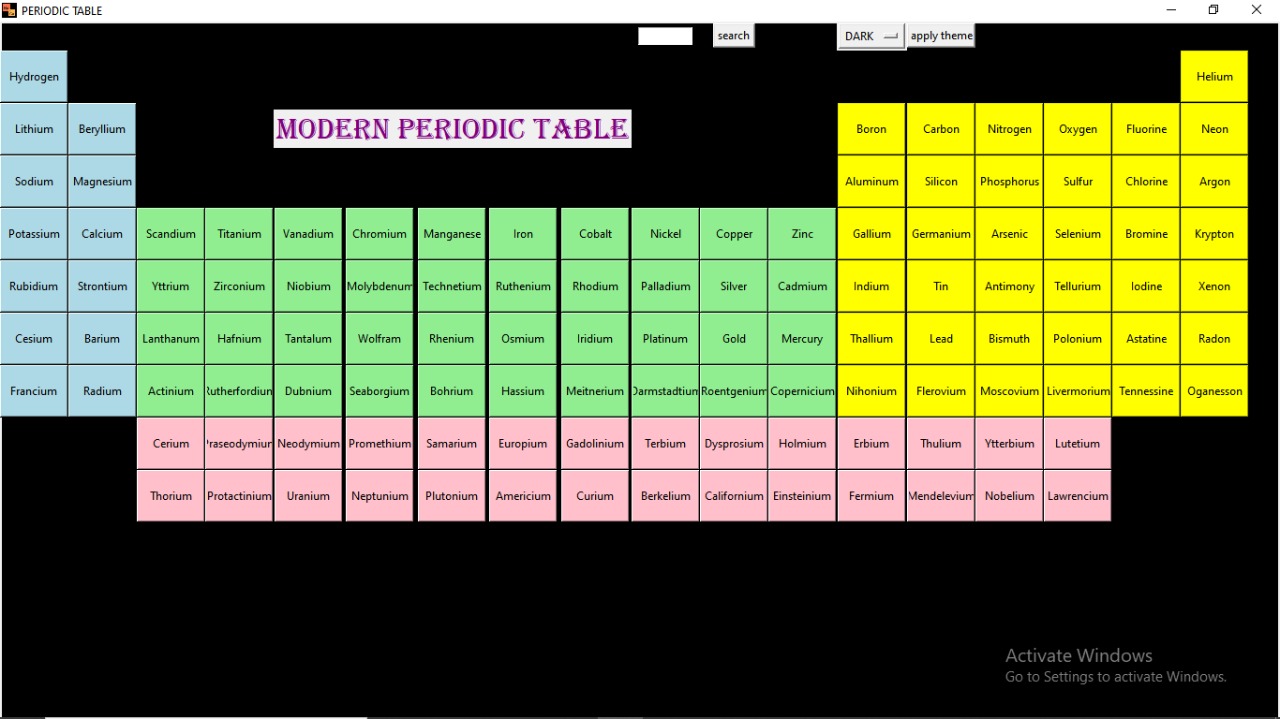


**CLASS XII Computer Science Project**

**PERODIC TABLE**



Name: Class: XII

Roll No: Section: B

`

ACKNOWLEDGEMENT

I hereby thank the school and the staff of the Computer Department for all the guidance and support extended to me during the course of this project.

My special thanks are due to the Principal for providing us with the right platform and environment to complete the project.

My sincere thanks to Ms. Madhurima Kashyap, our Computer Science teacher for her able guidance, unflinching support, and the affectionate touch without which this project would not have been a success.

Finally, I thank all those who have directly or indirectly contributed to the success of this project.

Name:

Roll No:

INDEX

Hardware Requirement ------------------------------------------- 5

Software Requirement -------------------------------------------- 5

Python ----------------------------------------------------------------- 6

MySQL ----------------------------------------------------------------- 7

Synopsis --------------------------------------------------------------- 8

Aims and Objective ------------------------------------------------- 9

Table Structure ------------------------------------------------------ 10

Screens and Code --------------------------------------------------- 16

Declaration ----------------------------------------------------------- 25

HARDWARE REQUIREMENT

* Processor: Intel Core i5 Processor or any other similar processor
* RAM: 8 GB
* System type: X64-based processor
* Storage: 500 GB

SOFTWARE REQUIREMENT

* Operating system: Windows 10
* Programming Language: Python 3.7 with connector & tkinter modules
* Database: MySQL 8.0 command line client
* Development Environment: Spider IDE

Python

Python is an object-oriented programming language created by Guido van Rossum in 1989. It is designed for a rapid prototyping of complex applications. It has interfaces to many operating systems. Many large companies use the Python programming language include NASA, Google, YouTube, etc.

Python programming is also used in Artificial Intelligence and other advanced fields of Computer Science. Python's simple, easy to learn syntax makes readability easy; therefore it is rather easy to maintain a program written in Python.

Its high-level built in data structures make it very attractive not only for rapid application development but also for use as a scripting or glue language to connect existing components together.

Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available without charge for all major platforms, and can be freely distributed.

Python programs are generally expected to run slower than Java programs, but they also take much less time to develop. Python programs are typically 3-5 times shorter than equivalent Java programs. This difference can be attributed to Python's built-in high-level data types and its dynamic typing, which means that a Python programmer wastes no time declaring the types of arguments or variables.

MySQL

MySQL is a relational database management system based on SQL (Structured Query Language). The application is used in a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for MySQL however, is for the purpose of a web database.

MySQL is supported by Oracle Corporation and is released under an open-source license, so, user doesn’t have to pay to use it. It’s a very powerful program and easily rivals other paid expensive and powerful database packages.

MySQL works on many operating systems and with many languages including C++, JAVA, Python, etc. It works very quickly and works well even with large data sets.

SYNOPSIS

Periodic table is one of the most important parts of chemistry .In order to make chemistry interesting and easy to learn we have made a periodic table which is a chart that arranges all of the known elements according to their atomic numbers.

The struggle to arrange the elements in a sequential manner has been going on since centuries. There were many scientists who have tried to do so namely Dobereiner, Newland, Mendeleev, and finally in 1869 the MODERN PERIODIC TABLE was created which on getting updated as scientists discover new elements. In our project we have tried to make it more easy to use interface and add all the necessary information about the 118 elements present in the periodic table like their atomic name, atomic mass , atomic radius, symbol ,number of isotopes , radioactivity ,etc. to make the user more interested and gains the most out of it.

The Graphic User Interface (GUI) is developed using the tkinter module of Python. The student details are stored in MySQL database. The connector module of Python is used to connect to the database to insert and retrieve information. The Spyder IDE and MySQL command line client have been used for the development of this project.

AIMS AND OBJECTIVE

* Periodic table is one of the most important parts of chemistry.
* To make chemistry interesting and easy to learn we have made a periodic table which is a chart that arranges all the known elements according to their atomic numbers
* In our project we have tried to make it more easy to use interface and add all the necessary information about the 118 elements present in the periodic table like their atomic name, atomic mass , atomic radius, symbol ,number of isotopes , radioactivity ,etc. to make the user more interested and gains the most out of it
* Every year a new property of element is discovered so this module enables us to update the information about the elements without changing the main program easily

TABLE STRUCTURE

**The following tables are used in the “PERODIC TABLE “**

CREATE TABLE tableName

(

Atomic\_Number varchar(300),

Element varchar(300),

Symbol varchar(300),

Atomic\_Mass varchar(300),

Number\_of\_Neutrons varchar(300),

Number\_of\_Protons varchar(300),

NumberofElectrons varchar(300),

Period varchar(300),

Grp varchar(300),

Phase varchar(300),

Radioactive varchar(300),

Naturally\_occuring varchar(300),

Metal varchar(300),

Nonmetal varchar(300),

Metalloid varchar(300),

Type varchar(300),

Atomic\_Radius varchar(300),

Electronegativity varchar(300),

First\_Ionization varchar(300),

Density varchar(300),

Melting\_Point varchar(300),

Boiling\_Point varchar(300),

Number\_Of\_Isotopes varchar(300),

Discoverer varchar(300),

Year varchar(300),

SpecificHeat varchar(300),

Number\_of\_Shells varchar(300),

Number\_of\_Valence varchar(255),

Block varchar(10)

);

TABLE STRUCTURE (contd..)

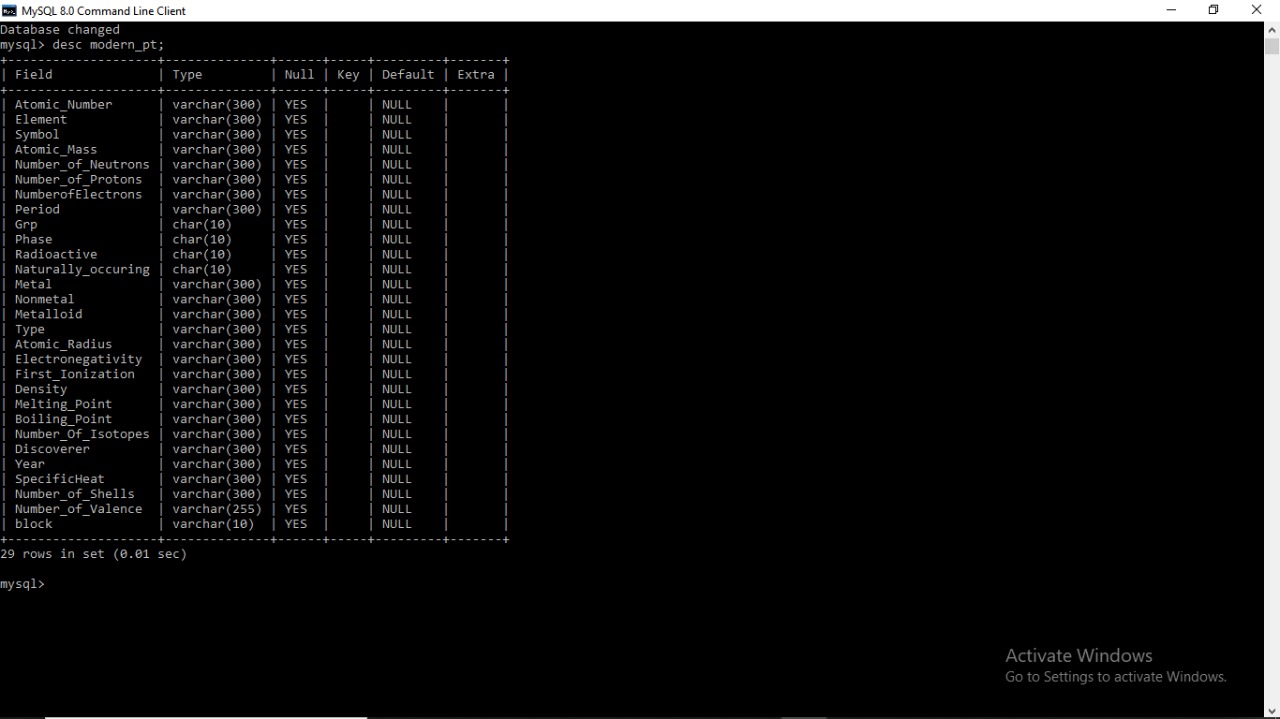


Table Values

Text

Description automatically generated

Text

Description automatically generated

Text

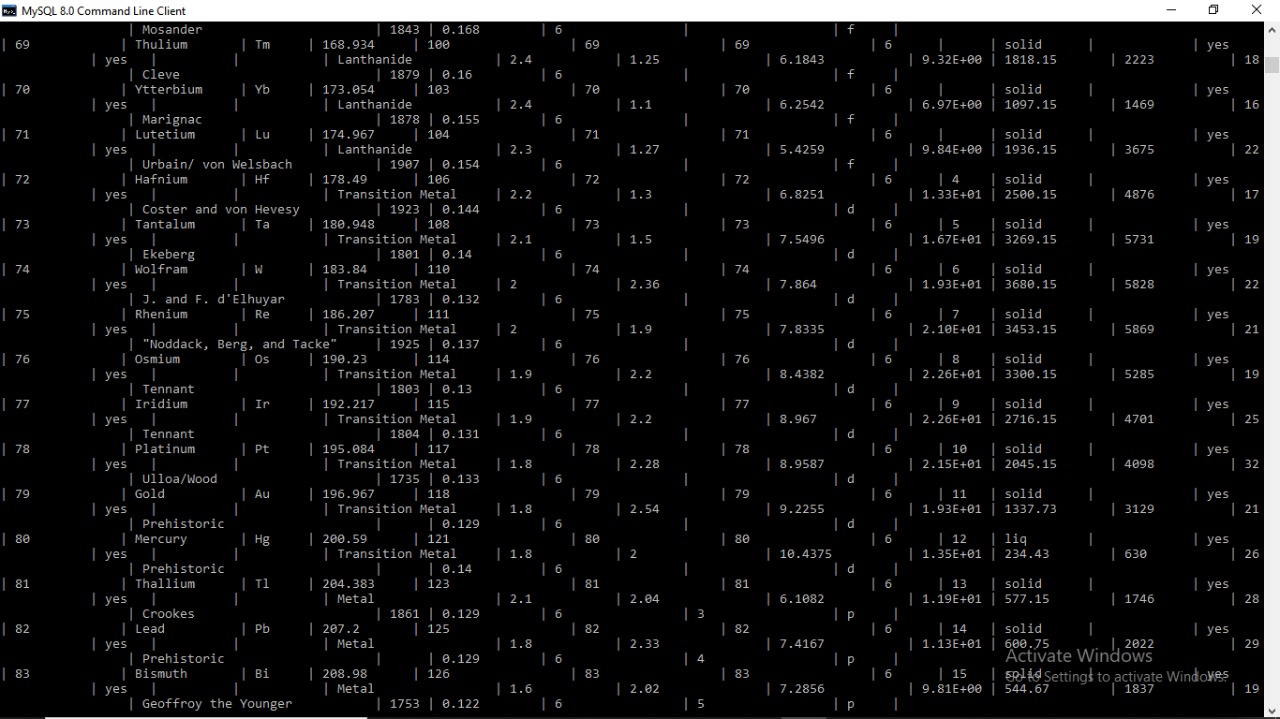
Description automatically generated

Text

Description automatically generated

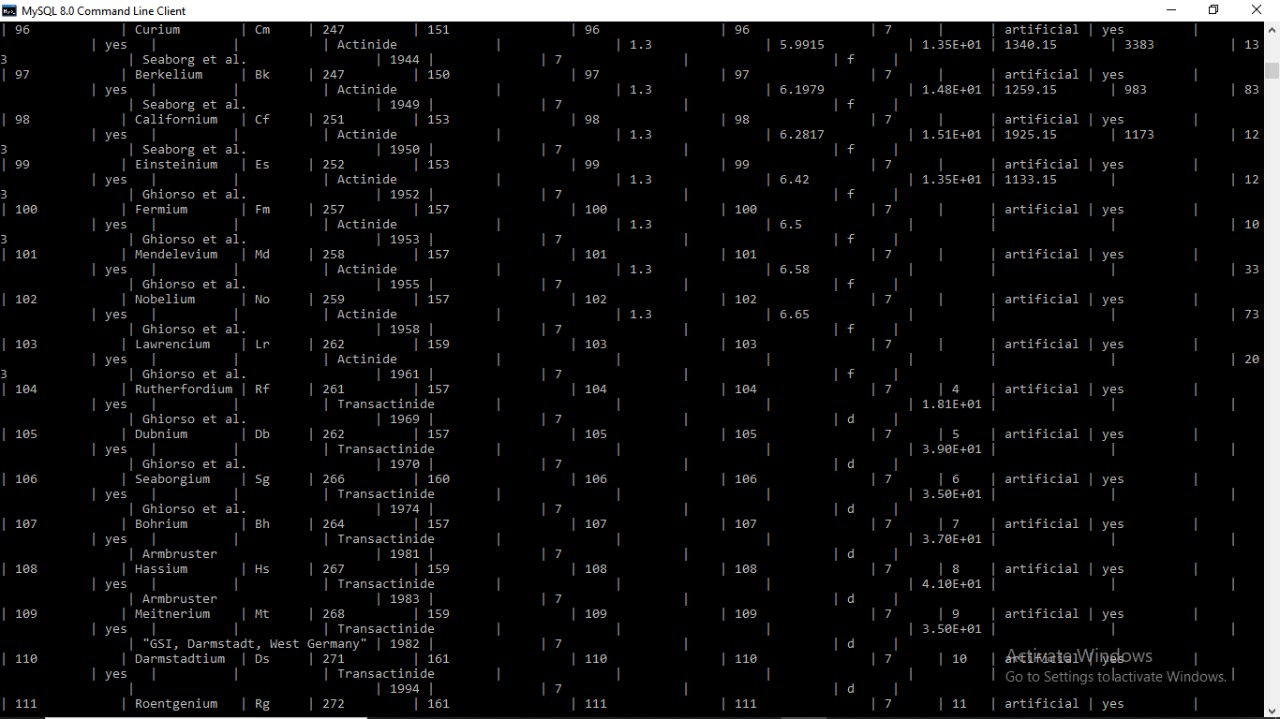
A picture containing text, window

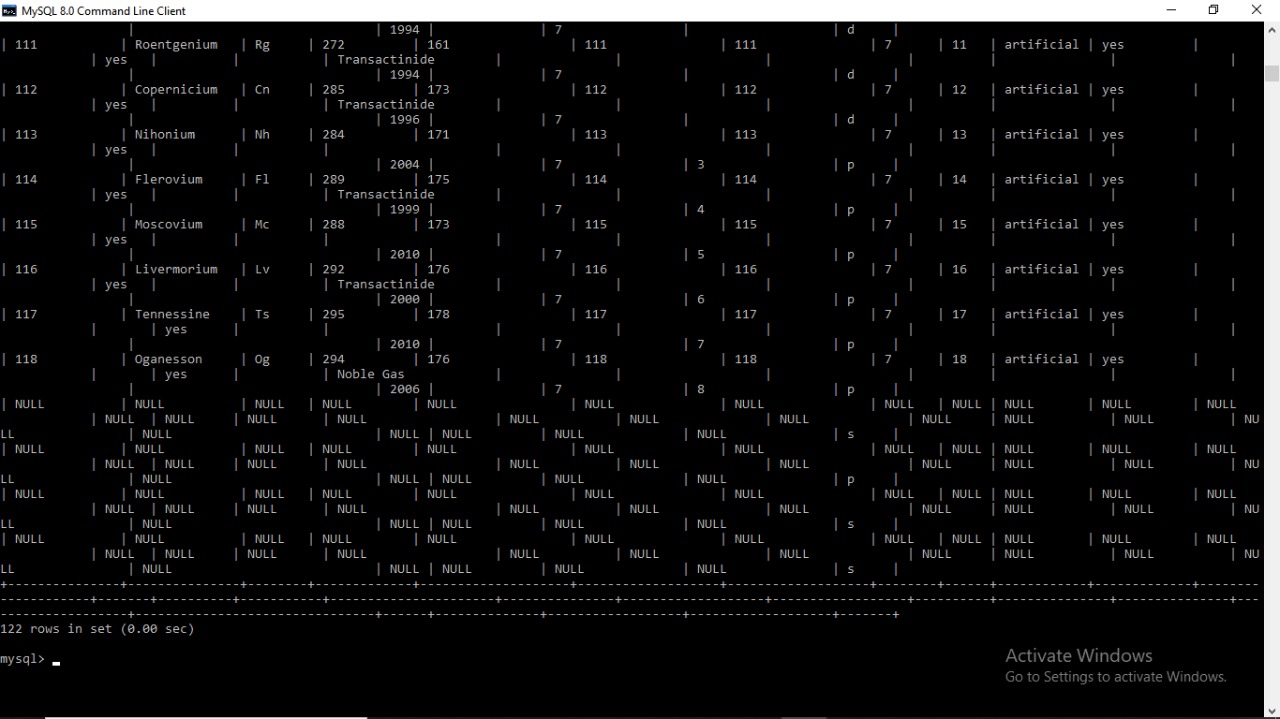
Description automatically generated



Text

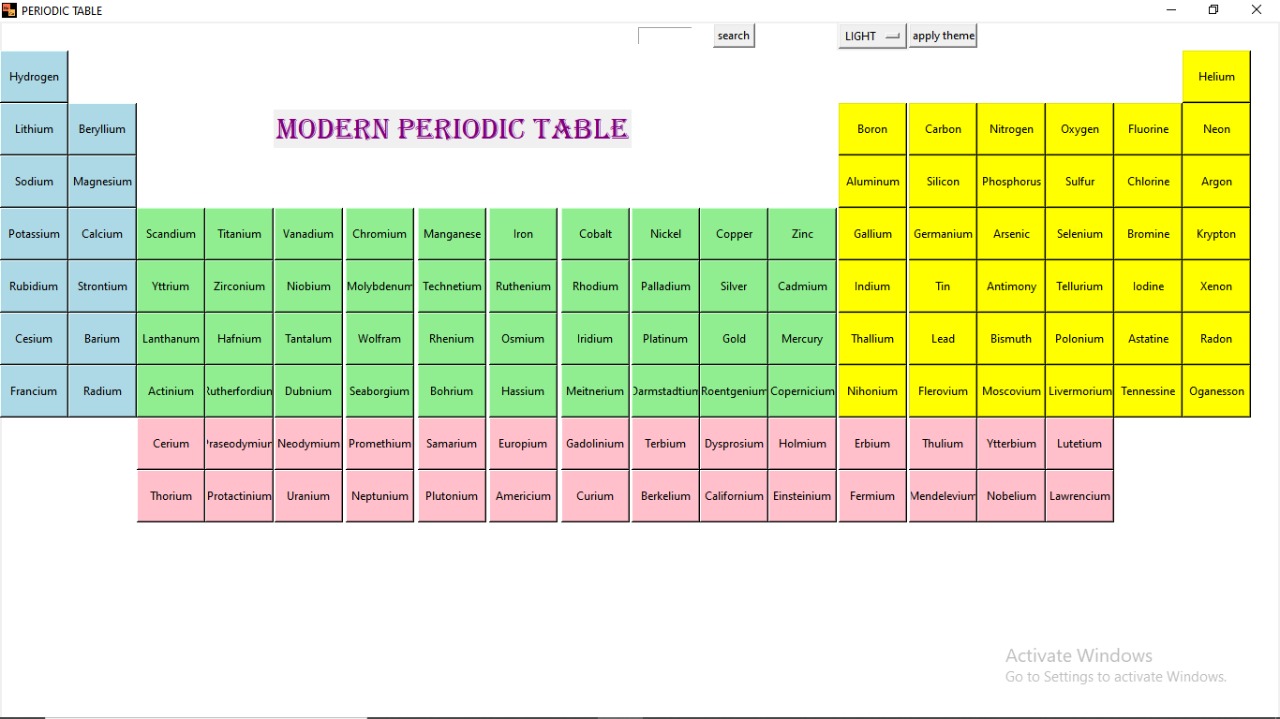
Description automatically generated



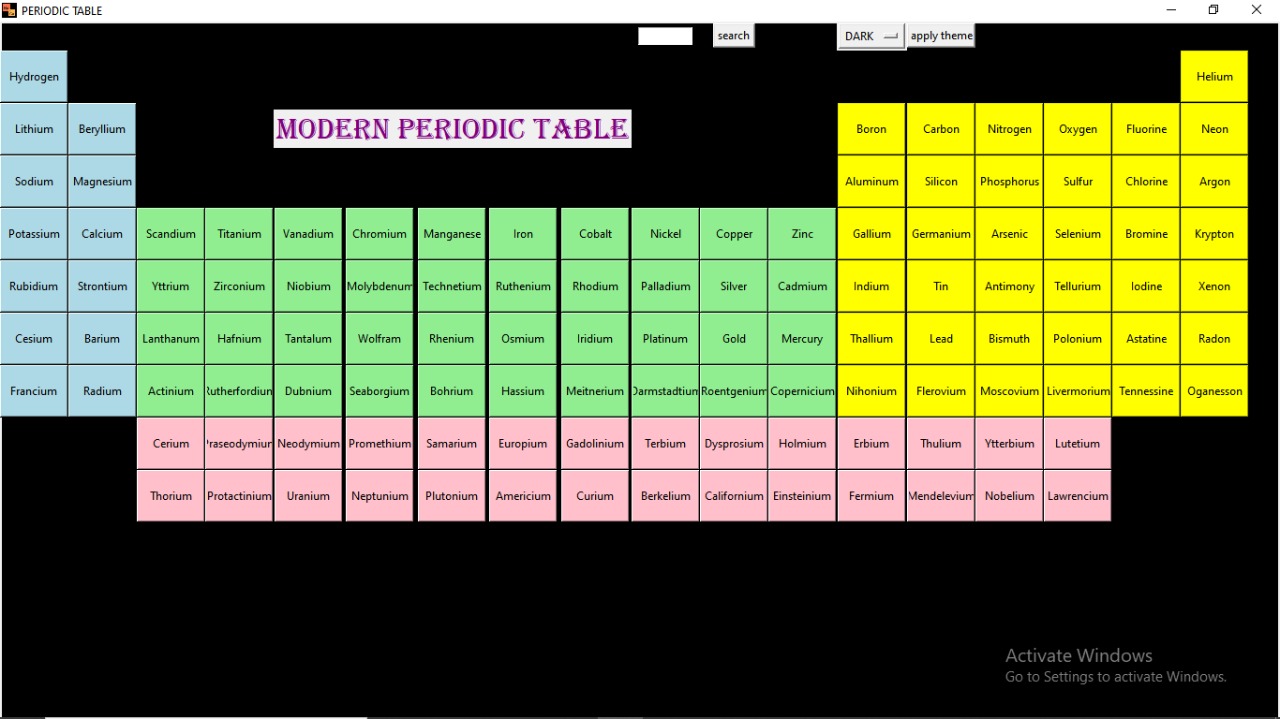


**SCREENS AND CODE**

1. All screen output shown below are at run time. The first screen that appears on launching the application is shown below (screen 1):



1. Application shown below (screen 2)



**## Code for screen 1:**

import mysql.connector as sql

import tkinter as tk

from tkinter import \*

import aaproject.buttoncomm as bc

from PIL import ImageTk,Image

root=tk.Tk()

root.iconbitmap('D:/python/icon.ico')

mysql=sql.connect(host = 'localhost', user ='root', password = '\*\*\*\*', database = 'periodic')

if mysql.is\_connected:

print("\_\_CONNECTED\_\_")

data=mysql.cursor()

root.title("PERIODIC TABLE")

canvas=tk.Canvas(root , height = "17000" , width = "17050" , bg="black")

canvas.pack()

base=tk.Canvas(canvas,bg="black")

base.place( relheight = "1" , relwidth = "1" , relx = "0" , rely = "0")

sbox=tk.Entry(base , width = 9)

sbox.grid(row=0 , column = 9)

sbutton=tk.Button(base,text="search",command=search)

sbutton.grid(row=0,column=10)

sel\_theme=StringVar()

sel\_theme.set("DARK")

theme=tk.OptionMenu(base, sel\_theme , "DARK", "LIGHT")

theme.grid(row=0,column=12)

theme\_button=tk.Button(base, text='apply theme' , command= lambda:theme\_change(sel\_theme.get()), width = 9).grid(row=0,column=13)

header=tk.Label(base, text = 'MODERN PERIODIC TABLE', font = ('algerian' , 24) , fg='purple').grid(row =2 , column =4 ,columnspan = 5)

com = [bc.sblock()]

for i in range(13):

exec(com[0][i])

com1=[bc.dblock()]

for i in range(40):

exec(com1[0][i])

com2=[bc.pblock()]

for i in range(37):

exec(com2[0][i])

com3=[bc.fblock()]

for i in range(28):

exec(com3[0][i])

def theme\_change(selected):

if selected=='LIGHT':

canvas.configure(background= "white")

base.configure(background= "white")

elif selected=="DARK":

canvas.configure(background= "black")

base.configure(background= "black")

def info(args):

global back,bas

cou=0

top = tk.Toplevel()

back=tk.Canvas(top, height = "500" , width = "500" , bg="black").pack()

bas=tk.Canvas(top,bg="black")

bas.place( relheight = "0.8" , relwidth = "0.8" , relx = "0.1" , rely = "0.1")

data.execute("desc modern\_pt;")

for i in data:

labl=tk.Label(bas,bg="black",fg="white",text = i[0]).grid(row=cou)

cou+=1

cou=0

data.execute("select \* from modern\_pt where Atomic\_number='%s';"%(args))

for k in data:

name=k[2]

#imgs(name)

for n in k:

l=tk.Label(bas,bg="black",fg="white",text = n).grid(row=cou,column=1)

cou+=1

def search():

global name

sboxout=sbox.get()

sboxout=str(sboxout.capitalize())

data.execute("select \* from modern\_pt where element='%s';"%(sboxout))

for i in data:

name=i[0]

if name==None:

return

else:

info(name)

name=None

root.mainloop()

**To generate the buttons for 118 elements in a periodic table, we have made a module called button command.py which will generate all buttons for elements in a periodic table block wise .**

* Every year a new property of element is discovered so this module enables us to update the information about the elements without changing the main program easily

import mysql.connector as sql

pd=sql.connect(host = 'localhost', user ='root', password = '\*\*\*\*')

if pd.is\_connected:

print("\_\_CONNECTED\_\_")

c1=pd.cursor()

c1.execute("use periodic;")

def sblock():

'''generates buttons for the sblock'''

c1.execute("select Atomic\_number, Element from modern\_pt where block = 's';")

l=[]

for i in c1:

l+=i

ename=''

bname=None

c=1

v=0

row=1

column=0

bsyntax=[]

while c<14:

bname=l[v]

ename=l[v+1]

bsyntax.append("""l=tk.Button(base, height = "3" , width = "9" , text = "%s" , command = lambda:info(%s),fg="black",bg="light blue").grid(row=%d, column=%d)"""%(ename,bname,row,column))

c+=1

v+=2

column+=1

if row==1:

column=0

row=2

if column==2:

row+=1

column=0

return bsyntax

def dblock():

''' generates buttons for the dblock'''

c1.execute("select Atomic\_number, Element from modern\_pt where Block = 'd';")

l=[]

for i in c1:

l+=i

ename=''

bname=None

c=1

v=0

row=4

column=2

bsyntax=[]

while c<41:

bname=l[v]

ename=l[v+1]

bsyntax.append('l=tk.Button(base, height = "3" , width = "9" , text = "%s" , command = lambda:info(%s),fg="black",bg="light green").grid(row=%d , column=%d)'%(ename,bname,row,column))

c+=1

v+=2

column+=1

if ename in ["Zinc","Cadmium","Mercury","Copernicium"]:

row+=1

column=2

return bsyntax

def pblock():

''' generates buttons for the pblock'''

c1.execute("select Atomic\_number, Element from modern\_pt where Block = 'p';")

l=[]

for i in c1:

l+=i

ename=''

bname=None

c=1

v=0

row=1

column=17

bsyntax=[]

while c<38:

bname=l[v]

ename=l[v+1]

bsyntax.append('l=tk.Button(base, height = "3" , width = "9" , text = "%s" , command = lambda:info(%s),fg="black",bg="yellow").grid(row=%d , column=%d)'%(ename,bname,row,column))

c+=1

v+=2

column+=1

if column==18:

row+=1

column=12

return bsyntax

def fblock():

'''generates buttons for the fblock'''

c1.execute("select Atomic\_number, Element from modern\_pt where Block = 'f';")

l=[]

for i in c1:

l+=i

ename=''

bname=None

c=1

v=0

row=11

column=2

bsyntax=[]

while c<29:

bname=l[v]

ename=l[v+1]

bsyntax.append('l=tk.Button(base, height = "3" , width = "9" , text ="%s" , command = lambda:info(%s),fg="black",bg="pink").grid(row=%d, column=%d)'%(ename,bname,row,column))

c+=1

v+=2

column+=1

if column==16:

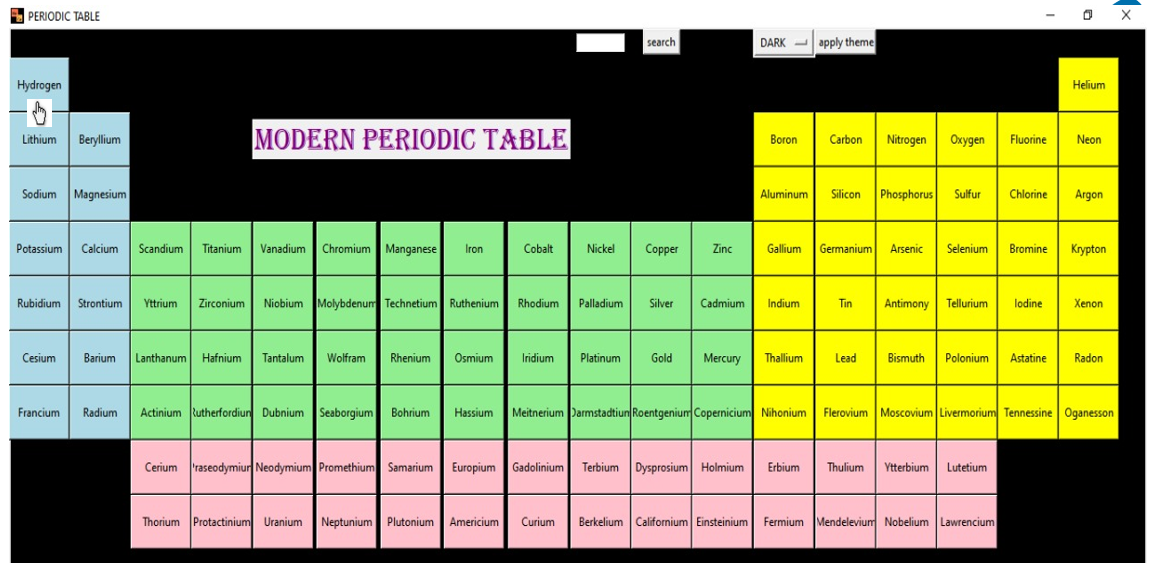
row=12

column=2

return bsyntax

**Information display of elements**

Click on the elements you want to see or search in the search box.



A picture containing text

Description automatically generated

DECLARATION

I, AATUR BORDIA , Roll No. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ a student of DPS, Navi Mumbai humbly submit that I have completed the project work as described in the report by my own skill and study as per the instruction of my teacher Ms. Madhurima Kashyap and that I have not copied the report or its any appreciable parts from any other literature in contravention of the academic ethics.

DATE: SIGNATURE OF STUDENT