

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

_____ Наталія АУШЕВА

«___» _____ 2022 р.

Дипломна робота

на здобуття ступеня бакалавра

спеціальності 122 «Комп'ютерні науки»

**освітня програма «Комп'ютерний моніторинг та геометричне
моделювання процесів і систем»**

на тему: «Веб-портал для фітнес тренувань»

Виконала :

студент IV курсу, групи ТР-81

Василенко Ірина Вікторівна _____

Керівник:

Ст.вик.,

Колумбет Вадим Петрович _____

Рецензент: _____

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2022

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший

спеціальність 122 “Комп’ютерні науки”

освітня програма «Комп’ютерний моніторинг та геометричне моделювання
процесів і систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Наталія АУШЕВА
(підпис)

” ____ ” _____ 2022 р.

ЗАВДАННЯ

на дипломну роботу студенту

Василенко Ірині Вікторівні

(прізвище, ім’я, по батькові)

1. Тема роботи _____ “Веб-портал для фітнес тренувань”

керівник роботи _____ Колумбет Вадим Петрович, ст. вик.
(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ” 25 ” травня _____ 2022р.
№ _____

2. Строк подання студентом роботи 10 червня _____ 2022 р.

3. Вихідні дані до роботи мова програмування PHP, середовище розробки IntelliJ Idea.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) розробити веб-портал для фітнес тренувань.

5. Перелік ілюстративного матеріалу: 44 рисунка.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

Дата видачі завдання ” 10 ” вересня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	25.05.2022	
2.	Вивчення та аналіз задачі	02-08.05.2022	
3.	Розробка архітектури та загальної структури системи	09-15.05.2022	
4.	Розробка структур окремих підсистем	16-22.05.2022	
5.	Програмна реалізація системи	09-28.05.2022	
6.	Оформлення пояснювальної записки	28-31.05.2022	
7.	Захист програмного продукту	01-08.06.2022	
8.	Передзахист	09.06.2022	
9.	Захист	15-30.06.2022	

Студент

_____ (підпис)

Василенко І. В.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Колумбет В. П.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Головною метою моєї дипломної роботи є створення веб-порталу для фітнес тренувань.

Саме цей проект розроблявся мною, щоб отримати досвід в роботі з веб-порталом і ведення проекту з нуля.

Складові пояснювальної записки: 60 сторінок, 44 рисунка, 1 додаток, 6 посилань.

При розробці програмного продукту я проаналізувала існуючі сайти, які заповнені інформацією про фітнес тренування та підкреслила для себе недоліки і переваги. При написанні продукту була створена серверна частина, яка написана на мові програмування PHP, та створений додатково чат-бот для телеграму.

Мною було створено зручний портал для відслідковування різної інформації, яка пов'язана з фітнес тренуваннями. Також, був розроблений чат-бот, який відслідковує біоритм.

Ключові слова: веб-портал, чат-бот, фітнес тренування, клієнт, адміністратор.

ABSTRACT

The main purpose of my thesis is to create a web portal for fitness training.

This project was developed by me to gain experience in working with a web portal and running a project from scratch. Components of the explanatory note: 60 pages, 44 figures, 1 appendix, 6 links.

When developing the software product, I analyzed the existing sites, which are filled with information about fitness training and highlighted the disadvantages and advantages. When writing the product, a server part was created, which is written in the PHP programming language, and an additional chatbot was created for the telegram.

I have created a convenient portal for tracking various information related to fitness training. Also, a chatbot has been developed that monitors biorhythms.

Keywords: web portal, chatbot, fitness training, client, administrator.

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	8
ВСТУП.....	10
1 ПОСТАНОВКА ЗАДАЧІ ТА ПРОЕКТУВАННЯ ДОДАТКУ	12
1.1 Часові рамки	12
1.2 Призначення та користувачі	14
1.3 Основні задачі	14
1.4 Програмна архітектура	15
1.5 Діаграма прецедентів	18
1.6 Діаграма розгортання	19
1.7 Діаграма розгортання процесів	22
2 ОПИС ПРЕДМЕТРОЇ ОБЛАСТІ	23
2.1 Sport Life	24
2.2 Cloe Ting	25
2.3 Fitness Blender	26
2.4 Ferroom	27
3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДОДАТКУ	28
3.1 Опис моделі	25
3.2 Опис структури бази даних	32
3.3 Опис запрограмованого чат-боту	34
4 ЗАСОБИ РОЗРОБКИ	36
4.1 Серверна частина	36
4.2 Веб-портал	42
4.3 Чат-бот	44
5 КОРИСТУВАННЯ РОБОЧИМ ВЕБ-ПОРТАЛОМ	47
5.1 Користувач — клієнт	48
5.2 Користувач — адміністратор	57

ВИСНОВКИ	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	60
ДОДАТОК А	61

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

СКБД — це система керування базами даних, сукупність набору пов'язаних між собою даних (баз даних) та програмних продуктів для користування та доступу до даних, яких потрібно. Програмні складові СКБД мають основні інструменти, які використовуються для ведення адміністратором баз даних. Можна швидко створити, оновити та знайти потрібну інформацію в базах даних.

БД — сполучені дані, які взаємодіють між собою та мають зв'язки. Складається із таблицок, куди записуються дані.

Веб-портал — це сторінка сайту, яка має місце в інтернеті та включає в себе різні інтерактивні можливості, якими може користуватись будь-хто.

UML — мова моделювання уніфікована.

URL — це певна адреса, за якою користувач в інтернеті може перейти на ресурс, який йому потрібно.

HTML — мова гіпертекстової розмітки. Мова розмітки, яка зберігає в собі певну інформацію. При використанні HTML ми позначаємо текст, який хочемо, щоб побачили у браузері.

CSS — це унікальна мова стилів сторінок. Зазвичай, вона описує зовнішній вигляд, додає стилю сторінкам, які написані на HTML.

PHP — це для широкого користування людей призначена мова програмування. Розроблена мова слугує для веб-додатків.

Frontend — це сторінка в браузері, де людина може переглядати інформацію на сайті та взаємодіяти з відразу з різними функціями. Клієнт може посилати різні запити саме на цій частині.

Backend — це частина програмного проекту, яку користувача не бачить. Вона являється програмним апаратом за допомогою якого реалізовані функції, якими користується клієнт.

Web-Api — це комплект із різних протоколів, програм, які допомагають взаємодіяти різним програмам між собою.

MySQL — це найпопулярніша СУБД, яка має систему керувань з реляційними БД. MySQL використовується у різних програмах, зокрема зустріти можна на веб-серверах, створює динамічні сторінки і має гарну допомогу від різних мов програмування.

HTTP — широковідомий протокол в комп'ютерних мережах, який передає дані.

SQL — призначена мова для формування баз даних та створення різних запитів, взаємодій з ними.

PhpMyAdmin — зручний інструмент за допомогою, якого можна взаємодіяти з базами даних.

Python — інтерпретована мова програмування.

MVC — при проектуванні програми часто використовується даний шаблон. Він допомагає створювати зручний та гнучкий дизайн програми та надає використовувати в подальшому компоненти.

ВСТУП

На сьогоднішній день існує велика кількість фітнес залів та клієнтів, які їх відвідують. Більшість фітнес залів не мають адекватної системи контролю клієнтів, деякі використовують паперові абонементи, деякі карточки. Тому, як ніколи, є актуальним розробка веб-порталу, за допомогою якого буде легко відслідковувати контроль клієнтів.

Наше життя потребує догляду духовного та фізичного. Головним чинником нашого здоров'я є догляд за нашим тілом та підтримка фізичної форми. Відвідування занять з фітнесу в спортивному залі - це запорука бути здоровим у майбутньому.

Також, актуальним є різна інформація, яка може допомогти при тренуваннях. Дуже важливим є відслідковування свого харчового раціону. Тому, я вирішила додати його до порталу.

Думка про створення даного продукту виникла після того, як я відвідувала декілька фітнес залів, а там були складнощі з контролем клієнтів. Ці складнощі відображались саме для керівників фітнес залів, вони вели контроль з паперовими абонементом.

Саме тому необхідність створення саме такого програмного продукту виключить проблеми з контролем клієнтів. Особливо важливо дотримуватись певного режиму тренувань.

Часто радять займатись тричі на тиждень, щоб ваше фізичне тіло було в певній мірі підтягнуте та ви були здорові. Отже, ходити в фітнес зал піде на користь кожній людині.

Розроблений продукт має цінність, також може використовуватись в будь-яких фітнес залах, адже контроль клієнтів потрібен всюди та поради нікому не бувають лишніми.

Актуальним рішенням я додала створити власний чат-бот, який буде відслідковувати біоритм людини.

Зрозуміло, що існують подібні сайти з такою тематикою, але не всі вони мають зручний інтерфейс та не мають додаткового чат-боту.

Часто людина заходить в телеграм кожен день. Ми постійно спілкуємось з людьми, вирішуємо проблеми через даний месенджер.

Чат-боти можуть гарно спілкуватись з користувачами в повідомленнях. Бот розпізнає та виконує різні дії, які були запрограмовані. Біоритми бувають різні, але я обрала біоритм, який відслідковується за датою дня народження та поточним днем і часом.

За допомогою боту можна відслідковувати свої дані, такі як фізичний, емоційний та інтелектуальний стан. Це дуже важливо знати та відслідковувати, коли ваші поточні стани мають найбільший відсоток. За вдяки цьому ви зможете продуктивніше робити певні задачі, які вам потрібно.

Отже, даний веб-портал буде корисний кожному, хто зацікавлений підтримувати своє здоров'я та емоційний стан. Також, гарним додатком є відслідковування клієнтів в спорт-залі адміністратором.

1 ПОСТАНОВКА ЗАДАЧІ ТА ПРОЕКТУВАННЯ ДОДАТКУ

Постановка задачі в створенні функціонованого веб-сайту зі швидким доступом для клієнта, який може купити абонемент та адміністратора, який контролює клієнтів. Клієнт може переглядати рецепти, обирати раціон та тренування і має змогу подивитись спортивні зали поруч і перейти за посиланням в чат-бот.

1.1. Часові рамки

При оцінці та розробці програмного продукту необхідно дотримуватись часових проміжків, щоб вчасно зробити поставлені задачі та здати все вчасно. Даний розподіл допомагає, коли робити певні задачі.

Розробка програмного продукту була поділена на наступні процеси:

1. Проаналізувати існуючі веб-портали
2. Формулювання переліку функцій програми
3. Виставлення часових рамок на роботу
4. Схвалити технічне завдання
5. Зробити use-case діаграму
6. Зробити дизайн
7. Формування баз даних
8. Внести дані в базу даних
9. Запрограмувати Frontend
10. Запрограмувати Backend
11. Поєднання Frontend і Backend частин
12. Тестування роботи
13. Відредагувати помилки

Задачі, які потрібно було виконати розбиті на певний період часу, початок переддипломної практики з 02 травня по 29 травня згідно.

Розроблені задачі, які були вписані на певні часові рамки були розроблені саме в той час, який зазначений на діаграмі, яка полегшила розробку продукту.

Діаграма Ганта — це особливий вид роботи з завданнями прямо в календарній сітці. На ній ви бачите візуальне представлення тривалості та крайніх термінів за всіма своїми завданнями.

Діаграму я створила у додатку Smartsheet, ця програма являється для мене доволі зручною та її достатньо відкрити за посиланням в браузері і не скачувати.

Діаграма зображена на рисунку 1.1, показує часові межі роботи.

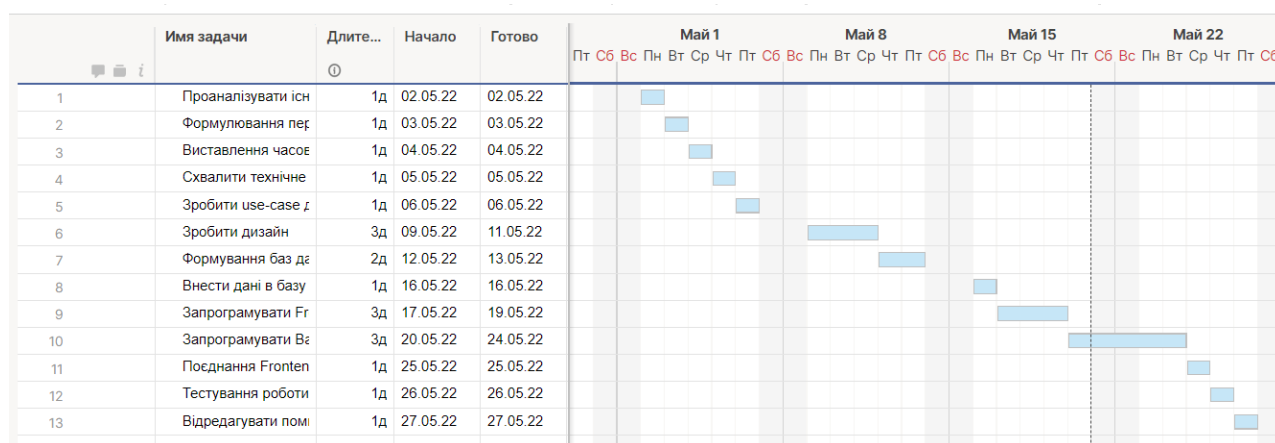


Рисунок 1.1 — Діаграма Ганта

Завдяки діаграмі практика була виконана вчасно і це гарна допомога при створенні програмного продукту.

1.2 Призначення та користувачі

Веб-портал призначений для будь-яких фітнес залів, де потрібно вести систему контролю клієнтів. Саме за допомогою сайту легко відбувається процес контролю, як для керівника фітнес залу. В наші дні це актуально, як ніколи. Багато людей зараз підтримують здоровий спосіб життя, тому і записуються в фітнес зали.

Використовувати програмний продукт може будь-хто. Починаючи від школярів до пенсіонерів. Кожен може зареєструватись на сайті та отримати абонемент в фітнес зал.

1.3 Основні задачі

Веб-портал в майбутньому можна розширювати та додавати нові функції. Основна ціль розробленого програмного продукту — це полегшений контроль клієнтів в фітнес залах.

Головні функції, які виконує даний продукт:

- реєстрація клієнту
- вибір абонементу клієнтом
- покупка абонементу
- перегляд та вибір тренувань
- вибір харчового раціону
- перегляд актуальних рецептів
- відслідковування найближчих спорт залів на мапі
- відслідковування адміністратором інформації про клієнта
- посилення на чат-бот, який відслідковує біоритм

За допомогою вище перерахованих задач, можна отримати готовий програмний продукт, який допоможе вирішити проблеми відслідковування всіх

клієнтів в фітнес залі та перегляд різноманітних функцій, які допоможуть для тренувань. Завдяки цьому він має широку орієнтацію на клієнтів, що дозволило б йому мати попит на ринку.

1.4 Програмна архітектура

Головним пунктом є створення комунікації між ключовими зацікавленими сторонами системи. Тому, що добре прописаний зв'язок між програмами є ключем при створенні та виведені даних швидке отримання їх. Коректні дані та представлення своїм користувачам. На рисунку 1.2 зображений зв'язок між програмами повинен створюватись при новітніх і зручних компонентів створення.

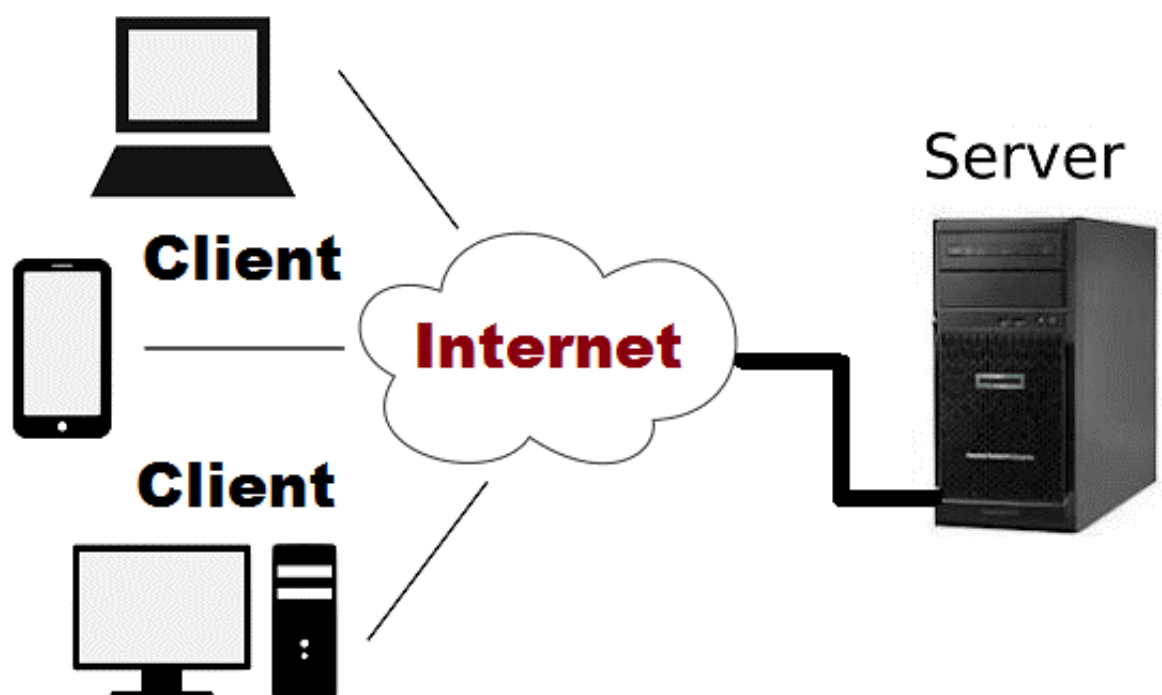


Рисунок 1.2 — Взаємодія між сервером та клієнтом

Користувач повинен мати швидкий доступ до сервера, якщо вірно була створена програма та правильно побудовані зв'язки. Також, інтерфейс для клієнта буде швидкий та

Щоб створити вірну архітектуру бази даних потрібно запрограмувати додаток.

Запрограмований додаток цілком потребує вірної архітектури БД. Щоб її реалізувати слідє підібрати вид БД, яка буде підходити до створеного додатку. Існує перелік БД за типом організації:

- мережева БД. Більше розгорнута, ніж ієрархічна модель. Відмінність слугує в тому, що різні об'єкти можуть належати до кількох предків.
- ієрархічна БД. Має деревовидний вигляд, який створений за допомогою різноманітних рівнів. "Предок- нащадок" - таким чином пов'язані об'єкти.
- об'єктно-орієнтована БД. Створені дані мають вигляді об'єктних моделей.
- реляційна БД. Найпопулярніша система збереження бази даних, яка представлена таблицями даних.

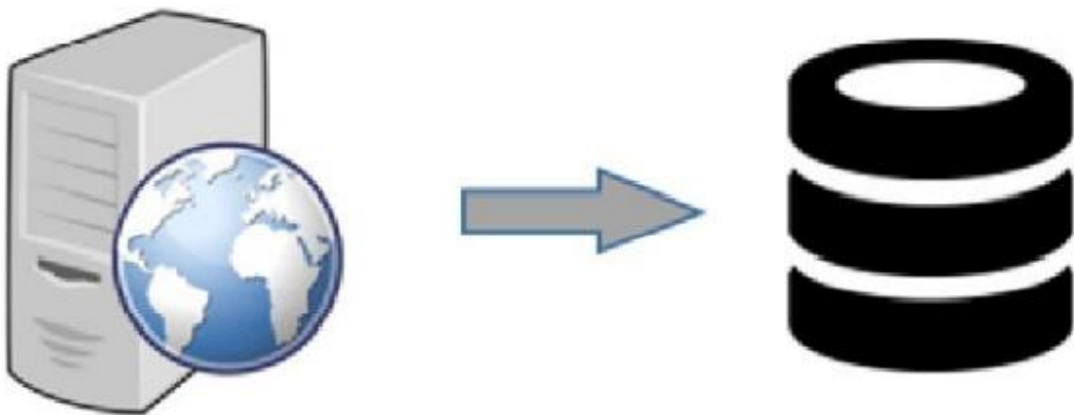


Рисунок 1.3 — Взаємодія сервера з БД

Для створення БД у своїй програмі я обрала реляційний метод.

Також, розглянемо схеми деяких процесів роботи програмного продукту.

Щоб користувача мав змогу швидко та надійно користуватись веб-порталом слід пройти реєстрацію. Згодом, користувач, який зареєструвався зможе купити абонемент та дає дані, щоб відслідковувати дані в чат-боті.

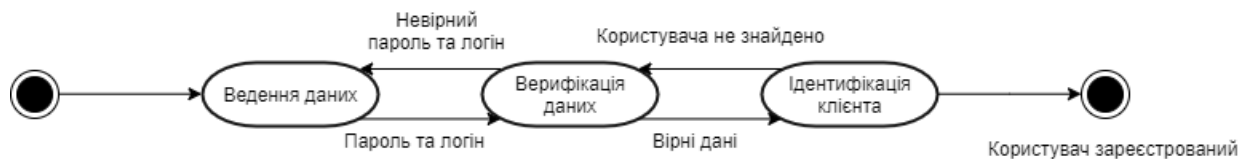


Рисунок 1.4 — Алгоритм перевірки, чи користувач зареєстрований

Щоб увійти до сайту, потрібно ввести свій логін та пароль. Але, щоб придбати абонемент потрібно пройти реєстрацію. Коли користувач вводить невірний пароль та логін йому підсвічується, що він ввів дані неправильно. Якщо дані введені вірно, то користувач потрапляє до свого особистого кабінету.

Одна з головних функцій на сайті є придбання клієнтом абонемента.

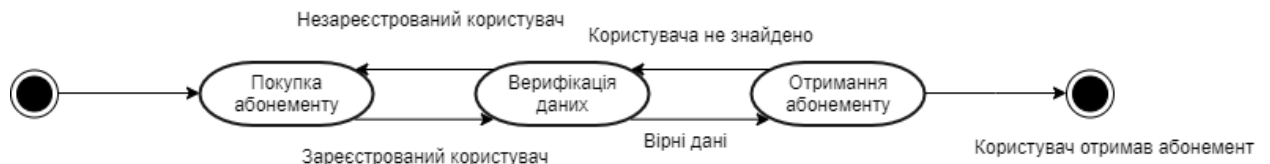


Рисунок 1.5 — Алгоритм придбання абонемента

Придбати абонемент може тільки зареєстрований користувач. Коли користувач обирає абонемент та хоче його купити, він повинен увійти на сайт, тому

спочатку проходить верифікацію, чи зареєстрований на сайті, чи ні. Якщо клієнт зареєстрований, то він може придбати абонемент.

1.5 Діаграма прецедентів

При потребі, які існують при виконанні програмного продукту я створила діаграму прецедентів.

Дана діаграма допомагає зрозуміти різні історії та сценарії створені між прецедентами(тоді, коли використовується) і акторами(певними користувачами). Основною функцією є опис бізнес логіки. Діаграма прецедентів займає головну ціль не тільки у створенні зв'язків з потенційними клієнтами (користувачами) і збирачами певних критеріїв до проекту. Також, Діаграми прецедентів додаються бізнес-логікою та детальною специфікацією прецедентів як вихідної інформації, яку учасники проекту успішно використовують на всіх його фазах (виникнення, проектування, тестування, програмування, документація). Також, діаграма прецедентів є частиною UML.

UML — мова моделювання уніфікована, слугує важливою частиною процесу, який являється уніфікованим. Цей процес забезпечує розробку програмного забезпечення. Дана мова являється широкого профілю. Слугує відкритим стандартом, яким користуються різні графічні позначення при створенні різних моделей для систем.

На рисунку 1.6 візуально показано, які можливості та функції ми маємо.

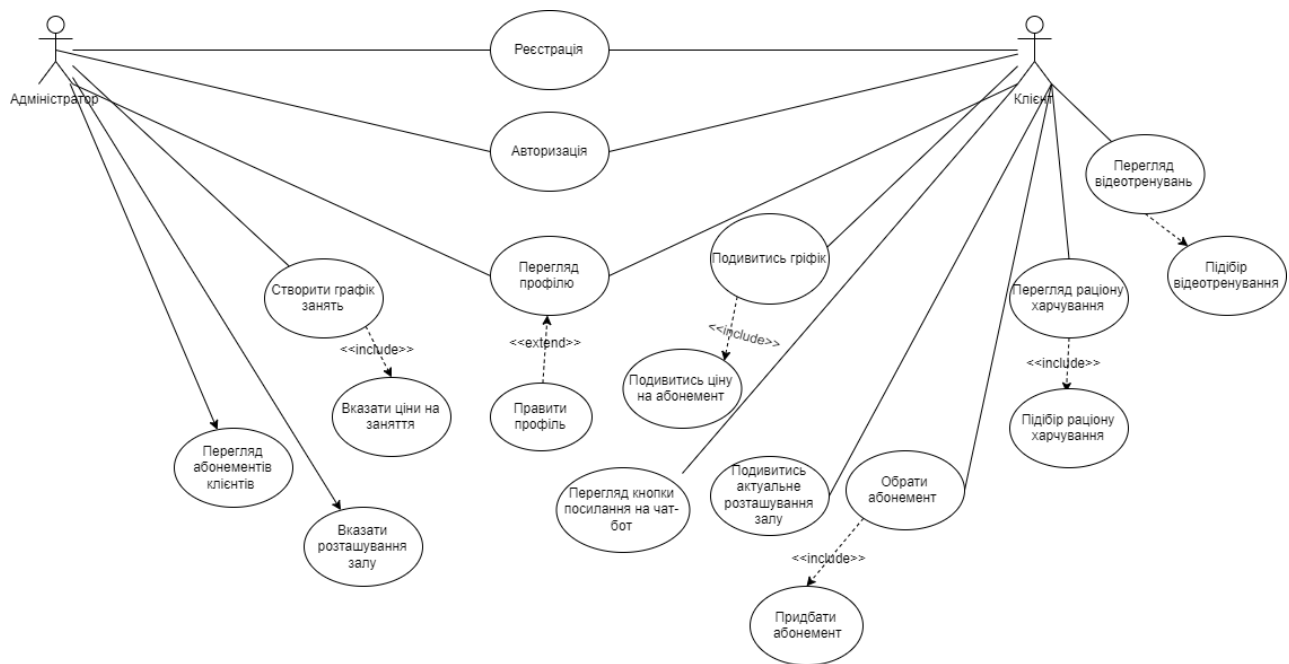


Рисунок 1.6 — Діаграма прецедентів

На рисунку ми бачимо діаграму прецедентів та її функції. Програма забезпечує повністю всі функції, щоб користувач був повністю задоволений.

Також, ми можемо отримати повний погляд на те, як розгортаються задачі, як і для клієнта, так і для адміністратора. Розширення та включення деталізують функції та особливості програмного продукту. Через таку діаграму стає краще, що потрібно розробити та кому саме підходить програмний продукт.

1.6 Діаграма розгортання

Діаграма розгортання - графік, що показує вузли, які були обчислені при запусненій програмі. Компоненти — це вигляд робочого екземпляра блоку коду.

Діаграма створена для показу програмного забезпечення та обладнання у створеному додатку. За допомогою даної діаграми можна отримати повну картинку того, як буде працювати програма на апаратному рівні.

Створення вузлів має вигляд прямокутних паралелепіпедів з вказаними додатками, які знаходяться в цих прямокутних паралелепіпедах і мають вигляд прямокутників. Також, певні вузли можуть мати ще додаткові підвузли. Вони мають вигляд прямокутних паралелепіпедів. Лише один пункт, так описаний вузол може включати в себе необмежену кількість різних вузлів. Вузли поділяються на декілька типів. Одним із типів являється вузол пристрою. Він розраховує певні ресурси, які безе з пам'яті. Інший обчислює програмний ресурс.

Мною створена діаграма, яка вміщує в себе такі прилади:

- персональний комп'ютер
- сервер

Програма зберігається на сервері, завдяки якому кожен користувач має змогу зайти на сайт та подивитись інформацію. Сервер взаємодіє з базою даних, з неї береться інформація. Потрапити на сайт можна з будь-якого пристрою, де є браузер та підключення до інтернету. Зробити це можна за допомогою певного посилання, яке перекине на веб-портал.

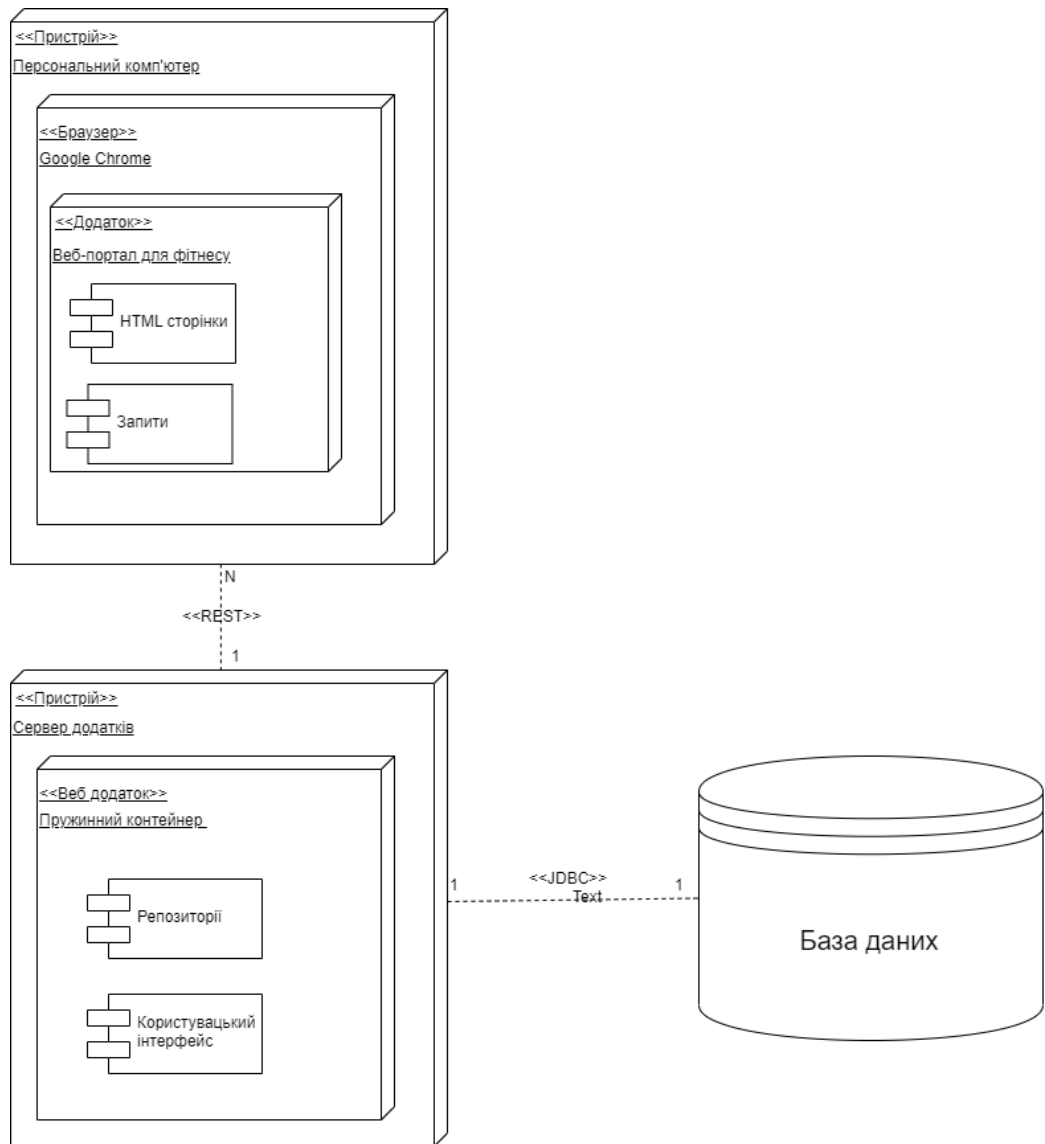


Рисунок 1.7 — Діаграма розгортання

Такий вигляд має моя діаграма розгортання. Дуже зручно переглядати, які компоненти між собою мають зв'язок.

1.7 Діаграма розгортання процесів

Для зручного перегляду різних залежностей і компонентів програми я вирішила побудувати діаграму компонентів.

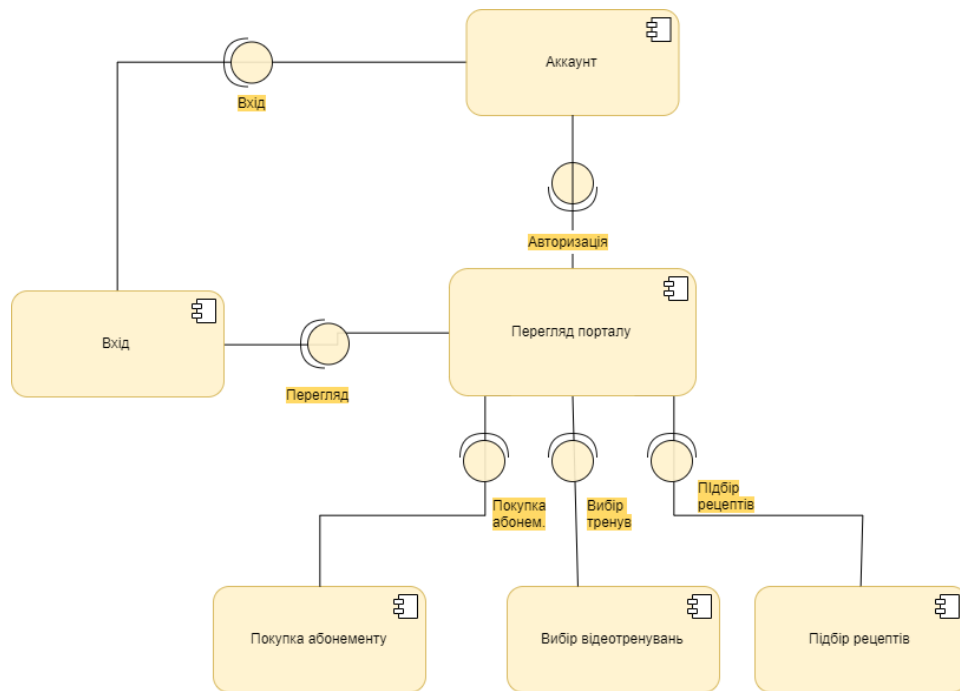


Рисунок 1.8 — Діаграма розгортання процесів

Діаграма компонентів — це певна діаграма, яка відтворює залежності компонентів. Також, вона дає розуміння, як утворюються зв'язки в програмі.

Я відтворила діаграму до свого програмного продукту, на діаграмі зображені такі пункти:

- вхід
- аккаунт
- перегляд порталу
- покупка абонементу
- вибір відео тренувань
- Підбір рецептів

2 ОПИС ПРЕДМЕТРОЇ ОБЛАСТІ

Наше життя потребує догляду духовного та фізичного. Головним чинником нашого здоров'я є догляд за нашим тілом та підтримка фізичної форми. Відвідування занять з фітнесу в спортивному залі - це запорука бути здоровим у майбутньому.

В наші часи можна обрати для себе певний вид спорту, який подобається та підходить саме вам. Займатись спортом може будь-хто. Існує безліч спортивних гуртків та секцій, які можуть відвідувати навіть маленькі діти починаючи за 4-х років. Також, важливо підтримувати здорове харчування. Тому до сайту я додала перегляд рецептів та підбір раціону харчування.

Користувач сам обирає за певними критеріями раціон. Для вибору доступні три пункти: обрати свою вікову категорію, обрати свою стать та скільки часу на день ви уділяєте тренуванням.

Багато відомих спортсменів виконавши свої поставлені цілі та здобувши медалі згодом надають свої послуги саме в фітнес та тренажерних залах. Ви можете підібрати собі тренера, який до вподоби або ходити на групові заняття. Також, можна відвідувати басейн, який дуже корисний для вашого здоров'я.

Особливо важливо дотримуватись певного режиму тренувань. Часто радять займатись тричі на тиждень, щоб ваше фізичне тіло було в певній мірі підтягнуте та ви були здорові. Отже, ходити в фітнес зал піде на користь кожній людині.

Саме для таких корисних цілей і був створений веб-портал для фітнес тренувань.

Такого веб-порталу немає аналогу в світі, але є схожі веб-сайти. Прикладами слугують Sport Life, CHLOE TING, Fitness Blender.

1.2. Sport Life

Sport Life — це сучасна мережа фітнес центрів на території України. Веб сайт Sport Life дозволяє користувачам переглянути, які існують спортивні заняття та обрати для себе абонемент, щоб відвідати певний зал, який знаходиться поруч з вашим домом [5]. Це головний приклад сайту, який я використовувала для створення свого продукту. З "Sport Life " я взяла за основу:

- які є абонементи
- графік занять
- оформлення абонементу

Головною різницею є зручний інтерфейс та швидка реєстрація, як для клієнтів, так і для адміністратора.

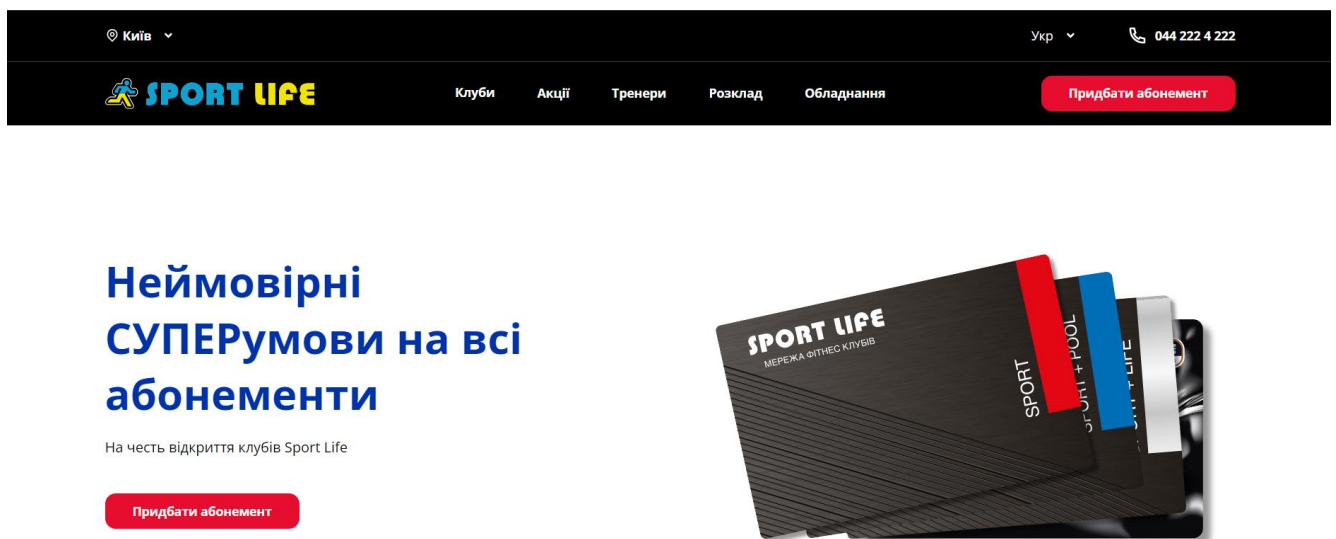


Рисунок 2.1 — Головна сторінка Sport Life

Досить непоганий сайт, але немає додаткових функцій, які допомагають клієнту пізнавати світ про фітнес тренування.

1.3. CHLOE TING

CHLOE TING — це онлайн сайт для фітнес занять, куди кожен може зайти та обрати для себе певний вид тренувань онлайн [6]. Також, можна онлайн замовити консультацію з тренером.

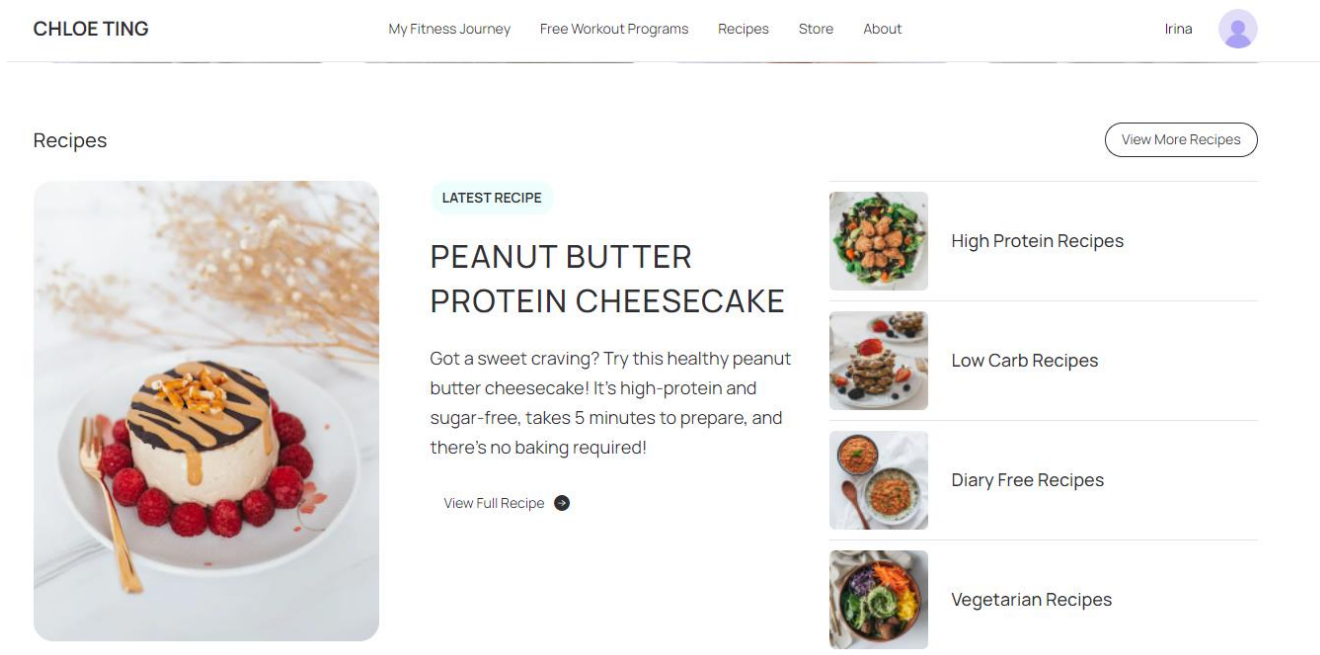


Рисунок 2.2 — Рецепти на сторінці CHLOE TING

На цьому сайті мене зацікавили аспекти, які допомагають людям розвиватись та займатись спортом вдома, коли немає можливості піти в зал. Гарно та зрозуміло виставлені на сайті онлайн тренування. Найголовніше, що мені сподобалось це рецепти салатів та сніданків, які допомагають при тренуваннях і тримати гарну фізичну форму.

Тому я вирішила додати на свій сайт розділ з правильним харчуванням.

2.3 Fitness Blender

Fitness Blender — це веб-сайт про тренування [7]. Загалом непоганий, цілком достатньо різної інформації. Можна також переглядати рецепти, але найбільше мені сподобалось, що є підбір відео тренувань, які я також додала собі до порталу.

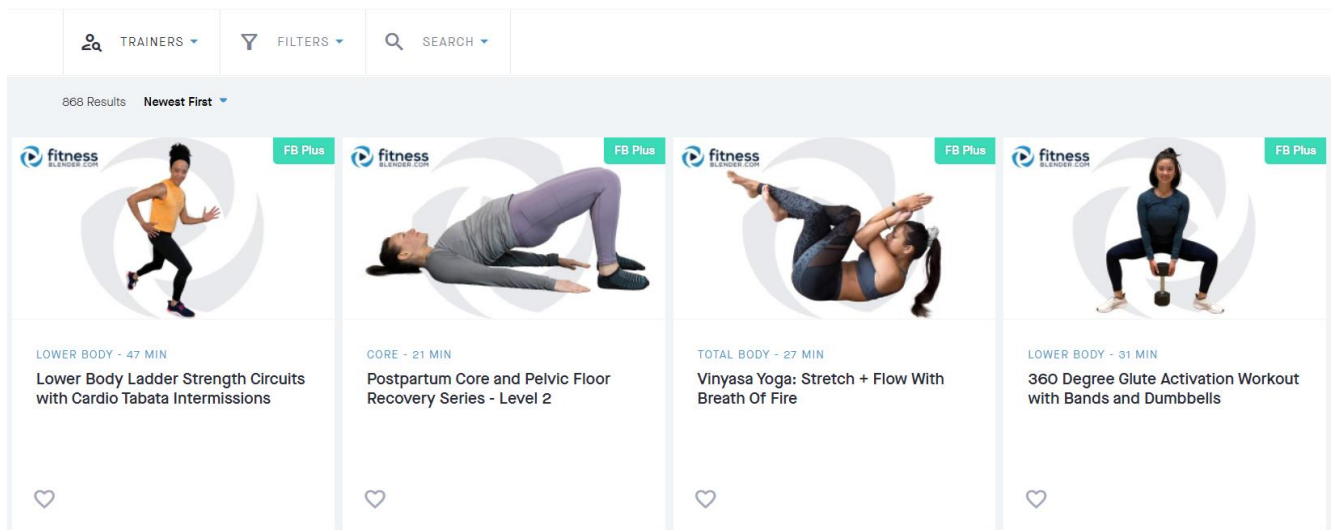


Рисунок 2.3 — Сторінка з відео тренуваннями

Сайт може переглядати будь-хто, але дивитись тренування та рецепти платно.. Можна переглядати та обирати рецепти. На сайті є відео тренування. Тренування існують короткими та довгими. Якщо ви кудись поспішаєте, то можна включити коротке тренування і позайматись, щоб не втрачати форму. Регулярні заняття дуже корисні і потрібні, щоб підтримувати вашу форму. Довгі заняття тривають близько години та для них потрібна підготовка. Можливо вам знадобляться гантелі.

Мінус сайту полягає в тому, що сайт є платним. А я реалізувала на своєму порталі безкоштовний підбір та перегляд відео тренувань. Будь-хто може відкрити та переглянути відео і почати по ньому займатись. Також, підбір тренувань швидкий та зрозумілий кожному користувачу.

2.4 Ferroom

Ferroom — це сайт спортивного залу [8]. Користувач зайшовши на сайт може переглянути абонементи, акції, контакти. Цілком зручний для використання веб-сайт. Мені сподобалось на даному сайті, що є можливість перегляду контактних даних, а саме мапи, де знаходиться спорт-зал.

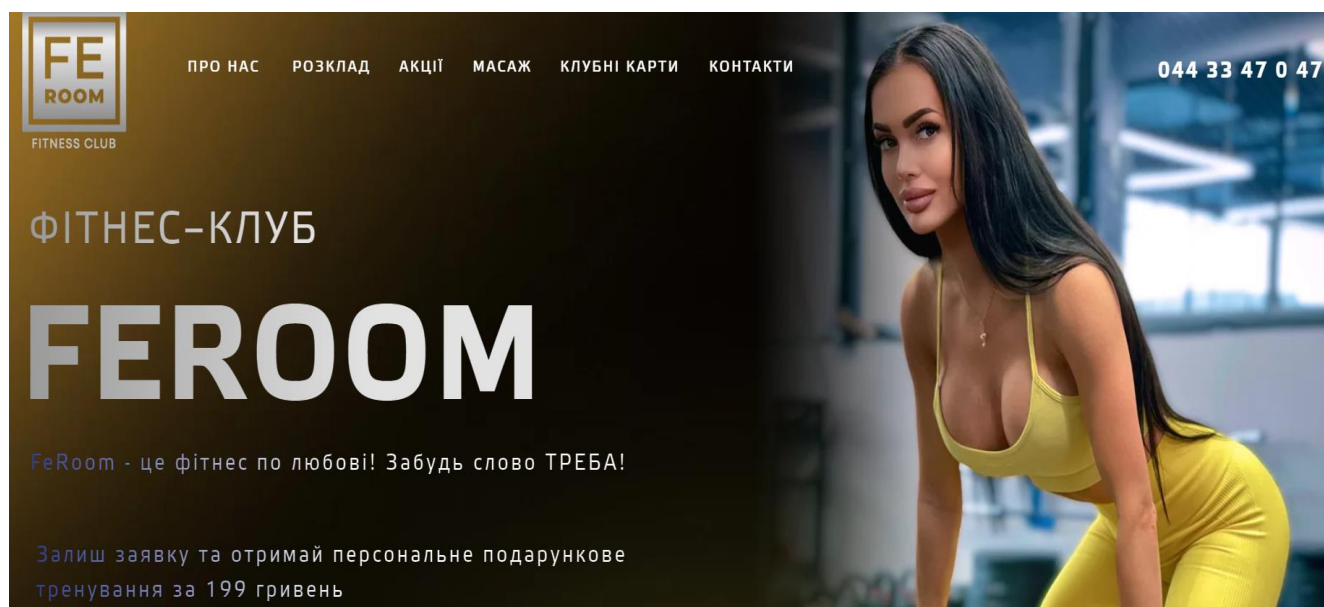


Рисунок 2.4 — Сторінка з відео тренуваннями

Тому я вирішила на свій веб-портал додати мапу, на яку можна нажати та подивитись де знаходиться фітнес-зал. Це дуже зручна функція, якою користується кожна людина, яка вперше піде тренуватись в залі.

3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДОДАТКУ

3.1 Опис моделі

Архітектура системи створена, щоб поєднувати зв'язки між різноманітними частинами системи. Я використовую систему за шаблоном MVC.

Ця веб-програма заснована на шаблоні MVC (модель-представлення-контролер).

MVC шаблон описує легкий спосіб створення побудови даної програми, спрямованої на відділення бізнес-логіки від користувацького інтерфейсу. Результатом є легше масштабування програми, тестування, обслуговування та, звичайно, впровадження.

Простіше кажучи, це спосіб організації створення програмного коду, включаючи блочних виділень, які слугують за вирішення різних проблем.

Даний спосіб поділяється на 3 різних блоки. Перший включає в себе дані програми, другий інтерфейс та останній має контроль над працюючим програмним продуктом.

У модель MVC включаються такі пункти:

- модель – цей розділ має контроль за даними та показує структуру програмного забезпечення.

- представлення – компонент, який забезпечує користувацьку взаємодію з програмою. Код у розділі «Перегляд» надає огляд і досвід роботи програми. Форматування та редагування даних вимагає форматування коду HTML та PHP. Частина цього розділу містить стандартний шаблон, який містить мітки, спільні для всіх сторінок (наприклад, "footer" і "header"), і шаблон, який відображає форму введення.

- контролер – здійснює зв'язки між компонентом моделі та компонентом перегляду. Коли користувач взаємодіє з сайтом, то його код на це реагує. До речі, це програмне забезпечення MVC.

Наприклад, на рисунку 3.1 представлено узагальнену схему системи автоматичного регулювання.

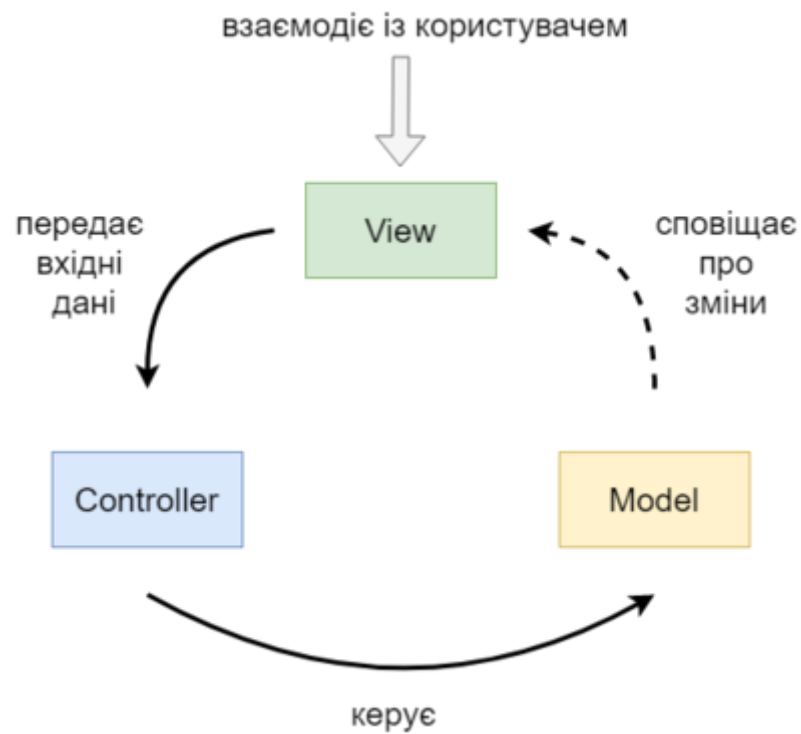


Рисунок 1.1 — Працюючі модулі MVC.

На рисунку 3.2 зображено, як при написанні мого продукту була створена певну послідовність з папок і файлів.

idea	21.05.2022 18:08	Папка с файлами	
Controllers	21.05.2022 18:03	Папка с файлами	
css	21.05.2022 18:03	Папка с файлами	
data	21.05.2022 18:03	Папка с файлами	
images	21.05.2022 18:03	Папка с файлами	
Models	21.05.2022 18:03	Папка с файлами	
partials	21.05.2022 18:03	Папка с файлами	
System	21.05.2022 18:03	Папка с файлами	
Views	21.05.2022 18:03	Папка с файлами	
._DS_Store	17.05.2022 16:15	Файл "DS_STORE"	1 КБ
._htaccess	07.05.2022 20:13	Файл "HTACCESS"	1 КБ
._idea	17.05.2022 14:27	Файл "IDEA"	1 КБ
._base.sql	17.05.2022 13:29	sql	1 КБ
._Controllers	17.05.2022 14:27	Файл "_CONTROL..."	1 КБ
._css	17.05.2022 14:27	Файл "_CSS"	1 КБ
._data	17.05.2022 14:27	Файл "_DATA"	1 КБ
._images	17.05.2022 14:27	Файл "_IMAGES"	1 КБ
._index.php	17.05.2022 13:44	Исходный файл Р...	1 КБ
._Models	17.05.2022 14:27	Файл "_MODELS"	1 КБ
._partials	17.05.2022 14:27	Файл "_PARTIALS"	1 КБ
._System	17.05.2022 14:27	Файл "_SYSTEM"	1 КБ
._Views	17.05.2022 14:27	Файл "_VIEWS"	1 КБ

Рисунок 3.2 — Послідовність з файлів та папок

Головною функцією на веб-порталі є вибір та замовлення клієнтом абонемента. Після реєстрації клієнт може переглянути абонементи та обрати собі, який найкраще підходить і його замовити. На рисунку 3.3 зображена модель, де клієнт обирає абонемент.

```

use Models\User;

class Order extends Model
{
    public const table = 'orders';

    public const TYPE_ORDER = [
        '100' => 'Разове заняття',
        '600' => '1 міс. СТАНДАРТ (12 трен.)',
        '700' => '1 міс. БЕЗЛІМІТ',
        '1400' => '3 місяці СТАНДАРТ',
        '2680' => 'пів-річний СТАНДАРТ',
        '4400' => 'річний СТАНДАРТ',
        '1700' => '3 місяці БЕЗЛІМІТ',
        '3200' => 'пів-річний БЕЗЛІМІТ',
        '5000' => 'річний БЕЗЛІМІТ'
    ];

    public function create($data)
    {
        $sql = "INSERT INTO ".self::table."(user_id, type, price)
            VALUES (:user_id, :type, :price)";

        $stmt = $this->pdo->prepare($sql);
        $stmt->bindValue(":user_id", $data['user_id']);
        $stmt->bindValue(":price", $data['price']);
        $stmt->bindValue(":type", self::TYPE_ORDER[$data['price']]);

        return $stmt->execute();
    }

    public function getOrders()
    {
        $sql = "SELECT u.name as user_name, u.phone, u.email, o.type FROM ".self::table." as o JOIN ".User::table." as u ON u.id = o.user_id";

        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();

        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
}

```

Рисунок 2.3 — Замовлення абонементу

Приклад коду зображений на рисунку 3.4, де одним з контролерів є отримання даних, які заповнив клієнт

```

<?php
namespace Controllers;

use Models\Order;
use Models\User;
class MainController
{
    public function index($view = '')
    {
        if ($view) {
            if (($view == 'login' || $view == 'reg') && !empty($_SESSION['user'])) {
                header('Location: /');
            } else {
                include "Views/{$view}.php";
            }
        }
    }
}

```

Рисунок 3.4 — Вигляд контролерів

Цей драйвер перевіряє, чи зареєстрований користувач у системі. Тут ми використовуємо модель даних, яка надсилається або отримується з точки зору.

За графічний вигляд відповідає код, який зображений на рисунку 3.5, тут створені:

- вхід
- головна сторінка
- перегляд абонементів
- перегляд рецептів
- замовлення абонементів
- контакти дані та карта, де знаходиться зал

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Sport Life - Головна сторінка</title>
  <!-- links -->
  <link rel="icon" href="/images/logo.png">
  <link rel="stylesheet" href="/css/index.css">
</head>
<body>

  <?php include 'partials/header.php'?>

  <div class="main-window">
    <div class="main-description">
      <h2>Займайся спортом</h2>
      <p>Наш зал обладнаний сучасними тренажерами на яких ви зможете отримувати задоволення під час тренування та ефективно проводити свій час. </p>
      <p>Знаходимося ми в самому центрі міста та працюємо з 8:30 до 20:30.</p>
      <?php if(isset($_SESSION['user'])) && $_SESSION['user']['name'] != 'admin':?>
      |   <a href="/order">Придбати абонемент</a>
      <?php endif?>
      <?php if(!isset($_SESSION['user'])):?>
      |   <a href="/login">Придбати абонемент</a>
      <?php endif?>
    </div>
  </div>
  <a name="time"></a>
  <div class="block-time">
```

Рисунок 4.5 — Інтерфейс користувача

На такій структурі тримається весь сайт. Дає розробити продукт швидко та якісно.

3.2 Опис структури бази даних

При створенні веб-порталу була розроблена база даних. Щоб швидко її створити я використала вільну систему керування БД MySQL.

За допомогою коду, який зображений на рисунку 3.6 відбулося основне підключення БД.

```
<?php

namespace Models;
class Model {
    public $pdo;

    public function __construct()
    {
        $dbOptions = ['host' => 'localhost', 'dbname' => 'sport', 'user' => 'user', 'password' => '1234'];

        $this->pdo = new \PDO(
            'mysql:host=' . $dbOptions['host'] . ';dbname=' . $dbOptions['dbname'],
            $dbOptions['user'],
            $dbOptions['password']
        );
        $this->pdo->exec('SET NAMES UTF8');
    }
}
```

Рисунок 5.6 — Підключення бази даних

Основним пунктом для адміністратора сайту є облік клієнтів. Адміністратор в базі даних, яка зображена на рисунку 3.7 отримує дані про клієнта, який купив абонемент. Він може переглянути хто купив абонемент, який саме , ціну та дату покупки.

☐		Редагувати		Копіювати		Видалити	1	річний БЕЗЛІМІТ	3	2022-05-17 19:44:37	5000
☐		Редагувати		Копіювати		Видалити	2	3 місяці СТАНДАРТ	5	2022-05-21 15:44:06	1400
☐		Редагувати		Копіювати		Видалити	3	1 міс. БЕЗЛІМІТ	5	2022-05-25 12:10:21	700
☐		Редагувати		Копіювати		Видалити	4	річний СТАНДАРТ	6	2022-05-25 20:33:55	4400
☐		Редагувати		Копіювати		Видалити	5	3 місяці БЕЗЛІМІТ	7	2022-05-25 20:35:16	1700
☐		Редагувати		Копіювати		Видалити	6	пів-річний СТАНДАРТ	5	2022-05-26 10:44:39	2680
☐		Редагувати		Копіювати		Видалити	7	1 міс. СТАНДАРТ (12 трен.)	5	2022-05-26 13:17:49	600
☐		Редагувати		Копіювати		Видалити	8	3 місяці СТАНДАРТ	5	2022-05-26 19:52:38	1400
☐		Редагувати		Копіювати		Видалити	9	3 місяці СТАНДАРТ	5	2022-05-26 19:52:42	1400
☐		Редагувати		Копіювати		Видалити	10	1 міс. СТАНДАРТ (12 трен.)	5	2022-05-26 21:39:53	600
☐		Редагувати		Копіювати		Видалити	11	1 міс. БЕЗЛІМІТ	5	2022-05-27 14:44:51	700

Рисунок 3.7 — БД із зібраними даними

Перегляд БД є дуже зручним та зрозумілим.

Також, моя база даних створена саме реляційним методом. Який є дуже популярним та найзручнішим.

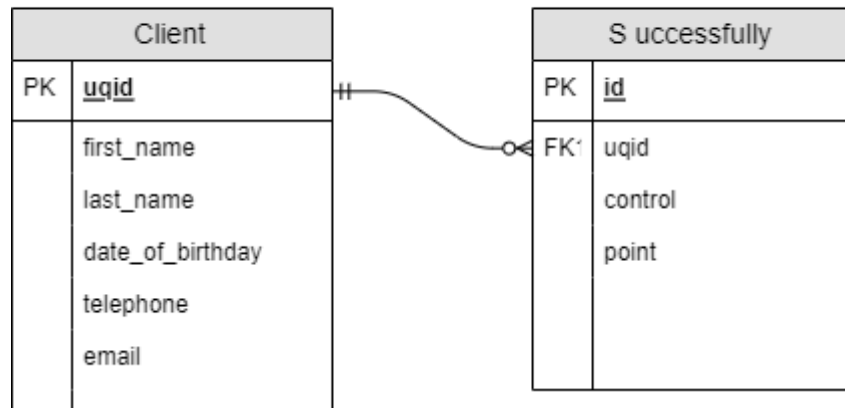


Рисунок 3.8 — Схема таблиць бази даних утворених реляційним методом

Швидко та зрозуміло можна переглянути всі дані.

3.3 Опис запрограмованого чат-боту

Сьогодні дуже актуальним є створення чат-боту в телеграмі. Кожна людина заходить в даний месенджер кожен день. Ми постійно спілкуємось з людьми, вирішуємо проблеми через даний месенджер.

Чат-боти можуть гарно спілкуватись з користувачами в повідомленнях. Бот розпізнає та виконує різні дії, які були запрограмовані.

Тому я вирішила створити бот, який відслідковує біоритм.

Біоритми бувають різні, але я обрала біоритм, який відслідковується за датою дня народження та поточним днем і часом.

```
{
  "statusCode": 150,
  "count": 5862555,
  "message": "Ok",
  "data": [
    {
      "first_name": "Інна",
      "Id": "14",
      "last_name": "Слободяник",
      "date_of_birthday": "05.04.2004",
      "telephone": "+38054582651",
      "email": "innablack@gmail.com",
      "user_url": "https://ua.000webhost.com/members/website/unoxidized-births/database",
    },
    {
      "first_name": "Ігорь",
      "Id": "10",
      "last_name": "Лисенко",
      "date_of_birthday": "04.04.2002",
      "telephone": "+380505045514",
      "email": "igorban@ukr.net",
      "user_url": "https://ua.000webhost.com/members/website/unoxidized-births/database"
    }
  ]
}
```

Рисунок 3.9 — Отримання даних за допомогою JSON

Приклад формули зображень на рисунку 3.10, який виводить біоритм.

```
def calculate biorhythm(birthdate):
    today = datetime.date.today()
    delta = today - birthdate
    days = delta.days
    phisical = math.sin(2*math.pi*(days/23))
    emotional = math.sin(2*math.pi*(days/28))
    intellectual = math.sin(2*math.pi*(days/33))
    biorhythm = {}
    biorhythm['phisical'] = int(phisical * 10000)/100
    biorhythm['emotional'] = int(emotional * 10000)/100
    biorhythm['intellectual'] = int(intellectual * 10000)/100
    biorhythm['phisical_critical_day'] = (phisical == 0)
    biorhythm['emotional_critical_day'] = (emotional == 0)
    biorhythm['intellectual_critical_day'] = (intellectual == 0)
    return biorhythm
```

Рисунок 3.10 — Формула розрахунку біоритму

Бот виводить дані про фізичний стан, емоційний та інтелектуальний

Клієнт, який пройшов реєстрацію на сайті, він заповнив інформацію та вказав дату народження. З бази даних сайту бот отримує дані.

4 ЗАСОБИ РОЗРОБКИ

Після того, як я провела повний аналіз своєї предметної області я вирішила розробити додаток з клієнт-серверною архітектурою. Програмний продукт може запускатись, як веб-програма. Розглянемо програмні засоби, за допомогою яких була створена система. Було створено дані пункти:

- розробка серверної програми
- розробка веб застосунку
- розробка чат-боту

4.1 Серверна частина

В розробці даної програми, яка стосується серверної частини були застосовані тільки найсучасніші інструменти. При розробці серверної частини була побудована база даних зі зв'язками, які допомагають проводити певні операції з роботу бази даних.

4.1.1 Мова програмування PHP

Я обрала для себе зручну та досить популярну мову програмування – PHP.



Рисунок 4.1 — Логотип мови PHP

PHP — являється скриптовою мовою програмування.

Мовний синтаксис бере початок від таких мов програмування, як Java, C та Perl. Для опанування дуже легкий. Одна з найпопулярніших мов програмування, яка має використання у веб-розробці (разом з Ruby, .NET, Python, Java, JavaScript). Також, PHP може підтримуватись майже усіма хостинг-провайдерами. У порівнянні зі скриптовою мовою JavaScript звичайний користувач не зможе подивитись код, який був написаний на PHP [1]. Це через те, що браузер відразу отримує готовий HTML код. З боку безпеки саме це є перевагою, тільки є один мінус — погіршується інтерактивність сторінок. Щоб таке усунути, можна генерувати JavaScript-коди.

PHP - це мова програмування, яка у свій код може вміщувати html-код, який буде справно працювати та правильно оброблятися PHP-інтерпретатором. Після використання тегу (`<?php`) код починає виконуватись і закінчується при закритті тегу.

Певний набір функцій, а їх досить є різноманітна кількість, PHP пропонує уникнути велику кількість рядків при написанні багаторядкових функцій, як в мовах програмування Pascal або C.

Дуже важливим додатком у PHP є вбудовані бібліотеки для праці з PostgreSQL, MySQL, InterBase, Sybase, mSQL, SQLite, Oracle, dbm, Hyperware.

Також, дуже легко починати програмувати на PHP, якщо ви раніше програмували на Perl або C. Це допомагає швидше опанувати нову мову. PHP — це мова програмування, яка спрямована саме на роботу в Інтернеті.

Користуватись РНР може будь хто, так як мова програмування являється безкоштовною. Також, існує дуже гарна підтримка РНР, люди зі всього світу вирішують різні задачі та складні питання саме на цій мові і можна знайти різні відповіді на питання, які вас бентежать.

Основним гарним фактором є ефективність у програмуванні. Завдяки РНР можна швидко обробляти різні сценарії, ніж написані на Perl. І цілком продуктивність РНР є досить хорошою для створення різноманітних веб-застосунків.

4.1.2 Система керування базами даних

Система керування базами даних — це сукупність набору пов'язаних між собою даних (баз даних) та програмних продуктів для користування та доступу до даних, яких потрібно [3]. Програмні складові СКБД мають основні інструменти, які використовуються для ведення адміністратором баз даних. Можна швидко створити, оновити та знайти потрібну інформацію в базах даних.

Основними характеристиками СКБД являється несуперечливість даних, використання даних не однією людиною, високий та захищений рівень безпеки даних, можливість повністю контролювати дані за їх надлишковістю, несуперечливість та коректність баз даних, також незалежне використання різних програм з даними.

Існує доволі невеликий список можливостей використання СКБД. Дозволено оновляти, додавати, читати та видаляти інформацію з Бази Даних. Доступ до контроль з БД можна надавати за допомогою: системи відслідковування паралельної роботи програм, які тримають під контролем процеси спільного доступу до бази даних. Сама архітектура СКБД складається з трирівневої системи ANSI-SPARC, у якій є окремий рівень, який ізолює програму від функцій представлення даних нижнього рівня. Існують такі рівні: зовнішній (бази даних,

які представлені користувачу), концептуальний (широке уявлення про базу даних, що описує дані, які зберігаються в базі даних та зв'язки між ними, є підтримка на внутрішньому рівні) та внутрішній рівень (так представляються бази даних в самому комп'ютері).

Досить популярними СКБД є PostgreSQL, Oracle, Microsoft SQL Server, Interbase, Firebird, IBM Db2.

При перегляді різних СКБД я вирішила для себе використовувати MySQL. Оскільки ця система керування є перевіреною, дуже надійною і виконує повністю всі функції, які потрібні для системи. Також, вона часто використовується в поєднанні з PHP.

MySQL- це найпопулярніша СКБД, яка має систему керувань з реляційними БД. MySQL використовується у різних програмах, зокрема зустріти можна на веб-серверах, створює динамічні сторінки і має гарну допомогу від різних мов програмування.



Рисунок 4.2 — Логотип СКБД MySQL

Користування MySQL є простим та зручним.

4.1.3 Середовище розробки IntelliJ Idea

IntelliJ IDEA — це створене інтегроване середовище розробки, в якому можна писати на будь-якій мові програмування (PHP, Python, Java, Scala). Розроблена компанією JetBrains. Система програми являється у вигляді функціональної комерційної версії «Ultimate Edition» та версії «Community Edition», яка являється безкоштовною. Community-версії мають Сирцеві тексти, які мають ліцензію Apache 2.0. Збірки бінарні можуть слугувати як для Mac OS X, Linux та Windows.

Компоненти IntelliJ IDEA розроблені для максимального використання, щоб розвинути швидкість розробки програмного продукту. IntelliJ IDEA являється розумним редактором коду та зі зручним інтерфейсом роблять написання коду приємним та зручним і дуже ефективним. Розумний редактор дуже гарно спрощує роботу при написанні коду та чудово справляється з допомогою при написанні коду. Можна швидко та легко знаходити елементи в списку, підлаштовувати налаштування під себе, змінювати теми фону, щоб було зручніше працювати.

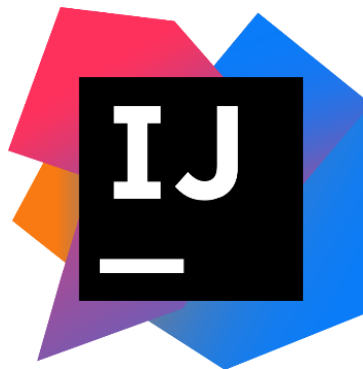


Рисунок 4.3 — IntelliJ IDEA логотип

IntelliJ IDEA була використана при створенні серверної частини.

4.1.4 Workbench - середовище управління баз даних

Щоб справно працювати з різними базами даних потрібно використовувати програму, яка підходить. Прикладом слугує програма MySQL Workbench, яку створила компанія Microsoft.

MySQL Workbench - Візуальний інструмент для проектування бази даних, який об'єднує проектування, моделювання, створення та маніпулювання базою даних в єдине безкоштовне середовище для системи баз даних MySQL. Він є наступником FabForce з DBDesigner 4 . MySQL Workbench поставляється в двох версіях: Standard Edition та Community Edition.



Рисунок 4.4 — MySQL Workbench логотип

Використання даної програми дає такі функції у використанні:

- під'єднання баз даних і управління ними
- можливість перевіряти та переглядати дані, які внесені до бази даних
- можна редагувати та створювати структури бази даних
- заповнення тестовими даними бази даних

4.2 Веб-портал

Щоб веб-додаток вірно працював необхідно було виконати такі пункти:

- створення дизайну сторінок веб сайту
- верстка сторінок
- поєднати весь функціонал, щоб точно відображались введені дані з серверу у веб-порталі

4.2.1 Інструменти для побудови веб-порталу

Щоб зверстати сторінки потрібно було використати HTML і CSS [2]. За допомогою цих інструментів ми можемо бачити зовнішній вигляд сайту.

HTML — це мова, створена для зручного огляду сторінок у браузері. Мова розмітки, яка зберігає в собі певну інформацію. При використанні HTML ми позначаємо текст, який хочемо, щоб побачили у браузері. Вся інформація є збереженою в певних блоках, так званих кластерах та за допомогою певної структури комп'ютер сприймає інформацію, яку потрібно зчитувати, а яку ні. HTML дає змогу для утворення різних структурних документів, які позначають структуровану семантику. Прикладом є посилання, абзаци, цитати, заголовки.

Користуватись HTML можливо за допомогою тегів (це елемент, який записується в такі дужки < >). Майже кожен тег при використанні потрібно відкрити та закрити. Також, на веб сайті користувач може сам переглянути код написаний на HTML.



Рисунок 4.5 — Логотип HTML

CSS (Cascading Style Sheets — таблиці стилів каскадні) - це унікальна мова стилів сторінок. Зазвичай, вона описує зовнішній вигляд, додає стилю сторінкам, які написані на HTML.



Рисунок 4.6 — Логотип CSS

Також, можна використовувати до сторінок, які написані на XHTML та з іншими видами XML-документів. Перевагами CSS являється наявність того, що

можна розділити зміст від оформлення сторінки. Також, можна відтворювати одні і ті самі документи, але в зовсім різних стилях та різному оформленні.

4.3 Чат-бот

Реалізація чат-боту була створена мовою Python в середовищі PyCharm [11].

Python це мова програмування, яка проста у використанні. Має різноманітність бібліотек, які допомагають при створенні різноманітних алгоритмів, програм. У порівнянні з мовою C виправляє помилки швидше та краще, бо має інтегровані типи даних різних рівнів, а також високих.

Python дає можливість розділяти програмний продукт на різні модулі, які згодом можуть бути використані в інших розробках.

Python інтерпретована мова, яка значно економить час компіляції. Таким чином можна створювати швидко шаблони програм та їх тестувати.



Рисунок 4.7 — Логотип Python

Функціонал Python має доволі великі відмінності в порівнянні з іншими мовами. Перелік головних відмінностей:

- автоматичне використання пам'яті
- швидкий спосіб виконання різних задач та операцій

Також, Python є популярною мовою програмування.

PyCharm редактор коду для Python. Має зручний та простий інтерфейс для використання. Підтримує різні інструменти за допомогою яких можна швидко написати програмний код.



Рисунок 4.8 — Логотип PyCharm

Підтримується PyCharm на різних операційних системах, таких як Mac OS, Windows, Linux.

PyCharm досить гарно допомагає при створенні програмного продукту, є швидка можливість написати код та виявити помилки в коді, які підсвічуються, коли вони з'являються. Також, є можливість додавати автоматичні засоби, при

починанні вписування коду, різних функцій PyCharm сам дописує. Такий спосіб називається автоматичним рефакторингом. Навігація даного додатку є зручною та надає можливості миттєво переходити з одного проекту в інший.

PyCharm має досить широкий набір інструментів, які допомагають швидко вбудовувати та запускати тести, які написані мовою Python.

Також, має повний функціонал та інтеграцію з Vagrant і Docker. Підтримує різні фреймворки, прикладом яких є Pyramid, Django, Flask та Google App Engine. Підтримує мови програмування CoffeeScript, JavaScript, CSS і HTML.

PyCharm має вбудований відкладчик, за допомогою якого можна користуватись JavaScript в Django.

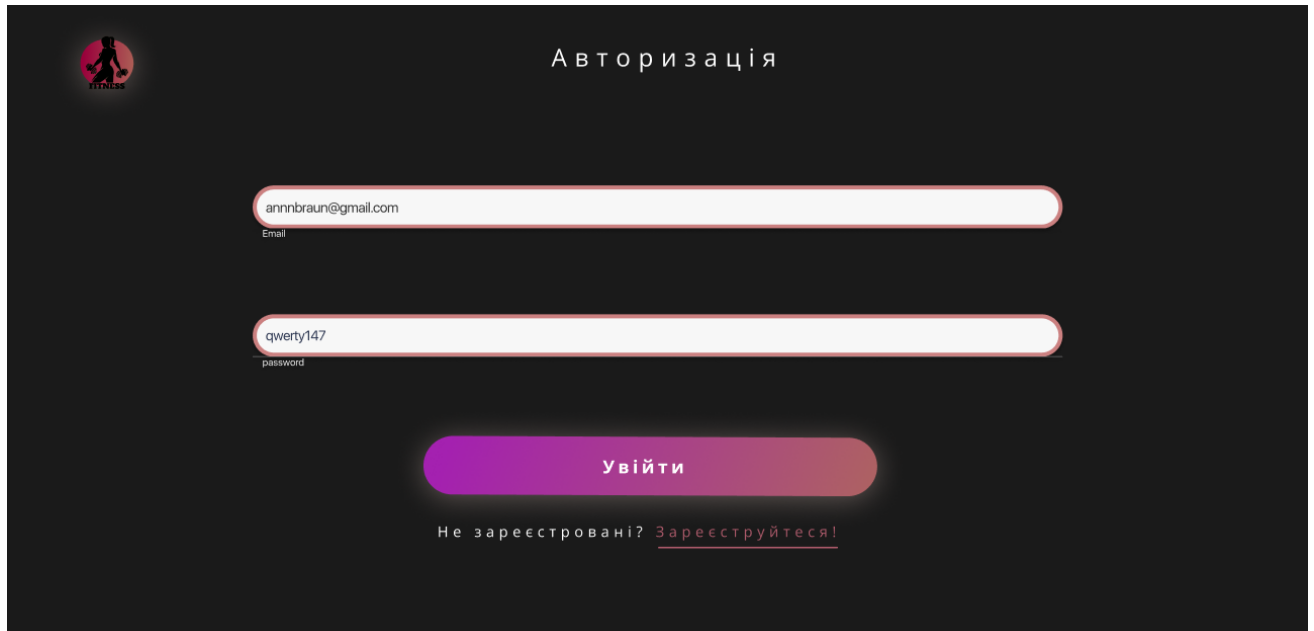
Отже, це найзручніше середовище для програмування. Має інтуїтивний дизайн, поєднує різні файли і надає швидкий доступ до їх керуваннями.

5 КОРИСТУВАННЯ РОБОЧИМ ВЕБ-ПОРТАЛОМ ТА ЧАТ-БОТОМ

Даним сайтом може користуватись будь-хто, так як система користування дуже зручна та проста. Різні сайти зроблені майже за одним шаблоном, щоб користувач швидко почав працювати з сайтом та отримав інформацію, яку йому потрібно і не шукав куди потрібно нажимати.

Клієнт сайту може зайти та переглянути повну інформацію про фітнес зал та обрати тренування, які йому підходять і відразу купити абонемент. Адміністратор в своєму кабінеті може переглянути, хто відвідує зал та змінювати вміст на сайті.

Спочатку при використанні веб-порталу потрібно увійти на сайт, пройти реєстрацію. Адміністратор та клієнт вводять свій логін та пароль . Якщо клієнт, ще не зареєстрований на сайті він повинен пройти повну реєстрацію.



The image shows a login window titled "Авторизація" (Authorization) on a dark background. In the top left corner, there is a logo featuring a silhouette of a person with a dumbbell. The main area contains two input fields: the first is for the email address, with the placeholder text "annnbraun@gmail.com" and the label "Email" below it; the second is for the password, with the placeholder text "qwerty147" and the label "password" below it. Below these fields is a large, rounded button with a gradient from purple to pink, labeled "Увійти" (Login). At the bottom, there is a link that says "Не зареєстровані? [Зареєструйтеся!](#)" (Not registered? [Register!](#)).

Рисунок 5.1 — Вікно авторизації

The image shows a registration form titled "Реєстрація" (Registration) on a dark background. In the top left corner, there is a small logo featuring a stylized figure. The form consists of five input fields, each with a label below it: "Name" (containing "Ann"), "Surname" (containing "Braun"), "Phone number" (containing "+3898558754"), "Email" (containing "annnbraun@gmail.com"), and "password" (containing "qwerty147"). Below these fields is a large, rounded rectangular button with a gradient from purple to pink, labeled "Зареєструватись" (Register).

Рисунок 5.2 — Вікно реєстрації

Використання програмного продукту для клієнта та адміністратора має певні відмінності. Спочатку розглянемо клієнта, як користувача, а потім адміністратора.

5.1 Користувач — клієнт

Так, як клієнт зареєструвався на сайті, на рисунку 5.3 зображено перегляд існуючих абонементів. Можна ознайомитись з абонементом та обрати актуальний для себе.

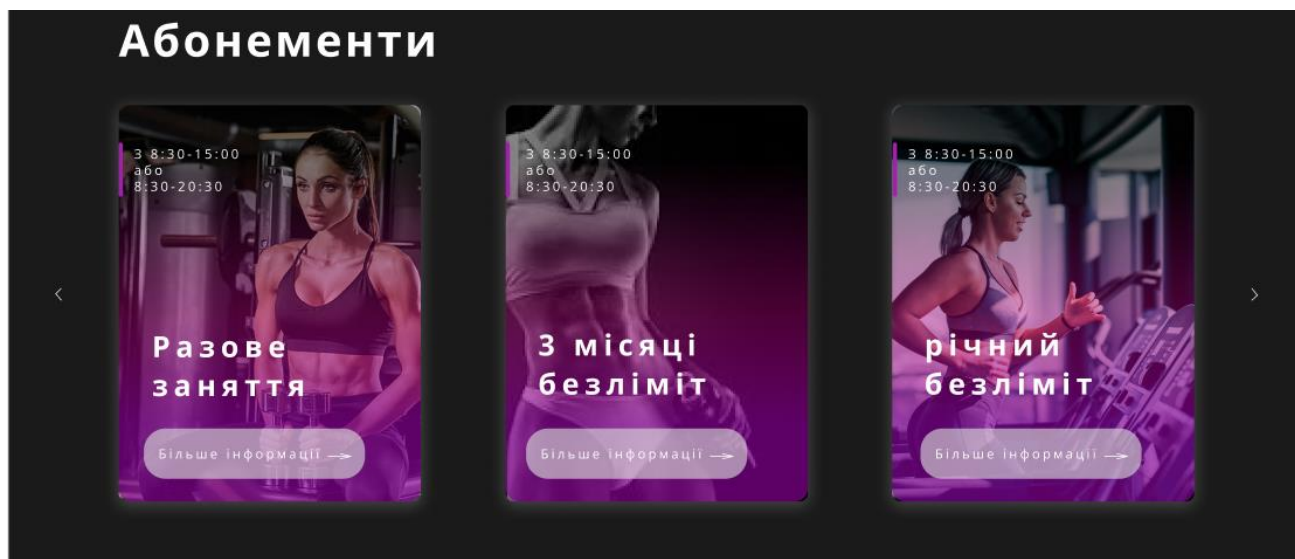


Рисунок 5.3 — Перегляд абонементів

Наступним кроком можна побачити на рисунку 5.4 актуальні рецепти, які дуже корисні та важливі, якщо збираєшся відвідувати фітнес.

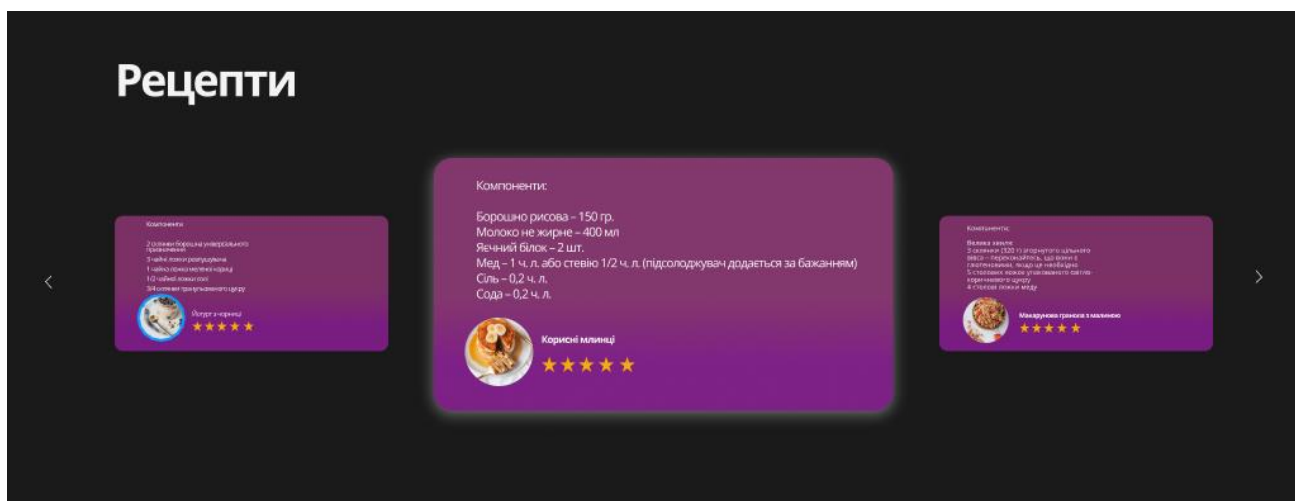


Рисунок 5.4 — Перегляд рецептів

Основною функцією являється придбання абонементу. Користувач може натиснути на кнопку "Придбати абонемент" і його перекине на сторінку, де він може обрати та забронювати абонемент.

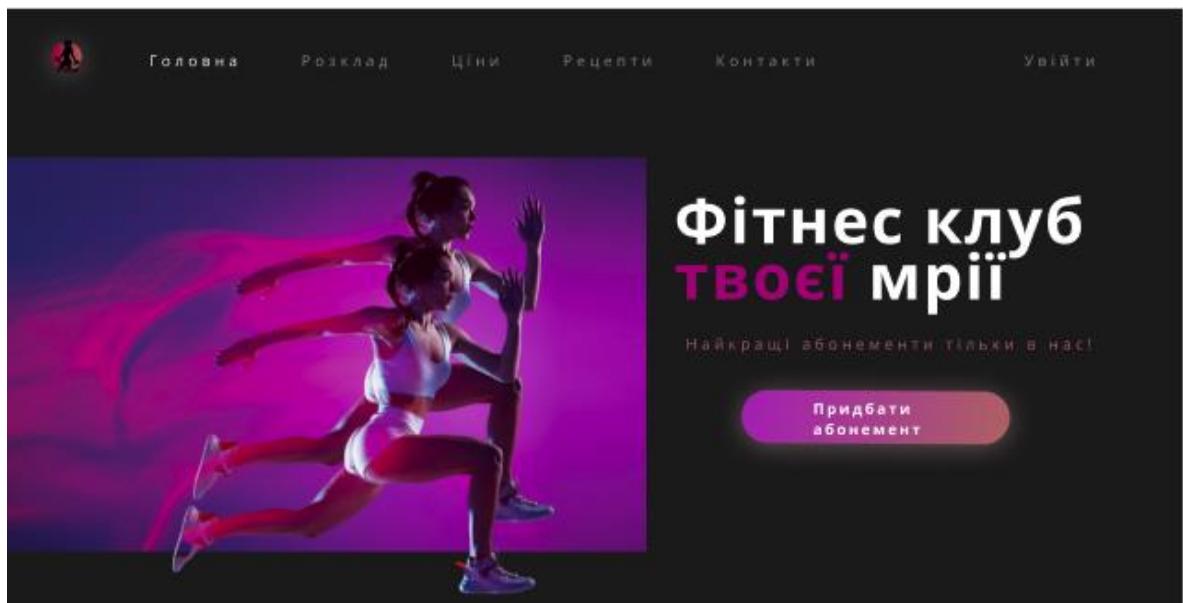
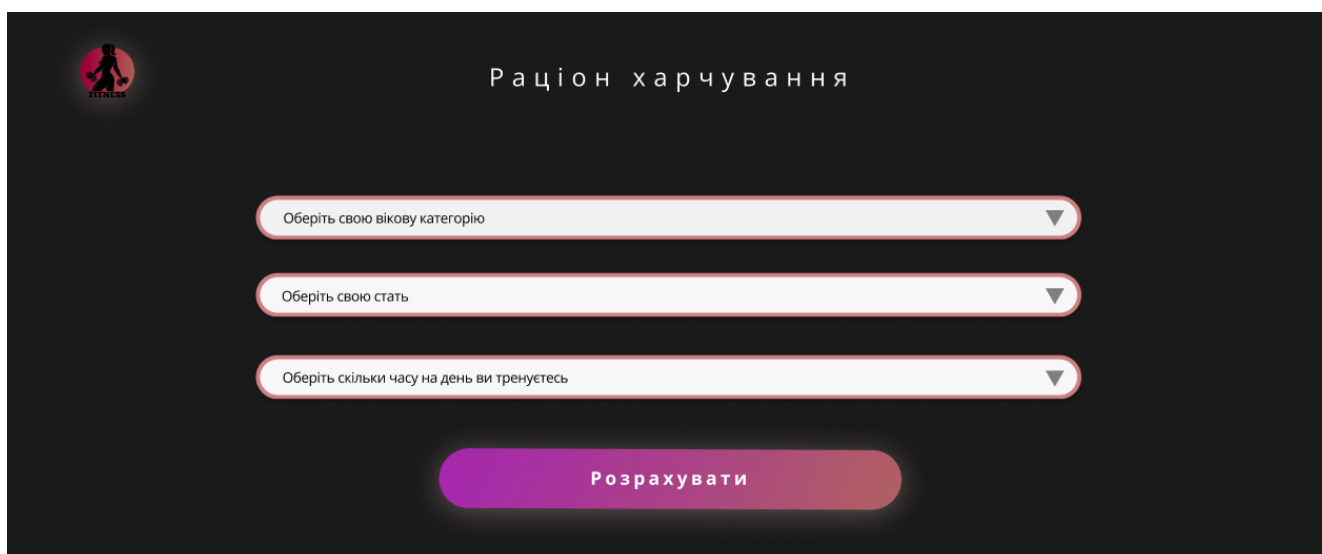


Рисунок 5.5 — Кнопка придбання абонементу

The image displays a membership selection form on the website. The form is set against a dark background with a purple-to-pink gradient. It begins with a personalized greeting 'Привіт, Ірино!' (Hello, Irina!) and the title 'Форма для заповнення абонементу' (Form for filling out membership). The form contains two input fields: the first is for the membership type, currently showing '1 міс. СТАНДАРТ (12 трен.)' (1 month. STANDARD (12 train.)), and the second is for the price, showing 'від 600 грн' (from 600 UAH). At the bottom of the form is a rounded button with a gradient, labeled 'Забронювати абонемент' (Book membership).

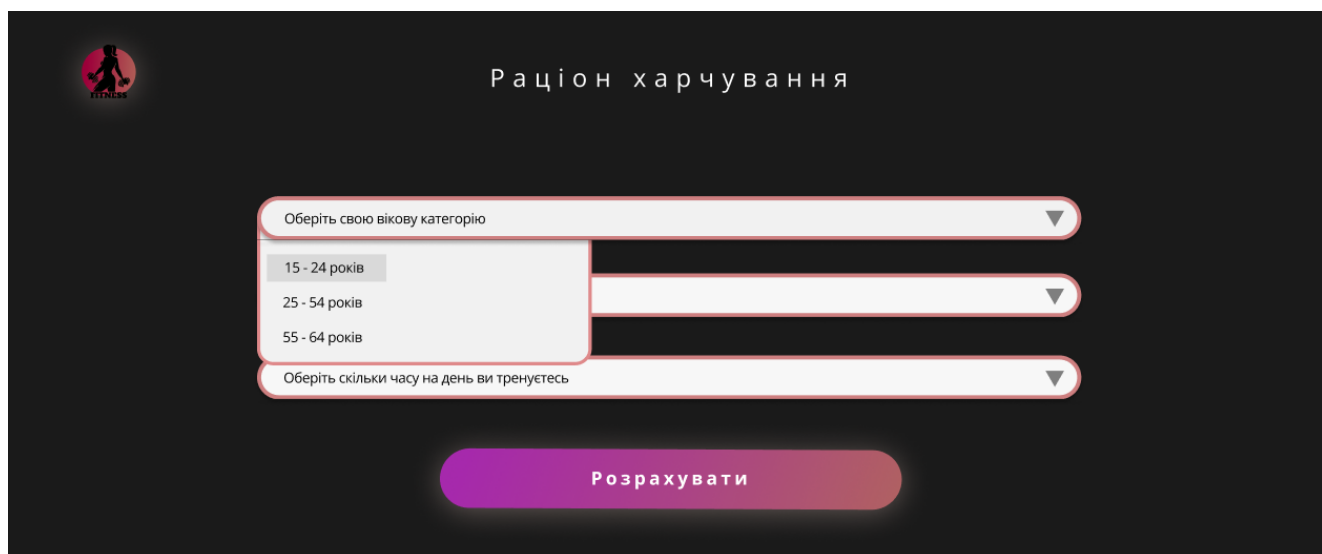
Рисунок 5.6 — Форма для вибору абонементу

Також, користувач може обрати раціон харчування, який йому підходить.



The screenshot shows a web form titled "Рацион харчування" (Diet) on a dark background. In the top left corner is a logo with a silhouette of a person running and the word "FITNESS". The form contains three white dropdown menus with red borders and downward-pointing arrows. The first menu is labeled "Оберіть свою вікову категорію" (Select your age category), the second "Оберіть свою стать" (Select your gender), and the third "Оберіть скільки часу на день ви тренуєтесь" (Select how much time you train per day). Below these menus is a large, rounded purple button with the text "Розрахувати" (Calculate).

Рисунок 5.7 — Форма для вибору раціону харчування



This screenshot shows the same "Рацион харчування" form, but the first dropdown menu is open, displaying a list of age categories: "15 - 24 років", "25 - 54 років", and "55 - 64 років". The "15 - 24 років" option is highlighted with a grey background. The other two dropdown menus and the "Розрахувати" button remain in the same positions as in the previous image.

Рисунок 5.8 — Заповнення даними



Рацион харчування

Оберіть свою вікову категорію

15 - 24 роки

25 - 54 роки

55 - 64 роки

Оберіть об'єм часу на день ви тренуєтесь

Розрахувати

Ваш ідеальний раціон:

Для цього віку характерно вкрай нерегулярне харчування, щедро приправлене фаст-фудом і снеками, вживанням алкоголю, активним курінням, дотриманням жорстких дієт - всі ці фактори мають негативний вплив на здоров'я і не відповідають уявленню про те, як виглядає раціон здорового харчування. Однак завдяки загальній «фортеці» організму і швидкому обміну речовин, наслідки від такого способу життя в молодості практично непомітні. Але це не означає, що турботою про здоров'я можна нехтувати.

Саме в молодому віці необхідно «запасатися» вітамінами, а також мікроелементами, тому основний упор варто робити на продукти, що містять цинк, хром, магній, вітаміни групи B. Здорове харчування для чоловіків і жінок має містити яйця, цільнозерновий хліб, морепродукти, горіхи, гриби, бобові, червоне м'ясо, шоколад.

Молодим дівчатам в цьому віці слід звернути увагу на зміну ваги в другій половині менструального циклу, якщо вони не приймають оральні контрацептиви. Адже біологічно закладено, що жінки перед початком нового менструального циклу схильні набирати вагу внаслідок активної вироблення прогестерону, який відповідає за вагітність. Не варто переживати з цього приводу, адже в новому циклі ця вага піде сам по собі. Через те що в молодому віці обмінні процеси протікають швидко, то за тиждень з легкістю можна позбутися від 2-3 кілограмів.

Індивідуальний раціон харчування у віці до 30 років може виглядати так:

Сніданок:
свіжовичавлений цитрусовий сік
150 гр нежирного сиру / 100 мл кефіру / мюслі з нежирним молоком, чай або кава;

Перекус о 11:00:
яйце круто / половина філе курячої грудки / 1 помідор / 1 яблуко;

Обід:
морепродукти / 150 гр червоної квасолі / теляча печінка, листя салату / два свіжих помідора з сіллю;

Перекус о 17:00:
1 банан / 150 гр пюре з фруктів / 150 гр нежирного сиру;

Вечеря:
овочевий / рибний суп, дві скибочки нежирної шинки / салат з морепродуктів, зелений салат.

Рисунок 5.9 — Отримання раціону харчування

Користувач може обрати відео за яким буде займатись. Тренування підходять всім, тому що рівень для початківців.

52



Рисунок 5.10 — Вибір та перегляд відеотренувань

Далі відвідувач сайту може відкрити карту. Мапа інтегрована в сайт, інтерактивна карта від сервісу google maps, яка дозволяє користувачу подивитись, де знаходяться спортивні центри та обрати для себе найбільш підходящий.

Карту можна скролити, переглядати геопозиції філій, дивитися адреси. Також для зручнішої взаємодії її можна відкрити у новій вкладці (переадресація на сайт google maps). Далі користувача веде на перегляд чат-боту. Користувач, який зареєструвався на веб-порталі може перейти за посиланням в чат-бот та відразу натиснути /biorhythm і йому видають інформацію.

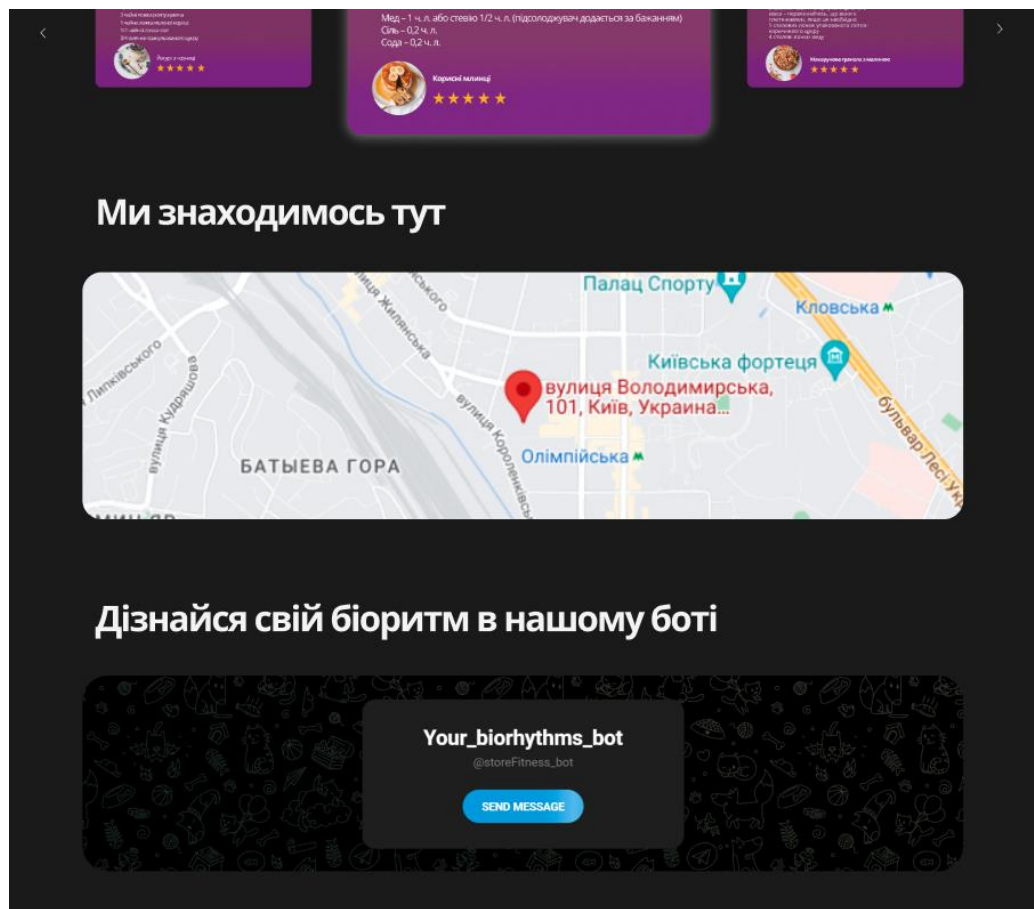


Рисунок 5.11 — Перегляд мапи та посилання на бот

Мапа інтегрована в сайт інтерактивна карта від сервісу google maps, яка дозволяє користувачу подивитись де знаходяться спортивні центри та обрати для себе найбільш підходящий.

Карту можна двигати, переглядати геопозиції філій, дивитися адреси. Також для зручнішої взаємодії її можна відкрити у новій вкладці (переадресація на сайт google maps).



Рисунок 5.12 — Отримання розрахунків біоритму в чат-боті

Користувач, який не зареєструвався на сайті може натиснути /start та ввести свою дату дня народження для відслідковування біоритму.

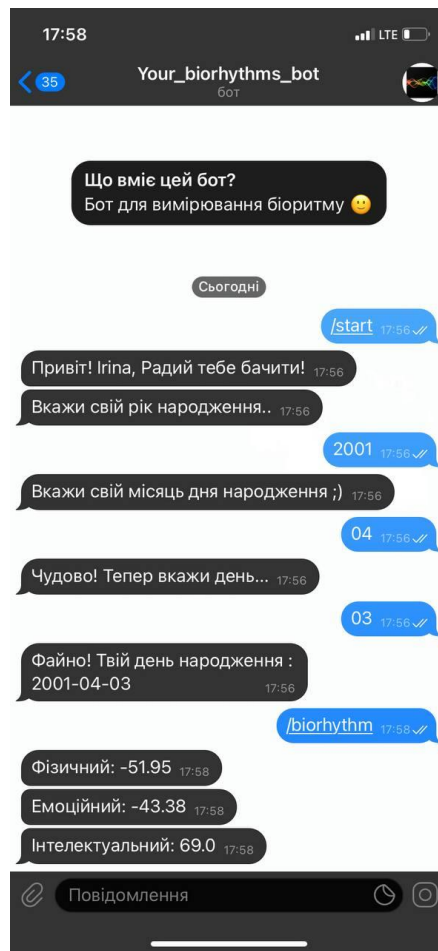
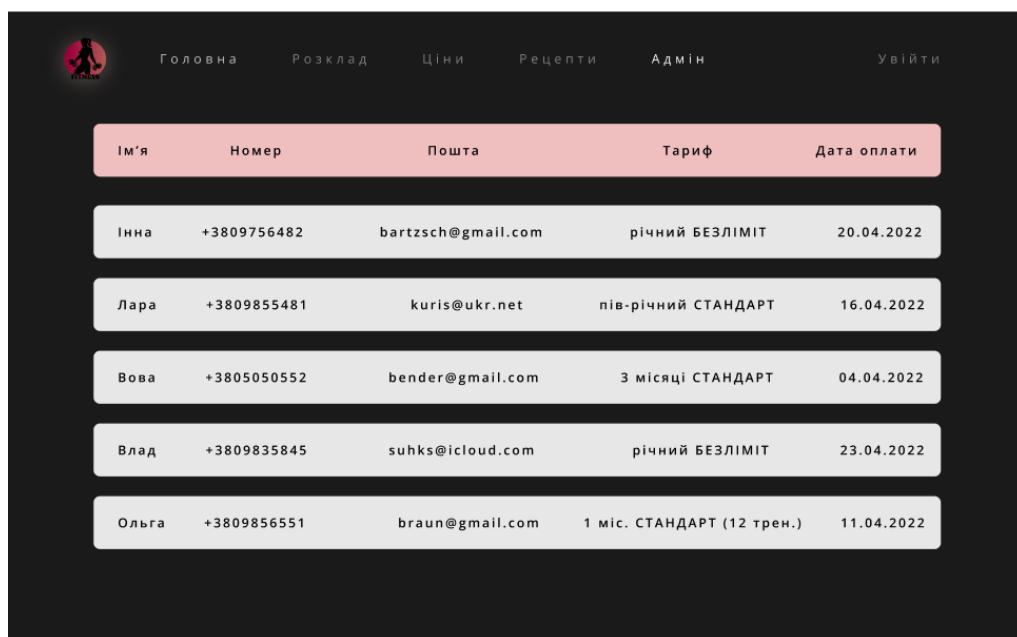


Рисунок 5.13 — Отримання розрахунків для незареєстрованого користувача

Бот досить зручний та швидко працює.

5.2 Користувач — адміністратор

Адміністратор має майже такий самий інтерфейс, як клієнт, але має розділ, де відслідковує клієнтів.



The screenshot shows a web application interface for an administrator. At the top, there is a navigation bar with a logo on the left and links for 'Головна', 'Розклад', 'Ціни', 'Рецепти', 'Адмін', and 'Увійти'. Below the navigation bar is a table with five columns: 'Ім'я', 'Номер', 'Пошта', 'Тариф', and 'Дата оплати'. The table contains five rows of client data.

Ім'я	Номер	Пошта	Тариф	Дата оплати
Інна	+3809756482	bartzsch@gmail.com	річний БЕЗЛІМІТ	20.04.2022
Лара	+3809855481	kuris@ukr.net	пів-річний СТАНДАРТ	16.04.2022
Вова	+3805050552	bender@gmail.com	3 місяці СТАНДАРТ	04.04.2022
Влад	+3809835845	suhks@icloud.com	річний БЕЗЛІМІТ	23.04.2022
Ольга	+3809856551	braun@gmail.com	1 міс. СТАНДАРТ (12 трен.)	11.04.2022

Рисунок 5.1 — Відслідковування адміністратором клієнтів

Зручно відслідковувати по датам, коли клієнт оформив абонемент.

ВИСНОВКИ

В кінцевому результаті при розробці програмного продукту можна виявити такі висновки:

1. Створений веб-портал для перегляду даних про фітнес тренувань та ведення контролю абонементів адміністратором.

2. Створений додаток має поділ на дві ролі. Роль клієнту та адміністратора.

3. Веб-портал для фітнесу може надавати клієнтові певний перелік розроблених функцій:

- реєстрація клієнту
- вибір абонементу клієнтом
- покупка абонементу
- перегляд та вибір тренувань
- вибір харчового раціону
- перегляд актуальних рецептів
- відслідковування найближчих спорт залів на мапі
- посилання на чат-бот, який відслідковує біоритм

Та такі функції для адміністратора:

- відслідковування клієнтів, які придбали абонемент.

4. Також, були створені різні діаграми, такі як діаграма Ганта, діаграма прецедентів, діаграма розгортання. Вони цілком допомогли при створенні програмного продукту. Поставлені задачі були виконані вчасно, діаграми давали картинку, що потрібно розробити і як працює програма.

5. Дана веб-програма заснована на шаблоні MVC. MVC шаблон описує легкий спосіб створення побудови даної програми, спрямованої на відділення бізнес-логіки від користувацького інтерфейсу. Результатом є легше масштабування програми, тестування, обслуговування та, звичайно, впровадження.

6. Додатково був розроблений чат-бот на мові програмування Python.

7. Веб-портал цілком має в майбутньому розширення, можливо додавати різні функції та змінювати інтерфейс.

8. Я також освоїла нові програмні технології, такі як створення чат-боту та реалізації веб-порталу. А також згадала набуті навички з програмуванням на PHP і побудові різноманітних діаграм.

Отже, програмний продукт працює без зайвих помилок та ним може користуватись будь-хто, так як сайт цілком безкоштовний.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Каллум Хопкінс PHP. Швидкий старт: навчальний посібник Лос-Анджелес, 2014. 159 с.
2. Джон Дакетт HTML і CSS. Розробка і дизайн веб-сайтів: навчальний посібник. Індіанаполіс, 2011. 478с.
3. Julie C. Meloni PHP, MySQL & JavaScript All in One, Sams Teach Yourself. 6th Edition, United States of America, 2018, 1625 с.
4. Берко А.Ю., Верес О.М., Пасічник В.В. Системи баз даних та знань, книга 2: системи управління базами даних та знань: навчальний посібник. Львів, 2021. 584с.
5. Sport Life [Електронний ресурс] — Режим доступу до ресурсу: <https://sportlife.ua/uk/> (15.05.22).
6. Cloe Ting [Електронний ресурс] — Режим доступу до ресурсу: <https://chloeting.com/> (17.05.22).
7. Fitness Blender [Електронний ресурс] — Режим доступу до ресурсу: <https://www.fitnessblender.com/> (17.05.22)
8. Ferroom [Електронний ресурс] — Режим доступу до ресурсу: <https://ferroom.com.ua/> (18.05.22)
9. Посібник з PHP — Режим доступу до ресурсу: <https://www.php.net/manual/ru/book.mysql.php> (18.05.22)
10. Сударшан С, Сибельшатц А, Концепції системи БД: навч. посібник Нью-Йорк: McGrawHill, 2011, 1244 с.
11. Путівник мовою програмування Python — Режим доступу до ресурсу: <https://pythonguide.rozh2sch.org.ua/>

ДОДАТОК А

Веб-портал для фітнес тренувань (Організація доступу для клієнта)

Текст програми

УКР.НТУУ “КПІ ім. Ігоря Сікорського” ТЕФ_АПЕПС_ТР81____22Б12-1

Аркушів 7

Київ – 2022

Приклад програмного коду PHP вибору абонементу "main_register.php"

```
<?php
namespace Controllers;

use Models\Order;
use Models\User;
class MainController
{
    public function index($view = "")
    {
        if ($view) {

            if (($view == 'login' || $view == 'reg') && !empty($_SESSION['user'])) {
                header('Location: /');
            } else {
                include "Views/{$view}.php";
            }
        }
    }

    public function login()
    {
        $user = new User();
        $loginUserArr = $user->login($_POST);

        if ($loginUserArr['isAdmin'] == true) {
```

```

        $_SESSION['user']['id'] = 0;
        $_SESSION['user']['name'] = 'admin';
        header('Location: /admin');
    }

    $loginUser = $loginUserArr['user'];

    if (empty($loginUser)) {
        header('Refresh:2; url=/login');
        echo 'Не знайдено користувача, спробуйте ще. Вас буде перенаправлено
через 2с';
    } else {
        $_SESSION['user']['id'] = $loginUser['id'];
        $_SESSION['user']['name'] = $loginUser['name'];

        header('Location: /');
    }
}

public function register()
{
    $user = new User();
    $statusRegister = $user->register($_POST);

    if ($statusRegister) {
        header('Location: /login');
    } else {

```

```

        header('Refresh:2; url=/reg');
        echo 'Помилка при реєстрації, спробуйте ще. Вас буде перенаправлено
через 2с';
    }
}

public function logout()
{
    $user = new User();
    $user->logout();

    header('Location: /');
}

public function getOrders($view)
{
    /**
     * Якщо юзер не адмін - редірект на головну сторінку
     */
    if (isset($_SESSION['user']) && $_SESSION['user']['name'] == 'admin') {
        $orderObj = new Order();
        $orders = $orderObj->getOrders();
        include "Views/{$view}.php";
    } else {
        header('Refresh:1; url=/');
        echo 'Ви не маєте доступ до адмінки. Вас буде перенаправлено через 1с';
    }
}

```

```

    }

    public function order()
    {
        if (isset($_SESSION['user']) && $_SESSION['user']['name'] != 'admin') {
            $user = $_SESSION['user'];
            $types = Order::TYPE_ORDER;
            include "Views/order.php";
        } elseif (isset($_SESSION['user']) && $_SESSION['user']['name'] == 'admin') {
            header('Refresh:2; url=/');
            echo 'Адмін не має права залишати заявки. Авторизуйтеся як юзер. Вас
буде перенаправлено через 2с';
        } else {
            header('Refresh:1; url=/');
            echo 'Авторизуйтеся для права залишати заявки. Вас буде
перенаправлено через 1с';
        }
    }

    public function create_order()
    {
        $orderObj = new Order();
        $status = $orderObj->create($_POST);

        if ($status) {
            header('Refresh:2; url=/');
            echo 'Вітаю, Ви успішно створили заявку на абонемент. Чекайте дзвінка.
Вас буде перенаправлено через 2с на головну сторінку.';
        }
    }
}

```

```

    } else {
        header('Refresh:1; url=/order');
        echo 'Помилка при створенні заявки. Спробуйте ще раз. Вас буде
перенаправлено через 1с';
    }
}
}
}

```

Приклад програмного коду PHP реєстрації (user.php)

```

<?php

namespace Models;

class User extends Model
{
    public const table = 'users';

    public function login($data)
    {
        $login = $data['login'];
        $password = $data['password'];
        $user = [];

        /**
         * Адмін відповідає login = admin && pass = admin
         */
        if ($login == 'admin' && $password == 'admin') {

```

```

        $isAdmin = true;
    } else {
        $isAdmin = false;

        $sql = "SELECT * FROM ".self::table." WHERE login = :login AND
password = :password";
        $stmt = $this->pdo->prepare($sql);
        $stmt->bindValue(":login", $login);
        $stmt->bindValue(":password", md5($password));
        $stmt->execute();

        $user = $stmt->fetch(\PDO::FETCH_ASSOC);
    }

    return ['isAdmin' => $isAdmin, 'user' => $user];
}

public function register($data)
{
    $sql = "INSERT INTO ".self::table."(login, name, phone, email, password)
        VALUES (:login, :name, :phone, :email, :password)";

    $stmt = $this->pdo->prepare($sql);
    $stmt->bindValue(":login", $data['login']);
    $stmt->bindValue(":name", $data['name']);
    $stmt->bindValue(":phone", $data['phone']);
    $stmt->bindValue(":email", $data['email']);

```

```
$stmt->bindValue(':password', md5($data['password']));

return $stmt->execute();
}

public function logout()
{
    unset($_SESSION['user']);
}
}
```