**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **5/June/2020** | **Name:** | **Prashantha naik** |
| **Course:** | **DIGITAL DESIGN USING HDL** | **USN:** | **4al17ec074** |
| **Topic:** | 1. **Verilog Tutorials and practice programs** 2. **Building/ Demo projects using FPGA** | **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.**  **Verilog Tutorials and practice programs**   * **Learnt the basics of Verilog hdl** * **Learnt how write the program in Verilog in all modeling style**         **Implement a Verilog module to count number of 0’s in a 16-bit number.**  module num\_zeros\_for(  input [15:0] A,  output reg [4:0] ones  );  integer i;  always@(A)  begin  ones = 0;  for(i=0;i<16;i=i+1)  if(A[i] == 0'b1)  ones = ones + 1;  end  endmodule  output  Input = "1010\_0010\_1011\_0010" => Output = "01001" ( 9 in decimal)  Input = "0011\_0110\_1000\_1011" => Output = "01000" (8 in decimal) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **5/June/2020** | **Name:** | **Prashantha naik** | |
| **Course:** | **Python** | **USN:** | **4al17ec074** | |
| **Topic:** | **Application 10: Build a Data Collector Web App with PostgreSQL and Flask** | **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.**  **app.py**  **from flask import Flask, render\_template, request**  **from flask.ext. sqlalchemy import SQLAlchemy**  **from send\_email import send\_email**  **from sqlalchemy.sql import func**  **app=Flask(\_\_name\_\_)**  **app.config['SQLALCHEMY\_DATABASE\_URI']='postgresql://postgres:postgr es123@localhost/height\_collector'**  **db=SQLAlchemy(app)**  **class Data(db.Model):**  **\_\_tablename\_\_="data"**  **id=db.Column(db.Integer, primary\_key=True)**  **email\_=db.Column(db.String(120), unique=True)**  **height\_=db.Column(db.Integer)**    **def \_\_init\_\_(self, email\_, height\_):**  **self.email\_=email\_**  **self.height\_=height\_**  **@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)**  **if db.session.query(Data).filter(Data.email\_ ==**  **email).count()== 0:**  **data=Data (email, height)**  **db.session.add(data)**  **db.session.commit()**  **average\_height=db. session. Query(func.avg (Data.height\_)).scalar() average\_height=round(average\_height, 1)**  **count = db.session.query(Data.height\_).count()**  **send\_email(email, height, average\_height, count) print(average\_height) return render\_template("success.html")**  **return render\_template('index.html', text="Seems like we got something from that email once!")**  **if \_\_name\_\_ == '\_\_main\_\_':**  **app.debug=True**  **app.run(port=5005)** | | | |