HYDROBOT

- Team Name: RobotGirlz
- Team Members: Mira Sivaprasad, Catie Robinson, Lubha Churiwala
- Github Repository URL: https://github.com/upenn-embedded/final-project-f25-f25-final_project_robotgirlz.git
- Github Pages Website URL: https://github.com/mira10-git/robotgirlz.github.io.git
- Description of hardware: ATmega328PB, MLX90632 FIR temperature sensor, LCD screen, Stemma speaker with audio amplifier, High sensitivity water level sensor, Audio FX Mini Sound Board, Linear Voltage regulator, Red and Blue LED lights

1. Abstract

This page is a brief of the working behind YOUR 'new hydration buddy' - HYROBOT. The bottle reminds the user to drink water if the water level in the bottle has not reduced every hour. It consists of an LCD which gives information such as water temperature and water level. The LCD has 5 different personality modes which can be switched by pressing the button. Along with the LCD, the bottle also has a speaker which gives the user auditory reminder for drinking water. The 5 different modes are: Happy, Sad, Angry, Sassy and Bored. If the user does not wish to have auditory interuptions, the bottle can be set to mode 5 (bored) which is the 'DO NOT DISTURB (DND)' mode. When the water temperature is below 30 °C, a blue LED lights up and when the water temperature is above 30°C, the red LED lights up.

2. Motivation

Everyone is so busy and occupied in their daily lives. Sometimes even the basic things such as drinking water or eating food on time are forgotten or ignored due to the strenuous schedule. For example, the day of a university student in UPenn includes - attending multiple classes + labs, completing back to back deadlines, sometimes working part-time, perhaps working on additional projects in GRASP labs, looking for summer jobs, trying to maintain a social life, taking up extra-curricular activities and so on. The list of milestones to complete are endless and time constraining. Due to the hustle of life, we forget to drink water -> get dehydrated -> loose energy to complete tasks -> fall sick -> create back logs and this cycle is hopelessly repetitive. Here's introducing **Hydrobot**, a bottle that reminds you to hydrate yourself and stay healthy!!!

Don't we all also have a preferred temperature to have our drinks in? Sometimes a little too cold or a little too hot than the normal temperature? For example, most university students fall sick, quite often catching a flu during certain months of the season. When you catch a cold, most of us prefer to drink warm water to ease our throat. Even as the weather gets more chilly in Philly, we start avoiding colder water. While getting hot beverages, how many of us have slightly burnt our tongue? I am sure quite a few of us. Here's why we are adding the feature of temperature readings to **Hydrobot**, so that everyone can enjoy their drinks in the temperature they'd like and avoid mishaps such as burning your toungue. Even when you don’t need a reminder **Hydrobot** is a friendly face monitoring the temperature of your water using infrared technology, so you know the temperature is just right!!!

HYDROBOT takes you through your own personalised hydration journey:)))

3. Goals

- Create an water bottle that would remind the user automatically to intake a significant amount of water every hour.
- The bottle gives the temperature read of the beverage inside, so that the consumers can have their preferred beverage in the exact temperature of their choice (hot/cold).
- Two LED lights to quicly know if the water is simple hot (red) or cold (blue).
- A speaker that talks in the voice of the user's personality choice and reminds them to hydrate themself.
- An LCD that tries to interact with the user by showing emojis, temperature value and how much water is left in the bottle.
- A switch that lets the user choose their preferred mood (happy, sad, angry, sassy or bored) and if they want the speaker to interact with them or not (so that the bottle does not create any inconvinience in public places)

4. Users

This water bottle can be used by users of all age and gender as drinking water is essential to existing and most people forget to hydrate themselves regularly. In addition to this, we have auditory aid from the speaker and visual aid from the LCD; this could further help differently abled people who are blind and deaf.

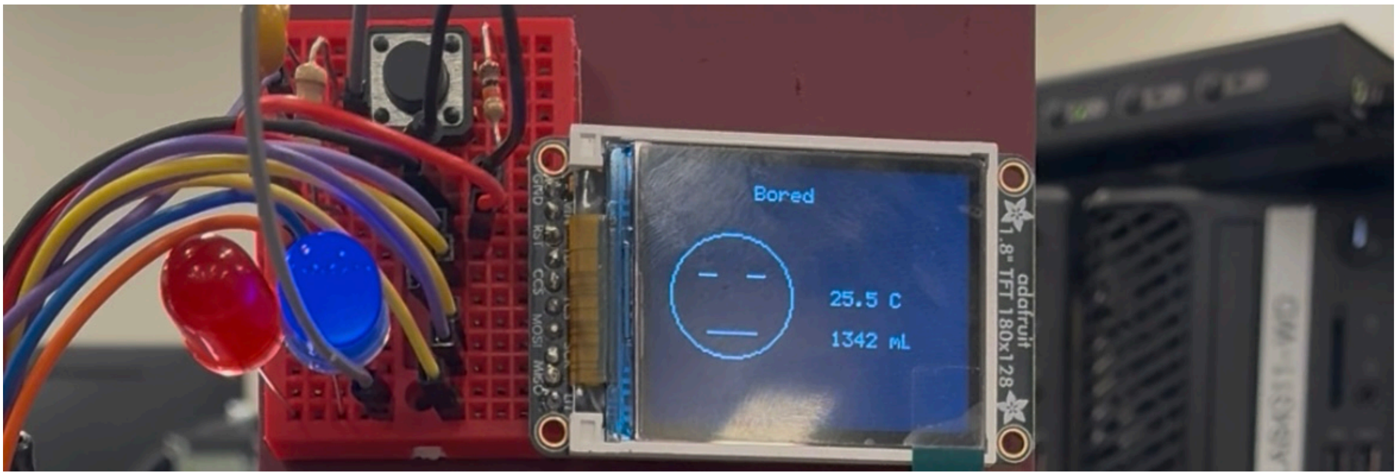
5. Video

https://drive.google.com/file/d/1J3wb_-LD5Ea_SrVyvrk9SyJ9oD1yoAxu/view?usp=share_link

6. Images

The speaaker fit to the case can be seen below: Speaker_case

The LCD screen mounted on the case can be seen here displaying 5 different modalities can be seen there:



```
<img width="1050" height="850" alt="LCD - happy" src="https://github.com/user-attachments/assets/64ad1bbd-d2f0-4358-a85c-94094f36baa2" />
```

The customised stacked up water level sensor can be seen below. The first image shows how it is placed inside the bottle and the second image gives a clear image on the stacking of 5 differet individual water level sensors to create a one long one: Water level inside bottle

Below the interior of the mechanical casing can be seen. It has securely placed the ATmega328PB, the speaker and the breadboard along with all wirings:

Below the exterior of the mechanical casing can be seen:

Below the extended battery case taped up to the bottle can be viewed:

7. System Block Diagram

The image of the system block diagram can be seen below: ![alt text](WATERBOTTLE BLOCK DIAGRAMfinal.drawio.png)

8. Methodology

9. Electrical

Below is the diagram of the electrical schematic:

10. Mechanical

Below are the 3D designs created on Fusion 360 that were used for 3D printing from the RPL lab: `<img width="689" height="956" alt="Front view" src="https://github.com/user-attachments/assets/c167dc75-82b5-4d11-816d-e4238ee88f15" />` `<img width="689" height="956" alt="Left view" src="https://github.com/user-attachments/assets/6e8650ec-fbae-42bf-b5b6-e18ff919724a" />` `<img width="689" height="956" alt="Back view" src="https://github.com/user-attachments/assets/30cced87-af1f-4381-8c44-a648baa546b7" />` `<img width="689" height="956" alt="Right view" src="https://github.com/user-attachments/assets/ec93c5a8-2908-4e06-9973-e2058d607f51" />`

Below is an image of the 3D printed case before any changes were made to the model: `<img width="289" height="405" alt="3D printed case" src="https://github.com/user-attachments/assets/c6b21fda-8816-4d67-b510-184cfabbe9e2" />`

Before the case was spray painted, some design changes were made by cutting certain sections off the case (due to size constraints) from the Garage Lab using a Uni-point Radial Arm Saw. Below is an image of both the case for electrical components and battery case spray painted from the Venture Lab: `<img width="190" height="302" alt="Screenshot 2025-12-07 at 3 57 39 PM" src="https://github.com/user-attachments/assets/4bad1dcd-f2b8-4b60-bad3-e82be13ff97d" />`

The below image shows the final 3D printed model mounted on top of the bottle comprising of all electrical components, breadboards and wirings inside of the print: `<img width="390" height="467" alt="3D printed model mounted on bottle" src="https://github.com/user-attachments/assets/b2a7cac0-f196-41be-9565-467dc0a5e37d" />`

11. Project Complexity

12. Challenges

The team encountered various obstacles that were not anticipated:

- The team went through 3 different set ups of failure to measure water level sensor before a customised version was finally working. The individual high sensitivity

water level sensors were glued together with epoxy. They kept coming off the more number of testing were done (as they were getting too drenched in water for a prolonged period of time).
- It was a challenge more than expected to integrate the temperature sensor with the rest of the electrical components. The water level sensor would interrupt with the temperature sensor and force it to give incorrect readings initially.
- The speaker modeule that was used initially was not compatible with the audio driver.
- The buck-boost converter was faulty and the linear regulator heated up quickly due to the larger step down voltage value.
- There were multiple versions of the 3D design (that did not fit the purpose of the design) before the team finalised on a casing with apt form and function.

13. Results

Our final result is that we created a bottle that gives regular reminders to drink water every hour depending on the water level inside the bottle, while also displaying the temperature of the water inside.

- Temperature sensor
  - The device consistently measures the temperature through an infrared object temperature sensor operating in its extended range, that communicates to the MCU through I2C.
  - Additionally, when the temperature of the bottle is more than 30 degree celcius, a red LED would light up indicating that the water is hot. Similarily, when the temperature of the bottle is less than or equal to 30 degree celcius, the blue LED would light up, signifying the water is colder in temperature.
- Water Level Sensor
  - The device also consistently measures the water level through resistive analog waterlevel sensors that are read by the MCU using ADC with its internal 1.1 V reference voltage.
- LCD screen
  - The raw data from each of the sensors was reinterpreted by the MCU to return values in degree celsius and mililiters respectively. These values are then displayed and periodically updated by an LCD screen that communicates to the MCU via SPI.

    - Alongside the displayed values is a color coded emoji associated with a particular personality option.
- Personality modes and speaker
    - We were able to create five different modes: Happy, Sad, Angry, Sassy and Bored.
    - Additionally, a speaker with in-built amplifier was used so that the reminder was not only visual but also auditory. These audios were customised in accordance to each personality modes.
    - The last mode (bored) is a 'DO NOT DISTURB' mode which means, the speaker will not be playing anything at that time. This mode can be taken advantage of when the user is in public place and do not prefer to be disturbed woth load noices. These personality modes can be changed at a click of a button.
- Since, there are both visual and auditory reminders, this bottle can be additionally helpful for both deaf and blind people.

14. Specification Results

14.1 SOFTWARE REQUIREMENT SPECIFICATIONS

Most of the software requirements were achieved. Values were received from the waterlevel sensor and temperature sensor. The audio files were plaing audible audio according to the display in the LCD which changed at every click of a button. There was an intial plan to diplay a string of color changing LEDs to show if the water was hot or cold. However, this was a tough software implementation as far as bare metal coding was concerned. It would have also added additional pressure on the memory use of ATmega328PB. So instead, a simplified logic of the same idea was deliberately use. Two individual blue and red LEDs were used that indicated temperature of the water. Only one of the software requirement was changed. From using a strip of LEDs that changes color to using two individual LEDs indicating red for hot water and blue for cold water. The reason this chnage was made was because it was much easier to implement code-wise and much efficient memory-wise.

ID	*Description*	**Validation Outcome**
SRS-01	The temperature sensor shall operate and report values every 1 second.	INSERT PIC OF THE LOGIC ANALYZER TAKING MEASUREMENTS IN TIME Confirmed. The IR temperature sensor starts and completes the 3 measurements necessary for an object temperature reading in < 100ms.
SRS-02	The water level sensor shall operate and report values no longer than 1 second after the opateration starts .	Confirmed. The analog water level sensors are continuously providing an input voltage to the ADC. Since the ATmega328PB's ADC is already enabled in single conversion mode after initialization mode, each ADC reading takes only takes 13 clock cycles, or 1.04ms from a 12.5kHz clock frequency (16Mhz with a 128 prescaler), making the enitre waterlevel reading procedure take <
SRS-03	Upon non-trivial temperature change (≥ ±0.5°C), the display shall update within 1 second.	Confirmed. The LCD screen updates at least twice every time the processor cycles though our main while loop taking TIME.
SRS-04	The audio processor shall store and play up to 4 pre-recorded 10-second audio files, with playback starting ≤1 second after trigger.	Confirmed. a signal comes trhough the speaker at TIME after GPIO trigger TAKE PIC
SRS-05	When the received temperature enters a new range (cold/normal/hot), the LED color shall change within 1 second.	Confirmed. When the temperature changes the LED color updates on the next cylces of our while loop (the loop may take longer)
SRS-06	The reminder timer shall be user-adjustable between 30–90 minutes through a simple switch or software parameter.	Confirmed. The timer period can be adjusted through the software parameter called ""PERIOD""

ID	*Description*	**Validation Outcome**
SRS-07	The system shall log daily total water intake (number of refills) and reset the count at midnight or on power-cycle.	Unconfirmed. The size of our water bottle made this functionality unreasonable.

14.2 HARDWARE REQUIREMENT SPECIFICATIONS

We were able to achieve most our hardware requirements, with some inoptimalities from our sensors. We wired the temperature sensor, customised water level sensor, speaker connected to the audio diver, LCD screen connected to 2 different colored LEDs and a button. For power, we used four 1.5V AA batteries that adds up to 6V. Them, used a linear voltage regulator to step down the voltage to 5V (which was the highest voltage required to power all components). This was then connected to ATmega328PB from where 5V input was delivered to the temperature sensor, LCD screen and speaker attached to the audio driver. The 3.3V input was used to operate the water level sensor. Our requirements did not change much at all. However, we did make slight changes in power regulation. Initially a single 9V battery was connected to a buck-boost convertor to step down voltage. However, the buck-boost convertor in the lab was faulty. This is when we switched to a linear voltage regulator. Since the step-down was from 9V to 5V, it heated up the voltage regulator quickly and it also seemed like a non-efficient set up. At this point, we switched to using four 1.5V batteries, which steps down voltage from 6V to 5V. This power setup was much more efficient and simpler than the earlier one.

ID	Description	Validation outcome
HRS-01	A ToF or ultrasonic sensor shall be used to measure the water level in the bottle. The sensor shall detect levels from 0 cm (empty) to 30 cm (full) with a resolution of less than or equal to 1 cm.	Unconfirmed. A cascade of resistive waterlevel sensors (linked here:Adafruit water level sensor.) were used and detected the water level from 0 mL to 1900 mL with an accuracy of +- 100mL, with higher quality reported values at higher volumes. This is suboptimal perfomance

ID	Description	Validation outcome
		due to the varying floating charge that exists on the water level sensors that are not in use (wet) at a given time.
HRS-02	A temperature sensor shall measure water temperature in the range of 0 °C - 60 °C with an accuracy of +- 1 °C. Validation: It should be able to measure the temperature of water without being in contact with it.	Unconfirmed. The temperature sensor accurately measures the temeprature of the water from 0°C - 40 °C. At hot temperatures the steam in the bottle clouds the sensor, blocking it from reading the water itself.
HRS-03	A speaker shall emit an audible alert of at least 60 dB.	Confirmed. The reminder is audible!
HRS-04	A bright display shall display water temperature and amount of water drank legibly under typical indoor lighting	Confirmed. The screen is bright and visible.
HRS-05	ATMega328PB should operate within its specified voltage range, that is 3.3 V-5V. It should be able to send display signals to the display screen via SPI.	Confirmed. The ATmega is able to operate this way via battery power
HRS-06	Power supply should be suffient to run the peripherals (temperature sensor, water level sensor, display screen, ATMega328PB) continuously without interruption for atleast 4 hours.	Confirmed. The system was left on for an afternoon and still operated normally throughout.
HRS-07	Water bottle should be made of metal and it should be able to withstand both hot water (up to 60°C) and cold water (till	Unconfirmed. The water bottle is plastic, but its still able to withstand the range of water temperature.

ID	Description	Validation outcome
	0°C) without deformation or leakage.	

Our final result is that we create a bottle that gives regular reminders to drink water every hour depending on the water level inside the bottle. We were able to create five different modes: Happy, Sad, Angry, Sassy and Bored. Each mode had a color coded emoji and information display on the LCD. The LCD diplayed information such as the temperature of the bottle and water level. When the temperature of the bottle is >35, the red LED would lit up indicating that the water is hot. Similarily, when the temperature of the bottle id <35, the blue LED would lit up, signifying the water is colder in temperature. Additionally, a speaker with in-built amplifier was used so that the reminder was not only visual but also auditory. These audios were customised in accordance to each personality modes. Since, there are both visual and auditory reminders, this bottle can be additionally helpful for both deaf and blind people. The last mode (bored) is a 'DO NOT DISTURB' mode which means, the speaker will not be playing anything at that time. This mode can be taken advantage of when the user is in public place and do not prefer to be disturbed woth load noices. These personality modes can be changed at a click of a button.

15. Timeline

Below is a Gantt Chart that was followed throughout the project. All milestones were achieved as per deadlines by the team. `<img width="2074" height="933" alt="Gantt Chart 1" src="https://github.com/user-attachments/assets/177f7e83-c346-4b68-843c-78a4aff31d4a" />` `<img width="2075" height="856" alt="Gantt Chart 2" src="https://github.com/user-attachments/assets/95cb1f24-ced2-4981-9888-e29c75c4f600" />` `<img width="2128" height="827" alt="Gantt Chart 4" src="https://github.com/user-attachments/assets/feaac036-15cf-4589-8c4f-f01e1b967844" />` `<img width="2136" height="951" alt="Gantt Chart 3" src="https://github.com/user-attachments/assets/f44e8db2-7284-43f6-ac57-f95e27c27168" />`

16. Future Improvements

There could be multiple improvements to this project:

- The mechanical casing can be an enclosed box with a latch system. Such a design would encompass all components without anything been shown outside. This was not implemented in the prototype so that all internal setup could be clearly seen during the demo.
- We could soldered down the device on protoboards, to prevent loose wiring.
- Obviously, a single water level sensor the length of the bottle could be used to give more accurate and consistent readings than the customised one that we have used in the prototype.
- The temperature readings works best for lower temperature than higher water temperature. Increasing the robustness of the code (taking and averaging entire measurement tables opposed to taking one reading at a time) and further waterproofing of the sensor set up itself would improve the performance of the temperature readings.
- The LED strips could be further implemented, as it does not only display two colors, but a range of red/blue could be shown as the temperature varied.
- A switch to power on the device could be implemented to avoid th edelicate process of of plugging and unplugging wires every time the device is turned on.

17. Reflection and Conclusions

- **What did you learn from it?** From this project, we got a wholistic view of the development process of an embedded system prototype. From planning out desired functions and building bill of materials, to testing individual components and final integration, this project expanded upon what we learned in ESE 5190's worksheets and labs, giving us one specific design where we had to implement everything we'd been learning about from scratch. Learning how to navigate multiple datasheets for desired information, write code for I2C communication for a device, find the necessary initializations for desired implementation, set up a linear regulatior, and how putting all these pieces together affected entire system operation, was all necessary for the succesful operation of our final design. We also gained experience with the arduous debugging process, isolating each component one by one. This is a very important skill for when we are working in the industry as most of the times, the components that work fine individually, stop working during integration and so, finding the root cause of the issue is a very complex problem. The project gave us hands on experience with bare metal coding, which one of our team members had had little to no experience with before. We learned individually about the intricacies of each main component: the temperature sensor, water level sensor, LCD screen and speaker and audio driver and how to integrate these together using ATmega328PB. We also got some experience in 3D designing and 3D printing.

- **What went well?** Many aspects of our project went well. Most notably the speaker with an in-built amplifier attached to the audio driver worked very well. There were no complications during any of the testing sessions and the audio were quite audible clearly differentiating between four different personality mode voices. Additionally, though it was the first part to be affected by integration bugs, the LCD screen also consistently displayed the information reported from the ATmega, giving us a clear place to look to see if there was an issue. If the code was hung up, we could see the LCD freeze, if a sensor wasn't working right we could see the garbage value on display, or if a button wasn't debouncing properly we could see the effect directly. This really helped with the debugging process.
- **What accomplishments are you proud of?** We were particularily proud of implementing the temperature sensor. Since it communicated through I2C,it required a lot of firmware implementation specific to the part itself. It took a lot of time to research, write code for, and test the necessary initializations, calibrations, and measurement start parameters, along with the firmware required to turn the raw data from the sensor into information that was interpretable. The temperature sensor took to most time to intergate with other parts, because the lack of an expected acknowledgement from the peripheral would hang up the entire application function. It was very rewarding when we figured out what was going on and got everything to work (code will get hung up waiting for an ACK if we did not have our I2C peripherals plugged in). Aside from the temperature sensor, finally putting everything together was very exciting even if it did take significant time and effort. The working of the entire project felt worth the effort.
- **What did you learn/gain from this experience?** In addition to the general practice applying what we learned in the course, we also learned how executing a project like this takes a lot of time, trial and error, testing, and adapting and overcoming the inevitable obstacles along the way. We learnt how we can approach the same problem in multiple ways. For example, we tried many different methods to measure the level of the water inside the bottle. This showed us that it is okay not to be stuck if the original plan did not work out, and that we can improvise when needed.
- **Did you have to change your approach?** Yes, due to many factors like time and feasibility, component compatability and information availability, we did have to

change our approach to a few functionalities multiple times.

  - We changed from using neopixel LED strips to individual LEDs to indicate water temperature. This was due to the complexity of the LED strips communication proccess that would have taken too much time that was better spent improving other parts of the device.

  - The speaker model was changed to one with an in-built amplifier to simplify the design instead of the two separate amplifiers required to boost the signal enough for the original 5W speaker.

  - Most notably, our approach to sensing the water level completely changed from a distance based sensor to a stack of 2-inch resistive waterlevel sensors.

    1. This change was initialy due to our first chosen part having intentionally incomplete documentation, due to the part manufacturers wanting the user to only implement the device using their API library to guarantee quality. This made it near impossible to write the necessary I2C code for the device.
    2. From here, our first back up plan was to use the ultrasonic sensor we had experience with from class. This however also did not work for our application, as the field of view of the ultrasonic sensor was too wide for the water bottle. It would sense the walls of the water botle before the water.
    3. After these, we looked into how typical resistive water level sensors operated. We had initially avoided many of these parts during our initial proposal because of their size, either being too big to fit in our initial bottle choice or too short for the length of the bottle. We looked into multiple options from here: a load sensor that communicated through I2C, two homemade water level sensor deisgns, and the two in-factory made water level sensor.
    4. From here we went with this final option since it had the most consistent behavior that we could implement in the remaining time we had. It took some more effort than anticipated, but we eventually got it working where it would return values reasonable for the set up.

- **What could have been done differently?** The final water level sensor implementation could have gone a lot differently. We could have continued researching other options or ordered different back up plan that would have avoid the floating charge errors we get now, but for the options and time we had our design is reasonable. A different water level set up would also avoid the issues of the waterlevel sensor blocking the temperature sensor when taking its readings.

- **Did you encounter obstacles that you didn’t anticipate?** Yes, we encountered obstacles that we did not anticipate:
    - It was a challenge to integrate the temperature sensor with the rest of the electrical components. The water level sensor would interrupt with the temperature sensor and force it give incorrect readings initially.
    - We went through 3 different set ups to measure water level sensor before we got a customised version working. The individual sensors were glued together with epoxy and they kept coming off with the different kind of testing we did (as the prolonged exposure to warm/hot water softened the glue).
    - The speaker module we were using initially was not compatible with the audio driver.
    - The buck-boost converter was faulty and the linear regulator heated up quickly due to the larger step down voltage value.
    - There were multiple versions of the 3D design (that did not fit the purpose of the design) before we finalised on a casing with apt form and function.
- **What could be a next step for this project?** There could be multiple improvements to this project:
    - The mechanical casing can be an enclosed box with a latch system. Such a design would encompass all components without anything been shown outside. This was not implemented in the prototype so that all internal setup could be clearly seen during the demo.
    - We could have prototyped boards. This way all the wiring set up would be customized.
    - There are commercial water level sensors that could be used. It would give more accurate and consistent readings than the customised one that we have used in the prototype.
    - The temperature readings works best for lower temperature than higher water temperature. So the sensor could be worked on more, in order to recieve better readings even at higher temperatures.
    - The LED strips could be further implemented, as it does not only display two colors, but a range of red/blue could be shown as the temperature varied.

19. Project Proposal Presentation

Below is the link used for the Project Proposal Presentation: https://docs.google.com/presentation/d/1BS_oT94HW1I447_tcnxJzq2WIQPKSj7Fw4KtnmdRAFw/edit?usp=sharing

Below is the link used for the Final Project Presentation: https://docs.google.com/presentation/d/1IQ7ptsauxRLAOfhBziD9jSM5K8ZaqktJYuk62H05T6E/edit?usp=share_link

20. References