# #SOURCE CODE

import requests

PUBLIC\_KEY = 'YOUR\_PUBLIC\_KEY'

SECRET\_KEY = 'YOUR\_SECRET\_KEY'

API\_BASE\_URL = 'https://api.ilovepdf.com/v1'

def perform\_pdf\_task(task, input\_files, output\_file, options={}):

headers = {

'Authorization': f'Token {PUBLIC\_KEY}:{SECRET\_KEY}'

}

if task == 1:

# Combine two or more PDF files into one

url = f'{API\_BASE\_URL}/tasks/pdfmerge'

data = {

'files': input\_files,

}

elif task == 2:

# Separate PDF pages or extract all pages into a PDF

url = f'{API\_BASE\_URL}/tasks/pdfsplit'

data = {

'range': 'all',

'files': input\_files,

}

elif task == 3:

# Remove PDF password security for reading and editing

url = f'{API\_BASE\_URL}/tasks/pdfunlock'

data = {

'files': input\_files,

}

elif task == 4:

# Extract all text from a PDF file to a TXT file

url = f'{API\_BASE\_URL}/tasks/text'

data = {

'files': input\_files,

}

elif task == 5:

# Convert JPG, TIFF, and PNG images to PDF

url = f'{API\_BASE\_URL}/tasks/imagepdf'

data = {

'files': input\_files,

}

response = requests.post(url, headers=headers, data=data, files=input\_files)

if response.status\_code == 200:

task\_id = response.json().get('task')

download\_url = f'{API\_BASE\_URL}/tasks/{task\_id}/download'

response = requests.get(download\_url, stream=True)

if response.status\_code == 200:

with open(output\_file, 'wb') as f:

for chunk in response.iter\_content(1024):

f.write(chunk)

print(f'Task completed. Output saved to {output\_file}')

else:

print(f'Failed to download the output file. Error: {response.text}')

else:

print(f'Task failed. Error: {response.text}')

if \_name\_ == '\_main\_':

task = int(input('Select a task (1-5): '))

input\_files = {'file': open('input.pdf', 'rb')} # Provide the input files

output\_file = 'output.pdf' # Define the output file path

perform\_pdf\_task(task, input\_files, output\_file)