from docx import Document #used to create the word documents and manipulate the word documents

from docx.enum.text import WD\_PARAGRAPH\_ALIGNMENT #setting the paragraph alignment

from docx.shared import Pt, Inches #size of the font size of paragraph in the word document

from datetime import datetime #date of the exam

import random #for selecting the random questions

def load\_questions(input\_file):

# Load questions from the input .docx file

doc = Document(input\_file) #Opens the input Word document

questions = [para.text.strip() for para in doc.paragraphs if para.text.strip() != ""] #Iterates over each paragraph in the document, strips any leading or trailing whitespace,

if paragraph is not empty: #and adds it to the list of question

return questions

def generate\_question\_paper(questions, num\_questions):

# Generate a question paper based on constraints

question\_paper = [] #Initializes an empty list to store selected questions

for i in range(0, len(questions), 5):

block = questions[i:i+5] #Creates a block of up to 5 questions

if block: #if the block is not empty

question\_paper.append(random.choice(block))#select the random question

return question\_paper[:num\_questions] #Returns the first num\_questions questions from the generated list

def create\_header(doc, marks,branch, time\_duration, exam\_date, sub,):

# Create header for question paper

section=doc.sections[0] #Gets the first section of the document

header=section.header #Accesses the header of the section.

header\_paragraph=header.paragraphs[0]#Gets the first paragraph in the header

# Adds text to the header

run = header\_paragraph.add\_run(

f"SRI VENKATESWARA COLLEGE OF ENGINEERING, TIRUPATI\n"

f"Name of the department: {branch}\n"

f"Subject: {sub}\n"

f"Marks:{marks}\n"

f"Date:{exam\_date}\t\tTime:{time\_duration} mins\n\n"# Ensure a double newline before QUESTION PAPER

f"Answer the following questions"

)

header\_paragraph.alignment=WD\_PARAGRAPH\_ALIGNMENT.CENTER

# Centers the header text.

run.font.size = Pt(12) #Sets the font size to 12 points.

def save\_question\_paper(question\_paper, output\_file, marks, branch,time\_duration, exam\_date, sub):

# Save the generated question paper to a .docx file

doc = Document()

# Add header

create\_header(doc, marks, branch,time\_duration, exam\_date, sub)

for i, question in enumerate(question\_paper, 1):

paragraph = doc.add\_paragraph(f"{i}.{question}"+" " +"[10 Marks]")

# Add marks at the end of each question

# Add 'or' in the middle of two consecutive questions

if i % 2 == 1 and i != len(question\_paper):

or\_paragraph = doc.add\_paragraph()

or\_run = or\_paragraph.add\_run('or')

or\_paragraph.alignment = WD\_PARAGRAPH\_ALIGNMENT.CENTER

# Add footer information

footer = doc.sections[0].footer

footer\_paragraph = footer.paragraphs[0]

footer\_paragraph.text = "End of Question Paper"

footer\_paragraph.alignment = WD\_PARAGRAPH\_ALIGNMENT.CENTER

doc.save(output\_file)

def main():

input\_file = input("Enter the path to the input .docx file: ")

sub = input("Enter the exam subject: ")

branch=input("enter the name of the department:")

num\_sets = int(input("Enter the number of question paper sets to generate: "))

num\_questions = int(input("Enter the number of questions per question paper: "))

marks = input("Enter the total marks: ")

time\_duration = input("Enter the time duration: ")

exam\_date = input("Enter the date of the exam (dd-mm-yyyy): ")

# Validate the date

try:

datetime.strptime(exam\_date, "%d-%m-%Y")

except ValueError:

print("Invalid date format. Please enter the date in dd-mm-yyyy format.")

return

questions = load\_questions(input\_file)

for i in range(num\_sets):

output\_file = input(f"Enter the path to save Question Paper Set {i+1}: ")

question\_paper = generate\_question\_paper(questions, num\_questions)

save\_question\_paper(question\_paper, output\_file, marks, branch,time\_duration, exam\_date, sub)

print("Question papers generated successfully.")

if \_\_name\_\_ == "\_\_main\_\_":

main()