

Individual Project

Contents

Goals:.....	1
Deliverables	1
Brainstorming.....	1
Instructions.....	1

Goals:

1. Deepen your software design and development skills.
2. Individually build and work on a project that you feel proud to talk about with a recruiter.
3. Go deeper into a subject/topic that you are either curious or excited about.
4. Develop strategic skills as you plan the scope of your project and set realistic goals given the time available and course-specific requirements (which can be generalized to a company's culture of reporting results).
5. Develop your communication skills as you present your idea (pitch) and project.

Deliverables

The individual project splits into five deliverables (each with a different deadline to scaffold your project's development):

1. Elevator pitch
2. Project scope and software architecture
3. Software testing activity
4. Demo video
5. Final report.

Brainstorming

1. Identify your project's topic and purpose. Choose:
 - A topic that you would like to explore, and/or
 - A problem that you would like to provide a solution for and/or
 - A task that you would like your users to be able to accomplish.
2. Identify your audience – who is your target audience?
3. Needs – what need does your project seek to solve or address?
4. Outline – what does your project do? What is it for? What topics are related to it?
5. Skills – what skills do you plan on enhancing through working on this project?
6. Career – in what ways can this project benefit your career goals?

Instructions

- The use of Java is *recommended*, although optional.

You have two pathways to choose from:

1. **Not using code-completion tools or prior source code as a reference.** Identify a project topic to then design, develop, and test a computational artifact.
 - a. To help you brainstorm, I offer a few examples: implement and compare different [sorting algorithms](#), [Dijkstra's algorithm](#), [Conway's Game of Life](#), [Minesweeper](#), [Triominoes](#), [Tour of Hanoi](#), [Musical Chairs](#), [Josephus problem](#), [Langton's ant](#), build an application to illustrate [pseudorandom numbers in Java](#), play with [Steganography](#), or build a web framework (look for [inspiration here](#) and explore [Awesome Shiny Extensions](#), and [shinyjs](#)).
2. **Using code-completion tools or inspired by some existing source code.** Search for different approaches to solve a specific problem through coding; finally, build a strategy based on various references (make sure to cite all of them!), or explore some GenAI tool of choice.
 - a. E.g., [T-Rex Chrome Dino Game](#), [Conway's Game of Life](#) , [snake game](#), [Jeff Heaton's Cellular Automata Project](#), or [train a neural network on MNIST with Keras](#). And remember to revisit the Carrer Readiness 1, where I added a few examples that are explicitly empowered by GenAI tools.

If you need, here are Potential Data Repositories:

1. [UCI Machine Learning repository](#)
2. [The world FactBook](#)
3. [Kaggle datasets](#)
4. [GapMinder](#)
5. [Data gov](#)
6. [World happiness](#)
7. [World in data](#)
8. [Global Fishing Watch](#)