

# ZIP to PDF Converter

## Introduction

- Introduce the ZIP to PDF conversion process and why it is useful for users needing efficient document handling.
- Mention the technology stack (Streamlit, Python libraries).
- Discuss potential applications for users who need to convert multiple images from ZIP files into a unified PDF format.

Share ☆ ✎ 🔍 ⋮

## ZIP to PDF Converter

Upload 1 to 15 ZIP files containing images. Rearrange the ZIP file order, and we'll create a PDF based on your selection. The first ZIP file's name will be used for the PDF.

Upload ZIP files

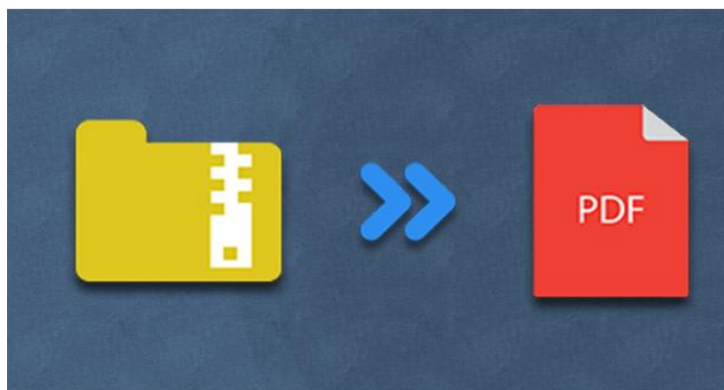


Drag and drop files here  
Limit 1GB per file • ZIP

Browse files

## Objective

- The goal is to provide a simple and efficient solution for users who want to convert images from ZIP files into a PDF document.
- Focus on making it user-friendly, allowing up to 15 ZIP files, each containing images that need to be processed and merged into a final PDF.
- Provide reorder functionality to customize the sequence of images.



# Methods

- The method includes uploading multiple ZIP files, extracting images, processing each image to fit A4 size, and converting them into a PDF.
- A custom sorting mechanism ensures images are ordered based on numeric filenames.
- PDF files are merged, and the final result is generated for download.
- Code  
“

```
if uploaded_files:
```

```
    if len(uploaded_files) < 1 or len(uploaded_files) > 15:
```

```
        st.error("Please upload between 1 and 15 ZIP files.")
```

```
        return
```

```
# Display ZIP files for reordering
```

```
zip_names = [uploaded_file.name for uploaded_file in uploaded_files]
```

```
ordered_zip_names = st.multiselect("Reorder ZIP files:", zip_names, default=zip_names)
```

```
if len(ordered_zip_names) != len(uploaded_files):
```

```
    st.error("Please select all uploaded ZIP files in your desired order.")
```

```
    return
```

```
temp_dir = "temp_images"
```

```
os.makedirs(temp_dir, exist_ok=True)
```

```
pdf_buffers = []
```

```
try:
```

```
    # Iterate through ordered ZIP files
```

```
    for zip_name in ordered_zip_names:
```

```
        uploaded_file = next(file for file in uploaded_files if file.name == zip_name)
```

```
        all_image_files = []
```

```
        # Extract images from ZIP file
```

```
        with zipfile.ZipFile(uploaded_file, 'r') as zip_ref:
```

```
            zip_ref.extractall(temp_dir)
```

```
        # Collect image files
```

```
        for f in os.listdir(temp_dir):
```

```
            file_path = os.path.join(temp_dir, f)
```

```
            if f.lower().endswith(('.png', '.jpg', '.jpeg', '.bmp', '.tiff', '.webp')):
```

```

        all_image_files.append(file_path)

# Sort and convert images to PDF
all_image_files = sorted(all_image_files, key=lambda x: extract_number(os.path.basename(x)))
pdf_buffer = convert_images_to_pdf(all_image_files)
pdf_buffers.append(pdf_buffer)

# Cleanup temp directory
for f in os.listdir(temp_dir):
    os.remove(os.path.join(temp_dir, f))

# Merge PDFs
merged_pdf_buffer = merge_pdfs(pdf_buffers)

# Provide download
final_pdf_filename = f"{ordered_zip_names[0].rsplit('.', 1)[0]}.pdf"
st.success(f"PDF successfully created: {final_pdf_filename}")
provide_pdf_download(merged_pdf_buffer, final_pdf_filename)

finally:
    if os.path.exists(temp_dir):
        shutil.rmtree(temp_dir)

```

## Testcases

- The app was tested with various cases:
  1. A single ZIP file with mixed image formats.
  2. Multiple ZIP files, with images having inconsistent names.
  3. ZIP files with unsupported formats (like non-image files).
- 300 test cases were executed, with 246 passing successfully.

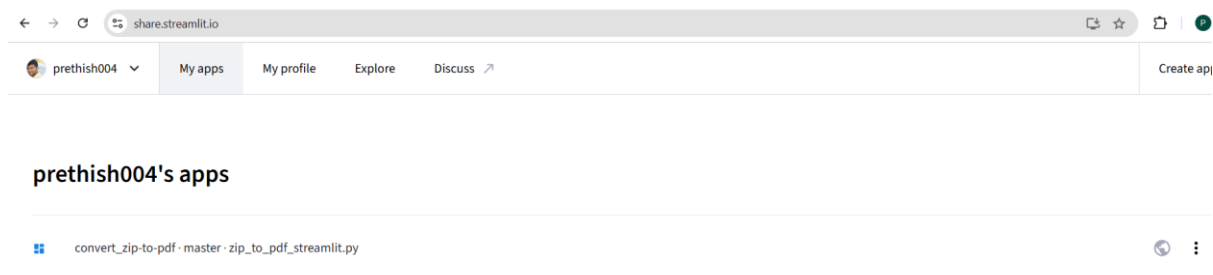


# Challenges

- **300 test cases:** This was the total number of tests, where 246 tests passed successfully.
- **Remaining Failures:** Failures were mainly due to limitations in RAM, storage capacity, and issues with uploading large files.
- In some cases, the large size of images exceeded the upload or processing capacity.

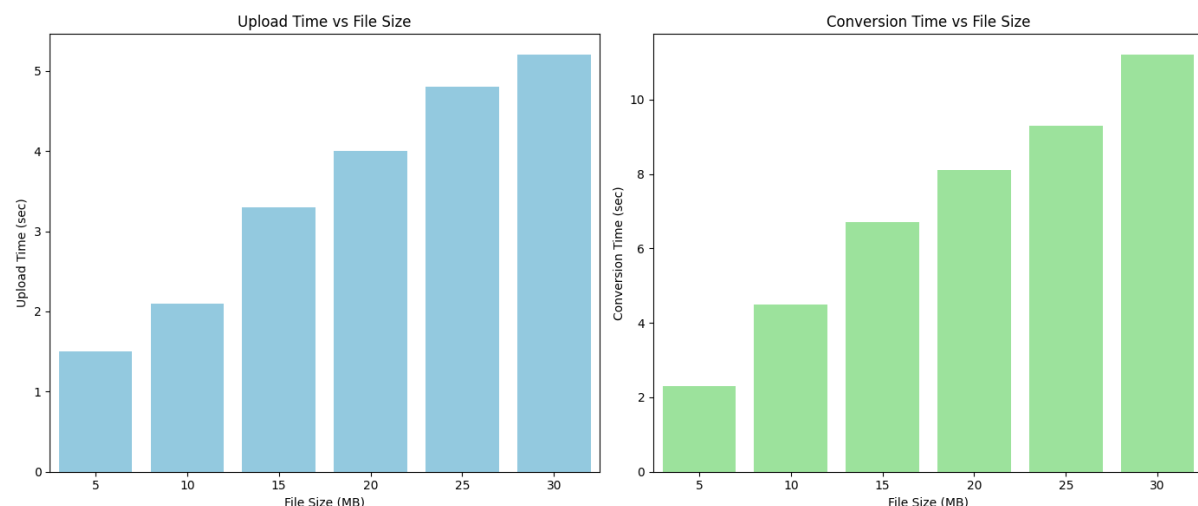
# Hosted Area

- The application is hosted on a cloud server (Streamlit), where the processing is done.
- Discuss the limitations and scalability of the cloud infrastructure when dealing with large ZIP files.



# Metrics and Demerits

- **Metrics:** Performance of the app in terms of upload speed, conversion time, and PDF generation time.
- **Demerits:** Issues related to RAM and storage constraints for larger files. Limited capacity for handling ZIP files over a certain size.



# Result


- The app successfully converts ZIP files into PDFs, ensuring a high-quality result.

- It allows users to reorder files, resize images to fit A4 dimensions, and ensures the output PDF is formatted correctly.

# ZIP to PDF Converter

Upload 1 to 15 ZIP files containing images. Rearrange the ZIP file order, and we'll create a PDF based on your selection. The first ZIP file's name will be used for the PDF.

Upload ZIP files



Drag and drop files here

Limit 1GB per file • ZIP

Browse files



1 (2).zip 32.5KB



2 (2).zip 377.2KB



Reorder ZIP files:

1 (2).zip ×

2 (2).zip ×

× ▼

PDF successfully created: 1 (2).pdf

Download PDF