| **Job Sheet** |
| --- |

| **Module** | **:** | IoT Fundamentals | **Module Code** | **:** | EC33105FP | |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | |
| **Job No** | **:** | 1 | **Duration** | **:** | F/T: | 3 hours |
|  |  |  |  |  | P/T: |  |

| **Job Title** | **:** | Configure IoT Device |
| --- | --- | --- |
|  |  |  |
| **Objectives** | **:** | 1. Setup IoT Device 2. Configure Remote Access |

**Tools, Equipment and Materials**

| Raspberry Pi Kit (with Rpi v4) |
| --- |
| PC |
| Monitor with HDMI port |
| USB Keyboard |
| USB mouse |
| Micro HDMI to HDMI cable  4G WiFi router |
|  |

**Number of Tasks to Complete**: 4

**Part A: Setup IoT Device (1 hour)**

While IoT devices are often **low-powered devices** with a microcontroller as the core processor and a Wi-Fi module as the communications interface, they don’t have to be. For instance, a mainstream PC can be used as an IoT device, but doing so would prove to be difficult as PCs require mains power of many hundreds of watts of power. However, the Raspberry Pi, a computer far more powerful than many IoT devices, is a good candidate for IoT processing!

The **Raspberry Pi (RPi)** range of systems is incredibly small, being only slightly larger than a credit card and uses very little energy. They have some decent processing capabilities with some Pi computers (such as the Rpi v4) having a 64-bit quad core. On top of this, they have a choice of varying RAM sizes of 1GB, 2GB, 4GB, integrated Wi-Fi, and plenty of GPIO for interfacing with external hardware. But the real icing on top is that they can be programmed wirelessly and in easy-to-use languages like Python, which include a wide range of libraries ranging from face detection to speech-to-text.

**Diagram, engineering drawing

Description automatically generated**

Sources:

<https://www.digikey.com/en/maker/blogs/2019/how-to-set-up-a-raspberry-pi-as-an-iot-device>, <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>

**Task 1 – Setup RPi**

**Perform the following steps to set up the RPi.**

| 1. | Ensure the microSD card is seated well. Video steps below: |
| --- | --- |
| 2. | Connect keyboard and mouse to RPi: |
| 3. | Connect Monitor to RPi |
| 4. | Connect Power Adaptor to RPi |
| 5. | Turn on RPi |
| 6. | Configure RPi |
| 7 | Reboot the RPi and ensure the date and time are updated correctly |
| 8 | Show your lecturer the configured RPi. |

**Part B: Configure Remote Access (2 hours)**

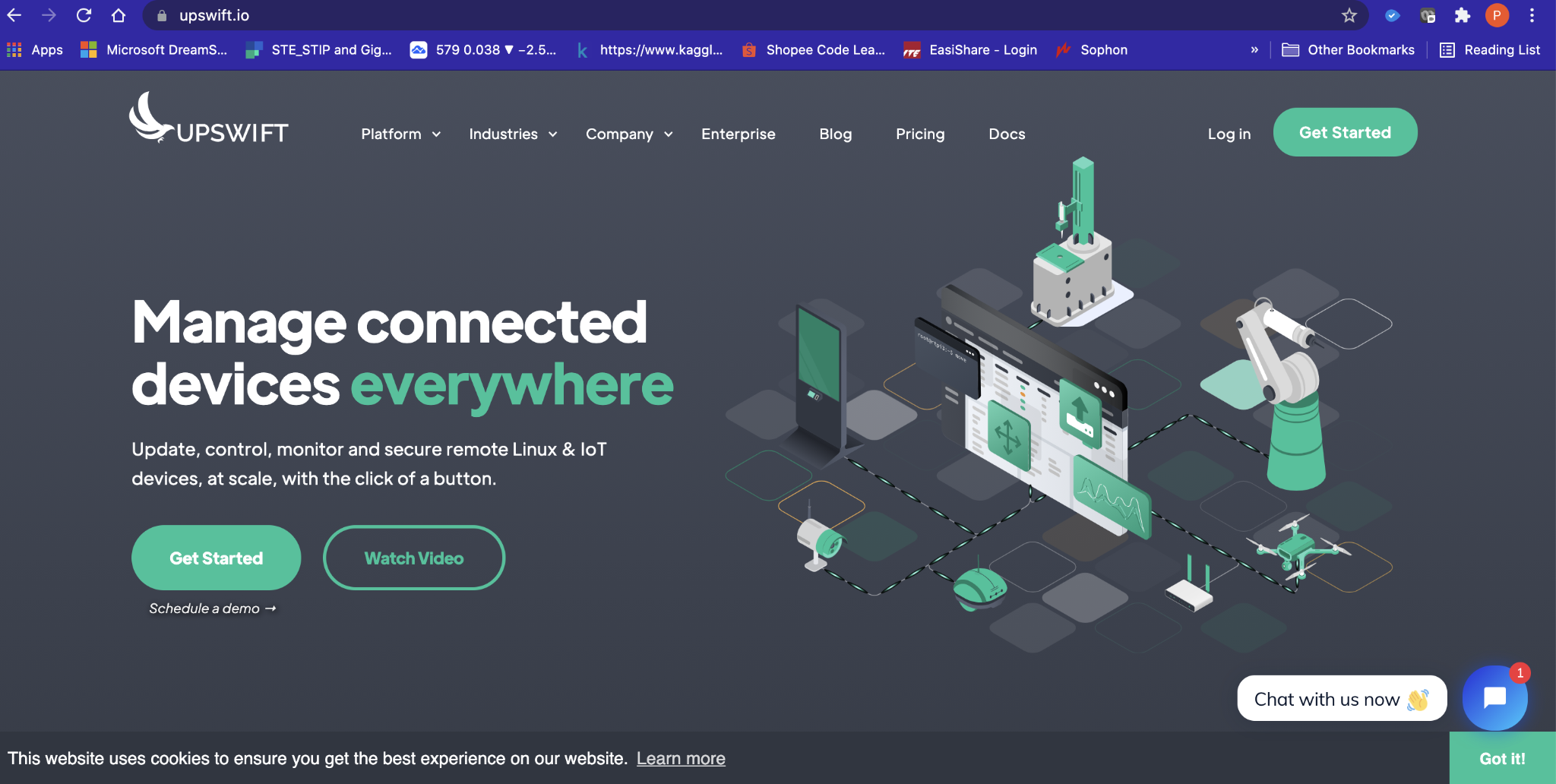
Remote access of devices allows IoT developers to be able to update, monitor, control, and manage their devices from **anywhere in the world, with an internet connection.**

Accessing the RPi remotely would require the device to be registered in an IoT platform. The IoT platform we are using in this module is Upswift. Upswift offers FREE accounts for hobbyists to connect up to 3 IoT devices. We will be

* Creating a FREE hobbyist account
* Registering the RPi
* Remotely accessing the RPi

Upswift makes connecting a device super simple, with less than 60 seconds. Their solutions are being used in over 42 countries as of 15 July 2021\*.

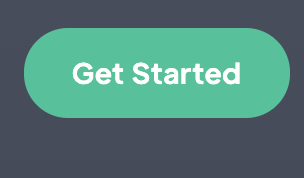
\*<https://www.upswift.io/pricing>



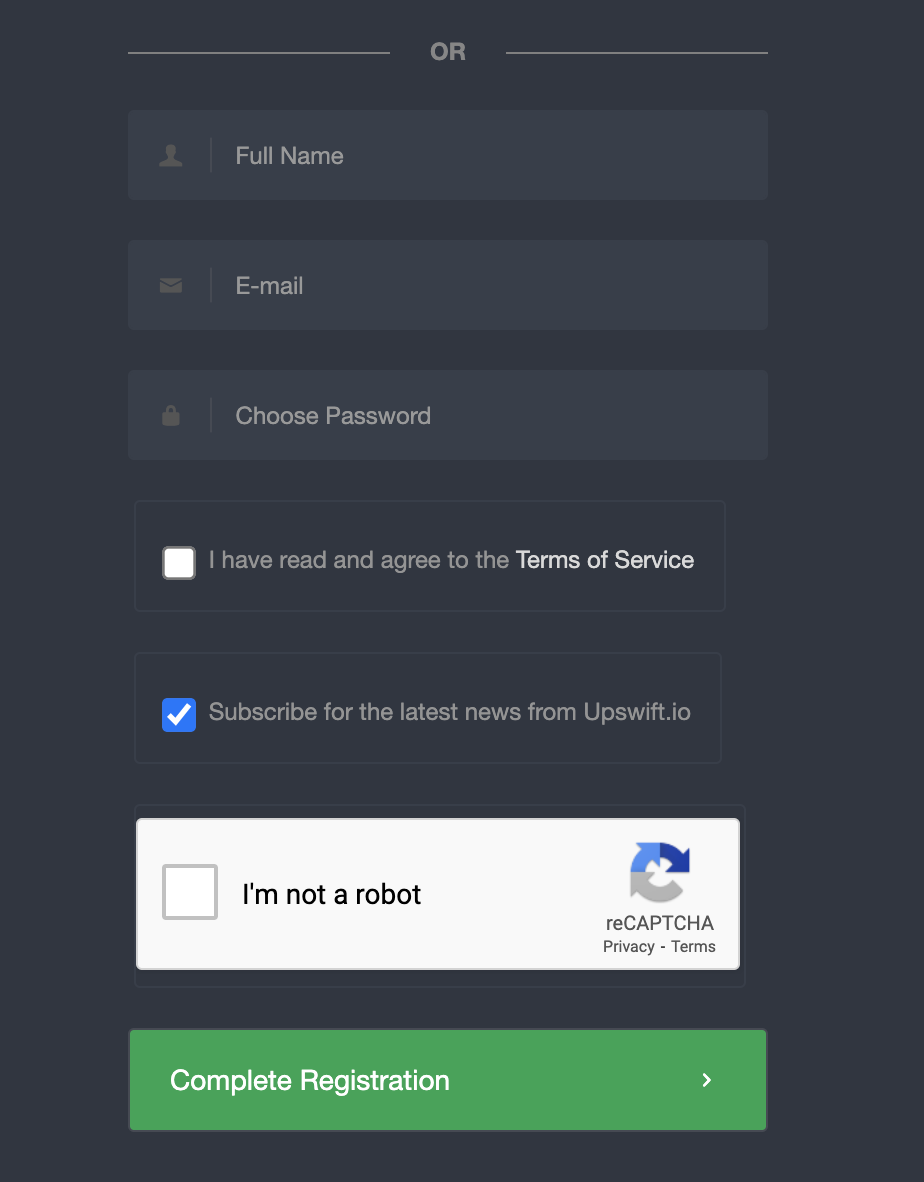
**Task 2 – Register for Upswift**

Perform the following steps to register for an Upswift account:

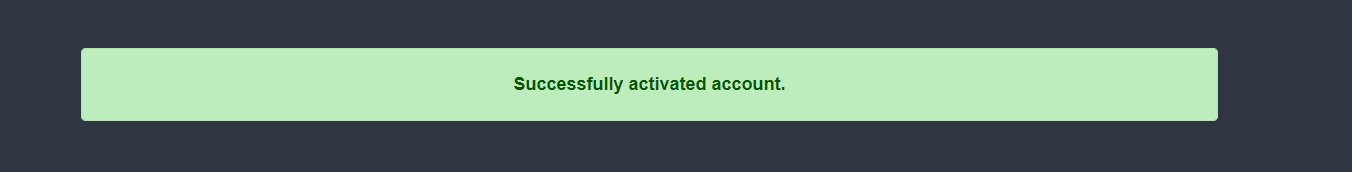
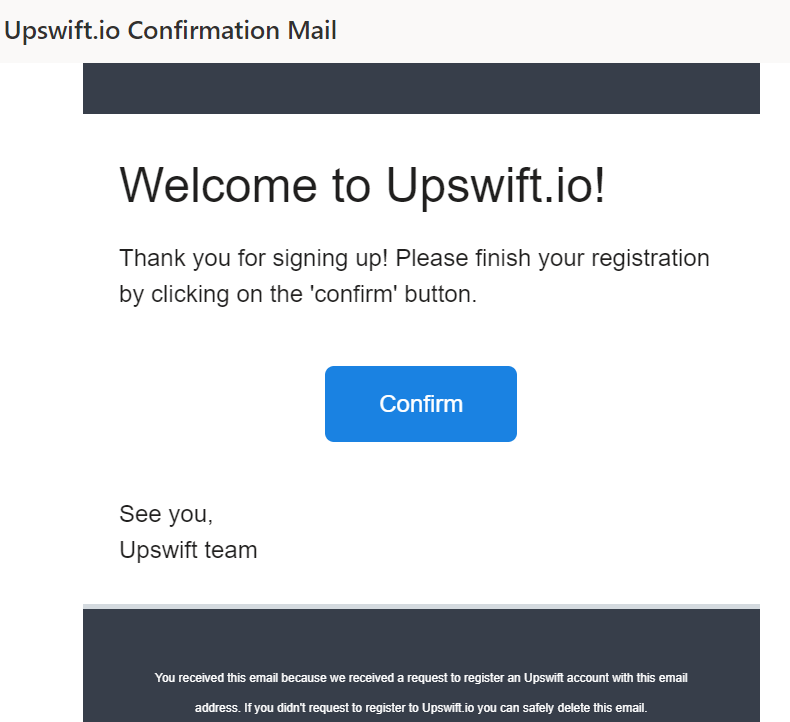
1. Go to <https://www.upswift.io/>
2. Click on the Get Started button on the top right.



1. Register for an account.



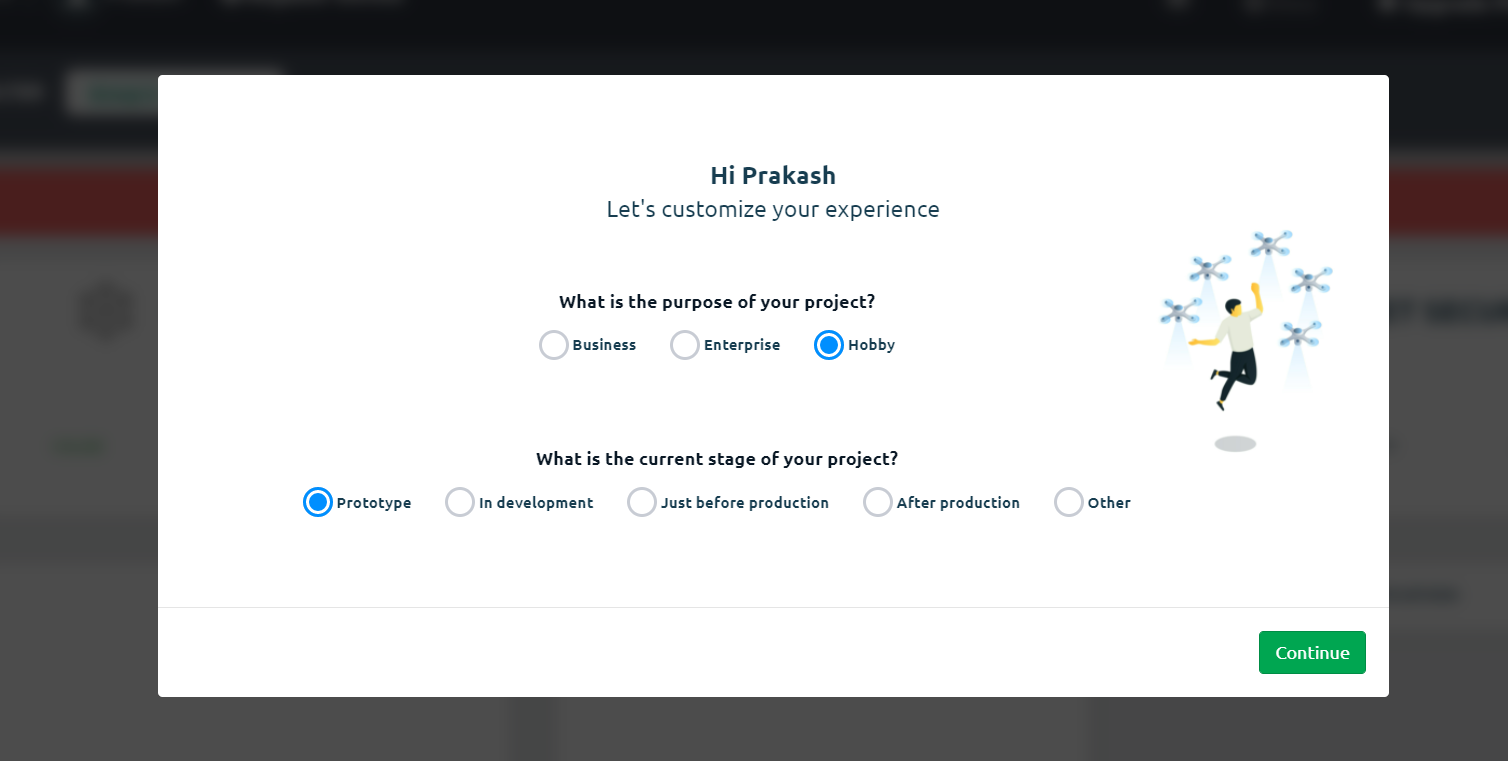
1. A confirmation email will be sent to your email account. Logon to your email and click on the confirm button.



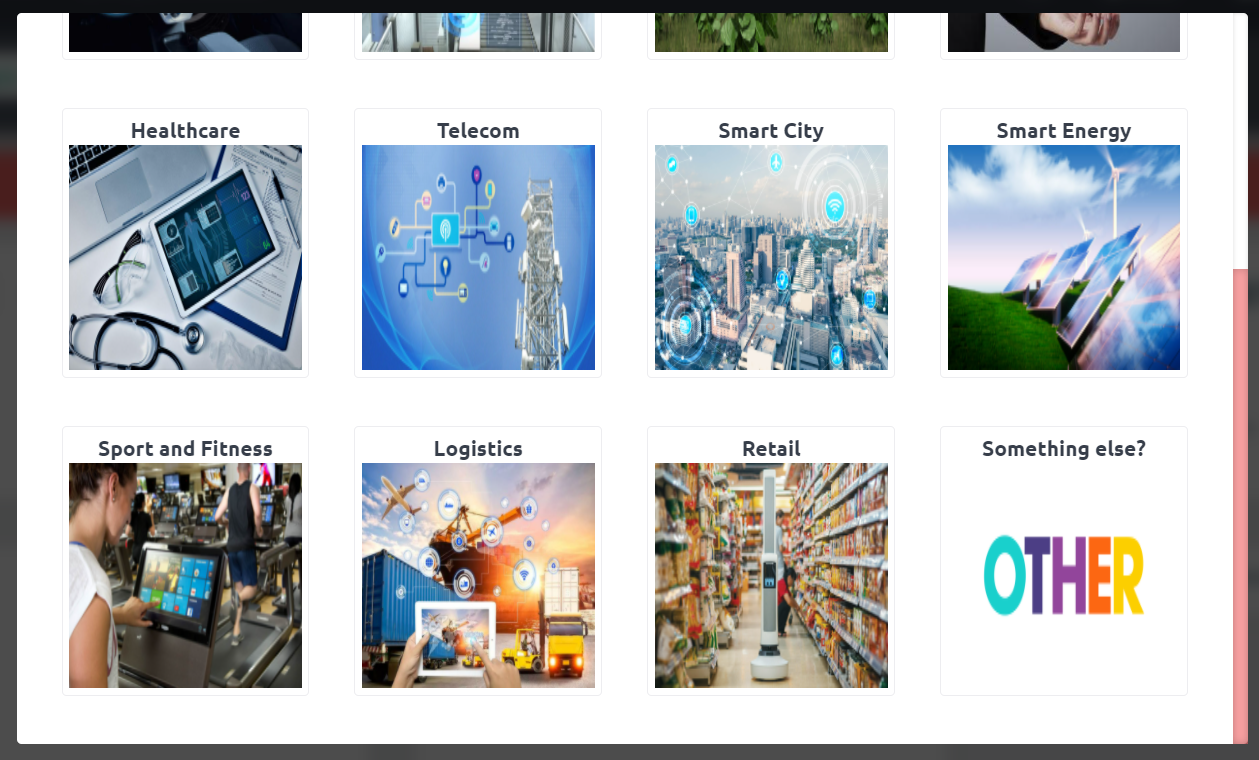
1. Log in to your account.
2. You will be presented with some questions. For question 1

Purpose of Project : **Hobby**

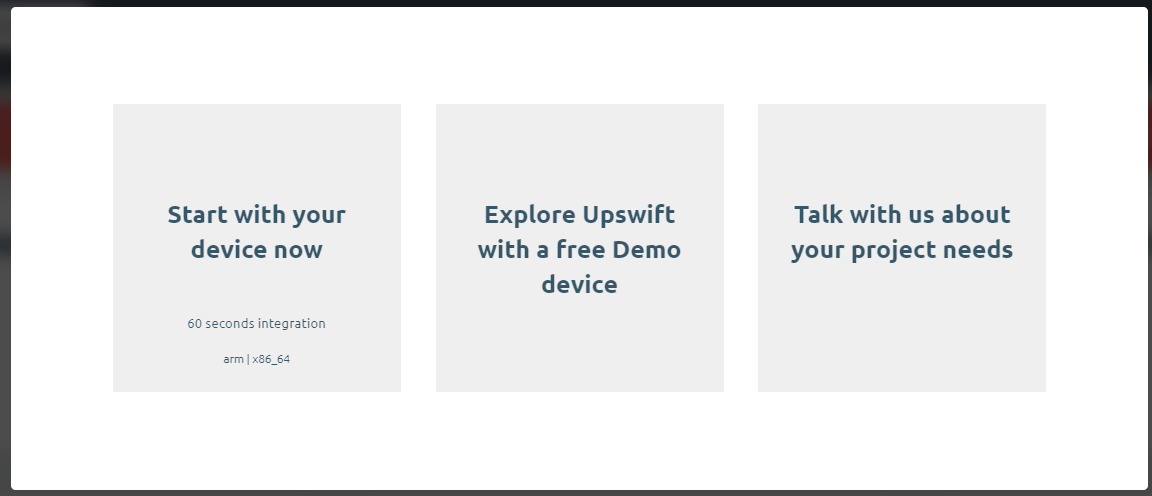
Current Stage of Project: **Prototype**



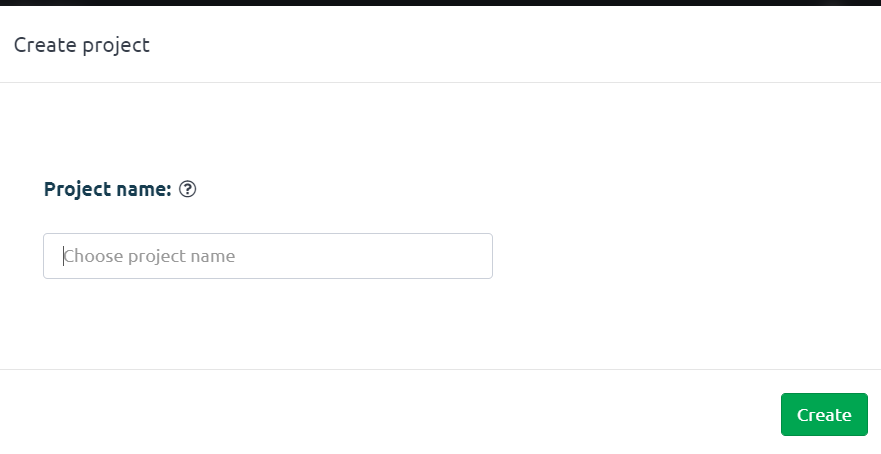
1. For question 2, select **other**.



1. Select **Start with your device now.**



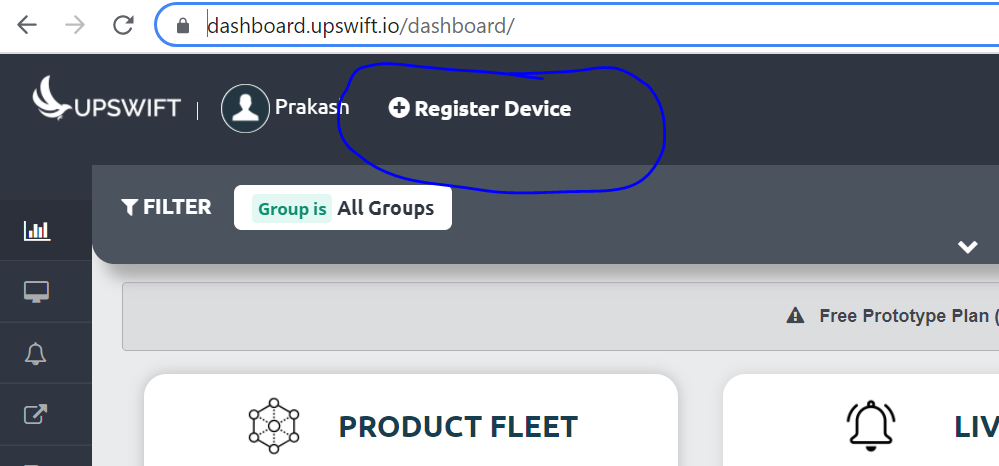
1. Give your project a name. I am using **Test. Click Create.**

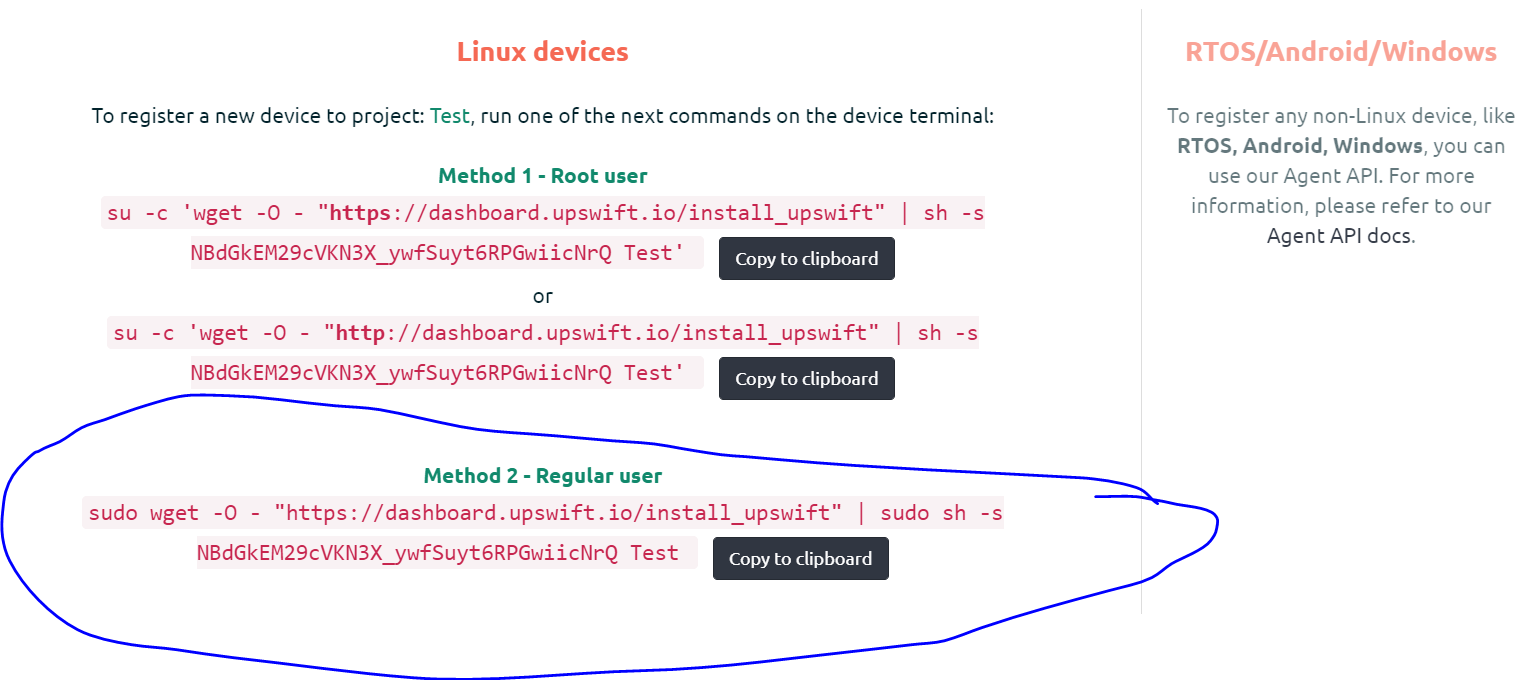


**Task 3 – Register Device**

1. Go to Upswift Dashboard on the RPi browser. Watch the video for steps.

<https://youtu.be/LmZppZQUCLw>

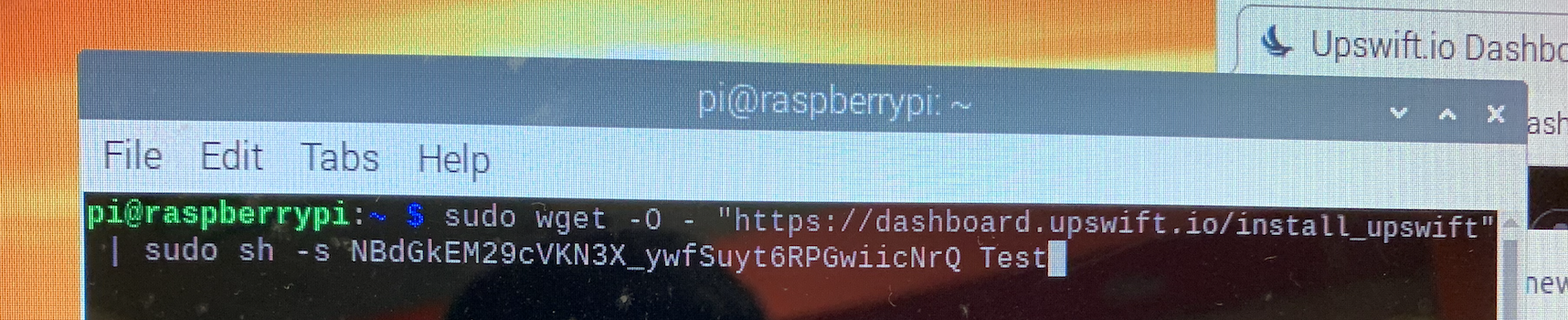
1. Click on register device
2. 
3. Click on **Copy to Clipboard** button for **Method 2 - Regular User**. This will copy a linux command.



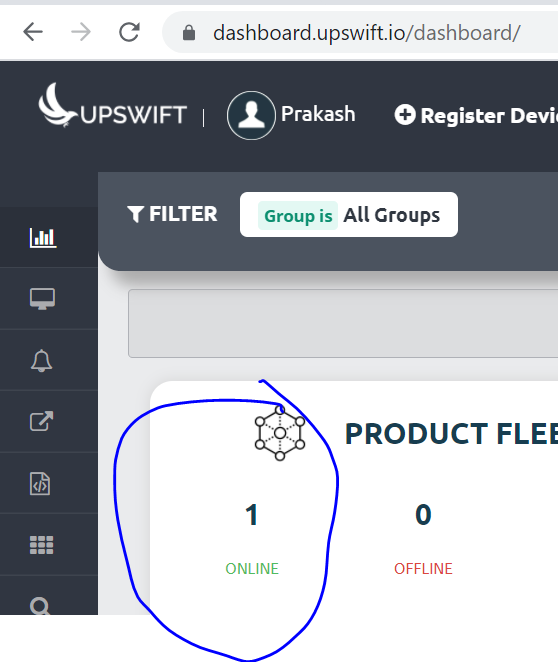
1. On the RPi, open a new terminal. Watch video for steps

<https://youtu.be/RMxIRpCJCto>

1. Right click and paste the copied linux command onto the terminal. Hit enter to execute the command



1. If the device is added successfully, you’ll see it appear in the dashboard.

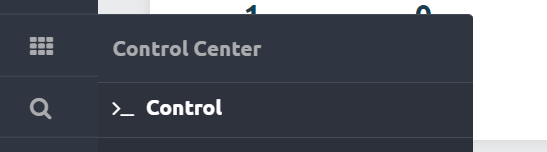


**Task 4 – Run Command Remotely**

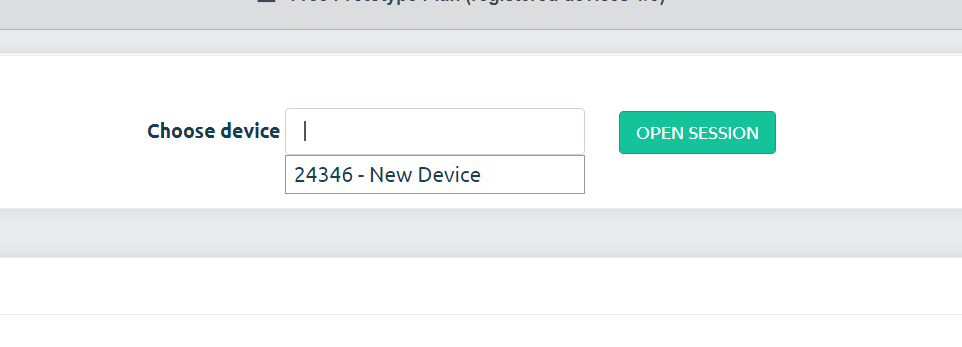
We are going to access the RPi’s terminal and execute a simple command on the terminal remotely.This will allow us to program the RPi from anywhere using the upswift account with a browser.

USE YOUR PC FOR THIS TASK (NOT RPi, so as to access remotely)

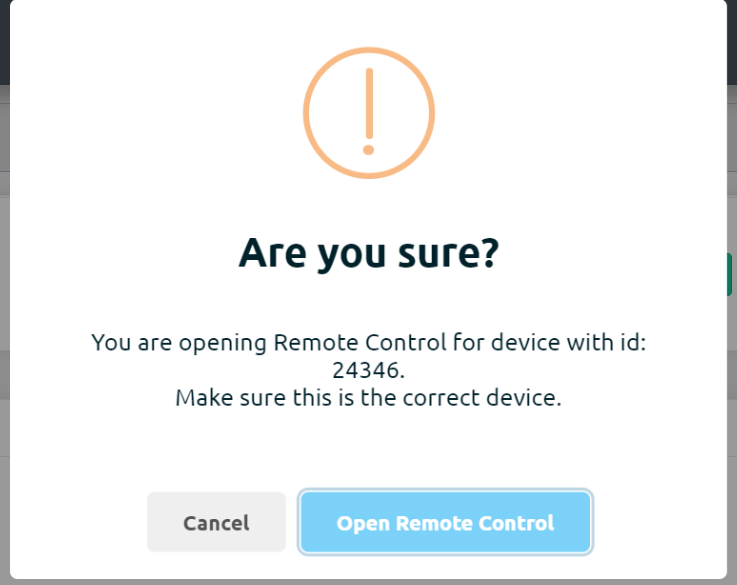
1. Go to <https://dashboard.upswift.io/dashboard/> on your PC (NOT the RPi monitor!) and login.
2. On the left side of your menu, click on the **Control** option under Control center.



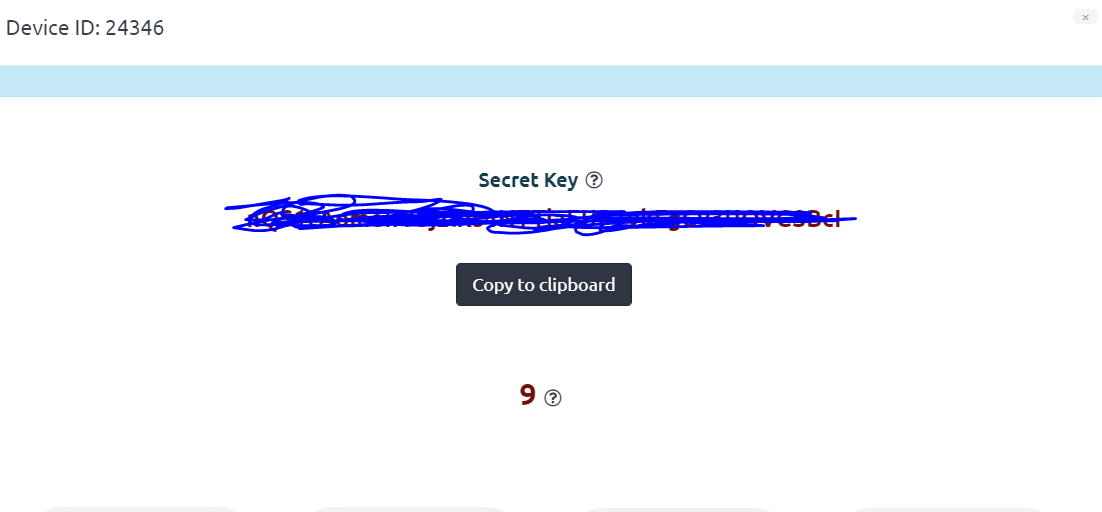
1. On the page center, select your device and click open session.



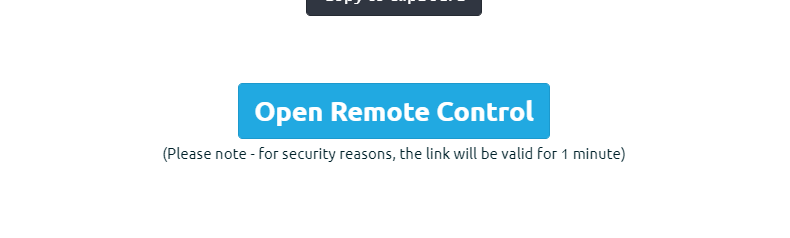
1. For the prompt, click **Open Remote Control.**



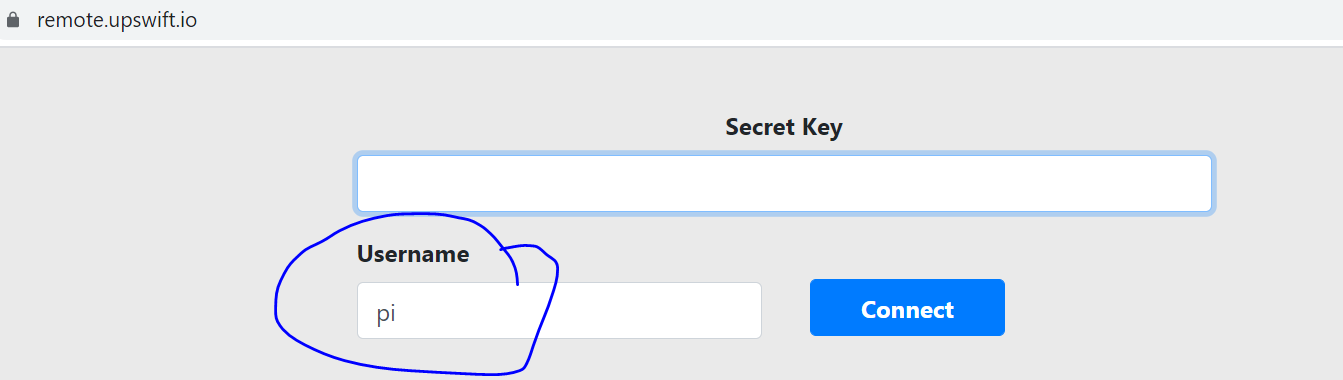
1. You will be shown a secret key. Click on **Copy to clipboard** button

****

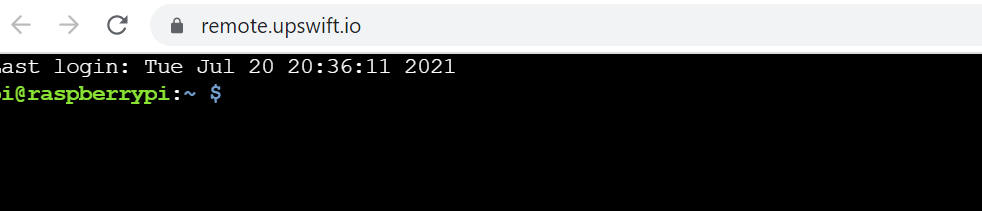
1. Once the timer is up, click on the **Open Remote Control** button.

****

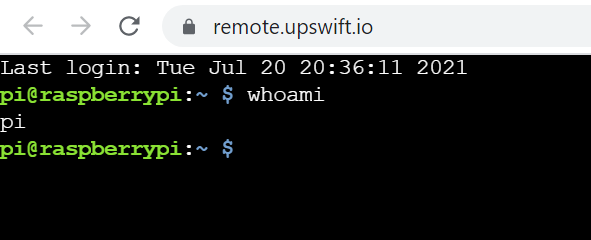
1. **Paste** the secret key in the input text box and change username to pi. Click on **Connect**.

****

1. You will be presented with a browser based RPi terminal.

****

1. Test the terminal by running a **whoami** command. The output should be as such:



1. Show your lecturer your configured RPi.