ERRORS

Error 1:

Description:
Immediately after uploading any file, the app crashes with:

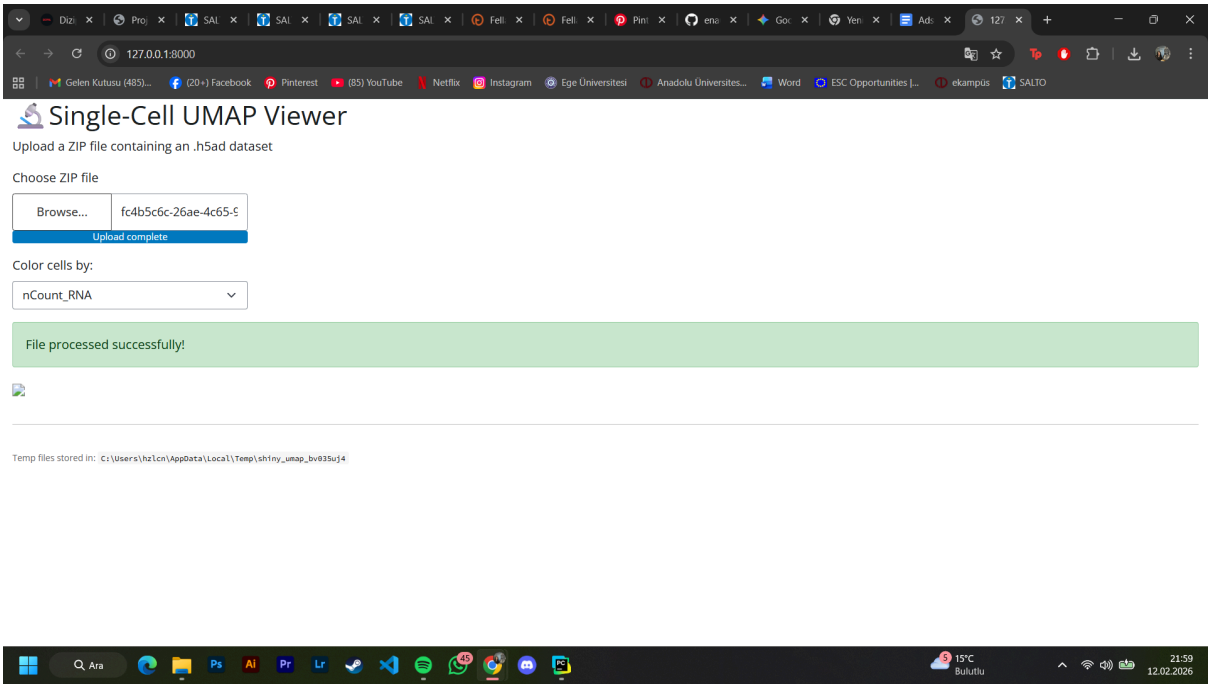
Step-by-Solution:

Step 1:
file_info = input.file_upload()
# file_info[0] is a dict with keys: ['name', 'size', 'type', 'datapath']
# file_info[0]["datapath"] is a STRING path, not a file object

Step 2:
src_path = file_info[0]["datapath"] # This is a string path

Open the file at that path
with open(src_path, "rb") as src:
    file_content = src.read() # Now you can read it

Error 2:



Description:

After uploading a valid ZIP file, the dropdown populates correctly but the UMAP plot shows a broken image icon (empty box with red "X").

Step-by-Step Solution:

Step 1: Locate the broken line

Find this line in your umap_plot() function:

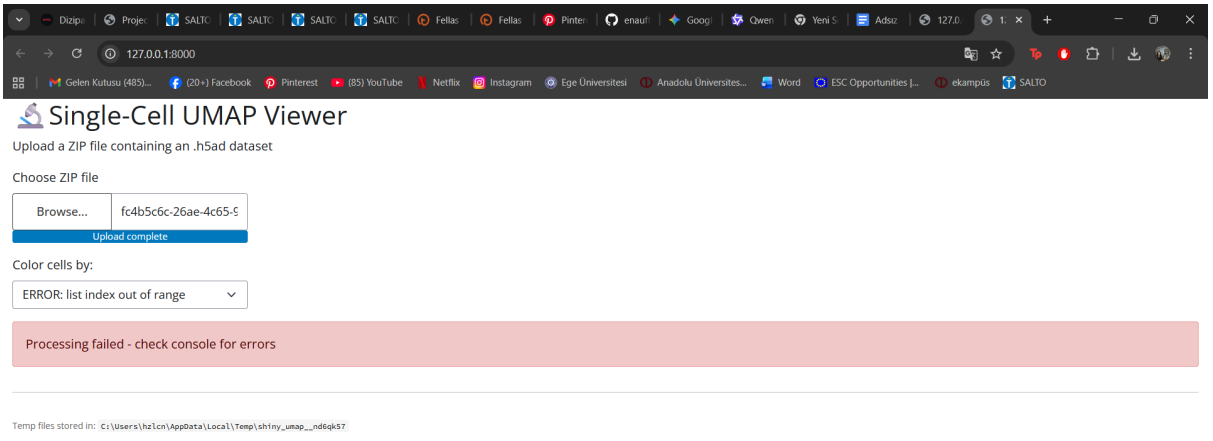
return ui.img(src=f"image/png;base64,{img_base64}", style="max-width: 100%;")

Step 2: Add the `` prefix

Fixed Code:

return ui.img(src=f"{img_base64}", style="max-width: 100%;")

Error 3:



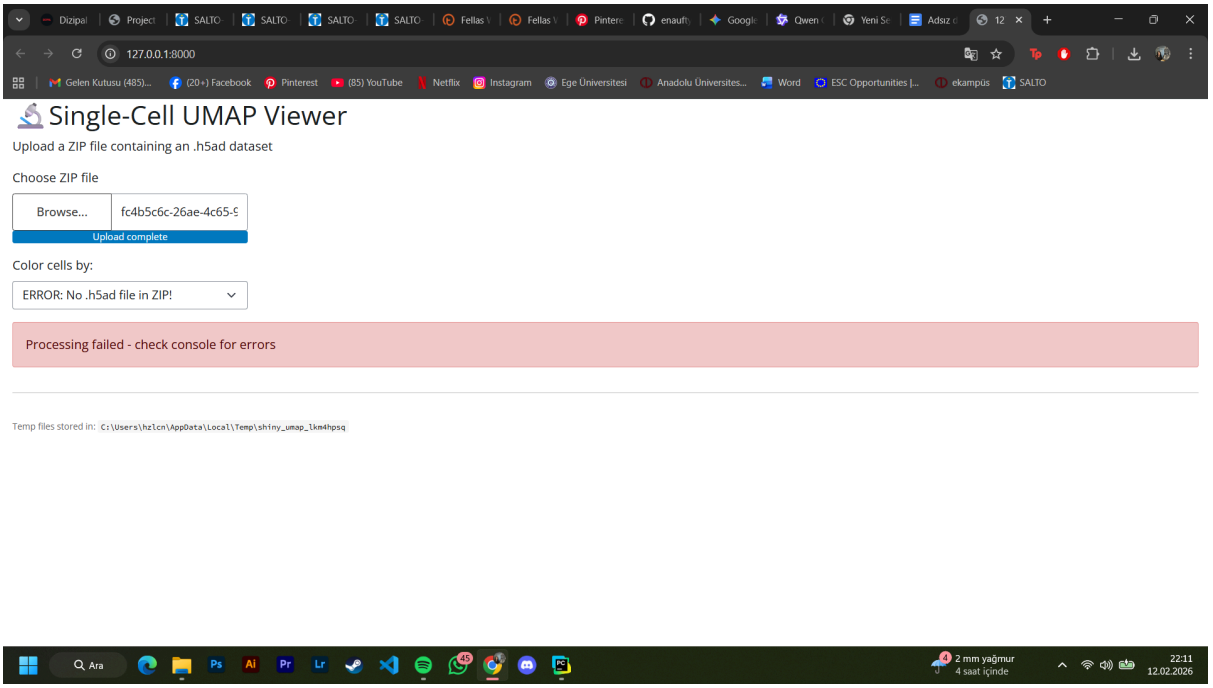
Description:
Uploading a ZIP file that contains no .h5ad file crashes the entire app with:

Step-by-Step Solution:
Step 1: Add validation before accessing the list
h5ad_files = [f for f in os.listdir(TEMP_DIR) if f.endswith(".h5ad")]

Check if list is empty

```
if not h5ad_files:
    # Show user-friendly error in UI
    ui.update_select("color_by", choices=["ERROR: No .h5ad file found in ZIP!"])
    print(f" ZIP contents: {os.listdir(TEMP_DIR)}")
    return  # Exit early
```

Error 4:



Description:
When a user uploads a ZIP file that does not contain any .h5ad file (e.g., ZIP with only text files or images), the application completely crashes with:

Step-by-Step Solution:
Step 1: Identify the Vulnerable Code Section
Locate the file extraction logic in your handle_upload() function (typically lines 40-50):
Find this section in your code

```
h5ad_files = [f for f in os.listdir(TEMP_DIR) if f.endswith(".h5ad")]
extracted_path = os.path.join(TEMP_DIR, h5ad_files[0])  # ← VULNERABLE LINE
h5ad_path.set(extracted_path)
```

Step 2: Add Validation Before List Access:
Add this validation block

```
h5ad_files = [f for f in os.listdir(TEMP_DIR) if f.endswith(".h5ad")]
print(f"Found .h5ad files: {h5ad_files}")
```

Important: Check if list is empty BEFORE accessing index

```
if not h5ad_files:
    # Get contents of ZIP for helpful debugging
    zip_contents = os.listdir(TEMP_DIR)

    # Create user-friendly error message
    error_msg = f"ERROR: No .h5ad file in ZIP! Found: {', '.join(zip_contents[:5])}"
    if len(zip_contents) > 5:
```

```
    error_msg += f" (+{len(zip_contents)-5} more files)"

# Log to terminal for debugging
print(f" {error_msg}")
print(f"   Full ZIP contents: {zip_contents}")

# Update UI with error message
ui.update_select("color_by", choices=[error_msg])

# Reset state
h5ad_path.set(None)

# EXIT EARLY - prevent crash
return
```