

## Project :- 1 (Extra\_Credit)

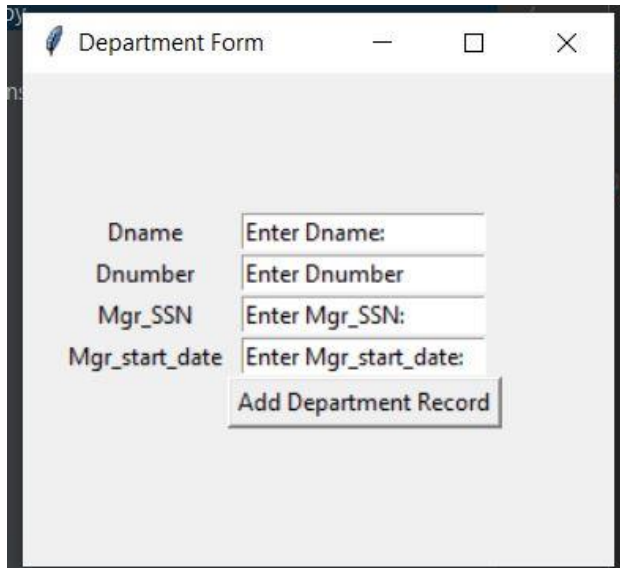
Name: Pruthvik H Kakadiya

1. Create 2 interactive (command line) or form or Web interfaces to INSERT a new record in each of the 2 tables DEPARTMENT and PROJECT. Each interface should allow the users to enter the attribute values of the new record, then INSERT the record in the table. Use the interfaces to insert 2 new records in each of the 2 tables. Turn in query results that show the new records that you inserted in each table.

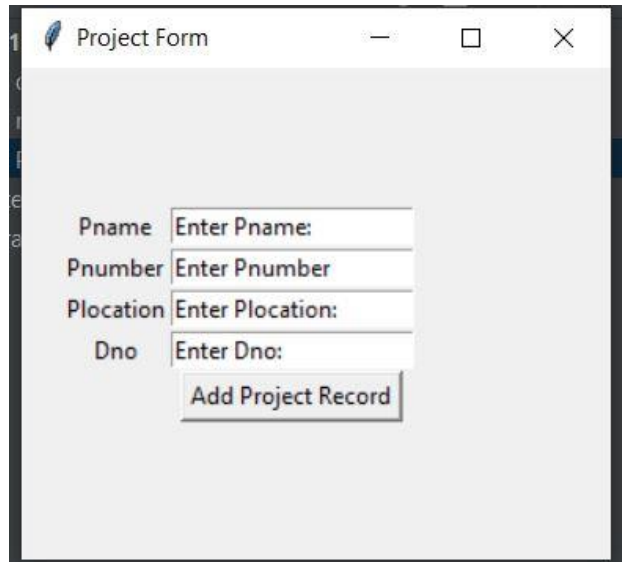
I have used python as my interactive programming language.

First using the Tkinter python library (Python Library to make simple and basic GUI applications ) I have made two forms

1: Department \_form

The screenshot shows a Tkinter window titled "Department Form". It contains four text input fields with labels to their left: "Dname" (with "Enter Dname:" inside), "Dnumber" (with "Enter Dnumber" inside), "Mgr\_SSN" (with "Enter Mgr\_SSN:" inside), and "Mgr\_start\_date" (with "Enter Mgr\_start\_date:" inside). Below these fields is a button labeled "Add Department Record".

2: Project\_form

The screenshot shows a Tkinter window titled "Project Form". It contains four text input fields with labels to their left: "Pname" (with "Enter Pname:" inside), "Pnumber" (with "Enter Pnumber" inside), "Plocation" (with "Enter Plocation:" inside), and "Dno" (with "Enter Dno:" inside). Below these fields is a button labeled "Add Project Record".

On the button click(for buttons Add Department Record & Add Project Record) events of these forms I have run the mysql connection code to connect these

form's codes to my mysql database and the query to INSERT records in tables Department and Project accordingly.

- Screenshots of INSERT INTO query:

```
sql = "INSERT INTO department VALUES(%s, %s, %s, %s)"
cursor.execute(sql, (cname1,cname2,cname3,cname4))
print("department record added")
mydb.commit()
```

```
sql = "INSERT INTO project VALUES(%s, %s, %s, %s)"
cursor.execute(sql, (cname1,cname2,cname3,cname4))
print("project record added")
mydb.commit()
```

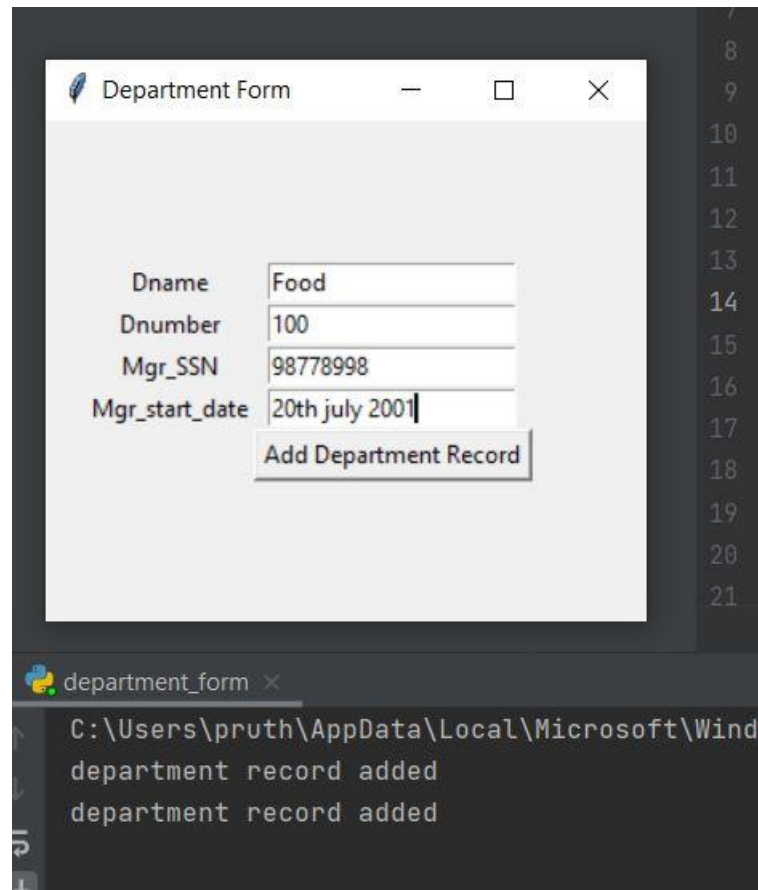
Here cname1-4 represents the columns 1-4 in respective areas Tables.

- Screenshots of records added:  
For department table I have inserted these two records.

The image shows a screenshot of a Python application window titled "Department Form". The form contains four text input fields with labels: "Dname" (containing "Fun"), "Dnumber" (containing "99"), "Mgr\_SSN" (containing "00000000"), and "Mgr\_start\_date" (containing "1st August 2021"). Below these fields is a button labeled "Add Department Record".

Below the form window, a terminal window is visible with the following output:

```
department record added
```



- Screenshot of query to show these records in database:

```

141 • select *
142   from project_1.department
143  where Dname = "Fun"
144  or Dname = "Food";
145

```

Result Grid				
Filter Rows: <input type="text"/>				
Edit:				
	Dname	Dnumber	Mgr_SSN	Mgr_start_date
	Fun	99	00000000	1st August 2021
	Food	100	98778998	20th july 2001
	NULL	NULL		NULL

For Project table I have inserted these two records.

The image displays two screenshots of a Windows application titled "Project Form". The application has a light gray background and contains four text input fields labeled "Pname", "Pnumber", "Plocation", and "Dno", followed by an "Add Project Record" button.

**Top Screenshot:** The input fields contain the following values: "Pname" is "Gaming", "Pnumber" is "99", "Plocation" is "Arlington", and "Dno" is "99". The "Add Project Record" button is visible below the fields. The Windows taskbar at the bottom shows the application path: `C:\Users\pruth\AppData\Local\Microsoft\WindowsApps\p` and the command prompt output: `project record added`.

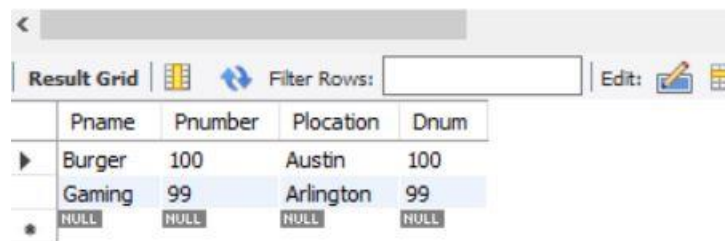
**Bottom Screenshot:** The input fields contain the following values: "Pname" is "Burger", "Pnumber" is "100", "Plocation" is "Austin", and "Dno" is "100". The "Add Project Record" button is visible below the fields. The Windows taskbar at the bottom shows the application path: `C:\Users\pruth\AppData\Local\Microsoft\WindowsApps\p` and the command prompt output: `project record added` followed by another `project record added`.

- Screenshot of query to show these records in database:

```

146 • select *
147 from project_1.project
148 where Pname = "Gaming"
149 or Pname = "Burger";
150

```



The screenshot shows a database interface with a 'Result Grid' tab. It displays two rows of data from a query. The first row is 'Burger' with Pnumber 100, Plocation Austin, and Dnum 100. The second row is 'Gaming' with Pnumber 99, Plocation Arlington, and Dnum 99. There is also a row with NULL values for all columns.

	Pname	Pnumber	Plocation	Dnum
▶	Burger	100	Austin	100
	Gaming	99	Arlington	99
*	NULL	NULL	NULL	NULL

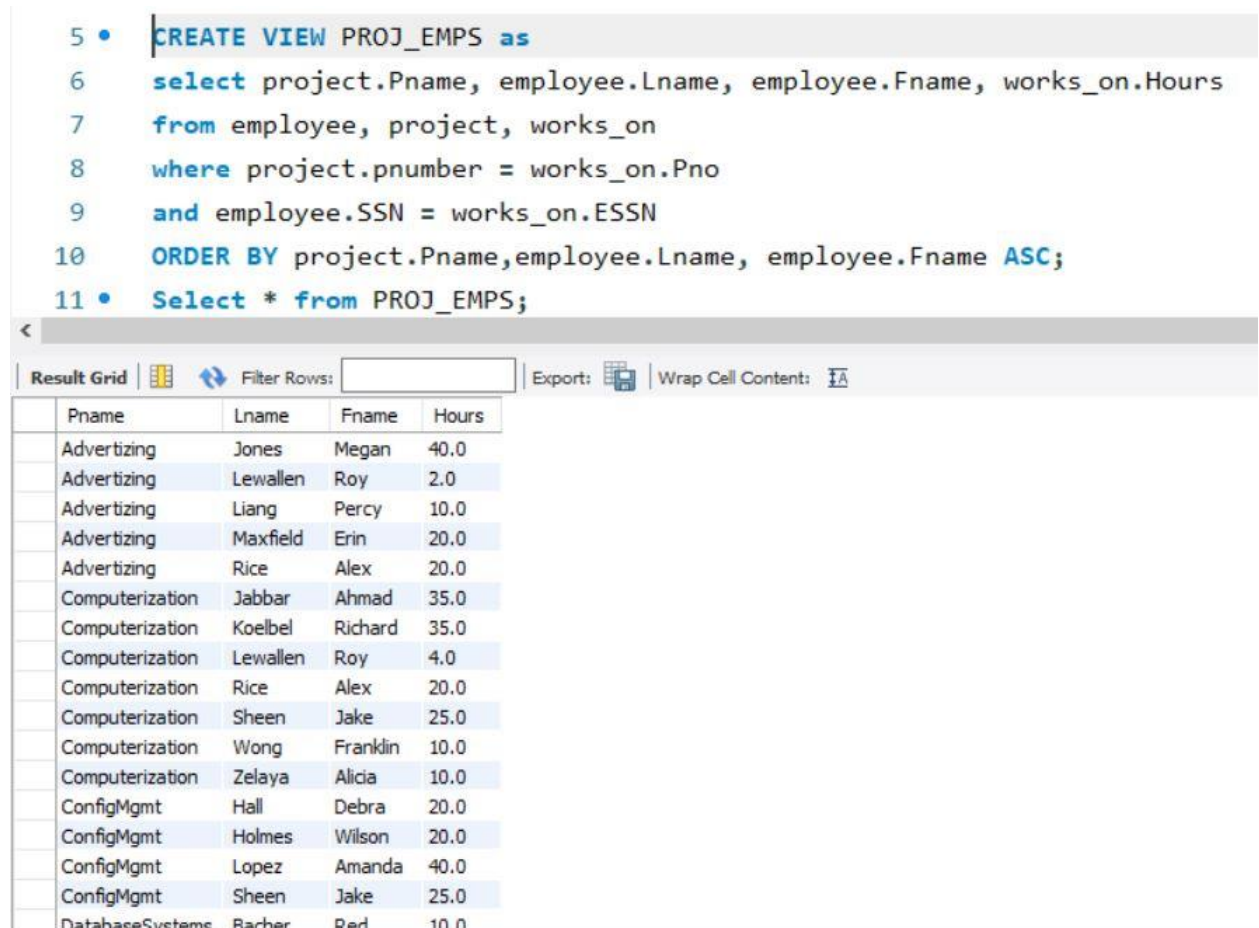
With this document I have also submitted the Department\_form.py and Project\_form.py which contains source codes of how I implemented each forms and connected them with mysql database.

2. Create 2 VIEWS as follows: the first view PROJ\_EMPS. The view will have the following information: PROJ\_EMPS (PNAME, LNAME, FNAME, HOURS) and will order the project info by PNAME in ascending order and within each PNAME the employees will be ordered by LNAME, FNAME in ascending order also. The second view PROJ\_SUMMARY will have one record for each project and will include the information PROJ\_SUMMARY (PNAME, NO\_OF\_EMPS, TOTAL\_HOURS), where NO\_OF\_EMPS is the number (COUNT) of employees currently working on the project and TOTAL\_HOURS is the SUM of the HOURS that the employees work on the project. Order the data by PNAME in descending order.

Query: view 1:

```
CREATE VIEW PROJ_EMPS as
select project.Pname, employee.Lname, employee.Fname, works_on.Hours
from employee, project, works_on
where project.pnumber = works_on.Pno
and employee.SSN = works_on.ESSN
ORDER BY project.Pname,employee.Lname, employee.Fname ASC;
Select * from PROJ_EMPS;
```

Screenshot:



```
5 • CREATE VIEW PROJ_EMPS as
6   select project.Pname, employee.Lname, employee.Fname, works_on.Hours
7   from employee, project, works_on
8   where project.pnumber = works_on.Pno
9   and employee.SSN = works_on.ESSN
10  ORDER BY project.Pname,employee.Lname, employee.Fname ASC;
11 • Select * from PROJ_EMPS;
```

Pname	Lname	Fname	Hours
Advertizing	Jones	Megan	40.0
Advertizing	Lewallen	Roy	2.0
Advertizing	Liang	Percy	10.0
Advertizing	Maxfield	Erin	20.0
Advertizing	Rice	Alex	20.0
Computerization	Jabbar	Ahmad	35.0
Computerization	Koelbel	Richard	35.0
Computerization	Lewallen	Roy	4.0
Computerization	Rice	Alex	20.0
Computerization	Sheen	Jake	25.0
Computerization	Wong	Franklin	10.0
Computerization	Zelaya	Alicia	10.0
ConfigMgmt	Hall	Debra	20.0
ConfigMgmt	Holmes	Wilson	20.0
ConfigMgmt	Lopez	Amanda	40.0
ConfigMgmt	Sheen	Jake	25.0
DatatabaseSystems	Racher	Red	10.0

Query: view 2:

```
CREATE VIEW PROJ_SUMMARY as
select project.Pname, count(works_on.ESSN) as Number_of_Employees,
sum(works_on.Hours) as Total_Hours
from project, works_on
where project.pnumber = works_on.Pno
group by Pname
ORDER BY project.Pname DESC;
Select * from PROJ_SUMMARY;
```

Screenshot:

```
23 • CREATE VIEW PROJ_SUMMARY as
24   select project.Pname, count(works_on.ESSN) as Number_of_Employees,
25   sum(works_on.Hours) as Total_Hours
26   from project, works_on
27   where project.pnumber = works_on.Pno
28   group by Pname
29   ORDER BY project.Pname DESC;
30 • Select * from PROJ_SUMMARY;
31
```

Pname	Number_of_Employees	Total_Hours
SearchEngine	2	34.0
Reorganization	5	44.0
ProductZ	6	124.0
ProductY	4	39.5
ProductX	5	94.5
OperatingSystem	11	358.0
Newbenefits	5	84.0
MotherBoard	3	40.0
Middleware	10	260.0
LaserPrinters	5	158.0
InkjetPrinters	10	332.0
Human1	4	101.0
EntityAnnot	3	61.0
DataMining	1	18.0
DatabaseSystems	11	347.0
ConfigMgmt	4	105.0
Computerization	7	139.0
Advertizing	5	92.0