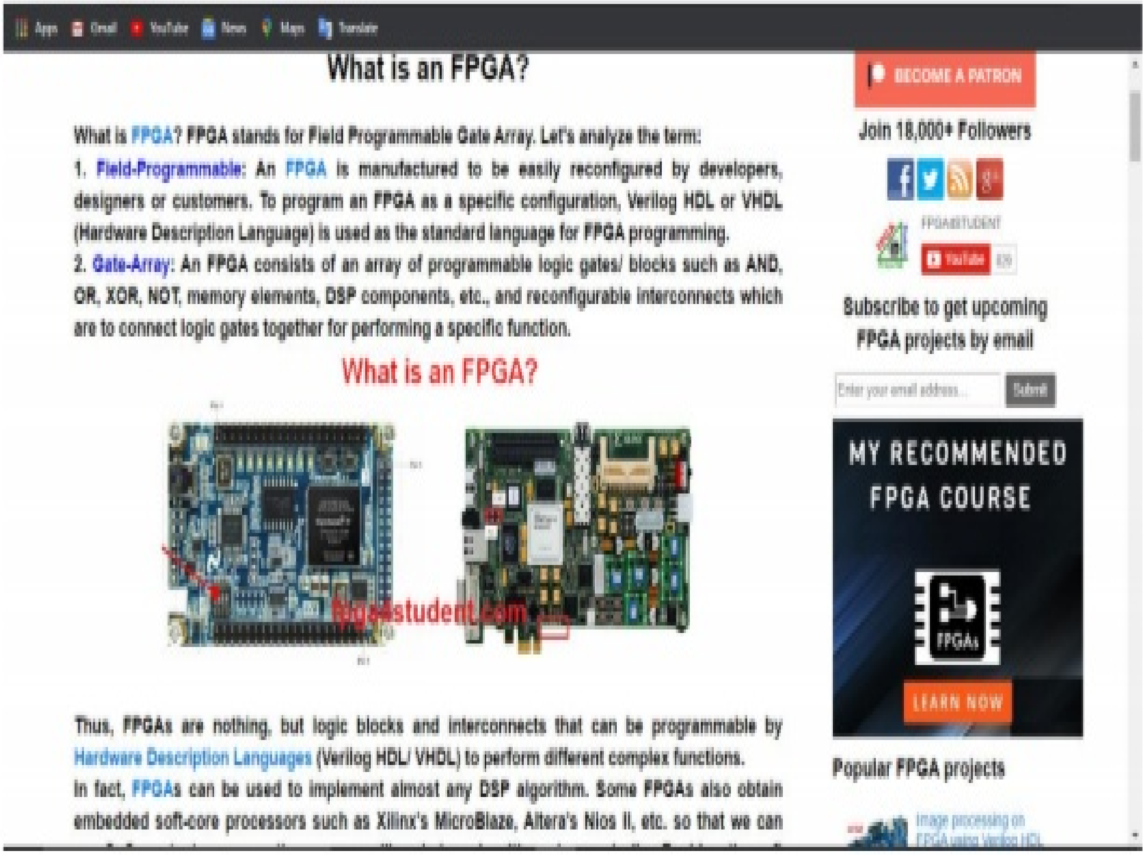
DAILY ASSESSMENT FORMAT

Date:	05-06-2020	Name:	M V Ramya
Course:	logic design	USN:	4AL17EC045
Topic:	FPGA	Semester & Section:	6th sem, A sec
Github Repository:	M V Ramya-045		

FORENOON SESSION DETAILS



What is an FPGA?

What is FPGA? FPGA stands for Field Programmable Gate Array. Let's analyze the term:

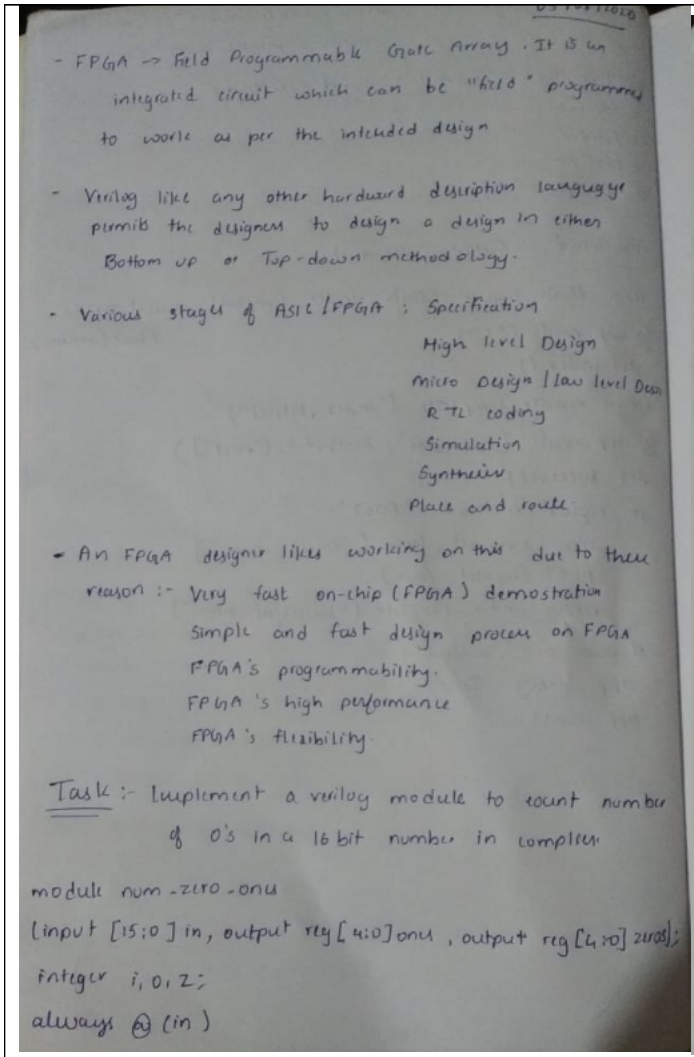
1. Field-Programmable: An FPGA is manufactured to be easily reconfigured by developers, designers or customers. To program an FPGA as a specific configuration, Verilog HDL or VHDL (Hardware Description Language) is used as the standard language for FPGA programming.

2. Gate-Array: An FPGA consists of an array of programmable logic gates/ blocks such as AND, OR, XOR, NOT, memory elements, DSP components, etc., and reconfigurable interconnects which are to connect logic gates together for performing a specific function.

Thus, FPGAs are nothing, but logic blocks and interconnects that can be programmable by Hardware Description Languages (Verilog HDL/ VHDL) to perform different complex functions.

In fact, FPGAs can be used to implement almost any DSP algorithm. Some FPGAs also obtain embedded soft-core processors such as Xilinx's MicroBlaze, Altera's Nios II, etc. so that we can

BECOME A PATRON
Join 18,000+ Followers
FPGA4STUDENT
YouTube
Subscribe to get upcoming FPGA projects by email
Enter your email address...
Submit
MY RECOMMENDED FPGA COURSE
FPGAs
LEARN NOW
Popular FPGA projects
Image processing on FPGA using Verilog HDL



- FPGA -> Field Programmable Gate Array. It is an integrated circuit which can be "field" programmed to work as per the intended design

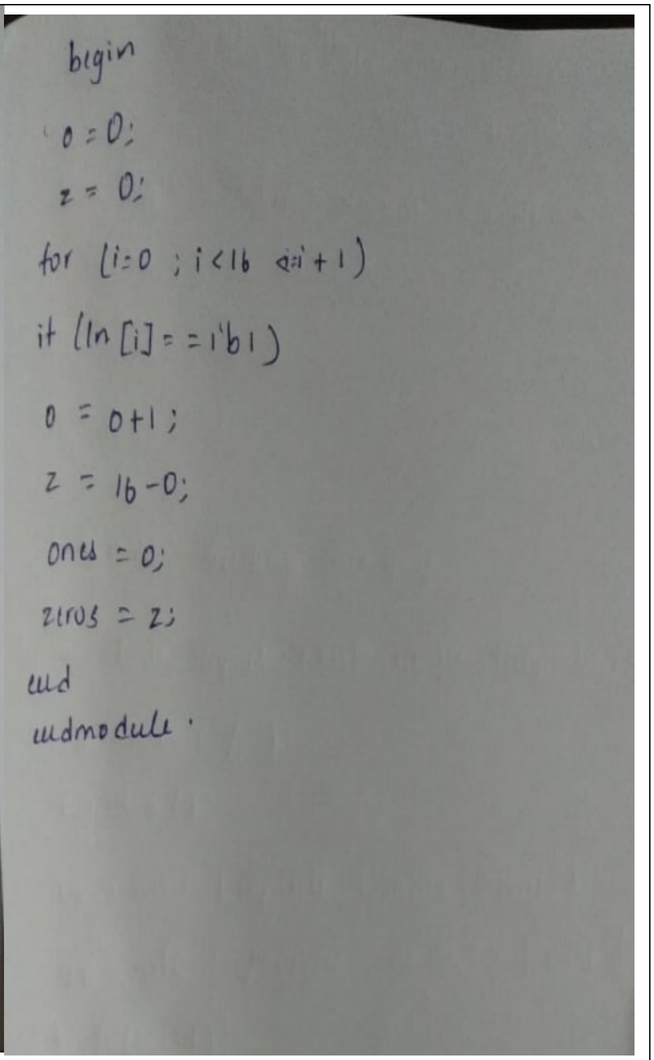
- Verilog like any other hardward description language permits the designers to design a design in either Bottom up or Top-down methodology.

- Various stages of ASIC/FPGA : Specification
  High level Design
  Micro Design/Low level Design
  RTL coding
  Simulation
  Synthesis
  Place and route

- An FPGA designer likes working on this due to these reason :- Very fast on-chip (FPGA) demostration
  Simple and fast design process on FPGA
  FPGA's programmability.
  FPGA's high performance
  FPGA's flexibility.

Task :- Implement a verilog module to count number of 0's in a 16 bit number in complier.

```
module num-zero-ones
(input [15:0] in, output reg [4:0] ones , output reg [4:0] zeros);
integer i, o, z;
always @ (in )
```



```
begin
o = 0;
z = 0;
for (i=0 ; i<16 ; i=i+1)
if (In[i] = = 1'b1)
o = o+1;
z = 16-o;
ones = o;
zeros = z;
end
endmodule.
```

Date:05june2020

Name: MV Ramya

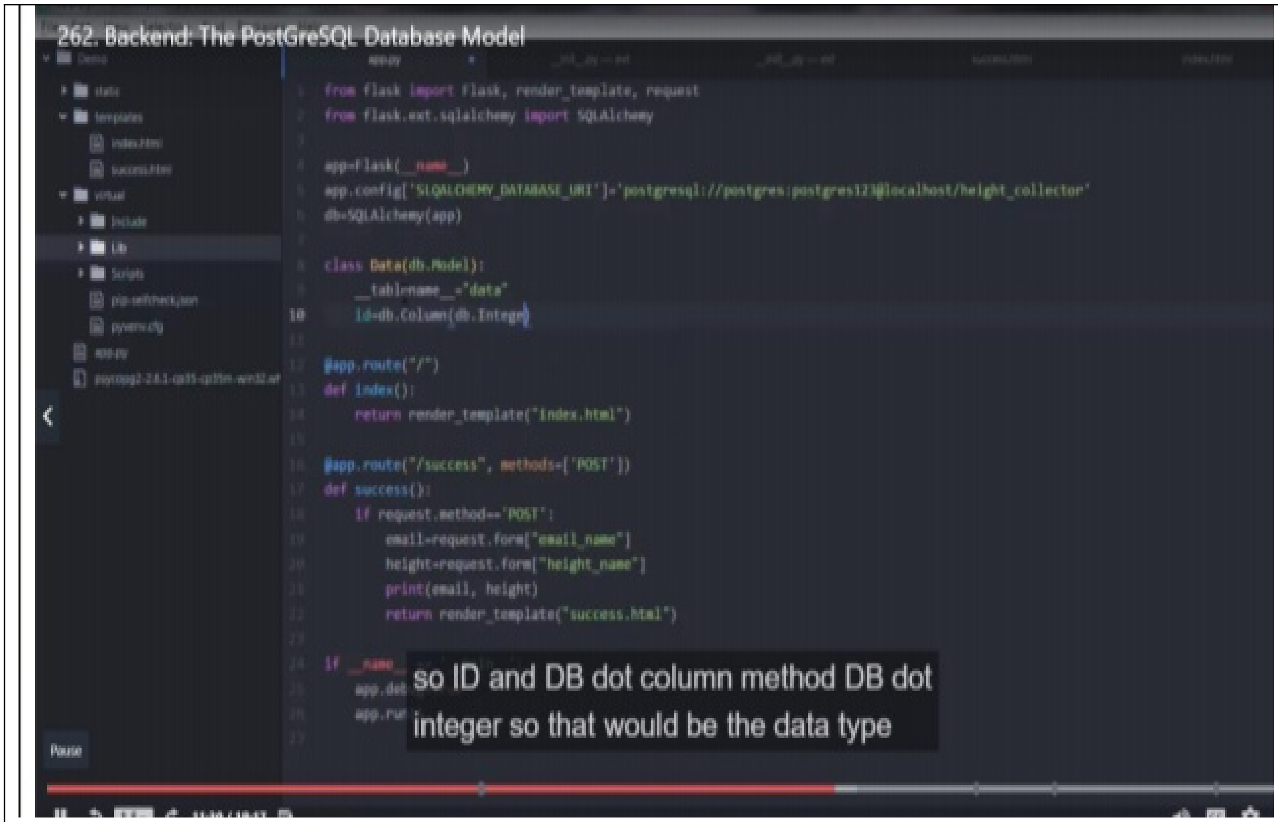
Course: python

USN:4AL17EC045

Topic:web app with PostGreSQL and Flask

Semester & Section: 6th sem Asec

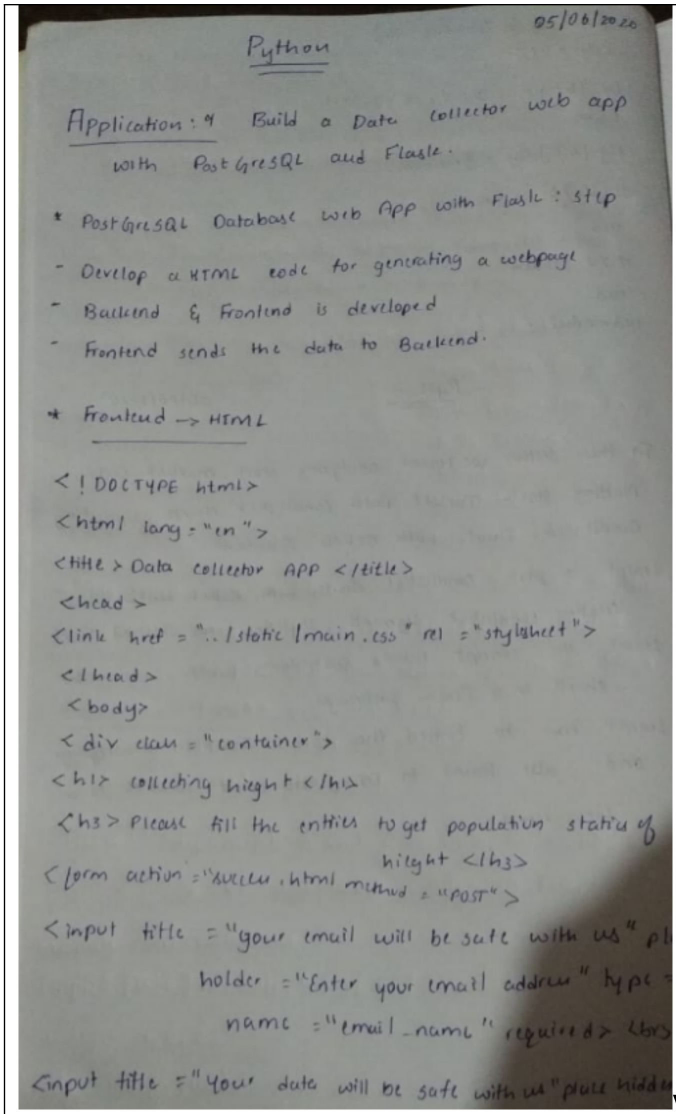
AFTERNOON SESSION DETAILS
Image of session



262. Backend: The PostGreSQL Database Model

```
from flask import Flask, render_template, request
from flask.ext.sqlalchemy import SQLAlchemy

app=Flask(__name__)
app.config['SLQALCHEMY_DATABASE_URI']='postgresql://postgres:postgres123@localhost/height_collector'
db=SQLAlchemy(app)

class Data(db.Model):
    __tabl=name__="data"
    id=db.Column(db.Intege)

@app.route("/")
def index():
    return render_template("index.html")

@app.route("/success", methods=['POST'])
def success():
    if request.method=='POST':
        email=request.form["email_name"]
        height=request.form["height_name"]
        print(email, height)
        return render_template("success.html")

if __name_
    app.de
    app.ru
```

so ID and DB dot column method DB dot integer so that would be the data type



05/06/2020

Python

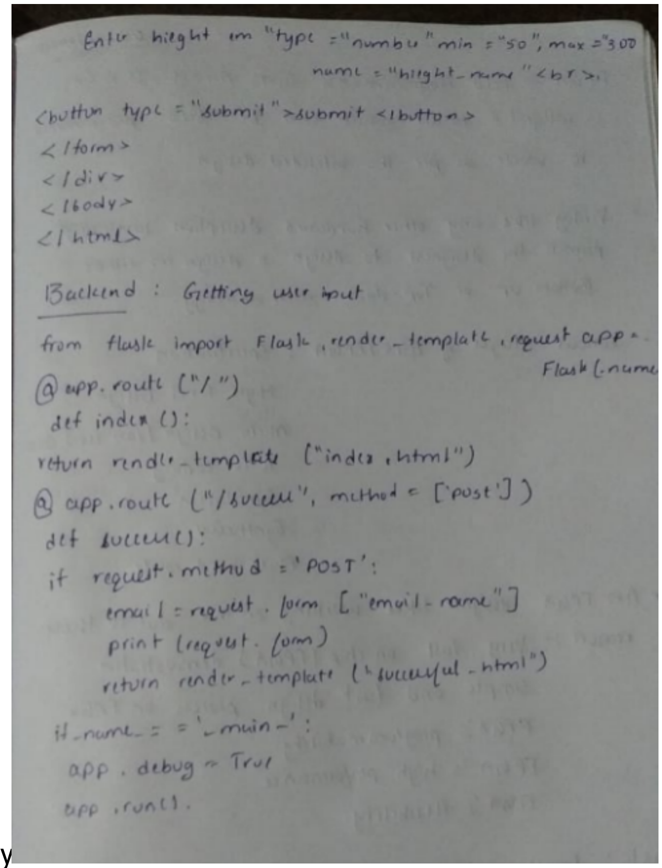
Application : 4 Build a Data collector web app with PostGreSQL and Flask.

- * PostGreSQL Database web App with Flask : step
- Develop a HTML code for generating a webpage
- Backend & Frontend is developed
- Frontend sends the data to Backend.

* Frontend -> HTML

```
<!DOCTYPE html>
<html lang="en">
<title> Data collector APP </title>
<head >
<link href = "../static/main.css" rel ="stylsheet">
<lhead>
<body>
<div class = "container">
<h1> collecting hieght </h1>
<h3> Please fill the entties to get population static of
                                   hieght </h3>
<form action ="succes.html method = "POST">
<input title = "your email will be safe with us" pl
               holder ="Enter your email address" type =
                  name ="email_name" required> <br>
<input title ="Your data will be safe with us" place hidde
```

y



```
     Enter hieght cm "type ="number" min ="50", max ="300
                                  name ="hieght_name" <br>.
<button type ="submit">submit </button>
</form>
</div>
</body>
</html>
```

Backend : Getting user input

```
from flask import Flask, render_template, request app =
                                               Flask (_name
@app.route ("/")
 def index ():
return render_template ("index.html")
@ app.route ("/success", method = ['post'])
def success():
 if request.method ='POST':
     email = request. form ["email_name"]
     print (request. form)
     return render_template ("successful_html")
 if_name_ = = '_main_':
  app . debug = True
  app .run().
```