WORKING DIRECTORY GENERATOR (Project 3)

GROUP 3:
Josiah MUTIE
Belise KANZIGA
Djadida UWITUZE
Patrick NIZEYIMANA
Kutlo Gaone KEJANG

African Institute Mathematical Science

October 8, 2025

Overview

1. Introduction
2. Project Objectives
3. Methodology
4. Key Features
5. Results
6. Applications
7. Project Impact
8. Real-World Applications
9. Conclusion
10. Q & A

Introduction

Managing coursework for a diverse student body at AIMS Rwanda presents a significant organizational challenge. When every student manually creates their own folders, the result is often a disorganized and inconsistent workspace, with naming errors, missing directories, and issues caused by special characters in international names. Our project addresses this directly by developing an automated Python tool that transforms simple student and course lists into a perfectly structured, standardized directory hierarchy for an entire cohort, saving time, eliminating manual errors and ensuring every student has a consistent and reliable workspace from day one.

Project Objectives

- Automatically create main directory: `AIMS-Rwanda-Workspace`
- Create standardized subdirectories for each student: `LastName, FirstName`
- Normalize names (remove accents, special characters)
- Create course subfolders for each student
- Generate placeholder `README.txt` files
- Validate input CSV files
- Prevent overwriting existing data
- Provide interactive menu system

Technical Approach

Modular Design

- Separate functions for each task
- Clear, maintainable code

Input Files

- `Student_List.csv`
- `Course_List.csv`

Python Libraries Used

Standard Library Only

- **os**: File operations
- **csv**: Read input files
- **unicodedata**: Name normalization
- **re**: Pattern cleaning
- **typing**: Type hints

Key Features

Encoding-Safe Reading

- Handles multiple encodings: UTF-8, UTF-8-SIG, Latin-1, cp1252
- Prevents crashes with international characters
- Fallback mechanism for robustness

Name Normalization

- Converts accented characters to ASCII: François → Francois
- Removes special characters and punctuation
- Ensures filesystem compatibility

Name Cleaning Example

Before Cleaning

```
François,␣Lukasz
Maria-José
O’Connor, Patrick
```

After Cleaning

```
Francois, Ukasz
Maria-Jose
OConnor, Patrick
```

Interactive Menu System

User-Friendly Features

- Default values for easy use
- Option to rerun without restarting

Menu Options

1 Generate new workspace

2 Quit

Testing and Results

Test Setup

- 19 student names
- 5 courses
- 95 course folders + 19 student folders generated

Successful Outcomes

- All directories and files created correctly
- Accented and hyphenated names handled properly
- Existing data preserved (no overwriting)
- Multiple encodings supported

Directory Structure

Generated Workspace

```
AIMS-Rwanda-Workspace/
  Mutie, Josiah/
    Introduction_to_Python_Programming/
      README.txt
    Statistical_Regression/
      README.txt
    Physical_Problem_Solving/
      README.txt
  Kanziga, Belise/
    Introduction_to_Python_Programming/
      README.txt
  ...
```

Practical Applications

- University lab folder management
- Corporate onboarding scripts
- Research collaboration setups
- Data science project templates

Broad Practical Applications

Academic Educational

- **University Course Management**
  Auto-setup for each semester's student workspaces
- **Research Lab Organization**
  Standardized folders for experiments and data
- **Thesis Supervision**
  Consistent structure for multiple student projects

Technical Development

- **Software Project Templates**
  Consistent skeleton for new code repositories
- **Data Science Workflows**
  Standardized folders for data, models, and results
- **DevOps Automation**
  Automated environment setup for team members

Conclusion

1. **Robust Automation Solution**
   Fully functional tool eliminating manual workspace setup
2. **Immediate Practical Value**
   Time savings and error reduction for AIMS Rwanda
3. **Foundation for Future Development**
   Scalable codebase ready for enhanced features

References

- Python Software Foundation. *Python 3.12 Documentation*
- Schwarz, N. (2019). Encoding and character handling approaches
- Smaragdakis, Y. and Batory, D. (2000). Application generators

Q & A

Thank your for your attention!

Questions?