

Project Checkpoint 2: Student Implementation & New Results

This second checkpoint builds on your initial work from Checkpoint 1. The goal is to demonstrate significant progress in your core implementation, show that you are actively writing code, and present a **new** result that moves your project forward. This assignment will require you to update your project context, showcase student-written code, and analyze your findings since the first checkpoint.

Note: It is okay to change project direction or goals between Checkpoint 1 and Checkpoint 2, but you must clearly explain any such changes in your submission.

This assignment requires you to submit an **8-slide presentation** in Quarto markdown.

1 Content Requirements

Your submission will consist of three parts spread across eight slides.

Part 1: Updated Project Context (2 slides) This section provides an updated summary of your project and your progress.

- **Slide 1: Updated Problem Statement & Goal**
 - Concisely restate your project's core problem.

- Clearly state the primary goal or hypothesis. **Highlight any changes or refinements** to your goal since Checkpoint 1.

- **Slide 2: Updated Methodology & Progress**

- Briefly describe your proposed technical approach.
- **Explain your progress** since Checkpoint 1 and note any changes to your planned model, algorithm, or system design.

Part 2: Core Code Implementation (4 slides) This section is meant to demonstrate your coding progress. The code shown **must be written by you**. While you may get help from LLMs, this should not be code copied from an external repository. It should represent a core part of your project (e.g., model architecture, methodology, dataset code, demo code, or experiment setup).

- **Slide 3: Code Snippet 1**

- Show a particularly interesting or important part of your **student-written** code.
- **Strict limit of 20 code lines.**

- **Slide 4: Explanation of Snippet 1**

- Explain what the code on the previous slide does.
- Justify why this is a key component of your project.

- **Slide 5: Code Snippet 2**

- Show a second, different **student-written** code snippet.
- **Strict limit of 20 code lines.**

- **Slide 6: Explanation of Snippet 2**

- Explain the function and importance of the second code snippet.

Part 3: New Result & Analysis (2 slides) This section is for sharing a new result from running your code.

- **Slide 7: New Preliminary Result**

- Present a **new result** obtained since Checkpoint 1. This must be a step beyond your initial finding.
- This can be a comparative result (e.g., method vs. baseline), a new figure, a more complete demo, or a results table.

- **Slide 8: Result Analysis & Next Steps**

- Briefly explain what this new result means. How does it compare to your expectations, the baseline, or the original paper’s claims?
- Describe your immediate next steps for the project based on this result.

Note on “No Results”: This checkpoint requires a tangible result to be considered satisfactory. If you were unable to get a new result, Slide 7 must instead show the primary roadblock (e.g., error message, failed experiment). Slide 8 must then provide a highly detailed explanation of your debugging process, what you tried, and your plan. **Submissions without a new result cannot score higher than ‘Okay’ (2) on the ‘Results & Analysis’ criterion, regardless of the quality of the explanation.**

2 Rubric for Grading

Your checkpoint will be graded according to the four equally weighted criteria below.

Criterion	Excellent (5)	Good (4)	Satisfactory (3)	Okay (2)	Poor (1)
Updated Project Context	The summary is exceptionally clear. Updates and progress since CP1 are explicitly and concisely stated. The project’s goal and method are immediately understood.	The summary is clear and establishes the project’s context well. Updates since CP1 are mentioned.	The summary is present but may be slightly unclear. Updates since CP1 are vague or missing.	The summary is confusing, incomplete, or a simple copy/paste from CP1 with no updates.	The context is missing or nonsensical.

Criterion	Excellent (5)	Good (4)	Satisfactory (3)	Okay (2)	Poor (1)
Code Snippet 1 & Explanation	The snippet is highly relevant, clearly student-written , and shows progress on a core project component. The explanation is crisp and demonstrates a strong understanding of the code's role.	The snippet is relevant to the project and appears to be student-written . The explanation is clear and connects the code to the project's goals.	The snippet is somewhat relevant and plausibly student-written, but the explanation is superficial. Or, the code is trivial (e.g., library imports).	The snippet is irrelevant, confusing, or appears to be copied from an external source . The explanation is missing or poor.	No code is provided for this snippet, or the snippet is trivial and unexplained.
Code Snippet 2 & Explanation	The snippet is highly relevant, clearly student-written , and shows progress on a <i>different</i> , core project component. The explanation is crisp and demonstrates a strong understanding.	The snippet is relevant to the project and appears to be student-written . The explanation is clear and connects the code to the project's goals.	The snippet is somewhat relevant and plausibly student-written, but the explanation is superficial. Or, the code is trivial (e.g., library imports).	The snippet is irrelevant, confusing, or appears to be copied from an external source . The explanation is missing or poor.	No code is provided for this snippet, or the snippet is trivial and unexplained.

Criterion	Excellent (5)	Good (4)	Satisfactory (3)	Okay (2)	Poor (1)
Results & Analysis	A clear, new result is presented with a thoughtful, deep analysis of its implications. Next steps are concrete, logical, and build directly on the new result.	A new result is presented with a clear explanation and logical next steps. The analysis is good.	A new result is present , but the analysis is superficial or next steps are vague. (This is the <i>minimum</i> level for a submission with a new result).	The result is unclear or analysis is missing. OR, no new result is presented , but the explanation of failure is detailed and demonstrates significant, good-faith debugging.	No new result or explanation of failure is provided, or the explanation does not demonstrate a good-faith effort.