CSULB – College of Engineering
Computer Engineering
Senior Design Project
CECS-490A

Group 10 – Seat Taken

Team Members:

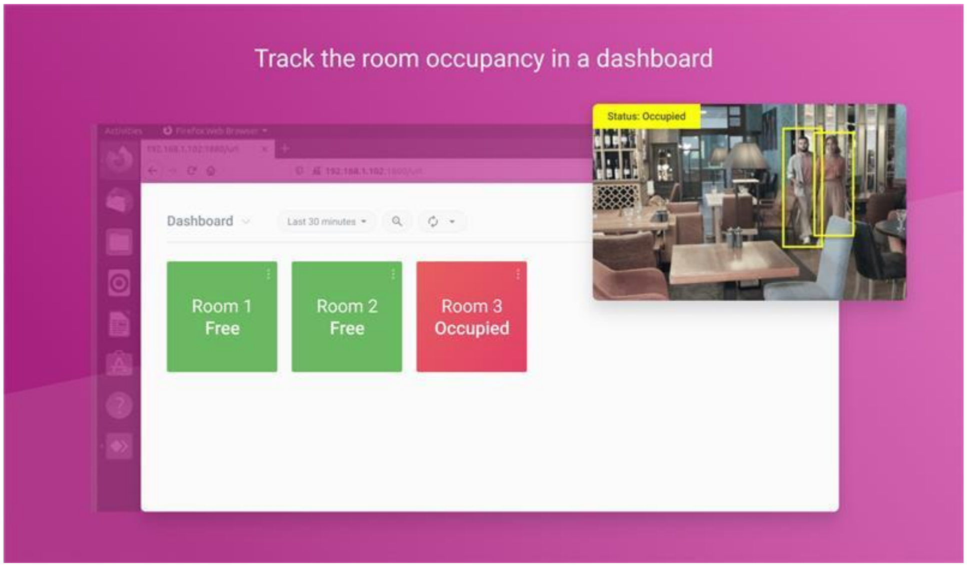
-Daniel Lee
-Andrew Jordan Gaspar
-Justin Miller
-Alexander Vallardez

Professor - Dan Cregg

Contents

Executive Summary:

Group 10's project, Seat Taken, seeks to create a system that tracks the occupancy of chairs in any given room. We will use artificial intelligence to recognize when a chair is being occupied by a person, being able to discern a human occupant against an inanimate object, such as a backpack. Then, using a website application, a user will be able to see a visual representation of the room layout with chairs and other identifying landmarks in the room as well as the occupancy of the chairs in the room. The planned system will be compatible with a wide variety of chair types and layouts, so it can be scaled to be usable in any situation, ranging from a quiet study section of a library, or even be augmented for use in high population areas.



Original Prototype:

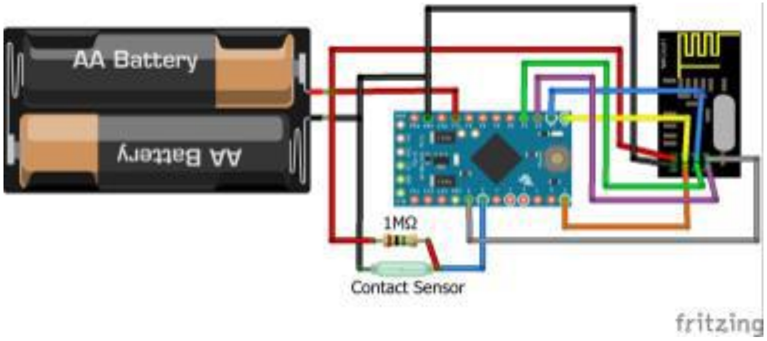
Our original ideas were to create a system of chairs that utilizes contact sensors to determine a seat's occupancy by a person. Each sensor would observe the weight of the object in the char and the duration of its stay to determine whether the seat is currently being occupied, or if another object is in its space. All the chairs would be connected to a central network that tracks the occupancy of each chair and relays the information to an application that provides a visual diagram of each seat's location and the state of its availability. We would have had a sensor in each chair as well as an MCU to operate and give it its functionality. We began to realize this would require a lot of parts, and these parts do not have a long shelf life when it comes to people sitting in chairs.

While this idea would work, it would not be sustainable on a large-scale use or constant repeated use. The parts would wear out/break and having to constantly repair or worry about part theft is something that we realized would be too much of a hassle and not feasible. This idea would be great for home use or a hobby project but not a professional product for commercial use. We want to design something that can be used on a large scale and is easy to adapt to future technologies.

This opened our eyes to design functionality and future purpose. We want to design something that can be upgraded remotely and easily. Having a camera and live feed system connected via Wi-Fi will allow us to easily manage our system and update it. Using multiple sensors and MCU's is asking for more problems than we could handle and would require us to be on site to fix/repair anything as well as creating a money pit for broken parts/stolen parts/newer parts.

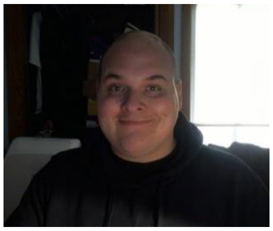
Previous Prototype Design:

- Battery life limits sensors capabilities, constantly having to replace batteries.
- Switch may break, flimsy, would need expensive sensors.
- ESP32 MCU would not be able to support more than one chair unless you wanted a bunch of wires running everywhere.
- Easy to steal if someone chooses to do so.
- Wi-Fi signal strength would not be strong enough.

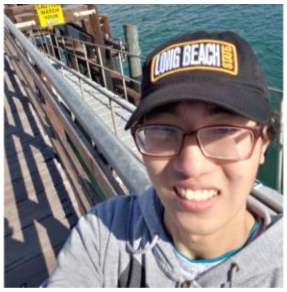


Team members:

Andrew Jordan Gaspar: A computer engineering major at CSULB, Andrew is knowledgeable in a multitude of coding languages and engineering technologies. With internship experience in the semiconductor industry, he can provide the ability to program microcontrollers and create algorithms for the team.



Justin Miller: CSULB Computer Engineer, hard worker with decent coding skills and good team management. Good task manager and will go above and beyond to make sure the team and everyone is on track and doing ok with the work and helping as much possible with coding/assembly/time management.



Daniel Lee: CSULB Computer Engineering student. Good at coding and time management. Is extremely interested in Object Detection. Will help us with the coding of our Object detecting algorithm along with training our AI to be best suited for detecting people in chairs.

Alexander Vallardez: CSULB Computer Engineering student. Heavy background in computers and anything to do with computers in general. Up to date with advanced topics and innovative technology.

Team Roles and Contributions:

Daniel Lee:

The general responsibilities we agreed upon during the declaration of our project for Daniel were getting the hardware to work. Daniel researched sensors for our original prototype. When the team decided to change prototypes, Daniel researched which cameras were best suited for our project and why. Daniel also has the responsibility to understand the general way we would set up our embedded system. Daniel was assigned to what math would be required for our project, security of our system, modeling ideas to structures out code, the coding program, program/algorithms/coding language to use, creating a basic schematic, and writing the first progress report.

After switching from a hardware contact sensor, Daniel's assignments were to research object detection and potential libraries to use and train our own object detecting AI. Daniel also created the Gantt Progress vs Actual timeline. Daniel primarily researched how computer vision works software wise along with finding libraries which may prove useful towards the goal of our project. Learning how the libraries worked with the other libraries was also a major role.

Andrew Jordan Gaspar:

Andrew's responsibilities for this project are implementing the website or application integration for our system, ensuring that the data collected by the Jetson Nano will have a seamless transition to our MySQL database, and then to our Django-based website. Currently, prototypes of both the database and the web framework have been created. The current prototype of the MySQL database is remarkably simple, as it is just to hold the value of each monitored chair's identification, and its occupation status. Each chair's identification is an integer that is held to each chair individually, and the status is a Boolean value that reads "False" for an empty chair and "True" for a filled one. As of now, there is no connection between the Jetson Nano and the database, but once the algorithm is set for recognizing occupied chairs, the Jetson will be able to update the database in real-time according to the data that the camera witnesses.

The website framework is currently sparse, but ready to be enhanced to the necessary level for the project. Django provides a very ample set of tools that will pair well with the MySQL database. The website already has the ability for holding a table of values much like a database, so once the two are paired, there will be no problem with the website updating its table to match the one in MySQL. As well, Django offers the ability to edit the page with images, which will allow the seating diagram to be implemented, and enable the ability to show the status of chairs in a way that is more user-friendly way.

Justin Miller:

Contributions to this project are adding tasks, making sure we all stay on task and are prepared for each week when we have our progress reports and must update the professor on our progress. Also, researching which MCU's/FPGA's/Computers will work for our project, which sensors will or will not work and why. Design our AI object detection recognition project around research. Find out what will work on the circuit level as well as contribute to the overall block diagram. Our new plan is to use AI Facial Recognition/Object Detection to monitor empty chairs versus chairs with people/faces. Justin is also researching our coding methods, specifically with Python in a Linux environment.

Justin is also contributing to the budget for materials and supplies, as well as offering my home and residence for testing purposes and group meetings if needed. He is also purchasing a second set of parts for testing and designing purposes, so team members have easy access to a Jetson device.

Alexander Vallardez:

Alexander's main contribution to the group is to work on the hardware. He must find the best hardware we can use for the project. Alexander will be gathering the materials we need for the project and researching what works best together. Other than that, most of his responsibilities are for project reports and task management. He will also be coordinating with Justin on the cost and getting parts that we need for the project. They will also be meeting up a lot to contribute to the building of the project. Currently Alexander is making a prototype 2 for us to work on. He has been implementing premade code into this prototype to make sure it is functioning as it should. This prototype is used as a backup and can be ready at hand just in case our original prototype has a defect. He will also be working on alternative power for the Jetson Nano. As we get closer to our real-world demos we want as many options as we can have when it comes to power. Lastly, He will be working on the schematics and drawings for the hardware. He is also in charge of all the port connects and all the externals that we will be using for this project.

Detailed Description of Our Project:

Objectives:

Our objective for this project is to provide a cost-efficient way of tracking the occupancy of seats in each room that is capable of being scaled up. Throughout our design process we originally planned to test a variety of MCU's and sensors, specifically any sensors that provided switch like capabilities like ultrasonic sensors, contact sensors, IR sensors, and magnetic field sensors. After thorough discussion and planning we decided having fewer moving parts would be better for our project. By using a Jetson Nano with AI technology (Object Detection) and live stream footage of any room from a camera or camera system, we can then cut out unnecessary costs such as theft or worn out/broken sensors. Over time these parts wear out and are more prone to break or not work. Cameras do not move or break as easily and people will not have easy access to our system. Attaching parts to a chair and leaving them in any public setting is asking to get things stolen/broken. By utilizing a neural network and artificial intelligence with the Jetson Nano, our team will be able to contrast between an empty and occupied chair by using our Jetson Nano with Python. Utilizing machine learning algorithms to help with object detection.

Features:

- Single-system installation. The occupancy-tracking system comes as a single, assembled unit that can be attached to a ceiling without needing external accessories across the tracking area
- Ability to track multiple chairs within the camera's range of vision
- Discrimination between humans and objects occupying seats. The artificial intelligence will be trained to recognize when a person is sitting as opposed to something like a backpack.
- Trainable AI that can be used to recognize different chairs
- Visual diagram that shows the real-time occupancy of every seat within the layout

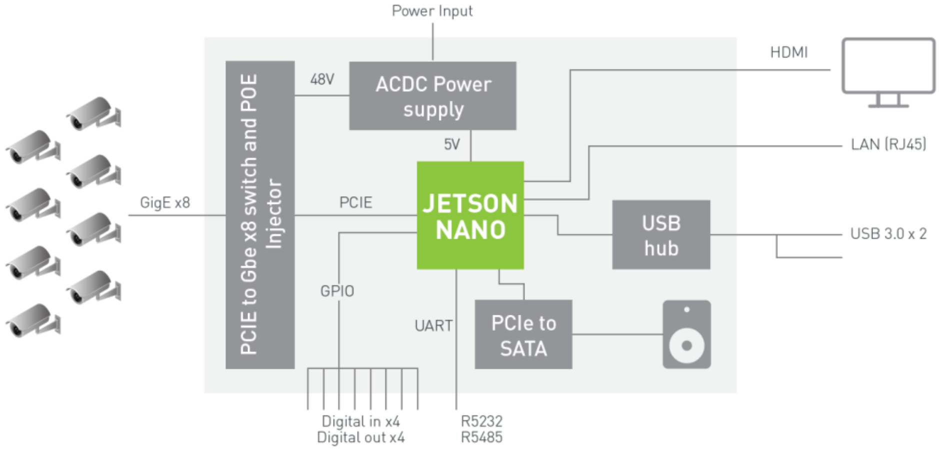
Benefits:

- Powerful data processing for AI object detection.
- Low-power solution when compared to sensors for each individual chair
- Easy scalability as the artificial intelligence can be trained and implemented in another Jetson Nano depending on the size of the area being tracked
- Unintrusive to the comfort of the individual occupying the seat. No sensors or microcontrollers will be attached to their seat
- While more expensive than our original idea, this allows it to be implemented on a large scale, easily reducing costs by only having one unit (monitoring system) per room etc.

Hardware Design Overview:

Our Jetson Nano (Outside of Case):

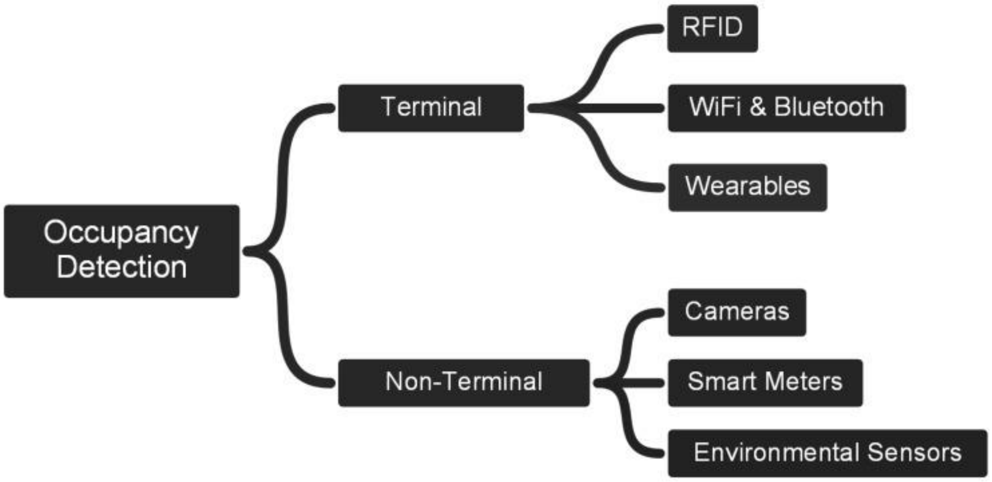
Overview Block Diagram of Jetson Nano AI System:



We will use a similar hardware schematic like the picture above. Our version will have the NVIDIA Jetson Nano 2GB computer with USB Wi-Fi connection hooked up to a Raspberry Pi 2 CSI camera (ribbon camera). It will be storing all its data and operating system on a micro-SD card formatted with NVIDIA's Jet-Pack Linux system. We will then be able to view and send data via VNC viewer, which allows us to remotely view and access our system. This feature will allow easy updates and maintenance. You could potentially connect this system to multiple cameras as seen above, while also being able to check the feed remotely via HDMI monitor or VNC viewer.

The Jetson Nano also comes with plenty of GPIO pins for more equipment and extra NVIDIA TX CUDA modules (more ram and data processing power, think of it as an extra memory stick).

Occupancy Detection Routes:



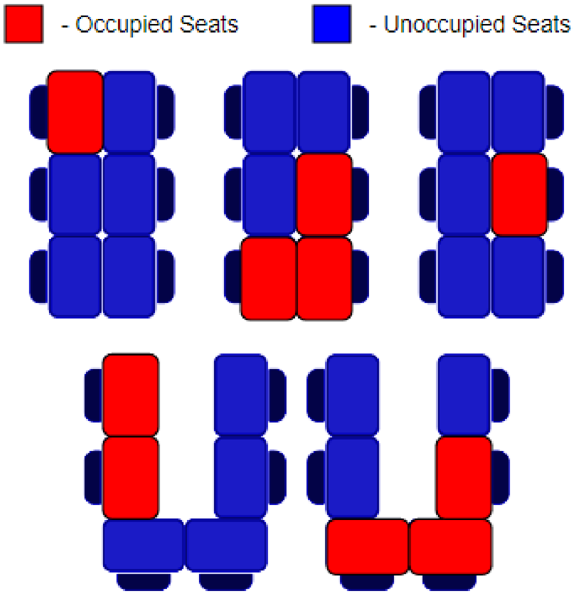
Object Detection Sample:

Web Application Design Overview:

As previously stated, our project aims to have a seating diagram that is openly available for public use. This will work by utilizing a MySQL database in tandem with a website running Django framework. The Jetson Nano will communicate directly with the database and update the tables for each chair's current status. Each chair will have its own ID within the table, which will allow us to monitor each chair individually without them overwriting each other's values. When the camera attached to the Jetson Nano witnesses an individual sitting in a chair, the database's value for the occupancy of the specific chair will be changed from 0 to 1, indicating that the seat is now taken.

From here, the website will be updated in tandem with the database's table. Django offers the ability for websites to have their own tables, much like a database, so we will be able to synchronize the website's tables with that of the MySQL database. As well, the website holds the image of a seating diagram that corresponds to the chairs that are tracked by the Jetson Nano. When the website detects an update in the occupancy of a chair, the image will be altered to display that information by changing the color of the seat, with red being occupied, and blue being unoccupied

Example of the Web Application's Seating Diagram



Verification:

Testing Procedures:

We have 2 things to test, our camera and our Jetson Nano's performance. It is important to know the performance of our camera and Jetson Nano.

The camera we chose will have to have good enough specifications so we can have smooth video feed, which will give us accurate results. We can test the performance of our camera by the picture quality and the bit rate of the video feed data as well as FPS.

The performance of our Jetson Nano will determine the pull rate of the AI output along with the number of objects to track on screen. To test our Jetson Nano' performance, we can use premade AI's that track people and objects and we can estimate the performance of our AI Seat Taken Project.

With the hardware finished we will need to make sure our output from th e Jetson Nano will be as accurate as possible. We can make sure the is as correct as it gets by taking as much training data of the angle where the camera will be sitting. As the camera will be on the ceiling or mounted in the wall, the chairs from the top down will be necessary for a more accurate output.

Tolerance Analysis:

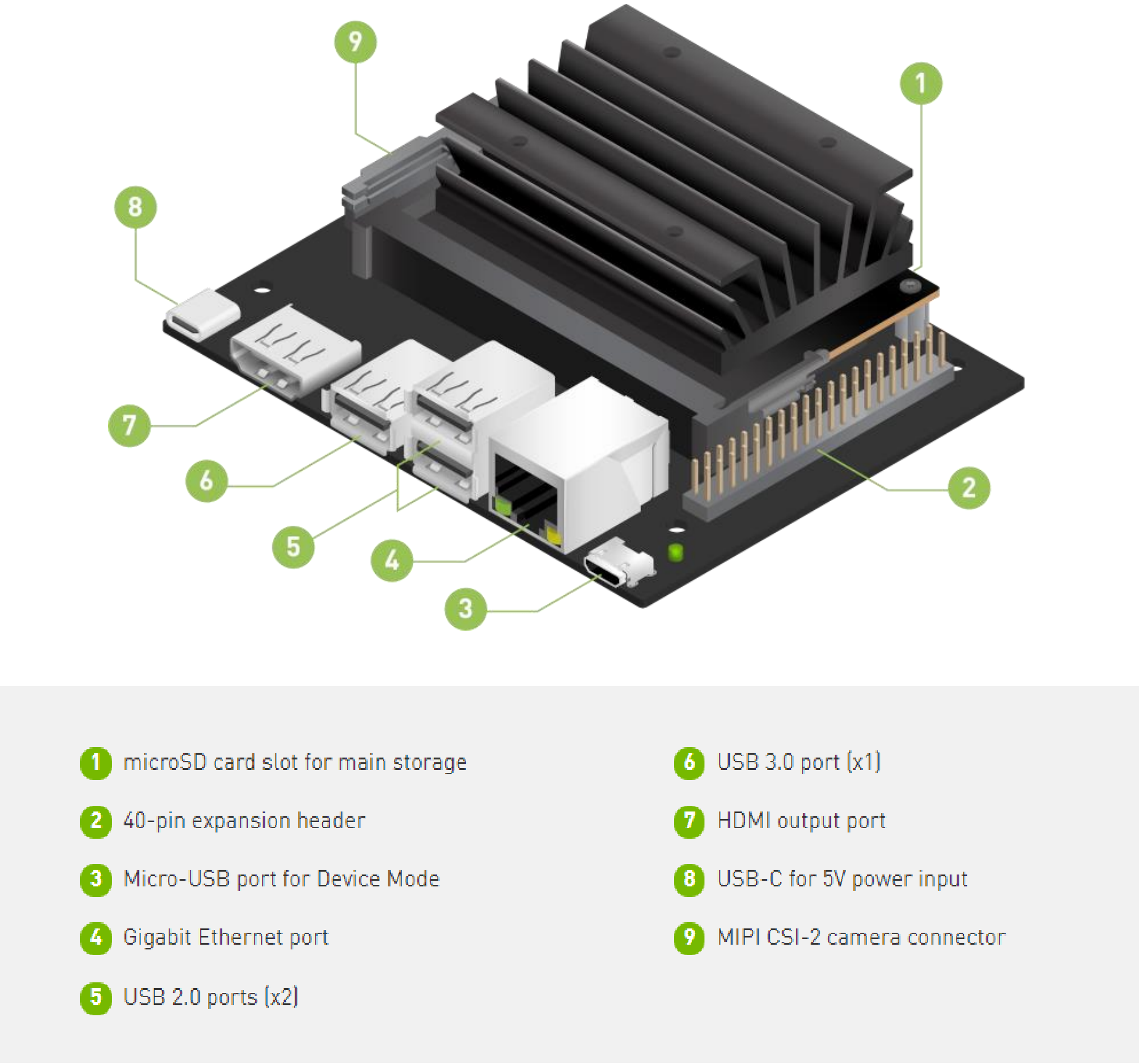
Our video capturing device or camera is one of the most important aspects of our project. We need one that can capture a wide enough angle, at least covering one section of the room, so we would need at least 160–180-degree FOV (field of view). Not to mention, our camera must also capture enough FPS for our systems accuracy and to ensure there are no bottlenecks.

Wi-Fi/Bluetooth strength will need to be strong for practical use but for testing purposes, we should be fine. something other than a person sitting in the chair like a bag or jacket for example. We will need to have our software filters out everything except human faces and chairs. Of course, there will be ways to trick our system (wearing a mask or large hoodie may cover the face, therefore not detecting someone in the chair).

Our system will also require a fast internet connection to easily upload our data and results to the website/application for people to track occupied seats and potentially reserve them if we are able to provide that capability. Fast data transfer is necessary for our system, or our video feed and data transfer will be crippled.

Technical Data:

Picture of FPGA/AI Kit:



1 microSD card slot for main storage

2 40-pin expansion header

3 Micro-USB port for Device Mode

4 Gigabit Ethernet port

5 USB 2.0 ports (x2)

6 USB 3.0 port (x1)

7 HDMI output port

8 USB-C for 5V power input

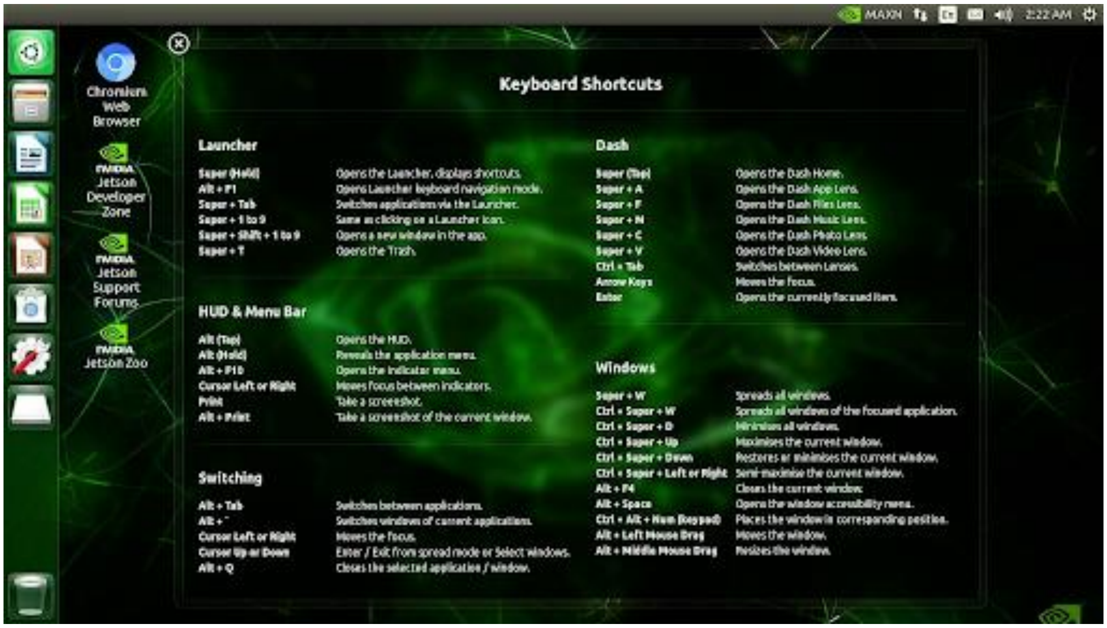
9 MIPI CSI-2 camera connector

Hardware Description:

We will use a similar hardware schematic like the picture above. Our version will have the NVIDIA Jetson Nano 2GB w/ Wi-Fi and Bluetooth adapter modification. There are connecter pins on the right-hand side of the board as well as an HDMI, USB, and Ethernet port although we will be using Wi-Fi with modular antennas for better connection. Also need to see how hot it gets during use and might need to add a fan to the top for better air circulation, we might just add it anyways because heat overtime just adds to the degradation of our parts.

This is what our final product will look like. For our design we will only be implementing one camera, the picture below shows a CSI camera. The case makes it easy to mount to a ceiling or easy to place anywhere and keeps all our hardware safe inside the case.

Jetson Nano's (JetPack 4.6) Linux Environment:



Keyboard Shortcuts

Launcher

Super (Hold)	Opens the Launcher, displays shortcuts.
Alt + F1	Opens Launcher keyboard navigation mode.
Super + Tab	Switches applications via the Launcher.
Super + 1 to 9	Same as clicking on a Launcher icon.
Super + Shift + 1 to 9	Opens a new window in the app.
Super + T	Opens the Trash.

HUD & Menu Bar

Alt (Tap)	Opens the HUD.
Alt (Hold)	Reveals the application menu.
Alt + F10	Opens the indicator menu.
Cursor Left or Right	Moves focus between indicators.
Print	Take a screenshot.
Alt + Print	Take a screenshot of the current window.

Switching

Alt + Tab	Switches between applications.
Alt + `	Switches windows of current applications.
Cursor Left or Right	Moves the focus.
Cursor Up or Down	Enter / Exit from spread mode or Select windows.
Alt + Q	Closes the selected application / window.

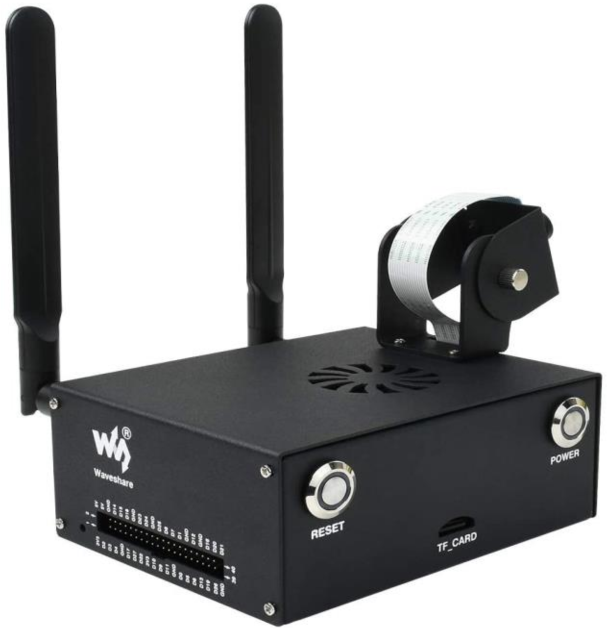
Dash

Super (Tap)	Opens the Dash Home.
Super + A	Opens the Dash App Lens.
Super + F	Opens the Dash Files Lens.
Super + M	Opens the Dash Music Lens.
Super + C	Opens the Dash Photo Lens.
Super + V	Opens the Dash Video Lens.
Ctrl + Tab	Switches between Lenses.
Arrow Keys	Moves the focus.
Enter	Opens the currently focused item.

Windows

Super + W	Spreads all windows.
Ctrl + Super + W	Spreads all windows of the focused application.
Ctrl + Super + D	Minimises all windows.
Ctrl + Super + Up	Maximises the current window.
Ctrl + Super + Down	Restores or minimises the current window.
Ctrl + Super + Left or Right	Semi-maximise the current window.
Alt + F4	Closes the current window.
Alt + Space	Opens the window accessibility menu.
Ctrl + Alt + Num (keypad)	Places the window in corresponding position.
Alt + Left Mouse Drag	Moves the window.
Alt + Middle Mouse Drag	Resizes the window.

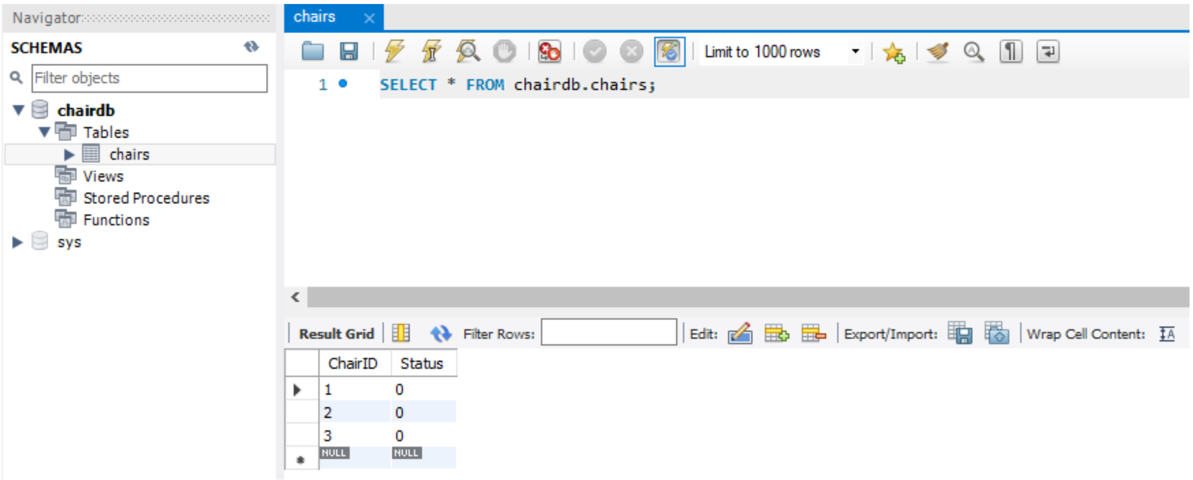
Our Completed Jetson Nano 2GB AI Detection Device:



Web Application Code:

MySQL:

Currently, the MySQL database is in a prototype stage, and will need to be hooked up to the Jetson Nano for any meaningful interactions to take place. It holds a very simple database called “chairdb,” in which a table exists called “chairs”. This chair table currently has three items, which are the three chairs we plan on first testing with the Nano’s AI recognition libraries. The chairs table’s items hold only two values now, which are “ChairID,” the integer identification we will use to differentiate the chairs, and “Status,” which is a Boolean value that is 0 or “False” for an empty chair, and 1 or “True” for an occupied chair. As of now, the values are only updated when changed by the administrator of the database, but once the Nano’s recognition algorithms are set to monitor the occupation of chairs, it will be able to alter this value on its own for each respective chair.

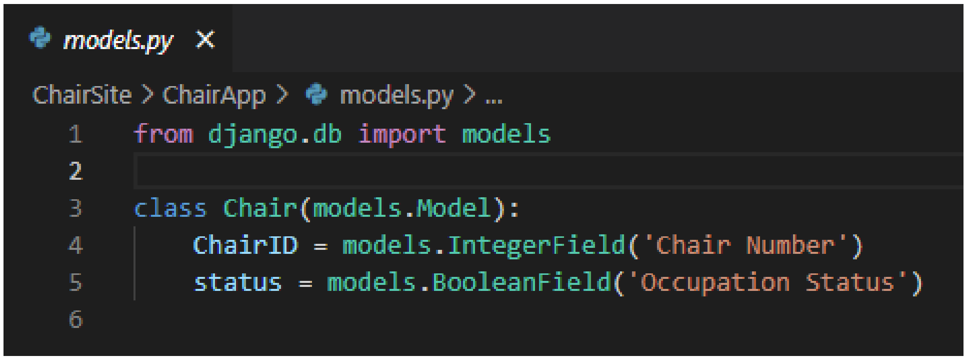


```
SELECT * FROM chairdb.chairs;
```

ChairID	Status
1	0
2	0
3	0
NULL	NULL

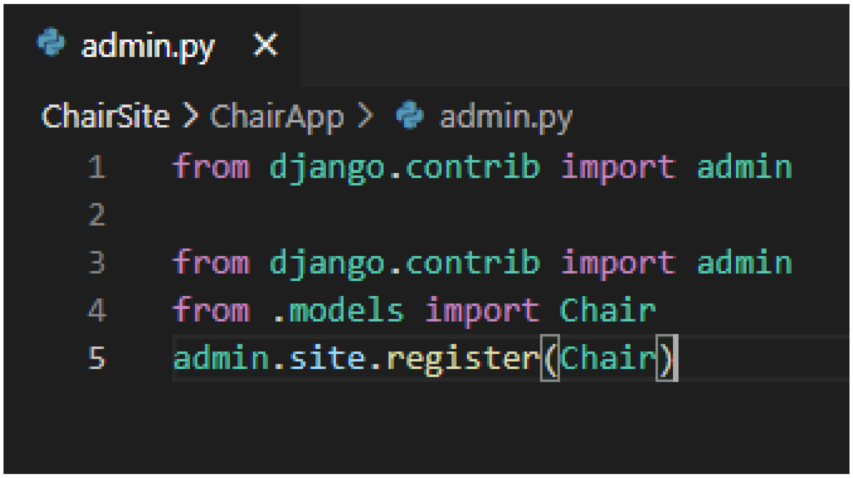
Django:

For the current prototype of the website, it does not yet have the seating diagram image, as it is something that will be set when the formation of the chairs is decided upon, but it does have the framework necessary to begin testing the interactions between it and the database. As previously said, Django offers the ability for websites to have tables that are much like that of the MySQL database. This is easily done with the template provided by Django, as their “models.py” file holds the different objects that are needed, which in this case is just the chair objects. Again, just like in the database, this object holds two values: the integer identification number and the Boolean status value.

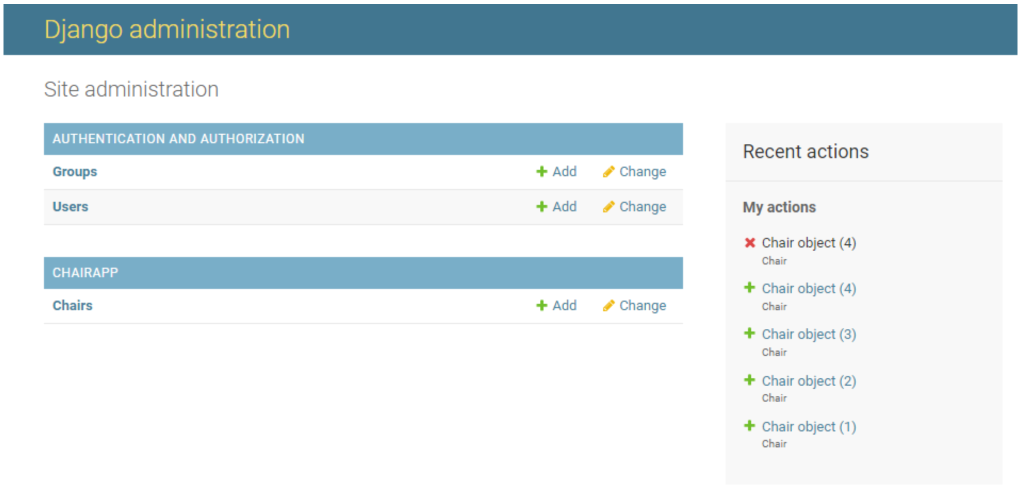


```python
from django.db import models

class Chair(models.Model):
    ChairID = models.IntegerField('Chair Number')
    status = models.BooleanField('Occupation Status')

```

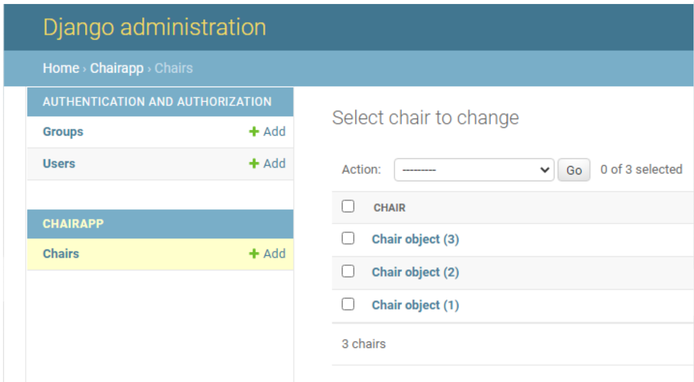
After the class objects are created, we can add a table of the chairs that can be controlled by an admin of the site. This is done by utilizing the "admin.py" file provided in Django, shown below. Once the basic framework of the website is completed, Django offers the ability to change data tables through the "admin" URL. Within the admin URL, we can see the application title, called "CHAIRAPP" here, and the table it currently holds called "Chairs," which is the one that was set within the code. As well, there are "Authentication and Authorization" abilities, which we are not implementing at the current time, but are something we will consider once the base project is completed.



```python
from django.contrib import admin

from django.contrib import admin
from .models import Chair
admin.site.register(Chair)
```



By clicking on the "Chairs" table, we can see the objects currently residing within the table. As of now, they are the three chairs, just like in the MySQL database, that we will be using for testing the Nano. Because Django makes it so simple to create these tables, it should be very simple to connect these mirroring values with those in the database, allowing them to be updated in tandem with one another.

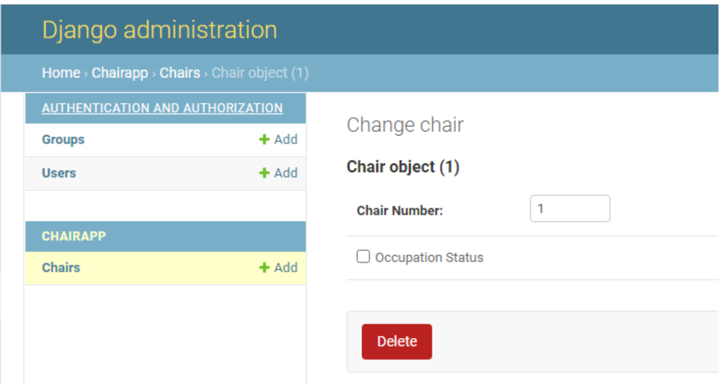


Select chair to change

Action: -------- 0 of 3 selected

□ CHAIR
□ Chair object (3)
□ Chair object (2)
□ Chair object (1)

3 chairs


Django administration
Home › Chairapp › Chairs › Chair object (1)
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
CHAIRAPP
Chairs + Add
Delete

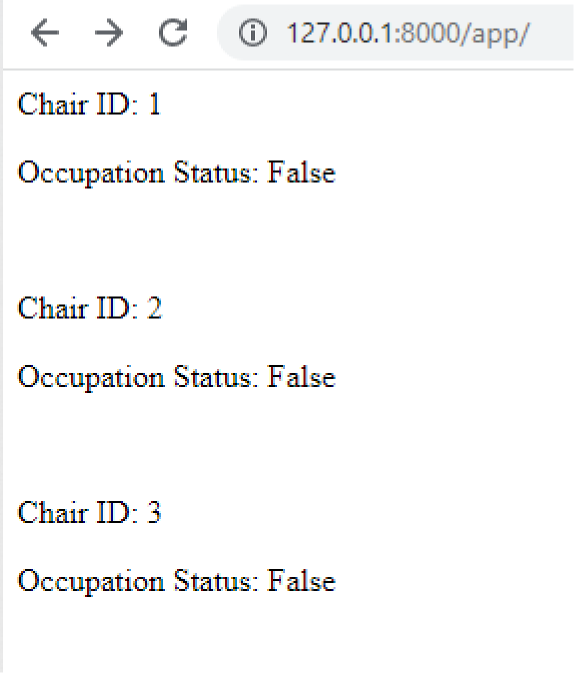
Change chair

Chair object (1)

Chair Number: 1

□ Occupation Status

Currently, the values of the chairs can be seen by a user utilizing the "app" URL. At this URL, we can see the current status of all the chair objects in the table. This is a read-only page that, unlike the admin URL, only allows the user to look at the data and not alter it. At this time, this will be the page we utilize for testing the communication between the database and the website. Once testing is completed, the text on this page will be replaced with a seating diagram that matches the monitored room, and the "Occupation Status" will be represented on the image by the color of the seat. However, this prototype will do just fine while the testing of the Nano's AI recognition system commences.



Chair ID: 1

Occupation Status: False

Chair ID: 2

Occupation Status: False

Chair ID: 3

Occupation Status: False

Future Technical Details: Libraries/Coding/Algorithms

We followed Dusty-NV's GitHub repo to install libraries of TensorFlow accelerated deep learning networks for image recognition, object detection with localization, and semantic segmentation. TensorFlow is NVIDIA's CUDA technology that gives the Jetson and its other hardware TX modules the capabilities for seamless object recognition and data gathering.

The main purpose of the repo is to help install libraries on the Jetson Nano which supports C++ and Python. This GitHub repo contains various pre-trained DNN models that are downloadable via Linux terminal. This repo has several libraries available for anyone to use to help achieve a baseline for their AI object detection system.

After flashing the Jetson Nano's JetPack (Operates on Linux OS), we cloned the repository and then proceeded to download and utilize the SSD-MobileNet V2 library for object detection. This specific library has tons of pretrained models and a vast neural network for machine learning. Our plan is to use this library but only train it for a person sitting in a chair, empty chair, backpack/random item in chair, and finally no chairs in case we want to monitor the amount of people in a single room.

While there are many libraries to choose from (Potential libraries listed below), we have been recently testing the SSD-MobileNet libraries because these are widely used by many developers. We can eventually create our own list of objects for it to detect, taking hundreds of pictures of the objects we want detected and then inputting them into our neural network for machine learning. This process will take us the longest, having to document and take pictures from every angle to make sure our object recognition is as accurate as possible.

Potential Libraries:

Coding examples:

Libraries	C++	Python
Image Recognition	imagenet.cpp	imagenet.py
Object Detection	detectnet.cpp	detectnet.py
Pose Estimation	posenet.cpp	posenet.py

Pre-trained Models for Image Recognition:

Network	CLI argument	NetworkType enum
AlexNet	alexnet	ALEXNET
GoogleNet	googlenet	GOOGLENET
GoogleNet-12	googlenet-12	GOOGLENET_12
ResNet-18	resnet-18	RESNET_18

ResNet-50	resnet-50	RESNET_50
ResNet-101	resnet-101	RESNET_101
ResNet-152	resnet-152	RESNET_152
VGG-16	vgg-16	VGG-16
VGG-19	vgg-19	VGG-19
Inception-v4	inception-v4	INCEPTION_V4

Pre-trained Models for Object Detection:

Network	CLI argument	NetworkType enum	Object classes
SSD-Mobilenet-v1	SSD-MobileNet-v1	SSD_MOBILENET_V1	91 (COCO classes)
SSD-Mobilenet-v2	SSD-MobileNet-v2	SSD_MOBILENET_V2	91 (COCO classes)
SSD-Inception-v2	SSD-Inception-v2	SSD_INCEPTION_V2	91 (COCO classes)
DetectNet-COCO-Chair	coco-chair	COCO_CHAIR	chairs
ped-100	pednet	PEDNET	pedestrians
multiped-500	multiped	PEDNET_MULTI	pedestrians, luggage
facenet-120	facenet	FACENET	faces

Pre-trained Models for Pose Estimation:

Model	CLI argument	NetworkType enum	Keypoints
Pose-ResNet18-Body	resnet18-body	RESNET18_BODY	18
Pose-DenseNet121-Body	densenet121-body	DENSENET121_BODY	18

Software & Hardware Tools Utilized:

Software:

NVIDIA Jetson Nano JetPack 4.6 Software:

Linux/Ubuntu 18.04 Aarch64:

Python v3.8 for Object Detection Algorithms/Code:

Django v3.2.9 for Web Application's Framework:

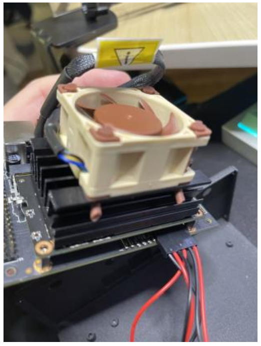
MySQL For Web Application's Relational Database:

Visual Studio Code For Use As an IDE:

Hardware:

NVIDIA Jetson Nano 2GB:

Jetson Nano and Cooling Fan (Outside of Case):



Jetson Nano with connections:

Budget Laptop for Flashing/Programming/Updating:

Waveshare Jetson Nano 4 PIN Fan:

SanDisk 64GB microSD Card w/Adapter:

Raspberry Pi-2 CSI Camera:

Logitech c922 USB Camera (for testing purposes):

Geekworm USB Wi-Fi adapter with Antenna:

Lazy-Boy chair (testing purposes):

Keyboard/Mouse for Jetson Nano setup:

HDMI Cable(s):

USB 3.0/2.0 cables:

Raspberry Pi USB-C Power supply:

Cost Analysis:

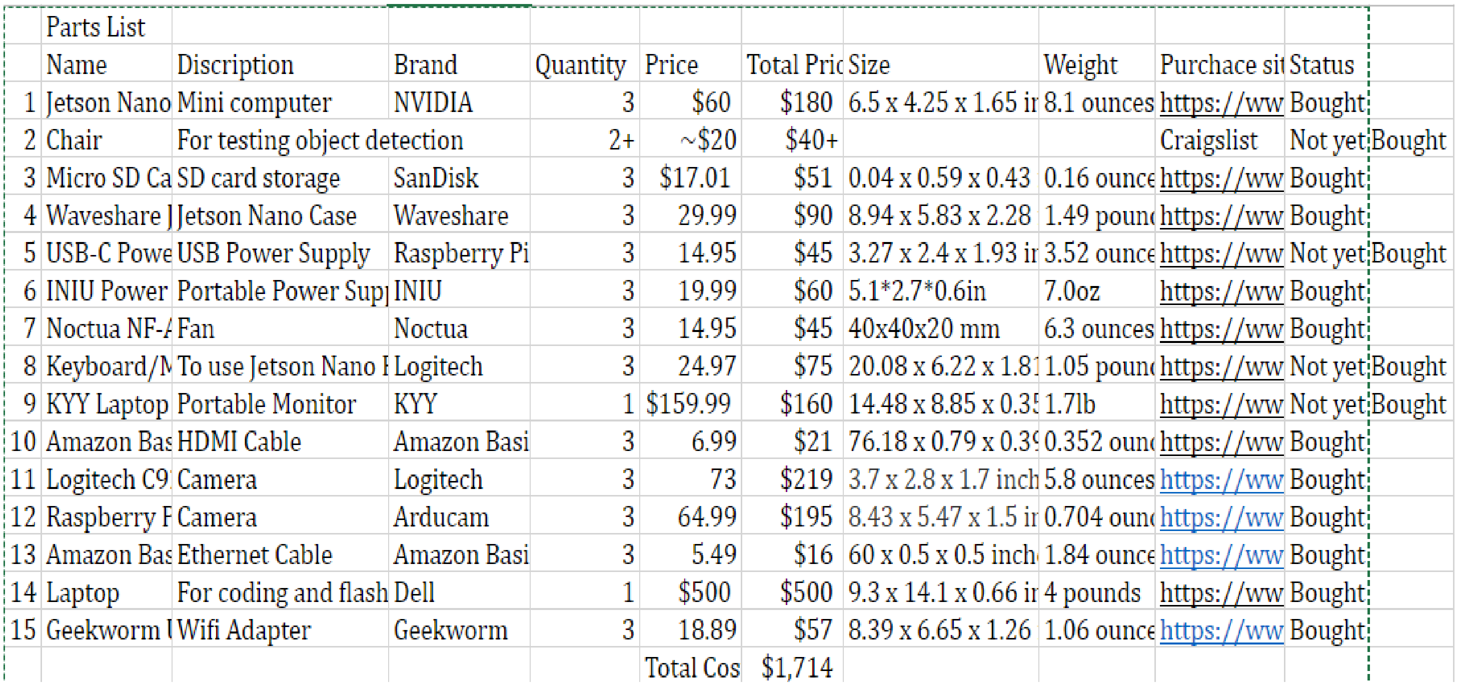
Labor:

- **Justin Miller:** ($150/hour) x 2.5 x 228 = $85,500 pre-tax.
- **Daniel Lee:** ($100/hour) x 2.5 x 228 = $57,000.
- **Andrew Gaspar:** ($100/hour) x 2.5 x 228 = $57,000.
- **Alexander Vallardez:** ($100/hour) x 2.5 x 228 = $57,000.
- **Total = $256,500 USD + Stock options available.**

The team will work around 6 hours a week each and will work on this project from the beginning of the school year to the end. We will work an estimated 38 weeks (about 8 and a half months) which comes out to 228 hours (about 1 and a half weeks) total.

Total Parts List for Research/Testing/Project Supplies:

- Parts List - Full Excel spread sheet parts list link with links to parts.



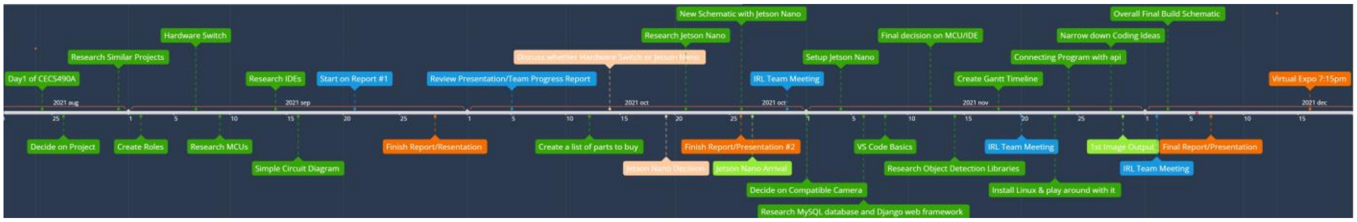
	Parts List									
	Name	Discription	Brand	Quantity	Price	Total Pric	Size	Weight	Purchace sit	Status
1	Jetson Nano	Mini computer	NVIDIA	3	$60	$180	6.5 x 4.25 x 1.65 ir	8.1 ounces	https://ww	Bought
2	Chair	For testing object detection		2+	~$20	$40+			Craigslist	Not yet Bought
3	Micro SD Ca	SD card storage	SanDisk	3	$17.01	$51	0.04 x 0.59 x 0.43	0.16 ounce	https://ww	Bought
4	Waveshare J	Jetson Nano Case	Waveshare	3	29.99	$90	8.94 x 5.83 x 2.28	1.49 poun	https://ww	Bought
5	USB-C Powe	USB Power Supply	Raspberry Pi	3	14.95	$45	3.27 x 2.4 x 1.93 ir	3.52 ounce	https://ww	Not yet Bought
6	INIU Power	Portable Power Sup	INIU	3	19.99	$60	5.1*2.7*0.6in	7.0oz	https://ww	Bought
7	Noctua NF-A	Fan	Noctua	3	14.95	$45	40x40x20 mm	6.3 ounces	https://ww	Bought
8	Keyboard/M	To use Jetson Nano I	Logitech	3	24.97	$75	20.08 x 6.22 x 1.81	1.05 poun	https://ww	Not yet Bought
9	KYY Laptop	Portable Monitor	KYY	1	$159.99	$160	14.48 x 8.85 x 0.35	1.7lb	https://ww	Not yet Bought
10	Amazon Bas	HDMI Cable	Amazon Basi	3	6.99	$21	76.18 x 0.79 x 0.39	0.352 oun	https://ww	Bought
11	Logitech C9	Camera	Logitech	3	73	$219	3.7 x 2.8 x 1.7 inch	5.8 ounces	https://ww	Bought
12	Raspberry P	Camera	Arducam	3	64.99	$195	8.43 x 5.47 x 1.5 ir	0.704 oun	https://ww	Bought
13	Amazon Bas	Ethernet Cable	Amazon Basi	3	5.49	$16	60 x 0.5 x 0.5 inch	1.84 ounce	https://ww	Bought
14	Laptop	For coding and flash	Dell	1	$500	$500	9.3 x 14.1 x 0.66 ir	4 pounds	https://ww	Bought
15	Geekworm U	Wifi Adapter	Geekworm	3	18.89	$57	8.39 x 6.65 x 1.26	1.06 ounce	https://ww	Bought
					Total Cos	$1,714				

- **Total Cost of all materials: $1,714 Before Tax**
- **Labor: $256,500.00**
- **Total Cost w/Labor: $258,214.00 USD**

Schedule/Timeline:

Overall Timeline:

https://time.graphics/line/591531

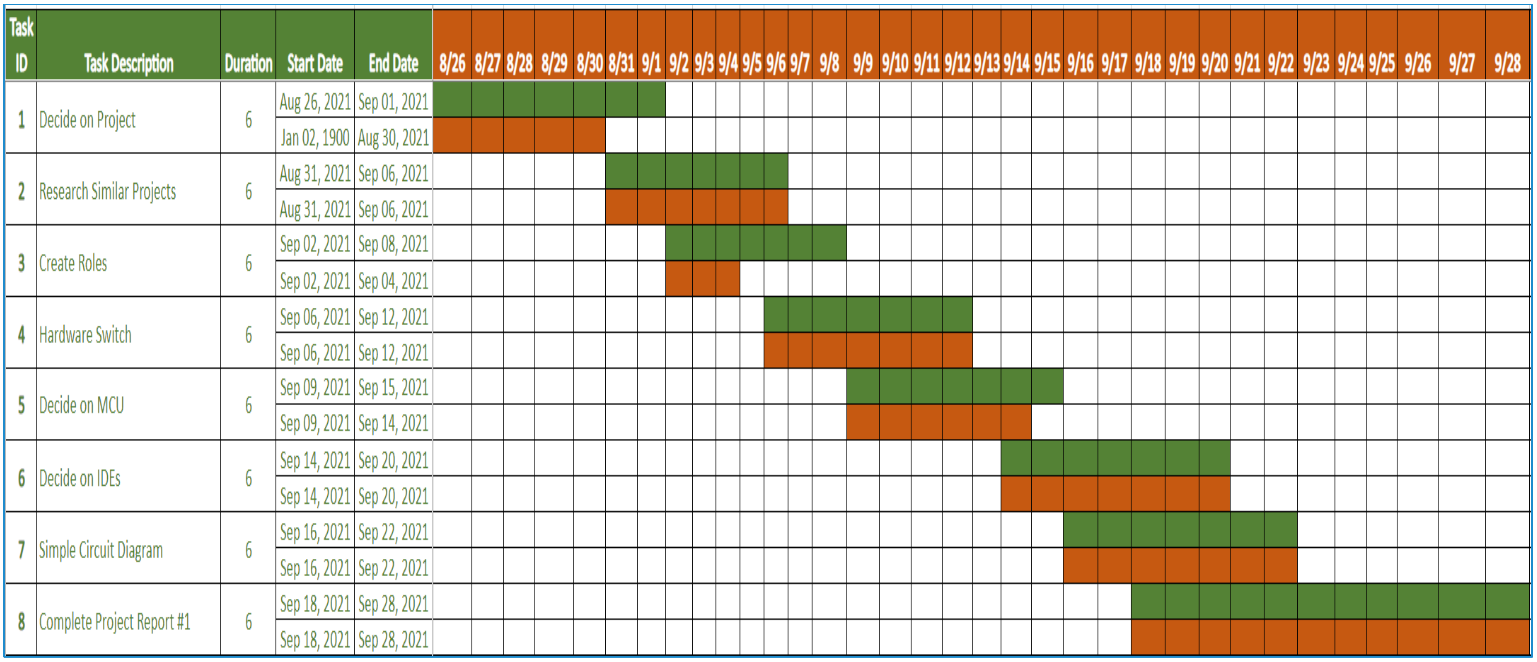

Day1 of CECS490A
Research Similar Projects
Hardware Switch
Research IDEs
Start on Report #1
Review Presentation/Team Progress Report
Discuss whether Hardware Switch or Jetson Nano
Research Jetson Nano
New Schematic with Jetson Nano
IRL Team Meeting
Setup Jetson Nano
Final decision on MCU/IDE
Create Gantt Timeline
Connecting Program with api
Narrow down Coding Ideas
Overall Final Build Schematic
Virtual Expo 7:15pm
2021 aug
2021 sep
2021 oct
2021 nov
2021 dec
25
1
5
10
15
20
25
1
5
10
15
20
25
1
5
10
15
20
25
1
5
10
15
Decide on Project
Create Roles
Research MCUs
Simple Circuit Diagram
Finish Report/Resentation
Create a list of parts to buy
Jetson Nano Decision
Finish Report/Presentation #2
Jetson Nano Arrival
Decide on Compatible Camera
Research MySQL database and Django web framework
VS Code Basics
Research Object Detection Libraries
IRL Team Meeting
Install Linux & play around with it
1st Image Output
Final Report/Presentation
IRL Team Meeting

Planned vs Actual Timeline

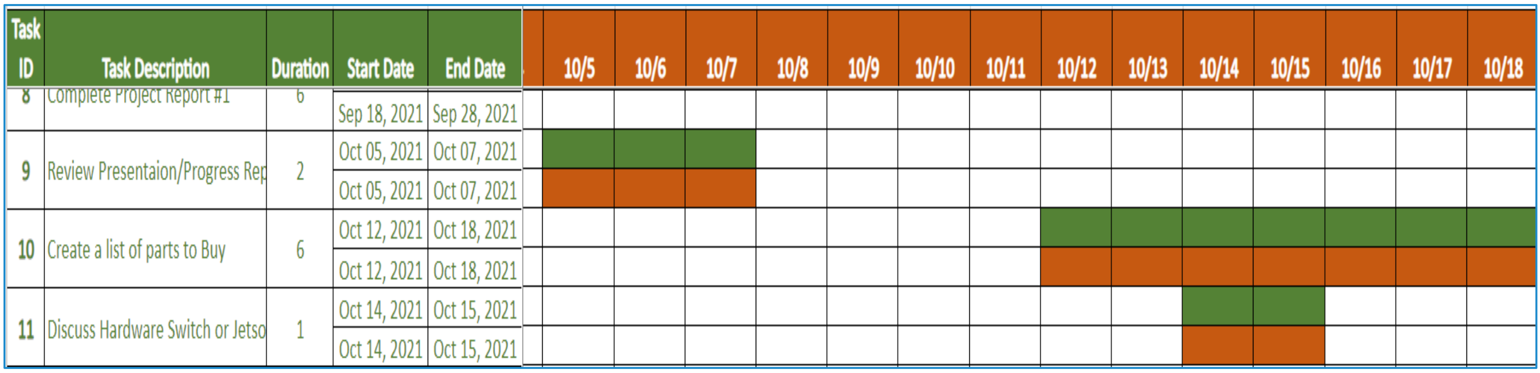
Green = Planned Time

Orange = Actual Time

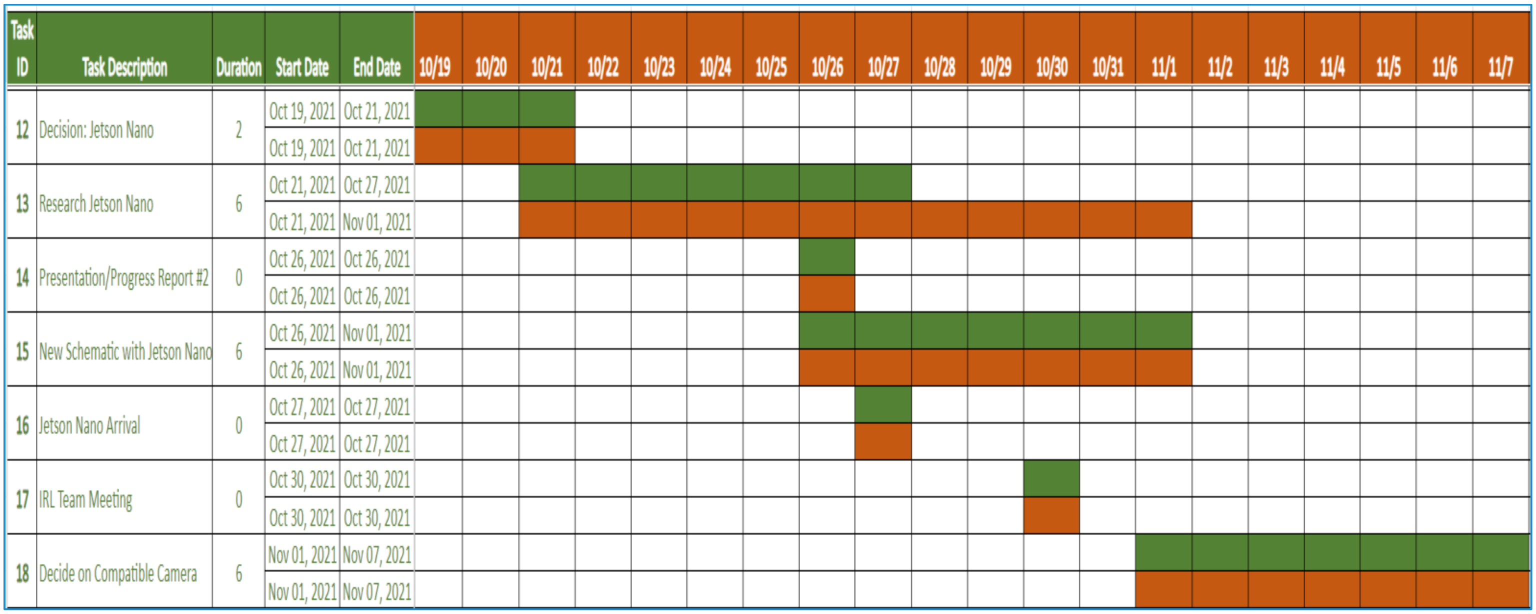
Link to timeline: Gantt Timeline



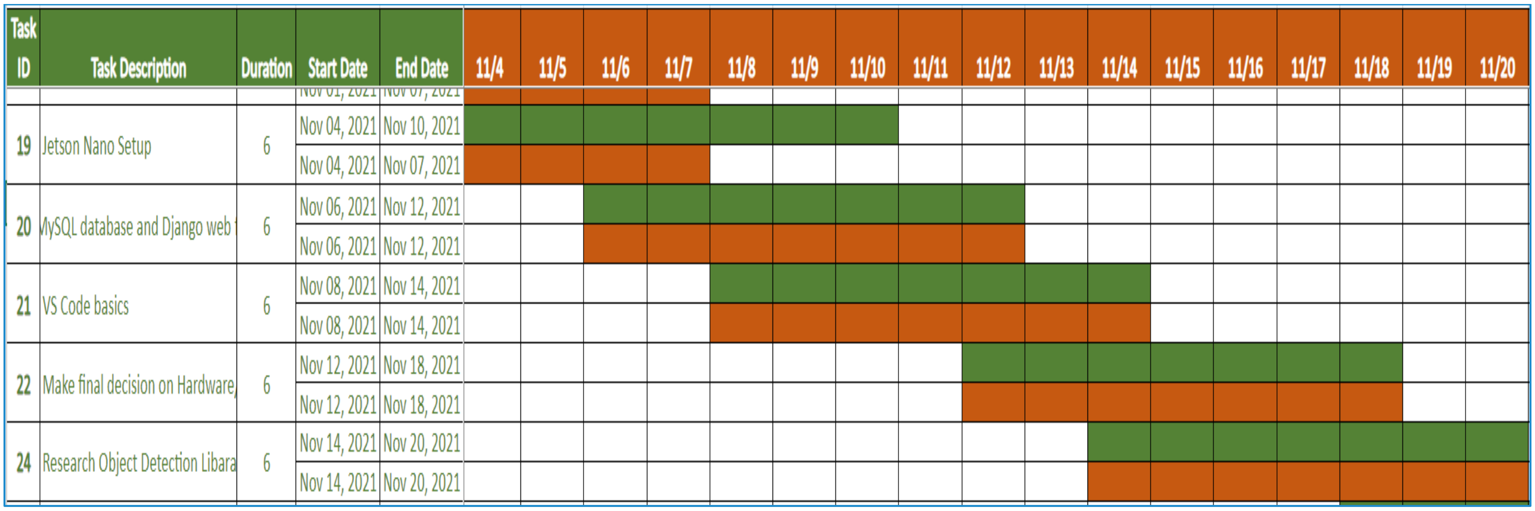
Task ID	Task Description	Duration	Start Date	End Date	8/26	8/27	8/28	8/29	8/30	8/31	9/1	9/2	9/3	9/4	9/5	9/6	9/7	9/8	9/9	9/10	9/11	9/12	9/13	9/14	9/15	9/16	9/17	9/18	9/19	9/20	9/21	9/22	9/23	9/24	9/25	9/26	9/27	9/28
1	Decide on Project	6	Aug 26, 2021	Sep 01, 2021	🟩	🟩	🟩	🟩	🟩	🟩	🟩																											
			Jan 02, 1900	Aug 30, 2021	🟧	🟧	🟧	🟧	🟧																													
2	Research Similar Projects	6	Aug 31, 2021	Sep 06, 2021						🟩	🟩	🟩	🟩	🟩	🟩	🟩																						
			Aug 31, 2021	Sep 06, 2021						🟧	🟧	🟧	🟧	🟧	🟧	🟧																						
3	Create Roles	6	Sep 02, 2021	Sep 08, 2021								🟩	🟩	🟩	🟩	🟩	🟩	🟩																				
			Sep 02, 2021	Sep 04, 2021								🟧	🟧	🟧																								
4	Hardware Switch	6	Sep 06, 2021	Sep 12, 2021												🟩	🟩	🟩	🟩	🟩	🟩	🟩																
			Sep 06, 2021	Sep 12, 2021												🟧	🟧	🟧	🟧	🟧	🟧	🟧																
5	Decide on MCU	6	Sep 09, 2021	Sep 15, 2021															🟩	🟩	🟩	🟩	🟩	🟩	🟩													
			Sep 09, 2021	Sep 14, 2021															🟧	🟧	🟧	🟧	🟧	🟧														
6	Decide on IDEs	6	Sep 14, 2021	Sep 20, 2021																				🟩	🟩	🟩	🟩	🟩	🟩	🟩								
			Sep 14, 2021	Sep 20, 2021																				🟧	🟧	🟧	🟧	🟧	🟧	🟧								
7	Simple Circuit Diagram	6	Sep 16, 2021	Sep 22, 2021																						🟩	🟩	🟩	🟩	🟩	🟩	🟩						
			Sep 16, 2021	Sep 22, 2021																						🟧	🟧	🟧	🟧	🟧	🟧	🟧						
8	Complete Project Report #1	6	Sep 18, 2021	Sep 28, 2021																								🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
			Sep 18, 2021	Sep 28, 2021																								🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧



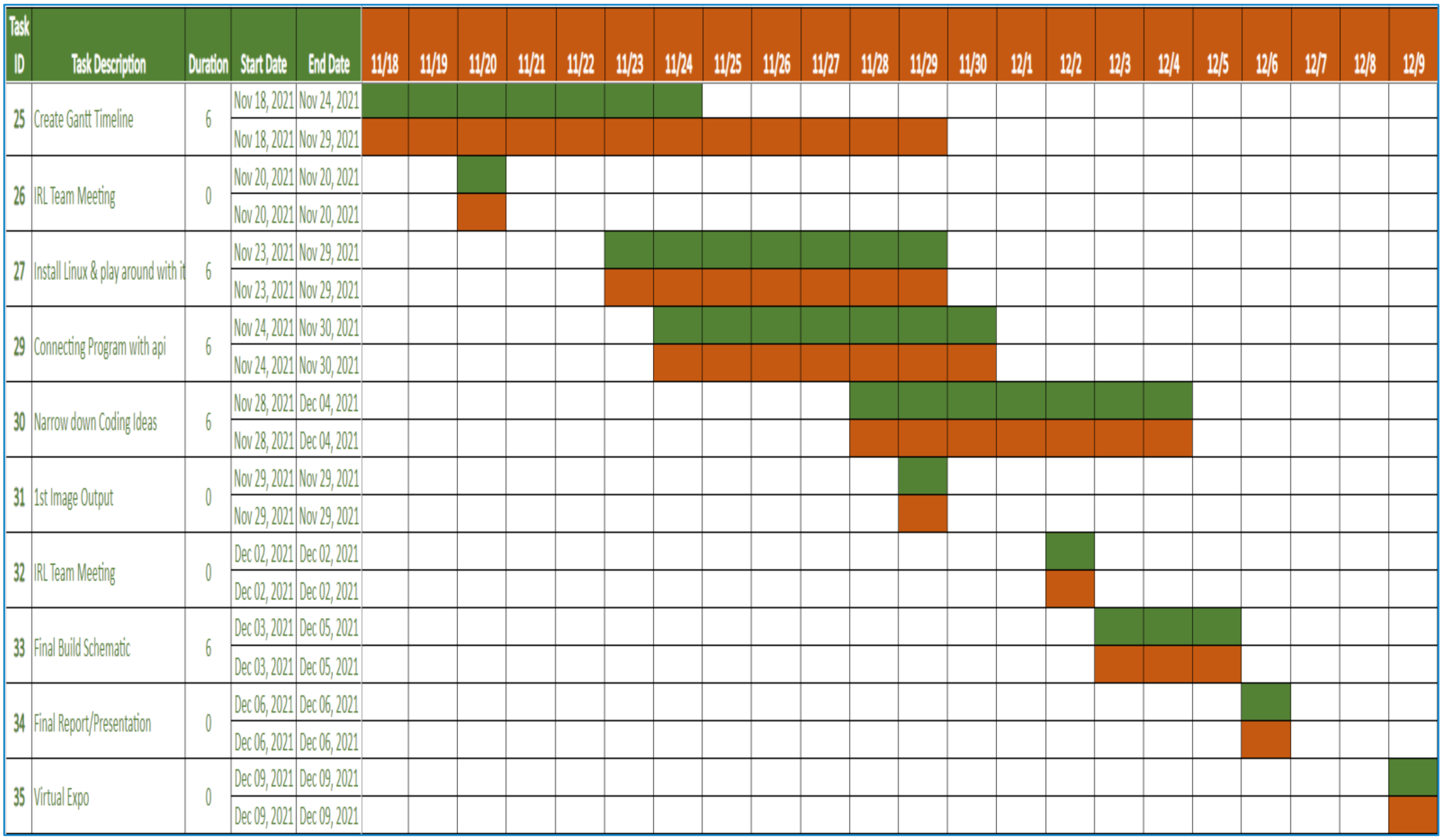
Task ID	Task Description	Duration	Start Date	End Date	10/5	10/6	10/7	10/8	10/9	10/10	10/11	10/12	10/13	10/14	10/15	10/16	10/17	10/18
8	Complete Project Report #1	6	Sep 18, 2021	Sep 28, 2021														
9	Review Presentaion/Progress Rep	2	Oct 05, 2021	Oct 07, 2021	🟩	🟩	🟩											
			Oct 05, 2021	Oct 07, 2021	🟧	🟧	🟧											
10	Create a list of parts to Buy	6	Oct 12, 2021	Oct 18, 2021								🟩	🟩	🟩	🟩	🟩	🟩	🟩
			Oct 12, 2021	Oct 18, 2021								🟧	🟧	🟧	🟧	🟧	🟧	🟧
11	Discuss Hardware Switch or Jetso	1	Oct 14, 2021	Oct 15, 2021										🟩	🟩			
			Oct 14, 2021	Oct 15, 2021										🟧	🟧			



Task ID	Task Description	Duration	Start Date	End Date	10/19	10/20	10/21	10/22	10/23	10/24	10/25	10/26	10/27	10/28	10/29	10/30	10/31	11/1	11/2	11/3	11/4	11/5	11/6	11/7
12	Decision: Jetson Nano	2	Oct 19, 2021	Oct 21, 2021	🟩	🟩	🟩																	
			Oct 19, 2021	Oct 21, 2021	🟧	🟧	🟧																	
13	Research Jetson Nano	6	Oct 21, 2021	Oct 27, 2021			🟩	🟩	🟩	🟩	🟩	🟩	🟩											
			Oct 21, 2021	Nov 01, 2021			🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧						
14	Presentation/Progress Report #2	0	Oct 26, 2021	Oct 26, 2021								🟩												
			Oct 26, 2021	Oct 26, 2021								🟧												
15	New Schematic with Jetson Nano	6	Oct 26, 2021	Nov 01, 2021								🟩	🟩	🟩	🟩	🟩	🟩	🟩						
			Oct 26, 2021	Nov 01, 2021								🟧	🟧	🟧	🟧	🟧	🟧	🟧						
16	Jetson Nano Arrival	0	Oct 27, 2021	Oct 27, 2021									🟩											
			Oct 27, 2021	Oct 27, 2021									🟧											
17	IRL Team Meeting	0	Oct 30, 2021	Oct 30, 2021												🟩								
			Oct 30, 2021	Oct 30, 2021												🟧								
18	Decide on Compatible Camera	6	Nov 01, 2021	Nov 07, 2021														🟩	🟩	🟩	🟩	🟩	🟩	🟩
			Nov 01, 2021	Nov 07, 2021														🟧	🟧	🟧	🟧	🟧	🟧	🟧

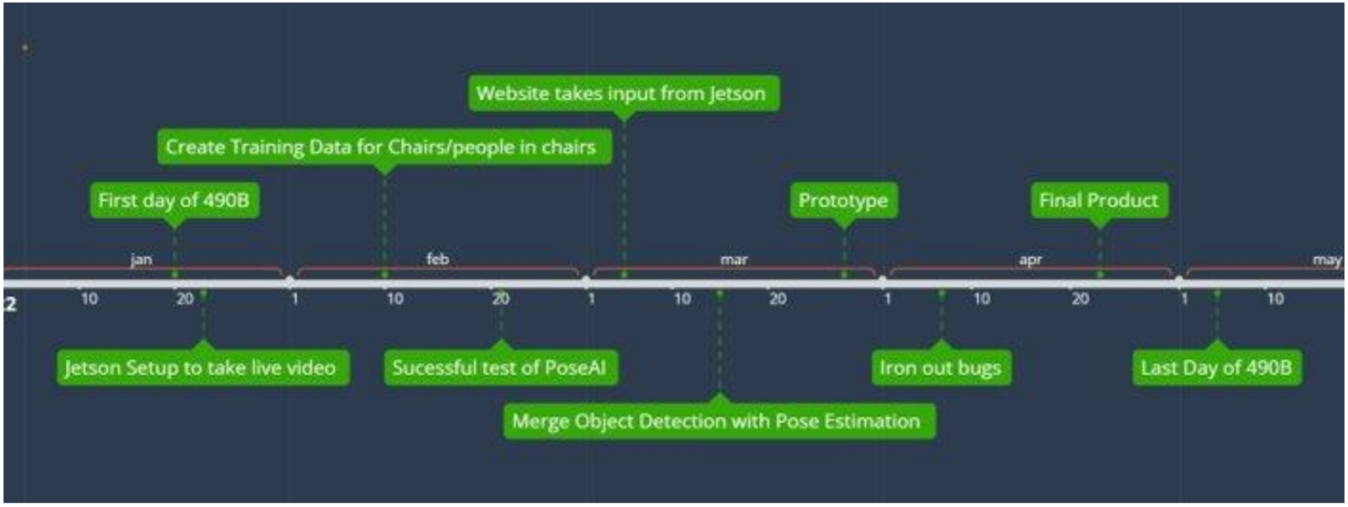


Task ID	Task Description	Duration	Start Date	End Date	11/4	11/5	11/6	11/7	11/8	11/9	11/10	11/11	11/12	11/13	11/14	11/15	11/16	11/17	11/18	11/19	11/20
			Nov 01, 2021	Nov 07, 2021	🟧	🟧	🟧	🟧													
19	Jetson Nano Setup	6	Nov 04, 2021	Nov 10, 2021	🟩	🟩	🟩	🟩	🟩	🟩	🟩										
			Nov 04, 2021	Nov 07, 2021	🟧	🟧	🟧	🟧													
20	MySQL database and Django web f	6	Nov 06, 2021	Nov 12, 2021			🟩	🟩	🟩	🟩	🟩	🟩	🟩								
			Nov 06, 2021	Nov 12, 2021			🟧	🟧	🟧	🟧	🟧	🟧	🟧								
21	VS Code basics	6	Nov 08, 2021	Nov 14, 2021					🟩	🟩	🟩	🟩	🟩	🟩	🟩						
			Nov 08, 2021	Nov 14, 2021					🟧	🟧	🟧	🟧	🟧	🟧	🟧						
22	Make final decision on Hardware,	6	Nov 12, 2021	Nov 18, 2021									🟩	🟩	🟩	🟩	🟩	🟩	🟩		
			Nov 12, 2021	Nov 18, 2021									🟧	🟧	🟧	🟧	🟧	🟧	🟧		
24	Research Object Detection Libara	6	Nov 14, 2021	Nov 20, 2021											🟩	🟩	🟩	🟩	🟩	🟩	🟩
			Nov 14, 2021	Nov 20, 2021											🟧	🟧	🟧	🟧	🟧	🟧	🟧



Task ID	Task Description	Duration	Start Date	End Date	11/18	11/19	11/20	11/21	11/22	11/23	11/24	11/25	11/26	11/27	11/28	11/29	11/30	12/1	12/2	12/3	12/4	12/5	12/6	12/7	12/8	12/9
25	Create Gantt Timeline	6	Nov 18, 2021	Nov 24, 2021	🟩	🟩	🟩	🟩	🟩	🟩	🟩															
			Nov 18, 2021	Nov 29, 2021	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧										
26	IRL Team Meeting	0	Nov 20, 2021	Nov 20, 2021			🟩																			
			Nov 20, 2021	Nov 20, 2021			🟧																			
27	Install Linux & play around with it	6	Nov 23, 2021	Nov 29, 2021						🟩	🟩	🟩	🟩	🟩	🟩	🟩										
			Nov 23, 2021	Nov 29, 2021						🟧	🟧	🟧	🟧	🟧	🟧	🟧										
29	Connecting Program with api	6	Nov 24, 2021	Nov 30, 2021							🟩	🟩	🟩	🟩	🟩	🟩	🟩									
			Nov 24, 2021	Nov 30, 2021							🟧	🟧	🟧	🟧	🟧	🟧	🟧									
30	Narrow down Coding Ideas	6	Nov 28, 2021	Dec 04, 2021											🟩	🟩	🟩	🟩	🟩	🟩	🟩					
			Nov 28, 2021	Dec 04, 2021											🟧	🟧	🟧	🟧	🟧	🟧	🟧					
31	1st Image Output	0	Nov 29, 2021	Nov 29, 2021												🟩										
			Nov 29, 2021	Nov 29, 2021												🟧										
32	IRL Team Meeting	0	Dec 02, 2021	Dec 02, 2021															🟩							
			Dec 02, 2021	Dec 02, 2021															🟧							
33	Final Build Schematic	6	Dec 03, 2021	Dec 05, 2021																🟩	🟩	🟩				
			Dec 03, 2021	Dec 05, 2021																🟧	🟧	🟧				
34	Final Report/Presentation	0	Dec 06, 2021	Dec 06, 2021																			🟩			
			Dec 06, 2021	Dec 06, 2021																			🟧			
35	Virtual Expo	0	Dec 09, 2021	Dec 09, 2021																						🟩
			Dec 09, 2021	Dec 09, 2021																						🟧

Semester 2 Timeline



Significant Challenges and Solutions:

The most significant challenge we will face with our project is the programming of Jetson's AI, as well as hardware limitations. As a technology we have not utilized in any of our classes, we will rely heavily on outside sources and libraries for knowledge to accomplish our goal in having the AI discern between objects. We know the broad details of what needs to be done regarding the training of the AI to recognize chairs and people, but once we begin, we are sure to encounter unforeseen problems in the coding. Also, we are expecting the hardware we plan to use to be adequate for what o ur project requires. If the resolution of our camera is not high enough for the AI to consistently detect objects, we will need to figure out a solution, which could range from decreasing the size of the room we are attempting to track, or outright purchasing a camera that can let the AI more clearly scan the area.

Stretch Goals:

At this moment, our stretch goals consist of having the ability to reserve seating, as well as sending alerts if a reserved spot is being taken. At this point, we are simply trying to have our system accurately track occupancy first, then occupancy of chairs next. Eventually, we want to have the ability for a user to reserve a single seat in any given public area where our devices are located. There are many aspects to this that we must account for, such as having timing blocks that do not become a hindrance to others attempting to find a seat, and having the ability to send these alerts, which may require specific accounts to find the individual who posted the reservation, but these are problems that we can work out once we reach a point where implementing it will not interfere with the completion of the base project.

Project Follow Up/Future Expansion:

With unlimited resources and funding, we would expand our Seat Taken idea to airports, schools, businesses, or any other crowded places. Most of our project would remain the same as far as the build goes except for more cameras per device. However, we would require much more equipment and data processing capabilities to handle such a large network to mass produce it. With our current resources, we are limited to having only a single room for our project to track the occupancy of, and with only having one camera limits the amount of coverage we have. Finally, it would also require an extensive amount of user testing and machine learning to accomplish it on such a big scale.

Intellectual Property:

Our website/application will allow users to know if a seat is taken. Our project's main goal is to save people's time looking for a seat instead of walking around in circles hoping to find an available one. Similar versions of this type of occupancy detection already exist for cars, how many people are in a room (occupancy limits), how many people enter any given area etc. However, our project is different because we will be connecting our device to the internet and can transmit data wirelessly to our server for our application/website for people to monitor open seats and areas where room is available. While Doctors are finding out resting more results in more productive workforce, making this the perfect application for a universities or businesses to implement for their populace.

Sample Object Detection Video:

https://youtu.be/WQwrGq4V-Ps

Appendix:

References

- *Use of Occupancy Sensors in Led Parking Lot and ... - PNNL.* https://www.pnnl.gov/main/publications/external/technical_reports/PNNL-21923.pdf.

- Dusty-Nv. "Dusty-NV/Jetson-Inference: Hello AI World Guide to Deploying Deep-Learning Inference Networks and Deep Vision Primitives with TENSORRT and Nvidia Jetson." *GitHub*, https://github.com/dusty-nv/jetson-inference.

- "Getting Started with Jetson Nano Developer Kit." *NVIDIA Developer*, 12 Apr. 2021, https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit.

- Brownlee, Jason. "A Gentle Introduction to Deep Learning for Face Recognition." *Machine Learning Mastery*, 5 July 2019, https://machinelearningmastery.com/introduction-to-deep-learning-for-face-recognition/

- "Adding Wi-Fi to the Jetson Nano." *Adding Wifi to the Nvidia Jetson*, https://learn.sparkfun.com/tutorials/adding-wifi-to-the-nvidia-jetson/all.

- "Object Detection Guide." *Fritz*, https://www.fritz.ai/object-detection/.

- "Smart Parking: Parking Occupancy Monitoring and Visualization System for Smart Cities." *IEEE Xplore*, https://ieeexplore.ieee.org/document/7506721?fbclid=IwAR2X-XnvwS5HFBtTYSb9dHthGGdLMrmsEMvtfCSM1mA59q45urm6JlSe9dg.