

## Final Project Requirement Sheet 100 points

### Project Goals

- Summarize all of your course knowledge in one significant coding project.
- A significant portion of your final course grade (25%) is the project component. For your project, you will work as a team or some cases (individually).

### Project Details

There are two tracks for the final project, namely, track 1 and track 2. Of course, if a student would want to pursue a completely different track, this option is also available. Consult with and get the Professor's approval, if you like to develop an idea outside track 1 and 2.

It is required for all students to follow the honor code. Some important points from the class honor code are outlined below for your reference:

1. Students are not allowed to share code files and/or other implementation details outside their team. It is acceptable to have a healthy discussion with your peers. However, this discussion should be limited to sharing ideas only.
2. Submitting a copy of the other team's program(s) is strictly not allowed. Please note that all work done during this project will be an opportunity for team members to learn, practice, and master the materials taught in this course. By doing the work individually, and within their team, students maximize the learning objectives.

At any duration during and/or after the project, students are recommended to team up with the Professor and/or the Technical Leader(s) to clarify if there is any confusion related to the items in the project sheet and/or class materials.

### Project Track 1 - Data-driven Scalable Tool

- The course project in this track, must solve a real-life problem or develop a real-life application using Dataset Collection, MultiThreading, and Cloud Computing. One example is what we did with Web Scraping on dogtime website. In this track, it is important to focus on automating the data collection, develop one of more computational blocks on the collected dataset(s), publishing services through client service architecture (such as web server and/or web applications), display results through outputs and experimental results by using multithreading and/or cloud computing.
- A few examples of course projects that were done by the previous batch students:
  1. Scrape web data and do sentiment analysis on the hotel reviews from the Tripadvisor website.
  2. Scrape web data and do sentiment analysis on automotive consumer review sites such as Edmunds, KBB, and MSN autos to identify what customers are thinking about their cars and most importantly different features on their cars, for example, fuel economy, comfort on seats, etc...
  3. Identify fake news automatically by scraping news articles from different websites.

- There are many references to all of these problems, and I'm sure as you think of your project, you will find resources for them as well. If you're completely stumped in coming up with a project idea, you can certainly talk to me and we will set up a brainstorming session. Be creative and choose something interesting to you!
- The course project must have a significant implementation part where you will develop a program(s) using Cloud Computing and Dataset Collection techniques. You may write your code from scratch, or reuse and extend some existing code. Anything you use that is not yours must be documented. You may program in any programming language that you like.
- The course project must be extensive enough to qualify as a project (think of work for at least 3 to 4 one-week lab assignments), but not too extensive so that you cannot finish it in the remainder of the semester (two months).

## Project Track 2 - Platform-driven Scalable Tool

- The course project in this track is required to develop a platform to facilitate Cloud Computing. One example is what we did with AWS automation. In our labs and practicals, we developed a platform to execute different patterns such as triangle, square, and diamond patterns. Can we provide a platform so that users can easily use their cloud profile and your tool to execute their code in the Cloud? You may develop the project in this track using a combination of web applications, and web services. Tools such as CanvasJS may be used to integrate experimental results in the web application using Javascript and JQuery.
- A few examples of course projects that were done by the previous batch students:
  1. Convert the AWS automation code series, to a web application. This application is then easily used (through the web) to create, terminate, start, and stop instances using a web interface. In this project, the user provides their cloud profile details, and the entire cloud is considered a black box to them.
  2. Convert the Dropbox automation code series, to a web application. Besides, develop a new code series, to automate Google Drive through the web application. The tool then compares the two Cloud storage web-based platforms to show experimental results on which one is faster or slower for an upload, download, etc ...
  3. Develop a new set of patterns (2 to 3), to be executed in the Cloud, based on triangle, square, and diamond patterns. This also involves developing a proper use-case to orchestrate and correlate with the new set of patterns.
- There are many references to all of these problems, and I'm sure as you think of your project, you will find resources for them as well. If you're completely stumped in coming up with a project idea, you can certainly talk to me and we will set up a brainstorming session. Be creative and choose something interesting to you!
- The course project must have a significant implementation part where you will develop a program(s) using Cloud Computing, Web Services and/or Web Applications. You may write your code from scratch, or reuse and extend some existing code. Anything you use that is not yours must be documented. You may program in any programming language that you like.
- The course project must be extensive enough to qualify as a project (think of work for at least 3 to 4 one-week lab assignments), but not too extensive so that you cannot finish it in the remainder of the semester (two months).

## Timeline and Deliverables

The timeline and deliverable details are provided below for your reference:

1. Proposal – Start developing an idea for your final project. Write a 2-page technical report (single or double spaced) of what you propose to do in your final project and submit a PDF copy of your proposal through the GitHub link shared. I don't expect the proposal to be very detailed at this point. But, it should summarize what project you are going to pursue, what you want to do (the real problem you will tackle, how you plan to solve this real problem, and at least a couple of references to indicate that you have done some research about the problem. **Deadline:** November 16th, 2020, 2:00 PM.
2. Progress Report 1 – Start developing a 3-page technical report (single/double spaced) to document the progress done by the team. By this point, you should have made a good amount of progress towards implementing your project. Were there any unexpected challenges? Did you have to change your initial model/framework or the project skeleton code? You should have also finished your experimental analysis of your proposed methodology by now. Include everything you have done so far in your progress report, even if it is incomplete. No need to include the actual code (unless you want my help with it), just describe what progress you have made with it. Submit a PDF copy of your progress report using the GitHub link shared. **Deadline:** November 23rd, 2020, 2:00 PM.
3. Progress Report 2 – Start expanding the technical report developed in the previous step (progress report 1). That is make a copy of progress report 1, and start adding the progress 2 content to it. Note; There should be separate submissions for progress report 1 and report 2 respectively. The progress report 2 should be a 6-page technical report (single/double spaced) to document the progress done by the team. By this point, you should have made a tremendous amount of progress towards implementing your project. Please refer to the previous bullet for what needs to go into the progress report. **Deadline:** November 30th, 2020, 2:00 PM.
4. Presentation – All teams should record a 10 minutes video to virtually present their course project. It is expected that students use Slides during their presentation. The deadline for this part is December 7th at 2:00 PM EST. The link below may be used to do the recordings. OBS is another option to record videos.

<https://www.apowersoft.com/free-online-screen-recorder>

By the presentation session, you should have finished implementation, run the experimental, and done some analysis. In the presentation, you should describe the motivation, problem definition, challenges, approaches, and results and analysis. Use diagrams and a few bullet points rather than long sentences and equations. The goal of the presentation is to convey the important high-level ideas and give intuition rather than be a formal specification of everything you did. Design at least 6 to 10 slides, including a slide with the title of your project and your name. Also, it is required to show a short demo of your tool at the end of the presentation. Team members should contribute equally to the presentation. You may upload one video with an edited version of the different team members presentation or upload separate videos. The presentation slides and video should be uploaded to the following google driver folder.

[https://drive.google.com/drive/folders/1Vp91uKvjVeY5GX4\\_i8kqOYZ2OsFz0BmM?usp=sharing](https://drive.google.com/drive/folders/1Vp91uKvjVeY5GX4_i8kqOYZ2OsFz0BmM?usp=sharing)

5. Final Report – The final report should be clear and well written, which includes no typos or grammatical errors. The report should be written professionally and technically. The report should include the following:
  - The motivation for your project. Why is the problem you decided to solve important or useful?
  - Background of the proposed problem. What have others done for it already? Include references.
  - Detailed project overview, a summary of the proposed implementation, and the methodology used. Include pseudocode, diagrams, and examples if appropriate. If you are extending existing work, briefly describe previous work and include references to it.
  - Description of your results. Make graphs, tables, and anything else that can help me understand your results.

- Conclusion. Give a short overview of your project and its results. Describe what you learned, what were the biggest challenges, and the biggest rewards.
- For each deliverable, you need to submit a PDF with your report (or presentation slides). For your final report, you need to submit any supplementary material (code, data, a README file documenting what everything is, and how to run your program) to the git repository using the GitHub link.

## Grading Rubric

1. Proposal – 10 points
2. Progress report 1 – 10 points
3. Progress report 2 – 10 points
4. Presentation – 35 points
5. Final report and implementation – 35 points
6. Please make sure to include the honor code statement in all submission files.
7. If a student needs any clarification on their project credits, it is strongly recommended to talk to the Professor. The project credits may be changed if deemed appropriate.

