**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **29 may 2020** | **Name:** | **Shreya poojary** |
| **Course:** | **python** | **USN:** | **4al16ec074** |
| **Topic:** | **Application to Build a desktop database application** | **Semester & Section:** | **8-B** |
| **Github Repository:** | **Shreya-test** |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Image of session** | | | | | | |
|  |  | |  | |  | | |
|  | |  | |  | |
|  |  | |  | |  | | |
|  | | | | | | | |
| **REPORT:**  **CONNECT BACKEND TO FRONTEND**  Python has been the most trending programming language used for object oriented programming. With python you can run simple statement over and over again without having to compile a whole program of which it's output functionality is superb.  Of course, Python is an interactive programming which has a diverse range of options for GUI (Graphical User Interface) framework (help developers create GUI applications in an easy and secure manner).  There are exactly two ways to perform what you are looking for, i.e., to insert data from data services of your backend into your frontend layout as content.   1. Server Side Rendering- This is how most and traditional websites specifically those are based on some CMS technology uses. The idea is to have some HTML template files with some placeholder for content that are later pushed by the backend program before sending to client browsers. In Python, you can have Jinja template framework at your disposal. You can wrap up variables into double braces and later providing values for each variable and compile them to static HTML file to render them on browser. 2. Client Side Rendering (Ajax)- This is the approach taken by progressive web applications. You will most probably use some JavaScript framework for the View and use some of its data services or another third party library to fetch data from Backend API endpoint and then update the view accordingly. Some of popular frameworks are- React, Vue (VDOM) or Angular, Mithril etc. 3. Hybrid- You may also however combine these two technique by using some server side rendering with a JavaScript plugin library with Ajax support such as jQuery.   **USER INTERFACE DESIGN**  There are many graphical user interface (GUI) toolkits that you can use with the Python programming language. The big three are Tkinter, wxPython, and PyQt. Each of these toolkits will work with Windows, macOS, and Linux, with PyQt having the additional capability of working on mobile.  A graphical user interface is an application that has buttons, windows, and lots of other widgets that the user can use to interact with your application. A good example would be a web browser. It has buttons, tabs, and a main window where all the content loads.  Interfaces play an important role in software engineering. As an application grows, updates and changes to the code base become more difficult to manage. More often than not, you wind up having classes that look very similar but are unrelated, which can lead to some confusion. In this tutorial, you’ll see how you can use a **Python interface** to help determine what class you should use to tackle the current problem.  nterfaces in Python are handled differently than in most other languages, and they can vary in their design complexity. By the end of this tutorial, you’ll have a better understanding of some aspects of Python’s data model, as well as how interfaces in Python compare to those in languages like Java, C++, and Go.  At a high level, an interface acts as a **blueprint** for designing classes. Like classes, interfaces define methods. Unlike classes, these methods are abstract. An **abstract method** is one that the interface simply defines. It doesn’t implement the methods. This is done by classes, which then **implement** the interface and give concrete meaning to the interface’s abstract methods.  Python’s approach to interface design is somewhat different when compared to languages like [Java](https://realpython.com/oop-in-python-vs-java/), Go, and C++. These languages all have an interface keyword, while Python does not. Python further deviates from other languages in one other aspect. It doesn’t require the class that’s implementing the interface to define all of the interface’s abstract methods.   1. FIXING A BUG 2. defget\_selected\_row(event): 3. try: 4. globalselected\_tuple 5. index=list1.curselection()[0] 6. selected\_tuple=list1.get(index) 7. e1.delete(0,END) 8. e1.insert(END,selected\_tuple[1]) 9. e2.delete(0,END) 10. e2.insert(END,selected\_tuple[2]) 11. e3.delete(0,END) 12. e3.insert(END,selected\_tuple[3]) 13. e4.delete(0,END) 14. e4.insert(END,selected\_tuple[4]) 15. exceptIndexError: 16. pass  Explanation The error was fixed by simply implementing a try and except block. When theget\_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. | | | | | | | |