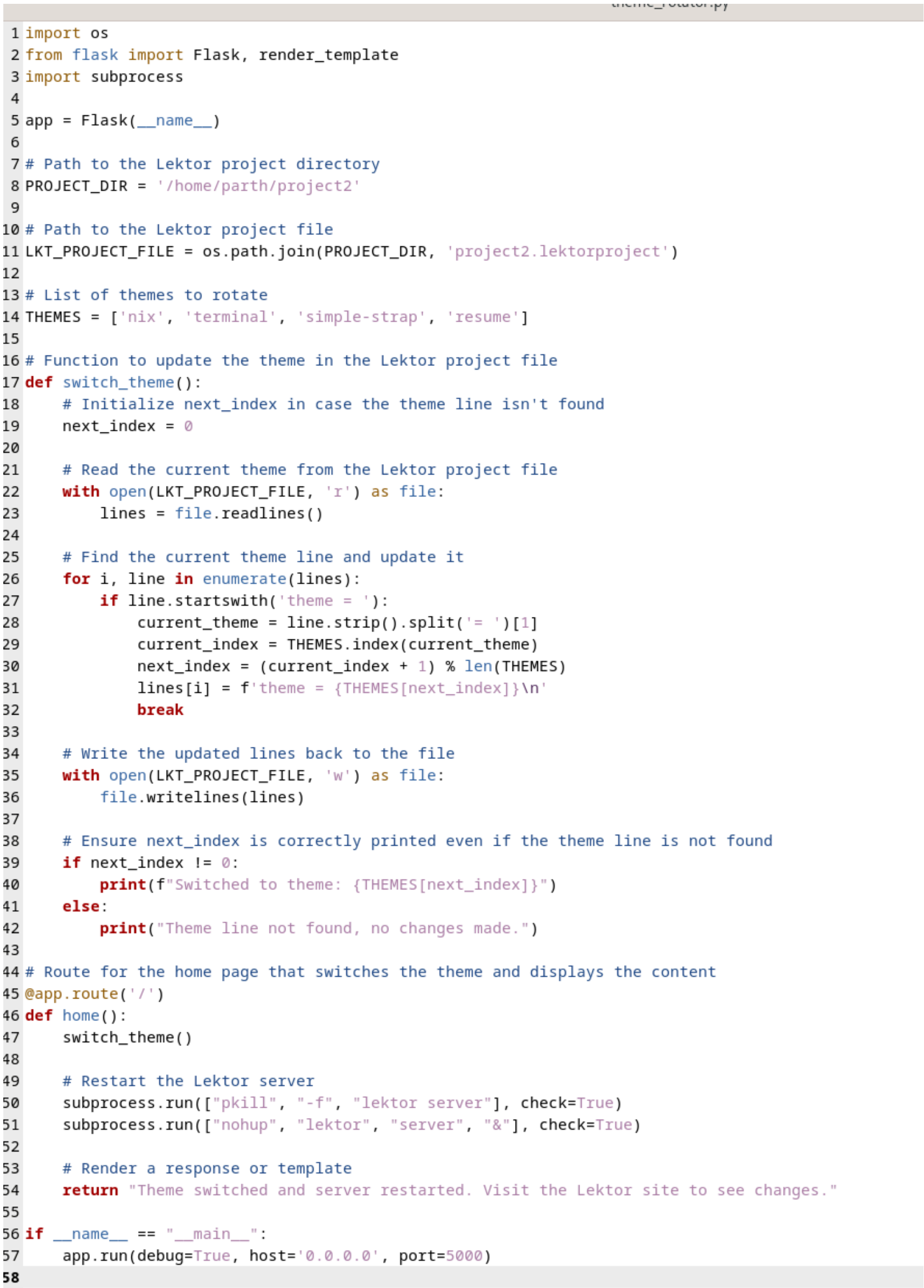
Journal 3:

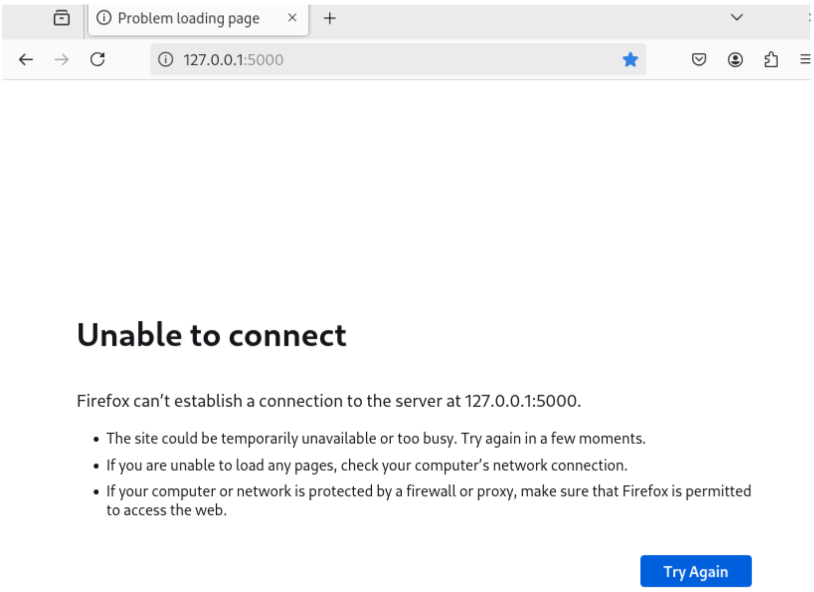
- This week was very challenging. I struggled a lot trying to fix problems.
- Wednesday, I tried to work with different themes with Lektor, and also do scripting.
- So initially, I didn't know the themes that have to be changed are themes from the Lektor documentation. I found that out on Friday.
- So Wednesday, I wasted my time trying to figure out a bash script to make things work and changing the css file.
- Then, on Friday during the lab, I figured out how to make the theme change one by one. First you need to find the themes, which are here → https://github.com/lektor/lektor-themes , then clone it in your lektor project folder under the themes folder.
- Once you do that, you have to edit the .lektorproject file and add:
  [project]
  themes = lektor-theme-name
- Once you do that, you have to do lektor build, in the terminal and lektor server, to see on your localhost the changes of the webpage.
- I figured that out in the Friday lab. On Saturday, I was trying to figure out the scripting, but wasn't able to figure it out. I even asked chat gpt to ask for a script to change the themes every minute, but it didn't work, I was getting errors with running the server..
- Then, my teammate recommended that I use a python script to do it, since he managed to make it work.

- So, I did a python script that looked like this:



```python
import os
from flask import Flask, render_template
import subprocess

app = Flask(__name__)

# Path to the Lektor project directory
PROJECT_DIR = '/home/parth/project2'

# Path to the Lektor project file
LKT_PROJECT_FILE = os.path.join(PROJECT_DIR, 'project2.lektorproject')

# List of themes to rotate
THEMES = ['nix', 'terminal', 'simple-strap', 'resume']

# Function to update the theme in the Lektor project file
def switch_theme():
    # Initialize next_index in case the theme line isn't found
    next_index = 0

    # Read the current theme from the Lektor project file
    with open(LKT_PROJECT_FILE, 'r') as file:
        lines = file.readlines()

    # Find the current theme line and update it
    for i, line in enumerate(lines):
        if line.startswith('theme = '):
            current_theme = line.strip().split('= ')[1]
            current_index = THEMES.index(current_theme)
            next_index = (current_index + 1) % len(THEMES)
            lines[i] = f'theme = {THEMES[next_index]}\n'
            break

    # Write the updated lines back to the file
    with open(LKT_PROJECT_FILE, 'w') as file:
        file.writelines(lines)

    # Ensure next_index is correctly printed even if the theme line is not found
    if next_index != 0:
        print(f"Switched to theme: {THEMES[next_index]}")
    else:
        print("Theme line not found, no changes made.")

# Route for the home page that switches the theme and displays the content
@app.route('/')
def home():
    switch_theme()

    # Restart the Lektor server
    subprocess.run(["pkill", "-f", "lektor server"], check=True)
    subprocess.run(["nohup", "lektor", "server", "&"], check=True)

    # Render a response or template
    return "Theme switched and server restarted. Visit the Lektor site to see changes."

if __name__ == "__main__":
    app.run(debug=True, host='0.0.0.0', port=5000)

```

- I never coded in python so I did ask chatgpt for some help.
- This, from what I saw in the terminal, was the closest thing to success, since I saw a message saying the themes were switching. The only problem was it wasn't switching on the web. After 1 minute, when the "theme switches", I see this on the webpage:



Unable to connect

Firefox can’t establish a connection to the server at 127.0.0.1:5000.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer’s network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the web.

- I tried to fix this, asking the AI again, which did not help me at all. Because it said I had to add a time interval to let the server restart, but that did not work. According to AI:

# Wait a bit to allow the server to start
time.sleep(5) → (solution)
# Adjust this wait time as needed # Check if the server is running

- So this week I still wasn’t able to figure out scripting, but my teammate has been able to figure out a solution. The only problem with that solution, is that it is not every minute, it is every 12 hours.
- The guide/manual of this project is also in progress, as well as the powerpoint for Tuesday.

Git Repository: https://github.com/ParthPatelA/UnixProject.git