Team Information

Name: Chess++
Group Members:

- Ahmad Bilal Carini
    - Position: Version Control Management
- Zeferino Araiza-Flores
    - Position: Timeline Management
- Zenas Ortega
    - Position: Chess Logic Management

Communication Tools:

- Microsoft Teams
- School Email

Communication Rules:

- Respond 24 hours from message
- Ping everyone for important messages
- If no response then an email will be sent

Relevant Project Artifacts:
https://github.com/ZeferinoA/Chess_plusplus

Product Information

- **Abstract:**
    - This program can simulate a chess game for beginners and advanced players who want to take the game into their own hands. Beginners will be able to learn the basics of how each piece moves up to the complex openings and book moves. Advanced players will be able to improve the game that they have come to learn with pieces they want to add and establish their moves. These players can then connect to the server for sorta-ish online multiplayer.

- **Goal:**
    - The goal is to teach users how to play chess, create new pieces and new ways to play, and allow for local server 2 player connectivity or single player games to improve with oneself.

- **Current practice:**
    - Chess exists, but can a chess board teach new chess players how to play? How about with new pieces that can do dynamic and interesting

moves? What about a simple CLI chess interface that supports complex moves like Castling and En Passants? Plenty of CLI chess games exist, but do they have the wealth of features that truly makes them worth using over a physical chess board?

- **Novelty:**
  - The novelty of this program will be the CLI chess interface that supports complex moves, the addition and use of new pieces and new ways to play the game, and local connectivity.

Do not reinvent the wheel or reimplement something that already exists, unless your approach is different.

- **Effects:**
  - New players that want to learn the basics and the complexities of the game will have a place to learn. Players that want a unique twist to the game can create new pieces to use and go against. It will allow all players to have fun either by learning to play for the first time, good ol simple chess, or have crazy modes/new pieces implemented

- **Technical Approach:**
  - The program will run in a linux environment and may support other OS and platforms, it will be developed primarily in C++, leveraging the use of rendering libraries for the graphical elements. It will be developed and tested through a command line interface. Git and Github will be used for planning and version control. Pull requests will be used to manage merging.

- **Risks:**
  - Creating an intuitive interface for moving pieces could pose a problem. Chess notations exist but require outside knowledge beyond just chess knowledge. Overall, the interface poses a risk as it may end up being too esoteric to understand how the program works if the interface ends up being too complex or limiting.
  - Maintaining the novelty of the program despite the ubiquity of chess could pose a challenge. Making novel versions and modes or pieces could add flavor to the project but making sure the game remains balanced with those new pieces or versions will pose a challenge.

- When working through Github, making sure merges don't break or overwrite features that others have created will pose a risk, even though Github allows easy access to rollbacks.'
- Programming lesser known features of chess, such as En Passants, may pose an issue and leaving these features out (because of knowledge limitations or negligence) would remove a core (if obscure) component of the game.

Major Project Features:

- Major Feature 1: Establishing the logic of the command-line chess, beginning with the creation of the array for the board, game pieces structures, and the rules for the game
- Major Feature 2: Establishing the logic for the chess teacher for basic moves such as how the pieces move and how to check and checkmate
- Major Feature 3: Expanding upon the learning logic for complex move such as special openings and book moves such as castling
- Major Feature 4: Allowing users to create their own chess pieces with their own move set in order to expand the game

Stretch Features:

- Stretch goal 1: Making a local multiplayer environment so that the game is playable between players on the same network but different devices through the use of network sockets and hosting clients (as needed).
- Stretch goal 2: Making a CPU mode. Wherein the player plays against a 'smart' cpu capable of determining moves itself, either through the use of machine learning or pretrained open-source models.