, esa, student_srn, student_sem):
    c.execute('INSERT IGNORE INTO evaluation (`Semester`, `SRN`, `Faculty_ID`) VALUES ((SELECT Sem from student where SRN=%s),%s, (SELECT faculty_id from team where
```

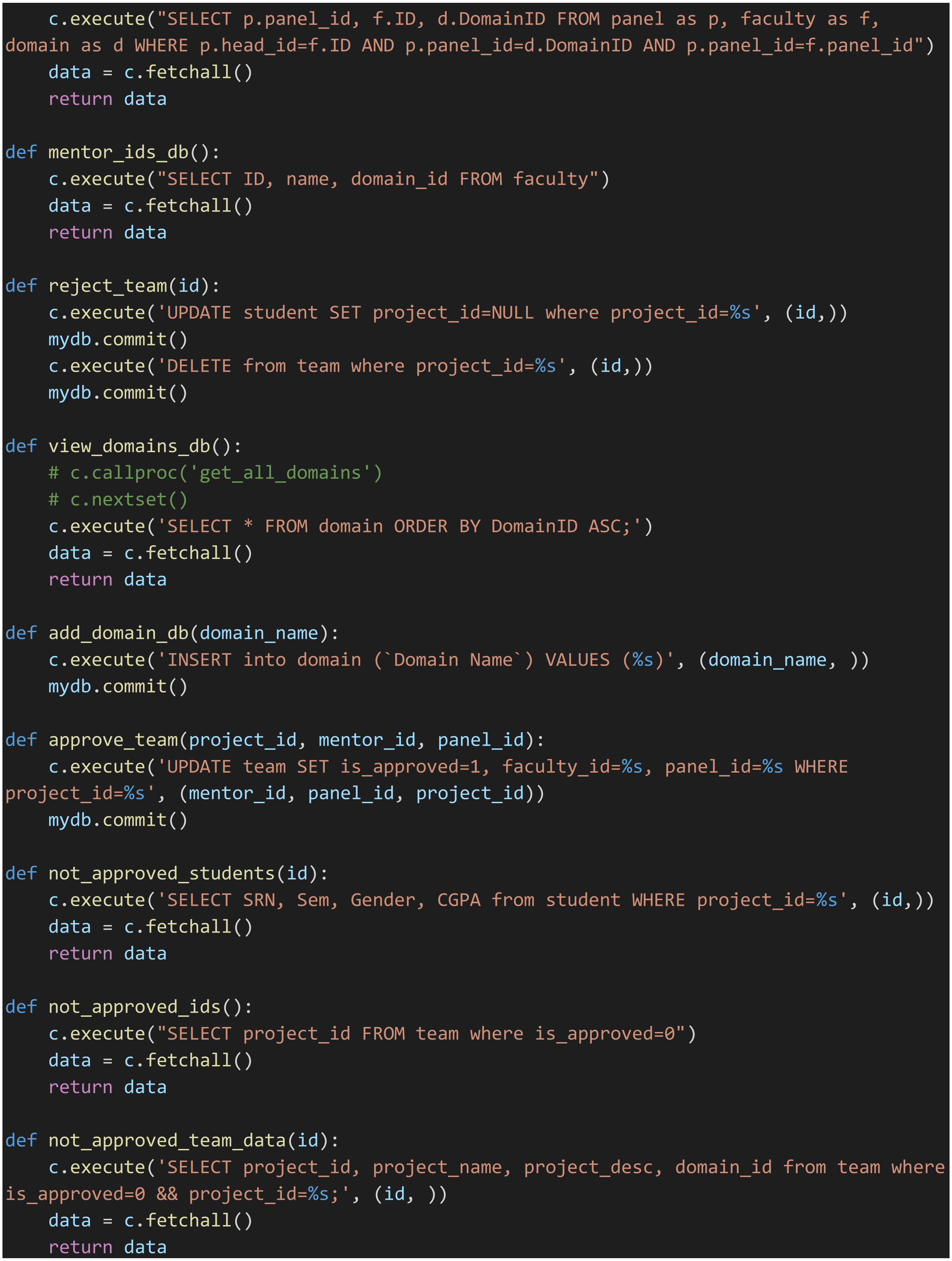


```
project_id=(SELECT project_id from student where SRN=%s)));', (student_srn,
student_srn, student_srn))
    mydb.commit()
    c.execute('UPDATE evaluation SET ISA1=%s, ISA2=%s, ISA3=%s, ESA=%s WHERE SRN=%s AND
Semester=%s;', (isa1, isa2, isa3, esa, student_srn, student_sem))
    mydb.commit()

def find_students(email_id):
    c.execute('SELECT SRN FROM student WHERE project_id IN (SELECT project_id FROM team
where faculty_id=(SELECT ID FROM faculty WHERE email_id=%s));', (email_id,))
    # data = c.execute('CALL find_students(%s);', (email_id,))
    data=c.fetchall()
    return data

def close():
    c.close()
    mydb.close()
```

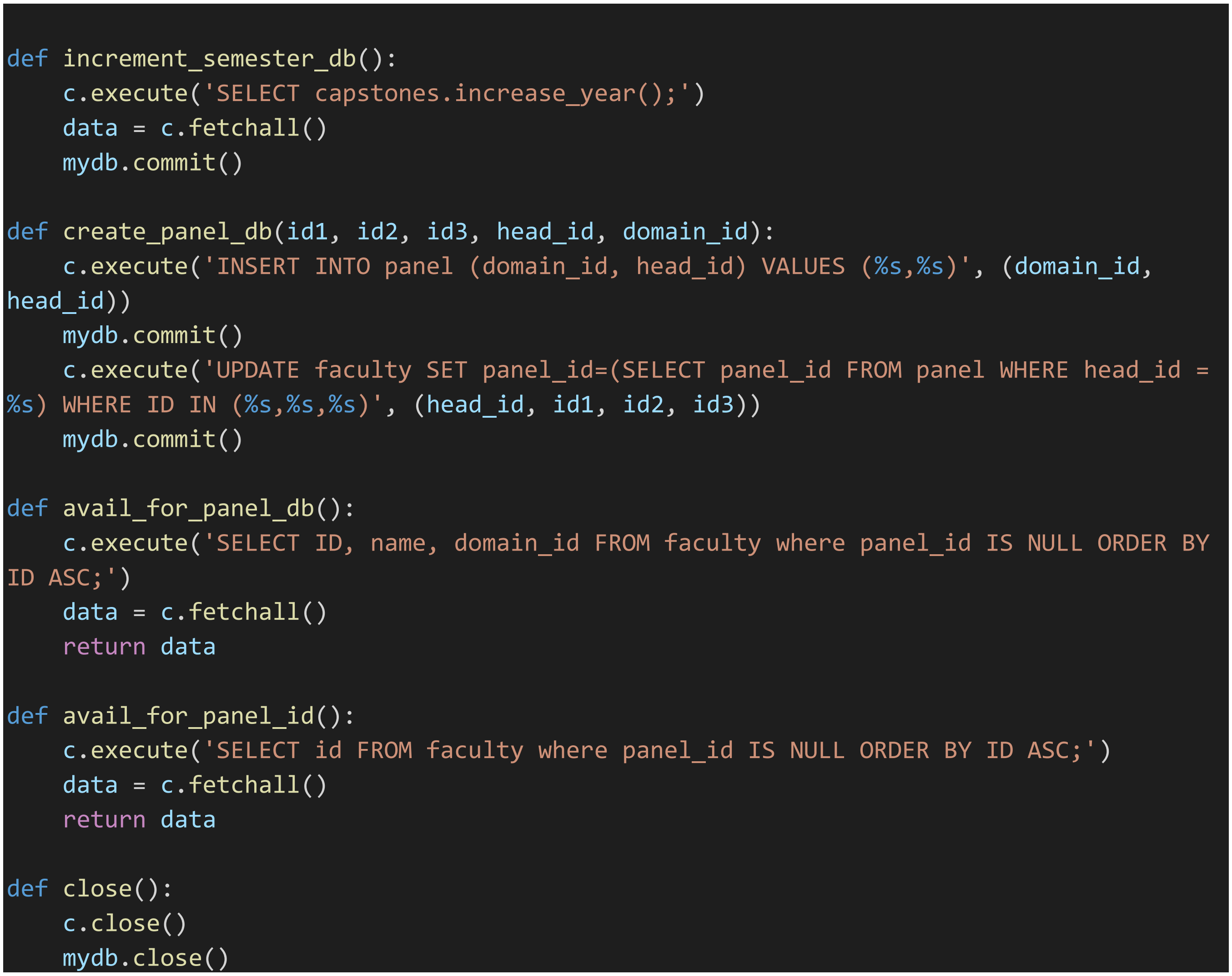
Admin Database Queries:



```
import mysql.connector
import pandas as pd
import streamlit as st

#set your mysql password
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="mysql@dbms@123",
    database="capstones"
)

c = mydb.cursor()

def delete_domain_db(id):
    c.execute("DELETE from domain WHERE DomainID=%s", (id,))
    mydb.commit()

def view_toppers_db():
    c.execute('SELECT * from evaluation where ISA1+ISA2+ISA3+ESA >= (SELECT
AVG(ISA1+ISA2+ISA3+ESA) from evaluation)')
    data = c.fetchall()
    return data

def avail_panels_db():
```

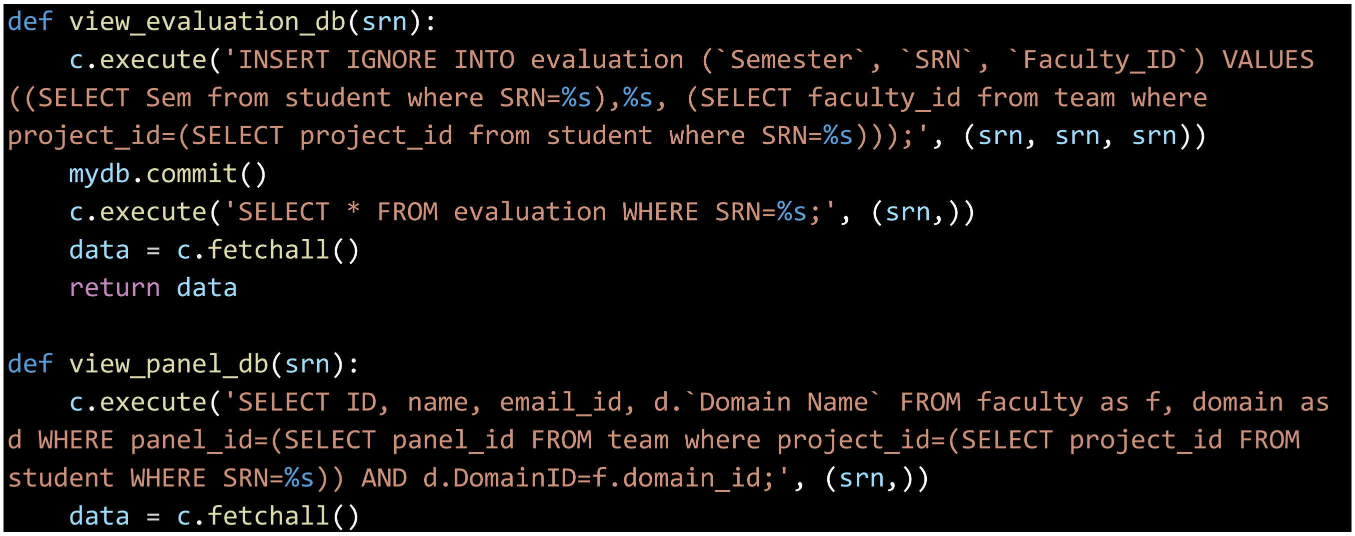


```
    c.execute("SELECT p.panel_id, f.ID, d.DomainID FROM panel as p, faculty as f,
domain as d WHERE p.head_id=f.ID AND p.panel_id=d.DomainID AND p.panel_id=f.panel_id")
    data = c.fetchall()
    return data

def mentor_ids_db():
    c.execute("SELECT ID, name, domain_id FROM faculty")
    data = c.fetchall()
    return data

def reject_team(id):
    c.execute('UPDATE student SET project_id=NULL where project_id=%s', (id,))
    mydb.commit()
    c.execute('DELETE from team where project_id=%s', (id,))
    mydb.commit()

def view_domains_db():
    # c.callproc('get_all_domains')
    # c.nextset()
    c.execute('SELECT * FROM domain ORDER BY DomainID ASC;')
    data = c.fetchall()
    return data

def add_domain_db(domain_name):
    c.execute('INSERT into domain (`Domain Name`) VALUES (%s)', (domain_name, ))
    mydb.commit()

def approve_team(project_id, mentor_id, panel_id):
    c.execute('UPDATE team SET is_approved=1, faculty_id=%s, panel_id=%s WHERE
project_id=%s', (mentor_id, panel_id, project_id))
    mydb.commit()

def not_approved_students(id):
    c.execute('SELECT SRN, Sem, Gender, CGPA from student WHERE project_id=%s', (id,))
    data = c.fetchall()
    return data

def not_approved_ids():
    c.execute("SELECT project_id FROM team where is_approved=0")
    data = c.fetchall()
    return data

def not_approved_team_data(id):
    c.execute('SELECT project_id, project_name, project_desc, domain_id from team where
is_approved=0 && project_id=%s;', (id, ))
    data = c.fetchall()
    return data
```



```
def increment_semester_db():
    c.execute('SELECT capstones.increase_year();')
    data = c.fetchall()
    mydb.commit()

def create_panel_db(id1, id2, id3, head_id, domain_id):
    c.execute('INSERT INTO panel (domain_id, head_id) VALUES (%s,%s)', (domain_id,
head_id))
    mydb.commit()
    c.execute('UPDATE faculty SET panel_id=(SELECT panel_id FROM panel WHERE head_id =
%s) WHERE ID IN (%s,%s,%s)', (head_id, id1, id2, id3))
    mydb.commit()

def avail_for_panel_db():
    c.execute('SELECT ID, name, domain_id FROM faculty where panel_id IS NULL ORDER BY
ID ASC;')
    data = c.fetchall()
    return data

def avail_for_panel_id():
    c.execute('SELECT id FROM faculty where panel_id IS NULL ORDER BY ID ASC;')
    data = c.fetchall()
    return data

def close():
    c.close()
    mydb.close()
```

Nested Queries:



```
def view_evaluation_db(srn):
    c.execute('INSERT IGNORE INTO evaluation (`Semester`, `SRN`, `Faculty_ID`) VALUES
((SELECT Sem from student where SRN=%s),%s, (SELECT faculty_id from team where
project_id=(SELECT project_id from student where SRN=%s)));', (srn, srn, srn))
    mydb.commit()
    c.execute('SELECT * FROM evaluation WHERE SRN=%s;', (srn,))
    data = c.fetchall()
    return data

def view_panel_db(srn):
    c.execute('SELECT ID, name, email_id, d.`Domain Name` FROM faculty as f, domain as
d WHERE panel_id=(SELECT panel_id FROM team where project_id=(SELECT project_id FROM
student WHERE SRN=%s)) AND d.DomainID=f.domain_id;', (srn,))
    data = c.fetchall()
    return data
```

```
def get_panel_head(srn):
    c.execute("SELECT head_id from panel where panel_id=(SELECT panel_id from team where project_id=(SELECT project_id from student WHERE SRN=%s));", (srn, ))
    head_id = c.fetchall()[0][0]
    return head_id
```

```
def your_students_db(email):
    c.execute("SELECT * FROM student where project_id IN (SELECT project_id FROM team WHERE faculty_id=(SELECT ID from faculty where email_id=%s))", (email,))
    data = c.fetchall()
    return data

def view_teams_db(email_id):
    c.execute('SELECT t.project_id, t.project_name, t.project_desc, d.`Domain Name` FROM team as t, domain as d WHERE t.panel_id=(SELECT panel_id FROM faculty where email_id=%s) AND t.domain_id=d.DomainID', (email_id,))
    data = c.fetchall()
    return data

def view_panel(email):
    c.execute('SELECT ID, name, email_id, domain_id from faculty WHERE panel_id=(SELECT panel_id FROM faculty WHERE email_id=%s);', (email,))
    data = c.fetchall()
    return data
def get_panel_head(email):
    c.execute('SELECT head_id from panel where panel_id=(SELECT panel_id FROM faculty WHERE email_id=%s);', (email,))
    data = c.fetchall()[0][0]
    return data

def see_eval(student_srn, student_sem):
    c.execute('INSERT IGNORE INTO evaluation (`Semester`, `SRN`, `Faculty_ID`) VALUES ((SELECT Sem from student where SRN=%s),%s, (SELECT faculty_id from team where project_id=(SELECT project_id from student where SRN=%s)));', (student_srn, student_srn, student_srn))
    mydb.commit()
```

```
def eval_db(isa1, isa2, isa3, esa, student_srn, student_sem):
    c.execute('INSERT IGNORE INTO evaluation (`Semester`, `SRN`, `Faculty_ID`) VALUES ((SELECT Sem from student where SRN=%s),%s, (SELECT faculty_id from team where project_id=(SELECT project_id from student where SRN=%s)));', (student_srn, student_srn, student_srn))
    mydb.commit()
```

```
def find_students(email_id):
    c.execute('SELECT SRN FROM student WHERE project_id IN (SELECT project_id FROM team
where faculty_id=(SELECT ID FROM faculty WHERE email_id=%s));', (email_id,))
    # data = c.execute('CALL find_students(%s);', (email_id,))
    data=c.fetchall()
    return data
```

Correlated Queries:

```
def view_toppers_db():
    c.execute('SELECT * from evaluation where ISA1+ISA2+ISA3+ESA >= (SELECT
AVG(ISA1+ISA2+ISA3+ESA) from evaluation)')
    data = c.fetchall()
    return data
```

8. STORED PROCEDURES, FUCNTIONS AND TRIGGERS

```
-- Stored Procedure to show all domains:
DELIMITER //
CREATE PROCEDURE get_all_domains()
BEGIN
   SELECT * FROM domain ORDER BY DomainID ASC;
END //
DELIMITER ;

-- Stored Procedure to show all students under a faculty
DELIMITER //
CREATE PROCEDURE find_students(in email varchar(45))
BEGIN
   SELECT SRN FROM student
   WHERE project_id IN (SELECT project_id FROM team where faculty_id=(SELECT ID FROM
faculty WHERE email_id=email));
END //
DELIMITER ;

-- Function to increase year by 1
SET SQL_SAFE_UPDATES = 0;
DELIMITER //
CREATE FUNCTION increase_year()
RETURNS INT
DETERMINISTIC
BEGIN
   DECLARE rows_affected INT;
   -- Increase the Sem of every student by 1
   UPDATE student
   SET Sem = Sem + 1
   WHERE Sem <= 8; -- Update only if Sem is less than or equal to 8
   -- Remove entries with Sem greater than 8
   DELETE FROM student
   WHERE Sem > 8;
   -- Get the total number of rows affected
   SELECT ROW_COUNT() INTO rows_affected;
   RETURN rows_affected;
```

```
END //
DELIMITER ;

-- Trigger to add Domain data into domain_backup before deleting:
DELIMITER //
CREATE TRIGGER before_delete_domain
BEFORE DELETE
ON capstones.domain FOR EACH ROW
BEGIN
   -- Insert deleted data into domain_backup table
   INSERT INTO domain_backup (DomainID, `Domain Name`)
   VALUES (OLD.DomainID, OLD.`Domain Name`);
END;
//
DELIMITER ;
```

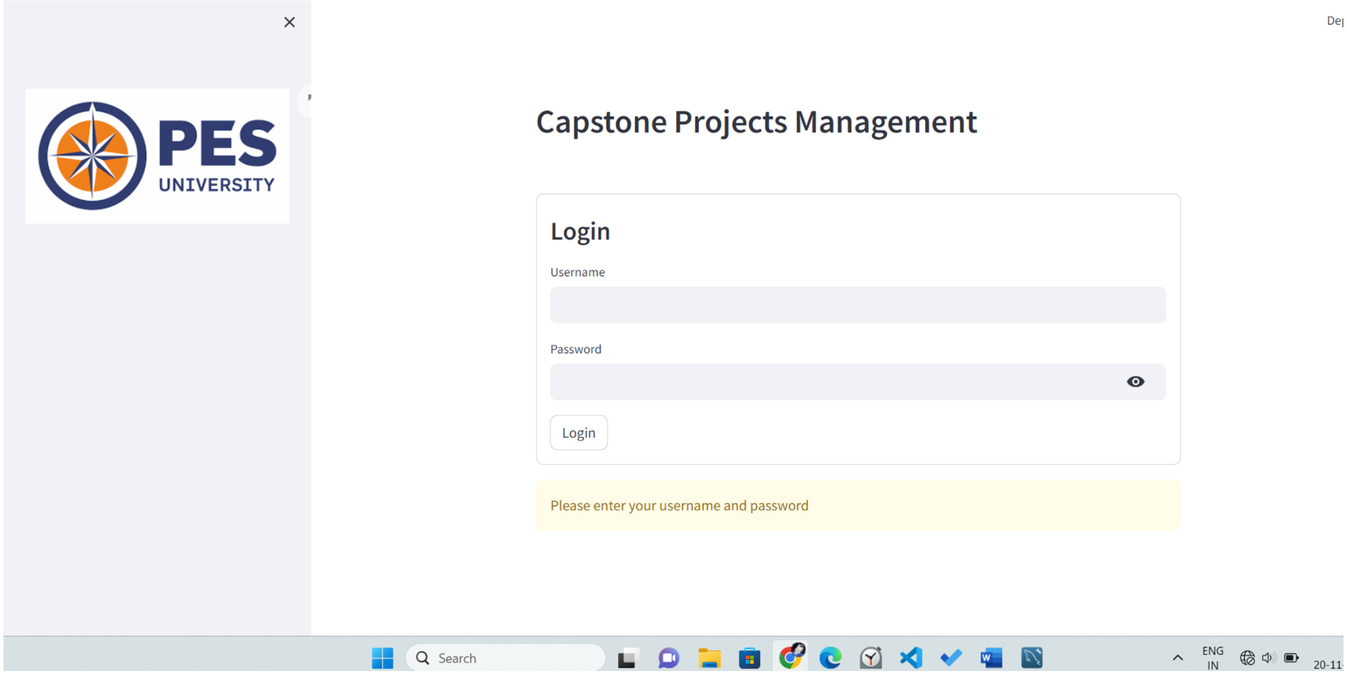
9. Database Users

```sql
-- Create a new user for the student
CREATE USER 'capstone_student'@'%' IDENTIFIED BY 'studentstudent';
-- Grant permissions on the necessary tables and columns
GRANT SELECT ON capstones.student TO 'capstone_student'@'%';
GRANT SELECT, INSERT ON capstones.team TO 'capstone_student'@'%';
GRANT SELECT ON capstones.domain TO 'capstone_student'@'%';
GRANT SELECT ON capstones.faculty TO 'capstone_student'@'%';
GRANT SELECT ON capstones.panel TO 'capstone_student'@'%';
GRANT SELECT, INSERT, UPDATE ON capstones.evaluation TO 'capstone_student'@'%';
-- If there are additional columns used in the queries, grant permissions on those columns
GRANT SELECT (head_id) ON capstones.panel TO 'capstone_student'@'%';
GRANT SELECT, UPDATE ON (project_id) ON capstones.student TO 'capstone_student'@'%';
GRANT SELECT, INSERT (project_name, project_desc, domain_id) ON capstones.team TO 'capstone_student'@'%';
GRANT SELECT, UPDATE (faculty_id) ON capstones.team TO 'capstone_student'@'%';
GRANT SELECT (is_approved) ON capstones.student TO 'capstone_student'@'%';
GRANT SELECT (ID, name, email_id) ON capstones.faculty TO 'capstone_student'@'%';
GRANT SELECT (DomainID, `Domain Name`) ON capstones.domain TO 'capstone_student'@'%';
GRANT SELECT, INSERT (`Semester`, `SRN`, `Faculty_ID`) ON capstones.evaluation TO 'capstone_student'@'%';
-- Flush privileges to apply the changes
FLUSH PRIVILEGES;



-- Create a new user for the faculty
CREATE USER 'capstone_faculty'@'%' IDENTIFIED BY 'facultyfaculty';
-- Grant permissions on the necessary tables and columns
GRANT SELECT ON capstones.student TO 'capstone_faculty'@'%';
GRANT SELECT ON capstones.team TO 'capstone_faculty'@'%';
GRANT SELECT ON capstones.domain TO 'capstone_faculty'@'%';
GRANT SELECT ON capstones.faculty TO 'capstone_faculty'@'%';
GRANT SELECT ON capstones.panel TO 'capstone_faculty'@'%';
GRANT SELECT, INSERT, UPDATE ON capstones.evaluation TO 'capstone_faculty'@'%';
-- If there are additional columns used in the queries, grant permissions on those columns
GRANT SELECT (ID, name, email_id) ON capstones.faculty TO 'capstone_faculty'@'%';
GRANT SELECT (ID, name, email_id, domain_id) ON capstones.panel TO 'capstone_faculty'@'%';
GRANT SELECT (project_id, project_name, project_desc, domain_id) ON capstones.team TO 'capstone_faculty'@'%';
GRANT SELECT, UPDATE (project_id) ON capstones.student TO 'capstone_faculty'@'%';
GRANT SELECT (DomainID, `Domain Name`) ON capstones.domain TO 'capstone_faculty'@'%';
GRANT SELECT, INSERT (`Semester`, `SRN`, `Faculty_ID`) ON capstones.evaluation TO 'capstone_faculty'@'%';
-- Flush privileges to apply the changes
FLUSH PRIVILEGES;
```

Admin will use the permission of ‘root’ for its functionalities.

11. FRONT END DEVELOPEMNT

Login:



Capstone Projects Management

Login

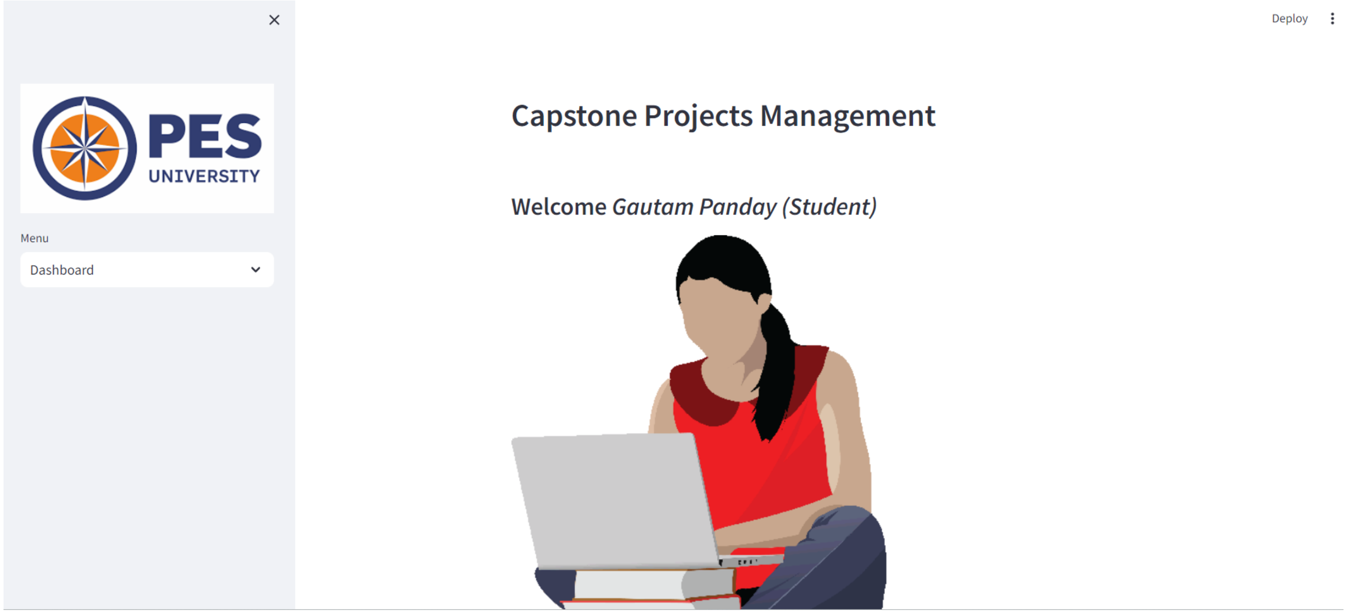
Username

Password

Login

Please enter your username and password

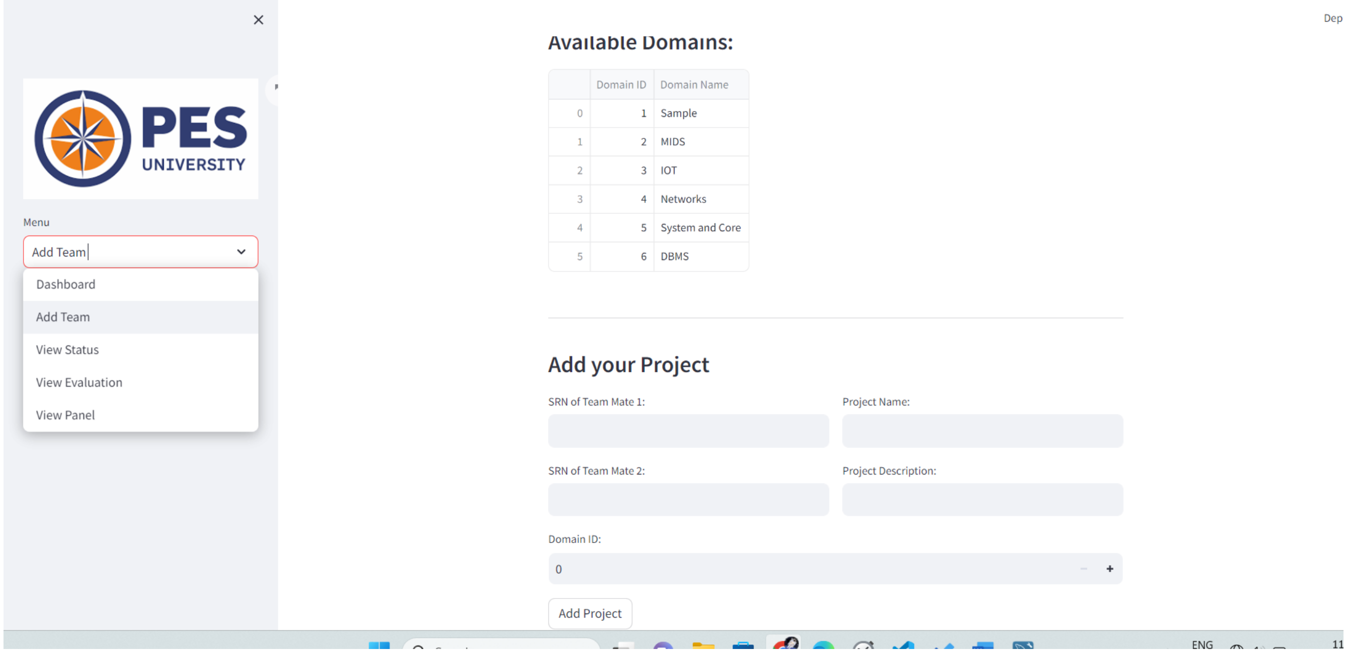
Student:



Capstone Projects Management

Welcome *Gautam Panday (Student)*

Menu: Dashboard



Menu: Add Team

Available Domains:

	Domain ID	Domain Name
0	1	Sample
1	2	MIDS
2	3	IOT
3	4	Networks
4	5	System and Core
5	6	DBMS

Add your Project

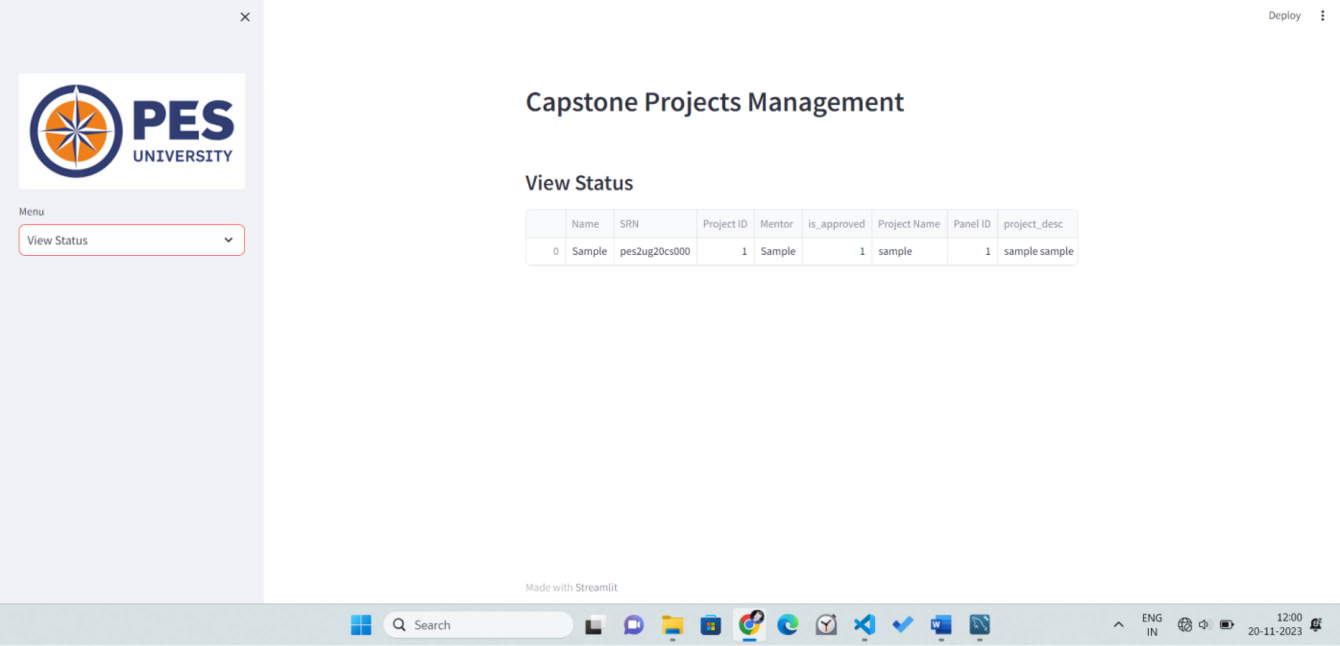
SRN of Team Mate 1:

Project Name:

SRN of Team Mate 2:

Project Description:

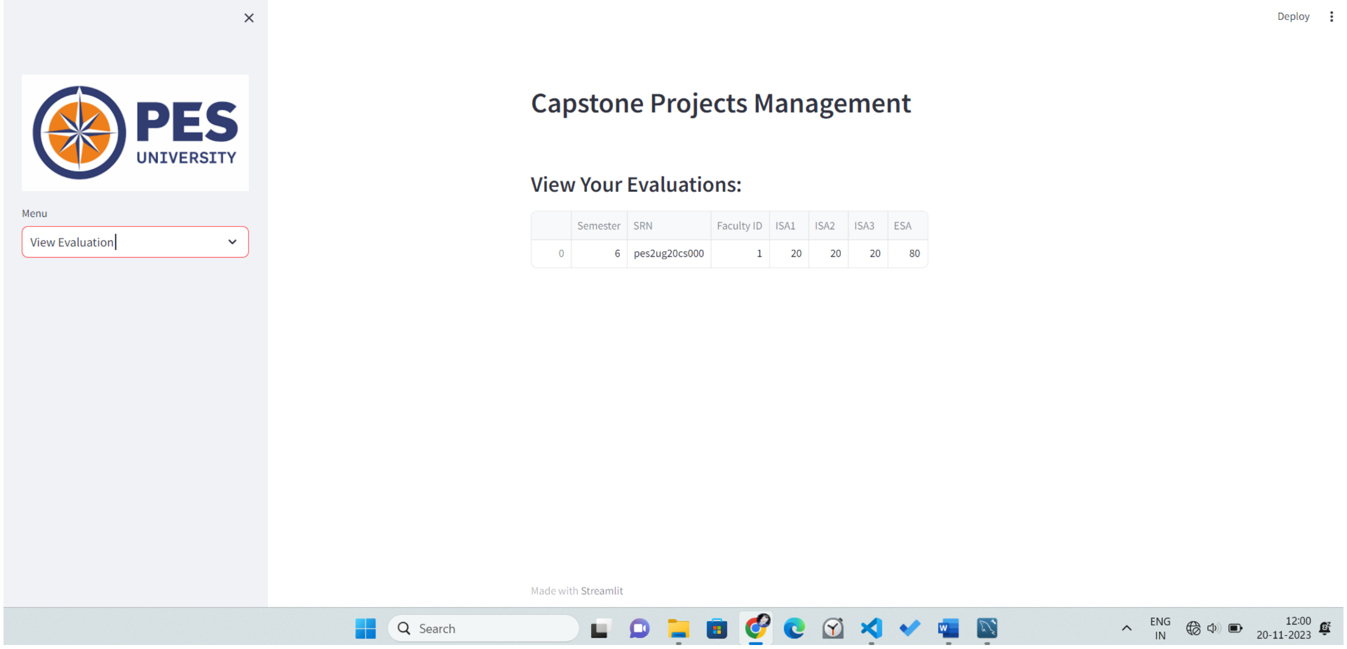
Domain ID: 0



Capstone Projects Management

View Status

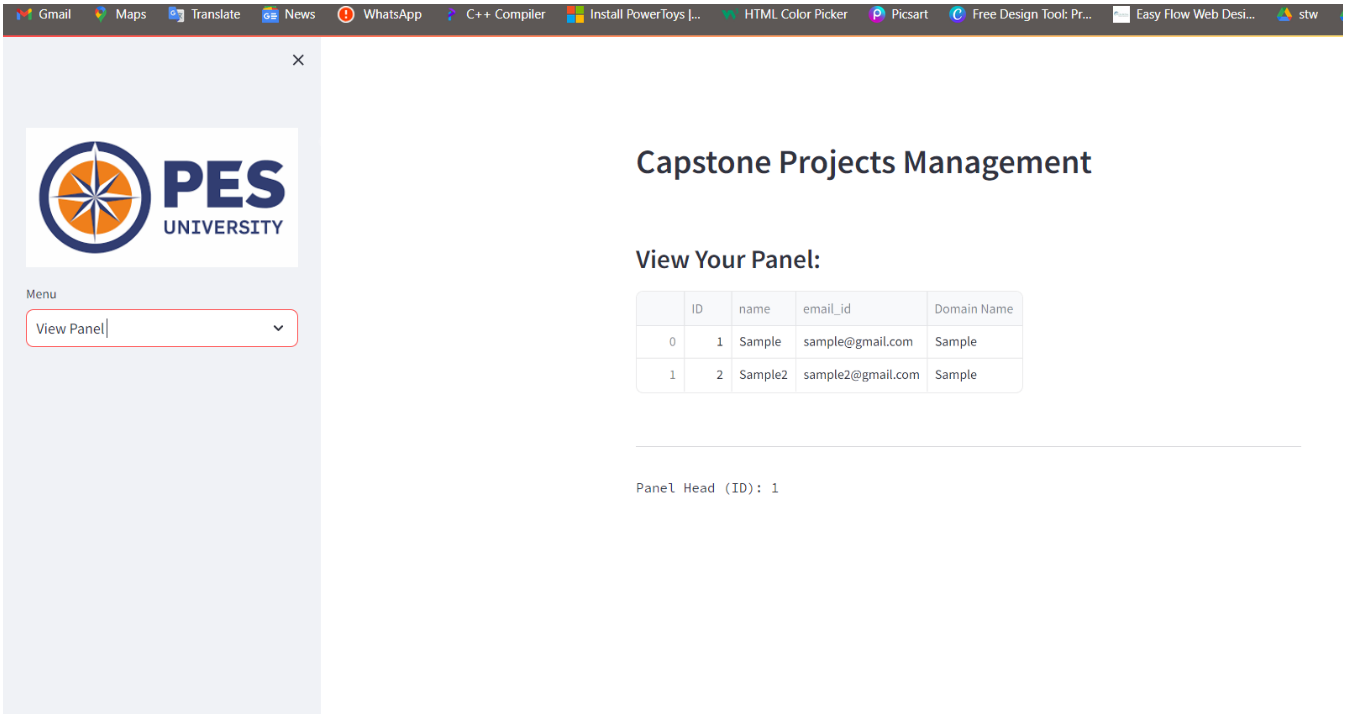
	Name	SRN	Project ID	Mentor	is_approved	Project Name	Panel ID	project_desc
0	Sample	pes2ug20cs000	1	Sample	1	sample	1	sample sample



Capstone Projects Management

View Your Evaluations:

	Semester	SRN	Faculty ID	ISA1	ISA2	ISA3	ESA
0	6	pes2ug20cs000	1	20	20	20	80



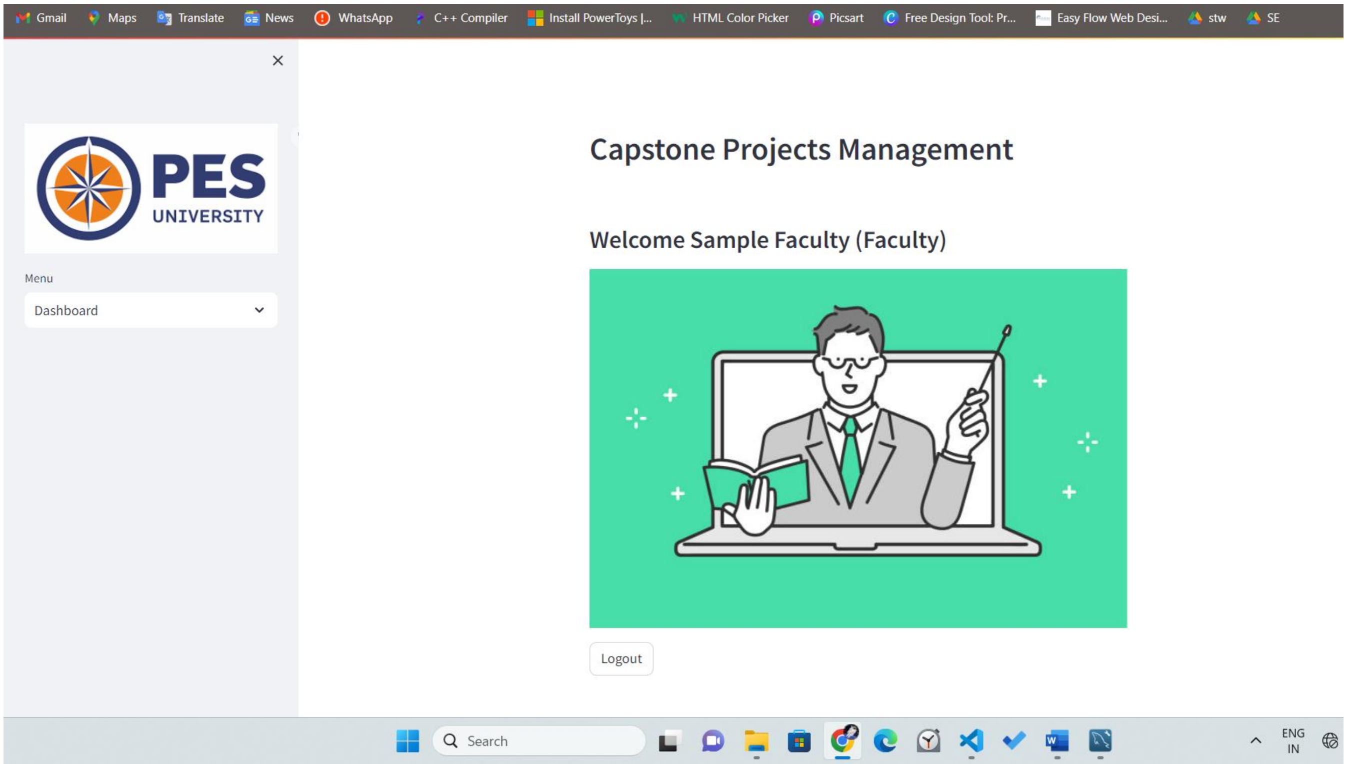
Capstone Projects Management

View Your Panel:

	ID	name	email_id	Domain Name
0	1	Sample	sample@gmail.com	Sample
1	2	Sample2	sample2@gmail.com	Sample

```
Panel Head (ID): 1
```

Faculty:

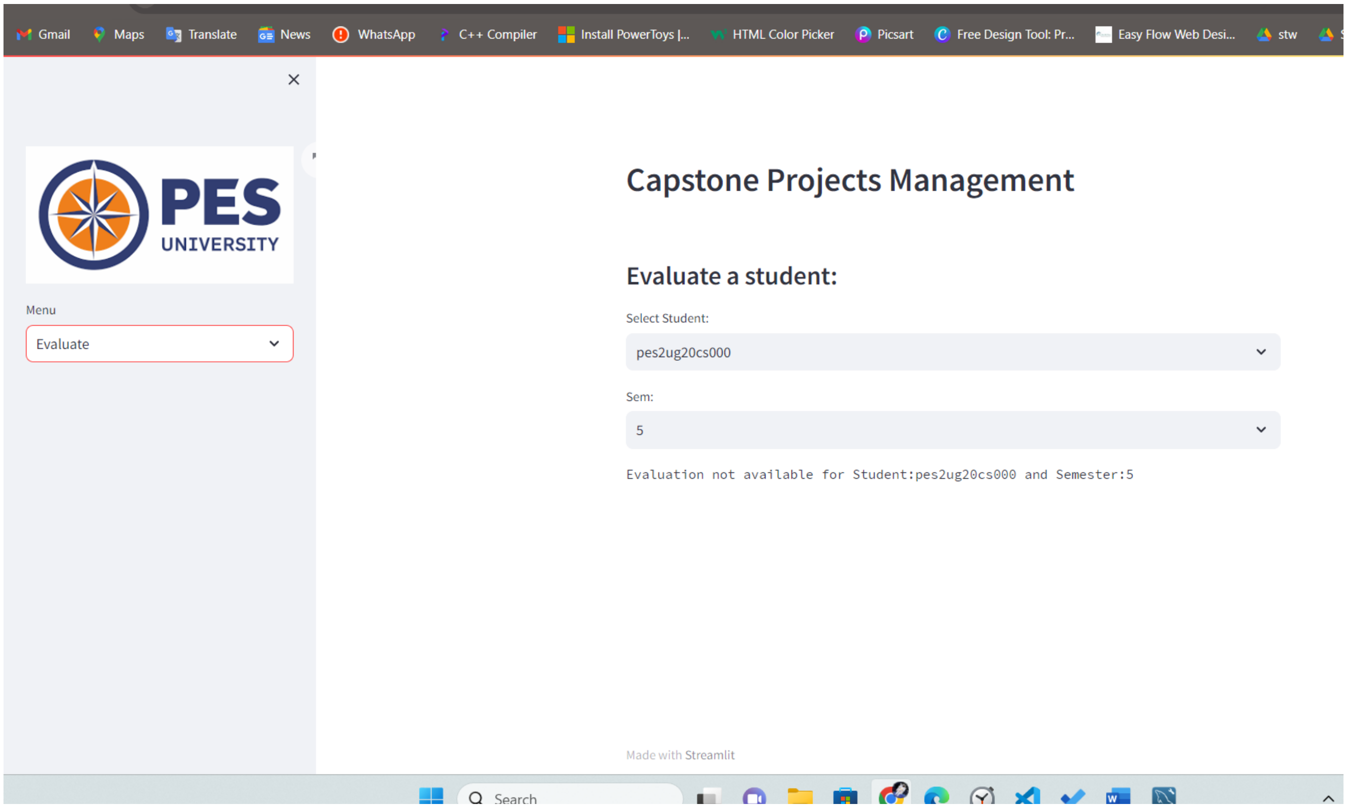


Capstone Projects Management

Welcome Sample Faculty (Faculty)

Menu

Dashboard



Capstone Projects Management

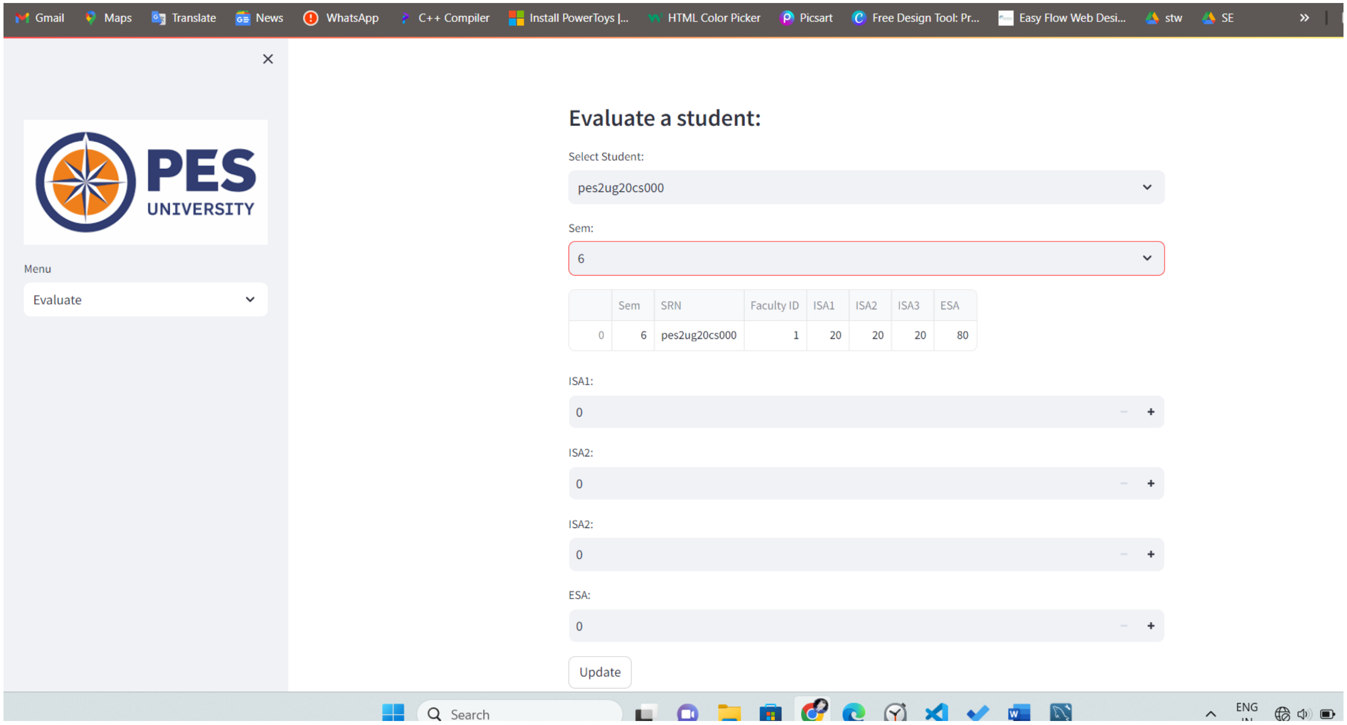
Evaluate a student:

Select Student: pes2ug20cs000

Sem: 5

```
Evaluation not available for Student:pes2ug20cs000 and Semester:5
```

Made with Streamlit



Evaluate a student:

Select Student: pes2ug20cs000

Sem: 6

	Sem	SRN	Faculty ID	ISA1	ISA2	ISA3	ESA
0	6	pes2ug20cs000	1	20	20	20	80

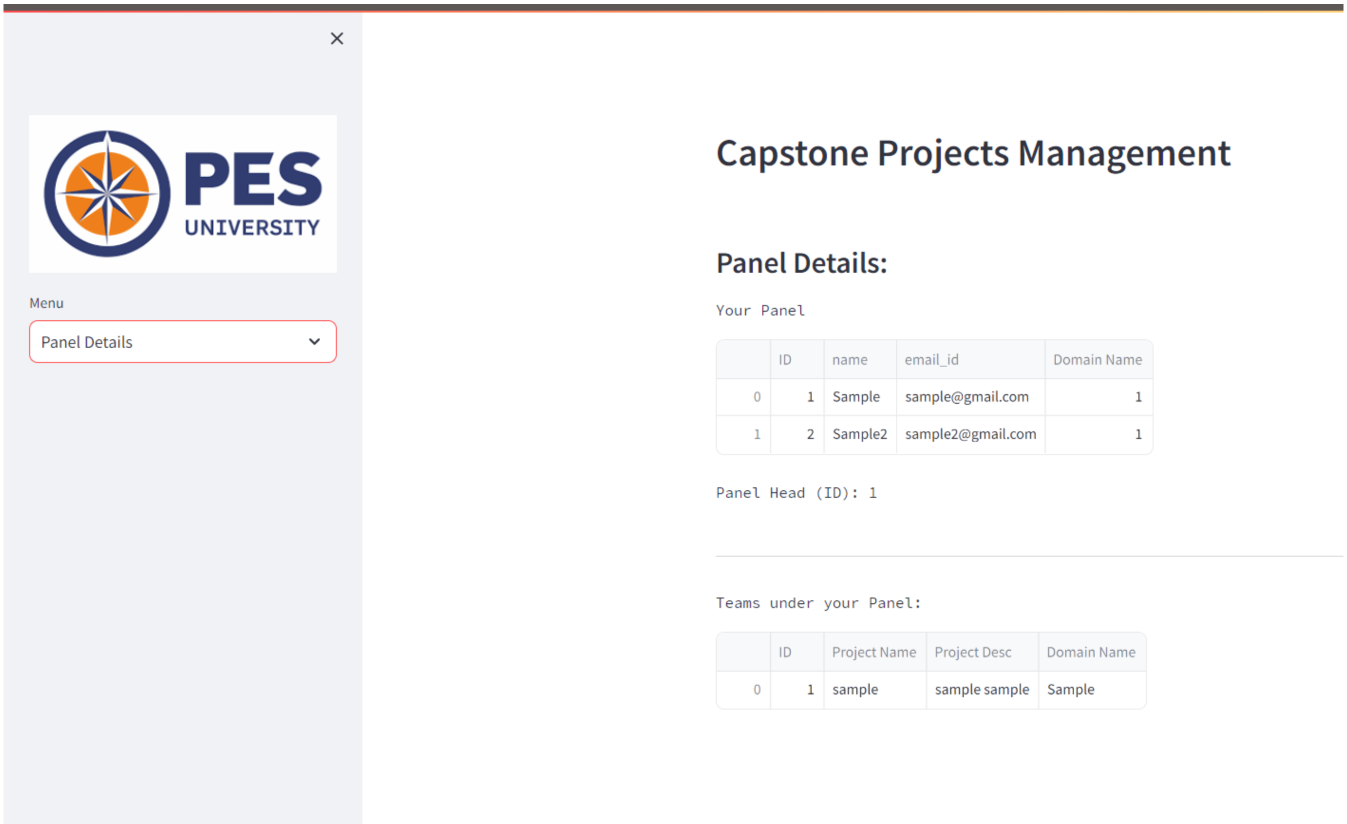
ISA1: 0

ISA2: 0

ISA2: 0

ESA: 0

Update



Capstone Projects Management

Panel Details:

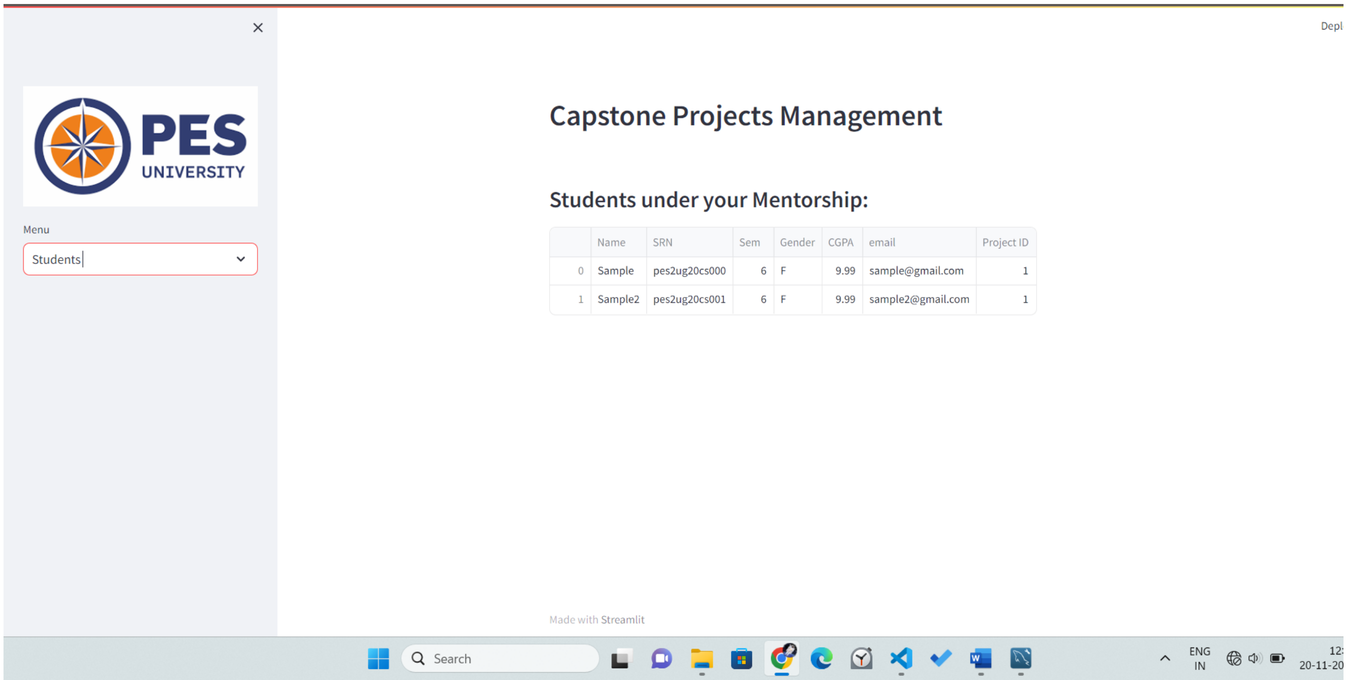
Your Panel

	ID	name	email_id	Domain Name
0	1	Sample	sample@gmail.com	1
1	2	Sample2	sample2@gmail.com	1

Panel Head (ID): 1

Teams under your Panel:

	ID	Project Name	Project Desc	Domain Name
0	1	sample	sample sample	Sample

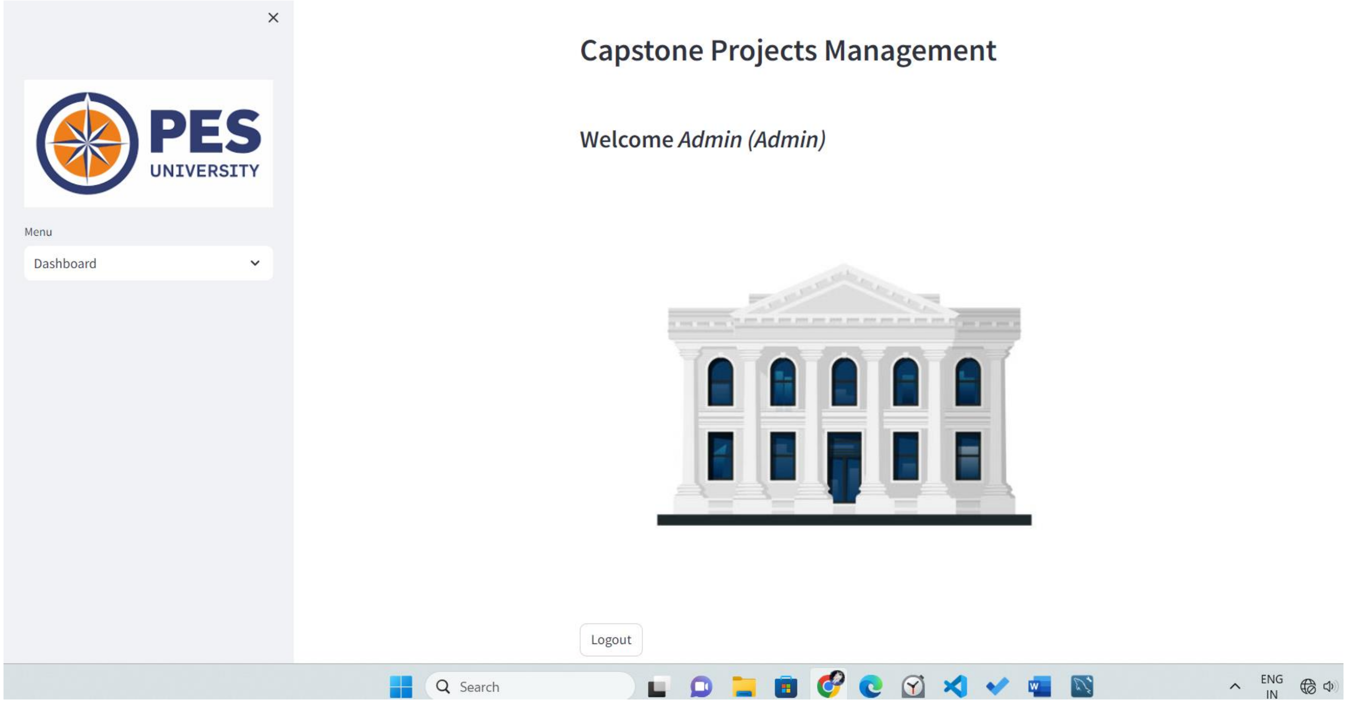


Capstone Projects Management

Students under your Mentorship:

	Name	SRN	Sem	Gender	CGPA	email	Project ID
0	Sample	pes2ug20cs000	6	F	9.99	sample@gmail.com	1
1	Sample2	pes2ug20cs001	6	F	9.99	sample2@gmail.com	1

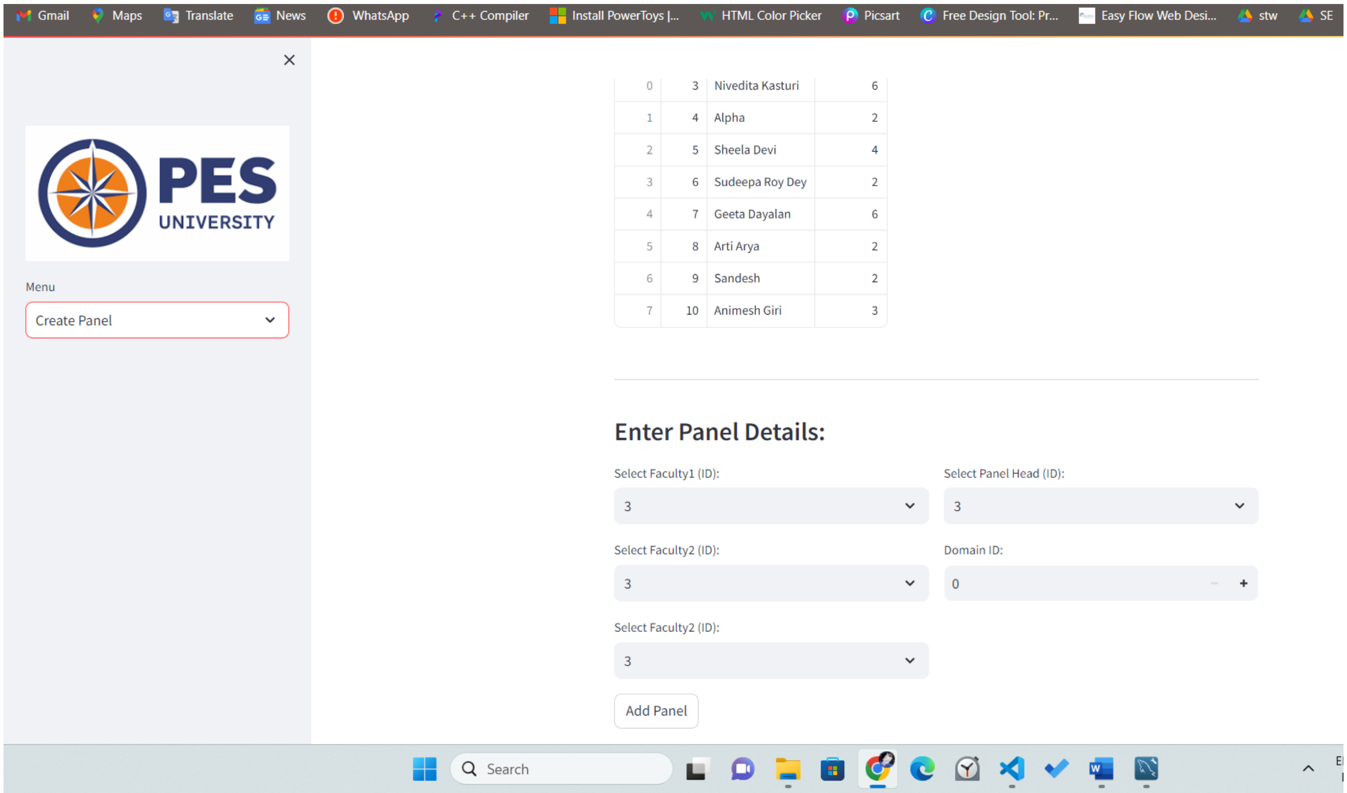
Admin:



Capstone Projects Management

Welcome *Admin (Admin)*

Menu: Dashboard



Menu: Create Panel

0	3	Nivedita Kasturi	6
1	4	Alpha	2
2	5	Sheela Devi	4
3	6	Sudeepa Roy Dey	2
4	7	Geeta Dayalan	6
5	8	Arti Arya	2
6	9	Sandesh	2
7	10	Animesh Giri	3

Enter Panel Details:

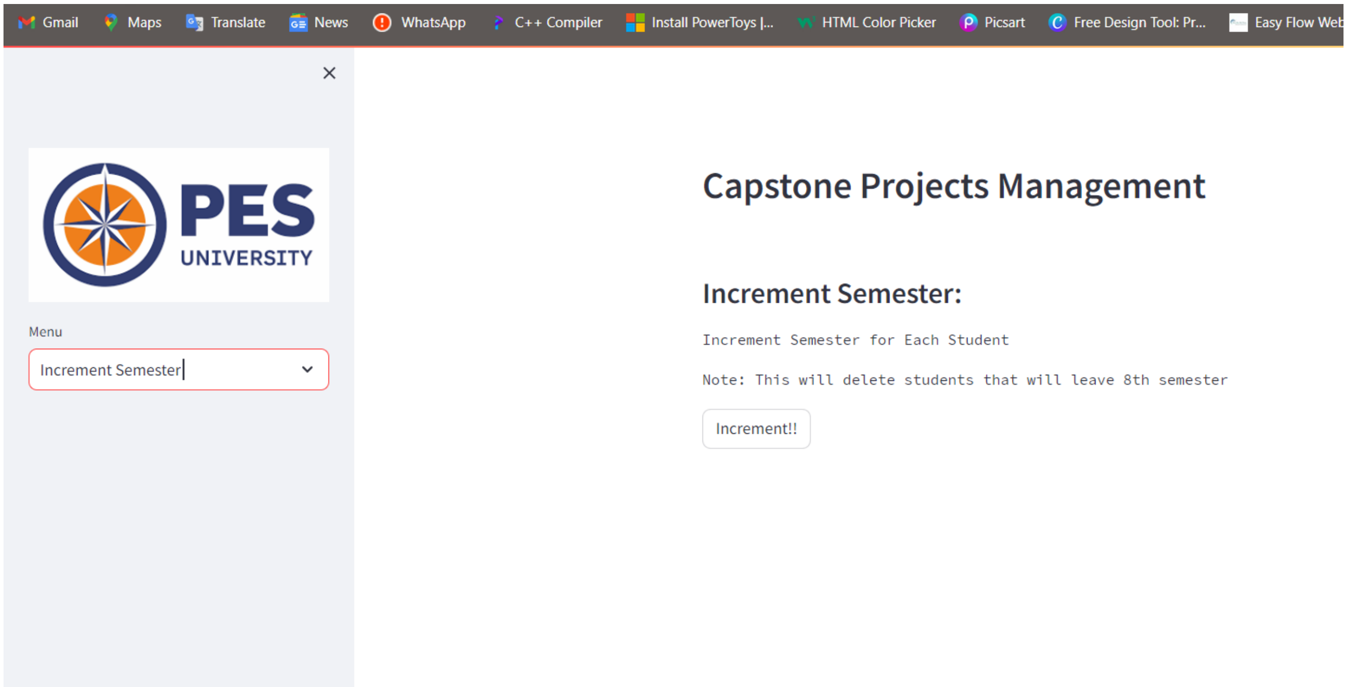
Select Faculty1 (ID): 3

Select Panel Head (ID): 3

Select Faculty2 (ID): 3

Domain ID: 0

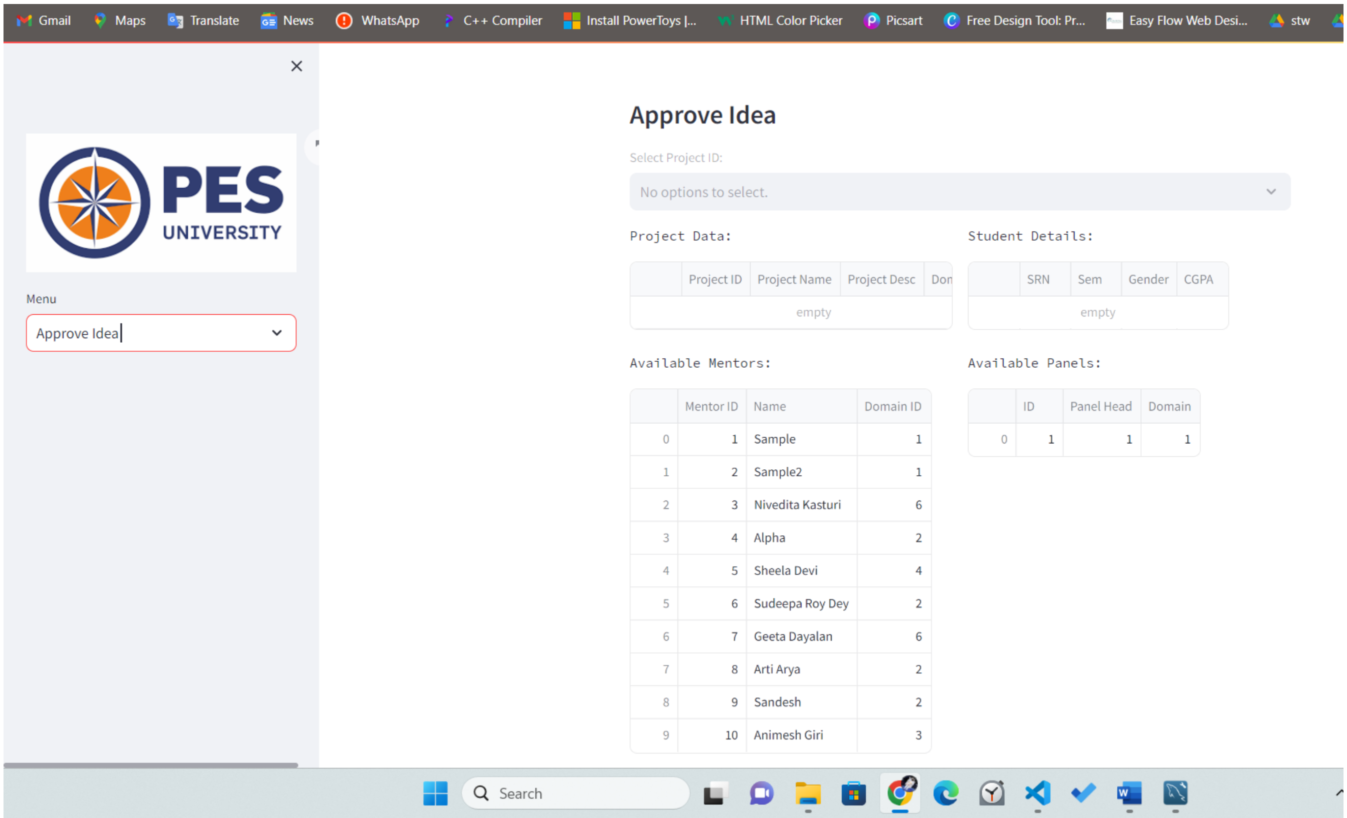
Select Faculty2 (ID): 3



Capstone Projects Management

Increment Semester:

```
Increment Semester for Each Student

Note: This will delete students that will leave 8th semester
```



Approve Idea

Select Project ID:

No options to select.

Project Data:

	Project ID	Project Name	Project Desc	Dor
		empty		

Student Details:

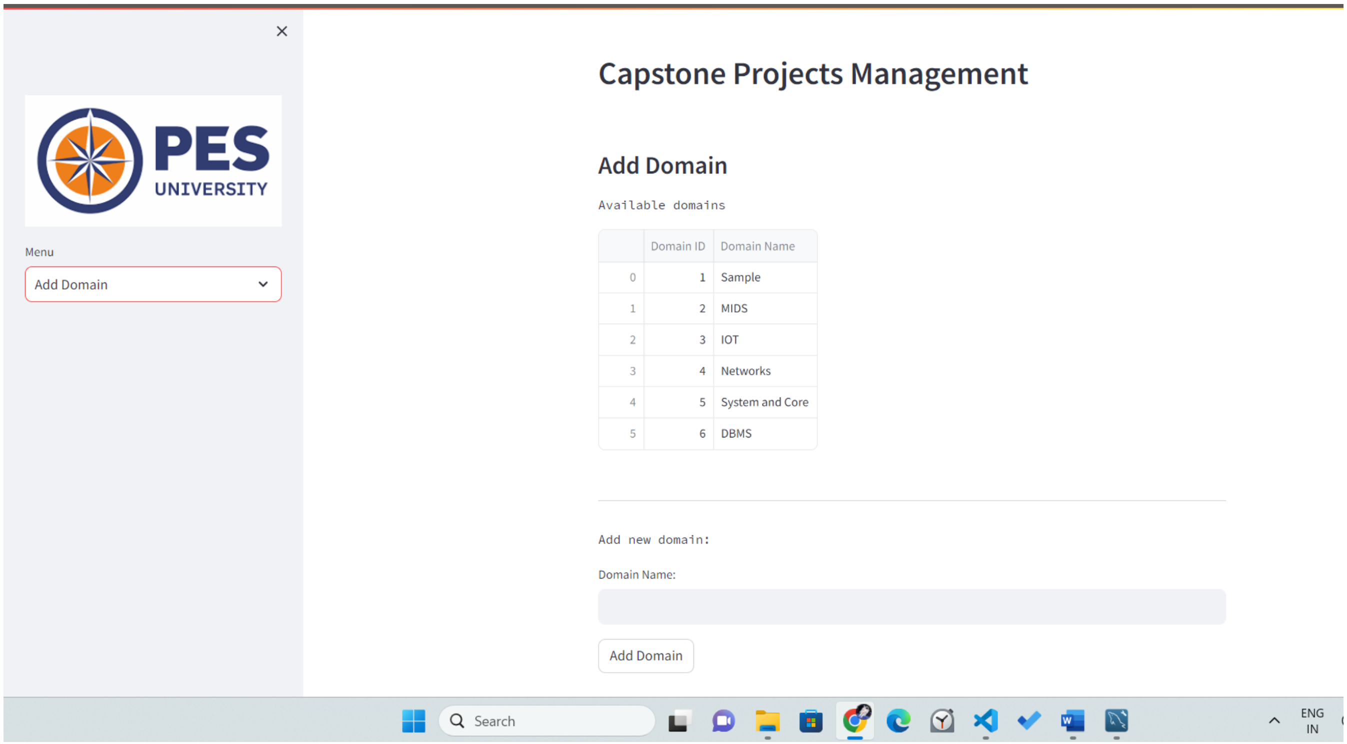
	SRN	Sem	Gender	CGPA
		empty		

Available Mentors:

	Mentor ID	Name	Domain ID
0	1	Sample	1
1	2	Sample2	1
2	3	Nivedita Kasturi	6
3	4	Alpha	2
4	5	Sheela Devi	4
5	6	Sudeepa Roy Dey	2
6	7	Geeta Dayalan	6
7	8	Arti Arya	2
8	9	Sandesh	2
9	10	Animesh Giri	3

Available Panels:

	ID	Panel Head	Domain
0	1	1	1



Menu: Add Domain

Capstone Projects Management

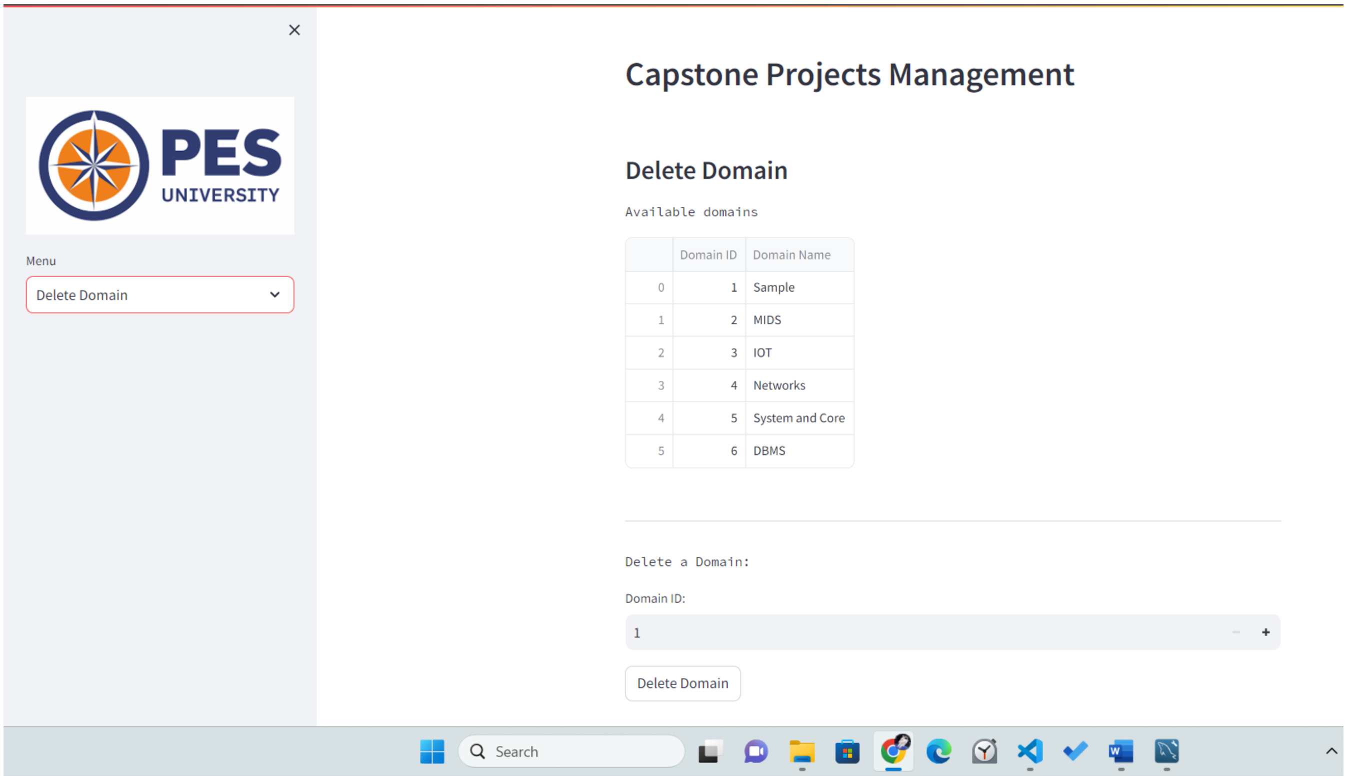
Add Domain

Available domains

	Domain ID	Domain Name
0	1	Sample
1	2	MIDS
2	3	IOT
3	4	Networks
4	5	System and Core
5	6	DBMS

Add new domain:

Domain Name:



Menu: Delete Domain

Capstone Projects Management

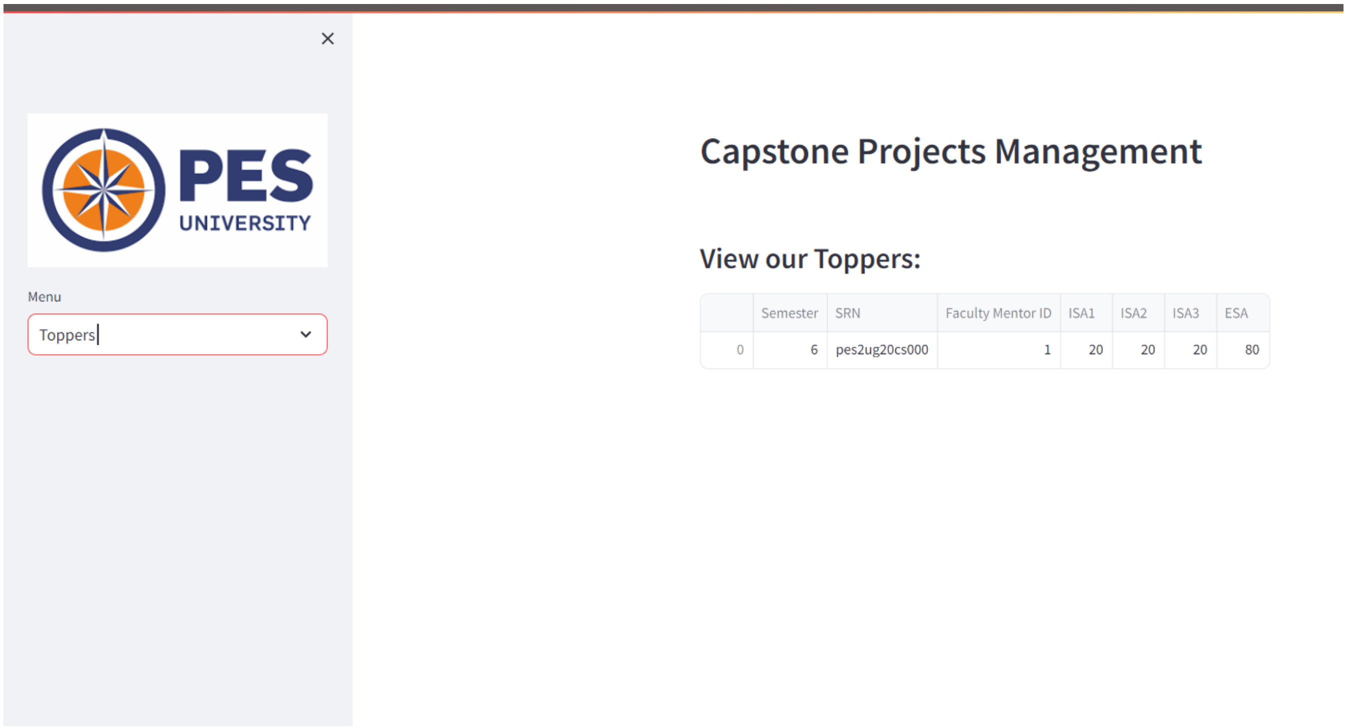
Delete Domain

Available domains

	Domain ID	Domain Name
0	1	Sample
1	2	MIDS
2	3	IOT
3	4	Networks
4	5	System and Core
5	6	DBMS

Delete a Domain:

Domain ID: 1



Menu: Toppers

Capstone Projects Management

View our Toppers:

	Semester	SRN	Faculty Mentor ID	ISA1	ISA2	ISA3	ESA
0	6	pes2ug20cs000	1	20	20	20	80

CONCLUSIONS

In conclusion, the implementation of the University Capstone Projects Management System represents a significant leap forward in addressing the challenges posed by the manual management of capstone projects within the university. By introducing a streamlined, digitized approach, this system serves as a catalyst for positive transformation, benefiting students, faculty, and administrators alike.

The system's core strengths lie in its ability to enhance efficiency, promote collaboration, and ensure accountability throughout the capstone project lifecycle. Students now experience a more seamless journey, from proposal submission to project evaluation, allowing them to focus on the substance of their work rather than administrative hurdles. Faculty advisors benefit from standardized evaluation criteria and streamlined communication, enabling them to provide timely guidance and feedback.

Administrators, in their managerial role, can rely on the system as a centralized hub for project-related information. The automation of workflows not only reduces the burden of manual record-keeping but also contributes to the accuracy and security of data. The transparency provided by the system fosters a culture of accountability and ensures that every stakeholder operates within their designated roles and permissions.

Furthermore, the system's adaptability and scalability position it as a sustainable solution for the evolving needs of the university. As project requirements, evaluation criteria, and university policies change over time, the system can seamlessly accommodate these adjustments, ensuring its relevance and effectiveness in the long run.

In essence, the University Capstone Projects Management System not only addresses the immediate challenges posed by manual processes but also sets the stage for a more dynamic, collaborative, and data-driven approach to managing capstone projects. By harnessing the power of technology, this system empowers the university community to fully realize the potential of capstone projects as integral components of academic and professional development. The journey toward streamlined project management has begun, promising a future where innovation and education intersect seamlessly.

REFERENCES

1.www.geeksforgeeks.com
2.www.openai.com
3.www.mysql.com