

# Web based shopping list

Joonas Partanen, TVT19SPO  
Information technology

## Introduction

This project is web based shopping list, that runs on homes wifi-network. The idea was to create simple to use, easy to reach and something that is always up to date list for the whole family to use. The site is also "secured" with simple user-password authentication. The web server runs on Raspberry pi 4. I used Python's Flask framework to set it up.

```
ostolist.py  index.html
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="utf-8">
6 <title>Ostoslista</title>
7 <meta http-equiv="Cache-control" content="no-cache">
8 <meta name="author" content="">
9 <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/ostoslista.css')}}">
10 <script type="text/javascript" src="{{ url_for('static', filename='js/ostoslista.js')}}"></script>
11 </head>
12 <body onload="readFromFile();">
13 <div class="list">
14 <div class="list">
15 <div class="list">
16 <div class="list">
17 <div class="list">
18 <div class="list">
19 <div class="list">
20 <div class="list">
21 <div class="list">
22 <div class="list">
23 <div class="list">
24 <div class="list">
25 <div class="list">
26 <div class="list">
27 <div class="list">
28 <div class="list">
29 <div class="list">
30 <div class="list">
31 <div class="list">
32 <div class="list">
33 <div class="list">
34 <div class="list">
35 <div class="list">
36 <div class="list">
37 <div class="list">
38 <div class="list">
39 </div>
40 </body>
41 </html>
```

FIGURE 1. HTML code for the page

## Objectives

In this mini-project goal was to create simple IoT like shopping list and learn basic web-development. I set myself a few basic features that must work in this application. Those were: basic authentication, loading always the most recent version of the list, adding new items to the list, and creating a new empty list. And the list must be stored in the server as a .txt file, not just in the HTML page.

## Methods

Raspberry pi 4 was the whole server for this simple web application. I used python on the servers side code such as handling http POST requests, javascript for the client side function and basic HTML and CSS for the site and styling.

```
ostolist.py
1 from flask import Flask, render_template, request, send_file, redirect
2 from flask_htpasswd import HtpasswdAuth
3 import subprocess
4 import os
5
6 app = Flask(__name__)
7 app.static_folder = 'static'
8 app.config['SEND_FILE_MAX_AGE_DEFAULT'] = 0
9 app.config['FLASK_HTTPSWD_PATH'] = '/home/pi/.htpasswd'
10 app.config['FLASK_AUTH_ALL'] = True
11
12 htpasswd = HtpasswdAuth(app)
13
14 @app.route('/', methods=['GET', 'POST'])
15
16 def home():
17
18     input_item=""
19     if request.method == 'POST':
20         input_item = request.form['item']
21         with open("static/ostoslista.txt", "a") as f:
22             f.write(str(input_item) + "\n")
23
24     return render_template("index.html")
25
26 @app.route('/static/ostoslista.txt')
27 def asd():
28     return send_file('static/ostoslista.txt')
29
30 @app.route('/config.json')
31 def qwe():
32     return send_file('static/js/config.json')
33
34 @app.route('/newlist')
35 def newfile():
36     subprocess.call(['/home/pi/ostoslista_app/newfile.bash'], shell=True)
37     return redirect("/")
38
39
40 if __name__ == '__main__':
41     app.run(debug = True, host='0.0.0.0')
```

FIGURE 2. Servers Python code.

Javascript funtion that sends GET request to the server is executed always when HTML page is loaded so the page always shows the newest version of the list. There is also a HTML button that executes bash script which takes care of storing old lists and creating a new empty list to right directory.

## Results

All the functions that I wanted to include work as intended. There is always better ways to do certain

things, but at the end I'm happy with the application.



FIGURE 3. View in browser

## Conclusions

This was fun and very educational little project for me. I learned a lot about web development. I also have some more ideas for the application and I will continue developing this a bit more. At first fixing scandic alphabets as input.

## References

### Flask

<https://flask.palletsprojects.com/en/1.1.x/>

### Authentication

<https://pypi.org/project/flask-htpasswd/>

### Running shell scripts

<https://docs.python.org/3.4/library/subprocess.html>