

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

**“Веб-орієнтована CRM-система для спортклубу
Титан”**

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Проценко О.Б.

Студент групи ІН–73–9

Філатова М.В.

СУМИ 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2021 р.

Завдання
до випускної роботи

Студента четвертого курсу, групи ІН-73-9 спеціальності “Комп'ютерні науки”
денної форми навчання Філатової Марини Володимирівни.

Тема: “Веб-орієнтована CRM-система для спортклубу Титан”

Затверджена наказом по СумДУ

№ _____ від _____ 2021 р.

Зміст пояснювальної записки: 1) аналітичний огляд методів побудови CRM-системи; 2) постановка завдання й формування завдань дослідження; 3) огляд і опис засобів для розробки; 4) розробка веб-орієнтованої CRM-системи для спортклубу «Титан»; 5) аналіз результатів.

Дата видачі завдання “ _____ ” _____ 2021 р.

Керівник випускної роботи _____ Проценко О.Б.

Завдання прийняв до виконання _____ Філатова М.В.

РЕФЕРАТ

Записка: 78 стор., 45 рис., 5 додатків, 15 джерел.

Об'єкт дослідження — Веб-орієнтована CRM-система для спортклубу “Титан”

Мета роботи — розробити веб-орієнтовану CRM-систему для спортклубу, яка повинна бути швидкою при великій кількості інформації, проста в дизайні і зрозуміла для користувача. Сайт повинен бути адаптивним до всіх пристроїв, також повинен працювати у всіх сучасних веббраузерах.

Результати — розроблено PHP об'єктно-орієнтований фреймворк, побудований по архітектурі шаблону MVC; на основі власного фреймворку реалізовано клієнтську частину CRM-системи для спортклубу «Титан».

CRM-СИСТЕМА, ФРЕЙМВОРК, MVC, HTML, CSS, JAVASCRIPT,
PHP, AJAX, JSON

ЗМІСТ

ВСТУП	5
1 ІНФОРМАЦІЙНИЙ ОГЛЯД	7
1.1 Коротка інформація про CRM-системи	7
1.2 Огляд вебресурсів спортивних клубів	10
1.3 Постановка задачі	14
2 ВИБІР МЕТОДІВ ВИРІШЕННЯ ЗАДАЧІ	16
2.1 Вибір методів розроблення	16
2.2 Вибір засобів програмування	18
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	22
3.1 Проектування структури вебсистеми	22
3.2 Реалізації основних компонентів	25
ВИСНОВКИ	49
СПИСОК ЛІТЕРАТУРИ	50
ДОДАТКИ	52

ВСТУП

У сучасному бізнесі попит на автоматизацію різних процесів стрімко зростає. На сьогодні досить складно уявити собі клієнтський, складський або бухгалтерський облік без застосування спеціалізованого програмного забезпечення. Оскільки спортклуби користуються величезною популярністю, тому бажаючих записатися на тренування дуже багато. Більшість представників сфери послуг або торгових представників вже використовують спеціальні програми, що допомагають вести клієнтів і оперувати необхідними даними. Втім, взаємодія з клієнтами в середньому і малому бізнесах все ще досить часто проводиться без впровадження автоматизації та значної уваги до обліку.

У разі, якщо робота ведеться без системи обліку, то кожен співробітник взаємодіє з клієнтами на власний розсуд: на папері, в Excel таблицях або ж взагалі не фіксує процес своєї роботи. Повідомлення клієнтам можуть надсилатись як з корпоративної, так і з особистої поштової скриньки, тому контроль якості роботи стає неможливим.

Вихід з цієї ситуації — автоматизація та стандартизація управління відносин з клієнтами, тобто впровадження CRM-системи (Customer Relationship Management System). Така система автоматично створює окремі картки для кожного клієнта на відміну від звичних Excel таблиць, де все зібрано в одну купу і легко заплутатися. Програмне забезпечення спортклубів позбавляє співробітників від непотрібної паперової роботи, яка забирає багато часу. Це не тільки зручно, але й мінімізує ймовірність спрацьовування людського фактора.

Автоматизація спортклубу допомагає створити організований інформаційний простір, впорядковуючи його діяльність, наприклад:

- спрощує спілкування персоналу закладу з відвідувачами (онлайн запис і скасування тренувань майже з будь-якого пристрою в будь-який час

через вебдодаток або чат-бот);

- здійснення оплати за тренування онлайн;
- дозволяє отримати стандартизовану базу клієнтів;
- вести простий, зручний облік клієнтів;
- ефективно здійснювати контроль якості роботи;
- мати особистий кабінет, в якому доступ до розкладу та графіку

тренувань 24/7;

– відстеження клієнтами станів абонементів, своїх фізичних показників, завантаженості клубу по годинах та ін. в особистих кабінетах;

- отримувати статистику ефективності власної роботи;
- вести фінансовий облік клубу;
- дозволяє легко контролювати роботу закладу;
- підвищує ефективність бізнесу і забезпечує організовану роботу всіх

співробітників [1].

Отже, метою даної роботи є розробка CRM-системи з унікальним функціоналом, інтерфейсом і можливостями, підібраними відповідно до цілей і завдань спортклубу.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Коротка інформація про CRM-системи

CRM-системи — це ПЗ (програмне забезпечення) для управління взаємовідносинами з клієнтами.

ПЗ CRM є особливою системою з однією основною метою — поліпшення відносин з клієнтами шляхом надання бізнесу інтелектуальних інструментів для ефективною розробки успішної стратегії. Іншими словами, ПЗ CRM може пришвидшити важливі бізнес-операції, такі як: маркетинг, обслуговування та продажі — щоб підвищити якість обслуговування і лояльність клієнтів від першої взаємодії до наступних.

Інструмент CRM починає зі збору даних з вебсайту, електронної пошти, телефону і соціальних мереж клієнта, а також інших даних з багатьох джерел і каналів. Він також може автоматично отримувати іншу інформацію, таку, як останні новини про діяльність компанії, і може зберігати особисті дані, такі, як особисті переваги клієнта в спілкуванні.

CRM-система по суті являє собою центральне місце, де підприємства можуть зберігати дані про клієнтів і потенційних клієнтів, відслідковувати взаємодії з клієнтами й ділитися цією інформацією з колегами. Це дозволяє компаніям управляти відносинами з клієнтами, допомагаючи бізнесу зростати.

Основними видами CRM-систем виділяють наступні:

1 операційні — найбільш поширений тип CRM, що дозволяє спростити процес взаємодії з клієнтом, підвищити його лояльність, структурувати і зберігати клієнтські дані та ін. Дає можливість отримати оперативний доступ до персональних даних клієнта в ході продажу або обслуговування;

2 аналітичні — відрізняються від операційних тим, що вмюють не тільки збирати і зберігати дані, але і дозволяють аналізувати їх, складати діаграми, графіки та ін. Тому основною метою цієї системи є формування

аналітики для розробки і планування сценаріїв взаємодії з клієнтами;

3 стратегічні — поліпшення комунікації та командної роботи всередині компанії. Спільна CRM допомагає об'єднати кілька відділів компанії (продажі, маркетинг, технічна підтримка) для досягнення спільної мети — використання накопичених даних для підвищення якості обслуговування, лояльності і залучення нових клієнтів;

4 комбіновані — зазвичай це операційні CRM-програми з обмеженим набором аналітичних інструментів (звіти по клієнтах, аналітика продажів і ефективності виконавців), а також з можливістю прямого спілкування з клієнтами та колегами за допомогою інтеграції з месенджерами, соціальними мережами та сайтом [2].

Вибір функцій CRM-системи залежить від тих завдань, які перед нею ставлять. Як правило, на це найбільше впливає розмір бізнесу:

- приватний підприємець. Зазвичай одиночним користувачам потрібна тільки база даних клієнтів для управління і відстеження зв'язків з ними. Це суттєво економить час і кошти;

- малий бізнес. Найбільш великим підприємцям в більшості випадків потрібна автоматизація відділу продажів аж до контролю й відстеження дій окремих торгових представників;

- середній бізнес. Такі компанії акцентують увагу на автоматизації продажів і маркетингу. Окрім цього, їм важлива аналітика, що дозволяє прогнозувати бізнес-процеси і створювати маркетингові стратегії;

- великий бізнес. Для великих компаній важливо все — від продажів до стратегії, тому вони потребують повноцінних комбінованих CRM-систем.

Можна виділити в окремий список основні переваги CRM-систем:

- створення повної бази клієнтських даних;
- зберігання історії взаємодії клієнта з компанією з доступом до неї;
- підвищення ефективності роботи відділу продажів;
- аналіз попиту і бажань споживача;

- автоматизація процесів менеджера (економія часу);
- автоматичне нагадування про необхідність зв'язатися з клієнтом.

Результат успішного впровадження CRM у бізнес — зростання продажів і прибутку компанії.

До трьох ключових типів CRM [3] можна віднести:

- настільні системи, що працюють на одному комп'ютері;
- система клієнт-сервер, що має центральну базу даних, яка зберігається на сервері, зазвичай розміщується на власному хостингу з програмним забезпеченням, встановленим на ПК (персональному комп'ютері) кожного користувача для доступу до неї;

- хмарні системи, які надаються та розміщуються в мережі третьої сторони і доступні в будь-якому місці через підключений пристрій.

Вибір платформи залежить від обладнання компанії, для якого створюється така система. У більшості випадків це ПК і ноутбуки, що працюють на базі операційних систем Windows і macOS, рідше – Linux або DOS. В останні кілька років також спостерігається тенденція мобілізації CRM-додатків. Основними перевагами CRM-додатків з підтримкою мобільних платформ можна виділити:

- більш точні дані. Можливість вносити інформацію в систему з будь-якого місця в будь-який час, що зменшує час затримки між подією і введенням інформації про цю подію (актуальність даних);

- вище ефективність. Мобільні пристрої дозволяють тримати потрібну інформацію перед очима. Це підвищує ефективність і сприяє більш економічному використанню часу;

- більше прийняття. Люди звикли до телефону в руці і постійного доступу до потрібної інформації. Використовуючи цю звичку, можна підвищити швидкість прийняття нової CRM-системи;

- краще обслуговування. Підтримка мобільних платформ сприяє скороченню відгуку на запит клієнта і при цьому дозволяє співробітникам

сприймати інформацію «на льоту».

1.2 Огляд вебресурсів спортивних клубів

Оскільки досить часто відбувається пряме спілкування з клієнтами за допомогою інтеграції CRM-системи з сайтом, тому надалі будуть розглянуті вебсайти спортклубів та CRM для сфери послуг.

Одним із завдань дипломного проєкту є розробка веб-орієнтованої системи для спортклубу. Для досягнення мети було проведено дослідження наступних сайтів подібної тематики:

1. <https://ironhouse.com.ua/> — сайт спортклубу IRON HOUSE, що містить змістовну та динамічно оновлюючу інформацію про послуги, які надає. Дозволяє користувачам зареєструватись у системі за умови, якщо гість сайту — клієнт даного клубу (має клубну карту, за номером якої відбувається реєстрація) (див.рис. 1.1).

З плюсів можна виділити:

- динамічне ведення сторінок сайту;
- наявність посилань на соціальні мережі;
- зручний та зрозумілий дизайн;
- можливість написання відгуків;
- наявність особистого кабінету для зареєстрованого клієнта клубу.

Недоліки вебсайту:

- не адаптивний дизайн деяких розділів сайту;
- відсутня валідація форм;
- час переходу між сторінками перевищує середній;
- відсутність форми зворотного зв'язку.

Рисунок 1.1 — Сторінка реєстрації користувачів

1. <https://b-tone.sumy.ua/> — сайт спортклубу B-tone, що вирізняється унікальним дизайном з-поміж конкурентів, розроблений агентством інтернет-маркетингу Володимира Стрілецького.

З переваг можна виділити:

- наявність форми зворотного зв'язку;
- ефектний дизайн із застосуванням сучасних технологій та підходів веброзробки (паралакс-ефект) (див. рис. 1.2);

- динамічне ведення сторінок сайту;

- наявність посилань на соціальні мережі;

До недоліків можна віднести:

- відсутність можливості написання відгуків спортклубу;

- немає можливості зареєструватися на сайті;

- відсутність розділу з прайс-листом.

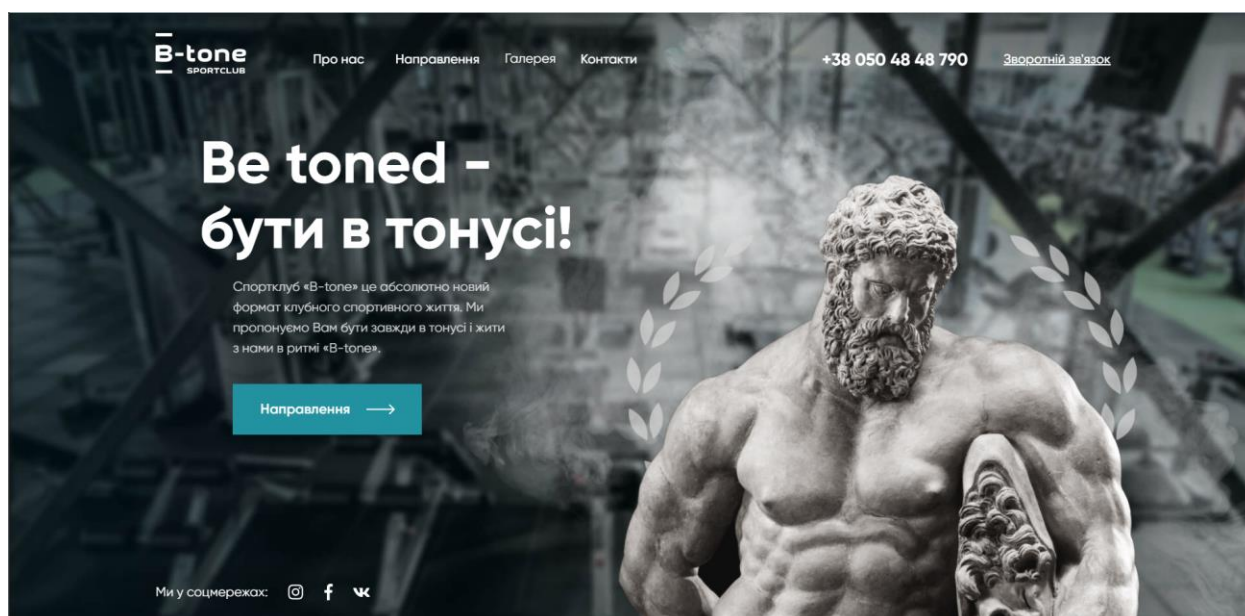


Рисунок 1.2 — Інтерфейс головної сторінки сайту

Більшість програм універсальні: допомагають збирати дані про клієнтів і контролювати продажі. При цьому такі CRM малоефективні коли потрібна автоматизація процесів в конкретній галузі. Тому на ринку існують галузеві CRM, розроблені спеціально під потреби бізнесу в конкретній ніші. Оскільки більшість з них платні, для огляду та аналізу були взяті наступні CRM зі сфери послуг:

1. «SalesapCRM» [4] – багатофункціональна система, що містить модулі управління продажами, завданнями (див. рис. 1.3) та фінансами, а також комунікації з клієнтами.

З плюсів можна виділити:

- простий та інтуїтивно зрозумілий інтерфейс;
- налаштування системи під користувача (програма розподіляє вхідні звернення між менеджерами, надсилає повідомлення про нове завдання);
- історія змін, завдяки якій легко знаходити відповідальних за дії в програмі по будь-якому клієнту або етапу.

До недоліків можна віднести:

- важливі для автоматизації бізнес-модулі відкриваються лише на платних тарифах: звіти, каси та бухгалтерія, інтеграція з месенджерами.

– відсутній зв'язок між поштою та сховищем, файли не можна прикріпити до листа.

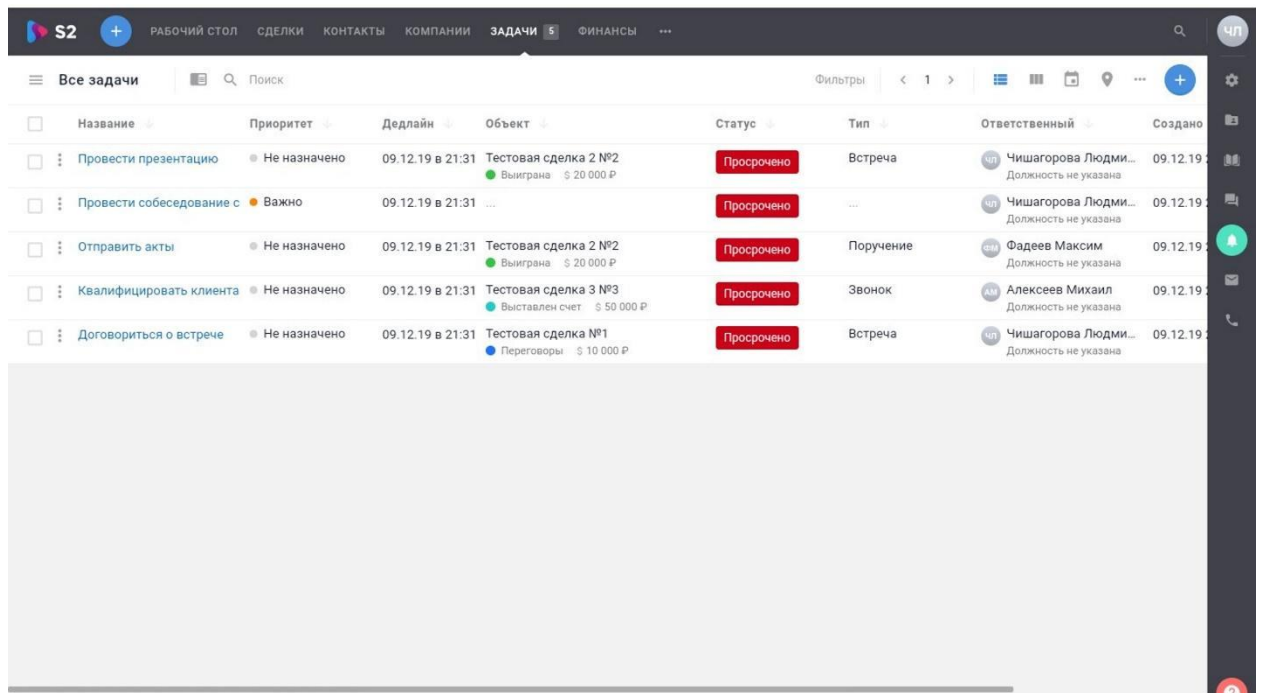


Рисунок 1.3 — Сторінка управління завданнями

2. «Арніка» [5] — сервіс для управління салоном краси. Онлайн-розклад наочно показує завантаженість майстрів (див. рис. 1.4).

З переваг можна виділити:

- автоматична розсилка SMS повідомлень клієнтам після внесення адміністратором інформації про візит;
- облік робочого часу майстрів, облік витратних матеріалів, збір даних для бухгалтерії, аналіз та рекомендації щодо закупівлі в майбутньому;
- працює служба підтримки, щоб адміністратори салонів могли швидко освоїти основи роботи в системі.

До недоліків можна віднести:

- у безкоштовній версії недоступні ключові функції CRM: онлайн-запис, управління товарами, продажами і фінансами, розрахунок зарплати майстрів, настройки прав доступу і бізнес-аналітика;
- висока вартість SMS розсилки.

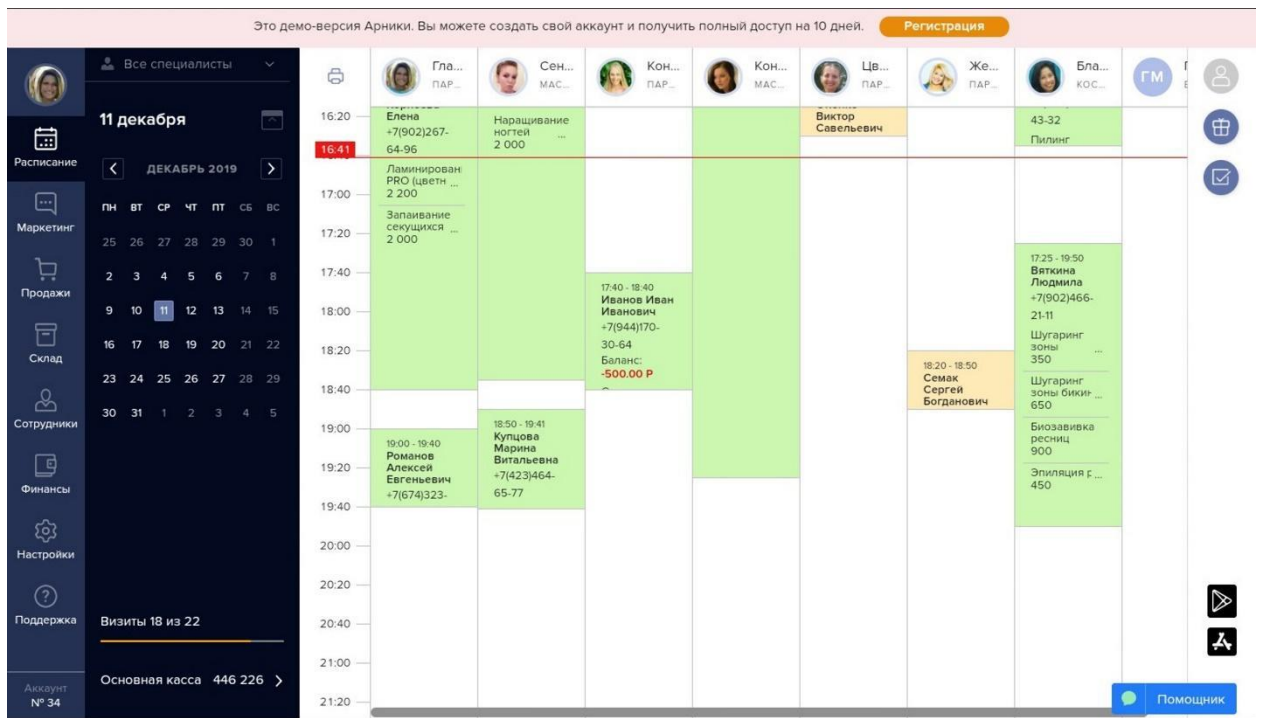


Рисунок 1.4 — Онлайн-роzkлад CRM «Арніка»

Все ж, незважаючи на численні CRM, індивідуальне рішення завжди виграє, і ось чому:

- індивідуальні інструменти дозволяють вирішувати складні завдання;
- відсутність великої кількості опцій і функцій, з яких впливає велика кількість полів та іконок на призначеному для користувача інтерфейсі,
- високий рівень безпеки;
- зручний та унікальний інтерфейс для співробітників;
- окремі маркетингові інструменти для просування бізнесу.

1.3 Постановка задачі

У даній роботі необхідно розробити вебсистему для спортклубу, яка повинна мати наступний функціонал:

- реєстрацію та авторизацію;
- 3 типи користувачів (гість, авторизований користувач (клієнт клубу) та адміністратор);
- форму зворотного зв'язку/запису на заняття;
- особистий кабінет для клієнтів, з можливістю: редагування власної

інформації; відстеження історії відвідувань спортклубу та історії оплати занять, станів абонементів; здійснення оплати тренувань онлайн; залишати відгуки.

– можливість створювати, редагувати та видаляти оголошення (з автоматичним сповіщенням про них на пошту або в особистих кабінетах користувачів або чат-боті Telegram);

– можливість ведення обліку оплат занять і відвідувань кожного клієнта;

– можливість редагування всього контенту сайту з відповідним оповіщенням клієнтів у разі зміни графіку роботи клубу;

– автоматичну розсилку індивідуальних нагадувань на пошту/в особистому кабінеті про закінчення терміну дії абонементу;

– фінансові звіти та звіти з відвідування клубу клієнтами;

Результатом роботи має стати вебсистема, яка повинна бути швидкою, простою в дизайні і зрозумілою для користувача. Сайт повинен бути адаптивним до всіх пристроїв, також повинен працювати у всіх сучасних веббраузерах.

2 ВИБІР МЕТОДІВ ВИРІШЕННЯ ЗАДАЧІ

2.1 Вибір методів розроблення

Бажання заощадити виражається в тому, що підприємці заради уявної економії віддають перевагу готовим продуктам замість того, щоб створити унікальне програмне рішення, яке буде враховувати всі особливості конкретного бізнесу і його цільової аудиторії.

Відповідно до основних переваг індивідуальних рішень перед готовими (див. 1.2.2) можна зробити висновок, що перед розробниками готових CRM стоїть складне завдання — вони повинні створити продукт, який сподобається всім. Зробити це неможливо, тому розробники відмовляються від деяких цільових аудиторій на користь більшості. Усі шаблонні рішення схожі, що дуже погано з точки зору маркетингу.

Щоб уникнути цих недоліків, краще розробляти системи CRM з нуля.

У число основних способів розробки CRM-систем, входять:

- CRM з відкритим вихідним кодом з можливістю кастомізації відповідно до власних потреб [6];
- php-фреймворки з можливістю гнучкого вибору інструментарію при розробці сайту.

Фреймворк PHP — це платформа для створення вебдодатків PHP. Він містить бібліотеки з попередньо упакованими функціями та класами і, найчастіше, елементи для реалізації шаблонів розробки програмного забезпечення.

Переваги використання фреймворків:

- розробка сайту будь-якої складності і функціоналу;
- висока продуктивність;
- фреймворки можна використовувати паралельно з сайтом, розробленому на CMS (системи управління контентом);
- наявність якісної і повної документації з прикладами;

- можливість тестування.

Серед недоліків:

- не підходить для великих додатків;
- поганий метод обробки помилок;
- зовнішня залежність.

Плюси CRM з відкритим вихідним кодом:

- нескінченна модульна настроюваність;
- «розширювана архітектура» і модульна структура, яка дозволяє поєднувати різні функції;
 - у деяких CRM в основі лежить php-фреймворк, тому достатньо легко внести потрібні зміни, що дозволяє відносно легко і економічно модифікувати платформу відповідно до потреб;
 - велика спільнота користувачів та розробників.

Недоліки CRM з відкритим вихідним кодом:

- інтеграції та функції безкоштовної версії з відкритим вихідним кодом не так розвинені, як у платних версіях;
 - слабка функція пошуку і неможливість прикріпляти файли з електронної пошти до контакту;
 - функції звітності дещо гірше в порівнянні з платною версією.

Ще одним методом для розробки вебсистеми є її написання з нуля за допомогою сучасних методів та вебтехнологій (див. 2.2). Основою для якої може слугувати власний простий PHP об'єктно-орієнтований фреймворк, який буде використовувати шаблон проектування MVC (Model-View-Controller) [7].

Кожен спосіб має плюси і мінуси. Останні два — найоптимальніші, тому що можна отримати не тільки працюючу систему в підсумку, але і чітке розуміння того, як це було зроблено. Втім, для спрощення роботи ніхто не забороняє використовувати альтернативні способи.

2.2 Вибір засобів програмування

На сьогодні існує велика кількість мов програмування, програмних засобів, СУБД, вебсерверів, IDE (інтегроване середовище розробки) та фреймворків за допомогою яких можна створити вебсистему. Відповідно до поставлених вище завдань (див. 1.3), найбільш оптимальним засобом для розробки такого продукту буде застосування мови розмітки гіпертекстових документів — HTML5, інструменту для керування зовнішнім виглядом вебсайту — CSS3, динамічної мови програмування JavaScript, скриптової мови програмування загального призначення PHP з використанням вебсервера Apache та реляційних баз даних MySQL.

Ще одним програмним засобом, без якого майже неможлива швидка розробка на PHP, є пакетний менеджер Composer [8]. Це потужний і зручний інструмент для управління бібліотеками. Пакетний менеджер вміє:

- знаходити бібліотеки за назвою, завантажувати і розпаковувати їх;
- автоматично завантажувати і встановлювати залежності для кожної бібліотеки;
- підключати класи бібліотек всередині сценаріїв;
- оновлювати бібліотеки разом із залежностями;
- викачувати пакети за замовчуванням з офіційного репозиторія packagist.org. Будь-яка людина може вільно додати туди свій пакет, щоб зробити його установку максимально легкою і зручною для всього світу;
- завантажувати пакети не тільки з packagist.org, але і з будь-якого GIT, Mercurial або SVN сховища;
- при скачуванні пакетів з github.com або bitbucket.org не потрібна встановлена система контролю версій (GIT або ін.), Composer працює через API (програмний інтерфейс) цих сайтів.

HTML5 (Hypertext Markup Language) — це остання версія мови гіпертекстової розмітки, коду, що описує вебсторінки. Він використовується для структурування та подання контенту на вебсторінках. HTML5 підтримує

традиційний синтаксис HTML і XHTML (розширювана мова гіпертекстової розмітки), а також інші нові функції в своїй розмітці, нові API, XHTML і обробку помилок. На відміну від старих версій HTML, які дозволяли створювати в основному статичні сайти, в які потрібно було додавати CSS і JavaScript, HTML5 набагато динамічніший і включає мультимедійні елементи. Тобто, HTML5 повністю здатний робити те, що раніше потрібно було робити, використовуючи старі інструменти.

CSS3 (Cascading Style Sheets) означає каскадну таблицю стилів 3-го рівня, яка є розширеною версією CSS і використовується для структурування, стилізації та форматування вебсторінок. В CSS3 було додано кілька нових функцій, і він підтримується всіма сучасними веббраузерами. Ця технологія надає функції JavaScript та мобільної розробки з додатковими функціями, такими як: переходи, градієнти, анімація, блоки, списки і багато іншого.

JavaScript (JS) — це легка об'єктно-орієнтована мова програмування, що забезпечує динамічну інтерактивність на вебсайтах при застосуванні до HTML-документа. Можливості JavaScript багато в чому залежать від середовища, в якому працює. JavaScript в основному використовується для додавання інтерактивності вебсайтам, розробки мобільних додатків, створення ігор на основі веббраузера та серверної веброботи.

PHP (Hypertext Preprocessor) — це поширена мова програмування загального призначення. Мова спеціально сконструйована для веброботи, код може впроваджуватися безпосередньо в HTML [9]. PHP простий у вивченні і ефективно працює на стороні сервера, працює на різних платформах, сумісний практично з усіма використовуваними сьогодні серверами та підтримує широкий спектр баз даних.

MySQL — це реляційна система управління базами даних. СУБД MySQL відзначається простотою використання, гнучкістю, низькою вартістю володіння (відносно платних СУБД), а також масштабованістю і продуктивністю. СУБД MySQL включає SQL-сервер і програми-клієнти, що

здійснюють доступ до серверу, засоби адміністрування і програмний інтерфейс для програмування своїх особистих програм. На сьогодні ця СУБД — одна з найбільш популярних, що використовують у вебдодатках.

Apache HTTP Server — вебсервер, який дуже часто використовується в якості сервера додатків для PHP-проектів. Взагалі, Apache працює на базі скриптових або веб-орієнтованих мов (PHP, Python, Tcl, Ruby, Perl, ASP), що робить його сумісним з найширшим спектром баз даних і серверного ПЗ. У Apache динамічна модульна структура. Можна швидко підключати додатковий функціонал (скачувати модулі), навіть без звернення до зовнішніх джерел. Вебсервер працює як на Windows, так і на всіх Unix-подібних системах. [10] З переваг можна відзначити універсальність і простоту налаштування.

JSON (JavaScript Object Notation) [11] — це загальний формат для представлення структурованої інформації (значень і об'єктів) і в основному використовується для передачі даних між сервером і клієнтом. Файл JSON являє собою більш просту і легку альтернативу розширенню з аналогічними функціями XML (Extensive Markup Language). Розробники використовують JSON для роботи з AJAX (асинхронний JavaScript і XML). Ці формати добре працюють разом для досягнення асинхронного завантаження даних, що зберігаються, а це означає, що вебсайт може оновлювати свою інформацію без оновлення сторінки.

AJAX (аббревіатура від «Asynchronous Javascript And Xml») — технологія звернення до сервера без перезавантаження сторінки [12]. Це група взаємопов'язаних технологій: JavaScript, DOM (Document Object Model), XML, HTML / XHTML, CSS, XMLHttpRequest та ін. AJAX дозволяє відправляти і отримувати дані асинхронно без перезавантаження вебсторінки.

Переваги використання AJAX:

- можливість створення зручного вебінтерфейсу;
- активна взаємодія з користувачем;
- часткове перезавантаження сторінки замість повної;
- зручність використання.

Visual Studio Code (VS Code) [13] відомий як один з найбільш повнофункціональних, безкоштовних і добре продуманих редакторів коду, що належить до сімейства ПО, розробленого технологічним гігантом Microsoft. Це розширюваний редактор коду з відкритим вихідним кодом, який підтримує широкий вибір мов програмування і фреймворків.

Особливості VS Code:

- рефакторинг і відладка коду;
- легко налаштовується під потреби;
- проста робота з системами контролю версій;
- автодоповнення різного типу конструкцій;
- велика кількість плагінів та тем;
- контекстна підказка;
- автозбереження;
- підтримка багатомовного середовища;
- крос-платформна підтримка.

Bootstrap [14] — це гігантська колекція зручних, багаторазових фрагментів коду, написаних на HTML, CSS і JavaScript. Це також середовище розробки зовнішнього інтерфейсу, яке дозволяє швидко створювати повністю адаптивні вебсайти. Тобто даний інструмент позбавляє від написання великої кількості коду CSS, даючи більше часу на розробку вебсторінок.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Проєктування структури вебсистеми

Вебдодаток реалізований за допомогою PHP об'єктно-орієнтованого фреймворку, побудованого за архітектурою шаблону MVC. Шаблон MVC описує простий спосіб побудови структури додатка, метою якого є відділення бізнес-логіки від призначеного для користувача інтерфейсу. В підсумку додаток легше масштабується, тестується, супроводжується і реалізується.

Структура сайту — це логічна побудова всіх сторінок ресурсу. Схема, за якою розподіляється шлях до папок, категорій. Неправильна побудова структури сайту суттєво ускладнює просування користувачів по сайту.

Структура проєкту має наступний вигляд:

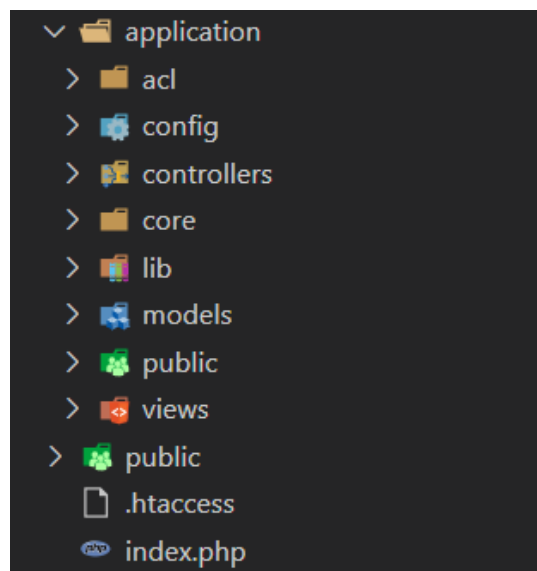


Рисунок 3.1 — Структура розробленого додатка

Папка `acl` (access control list) містить масиви з правами для кожного контролера. Директорія `controllers` зберігає всі створені контролери. Папка `models` містить моделі даних. Папка `views` — для зберігання уявлень проєкту, папки `layouts` з шаблонами сторінок та папки `errors` з шаблонами для сторінок-помилки. У папці `config` зберігаються всі налаштування для поточного проєкту (файл підключення до бази даних та файл маршрутів). В папці `core` знаходяться основні класи (батьківські класи `Controller`, `Model` та `View`, котрі

будуть наслідувати інші контролери та моделі) та клас Router, що відповідає за навігацію на сайті. В папці lib зберігаються різні класи, наприклад, клас для роботи з базою даних, пагінацією та сторонні файли Composer. Папка public містить директорії css, js, images з файлами відповідних типів.

Відповідно до вищенаведеної структури було розроблено шаблони сторінок вебсистеми. Шаблони досить зручні, тому що більшість сторінок верстають за принципом однієї сторінки майже автоматично. Структура шаблону складається з елементів, які повинні бути присутніми на всіх сторінках системи. В моєму проєкті використовується декілька шаблонів, для вебсторінок системи.

Перший шаблон для головної сторінки вебсайту спортклубу виконаний в форматі Landing page і містить всю інформацію про спортклуб (рис. 3.2).

Для переходу до самої системи, та її основних функцій на головній сторінці є кнопка, при натисканні на котру, гість попадає на сторінку авторизації в системі. Також на наступній сторінці гість може не тільки виконати процедуру авторизації, але ще у нього є можливість виконати відновлення пароля та процедуру реєстрації особистого кабінету.

Другий шаблон є основним з точки зору використання шаблону по всій системі в цілому (див. рис. 3.3). Шаблон містить меню навігації, яке розташовується з лівого боку сторінки до того моменту, поки розширення екрану зі значенням ширини не менше 992 px. Зі зменшенням даного значення, меню розміщується вгорі сторінки горизонтально у вигляді випадаючого списку. При натисканні на будь-який пункт меню, сторінка автоматично переходить в той розділ, за яким закріплено цей пункт.

Для різних категорій користувачів пункти навігації (меню) будуть різними.



Рисунок 3.2 — Шаблон головної сторінки

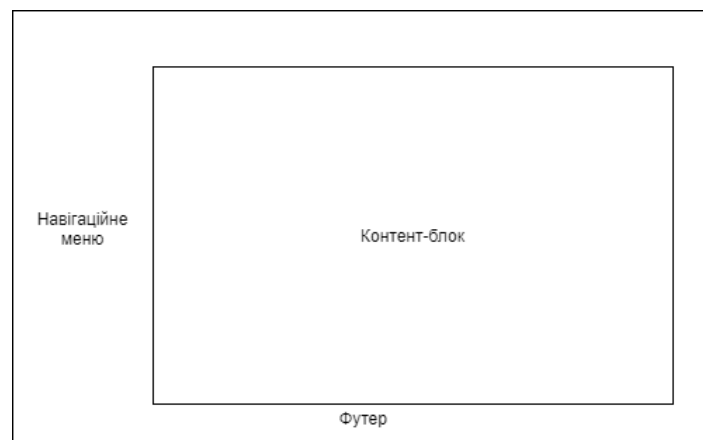


Рисунок 3.3 — Основний шаблон сторінок CRM-системи

3.2 Реалізації основних компонентів

На схемі взаємодії користувачів (use-case) вебсистеми, яка описує права користувачів на сайті (рис. 3.4), є три актори:

- гість;
- користувач, який авторизувався;
- адміністратор.

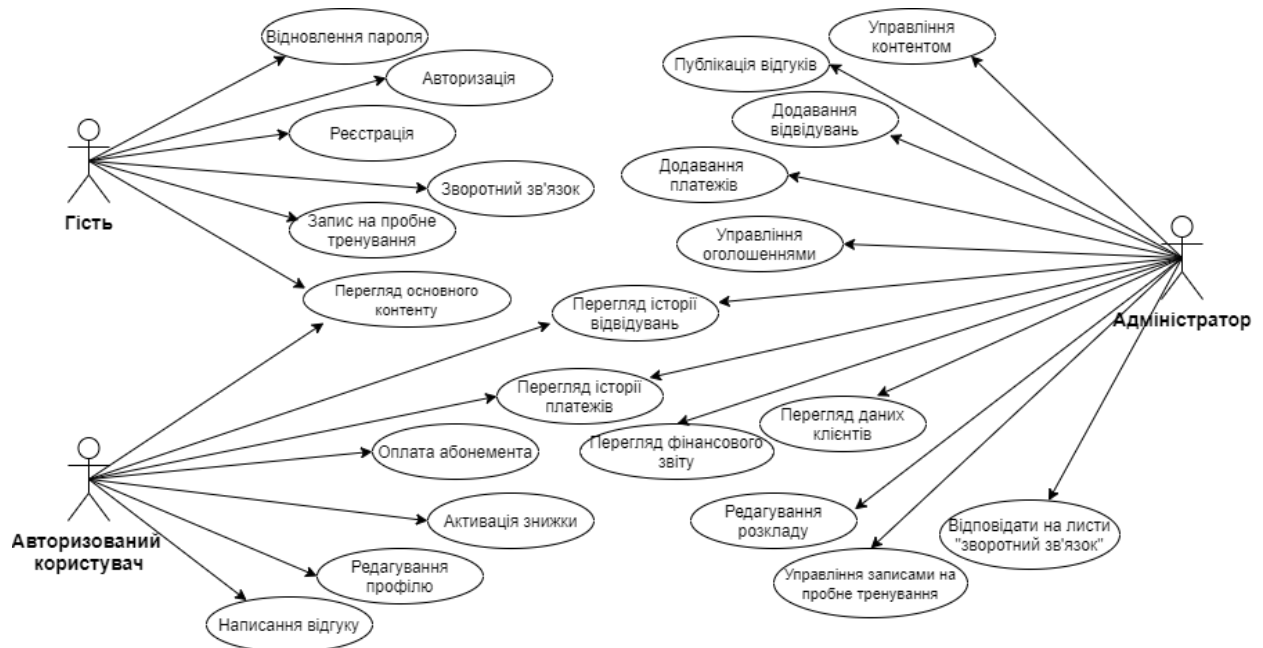


Рисунок 3.4 — Діаграма варіантів використання

Інтернет-додаток містить (БД) базу даних, розроблену в PhpMyAdmin (вебдодаток, що представляє собою вебінтерфейс для адміністрування СУБД).

На рисунку 3.5 представлена концептуальна модель БД системи.

В даній базі даних, яка була створена для даної вебсистеми міститься 11 таблиць:

- abonements — містить інформацію про абонементи;
- abonement_types — містить інформацію про типи абонементів;
- reviews — зберігає відгуки авторизованих користувачів;
- feedback — призначена для зберігання листів від відвідувачів;
- sport_directions — містить спортивні напрямки;
- notifications — містить оголошення;

- contact_info — зберігає контактну інформацію клубу;
- training_history — призначена для зберігання історії відвідувань;
- tariffs — містить інформацію про оплачені абонементи;
- schedule — зберігає графік роботи клубу;
- accounts — містить інформацію про зареєстрованих користувачів.

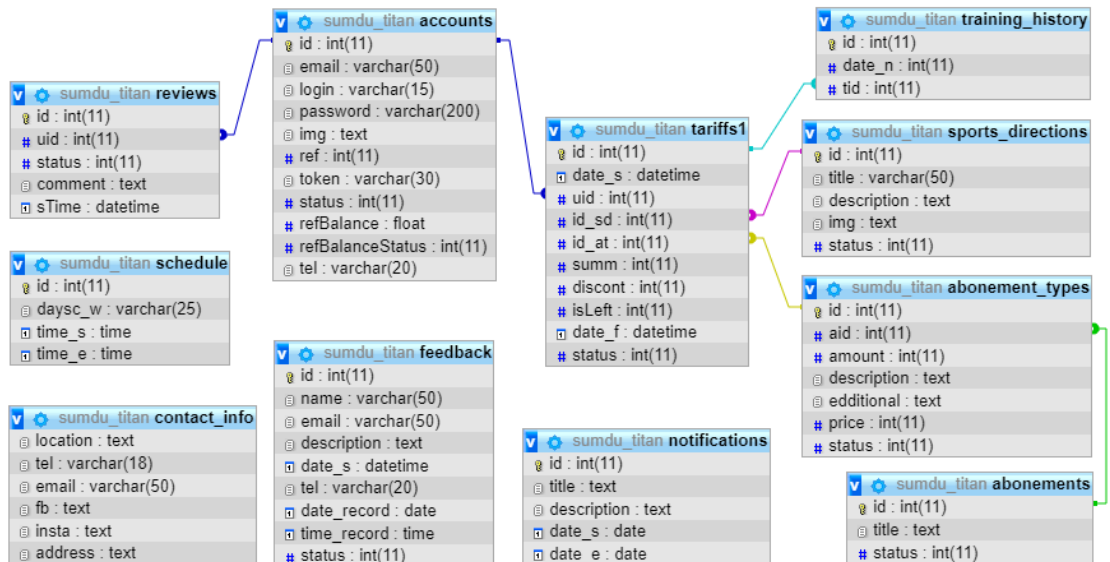


Рисунок 3.5 – Концептуальна модель БД

Головний розділ є обличчям усієї вебсистеми, тому важливо зробити його привабливим. В даній системі на головній сторінці використовується простий і в той же час привабливий дизайн (рис. 3.6). Контент даної сторінки завантажується по мірі прокручування її вниз, що реалізовано за допомогою Stellar.js (плагін jQuery). На головній сторінці можна побачити:

- шапку;
- блок з описом спортклубу;
- розділ з описом послуг (напрямоків), які надає клуб;
- розділ з відгуками клієнтів;
- розділ з прайс листом;
- підвал з контактною інформацією та формою зворотного зв'язку.

В результаті використання основних інструментів Bootstrap, був реалізований адаптивний інтерфейс під різні розширення екранів чи

смартфонів. Приклад роботи показано на рисунку 3.7.

На блоці з описом спортклубу головної сторінки є кнопка «Пробное занятие», що викликає модальне вікно, яке містить форму з полем для вводу номера телефону (рис. 3.8) з перевіркою валідності.

Форма для зворотного зв'язку (рис. 3.9) призначена для відправлення повідомлення адміністратору спортклубу в систему. Всі поля форми є обов'язковими для заповнення та підтримують валідацію на стороні клієнта.

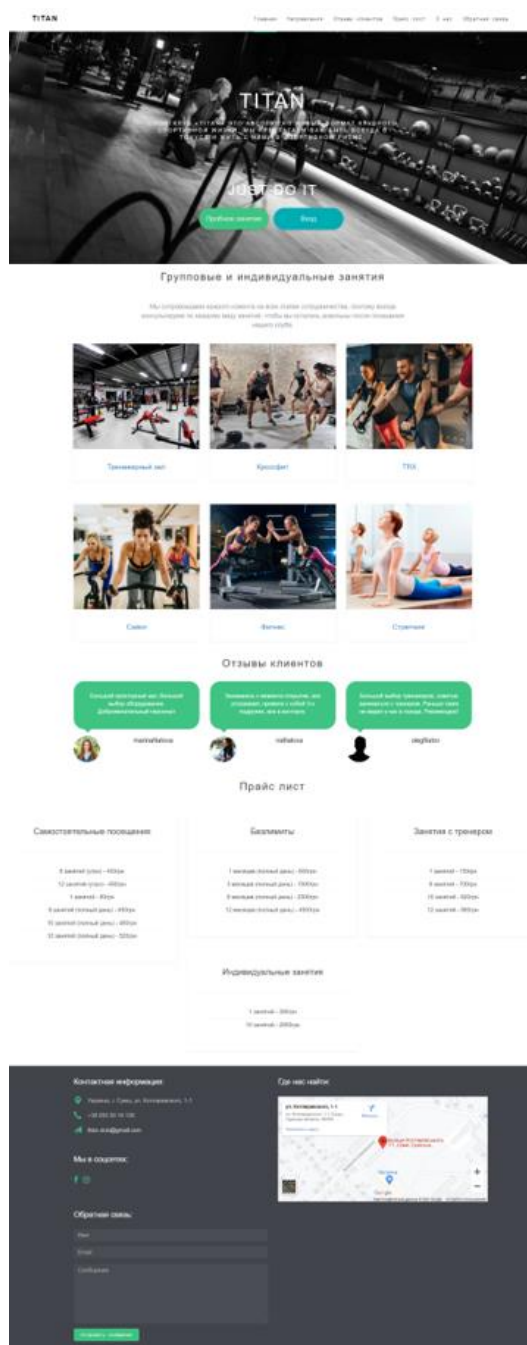


Рисунок 3.6 — Головна сторінка сайту

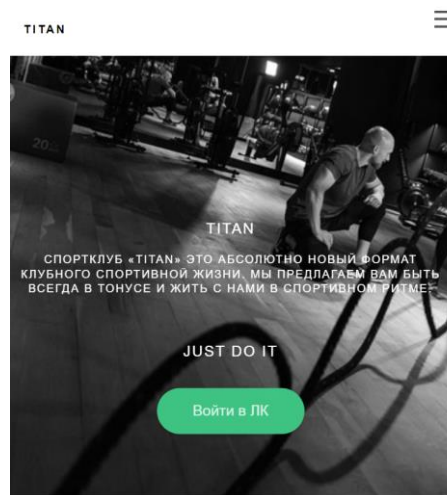


Рисунок 3.7 — Мобільний вигляд вебдодатку

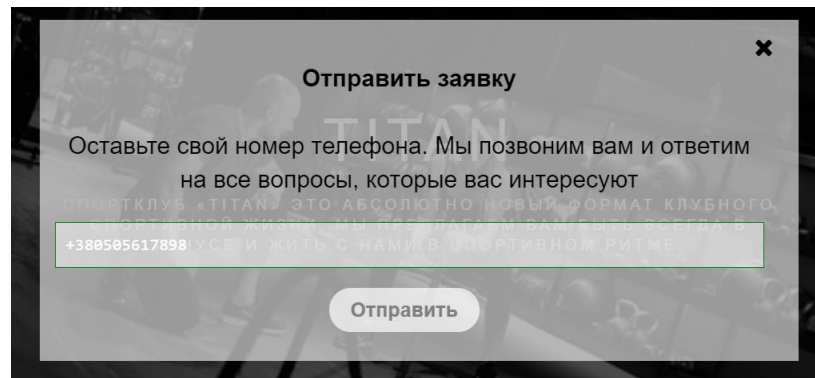


Рисунок 3.8 — Модальне вікно для відправки номера телефону

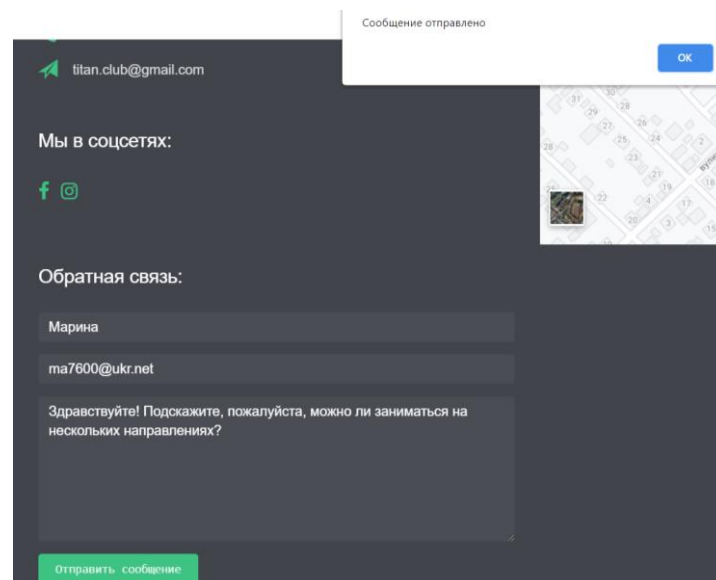


Рисунок 3.9 — Форма зворотного зв'язку з повідомленням про успішне виконання операції

У системі передбачено два варіанти реєстрації: через посилання-запрошення або самостійно.

Щоб зареєструватись у системі, користувачу необхідно заповнити наступні дані форми: електронна пошта, логін (поле дозволяє вводити латинські букви та цифри від 3 до 15 символів), пароль (до вводу дозволені латинські букви та цифри від 10 до 30 символів). Всі поля форми є обов'язковими до заповнення, а поле для логіна має перевірку на унікальність на стороні сервера (рис. 3.10).

При спробі введення некоректних даних, користувач отримає попереджувальне повідомлення.

Рисунок 3.10 — Сторінка реєстрації користувача системи

У випадку, якщо логін та пошта пройшли перевірку на унікальність та коректність в методах `validate`, `checkEmailExists` та `checkLoginExists`, наведених у Додатку Б, для реєстрації використовується метод `register`:

```
public function register($post) {
    $token = $this->createToken();
    if ($post['ref'] == 'none') {
        $ref = 0;
    }
    else {
        $ref = $this->checkRefExists($post['ref']);
        if (!$ref) {
            $ref = 0;
        }
    }
    $params = [
        'email' => $post['email'],
        'login' => $post['login'],
        'password' => password_hash($post['password'], PASSWORD_BCRYPT),
        'ref' => $ref,
```

```

        'token' => $token,
        'status' => 0,
        'refBalance' => 0,
    ];
    $this->db->query('INSERT INTO
accounts(email,login,password,ref,token,status,refBalance) VALUES (:email,
:login, :password, :ref, :token, :status, :refBalance)', $params);
    mail($post['email'], 'Спортклуб Titan - Регистрация', 'Для
подтверждения регистрации перейдите по ссылке: '.'
http://localhost/Titan_final.loc/account/confirm/'.$token);
    }

```

Відмінність способів реєстрації полягатиме в додатковому неактивному полі на формі реєстрації (рис. 3.11), що містить логін користувача, який запросив до спортклубу, і можливостей, описаних далі.

Рисунок 3.11 — Сторінка реєстрації користувача за запрошенням

Для процедури авторизації користувачеві необхідно використовувати логін та пароль від особистого кабінету (рис. 3.12).

Рисунок 3.12 — Сторінка авторизації користувача системи

Для відновлення пароля клієнту клубу необхідно вставити електронну пошту, на яку реєструвався на сторінці відновлення пароля (рис. 3.13).

Якщо користувач вкаже пошту, на яку він не реєструвався, йому буде відображатися повідомлення з текстом помилки. Якщо користувач вказав вірну пошту, йому буде відіслане повідомлення на електронну пошту з посиланням на відновлення пароля. Після переходу за посиланням користувачу буде надано новий пароль для входу у систему. Приклад наведено на рисунку 3.14.

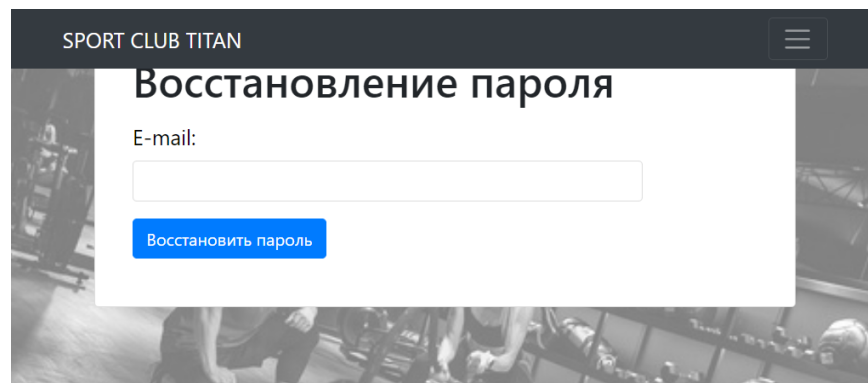


Рисунок 3.13 — Сторінка відновлення пароля у системі

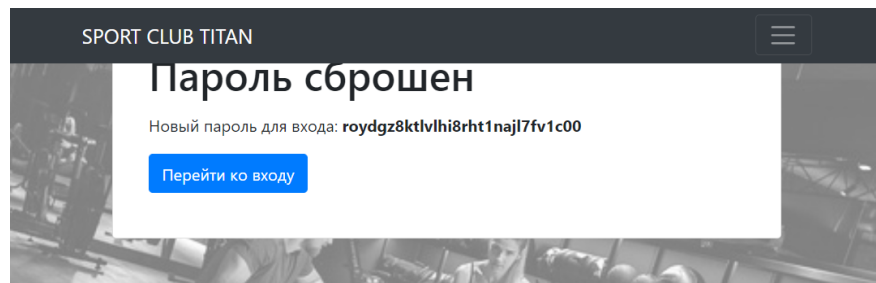


Рисунок 3.14 — Сторінка з новим паролем для входу

Після успішної авторизації клієнт клубу потрапляє на головну сторінку, де відображається меню зі всіма доступними сторінками для користувача, а саме: абонементи, історія платежів, історія відвідувань, бонуси, профіль, відгук, вихід та контент-блок. Тут можливо переглянути актуальний графік роботи клубу та оголошення на сьогоднішню дату, додані адміністратором з панелі управління. Вигляд головної сторінки представлений на рисунку 3.15.

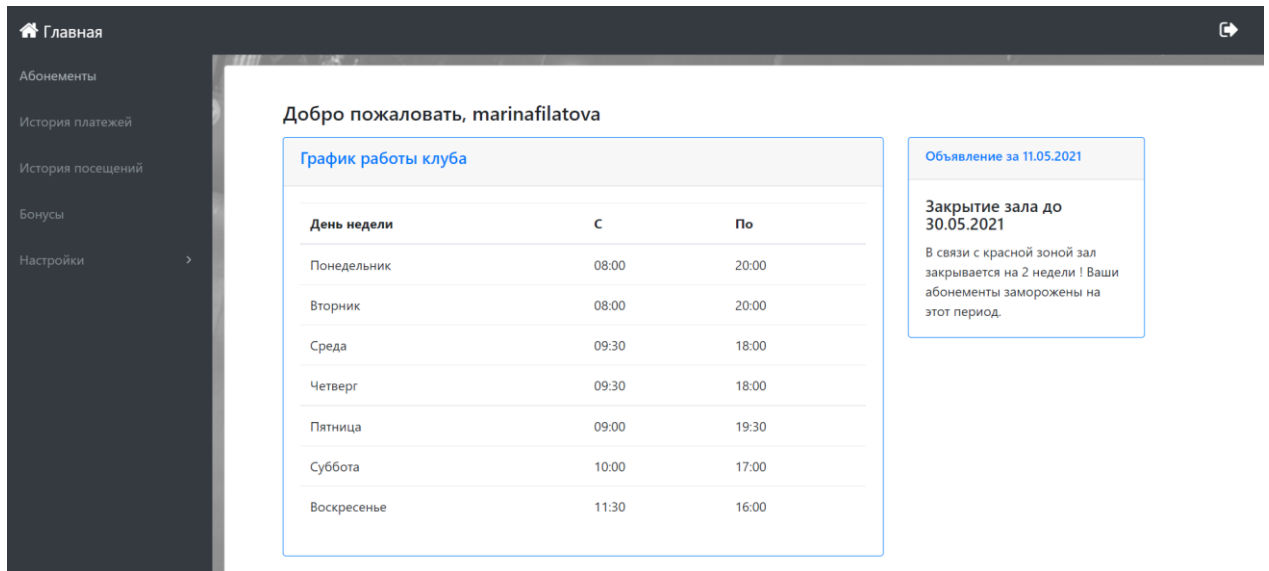


Рисунок 3.15 — Головна сторінка особистого кабінету клієнта

Розділ «Бонуси» (рис. 3.16). Оскільки спортклуб має систему знижок, тому її було реалізовано в CRM-системі. Якщо клієнт реєструється в спортклубі та оплачує абонемент за рекомендацією учасника спортклубу, то тому, хто рекомендував, автоматично нараховується знижка в розмірі 5% від вартості абонементу, яку надалі можна буде використати при оплаті абонементу. Контентом в цьому розділі є: посилання на реєстрацію від власного імені; накопичена сума знижки; список клієнтів, яких запросив користувач. Після активації знижки, користувач отримає повідомлення про успішну активацію та можливість оплатити абонемент зі знижкою.

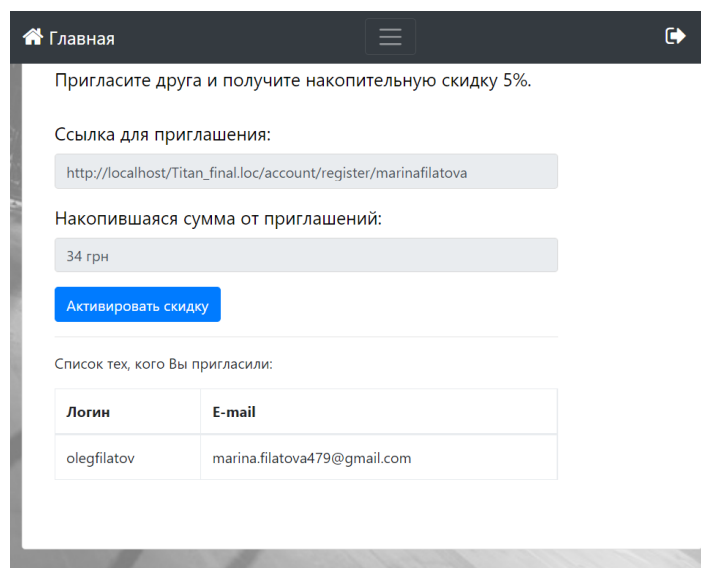


Рисунок 3.16 — Розділ «Бонуси» CRM-системи

При переході за пунктом «Абонементи», користувач потрапить на сторінку з переліком усіх абонементів та їх типів і кнопкою «Оплатить» до кожного (рис. 3.17). Дана кнопка є не тільки посиланням на формування квитанції, а й також посиланням для перегляду більш детальної інформації щодо абонементу, який буде оплачено.

Самостоятельные посещения			
Количество	Тип	Время	Стоимость
8	занятий	(утро)	400
12	занятий	(утро)	450
1	занятий		80
8	занятий	(полный день)	450
10	занятий	(полный день)	480
12	занятий	(полный день)	520

Оплатить

Безлимиты			
Количество	Тип	Время	Стоимость
1	месяцев	(полный день)	600
3	месяцев	(полный день)	1500
6	месяцев	(полный день)	2500
12	месяцев	(полный день)	4500

Оплатить

Рисунок 3.17 — Розділ з абонементами CRM-системи

Натиснувши на кнопку нижче кожного з абонементів, перед користувачем відкривається сторінка з відображенням більш детальної інформації з приводу абонементу, а саме: назва обраного абонементу, поле для вибору спортивного напрямку, поле для вибору типу відповідного абонементу та повідомлення про можливість здійснити оплату з накопиченою знижкою (якщо користувач накопичив), якщо її активувати (рис. 3.18) або додаткове поле з сумою вже активованої знижки (рис. 3.19).

У випадку оплати абонементу зі знижкою, сума, що буде відведена до оплати, розраховується як різниця вартості абонементу та знижки. У разі

перевищення суми знижки вартості абонементу, користувач отримає даний абонемент безкоштовно (сума оплати буде дорівнювати 0).

Активувати знижку можна перейшовши за посиланням «активировать её» під формою, де користувача буде перенаправлено на розділ «Бонуси».

Рисунок 3.18 — Сторінка редагування квитанції

Рисунок 3.19 — Сторінка редагування квитанції зі знижкою

Натиснувши на кнопку «Перейти к оплате» в майбутньому відбудеться запит до API LiqPay (платіжна система) [15]. Результатом виконання запиту буде редирект клієнта на нову вкладку браузера, де відкриється платіжна сторінка LiqPay з методами оплати. Клієнт обирає платіжний метод —

заповнює реквізити карти і підтверджує оплату на платіжній сторінці LiqPay. За результатом LiqPay повертає клієнта на сторінку оплати в CRM-систему зі статусом оплати.

На сьогодні ж відразу виконається запит, який дозволить додати запис про новий абонемент в БД та анулювати суму знижки. Перед доданням нового абонемента відбувається перевірка чи є комбінація обраного спортивного напрямку та типу абонемента унікальною серед існуючих у даного клієнта. В залежності від результату, абонемент стає активним або неактивним.

Логіка роботи описана в методі setRefBal:

```
public function setRefBal($post) {
    if ($dataRef != 0) {
        $refSum = round(($price * 0.05), 0);
        $this->db->query('UPDATE accounts SET refBalance = refBalance +
:sum, refBalanceStatus = 0 WHERE id = :id', $params);
    }
    $amount_days_month = $amount * 31;
    $razn = $price - $refBalanceMy;
    if($razn < 0) {$razn = 0;}
    $amount_copy = $this->db->column('SELECT COUNT(id) FROM tariffs1 where
id_at = :atid', $params);
    if($amount_copy > 0){ $status = 0;}
    else {$status = 1;}
    if ($description == 'занятий') {
        $this->db->query('INSERT INTO
tariffs1(date_s,uid,id_sd,id_at,summ,discont,isLeft,date_f,status) VALUES
(:date_s,:id_account,:id_sd,:id_at,:summ,:discont,:isLeft,NULL,:status)',
$params);
    } else {
        $params = [
            'date_s' => time(),
            'id_account' => $id_account,
            'id_sd' => $sdid,
            'id_at' => $atid,
            'summ' => $razn,
            'discont' => $refBalanceMy,
            'date_f' => strtotime('+ '.$amount_days_month.' days'),
            'status' => $status,
        ];
        $this->db->query('INSERT INTO
tariffs1(date_s,uid,id_sd,id_at,summ,discont,isLeft,date_f,status) VALUES
(:date_s,:id_account,:id_sd,:id_at,:summ,:discont,NULL,:date_f,:status)',
$params);
    }
    $this->db->query('UPDATE accounts SET refBalance = 0, refBalanceStatus
= 0 WHERE id = :id_account', $params);
    return true;
}
```

Для відстеження статусів власних абонементів необхідно перейти в розділ «История платежей» (рис. 3.20). Для користувача буде відображатися вся інформація про його платежі з аналізом статусу активності в залежності від типу тривалості абонементу («заняття» або «місяці»). На основі даних про кількість відвідувань тренувань по конкретному типу абонементу або датам оплати/закінчення абонементу формується журнал.

Дата оплаты	Спортивное направление	Описание	Оплачено	Скидка	Статус абонемента
27.05.2021	Стретчинг	3 месяцев (полный день) 1500грн	1500 грн	0 грн	Закончится 28.08.2021
27.05.2021	Тренажерный зал	8 занятий (утро) 400грн	400 грн	0 грн	Осталось 4 занятия
30.05.2021	Тренажерный зал	8 занятий (утро) 400грн	400 грн	0 грн	Неактивен
14.05.2021	Тренажерный зал	8 занятий (утро) 400грн	366 грн	34 грн	Закончился

Рисунок 3.20 — Розділ «История платежей» CRM-системи

На цій сторінці використовуються методи, що повертають дані з бази даних та аналізують для виводу:

```
public function tariffsAction() {
    $this->model->updateTariffs();
    $pagination = new Pagination($this->route, $this->model->tariffsCount());
    $vars = [
        'pagination' => $pagination->get(),
        'list' => $this->model->tariffsList($this->route),
        'sportList' => $this->model->sportDirectionsList(),
        'abonement_types' => $this->model->abonementTypesList(),
        'abonements' => $this->model->abonementsList(),
    ];
    $this->view->render('История платежей', $vars);
}
```

У розділі «История відвідувань» (рис. 3.21) користувач має змогу не лише переглядати історію власних тренувань, а й здійснювати пошук за власними

абонементами (як за назвою спортивного напрямку, так і за абонементом і його типом). Приклад пошуку наведено на рисунку 3.22.

Даний функціонал реалізовано в методах `historyAction`:

```
public function historyAction() {
    $this->model->updateTariffs();
    if (!empty($_POST)) {
        if ($_POST['type'] == 'search') {
            $res = $this->model->checkClientVisit($_POST['tid']);
            if ($res) { $this->view->location
                ('/Titan_final.loc/dashboard/history/' . $_POST['tid']);
            } else { $this->view->message('', 'Такого абонемента не
                существует. Выберите из списка.');
```

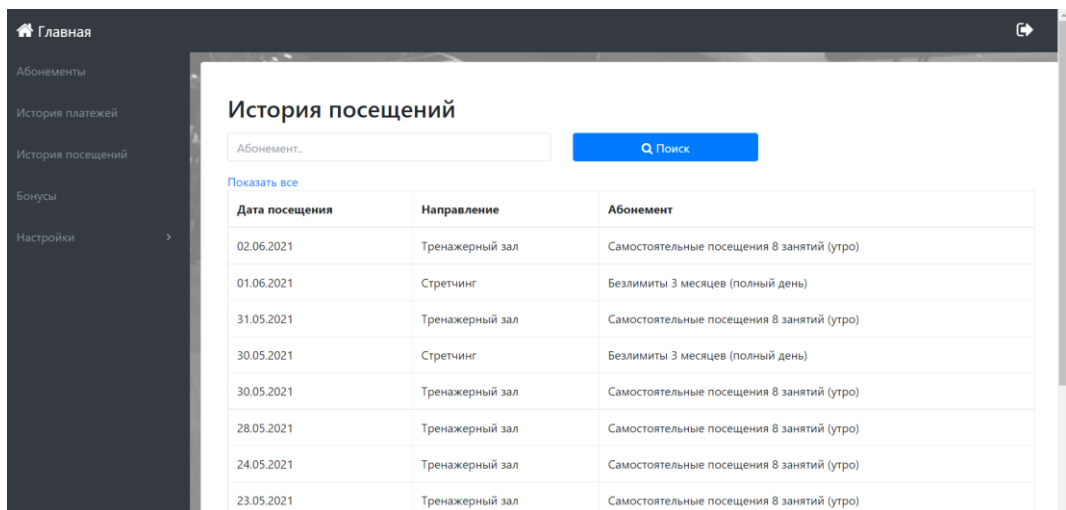


Рисунок 3.21 — Розділ «Історія відвідувань» CRM-системи

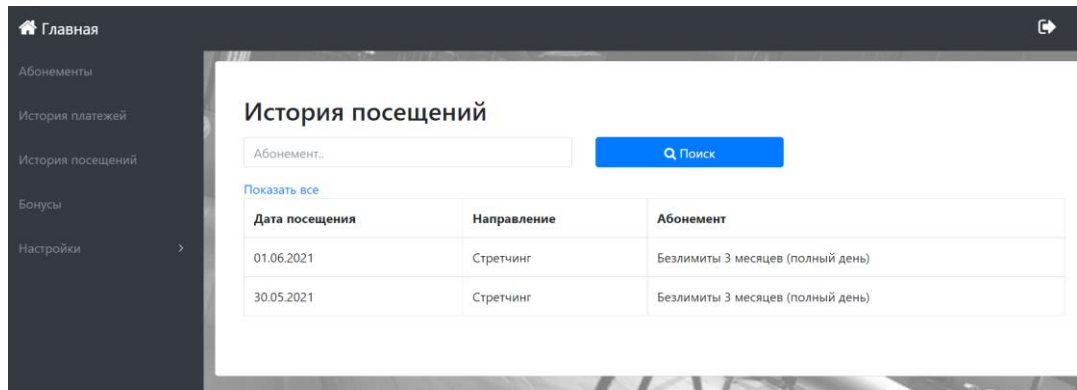


Рисунок 3.22 — Результат поиска в истории посещений

Наступний розділ в CRM-системі є «Мій відгук» (рис. 3.23). В даному розділі користувач має можливість написати відгук або змінити його, якщо вже існує.

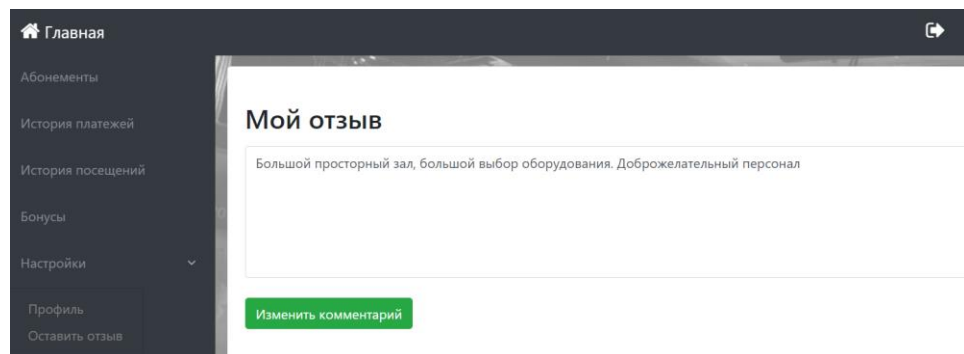


Рисунок 3.23 — Розділ «Мій відгук» CRM-системи

Наступний розділ в CRM-системі є «Профіль» (рис. 3.24). В даному розділі користувач має можливість додати номер телефону та фото, а також редагувати існуючу інформацію.

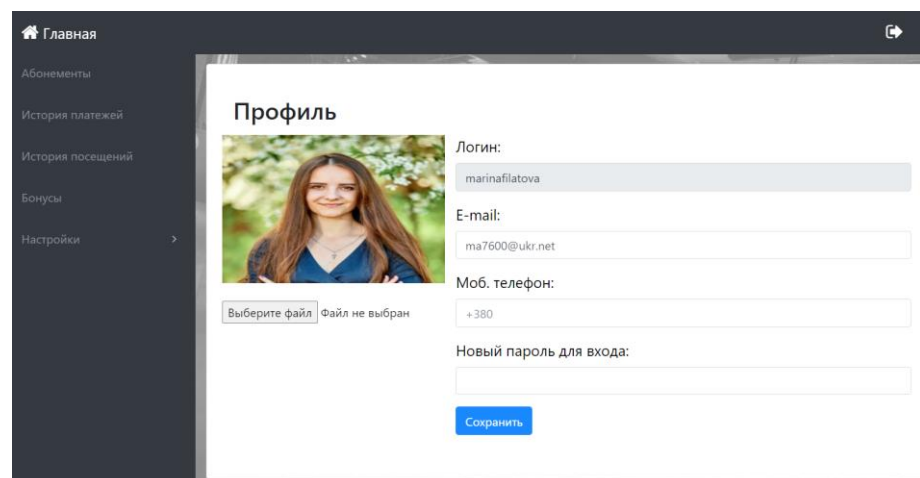


Рисунок 3.24 — Розділ «Історія відвідувань» CRM-системи

Останній пункт навігаційного меню — «Вихід» у вигляді значка. Натиснувши на нього користувач закінчує роботу з сесією і виходить із системи. Після чого він стає звичайним гостем в системі.

На сьогодні переважна більшість ресурсів в Інтернеті є динамічними сайтами. Попит на створення сайтів у вигляді набору статичних HTML-сторінок зовсім зник.

Для управління системою, адміністратору необхідно здійснити вхід ввівши логін та пароль (рис. 3.25). Після успішної авторизації він потрапляє на сторінку управління абонементом клієнтів (рис. 3.26), де може переглядати всю інформацію, здійснювати пошук за логіном клієнта.

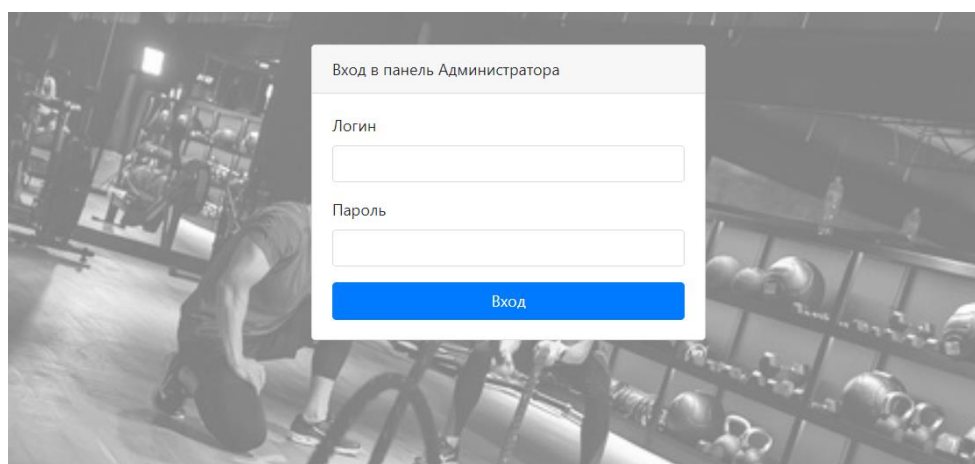


Рисунок 3.25 — Сторінка з формою авторизації адміністратора

Панель Администратора

Управление платежами

+ Добавить платеж Клиент... Поиск

Показать все

Дата оплаты	Клиент	Спортивное направление	Описание	Оплачено	Скидка	Статус абонемента
31.05.2021	irafilatova	TRX	1 занятий 150 грн	95 грн	55 грн	Осталось 1 занятий
30.05.2021	olegfilatov	Фитнес	1 занятий 300 грн	300 грн	0 грн	Осталось 1 занятий
30.05.2021	marinafilatova	Тренажерный зал	8 занятий (утро) 400 грн	400 грн	0 грн	Неактивен
27.05.2021	marinafilatova	Стретчинг	3 месяцев (полный день) 1500 грн	1500 грн	0 грн	Закончится 28.08.2021
27.05.2021	marinafilatova	Тренажерный зал	8 занятий (утро) 400 грн	400 грн	0 грн	Осталось 4 занятий
14.05.2021	marinafilatova	Тренажерный зал	8 занятий (утро)	366 грн	34 грн	Закончился

Рисунок 3.26 — Сторінка управління абонементом

Для контролю абонементів адміністратор має можливість додавати абонементи у разі неможливості клієнта здійснити оплату онлайн (рис. 3.27). Поле для вибору та пошуку напрямку або типу абонемента надає перелік лише тих, які існують (активовані) адміністратором (далі більш детально розглянуто в панелі управління контентом).

Рисунок 3.27 — Форма додавання оплати абонемента

Наступний розділ в CRM-системі є «Історія відвідувань» (рис. 3.28). У адміністратора є можливість здійснювати пошук по обраному абонементу конкретного клієнта (якщо вже були відвідування) та додавати нове, обравши з переліку активних абонементів (рис. 3.29).

На даній сторінці використовується три методи. Перший повертає інформацію про платежі, другий здійснює пошук по платежам, а третій додає новий платіж в БД:

```
public function historyAction() {
    $this->model->updateTariffs();
    if (!empty($_POST)) {
        if ($_POST['type'] == 'search') {
            $res = $this->model->checkClientVisit($_POST['tid']);
            if ($res) { $this->view-
>location('/Titan_final.loc/admin/history/' . $_POST['tid']);
            } else { $this->view->message('', 'Такого абонемента не существует.
Выберите из списка.');
```



```

        $this->view->message('', 'Посещение добавлено');
    }else { $this->view->message('', 'Данный клиент уже отмечен
сегодня');}
    } else {
        $this->view->message('', 'Такого абонемента не существует. Выберите
из списка.');
```

Панель Администратора

История посещений

[+ Добавить посещение](#) [Поиск](#)

[Показать все](#)

Дата посещения	Клиент	Направление	Абонемент
02.06.2021	marinafilatova	Тренажерный зал	Самостоятельные посещения 8 занятий (утро)
01.06.2021	marinafilatova	Стретчинг	Безлимиты 3 месяцев (полный день)
31.05.2021	marinafilatova	Тренажерный зал	Самостоятельные посещения 8 занятий (утро)
30.05.2021	marinafilatova	Стретчинг	Безлимиты 3 месяцев (полный день)
30.05.2021	marinafilatova	Тренажерный зал	Самостоятельные посещения 8 занятий (утро)
28.05.2021	marinafilatova	Тренажерный зал	Самостоятельные посещения 8 занятий (утро)
24.05.2021	marinafilatova	Тренажерный зал	Самостоятельные посещения 8 занятий (утро)
23.05.2021	marinafilatova	Тренажерный зал	Самостоятельные посещения 8 занятий (утро)
22.05.2021	marinafilatova	Тренажерный зал	Самостоятельные посещения 8 занятий (утро)
20.05.2021	marinafilatova	Тренажерный зал	Самостоятельные посещения 8 занятий (утро)
19.05.2021	marinafilatova	Тренажерный зал	Самостоятельные посещения 8 занятий (утро)

Рисунок 3.28 — Сторінка «Історія відвідувань»

Добавить посещение

Клиент

- 2 Тренажерный зал marinafilatova Самостоятельные посещения 8 занятий (утро)
- 3 Стретчинг marinafilatova Безлимиты 3 месяцев (полный день)
- 6 Фитнес olegfilatov Индивидуальные занятия 1 занятий
- 7 TRX irafilatova Занятия с тренером 1 занятий

Рисунок 3.29 — Форма додавання запису про відвідування

У розділі «Контент» є ряд функцій, які необхідні для повноцінної роботи всієї системи в цілому. Дії, які можна проводити над даними спортивних

напрямоків (рис. 3.30), абонементів та їх типів (рис. 3.31-3.33), контактною інформацією (рис. 3.34) та графіком роботи спортклубу (рис. 3.35) наступні: додавати, редагувати та видаляти/відновлювати.

Здесь Вы можете изменить информацию сайта спортклуба

Направления **Абонементы** Контактная информация График работы

Добавьте направление:

Название	Описание	Изображение	
<input type="text"/>	<input type="text"/>	Выберите файл <small>Файл не выбран</small>	<input type="button" value="Добавить"/>

Управление направлениями:

Название	Описание	Изображение	
Стретчинг	Растяжка мышц помогает развить гибкость, улучшить кровообращение и подтянуть все тело		<input type="button" value="Восстановить"/>
Фитнес	Чтобы улучшить общее состояние организма, спортивный клуб B-tone приглашает вас на занятия фитнесом		<input type="button" value="Удалить"/>
Сайкл	Это динамические упражнения на велотренажере, которые позволяют очень быстро избавиться от лишних калорий		<input type="button" value="Удалить"/>

Рисунок 3.30 — Вкладка управління спортивними напрямками

Здесь Вы можете изменить информацию сайта спортклуба

Направления **Абонементы** Контактная информация График работы

Управление абонементами **Управление типами абонементов**

Добавить абонемент

Название	
<input type="text"/>	<input type="button" value="Добавить"/>

Изменить абонемент

Выберите абонемент:

Занятия с тренером

Новое название:

[Удалить/Восстановить* абонемент](#)

*** Вы можете удалить (активировать/деактивировать) абонемент**

Название	
Индивидуальные занятия	<input type="button" value="Восстановить"/>
Занятия с тренером	<input type="button" value="Удалить"/>
Безлимиты	<input type="button" value="Удалить"/>
Самостоятельные посещения	<input type="button" value="Удалить"/>

Рисунок 3.31 — Вкладка управління абонементом

У разі видалення абонементу або типу, відредагувати його буде неможливо, а лише після відновлення. Дане правило застосовується і для спортивних напрямків.

Направления | Абонементы | Контактная информация | График работы

Управление абонементом | Управление типами абонементов

Добавить тип абонемента

Название абонемента	Количество	Тип	Время	Стоимость	
Индивидуальные за <input type="text"/>	<input type="text"/>	заня <input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Добавить"/>

Изменить тип абонемента

Название абонемента	Количество	Тип	Время	Стоимость	
Индивидуальные занятия	1	занятий		300	<input type="button" value="Сохранить"/>
Индивидуальные занятия	10	занятий		2000	<input type="button" value="Сохранить"/>
Занятия с тренером	1	занятий		150	<input type="button" value="Сохранить"/>
Занятия с тренером	8	занятий		700	<input type="button" value="Сохранить"/>
Занятия с тренером	10	занятий		820	<input type="button" value="Сохранить"/>
Занятия с тренером	12	занятий		900	<input type="button" value="Сохранить"/>
Безлимиты	1	месяцев	(полный день)	600	<input type="button" value="Сохранить"/>

Рисунок 3.32 — Вкладка для додавання та редагування типів абонементів

Направления | Абонементы | Контактная информация | График работы

Управление абонементом | Управление типами абонементов

Добавить тип абонемента

Изменить тип абонемента

Удалить/Восстановить* тип абонемента

* Вы можете удалить (активировать/деактивировать) тип абонемента

Название абонемента	Количество	Тип	Время	Стоимость	
Индивидуальные занятия	1	занятий		300	<input type="button" value="Восстановить"/>
Индивидуальные занятия	10	занятий		2000	<input type="button" value="Удалить"/>
Занятия с тренером	1	занятий		150	<input type="button" value="Удалить"/>
Занятия с тренером	8	занятий		700	<input type="button" value="Удалить"/>
Занятия с тренером	10	занятий		820	<input type="button" value="Удалить"/>
Занятия с тренером	12	занятий		900	<input type="button" value="Удалить"/>
Безлимиты	1	месяцев	(полный день)	600	<input type="button" value="Удалить"/>
Безлимиты	3	месяцев	(полный день)	1500	<input type="button" value="Удалить"/>
Безлимиты	6	месяцев	(полный день)	2500	<input type="button" value="Удалить"/>

Рисунок 3.33 — Вкладка для видалення/відновлення типів абонементів

Поля усіх форм є обов'язковими для заповнення, спеціальні символи будуть екрануватись для запобігання вставки шкідливого коду.

Здесь Вы можете изменить информацию сайта спортклуба

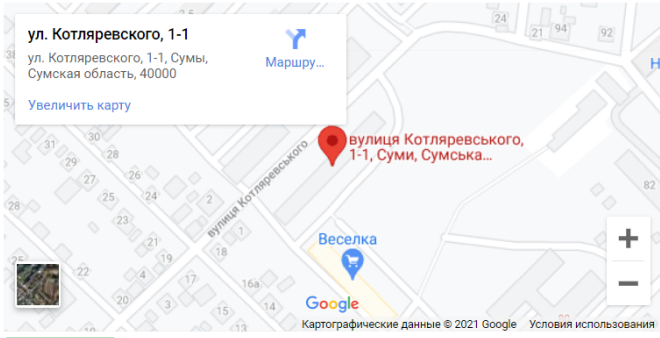
[Направления](#) [Абонементы](#) [Контактная информация](#) [График работы](#)

Контактная информация:

Украина, г. Сумы, ул. Ко: +38 050 56 19 156 titan.club@gmail.com

Мы в соцсетях:

Где нас найти:



ул. Котляревского, 1-1
ул. Котляревского, 1-1, Сумы, Сумская область, 40000

Увеличить карту

Маршру...

вулиця Котляревського, 1-1, Суми, Сумська...

Веселка

Google

Картографические данные © 2021 Google Условия использования

Сохранить

Рисунок 3.34 — Вкладка редагування контактної інформації

Здесь Вы можете изменить информацию сайта спортклуба

[Направления](#) [Абонементы](#) [Контактная информация](#) [График работы](#)

Вы можете изменить график работы

День недели:

Понедельник

Время начала работы: --:--

Время конца работы: --:--

Сохранить

День недели	С	По
Понедельник	08:00	20:00
Вторник	08:00	20:00
Среда	09:30	18:00
Четверг	09:30	18:00
Пятница	09:00	19:30
Суббота	10:00	17:00
Воскресенье	11:30	16:00

Рисунок 3.35 — Вкладка редагування графіка роботи клубу

Розділ «Оголошення» (рис. 3.36). Контентом в цьому розділі є інформація з приводу всіх запланованих оголошень, які створив адміністратор та форма для додавання нових у систему.

Для створення оголошення необхідно заповнити поля заголовку, опису, дати початку і кінця, які являються індикаторами старту і закінчення періоду відображення на головній сторінці у клієнтів (див. рис. 3.15) та натиснути кнопку «Добавить». В результаті кожен користувач не тільки побачить оголошення в зазначений період на головній сторінці особистого кабінету, а й отримає повідомлення на пошту зі змістом оголошення. Також є можливість видалення оголошень з системи.

Здесь Вы можете управлять объявлениями

Добавьте запланированное объявление:
* - дата начала/окончания отображения объявления на доске клиента

Заголовок	Описание	Дата начала *	Дата окончания *	
<input style="width: 90%;" type="text"/>	<input style="width: 95%;" type="text"/>	дд.мм.гггг	дд.мм.гггг	<input type="button" value="Добавить"/>

Объявления:

Заголовок	Описание	Дата начала	Дата окончания	
Зал закрыт 14.05.2021	Извините за неудобства. Происходят ремонтные работы.	2021-05-13	2021-05-17	<input type="button" value="Удалить"/>
Закрытие зала до 30.05.2021	В связи с красной зоной зал закрывается на 2 недели ! Ваши абонементы заморожены на этот период.	2021-05-11	2021-05-30	<input type="button" value="Удалить"/>

Рисунок 3.36 — Сторінка «Оголошення» CRM-системи

Метод, який створює оголошення та робить розсилку повідомлень на пошту клієнтам клубу:

```
public function addNotification($post) {
    $title = htmlspecialchars($post['title']);
    $description = htmlspecialchars($post['description']);
    $date_s = $post['date_s'];
    $date_e = $post['date_e'];
    $params = [
        'title' => htmlspecialchars($post['title']),
        'description' => htmlspecialchars($post['description']),
        'date_s' => $post['date_s'],
        'date_e' => $post['date_e'],
    ];
};
```

```

$this->db->query('insert into notifications(title,
description,date_s,date_e) values(:title, :description, :date_s,:date_e)',
$params);
if((date("Y-m-d") >= $date_s) && (date("Y-m-d") <= $date_e)) {
    $result = $this->db->row('SELECT * FROM accounts');
    if (!empty($result)) {
        foreach ($result as $key => $val) {
            mail($val['email'], 'Объявление: '.$title, $description);
        }
    }
}return true;
}

```

Всю інформацію про кожного клієнта, яка згадується в системі, а саме: логін, фотокартку, пошту, логін того, хто запросив, номер телефону, стан акаунта та суму накопиченої знижки адміністратор може переглянути на сторінці «Клієнти» (рис. 3.37)




Клиенты клуба						
Логин	Email	Моб. тел.	Кто пригласил	Статус акаунта	Сумма скидки	
	irafilatova	mf2617770@gmail.com		Активирован	0 грн	
	olegfilatov	marina.filatova479@gmail.com	+(380)505619065	marinafilatova	Активирован	0 грн
	marinafilatova	ma7600@ukr.net		irafilatova	Активирован	15 грн

Рисунок 3.37 — Сторінка «Клієнти» CRM-системи

Адміністратор може видалити чи схвалити відгук клієнта (рис. 3.38). Після схвалення адміністратором відгуку, він з'являється у відповідному розділі на головній сторінці сайту спортклубу (див. рис. 3.6).

Отзывы			
Дата	Клиент	Комментарий	
10.05.2021 в 17:44	olegfilatov	Большой выбор тренажеров, советую заниматься с тренером. Раньше таких не видел у нас в городе. Рекомендую!	Опубликовать
10.05.2021 в 17:37	irafilatova	Занимаюсь с момента открытия, все устраивает, привела с собой 3-х подружек, все в восторге.	Удалить
21.04.2021 в 16:48	marinafilatova	Большой просторный зал, большой выбор оборудования. Доброжелательный персонал	Удалить

Рисунок 3.38 — Сторінка управління відгуками

При переході до категорії звітів, відображається інформація про зароблені кошти за поточний місяць сумарно та по напрямках (рис. 3.39).

Отчет за <u>05.2021</u>	
Направление	Сумма, грн
Тренажерный зал	1166
TRX	95
Фитнес	300
Стретчинг	1500
	3061 грн

Рисунок 3.39 — Сторінка звітів

При переході до категорії зворотного зв'язку (рис. 3.40), відображаються усі листи відвідувачів сайту клубу з інформацією про час, пошту та ім'я відправника. Розгорнувши лист, можна одразу надіслати відповідь відправнику на пошту. Після відправлення листа запис стає сірого кольору та відбувається сортування записів.

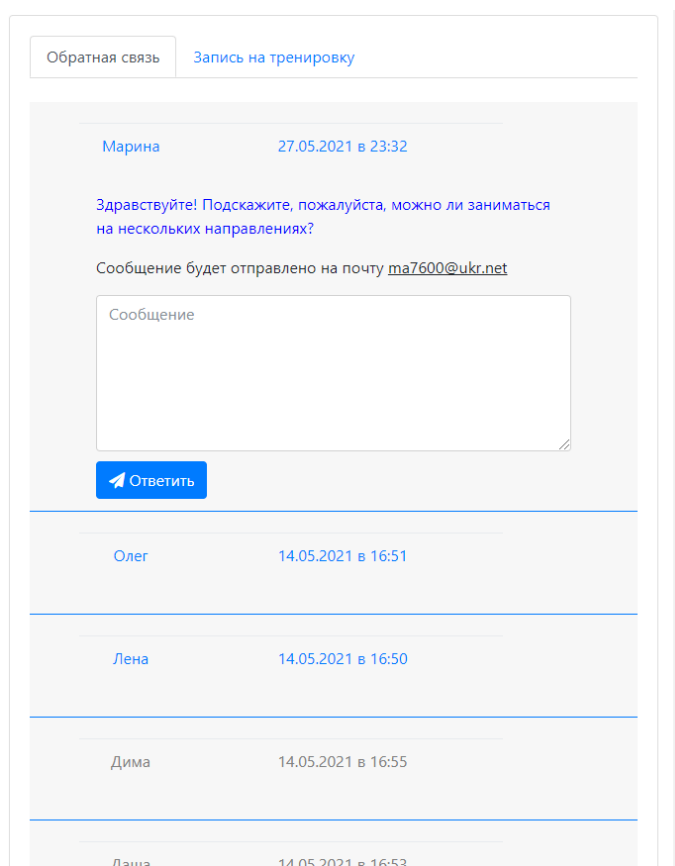
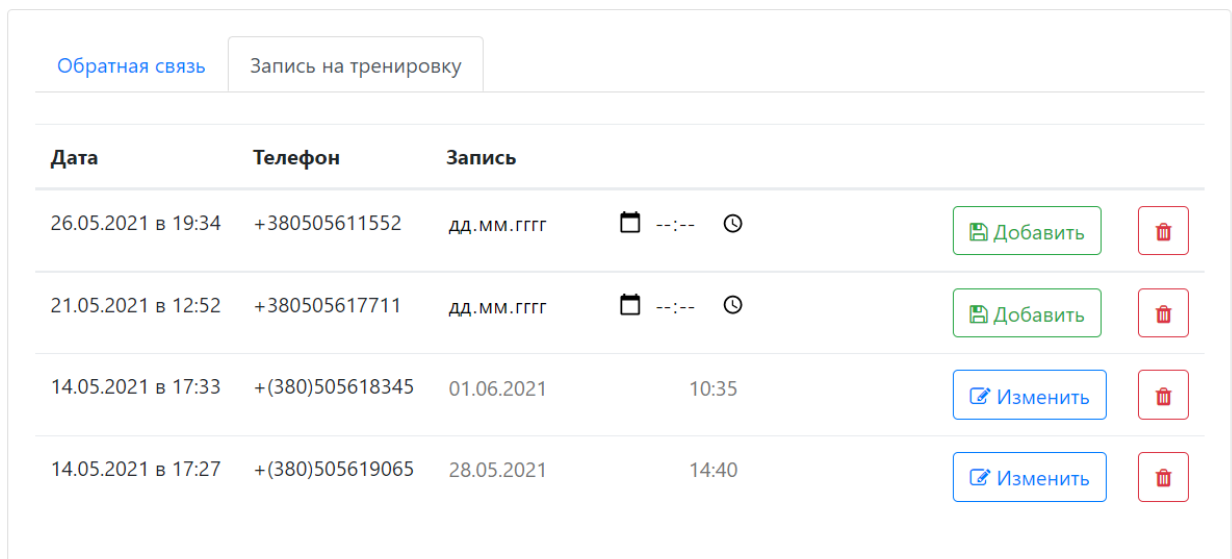


Рисунок 3.40 — Сторінка зворотного зв'язку

Дії щодо зміни статусу листів від відвідувачів сайту спортклубу та відправка відповідей на них наведено у методі response:

```
public function response($post) {
    $id = $post['id'];
    $email = $post['email'];
    $title = 'Ответ на Ваше сообщение';
    $message = $post['message'];
    $params = [
        'id' => $post['id'],
    ];
    $this->db->query('UPDATE feedback set status = 1 WHERE id = :id',
    $params);
    $result = $this->db->row('SELECT * FROM accounts');
    mail($email, $title, 'Спасибо за интерес к нашему спортклубу.
    '.$message);
    return true;
}
```

Перейшовши на вкладку записів на тренування тієї ж сторінки, будуть відображені номери телефонів зацікавлених осіб, які прагнуть прийти на пробне тренування (рис. 3.41). Тренер має можливість додавати або змінювати дату і час назначених тренувань, а також видаляти їх.



Дата	Телефон	Запись		
26.05.2021 в 19:34	+380505611552	дд.мм.гггг	📅 --:--	🕒
21.05.2021 в 12:52	+380505617711	дд.мм.гггг	📅 --:--	🕒
14.05.2021 в 17:33	+(380)505618345	01.06.2021	10:35	
14.05.2021 в 17:27	+(380)505619065	28.05.2021	14:40	

Рисунок 3.41 — Вкладка записів на тренування

ВИСНОВКИ

Була сформульована мета створити веб-орієнтовану CRM-систему для спортклубу «Титан», що дозволить систематизувати та структурувати всі дані про клієнтів і роботі з ними.

Під час створення дипломного проєкту був проведений детальний аналіз предметної області. На базі аналогів спортклубів та CRM були виявлені недоліки та переваги, відштовхуючись від яких були виділені функції, які повинна виконувати система; були визначені учасники інформаційного обміну у системі та створено схему взаємодії користувачів (use-case); було проведено вибір програмних засобів для реалізації. Також було спроектовано та реалізовано базу даних системи.

Завдяки використанню сучасних вебтехнологій вдалося створити функціональний, зручний та інтуїтивно-зрозумілий інтерфейс для різних категорій користувачів, яким легко користуватися у сучасних браузерах та при різних розширеннях екранів. Також було здійснено тестування системи.

В результаті роботи було створено вебсистему, яка безумовно полегшить процес ведення обліку клієнтів клубу, надасть можливість легко контролювати роботу закладу та оперативно знаходити потрібну інформацію. Окрім цього, автоматизація спортклубу позитивно вплине на динаміку продажів.

СПИСОК ЛІТЕРАТУРИ

1. Акулич Маргарита. Управление взаимоотношениями с клиентами (CRM)/Маргарита Акулич. — Москва: Издательские решения, 2019. — 102 с.
2. Фейдер Питер. Клиентоцентричность. Отношения с потребителями в цифровую эпоху / Питер Фейдер, Сара Томс. — Москва: Альпина Диджитал, 2018. — 150 с.
3. Неретин Юрий. Сборник бизнес-решений. Как прокачать свой бизнес с помощью IT / Юрий Неретин, Александр Огнев. — Москва: Издательские решения, 2018. — 60 с.
4. S2 (SalesapCRM) — CRM системы [Электронный ресурс]. — Режим доступа: <https://salesap.ru/> — (дата звернення 06.04.2021) — Назва з титулу екрана.
5. Арника — специализированная CRM для салона красоты. [Электронный ресурс]. — Режим доступа: <https://arnica.pro/crm> — 04.02.2017р. — Назва з титулу екрана.
6. Шполянская И. Ю. Архитектура адаптивной веб-ориентированной системы взаимодействия с клиентами / И. Ю. Шполянская. — Москва: Синергия, 2017. — 8 с.
7. Мартин Роберт. Чистый код: создание, анализ и рефакторинг / Роберт Мартин. — Санкт-Петербург: Питер, 2018. — 464 с.
8. Колисниченко Денис. PHP и MySQL. Разработка веб-приложений / Денис Колисниченко. — Санкт-Петербург: БХВ-Петербург, 2017. — 619 с.
9. Скляр Дэвид. PHP 7. Руководство по созданию интерактивных веб-сайтов / Дэвид Скляр — Москва: oReilly, 2017. — 382 с.
10. Apache HTTP Server. [Электронный ресурс]. — Режим доступа: https://uk.wikipedia.org/wiki/Apache_HTTP_Server — 25.03.2019р. — Назва з титулу екрана.

11. Резиг Джон. JavaScript для профессионалов + code / Джон Резиг, Расс Фергюсон — Москва: Вильямс, 2017. — 242 с.
12. AJAX — Руководства Web-разработчика. [Электронный ресурс]. — Режим доступа: <https://developer.mozilla.org/ru/docs/Web/Guide/AJAX> — (дата звернения 17.04.2021) — Назва з титулу екрана.
13. Visual Studio Code - Code Editing. Redefined [Электронный ресурс]. — Режим доступа: <https://code.visualstudio.com/> — (дата звернения 20.04.2021) — Назва з титулу екрана.
14. Сильвио Морето. Bootstrap в примерах / Сильвио Морето — Москва: ДМК Пресс, 2017. — 316 с.
15. Платежные и информационные APIs LiqPay. [Электронный ресурс]. — Режим доступа: <https://www.liqpay.ua/documentation/> — (дата звернения 16.05.2021) — Назва з титулу екрана.

ДОДАТКИ

Додаток А

В даному додатку представлений код шаблону головної сторінки спортклубу.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title><?php echo $title; ?></title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="Спортклуб Титан в г.Сумы.
Котляревського 1-1" />
    <meta property="og:type" content="article" />
    <meta property="og:title" content="Спортклуб Титан в г.Сумы" />
    <meta property="og:description" content="Спортклуб Титан в г.Сумы.
Котляревського 1-1" />
    <meta property="og:image"
content="/Titan_final.loc/public/images/1.jpg"/>
    <link rel="stylesheet" href="/Titan_final.loc/public/css/animate.css">
    <link rel="stylesheet"
href="/Titan_final.loc/public/css/icomoon.css">
    <link rel="stylesheet" href="/Titan_final.loc/public/css/simple-line-
icons.css">
    <link rel="stylesheet"
href="/Titan_final.loc/public/css/bootstrap.css">
    <link rel="stylesheet" href="/Titan_final.loc/public/css/style.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <script src="/Titan_final.loc/public/js/modernizr-
2.6.2.min.js"></script>
    <script src="/Titan_final.loc/public/scripts/jquery.js"></script>
    <script src="/Titan_final.loc/public/scripts/form.js"></script>
    <script src="/Titan_final.loc/public/scripts/popper.js"></script>
    <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js"></SCRI
PT>
  </head>
  <body>
    <header role="banner" id="fh5co-header"
style="margin:0px;padding:0px;width:100%;">
    <div class="container">
    <div class="row">
      <nav class="navbar navbar-default navbar-fixed-top">
        <div class="navbar-header">
<a href="#" class="js-fh5co-nav-toggle fh5co-nav-toggle" data-
toggle="collapse" data-target="#navbar" aria-expanded="false" aria-
controls="navbar"><i></i></a>
<a class="navbar-brand" href="/Titan_final.loc">TITAN</a>
        </div>

```

```

        <div id="navbar" class="navbar-collapse collapse">
            <ul class="nav navbar-nav navbar-right">
                <li class="active"><a href="#" data-nav-
section="home"><span>Главная</span></a></li>
                <li><a href="#" data-nav-
section="explore"><span>Направления</span></a></li>
                <li><a href="#" data-nav-section="testimony"><span>Отзывы
клиентов</span></a></li>
                <li><a href="#" data-nav-section="pricing"><span>Прайс
лист</span></a></li>
                <li><a href="#" data-nav-section="contentinfo"><span>0
нас</span></a></li>
                <li><a href="#" data-nav-
section="contentinfo"><span>Обратная связь</span></a></li></ul></div></nav>
            </div>
        </header>
        <?php echo $content; ?>
        <script src="/Titan_final.loc/public/js/jquery.min.js"></script>
        <script src="/Titan_final.loc/public/js/jquery.easing.1.3.js"></script>
        <script src="/Titan_final.loc/public/js/bootstrap.min.js"></script>
        <script
src="/Titan_final.loc/public/js/jquery.waypoints.min.js"></script>
        <script
src="/Titan_final.loc/public/js/jquery.stellar.min.js"></script>
        <script src="/Titan_final.loc/public/js/jquery.countTo.js"></script>
        <script src="/Titan_final.loc/public/js/main.js"></script>
        <script>
            var btn = document.getElementById("btn");
            btn.onclick = function() {open_form1()};
            function open_form1(){
                var form1 = document.getElementById("form1");
                form1.classList.toggle("show");
            }
            function checkTel(){
                var tel = document.getElementById("tel");
                var reg = /^[+]?[0-9\s-\(\)]{3,25}*/i;
                if(tel.value.search(reg) > -1){
                    tel.style.borderColor = "green";
                } else {
                    tel.style.borderColor = "red";
                }
            }
            var send = document.getElementById("send");
            function valid(form){
                var tel = form.tel.value;
                var tel_pattern = /^[+]?[0-9\s-\(\)]{3,25}*/i;
                if (tel_pattern.test(tel) == false){
                    alert("Номер неверного формата");
                    event.preventDefault();
                } else {open_form1()};
            }
        </script>
    </body>
</html>

```

Додаток Б

У даному додатку приведено код моделі Account.

```
<?php
namespace application\models;
use application\core\Model;
class Account extends Model {
    public function validate($input, $post) {
        $rules = [
            'email' => [
                'pattern' => '#^([a-z0-9_.-]{1,20})@([a-z0-9_.-
]+)\.([a-z\..]{2,10})$#',
                'message' => 'E-mail адрес указан неверно',
            ],
            'login' => [
                'pattern' => '#^[a-z0-9]{3,15}$#',
                'message' => 'ЛОГИН указан неверно (разрешены только
латинские буквы и цифры от 3 до 15 символов',
            ],
            'ref' => [
                'pattern' => '#^[a-z0-9]{3,15}$#',
                'message' => 'ЛОГИН пригласившего указан неверно',
            ],
            'password' => [
                'pattern' => '#^[a-z0-9]{10,30}$#',
                'message' => 'Пароль указан неверно (разрешены только
латинские буквы и цифры от 10 до 30 символов',
            ],
            'tel' => [
                'pattern' => '#^([+]?[0-9\s-\(\)]{3,25})*$#',
                'message' => 'Номер телефона указан неверно',
            ],
        ];
        foreach ($input as $val) {
            if (!isset($post[$val]) or
!preg_match($rules[$val]['pattern'], $post[$val])) {
                $this->error = $rules[$val]['message'];
                return false;
            }
        }
        if (isset($post['ref'])) {
            if ($post['login'] == $post['ref']) {
                $this->error = 'Регистрация невозможна';
                return false;
            }
        }
        return true;
    }

    public function checkEmailExists($email) {
        $params = [
            'email' => $email,
        ];
    }
}
```

```

        return $this->db->column('SELECT id FROM accounts WHERE email =
:email', $params);
    }

    public function checkLoginExists($login) {
        $params = [
            'login' => $login,
        ];
        if ($this->db->column('SELECT id FROM accounts WHERE login =
:login', $params)) {
            $this->error = 'Этот логин уже используется';
            return false;
        }
        return true;
    }

    public function activate($token) {
        $params = [
            'token' => $token,
        ];
        $this->db->query('UPDATE accounts SET status = 1, token = ""
WHERE token = :token', $params);
    }

    public function checkRefExists($login) {
        $params = [
            'login' => $login,
        ];
        return $this->db->column('SELECT id FROM accounts WHERE login =
:login', $params);
    }

    public function createToken() {
        return
substr(str_shuffle(str_repeat('0123456789abcdefghijklmnopqrstuvwxy', 30)),
0, 30);
    }

    public function checkTokenExists($token) {
        $params = [
            'token' => $token,
        ];
        return $this->db->column("SELECT id FROM accounts WHERE token =
:token", $params);
    }

    public function register($post) {
        $token = $this->createToken();
        if ($post['ref'] == 'none') {
            $ref = 0;
        }
        else {
            $ref = $this->checkRefExists($post['ref']);
            if (!$ref) {
                $ref = 0;
            }
        }
        $params = [
            'email' => $post['email'],
            'login' => $post['login'],
            'password' => password_hash($post['password'],
PASSWORD_BCRYPT),

```



```

        'ref' => $ref,
        'token' => $token,
        'status' => 0,
        'refBalance' => 0,
    ];
    $this->db->query('INSERT INTO
accounts(email,login,password,ref,token,status,refBalance) VALUES (:email,
:login, :password, :ref, :token, :status, :refBalance)', $params);
    mail($post['email'], 'Спортклуб Titan - Регистрация', 'Для
подтверждения регистрации перейдите по ссылке: '.'
http://localhost/Titan_final.loc/account/confirm/'.$token);
    }

    public function checkData($login, $password) {
        $params = [
            'login' => $login,
        ];
        $hash = $this->db->column('SELECT password FROM accounts WHERE
login = :login', $params);
        if (!$hash or !password_verify($password, $hash)) {
            return false;
        }
        return true;
    }

    public function checkStatus($type, $data) {
        $params = [
            $type => $data,
        ];
        $status = $this->db->column('SELECT status FROM accounts WHERE
'.$type.' = :'.$type, $params);
        if ($status != 1) {
            $this->error = 'Аккаунт ожидает подтверждения по E-mail';
            return false;
        }
        return true;
    }

    public function login($login) {
        $params = [
            'login' => $login,
        ];
        $data = $this->db->row('SELECT * FROM accounts WHERE login =
:login', $params);
        $_SESSION['account'] = $data[0];
    }

    public function recovery($post) {
        $token = $this->createToken();
        $params = [
            'email' => $post['email'],
            'token' => $token,
        ];
        $this->db->query('UPDATE accounts SET token = :token WHERE email
= :email', $params);
        mail($post['email'], 'Спортклуб Titan - Восстановление пароля',
'Для восстановления пароля перейдите по ссылке: '.'
http://localhost/Titan_final.loc/account/reset/'.$token);
    }

    public function save($post) {

```

```

        $file =
"/Titan_final.loc/public/images/" . $_FILES['path']['name'];
        move_uploaded_file($_FILES['path']['tmp_name'], $file);
        $params = [
            'id' => $_SESSION['account']['id'],
            'email' => $post['email'],
            'img' => $file,
            'tel' => $post['tel'],
        ];
        if (!empty($post['password'])) {
            $params['password'] = password_hash($post['password'],
PASSWORD_BCRYPT);
            $sql = ',password = :password';
        } else {
            $sql = '';
        }
        foreach ($params as $key => $val) {
            $_SESSION['account'][$key] = $val;
        }
        $this->db->query('UPDATE accounts SET email = :email, img = :img,
tel = :tel' . $sql . ' WHERE id = :id', $params);
        return true;
    }

    public function reset($token) {
        $new_password = $this->createToken();
        $params = [
            'token' => $token,
            'password' => password_hash($new_password,
PASSWORD_BCRYPT),
        ];
        $this->db->query('UPDATE accounts SET status = 1, token = "",
password = :password WHERE token = :token', $params);
        return $new_password;
    }

    public function getSchedule() {
        return $this->db->row('SELECT * from schedule');
    }

    public function getNotifications() {
        return $this->db->row('SELECT * from notifications order by id
desc');
    }
}

```

Додаток В

У даному додатку приведено код моделі Admin.

```
<?php
namespace application\models;
use application\core\Model;
class Admin extends Model {
    public function loginValidate($post) {
        $config = require 'application/config/admin.php';
        if ($config['login'] != $post['login'] or $config['password'] !=
$post['password']) {
            $this->error = 'Логин или пароль указан неверно';
            return false;
        }
        return true;
    }
    public function reviewsCount() {
        return $this->db->column('SELECT COUNT(id) FROM reviews');
    }

    public function checkClientTariff($id) {
        $params = [
            'id' => $id,
        ];
        return $this->db->column('SELECT id FROM accounts WHERE id =
:id', $params);
    }
    public function checkClientVisit($id) {
        $params = [
            'id' => $id,
        ];
        return $this->db->column('SELECT id FROM tariffs1 WHERE id =
:id', $params);
    }
    public function checkAddClientTariff($post) {
        $id_account = $post['uidAdd'];
        $atid = $post['atidAdd'];
        $sdid = $post['sdidAdd'];
        $params = [
            'id' => $id_account,
        ];
        $r1 = $this->db->column('SELECT id FROM accounts WHERE id = :id',
$params);
        $params = [
            'id' => $atid,
        ];
        $r2 = $this->db->column('SELECT id FROM abonement_types WHERE id
= :id', $params);
        $params = [
            'id' => $sdid,
        ];
        $r3 = $this->db->column('SELECT id FROM sports_directions WHERE
id = :id', $params);
        if (!$r1 || !$r2 || !$r3) {
            return false;
        } else {
            return true;
        }
    }
}
```

```

}
public function reviewsList($route) {
    $max = 10;
    $params = [
        'max' => $max,
        'start' => (((($route['page'] ?? 1) - 1) * $max),
    ];
    return $this->db->row('SELECT reviews.id,
reviews.comment, reviews.sTime, accounts.login, reviews.status FROM reviews
inner join accounts on(reviews.uid = accounts.id) ORDER BY reviews.id DESC
LIMIT :start, :max', $params);
}
public function reviewComplete($id) {
    $params = [
        'id' => $id,
    ];
    $this->db->query('UPDATE reviews SET status = 1 WHERE id = :id',
$params);
    return true;
}
public function reviewComplete2($id) {
    $params = [
        'id' => $id,
    ];
    $this->db->query('UPDATE reviews SET status = 0 WHERE id = :id',
$params);
    return true;
}
public function updateContent($post) {
    $file =
"/Titan_final.loc/public/images/" . $_FILES['path']['name'];
move_uploaded_file($_FILES['path']['tmp_name'], $file);
    $params = [
        'id' => $post['id'],
        'title' => htmlspecialchars($post['title']),
        'description' => htmlspecialchars($post['description']),
        'img' => $file,
    ];
    $this->db->query('UPDATE sports_directions SET title = :title,
img = :img, description = :description WHERE id = :id', $params);
    return true;
}
public function deleteContent($id) {
    $params = [
        'id' => $id,
    ];
    $this->db->query('UPDATE sports_directions set status = 0 WHERE
id = :id', $params);
    return true;
}
public function deleteContentBack($id) {
    $params = [
        'id' => $id,
    ];
    $this->db->query('UPDATE sports_directions set status = 1 WHERE
id = :id', $params);
    return true;
}
public function addContent($post) {
    $file =
"/Titan_final.loc/public/images/" . $_FILES['path']['name'];
move_uploaded_file($_FILES['path']['tmp_name'], $file);

```

```

$params = [
    'title' => htmlspecialchars($post['title']),
    'description' => htmlspecialchars($post['description']),
    'img' => $file,
];
$this->db->query('insert into sports_directions(title,
description,img) values(:title, :description, :img)', $params);
return true;
}
public function checkTitleDir($post) {
    $par = [
        'title' => htmlspecialchars($post['title']),
    ];
    return $this->db->column('SELECT title from sports_directions
where title = :title',$par);
}
public function checkTitleAb($post) {
    $par = [
        'title' => htmlspecialchars($post['title']),
    ];
    return $this->db->column('SELECT title from abonements where
title = :title',$par);
}
public function deleteNotification($id) {
    $params = [
        'id' => $id,
    ];
    $this->db->query('DELETE from notifications WHERE id = :id',
$params);
    return true;
}
public function addNotification($post) {
    $title = htmlspecialchars($post['title']);
    $description = htmlspecialchars($post['description']);
    $date_s = $post['date_s'];
    $date_e = $post['date_e'];
    $params = [
        'title' => htmlspecialchars($post['title']),
        'description' => htmlspecialchars($post['description']),
        'date_s' => $post['date_s'],
        'date_e' => $post['date_e'],
    ];
    $this->db->query('insert into notifications(title,
description,date_s,date_e) values(:title, :description, :date_s,:date_e)',
$params);
    if((date("Y-m-d") >= $date_s) && (date("Y-m-d") <= $date_e)) {
        $result = $this->db->row('SELECT * FROM accounts');
        if (!empty($result)) {
            foreach ($result as $key => $val) {
                mail($val['email'], 'Объявление: '.$title,
                $description);
            }
        }
    }
    return true;
}
public function updateContact($post) {
    $params = [
        'location' => htmlspecialchars($post['location']),
        'tel' => htmlspecialchars($post['tel']),
        'email' => $post['email'],
        'fb' => $post['fb'],
    ];
}

```

```

        'insta' => $post['insta'],
        'address' => $post['address'],
    ];
    $this->db->query('UPDATE contact_info set location = :location,
tel = :tel, email= :email,fb=:fb,insta=:insta,address=:address', $params);
    return true;
}

public function updateSchedule($post) {
    $params = [
        'daysc_w' => $post['daysc_w'],
        'time_s' => $post['time_s'],
        'time_e' => $post['time_e'],
    ];
    $this->db->query('UPDATE schedule set time_s = :time_s, time_e=
:time_e WHERE daysc_w = :daysc_w', $params);
    return true;
}

public function addTariffCllient($post) {
    $id_account = $post['uidAdd'];
    $atid = $post['atidAdd'];
    $sdid = $post['sdidAdd'];
    $discont = $post['discont'];
    $params = [
        'id_account' => $id_account,
    ];
    if($discont == "1") {
        $refBalanceMy = $this->db->column('SELECT refBalance FROM
accounts WHERE id = :id_account', $params);
    } else {
        $refBalanceMy = 0;
    }
    $dataRef = $this->db->column('SELECT ref FROM accounts WHERE id =
:id_account', $params);
    $params = [
        'atid' => $atid,
    ];
    $price = $this->db->column('SELECT price FROM abonement_types
WHERE id = :atid', $params);
    $description = $this->db->column('SELECT description FROM
abonement_types WHERE id = :atid', $params);
    $amount = $this->db->column('SELECT amount FROM abonement_types
WHERE id = :atid', $params);

    if ($dataRef != 0) {
        $refSum = round(($price * 5) / 100, 0);
        $params = [
            'sum' => $refSum,
            'id' => $dataRef,
        ];
    };
    $this->db->query('UPDATE accounts SET refBalance =
refBalance + :sum, refBalanceStatus = 0 WHERE id = :id', $params);
}

$amount_days_month = $amount * 31; //число дней (31 в месяце)
$razn = $price - $refBalanceMy;
if($razn < 0) {
    $razn = 0;
}
$params = [
    'id_account' => $id_account,

```

```

        'atid' => $atid,
        'id_sd' => $sdid,
    ];
    $amount_copy = $this->db->column('SELECT COUNT(id) FROM tariffs1
where id_at = :atid and uid = :id_account and id_sd = :id_sd', $params);
    if($amount_copy > 0){
        $status = 0;
    } else {
        $status = 1;
    }
    if ($description == 'занятий') {
        $params = [
            'date_s' => time(),
            'id_account' => $id_account,
            'id_sd' => $sdid,
            'id_at' => $atid,
            'summ' => $razn,
            'discont' => $refBalanceMy,
            'isLeft' => $amount,
            'status' => $status,
        ];
        $this->db->query('INSERT INTO
tariffs1(date_s,uid,id_sd,id_at,summ,discont,isLeft,date_f,status) VALUES
(:date_s,:id_account,:id_sd,:id_at,:summ,:discont,:isLeft,NULL,:status)',
$params);
    } elseif ($description == 'месяцев') {
        $params = [
            'date_s' => time(),
            'id_account' => $id_account,
            'id_sd' => $sdid,
            'id_at' => $atid,
            'summ' => $razn,
            'discont' => $refBalanceMy,
            'date_f' => strtotime('+ '.$amount_days_month.'
days'),
            'status' => $status,
        ];
        $this->db->query('INSERT INTO
tariffs1(date_s,uid,id_sd,id_at,summ,discont,isLeft,date_f,status) VALUES
(:date_s,:id_account,:id_sd,:id_at,:summ,:discont,NULL,:date_f,:status)',
$params);
    }
    $params = [
        'id_account' => $id_account,
    ];
    $this->db->query('UPDATE accounts SET refBalance = 0,
refBalanceStatus = 0 WHERE id = :id_account', $params);

    return true;
}

public function getTariffs() {
    return $this->db->row('SELECT * FROM tariffs1');
}
public function addAbonement($post) {
    $params = [
        'title' => htmlspecialchars($post['title']),
    ];
    $this->db->query('insert into abonements(title) values(:title)',
$params);
    return true;
}
}

```

```

public function deleteAbonement($id) {
    $params = [
        'id' => $id,
    ];
    $this->db->query('UPDATE abonements set status = 0 WHERE id = :id',
$params);
    return true;
}
public function deleteAbonementBack($id) {
    $params = [
        'id' => $id,
    ];
    $this->db->query('UPDATE abonements set status = 1 WHERE id = :id',
$params);
    return true;
}
public function updateAbonement($post) {
    $params = [
        'title' => htmlspecialchars($post['title']),
        'new_title' => htmlspecialchars($post['new_title']),
    ];
    $this->db->query('UPDATE abonements set title = :new_title WHERE title
= :title', $params);
    return true;
}
public function addAbonementType($post) {
    $params = [
        'aid' => $post['aid'],
        'amount' => $post['amount'],
        'description' => $post['description'],
        'edditional' => htmlspecialchars($post['edditional']),
        'price' => $post['price'],
    ];
    $this->db->query('insert into
abonement_types (aid,amount,description,edditional,price)
values (:aid,:amount,:description,:edditional,:price)', $params);
    return true;
}
public function updateAbonementType($post) {
    $params = [
        'id' => $post['id-type'],
        'price' => $post['price'],
    ];
    $this->db->query('UPDATE abonement_types set price = :price WHERE id =
:id', $params);
    return true;
}
public function response($post) {
    $id = $post['id'];
    $email = $post['email'];
    $title = 'Ответ на Ваше сообщение';
    $message = $post['message'];
    $params = [
        'id' => $post['id'],
    ];
    $this->db->query('UPDATE feedback set status = 1 WHERE id = :id',
$params);
    $result = $this->db->row('SELECT * FROM accounts');
    mail($email, $title, 'Спасибо за интерес к нашему спортклубу.
'.$message);
    return true;
}
}

```



```

public function deleteAbonementType($id) {
$params = [
    'id' => $id,
];
$this->db->query('UPDATE abonement_types set status = 0 WHERE id = :id',
$params);
return true;
}
public function deleteAbonementTypeBack($id) {
$params = [
    'id' => $id,
];
$this->db->query('UPDATE abonement_types set status = 1 WHERE id = :id',
$params);
return true;
}
    public function getAbonements() {
        return $this->db->row('SELECT * from abonements order by id
desc');
    }
    public function getClients() {
        return $this->db->row('SELECT * from accounts');
    }
    public function getTariffClients() {
        return $this->db->row('SELECT distinct uid, login from tariffs1 t
inner join accounts a on(t.uid=a.id)');
    }
    public function getTariffSportAll() {
        return $this->db->row('SELECT distinct id_sd, title from tariffs1
t inner join sports_directions s on(t.id_sd=s.id)');
    }
    public function getAbonementsTypes() {
        return $this->db->row('SELECT * from abonement_types');
    }
    public function directionsList() {
        return $this->db->row('SELECT * from sports_directions order by
id desc');
    }
    public function contactInfoList() {
        return $this->db->row('SELECT * from contact_info');
    }
    public function getSchedule() {
        return $this->db->row('SELECT * from schedule');
    }
    public function getNotifications() {
        return $this->db->row('SELECT * from notifications order by id
desc');
    }
    public function getFeedback() {
        return $this->db->row('SELECT * from feedback where name is not
null order by status asc,id desc');
    }
    public function getRecords() {
        return $this->db->row('SELECT * from feedback where tel is not
null order by id desc');
    }
    public function getReport() {
        return $this->db->row('SELECT t.id_sd, s.title, sum(t.summ) as
sumt from tariffs1 t inner join sports_directions s on(t.id_sd=s.id) group by
t.id_sd');
    }
    public function tariffsCount() {

```

```

        return $this->db->column('SELECT COUNT(id) FROM tariffs1');
    }
    public function tariffsCount2($login) {
        $params = [
            'login' => $login,
        ];
        return $this->db->column('SELECT COUNT(id) FROM tariffs1 where
login = :login', $params);
    }
    public function tariffsList() {
        $result = $this->db->row('SELECT * FROM tariffs1 ORDER BY id
DESC');
        return $result;
    }
    public function tariffsList2($route) {
        $params = [
            'uid' => $route['page'],
        ];
        $result = $this->db->row('SELECT * FROM tariffs1 where uid
= :uid ORDER BY id DESC', $params);
        return $result;
    }

    public function visitsList() {
        $result = $this->db->row('SELECT * FROM training_history ORDER BY
id DESC');
        return $result;
    }
    public function visitsList2($route) {
        $params = [
            'tid' => $route['page'],
        ];
        $result = $this->db->row('SELECT * FROM training_history
where tid = :tid ORDER BY id DESC', $params);
        return $result;
    }

    public function addNewVisit($post) {
        $tid = $post['tidAdd'];
        $params = [
            'tid' => $tid,
        ];
        //---проверка если уже оплачено сегодня данный абонемент
        $isExist = $this->db->column('SELECT count(*) from
training_history where tid = :tid and from_unixtime(date_n,"%d") =
DAYOFMONTH(now()) and from_unixtime(date_n,"%m") = MONTH(now()) and
from_unixtime(date_n,"%Y") = YEAR(now())', $params);
        if($isExist > 0) {
            return false;
        }
        //---
        $isLeft = $this->db->column('SELECT isLeft from tariffs1
where id = :tid', $params);
        $uid = $this->db->column('SELECT uid from tariffs1 where id
= :tid', $params);
        $atid = $this->db->column('SELECT id_at from tariffs1 where
id = :tid', $params);
        $sdid = $this->db->column('SELECT id_sd from tariffs1 where
id = :tid', $params);
        $date_f = $this->db->column('SELECT date_f from tariffs1
where id = :tid', $params);
        if($isLeft){
            if($isLeft == 1) {

```

```

        $this->db->query('UPDATE tariffs1 set isLeft =
isLeft - 1, status = 0 where id = :tid', $params);
        $params = [
            'tid' => $tid,
            'uid' => $uid,
            'id_sd' => $sdid,
            'id_at' => $atid,
        ];
        $tidNew = $this->db->column('SELECT id from
tariffs1 where id > :tid and uid = :uid and id_sd = :id_sd and id_at = :id_at
order by id limit 1', $params);
        $params = [
            'tidNew' => $tidNew,
        ];
        $this->db->query('UPDATE tariffs1 set status =
1 where id = :tidNew', $params);
        $params = [
            'tid' => $tid,
            'date_n' => time(),
        ];
        $this->db->query('INSERT INTO training_history
(date_n,tid) VALUES (:date_n,:tid)', $params);
    } else {
        $this->db->query('UPDATE tariffs1 set isLeft =
isLeft - 1 where id = :tid', $params);
        $params = [
            'tid' => $tid,
            'date_n' => time(),
        ];
        $this->db->query('INSERT INTO training_history
(date_n,tid) VALUES (:date_n,:tid)', $params);
    }
} else {
    if(time() < $date_f) {
        $params = [
            'tid' => $tid,
            'date_n' => time(),
        ];
        $this->db->query('INSERT INTO training_history
(date_n,tid) VALUES (:date_n,:tid)', $params);
    } elseif(time() == $date_f) {
        $params = [
            'tid' => $tid,
            'date_n' => time(),
        ];
        $this->db->query('INSERT INTO training_history
(date_n,tid) VALUES (:date_n,:tid)', $params);
    }
    $params = [
        'tid' => $tid,
    ];
    $this->db->query('UPDATE tariffs1 set status =
0 where id = :tid', $params);
    $params = [
        'tid' => $tid,
        'uid' => $uid,
        'id_sd' => $sdid,
        'id_at' => $atid,
    ];
    $tidNew = $this->db->column('SELECT id from
tariffs1 where id > :tid and uid = :uid and id_sd = :id_sd and id_at = :id_at
order by id limit 1', $params);
    $params = [

```


Додаток Г

Сторінка панелі управління контентом.

```

<div class="content-wrapper">
  <div class="container-fluid">
    <div class="card mb-3">
      <div class="card-header"><b>Здесь Вы можете изменить информацию
сайта спортклуба</b></div>
      <div class="card-body">
        <div class="row">
          <div class="col-sm-12">
            <ul class="nav nav-tabs">
              <li><a data-toggle="tab" class="nav-link active"
href="#directions">Направления</a></li>
              <li><a data-toggle="tab" class="nav-link"
href="#abonements">Абонементы</a></li>
              <li><a data-toggle="tab" class="nav-link" href="#contact-
info">Контактная информация</a></li>
              <li><a data-toggle="tab" class="nav-link" href="#schedule">График
работы</a></li>
            </ul>

            <div class="tab-content">
              <div id="directions" class="tab-pane fade show active">
                <br>
                <div class="card-header">Добавьте направление: </div>
                <form action="/Titan_final.loc/admin/content" method="post"
enctype="multipart/form-data">
                  <table class="table table-bordered table-responsive">
                    <thead>
                      <tr>
                        <th>Название</th>
                        <th>Описание</th>
                        <th>Изображение</th>
                        <th></th>
                      </tr>
                    </thead>
                    <tbody>
                      <tr>
                        <td><input type="text" name="title" style="outline:
none;border: none;border-bottom:1px solid grey;" required></td>
                        <td><textarea name="description" style="outline:
none;border: none;border-bottom:1px solid grey;" required></textarea>
                        <td><input type="file" class="form-control-file"
name="path" required></td>
                        <td><input type="hidden" name="type" value="create">
                        <td><button style="cursor:pointer;" type="submit"
class="btn btn-primary"><i class="fa fa-save" aria-hidden="true"></i>
Добавить</button></td>
                      </tr>
                    </tbody>
                  </table>
                </form>
                <div class="card-header">Управление направлениями: </div>
                <table class="table table-bordered table-responsive">
                  <thead>
                    <tr>
                      <th>Название</th>
                      <th>Описание</th>

```

```

        <th>Изображение</th>
        <th></th>
    </tr>
</thead>
<tbody>
<?php foreach ($sports_directions as $val): ?>
    <tr>
        <td><?php echo $val['title']; ?></td>
        <td><?php echo $val['description']; ?></td>
        <td>
            
        <td>
            <form action="/Titan_final.loc/admin/content"
method="post">
                <input type="hidden" name="id" value="<?php echo
$val['id']; ?>">
                <input type="hidden" name="type" value="delete">
                <?php if($val['status'] == 1): ?>
                <button style="cursor:pointer;" type="submit"
class="btn btn-outline-success"><i class="fa fa-trash" aria-
hidden="true"></i> Удалить</button>
                </form>
                <form action="/Titan_final.loc/admin/content"
method="post">
                    <? else: ?>
                    <input type="hidden" name="id" value="<?php echo
$val['id']; ?>">
                    <input type="hidden" name="type"
value="delete_sport_back">
                    <button style="cursor:pointer;" type="submit"
class="btn btn-outline-secondary"><i class="fa fa-save" aria-
hidden="true"></i> Восстановить</button>
                    <? endif; ?>
                </form>
            </td>
        </tr>
    <?php endforeach; ?>
</tbody>
</table>
</div>
<div id="abonements" class="tab-pane fade">
<br>
<div class="row">
    <div class="col-md-4">
        <a style="width:100%;margin-bottom:10px;" href="#spoiler-
1" data-toggle="collapse" class=" btn btn-outline-primary collapsed">
            Управление абонеменстами
        </a>
    </div>
    <div class="col-md-4">
        <a style="width:100%;" href="#spoiler-2" data-
toggle="collapse" class=" btn btn-outline-success collapsed">
            Управление типами абонементов
        </a>
    </div>
</div>
<div class="collapse" id="spoiler-1">
    <div class="card card-body border-primary">
        <a href="#spoiler-3" data-toggle="collapse" class=" btn
btn-link collapsed">

```

```

        Добавить абонемент
    </a>
    <div class="collapse" id="spoiler-3">
    <form action="/Titan_final.loc/admin/content"
method="post">
        <table class="table table-bordered">
        <thead>
        <tr>
        <th>Название</th>
        <th></th>
        </tr>
        </thead>
        <tbody>
        <tr>
            <td><input type="text" name="title"
style="outline: none;border: none;border-bottom:1px solid grey;"
required></td>
            <input type="hidden" name="type" value="create-
abonement">
            <td><button style="cursor:pointer;"type="submit"
class="btn btn-primary"><i class="fa fa-save" aria-hidden="true"></i>
Добавить</button></td>
        </tr>
        </tbody>
        </table>
        </form>
    </div>
    <a href="#spoiler-4" data-toggle="collapse" class=" btn
btn-link collapsed">
        Изменить абонемент
    </a>
    <div class="collapse" id="spoiler-4">
    <form action="/Titan_final.loc/admin/content"
method="post">
        <div class="form-group">
        <label for="daysc_w">Выберите абонемент:</label>
        <select class="form-control" name="title">
        <?php foreach ($abonements as $val): ?>
        <?if($val['status'] == 1) :?>
        <option value="<?php echo $val['title'];
?>"><?php echo $val['title']; ?></option>
        <?php endif; ?>
        <?php endforeach; ?>
        </select>
        </div>
        <div class="form-group">
        <label for="new_title">Новое название: </label>
        <input style="outline:none;border:none;border-
bottom:1px solid grey;" type="text" name="new_title" required/>
        </div>
        <input type="hidden" name="type" value="update-
abonement">
        <button style="cursor:pointer;"type="submit" class="btn
btn-primary">Сохранить</button>
        </form>
    </div>
    <a href="#spoiler-5" data-toggle="collapse" class=" btn
btn-link collapsed">
        Удалить/Восстановить<span style="color:red">*</span>
        абонемент
    </a>
    <div class="collapse" id="spoiler-5">

```

```

        <h6 style="color:red">* Вы можете удалить
(активировать/деактивировать) абонемент </h6>
        <table class="table table-bordered">
        <thead>
        <tr>
        <th>Название</th>
        <th></th>
        </tr>
        </thead>
        <tbody>
        <?php foreach ($abonements as $val): ?>
        <tr>
        <td><?php echo $val['title']; ?></td>
        <td>
        <form action="/Titan_final.loc/admin/content"
method="post">
                <input type="hidden" name="id" value="<?php
echo $val['id']; ?>">
                <input type="hidden" name="type"
value="delete-abonement">
                <?php if($val['status'] == 1): ?>
                <button
style="cursor:pointer;"type="submit" class="btn btn-outline-primary"><i
class="fa fa-trash" aria-hidden="true"></i> Удалить</button>
                </form>
                <form action="/Titan_final.loc/admin/content"
method="post">
                <? else: ?>
                <input type="hidden" name="id"
value="<?php echo $val['id']; ?>">
                <input type="hidden" name="type"
value="delete-abonement-back">
                <button
style="cursor:pointer;"type="submit" class="btn btn-outline-secondary"><i
class="fa fa-save" aria-hidden="true"></i> Восстановить</button>
                <? endif; ?>
                </form>
        </td>
        </tr>
        <?php endforeach; ?>
        </tbody>
        </table>
        </div>
        </div>
        <div class="collapse" id="spoiler-2">
        <div class="card card-body border-success">
        <a style="color:green;" href="#spoiler-6" data-
toggle="collapse" class="btn btn-link collapsed">
        Добавить тип абонемента
        </a>
        <div class="collapse" id="spoiler-6">
        <form action="/Titan_final.loc/admin/content"
method="post">
        <table class="table table-bordered table-responsive">
        <thead>
        <tr>
        <th>Название абонемента</th>
        <th>Количество</th>
        <th>Тип</th>
        <th>Время</th>
        <th>Стоимость</th>

```



```

        <th></th>
    </tr>
</thead>
<tbody>
    <tr>
        <td>
            <select class="form-control" name="aid">
                <?php foreach ($abonements as $val): ?>
                    <!--?php foreach ($abonement_types as $val2):
?-->
                        <option value="<?php echo $val['id'];
?>"><?php echo $val['title']; ?></option>
                            <!--?php endforeach; ?-->
                                <?php endforeach; ?>
                                    </select>
                                        </td>
                                            <td><input type="number" name="amount"
style="outline: none;border: none;border-bottom:1px solid grey;"
required></td>
                                                <td>
                                                    <select class="form-control" name="description"
required>
                                                        <option value="занятий">занятий</option>
                                                            <option value="месяцев">месяцев</option>
                                                                </select>
                                                                    </td>
                                                                        <td><input type="text" name="edditional"
style="outline: none;border: none;border-bottom:1px solid grey;"
required></td>
                                                                            <td><input type="number" name="price"
style="outline: none;border: none;border-bottom:1px solid grey;"
required></td>
                                                                                <input type="hidden" name="type" value="create-
abonement-type">
                                                                                    <td><button style="cursor:pointer;" type="submit"
class="btn btn-success"><i class="fa fa-save" aria-hidden="true"></i>
Добавить</button></td>
                                                                                        </tr>
                                                                                            </tbody>
                                                                                                </table>
                                                                                                    </form>
                                                                                                        </div>
                                                                                                            <a style="color:green;" href="#spoiler-7" data-
toggle="collapse" class=" btn btn-link collapsed">
Изменить тип абонеента
</a>
<div class="collapse" id="spoiler-7">
    <table class="table table-bordered table-responsive">
        <thead>
            <tr>
                <th>Название абонеента</th>
                <th>Количество</th>
                <th>Тип</th>
                <th>Время</th>
                <th>Стоимость</th>
                <th></th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($abonements as $val): ?>
                <?php foreach ($abonement_types as $val2): ?>

```

```

                <?if($val2['status'] == 1 && ($val['id'] ==
$val2['aid'])) && ($val['status'] == 1):?>
                <tr>
                    <td><?php echo $val['title']; ?></td>
                    <td><?php echo $val2['amount'];?></td>
                    <td><?php echo $val2['description'];?></td>
                    <td><?php echo $val2['edditional'];?></td>
                    <form action="/Titan_final.loc/admin/content"
method="post">
                        <td><input type="number" value="<?php echo
$val2['price'];?>" name="price" style="outline: none;border: none;border-
bottom:1px solid grey;" required></td>
                        <input type="hidden" name="id-type" value="<?php
echo $val2['id'];?>">
                        <input type="hidden" name="type" value="update-
abonement-type">
                        <td><button style="cursor:pointer;" type="submit"
class="btn btn-success"><i class="fa fa-save" aria-hidden="true"></i>
Сохранить</button></td>
                    </form>
                </tr>
                <?php endif; ?>
            <?php endforeach; ?>
        <?php endforeach; ?>
    </tbody>
</table>
</div>
<a style="color:green;" href="#spoiler-8" data-
toggle="collapse" class=" btn btn-link collapsed">
    Удалить/Восстановить<span style="color:red">*</span> тип
абонемента
</a>
<div class="collapse" id="spoiler-8">
    <h6 style="color:red">*</h6> Вы можете удалить
(активировать/деактивировать) тип абонемента </h6>
    <table class="table table-bordered table-responsive">
        <thead>
            <tr>
                <th>Название абонемента</th>
                <th>Количество</th>
                <th>Тип</th>
                <th>Время</th>
                <th>Стоимость</th>
                <th></th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($abonements as $val): ?>
                <?php foreach ($abonement_types as $val2): ?>
                    <?if( $val['id'] == $val2['aid'] ):?>
                        <tr>
                            <td><?php echo $val['title']; ?></td>
                            <td><?php echo $val2['amount'];?></td>
                            <td><?php echo $val2['description'];?></td>
                            <td><?php echo $val2['edditional'];?></td>
                            <td><?php echo $val2['price'];?></td>
                            <td>
                                <form action="/Titan_final.loc/admin/content"
method="post">
                                    <input type="hidden" name="id" value="<?php
echo $val2['id']; ?>">

```

```

                                <input type="hidden" name="type"
value="delete-abonement-type">
                                <?php if($val2['status'] == 1): ?>
                                    <button
style="cursor:pointer;"type="submit" class="btn btn-outline-success"><i
class="fa fa-trash" aria-hidden="true"></i> Удалить</button>
                                </form>
                                <form action="/Titan_final.loc/admin/content"
method="post">
                                    <? else: ?>
                                        <input type="hidden" name="id" value="<?php
echo $val2['id']; ?>">
                                        <input type="hidden" name="type"
value="delete-abonement-type-back">
                                        <button
style="cursor:pointer;"type="submit" class="btn btn-outline-secondary"><i
class="fa fa-save" aria-hidden="true"></i> Восстановить</button>
                                        <? endif; ?>
                                    </form>
                                </td>
                            </tr>
                        <?php endif; ?>
                    <?php endforeach; ?>
                <?php endforeach; ?>
            </tbody>
        </table>
    </div>
</div>
</div>
<div id="contact-info" class="tab-pane fade">
    <form action="/Titan_final.loc/admin/content" method="post">
        <?php foreach ($contact_info as $val): ?>
            <br>
            <h5 class="section-title">Контактная информация:</h5>
            <input type="text" name="location" value="<?php echo
$val['location']; ?>" required>
            <input type="text" name="tel" value="<?php echo
$val['tel']; ?>" required>
            <input type="text" name="email" value="<?php echo
$val['email']; ?>" required>
            <h5 class="section-title">Мы в соцсетях:</h5>
            <input type="text" name="fb" value="<?php echo
$val['fb']; ?>">
            <input type="text" name="insta" value="<?php echo
$val['insta']; ?>">
            <h5 class="section-title">Где нас найти:</h5>
            <p><input type="text" name="address" value="<?php echo
$val['address']; ?>" required</p>
            <iframe src="<?php echo $val['address']; ?>"
style="border:0; width:600px; height:300px;" allowfullscreen=""
loading="lazy"></iframe>
            <?php endforeach; ?>
            <input type="hidden" name="type" value="update-contact"><br>
            <button style="cursor:pointer;"type="submit" class="btn btn-
success">Сохранить</button>
        </form>
    </div>

<div id="schedule" class="tab-pane fade">
<br>

```


Додаток Д

Сторінка відображення панелі управління зворотним зв'язком.

```

<div class="content-wrapper">
  <div class="container-fluid">
    <div class="card mb-3">
      <div class="card-body">
        <div class="row">
          <div class="col-sm-12">
            <ul class="nav nav-tabs">
              <li><a data-toggle="tab" class="nav-link active"
href="#directions">Обратная связь</a></li>
              <li><a data-toggle="tab" class="nav-link"
href="#abonements">Запись на тренировку</a></li>
            </ul>
            <div class="tab-content">
              <div id="directions" class="tab-pane fade show active">
                <br>
                <?php foreach ($feedback as $val3): ?>
                  <div class="card-header card-body border-primary">
                    <a style="margin-left:17px;" href="#spoiler-<?php echo
$val3['id']; ?>" data-toggle="collapse" class=" btn btn-link collapsed">
                      <table class="table">
                        <tr>
                          <?if($val3['status'] == 1) :?>
                            <td><div style="width:80px;color:grey;"><?php
echo $val3['name']; ?></div></td>
                            <td><div style="width:300px;color:grey;"><?php
echo date('d.m.Y в H:i', $val3['date_s']);?></div></td>
                          <?else:?>
                            <td><div style="width:80px;"><?php echo
$val3['name']; ?></div></td>
                            <td><div style="width:300px;"><?php echo
date('d.m.Y в H:i', $val3['date_s']);?></div></td>
                          <?endif;?>
                        </tr>
                      </table>
                    </a>
                    <div class="collapse" id="spoiler-<?php echo
$val3['id']; ?>">
                      <div class="container">
                        <div class="col-md-6">
                          <form action="/Titan_final.loc/admin/feedback"
method="post">
                            <p style="color:blue;"><?php echo
$val3['description']; ?></p>
                            <p>Сообщение будет отправлено на почту <u><?php
echo $val3['email']; ?></u></p>
                            <textarea name="message" class="form-control"
id="message" rows="7" placeholder="Сообщение" required></textarea>
                            <input type="hidden" name="type"
value="response">
                            <input type="hidden" name="id" value="<?php
echo $val3['id']; ?>">
                            <input type="hidden" name="email" value="<?php
echo $val3['email']; ?>">
                            <button style="cursor:pointer;margin-
top:10px;" type="submit" class="btn btn-primary"><i class="fa fa-send" aria-
hidden="true"></i> Ответить</button>

```

```

        </form>
    </div>
</div>
</div>
</div>
</div>
<?endforeach;?>
</div>
<div id="abonements" class="tab-pane fade">
<br>
    <table class="table table-responsive">
        <thead>
            <tr>
                <th>Дата</th>
                <th>Телефон</th>
                <th>Запись</th>
                <th></th>
                <th></th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($record as $val): ?>
                <tr>
                    <td><?php echo date('d.m.Y в H:i',$val['date_s']);
?></td>
                    <td><?php echo $val['tel']; ?></td>
                    <?if((empty($val['date_record'])) &&
(empty($val['time_record']))) :?>
                        <form action="/Titan_final.loc/admin/feedback"
method="post">
                            <td>
                                <input style="outline:none;border:none;"
type="date" name="date_record" required/>
                                <input style="outline:none;border:none;"
type="time" name="time_record" required/>
                            </td>
                            <td>
                                <input type="hidden" name="type" value="update-
rec">
                                <input type="hidden" name="idRec" value="<?php
echo $val['id']; ?>">
                                <button style="cursor:pointer;"type="submit"
class="btn btn-outline-success"><i class="fa fa-save" aria-hidden="true"></i>
Добавить</button>
                            </td>
                        </form>
                        <form action="/Titan_final.loc/admin/feedback"
method="post">
                            <td>
                                <input type="hidden" name="type" value="delete-
rec">
                                <input type="hidden" name="idRec"
value="<?php echo $val['id']; ?>">
                                <button style="cursor:pointer;"type="submit"
class="btn btn-outline-danger"><i class="fa fa-trash" aria-
hidden="true"></i></button>
                            </td>
                        </form>
                    <?else :?>
                        <form action="/Titan_final.loc/admin/feedback"
method="post">
                            <td>

```

