**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **28/05/2020** | **Name:** | **Dhavala** |
| **Course:** | **Logic Design** | **USN:** | **4AL17EC027** |
| **Topic:** | * **Boolean equations for digital circuits. Combinational circuits: Conversion of MUX and Decoders to logic gates** * **design of 7 segment decoder with common anode display** | **Semester & Section:** | **6TH SEM & A Section** |
| **Github Repository:** | **Dhavala27** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
|  |

|  |
| --- |
| **Report** |

**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **28/05/2020** | **Name:** | **Dhavala** |
| **Course:** | **PYTHON** | **USN:** | **4AL17EC027** |
| **Topic:** | * **Application 5: Build a Desktop Database Application** | **Semester & Section:** | **6TH SEM & A Section** |
| **Github Repository:** | **Dhavala27** |  |  |

|  |
| --- |
| **AFTERNOON SESSION DETAILS** |
|  |

|  |
| --- |
| **Report** |
| DATABASE  A Database is defined as a structured set of data. So, in SQL the very first step to store  the data in a well-structured manner is to create a database. The CREATE DATABASE  statement is used to create a new database in SQL.  Syntax:  CREATE DATABASE database\_name;  database\_name: name of the database.  Example Query:This query will create a new database in SQL and name the database as my\_database. CREATE DATABASE my\_database;  CREATE TABLE  We have learned above about creating databases. Now to store the data we need a table to do that. The CREATE TABLE statement is used to create a table in SQL. We know that a table comprises of rows and columns. So while creating tables we have to provide all the information to SQL about the names of the columns, type of data to be stored in columns, size of the data etc. Let us now dive into details on how to use CREATE TABLE statement to  create tables in SQL.  Syntax:  CREATE TABLE table\_name  (  column1 data\_type(size),  column2 data\_type(size),  column3 data\_type(size),  ....  );  Fixing the Bug  We haven’t noticed, the program has a bug. When the listbox is empty and the user clicks the listbox, an *IndexError* is generated in the terminal:    Why does this error happen?  Well, everything starts with the user clicking on the listbox. Clicking the listbox executes the following code:  List1.blind(‘<<ListboxSelect>>’,get\_selected\_row)  That code calls the get\_selected\_row function:  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])  Since the listbox is empty, list1.curselection() will be an empty list with no items. Trying to access the first item on the list with [0] in line 3 will throw an error, because there is no first item in the list.  **Solution**  def get\_selected\_row(event):  try:  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])  except IndexError:  pass  **Explanation**  The error was fixed by simply implementing a try and except block. When the get\_selected\_row function is called, Python will execute the indented block under try. If there is an IndexError, none of the lines under try will be executed; the line under except will be executed, which is pass. The pass statement means "do nothing". Therefore, the function will do nothing when there's an empty listbox. |