COP 4521 Spring 2023
Project Requirement Description

Total Points: 100
Due: Wednesday, 04/26/2023

1 Objective

The semester project is a creative exercise where students are expected to take what they have learned about Information Management, Secure Computing, Parallel Computing, and Distributed Computing (both in and out of the classroom) and create a polished Python application solving a real problem. Programming in Python involves not only understanding the syntactic rules and constructs of the language, but also the ability to research the libraries that are needed for a task, as well as organizing the code well. The Python API is very large and has many pre-built features. After this project, you will have some experience looking into existing Python libraries and designing a small application in the areas in focus for this course.

2 Specifications

You are required to design and implement an application that solves a quasi real-world problem in Python. This is a group project. A group is a minimum of 3 people and a maximum of 5people. The project should prominently feature aspects of Information Management (including role Based Access Control), Secure Computing, Parallel Computing, and Distributed Computing. The final deliverable will be due at the end of the term.

2.1 Requirements

1. You may choose from a variety of applications. Ideally, each group will work on something different. Some ideas include:
    - A web application the involves a Web stack and makes use of front-end and back end Python libraries.
    - A web application that can solve engineering/math problems using the numerical libraries.
2. The project needs to be of an appropriate difficulty level. Teams need to pick ideas that would involve enough effort for all the people.
3. The project needs to have a component that interacts with a database (not a dataset in a file) along with role based access.
4. Your grade for the project will be based on:
    (a) Overall functionality. (15%)
    (b) Whether all of the requirements of the original proposal were met. (15%)
    (c) The size and quality of your individual contribution (in Python code) to the project. (40%)

(d) How your individual contribution integrates into the rest of the codebase. (20%)

(e) Overall quality of code. (PEP8, code organization, coverage, complexity, test system, build system, documentation). (10%)

5. All projects should be done using a repository. If group members do not agree on the division of labor during the evaluation, your repository contributions/activity will be used to assess this component.

3 Milestones

The project will have several milestones The final grade for the project will be the sum of your grades for the milestones.

- **Proposal - due 01/23/2023 - 5%** Your proposal is essentially an email (one email per group) that clearly states the following.
  - Your team members
  - The project title, with a small description of what you hope to achieve.
  - A couple of lines on how you plan to split up the workload.

  You will receive an email acknowledging the receipt of your proposal. All proposals will be reviewed for suitability, which is dependent on the group size, complexity of proposal, experience level of group members, etc. Student groups are not bound to their proposal and may make modifications to their original idea and/or approach. However, you may want to consult with the instructor before making major changes that may affect the suitability of the project.
- **Status Report 1 - due 02/20/2023 - 5%** This will also be an email (one email per group) with a couple of paragraphs detailing the progress you've made. By this time, you should have
  - Created a GitHub repository for your project and invited the instructor as a collaborator.
  - Designed a detailed project plan.
  - Started development on the project.
- **Status Report 2 - due 03/27/2023 - 10%** This will be an email to the instructor containing the following:
  - Any changes from the original plan
  - A list of Python libraries you are using/going to use for your project.
  - A summary of what you have already achieved.
- **Final Project Delivery - due 04/26/2023 - 80%** You should pack all relevant files for your assignment into one tarball. The file should contain:
  - All necessary .py files that make your program work (please do not include resource files like images or datasets).
  - A README file (this can be in text format or in html format) that contains any extra documentation for your project. This should include
    * A description of the problem you are trying to solve.
    * Any details regarding instructions for the user interface that is beyond the obvious.
    * A list of Python libraries you are using.
    * A list of other resources.

    - Descriptions of any extra features implemented (beyond the project proposal).
    - Include a description of the separation of work (who was responsible for what pieces of the program).

Submit the final deliverable through Canvas. Only one submission is required per team.

4 Demonstration:

You will be required to demonstrate your application sometime during the last two weeks of classes. We will determine the schedule later in the course. You will be assessed in the areas described above. More than anything, I want to see that you put forth a solid effort to achieve the goals of your original proposal. Even if you fall short of the proposal, a discussion of the obstacles and lessons learned will indicate to me that you put effort into completing the project.

5 Guidelines for researching resources

- It would probably be a good idea to look at examples and tutorials on the internet, to help you find techniques and tips for implementing topics not discussed in class.
- If you find a resource that gives you some ideas – and especially if you model a portion of your code after another example – be sure to cite your source in your documentation
- The coding, however, must be yours! You are to implement your own logic (i.e. you may not just find an existing implementation and use code found on web searches).
- Code needs to be generated by you (and not auto-generated)
- You may use icons, graphics pictures, sounds, datasets, etc that you find online. If and when possible, cite your source, and be sure to pay attention to any copyright notices regarding such resources. (Example: If you choose to design an AI for a card game, it would be a good idea to find image files of cards, for a nicer interface – it's best to find public domain images for this sort of thing).