CSC2510 Final Project

Company Name: DeelTech Solutions

Background:
DeelTech Solutions is a fast-growing technology consulting firm specializing in **cloud infrastructure automation and DevOps services**. The company has recently secured a major contract to help universities modernize their IT systems and streamline faculty collaboration tools. DeelTech prides itself on security, scalability, and automation.

The Challenge:
DeelTech has been hired to **create a secure internal portal for faculty members** at Tennessee Tech University's Computer Science Department. This portal will allow professors to manage course materials, collaborate on research, and access cloud-based resources. Before the portal can go live, DeelTech needs **user accounts for all CS faculty members** to be created in the system.

Your Role:
You are part of DeelTech's **DevOps onboarding team**. Your task is to:

1. **Gather professor names and details** from the TNTech Computer Science Department website.
2. **Create user accounts** for each professor in the new system using a standardized format (e.g., username, email, temporary password).
3. Ensure the accounts follow **security best practices** to include anti-piracy techniques.
4. Automate the process using **Bash scripting** or other tools covered in class.

Why This Matters:
DeelTech's reputation depends on delivering **efficient, automated solutions**. Manual account creation is error-prone and slow. Your automation will demonstrate the power of DevOps principles and scripting in real-world IT operations.

Technical Requirements:
You must design a program in BASH for the Network Operations (NetOps) team to deploy. A main script may call additional sub-scripts. The code will deployed on Ubuntu 24.04 with BASH v5.2.21. DeelTech uses virtualization technology. The Ubuntu instance will run on a x86 architecture virtual machine using Virtual Box 7.1.12. The program will need to perform the following tasks:

1. Parse names from a provided website
2. Create users on the Ubuntu system given specific criteria for username/password structure
3. Manual "add user" function utilizing defined username and password structure
4. Use AI assist in building part of the script to implement a licensing function

Assignment and Grading Details:
Team members will be graded on the presentation of the material, the working code, documentation, and maintenance of the application. Additionally, peers will evaluate group members which will determine what percentage of the overall grade they should receive.

1. Presentations = 40pts
2. Code = 40pts
3. Usage Manual / Documentation = 10pts
4. CI/CD = 10pts
5. Peer Review = % of total accumulated team points

Rubric:
Teams must provide presentations at various milestones, working code, usage documentation, and they also must provide maintenance for the application:

1. Phase 1: Plan / Code
    1. Present coherent approach to solving the problem
    2. Each team member must present during presentation
    3. Teams should use Power Point to lay-out and document plans
        1. Team name
        2. Introduction / Agenda
        3. Scope of effort / definition of work / problem to be solved
        4. Roles
        5. Recommended changes/improvements to requirement (optional)
        6. Development approach by each member tackling a task
        7. Collaboration methodology / tools
        8. Delivery timeline with possible pitfalls
            1. Probable scenario (e.g. team will deliver full product on-time)
            2. Best case scenario (e.g. team will deliver ahead of schedule and recommend additional capabilities)
            3. Worst case scenario (e.g. team will not accomplish all tasks, but will take X action to mitigate)
        9. CI/CD approach commitment
        10. Conclusion
        11. Acceptance discussion (does not count against time)
    4. Team will keep presentations to eight (8) minutes ($\pm$30 seconds for full credit)
2. Phase 2: Build / Test / Deploy
    1. Deliver code to deployment team (iLearn)
    2. Provide documentation for usage of application
    3. Partial credit will be given based on completeness of code
    4. Demonstration of application
        1. Live demo of integrated application on student provided test environment
        2. Explanation of usage manual
3. Phase 3: CI/CD
    1. Continuous integration and delivery
    2. Final presentation

1. Must use Power Point
2. CI/CD experience
3. Lessons learned
4. Process improvement for next time
5. Must be more than four (4) minutes long

Detailed Technical Requirements:
Teams must follow these provided guidelines for code:

1. Part 1
    a. Primary interface to choose program action
    b. Default action = web scraper and parser
        i. Scrape/download CS department site: https://www.tntech.edu/engineering/programs/csc/faculty-and-staff.php
        ii. Parse positions for names of faculty and staff
            1. first & last names
            2. no titles
2. Part 2
    a. Create username for faculty given new schema = first.last
    b. Create password from full name and phrase DEELTECH = firstnamelastnameDEELTECH
    c. Create Ubuntu users with new user/pass
3. Part 3
    a. Optional “Add User” function
    b. Takes first name and last name as input
    c. Generates username = first.last
    d. Password = firstnamelastnameDEELTECH
4. Part 4 (must use AI on key obfuscation technique)
    a. The main script should enforce licensing
        i. On first run of the program:
            1. A [license].dat file should be placed somewhere on the file system
            2. “license” in filename maybe a random name chosen by the team
            3. If a license file does not exist, then the user must be prompted for a key
        ii. A 16 digit license key should be provided by email to the stakeholders
            1. Scrum master should email license key (this is not automated)
            2. This will be the students emailing the key to the instructor and TA
        iii. The program should always check for a valid license file before running

iv. The key should be obfuscated in the main program so it cannot be seen in clear text

Presentation Grading:
AI may be used for any portion of the code, product manual, or presentations. AI must be used for at least part of the license generation component of code. Student will be graded on the overall project and assume risk associated with using AI for non-required parts of project.

- 10pts = Completing all three presentations
- 10pts = Phase 1 presentation
  - 3pts = Staying within time
  - 2pts = All members present some material
  - 4pts = Students address all 10 topic points stipulated above
  - 1pts = Students approach the presentation professionally and limit "ums"
- 10pts = Phase 2 presentation
  - 3pts = Staying within time
  - 2pts = All members present some material
  - 4pts = Students demonstrate live (not recorded) usage of required components of program and discuss usage documentation
  - 1pts = Students approach the presentation professionally and limit "ums"
- 10pts = Phase 3 presentation
  - 3pts = Staying within time
  - 2pts = All members present some material (each student may discuss a lesson learned)
  - 4pts = Students address all three (3) topic points stipulated above
  - 1pts = Students approach the presentation professionally and limit "ums"

User Manual Grading:
The user manual will be graded on professional tone, completeness of explanations, and grammar.

AI Statement:
AI may be used for any portion of the code, product manual, or presentations. AI must be used for at least part of the license generation component of code. Student will be graded on the overall project and assume risk associated with using AI for non-required parts of project. Use of AI should be cited where appropriate.