**Part 1: Accessing the assignment and obtaining the starter code**

Since this is a *group assignment*, cloning the starter code is slightly different. Please follow the below instructions carefully.

* Go to
  + If one of your group members has already created a team, select the name of your team from the list to join that team.
  + If you are the first person from your group to accept the assignment, you must create a new team. Enter the name of your team in the text field and click on the “create team” button.
  + Make sure that you are creating/joining the correct team, since you may not be able to undo these steps.
* Once you have created or joined a team, you will be given access to a cloned repo containing the starter code required for this assignment.
* Your entire group now shares a repository that any of you can edit. Make sure to commit and push changes carefully so that you do not overwrite each other’s code.

**Part 2: Sensor logger app for live data collection**

We'll be using Sensorlogger (https://www.tszheichoi.com/sensorlogger) app for streaming:

For iOS: https://apps.apple.com/app/id1531582925

For Android: https://play.google.com/store/apps/details?id=com.kelvin.sensorapp&pcampaignid=pcampaignidMKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1

The above link is also available on the app's website.

**Part 3: Step Detection**

1. Connect your laptop to your phone hotspot.
2. Reopen the app and navigate to the “Logger” tab.
3. Tap on the radio button beside the word “Accelerometer”.

**IMPORTANT: If you are unable to install and get the app running, let us know through Piazza.**

1. Once your app is up and running, find and open the python file “sensor\_logger.py” from the starter repository in any code editor of your choice
2. When you run the script in terminal python3 sensor\_logger.py you will see something like this

You will see something like these on the command prompt:

```

Dash is running on http://0.0.0.0:8000/

\* Serving Flask app 'sensor\_logger'

\* Debug mode: off

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

\* Running on all addresses (0.0.0.0)

\* Running on http://127.0.0.1:8000

\* Running on http://192.168.212.126:8000

Press CTRL+C to quit

```

The line:

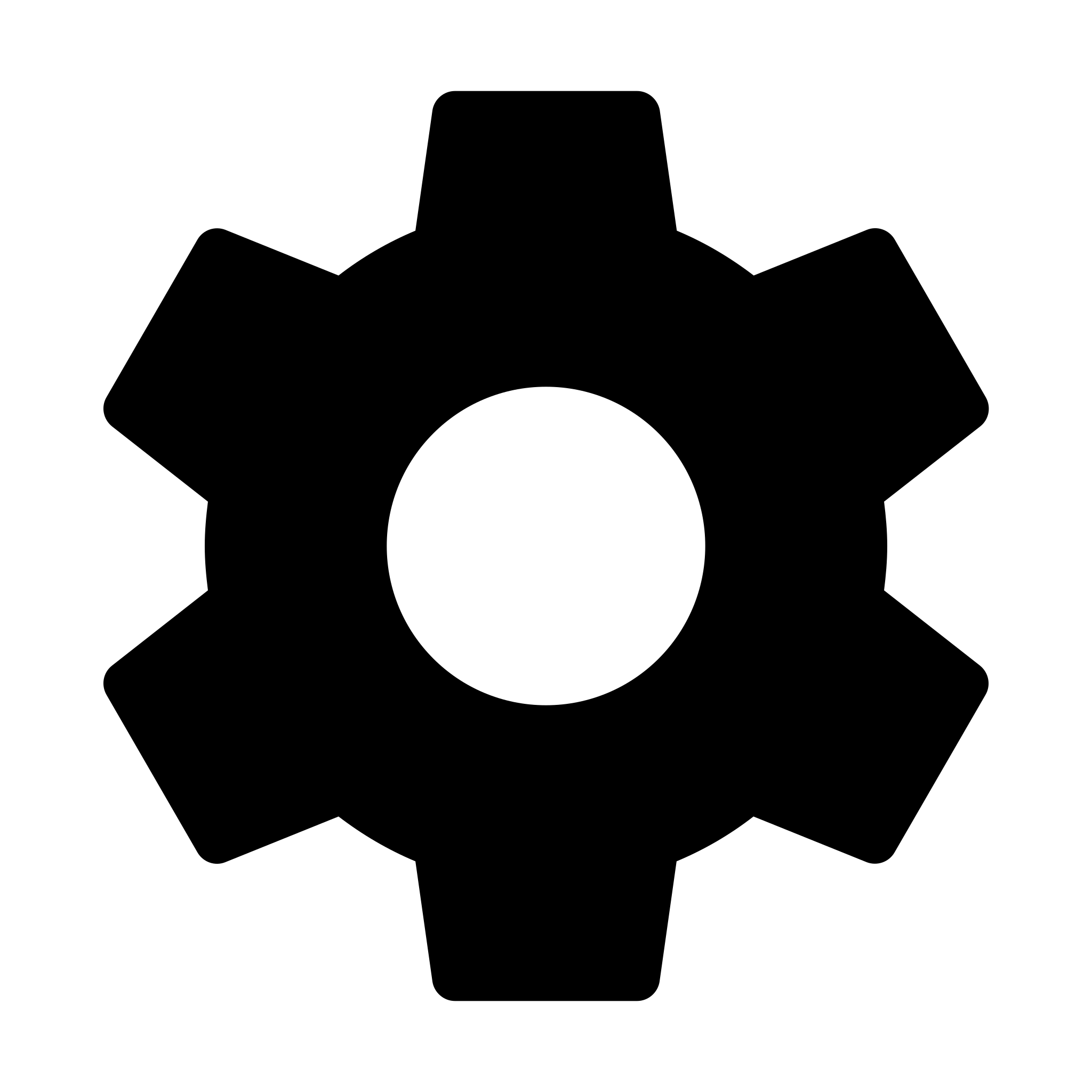
```

\* Running on http://192.168.212.126:8000

```

shows the IP address the server is running. set the same IP address on the senor logger app.

---



1. Copy the ip address. Go to settings and enable http push. Then paste the ip address e.g, <http://192.168.212.126:8000/data> in the push url and tap to test pushing.
2. Go back to the logger page and press start recording
3. Now navigate to the ip address on a web brower of your choice (other than Safari) and you will see the following web page if you run your updated script



1. To complete the assignment, you need to edit the Python script and add code to do the following:
   * count steps by following a pipeline similar to Part 1 of the assignment: filter the raw signal using appropriate filters and apply your own step detection algorithm on the filtered data.
   * plot the points where the steps occurred on the magnitude graph (as markers on the magnitude line).
2. Rerun the script and repeat the steps from 5 to 8

**Part 4: Submission and Grading**

The assignment is due by **Friday, March 24, 3:00 PM**. Note that this is right afterthe class timing on Friday. Make sure you commit and push your completed code to Github before the deadline, as late commits will not be accepted and thus cannot be graded.

In order to be evaluated, it is **MANDATORY** to present your working code during class hours (1:25 PM to 2:15 PM) on March 24.

* Note that since you have access to the live data from your own device, you have the ground truth for the number of steps taken. You can compare your algorithm’s performance to this known value. You get credit if you can demonstrate during the presentation that your algorithm detects a reasonably accurate number of steps while one of your team members walks around the room with the phone in his/her pocket.
* In-class evaluations are being done only so that we can evaluate your algorithms on the device you trained them on, and on the person(s) you trained them on, because it is generally difficult to get these algorithms to generalize across devices/people. Therefore, being present for in-class evaluation is mandatory.