

SPRINT - 4

Deployment

FLASH IMPLEMENTATION USING PYTHON CODE :

```
from flask import Flask, current_app, send_from_directory, render_template
from flask import jsonify
from flask import request
from flask_pymongo import PyMongo

from bson import json_util
import re
import os

app = Flask(__name__, static_folder="./client/build/static",
            template_folder="./client/build")

ENV = 'prod'

app.config['MONGO_DBNAME'] = 'test'

if ENV == 'dev':
    import config
    app.debug = True
    app.config['MONGO_URI'] = config.api_key
else:
    app.debug = False
    app.config['MONGO_URI'] = os.environ.get('MONGO_URI')

mongo = PyMongo(app)

@app.route('/api/foods/all', methods=['GET'])
def get_all_foods():
    food = mongo.db.foods
    output = []
    for food in food.find():
        output.append({'ndbno': food['ndbno'], 'name': food['name'], 'weight': food['weight'],
                        'measure': food['measure'], 'nutrients': food['nutrients']})
    return jsonify(output)

@app.route('/api/foods/search', methods=['GET'])
def get_queried_foods():
    food = mongo.db.foods
    fieldsets = []
    results = []
    queries = []
```

```

nutrients_params = request.args.get('nutrients')
mins_params = request.args.get('mins')
maxes_params = request.args.get('maxes')

if ',' in nutrients_params:
    nutrients = nutrients_params.split(',')
else:
    nutrientsList = []
    nutrientsList.append(nutrients_params)
    nutrients = nutrientsList

if ',' in mins_params:
    mins = mins_params.split(',')
else:
    minsList = []
    minsList.append(mins_params)
    mins = minsList

if ',' in maxes_params:
    maxes = maxes_params.split(',')
else:
    maxesList = []
    maxesList.append(maxes_params)
    maxes = maxesList

fieldsets.append(nutrients)
fieldsets.append(mins)
fieldsets.append(maxes)

for i in range(len(fieldsets[0])):
    regex = ".*" + fieldsets[0][i] + ".*"
    query = {
        'nutrients': {
            '$elemMatch': {
                'nutrient': {"$regex": regex, "$options": "-i"},
                'gm': {
                    '$gt': int(fieldsets[1][i]),
                    '$lte': int(fieldsets[2][i]),
                }
            }
        }
    }
    queries.append(query)

results = food.find({'$and' : queries})
return json_util.dumps(results, default=json_util.default)

@app.route('/')
def index():

```

```
return render_template("index.html")
# ungitignored client/build, ran npm run build, deployed.

if __name__ == '__main__':
    app.run()

# Run server in watch mode:
# FLASK_APP=app.py FLASK_ENV=development flask run --port=5000
```

HTML CODE

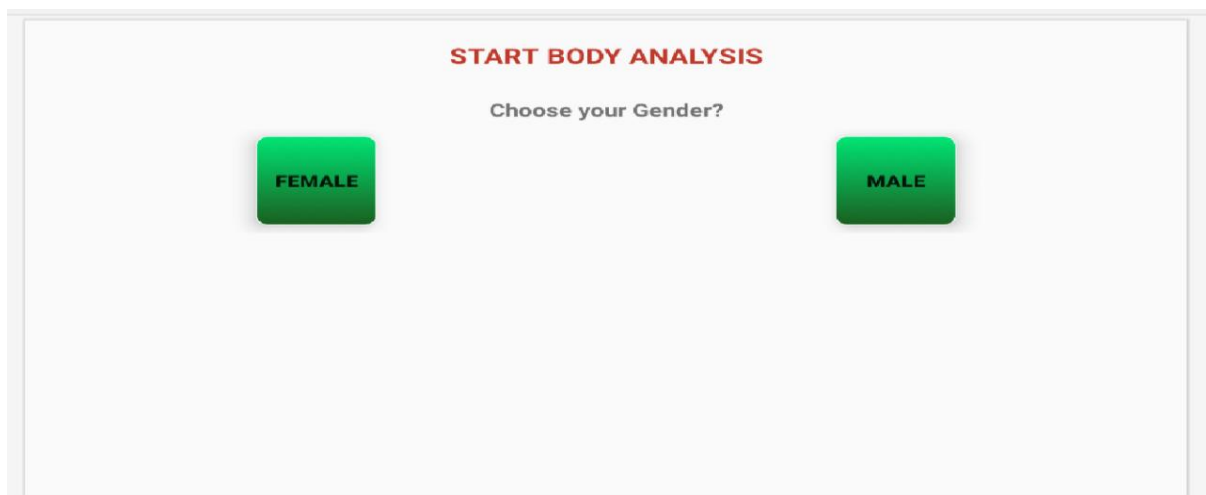
```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <meta name="theme-color" content="#000000" />
  <meta name="description" content="Web site created using create-react-app" />
  <title>Food Nutrition App</title>
</head>

<body>
  <noscript>You need to enable JavaScript to run this app.</noscript>
  <div id="root"></div>
</body>

</html>
```

Output for Deployment :



Physical Activity

ALMOST NO PHYSICAL ACTIVITY

FREQUENTLY WALKING

EXERCISE 1-2 A WEEK

EXERCISE 3-5 A WEEK

EXERCISE EVERYDAY

WATER REMINDER

Set Reminder Period:

1 HOUR

30 MIN

STOP REMINDER

START REMINDER

YOUR PROFILE SUMMARY

BMI Analysis [kg/m²]:



BMI

BMI



BODY WATER analysis:



WATER

SET REMINDER

Calories analysis:



Calories

CALCULATE CALORIES