



Design & Analysis of IoT Software Systems - SOFE4610U

Assignment 3

CRN: 44432

Group Number 8

Group Member 1

Name: Alden O'Cain

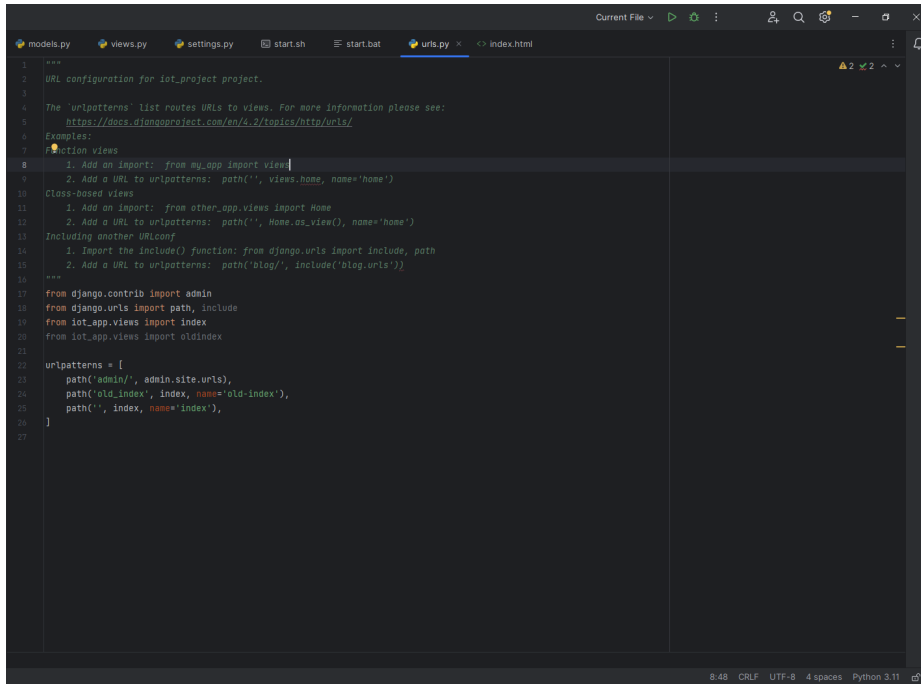
Student ID: 100558599

Group Member 2

Name: Liam Rea

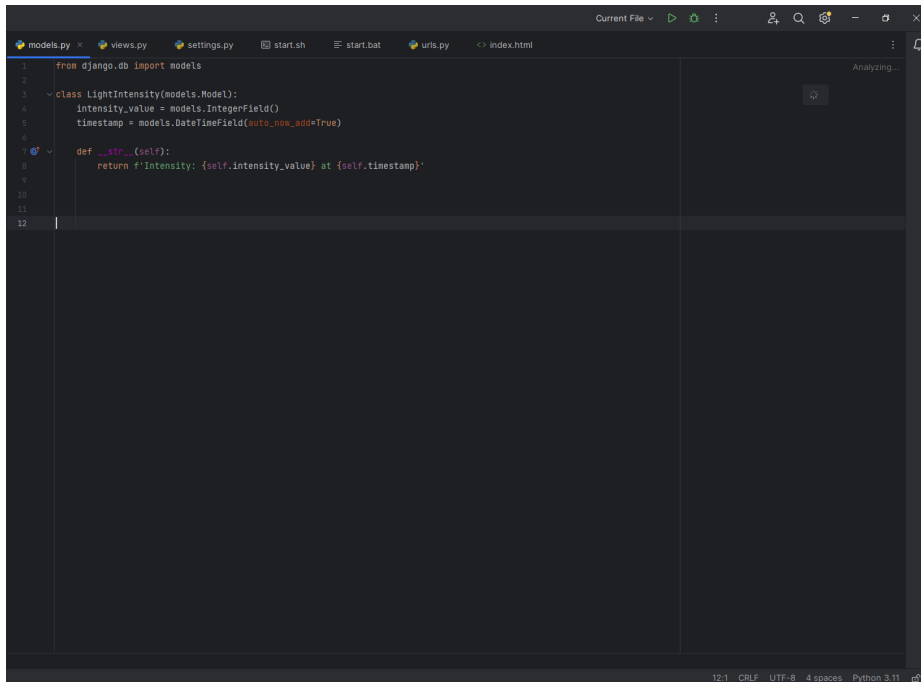
Student ID: 100743012

Project Repository



```
1 """
2 URL configuration for iot_project project.
3
4 The 'urlpatterns' list routes URLs to views. For more information please see:
5     https://docs.djangoproject.com/en/4.2/topics/http/urls/
6 Examples:
7 # Action views
8 1. Add an import: from my_app import views
9 2. Add a URL to urlpatterns: path('', views.home, name='home')
10 # Class-based views
11 1. Add an import: from other_app.views import Home
12 2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
13 Including another URLconf
14 1. Import the include() function: from django.urls import include, path
15 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import path, include
19 from iot_app.views import index
20 from iot_app.views import oldindex
21
22 urlpatterns = [
23     path('admin/', admin.site.urls),
24     path('old_index', index, name='old-index'),
25     path('', index, name='index'),
26 ]
27
```

Defining URL routes in the Django app



```
1 from django.db import models
2
3 class LightIntensity(models.Model):
4     intensity_value = models.IntegerField()
5     timestamp = models.DateTimeField(auto_now_add=True)
6
7     def __str__(self):
8         return f'Intensity: {self.intensity_value} at {self.timestamp}'
9
10
11
12
```

Creating a model for light intensity data

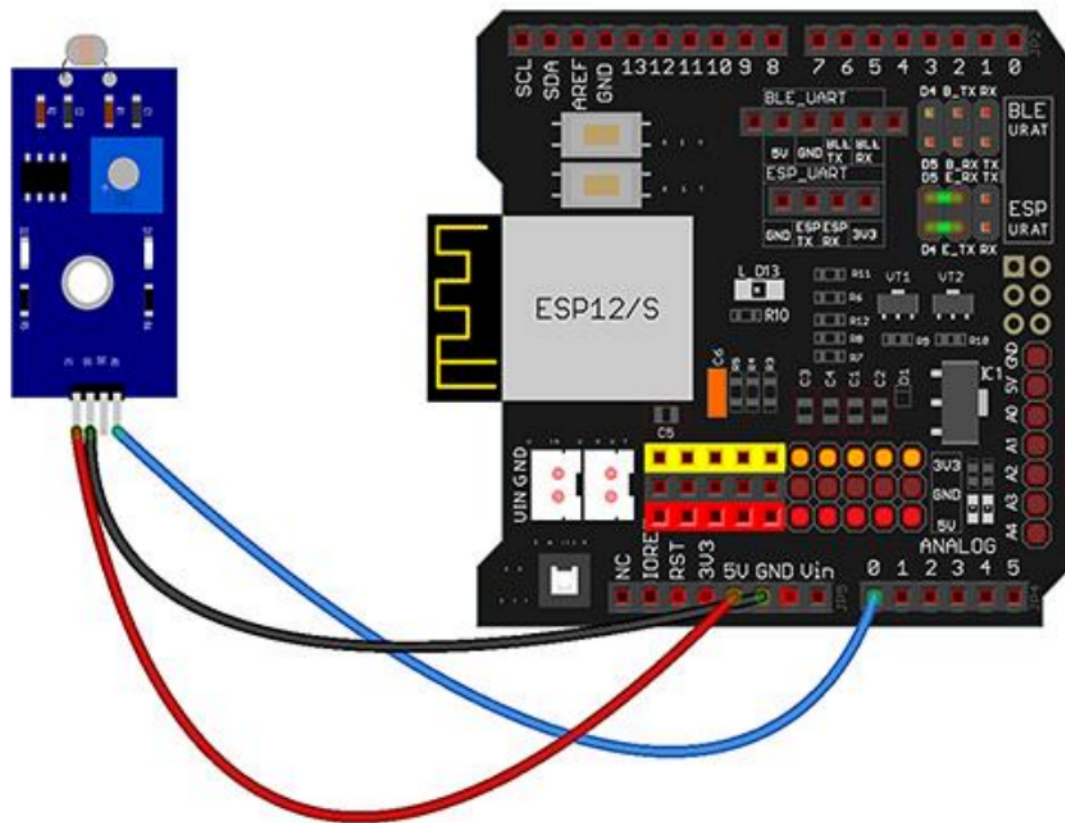
```
models.py  views.py  settings.py  start.sh  start.bat  urls.py  < index.html
7
8
9 intensity_value = 42
10 light_intensity = LightIntensity.objects.create(intensity_value=intensity_value)
11 light_intensity.save()
12
13 from django.http import HttpResponse
14 from django.views.decorators.csrf import csrf_exempt
15 from .models import LightIntensity # Import your LightIntensity model
16
17
18 @usage
19 def index(request):
20     ser = serial.Serial()
21     ser.port = "/dev/rfcomm0"
22     ser.baudrate = 115200
23     ser.timeout = 1
24     ser.setDTR(False)
25     ser.setRTS(False)
26     ser.open()
27     raw_data = 0
28     ser.flushInput()
29     ser.flush()
30     ser.flushOutput()
31     time.sleep(1)
32     raw_data = ser.readline()
33     try:
34         ser.flushInput()
35         raw_data = int(raw_data)
36         ser.flush()
37     except ValueError:
38         pass
39     latest_intensity = raw_data
40     # latest_intensity = LightIntensity.objects.last()
41     return render(request, template_name='index.html', context={'latest_intensity': latest_intensity if latest_intensity else 'N/A'})
42
43
44 @usage
45 def oldindex(request):
46     latest_intensity = LightIntensity.objects.last()
47     return render(request, template_name='index.html', context={'latest_intensity': latest_intensity.intensity_value if latest_intensity else 'N/A'})
48
```

Creating a view for the index page (oldIndex displays a static value)

Light Intensity Display

Latest Light Intensity: 42

Old index displays a static value to test the HTML template



Wiring the photoresistor sensor to the Arduino WiFi shield

```
IOT_lab2.ino
1 #include "SoftwareSerial.h"
2
3 SoftwareSerial softserial(4, 5); // RX, TX
4
5 const int analogInPin = A0; // photoresistor pin
6
7 void setup() {
8   Serial.begin(9600);
9 }
10
11 void loop() {
12   // Read the analog value from the photoresistor
13   int sensorValue = analogRead(analogInPin);
14
15   // Print the sensor value to the Serial Monitor
16   Serial.println("Sensor Value: " + String(sensorValue));
17
18   // Optionally, add a delay to slow down the serial output
19   delay(1000); // Adjust the delay time according to your needs
20 }
21
```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Uno' on 'COM3')

Sensor Value: 1024
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1017
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1022
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1023
Sensor Value: 1023

Ln 21, Col 1 Arduino Uno on COM3

Testing reading the sensor and printing to the Serial Monitor

```
IOT_lab2.ino
1 #include <SoftwareSerial.h>
2
3 #define DHTPIN A0
4
5 SoftwareSerial btSerial(4, 5); // TX, RX
6
7 void setup() {
8   btSerial.begin(9600);
9   Serial.begin(9600);
10  btSerial.flush();
11  Serial.flush();
12 }
13
14 void loop() {
15   int sensorValue = analogRead(A0);
16   btSerial.println(sensorValue);
17   delay(10000); // Send data every 10 seconds
18 }
19
```


Output Serial Monitor x

Not connected. Select a board and a port to connect automatically.

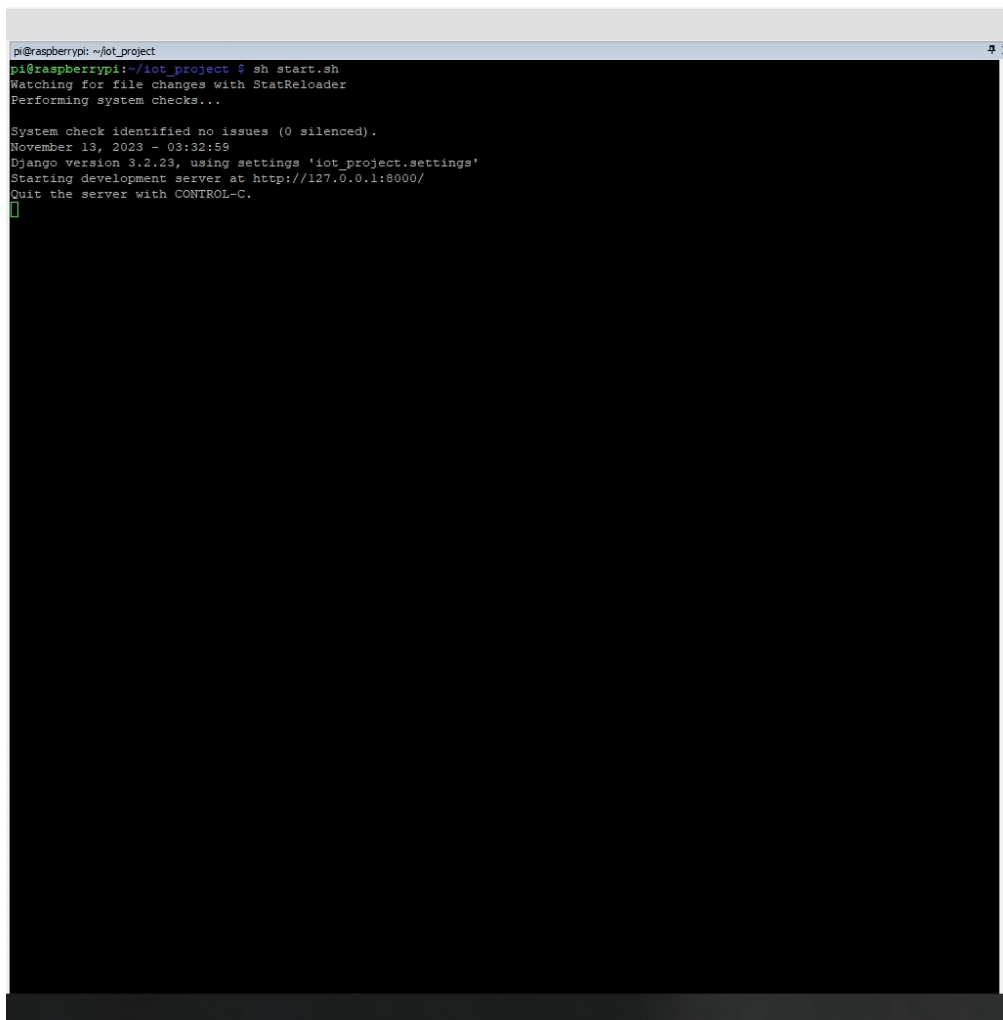
Ln 15, Col 37 Arduino Uno on COM3 [not connected]

Updating the Arduino sketch to send the sensor value through bluetooth

```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ python HCTest.py  
Start  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023  
  
1023
```



Testing capturing the photoresistor sensor values using a simple python script

A terminal window titled 'pi@raspberrypi: ~/iot_project' with standard window controls. The terminal shows the execution of 'sh start.sh', which initiates a Django development server. The output includes system checks, the date and time, the Django version (3.2.23), and the server URL (http://127.0.0.1:8000/). A green cursor is visible on the line following the instructions.

```
pi@raspberrypi: ~/iot_project
pi@raspberrypi:~/iot_project $ sh start.sh
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
November 13, 2023 - 03:32:59
Django version 3.2.23, using settings 'iot_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
█
```

Running the updated Django server on the Raspberry Pi



```
pi@raspberrypi: ~/docker-example-python-web-server X
pi@raspberrypi:~/docker-example-python-web-server $ curl http://localhost:8000
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Light Intensity Display</title>
</head>
<body>
  <h1>Light Intensity Display</h1>
  <p>Latest Light Intensity: b6#x27;1023\r\n&#x27;</p>
</body>
</html>
pi@raspberrypi:~/docker-example-python-web-server $
```

Testing the output of the Django web server locally on the Raspberry pi using curl after incorporating reading the bluetooth data inside the Django view

Light Intensity Display

Latest Light Intensity: 1023

Django server displaying the correct values to another computer on the network after changing the Django settings to bind to the private IP address of the Raspberry Pi