Intro to Computer System and Platform Technologies:

Step 1: Understanding the Learning Outcomes

Before starting, ensure you understand the course learning outcomes related to number systems and digital logic. Your project should demonstrate these concepts using the Micro:bit.

Step 2: Forming a Team

Partner with a classmate to form a two-person team. Discuss each other's strengths and how you can collaborate effectively.

Step 3: Brainstorming Project Ideas

Brainstorm ideas for a project that applies number systems or digital logic. For example, you could create:

- A binary counter that increments or decrements with button presses.
- A simple game that uses binary logic to determine winning conditions.
- An encoder/decoder that translates between different number systems.

Step 4: Planning Your Project

Once you have an idea, plan your project. Determine what components you'll need, what each team member will do, and set a timeline for your work.

Step 5: Building the Project

Use the resources available at MakeCode for micro:bit to program your Micro:bit. You may need to learn the platform's specifics and possibly some JavaScript or Python if you're not already familiar.

Step 6: Overcoming Challenges

You may face technical challenges, such as getting the Micro:bit to perform as expected or integrating different components. Use online forums, official documentation, and class resources to troubleshoot issues.

Step 7: Writing the Reflection

For your individual reflection, structure it as follows:

Introduction

- Briefly describe the project and its objectives.

Demonstrated Knowledge

- Discuss the concepts you applied that you were already familiar with, such as number systems or basic programming constructs.

New Learnings

- Reflect on any new skills or knowledge you acquired during the project. This could be learning how to code for hardware if your experience is mostly with software.

Challenges and Solutions

- Describe specific challenges you encountered and how you addressed them. This can include both technical hurdles and teamwork issues.

Areas for Improvement

- Critically assess your project and identify areas where you could improve, such as in planning, execution, or deeper understanding of the concepts.

Conclusion

- Summarize what the project achieved and how it has prepared you for future work.

Step 8: Code Submission

Make sure your code is well-commented and adheres to good coding practices. Submit it through the provided link, ensuring that only one team member does the submission.

Step 9: Submit Reflection

Submit your individual reflection to Turnitin, adhering to the page limit and any formatting guidelines provided by your instructor.

Remember, while working on this project, regular communication with your team member is key. Use version control systems like Git to manage code changes effectively, if permitted. And make sure your project not only works but also clearly demonstrates the learning outcomes related to number systems and digital logic.

Example:

↳ *2 cells hidden*