import os import subprocess

def generate_materials_informatics_course_markdown(course_title=“Introduction to Materials Informatics”, output_format=“md”): """ Generates a Markdown file with a course structure for Materials Informatics.

```
Args:
    course_title (str): The title of the course.  Defaults to "Introduction to Materials Inf
    output_format (str): The output format. Can be "md" (Markdown) or "pdf".
        Defaults to "md".
Returns:
    None.  Saves the generated file.
"""

# Ensure the filename is valid
filename = course_title.replace(" ", "_").lower() + "." + output_format

# Course content
course_content = f"""
```

{course_title}

Course Overview

This course provides an introduction to the emerging field of Materials Informatics, which applies data science and machine learning techniques to accelerate the discovery, design, and development of new materials. Students will learn how to acquire, analyze, and model materials data to solve real-world materials science problems.

Prerequisites

- Basic understanding of materials science concepts
- Familiarity with programming concepts (Python preferred)
- Linear algebra and statistics fundamentals

Course Objectives

Upon successful completion of this course, students will be able to:

- Understand the fundamental concepts of Materials Informatics.
- Apply data mining and machine learning techniques to materials data.
- Utilize Python libraries for materials data analysis and modeling.
- Design and execute materials discovery workflows.
- Evaluate the performance of machine learning models in materials science.
- Communicate materials informatics findings effectively.

Course Structure

The course is divided into the following modules:

Module 1: Introduction to Materials Informatics

- What is Materials Informatics?
- History and evolution of Materials Informatics
- The materials data landscape
- Applications of Materials Informatics
  - Materials discovery
  - Materials design
  - Process optimization
- Challenges and opportunities in the field

[Image of materials science applications, e.g., a graph showing accelerated discovery, could go here]

Module 2: Materials Data Acquisition and Management

- Sources of materials data
  - Experimental databases
  - Computational databases
  - Materials characterization techniques
- Data formats and standards
- Data cleaning and preprocessing
  - Handling missing data
  - Data normalization and scaling
- Data storage and management

    - Relational databases
    - NoSQL databases
    - Cloud-based solutions

[Image of a materials database or data flow diagram could go here]

Module 3: Data Mining and Machine Learning for Materials Science

- Fundamentals of machine learning
    - Supervised learning
    - Unsupervised learning
    - Reinforcement learning (brief overview)
- Regression techniques
    - Linear regression
    - Polynomial regression
    - Support vector regression
- Classification techniques
    - Logistic regression
    - Decision trees
    - Random forests
    - Support vector machines
- Clustering techniques
    - K-means clustering
    - Hierarchical clustering
- Dimensionality reduction
    - Principal component analysis (PCA)
    - t-distributed stochastic neighbor embedding (t-SNE)

Module 4: Materials Modeling and Simulation

- Connecting materials data to models
- Physics-based modeling vs. data-driven modeling
- Building surrogate models
  - Gaussian process regression
  - Neural networks
- Model validation and uncertainty quantification
- Active learning for materials discovery

[Image of a machine learning workflow for materials modeling could go here]

Module 5: Applications of Materials Informatics

- Case studies in materials discovery
  - High-throughput computational screening
  - Autonomous experimentation
- Case studies in materials design
  - Tailoring materials properties
  - Inverse design
- Case studies in process optimization
  - Predictive maintenance
  - Process control

Module 6: Advanced Topics in Materials Informatics

- Graph neural networks for materials
- Natural language processing for materials science literature

- Multi-modal materials data analysis
- Explainable AI (XAI) for materials science
- The future of Materials Informatics

Assessment

- Homework assignments (40%)
- Midterm exam (20%)
- Final project (30%)
- Class participation (10%)

Grading Scale

Grade	Percentage
A	90-100
B	80-89
C	70-79
D	60-69
F	<60

Recommended Textbooks

- **Title:** *Python Data Science Handbook*
  - **Author:** Jake VanderPlas
  - **Publisher:** O'Reilly Media
- **Title:** *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*
  - **Author:** Aurélien Géron
  - **Publisher:** O'Reilly Media
- **Title:** *Mathematics for Machine Learning*
  - **Authors:** Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong
  - **Publisher:** Cambridge University Press

Additional Resources

- The Materials Project
- NIST Materials Data Repository
- Journal of Materials Informatics

Note: This is a sample course structure and may be adjusted based on the specific needs of the students and the instructor. """

```
# Ensure the filename is valid
filename = course_title.replace(" ", "_").lower() + "." + output_format
```

```
# Write the markdown content to a file
try:
    with open(filename, "w") as f:
        f.write(course_content)
    print(f"Successfully created course structure markdown file: {filename}")
except Exception as e:
    print(f"Error writing to file: {e}")

if output_format == "pdf":
    try:
        # Get the path to the default template
        default_template_path = subprocess.run(
            ["pandoc", "--print-default-data-dir"],
            check=True,
            stdout=subprocess.PIPE,
            text=True
        ).stdout.strip()
        default_template = os.path.join(default_template_path, "templates", "default.latex")

        subprocess.run(
            ["pandoc", filename, "-o", course_title.replace(" ", "_").lower() + ".pdf", f"--
            check=True,
            stderr=subprocess.PIPE,  # Capture standard error
            text=True
        )
        print(f"Successfully converted {filename} to PDF using template: {default_template}"
    except subprocess.CalledProcessError as e:
        print(f"Error converting to PDF: {e}")
        print(f"Pandoc error output:\n{e.stderr}")
        print("Make sure Pandoc and LaTeX are installed and in your system's PATH.")
    except FileNotFoundError:
        print("Error: Could not find Pandoc's default template.  Ensure Pandoc is correctly
```

if **name** == “**main**”: generate_materials_informatics_course_markdown(output_format=“pdf”)