**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **04/06/2020** | **Name:** | **Shilpa C** |
| **Course:** | **Digital design using HDL** | **USN:** | **4al17ec086** |
| **Topic:** | 1. Hardware modelling using verilog 2. FPGA and ASIC Interview questions | **Semester & Section:** | **6th ,B sec** |
| **Github Repository:** | **shilpa-c** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  Hardware modelling using Verilog:  Learnt about the Verilog hardware description language.•  Understood the difference between behavioral and structural design styles.  • Learnt to write test benches and analyze simulation results  .• Learnt to model combinational and sequential circuits  .• Distinguish between good and bad coding practices  .• Case studies with some complex designs.  WhatsApp Image 2020-06-04 at 18.38.16.jpeg  WhatsApp Image 2020-06-04 at 18.38.16 (1).jpeg  WhatsApp Image 2020-06-04 at 18.38.16 (1).jpeg |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:04/06/2020** |  | **Name:** | **Shilpa.c** | |
| **Course:python** |  | **USN:** | **4al17ec086** | |
| **Topic: Application 10: Build a Data Collector Web App with PostGreSQL and Flask** |  | **Semester & Section:** | **6th .b sec** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages.**   1. Install PostgreSQL to local machine 2. Install Heroku CLI 3. Create python virtual environment for the project 4. Create a sample code with Flask to check 5. Create database 6. Create configurations 7. Database migration 8. Finish the code   **1. Install PostgreSQL to local machine**  Follow this step if you already haven’t installed PostgreSQL on your machine.  Install PostgreSQL in Linux using the command,  sudo apt-get install postgresql postgresql-contrib  Now create a superuser for PostgreSQL  sudo -u postgres createuser --superuser ***name\_of\_user***  And create a database using created user account  sudo -u ***name\_of\_user*** createdb ***name\_of\_database***  You can access created database with created user by,  psql -U ***name\_of\_user*** -d ***name\_of\_database***  *Note that if you created the***name\_of\_user***and***name\_of\_database***as your user name on your machine, you can access that database with that user with psql command.*  **Create a sample code with Flask to check**  For using Flask, first you need to install Flask. (Make sure that you have activated the virtual environment)  pip install Flask  Now create a file named ***app.py***in ***books\_server***directory and put below code to test Flask before we move into real application development  to execute above code run  python app.py  or  FLASK\_APP=app.py flask run  you can check the deployed server on [***http://127.0.0.1:5000/***](http://127.0.0.1:5000/)  *Here we have created 3 methods with 3 routes.*  *1. first method is root URL.*  [**http://127.0.0.1:5000/**](http://127.0.0.1:5000/)*will return***Hello World!***on your browser.*  *another 2 methods are used to get inputs. here we have used 2 types of data input methods.*  [**http://127.0.0.1:5000/name**](http://127.0.0.1:5000/)[*/Twilight*](https://www.goodreads.com/book/show/41865.Twilight)*will return***name : Twilight**  [**http://127.0.0.1:5000/details?author=***Stephenie Meyer***&published=***2006*](http://127.0.0.1:5000/details?author=Stephenie%20Meyer&published=2006) *will return***Author : Stephenie Meyer, Published: 2006**  *This***app.py***file won’t be used in the project. We will create***app.py***later as required for the project*  From here let’s move to create our book details storing application.  **Create database**  First create the database we need here for our application named ***books\_store***  sudo -u ***name\_of\_user*** createdb books\_store  Now you can check the created database with,  psql -U ***name\_of\_user*** -d books\_store  You should log into***books\_store***data base if above command was success.  **Create configurations**  We need to define configurations for deploying environments. create a file named ***config.py***with below code.  According to created configurations set ***“APP\_SETTINGS”*** environment variable by running this in the terminal  export APP\_SETTINGS="config.DevelopmentConfig"  Also add ***“DATABASE\_URL”***to environment variables. In this case our database URL is based on the created database. So, export the environment variable by this command in the terminal,  export DATABASE\_URL="postgresql://localhost/books\_store" | | | |