**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **28/may/2020** | **Name:** | **Prashantha naik** |
| **Course:** | **Logic Design** | **USN:** | **4al17ec074** |
| **Topic:** | **1.Boolean equations for digital circuits. Combinational circuits: Conversion of MUX and Decoders to logic gates.**  **2.** **design of 7 segment decoder with common anode display** | **Semester & Section:** | **6th b** |
| **Github Repository:** | **prashanth\_course** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **28/may/2020** | **Name:** | **Prashantha naik** | |
| **Course:** | **python** | **USN:** | **4al17ec074** | |
| **Topic:** | **Application 5: Build a Desktop Database Application** | **Semester&Section:** | **6th b** | |
| **Git hub repository** | **prashanth\_couse** |  |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages.**  **Application 5: Build a Desktop Database Application**   * **I learnt about how to build a desktop database Application**   **Front end interface**  from tkinter import \*  import backend  def get\_selected\_row(event):  global selected\_tuple  index=list1.curselection()[0]  selected\_tuple=list1.get(index)  e1.delete(0,END)  e1.insert(END,selected\_tuple[1])  e2.delete(0,END)  e2.insert(END,selected\_tuple[2])  e3.delete(0,END)  e3.insert(END,selected\_tuple[3])  e4.delete(0,END)  e4.insert(END,selected\_tuple[4])  def view\_command():  list1.delete(0,END)  for row in backend.view():  list1.insert(END,row)  def search\_command():  list1.delete(0,END)  for row in backend.search(title\_text.get(),author\_text.get(),year\_text.get(),isbn\_text.get()):  list1.insert(END,row)  def add\_command():  backend.insert(title\_text.get(),author\_text.get(),year\_text.get(),isbn\_text.get())  list1.delete(0,END)  list1.insert(END,(title\_text.get(),author\_text.get(),year\_text.get(),isbn\_text.get()))  def delete\_command():  backend.delete(selected\_tuple[0])  def update\_command():  backend.update(selected\_tuple[0],title\_text.get(),author\_text.get(),year\_text.get(),isbn\_text.get())  window=Tk()  window.wm\_title("BookStore")  l1=Label(window,text="Title")  l1.grid(row=0,column=0)  l2=Label(window,text="Author")  l2.grid(row=0,column=2)  l3=Label(window,text="Year")  l3.grid(row=1,column=0)  l4=Label(window,text="ISBN")  l4.grid(row=1,column=2)  title\_text=StringVar()  e1=Entry(window,textvariable=title\_text)  e1.grid(row=0,column=1)  author\_text=StringVar()  e2=Entry(window,textvariable=author\_text)  e2.grid(row=0,column=3)  year\_text=StringVar()  e3=Entry(window,textvariable=year\_text)  e3.grid(row=1,column=1)  isbn\_text=StringVar()  e4=Entry(window,textvariable=isbn\_text)  e4.grid(row=1,column=3)  list1=Listbox(window, height=6,width=35)  list1.grid(row=2,column=0,rowspan=6,columnspan=2)  sb1=Scrollbar(window)  sb1.grid(row=2,column=2,rowspan=6)  list1.configure(yscrollcommand=sb1.set)  sb1.configure(command=list1.yview)  list1.bind('<<ListboxSelect>>',get\_selected\_row)  b1=Button(window,text="View all", width=12,command=view\_command)  b1.grid(row=2,column=3)  b2=Button(window,text="Search entry", width=12,command=search\_command)  b2.grid(row=3,column=3)  b3=Button(window,text="Add entry", width=12,command=add\_command)  b3.grid(row=4,column=3)  b4=Button(window,text="Update selected", width=12,command=update\_command)  b4.grid(row=5,column=3)  b5=Button(window,text="Delete selected", width=12,command=delete\_command)  b5.grid(row=6,column=3)  b6=Button(window,text="Close", width=12,command=window.destroy)  b6.grid(row=7,column=3)  window.mainloop()  **back end interface**  import sqlite3  def connect():  conn=sqlite3.connect("books.db")  cur=conn.cursor()  cur.execute("CREATE TABLE IF NOT EXISTS book (id INTEGER PRIMARY KEY, title text, author text, year integer, isbn integer)")  conn.commit()  conn.close()  def insert(title,author,year,isbn):  conn=sqlite3.connect("books.db")  cur=conn.cursor()  cur.execute("INSERT INTO book VALUES (NULL,?,?,?,?)",(title,author,year,isbn))  conn.commit()  conn.close()  view()  def view():  conn=sqlite3.connect("books.db")  cur=conn.cursor()  cur.execute("SELECT \* FROM book")  rows=cur.fetchall()  conn.close()  return rows  def search(title="",author="",year="",isbn=""):  conn=sqlite3.connect("books.db")  cur=conn.cursor()  cur.execute("SELECT \* FROM book WHERE title=? OR author=? OR year=? OR isbn=?", (title,author,year,isbn))  rows=cur.fetchall()  conn.close()  return rows  def delete(id):  conn=sqlite3.connect("books.db")  cur=conn.cursor()  cur.execute("DELETE FROM book WHERE id=?",(id,))  conn.commit()  conn.close()  def update(id,title,author,year,isbn):  conn=sqlite3.connect("books.db")  cur=conn.cursor()  cur.execute("UPDATE book SET title=?, author=?, year=?, isbn=? WHERE id=?",(title,author,year,isbn,id))  conn.commit()  conn.close()  connect()  insert("The Sun","John Smith",1918,913123132)  delete(3)  update(4,"The moon","John Smooth",1917,99999)  print(view())  print(search(author="John Smooth")) | | | |