PYTHON PPROJECT SEM II

By Group 10

Topic:

Video Clip to Showcase Importance of Green IT

Name	Roll No	Topic
Kshiti Vartak	4267	Regulatory Compliance
Darpan	4226	Resource Conservation
Aastha Meher	4231	Environmental Sustainability
Viraj Kamble	4220	Innovation
Anuja Kharat	4225	Cost Savings
Adishree Khamgaonkar	4224	Health and Well Being

Drive Link:
https://drive.google.com/drive/folders/160Pa_8du4JKCxCNvmmHnTqDKAJRr2Ykc?usp=drive_link

Regulatory Compliance Video: By Kshiti

Code:

Main Code:

```
from moviepy.editor import *

clips = []
clip1 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0001.jpg').set_duration(1)
clip2 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0002.jpg').set_duration(0.5)
clip3 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0003.jpg').set_duration(0.5)
clip4 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0004.jpg').set_duration(1)
clip5 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0005.jpg').set_duration(1)
clip6 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0006.jpg').set_duration(1)
clip7 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0007.jpg').set_duration(1)
clip8 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0008.jpg').set_duration(1)
clip9 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0009.jpg').set_duration(2)
clip10 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0010.jpg').set_duration(1.5)
clip11 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0011.jpg').set_duration(1.5)
clip12 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0012.jpg').set_duration(1.5)
clip13 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0013.jpg').set_duration(1.5)
clip14 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0014.jpg').set_duration(1.5)
clip15 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0015.jpg').set_duration(1)
clip16 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0016.jpg').set_duration(2)
clip17 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0017.jpg').set_duration(1.5)
clip18 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0018.jpg').set_duration(1.5)
clip19 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0019.jpg').set_duration(1.5)
clip20 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0020.jpg').set_duration(1.5)
clip21 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0021.jpg').set_duration(1.5)
clip22 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0022.jpg').set_duration(2)
```

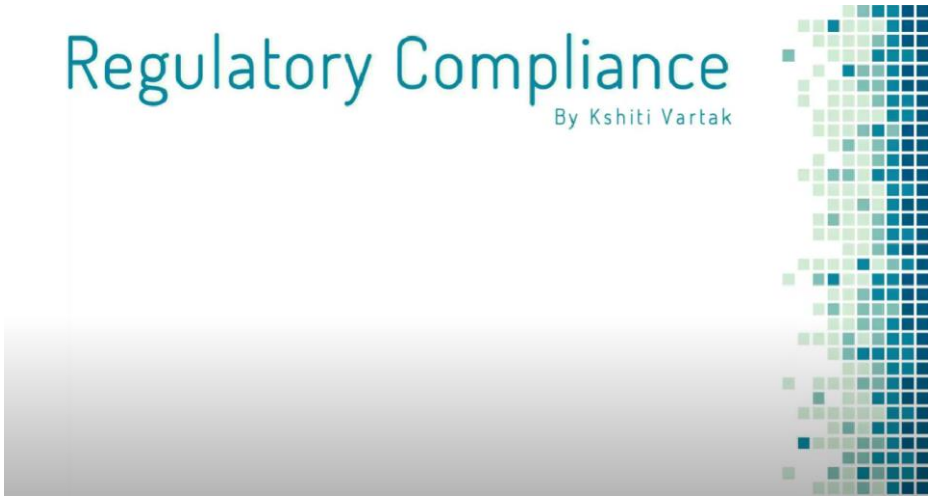
```
clip23 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0023.jpg').set_duration(1.5)
clip24 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0024.jpg').set_duration(1.5)
clip25 =  ImageClip(r'C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\0025.jpg').set_duration(2)


clips.append(clip1)
clips.append(clip2)
clips.append(clip3)
clips.append(clip4)
clips.append(clip5)
clips.append(clip6)
clips.append(clip7)
clips.append(clip8)
clips.append(clip9)
clips.append(clip10)
clips.append(clip11)
clips.append(clip12)
clips.append(clip13)
clips.append(clip14)
clips.append(clip15)
clips.append(clip16)
clips.append(clip17)
clips.append(clip18)
clips.append(clip19)
clips.append(clip20)
clips.append(clip21)
clips.append(clip22)
clips.append(clip23)
clips.append(clip24)
clips.append(clip25)


video_clip = concatenate_videoclips(clips, method='compose')
video_clip.write_videofile("FinalKshiti.mp4", fps=24, remove_temp=True,
codec="libx264", audio_codec="aac")
```

Text on Image

```
from PIL import Image
from PIL import ImageDraw
from PIL import ImageFont

# Open an Image
lastframe = Image.open(r"C:\Users\Kshiti\OneDrive\Desktop\Kshiti Video Python
Project\lastframe.jpg")

# Call draw Method to add 2D graphics in an image
textonimg = ImageDraw.Draw(lastframe)

# Custom font style and font size
font_path = "C:/Users/Kshiti/OneDrive/Desktop/Kshiti Video Python
Project/TitilliumWeb-Regular.ttf"
myFont = ImageFont.truetype(font_path, 80)
```

```
# Add Text to an image
textonimg.text((200, 536), "These are just a few examples of the
",font=myFont, fill=(0, 89, 127))
textonimg.text((200, 636), "regulatory compliance measures ",font=myFont,
fill=(0, 89, 127))
textonimg.text((200, 736), "currently in place related to Green
IT.  ",font=myFont, fill=(0, 89, 127))
textonimg.text((200, 936), "Organizations operating in these regions
",font=myFont, fill=(0, 89, 127))
textonimg.text((200, 1036), "or markets need to be aware of these
",font=myFont, fill=(0, 89, 127))
textonimg.text((200, 1136), "regulations and ensure compliance to avoid
",font=myFont, fill=(0, 89, 127))
textonimg.text((200, 1236), "fines and penalties and demonstrate their
",font=myFont, fill=(0, 89, 127))

# Display edited image
lastframe.show()

# Save the edited image
lastframe.save("LastFrame1.jpeg")
```

Video in Drive Link:



Resource Conversation Video By Darpan

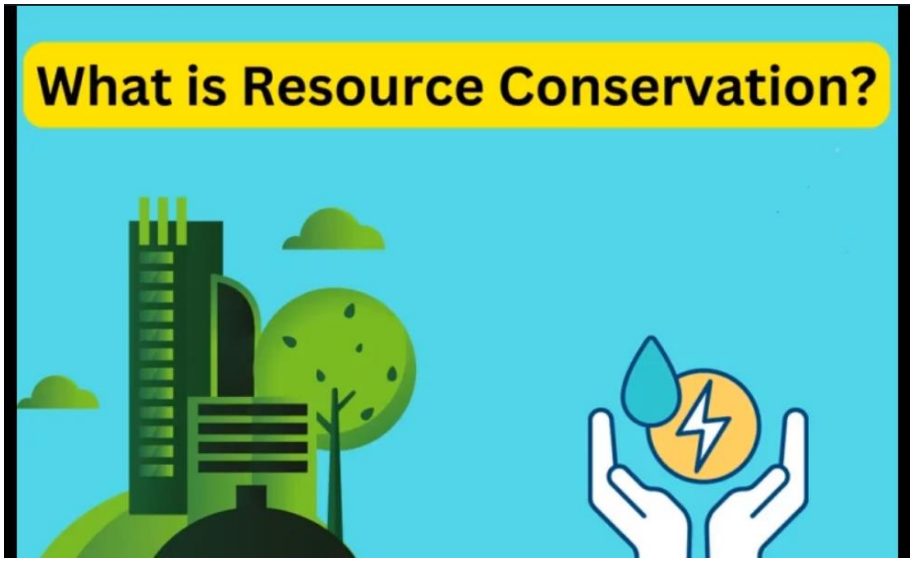
Code:

```
from moviepy.editor import ImageClip, concatenate_videoclips, TextClip

import textwrap



# List of image paths and durations

image_data = [
```

```
    {'path': 'C:/Users/Rsc/Documents/py/resource.png', 'duration': 3, 'key_aspect':
'RESOURCE CONSERVATION \n Efficient utilization and management of resources to reduce
waste and promote sustainability. Helps in preserving natural resources, minimizing
environmental impact, and achieving long-term sustainability goals.'},

    {'path': 'C:/Users/Rsc/Documents/py/data.jpg', 'duration': 3, 'key_aspect': 'DATA
MANAGEMENT \n The process of organizing, storing, and efficiently retrieving data to
ensure its accuracy, availability, and security. Facilitates better decision-making, enhances
productivity, and ensures compliance with data protection regulations.'},

    {'path': 'C:/Users/Rsc/Documents/py/cooling.jpg', 'duration': 3, 'key_aspect': 'EFFICIENT
COOLING \n Implementing technologies and strategies to optimize cooling systems in data
centers and electronic devices. Reduces energy consumption, extends equipment lifespan,
and maintains optimal operating conditions.'},

    {'path': 'C:/Users/Rsc/Documents/py/green_computing.jpg', 'duration': 3, 'key_aspect':
'GREEN COMPUTING \n Designing, using, and disposing of computing devices and systems in
an environmentally friendly manner. Reduces environmental impact, lowers energy
consumption, and promotes sustainability in the IT industry.'},

    {'path': 'C:/Users/Rsc/Documents/py/sustainable.jpg', 'duration': 3, 'key_aspect':
'SUSTAINABILITY \n Meeting the needs of the present without compromising the ability of
future generations to meet their own needs. Addresses environmental, social, and
economic concerns to create a balanced and enduring system.'},

    {'path': 'C:/Users/Rsc/Documents/py/ewaste.jpg', 'duration': 3, 'key_aspect': 'E-WASTE
MANAGEMENT \n The proper disposal and recycling of electronic waste to prevent
environmental pollution and health hazards. Reduces environmental impact, recovers
valuable materials, and mitigates the negative effects of electronic waste.'},

    {'path': 'C:/Users/Rsc/Documents/py/remote.webp', 'duration': 3, 'key_aspect': 'REMOTE
WORK \n Technologies and practices that enable individuals to work from locations outside
the traditional office setting. Enhances flexibility, work-life balance, and business continuity,
especially during unforeseen events.'},

]



# Fixed size for all clips (width, height)
fixed_size = (640, 480)
# Function to create a TextClip with black background and white text
def create_key_aspect_clip(key_aspect, duration):
    wrapped_text = textwrap.fill(key_aspect, width=30)
    text_clip = TextClip(wrapped_text, fontsize=30, color='black', bg_color='white',
size=fixed_size)

    text_clip = text_clip.set_position(('left', 'center'))
    return text_clip.set_duration(duration)
```

```
# Load and resize images using ImageClip
video_clips = []
for data in image_data:
    image_clip = ImageClip(data['path'], duration=data['duration']).resize(width=fixed_size[0],
height=fixed_size[1])
    key_aspect_clip = create_key_aspect_clip(data['key_aspect'], data['duration'])
    video_clips.extend([image_clip, key_aspect_clip])

# Concatenate video clips
video_clip = concatenate_videoclips(video_clips, method="compose")

# Output the final video
output_path = 'output_video_with_key_aspects.mp4'
video_clip.write_videofile(output_path, codec='libx264', fps=24)
```

Video in Drive Link:



Environmental Sustainability Video By Aastha

Code:

```
from moviepy.editor import ImageSequenceClip, concatenate_videoclips, VideoFileClip
from PIL import Image
import numpy as np
import os

# Image file paths (excluding the intro image)
image_files = [
    r'C:\Users\HP\OneDrive\Pictures\img1.jpg',
    r'C:\Users\HP\OneDrive\Pictures\img2.jpg',
    r'C:\Users\HP\OneDrive\Pictures\img3.jpg',
    r'C:\Users\HP\OneDrive\Pictures\img4.jpg',
    r'C:\Users\HP\OneDrive\Pictures\img5.jpg',
```

```
    r'C:\Users\HP\OneDrive\Pictures\img6.jpg',
    r'C:\Users\HP\OneDrive\Pictures\img7.jpg',
    r'C:\Users\HP\OneDrive\Pictures\img8.jpg',
    r'C:\Users\HP\OneDrive\Pictures\img9.jpg',
]

# Image and video durations
duration = 3

# Fixed size for all clips (width, height)
fixed_size = (640, 480)

# Create video clips for each image with fixed size
video_clips = []

for image_path in image_files:
    # Convert the Image object to a NumPy array
    image_pil = Image.open(image_path).resize(fixed_size)
    image_array = np.array(image_pil)

    # Create the ImageSequenceClip with the NumPy array
    video_clip = ImageSequenceClip([image_array], fps=1 / duration)

    video_clips.append(video_clip)

# Concatenate all video clips
final_clip = concatenate_videoclips(video_clips, method="compose")

# Export the final video with fixed size
output_path = 'output_video_only_images.mp4'
final_clip.write_videofile(output_path, codec='libx264', fps=24)
```

Video in Drive Link:

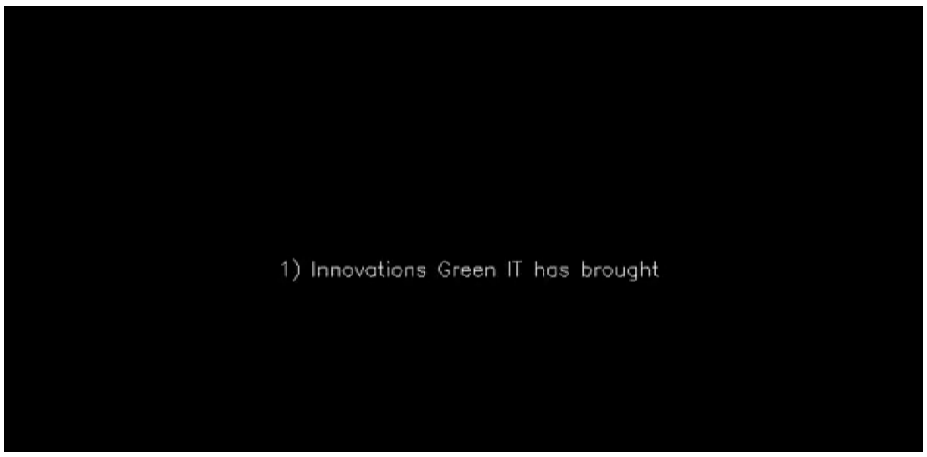
Innovation By Viraj

Code:

```
import cv2
import numpy as np  # Import NumPy module for array operations

# Output video file name
output_video = 'fourthlast.mp4'

# Define video properties
frame_width = 640
frame_height = 480
fps = 24
duration_sec = 10

# Initialize video writer object
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
out = cv2.VideoWriter(output_video, fourcc, fps, (frame_width, frame_height))

# Define text messages
texts = [
   "1) Innovations Green IT has brought",
   "2) Energy-Efficient Hardware: \nManufacturers have been developing energy-efficient \ncomponents such as processors, servers, and storage devices.",
   "3) Virtualization: Virtualization \ntechnology allows multiple virtual machines to run \non a single physical server, thereby optimizing resource \nutilization and reducing the number of physical servers required.",
   "4) Cloud Computing: Cloud computing \nenables organizations to access computing resources, \nsuch as servers, storage, and applications, over the internet on \na pay-per-use basis."
]

# Define image paths
image_paths = [
   "C:/Users/admin/Desktop/pythonprograms/EEH.png",
    "C:/Users/admin/Desktop/pythonprograms/image2.jpg",
   "C:/Users/admin/Desktop/pythonprograms/image3.jpg",
    "C:/Users/admin/Desktop/pythonprograms/image4.jpg"
]

# Duration for each text and image combination (in seconds)
duration_sec_per_item = 5

# Write text and image combinations to video
for i in range(len(texts)):
   # Split text into lines
```

```
    lines = texts[i].split('\n')

    # Write each line of text to video
    font_scale = 0.6
    thickness = 1
    y_offset = (frame_height - (len(lines) * 30)) // 2  # Center vertically
    for line in lines:
        text_size = cv2.getTextSize(line, cv2.FONT_HERSHEY_SIMPLEX, font_scale, thickness)[0]
        text_x = (frame_width - text_size[0]) // 2
        text_y = y_offset + text_size[1] + 10  # Add 10 pixels of vertical spacing between lines
        if text_y + text_size[1] < frame_height:  # Ensure text is within frame boundaries
            frame = np.zeros((frame_height, frame_width, 3), dtype=np.uint8)
            frame = cv2.putText(
                frame,
                line,
                (text_x, text_y),
                cv2.FONT_HERSHEY_SIMPLEX,
                font_scale,
                (255, 255, 255),
                thickness
            )
            # Write duplicate frames to slow down the text animation
            for _ in range(int(duration_sec_per_item * fps / len(texts))):
                out.write(frame)
        y_offset += 30  # Increment y-offset for next line

    # Write image to video
    for _ in range(int(duration_sec_per_item * fps)):
        frame = cv2.imread(image_paths[i % len(image_paths)])  # Ensure cyclic access to
images
        frame = cv2.resize(frame, (frame_width, frame_height))
        out.write(frame)

# Release video writer object
out.release()

print("Video created successfully!")
```

Video in Drive Link:



Cost Savings by Anuja

Code:

```
from moviepy.editor import ImageSequenceClip, concatenate_videoclips, VideoFileClip
from PIL import Image
import numpy as np
import os

# Image file paths (excluding the intro image)
image_files = [
    r'C:\Users\HP\OneDrive\Pictures\imgg1.jpg',
    r'C:\Users\HP\OneDrive\Pictures\imgg2.jpg',
    r'C:\Users\HP\OneDrive\Pictures\imgg3.jpg',
    r'C:\Users\HP\OneDrive\Pictures\imgg4.jpg',
    r'C:\Users\HP\OneDrive\Pictures\imgg5.jpg',
    r'C:\Users\HP\OneDrive\Pictures\imgg6.jpg',
    r'C:\Users\HP\OneDrive\Pictures\imgg7.jpg',
]

# Image and video durations
duration = 3

# Fixed size for all clips (width, height)
fixed_size = (640, 480)

# Create video clips for each image with fixed size
video_clips = []

for image_path in image_files:
    # Convert the Image object to a NumPy array
    image_pil = Image.open(image_path).resize(fixed_size)
    image_array = np.array(image_pil)

    # Create the ImageSequenceClip with the NumPy array
```

```
    video_clip = ImageSequenceClip([image_array], fps=1 / duration)

    video_clips.append(video_clip)

# Concatenate all video clips
final_clip = concatenate_videoclips(video_clips, method="compose")

# Export the final video with fixed size
output_path = 'output_video_only_images.mp4'
final_clip.write_videofile(output_path, codec='libx264', fps=24)
```

Video in Drive Link:

Health and Well Being by Adishree

Code:

```
from moviepy.editor import ImageSequenceClip, concatenate_videoclips, VideoFileClip
from PIL import Image
import numpy as np
import os

# Image file paths (excluding the intro image)
image_files = [
    r'C:\Users\HP\OneDrive\Pictures\imggg3.jpg',
    r'C:\Users\HP\OneDrive\Pictures\imggg6.jpg',
    r'C:\Users\HP\OneDrive\Pictures\imggg1.jpg',
    r'C:\Users\HP\OneDrive\Pictures\imggg4.jpg',
    r'C:\Users\HP\OneDrive\Pictures\imggg2.jpg',
    r'C:\Users\HP\OneDrive\Pictures\imggg5.jpg',
]
```

```
# Image and video durations
duration = 3

# Fixed size for all clips (width, height)
fixed_size = (640, 480)

# Create video clips for each image with fixed size
video_clips = []

for image_path in image_files:
    # Convert the Image object to a NumPy array
    image_pil = Image.open(image_path).resize(fixed_size)
    image_array = np.array(image_pil)

    # Create the ImageSequenceClip with the NumPy array
    video_clip = ImageSequenceClip([image_array], fps=1 / duration)

    video_clips.append(video_clip)

# Concatenate all video clips
final_clip = concatenate_videoclips(video_clips, method="compose")

# Export the final video with fixed size
output_path = 'output_video_only_images.mp4'
final_clip.write_videofile(output_path, codec='libx264', fps=24)
```

Video in Drive Link: