**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**КУРСОВ ПРОЕКТ**

Предмет: Въведение в скриптовите езици

Тема: Уеб сайт за проследяване на физическата активност

Ученици: Научен ръководител:

*Денис Димитров Даниел Димитров*

*Деян Нъков*

*Калоян Тодоров*

СОФИЯ

2024

**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 проследяване на фитнес, предназначено да помага на потребителите да записват своите тренировъчни дейности, да наблюдават напредъка си във времето и да предоставят персонализирани препоръки за подобряване на тяхното фитнес преживяване. 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 за проследяване на активността на потребителя в тренировъчна среда и предоставяне на ефективно наблюдението на напредъка.

**Специфични цели:**

* Да се разработи и внедри удобен за потребителя интерфейс за проследяване на различни тренировъчни дейности.
* Да се даде възможност на потребителите да преглеждат напредъка си чрез визуални изображения, като графики и диаграми.
* Да се предоставят препоръки за тренировки въз основа на действащи тренировъчни планове.

**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.

**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.

**Matplotlib**: Библиотека за изчертаване на графики за Python, която ни помогна да генерираме графики и диаграми за визуализиране на напредъка на потребителя при различните упражнения.

**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.
* Matplotlib: Улеснява създаването на подробни и персонализирани визуализации.(в web приложението за създаване на диаграми,следящи прогреса във времето)

**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.

**б)Проследяване на тренировките:**

Потребителите могат да добавят данни за тренировката, включително вид упражнение, килограми, серии и повторения.

Данните се съхраняват в базата данни SQLite.

**в)Визуализация на напредъка:**

Приложението извлича данни за тренировките и генерира визуални отчети за напредъка с помощта на Matplotlib.

Графиките показват тенденциите във времето, като помагат на потребителите да следят прогреса си.

**г)Предложени тренировки:**

За потребителите ние изготвихме едно-седмичен тренировъчен план включващ 5 тренировъчни дни и 2 почивни

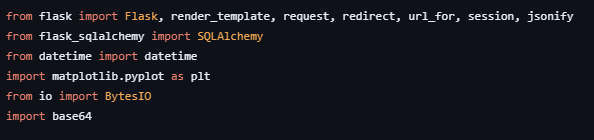
**3.Трета глава: Програмна реализация на проекта**

**3.1 Структура на проекта**

* static/: Съдържа статичният файл styles.css.
* templates/: Съдържа всички HTML файлове.
* app.py: Основният файл съдържащ кодът написан на Python
* my\_body\_tracker.db: Файл съдържащ базите данни

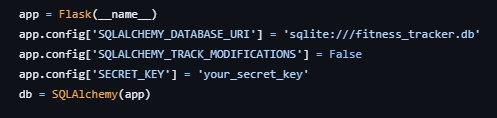
**3.2 app.py**

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

****

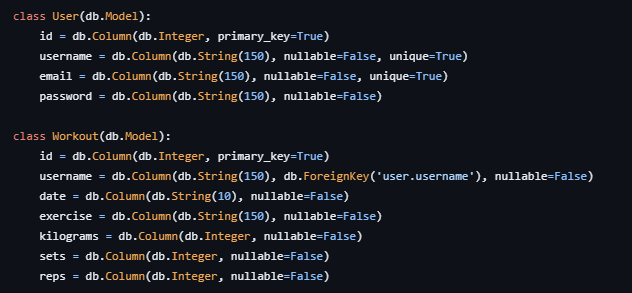
Тук се импортират нужните библиотеки. Flask е уеб рамка, SQLAlchemy е ORM (Object Relational Mapper), datetime за работа с дати, matplotlib за създаване на графики, io за работа с байтове, base64 за кодиране на изображения и webview за показване на уеб интерфейса.

**3.2.2 Конфигурация на Flask приложение**



Създаваме Flask приложение и го конфигурираме. Настройваме база данни SQLite, изключваме проследяването на модификации за да не получаваме предупреждения и задаваме секретен ключ за сесиите.

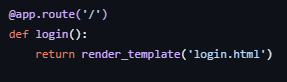
**3.2.3 Модели на базата данни**



Тук дефинираме два модела за базата данни. User моделът съдържа информация за потребителя, а Workout моделът съдържа информация за тренировките.

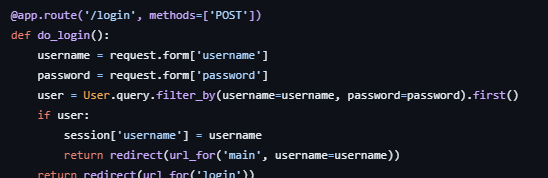
**3.2.4 Пътища за приложението**

**3.2.4 A) Логин страница**



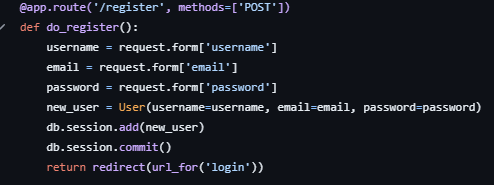
Показва страницата за вход (login.html).

**3.2.4 B) Обработка на логин формата**

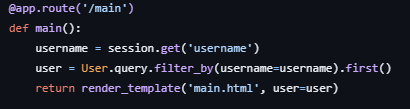
****

Този route 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.

**3.2.4 C) Регистрация на нов потребител**

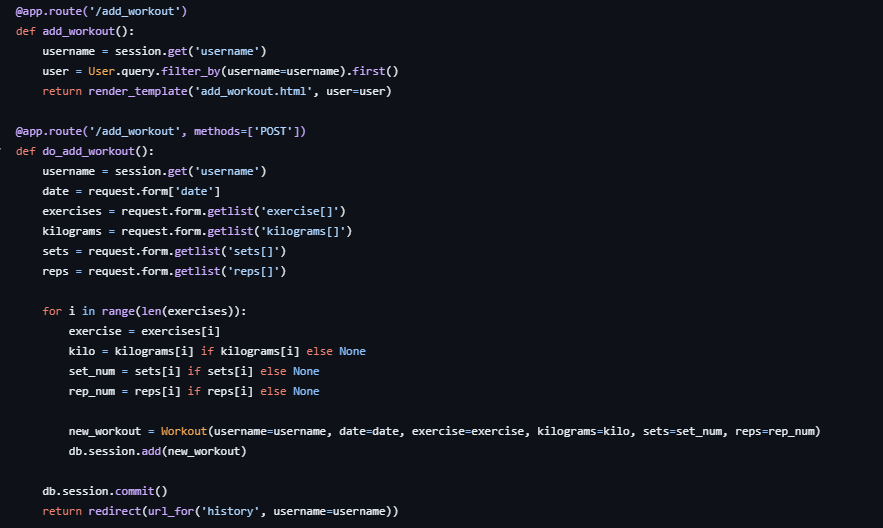


Този route се грижи за показването на регистрационната форма и обработката на данните от нея. Добавят нов потребител в базата данни.

**3.2.4 D) Главна страница**

Показва главната страница за текущия потребител.

**3.2.4 E) Добавяне на тренировка**

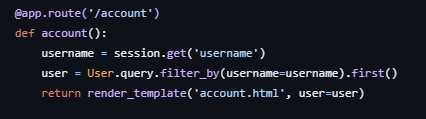
****

Показва формата за добавяне на тренировка и обработва данните от нея. Добавя нова тренировка в базата данни за текущия потребител.

**3.2.4 F) История на тренировките**

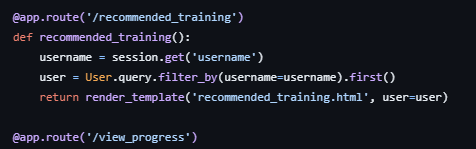
Тази част от кода показва историята на тренировките на текущия потребител и поставя датата в правилния формат за нашия континент.

**3.2.4 G) Страница с информация за акаунта**

****

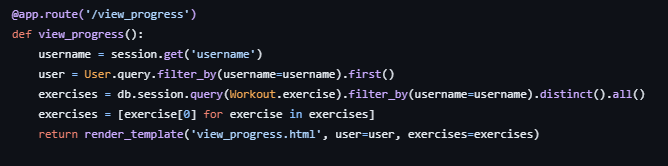
Показва информацията за акаунта на текущия потребител.

**3.2.4 H) Препоръчани тренировки**



Показва препоръчани тренировки за текущия потребител.

**3.2.4 I) Преглед на напредъка**

****

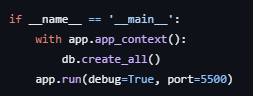
Показва списък с бутони съдържащи различните упражнения в страницата за преглед на прогреса.

**3.2.4 J) Напредък на конкретно упражнение**



Показва графика на напредъка за дадено упражнение.

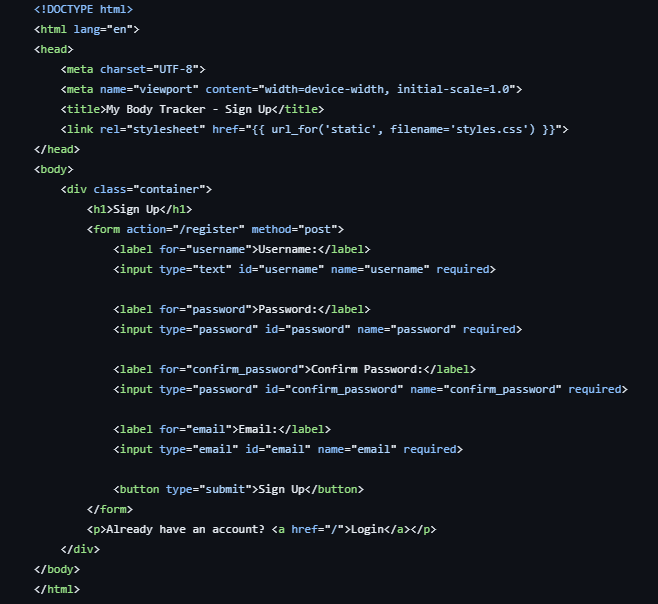
**3.2.4 K) Стартиране на приложението**



Стартира Flask приложението и създава базата данни, ако тя не съществува.

**3.3 HTML files**

**3.3.1 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 "Sign Up", 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.

**3.3.2 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 Main.html**

****

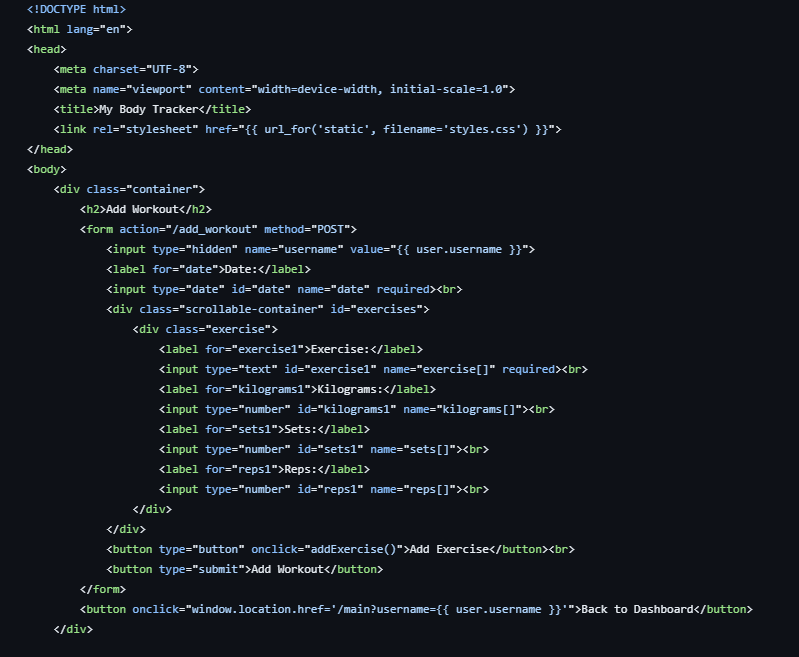
Този HTML файл е главната страница на нашия уеб сайт. Той посреща потребителя по име и предоставя навигационни бутони за достъп до различни функционалности на уеб сайт.

**3.3.4 view\_progress.html**

****

Този HTML файл създава страница, която позволява на потребителите да преглеждат напредъка си в различни упражнения. На страницата се показват бутони за всяко упражнение, което потребителят е изпълнявал. Когато потребителят натисне бутон за конкретно упражнение, той се пренасочва към страница, която показва напредъка му в това упражнение. Има и бутон за връщане към главната страница на приложението.

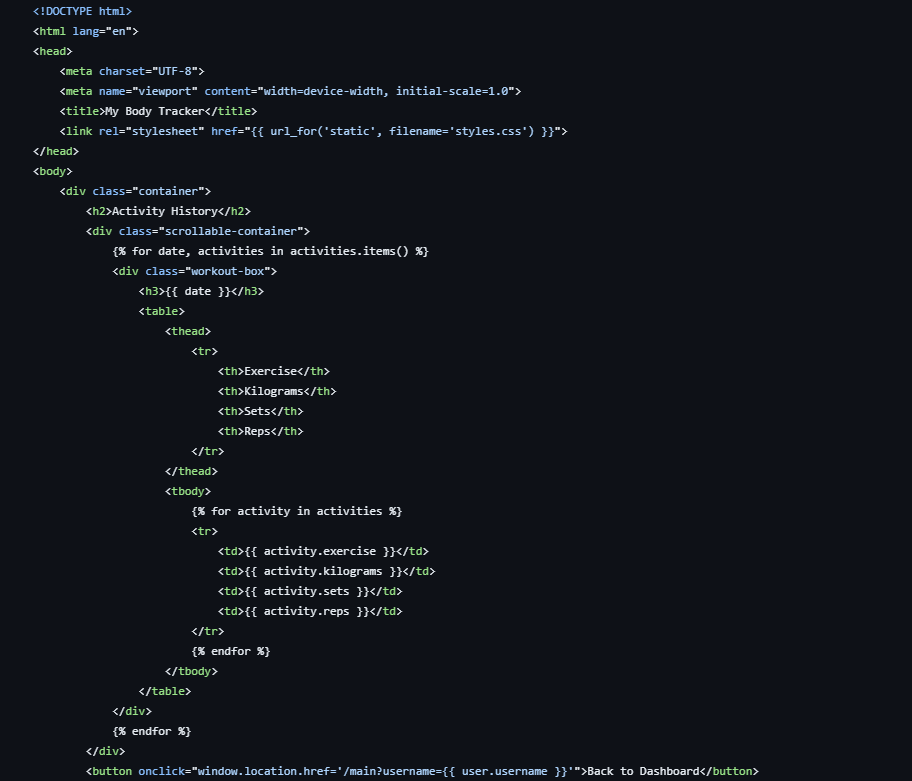
**3.3.5 add\_workout.html**

****

****

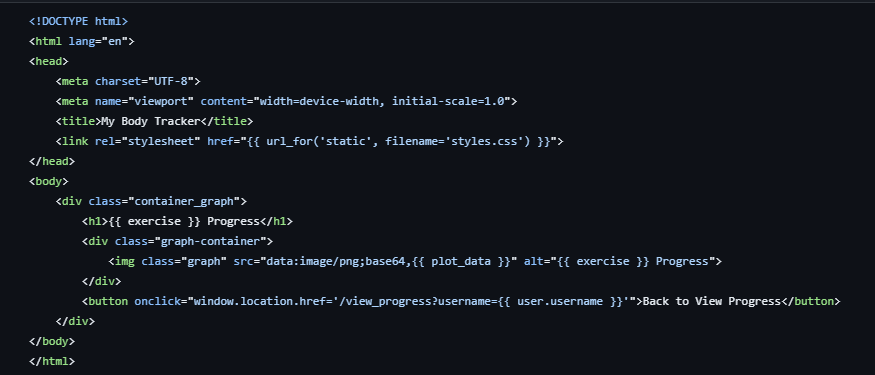
Този HTML файл създава страница за добавяне на тренировки, която включва формуляр за въвеждане на дата и различни упражнения с детайли за килограми, серии и повторения. Потребителите могат да добавят повече упражнения чрез бутон, който динамично добавя нови полета към формата. След попълването на формата, данните се изпращат на сървъра за обработка и съхранение.

**3.3.6 history.html**

****

Този HTML файл създава страница, която показва историята на дейностите на потребителя. За всяка дата, за която има регистрирани дейности, се създава нова секция с таблица, която показва подробности за упражненията, килограмите, сериите и повторенията. Има и бутон за връщане към главната страница на приложението.

**3.3.7 exercise\_progress.html**

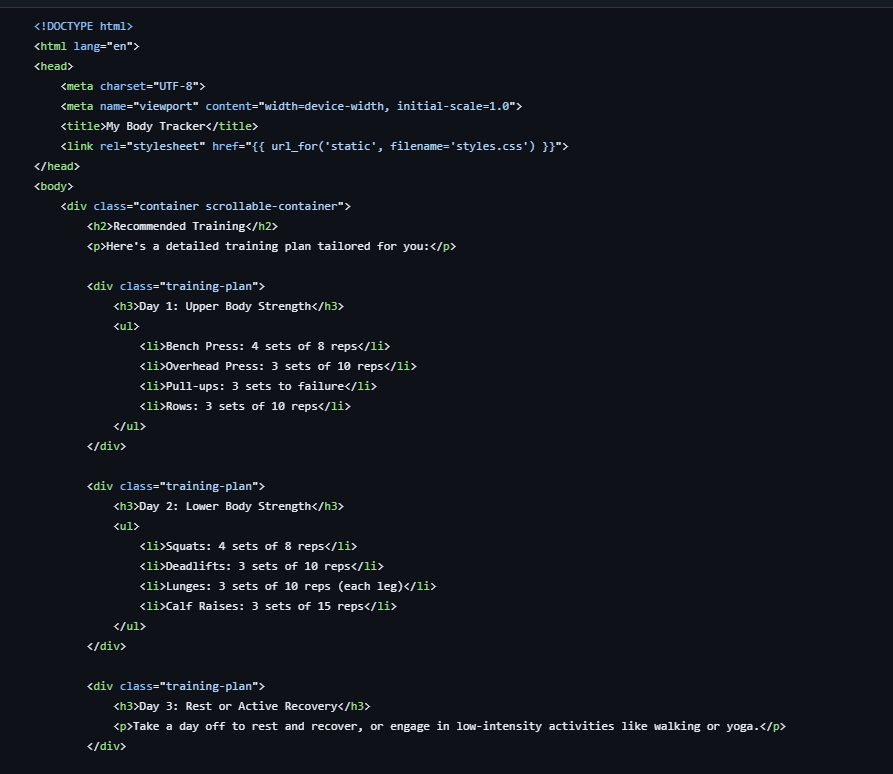
****

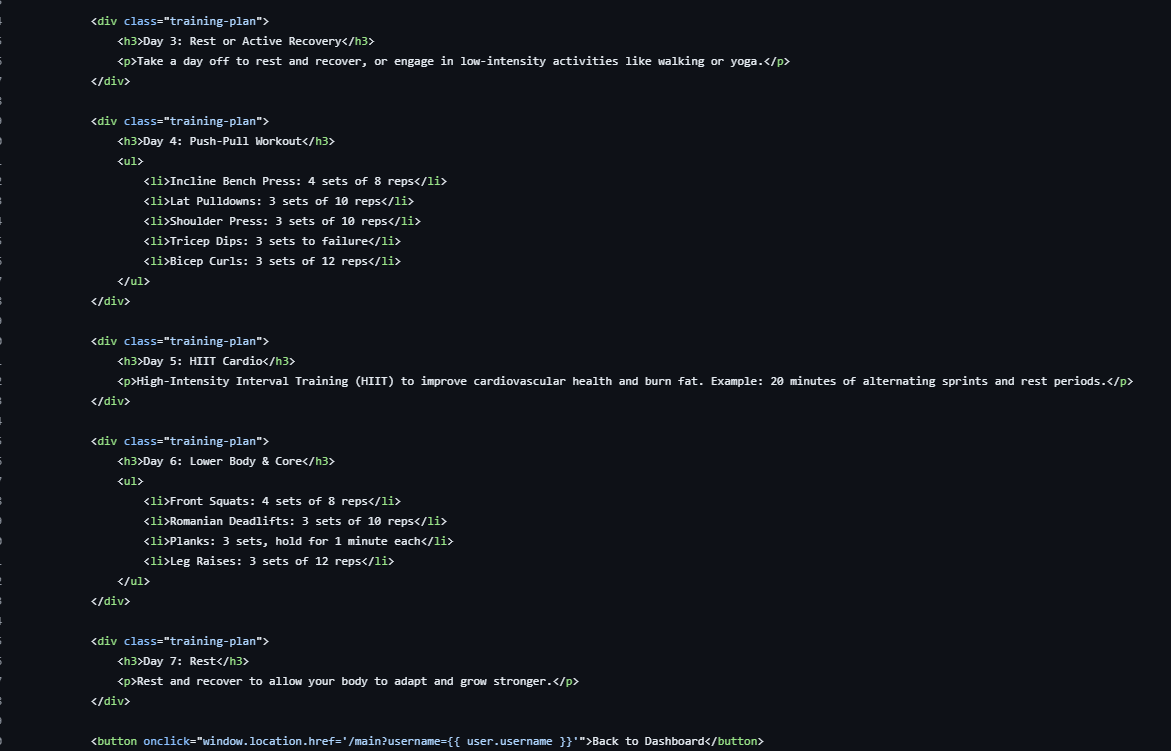
Този HTML файл създава страница, която показва графика на напредъка в определено упражнение. Потребителят вижда заглавие с името на упражнението, графика на напредъка в това упражнение и бутон за връщане към страницата за преглед на напредъка в упражненията.

**3.3.8 account.html**

****

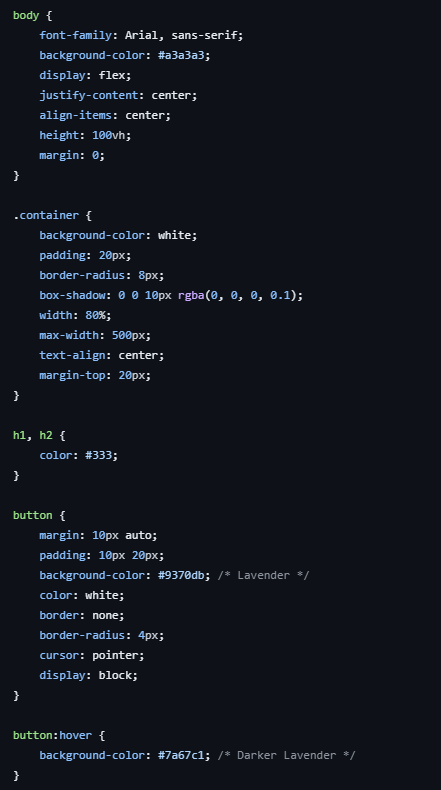
Този HTML файл създава страница, която показва информация за акаунта на потребителя, включително потребителското име и имейл адрес. Потребителят има възможност да се върне към главната страница на приложението, използвайки бутона "Back to Dashboard".

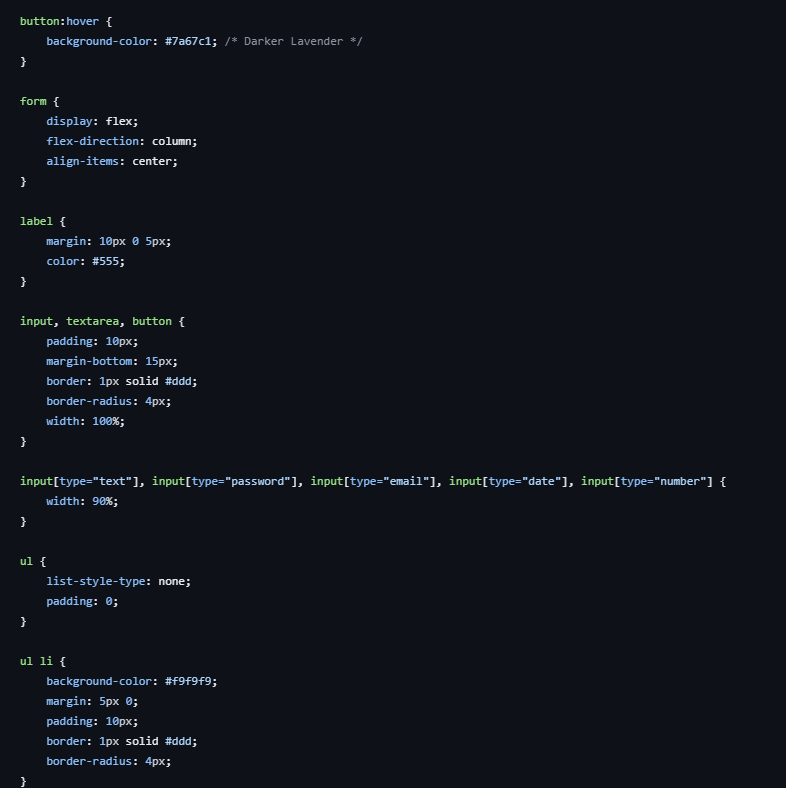
**3.3.9 recommended.html**

****

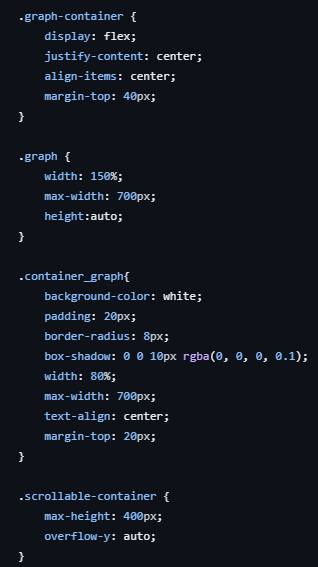
Този HTML файл съдържа детайлна тренировъчна програма, която е предназначена за препоръка за потребителя на приложението "My Body Tracker". Съдържанието е организирано в различни дни на седмицата, всяка със списък от упражнения. Потребителят може да се върне към главната страница на приложението, използвайки бутона за навигация "Back to Dashboard".

**3.4 CSS file**

****

****

****

****

CSS файлът доставя форматирането на нашия сайт и подпомага пр му интерфейс

**4. Четвърта глава: Ръководство на потребителя**

**4.1 Build инструкции**

За да може потребителят да използва “My Body Tracker” са му нужни:

* Code editor
* инсталиран Flask
* Mathplotlib
* Последна версия на Python

**4.2 Инструкции за използване**

**4.2.1 Регистрация и логин**

Когато потребителят успешно включи 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 Добавяне на тренировка**

Потребителят може да добави тренировка, като избере упражнение, дата, килограми, серии и повторения.

**4.2.3 История на активността**

След въвеждане на тренировки в сайта можете да видите историята на своята активност.

**4.2.4 Преглеждане на прогреса**

В сайта потребителят може да види персонализирани графики, които показват прогреса на килограмите, които е използва за всяко отделно упражнение.

**4.2.5 Препоръки**

В сайта може да намерите изготвена от специалисти седем дневна тренировъчна програма, която ще Ви помогне да постигнете желаните резултати.

**4.2.6 Информация за акаунта**

В нашия сайт можете да видите информацията за акаунта(име, имейл).

**5. Пета глава: Заключение**

**5.1 Обобщение**

Проектът прави следенето на прогреса във фитнеса по-лесно.

**5.2 Виждане за усъвършенстване**

Може да усъвършенстваме проекта, като го направим във форма на мобилно приложение. Това ще го направи много по-удобен за използване. Може да направим community, което да споделя тренировки, за да се развиват взаимно. Може да добавим и физически дейности (като бягане, бокс, лостове), което значително ще развие приложението.

**6. Шеста глава: Използвана литература**

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

* Сайт: Flask Documentation

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

* Сайт: SQLAlchemy Documentation

**Matplotlib Documentation:** Официалната документация на Matplotlib за генериране на графики.

* Сайт: [Matplotlib Documentation](https://matplotlib.org/)

**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

**7. Седма глава: Съдържание**

**УВОД**

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

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

**3. Трета глава: Програмна реализация на проекта**

**4. Четвърта глава: Ръководство на потребителя**

**5. Пета глава: Заключение**

**6. Шеста глава: Използвана литература**

**cececccd**