1. Create a user interface that will allow users to play against four different modes, easy, medium, hard, and website versus.

|  |  |
| --- | --- |
| Test | We will have a submission protocol on the website, and we will have a received protocol for the MCU running. There will be four different mode buttons, and from the user selection the website will transmit the mode to the Microcontroller Unit using the MQTT protocol.  We will have a print () function to validate that the mode was successfully transmitted to the Microcontroller. |
| Results | To be determined |
| Conclusion | To be determined |

1. Detect token placement, and track each player’s turn and moves

|  |  |
| --- | --- |
| Test | This process will have the 7 Beam IR sensor connected to the 7 GPIO pins in the Microcontroller, which will have an External Interrupt function so we will be able to validate the user or computer moves. With the sensor it will be an active high (5V) so when the token blocks the beam it will have a low state (0V).  From the top sensor we can determine where the user dropped the token. We will also have a print () function to debug if the placement from each sensor being triggered has a validate MCU print. |
| Results | So far, we have tried using two sensors that the program is catch but more tests will be handle (on GitHub->program->logic->blink.py). Also have a prototype shield for multiple sensor that fits with our connect 4 board and it doesn’t interfere with other sensor side by side to each other (on GitHub->Design->Connect4.stl). More to be determined |
| Conclusion | More to be determined |

1. Dispense the disc into the dropper.

|  |  |
| --- | --- |
| Test | This will be a super easy process we will have a solenoid that will realise the disc when the computer move / versus process has made the move on the website / computer\_move (). After the MCU has received the coordinate to where disc must be dropping then this system will come on play.  The hardest part will be to have a good timing to be able to open the hole using the solenoid so only one of the disc drops. We assume it will be like a around 800ms delay but will have an exact time from calibrating at the spot. |
| Results | To be determined |
| Conclusion | To be determined |

1. Create a web browser interface to play game

|  |  |
| --- | --- |
| Test | This test will take the MCU and the Webserver having a seamless communication with each other. This is for the versus with a user potion from our project.  To be able to test this we would have debugged system in the console and the website. Using the MQTT protocol we will have a 7 button that will send the coordinate of the placement of the board to the MCU. |
| Results | To be determined |
| Conclusion | To be determined |

1. Create a web browser interface to play game

|  |  |
| --- | --- |
| Test | This test will take the MCU and the Webserver having a seamless communication with each other. This is for the versus with a user potion from our project.  To be able to test this we would have debugged system in the console and the website. Using the MQTT protocol we will have a 7 button that will send the coordinate of the placement of the board to the MCU. |
| Results | To be determined |
| Conclusion | To be determined |

1. Send a dropped token's data to display its placement on the browser / Versing another player on the web-browser

|  |  |
| --- | --- |
| Test | This test will take the MCU and the Webserver having a seamless communication with each other. This is for the versus with a user potion from our project.  To be able to test this we would have debugged system in the console and the website. Using the MQTT protocol we will have a 7 button that will send the coordinate of the placement of the board to the MCU.  Then similarly same process will be reverse so that the user who is playing on the board coordinate (Know by the sensors) will be send to the website. |
| Results | To be determined |
| Conclusion | To be determined |

1. Save the winning player's name in the database with their win count and win rate percentage.

|  |  |
| --- | --- |
| Test | This test will have used for the database side of our project having the win % / count/ rate. All being saved to the database on the website after the game is over. |
| Results | To be determined |
| Conclusion | To be determined |

1. Stepper Motor Gear system used to drop the disc on the board:

* The stepper motor will move the dropper and travel a distance between 1 to 34 inches along a guided rail system to position and drop a token in a slot on the board.
* The calibration process will move the dropper from one end of the board to the other, calibrating the motor. This entire process should be completed within one minute or less to ensure the dropper's positioning is accurate and consistent across the board's full range.
* The stepper motor will take between 2 to 5 seconds to travel and position the hopper for a turn, depending on the distance needed for each move. This ensures the proper functionality of the game.
* The process of dispensing and dropping the token onto the game board will take 15 seconds, allowing time for the stepper motor to grab a piece from the dispenser, position the hopper, release the token and have it land in the slot on the board.

|  |  |
| --- | --- |
| Test | With previous process working (no. 3), the disc will be dropped to the hopper that will have it drop to the position the user wants to place.   * Have the hopper going back and forward to the exact board length using the motor and gear system that is 3d printed to our need, * Have hopper go to specified location section of the board, * Have the solenoid open the hole of the hopper so it drops the disc to that location   This will have a overall big process as it is need for each step to be completed accordingly for the main gear system of the process to succeed.  Overall, timing for the stepper motor for each step:   * From start to end: Max. 2 sec, * At location dropping disc: less than 1 sec (will have a fixed buffer of 2 sec), * Reset to Start: Max 2 sec,   Total time for this process with buffer should be around 10 sec. As this timing would differ because there are multiple location to travel (7). |
| Results | To be determined |
| Conclusion | To be determined |