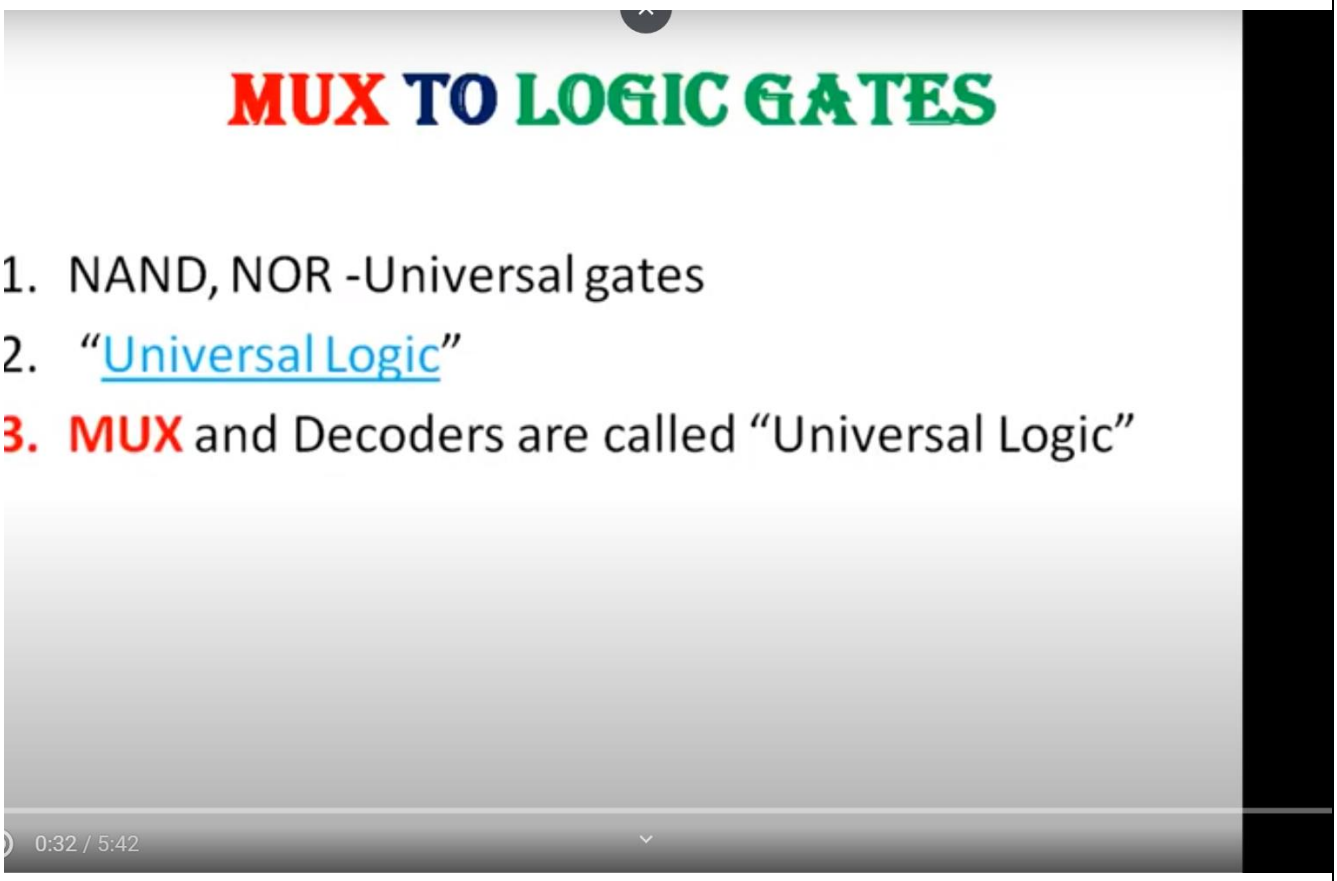
# DAILY ASSESSMENT FORMAT

| | | | |
|---|---|---|---|
| Date: | 28/5/2020 | Name: | SHILPA N |
| Course: | LOGIC DESIGN | USN: | 4AL16EC071 |
| Topic: | DAY 1 | Semester & Section: | 8th sem B sec |
| Github Repository: | Shilpan-test | | |

**FORENOON SESSION DETAILS**

**Image of session**




**Report –**

**BOOLEAN ALGEBRA OR LOGIC GATES**

The digital system understands logic level '0' or '1'. When designing the system, the designer look into cost of the circuit which depends n number of components and simple realization of circuit. To do this we require set of rules and is called Boolean algebra. In 1854 George Boole developed an algebraic system called Boolean algebra. Boolean algebra is a system of mathematic logic. It is defined with "set of elements, a set of operators and a number of axioms or postulates".

Set of elements – (0,1)

Two binary operator – OR (+) & AND(.)

Unary operator – NOT or invert

Axioms and Laws of Boolean algebra

Axioms or postulates of Boolean algebra are a logical expression on which we can build useful theorems.

AND operation

- 0.0=0
- 0.1=0
- 1.0=0
- 1.1=1

OR operation

- 0+0=0
- 0+1=1
- 1+0=1
- 1+1=1

NOT operation

- 0'=1
- 1'=0

Difference between Boolean algebra, ordinary algebra & binary number system

In Boolean algebra,
A+A= A->1+1=1
A. A=A->1.1=1

In Ordinary algebra,
A+A= 2A->1+1=2
A. A=$A^2$->1.1=$1^2$

In binary number system

1+1= (10)
1.1=1

Axioms or postulates

- X+0=X
- X.0=0
- X+1=1
- X.1=X
- X+X=X
- X.X=X
- X+X'=1
- X.X'=0
- (x')'=X

Identity element

OR and AND operations have some identity element.
When we do some operation between with 0 or 1 and other variable, we should get other variable.
X+0=x
0+x=x
Addition identity is zero
X.1=x
Multiplication identity is 1.

Laws of Boolean Algebra

1) Commutative Law
   X+y=y+x
   X.y=y.x

2) Associative Law
   X+(y+z)=(x+y)+z
   X.(y.z)=(x.y).z

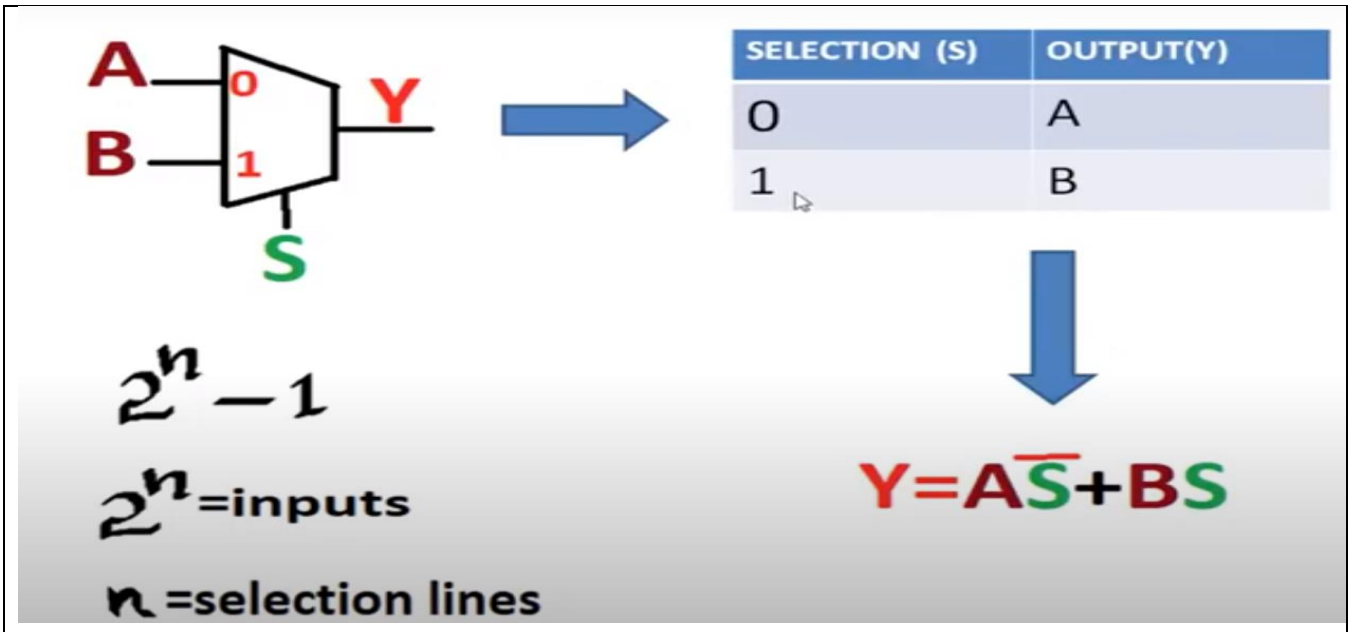
3) Distributive Law
   1. X(y+z) = xy+yz
   2. X+yz = (x+y) (x+z)
      = x.x +xz +xy +yz
      =x(1+z+y)+yz
      =x + yz

MUX to Logic Gates

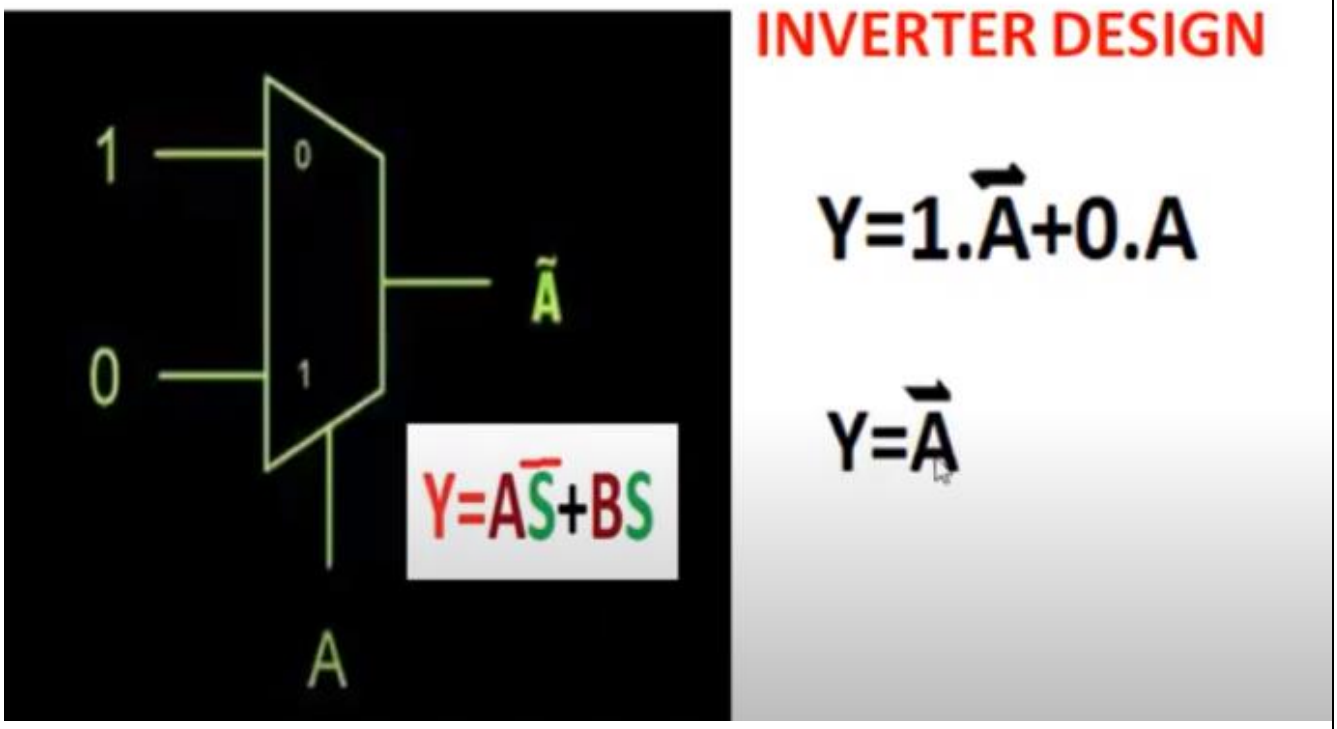
1. NAND, NOR – universal gates
2. MUX and Decoders are called universal logic

Multiplexer is device which select multiple input lines and gives single the output line.

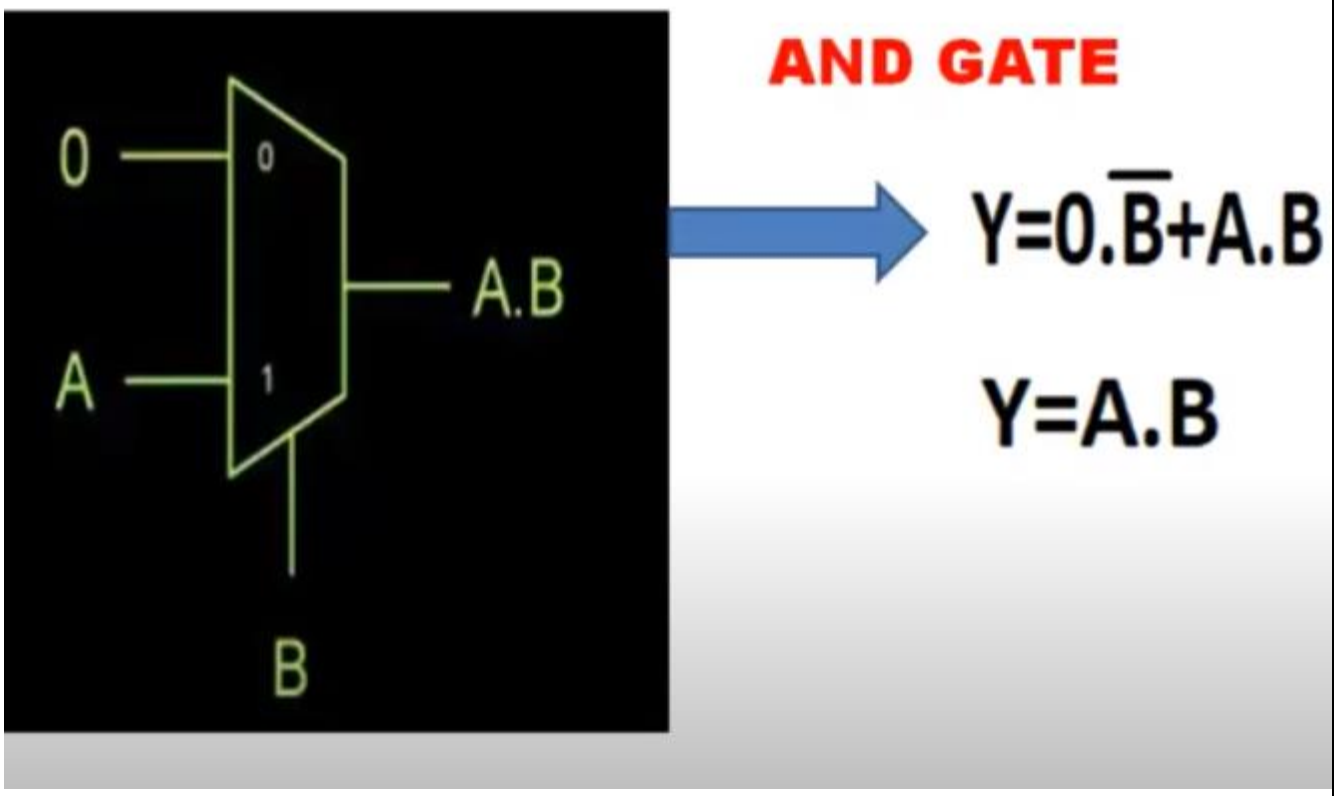
2:1 MUX-



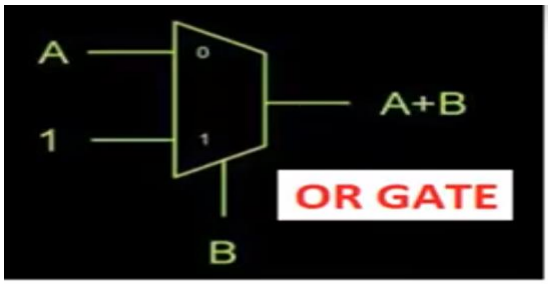

INVERTER design with the help of MUX

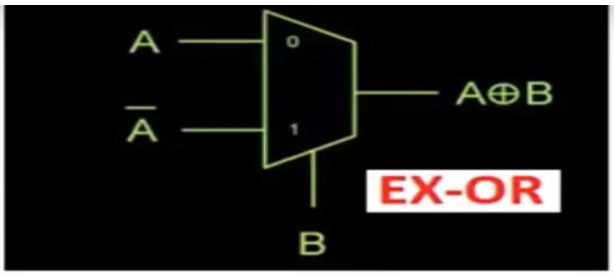


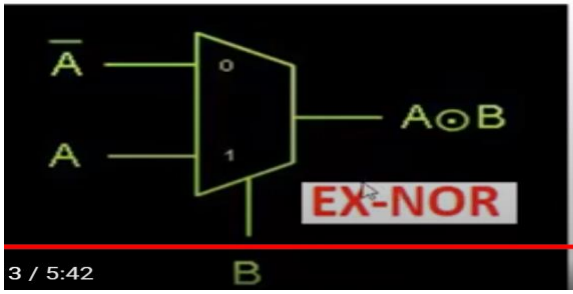

AND gate design




OR, E-XOR, EX-NOR gate design



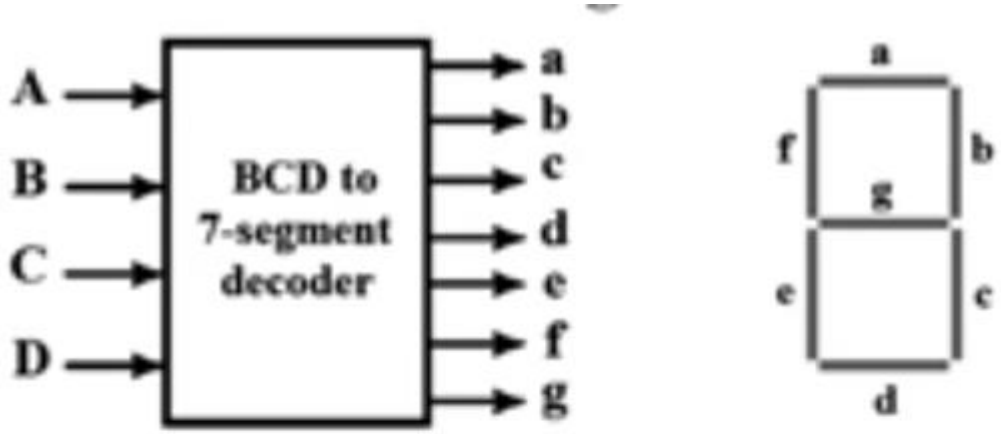




3 / 5:42

BCD TO 7-SEGMENT ENCODER

BCD to 7-segment encoder has 4input line and 7output line.

The decoder is an essential component in BCD to seven segment decoder. A decoder is nothing but a combinational logic circuit mainly used for converting a BCD to an equivalent decimal number. It can be a BCD to seven segment decoders. A combinational logic circuit can be built with logic gates which include inputs as well as outputs. The output of this circuit mainly lies in the current condition of the inputs. The best examples of this circuit are multiplexers, demultiplexers, adders, subtractors, encoders, decoders, etc. The truth table and its display is as shown below,







| A | B | C | D | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

BONUS SESSION-WHY YOU SHIULD WRITE TOUR OWN RESUME?

The presenter was the session was Sajjad Ahmed, HR, talent management, Capgemini. Resume is a French word which means summary or Bio data or curriculum vitae. The flow of the lecture is discussed using resume word. The basis is the job requirement should be there in the company. Job description might be different interviewer. Sourcing team their responsibility to ensure that the job description is published either through newspaper etc... Profile evaluation also come under basics.

The first letter r is relevant information- we should put personal details like name as on certificate, phone number, correct email, city of residence. Objective/synopsis-freshers can put motivation as their objective, profile summary of experience. Education- name of the college, city, year of passing, %marks for freshers. For experiencers organizations- name, location, start and end date of employment, job designation, responsibilities, projects, acknowledgements. Skills- primary skill, secondary skills. Training & certifications.

The second letter is easy to read- put a picture which is very clear. The font should be proper. Don’t do mistakes. Tip is Have someone proof read of your resume.

The third letter is specific details- mention only specifics in the resume. Always put your contribution for the role instead of just description. Freshers can relate this by giving examples of the activities he has done. Tip is resume is a medium for advertising yourself.

The fourth letter is use keywords- match rate is jd requires some one should hv MBA, so instead of putting the expansion along with that put MBA in braces. Tip is content & format is the key

The fifth letter is Match the requirement- always create your resume for similar to the Job Description. Explained about job title, duties, results, education, skills, location. Tip is different resume for different positions.

The last letter is Evaluable- mention the character instead of logos, and save the resume in pdf, name the pdf with your name properly. Don’t have use tables in the resume because some company uses bot which cannot read table. Tip is always trying to make resume similar to job description.

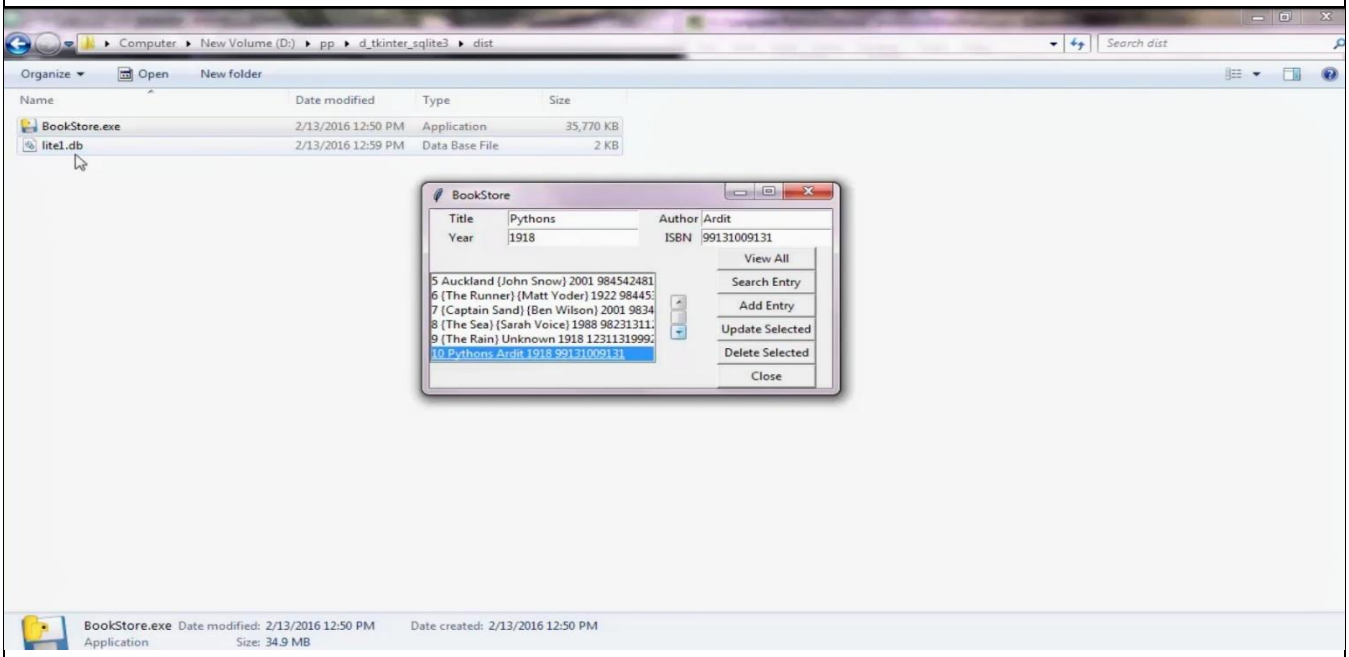
concludes session by saying that “you are not your resume; you are your work”.

| | | | |
|---|---|---|---|
| **Date:** | **28/05/2020** | **Name:** | **Shilpa N** |
| **Course:** | **Python** | **USN:** | **4AL16EC071** |
| **Topic:** | **Day 10** | **Semester & Section:** | **8th sem,'B' section** |

**AFTERNOON SESSION DETAILS**

In this section you're going to learn how to build a real-world program that allows the user to store information about books, so I have already builded a program and I'll show you step by step how you can do the same. I have the program here and it's an executable So apart from learning how to build the a dot py program ,you'll also to learn how to build executable files and that run on windows and dot app files that run on Mac So I called the program Bookstore, and this was built mainly with tkinter which is a graphical user interface library and the SQLite3 library which is a library to interact with SQLite database. So what we got here are four entries where you can enter a new book record, so let's say you've got a new book in your store and you want to add it in your database.And here you can see all the books that have been added to the program. You can add a new book, let's say pythons for title.Ardit, quickly and 1918. You have the ISBN number which is a unique identification number for books. It's quite long.

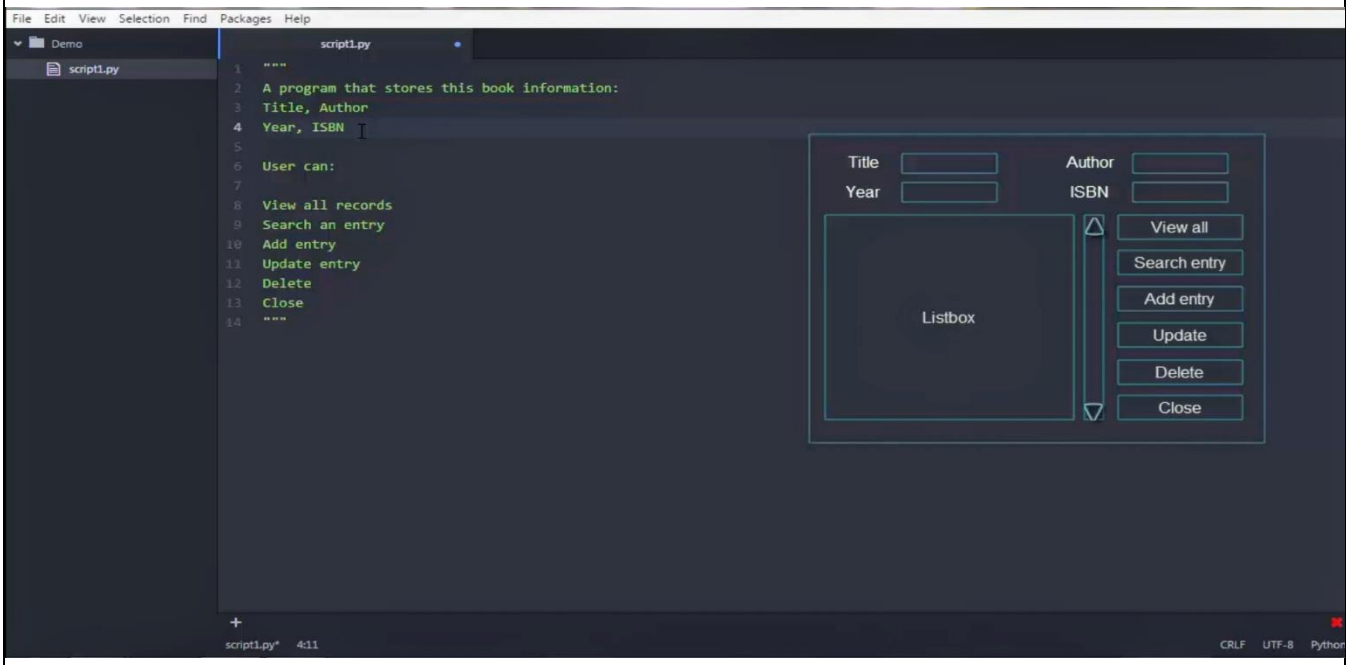
When you click on the entry you get all the fields displayed in each of the text box here.So, view all again and you can also update and delete the records.And finally you can close the window Bookstore program, and specifically we're going to work on the graphical user interface because you know this program has two parts so to say. So it has a backend and it has frontend.The backend is the sqlite3

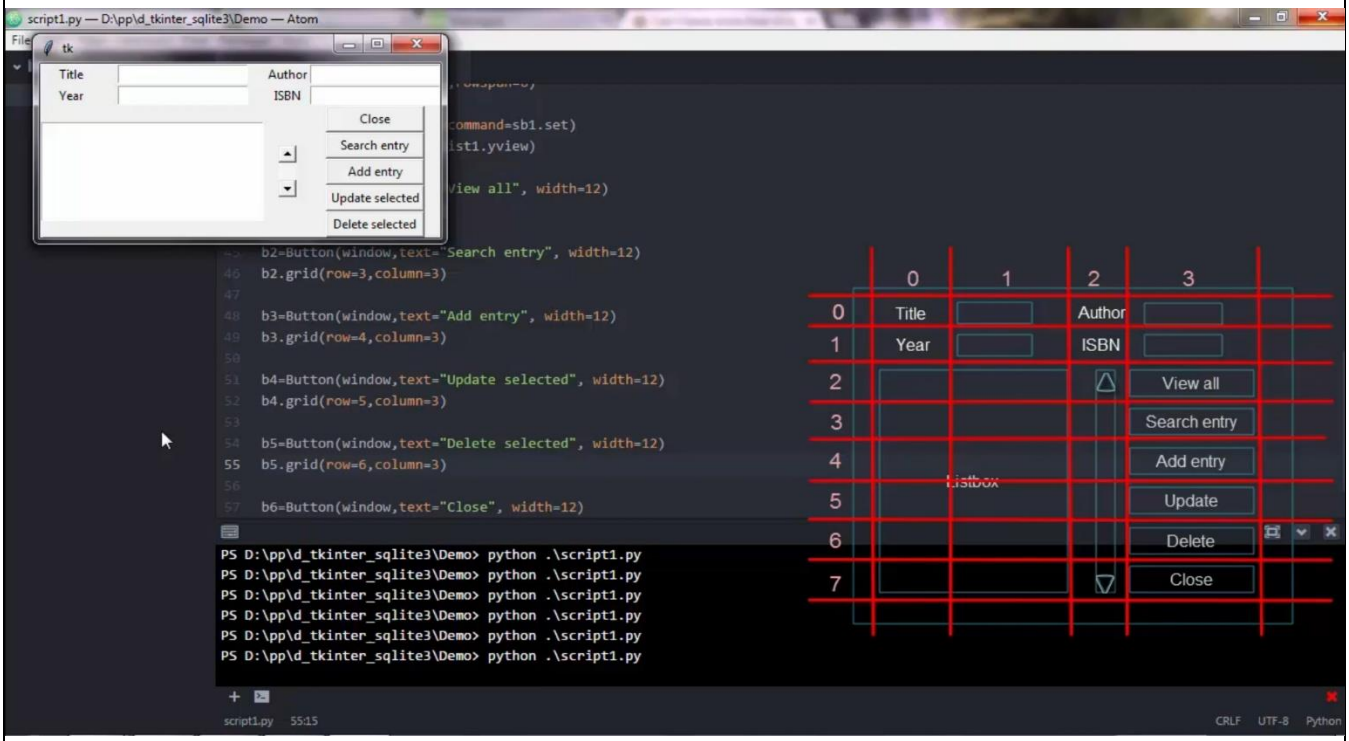



database code, so the code that interacts with the database and the frontend is a code of the Tkinter

library, so the code that builds up all these buttons, these entries, this text, this list box, and this scroll bar.

**Front end:**



**Backend:**

frontend.py backend.py

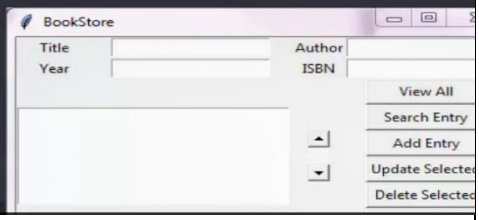
```
import sqlite3

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()

def view():
    conn=sqlite3.connect("books.db")
    cur=conn.cursor()
    cur.execute("SELECT * FROM book")
    rows=cur.fetchall()
    conn.close()
    return rows


connect()
insert("The sea","John Tablet")
```



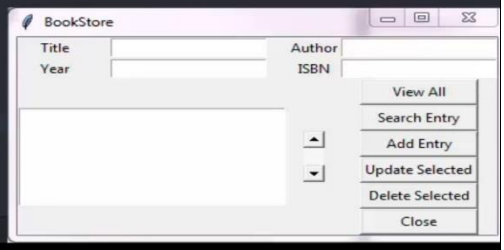

frontend.py backend.py

```
    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,)

connect()
insert("The Sun","John Smith",1918,913123132)
delete(3)
```




```
PS D:\pp\d_tkinter_sqlite3\Demo> python .\backend.py
[(1, 'The sea', 'John Tablet', 1918, 913123132), (2, 'The sea', 'John Tablet', 1918, 913123132)]
PS D:\pp\d_tkinter_sqlite3\Demo> python .\backend.py
[(1, 'The sea', 'John Tablet', 1918, 913123132), (2, 'The sea', 'John Tablet', 1918, 913123132), (3, 'The Earth', 'John Smith', 1918, 9131
23132)]
[(3, 'The Earth', 'John Smith', 1918, 913123132)]
PS D:\pp\d_tkinter_sqlite3\Demo> python .\backend.py
[(1, 'The sea', 'John Tablet', 1918, 913123132), (2, 'The sea', 'John Tablet', 1918, 913123132), (4, 'The Sun', 'John Smith', 1918, 913123
132)]
[(4, 'The Sun', 'John Smith', 1918, 913123132)]
PS D:\pp\d_tkinter_sqlite3\Demo>
```