



HatSight

“See the world through sound”

FSE 100 - Dr Tuzmen - Team 1

Aiden Reynolds
Cormac Kneup
Tarun Veeraraghavan
Ryder Granson

Problem Statement

- Visually impaired individuals can have difficulty with being to identify objects.
- Daily tasks such as finding an object, reading a label, or navigating the world safely can be a challenge.
- Existing solutions can be slow, overly complicated, or provide incomplete information.



- Our product focuses on helping visually impaired individuals quickly access their surroundings with the press of a button.

The Solution

- A hat which has a camera and button as the inputs and then the speaker and the vibrator as the outputs.
- When the button is clicked, the camera captures an image of what is in front of it
- Our python code with OpenAI client helps to convert the captured image to text
- And then another module to convert text to speech and outputs from a headphone speaker.
- This will be portable and functional.



Product Development Process

1. Ideation: ([See complete process here](#))

- This was the first step we did.
- We first watched multiple interviews of the blind and understood their specific problems.
- We then merged all the problems and started to develop a solution.
- We found the target audience which was “A visually impaired college student who wants an affordable device that will make their everyday life easier.”

Pain Points:

- Actual problem
- Current state
- Scope

<u>Actual Problem</u>	<u>Current state of problem</u>	<u>Why is it a problem?</u>
Cooking	Marifer has to make sure everything is in the same place and nothing gets moved to know where everything is.	If anything ends up getting moved, it creates difficulty knowing what each ingredient is.
AI image to text recognition is not specialized enough	Olivia thinks that books which have images when asked to AI for the description doesn't properly say what's in the image. She also feels that when trying to recognize different signs such as 'stop' signs AI is not capable to recognize it properly	This is a problem because the visually impaired are not able to enjoy visual books. Navigation is also tempered
Overhead objects hitting	A cane cannot detect branches or signs hanging from above, causing impacts to the head	This can hurt people, causing scratches, bruises, etc

Product Development Process

Potential products (around 5):

- A cane that can detect where the door is in the room. Using a video scanner and ai to identify where the door is, if it's open, closed. And using a beeper to alert where the door is facing in comparison to the person.
- An AI scanner that can quickly identify an object or thing using an image recognition scanner. This scanner will be attached to a cap.
- A bracelet trained to say whatever kitchen device is in front of the customer's wrist. It will be stylish and fun while also being very helpful while cooking.
- A device that makes a beeping noise on either side of the person, the beep gets faster the closer they are to hitting something on that side, like a cabinet or other. It uses ultrasonic sensors to detect how close the person is to hitting something. It is like a jacket with sensors along the side.
- A device that utilizes a PIR sensor and ultrasonic sensor in order to detect the nearest individual and the person's proximity to them. It can also have a function that can be toggled which notifies the visually impaired individual utilizing an active buzzer. One example of a use for this is to navigate crowded spaces like crowds or other general public spaces.

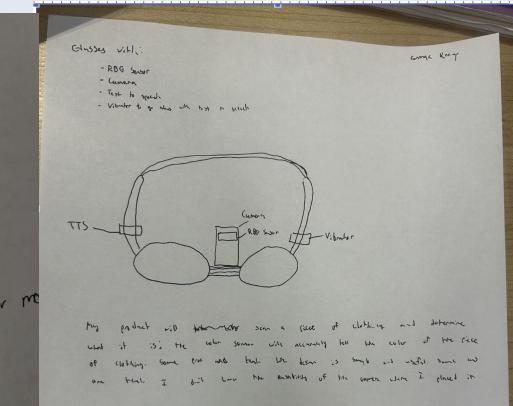
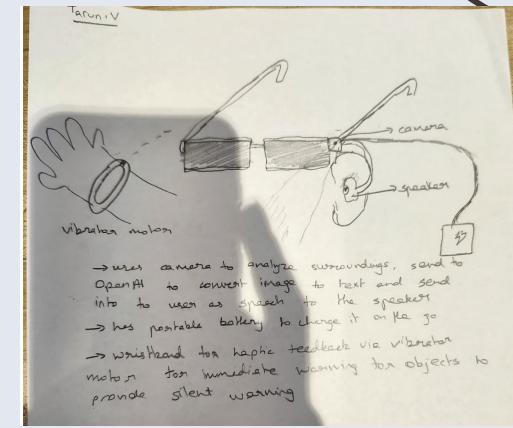
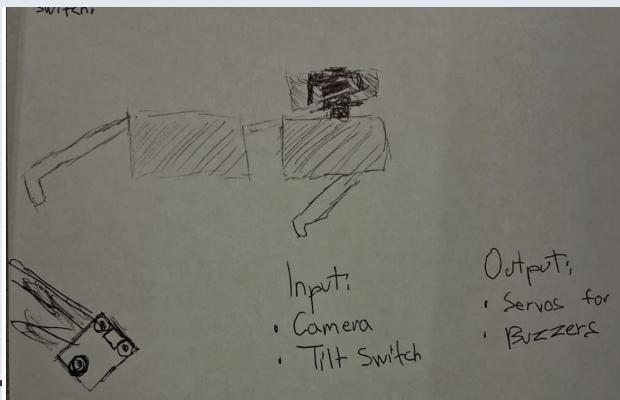
2. Product Research:

- We brainstormed different products in the market that solved this problem and made a list of functional and nonfunctional requirements.
- We initially made 5 product ideas that could solve the problems we had listed out in the “Ideation” step

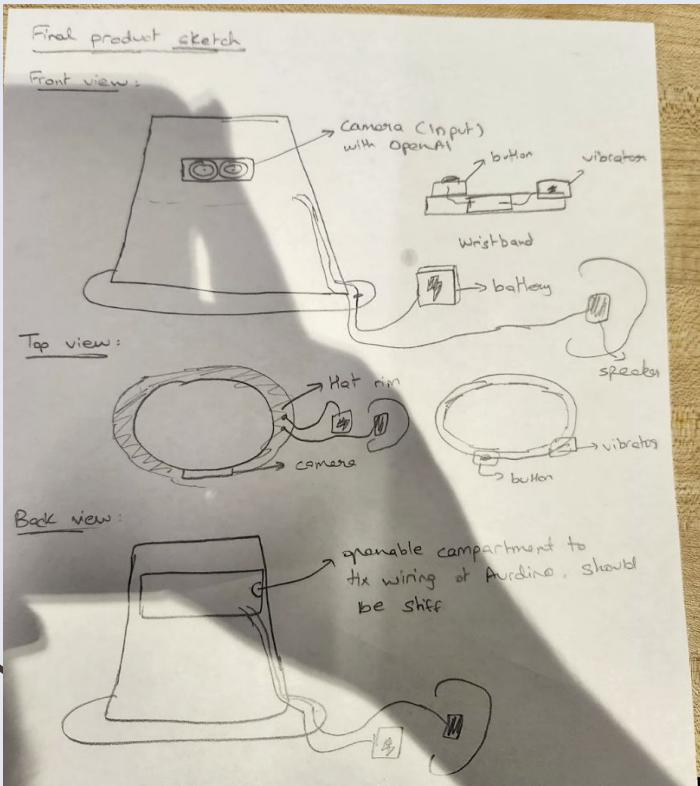
Product Development Process

3. First sketches: (See complete process here)

Once we had solid ideas, we created product sketches and tried to validate which idea was the best. You can see different sketches from our teammates here



Product Development Process



4. Initial product design ([See full development proposal here](#))

- After talking a bit about each other's idea of product, we were able to combine the best parts of each idea and come up with our first product design.
- We used Tarun's wristband control idea, and Aiden made the physical design and worked on our development proposal

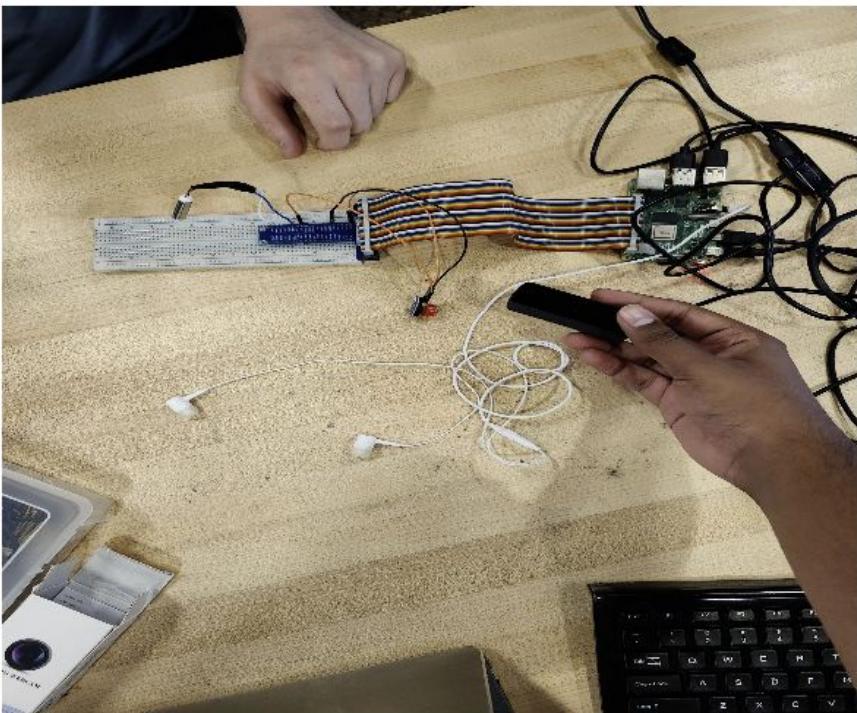
Product Development Process

5. Deciding our product items / shopping list (See here)

With the development proposal, we thought about what we need to buy and what existing items we can salvage. We decided that we needed to buy only a wristband

Group Number	1	
Material	Quantity	Total
Camera	1	1
Top hat	1	1
Button	1	1
Vibrator	1	1
<u>Braclet</u>	1	1
Aurdino	1	1
Breadboard	1	1
Wires	25	25
Cardboard sheet	1	1
Powerbank that can charge the	1	1
<u>Headphones</u>	1	1
Scissors	1	1
Tape	1	1

Product Development Process



6. Building 1 (Arduino wiring implementation)

- Cormac and Aiden worked on wiring of the Arduino, PI board and other connections.
- We were able to use the Openai connection for camera setup and buzzer_button assignment's connection for button to camera and button to vibrator connection.
- From pic, you can see the camera, button, speaker and vibrator

Product Development Process

7. Building 2 (Coding implementation) ([See full code here](#))

- Tarun and Aiden were responsible for the coding.
- They had to handle the core logic of button click , capturing image, sending image to OpenAI Client, converting it to text and output the text into speech through the headphones.
- We made sure to make it as modular and scalable as possible. Will explain code in detail later

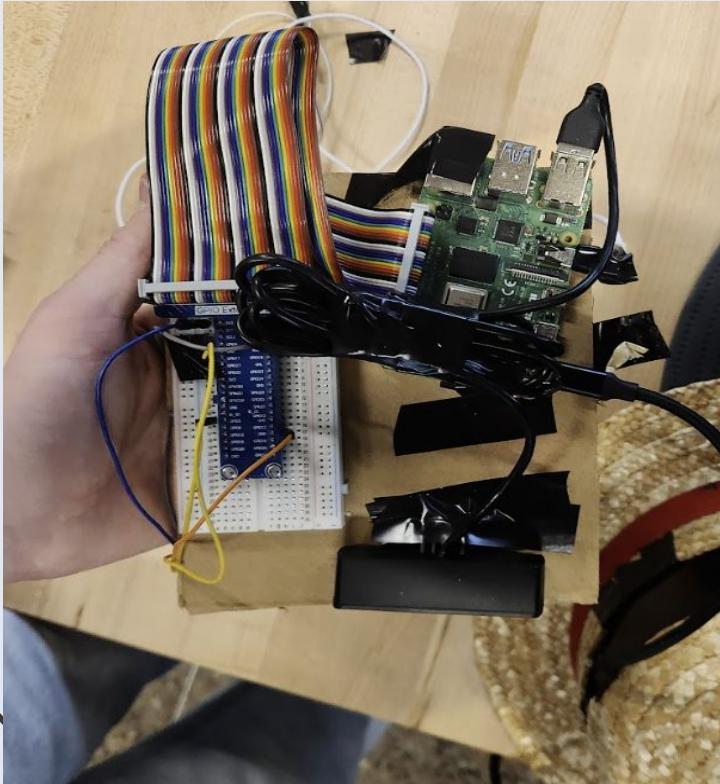
```
27 # OPENAI MODEL + FILE SETTINGS
28 # =====
29 client = OpenAI(api_key="MY_API_KEY") # API key replaced later for safety
30 MODEL = "gpt-5-nano-2025-08-07" # Vision-capable lightweight model
31 IMAGE_PATH = "captured_image.jpg" # Temporary saved camera image
32 RESOLUTION = "640x480" # Webcam capture resolution
33
34 # =====
35 # CAMERA CAPTURE FUNCTION
36 # =====
37 def capture_image(path: str):
38     """Capture an image using fswebcam and save it to disk."""
39     print(" Capturing image...")
40
41     # -r sets the resolution
42     # -S 2 skips first 2 frames for clarity
43     # --no-banner removes timestamp/banner overlay
44     subprocess.run([
45         "fswebcam", "-r", RESOLUTION, "-S", "2", "--no-banner", path
46     ], check=True)
47
48     print(" Image captured successfully!")
49
50 # =====
51 # CONVERT IMAGE TO BASE64 DATA URL
52 # =====
53 def to_data_url(path: str) -> str:
```

Product Development Process

Building 1 (Working implementation)

You can watch the video of the setup working here before setting it on the hat here: <https://www.youtube.com/shorts/ZFhytpx9R3o>

Product Development Process



8. Building 3 (Adding components to the hat)

- Once we had our basic Arduino working, we now had to attach the components to the hat to make it useable.
- We first cut out a small cardboard piece and taped all the components together on it.
- And then we made a hole in the hat to fit it in

Product Development Process

9. Building 4 (Adding components to the hat)

- And then we made a hole in the hat to make sure it was slidible.
- So, if any wiring issue was present, we could easily fix it and detach it back.
- We have the images of the back and side view



Product Development Process



10. Final working model

- After fixing some wiring issue, we were able to piece together and make a great product which is ready for testing.
- Image shows Cormac wearing the hat and the bracelet. You can see a working video here:
https://www.youtube.com/watch?v=k_EpGdcFMfQ

Product Development Process

11. UX testing (See review excel file here)

- Once we had our product working.
- We gave it to classmates and got their feedback on our idea, product design, wearability and usability. We properly managed their reviews in a Excel file and analyzed the changes we can make about them. These gave us new insights that we previously didn't have.

How accurate is the technology (output sensor)	Please provide your comments or recommendations	How easy is to wear and/or move and/or manipulate
7	6 Looks really good	6
7	7 Seems very intuitive and is put together very well.	6
5	7 Too quiet, the battery could've been attached to the earbuds	2
5	7 The product was a bit quiet. It would be nice if the earbuds were louder	2
7	6 The product is really well done	6
6	3 Camera would say "cannot detect" or give an error	5
7	4 I tried the product(hat) and i was not getting clear audio from the earbuds	5
6	6 The sensors are well placed	7
5	6 The technology was finicky and difficult to use.	6
6	6 None	4
7	7 All of the technology used as an adequate purpose	6
6	6 The strap that goes around the users wrist is a very good idea	6
6	6 Good choice in using the camera; I wish the earbuds had a built-in camera	4

Product Development Process

Its easy to wear but the hat balancing is a bit of an issue and adjusting the head everytime to capture the image in front is like a problem

the product has good wire management , and is easy to put on

There were a lot of loose wires

It could use a better strap but its still good

Wires extend long which could make them a little hard to wear, but other than that it is great.

- The images shows some of the feedback we got, if not readable. This is what it says:
- “Its easy to wear but the hat balancing is a bit of an issue and adjusting the head everytime to capture the image in front is like a problem
- the product has good wire management , and is easy to put on
- There were a lot of loose wires

12. Fixing user issues to make our product more user friendly

We got some feedback on existing issues such as “the hat not being stable or how the wiring was loose”. We also got some new ideas to work on such as “make the camera movable or make the headset bluetooth compatible”. We put some of them in action which we’ll discuss later.

Code Overview ([Full Code here](#))

A quick breakdown of our code to help you understand our product better

1. Image Capture

We got this function from the OCR_OpenAI project. It helps in capturing the image when the button is pressed

```
resp = client.responses.create(  
    model=MODEL,  
    reasoning={"effort": "low"},  
    max_output_tokens=256,  
    input=[{  
        "role": "user",  
        "content": [  
            {"type": "input_text", "text": prompt},      # Instruction text  
            {"type": "input_image", "image_url": data_url} # Encoded image  
        ]  
    }  
)
```

```
# =====  
# CAMERA CAPTURE FUNCTION  
# =====  
  
def capture_image(path: str):  
    """Capture an image using fswebcam and save it to disk."""  
    print(" Capturing image...")  
  
    # -r sets the resolution  
    # -S 2 skips first 2 frames for clarity  
    # --no-banner removes timestamp/banner overlay  
    subprocess.run([  
        "fswebcam", "-r", RESOLUTION, "-S", "2", "--no-banner", path  
    ], check=True)  
  
    print(" Image captured successfully!")
```

2. Vision to text

We got this function from the OCR_OpenAI project. It sends the captured image to OpenAI and gets the resulted Text

Code Overview ([Full Code here](#))

3. TTS (Text to Speech)

We used espeak python module for this purpose. It takes in the result from OpenAI and then creates a subprocess to play the audio through the configured headphone jack output

```
def speak(text: str):
    """Convert text to speech and play it via headphone jack."""
    safe_text = text.replace("'", '').replace('"', '') # Avoid breaking shell command

    try:
        # espeak generates audio → piped to aplay → played out loud
        subprocess.Popen(
            f'espeak "{safe_text}" --stdout | aplay -D plughw:2,0',
            shell=True,
            stderr=subprocess.DEVNULL
        )
        print(f" Speaking: {safe_text}")
    except Exception as e:
        print(" Speech failed:", e)
```

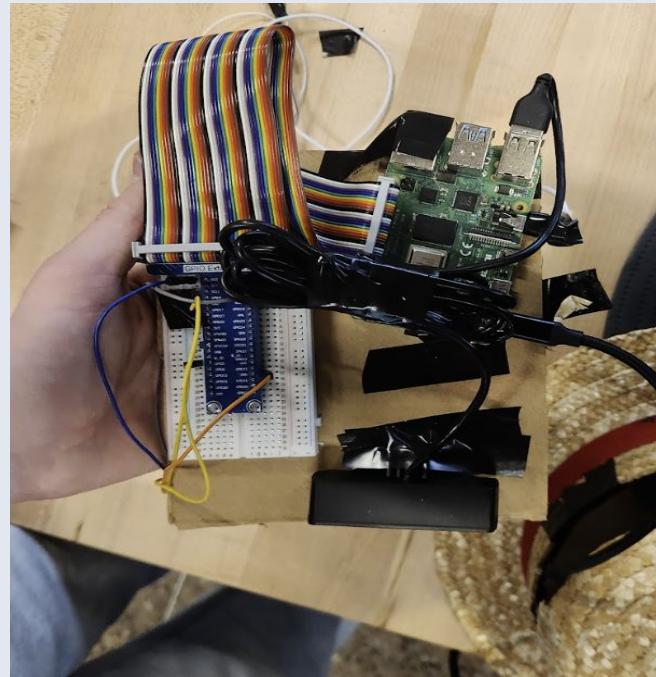
```
def vibrate(duration=0.3):
    """Activate the vibration motor for haptic feedback."""
    GPIO.output(VibPin, GPIO.HIGH)
    sleep(duration)
    GPIO.output(VibPin, GPIO.LOW)
```

4. Haptic feedback

We got this from the buzzer button python code. We modified this a bit so the vibrator vibrates when the button is clicked

Summary of feedback / Changes we made

- Lengthened the wires and taped them together to tighten them
- Changed the button and vibrator used



Summary of feedback / Changes we made



Widened the cutouts in the hat to make it easier to adjust the hardware. If a wiring issue occurs we can easily take off the electric component from the hat, fix it and slide it back in

- Adding an adjustable strap to the hat in order to secure it. The hat is more stable , firm and easy to wear



What we would do next time

- Make the hat more compatible with other head shapes.
- Choose a different audio output that makes the text-to-speech slower and louder.
- Work on the accuracy of the camera to make sure there are no errors.
- Create separate compartments to store the wires or tape them into clusters.
- Change the wristband to make it more compatible with the project design.



Thanks!