

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мережних технологій факультету інформатики

РОЗРОБКА ВЕБ-СЕРВІСУ ДЛЯ КОНТРОЛЮ ХАРЧУВАННЯ ТА ФІЗИЧНИХ НАВАНТАЖЕНЬ

**Текстова частина до курсової роботи
за спеціальністю «Інженерія програмного забезпечення»**

Керівник курсової роботи

Канд.фіз.-матем. наук

Гречко А.В

Виконала студентка 3 курсу

Жуковська М. С.

Київ 2021

Тема: Розробка веб-сервісу для контролю харчування та фізичних навантажень

Календарний план виконання роботи:

№	Назва етапу	Термін виконання	Примітка
1.	Отримання теми курсової роботи	Жовтень 2021р.	
2.	Пошук тематичної літератури	Березень-квітень 2021р.	
3.	Ознайомлення з літературою	Березень-квітень 2021р.	
4.	Ознайомлення з існуючими системами-аналогами роботи	Березень-квітень 2021р.	
5.	Планування веб-сайту	Березень-квітень 2021р.	
10.	Розробка функціоналу проекту	Травень 2021р.	
12.	Додавання стилів до сторінок	Травень 2021р.	
13.	Написання текстової частини	Травень 2021р.	
16.	Створення презентації	Травень 2021р.	
18.	Захист роботи		

Студентка Жуковська М.С.

Керівник Гречко А.В.

“ ”

Перелік умовних позначень	4
ВСТУП	5
РОЗДІЛ 1: Аналіз предметної області.....	7
1.1 Аналіз сучасного стану питання та обґрунтування теми.....	7
1.2. Огляд існуючих аналогів розробки.....	8
1.3. Постановка завдання	11
РОЗДІЛ 2: Теоретичні відомості	12
2.1. Основні принципи розробки веб-застосунків.....	12
2.2. Класифікація сайтів	15
РОЗДІЛ 3: Опис реалізації програмного продукту.....	18
3.1 Аналіз технічного завдання	18
3.2. Обґрунтування вибору засобів розробки.....	19
3.3 Опис розробки програм.....	21
3.4 Опис файлів даних та інтерфейсу програми	22
3.5. Тестування програми і результати її виконання.....	24
Висновки	29
Список використаної літератури	30

Перелік умовних позначень

1. Валідація – процес перевірки інформації та її підтвердження на відповідність критеріям конкретного користувача, продукту, послуги або системи.
2. Шаблонізатор – програмне забезпечення, яке підключає файл шаблону, передає йому дані і повертає згенерований HTML.
3. HTTP-протокол (Hyper Text Transfer Protocol) – це протокол, який дозволяє отримувати різні ресурси, наприклад HTML-документи.
4. БД – база даних.

ВСТУП

В сучасному світі Інтернет є важливим та потрібним джерелом інформації для більшої частини суспільства. За допомогою веб-сайтів людина може отримати доступ для актуальних новин та тем, які її турбують. Різноманітні веб-платформи та сервіси надають корисну інформацію щодо інтересів та захоплень, групують однодумців та сприяють створенню тематичних клубів та співтовариств. Особливо актуальність веб-сервісів зросла під час карантинних умов, оскільки майже все спілкування перейшло в онлайн-сферу, це збільшило потреби в ній.

З кожним роком мода на здоровий спосіб життя набирає обертів, люди починають все більше осмислювати своє ставлення до свого тіла, здоров'я та самопочуття. Великий вплив також має карантин та постійне знаходження вдома, недоступність спортивних клубів, фітнес-центрів та обмеження щодо епідемії є причиною не тільки недостатньої рухливості людей, але й їх поганого психологічного та ментального стану. На сучасному українському ринку існує не так багато веб-сайтів, які надають доступ до тренувань, основана маса таких застосунків зосереджена у англомовній спільноті. Таким чином актуальність дослідження питання створення фітнес-сервісів та розробка аналогічних продуктів зумовлена безперервним збільшенням числа активних користувачів мережі Інтернет та корисністю таких ресурсів.

Метою роботи є створення веб-сервісу для контролю харчування та фізичних навантажень для полегшення доступу з дома.

Завдання: Розробити онлайн - платформу, створити систему керування цією платформою, створити базу даних фізичних вправ та програм, створити можливість слідкувати за своїм харчуванням, та створити невеликий блог для обміну інформації.

Об'єктом дослідження є створення веб-сервісу для контролю харчування та фізичних навантажень.

Предметом дослідження є огляд існуючих додатків та веб-платформ для фітнесу.

Веб-застосунок розроблений на мові Python за допомогою фреймворку Django. Для збереження даних використано базу даних SQLite. Середовище розробки – PyCharm від JetBrains.

У навчальних цілях було вивчено багато Інтернет-ресурсів, враховуючи різноманітні туторіали та навчальні відеороліки, наукові статті, офіційну документацію та книги.

Курсова робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

РОЗДІЛ 1: Аналіз предметної області

1.1 Аналіз сучасного стану питання та обґрунтування теми

Інтернет стає все більш важливою частиною нашого повсякденного життя зі своїм зростаючим впливом як на окремих споживачів, так і на великі економіки. Згідно статистиці платформи бізнес-даних «Statista» кількість користувачів, які наразі підключені до всесвітньої мережі, становить майже 4 мільярди — це більше половини населення нашої планети. У 2021 році глобальний трафік збільшився на 39,7 відсотків порівняно з 2020 роком, це говорить про невпинно зростаючий попит на онлайн-послуги у всьому світі.

Пандемія спричинила закриття фітнес-залів та майданчиків, що відобразилося на притоку людей до «онлайн-залів» та створення фітнес-клубами своїх онлайн-тренерів. В Україні прибуток від таких додатків збільшився на 36%, люди почали частіше займатися спортом в домашніх умовах. Ринок онлайн-фітнесу, за оцінками представників сервісу, буде рости щорічно на 15-20% і тим самим сповільнювати зростання оффлайн-сегмента, який в останні роки становив не більше 2-3%. Навіть вихід спорт-залів з карантину не гарантує повного відновлення бізнесу, адже залишатимуться питання з орендодавцями та співробітниками, це все буде великим навантаженням для оффлайн-сфери. Велика кількість банкрутств, особливо маленьких клубів, відсрочки та кредити. А на виході з пандемії обсяг валової виручки навряд чи можна буде порівняти з минулим роком. Таким чином, коронавірус підштовхнув населення до більшого занепокоєння своїм здоров'ям, і всі більше споживачів та власників бізнесів розуміють зручність використання онлайн-тренувань. Отже, інформаційна система, присвячена для упорядкування даних щодо харчування та тренувань та яка поєднує в собі основні аспекти ведення здорового життя —

актуальна та перспективна тема в сучасних реаліях, яка привертає до себе увагу спонсорів та нових користувачів.

1.2.Огляд існуючих аналогів розробки

На ринку представлено дуже багато саме додатків для телефонів, які можна скачати з «Google Play», «App Store», «Galaxy Store» і т.д. Вони дозволяють відслідковувати фізичні навантаження та дають можливість контролювати свій раціон, але по-перше, україномовних застосунків дуже мало, по-друге, веб-сайти мають свої переваги відносно подібних додатків. Тому україномовних веб-сервісів по цій тематиці вкрай мало, і більшість з них є сайтами зачинених фітнес-клубів, які в умовах карантину намагаються скоріше пристосувати свої системи до онлайн послуг.

Fitness Blender — найбільший англomовний фітнес-канал на YouTube т, який зібрав 6 мільйонів підписників та 1 мільярд переглядів та згодом перетворився у веб-сервіс. Він має понад 600 відеороликів, що пропонують тренування від пілатесу та йоги до силових тренувань. На платформі представлені відео з вправами, програми занять та харчування, блог на відповідну тематику та велике товариство однодумців. Основними перевагами Fitness Blender є велике різноманіття та доступність на безкоштовній основі більшості тренувань та увага до своєї аудиторії. З мінусів можна виділити велику кількість інформації та не завжди зрозумілий інтерфейс.

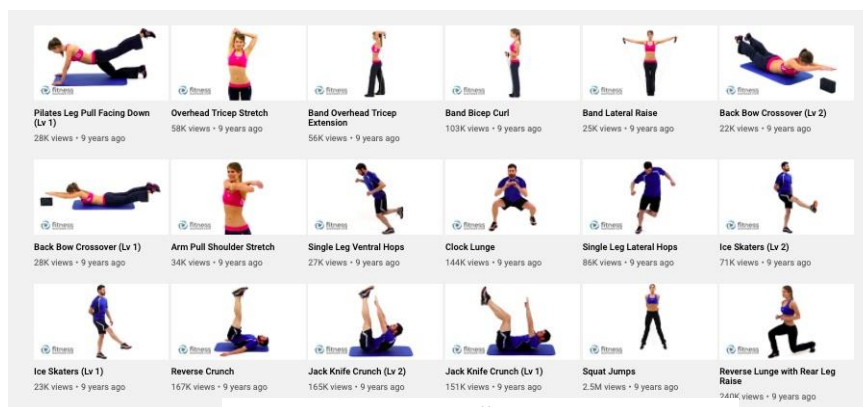


Рисунок 1 - Інтерфейс Fitness Blender

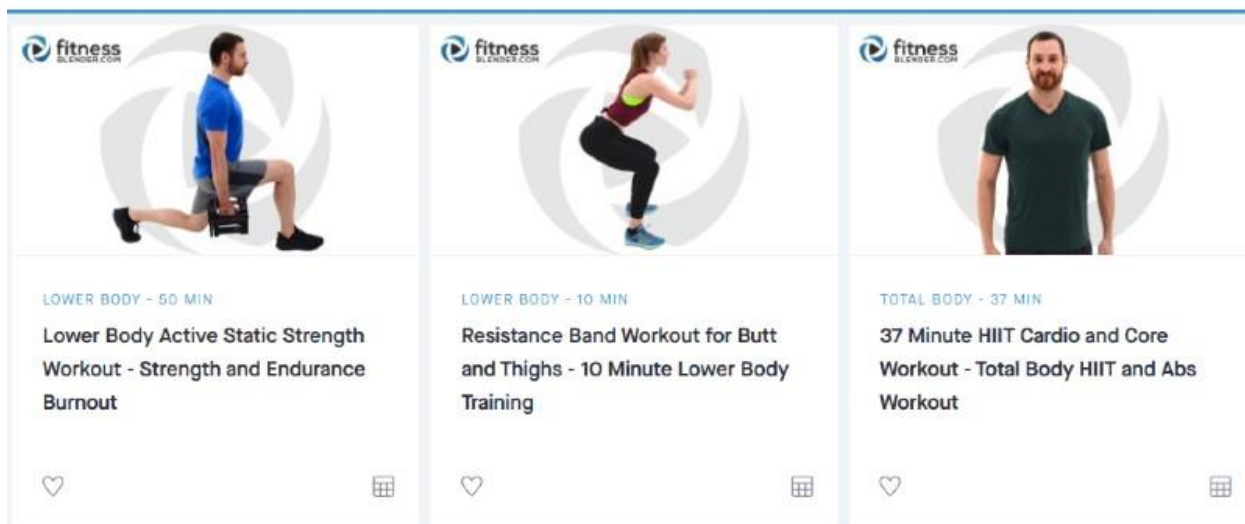


Рисунок 2 - Інтерфейс Fitness

Free Trainers — веб-платформа з онлайн заняттями з фітнесу, має додаток на IOS та Android. Сайтом користується понад 2 мільйона людей. Інтерфейс сайту є інтуїтивно-зрозумілим та мінімалістичним. Плюси: види тренувань для різних цілей, підбирання вправ під конкретну ціль, можливість записувати тренування до календаря, наявність груп та блогів по тематиці сайту, можливість зареєструватися та зберігати свій результат. Недоліком є відсутність будь-яких тем та функціоналу, які пов'язані з харчуванням та раціоном, також сайт повністю англomовний.

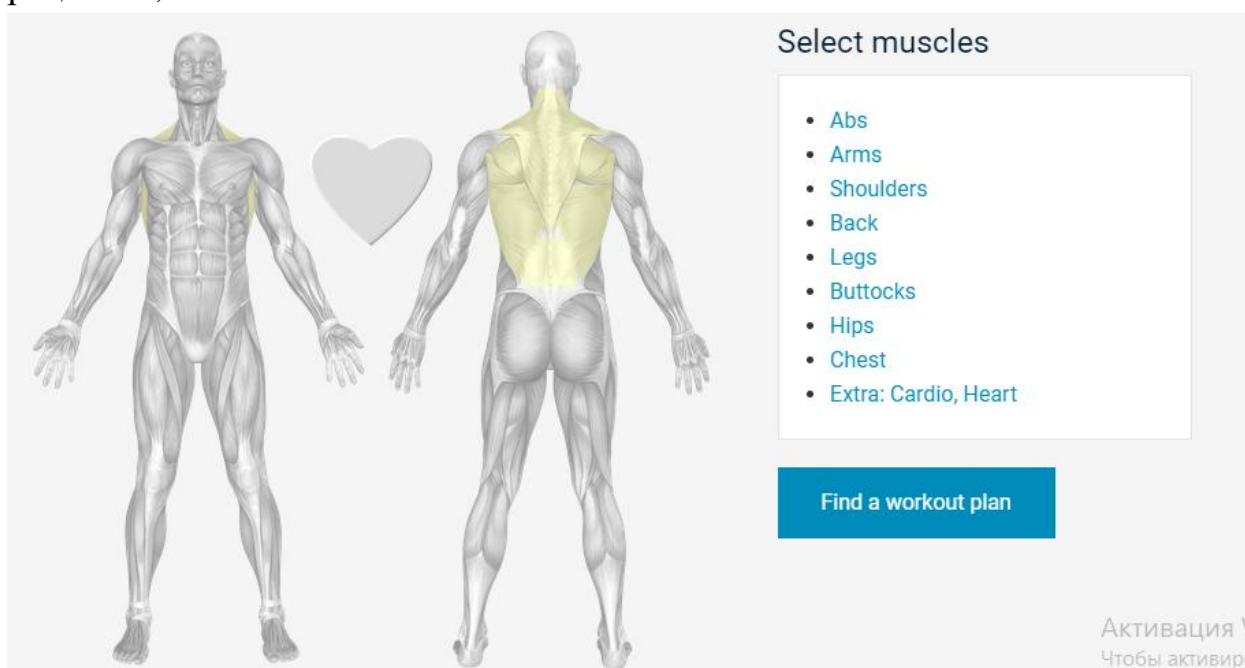


Рисунок 3 - Інтерфейс Free Trainers

Wger Workout Manager — веб-сайт з відкритим кодом, написаний на Python з використанням Django. Він створений для сумісного покращення волонтерами. Має розділи з тренуваннями, програмами тренувань та можливістю обрахувати масу тіла, контролювати свій раціон. Є можливість обрати вправи за категоріями. На відміну від попередніх платформ підтримує українську мову. Має можливість авторизації та створення свого приватного кабінету. Основним плюсом цієї платформи я вбачаю заохочення усіх бажаючих до її розвитку та розбудови. З мінусів відзначаю відсутність блогу або груп для спілкування на сайті.

Панель

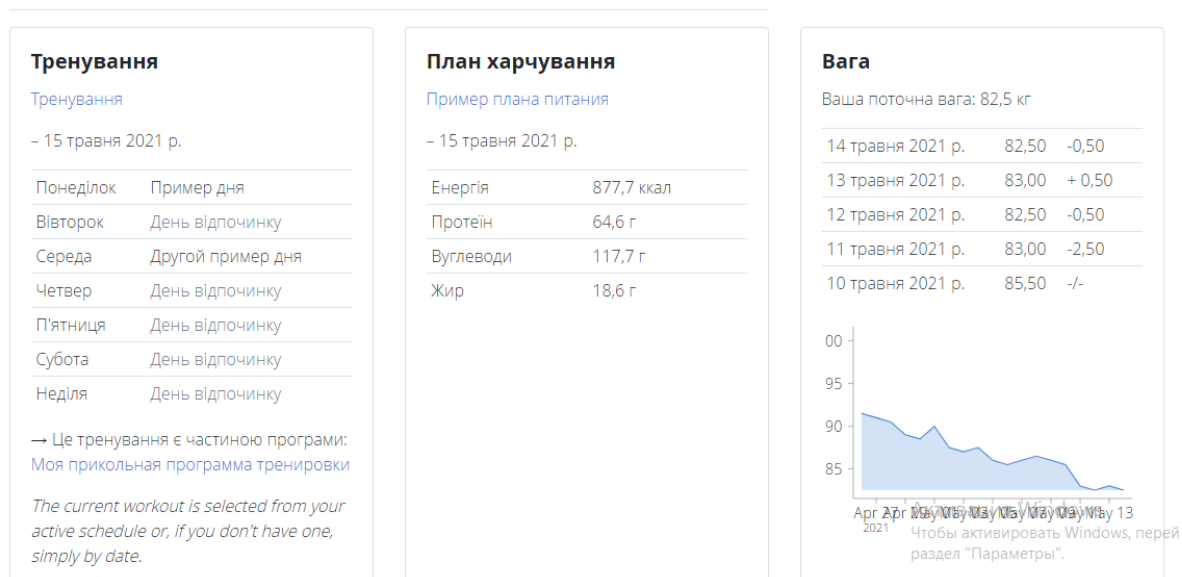


Рисунок 4 - Інтерфейс Wger

The interface shows a detailed view of a workout example for Monday:

- Пример дня (Example Day):** Monday.
 - #1: Biceps Curls With Dumbbell (4 x 8). Description: Fluid movements with no pauses at the top or the bottom.
 - #2: French Press (skullcrusher) SZ-bar (4 x 8). Description: Don't move the elbows during the exercise.
 - #3: Squats (4 x 10). Description: Make a pause at the bottom of 1sec, at the top, 3. Go down till your thighs are parallel to the floor.
- Треновані м'язи (Trained Muscles):** Shows two anatomical diagrams of a human body. The first diagram highlights the biceps and triceps muscles in red, and the second diagram highlights the quadriceps and gluteal muscles in red. A list of instructions is provided:
 1. додати тренувальні дні
 2. Додати набори підходів до тренувальних днів
 3. Встановити повторення для кожного підходу

Рисунок 5 - Інтерфейс Wger

1.3. Постановка завдання

Як було зазначено вище, основна частина онлайн-ресурсів з теми фітнесу та харчування є англомовною, україномовних же сервісів не так багато і більша частка припадає на додатки для мобільних пристроїв. Ідея моєї курсової роботи - створити таку платформу, яка була б популярною серед українських інтернет-користувачів та полегшила б для них процес пошуку тренувань та контроль за харчуванням.

Обрано наступні функції, які мають бути реалізовані на веб-платформі:

- Розробити інтуїтивно-зрозумілий веб-інтерфейс.
- Автентифікація та реєстрація користувача.
- Перегляд тренувань та програм.
- Доступ до тренувань без реєстрації на сайті, а також забезпечення можливості додання коментарів до тренувань та постів у блогу.
- Форма контактного зв'язку.
- Створення, редагування, видалення, схвалення статей.
- Перегляд та зміна інформації про вправи, програми, статті та користувачів (для адміністратора).

РОЗДІЛ 2: Теоретичні відомості

2.1. Основні принципи розробки веб-застосувань

MVC (Model-View-Controller) — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення. Він розділяє логіку проекту на три частини:

- Model (модель). Отримує дані від контролера, звертається до баз даних, представляє внутрішній устрій системи, виконує необхідні операції і передає їх в вид/уявлення. Не взаємодіє з користувачем напряду.
- View (вид або уявлення). Отримує дані від моделі і виводить їх для користувача, відповідає за зовнішній вигляд додатку.
- Controller (контролер). Зв'язує модель та вид, обробляє дії та отримує відповідь від користувача, перевіряє отримані дані і передає їх моделі, результат передає у вид. Як правило, на рівні контролера здійснюється авторизація і фільтрація отриманих даних - перевіряються права користувача на отримання інформації або виконання дій.

Отже, MVC допомагає відокремлювати візуалізацію, логіку проекту та взаємодію з користувачем. Зміна інтерфейсу не впливає на логіку проекту, а зміна логіки в свою чергу не впливає на інтерфейс. Це сприяє створенню більш структурованого коду, дозволяє працювати над проектом більш спеціалізованим людям, спрощує підтримку коду, робить його більш логічним і зрозумілим.

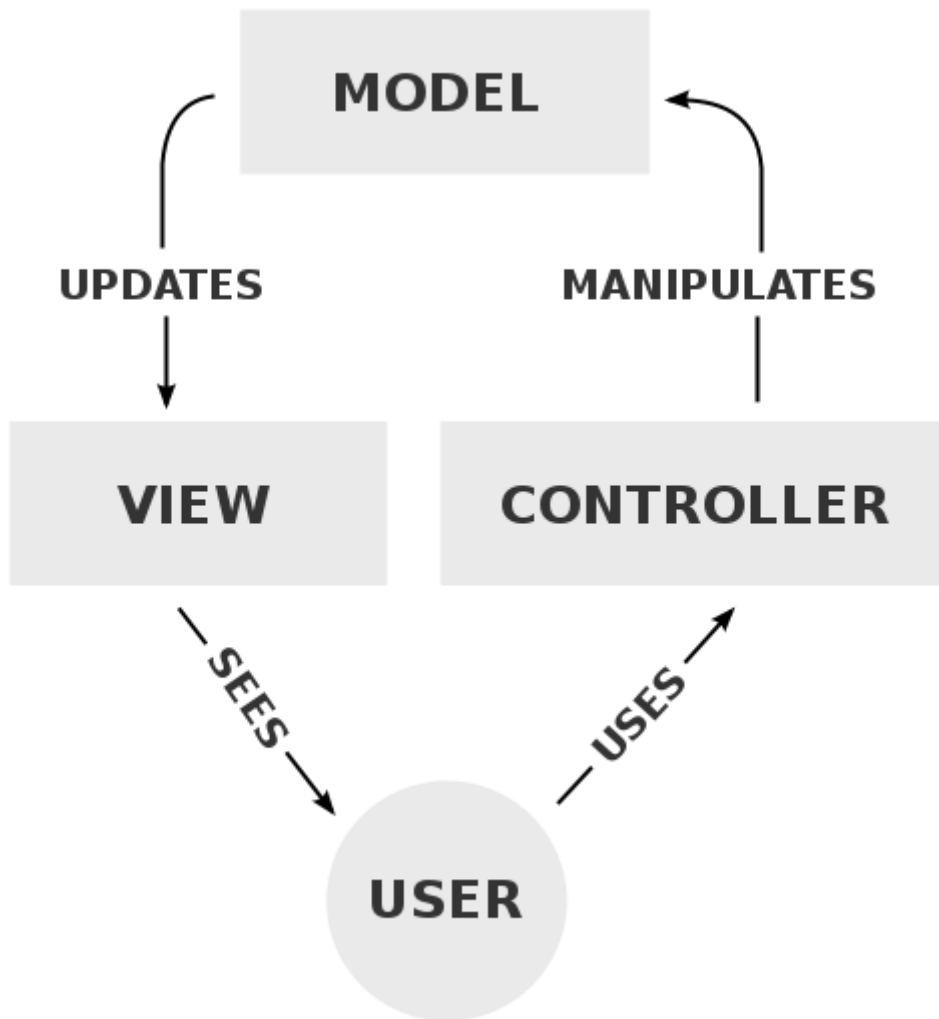


Рисунок 6 - Принцип роботи MVC – паттерну

Фреймворк Django ввів нову термінологію MTV, це різновид шаблону MVC. В Django функції, що відповідають за обробку логіки, відповідають частині Controller з MVC, але називаються View, а відображення відповідає частині View з MVC, але називається Template. **Отже, за логіку проекту відповідає View, а за відображення даних Template.**

Концепція моделі все ще залишається незмінною. Модель містить логічну файлову структуру проекту, є проміжне програмне забезпечення та обробка даних між базою даних та видом. Модель **обробляє** формат даних з **виду**, вони зберігаються в базі даних і навпаки, отримання інформації з бази даних передається до **виду**. **Вид в MTV моделі** форматує дані за допомогою

моделі. У свою чергу, **модель** передає в базу даних і ті дані, які передаються в шаблон для перегляду. Django має файл з ім'ям `urls.py`, який зіставляє вхідні запити з **видами**.

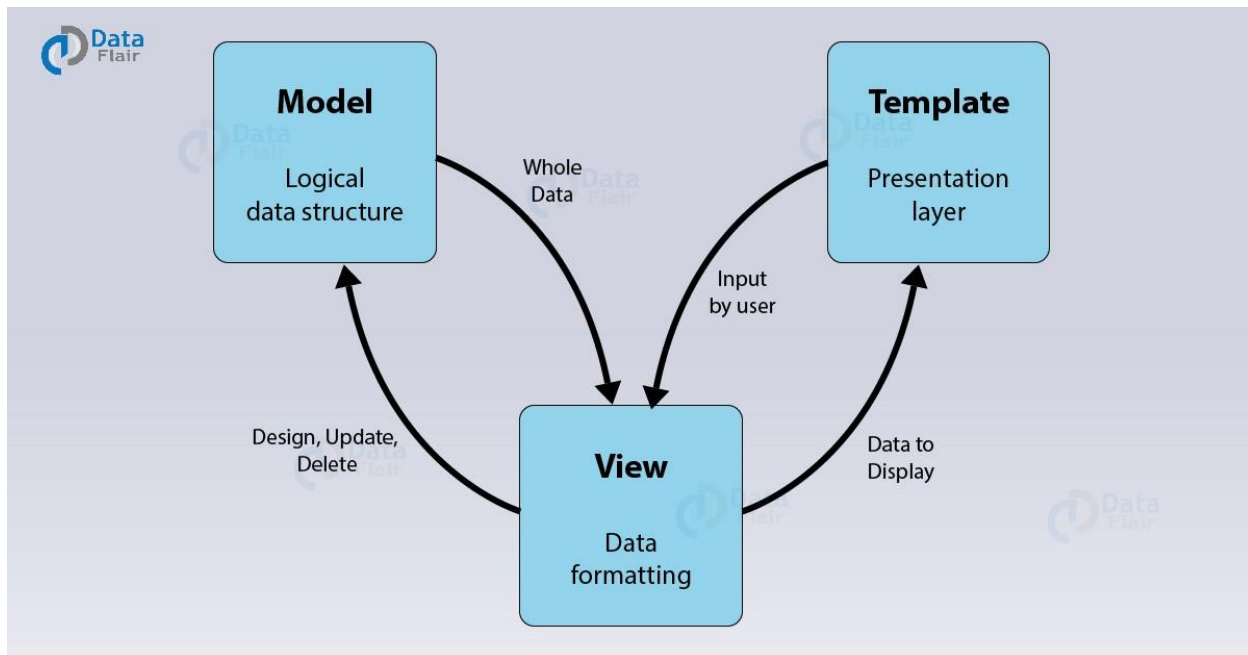


Рисунок 7 - Принцип роботи MTV – паттерну

Як видно з наведеної нижче схеми, ми маємо деякі клієнтську та серверну сторони. Можна помітити, що вид знаходиться на стороні сервера, а шаблон - на стороні клієнта.

Кроки роботи застосування:

1. Ми робимо запит до веб-сторінки, шаблон — інтерфейс, за допомогою якого ми робимо цей запит через браузер.
2. Коли наш шаблон оновлюється, запит потрапляє до View, а потім перенаправляється до URL.
3. Після надсилання запиту на правильну URL-адресу застосовується логіка програми, і модель ініціює правильну відповідь на даний запит.
4. Потім ця відповідь надсилається назад до виду, де вона знову перевіряє відповідь і передає її як відповідь HTTP або бажаний формат користувача.

5. Потім він знову відображається браузером за допомогою шаблонів.

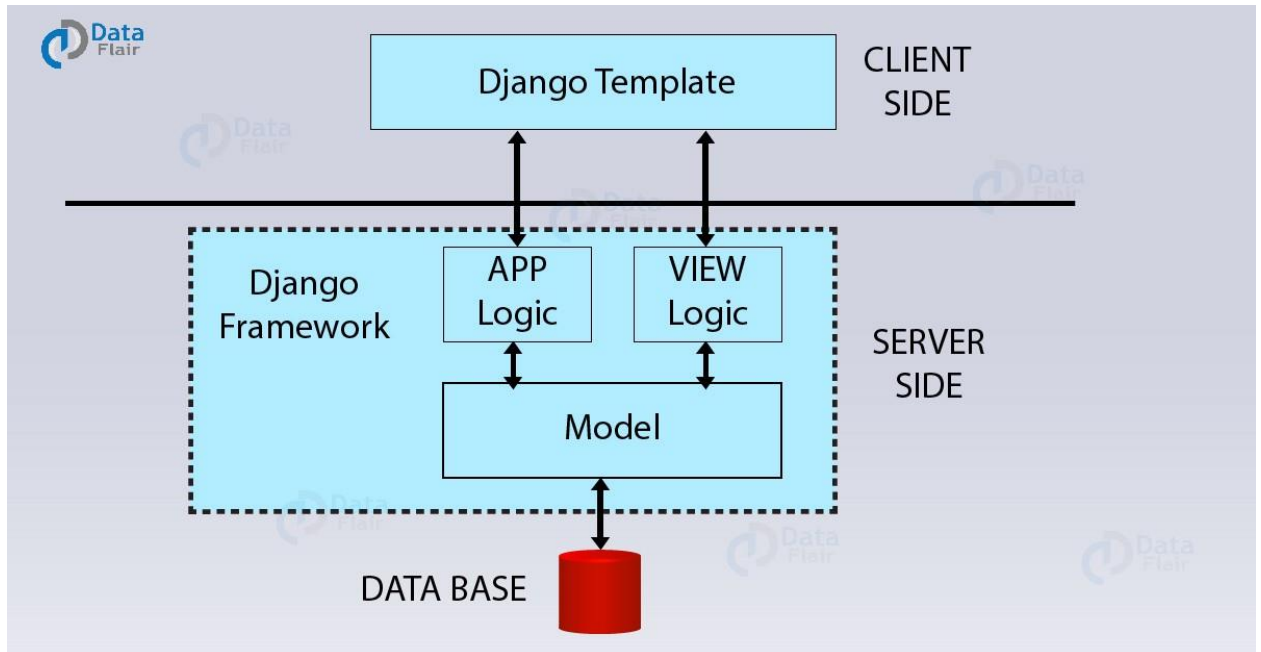


Рисунок 8 - архітектура Django

2.2. Класифікація сайтів

Наразі кількість інформації у веб-просторі збільшується з кожним днем. Для її впорядкування створюються різноманітні сайти, які різняться за задачами, призначенням, тематикою, функціональністю або технічними характеристиками. Таке впорядкування корисне для розуміння переваг та недоліків того чи іншого способу та безпосередньо впливає на вибір інструментів, планування сайту.

Оскільки сайти можуть відрізнятися по багатьом характеристикам, то розглянемо основну з них — класифікацію відносно задач та призначення:

- Сайт-визитка - це додатковий онлайн-інструмент маркетингу, який складається максимум з 5-7 сторінок. Він містить відомості про компанію, інформацію про продаж товарів/послуг, контактні

дані. Переваги: низька вартість створення та швидкий запуск. Недоліки: слабо розвинутий функціонал та дизайн.

- Промо сайт - створено для просування конкретного продукту, послуги компанії, маркетингової акції, події або заходу. В основному використовується для збільшення продажів та привернення уваги. Переваги: яскравий дизайн, який відгукується користувачеві. Недоліки: слабо розвинутий функціонал та структура.
- Корпоративні сайти - найбільш корисні великим компаніям, які мають різні філії в регіонах або дилерську мережу. Такі сайти є багатофункціональними, вони надають інформацію про компанію, її товари або послуги. Можуть містити в собі функціонал достатній для роботи працівників. Переваги: централізоване сховище докладної інформації, можливість взаємодії працівників, клієнтоорієнтованість. Недоліки: чітко виражених немає.
- Інтернет-магазин – спрямований на торгівлю в інтернеті. Для кожного клієнта передбачений особистий кабінет з відображенням історії покупок, корзиною, є система бонусів і знижок, а також розділи новинок і хітів продажів. Переваги: можливість покупки за кілька кліків мишкою, можливість порівняння і оцінки товарів, цілодобова доступність. Недоліки: неможливість побачити товар наочно.
- Інформаційний сайт - інформаційний ресурс по заданій темі. Зазвичай великі за обсягом, мають від 500 сторінок. Мають багато підрозділів та категорій. Переваги: розвинута навігація. Недоліки: слабкий дизайн, адже увага більше приділяється інформації.

- Форум – сайт з цілю спілкування. Може бути як цілим сайтом, так і його частиною. Надає можливість висловитися на конкретну тему, реалізован кабінет користувача та можливість створювати свої теми. Переваги: високий рівень можливостей спілкування. Недоліки: потребує багато часу та сил для розробки, системи захисту даних та модерації.
- Блог – інтернет-журнали або щоденники, де користувачі діляться своїми думками та новинами. Є можливість додавати коментарі до кожного блогу. Блог може вести конкретна людина або ціла організація. Переваги: корисна інформація для клієнта, гарна можливість просувати сайт в інтернеті. Недоліки: потребує постійного оновлення інформації та додавання актуальних тем.
- Соціальна мережа – достатньо складна система, яка вміщує в себе багато складових. Створена для спілкування користувачів, також може мати рейтингову систему, та розповсюдження медіа. Переваги: максимальне охоплення аудиторії. Недоліки: висока вартість, складність розробки, постійна підтримка.
- Тематичний сайт - інтернет-енциклопедія, довідник з конкретної теми. Забезпечують інформацією по заданій темі. Функціонал може сильно різнитися від теми до теми. Переваги: максимальне охоплення конкретної теми. Недоліки: привертає тільки людей, які цікавляться тематикою сайту.
- Ігровий портал – складний ресурс для інтерактивних онлайн-розваг. Має реєстрацію користувачів, стрічку новин, можливість зворотного зв'язку. Переваги: чудова графіка. Недоліки: складна структура та розробка, потребує постійної підтримки.

РОЗДІЛ 3: Опис реалізації програмного продукту

3.1 Аналіз технічного завдання

У сервісу наявні наступні групи користувачів: адміністратор та користувач. Визначення адміністратора магазину відбувається розробником, для цього потрібно створити обліковий запис "superuser", який дає повний доступ до сайту і всі необхідні дозволи, використовуючи manage.py. Адміністратор має право:

- додавати нові вправи, програми, користувачів, групи користувачів, редагувати інформацію про них та видаляти з бази;
- додавати, редагувати та видаляти статті, опубліковувати не опубліковані статті;
- схвалення та видалення коментарів.
- Користувач має право:
- переглядати каталог вправ та програм;
- реєстрація на сайті;
- використання контактної форми для зв'язку з адміністратором;
- додавати, редагувати та видаляти статті, опубліковувати не опубліковані статті;
- схвалення та видалення коментарів.

В основі проекту – взаємодія веб-застосунку з базою даних, яка зберігає інформацію про користувачів, статті, тренування, програми, харчування.

Інтерфейс сайту є простим та інтуїтивно-зрозумілим. Веб-сайт налічує п'ять основних сторінок:

«Програми» – добірка програм, яка складається з вправ. Є можливість здійснювати просту навігацію за пунктами меню: переходити на сторінки «Вправи», «Харчування», «Блог», «Контакти», «Вхід» та «Реєстрація».

«Вправи» – добірка вправ, у які можна зайти, почитати інструкцію щодо виконання, короткі відомості про вправу та подивитися відео-виконання.

«Харчування» – можливість додати їжу до свого раціону, перевірити індекс маси тіла.

«Блог» – невеличкий блог, у якому можна створити статтю, редагувати її, задати дату її публікації. Додавати та видаляти коментарі на сторінці блогу.

«Контакти» – відображає контактну форму, за допомогою якої можна написати адміністратору.

«Вхід» та «Реєстрація» – модальне вікно входу та сторінка реєстрації відповідно.

3.2. Обґрунтування вибору засобів розробки

Для розробки своєї платформи я обрала мову Python. Хочеться відзначити, що Python - це затребувана мова програмування з великим потенціалом в майбутнє.

Переваги мови Python:

- Зрозуміла і проста мова програмування. Синтаксис немає нічого зайвого, код чистий і зрозумілий без зайвих дужок і виразів;

- Можливість розширення. Існують бібліотеки і фреймворки під будь-який тип завдань і потреб. Також величезним плюсом є те, що ми можемо використовувати С код з Python;
- PEP - єдиний стандарт для написання коду, що робить код підтримуваним і читабельним навіть при переході від одного програміста до іншого. Це підтримує популярність Python;
- Open Source - код інтерпретатора Python є відкритим, що дозволяє будь-кому, хто зацікавлений у розвитку мови взяти участь в його розробці і поліпшити його;
- Широке застосування. Використовується для розробки веб-додатків, ігор, зручний для автоматизації, математичних обчислень, машинного навчання, в області інтернету речей;
- Велика інтернет-спільнота. Якщо розробник стикається з питаннями і труднощами, він завжди може запитати поради у колег, що значно прискорює вирішення проблем. Також є велика кількість навчальних матеріалів.

Разом із мовою Python я використовую фреймворк Django. Він містить величезну кількість функціональності для вирішення більшості завдань веб-розробки.

Переваги Django:

- Django як фреймворк задає структуру проекту. Вона допомагає розробникам розуміти, де і як додавати нову функціональність. Завдяки однаковій для всіх проектів структурі набагато простіше знайти вже готові рішення або отримати допомогу від спільноти.
- Додатки в Django дозволяють розділити проект на декілька частин. Цей підхід дозволяє легко інтегрувати готові рішення.

- Django працює з десятками додаткових функцій, які помітно допомагають з аутентифікацією користувача, картами сайту, адмініструванням вмісту, RSS і багато чим іншим. Дані аспекти допомагають здійснити кожен етап веб розробки.
- Django включає механізми запобігання поширених атак на зразок SQL-ін'єкцій (XSS) і підробки міжсайтових запитів (CSRF).

Фронтенд розробка виконувалась за допомогою HTML, CSS та JavaScript.

3.3 Опис розробки програм

Використовується клієнт-серверна архітектура. Використовується HTTP-протокол. Браузер формує запит і відправляє його на сервер, після чого серверне програмне забезпечення обробляє цей запит, формує відповідь і передає його назад клієнтові. Після цього клієнтську програму може продовжити відправляти інші запити, які будуть оброблені аналогічним чином.

Спочатку запит у Django надходить до urls.py, а потім переходить до відповідної функції в views.py. Функції Python у views.py приймають веб-запит з urls.py і надають веб-відповідь у шаблони.

У Django за замовчуванням у проекті є файл urls.py. Він також має визначений попередньо шлях для програми адміністратора. Однак Django рекомендує зіставляти всі ресурси за допомогою іншого urls.py, **який створюється** в додатку. Зіставлення URL-адрес перенаправить запит з URL-адрес проекту на URL-адреси додатків, а потім на відповідну функцію у **виду**.

Кожна програма django за замовчуванням отримує файл models.py для створення таблиць баз даних. За замовчуванням механізм бази даних у фреймворку django є sqlite3. Його можна змінити, відвідавши settings.py в каталозі проекту. Django використовує ORM (Object Relational Mapping), що дозволяє нам виконувати операції з базами даних без написання SQL-запитів, але з простими об'єктами python.

Django Admin - це внутрішній інтерфейс користувача для суперкористувачів та **розробників** веб-додатків. **Можна** зареєструвати свої моделі на сайті адміністратора та контролювати створення, перегляд, зміну та видалення дій у моделях. Спочатку суперкористувача можна створити за допомогою команди create_superuser, а потім інших користувачів можна створити з інтерфейсу адміністратора.

3.4 Опис файлів даних та інтерфейсу програми

Перша сторінка сайту, на яку потрапляє користувач – список програм тренувань.

```
{% for program in programs %}
  <div class="item program-item">
    
    <h2><a href="{% url 'program_detail' pk=program.pk %}">{{
program.program_name }}</a></h2>
    <p>Вид тренування: {{ program.program_type }}</p>
    <p>Група м'язів: {{ program.program_group }}</p>
    <p>Обладнання: {{ program.program_equipment|linebreaksbr }}</p>
  </div>
{% endfor %}
```

Автентифікація підтверджує особистість користувача, а авторизація визначає, що дозволено автентифікованому користувачеві. Користувач може авторизуватися в програмі. Це відбувається за допомогою вбудованої платформи аутентифікації, яка обробляє облікові записи користувачів, групи, дозволи та сеанси користувачів. Платформа аутентифікації створюється по замовчуванню при створенні проекту. Вона містить такі моделі:

- User: Модель користувача (username, password, email, first_name, last_name, і is_active).
- Group: Модель для класифікації користувачів.
- Permission: Прапори для виконання певних дій.
- Система хешування паролів

Також платформа аутентифікації має такі класи:

- AuthenticationMiddleware: Пов'язує користувачів із запитом за допомогою сесій
- SessionMiddleware: Обробляє поточну сесію за запитом

У моєму проєкті форма LoginForm використовується для аутентифікації користувачів через базу даних.

```
class LoginForm(forms.Form):
    username = forms.CharField()
    password = forms.CharField(widget=forms.PasswordInput)
    next = forms.CharField(widget=forms.HiddenInput(), required=False)
```

Також є вид user_login, який викликається з запитом GET :

```
def user_login(request):
    if request.method == 'POST':
        form = LoginForm(request.POST)
        if form.is_valid():
            cd = form.cleaned_data
            user = authenticate(username=cd['username'],
password=cd['password'])
            if user is not None and user.is_active:
                login(request, user)
                return render(request, 'registration/profile.html', {'form':
form})
            else:
                form.add_error(None, 'Невірний логін або пароль')
                return render(request, 'registration/login.html', {'form':
form})
        else:
            form = LoginForm()
            return render(request, 'registration/login.html', {'form': form})
```

Коли користувач відправляє форму за допомогою POST, відбувається наступне:

1. Створюється екземпляр форми з відправленими даними.
2. Перевіряється валідність форми, надається інформація про помилки, якщо вони є.
3. Якщо надіслані дані є валідними, ми шукаємо користувача в базі даних за допомогою методу `authenticate()`, який приймає `username` і `password` і повертає об'єкт `User`.
4. Якщо користувач успішно пройшов аутентифікацію, ми перевіряємо, чи активний він за допомогою атрибуту `is_active`.
5. Якщо користувач активний, він входить на сайт. Ми встановлюємо сесію, викликавши метод `login()` і перенаправляємо користувача на сторінку профілю.

3.5. Тестування програми і результати її виконання

Тестування програми проводилося в браузері Орега.

Спочатку користувач потрапляє на сторінку з програмами тренувань. Користувач може натиснути на них та отримати більш детальну інформацію.

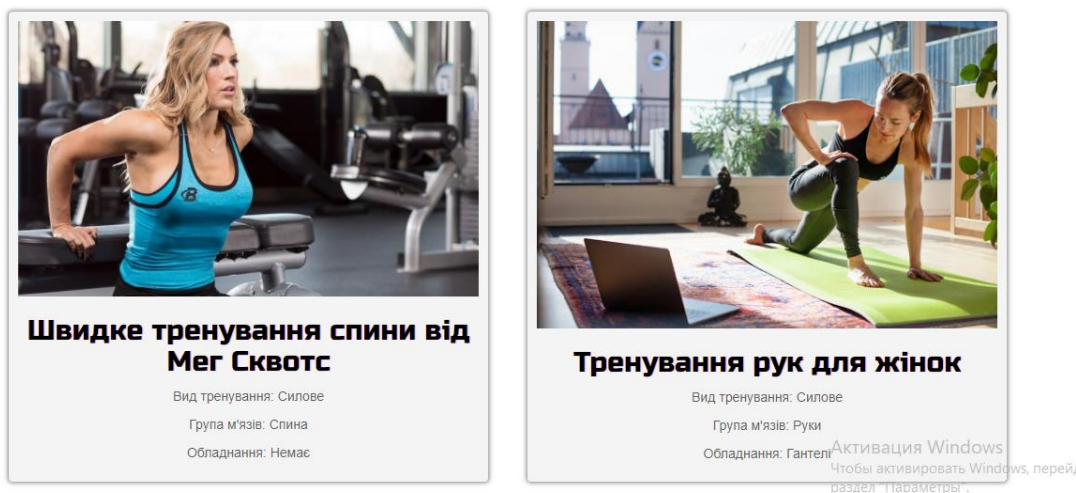


Рисунок 9 - Програма тренувань

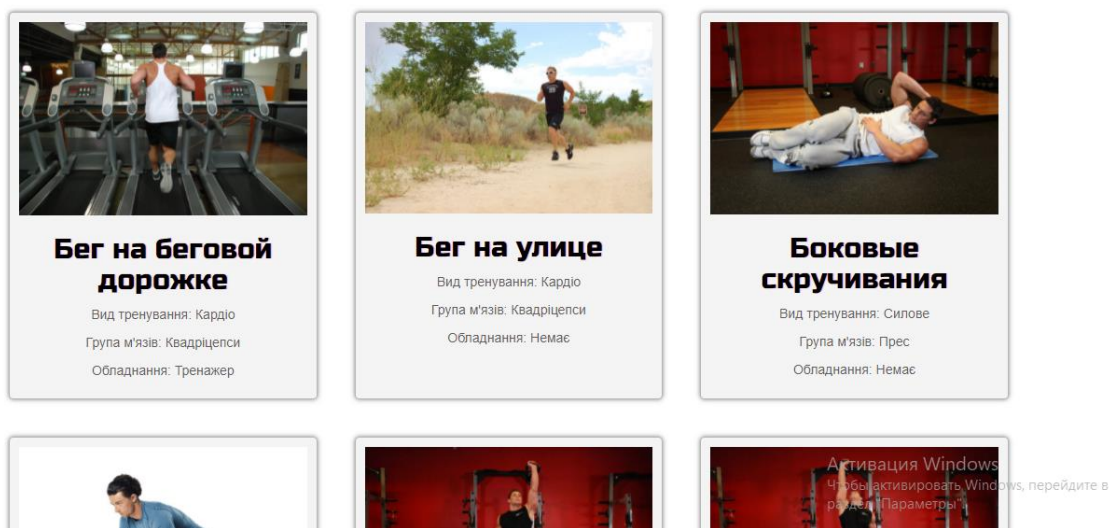


Рисунок 10 - Вправи

У вкладці «Вправи» можна побачити перелік доступних вправ. Сторінка з вправою, вміщує відео, короткі відомості про вправу та інструкцію. Також є можливість додання коментарів.

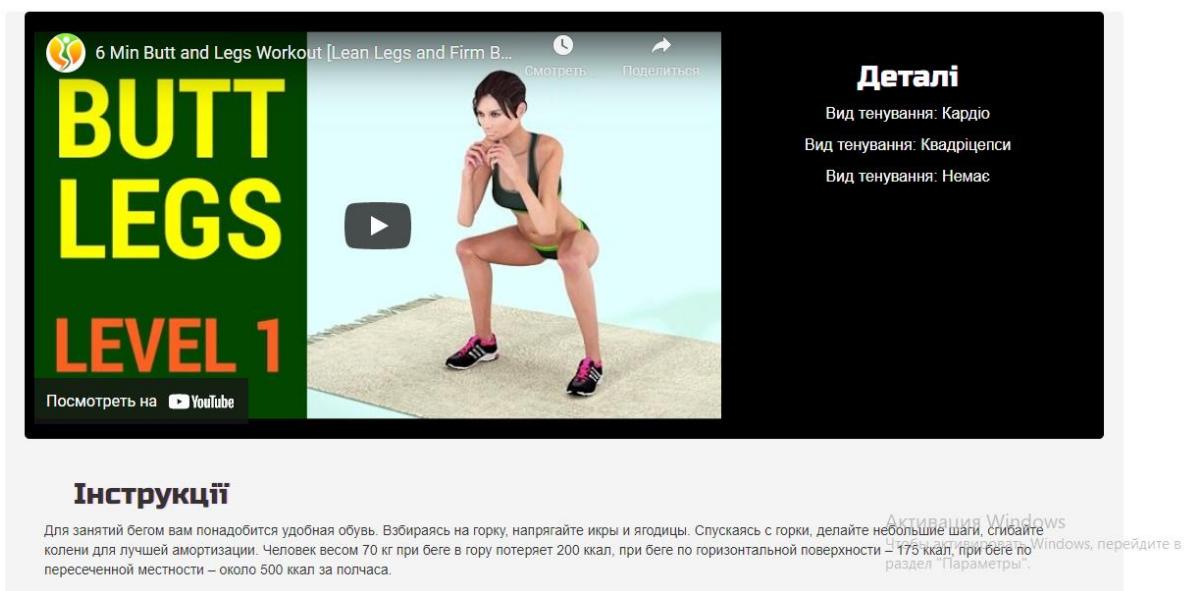


Рисунок 11 - права

У вкладці «Блог» можна потрапити на перелік статей, наявних на сайті. На кожну статтю можна натиснути та подивитися детальніше, залишити коментар. Також статтю можна редагувати та видаляти.

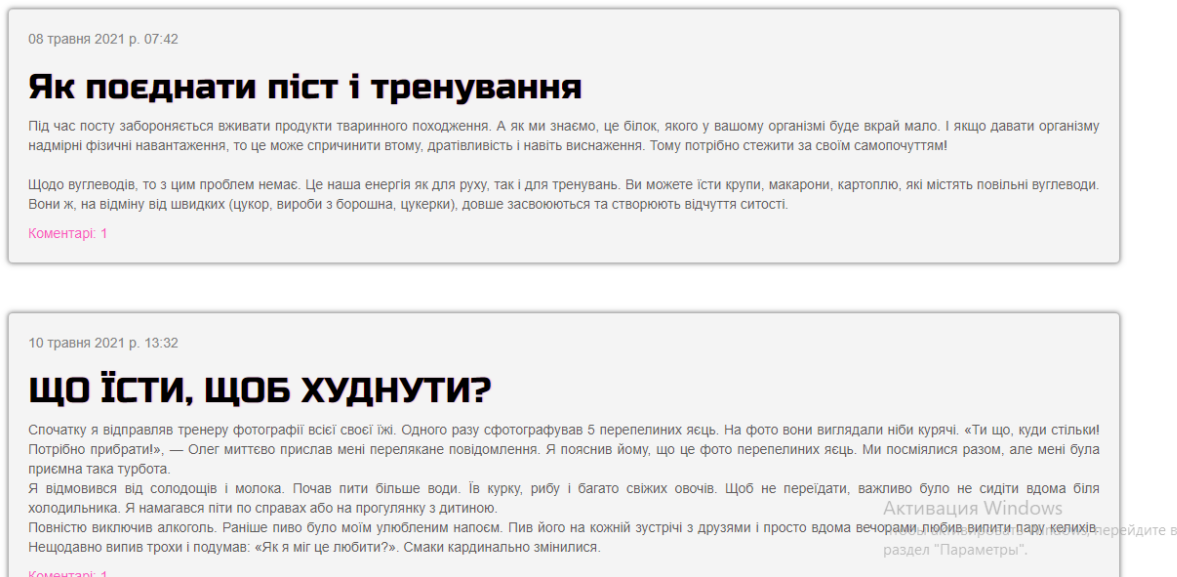


Рисунок 12 - сторінка блогу

У вкладці «Контакти», можна знайти форму для зв'язку через електронну пошту.

Залиште повідомлення

Email:

Тема:

Надіслати

Рисунок 13 - Контактна форма

У вкладці «Харчування» є можливість додати їжу та подивитися її енергетичну цінність, кількість білків, жирів та вуглеводів.

Додати їжу:

вода питна

Їжа за сьогодні:

Їжа	Енергія(ккал.)	Білки(г)	Вуглеводи(г)	Жири(г)	Видалити
вода питна	0	0,0	0,0	0,0	Видалити
кава натуральна без цукру	2	0,12	0,0	0,2	Видалити
кава натуральна без цукру	2	0,12	0,0	0,2	Видалити
яйце куряче варене	152	13,0	1,0	11,0	Видалити
вода питна	0	0,0	0,0	0,0	Видалити
банан	94	1,0	22,0	0,2	Видалити

Рисунок 14 - розділ "Харчування"

Форма додання коментарів.

Новий коментар

Author:

Text:

Рисунок 15 - форма додання коментарів

12 травня 2021 р. 01:22

Me

Cool)

12 травня 2021 р. 01:23

No aut

Lol

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Рисунок 16 - коментарі

Коментарі на сторінці відображаються під постами або під вправами.

Висновки

Отже, поставлена задача була виконана: розроблено веб-застосунок для контролю стану здоров'я, який задовольняє потреби сучасного користувача. У результаті виконання роботи створено програмний застосунок з зручним та інтуїтивно-зрозумілим інтерфейсом, завдяки якому користувачі можуть займатися спортом в домашніх умовах.

Окрім цього, завдяки гнучкості фреймворку та програмного коду можна з легкістю продовжувати оснащувати платформу різноманітним функціоналом. Додавати більше тренувань, вправ під різні потреби користувачів.

Список використаної літератури

1. ТUTORІАЛ по створенню блогу на Django [Електронний ресурс]
<https://tutorial.djangogirls.org/uk/>
2. ТUTORІАЛИ з прикладами реалізацій програм на Django
[Електронний ресурс] <https://pocoz.gitbooks.io/django-v-primerah/content/>
3. Відео пояснення праці з фрейворком Django [Електронний ресурс]
<https://www.youtube.com/watch?v=w4nrT7emiVc&t=1538s>
4. Відео пояснення праці з фрейворком Django [Електронний ресурс]
https://www.youtube.com/watch?v=QFy_RmpJ4ZQ
5. Офіційна документація [Електронний ресурс]
<https://docs.djangoproject.com>
6. ТUTORІАЛ по створенню блогу на Django [Електронний ресурс]
<https://tutorial-extensions.djangogirls.org>
7. MVC та MTV [Електронний ресурс]
<https://lectureswww.readthedocs.io/6.www.sync/2.coding/1.mvc.html>
8. Стаття, у якій розглянуто класифікацію сайтів [Електронний ресурс]
<https://99designs.com/blog/web-digital/types-of-websites/>