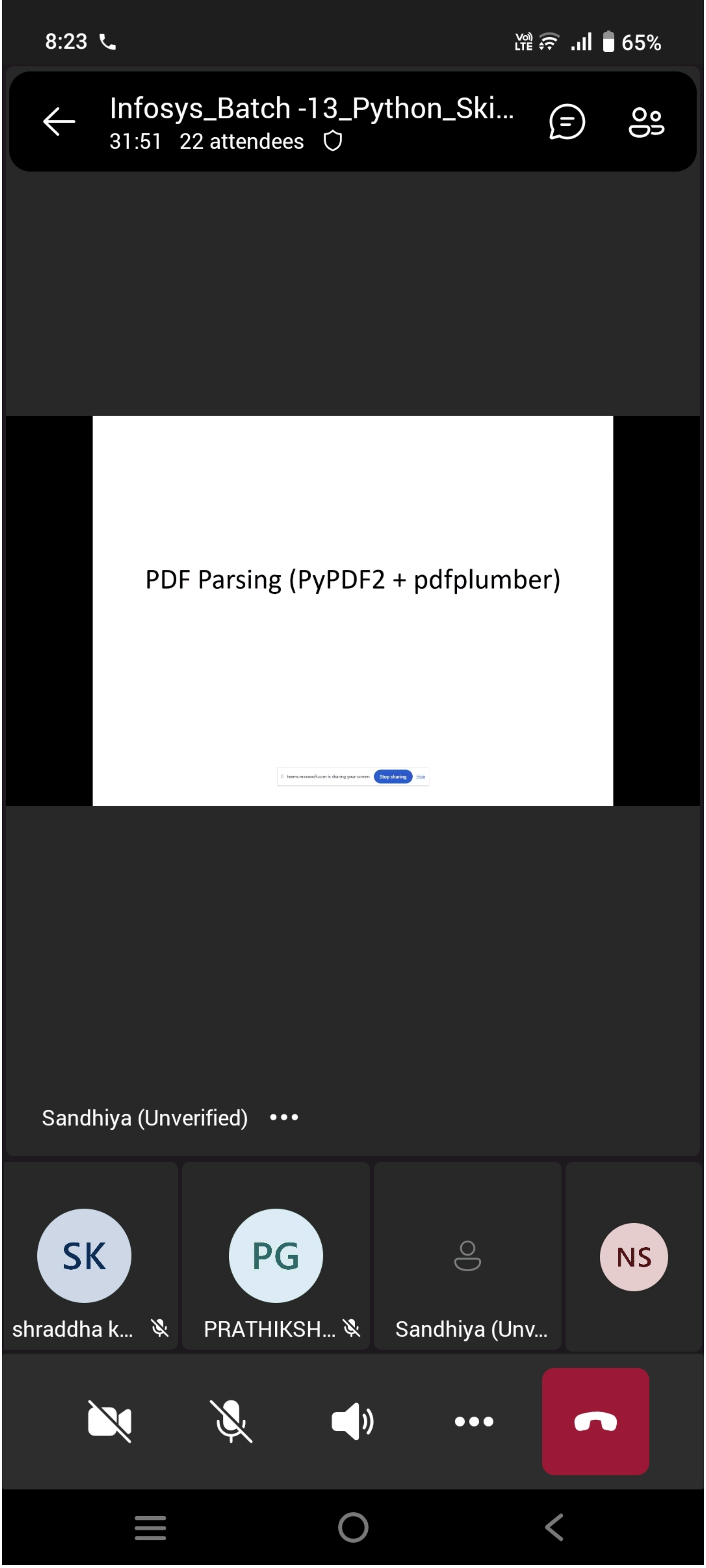
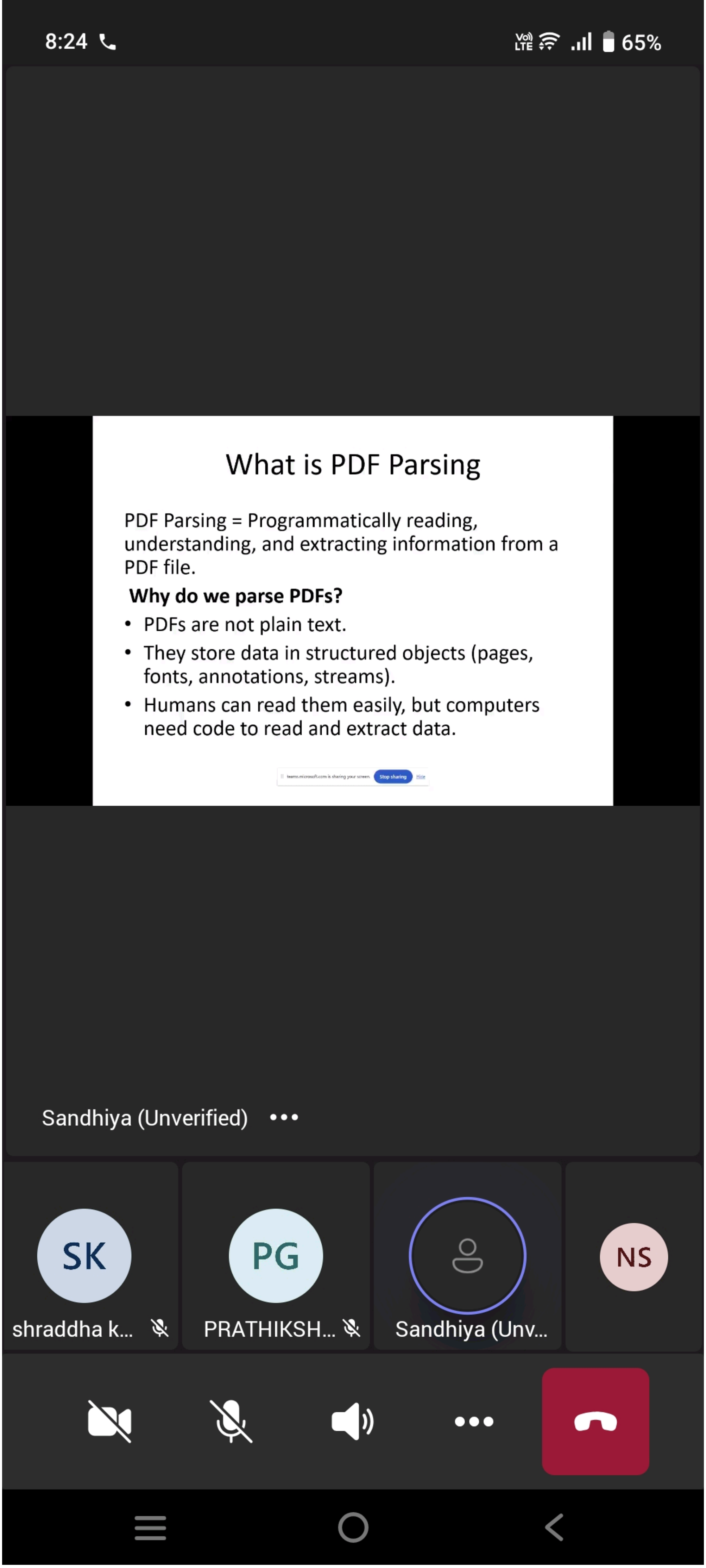


PDF Parsing (PyPDF2 + pdfplumber)

teams.microsoft.com is sharing your screen.

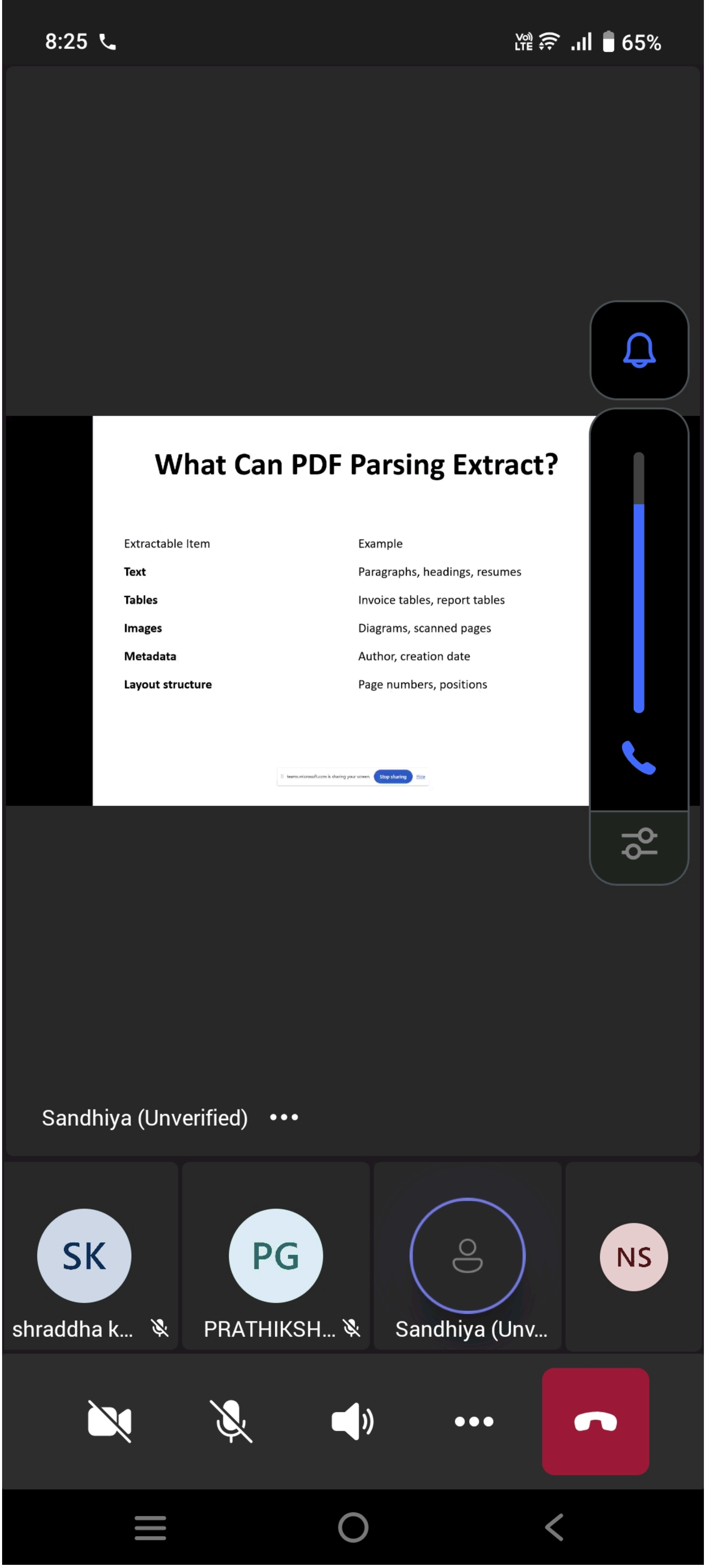


What is PDF Parsing

PDF Parsing = Programmatically reading, understanding, and extracting information from a PDF file.

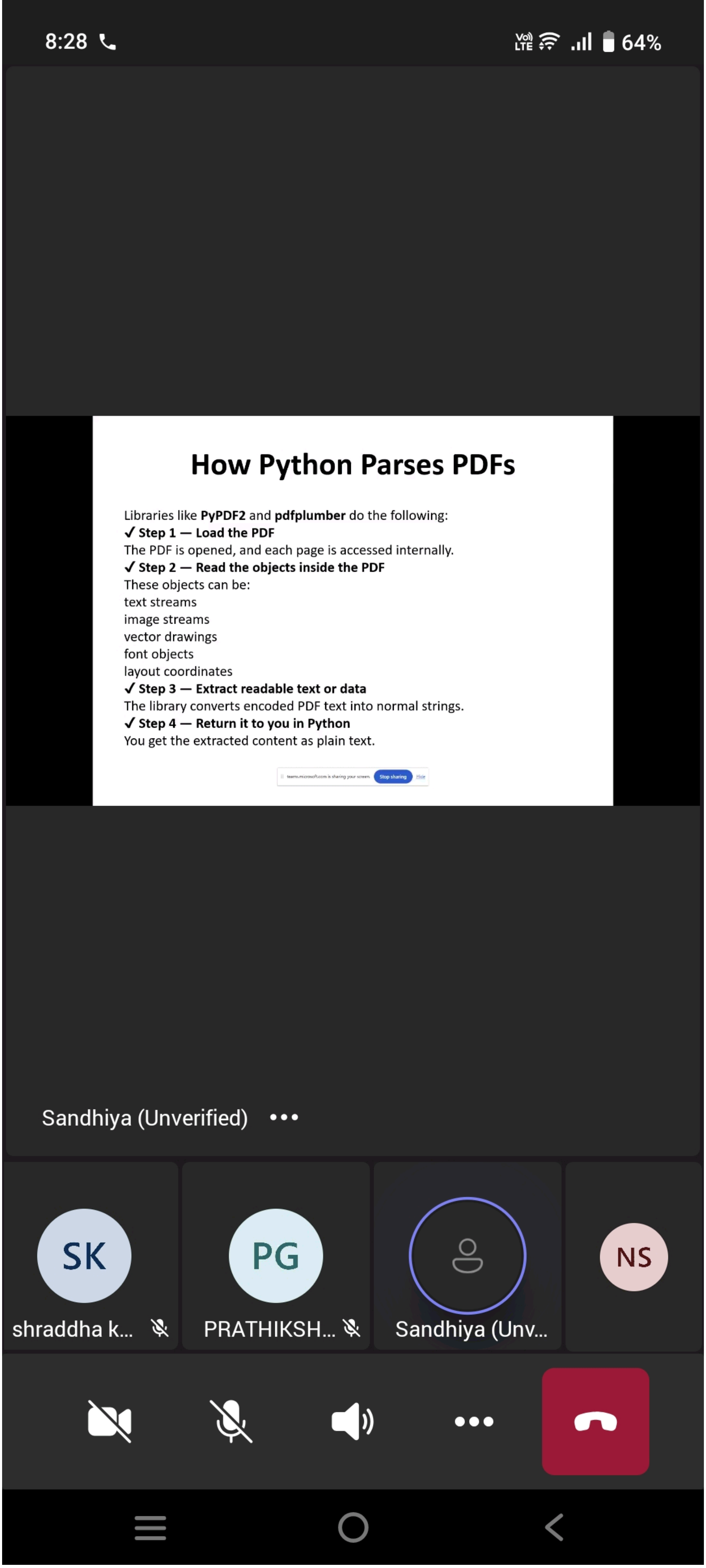
Why do we parse PDFs?

- PDFs are not plain text.
- They store data in structured objects (pages, fonts, annotations, streams).
- Humans can read them easily, but computers need code to read and extract data.



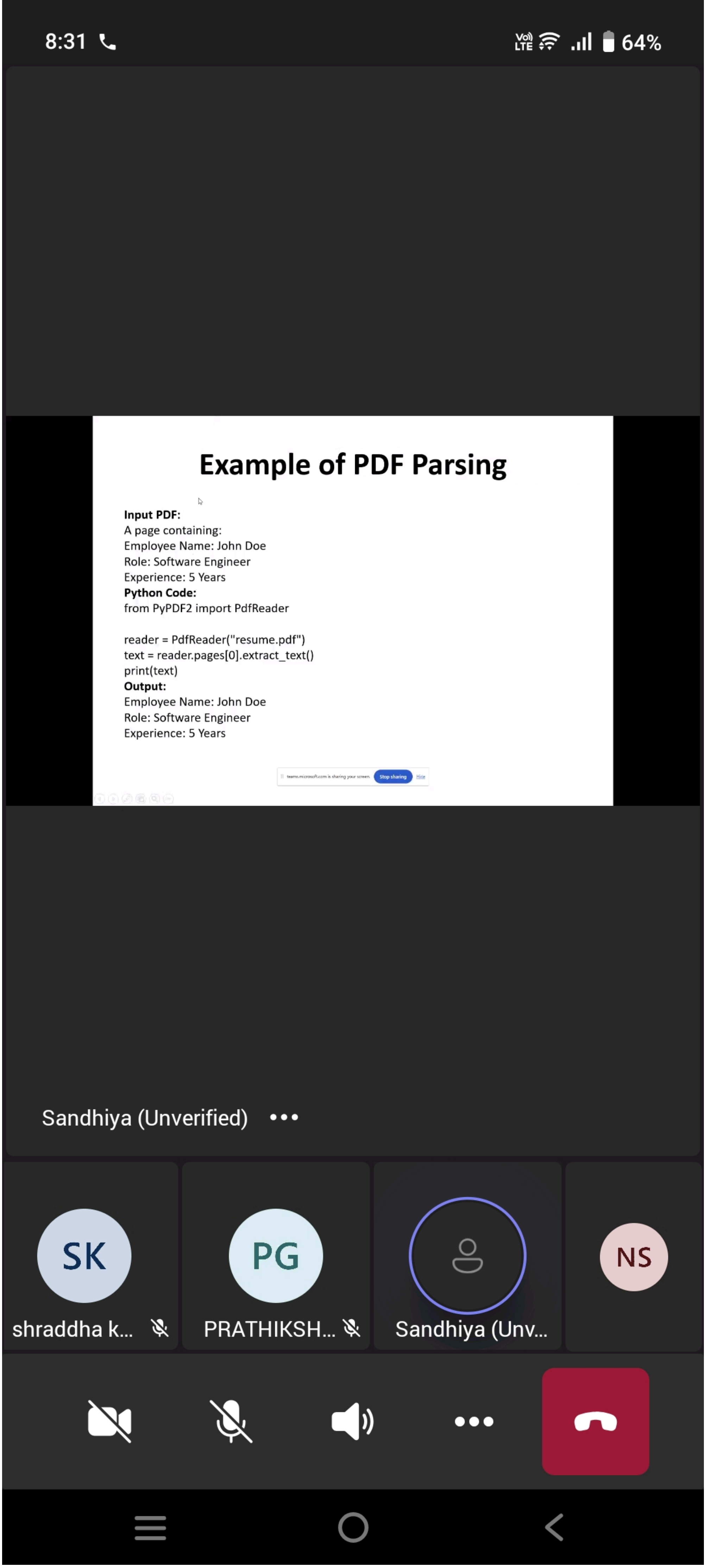
What Can PDF Parsing Extract?

Extractable Item	Example
Text	Paragraphs, headings, resumes
Tables	Invoice tables, report tables
Images	Diagrams, scanned pages
Metadata	Author, creation date
Layout structure	Page numbers, positions



How Python Parses PDFs

Libraries like **PyPDF2** and **pdfplumber** do the following:
✓ Step 1 — Load the PDF
The PDF is opened, and each page is accessed internally.
✓ Step 2 — Read the objects inside the PDF
These objects can be:
text streams
image streams
vector drawings
font objects
layout coordinates
✓ Step 3 — Extract readable text or data
The library converts encoded PDF text into normal strings.
✓ Step 4 — Return it to you in Python
You get the extracted content as plain text.



Example of PDF Parsing

Input PDF:

A page containing:
Employee Name: John Doe
Role: Software Engineer
Experience: 5 Years

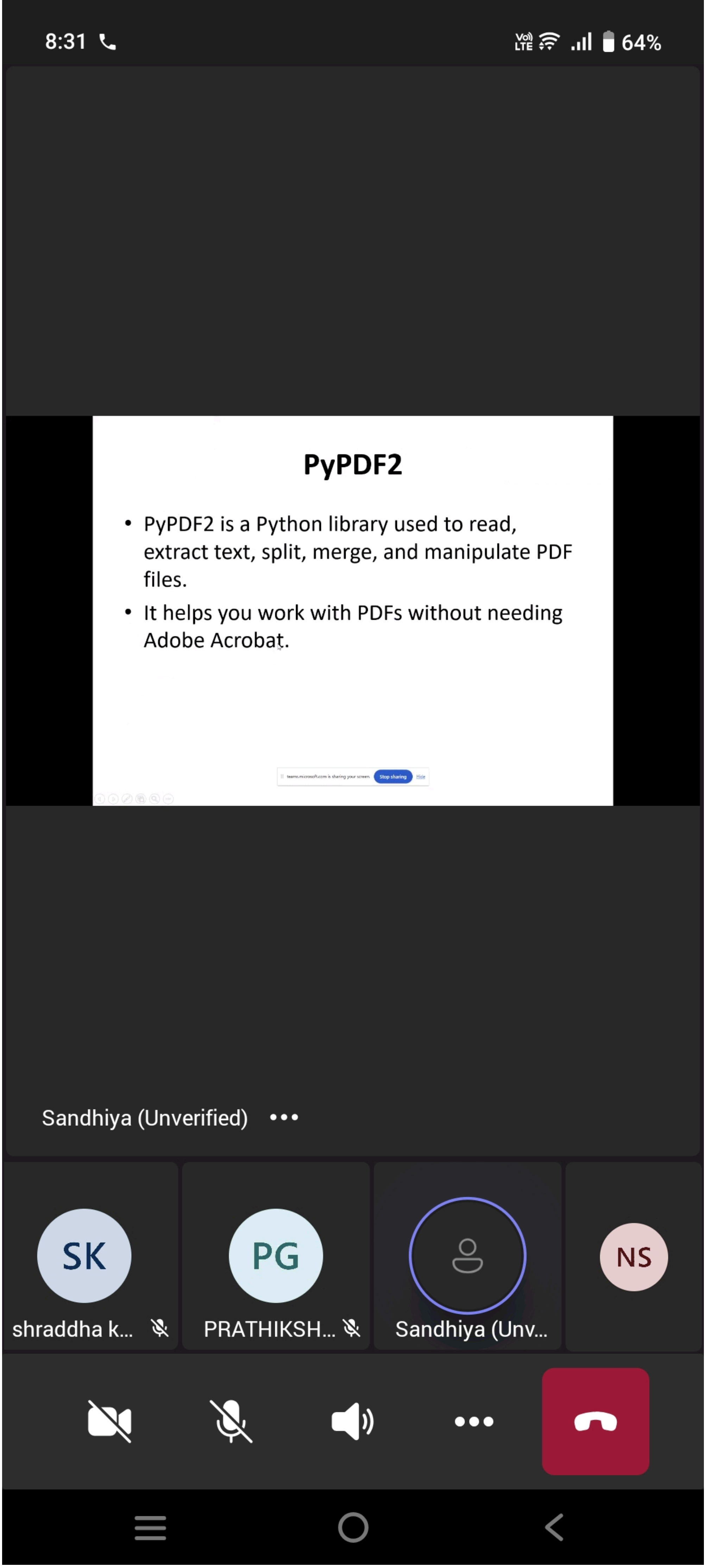
Python Code:

```
from PyPDF2 import PdfReader

reader = PdfReader("resume.pdf")
text = reader.pages[0].extract_text()
print(text)
```

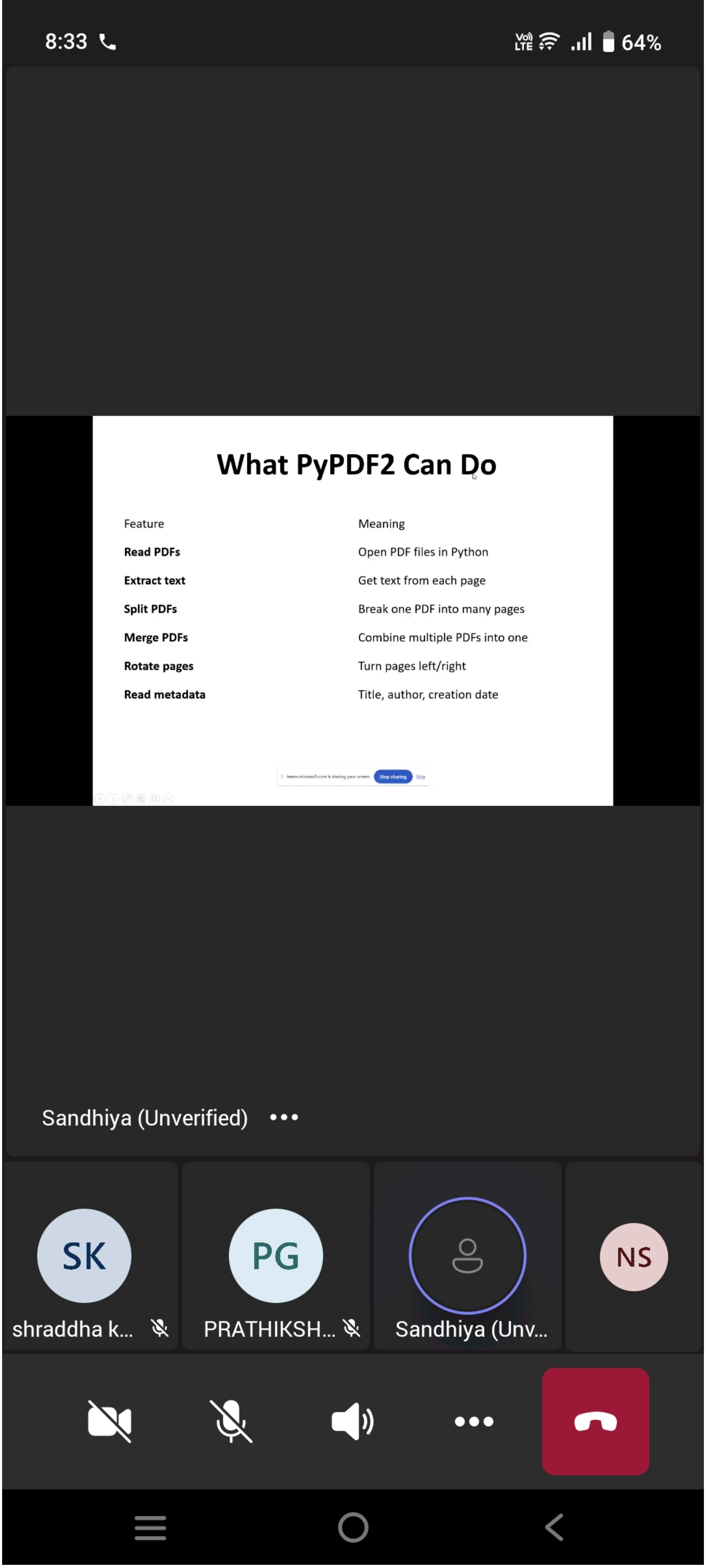
Output:

```
Employee Name: John Doe
Role: Software Engineer
Experience: 5 Years
```



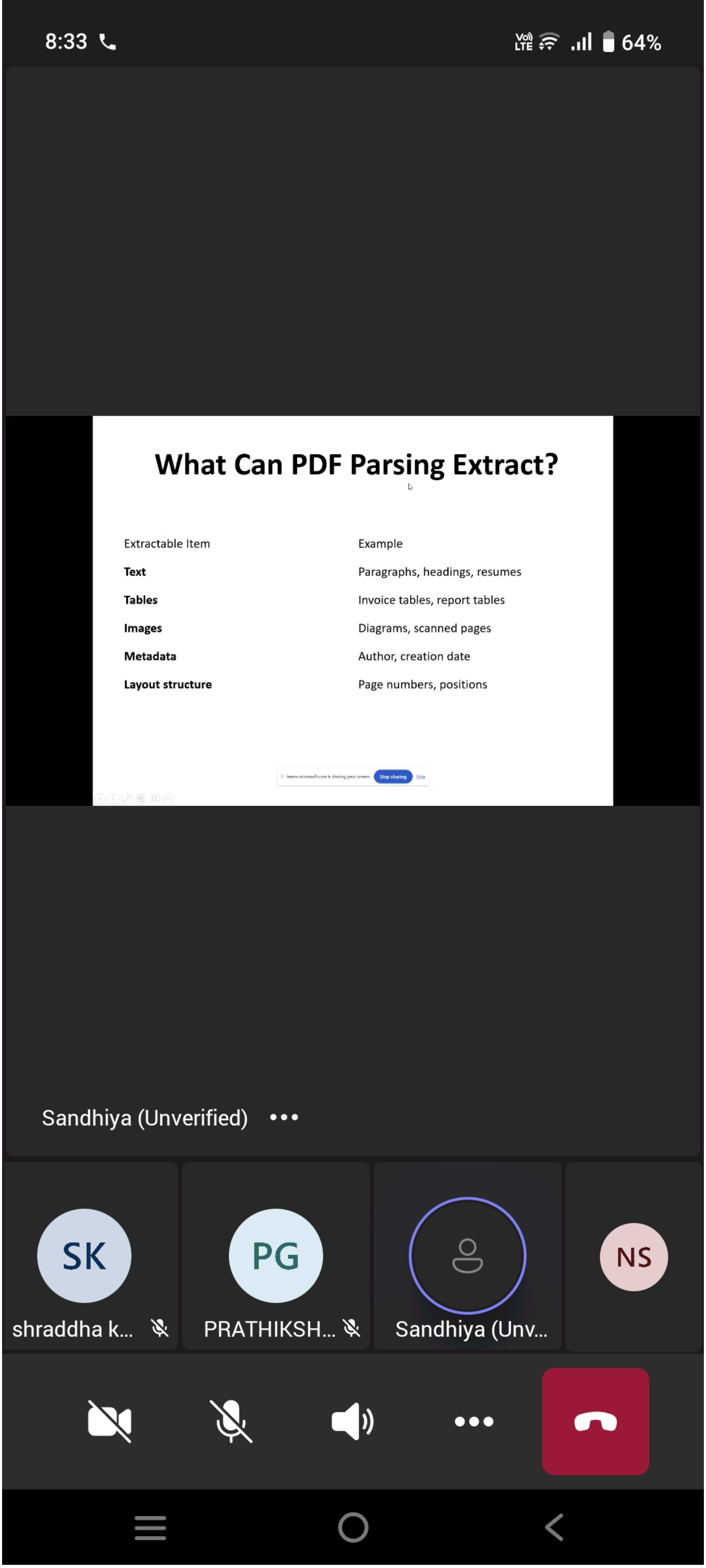
PyPDF2

- PyPDF2 is a Python library used to read, extract text, split, merge, and manipulate PDF files.
- It helps you work with PDFs without needing Adobe Acrobat.



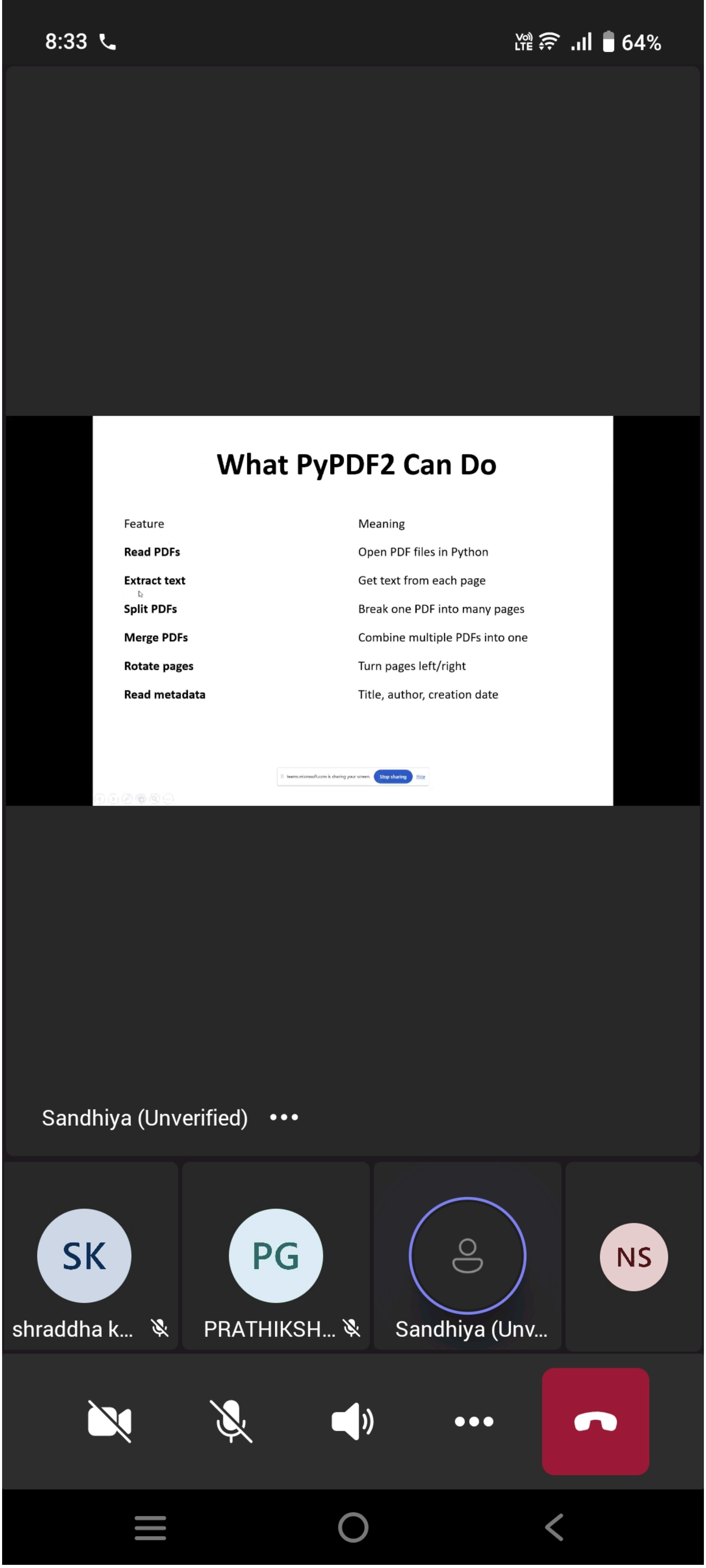
What PyPDF2 Can Do

Feature	Meaning
Read PDFs	Open PDF files in Python
Extract text	Get text from each page
Split PDFs	Break one PDF into many pages
Merge PDFs	Combine multiple PDFs into one
Rotate pages	Turn pages left/right
Read metadata	Title, author, creation date



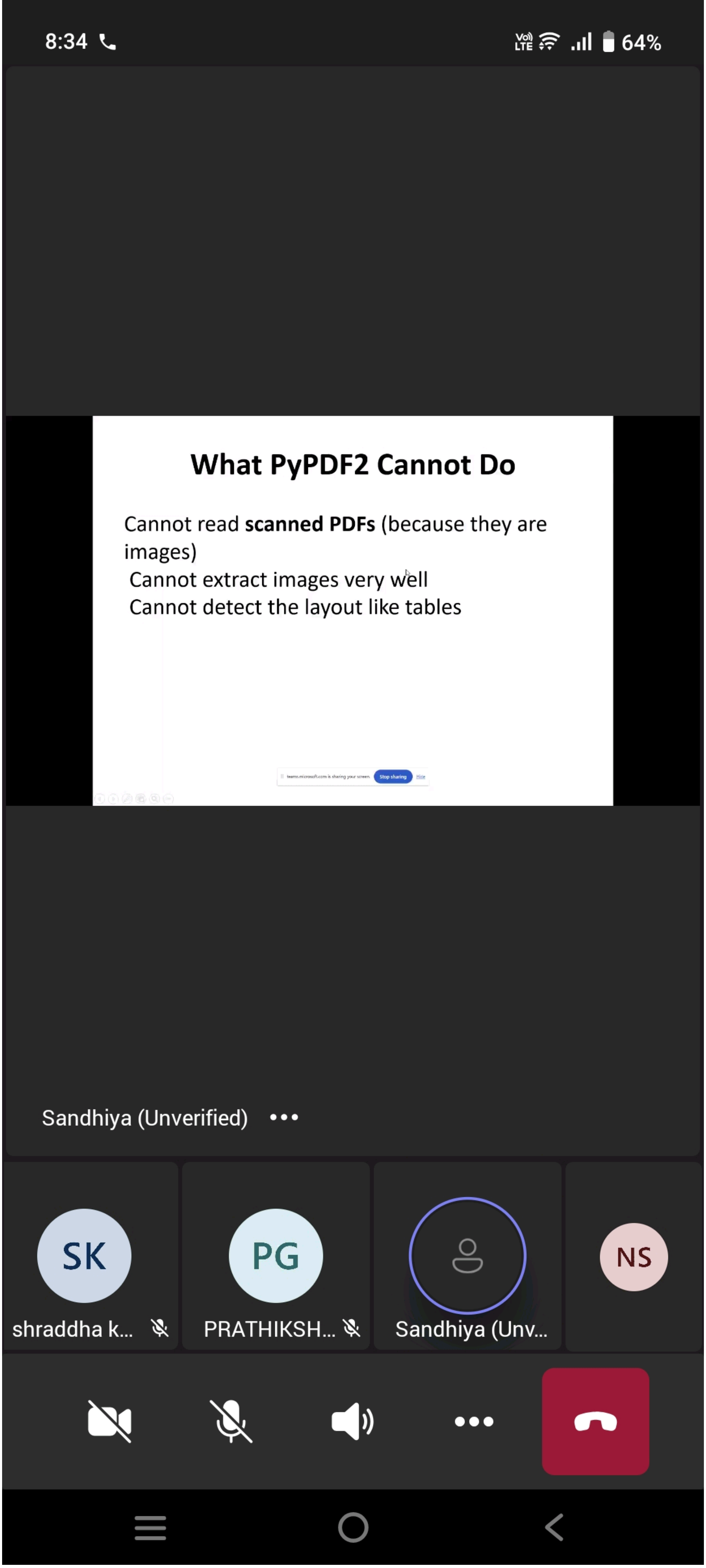
What Can PDF Parsing Extract?

Extractable Item	Example
Text	Paragraphs, headings, resumes
Tables	Invoice tables, report tables
Images	Diagrams, scanned pages
Metadata	Author, creation date
Layout structure	Page numbers, positions



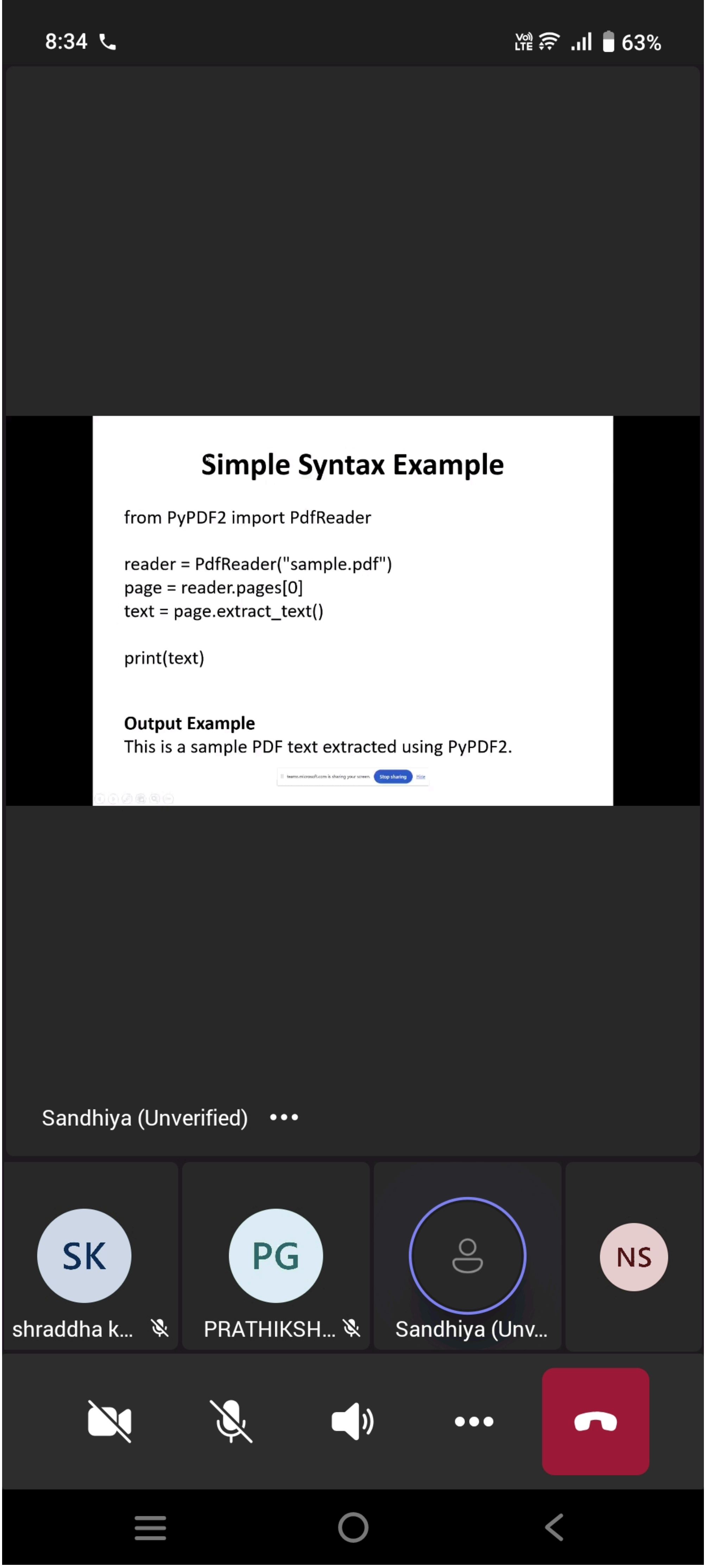
What PyPDF2 Can Do

Feature	Meaning
Read PDFs	Open PDF files in Python
Extract text	Get text from each page
Split PDFs	Break one PDF into many pages
Merge PDFs	Combine multiple PDFs into one
Rotate pages	Turn pages left/right
Read metadata	Title, author, creation date



What PyPDF2 Cannot Do

Cannot read **scanned PDFs** (because they are images)
Cannot extract images very well
Cannot detect the layout like tables



Simple Syntax Example

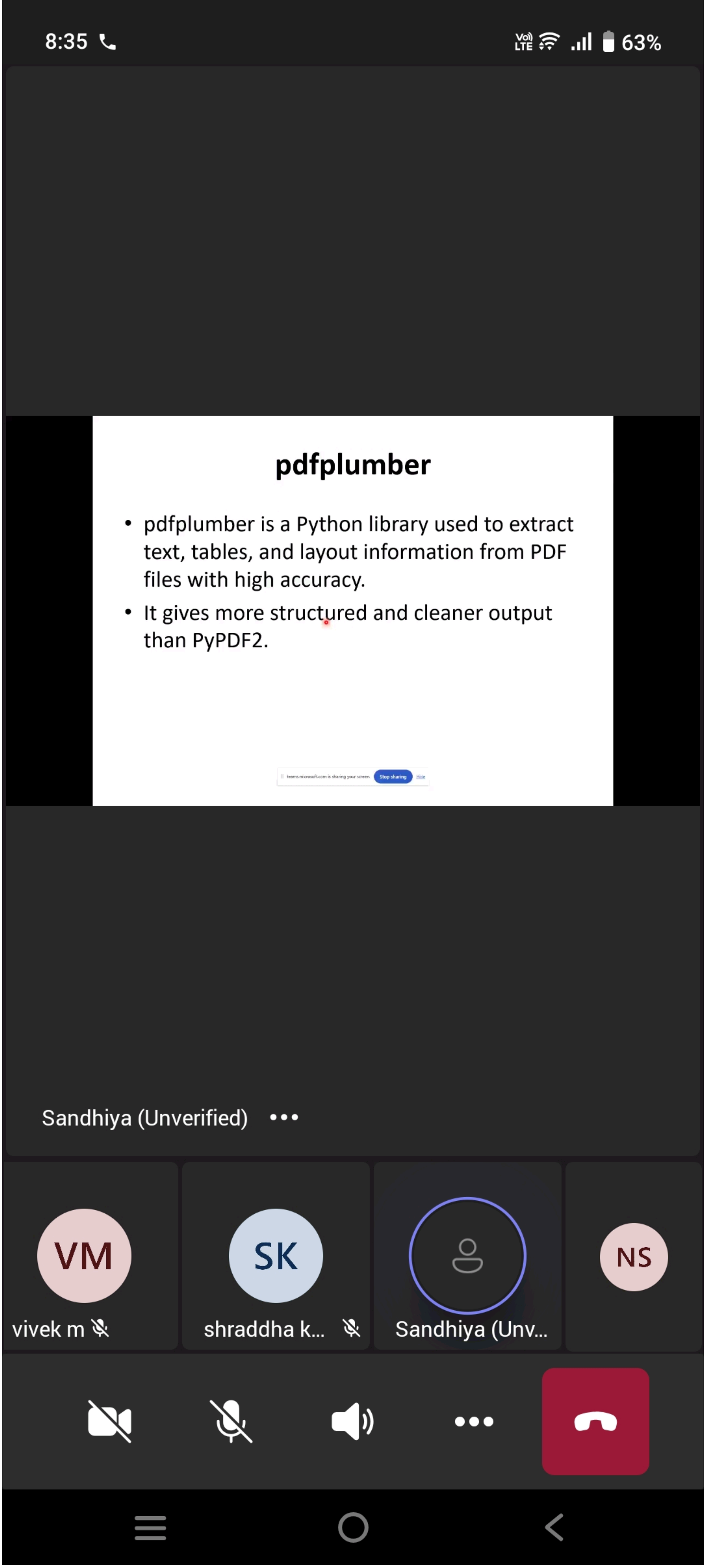
```
from PyPDF2 import PdfReader

reader = PdfReader("sample.pdf")
page = reader.pages[0]
text = page.extract_text()

print(text)
```

Output Example

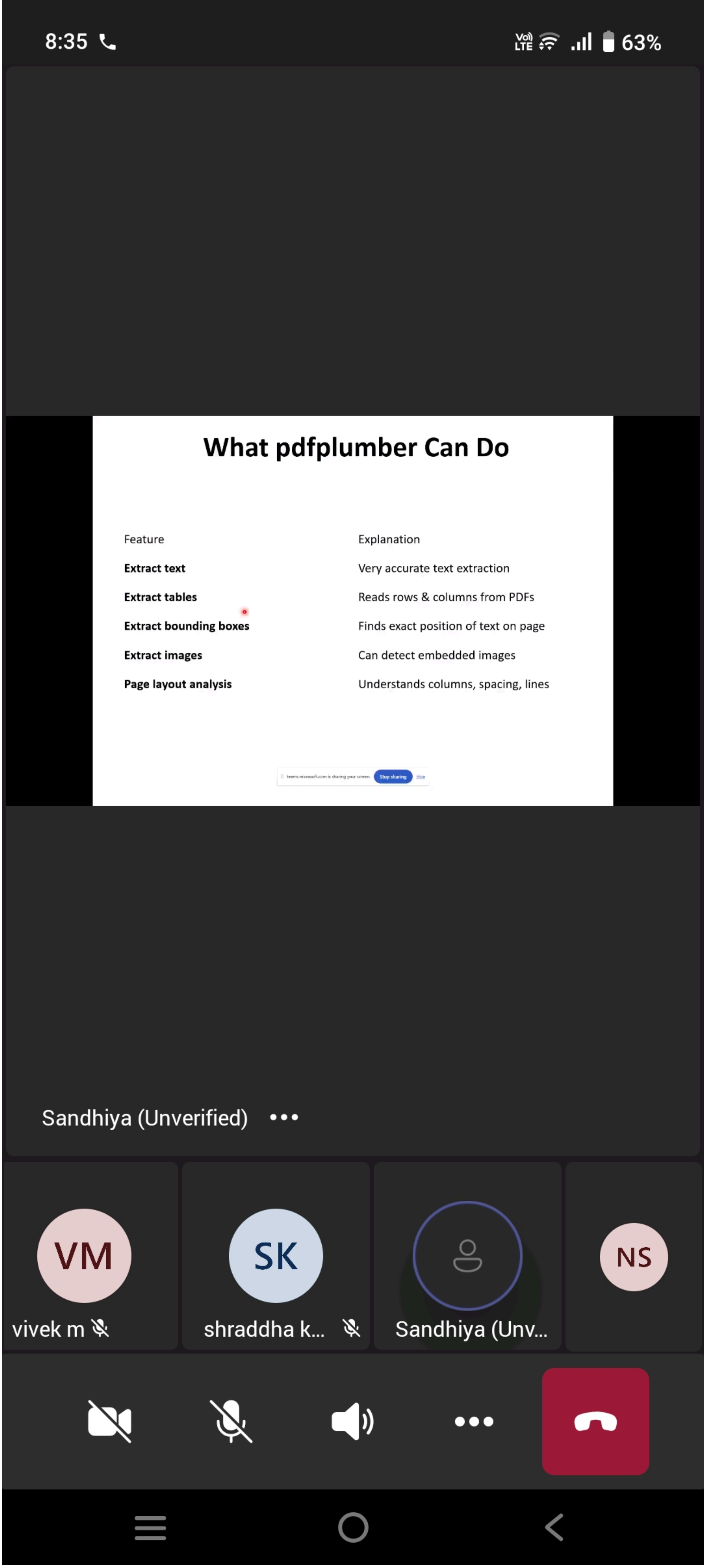
This is a sample PDF text extracted using PyPDF2.



pdfplumber

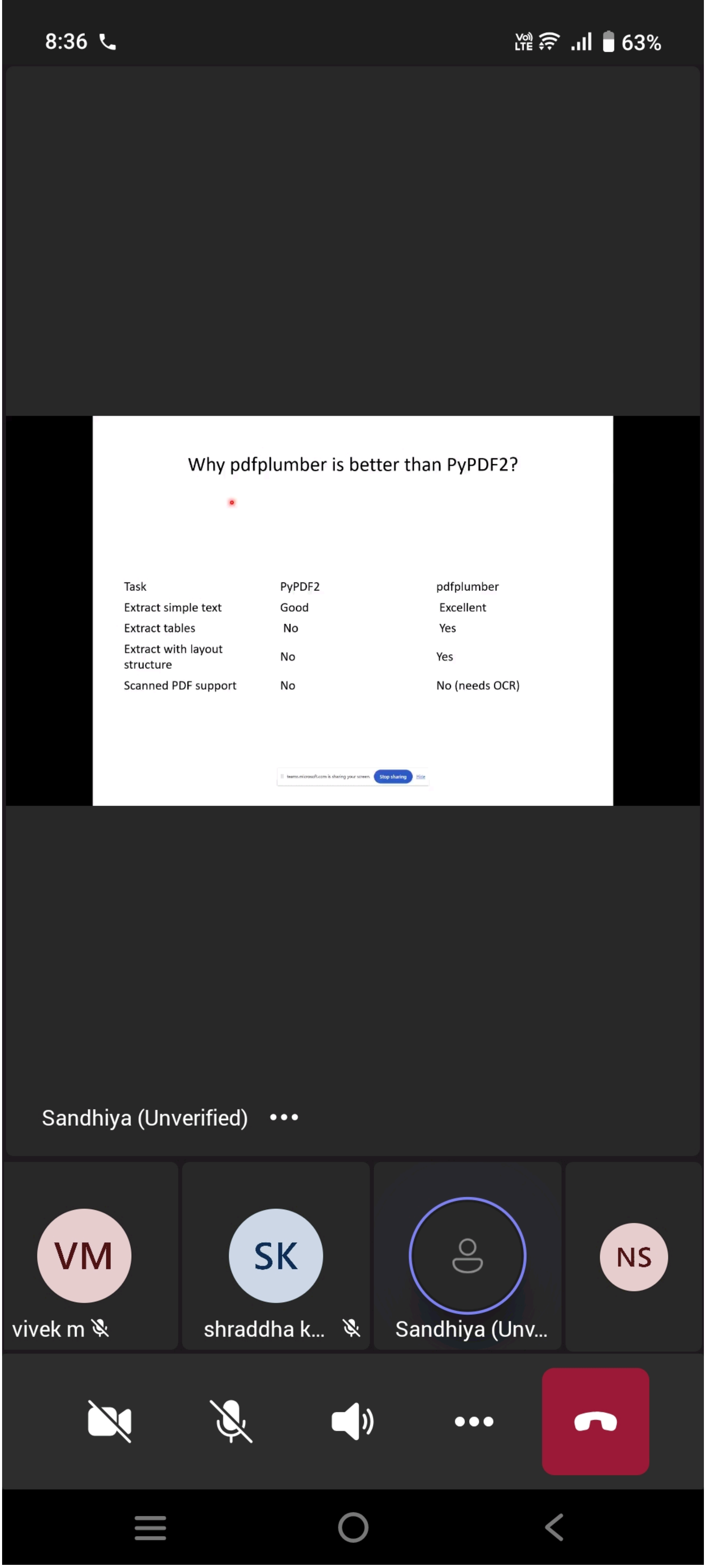
- pdfplumber is a Python library used to extract text, tables, and layout information from PDF files with high accuracy.
- It gives more structured and cleaner output than PyPDF2.

teams.microsoft.com is sharing your screen.



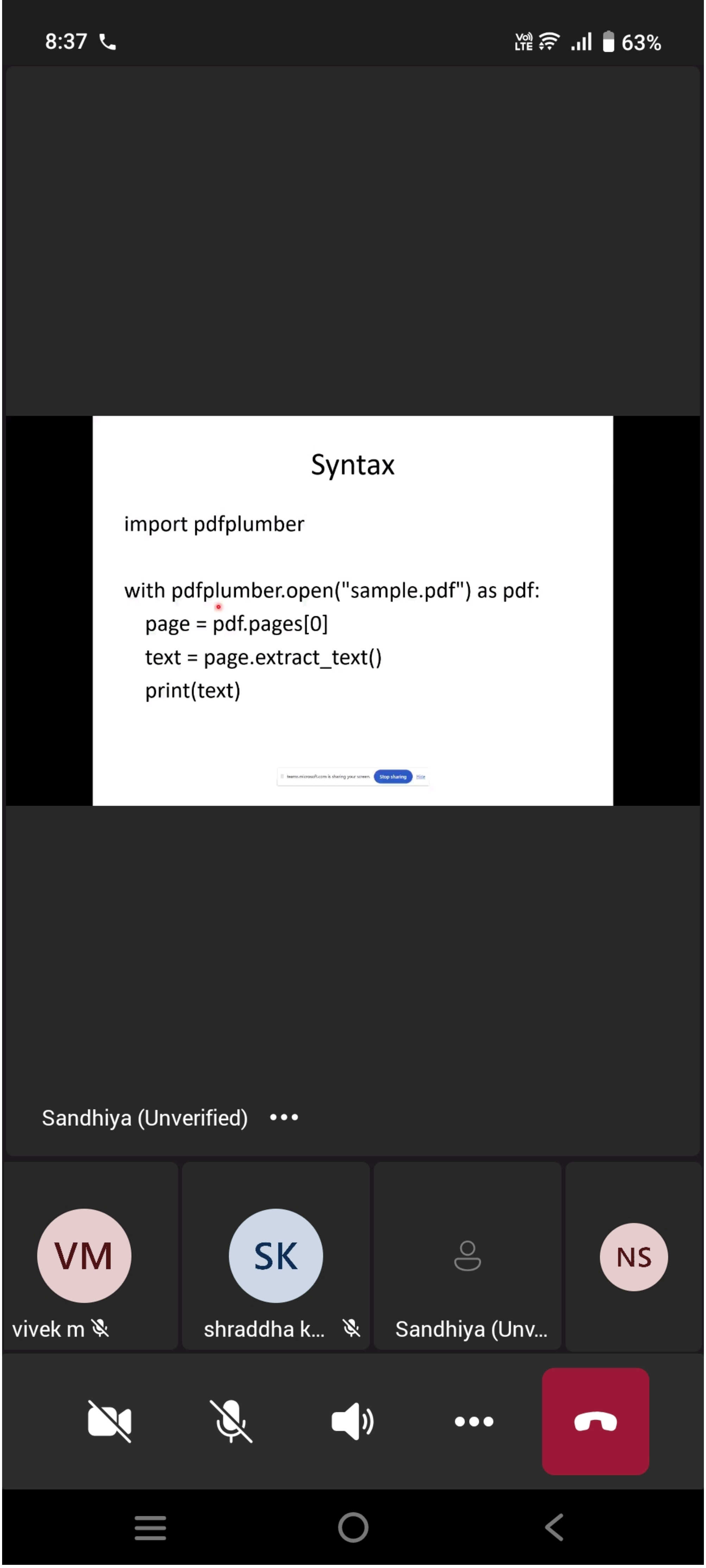
What pdfplumber Can Do

Feature	Explanation
Extract text	Very accurate text extraction
Extract tables	Reads rows & columns from PDFs
Extract bounding boxes	Finds exact position of text on page
Extract images	Can detect embedded images
Page layout analysis	Understands columns, spacing, lines



Why pdfplumber is better than PyPDF2?

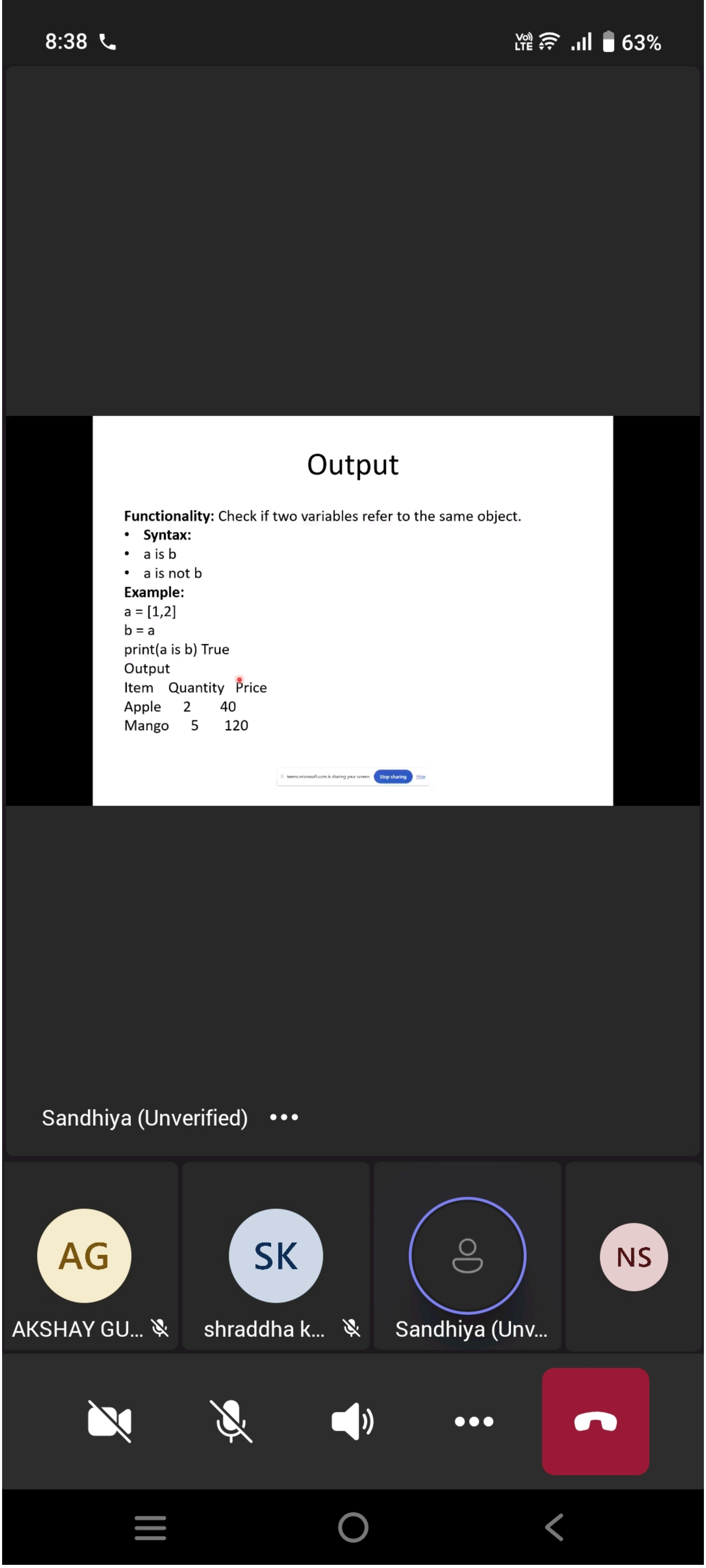
Task	PyPDF2	pdfplumber
Extract simple text	Good	Excellent
Extract tables	No	Yes
Extract with layout structure	No	Yes
Scanned PDF support	No	No (needs OCR)



Syntax

```
import pdfplumber

with pdfplumber.open("sample.pdf") as pdf:
    page = pdf.pages[0]
    text = page.extract_text()
    print(text)
```



Output

Functionality: Check if two variables refer to the same object.

- **Syntax:**
- a is b
- a is not b

Example:

```
a = [1,2]
b = a
print(a is b) True
Output
Item    Quantity   Price
Apple      2       40
Mango      5       120
```