**Part 1: Identify the Problem**

**-Automated file Organiser-**Create a script to sort and organize files on a computer by type into designated folders and set a time to make it happen periodically.

-**Local Event Calendar Scraper** - Build a tool to scrape local websites or event pages and consolidate the information into a user-friendly calendar format.

**Part 3: Reflect on the Solution**

**Automated file Organiser**

**-** Working on this file organizer project made me discover and learn lot of stuffs. I believe I chose a suitably challenging problem that aligns well with real-world applications and also solves my problem of having an unorganised file system. it involved integrating multiple Python libraries like os, shutil, time and thinking through edge cases like handling unsupported file types or pre-existing folder structures. At the same time, it wasn’t overly complex, allowing me to focus on learning without being overwhelmed.

The most challenging aspect was ensuring that the script could handle unexpected situations well. For instance, managing file permissions, avoiding conflicts with files that already exist in target folders, and making the program robust enough to skip directories while still organizing all valid files required careful planning and testing.

To learn about the libraries used, I relied heavily on the official Python documentation for os and shutil. Stack Overflow and geeksforgeeks was also a great resource for troubleshooting specific errors and understanding best practices for handling file operations in Python. These resources provided clarity on how to manage paths dynamically and use functions like os.makedirs() and shutil.move() effectively.

The most valuable thing I learned from this assignment is the importance of handling edge cases in automation scripts. Initially, I overlooked scenarios like files with no extensions or pre-existing folders with the same name, which led to errors during testing. Addressing these challenges taught me to think critically about real-world implementation issues and to write cleaner, more resilient code.

To test the completed product, I created a mock directory with various file types, including some edge cases, like hidden files, directories, and files with uncommon extensions. I ran the script multiple times to verify that it worked as expected in different scenarios. I have also added a time loop for it to repeat the execution after 24hr so that everyday the folder gets organised, unfortunately I have not been able to test it as it was a last minute addition and I didn’t have 24 hrs to test it.

Overall, I thoroughly enjoyed working on this assignment. It was very intresting to see how a small problem could be solved by scripting and turned into a practical tool that has the potential to save time , improve organization and make me look cool infront of my non techy friends. This project also inspired me to think about other automation scripts I could create, It got me interested in Python all over again, hopefully I keep making more such tools.