Университет по библиотекознание и информационни технологии

**КУРСОВ ПРОЕКТ – Приложение за фитнес, здраве и здравословно хранене**

*Изготвили: Йордан Стоилов, Георги Иванов*

**УВОД**

Здравето и физическата форма са значим акцент в съвременното общество, което се дължи на нарастващото осъзнаване на ползите от активния начин на живот и рисковете, свързани със заседналото поведение. Появата на технологиите е от решаваща роля за тази промяна, предоставяйки инструменти и приложения, които помагат на хората да наблюдават и подобряват физическото си състояние. По-специално, приложенията за проследяване на физическата активност притежават огромна популярност, предлагайки на потребителите възможността да проследят тренировките си , да следят напредъка си и да постигат своите фитнес цели.

Проектът представя разработването на уеб сайт за намиране на упражнения, здравословни рецепти, музикални плейлисти и тестове на здравословното състояние, предназначено да помага на потребителите да бъдат мотивирани и да поддържат форма, както и да се хранят правилно. Използват се съвременни уеб технологии, за да се осигури безпроблемно потребителско изживяване, което гарантира достъпност и лекота на използване.

**Цели:**

Основната цел на този проект е да се разработи уеб сайт, който да улеснява трениращите като им предоставя платформа, която да им помага да следят здравословното си състояние, да се мотивират с музика, както и да намират и запазват в любими упражнения и рецепти за ястия.

**Специфични цели:**

* Да се разработи и внедри удобен за потребителя интерфейс за намиране на упражнения с различна трудност и мускулна група.
* Да се даде възможност на потребителите да намират Spotify плейлисти с музика, която да слушат по време на тренировка.
* Да се предоставят здравословни рецепти за приготвяне на храна.
* Да се извършват тестове на здравословното състояние на потребителя.

**1.Първа глава: Методи и технологии за реализиране на проекта**

* **1.1.** Приложението за следене на физическата активност използва комбинация от front end и back end технологии, за да осигури стабилна и удобна за потребителя платформа.
* **1.2.Основните технологии, използвани в този проект, включват:**
* **Flask**: Микро уеб рамка за Python, която избрахме, защото е сравнително проста и гъвкава при изграждането на уеб приложения.
* **CS50 SQLite**: Използвахме тази база данни, защото е сравнително лека и лесна за употреба. В уеб сайта е използвана именно тя, за да можем ефективно да съхраняваме потребителските данни
* **HTML/CSS**: Стандартни технологии за създаване и оформяне на уеб интерфейса.
* **JavaScript**: Използвахме този език на малко места - в някои от HTML файловете, за да подобрим интерактивността и бързото реагиране на уеб сайта.
* **Bootstrap**: Лека и лесна за използване open-source front-end уеб рамка, с помощта на който постигнахме голяма част от дизайна на менюта, полета и други визуални елементи.

**2.Втора глава: Проекция на структурата на Фитнес Web сайта**

**2.1.Функционални изисквания към Фитнес Web сайта**

**2.2. Избор на програмни средства и**

**развойната среда**

* Езикът “Python” е избран поради лесната му четимост , лекота на използване и обширни библиотеки за web разработка и визуализация на данни.
* Flask(web рамка),която е базирана на Python, осигурява гъвкавостта, необходима за ефективното изграждане и разширяване на web приложението и позволява бързо разработване.
* SQLite: Избрахме този вграден механизъм за бази данни, защото осигурява простота и лекота.
* Bootstrap: Избрахме тази уеб рамка, тъй като улеснява създаването на красив и отзивчив front-end. Колекцията от готови и предварително зададени елементи, заедно с възможността им за персонализиране и обширната документация, направиха тази библиотека страхотен избор.

**2.3. Алгоритъм**Алгоритъмът за web сайта за проследяване на физическата активността включва няколко ключови процеса:

**а) Регистрация и удостоверяване на потребителя:**

Потребителите могат да създават акаунти и да влизат в web сайта.

Данните на потребителите се съхраняват и удостоверяват по сигурен начин с помощта на управлението на сесиите на Flask и хеширащите алгоритми в werkzeug.security библиотеката.

**б) Тест на текущото здравно състояние:**

Потребителите могат да извършат тест за текущото си здравословно състояние, въвеждайки данни като своите килограми, ръст, възраст, данни за кръвно налягане и т.н. Данните се съхраняват в базата данни SQLite и са достъпни всеки път при посещение на персоналния профилs.

**в) Плейлисти в Spotify по дадена ключова дума:**

Приложението предлага плейлисти в платформата Spotify по зададена от потребителя ключова дума (напр. „хип-хоп“, „хаус“, т.н.), представени по красив начин чрез Bootstrap.

**г) Предложени тренировки:**

Потребителят може да намира упражнения, като ги филтрира по тип упражнение, мускулна група, и трудност. Упражненията са представени като карти със подходяща картинка и допълнителна информация за начина на изпълнение и друга информация. При желание, всяко едно упражнение може да бъде добавено в Любими.

**д) Рецепти за храна:**

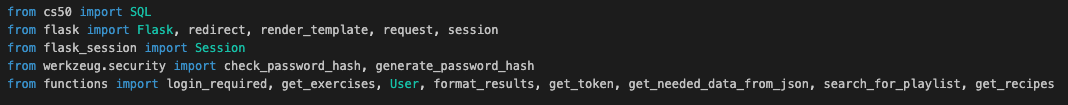
Приложението дава възможност за търсене на здравословни рецепти по ключова дума, които са представени по подобен начин на тренировките.

**3.1 Структура на проекта**

* static/: Съдържа статичният файл styles.css и изображенията за заден фон.
* templates/: Съдържа всички HTML файлове.
* app.py: Основният файл съдържащ кодът написан на Python
* functions.py: Допълнителни (помощни) функции
* database.db: Файл съдържащ базите данни

**3.2 app.py**

**3.2.1 Импортиране на библиотеки и създаване на приложение**

****

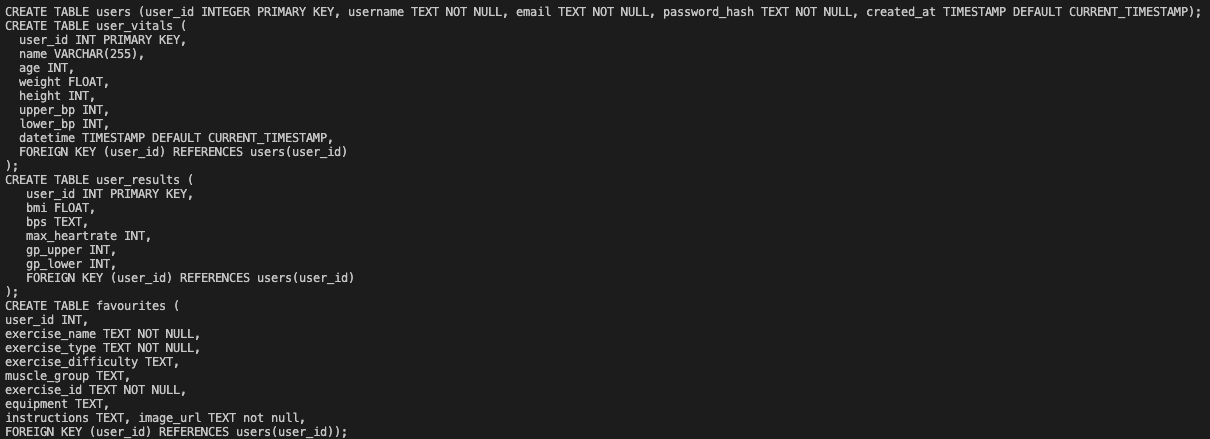
Тук се импортират нужните библиотеки. Flask е уеб рамка, cs50.SQL е рамка, която позволява лесна работа със SQLite без нуждата от ORM, werkzeug е библиотека, която съдържа функции за хеширане и проверка на хешове, необходима за правилно съхранение и проверка на пароли, functions е гореспоменатият Python файл, написан от нас за улеснение при работа с app.py.

**3.2.2 Конфигурация на Flask приложение**

****

Създаваме Flask приложение и го конфигурираме. Настройваме база данни CS50 SQLite и създаваме сесия.

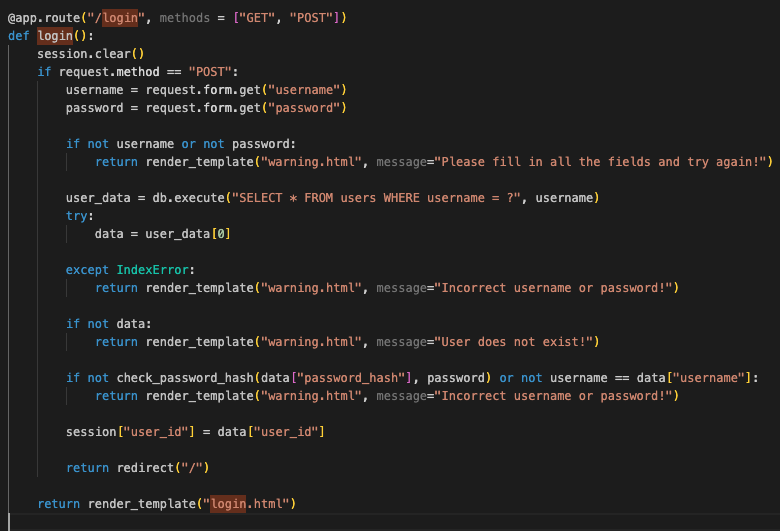
**3.2.3 Модели на базата данни**



Таблицата Users съдържа информацията за потребителите, User Results съдържа информация за здравословното му състояние, а Favourites – любимите му упражнения.

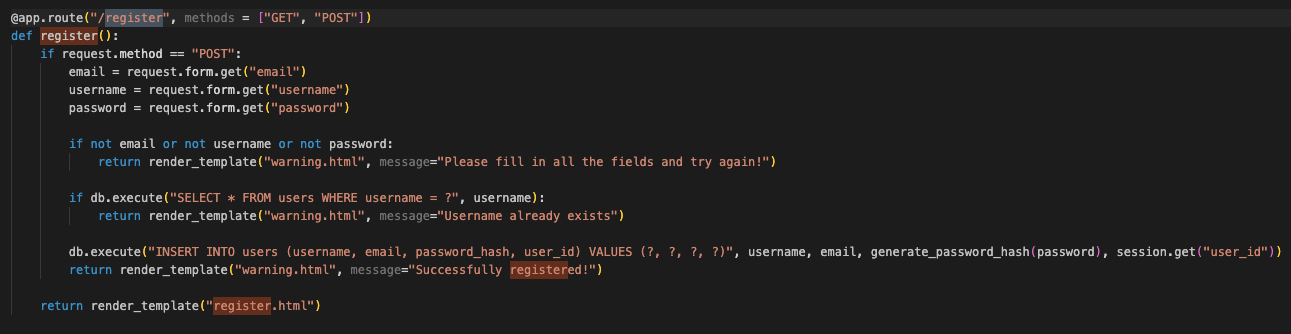
**3.2.4 Пътища на приложението**

**3.2.4 Логин страница**

****

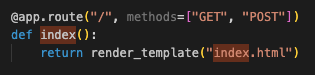
Този път обработва данните от формата за вход. Проверява дали потребителят съществува в базата данни и ако да, запазва потребителското име в сесията и пренасочва към главната страница, а при неуспех, го уведомява със съобщение да опита отново.

3.2.5 Регистрация на нов потребител



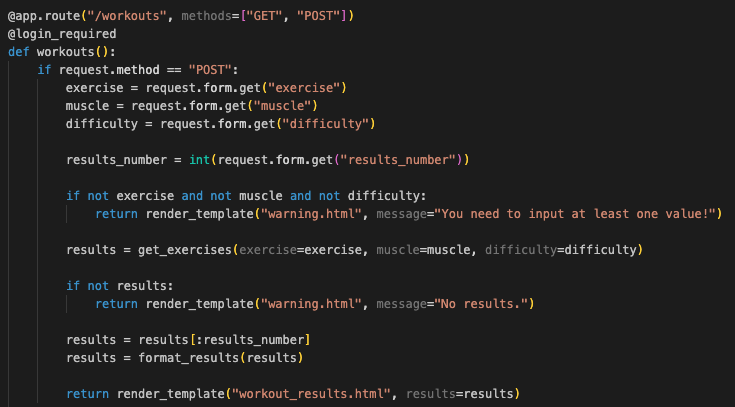
Този път се грижи за показването на регистрационната форма и обработката на данните от нея. Добавят нов потребител в базата данни.

3.2.6 Главна страница

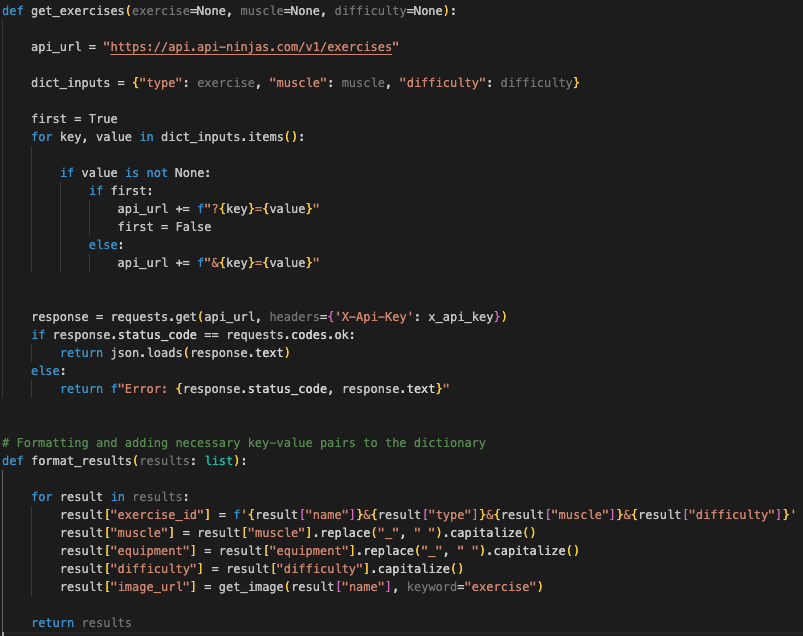


Показва главната страница за текущия потребител.

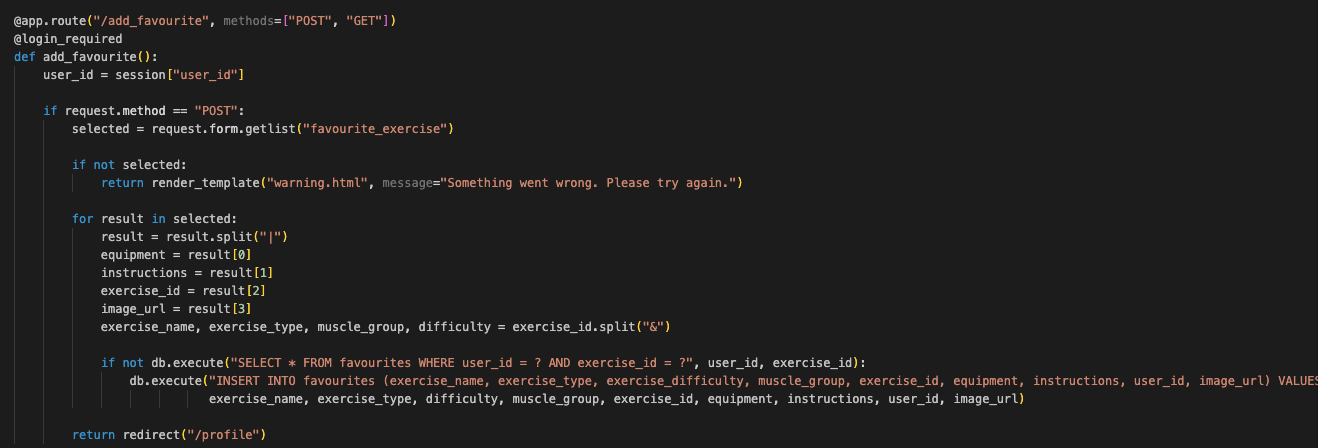
3.2.7 Търсене на тренировка



Показва формата за търсене на тренировка и обработва данните от нея. Препраща към страницата с резултатите от търсенето, като й подава резултатите от търсенето, които да бъдат показани (ако има такива). Логиката на самото търсене е изнесена в отделна функция във functions.py, както е показано по-долу:

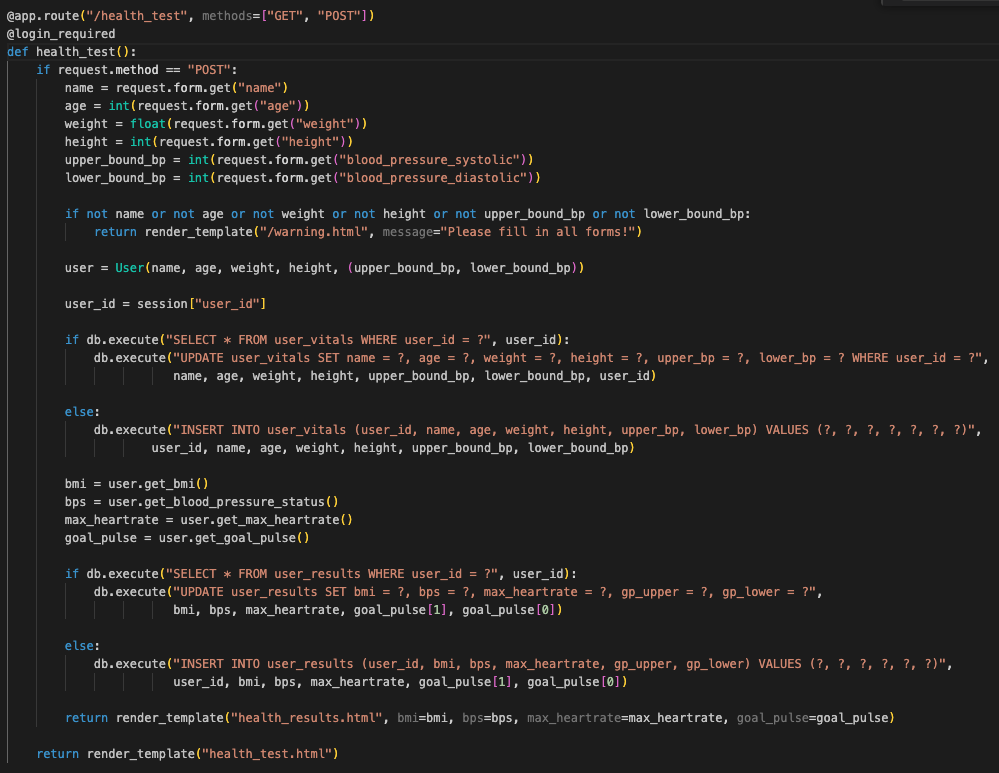


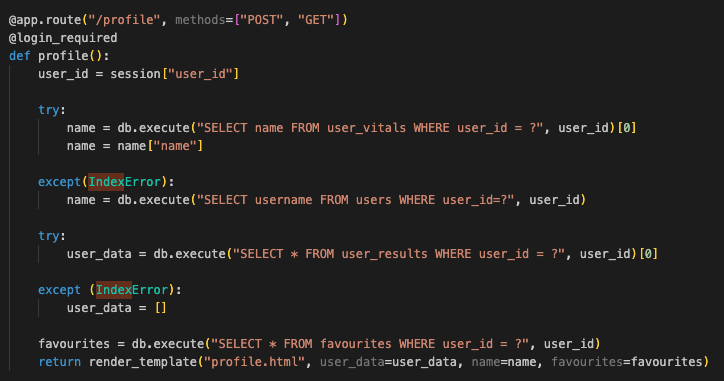
3.2.8 Добавяне на упражнение в любими

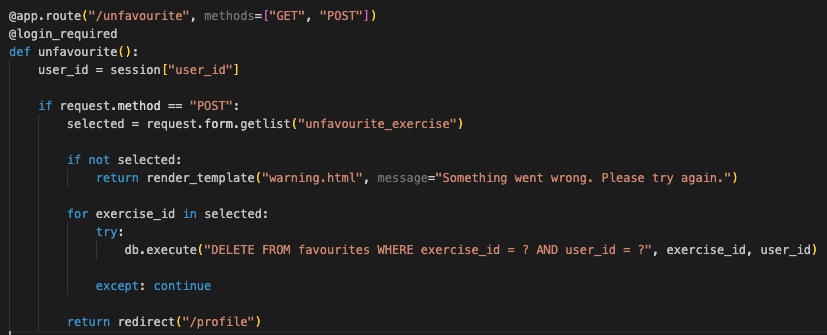


Тази част от кода показва логиката за добавяне на упражнение в любими – ако потребителя се опита да добави без да е отбелязал упражнение, той бива препратен към екран, който описва грешката, а при отбелязано упражнение, то се запаметява в базата данни чрез db.execute командата.

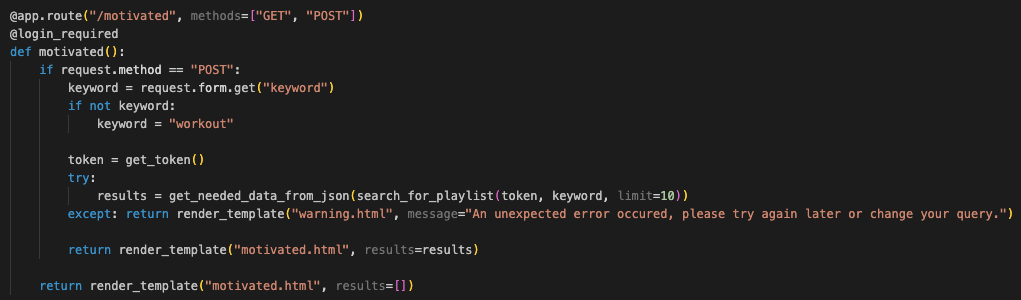
3.2.9 Тест на здравето

  
Тази функция взима подадените данни от HTML-а и от тях създава клас User, който съдържа всички тестове, резултатите от които ще бъдат показани. Данните за здравните показания на потребителя се записват в базата данни, а резултатите от тестовете се подават на health\_results.html, където потребителят бива препратен и където ще му бъдат представени.

3.2.10 Профил  
  


Тази страница показва информация за потребителя, като име и здравословно състояние. В случай, че тези данни не са налични, се показва различно съобщение. Премахването от любими става с помощта на долупосочената функция:  
  


3.2.11 Музикални плейлисти от Spotify



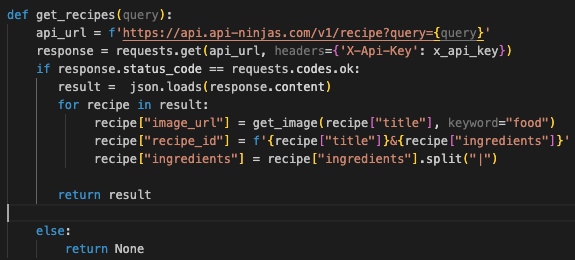
Този код извиква Spotify API с подадена от потребителя ключова дума, като при липса на такава, се използва “workout”. Самата функционалност се намира в functions.py файла, както е показано по-долу:



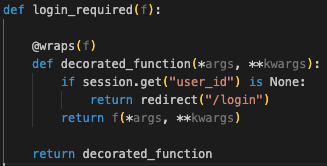
3.2.12 Търсене на рецепти



В тази част от приложението потребителят подава ключова дума за търсене на рецепти и получава резултати обратно. Логиката на самото търсене е изнесена в отделна функция във functions.py, както е показано по-долу:

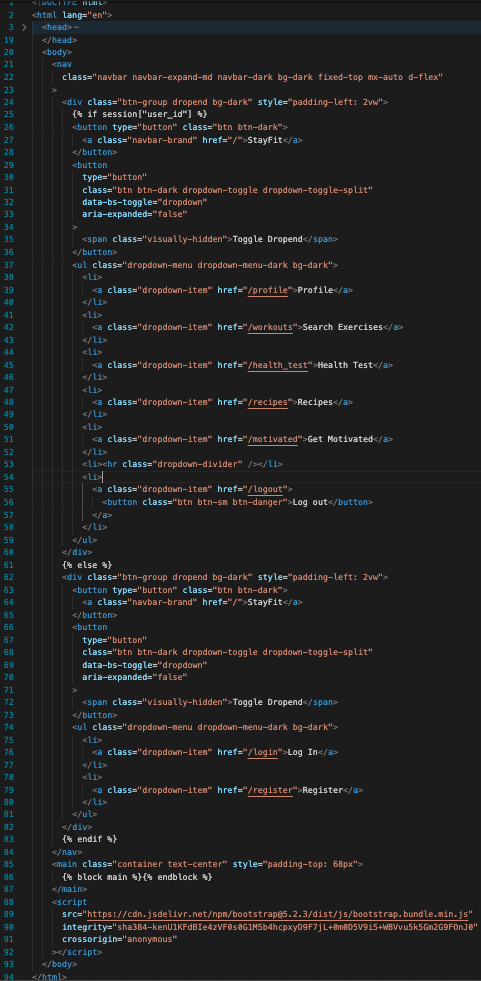


3.2.13 Декоратор, който подсигурява, че потребителят е логнат преди да извърши даденото действие



**3.3 HTML фйалове**

3.3.1 layout.html



Този документ съдържа основния външен вид на приложението, като останалите документи просто го разширяват. Тук и при всички други се използва Jinja, за да се „подпъхне“ необходимата информация в HTML-а, както и да се преизползва обща функционалност.

3.3.2 login.html



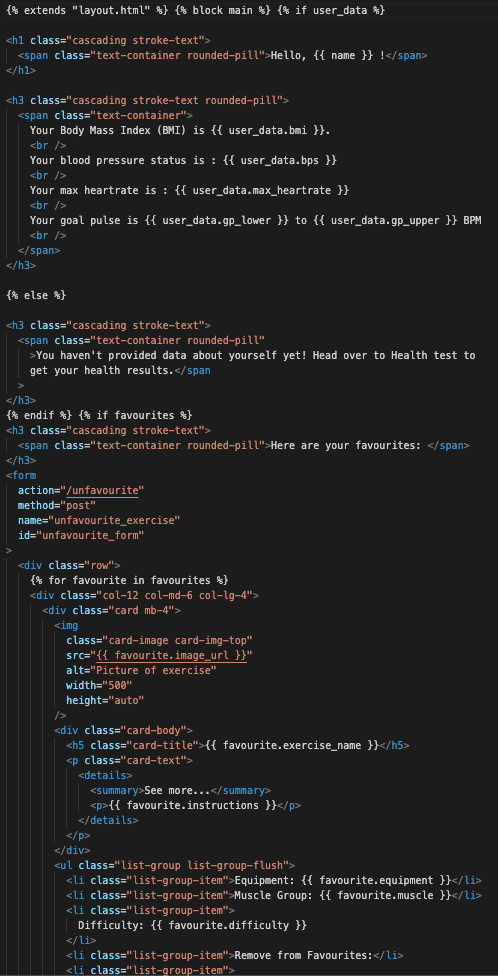
Този HTML файл създава страница за вход (логин), която включва формуляр за въвеждане на потребителско име и парола. Когато потребителят попълни формата и натисне бутона "Login", данните се изпращат на сървъра чрез POST заявка към /login, където се обработват от сървърната логика на приложението. Страницата също така предлага линк за регистрация на потребители, които все още нямат акаунт.

3.3.3 motivated.html



Този HTML предоставя на потребителя поле, в което да въведе ключова дума, по която да се намери музикален плейлист, като след това го представя като резултат. Плейлистите са форматирани като картички, през които може да се скролва.

3.3.4 profile.html



Този HTML представя на потребителя неговите данни за здравето и любими тренировки, представени отново под формата на картички.

3.3.5 recipes.html



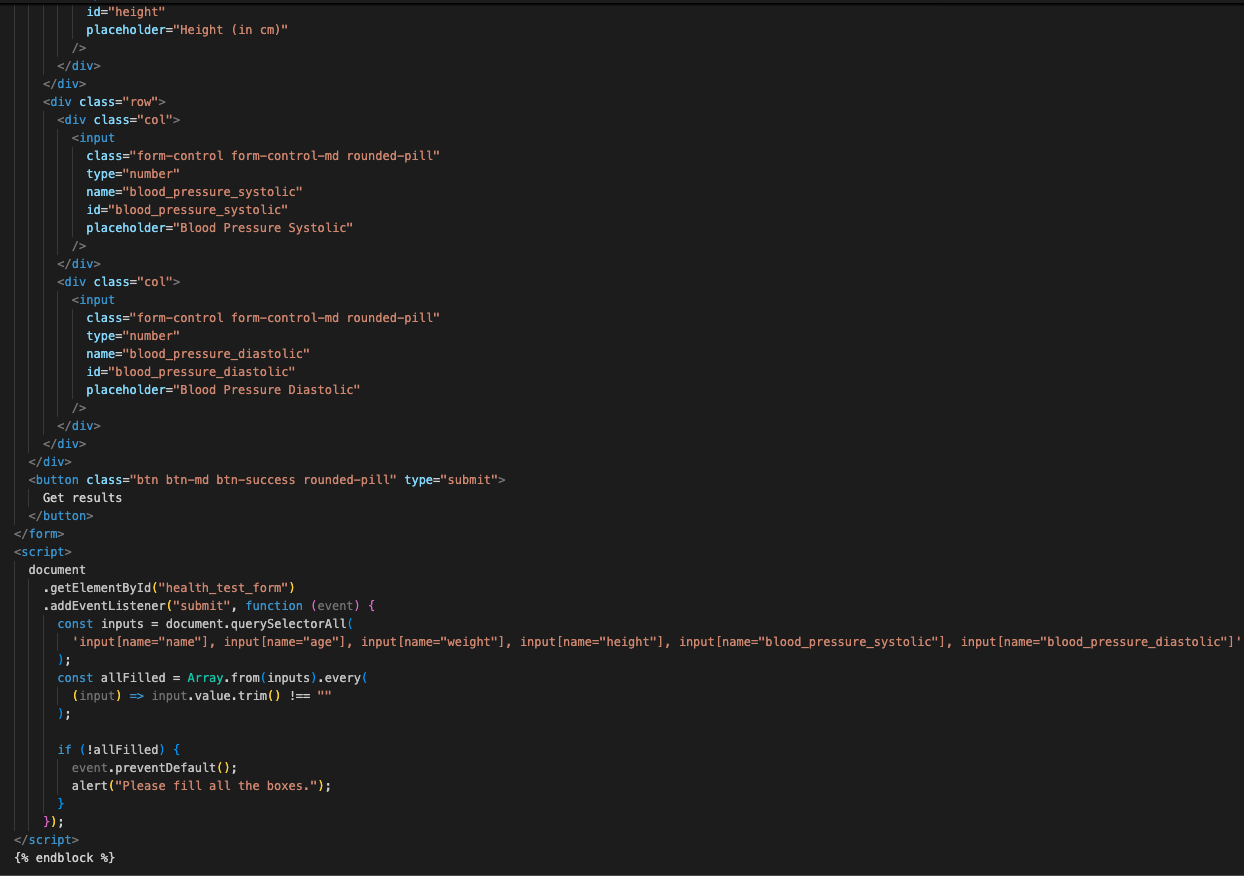
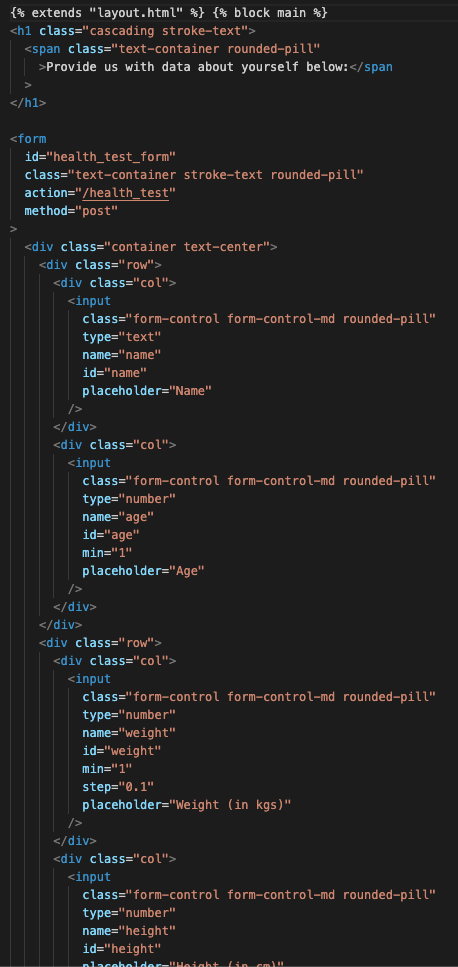
Този HTML файл създава страница, която позволява на потребителите да търсят здравословни рецепти за храна и представя получената информация под формат на карти.

3.3.6 register.html



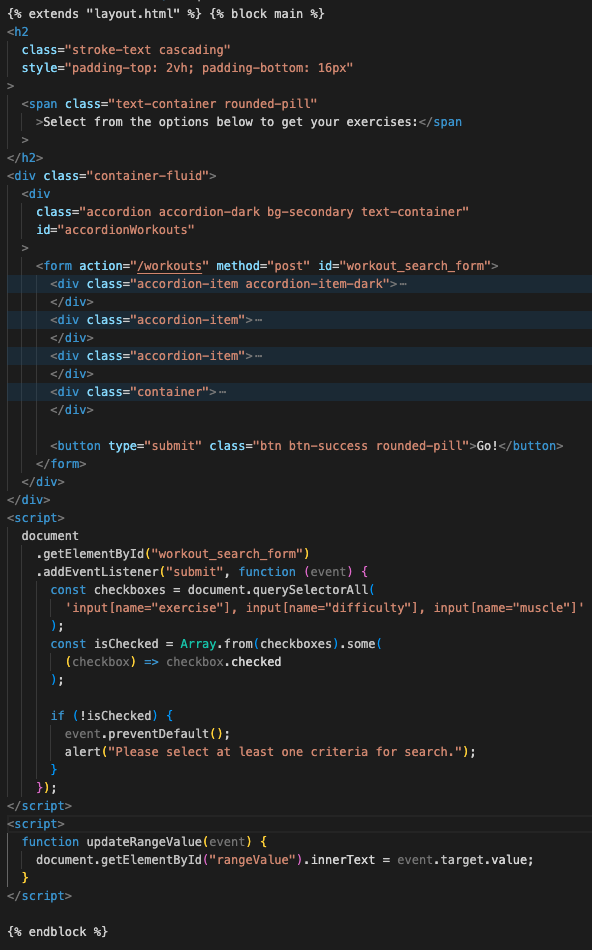
Този HTML файл създава страница за регистрация, която включва формуляр за въвеждане на потребителско име, парола, потвърждение на паролата и имейл адрес. Когато потребителят попълни формата и натисне бутона "Go!", данните се изпращат на сървъра чрез POST заявка към /register, където се обработват от сървърната логика на приложението. Страницата също така предлага линк за вход към потребители, които вече имат акаунт.

3.3.7 health\_test.html



Този HTML файл отговаря за приемането на данни от потребителя, свързани с неговото тяло и различни здравни параметри и му връща резултати, които отговарят на текущото му здравословно състояние. Резултатите се записват в базата данни и се показват при всяко отваряне на профила. Всеки следващ тест презаписва предишните данни.

3.3.8 workouts.html

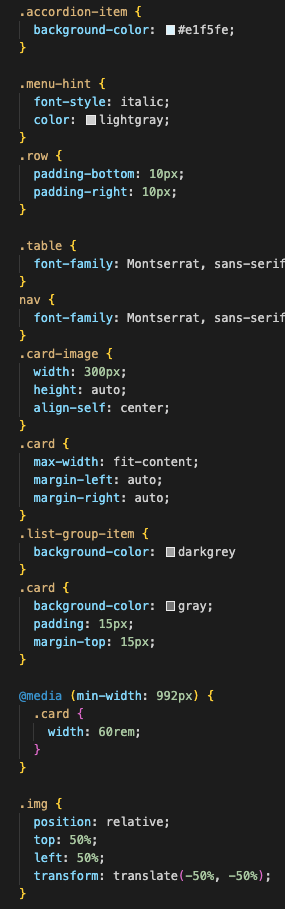


Този HTML предоставя интерфейса за търсене на упражнения, Потребителят избира тип упражнение, мускулна група и брой упражнения. Първите два критерия са представени чрез бутони тип „радио“, а броят упражнения се избира чрез плъзгач. При натискане на „Go!“, се изпраща заявка към външно API, като резултата се обработва от сървърната логика на приложението и се връща на потребителя под форма на карти, които той може да добави към своите любими в профила.

**3.4 CSS**

****

****



CSS файлът доставя форматирането на нашия сайт и подсигурява, че той ще изглежда добре както на компютри, така и на мобилни устройства.

**4. Ръководство на потребителя**

**4.1 Необходими инструменти**

* Code editor (VSCode, Pycharm, т.н.)
* Flask
* Последна версия на Python

**4.2 Инструкции за ползване**

**4.2.1 Регистрация и логин**

Когато потребителят успешно стартира проекта трябва да влезе в акаунта си, като натисне бутона “Log In”. Ако няма акаунт, трябва да си направи. Това става, когато натисне бутона “Sign Up”, въведе име, e-mail, парола и след това потвърди паролата си.

**4.2.2 Търсене на тренировка**

Потребителят може да намери упражнение като въведе неговия тип, мускулна група и брой упражнения, които желае да получи, и след което натисне “Go!”.

**4.2.3. Любими**

Потребителят може да добавя и премахва намерени упражнения от любимите си.

**4.2.4 Проверка на здравословно състояние**

Потребителят може да извърши проверка на здравословното си състояние като въведе данни за своите възраст, кръвно налягане и физически параметри, и след което натисне “Go!”.

**4.2.5 Търсене на рецепти**

Потребителят може да намери рецепти за приготвяне на храна като въведе ключова дума, и след което натисне “Go!”.

**4.2.6 Търсене на музикални плейлисти в Spotify**

Потребителят може да намери музикални плейлисти в даден стил като въведе ключова дума, и след което натисне “Go!”.

**5. Заключение**

**5.1 Обобщение**

Проектът прави търсенето на упражнения, здравословни рецепти, музикални плейлисти за тренировка и измерването на здравословното състояние по-лесно.

**5.2 Усъвършенстване**

Проектът може бъде усъвършенстван като се добавят функционалности от сорта на: запазване на рецепти в любими, създаване на отделни тренировки от намерените упражнения, запазване на плейлисти в любими, подобрен потребителски интерфейс, добавяне на ORM за по-лесно надграждане на back-end логиката и други.

**6. Използвана литература**

**Flask Documentation**: Официалната документация на Flask за ръководство и примери.

* Сайт: [Flask documentation](https://flask.palletsprojects.com/en/stable/)

**CS50 SQLite Documentation**: Официалната документация на **CS50 SQLite** за работа с бази данни в Flask.

* Сайт: [CS50 SQLite documentation](https://cs50.readthedocs.io/library/python/)

**HTML and CSS Tutorials**: Различни онлайн ресурси и уроци за уеб дизайн и стилизиране.

* Сайтове: W3Schools, MDN Web Docs, CSS-Tricks

**Python Documentation**: Официалната документация на Python за справка по езика и вградените библиотеки.

* Сайт: [Python Documentation](https://docs.python.org/)

**Stack Overflow**: Онлайн общност за програмисти, където се споделят примери и решения на проблеми.

* Сайт: [Stack Overflow](https://stackoverflow.com/)

**You Tube**: Туториали по всички използвани езици