

Faculty of Engineering & Applied Science

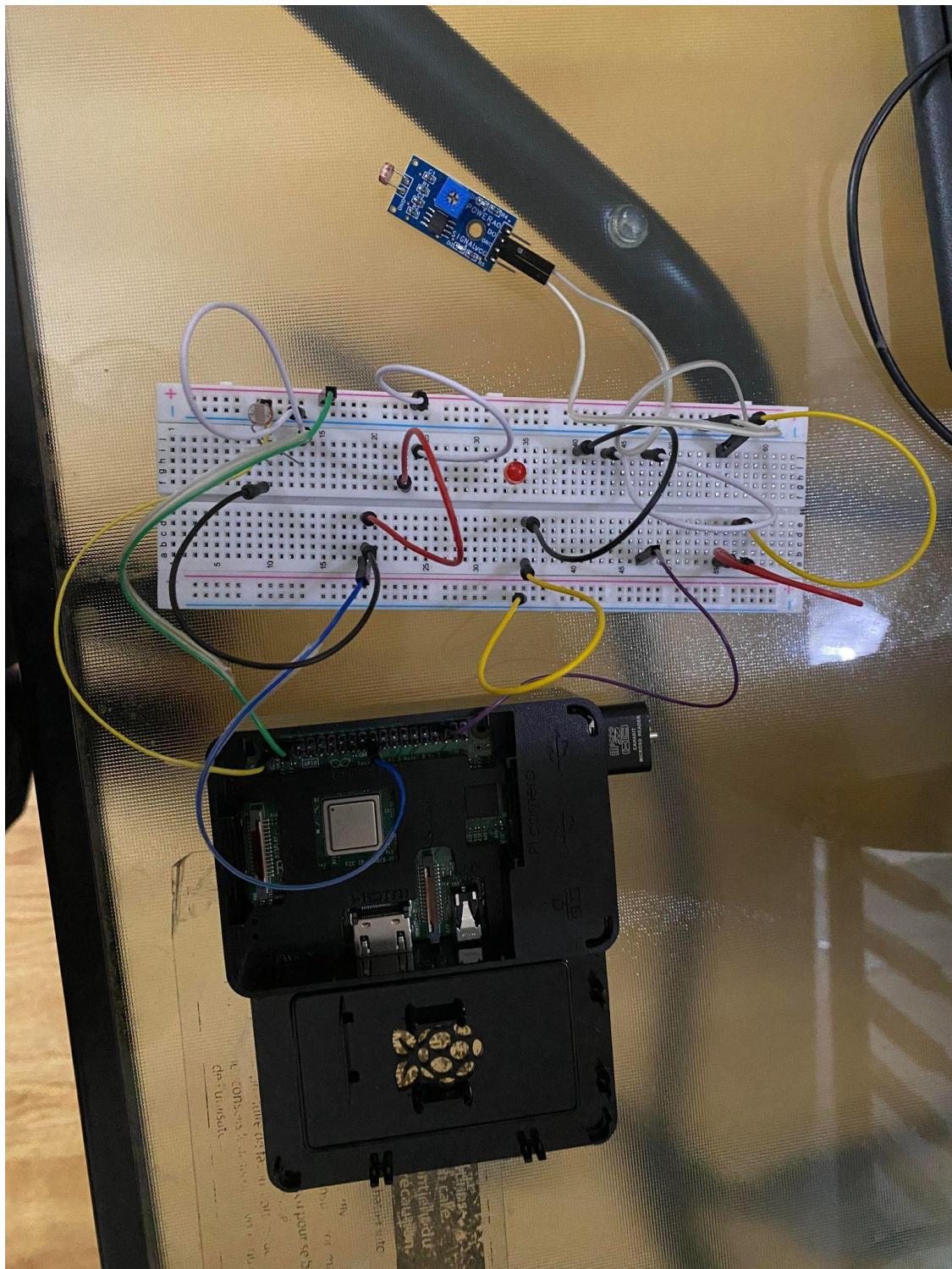


SOFE 4610U Design & Analysis of IoT

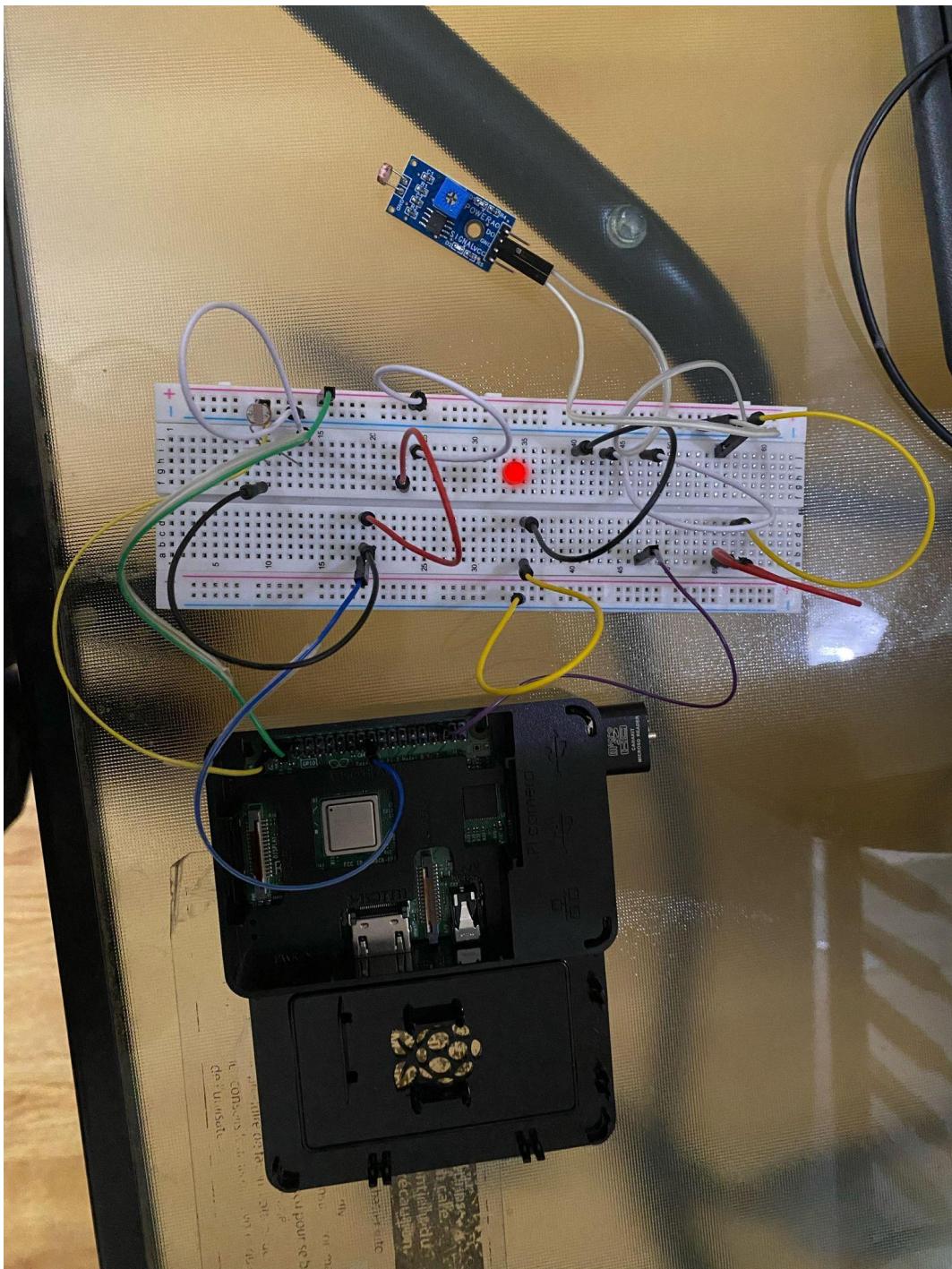
Assignment-3

Group#: 2

First Name	Last Name	Student Number
Nivetha	Gnaneswaran	100695935
Rodaba	Ebadi	100708585
Munazza	Fahmeen	



In this screenshot you can see that our setup of the system. This includes the use of the Raspberry Pi which was used as the sensor device, with the photoresister sensor which was used for detecting the light intensity, and then the LED which was used for the light itself. Jumper cables were used to ensure all the components were connected as they should be.



In this screenshot you can see that we successfully turned on the LED light using the resistor and the capacitor accordingly.

The screenshot shows a terminal window on a Raspberry Pi. The user is in a virtual environment named 'djenv'. They run the command `sudo python3 -m pip install django`. The terminal shows the download progress of several packages from PyPI and the Pi Wheeels repository. After the packages are collected, they are installed. The user then runs `sudo nano /home/pi/pidjango/pidjango/settings.py` three times, likely to edit the settings file. They also run `sudo nano /var/log/apache2/error.log` and `rm -rf venv`. Finally, they attempt to create a new virtual environment with `virtualenv -p /usr/bin/python3 venv`, which fails because Python 3 is not found. The user then tries to activate the environment with `source venv/bin/activate`, which also fails due to the same reason. A tooltip at the bottom right provides information about the PYTHONHOME environment variable.

```
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango $ sudo python3 -m pip install django
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting django
  Downloading https://www.piwheels.org/simple/django/Django-4.1.3-py3-none-any.whl (8.1 MB)
    |████████| 8.1 MB 57 kB/s
Collecting sqlparse>=0.2.2
  Downloading https://www.piwheels.org/simple/sqlparse/sqlparse-0.4.3-py3-none-any.whl (42 kB)
    |████████| 42 kB 169 kB/s
Re-installing collected packages: sqlparse, asgiref, django
Successfully installed asgiref-3.5.2 django-4.1.3 sqlparse-0.4.3
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango $ sudo django-admin startproject pidjango .
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango $ hostname -I
192.168.2.43 172.17.0.1
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango $ sudo nano /home/pi/pidjango/pidjango/settings.py
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango $ sudo nano /var/log/apache2/error.log
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango $ rm -rf venv
virtualenv -p /usr/bin/python3 venv
source venv/bin/activate
pip install -r requirements.txt
bash: virtualenv: command not found
bash: venv/bin/activate: No such file or directory
ERROR: Could not open requirements file: [Errno 2] No such file or directory: 'requirements.txt'
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango $ sudo pip install -r requirements.txt
ERROR: Could not open requirements file: [Errno 2] No such file or directory: 'requirements.txt'
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango $ ^[[200~rm -rf venv
bash: $'^[[200~rm': command not found
bash: venv/bin/activate: No such file or directory
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango $ virtualenv -p /usr/bin/python3 venv/
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango $ source venv/bin/activate
bash: venv/bin/activate: No such file or directory
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango $ cd
(djenv) nivethagnan@raspberrypi:~/ $ cd
(djenv) nivethagnan@raspberrypi:~ $
```

In the above screenshot the `python3 -m pip install django` command is used to install the needed packages for django. The next commands are used to create the different python files as needed.

The screenshot shows a terminal window on a Raspberry Pi desktop. The terminal output is as follows:

```
Setting up lxplug-ejecter (0.21) ...
  Processing triggers for libvlc-bin:armhf (3.0.17.4-0+deb11u1+rpi1+rpt4) ...
  Processing triggers for libc-bin (2.31-13+rpt2+rpi1+deb11u5) ...
[nivethagnan@raspberrypi: ~]$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
    libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
    libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,218 kB of archives.
After this operation, 6,929 kB of additional disk space will be used.
Get:1 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libapr1 armhf 1.7.0-6+deb11u1 [87.1 kB]
Get:2 http://raspbian.freemirror.org/raspbian bullseye/main armhf libaprutil1 armhf 1.6.1-5 [81.2 kB]
Get:3 http://raspbian.freemirror.org/raspbian bullseye/main armhf libaprutil1-dbd-sqlite3 armhf 1.6.1-5 [17.5 kB]
Get:4 http://raspbian.freemirror.org/raspbian bullseye/main armhf libaprutil1-ldap armhf 1.6.1-5 [16.5 kB]
Get:5 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf liblua5.3-0 armhf 5.3.3-1+b1 [95.0 kB]
Get:6 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf apache2-bin armhf 2.4.54-1-deb11u1 [1,227 kB]
Get:7 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf apache2-data all 2.4.54-1-deb11u1 [260 kB]
Get:8 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf apache2-utils armhf 2.4.54-1-deb11u1 [275 kB]
Get:9 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf apache2-armhf 2.4.54-1-deb11u1 [87.1 kB]
Fetched 2,218 kB in 3s (815 kB/s)
Selecting previously unselected package libapr1:armhf.
(Reading database ... 106684 files and directories currently installed.)
Preparing to unpack .../0-libapr1_1.7.0-6+deb11u1_armhf.deb ...
Unpacking libapr1:armhf (1.7.0-6+deb11u1) ...
Selecting previously unselected package libaprutil1:armhf.
Preparing to unpack .../1-libaprutil1_1.6.1-5_armhf.deb ...
Unpacking libaprutil1:armhf (1.6.1-5) ...
Selecting previously unselected package libaprutil1-dbd-sqlite3:armhf.
```

A tooltip in the bottom right corner provides information about Python encoding issues:

PYTHONHOME env variable. If I command python in CMD console, It kept using python3.4. Whenever I command python in CMD console, it starts showing an error below.
Fatal Python error: Py_Initialize: Unable to get the locale encoding
ImportError: No module named 'encodings'
I searched to figure out my problem. Solution was simple. When you install python3.5, you can custom install and check Add Python to environment variables in Advanced Options.
I just leave here for case that someone have similar issues visit here so that they don't waste time.

The apache2 packages also needed to be installed to use Django

The screenshot shows a terminal window titled "monny - /home/nivethagnan/Desktop/photoResistorTest.py @ 21:5". The terminal session starts with enabling the Apache site:

```
Enabling site 000-default.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib  
/systemd/system/apache2.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.  
service → /lib/systemd/system/apache-htcacheclean.service.  
Processing triggers for man-db (2.9.4-2) ...  
Processing triggers for libc-bin (2.31-13+rpi2+rp11+deb11u5) ...
```

Then it runs the command `hostname -I` to find the IP address:

```
nivethagnan@raspberrypi:~ $ hostname -I  
192.168.2.43 172.17.0.1
```

It then attempts to add a user "pi" to the "www-data" group:

```
nivethagnan@raspberrypi:~ $ ^C  
nivethagnan@raspberrypi:~ $ ^C  
nivethagnan@raspberrypi:~ $ sudo usermod -a -G www-data pi  
usermod: user 'pi' does not exist
```

Next, it installs the Apache2 module "libapache2-mod-wsgi-py3":

```
nivethagnan@raspberrypi:~ $ sudo apt install libapache2-mod-wsgi-py3  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
  libfuse2  
Use 'sudo apt autoremove' to remove it.  
The following NEW packages will be installed:  
  libapache2-mod-wsgi-py3  
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.  
Need to get 90.2 kB of archives.  
After this operation, 265 kB of additional disk space will be used.  
Get:1 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libapache2-mod-wsgi-py3 armhf 4.7.1-3+b1 [90.2 kB]  
Fetched 90.2 kB in 1s (128 kB/s)  
Selecting previously unselected package libapache2-mod-wsgi-py3.  
(Reading database ... 107394 files and directories currently installed.)  
Preparing to unpack .../libapache2-mod-wsgi-py3_4.7.1-3+b1_armhf.deb ...  
Unpacking libapache2-mod-wsgi-py3 (4.7.1-3+b1) ...  
Setting up libapache2-mod-wsgi-py3 (4.7.1-3+b1) ...  
apache2_invoke: Enable module wsgi  
nivethagnan@raspberrypi:~ $ sudo apt install python3 python3-virtualenv python3-pip
```

Finally, it installs Python 3 and its dependencies:

```
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
python3 is already the newest version (3.9.2-3).  
python3-virtualenv is already the newest version (3.9.2-3).
```

A tooltip at the bottom right of the terminal window provides information about the `PYTHONHOME` environment variable.

The `hostname -I` command here in the above screenshot is used to find the IP address to view the server under the correct IP.

The following package was automatically installed and is no longer required:
libfuse2
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
nivethagnan@raspberrypi:~ \$ sudo nano /etc/apache2/sites-enabled/000-default.conf
nivethagnan@raspberrypi:~ \$ sudo systemctl restart apache2
nivethagnan@raspberrypi:~ \$ mkdir -p /home/pi/pidjango/static
mkdir: cannot create directory '/home/pi': Permission denied
nivethagnan@raspberrypi:~ \$ sudo mkdir -p /home/pi/pidjango/static
nivethagnan@raspberrypi:~ \$ cd /home/pi/pidjango
nivethagnan@raspberrypi:/home/pi/pidjango \$ python3 -m venv djenv
nivethagnan@raspberrypi:/home/pi/pidjango \$ sudo python3 -m venv djenv
Error: [Errno 13] Permission denied: '/home/pi/pidjango/djenv'
nivethagnan@raspberrypi:/home/pi/pidjango \$ source djenv/bin/activate
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango \$ python3 -m pip install django
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting django
 Downloading https://www.piwheels.org/simple/django/Django-4.1.3-py3-none-any.whl (8.1 MB)
 8.1 MB 48 kB/s
Collecting sqlparse>=0.2.2
 Downloading https://www.piwheels.org/simple/sqlparse/sqlparse-0.4.3-py3-none-any.whl (42 kB)
 42 kB 181 kB/s
Collecting asgiref<4,>=3.5.2
 Downloading https://www.piwheels.org/simple/asgiref/asgiref-3.5.2-py3-none-any.whl (22 kB)
Installing collected packages: sqlparse, asgiref, django
ERROR: Could not install packages due to an EnvironmentError: [Errno 13] Permission denied: '/home/pi/pidjango/djenv/lib/python3.9/site-packages'
Consider using the '--user' option or check the permissions.
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango \$ sudo python3 -m pip install django
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting django
 Downloading https://www.piwheels.org/simple/django/Django-4.1.3-py3-none-any.whl (8.1 MB)
 8.1 MB 57 kB/s
Collecting sqlparse>=0.2.2
 Downloading https://www.piwheels.org/simple/sqlparse/sqlparse-0.4.3-py3-none-any.whl (42 kB)
 42 kB 169 kB/s
Collecting asgiref<4,>=3.5.2
 Downloading https://www.piwheels.org/simple/asgiref/asgiref-3.5.2-py3-none-any.whl (22 kB)
PYTHONDHOME env variable, if I command python in CMD console, it keeps using deleted python3.4. Whenever I command python in CMD console, it starts showing below.
Python error: Py_Initialize: Unable to get the locale encoding
No module named 'encodings'
I

The commands here are used to install the needed packages and python files for Django.

Consider using the '--user' option or check the permissions.
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango \$ sudo python3 -m pip install django
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting django
 Downloading https://www.piwheels.org/simple/django/Django-4.1.3-py3-none-any.whl (8.1 MB)
 8.1 MB 57 kB/s
Collecting sqlparse>=0.2.2
 Downloading https://www.piwheels.org/simple/sqlparse/sqlparse-0.4.3-py3-none-any.whl (42 kB)
 42 kB 169 kB/s
Collecting asgiref<4,>=3.5.2
 Downloading https://www.piwheels.org/simple/asgiref/asgiref-3.5.2-py3-none-any.whl (22 kB)
Installing collected packages: sqlparse, asgiref, django
Successfully installed asgiref-3.5.2 django-4.1.3 sqlparse-0.4.3
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango \$ sudo django-admin startproject pidjango .
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango \$ hostname -I
192.168.2.43 172.17.0.1
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango \$ sudo nano /home/pi/pidjango/pidjango/settings.py
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango \$ sudo nano /home/pi/pidjango/pidjango/settings.py
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango \$ sudo nano /home/pi/pidjango/pidjango/settings.py
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango \$ sudo nano /var/log/apache2/error.log
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango \$ rm -rf venv
(djenv) nivethagnan@raspberrypi:/home/pi/pidjango \$ virtualenv -p /usr/bin/python3 venv/
virtualenv -p /usr/bin/python3 venv/
source venv/bin/activate
pip install -r requirements.txt
pip: error: No such file or directory: 'requirements.txt'
No such file or directory: 'requirements.txt'

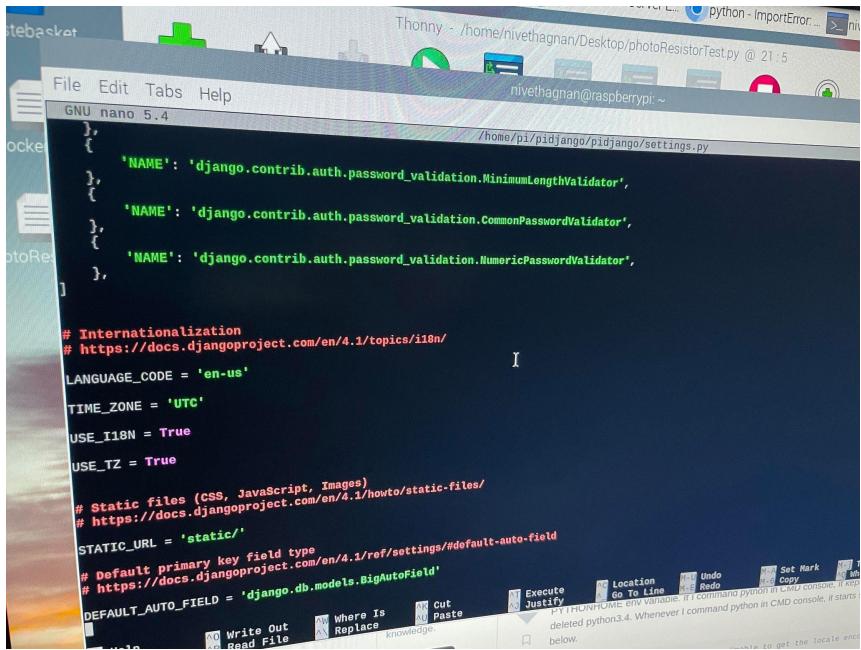
The virtual environment created within the raspberry pi is used to download the python files in the correct virtual environment.

```
Django settings for pidjango project.  
Generated by 'django-admin startproject' using Django 4.1.3.  
For more information on this file, see  
https://docs.djangoproject.com/en/4.1/topics/settings/  
For the full list of settings and their values, see  
https://docs.djangoproject.com/en/4.1/ref/settings/  
  
from pathlib import Path  
  
# Build paths inside the project like this: BASE_DIR / 'subdir'.  
BASE_DIR = Path(__file__).resolve().parent.parent  
  
# Quick-start development settings - unsuitable for production  
# See https://docs.djangoproject.com/en/4.1/howto/deployment/checklist/  
  
# SECURITY WARNING: keep the secret key used in production secret!  
SECRET_KEY = 'django-insecure-xg8c_!m-7v-$x9&wiu^7!2h(c1np757om2l@0s#bm+g3vo9tg='  
# SECURITY WARNING: don't run with debug turned on in production!  
DEBUG = True  
ALLOWED_HOSTS = ["192.168.2.43"]  
  
# Application definition
```

The script is edited for the ALLOWED_HOSTS part.

```
File Edit Tabs Help  
GNU nano 5.4  
[...]  
'django.contrib.sessions',  
'django.contrib.messages',  
'django.contrib.staticfiles',  
]  
  
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
]  
ROOT_URLCONF = 'pidjango.urls'  
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    },  
]  
WSGI_APPLICATION = 'pidjango.wsgi.application'  
  
nivethagnan@raspberrypi: ~
```

This is the full unedited script.



The screenshot shows a Raspberry Pi desktop environment with several windows open. In the foreground, a terminal window titled 'GNU nano 5.4' displays the contents of a Django settings file. The code includes sections for password validation, internationalization, static files, and database configuration. In the background, there are icons for Thonny (Python IDE), a file browser, and other system applications. The desktop has a blue and white theme with a sunset wallpaper.

```
File Edit Tabs Help /home/pi/pidjango/pidjango/settings.py niyethagnan@raspberrypi: ~
[{"NAME": "django.contrib.auth.password_validation.MinimumLengthValidator"}, {"NAME": "django.contrib.auth.password_validation.CommonPasswordValidator"}, {"NAME": "django.contrib.auth.password_validation.NumericPasswordValidator"}]

# Internationalization
# https://docs.djangoproject.com/en/4.1/topics/i18n/
LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.1/howto/static-files/
STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/4.1/ref/settings/#default-auto-field
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Github repo: <https://github.com/MunazzaF/Assignment3IoT>

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, POST, HEAD, OPTIONS

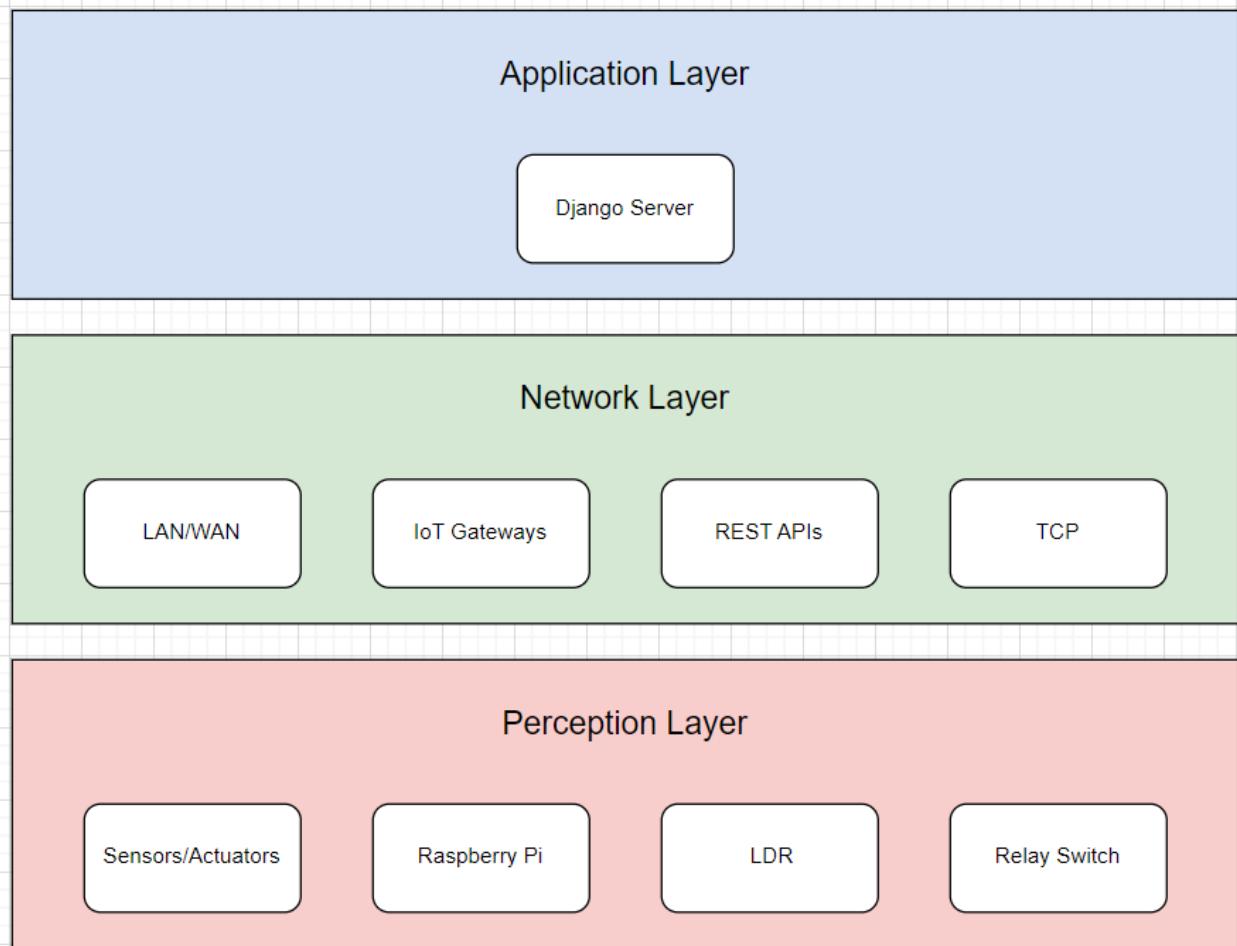
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://localhost:8000/state/1/",
            "name": "on"
        }
    ]
}
```

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, PUT, DELETE, HEAD, OPTIONS, PATCH

{
    "url": "http://localhost:8000/mode/1/",
    "name": "manual"
}
```

This shows the front end Django for the Auto and Light.





The above screenshot shows the architecture design of the system. It shows each layer of the system and which components were within their respective layer. In the application layer we have the Django server. The Django server is used for the server-side web framework which is written in python scripts. In the network layer we have four different components, the LAN/WAN, IoT Gateways, REST APIs and TCP. The network layer has all the components that are required for packet forwarding and routing. The last layer which is the perception layer has four components, which includes the sensors and actuators, the Raspberry Pi, the LDR, and the Relay Switch. The Raspberry Pi was used as the sensor device, and the LDR which is the photoresistor sensor was used as the light intensity detector.