Proiect

Python

Cuprins

1. Prezentarea temei - argument
2. Funcționalitatea aplicației
3. Programe folosite
4. Implementarea aplicației
5. Bibliografie
6. Anexă
7. **Prezentarea temei**

Indexul de masă corporală (IMC sau BMI, în engleză) reprezintă o metrică esențială în monitorizarea greutății corporale și este un instrument important în eforturile de menținere a unei sănătăți optime. Prin evaluarea raportului dintre greutate și înălțime, IMC oferă o perspectivă generală asupra compoziției corporale și a potențialelor riscuri pentru sănătate.

Utilizarea unei aplicații dedicate pentru calculul și monitorizarea IMC aduce numeroase beneficii. Prin intermediul unei astfel de aplicații, utilizatorii pot accesa rapid și ușor informații despre starea lor de sănătate, oferindu-le posibilitatea de a lua decizii informate în ceea ce privește regimul alimentar și nivelul de activitate fizică.

Prin accesibilitatea crescută oferită de aplicații, utilizatorii pot beneficia de o monitorizare constantă a IMC-ului lor, ceea ce poate contribui la prevenirea problemelor de sănătate asociate cu fluctuațiile extreme ale greutății corporale. În final, integrarea tehnologiei în gestionarea sănătății personale prin intermediul aplicațiilor pentru IMC poate juca un rol semnificativ în promovarea unui stil de viață echilibrat și a unei sănătăți durabile.

1. **Funcționalitatea aplicației**

Aplicația a fost proiectată astfel încât să aibă o interfață cât mai simplă și interactivă. Atunci când este lansată, se încarcă prima pagină în care utilizatorul își poate introduce datele necesare calculării BMI

A screenshot of a computer

Description automatically generated

Odată ce utilizatorul completează formularul cerut și apasă pe butonul „Calculate BMI” aplicația afișează rezultatul obținut în urma calculului, încadrarea utilizatorului într-o anumită categorie de greutate bazată pe valoarea BMI și un feedback:

A screenshot of a computer

Description automatically generated

Formularul rămâne activ pentru a prelua date noi, în cazul în care se dorește o reevaluare. Acesta poate fi completat din nou, iar pentru o recalculare se apasă butonul „Calculate BMI” rămas activ. Mai mult, imediat după afișarea categoriei, un nou link devine disponibil, pe care utilizatorul îl poate accesa pentru a fi trimis la o pagină dedicată descrierii diferitelor categorii de greutate:

A screenshot of a computer

Description automatically generated

Dacă se dorește întoarcerea la pagina principală a aplicației, se poate accesa linkul de la sfârșitul descrierilor.

1. **Programe folosite**

Pentru implementarea aplicației a fost folosită librăria dedicată Flask pentru randarea paginilor și formatele HMTL și CSS pentru design-ul acestora. De asemenea Flask include framework-ul Jinja ce permite modificarea conținutului diverselor pagini în funcție de parametrii introduși de utilizator. Din motive de portabilitate am utilizat un Virtual Environment (VENV) pentru a putea rula aplicația de pe orice stație fără a mai fi nevoie ca pe sistemul respectiv să fie instalate oricare dintre celelalte aplicații numerotate mai sus. Instalarea interpretorului Python reprezintă astfel, singura precondiție pentru a putea rula programul ce pornește aplicația pe un anumit server sau pe un server local, de test.

1. **Implementarea aplicației**

Așa cum am început în finalul capitolului dedicat software-ului folosit în implementare, rularea aplicației se face pe un server local (în faza de testare, ulterior acesta poate fi înlocuit cu unul închiriat) care încarcă pagini web cu conținut adaptat cererilor efectuate de client/utilizator către server.

Flask folosește un format standardizat al directorului de lucru pentru rularea aplicațiilor. Paginile HTML ce urmează a fi afișate trebuie puse într-un director „templates”, iar fișierele CSS, responsabile de design și formatul conținututlui se regăsesc în directorul „static”.

Am anexat la sfârșitul documentației implementarea codului. Voi trece prin logica codului în paralel cu modificările pe care acesta le determină asupra conținutului paginilor încărcate.

La începutul programului linia 1 încarcă din librăria Flask constructorul de bază al unei aplicației Flask, funcția render\_template ce formatează pagini HTML cu parametrii transmiși cu ajutorul librăriei Jinja și funcția request ce se ocupă cu cererile HTTP (cereri de tip POST) făcute de utilizator către server.

Linia 3 apelează constructorul Flask pentru a crea aplicația și îi trebuie transmis ca parametru variabila \_\_name\_\_ (convenție stabilită de dezvoltatori, în principiu, pentru a verifica că aplicația este inițializată într-un cod sursă rulat și nu într-o librărie ce urmează a fi inclusă într-un altul).

Liniile 5 – 19 definesc cele două funcții de bază ale aplicației, calculate\_bmi și get\_weight\_category ce permit întâi calcularea BMI-ului și ulterior încadrarea utilizatorului într-o anumită categorie de greutate (prima returnează valoarea BMI calculată pe baza înălțimii și a greutății introduse ca mai apoi următoarea să preia acest rezultat și să returneze categoria de greutate sub formă de șir de caractere).

În continuare sunt definite, între liniile 20 și 37 sunt prezentate rutele pe care aplicația va afișa conținut și va gestiona cererile făcute de utilizator către server, după ce a prelucrat datele trimise de acesta

Prima rută (linia 20) modelează conținutul primei pagini (aflată la adresa de bază „/”) aceasta acceptând două tipuri de cereri „GET” și „POST”. Prima este folosită pentru încărcarea în primă instanță a paginii principale (atunci când utilizatorul nu a introdus nicio dată), iar cea de a doua pentru reactualizarea paginii și afișarea rezultatelor în baza parametrilor transmiși de client prin formular.

În prima situație, la cererea de tip „GET”, parametrii sunt setați la valoarea neutră „None” folosită în Python pentru a declara o variabilă neinițializată. În urma verificării request.method == 'POST', se va trece peste execuția blocului de instrucțiuni, iar render\_template va genera o pagină în care nu este afișat decât formularul inițial (vezi conținutul index.html, acolo corpul {%if bmi %} verifică dacă valoarea lui bmi a fost inițializată, deci în acest caz sare peste afișarea liniilor rezervate rezultatului care nu există în acest punct inițial).

În cea de a doua situație, la cererea de tip „POST”, condiția request.method == 'POST' este îndeplinită și variabilele bmi și category sunt suprascrise de rezultatele returnate de funcțiile dedicate ce primesc ca parametrii valorile de la câmpurile formularului de pe pagina inițială de pe care a fost făcută cererea. De data aceasta constructorului render\_template îi sunt transmiși parametrii diferiți de valoarea neutră „None”, și când pagina va fi reîncărcată, în sintaxa de Jinja condiția {%if bmi %} va fi îndeplinită (variabila bmi a înregistrat o valoarea la această cerere) iar conținutul paginii principale se va modifica, incluzând de data aceasta și liniile 24-25 din index.html ce primesc valorile bmi și category prin constructorul render\_template returnat de funcția index().

La prima cerere de tip „POST” , precedată de cererea inițială „GET” formularul va fi golit și disponibil pentru a lansa o nouă cerere de tip „POST” pentru a permite o calculare nouă, afișare a noului rezultat și tot așa.

În plus, în codul HTML al paginii principale (index.html) se fac aceleași verificări (de inițializare a variabilelor și obținere a valorii lor, în caz afrmativ) pentru afișarea unui feedback interactiv în funcție de categoria de greutate obținută.

În codul sursă se mai află și o a doua rută, (@app.route('/weight\_categories')), care permite randarea unei pagini dedicate informațiilor suplimentare despre difertitele categorii de greutate. Pagina conține un link de revenire către pagina principală. Accesarea paginii principale se face prin cerere de tip „GET”, despre care am vorbit mai sus (doar afișează pagina, cu date neințializate și conținut în conformitate cu datele neinițializate).

În final, în ultimele două linii se rulează aplicația Flask creată anterior, verificând întâi că crearea obiectului-aplicație se face din codul sursă rulat și nu unul importat, așa cum am mai spus mai sus.

Despre design-ul proiectului nu voi intra în detalii; atașez la final fișierul CSS. Menționez că a fost o provocare să creez un fundal dinamic care rulează în buclă un videoclip.

1. **Bibliografie**

* [Our Documentation | Python.org](https://www.python.org/doc/)
* [Welcome to Flask — Flask Documentation (3.0.x) (palletsprojects.com)](https://flask.palletsprojects.com/en/3.0.x/)
* [CS50x 2023 - Lecture 9 - Flask (youtube.com)](https://www.youtube.com/watch?v=oVA0fD13NGI&t=1970s)
* [HTML: HyperText Markup Language | MDN (mozilla.org)](https://developer.mozilla.org/en-US/docs/Web/HTML)
* [ChatGPT (openai.com)](https://chat.openai.com/)

1. **Anexă**

Cod sursă:

from flask import Flask, render\_template, request

app = Flask(\_\_name\_\_)

def calculate\_bmi(weight, height):

    height\_in\_meters = height / 100  # convert height from cm to meters

    bmi = weight / (height\_in\_meters \*\* 2)

    return round(bmi, 2)

def get\_weight\_category(bmi):

    if bmi < 18.5:

        return 'Underweight'

    elif 18.5 <= bmi < 25:

        return 'Normal weight'

    elif 25 <= bmi < 30:

        return 'Overweight'

    else:

        return 'Obese'

@app.route('/', methods=['GET', 'POST'])

def index():

    bmi = None

    category = None

    if request.method == 'POST':

        weight = float(request.form['weight'])

        height = float(request.form['height'])

        bmi = calculate\_bmi(weight, height)

        category = get\_weight\_category(bmi)

    return render\_template('index.html', bmi=bmi, category=category)

@app.route('/weight\_categories')

def result():

    # You can include any necessary data processing or logic here

    return render\_template('weight\_categories.html')

if \_\_name\_\_ == '\_\_main\_\_':

    app.run()

**Conținutul paginii principale (index.html)**

<!DOCTYPE html>

<html>

<div class="video-container">

    <video autoplay muted loop id="video-background">

        <source src="../static/video.mp4" type="video/mp4">

        Your browser does not support HTML5 video.

    </video>

</div>

<div class="centered-div">

    <title>BMI Calculator</title>

    <link href="../static/styles.css" rel="stylesheet">

    <h1>BMI Calculator</h1>

    <form method="post">

        <label for="weight">Enter your weight (in kg):</label>

        <input type="text" id="weight" name="weight" required><br><br>

        <label for="height">Enter your height (in cm):</label>

        <input type="text" id="height" name="height" required><br><br>

        <input id="submit-button" type="submit" value="Calculate BMI">

    </form>

    {% if bmi %}

    <h2>Your BMI is: {{ bmi }}</h2>

    <h2>Your weight category is: {{ category }}</h2>

    <a href="/weight\_categories">Find out more about weight categories!</a> <!-- Add a link to view the result -->

    {% endif %}

    {% if category %}

    <h3>Here's some feedback:</h3>

    {% if category == "Underweight" %}

    <p>Your health journey matters! Consider consulting a nutritionist or dietitian to explore healthy ways to gain

        weight.

        Incorporating nutrient-dense foods and a balanced meal plan tailored to your needs could be a great start.

        Remember, gradual changes can lead to lasting improvements in your well-being</p>

    {% endif %}

    {% if category == "Normal weight" %}

    <p>Keep up the fantastic work maintaining a healthy weight! Balanced nutrition and regular exercise are key.

        Remember to listen to your body's signals and prioritize self-care. Your commitment to overall wellness is

        inspiring!</p>

    {% endif %}

    {% if category == "Overweight" %}

    <p>Taking steps toward a healthier lifestyle is incredible! Consider small, sustainable changes like portion

        control, adding more veggies to meals, and finding enjoyable physical activities.

        You're already on the path to a healthier you—keep going, one step at a time! </p>

    {% endif %}

    {% if category == "Obese" %}

    <p>Your dedication to improving your health is admirable! Seeking support from healthcare professionals, such as a

        nutritionist or doctor, can offer guidance tailored to your needs. Small, gradual changes in diet and physical

        activity can make a significant difference.

        Remember, every positive choice brings you closer to a healthier, happier you. You've got this!</p>

    {% endif %}

    {% endif %}

</div>

</html>

**Conținutul paginii secundare:**

<!DOCTYPE html>

<html>

<div class="video-container">

    <video autoplay muted loop id="video-background">

        <source src="../static/video.mp4" type="video/mp4">

        Your browser does not support HTML5 video.

    </video>

</div>

<div class="centered-div">

    <title>Weight categories</title>

    <h2>Weight categories</h2>

    <h2>Underweight</h2>

    <ul>

        <li>Nutritional deficiencies: Insufficient intake of essential nutrients can weaken the immune system, lead to

            fatigue, and impair organ function.</li>

        <li>Osteoporosis: Lower body weight can increase the risk of bone weakening and fractures.</li>

        <li>Fertility issues: Women may face irregular menstrual cycles or difficulty conceiving due to low body weight.

        </li>

        <li>Weakened immune system: Underweight individuals may be more susceptible to infections and illness due to a

            compromised immune system.</li>

    </ul>

    <h2>Overweight</h2>

    <ul>

        <li>Heart disease and stroke: Excess weight can raise blood pressure and cholesterol levels, increasing the risk

            of heart disease and stroke.</li>

        <li>Type 2 diabetes: Obesity is a significant risk factor for developing insulin resistance and type 2 diabetes.

        </li>

        <li>Joint problems: The additional weight puts strain on joints, leading to conditions like osteoarthritis.</li>

        <li>Sleep apnea: Obesity increases the likelihood of experiencing interrupted breathing during sleep, leading to

            sleep apnea.</li>

    </ul>

    <h2>Obese</h2>

    <ul>

        <li>Higher risk of cancers: Obesity is associated with an increased risk of various cancers, including breast,

            colon, and pancreatic cancer.</li>

        <li>Mental health issues: Obesity can contribute to depression, low self-esteem, and body image concerns.</li>

        <li>Liver disease: Non-alcoholic fatty liver disease (NAFLD) is more prevalent among obese individuals.</li>

        <li>Respiratory problems: Breathing difficulties, including asthma and shortness of breath, are more common in

            obese individuals.</li>

    </ul>

    <a href="/">Get back to the calculator</a>

    <link href="../static/styles.css" rel="stylesheet">

</div>

</html>