Agenda

- Flask and MySQL
- Scrumwise
- Sprint 2 review
- Sprint 3 planning

Flask & MySQL

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by Armin Ronacher who leads an international group of python enthusiasts (POCCO). It is based on WSGI toolkit and jinja2 template engine. Flask is considered as a micro framework.

Installing FLASK

The first thing to do is to install Flask on your Raspberry Pi. Go to Terminal and enter:

```
sudo apt-get install python3-flask
```

Create 2 sub-folders:

- **static** *for CSS files*
- **templates** *for HTML files*

The final folder *tree* will be:

```
/flaskdemo
    /static
    /templates
```

Python WebServer Application

Create the Python *WebServer* with Flask.

- Create a file (*app.py*) in the *flaskdemo* folder
- Insert this code

```
    from flask import Flask

    app = Flask(__name__)

    @app.route('/')
    def index():
        return 'Hello 3. semester'

    if __name__ == '__main__':
        app.run(debug=True, port=5000, host='0.0.0.0')
```

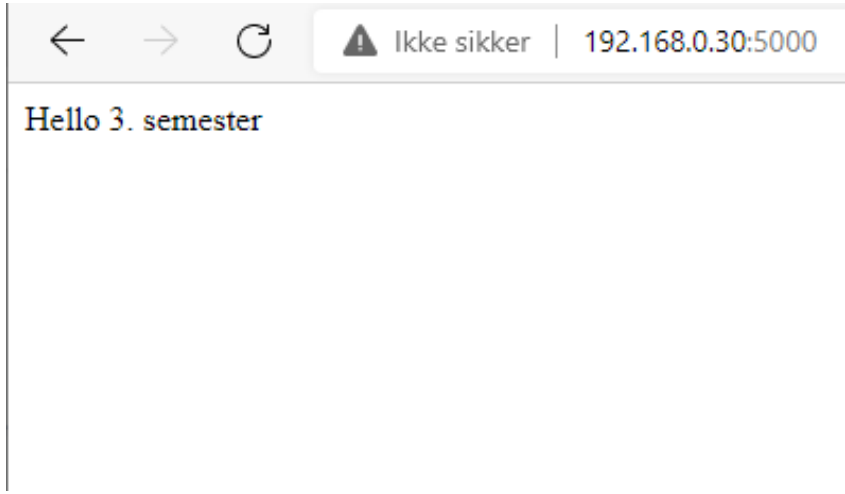
You use port **5000** and it can be accessed from all ip addresses - **host='0.0.0.0'**

Run the application - app.py

```
python3 app.py
```

Point your browser to the IP address of the Raspberry PI and port *5000*

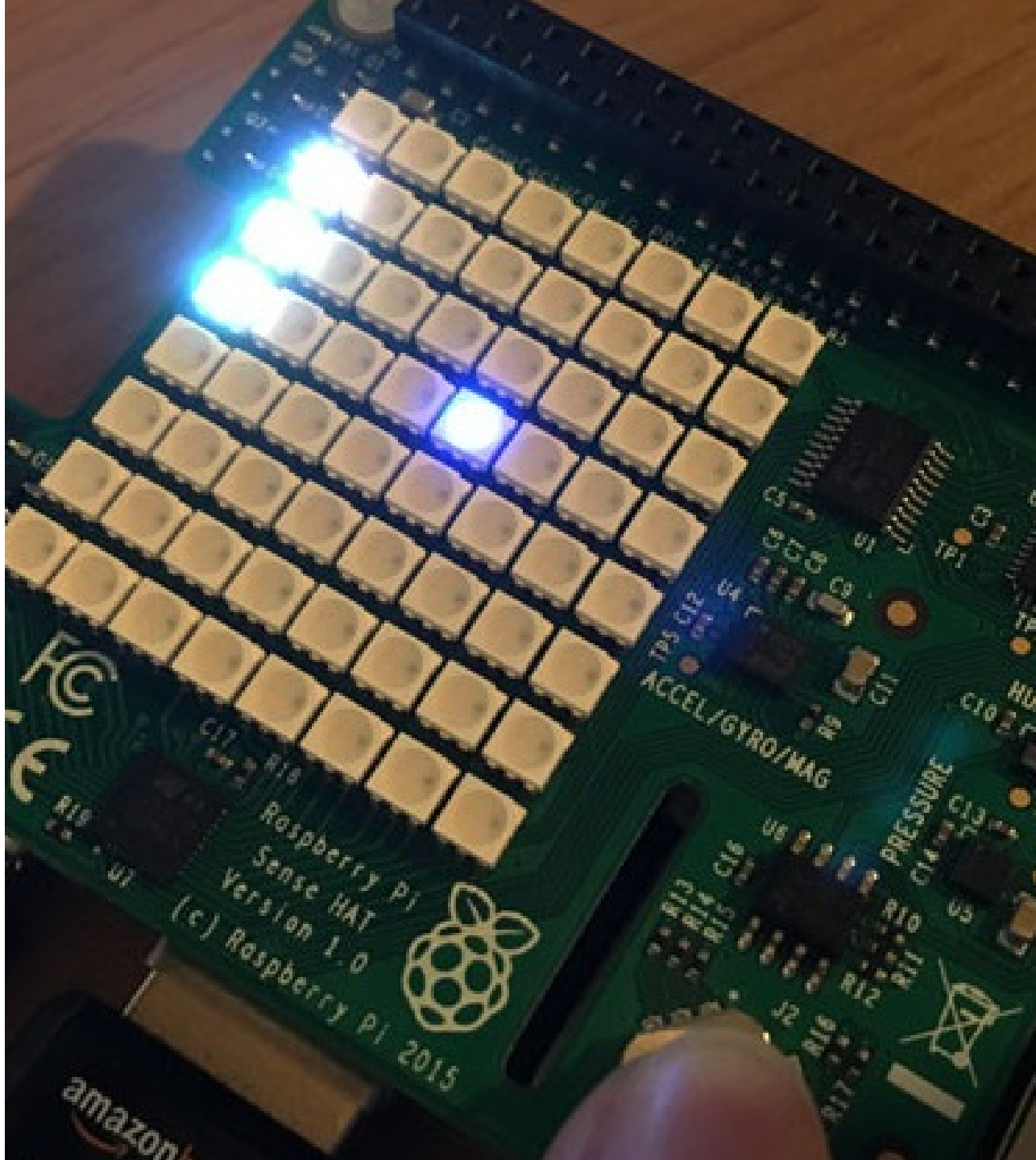
```
http://raspberryip_ip:5000
```



Sense Hat Data

Show the data from the Sense Hat in a webpage

- senesdata.py
- index.html
- style.css
- database.html



index.html

Create a file named **index.html**, save it in the **/templates** folder.

```
<!DOCTYPE html>
   <head>
      <title>{{ time }}</title>
      <link rel="stylesheet" href="../static/style.css/">
   </head>
   <body>
      <h1>Sense Hat Output</h1>
      <h2>The temperatur is: {{ temperatur }}</h2>
      <h2>The pressure is: {{ pressure }}</h2>
      <h2>The humidity is: {{ humidity }}</h2>
   </body>
</html>
```

Anything in *double curly* braces within the HTML template is interpreted as a **variable** that would be passed to it from the Python script via the **render_template** function.

senesdata.py - 1. route

```
from flask import Flask, render_template
import datetime
from sense_hat import SenseHat
import mysql.connector

# Sense data
sense = SenseHat()

# Flask
app = Flask(__name__)

@app.route("/")
def templetdata():
    now = datetime.datetime.now()
    timeString = now.strftime("%d-%m-%Y %H:%M")
    temperaturString = round(sense.get_temperature(),1)
    pressureString = round(sense.get_pressure(),1)
    humidityString = round(sense.get_humidity(),1)

    templateData = {
        'time': timeString,
        'temperatur': temperaturString,
        'pressure': pressureString,
        'humidity': humidityString
    }
    return render_template('index.html', **templateData)

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=5000, debug=True)
```

style.css

```
body {
    background: grey;
}

h1, h2 {
    color:white;
}

p {
    color:white;
}
```

```python
from flask import Flask, render_template
import datetime
from sense_hat import SenseHat
import mysql.connector

# Sense data
sense = SenseHat()

# Flask
app = Flask(__name__)

@app.route("/")
def templetdata():
    now = datetime.datetime.now()
    timeString = now.strftime("%d-%m-%Y %H:%M")
    temperaturString = round(sense.get_temperature(),1)
    pressureString = round(sense.get_pressure(),1)
    humidityString = round(sense.get_humidity(),1)

    templateData = {
        'time': timeString,
        'temperatur': temperaturString,
        'pressure': pressureString,
        'humidity': humidityString
    }
    return render_template('index.html', **templateData)

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=5000, debug=True)
```

```html
<!DOCTYPE html>
   <head>
      <title>{{ time }}</title>
      <link rel="stylesheet" href="../static/style.css/">
   </head>
   <body>
      <h1>Sense Hat Output</h1>
      <h2>The temperatur is: {{ temperatur }}</h2>
      <h2>The pressure is: {{ pressure }}</h2>
      <h2>The humidity is: {{ humidity }}</h2>
   </body>
</html>
```

senesdata.py - 2. route

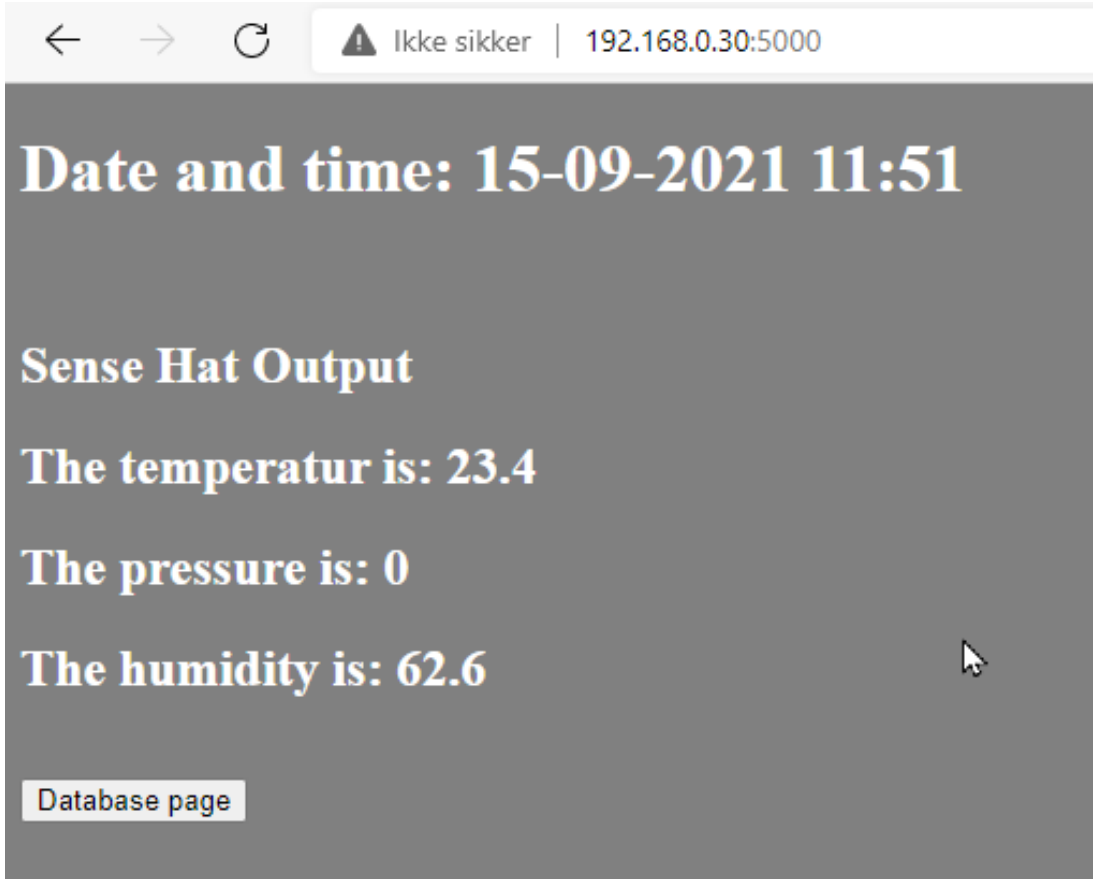
```
@app.route("/database")
def databasedata():
    try:
        conn = mysql.connector.connect(
            host="localhost",
            user="pi",
            password="Your_Password",
            database="sensedata"
        )

        mycursor = conn.cursor()
        mycursor.execute("SELECT * FROM hatdata")
        data = mycursor.fetchall()

        return render_template("database.html", data=data)

    except Exception as e:
        return (str(e))

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=5000, debug=True)
```

database.html

```
<!DOCTYPE html>
   <head>
      <title>Database</title>
      <link rel="stylesheet" href="../static/style.css/">
   </head>

   <body>
        <h1>Database info</h1>
        <br>
        <table border="1" cellpadding="5" cellspacing="5">
            {% for row in data %}
                <tr>
                {% for d in row %}
                    <td>{{ d }}</td>
                {% endfor %}
                </tr>
            {% endfor %}
        </table>
   </body>
</html>
```

```python
@app.route("/database")
def databasedata():
    try:
        conn = mysql.connector.connect(
            host="localhost",
            user="pi",
            password="[redacted]",
            database="sensedata"
        )

        mycursor = conn.cursor()
        mycursor.execute("SELECT * FROM hatdata")
        data = mycursor.fetchall()

        return render_template("database.html", data=data)

    except Exception as e:
        return (str(e))
```

```html
<!DOCTYPE html>
   <head>
      <title>Database</title>
      <link rel="stylesheet" href="../static/style.css/">
   </head>

   <body>
        <h1>Database info</h1>
        <br>
        <table border="1" cellpadding="5" cellspacing="5">
            {% for row in data %}
                <tr>
                {% for d in row %}
                    <td>{{ d }}</td>
                {% endfor %}
                </tr>
            {% endfor %}
        </table>
   </body>
</html>
```

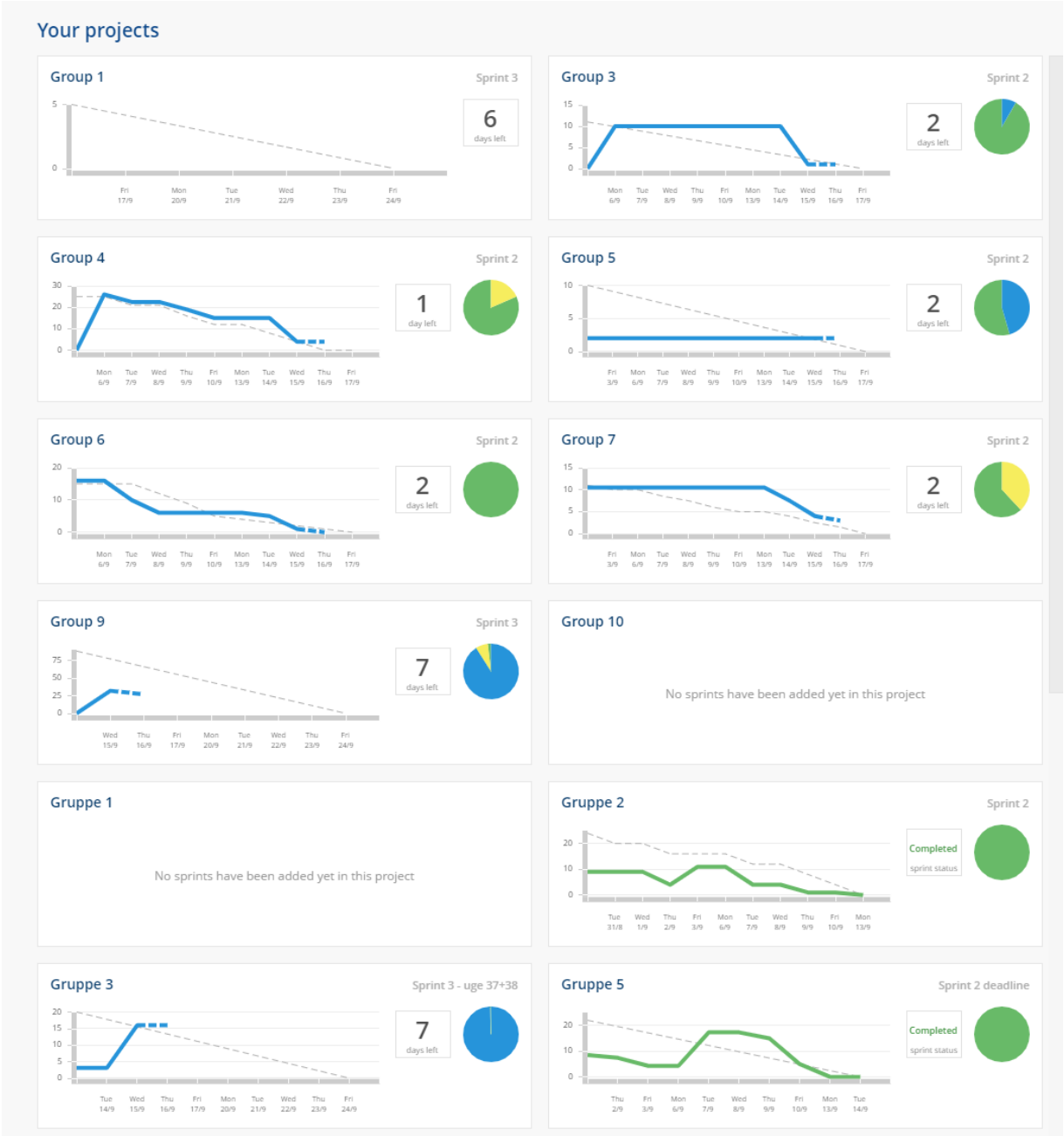


Database info

1	2021-08-06 13:47:51	34.92308044433594	38.67486572265625	0.0
2	2021-08-06 13:47:56	34.903846740722656	38.48683547973633	1008.412353515625
3	2021-08-06 13:48:01	34.96154022216797	38.56541442871094	1008.391845703125
4	2021-08-06 13:48:06	34.88461685180664	38.32125473022461	1008.3671875
5	2021-08-06 13:48:11	34.846153259277344	38.478416442871094	1008.3564453125
6	2021-08-06 13:50:07	34.96154022216797	38.2398681640625	1008.40087890625
7	2021-08-06 13:51:35	35.17308044433594	38.077091217041016	1008.419921875
8	2021-08-06 13:54:50	35.67308044433594	37.03870391845703	1008.384765625
9	2021-08-06 13:56:19	35.42308044433594	38.10515594482422	1008.366455078125
10	2021-08-06 13:56:37	35.596153259277344	37.61122131347656	1008.32763671875
11	2021-08-06 13:56:56	35.5384635925293	37.3193473815918	1008.32958984375
12	2021-08-06 13:57:14	35.653846740722656	37.3782844543457	1008.37109375
13	2021-08-06 13:57:33	35.75	37.209896087646484	1008.32421875
14	2021-08-06 13:57:52	35.615386962890625	37.20147705078125	1008.37109375
15	2021-09-14 16:46:40	27.865385055541992	49.76318359375	0.0
16	2021-09-14 16:46:56	27.846155166625977	50.46479797363281	1019.104736328125
17	2021-09-14 16:47:14	27.903846740722656	50.53215408325195	1019.09912109375

Scrumwise

https://www.scrumwise.com/scrum/#/overview/project/group-1/id-199424-3369-1



2021-37: Uge 37

- Sprint Retrospective
- **SCRUM** - *Tirsdag 14-09*
  - Sprint Review
  - Sprint Planning
- **IT** - *Fredag 17-09*
  - SCRUM Backlog
  - IT emne - *Hardware - computerkomponenter*

2021-38: Uge 38 - Afslutning

- Sprint Retrospective
- **SCRUM** - *Tirsdag 21-09*
  - Sprint Review
- **IT** - *Fredag 24-09 - Kl. 08:30 til 12:30*
  - Præsentation
  - IT emne - *Operating systems*

Aflevering

- Kode
- IT-emner
- PowerPoint