Project

```
<html>
<head>
<title>Database Application</title>
</head>
<body>'
<form name ="Hello" action="2.py"method="post">
<h2>Employee Data Entry</h2>
<p><label for="first_name">First name:</label>
<input id="first_name" name="first_name" type="text" placeholder="Enter first
name:"></p>
<p><label for="last_name">Last name:</label>
<input id="last_name" name="last_name" type="text" placeholder="Enter last
name:"></p>
<p><label for="age">Age:</label>
<input id="age" name="age" type="text" placeholder="Enter age:"></p>
<p><label for="gender">Gender:</label>
<input id="gender" name="gender" type="text" placeholder="Enter gender:"></p>
<p><label for="income">Income:</label>
<input id="income" name="income" type="text" placeholder="Enter income:"></p>
<input id="Done" name="Done" type="Submit" value="store">
</form>
</body>
</html>
```

```
#!c:\Python34\python.exe
import cgi,cgitb,mysql.connector
cgitb.enable()
config={
        "user":"Poorvi",
        "password":"mice",
        "host":"localhost",
        "database":"Ridhi",
        "raise_on_warnings":True}
print("Content-type:text/html\r\n\r\n")
form=cgi.FieldStorage()
first_name=form.getvalue("first_name")
last_name=form.getvalue("last_name")
age=form.getvalue("age")
```

```
gender=form.getvalue("gender")
income=form.getvalue("income")
cn=mysql.connector.connect(**config)
cursor=cn.cursor()
sql="INSERT INTO
EMPLOYEE(FIRST_NAME,LAST_NAME,AGE,GENDER,INCOME)Values('%s','%s','%
d','%s','%d')"%(first_name,last_name,int(age),gender,int(income))
try:
    cursor.execute(sql)
    cn.commit()
    print("Record inserted.")
except:
    cn.rollback()
    print("error storing record")
print("<html>")
print("<head>")
print("<title>Employee Records</title>")
print("</head>")
print("<body>")
try:
    cursor.execute("select * from employee")
    results=cursor.fetchall()
    print("<table align='center'border=2>")
    print("<tr><td>First name</td><td>Last
name</td><td>age</td><td>gender</td><td>income</td></tr>")
    for row in results:

print("<tr><td>%s</td><td>%s</td><td>%d</td><td>%s</td><td>%d</td></tr>"%(row[0
],row[1],row[2],row[3],row[4]))
    print("</table>")
except:
    print("Error fetching records.")
print("</body>")
print("</html>")
cn.close()
```

Project

```
from tkinter import *
import mysql.connector
config={
        "user":"Poorvi",
        "password":"mice",
        "host":"localhost",
        "database":"Ridhi",
        "raise_on_warnings":True}

print("Connected to database")
class Database_application():
    def new_record(self):
        self.b_new['state']=DISABLED
        self.populate_blank();
    def save_record(self):
        first_name=self.entry_first_name.get()
        last_name=self.entry_last_name.get()
        age=int(self.entry_age.get())
        gender=self.entry_gender.get()
        income=float(self.entry_income.get())
        if self.b_new['state']==DISABLED:
            sql="INSERT INTO
EMPLOYEE(FIRST_NAME,LAST_NAME,AGE,GENDER,INCOME)values('%s','%s','%d
','%s','%f')"%(first_name,last_name,age,gender,income)
        else:
            sql="UPDATE EMPLOYEE SET
LAST_NAME='%s',AGE='%d',GENDER='%s',INCOME='%f' WHERE first_name like
'%s'"%(last_name,age,gender,income,first_name)
        try:

            self.cursor.execute(sql)
            self.cn.commit()
            sql="SELECT * FROM EMPLOYEE"
            self.cursor.execute(sql)
            self.results=self.cursor.fetchall()
        except:
            self.cn.rollback()
            print("Some error")
        self.b_new['state']=NORMAL
    def delete_record(self):
        first_name=self.entry_first_name.get()
        sql="DELETE FROM EMPLOYEE WHERE first_name like'%s'"%first_name
        try:
            self.cursor.execute(sql)
            self.cn.commit()
            sql="SELECT * FROM EMPLOYEE"
            self.cursor.execute(sql)
            self.results=self.cursor.fetchall()
            self.populate_blank()
            self.previous_record()
        except:
            self.cn.rollback()
    def exit_form(self):
        self.root.destroy()
    def first_record(self):
        if len(self.results)>0:
            self.current_record=0
        self.populate_record()
    def next_record(self):
        if len(self.results)>0:
            self.current_record+=1
            if self.current_record>=len(self.results):
                self.current_record=0
            self.populate_record()
    def previous_record(self):
        if len(self.results)>0:
            self.current_record-=1
            if self.current_record<0:
                self.current_record=len(self.results)-1
            self.populate_record()
    def last_record(self):
        if len(self.results)>0:
                self.current_record=len(self.results)-1
        self.populate_record()
    def __init__(self):
        self.root=Tk()
        self.cn=mysql.connector.connect(**config)
        self.root.title("Records")
        self.root.geometry("225x270")
        label_first_name=Label(self.root,text="First name:")
        self.entry_first_name=Entry(self.root)
        label_last_name=Label(self.root,text="Last name:")
        self.entry_last_name=Entry(self.root)
        label_age=Label(self.root,text="Age:")
        self.entry_age=Entry(self.root)
        label_gender=Label(self.root,text="Gender:")
        self.entry_gender=Entry(self.root)
        label_income=Label(self.root,text="Income:")
        self.entry_income=Entry(self.root)
        self.b_new=Button(self.root,text="New",command=self.new_record)
        self.b_save=Button(self.root,text="Save",command=self.save_record)
        self.b_delete=Button(self.root,text="Delete",command=self.delete_record)
        self.b_exit=Button(self.root,text="Exit",command=self.exit_form)
        self.b_first=Button(self.root,text="First",command=self.first_record)
        self.b_next=Button(self.root,text="Next",command=self.next_record)
        self.b_previous=Button(self.root,text="previous",command=self.previous_record)
        self.b_last=Button(self.root,text="last",command=self.last_record)
        label_first_name.place(x=20,y=20,width=60,height=25)
        self.entry_first_name.place(x=90,y=20,width=100,height=25)
        label_last_name.place(x=20,y=55,width=60,height=25)
        self.entry_last_name.place(x=90,y=55,width=100,height=25)
        label_age.place(x=20,y=90,width=60,height=25)
        self.entry_age.place(x=90,y=90,width=100,height=25)
        label_gender.place(x=20,y=125,width=60,height=25)
        self.entry_gender.place(x=90,y=125,width=100,height=25)
        label_income.place(x=20,y=160,width=60,height=25)
        self.entry_income.place(x=90,y=160,width=100,height=25)

        self.b_new.place(x=20,y=195,width=40,height=25)
        self.b_save.place(x=70,y=195,width=40,height=25)
        self.b_delete.place(x=120,y=195,width=40,height=25)
        self.b_exit.place(x=170,y=195,width=40,height=25)
        self.b_first.place(x=20,y=230,width=40,height=25)
        self.b_next.place(x=70,y=230,width=40,height=25)
        self.b_previous.place(x=120,y=230,width=40,height=25)
        self.b_last.place(x=170,y=230,width=40,height=25)

        try:
            self.cursor=self.cn.cursor()
            sql="SELECT * FROM EMPLOYEE"
            self.cursor.execute(sql)
            self.results=self.cursor.fetchall()
            self.current_record=0
        except:
            print("Error fetching data.")
        self.first_record()
        self.root.mainloop()
    def populate_record(self):
        row=self.results[self.current_record]
        self.entry_first_name.delete(0,END)
        self.entry_first_name.insert(0,row[0])
        self.entry_last_name.delete(0,END)
        self.entry_last_name.insert(0,row[1])
        self.entry_age.delete(0,END)
        self.entry_age.insert(0,row[2])
        self.entry_gender.delete(0,END)
        self.entry_gender.insert(0,row[3])
        self.entry_income.delete(0,END)
        self.entry_income.insert(0,row[4])
    def populate_blank(self):
        self.entry_first_name.delete(0,END)
        self.entry_last_name.delete(0,END)
        self.entry_age.delete(0,END)
        self.entry_gender.delete(0,END)
        self.entry_income.delete(0,END)
d=Database_application();
```