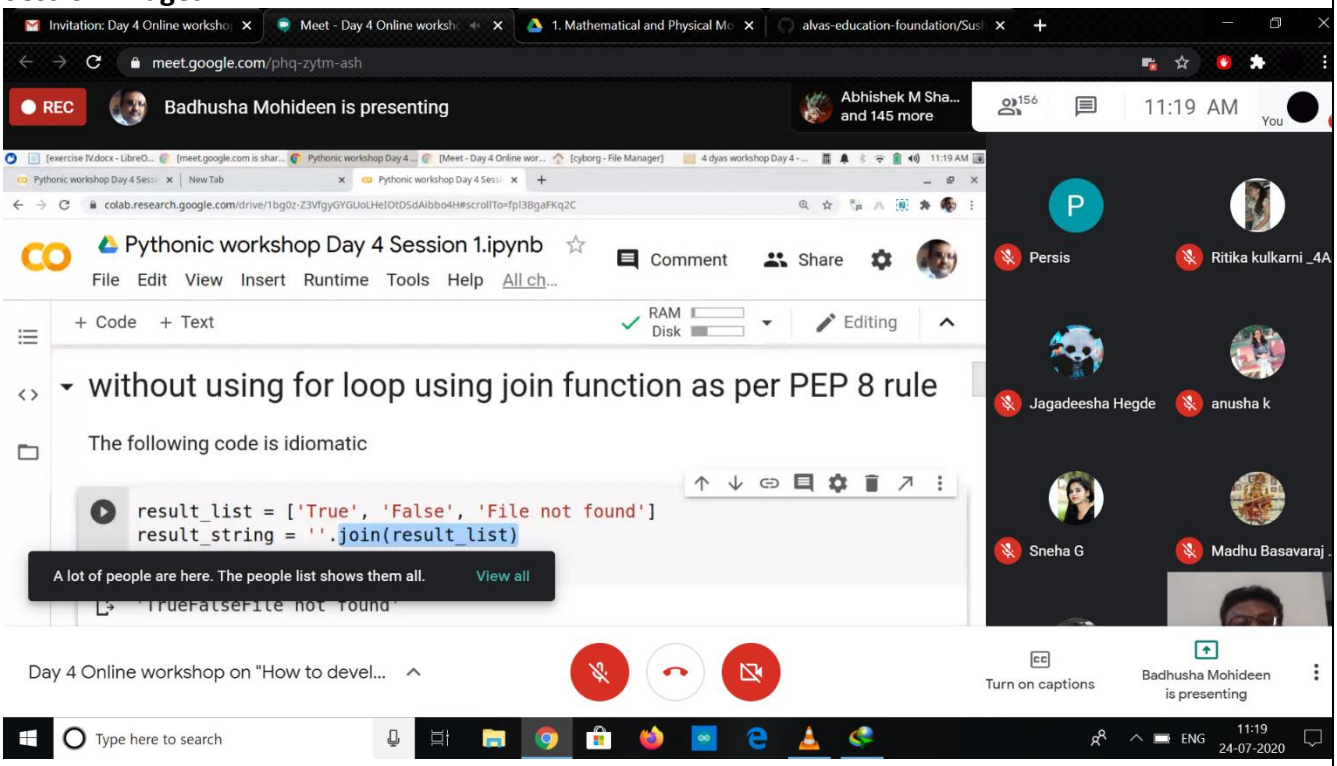
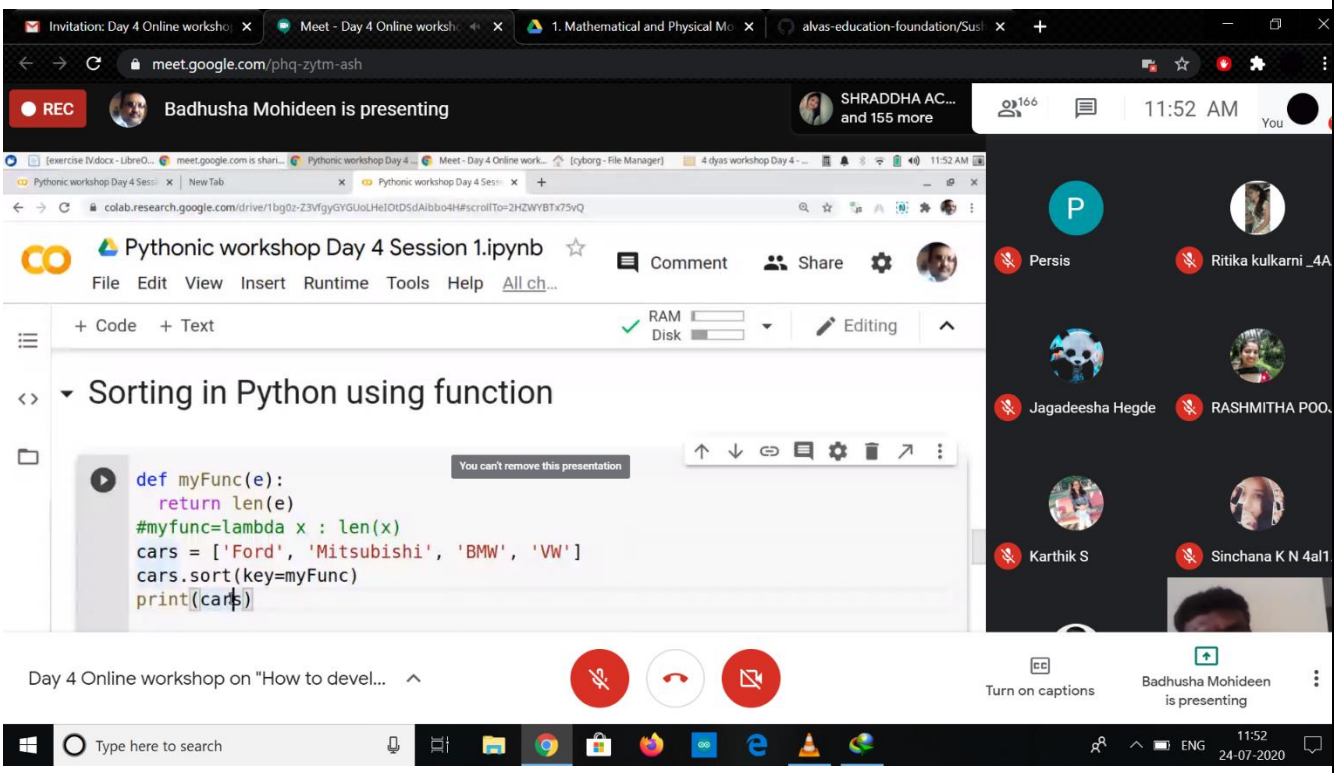
DAILY ASSESSMENT FORMAT

Date:	**24th July 2020**	**Name:**	**K B KUSHI**
Course:	**workshop**	**USN:**	**4AL17EC107**
Topic:	• How to develop Pythonic coding rather than Python coding • Certificate	**Semester & Section:**	**6 & B**
GitHub Repository:	**KUSHI-COURSES**		

SESSION DETAILS

Session images





Report:

A common neologism in the Python community is pythonic, which can have a wide range of meanings related to program style. To say that code is pythonic is to say that it uses Python idioms well, that it is natural or shows fluency in the language. Likewise, to say of an interface or language feature that it is pythonic is to say that it works well with Python idioms, that its use meshes well with the rest of the language.

Python scripts can put the system into different states, set configurations, and test all sorts of real-world use cases. Python can also be used to receive embedded system data that can be stored for analysis. Programmers can then use Python to develop parameters and other methods of analyzing that data.

There are certain things you can do with all sequence types. These operations include indexing, slicing, adding, multiplying, and checking for membership. In addition, Python has built-in functions for finding the length of a sequence and for finding its largest and smallest elements.

One of the special concepts in Python is the idea of writing idiomatic code that is most aligned with the language features and ideals. In Python, we call this idiomatic code Pythonic. While this idea is easy to understand, it turns out to be fairly hard to make concrete.

This course will take you on a tour of over 50 of the more popular and useful code examples demonstrating examples of Pythonic code. In the examples, you'll first see non-Pythonic code and then the more natural Pythonic version. Topics covered include the expansive use of dictionaries, hacking Python's memory usage via slots, using generators, comprehensions, and generator expressions, creating subsets of collections via slices (all the way to the database) and more.

Several of these are Python 3 features so you'll have even more reason to adopt Python 3 for your next project.