# CS Club Weekly Meeting

03/18/21

# Agenda

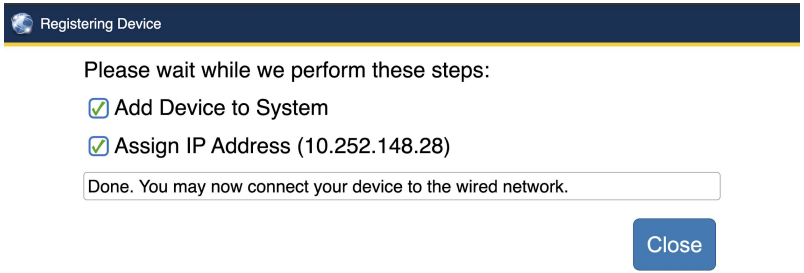
- Exploring the use of our new server!
  - SSH
  - Flask
  - SQLAlchemy
  - and more!

# What have we done so far?

- Plugged the Raspberry Pi into a power source
- Connected it to the Trinity's network via ethernet (wired connection)

# What have we done so far?

- Registered the device with Trinity which gave us a static IP address anyone on Trinity's network can connect to.



# What have we done so far?

- Modified the code in the [python-flask repository] to account for the new host - 10.252.148.28
- Turned on debug mode which is extremely useful for testing and developing
    - Debug mode picks up on any changes made to the app.py code and restarts the server when any changes occur.

```
31
32  if __name__ == "__main__":
33      db.create_all()
34      app.run(host='10.252.148.28', debug=True)
35
```

You vs. the guy she tells you not to worry about web server edition

```python
from flask import Flask

app = Flask(__name__)

@app.route("/")
def helloworld():
    return "Hello World!"

@app.route("/get_data")
def getdata():
    return "Your data"

if __name__ == "__main__":
    app.run()
```

# Recall that Flask is… Simple

- It follows the ‘batteries not included’ montra
- We’ve chosen to use Flask because of its simplicity
- Useful things not included in Flask
  - Security
  - **Persistent Storage**
  - API Management/Builder
  - A lot more that I just can’t think of off the top of my head.

# What have we done so far?

- Addison added some primitive persistent storage capabilities using SQLAlchemy

```python
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from flask import render_template


app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///db.sqlite3'
db = SQLAlchemy(app)

class Analytics(db.Model):

    id = db.Column('analytics_id', db.Integer, primary_key = True)
    views = db.Column(db.Integer)

    def __init__(self, views):
        self.views = views

    def __repr__(self):
        return f"{self.views}"

@app.route('/')
def index():
    if Analytics.query.first() == None:
        analytics = Analytics(0)
        db.session.add(analytics)
        db.session.commit()
    Analytics.query.first().views = Analytics.query.first().views + 1
    db.session.commit()

    return render_template("index.html", num_page_views=Analytics.query.first())
```

# How to access the server (2-methods)

1. Via browser - http://10.252.148.28:5000/
2. Via terminal
    a. ssh pi@10.252.148.28
    b. You will need to enter the password: raspberry
    c.

**Welcome to the student-run website for the Trinity College Computer Science Club!**

This page has been viewed 19 times.

Stored in the database on the Raspberry PI using SQLAlchemy

# Goal for tonight

- Some suggestions
  - Store the names of club members in the db and render them dynamically on the page
  - Store the name of the last person to visit the page in the db and render that on the page