CMPSC 481
Software Innovation II
Spring 2019

Software Project Two
Assigned: Thursday, March 28, 2019 at 1:30 pm
Due: Thursday, May 2, 2019 at 1:30 pm

Objectives

To use GitHub and the GitHub flow model to collaboratively engineer, deliver, and evaluate a software product. Along with using GitHub features like the issue tracker and reviewing pull requests, in this assignment you will use Markdown to complete technical writing tasks and the Python programming language and many Python packages (e.g., Pytest for automated testing) to create production-quality open-source programs for use in an educational setting. As a side effect of working in a team, you will also experience challenges (e.g., the creation of merge conflicts in a version control repository) that force you to develop practical solutions. You will also gain experience in interacting with team members, technical leaders, members of industry, and the course instructor. Students will work together in a development team while mastering the technical and professional skills in the field of software engineering, working towards becoming a recognized software innovator who can design, implement, and release software in production use at Allegheny College. Finally, students will gain experience to distinguish themselves as software innovators who can use systems such as Amazon Elastic Beanstalk and Amazon Lambda for serverless computing.

Suggestions for Success

- **Use the laboratory computers**. The computers in the departmental laboratories feature specialized software for completing this course's assignments. If it is necessary for you to work on a different machine, be sure to regularly transfer your work to a laboratory machine so that you can check its correctness. If you cannot use a laboratory computer and you need help with the configuration of your own laptop, then please carefully explain its setup to a teaching assistant or the course instructor when you are asking questions.
- **Explore teamwork and technologies**. While certain aspects of these assignments will be challenging for you, each part is designed to give you the opportunity to learn both fundamental concepts in the field of computer science and explore advanced technologies that are commonly employed at a wide variety of companies. To explore and develop new ideas, you should regularly communicate with your team and/or the teaching assistants and tutors.
- **Hone your technical writing skills**. Computer science assignments require to you write technical documentation and descriptions of your experiences when completing each task. Take extra care to ensure that your writing is interesting and both grammatically and technically correct, remembering that computer scientists must effectively communicate and collaborate with their team members and the tutors, teaching assistants, and course instructor.

Creating a Suite of Educational Software Tools

To access this assignment, you should go into the `#announcements` channel in our Slack team and find the announcement that provides a link for it. Copy this link and paste it into your web browser. Now, you should accept the laboratory assignment and see that GitHub Classroom created a new GitHub repository for you to access the assignment's starting materials and to store the completed version of your assignment. Specifically, to access your new GitHub repository for this assignment, please click the green “Accept” button and then click the link that is prefaced with the label “Your assignment has been created here”. If you accepted the assignment and correctly followed these steps, you should have created a GitHub repository with a name like “Allegheny-Computer-Science-481-Spring-2019/computer-science-481-spring-2019-lab-2-gkapfham”. Unless you provide the instructor with documentation of the extenuating circumstances that you are facing, not accepting the assignment means that you will receive a failing grade for it. Instead of giving you access to “starter” code for this assignment, the purpose of this GitHub repository is to facilitate the assessment of your own mastery of the professional and technical skills in software innovation.

This assignment invites you to work in a entire-class team to create a suite of educational software tools for use by the students and faculty in the Department of Computer Science at Allegheny College. Your goal for this project is to implement a suite of useful, production-quality software tools, deployed as either web sites or command-line applications that can be used starting in the Fall 2019 semester. For this assignment, you will collaborate with the members of your class to implement these tools in the Python programming language, leveraging, for instance, either the Django web application framework or the Chalice framework for creating and deploying Lambda functions on the Amazon serverless computing cloud. Your development team should organize itself into sub-teams that will focus on implementing a total of five educational software tools. Then, each sub-team should pick a project, decide on a name for that project, create a GitHub repository in the GatorEducator GitHub organization, and start using the issue tracker to identify the key projects on which you will focus. Every team member is responsible for working together to handle these issues and any others that arise during the completion of this long-term software project.

Using this list as a starting point, the sub-teams should each pick a distinct software project; the entire class is tasked with implementing a total of five software tools as part of a cohesive suite.

1. **Course Survey**: A survey that students can complete to share, for instance, their concerns about course content and instruction. This survey should be customizable, thereby allowing an instructor to ask new questions of students on a regular basis. The survey should support a wide range of question types and be available through the terminal window or a web interface.
2. **Academic Advising**: A means by which students can effectively communicate with their academic adviser, supporting, for instance, the submission of a status update and a four-year course plan. It would also allow faculty to record that they have met with their advisees.
3. **Interactive Quiz**: A customizable quiz system that allows faculty to define questions, administer a quiz, and receive a copy of results in a fashion suitable for semi-automated grading.
4. **Student Petition**: A system that will allow students to upload a petition for a change in, for instance, their graduation requirements. Along with giving faculty a way to vote on the petition, this system should then automatically notify students of the result of the vote.
5. **Institutional Review Board Checklist**: A system that will allow students to upload a copy of the proposal for submission to the College’s institutional review board (IRB). This tool should also ask students to complete a checklist of points to consider about their proposal and the facilitate a faculty members submission of the document to the IRB’s current chair.

Collaborating with Your Software Engineering Team

Your team should use GitHub and its features (e.g., issue tracker, pull requests, commit log, and code review request) to complete all of the tasks referenced in the previous section. Aiming to manage risk and estimate the effort required for individual team members to complete this project, you should assign people to teams, roles, and tasks. While it is acceptable for you to have in-person discussions with your team members or to talk about the project through Slack, please remember that all important discussions and decisions must be documented through GitHub. Finally, as you are working with your team, you should carefully document your experiences and contributions so that you can share them through writing stored in the repository created by GitHub Classroom.

Since multiple approaches may support the effective completion of the required software, this assignment does not dictate team organization or communication strategies. The students in the course should instead work with each other, the team leaders, and the course instructor to identify team roles and strategies for effective organization and communication. With that said, you should plan to use either forks or branches of a GitHub repository to organize your work. Once a specific branch/fork contains the finished version of its associated deliverable, a team member should create a pull request for discussion and review. If the team leaders, the technical leaders, and the course instructor judge that the pull request has all of the expected characteristics, then it should be merged into the "master" branch of the appropriate repository. If the pull request is not accepted, then team member(s) should improve it until it meets every reviewer's expectations. Your team should continue to use this model, called "GitHub flow", to support the completion of all deliverables. Students with questions about the use of GitHub should first talk to a team leader.

Self Evaluation and Project Retrospective

Your GitHub repository for this assignment contains a Markdown file that you will use to document, evaluate, and reflect on your contributions to this project. Your evaluation of your own work should focus on your mastery of the technical and professional skills that are necessary to become a software innovator. You should thoughtfully reflect on your current areas of expertise and opportunities for improvement. As you work on this project, you must be proactive in finding ways to master skills such as using version control, documenting and refactoring source code, adding and testing new source code modules, and writing technical documentation. In addition to writing a publicly available blog post that documents and reflects on your experiences, your private reflection should analyze the challenges that you faced and the strategies that you adopted to overcome them.

Please remember that Travis CI is configured to use both "`mdl`" and "`proselint`" to check the Markdown files in your repository created by GitHub Classroom. If you saved the files correctly and your writing meets all of the requirements set by these linting tools, then you will see a green ✓ in the listing of commits in GitHub after awhile. If your submission does not meet the requirements, a red ✗ will appear instead. The instructor will reduce a student's grade for this assignment if the red ✗ appears on the last commit in GitHub immediately before the assignment's due date. Yet, if the green ✓ appears on the last commit in your GitHub repository, then you satisfied the basic linting checks for the Markdown file that contains your work assessment and reflection.

Suggested Schedule for the Software Project

The course instructor invites the students in this class to work together to devise a schedule by which they can complete the software product by the stated deadline. Overall, you will work on this assignment for six weeks. Here is a suggestion for a schedule to complete the suite of tools:

- **March 28**: As you start this project, please set a regular out-of-class meeting schedule and make arrangements to regularly report your progress to both your team leader and the course instructor. As you develop your ideas, make sure that you can carefully leverage GitHub's features to ensure that there is no duplicative work and that collaboration proceeds smoothly. Please see the instructor if you have questions about picking the project on which your sub-team will focus as you implement this suite of innovative educational software. You should develop a list of features for your program and explore what technologies you will use to support its implementation, focusing on frameworks supported by the Python language.

- **April 4**: Once you have finished, for instance, organizing your teams and raising issues in GitHub, further explore the Python language features and packages that you will need to implement your features. You should start to implement a prototype of your assigned features and rapidly add needed defect fixes. Please see the instructor and your team leader if you are facing severe challenges that may prevent you from finishing a prototype during this week.
- **April 11**: Finalize implementation of a prototype tool and give a demonstration to the instructors and students who will use your tool, requesting feedback that you incorporate into subsequent revisions. Your aim should be to finish all implementation and debugging tasks, ultimately leading to a full-featured demonstration that surfaces detailed feedback.
- **April 18**: Finish implementing and debugging all key features and complete all major defect fixes, ultimately leading to a full-featured demonstration and further feedback from additional people who use the tool. At this stage, you should have demonstrated your tool to multiple students and faculty at Allegheny, helping them to use the tool in their own work.
- **April 25**: While continuing to incrementally enhance your tool, release a completed version so that external individuals can use it and provide detailed feedback through the issue tracker. Complete your tool's documentation and reflection and publish a draft of your blog post so that it is available for peer review by the instructor and your colleagues in this course.
- **May 2**: Release a production quality tool suitable for use by other students and by all the Computer Science instructors at Allegheny College, ensuring that your system is correctly released through GitHub. Publish the final version of your technical blog post and submit your private reflection through your repository created by GitHub Classroom.

After project completion you will receive a letter grade for your cumulative work on this project. During the completion of this project you will also receive feedback and advice from the instructor.

Summary of the Required Deliverables

Students do not need to submit printed source code or technical writing for any assignment in this course. Instead, this assignment invites you to submit, using GitHub, the following deliverables. Unless you provide the instructor with documentation of the severe and extenuating circumstances that you are facing, no late work will be considered towards your grade for this software project. Using the GitHub repository for the GatorGrader project, the instructor will, as appropriate, share regular public feedback with all of the team members. All students will privately receive a letter grade for each of the following aspects of their contribution to this long-term software project.

1. Available for download from a GitHub repository hosted in the organization at `https://github.com/GatorEducator/`, a suite of educational software tools ready for production use.
2. Submitted through your repository created by GitHub Classroom, a self evaluation that documents what you learned, the challenges that you faced, and the contributions that you made to the GatorGrader project. Along with explaining how you interacted with non-team members (e.g., experts who provided feedback on the tool), this evaluation should also detail the ways in which you helped teachers and students learn how to use the GatorGrader tool.
3. Published on your web site, a publicly available blog post that overviews the suite of tools, documents the tasks that you completed, and reflects on your experiences during the enhancement of your tool. In addition to linking to your project's GitHub repository and your contributions, your blog post should use source code and command-line examples to give a detailed introduction to how your tool supports work in an educational setting at Allegheny.