

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет водного господарства та природокористування
Навчально-науковий інститут кібернетики, інформаційних технологій та інженерії
Кафедра комп'ютерних технологій та економічної кібернетики

Допущено до захисту:

Завідувач кафедри

_____ д. е. н., проф. П. М. Грицюк

« _____ » _____ 20__ р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня «бакалавр»

за освітньо-професійною програмою «Інформаційні системи та технології»
спеціальності 126 «Інформаційні системи та технології»

на тему: «Розробка та впровадження Веб-сервісу для автоматизації
управління відносинами з клієнтами спортзалу»

Виконала:

здобувач вищої освіти 4 курсу, групи ІСТ-41
Чаплик Олеся Андріївна

Керівник:

канд. техн. наук, доцент Барановський С. В.

Рецензент:

ст. викладач, Шевченко І. М.

Рівне - 2024

Національний університет водного господарства та природокористування
 ННІ кібернетики, інформаційних технологій та інженерії
 Кафедра комп'ютерних технологій та економічної кібернетики

Освітньо-кваліфікаційний рівень – *бакалавр*
 Освітньо-професійна програма «*Інформаційні системи і технології*»
 Спеціальність 126 «*Інформаційні системи і технології*»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ д. е. н., проф. П.М. Грицюк

« _____ » _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачу _____

Чаплик Олесі Андріївни

(прізвище, ім'я, по-батькові)

1. Тема роботи _____ *Розробка та впровадження Веб-сервісу для автоматизації управління відносинами з клієнтами спортзалу* _____

керівник роботи _____ *Барановський Сергій Віталійович, канд. техн. наук, доцент* _____
 (прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)

затверджена наказом по університету від “ _____ “ _____ 202__ року № _____

2. Термін здачі студентом закінченої роботи _____

3. Вихідні дані до роботи _____ *Важливою проблемою і на сьогодні залишається підвищення ефективності діяльності фітнес-закладів шляхом більш широкого впровадження сучасних інформаційних систем та технологій. Завданням кваліфікаційної роботи є дослідження та аналіз перспектив впровадження Веб-технологій та Веб-сервісів для інформаційної підтримки процесів управління діяльністю фітнес-закладів, а також створення на цій основі бази даних та інформаційної системи з інтегрованими Веб-сервісами для автоматизації окремих функцій обліку та управління діяльністю закладом* _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Вступ

1. *Сучасний стан та перспективи розвитку сфери надання послуг фітнес-закладів*

2. *Огляд та аналіз інформаційних технологій управління діяльністю фітнес-закладів*

3. *Особливості застосування Веб-технологій в бізнесі*

4. *Технологія інтегрування XML у Веб-сервіс*

5. *Організація роботи з базами даних ІТ проекту*

6. *Технології та засоби захисту інформації в ІТ проекті*

7. *Програмна реалізація ІТ проекту*

8. Опис функціональних можливостей та інтерфейсу ІС

Висновки

5. Перелік графічного матеріалу _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання отримав
1.	<i>Барановський С.В., доцент</i>		
2.	<i>Барановський С.В., доцент</i>		
3.	<i>Барановський С.В., доцент</i>		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Терміни виконання етапів роботи	Примітка
1.	<i>Аналіз інформаційних процесів планування та управління діяльністю фітнес-закладу</i>	<i>до 20.01.2024</i>	
2.	<i>Аналіз Веб-технологій інформаційної підтримки діяльності фітнес-закладів</i>	<i>до 20.02.2024</i>	
3.	<i>Особливості застосування та технології інтегрування XML у Веб-сервіси для управління діяльністю фітнес-закладу</i>	<i>до 20.03.2024</i>	
4.	<i>Моделювання інформаційних потоків та проектування бази даних</i>	<i>до 10.04.2024</i>	
5.	<i>Організація системи захисту інформації в ІТ проекті</i>	<i>до 25.04.2024</i>	
6.	<i>Програмна реалізація ІТ проекту</i>	<i>до 25.05.2024</i>	
7.	<i>Оформлення та представлення до захисту бакалаврської роботи</i>	<i>до 10.06.2024</i>	

Здобувач вищої освіти _____
(підпис)

_____ (прізвище та ініціали)

Керівник кваліфікаційної роботи _____
(підпис)

_____ (прізвище та ініціали)

АНОТАЦІЯ

Чаплик О.А.. Розробка та впровадження Веб-сервісу для автоматизації управління відносинами з клієнтами спортзалу. Кваліфікаційна робота на здобуття освітнього ступеня «бакалавр» : 92 с., рис.12, 2 табл., 5 додатків на 22 стор., 7 літературних джерел.

Метою даного проєкту є розробка та впровадження Веб-сервісу для автоматизації управління відносинами з клієнтами спортзалу, який допоможе підвищити ефективність роботи персоналу, покращити взаємодію з клієнтами, збільшити їх лояльність та, як наслідок, сприяти зростанню бізнесу.

Сайт розроблений в середовищі Visual Studio, з використанням сучасних технологій мови програмування, таких як C#, а також HTML, CSS, JavaScript. Веб-сервіс пропонує широкий спектр функціональних можливостей, спрямованих на покращення взаємодії з клієнтами та підвищення рівня їх задоволеності. Користувачі отримують зручний та інтуїтивно зрозумілий інтерфейс, який дозволяє легко знаходити необхідну інформацію про послуги спортзалу, розклад занять, тренерів. Крім того, сервіс надає можливість зворотнього зв'язку, дозволяючи клієнтам задавати питання та отримувати оперативну підтримку від персоналу спортзалу. Використання XML-файлу для зберігання бази даних користувачів забезпечує надійність та безпеку даних, а також дозволяє легко інтегрувати сервіс з іншими системами управління спортзалом. Завдяки цьому, клієнти можуть відстежувати свій поточний абонемент, термін його дії у своїх облікових записах.

ЗМІСТ

РОЗДІЛ 1. СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ СФЕРИ.....	9
НАДАННЯ ПОСЛУГ ФІТНЕС-ЗАКЛАДІВ	9
1.1. Аналіз сучасного стану та перспектив надання послуг фітнес-закладами	9
1.2. Огляд інформаційних технологій управління діяльністю фітнес-закладів.....	12
1.3. Веб-технології інформаційної підтримки діяльності фітнес-закладів	18
1.4. Перспективи впровадження XML для підвищення ефективності	24
діяльності фітнес-закладів.....	24
РОЗДІЛ 2. ІНТЕГРУВАННЯ XML У ВЕБ-СЕРВІСИ ДЛЯ УПРАВЛІННЯ ДІЯЛЬНІСЮ ФІТНЕС-ЗАКЛАДУ.....	27
2.1. Загальна характеристика та аналіз сучасного використання XML	27
2.2. Особливості застосування технологій XML у бізнесі.....	35
2.3. Технологія інтегрування XML у Веб-сервіс.....	39
РОЗДІЛ 3. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	46
3.1. Обґрунтування вибору засобів реалізації	46
3.2. Опис інтерфейсу та функціональних можливостей веб-сайту.....	55
3.3. Тестові приклади роботи системи, аналіз результатів.....	60
3.4. Рекомендації щодо впровадження та обґрунтування	64
ефективності.	64
ВИСНОВКИ.....	67
СПИСОК ЛІТЕРАТУРИ.....	69
ДОДАТКИ	70

ВСТУП

У сучасному світі інноваційних технологій та активного способу життя, спортивні заклади мають велике значення в підтримці фізичного здоров'я та психологічного благополуччя населення. Зростаюча свідомість про важливість активного способу життя підштовхує спортзали до пошуку ефективних рішень для поліпшення свого функціонування та підвищення якості обслуговування клієнтів.

Розробка та впровадження Веб-сервісу для автоматизації управління відносинами з клієнтами спортзалу є актуальним завданням, спрямованим на оптимізацію рутинних процесів, покращення комунікації та забезпечення персоналізованого досвіду для кожного клієнта. Цей веб-сервіс прагне стати ключовим інструментом для спортивних закладів, дозволяючи їм ефективно керувати відносинами з клієнтами, підвищуючи їхню лояльність та рівень задоволення від занять фітнесом. У даному дослідженні ми ретельно розглянемо об'єкт дослідження та виявимо ключові проблеми, що потребують вирішення через впровадження цього інноваційного Веб-сервісу.

У зазначеному контексті, де конкуренція на ринку фітнес-послуг стрімко зростає, важливо зазначити, що сучасні клієнти спортзалів не лише очікують високоякісних тренувань, але й вимагають ефективного та індивідуалізованого підходу в управлінні відносинами.

Процеси ручного обліку, недостатня комунікація та несистематизований розклад тренувань вже стали причиною нестабільності та невдоволення серед клієнтів. Застосування Веб-сервісу для автоматизації управління відносинами з клієнтами є стратегічною відповіддю на ці виклики, оскільки воно не лише оптимізує роботу адміністрації, а й створює нові можливості для вдосконалення відносин з клієнтами.

Метою цього дослідження є глибше розуміння суті проблем та визначення способів, які дозволять розробити та успішно впровадити Веб-сервіс для оптимізації управління відносинами з клієнтами спортзалу.

Завдання дослідження включають:

- Детальний аналіз існуючих проблем: Розгляд найважливіших аспектів, таких як нестабільний облік клієнтів, втрата даних, неефективна комунікація та складнощі при формуванні розкладу тренувань.
- Визначення вимог до Веб-сервісу: Встановлення ключових функціональностей та можливостей, які повинен надавати розроблений сервіс.
- Розробка концепції Веб-сервісу: Створення обґрунтованої концепції, яка враховує виявлені проблеми та потреби клієнтів спортзалу.
- Оцінка потенційних вигод: Визначення можливостей, які впровадження Веб-сервісу може принести для оптимізації управління та підвищення задоволеності клієнтів.

Для досягнення визначених завдань, використовуватимемо комбінацію кількох методологічних підходів. Перш за все, аналіз існуючих систем та сервісів, що використовуються в сфері фітнесу та управління спортзалами. Це дозволить визначити найкращі практики та потреби клієнтів, які необхідно врахувати при розробці Веб-сервісу.

Далі, провести опитування серед адміністраторів та клієнтів спортзалів з метою визначення конкретних проблем, із якими вони стикаються у щоденній роботі або під час занять. Це дозволить отримати важливий внутрішній вигляд на ситуацію та точно визначити пріоритети у розробці Веб-сервісу.

Це дослідження є кроком у напрямку розвитку інноваційних рішень для спортзалів, які бажають підтримувати високий стандарт обслуговування та відповідати потребам сучасного споживача.

РОЗДІЛ 1. СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ СФЕРИ НАДАННЯ ПОСЛУГ ФІТНЕС-ЗАКЛАДІВ

1.1. Аналіз сучасного стану та перспектив надання послуг фітнес-закладами

Об'єкт дослідження - розробка та впровадження Веб-сервісу для автоматизації управління відносинами з клієнтами спортзалу. Дана розробка у кінцевому результаті має не лише полегшити роботу адміністраторів, а й значно покращити взаємодію з клієнтами. Під час аналізу сучасного стану управління спортзалами, виявлено ряд проблем, які ускладнюють їхнє ефективне функціонування. В першу чергу, нестабільний облік клієнтів, втрата даних та неефективна комунікація з клієнтами стали серйозними труднощами. Ці недоліки стали основою для визначення актуальності та необхідності впровадження вказаного Веб-сервісу.

Ці проблеми стають кричущими, оглядаючи роботу сучасних спортзалів. Нестабільний облік клієнтів породжує плутанину в адміністративних процесах, а втрата даних призводить до недостовірного аналізу роботи закладу. Не менш важливим є інша аспект – неефективна комунікація з клієнтами. Відсутність зручних інструментів сповіщення та взаємодії може вплинути на задоволеність клієнтів та збереження їх лояльності.

Зокрема, складнощі при формуванні розкладу тренувань можуть призводити до невдоволення клієнтів, адже вони не можуть зручно планувати свої візити. Ураховуючи ці проблеми, розробка Веб-сервісу, спрямованого на автоматизацію управління відносинами з клієнтами, виявляється вельми обґрунтованою.

Зазначений сервіс покликаний вирішити ці проблеми шляхом надання інструментів для стабільного обліку клієнтів, уникнення втрати даних та покращення комунікації з клієнтами. Також, він має забезпечити зручний інтерфейс для формування розкладу тренувань, що враховує індивідуальні потреби клієнтів та покращує їхнє спортивне досвід.

Запровадження такого сервісу передбачає використання сучасних технологій та впровадження інноваційних підходів до управління спортивним закладом. Відповідно, адміністратори отримають інструменти для ефективного моніторингу клієнтської бази, автоматизованого обліку абонементів та тренувань, а також можливість оперативно реагувати на зміни та побажання клієнтів.

Додатково, Веб-сервіс спростить взаємодію клієнтів із спортзалом, забезпечивши їхній зручний доступ до персональних даних, актуального розкладу та інформації про акції та події. Це не тільки підвищить рівень задоволення клієнтів, але й сприятиме збільшенню лояльності та рекомендацій клієнтів, що є ключовим для успіху будь-якого спортивного закладу.

В подальших розділах дослідження буде розглянуто більший обсяг функціоналу Веб-сервісу, його переваги та можливості, щоб обґрунтувати його впровадження та визначити його роль у вдосконаленні управління спортзалом та підвищенні якості обслуговування клієнтів.

Основною метою введення Веб-сервісу є покращення ефективності управління клієнтами та оптимізація внутрішніх процесів спортзалу. Його впровадження дозволить адміністраторам спортзалу зосередитися на наданні якісного обслуговування, замість витрат на ручний облік та вирішення технічних проблем.

Важливою функцією Веб-сервісу буде забезпечення надійного обліку клієнтів та їхніх абонементів. Автоматизований контроль за термінами абонементів та оплата онлайн сприятимуть уникненню непорозумінь та забезпечать стабільність фінансового потоку для спортзалу.

У контексті покращення комунікації, Веб-сервіс передбачає впровадження системи повідомлень та розсилок, що дозволить спортзалу ефективно спілкуватися

з клієнтами. Запропонований інструментарій також дозволить адміністраторам легко інформувати клієнтів про акції, новини та будь-які зміни у розкладі.

Враховуючи індивідуальні потреби клієнтів, система формування розкладу тренувань дозволить кожному клієнту зручно планувати свій час та вибирати оптимальний графік тренувань. Це сприятиме підвищенню задоволеності клієнтів та створить умови для довгострокових відносин.

Загальною метою є створення інтегрованого інструменту, який покликаний не лише вирішити існуючі проблеми, а й підняти рівень обслуговування та конкурентоспроможність спортзалу на ринку. У подальших дослідженнях буде розглянуто більше деталей щодо функціоналу Веб-сервісу та його можливостей.

Нестабільний облік клієнтів та абонементів визначається відсутністю ефективних інструментів, що призводить до плутанини та непорозумінь в адміністративних процесах через ручний облік. З іншого боку, проблема втрати даних та недостовірного аналізу породжена не продуманим обліком, що може призвести до втрати важливої інформації та ускладнює аналіз роботи спортзалу.

Недостатня комунікація з клієнтами визначається відсутністю зручних інструментів сповіщення та взаємодії, що може впливати на загальну задоволеність клієнтів та їхню лояльність. Окрім того, складнощі при формуванні розкладу тренувань свідчать про брак систематизованого інструменту для планування та оновлення графіку тренувань, що може викликати невдоволення серед клієнтів через неефективний розподіл часу тренувань. Розгляд цих проблем є необхідним кроком для подальшого вирішення та впровадження Веб-сервісу, спрямованого на покращення управління та спілкування з клієнтами.

Отже, говорячи про конкретні проблеми, можна виділити наступні пункти:

1. Нестабільний облік клієнтів:

- Відсутність ефективних інструментів для точного та стабільного обліку клієнтів та їхніх абонементів.

- Плутанина та непорозуміння в адміністративних процесах через ручний облік.

2. Втрата даних та невірний аналіз:

- Непродуманий облік може призводити до втрати важливих даних.

- Недостовірний аналіз роботи спортзалу через відсутність автоматизованого інструменту.

3. Неefективна комунікація з клієнтами:

- Відсутність зручних інструментів сповіщення та взаємодії з клієнтами.

- Погана комунікація може впливати на задоволеність клієнтів та їхню лояльність.

4. Складнощі при формуванні розкладу тренувань:

- Брак систематизованого інструменту для зручного планування та оновлення розкладу тренувань.

- Невдоволення клієнтів через неефективний розподіл часу тренувань.

1.2. Огляд інформаційних технологій управління діяльністю фітнес-закладів

Успішне управління фітнес-закладом сьогодні неможливе без використання сучасних інформаційних технологій (ІТ). Вони забезпечують ефективну взаємодію з клієнтами, оптимізацію бізнес-процесів та підвищення конкурентоспроможності. Розглянемо основні категорії ІТ-рішень, що знаходять застосування у фітнес-індустрії.

CRM-системи – це потужний інструментарій, що ставить клієнта в центр уваги фітнес-закладу. Вони допомагають зібрати, систематизувати та проаналізувати величезний обсяг даних про кожного відвідувача, перетворюючи їх на цінні знання для розвитку бізнесу. Головна мета CRM:

- Персоналізація сервісу: crm дозволяє створити індивідуальний підхід до кожного клієнта, враховуючи його вподобання, історію відвідувань, цілі та потреби. Це допомагає запропонувати найрелевантніші послуги та продукти, що підвищує задоволеність та лояльність.
- Підвищення лояльності: завдяки персоналізації, своєчасним пропозиціям та індивідуальному спілкуванню, crm допомагає зміцнити зв'язок з клієнтами, перетворюючи їх на постійних відвідувачів.
- Стимулювання продажів: аналіз даних crm дозволяє виявити найбільш ефективні канали залучення клієнтів, визначити найпопулярніші послуги та розробити цільові маркетингові кампанії, що призводить до збільшення продажів.

Ключові функціональні можливості CRM-систем:

- Ведення клієнтської бази: зберігання та оновлення контактної інформації, даних про абонементи, історії відвідувань, участі у заходах, анкетних даних та іншої важливої інформації про клієнтів.
- Відстеження історії взаємодій: фіксація всіх контактів з клієнтами – дзвінків, електронних листів, повідомлень у месенджерах, зустрічей, опитувань тощо. Це дозволяє відстежувати ефективність комунікації та виявляти потенційні проблеми.
- Сегментація клієнтів: розподіл клієнтів на групи за різними критеріями (вік, стать, інтереси, рівень доходу, цілі тренувань тощо) для більш точного та ефективного таргетування маркетингових кампаній.
- Автоматизація маркетингових кампаній: створення та запуск персоналізованих розсилок, sms-повідомлень, push-сповіщень, вітальних повідомлень з днем народження та інших маркетингових активностей.

- Аналіз ефективності продажів: відстеження динаміки продажів, визначення найбільш прибуткових послуг та продуктів, оцінка ефективності маркетингових кампаній, прогнозування доходів.

Системи управління розкладом та бронювання є ключовим інструментом для ефективної організації роботи фітнес-центру. Вони дозволяють автоматизувати процес планування занять, забезпечуючи зручність як для клієнтів, так і для персоналу. Основні цілі:

- Зручність для клієнтів: онлайн-платформи та мобільні додатки дозволяють клієнтам легко переглядати розклад занять, вибирати зручний час та тренера, записуватися на заняття в будь-який час доби. Це економить час та підвищує задоволеність клієнтів.
- Оптимізація завантаження: системи допомагають розподілити навантаження на тренерів та приміщення, уникаючи перевантажень та простоїв. Це дозволяє максимально ефективно використовувати ресурси фітнес-центру.
- Збільшення продажів: можливість онлайн-бронювання та оплати занять стимулює клієнтів до більш активного використання послуг фітнес-центру, що призводить до збільшення продажів.

Ключові функціональні можливості:

- Онлайн-запис на заняття: клієнти можуть самостійно вибирати заняття, переглядати доступні місця та записуватися на тренування через веб-сайт або мобільний додаток.
- Управління розкладом: адміністратори можуть легко створювати, редагувати та публікувати розклад групових та індивідуальних тренувань, враховуючи доступність тренерів та приміщень.

- Інтеграція з платіжними системами: можливість онлайн-оплати занять за допомогою банківських карток, електронних гаманців та інших платіжних інструментів.
- Мобільні додатки для клієнтів: зручні мобільні додатки дозволяють клієнтам переглядати розклад, записуватися на заняття, відстежувати свої тренування та отримувати сповіщення про зміни у розкладі.
- Управління тренерським складом: система дозволяє відстежувати завантаженість тренерів, планувати їх робочий графік та розподіляти навантаження.
- Аналітика та звітність: системи надають детальну інформацію про відвідуваність занять, популярність тренерів, ефективність маркетингових кампаній та інші важливі показники, що допомагає приймати обґрунтовані управлінські рішення.

Системи контролю доступу та безпеки є невід'ємною складовою сучасного фітнес-центру. Вони забезпечують захист клієнтів, персоналу та майна, створюючи атмосферу безпеки та довіри. Це не просто технічні засоби, а комплексний підхід до забезпечення безпеки, який включає як фізичні, так і інформаційні компоненти. Основна мета:

- Захист клієнтів: безпека клієнтів – це пріоритет для будь-якого фітнес-центру. Системи контролю доступу допомагають запобігти проникненню сторонніх осіб, забезпечуючи комфорт та спокій відвідувачів.
- Захист персоналу: співробітники фітнес-центру також потребують захисту. Системи безпеки допомагають запобігти конфліктам, крадіжкам та іншим неприємним інцидентам, створюючи безпечне робоче середовище.
- Захист майна: фітнес-центри зазвичай мають дороге обладнання та інвентар, які потребують захисту від крадіжок та пошкоджень. Системи безпеки допомагають запобігти втратам та зберегти майно фітнес-центру.

- Запобігання надзвичайним ситуаціям: системи пожежної безпеки та відеоспостереження допомагають швидко виявляти та реагувати на надзвичайні ситуації, такі як пожежі, затоплення, витік газу тощо. Це дозволяє мінімізувати збитки та забезпечити безпеку людей.

Ключові функціональні можливості:

- Контроль доступу: системи контролю доступу дозволяють обмежити доступ до приміщень фітнес-центру лише для авторизованих осіб. Для цього використовуються різні технології:
- Картки доступу: це найпоширеніший спосіб контролю доступу. Кожному клієнту та співробітнику видається індивідуальна картка, яка дозволяє відкривати двері та проходити через турнікети.
- Біометрія: біометричні системи використовують унікальні фізичні характеристики людини (відбитки пальців, розпізнавання обличчя, сканування сітківки ока) для ідентифікації та надання доступу.
- Мобільні ідентифікатори: сучасні системи дозволяють використовувати смартфони як ідентифікатори для відкриття дверей та проходження через турнікети.
- Відеоспостереження: відеокамери встановлюються у всіх ключових зонах фітнес-центру, дозволяючи відстежувати ситуацію в режимі реального часу та записувати відео для подальшого аналізу. Це допомагає запобігти крадіжкам, вандалізму та іншим правопорушенням, а також розслідувати інциденти, якщо вони все ж таки відбулися.
- Інтеграція з системами пожежної безпеки: системи контролю доступу можуть бути інтегровані з системами пожежної сигналізації та оповіщення, що дозволяє автоматично розблокувати двері та евакуювати людей у разі пожежі.

Впровадження сучасних систем контролю доступу та безпеки – це інвестиція у майбутнє фітнес-центру. Це дозволяє створити безпечне та комфортне

середовище для клієнтів та персоналу, підвищити рівень довіри та лояльності, а також мінімізувати ризики та втрати.

Бізнес-аналітика та звітність – це потужний інструмент, який дозволяє фітнес-центрам перетворити величезний обсяг даних на цінну інформацію, необхідну для прийняття обґрунтованих управлінських рішень. Це свого роду "мозок" фітнес-бізнесу, який допомагає побачити повну картину діяльності, виявити сильні та слабкі сторони, зрозуміти потреби клієнтів та визначити шляхи розвитку. Основна мета:

- **Прийняття обґрунтованих рішень:** бізнес-аналітика надає керівництву фітнес-центру точну та актуальну інформацію про всі аспекти діяльності. Це дозволяє оцінити ефективність роботи, виявити проблемні зони та прийняти рішення, що базуються на фактах, а не на припущеннях.
- **Оптимізація бізнес-процесів:** аналіз даних допомагає виявити неефективні процеси, визначити вузькі місця та розробити заходи щодо їх покращення. Це дозволяє підвищити продуктивність, знизити витрати та збільшити прибутковість.
- **Розуміння потреб клієнтів:** аналіз даних про поведінку клієнтів, їх вподобання та відгуки допомагає краще зрозуміти їх потреби та очікування. Це дозволяє розробити нові послуги та продукти, які будуть користуватися попитом, а також покращити якість обслуговування.
- **Планування стратегії розвитку:** бізнес-аналітика допомагає визначити тенденції ринку, оцінити конкурентне середовище та спрогнозувати майбутні потреби клієнтів. Це дозволяє розробити ефективну стратегію розвитку фітнес-центру та досягти довгострокових цілей.

Ключові функціональні можливості:

- Збір та аналіз даних: системи бізнес-аналітики збирають дані з різних джерел – crm-систем, систем управління розкладом, касових апаратів, соціальних мереж, веб-сайту та інших. Ці дані аналізуються за допомогою спеціальних алгоритмів та інструментів, що дозволяє виявити закономірності, тенденції та взаємозв'язки.
- Формування звітів та дашбордів: системи генерують різноманітні звіти, що відображають ключові показники ефективності фітнес-центру – продажі, відвідуваність, прибутковість, ефективність маркетингових кампаній, задоволеність клієнтів тощо. Дашборди – це інтерактивні панелі, які візуалізують дані у вигляді графіків, діаграм та таблиць, що полегшує їх сприйняття та аналіз.
- Прогнозування: бізнес-аналітика дозволяє прогнозувати майбутні тенденції, наприклад, попит на різні види послуг, ефективність маркетингових кампаній, фінансові результати. Це допомагає приймати проактивні рішення та підготуватися до майбутніх викликів.

Бізнес-аналітика – це невід'ємна складова успішного управління фітнес-центром. Вона дозволяє приймати обґрунтовані рішення, оптимізувати роботу, задовольняти потреби клієнтів та досягати стабільного зростання.

1.3. Веб-технології інформаційної підтримки діяльності фітнес-закладів

Сучасні веб-технології відіграють важливу роль у забезпеченні інформаційної підтримки діяльності фітнес-закладів. Вони допомагають залучати нових клієнтів, підвищувати лояльність існуючих, оптимізувати бізнес-процеси та покращувати загальну ефективність роботи. Розглянемо основні веб-технології, які активно використовуються у фітнес-індустрії.

Веб-сайт – це не просто сторінка в Інтернеті, це віртуальне представництво фітнес-закладу, його обличчя в цифровому світі. Він є першим місцем, куди звертаються потенційні клієнти, щоб дізнатися більше про клуб, його послуги та атмосферу. Тому важливо, щоб веб-сайт був інформативним, зручним та привабливим. Основна мета веб-сайту фітнес-клубу:

Залучити та інформувати потенційних клієнтів, надаючи їм всю необхідну інформацію для прийняття рішення про відвідування клубу. Це включає:

- Інформаційні сторінки: детальний опис клубу, його історії, місії та цінностей, перелік послуг (групові заняття, персональні тренування, дитячий фітнес, спа-зона тощо), інформація про тренерів (кваліфікація, досвід, спеціалізація), контакти (адреса, телефон, email).
- Розклад занять: зручний онлайн-розклад, який дозволяє клієнтам швидко знайти потрібне заняття за типом, часом, тренером або рівнем складності. Важливо, щоб розклад був актуальним та легко оновлювався.
- Онлайн-запис: можливість забронювати заняття або персональне тренування безпосередньо на сайті, що економить час клієнтів та підвищує їх задоволеність. Це також допомагає клубу відстежувати попит на різні послуги та планувати завантаження тренерів.
- Блог: регулярне оновлення блогу з цікавими та корисними статтями про фітнес, здорове харчування, мотивацію та інші актуальні теми допомагає залучити нових відвідувачів та утримати існуючих. Блог також може використовуватися для просування послуг клубу та підвищення його експертності.
- Галерея: фото та відео з життя фітнес-клубу створюють атмосферу та допомагають потенційним клієнтам уявити, як проходять тренування, які умови в залах та яка атмосфера панує в клубі.

- Відгуки: можливість залишити відгук про клуб або тренера на сайті підвищує довіру до клубу та допомагає потенційним клієнтам зробити вибір. Важливо відповідати на відгуки, як позитивні, так і негативні, демонструючи відкритість та готовність до діалогу.

Ефективний веб-сайт – це потужний інструмент маркетингу та продажів для фітнес-клубу. Він допомагає залучати нових клієнтів, підвищувати їх лояльність та формувати позитивний імідж бренду.

SEO-оптимізація (Search Engine Optimization) – це комплекс заходів, спрямованих на підвищення позицій веб-сайту фітнес-клубу у результатах пошуку. Це як невидима рука, яка піднімає сайт вище у списку, роблячи його більш помітним для потенційних клієнтів. У світі, де більшість користувачів рідко переглядають результати пошуку далі першої сторінки, високі позиції – це ключ до успіху. Основна мета SEO-оптимізації:

- Збільшити органічний трафік на сайт фітнес-клубу, тобто кількість відвідувачів, які знаходять сайт через пошукові системи, не вкладаючи кошти в платну рекламу. Це досягається шляхом:
- Підвищення видимості сайту: seo допомагає сайту фітнес-клубу з'являтися на перших сторінках результатів пошуку за релевантними запитами, такими як "фітнес-клуб у [місто]", "групові заняття з йоги", "персональний тренер" тощо.
- Залучення цільової аудиторії: seo-оптимізація дозволяє залучати на сайт саме тих користувачів, які активно шукають послуги фітнес-клубу, що підвищує ймовірність їх конверсії у клієнтів.
- Зміцнення авторитету бренду: високі позиції у пошукових результатах свідчать про авторитетність та надійність фітнес-клубу, що позитивно впливає на його імідж.

Основні методи SEO-оптимізації:

- **Ключові слова:** це слова та фрази, які користувачі вводять у пошуковий рядок, щоб знайти потрібну інформацію. Важливо використовувати релевантні ключові слова у текстах на сайті, заголовках, описах, url-адресах тощо, щоб пошукові системи могли зрозуміти, про що йдеться на сайті та кому він може бути корисним.
- **Мета-теги:** це спеціальні теги, які не відображаються на сторінці, але містять важливу інформацію для пошукових систем. Мета-теги title (заголовок сторінки) та description (опис сторінки) допомагають пошуковим системам зрозуміти, про що йдеться на сторінці, та формують сніппет – короткий опис сайту, який відображається у результатах пошуку.
- **Зовнішні посилання:** це посилання на сайт фітнес-клубу з інших сайтів. Чим більше якісних посилань з авторитетних ресурсів веде на сайт, тим вище його рейтинг у пошукових системах.
- **Технічна оптимізація:** це комплекс заходів, спрямованих на покращення технічних характеристик сайту, таких як швидкість завантаження сторінок, адаптивність під мобільні пристрої, коректність коду тощо. Це важливо не лише для пошукових систем, але й для користувачів, які очікують швидкого та зручного доступу до інформації.

SEO-оптимізація – це довгостроковий процес, який вимагає постійної уваги та зусиль. Але результати варті того: високі позиції у пошукових результатах забезпечують стабільний потік цільового трафіку та допомагають фітнес-клубу досягти успіху в онлайн-просторі.

Контент-маркетинг – це стратегія залучення та утримання клієнтів за допомогою цінного та цікавого контенту. Це не просто реклама, а спосіб побудувати довгострокові відносини з аудиторією, стати для неї джерелом інформації, натхнення та мотивації. Основна мета контент-маркетингу:

- Залучення нових клієнтів: цікавий та корисний контент привертає увагу потенційних клієнтів, викликає інтерес до фітнес-клубу та спонукає їх до дії – відвідати сайт, записатися на пробне заняття, підписатися на розсилку тощо.
- Підвищення лояльності існуючих клієнтів: регулярне надання цінного контенту допомагає утримувати клієнтів, підтримувати їх інтерес до клубу та зміцнювати їх лояльність.
- Формування експертного іміджу: якісний контент позиціонує фітнес-клуб як експерта у своїй галузі, що підвищує довіру до бренду та робить його більш привабливим для клієнтів.

Формати контенту:

- Статті: інформативні статті про фітнес, здорове харчування, мотивацію, поради щодо тренувань та досягнення цілей допомагають клієнтам отримати корисні знання та навички, що підвищує їхню зацікавленість у послугах клубу.
- Відео: відео-тренування, інтерв'ю з тренерами, огляди клубу, історії успіху клієнтів – це чудовий спосіб візуально представити клуб та його послуги, залучити аудиторію та підвищити її мотивацію.
- Інфографіка: візуалізація статистичних даних, порад, інструкцій у вигляді інфографіки робить контент більш привабливим та легким для сприйняття.
- Вебінари: онлайн-семінари на актуальні теми, проведені експертами клубу, дозволяють залучити широку аудиторію, продемонструвати експертизу та надати цінну інформацію.

Соціальні мережі – це потужний інструмент для взаємодії з цільовою аудиторією. Вони дозволяють фітнес-клубу бути ближче до своїх клієнтів, спілкуватися з ними, ділитися новинами та отримувати зворотний зв'язок. Основна мета SMM:

- Залучення та спілкування з цільовою аудиторією: Соціальні мережі дозволяють фітнес-клубу знайти та залучити свою цільову аудиторію, спілкуватися з нею, відповідати на запитання, проводити опитування та конкурси, створювати спільноту однодумців.
- Підвищення впізнаваності бренду: Активна присутність у соціальних мережах допомагає підвищити впізнаваність бренду, зробити його більш близьким та зрозумілим для аудиторії.

Платформи:

- Facebook: створення сторінки клубу дозволяє публікувати новини, акції, фото, відео, проводити конкурси та опитування, взаємодіяти з підписниками.
- Instagram: візуальна платформа, ідеально підходить для публікації фото та відео з життя клубу, історій успіху клієнтів, проведення прямих ефірів та конкурсів.
- Youtube: розміщення відео-тренувань, оглядів обладнання, інтерв'ю з тренерами допомагає залучити нову аудиторію та продемонструвати експертизу клубу.

Email-маркетинг – це ефективний спосіб підтримувати зв'язок з клієнтами, інформувати їх про новини, акції та спеціальні пропозиції. Основна мета email-маркетингу:

- Інформування клієнтів: розсилки допомагають тримати клієнтів в курсі подій у фітнес-клубі, інформувати їх про нові послуги, акції, знижки, зміни у розкладі тощо.
- Підвищення продажів: промо-розсилки з привабливими пропозиціями можуть стимулювати клієнтів до покупки абонементів, відвідування занять та використання додаткових послуг.

- Підтримка лояльності: персоналізовані розсилки з привітаннями з днем народження, нагадуваннями про заняття та індивідуальними пропозиціями допомагають зміцнити відносини з клієнтами та підвищити їх лояльність.

Формати розсилок:

- Інформаційні: розклад занять, новини клубу, статті про фітнес, поради щодо здорового способу життя.
- Промо-розсилки: акції, знижки, спеціальні пропозиції, бонуси.
- Тригерні розсилки: привітання з днем народження, нагадування про заняття, реактивація неактивних клієнтів, персоналізовані пропозиції на основі поведінки клієнта.

Веб-технології є потужним інструментом для розвитку фітнес-бізнесу. Вони допомагають залучати нових клієнтів, підвищувати їх лояльність, покращувати комунікацію та ефективність роботи. Правильне використання веб-технологій дозволить фітнес-клубу зайняти лідируючі позиції на ринку та досягти успіху.

1.4. Перспективи впровадження XML для підвищення ефективності діяльності фітнес-закладів

Впровадження XML у фітнес-закладах відкриває широкі перспективи для підвищення ефективності їхньої діяльності. Завдяки своїй гнучкості та універсальності, XML може бути використаний для вирішення різноманітних завдань, пов'язаних з управлінням клієнтами, персоналом, фінансами та іншими аспектами роботи фітнес-центру.

XML дозволяє створювати структуровані бази даних клієнтів, що містять інформацію про їхні особисті дані, історію відвідувань, придбані послуги, уподобання тощо. Це дозволяє персоналу фітнес-закладу швидко знаходити необхідну інформацію, персоналізувати спілкування з клієнтами та пропонувати їм релевантні послуги.

XML може бути використаний для автоматизації різних бізнес-процесів, таких як реєстрація клієнтів, оформлення абонементів, розрахунок заробітної плати персоналу, формування звітності тощо. Це дозволяє зменшити кількість ручної роботи, мінімізувати помилки та підвищити ефективність роботи фітнес-закладу.

XML забезпечує легку інтеграцію з іншими системами, такими як бухгалтерські програми, системи відеоспостереження, системи контролю доступу тощо. Це дозволяє створити єдину інформаційну систему, що охоплює всі аспекти діяльності фітнес-закладу, та забезпечує комплексний підхід до управління.

XML може бути використаний для створення інтерактивних веб-сайтів та мобільних додатків, які надають клієнтам доступ до інформації про розклад занять, тренерів, послуги, акції тощо. Це дозволяє покращити комунікацію з клієнтами, підвищити їхню лояльність та залучити нових відвідувачів.

XML дозволяє збирати та аналізувати дані про діяльність фітнес-закладу, такі як кількість відвідувачів, популярність різних послуг, ефективність маркетингових кампаній тощо. Це дозволяє керівництву приймати обґрунтовані рішення щодо розвитку бізнесу, оптимізації витрат та підвищення прибутковості.

XML може бути використаний для створення індивідуальних програм тренувань для клієнтів фітнес-закладів. Тренери можуть використовувати XML для зберігання інформації про вправи, підходи, повторення, вагу та інші параметри тренувань. Ці дані можна легко обмінюватися між тренерами та клієнтами, а також інтегрувати з мобільними додатками для відстеження прогресу та мотивації.

XML може бути використаний для створення навчальних матеріалів для тренерів та іншого персоналу фітнес-закладів. Це можуть бути електронні підручники, інструкції, відеоуроки тощо. Завдяки XML, ці матеріали можна легко оновлювати та адаптувати до потреб конкретного фітнес-закладу.

XML може бути використаний для створення та поширення маркетингових матеріалів, таких як електронні розсилки, рекламні банери, статті тощо. Завдяки XML, ці матеріали можна легко адаптувати до різних каналів комунікації та цільових аудиторій.

Впровадження XML у фітнес-закладах має значний потенціал для підвищення ефективності їхньої діяльності. Завдяки використанню XML, фітнес-центри можуть оптимізувати управління клієнтськими даними, автоматизувати бізнес-процеси, покращити взаємодію з клієнтами, інтегруватися з іншими системами та приймати обґрунтовані рішення на основі аналізу даних. Це дозволить фітнес-закладам підвищити свою конкурентоспроможність, збільшити прибутковість та забезпечити високий рівень обслуговування клієнтів.

Застосування XML у фітнес-індустрії є перспективним напрямком, який може значно покращити ефективність та якість послуг, що надаються. Впровадження XML-технологій дозволить фітнес-закладам автоматизувати рутинні операції, покращити взаємодію з клієнтами та персоналом, а також відкрити нові можливості для розвитку бізнесу.

РОЗДІЛ 2. ІНТЕГРУВАННЯ XML У ВЕБ-СЕРВІСИ ДЛЯ УПРАВЛІННЯ ДІЯЛЬНІСЮ ФІТНЕС-ЗАКЛАДУ

2.1. Загальна характеристика та аналіз сучасного використання XML

XML (Розширювана Мова Розмітки) — це потужний інструмент для представлення, зберігання та обміну структурованими даними. Його можна уявити як універсальну мову, за допомогою якої різні програми та системи можуть "спілкуватися" між собою, розуміючи зміст переданої інформації.

XML (Розширювана Мова Розмітки) являє собою універсальний текстовий формат, призначений для структурування, зберігання та обміну даними між різними програмними системами та платформами. Він відрізняється від багатьох інших форматів тим, що не лише зберігає інформацію, а й описує її структуру та значення, що робить його "самоописовим".

Однією з ключових характеристик XML є його розширюваність. На відміну від HTML, де набір тегів фіксований, XML дозволяє користувачам створювати власні теги, які найкраще відповідають їхнім потребам та специфіці даних. Це надає величезну гнучкість та адаптивність, роблячи XML придатним для найрізноманітніших застосувань – від опису структури книг у бібліотеці до обміну складними фінансовими даними між банками.

XML також відрізняється високою структурованістю. Дані в ньому організовані у вигляді ієрархічного дерева, де кожен елемент може мати вкладені елементи. Така організація забезпечує чітку та логічну структуру даних, що полегшує їх обробку, пошук та аналіз. Наприклад, XML-документ, що описує книгу, може мати елемент "книга", який містить вкладені елементи "автор", "назва", "рік видання" тощо.

Важливою перевагою XML є можливість перевірки валідності документів. За допомогою спеціальних схем (DTD або XSD) можна визначити правила, яким повинен відповідати XML-документ. Це дозволяє переконатися, що дані

структуровані коректно та не містять помилок, що особливо важливо при обміні інформацією між різними системами. Наприклад, схема може вимагати, щоб кожен елемент "книга" обов'язково містив елементи "автор" та "назва".

Нарешті, XML є універсальним форматом, який знайшов застосування в багатьох сферах. Він використовується у веб-технологіях (XHTML, SVG, RSS), для обміну даними між різними програмами (SOAP, XML-RPC), для зберігання даних (у файлах або базах даних), а також для конфігурації програмного забезпечення. Завдяки своїй гнучкості, розширюваності та можливості перевірки валідності, XML став незамінним інструментом для роботи зі структурованими даними в сучасному цифровому світі.

XML є невід'ємною частиною сучасного вебу, забезпечуючи основу для багатьох ключових технологій. XHTML (розширювана гіпертекстова мова розмітки) є розширенням HTML, яке використовує синтаксис XML для більш суворої та структурованої розмітки веб-сторінок. SVG (масштабована векторна графіка) дозволяє створювати високоякісні векторні зображення, які можна масштабувати без втрати якості. RSS (формат стрічок новин) використовується для поширення оновлень контенту веб-сайтів, блогів та інших джерел інформації.

XML відіграє важливу роль у забезпеченні ефективного обміну даними між різними програмними системами та платформами. Він є основою для SOAP (простий протокол доступу до об'єктів), який дозволяє програмам, написаним на різних мовах програмування та працюючим на різних операційних системах, взаємодіяти між собою через Інтернет. XML-RPC (віддалений виклик процедур XML) є простішим протоколом, який використовується для віддаленого виклику функцій та процедур.

XML є зручним форматом для зберігання структурованих даних. Він може використовуватися для зберігання інформації у файлах, що дозволяє легко

переносити та обмінюватися даними між різними програмами та платформами. Крім того, XML може використовуватися для зберігання даних у базах даних, забезпечуючи гнучкість та масштабованість.

Багато програм використовують XML-файли для зберігання своїх налаштувань та параметрів. Це дозволяє легко змінювати конфігурацію програми без необхідності вносити зміни до її вихідного коду. XML-файли також зручні для читання та редагування як людьми, так і комп'ютерами, що спрощує процес налаштування програмного забезпечення.

Сучасне використання XML залишається актуальним, хоча й зазнало певних змін у порівнянні з піком його популярності. XML досі є важливим інструментом у багатьох сферах, але його роль дещо змінилася під впливом нових технологій та підходів.

Обмін даними між корпоративними системами: XML залишається незамінним інструментом для інтеграції різноманітних корпоративних систем, особливо у великих компаніях зі складною IT-інфраструктурою. Його здатність описувати структуру даних та забезпечувати їхню валідність робить його ідеальним вибором для обміну інформацією між різними відділами, програмами та платформами. Наприклад, XML може використовуватися для синхронізації даних про клієнтів, замовлення, інвентар та інші бізнес-процеси.

Фінансові та банківські операції: У фінансовому секторі, де точність, надійність та безпека даних мають вирішальне значення, XML знайшов широке застосування. Він використовується для обміну інформацією про транзакції, рахунки, платежі, звіти та інші фінансові документи. Завдяки своїй здатності до валідації та суворій структурі, XML забезпечує цілісність та коректність фінансових даних, що є критично важливим для запобігання помилкам та шахрайству.

Електронний документообіг: XML є основою для багатьох міжнародних стандартів електронного документообігу, таких як XBRL (розширювана мова ділової звітності) та UBL (універсальна бізнес-мова). XBRL використовується для стандартизації фінансової звітності, що полегшує її аналіз та порівняння. UBL надає єдиний формат для різних бізнес-документів, таких як рахунки-фактури, замовлення на покупку та транспортні накладні, що спрощує їх обробку та автоматизацію бізнес-процесів.

Медичні дані: У сфері охорони здоров'я XML відіграє важливу роль у забезпеченні обміну медичною інформацією між різними закладами та системами. Він використовується для представлення електронних медичних карток (ЕМК), які містять історію хвороби пацієнта, результати обстежень, призначення ліків та інші важливі дані. XML також використовується для обміну результатами лабораторних досліджень, рентгенівськими знімками та іншими медичними зображеннями.

Публікація контенту: XML залишається важливим інструментом для публікації контенту в різних форматах. Він широко використовується для створення електронних книг (e-books), наукових статей, технічної документації та інших видів публікацій. XML дозволяє структурувати контент, додавати метадані, забезпечувати посилання та створювати різні формати виведення (PDF, HTML, ePub тощо).

Хоча XML відіграв важливу роль у становленні сучасного вебу, його використання для розмітки веб-сторінок поступово зменшується. З появою HTML5 та CSS3, розробники отримали більш зручні та потужні інструменти для створення динамічних та інтерактивних веб-сайтів. HTML5 пропонує нові семантичні елементи, такі як <header>, <nav>, <article> та <footer>, які дозволяють чітко структурувати контент сторінки, тоді як CSS3 надає розширені можливості для оформлення та стилізації елементів. Хоча XML все ще використовується для деяких

веб-технологій, таких як SVG (масштабована векторна графіка), його роль у розмітці сторінок значно зменшилася.

У сфері веб-сервісів, які забезпечують взаємодію між різними програмами через Інтернет, XML також втрачає свої позиції. SOAP (простий протокол доступу до об'єктів), який базується на XML, був популярним вибором для розробки веб-сервісів завдяки своїй строгій структурі та можливості валідації. Однак, з появою RESTful API (інтерфейс прикладного програмування на основі репрезентативного стану), який використовує більш легкий та зручний формат JSON (нотація об'єктів JavaScript) для представлення даних, SOAP почав втрачати свою популярність. RESTful API виявилися більш простими у розробці та використанні, що сприяло їхньому швидкому поширенню та витісненню SOAP з багатьох проектів.

XML у комбінації з JSON: У сучасному світі розробки програмного забезпечення, де гнучкість та ефективність є ключовими факторами, XML та JSON часто використовуються разом, доповнюючи один одного. XML, завдяки своїй строгій структурі та можливості валідації, ідеально підходить для опису складних даних та забезпечення їхньої цілісності. JSON, зі свого боку, пропонує більш компактне та легке для читання представлення даних, що робить його ідеальним вибором для передачі інформації через мережу.

Такий підхід дозволяє поєднати переваги обох форматів: XML забезпечує надійність та структурованість даних на стороні сервера, а JSON спрощує їх обробку та передачу на стороні клієнта. Наприклад, конфігурація програми може зберігатися у форматі XML, що дозволяє легко перевірити її коректність, а потім перетворюватися у JSON для передачі на фронтенд, де вона буде використовуватися для налаштування інтерфейсу користувача.

XML для зберігання конфігурації: Незважаючи на появу нових форматів даних, XML залишається популярним вибором для зберігання конфігурації програмного забезпечення. Його ієрархічна структура дозволяє зручно

організувати налаштування та параметри, а можливість валідації забезпечує їхню коректність. Крім того, XML є текстовим форматом, що робить його легко читабельним та редагованим як людьми, так і комп'ютерами.

Завдяки цим перевагам, XML-файли конфігурації широко використовуються у різноманітних програмах – від простих утиліт до складних корпоративних систем. Вони дозволяють легко змінювати налаштування програми без необхідності втручатися у її вихідний код, що спрощує процес налаштування та підтримки програмного забезпечення.

XML, як і будь-яка технологія, має свої сильні та слабкі сторони. Розуміння цих аспектів є важливим для прийняття обґрунтованого рішення щодо його використання у конкретному проекті.

Таблиця 1

Переваги та недоліки XML	
ПЕРЕВАГИ	НЕДОЛІКИ
Розширюваність та гнучкість	Надмірність та складність синтаксису
Строга структура та можливість валідації	Великий розмір файлів
Самоописуваність даних	Відносно повільна обробка
Підтримка Unicode	Потреба у додаткових інструментах для роботи
Незалежність від платформи та мови	Більш складний для вивчення, ніж JSON
Широке застосування у різних сферах	

Незважаючи на появу нових технологій та форматів даних, XML (Розширювана Мова Розмітки) продовжує відігравати важливу роль у сучасному

цифровому світі. Його універсальність, розширюваність та здатність до валідації роблять його незамінним інструментом у багатьох сферах, від веб-розробки до фінансових операцій.

Веб-технології: Хоча частка веб-сайтів, що безпосередньо використовують XML у своїй розмітці, може здаватися відносно невеликою (28,5% станом на червень 2024 року, за даними W3Techs), його вплив на веб-технології набагато ширший. XML є основою для багатьох інших технологій, таких як RSS (формат стрічок новин), SVG (масштабована векторна графіка) та XHTML (розширений HTML), які використовуються на значно більшій кількості сайтів.

Корпоративні системи: У корпоративному середовищі, де інтеграція різноманітних систем та обмін даними є критично важливими, XML залишається популярним вибором. Його здатність описувати структуру даних та забезпечувати їхню валідність робить його ідеальним інструментом для обміну інформацією між різними відділами, програмами та платформами. Згідно з опитуванням Forrester Research, понад 70% компаній зі списку Fortune 500 використовують XML для інтеграції своїх систем.

Фінансовий сектор: У фінансовому секторі, де точність, надійність та безпека даних мають вирішальне значення, XML знайшов широке застосування. SWIFT, глобальна система міжбанківських фінансових телекомунікацій, використовує XML для більшості своїх повідомлень, що підтверджує його важливість для обміну фінансовою інформацією на міжнародному рівні.

XML також знайшов широке застосування в Україні, особливо в державному секторі та фінансовій сфері, де важливість структурованих та валідних даних є особливо високою. В Україні XML відіграє ключову роль у розвитку електронного документообігу, спрощуючи та пришвидшуючи обмін важливою інформацією між підприємствами, державними органами та громадянами. Завдяки своїй структурованості та можливості валідації, XML забезпечує надійність та цілісність

даних, що є особливо важливим при роботі з фінансовими та юридичними документами.

Впровадження електронних систем документообігу на основі XML дозволило значно скоротити час та витрати на обробку документів, усунути необхідність у паперових носіях та зменшити ризик помилок. Сьогодні українські підприємства активно використовують XML для обміну податковими накладними, актами виконаних робіт, договорами та іншими документами, що суттєво підвищує ефективність їхньої роботи та сприяє розвитку електронної комерції.

XML також відіграє важливу роль у банківському секторі України, де він використовується для обміну фінансовою інформацією між банками, платіжними системами та іншими фінансовими установами. Завдяки XML забезпечується швидкість, надійність та безпека фінансових транзакцій, що є критично важливим для стабільності та розвитку фінансової системи країни. Використання XML у банківському секторі також сприяє впровадженню нових фінансових технологій та послуг, таких як онлайн-банкінг та мобільні платежі.

Крім того, XML активно використовується в електронних системах державних послуг, що дозволяє громадянам отримувати різноманітні послуги, такі як реєстрація бізнесу, отримання довідок та дозволів, подання декларацій та інші, в електронному вигляді. Це значно спрощує та прискорює процес отримання послуг, зменшує бюрократичні бар'єри та сприяє підвищенню прозорості та ефективності роботи державних органів.

Загалом, XML залишається важливим інструментом у сучасному світі інформаційних технологій, особливо у сферах, де потрібна строга структура даних, валідація та надійність. Хоча його використання у деяких областях зменшилося, XML продовжує розвиватися та адаптуватися до нових вимог, зберігаючи свою актуальність у багатьох важливих галузях.

2.2. Особливості застосування технологій XML у бізнесі

Технології XML (eXtensible Markup Language) відіграють важливу роль у сучасному бізнес-середовищі, забезпечуючи ефективну обробку, обмін та зберігання даних. Завдяки своїй гнучкості, розширюваності та здатності структурувати інформацію, XML знаходить застосування у різних аспектах бізнес-діяльності.

XML дозволяє компаніям легко обмінюватися інформацією з партнерами, постачальниками та клієнтами, незалежно від використовуваних ними платформ та систем. Це досягається завдяки стандартизованій структурі XML-документів, що забезпечує їх коректну інтерпретацію різними програмними засобами. Такий універсальний обмін даними спрощує інтеграцію різних бізнес-процесів, покращує координацію між відділами компанії та сприяє ефективній взаємодії з зовнішніми контрагентами.

XML надає можливість створювати власні теги для опису даних, що робить його ідеальним інструментом для структурування різноманітної інформації. Наприклад, компанії можуть використовувати XML для створення каталогів продукції, де кожен товар буде описаний за допомогою тегів, що містять інформацію про його назву, ціну, характеристики тощо. Також XML використовується для структурування фінансових звітів, замовлень, медичних карток пацієнтів та інших типів даних. Структуровані дані легше обробляти, аналізувати та зберігати, що сприяє підвищенню ефективності бізнес-процесів та прийняттю обґрунтованих рішень.

XML відіграє важливу роль у створенні динамічних веб-сторінок, які можуть адаптуватися до потреб користувачів та відображати актуальну інформацію. За допомогою XML можна розділити вміст веб-сторінки від її оформлення, що дозволяє легко змінювати зовнішній вигляд сайту без необхідності вносити зміни у кожну сторінку окремо. Також XML використовується для створення

інтерактивних елементів, таких як форми зворотного зв'язку, кошики для покупок та інші компоненти, що забезпечують зручну взаємодію з користувачами.

XML дозволяє автоматизувати обмін даними між різними системами, такими як ERP (Enterprise Resource Planning), CRM (Customer Relationship Management) та електронна комерція. Наприклад, замовлення, оформлене клієнтом в інтернет-магазині, може бути автоматично передане до ERP-системи для обробки та виконання. Також XML використовується для автоматичного формування рахунків-фактур, накладних та інших документів, що супроводжують бізнес-процеси. Автоматизація обміну даними зменшує кількість ручної роботи, мінімізує помилки та прискорює виконання завдань, що сприяє підвищенню ефективності бізнесу.

XML використовується для створення електронних документів, які можна легко редагувати, формувати та поширювати. На відміну від традиційних текстових документів, XML-документи містять інформацію про структуру та семантику даних, що дозволяє використовувати їх у різних контекстах. Наприклад, XML-документ, що містить інформацію про товар, може бути використаний для створення каталогу продукції, формування замовлення на покупку або автоматичного заповнення форми на веб-сайті. Також XML використовується для створення електронних книг, журналів та інших публікацій, що забезпечує зручний доступ до інформації та знижує витрати на друк та зберігання паперу.

XML підтримує різні механізми шифрування та цифрового підпису, що дозволяє захистити конфіденційну інформацію від несанкціонованого доступу. Наприклад, компанії можуть використовувати XML для шифрування фінансових звітів, медичних карток пацієнтів та інших документів, що містять чутливу інформацію. Також XML використовується для створення цифрових підписів, що підтверджують автентичність документів та цілісність даних.

XML дозволяє створювати документи та веб-сторінки, які можуть відображатися різними мовами. Це досягається завдяки використанню спеціальних тегів, що містять інформацію про мову тексту. Наприклад, тег <en> може використовуватися для позначення англійського тексту, а тег <uk> - для українського. Завдяки цьому, компанії можуть легко створювати багатомовні веб-сайти та документи, що розширює їхню аудиторію та покращує комунікацію з міжнародними партнерами та клієнтами.

XML-структуровані дані допомагають пошуковим системам краще розуміти вміст веб-сайтів, що сприяє підвищенню їх позицій у результатах пошуку. Наприклад, використовуючи мікророзмітку Schema.org, можна надати пошуковим системам детальну інформацію про товари, послуги, події, рецепти та інші типи контенту. Це дозволяє пошуковим системам відображати розширені результати пошуку (rich snippets), які містять додаткову інформацію, таку як рейтинги, ціни, наявність тощо. Розширені результати пошуку привертають більше уваги користувачів та збільшують ймовірність переходів на сайт, що позитивно впливає на його трафік та конверсію.

XML легко інтегрується з іншими веб-технологіями, такими як HTML, CSS та JavaScript. Це дозволяє створювати складні веб-додатки, які поєднують структуровані дані з інтерактивними елементами та візуальним оформленням. Наприклад, XML може використовуватися для зберігання даних про товари, які потім відображаються на веб-сторінці за допомогою HTML та CSS. JavaScript може використовуватися для обробки даних з XML-файлу та динамічного оновлення вмісту сторінки. Така інтеграція дозволяє створювати гнучкі та масштабовані веб-рішення, які легко підтримувати та розвивати.

Завдяки своїй гнучкості та модульності, XML дозволяє розробникам створювати багаторазові компоненти, які можна використовувати у різних проєктах. Це зменшує час та зусилля, необхідні для розробки нових рішень, та

спрощує їх підтримку. Наприклад, розробивши XML-шаблон для створення каталогу продукції, компанія може легко використовувати його для створення каталогів для різних категорій товарів або для різних мовних версій сайту. Також XML дозволяє легко вносити зміни до структури даних без необхідності переробляти весь код програми, що спрощує оновлення та модернізацію веб-сайтів та додатків.

Технологія XML знаходить широке застосування в різних сферах бізнесу завдяки своїй гнучкості, розширюваності та здатності структурувати дані. Нижче наведена таблиця 2, яка ілюструє основні місця застосування XML та їхні особливості:

Таблиця 2

МІСЦЕ ЗАСТОСУВАННЯ XML	ОСОБЛИВОСТІ ЗАСТОСУВАННЯ
Обмін даними між різними системами (ERP, CRM, електронна комерція)	Стандартизація формату даних, спрощення інтеграції, автоматизація процесів
Веб-розробка (створення динамічних сторінок, інтерактивних елементів)	Розділення вмісту та оформлення, адаптивність до потреб користувачів, підтримка багатомовності
Структурування та зберігання даних (каталоги продукції, фінансові звіти, медичні картки)	Створення власних тегів, легкість пошуку та аналізу інформації, покращення організації даних
Створення електронних документів (електронні книги, журнали, звіти)	Легкість редагування та форматування, можливість використання в різних контекстах, зниження витрат на друк

<p>Автоматизація бізнес-процесів (обробка замовлень, формування рахунків)</p>	<p>Зменшення кількості ручної роботи, мінімізація помилок, прискорення виконання завдань</p>
<p>Забезпечення безпеки даних (шифрування, цифровий підпис)</p>	<p>Захист конфіденційної інформації, підтвердження автентичності документів</p>

Таким чином, технологія XML є універсальним інструментом, що значно розширює можливості бізнесу в різних сферах. Застосування XML дозволяє компаніям оптимізувати внутрішні процеси, покращити взаємодію з партнерами та клієнтами, підвищити ефективність роботи з даними та забезпечити їх безпеку. Гнучкість та масштабованість XML роблять його цінним активом для компаній, які прагнуть досягти успіху в сучасному цифровому світі.

2.3. Технологія інтегрування XML у Веб-сервіс.

Інтеграція XML у веб-сервіси є важливим аспектом сучасної розробки, оскільки XML забезпечує стандартизований та гнучкий формат для обміну даними між різними системами та платформами. Веб-сервіси, що використовують XML, можуть легко взаємодіяти з іншими веб-сервісами та додатками, незалежно від мови програмування чи технологій, на яких вони побудовані.

1. SOAP (Simple Object Access Protocol)

SOAP є одним з найпоширеніших протоколів для інтеграції XML у веб-сервіси. Він визначає стандартний формат для структурування повідомлень, що дозволяє веб-сервісам обмінюватися інформацією незалежно від платформи, мови програмування чи технології. SOAP-повідомлення зазвичай передаються через протокол HTTP, але можуть використовувати й інші транспортні протоколи.

Структура SOAP-повідомлення включає конверт, заголовок та тіло. Конверт визначає XML-простір імен та кодування повідомлення. Заголовок може містити

інформацію про маршрутизацію, безпеку та інші додаткові дані. Тіло повідомлення містить власне XML-дані, які представляють запит до веб-сервісу або відповідь від нього.

2. WSDL (Web Services Description Language)

WSDL є важливим компонентом інтеграції XML у веб-сервіси, оскільки він надає машинно-читаний опис інтерфейсу веб-сервісу. WSDL-документ визначає операції, які може виконувати веб-сервіс, повідомлення, що використовуються для запитів та відповідей, типи даних, протоколи зв'язку та адреси кінцевих точок.

WSDL-документи дозволяють клієнтам веб-сервісів автоматично генерувати код для взаємодії з сервісом, що значно спрощує процес інтеграції. Крім того, WSDL-документи можуть бути використані для створення інструментів тестування та моніторингу веб-сервісів.

3. XML Schema

XML Schema є мовою визначення структури XML-документів. Він дозволяє визначати елементи, атрибути та типи даних, які можуть бути використані в XML-документі, а також встановлювати обмеження на їх значення та взаємозв'язки.

XML Schema відіграє важливу роль у забезпеченні валідності та надійності обміну даними між веб-сервісами. Використовуючи XML Schema, розробники можуть переконатися, що XML-дані, що надсилаються та отримуються веб-сервісом, відповідають очікуваній структурі та формату.

4. REST (Representational State Transfer)

REST є альтернативним підходом до побудови веб-сервісів, який відрізняється від SOAP своєю простотою та гнучкістю. RESTful веб-сервіси використовують стандартні HTTP-методи (GET, POST, PUT, DELETE) для виконання операцій над ресурсами, представленими у вигляді унікальних URI.

RESTful веб-сервіси часто використовують XML або JSON для представлення даних. Хоча XML не є обов'язковим для REST, він може бути ефективно використаний для структурування даних та забезпечення їхньої валідності.

Вибір між SOAP та REST залежить від конкретних вимог проекту. SOAP зазвичай використовується у складних корпоративних системах, де важлива стандартизація, безпека та надійність. REST, у свою чергу, краще підходить для простих веб-сервісів, де важлива швидкість та легкість розробки.

У сучасному світі, де інформаційні технології відіграють ключову роль у розвитку бізнесу, веб-сервіси стають незамінним інструментом для обміну даними між різними системами та платформами. XML (eXtensible Markup Language), завдяки своїм численним перевагам, займає особливе місце серед технологій, що використовуються для інтеграції у веб-сервісах.

➤ Стандартизація

XML є універсальною мовою розмітки, визнаною та підтримуваною більшістю сучасних платформ та технологій. Це означає, що веб-сервіси, побудовані на XML, можуть без проблем обмінюватися даними з іншими системами, незалежно від того, на якій мові програмування вони написані або на якій операційній системі працюють. Така стандартизація значно спрощує інтеграцію різних компонентів та забезпечує високий рівень сумісності між ними. Крім того, XML підтримується великою кількістю інструментів та бібліотек, що полегшує розробку та підтримку веб-сервісів.

➤ Гнучкість

Однією з ключових переваг XML є його гнучкість. На відміну від багатьох інших форматів даних, XML не має жорстко визначеної структури. Розробники можуть створювати власні теги та атрибути, що дозволяє їм точно відображати

специфіку даних, з якими вони працюють. Така гнучкість дає змогу легко адаптувати XML до різних потреб та вимог проекту. Наприклад, можна створити XML-схему для опису даних про клієнтів фітнес-клубу, яка буде містити інформацію про їхні особисті дані, історію відвідувань, придбані послуги тощо.

➤ **Розширюваність**

XML легко розширюється за рахунок додавання нових тегів та атрибутів. Це особливо важливо для веб-сервісів, які можуть еволюціонувати з часом. Наприклад, якщо у фітнес-клубі з'являються нові послуги, можна легко додати відповідні теги до XML-схеми, не порушуючи сумісність з існуючими клієнтами веб-сервісу. Така розширюваність дозволяє веб-сервісам залишатися актуальними та адаптуватися до нових вимог бізнесу.

➤ **Читабельність**

XML-документи мають чітку ієрархічну структуру, що робить їх легко читабельними як для людей, так і для машин. Теги та атрибути в XML мають змістовні назви, що дозволяє швидко зрозуміти, яку інформацію вони містять. Це значно полегшує процес розробки, налагодження та підтримки веб-сервісів. Крім того, читабельність XML дозволяє використовувати його для створення документації та інших матеріалів, призначених для людей.

➤ **Підтримка інтернаціоналізації**

XML підтримує різні кодування символів, включаючи Unicode, що дозволяє використовувати його для роботи з даними, написаними різними мовами. Це особливо важливо для веб-сервісів, які можуть використовуватися у різних країнах та регіонах. Крім того, XML дозволяє вказувати мову документа за допомогою спеціального атрибута `xml:lang`, що полегшує обробку багатомовних даних.

Після визначення концептуальних основ та теоретичних аспектів використання XML у веб-сервісах, важливо розглянути практичні кроки та інструменти, необхідні для успішної реалізації інтеграції. Цей процес включає кілька ключових етапів, кожен з яких відіграє важливу роль у забезпеченні функціональності, надійності та ефективності веб-сервісу.

➤ Розробка WSDL-документа

WSDL (Web Services Description Language) є основою для визначення та опису інтерфейсу веб-сервісу. Це XML-документ, який містить детальну інформацію про функціональність веб-сервісу, включаючи доступні операції, формати запитів та відповідей, використовувані протоколи (наприклад, SOAP) та адреси кінцевих точок. Розробка WSDL-документа є критично важливим кроком, оскільки він служить своєрідним контрактом між веб-сервісом та його клієнтами, визначаючи, як вони повинні взаємодіяти.

У WSDL-документі кожна операція веб-сервісу описується за допомогою елементів <operation>, які містять інформацію про назву операції, вхідні та вихідні повідомлення, а також можливі помилки. Повідомлення описуються за допомогою елементів <message>, які вказують на XML-схеми, що визначають структуру даних. Крім того, WSDL-документ містить інформацію про прив'язки (bindings), які визначають, як повідомлення кодуються та передаються через мережу (наприклад, за допомогою SOAP через HTTP).

➤ Створення XML Schema

XML Schema (XSD) є мовою визначення структури XML-документів. Вона дозволяє визначати елементи, атрибути та типи даних, які можуть бути використані в XML-документі, а також обмеження на їх значення та взаємозв'язки. XML Schema

відіграє важливу роль у забезпеченні валідності та надійності обміну даними між веб-сервісом та його клієнтами.

При створенні XML Schema для веб-сервісу, розробники визначають структуру вхідних та вихідних повідомлень для кожної операції. Це включає визначення елементів, їхніх типів даних (наприклад, string, integer, date), атрибутів та обмежень на їх значення (наприклад, мінімальна та максимальна довжина рядка, діапазон значень для чисел). XML Schema також дозволяє визначати складні типи даних, які складаються з інших елементів та атрибутів.

➤ **Реалізація логіки веб-сервісу**

Після визначення інтерфейсу веб-сервісу за допомогою WSDL та структури даних за допомогою XML Schema, наступним кроком є реалізація його логіки. Це включає написання коду, який обробляє вхідні запити від клієнтів, виконує необхідні операції (наприклад, доступ до бази даних, виконання обчислень) та формує вихідні XML-відповіді.

Реалізація логіки веб-сервісу може бути виконана на різних мовах програмування, таких як Java, C#, Python тощо. Існують різні фреймворки та бібліотеки, які спрощують цей процес, надаючи готові інструменти для роботи з SOAP, WSDL та XML Schema.

➤ **Тестування та налагодження**

Після реалізації веб-сервісу необхідно провести його ретельне тестування та налагодження. Це включає перевірку роботи веб-сервісу з різними типами запитів та даних, виявлення та виправлення помилок, а також оптимізацію продуктивності.

Для тестування веб-сервісів можна використовувати різні інструменти, такі як SoapUI, Postman тощо. Ці інструменти дозволяють відправляти запити до веб-

сервісу, переглядати відповіді, перевіряти валідність XML-даних та аналізувати продуктивність. Налагодження веб-сервісу може включати використання відлагоджувачів, логування та інших методів для виявлення та виправлення проблем у коді.

РОЗДІЛ 3. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

3.1. Обґрунтування вибору засобів реалізації

Front-End (Клієнтська частина)

Front-End - це те, що користувач бачить і з чим безпосередньо взаємодіє у своєму браузері. Для створення привабливого, функціонального та зручного інтерфейсу веб-сервісу CRM для спортзалу ми пропонуємо використовувати наступні технології:

1. HTML 5 (HyperText Markup Language 5)

HTML5 – це п'ята версія мови розмітки гіпертексту, яка є стандартом для створення веб-сторінок. Вона визначає, які елементи будуть присутні на сторінці та як вони будуть розташовані один відносно одного. HTML5 надає теги (спеціальні мітки) для визначення заголовків (<h1>, <h2> тощо), абзаців (<p>), зображень (), посилань (<a>), таблиць (<table>), списків (,) та багатьох інших елементів.

Одна з ключових переваг HTML5 полягає у розширених можливостях семантичної розмітки. Це означає, що замість використання загальних тегів, таких як <div> та , ми можемо використовувати більш спеціалізовані теги, які несуть змістовне значення. Наприклад, тег <header> позначає заголовок сторінки, <nav> – навігаційне меню, <article> – основний контент, <footer> – підвал сторінки.

Використання семантичної розмітки має кілька важливих переваг:

- **Покращення читабельності коду:** Код стає більш структурованим та зрозумілим для розробників, що полегшує його підтримку та модифікацію.

- **Покращення доступності:** Сайти з семантичною розміткою краще сприймаються допоміжними технологіями (скрінрідерами), що робить їх доступнішими для людей з обмеженими можливостями.
- **Покращення SEO:** Пошукові системи, такі як Google, використовують семантичну розмітку для кращого розуміння структури та змісту сторінки, що може позитивно вплинути на її позицію в результатах пошуку.

Таким чином, HTML5 є невід'ємною частиною сучасної веб-розробки, забезпечуючи міцний фундамент для створення зручних, доступних та добре оптимізованих веб-сайтів та додатків, включаючи CRM-системи для спортзалів.

2. CSS 3 (Cascading Style Sheets 3)

CSS3 - Ця мова стилів дозволяє нам перетворити сухий та нудний HTML-скелет у візуально привабливий, стильний та зручний для користувача інтерфейс.

CSS3 дає нам повний контроль над зовнішнім виглядом кожного елемента на сторінці. Ми можемо змінювати кольори тексту та фону, вибирати шрифти та їх розміри, встановлювати відступи та поля, позиціонувати елементи на сторінці, додавати тіні, градієнти, закруглені кути, рамки та інші візуальні ефекти. За допомогою CSS3 можна створювати вражаючі анімації та переходи, які зроблять взаємодію з веб-сервісом більш приємною та захоплюючою.

Однією з найважливіших можливостей CSS3 є створення адаптивного дизайну. У сучасному світі люди використовують для доступу до Інтернету найрізноманітніші пристрої – від великих моніторів до планшетів та смартфонів. Адаптивний дизайн дозволяє веб-сторінці автоматично підлаштовуватися під розмір екрану, забезпечуючи оптимальний вигляд та зручність використання на будь-якому пристрої. Це особливо важливо для веб-сайту спортзалу, яким співробітники можуть користуватися як на робочому місці, так і в дорозі.

Застосування CSS3 дозволяє створити унікальний та впізнаваний стиль для веб-сервісу, який буде відображати бренд спортзалу та привертати увагу користувачів. Гармонійне поєднання кольорів, шрифтів та візуальних ефектів зробить інтерфейс не лише привабливим, але й інтуїтивно зрозумілим, що сприятиме підвищенню ефективності роботи співробітників та задоволеності клієнтів.

3. JavaScript (ES6+)

JavaScript – це мова програмування, яка надає веб-сторінкам життя. Завдяки JavaScript сторінки можуть реагувати на дії користувача, змінювати свій вигляд та функціональність без необхідності перезавантаження. Це відкриває величезні можливості для створення динамічних та інтерактивних інтерфейсів.

JavaScript дозволяє реалізувати безліч функцій, які є критично важливими для веб-сайту спортзалу:

- **Динамічне оновлення контенту:** Наприклад, коли користувач вибирає дату у календарі, JavaScript може автоматично оновити список доступних занять або тренерів.
- **Валідація форм:** JavaScript може перевіряти, чи правильно користувач заповнив форму (наприклад, чи ввів коректну адресу електронної пошти), і повідомляти про помилки, не відправляючи дані на сервер.
- **Обробка подій користувача:** JavaScript може реагувати на кліки миші, наведення курсора, натискання клавіш та інші дії користувача, виконуючи відповідні команди.
- **Взаємодія з сервером через API:** JavaScript може відправляти запити на сервер для отримання або оновлення даних, що дозволяє реалізувати функціональність, яка вимагає доступу до бази даних або інших ресурсів сервера.

ES6+ (ECMAScript 2015+) – це сучасний стандарт JavaScript, який впроваджує багато нових можливостей та покращень, що спрощують та прискорюють розробку. Наприклад, ES6+ додає нові способи оголошення змінних (let та const), стрілкові функції, шаблонні рядки, деструктуризацію, класи та багато іншого.

Використання JavaScript (ES6+) дозволяє створити кращий функціонал веб-сайту для спортзалу, який буде не просто статичним набором сторінок, а потужним інтерактивним інструментом, який допоможе співробітникам ефективно управляти відносинами з клієнтами, а клієнтам – зручно записуватися на заняття, відстежувати свої результати та спілкуватися з тренерами.

Фреймворк/Бібліотека (React, Vue.js, Angular): Для спрощення та прискорення розробки складних інтерфейсів рекомендується використовувати фреймворк або бібліотеку JavaScript. Це готові набори інструментів та компонентів, які дозволяють швидше та ефективніше створювати інтерактивні веб-додатки.

- **React:** Розроблений Facebook, він використовує декларативний підхід, що дозволяє описувати, як інтерфейс має виглядати в залежності від даних, а не як його створювати крок за кроком. Це робить код більш читабельним та легким у підтримці. React також відомий своєю високою продуктивністю та великою спільнотою розробників.
- **Vue.js:** Це прогресивний фреймворк, який легко інтегрується в існуючі проекти. Він простий у вивченні та використанні, але водночас досить потужний для створення складних додатків. Vue.js також має гнучку архітектуру, що дозволяє вибрати підхід до розробки, який найкраще підходить для конкретного проекту.
- **Angular:** Розроблений Google, Angular – це повноцінний фреймворк, який пропонує широкий набір інструментів для розробки складних веб-додатків.

Він використовує TypeScript (надмножину JavaScript з статичною типізацією), що допомагає виявляти помилки на ранніх етапах розробки.

У світі веб-розробки технології постійно розвиваються, з'являються нові інструменти та підходи. Однак, деякі технології зарекомендували себе як надійні, ефективні та універсальні, ставши стандартом у галузі. Саме такі технології ми пропонуємо використовувати для створення CRM-системи для вашого спортзалу.

HTML5, CSS3 та JavaScript (ES6+) – це тріада, на якій базується практично кожен сучасний веб-сайт та веб-додаток. Ці технології є відкритими стандартами, підтримуваними усіма основними браузерами (Chrome, Firefox, Safari, Edge, Opera). Це означає, що веб-сервіс, створений з їх використанням, буде коректно відображатися та працювати на різних пристроях та операційних системах, забезпечуючи кросбраузерну сумісність.

Висока продуктивність – ще одна важлива перевага цих технологій. Сучасні браузери оптимізовані для роботи з HTML5, CSS3 та JavaScript, що дозволяє досягти швидкого завантаження сторінок та миттєвої реакції на дії користувача. Це особливо важливо для веб-сайту спортзалу, де швидкість та зручність роботи є критичними факторами для ефективного управління відносинами з клієнтами.

Крім того, HTML5, CSS3 та JavaScript (ES6+) надають величезні можливості для створення сучасних, інтерактивних та зручних інтерфейсів. Завдяки широкому спектру візуальних ефектів, анімацій, можливостей динамічного оновлення контенту та взаємодії з користувачем, ми зможемо створити веб-сайт, який буде не лише функціональним, але й естетично привабливим та інтуїтивно зрозумілим.

Використання фреймворків або бібліотек JavaScript, таких як React, Vue.js або Angular, дозволить ще більше прискорити та спростити процес розробки, а також забезпечити високу якість коду та легкість його підтримки.

Back-End (Серверна частина)

Back-end – це серверна частина веб-сервісу, яка відповідає за логіку роботи, обробку даних, взаємодію з базою даних та інші процеси, приховані від очей користувача. Для створення back-end веб-сайту для спортзалу. Для реалізації використовуються наступні технології:

1. C# (з ASP.NET Core MVC)

C# – це мова програмування, яку можна порівняти з універсальним інструментом у руках майстра. Вона дозволяє створювати різноманітні програми, від простих консольних утиліт до складних корпоративних систем. Розроблена компанією Microsoft, C# поєднує в собі елегантність та простоту з потужністю та гнучкістю, що робить її ідеальним вибором для розробки back-end частини CRM-системи для спортзалу.

C# – це об'єктно-орієнтована мова програмування, що означає, що вона дозволяє розробникам моделювати реальний світ у вигляді об'єктів, які мають свої властивості та методи. Такий підхід робить код більш структурованим, зрозумілим та легким у підтримці, що особливо важливо при розробці складних систем, таких як CRM.

ASP.NET Core MVC – це фреймворк, який спрощує та прискорює розробку веб-додатків на базі C#. MVC (Model-View-Controller) – це архітектурний патерн, який розділяє додаток на три основні компоненти:

- **Модель (Model):** Відповідає за дані та бізнес-логіку додатка.
- **Представлення (View):** Відповідає за відображення даних користувачеві.
- **Контролер (Controller):** Обробляє запити користувача, взаємодіє з моделлю та вибирає відповідне представлення.

Такий поділ дозволяє розробникам зосередитися на окремих аспектах додатка, не заглиблюючись у деталі інших компонентів. Це робить код більш модульним, легким у тестуванні та підтримці.

ASP.NET Core MVC надає розробникам широкий набір готових інструментів та шаблонів для реалізації типових задач, таких як маршрутизація, автентифікація, авторизація, робота з формами, валідація даних тощо. Це дозволяє значно скоротити час розробки та зосередитися на реалізації унікальної бізнес-логіки CRM-системи.

Використання C# (з ASP.NET Core MVC) для розробки back-end частини CRM-системи для спортзалу дозволить створити надійний, масштабований та ефективний серверний додаток, який буде відповідати всім вимогам сучасного бізнесу.

2. Node.js (з Express.js)

Node.js – це середовище виконання JavaScript, яке працює на сервері. Це означає, що ви можете використовувати JavaScript для написання коду, який буде обробляти запити від клієнтів, взаємодіяти з базою даних, виконувати обчислення та інші операції, які зазвичай виконуються на сервері.

Express.js – це фреймворк для Node.js, який значно спрощує створення веб-серверів та API (Application Programming Interface). API – це набір правил та специфікацій, які дозволяють різним програмам обмінюватися даними. У випадку CRM-системи, API буде використовуватися для обміну даними між front-end (інтерфейсом користувача) та back-end (серверною частиною).

Express.js надає розробникам зручний набір інструментів та функцій для створення маршрутів (шляхів, за якими клієнти можуть звертатися до сервера),

обробки запитів, роботи з шаблонами, налаштування middleware (програмного забезпечення, яке виконується між запитом та відповіддю) та багато іншого.

Використання Node.js та Express.js для розробки back-end частини CRM-системи має кілька переваг:

- Швидкість та продуктивність: node.js використовує асинхронну модель обробки запитів, що дозволяє йому обробляти велику кількість запитів одночасно без блокування. Це робить node.js дуже швидким та ефективним, що особливо важливо для веб-додатків з високим навантаженням.
- Масштабованість: node.js легко масштабується, що дозволяє додавати нові сервери для обробки збільшеного навантаження.
- Уніфікація технологій: використання javascript як на фронтенді, так і на бекенді дозволяє використовувати одні й ті ж знання та навички, що спрощує розробку та підтримку сrm-системи.
- Велика спільнота та екосистема: node.js має величезну спільноту розробників та велику кількість готових модулів та бібліотек, які можна використовувати для вирішення різних задач.

Завдяки цим перевагам, Node.js (з Express.js) є відмінним вибором для створення сучасних, швидких та масштабованих веб-додатків, включаючи CRM-системи для спортзалів.

3. XML (eXtensible Markup Language)

XML (eXtensible Markup Language) – це мова розмітки, схожа на HTML, яка використовується для опису структури даних. Вона дозволяє зберігати дані у вигляді ієрархії тегів, де кожен тег представляє певний елемент даних. Наприклад, тег <client> може містити інформацію про клієнта (ім'я, прізвище, контактні дані), а тег <visit> – інформацію про відвідування клієнта (дата, час, заняття).

Хоча XML не є повноцінною базою даних, як, наприклад, реляційні бази даних (PostgreSQL, MySQL) або NoSQL бази даних (MongoDB), він має свої переваги:

- **Простота:** xml-файли легко створювати, читати та редагувати за допомогою будь-якого текстового редактора.
- **Портативність:** xml-файли є текстовими файлами, які можна легко переносити між різними системами та платформами.
- **Структурованість:** xml дозволяє описувати структуру даних, що робить їх більш зрозумілими та легкими в обробці.
- **Розширюваність:** xml дозволяє додавати нові теги та атрибути без необхідності змінювати структуру існуючих даних.

Однак, використання XML як бази даних має і свої обмеження:

- **Продуктивність:** xml не оптимізований для швидкого пошуку та обробки великих обсягів даних, як спеціалізовані бази даних.
- **Масштабованість:** зі збільшенням обсягу даних робота з xml-файлами може стати повільною та незручною.
- **Відсутність вбудованих механізмів:** xml не надає вбудованих механізмів для управління доступом до даних, забезпечення цілісності даних та інших функцій, які зазвичай присутні у повноцінних базах даних.

Тому, використання XML як бази даних може бути виправданим для невеликих CRM-систем з обмеженим обсягом даних та невисокими вимогами до продуктивності. Для більших та складніших систем рекомендується використовувати спеціалізовані бази даних, які забезпечують більш високу продуктивність, масштабованість та надійність.

Вибір C# (з ASP.NET Core MVC) або Node.js (з Express.js) залежить від конкретних вимог проекту, досвіду команди розробників та інших факторів. Обидві технології є потужними та гнучкими інструментами для створення back-end частини веб-сервісів. Використання XML як бази даних може бути доцільним у певних випадках, але для більших проектів або проектів з високими вимогами до продуктивності та масштабованості рекомендується використовувати більш спеціалізовані бази даних, такі як PostgreSQL або MySQL.

Таким чином, вибір цих технологій є оптимальним рішенням для створення веб-сайту спортзалу, який буде відповідати всім сучасним вимогам, забезпечуючи високу продуктивність, зручність використання та широкі можливості для розвитку та розширення функціональності.

3.2. Опис інтерфейсу та функціональних можливостей веб-сайту.

Проаналізувавши декілька сайтів, було помічено, що розробники використовують досить багато тексту і користувач від цього губиться, оскільки не розуміє, яка інформація є для нього важлива. Тому у своїй роботі я приділила велику увагу інтерфейсу та працювала за таким принципом, щоб на сайті була виключно необхідна інформація та збалансована палітра кольорів.



Рис.3.2.1. Головна сторінка спортивного залу PowerGym

Коли користувач потрапляє на сайт спортзалу, то одразу його зустрічає головна сторінка, на якій розміщено мотиваційних слоган, а знизу кнопка для більшого ознайомлення з сайтом. Зверху розміщена панель для швидкої навігації по сайту. На ній розміщено:

- Головна сторінка веб-сайту
- Про нас
- Абонементи
- Розклад
- Контакти
- Вхід
- Посилання на соціальні мережі

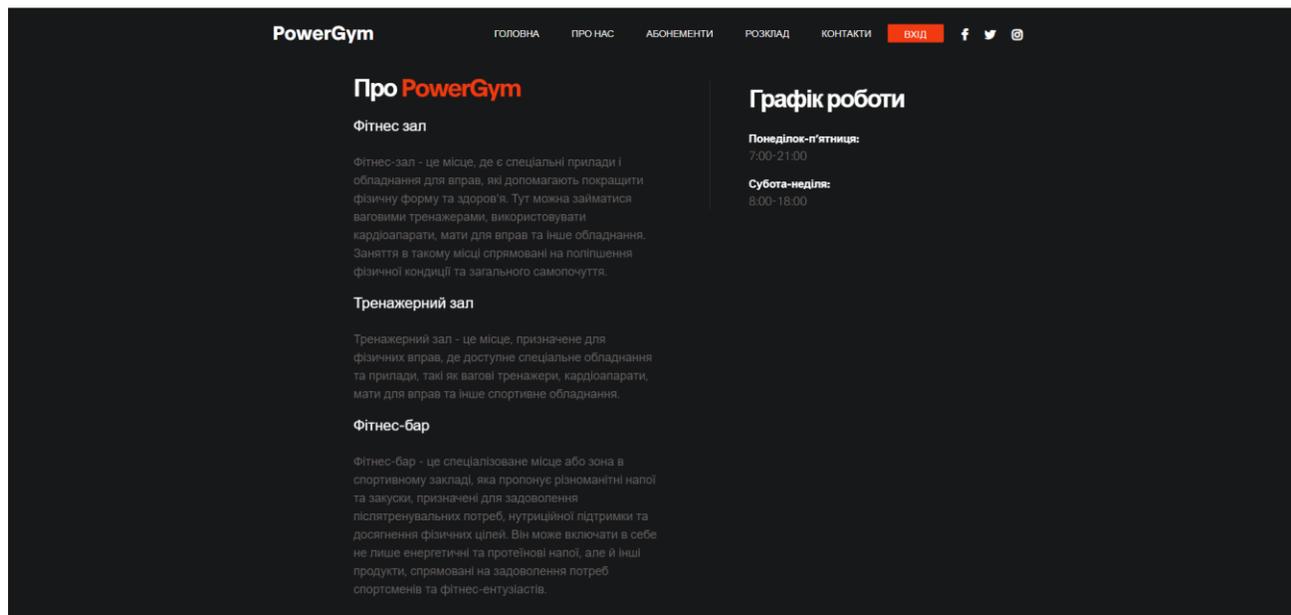


Рис.3.2.2. Інформаційний блок “Про нас” спортивного залу PowerGym

У розділі “Про нас” розташована інформація про фітнес-зал, тренажерний зал, а також фітнес-бар. Це зроблено з тією метою, щоб користувач розумівся в

термінології. Тоді людина краще розуміє, що саме їй необхідно, коли захоче скористуватися послугами спортзалу. До того ж розміщено графік роботи.

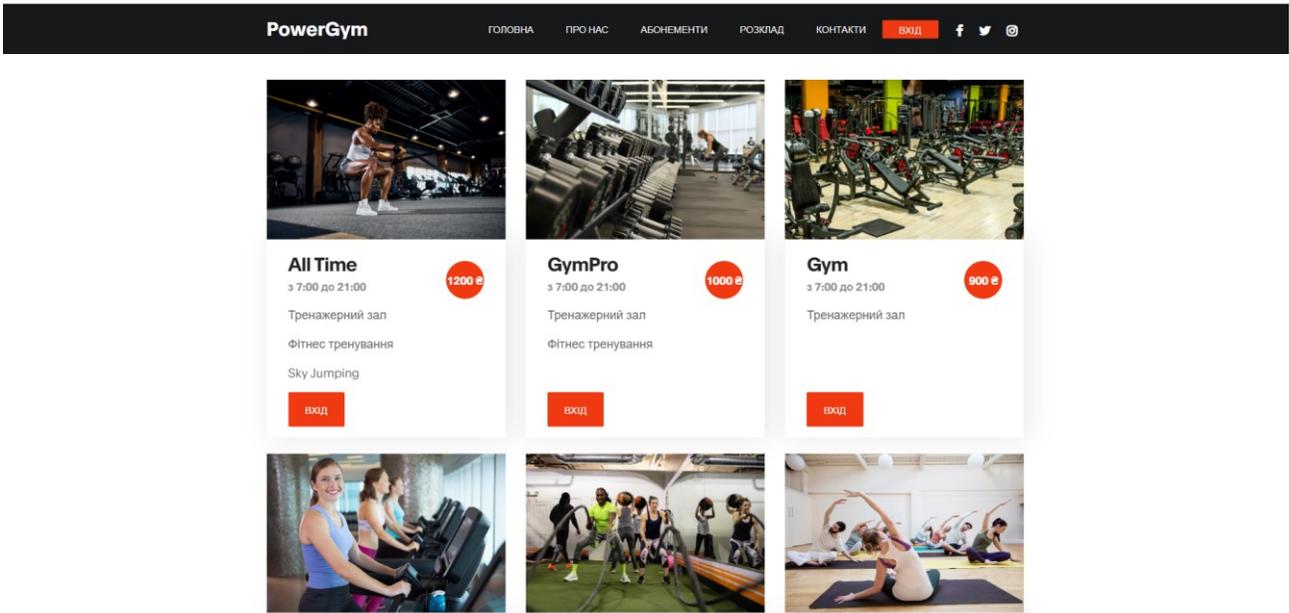


Рис.3.2.3. Розділ “Абонементи” спортивного залу PowerGym

У розділі “Абонементи” розташована інформація про можливі абонементи, які пропонує спортивний зал. До кожного абонементу є інформація про:

- Назву абонементу
- Дія абонементу у певні години та дні
- Послуги, які дає конкретний абонемент
- Ціну

Варто зазначити, що користувач може користуватись лише одним абонементом. Якщо є активний абонемент, то користувач не може підписатись на інший. Лише по закінченню попереднього абонементу.

	ПН	ВТ	СР	ЧТ	ПТ	СБ
7:00	Фітнес 7:00 - 7:45	Кардіо 7:00 - 7:45	Фітнес 7:00 - 7:45	Кардіо 7:00 - 7:45	Фітнес 7:00 - 7:45	
8:00	Sky Jumping 8:00 - 8:45	Фітнес 8:00 - 8:45	Кросфіт 8:00 - 8:45	Sky Jumping 8:00 - 8:45	Кросфіт 8:00 - 8:45	Кросфіт 8:00 - 8:45
9:00		Sky Jumping 9:00 - 9:45		Sky Jumping 9:00 - 9:45		Фітнес 9:00 - 9:45
18:00	Кардіо 18:00 - 18:45	Sky Jumping 18:00 - 18:45		Sky Jumping 18:00 - 18:45	Фітнес 18:00 - 18:45	
19:00	Фітнес 19:00 - 19:45 PM	Кросфіт 19:00 - 19:45	Кардіо 19:00 - 19:45	Кросфіт 19:00 - 19:45	Кардіо 19:00 - 19:45	Кардіо 19:00 - 19:45
20:00	Sky Jumping 20:00 - 20:45	Фітнес 20:00 - 20:45	Кросфіт 20:00 - 20:45	Фітнес 20:00 - 20:45	Sky Jumping 20:00 - 20:45	Кросфіт 20:00 - 20:45

Рис.3.2.4. Розділ “Розклад” спортивного залу PowerGym

У розділі “Розклад” розташована інформація про групові заняття, які надає спортивний клуб. Розклад сформований таким чином, що вся інформація розписана по дням, а також зазначається години. Це є досить зручно, оскільки користувач може з легкістю орієнтуватись та розподіляти свій день. Це робить спортивний зал більш зручним для суспільства, адже задоволеність клієнта понад усе. Це мотивує робити зал більш краще, а людей відвідувати для них чудовий та зручний спортивний зал.

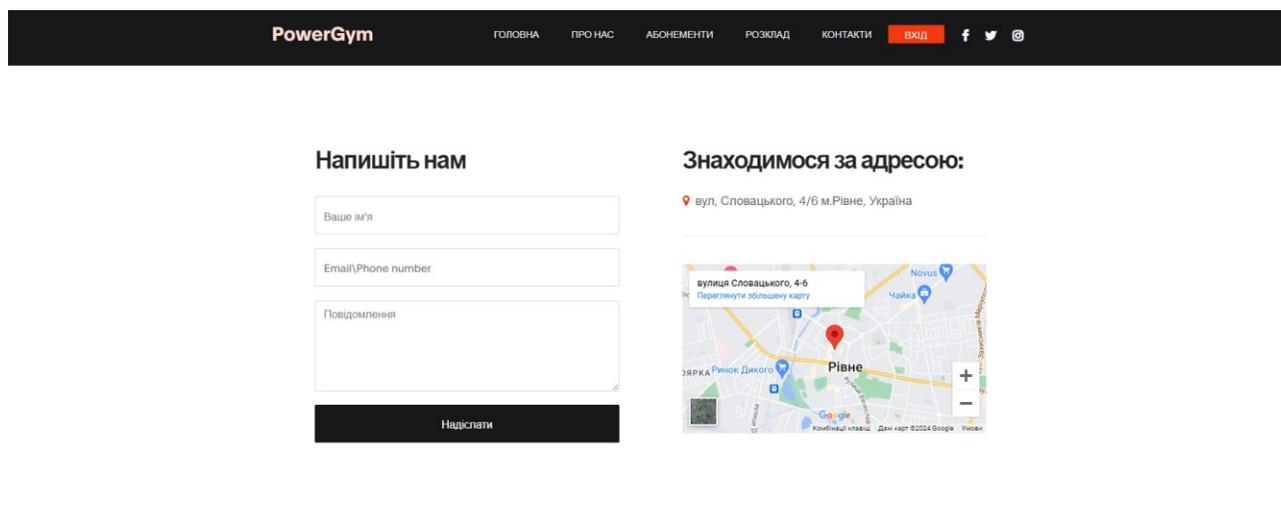
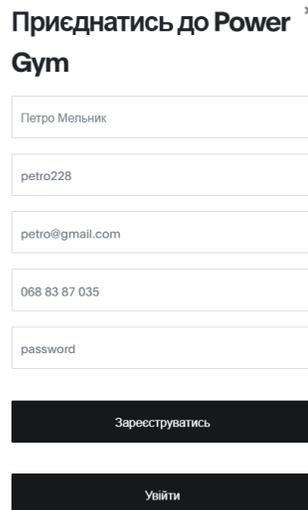


Рис.3.2.5. Розділ “Контакти” спортивного залу PowerGym

У розділі “Контакти” розташована інформація про адресу розташування спортивного залу, а також контактні дані. Розташований спосіб комунікації: поля для відправки повідомлення до спортивного залу у службу підтримки.

На даному сайті є можливість створити свій обліковий запис. Через який можна підписатись на абонемент або відписатись від нього. А також в особовому кабінеті відображається таймер, який відображає скільки часу дійсний абонемент у клієнта.

Щоб зареєструватись на сайті необхідно ввести необхідну інформацію у відповідну форму (рис.3.2.6.)



Приєднатись до Power ^x
Gym

Петро Мельник

petro228

petro@gmail.com

068 83 87 035

password

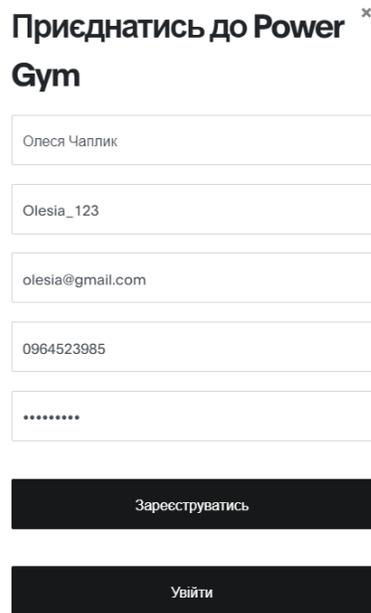
Зареєструватись

Увійти

Рис.3.2.6. Форма реєстрації для облікового запису спортивного залу PowerGym

3.3. Тестові приклади роботи системи, аналіз результатів.

Щоб стати клієнтом спортивного клубу PowerGym необхідно створити обліковий запис. Для цього заповнюємо поля у формі реєстрації, яка знаходиться, натиснувши кнопку “Вхід” (рис. 3.3.1.). Система вже показує, що саме необхідно вписати. Після того як все вказали ми натискаємо “Зареєструватись” і стаємо повноцінним клієнтом спортивного залу PowerGym.



Присєднатись до Power Gym

Олеся Чаплик

Olesia_123

olesia@gmail.com

0964523985

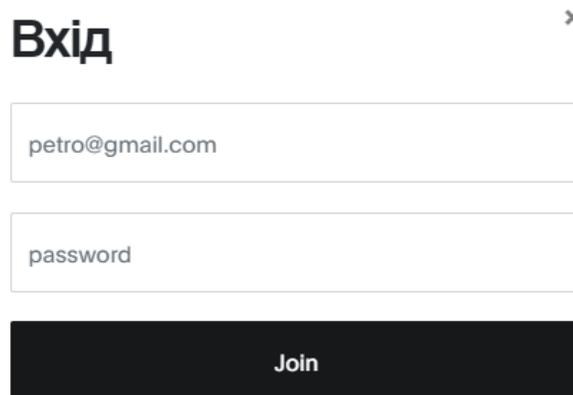
.....

Зареєструватись

Увійти

Рис.3.3.1. Етап реєстрації на сайті спортивного залу PowerGym

Варто зазначити, що якщо поля невірно заповнено, то система сповістить про помилку і попросить ввести коректно. Якщо поля заповнено вірно система направить у головне меню. Потім необхідно натиснути кнопку “Вхід”, далі натискаємо “Увійти” і вказуємо наступні поля, які просить система на рис. 3.3.2.



Вхід

petro@gmail.com

password

Join

Рис.3.3.2. Форма входу на сайті спортивного залу PowerGym

Після успішного входу в обліковий запис у верхній панелі швидкого доступу з'являється кнопка “Аккаунт” (рис.3.3.3.), через яку ми можемо переглянути інформацію про абонемент.

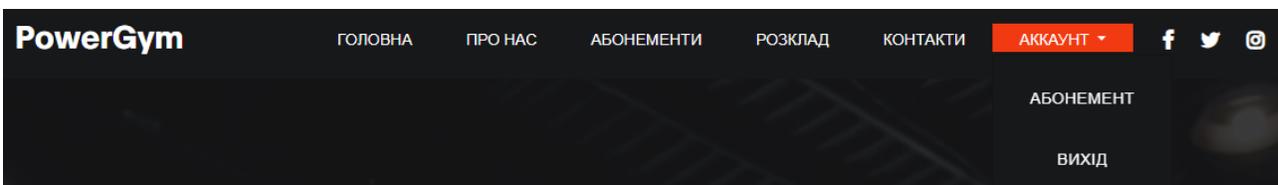


Рис.3.3.3. Панель швидкого доступу з врахуванням облікового запису на сайті спортивного залу PowerGym

Коли настикається кнопка “Аккаунт”, то сайт інформує користувача, про те, що він не має ніякого абонементу. Для того щоб активувати його необхідно перейти у розділ “Абонементи” та обрати абонемент.

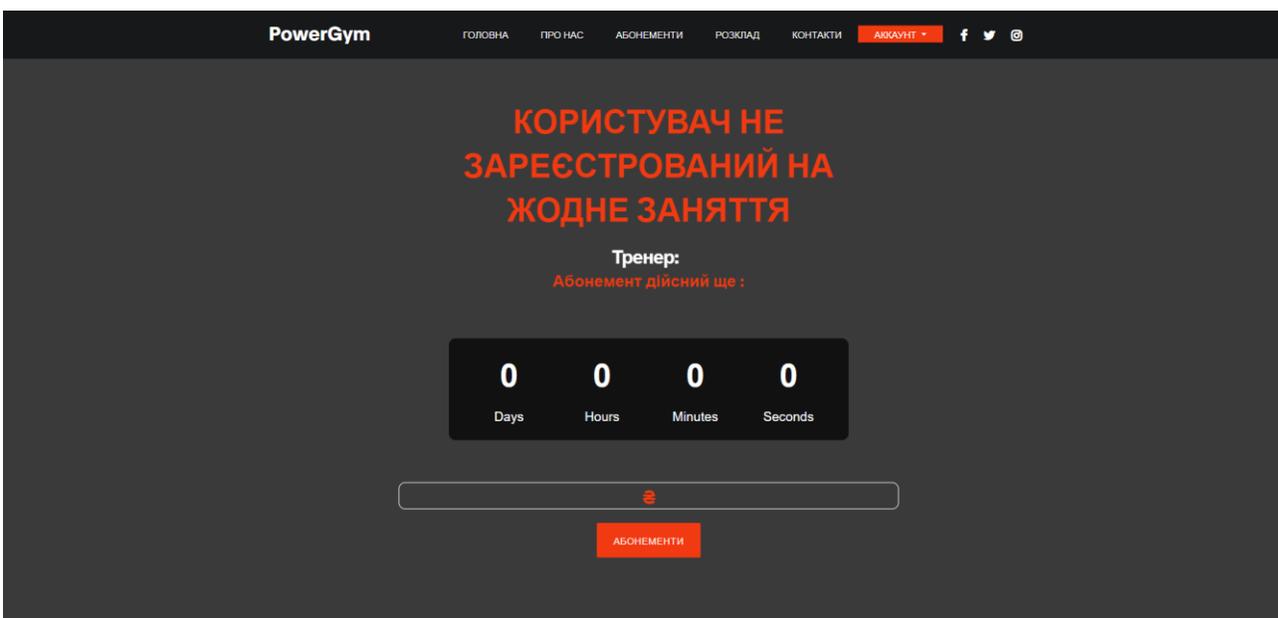


Рис.3.3.4. Інформування користувача про відсутність абонементу на сайті спортивного залу PowerGym

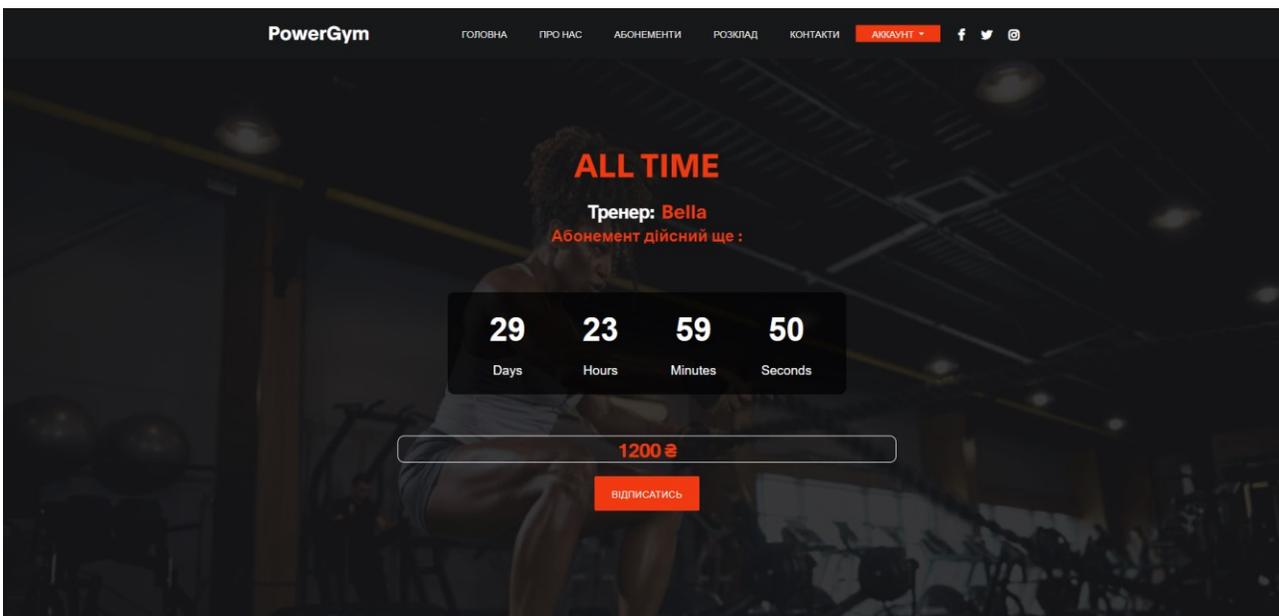


Рис.3.3.5. Інформування користувача про активний абонемент на сайті спортивного залу PowerGym

Після того як людина підписалася на абонемент у обліковому записі з'являється інформація про абонемент, який попередньо був обраний. Вказано тренера, який закріплюється за клієнтом та час тривання абонементу – 1 місяць. Знизу розміщена вартість абонементу та кнопка “Відписатись”.

Варто зазначити, що система сповіщує користувача, якщо він підписується на абонемент, тривалість попереднього якого ще не закінчилась. Сповіщення інформує користувача про некоректну дію (рис.3.3.5.)

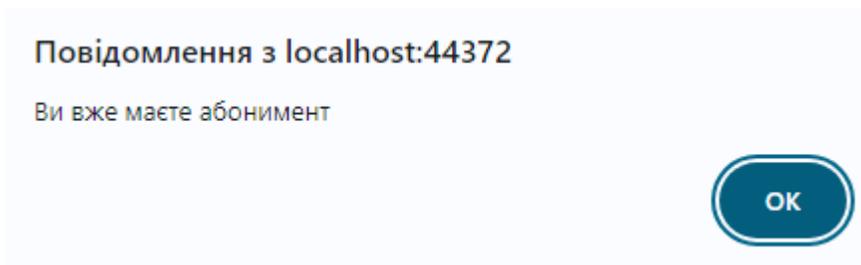


Рис.3.3.6. Інформативне сповіщення про абонемент на сайті спортивного залу PowerGym

Аналіз можливостей сайту показує, що сервіс успішно справляється з поставленими завданнями та відрізняється зручністю у використанні. Розробники створили доступний та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам швидко освоїтися на платформі.

Сайт надає вичерпну інформацію про зал, ефективно взаємодіє з відвідувачами та пропонує зручні функції. Користувачі можуть легко створити обліковий запис, підібрати абонемент, що відповідає їхнім індивідуальним потребам та цілям.

Важливою перевагою є наявність різноманітних способів комунікації зі службою підтримки та відділами зворотного зв'язку. Це дозволяє оперативно отримувати відповіді на питання, вирішувати проблеми та отримувати консультації від фахівців.

3.4. Рекомендації щодо впровадження та обґрунтування ефективності.

Впровадження спеціалізованої CRM-системи для спортзалу є стратегічно важливим кроком для підвищення ефективності бізнесу, покращення якості обслуговування клієнтів та збільшення прибутковості. Для досягнення максимальної ефективності від впровадження CRM-системи, важливо врахувати ряд рекомендацій, які стосуються як технічних аспектів розробки, так і стратегічних питань розвитку бізнесу.

Рекомендації щодо впровадження та обґрунтування ефективності, аби даний веб-сайт працював більш злагоджено та ефективно:

- **Багатомовність:** для спортзалу, який прагне залучити широку аудиторію, важливо забезпечити багатомовність. Це дозволить клієнтам з різних країн та мовних груп комфортно користуватися сервісом, обирати мову інтерфейсу,

отримувати сповіщення та інформацію рідною мовою. Це сприятиме підвищенню лояльності клієнтів та розширенню географії бізнесу.

- **Інтеграція з іншими сервісами:** для оптимізації бізнес-процесів та підвищення ефективності роботи спортзалу рекомендується забезпечити інтеграцію crm-системи та з іншими сервісами, такими як системи бухгалтерського обліку, email-маркетингу, sms-розсилок, календарі тощо. Це дозволить уникнути дублювання даних, автоматизувати рутинні операції та отримати більш повну картину взаємодії з клієнтами. Наприклад, інтеграція з платіжними системами дозволить автоматизувати процес оплати абонементів та послуг, а інтеграція з календарем допоможе співробітникам планувати робочий час та розклад занять.
- **Перехід на sql бази даних при масштабуванні:** якщо планується масштабування проекту та збільшення кількості користувачів, рекомендується перейти від використання xml до sql баз даних (postgresql, mysql). Sql бази даних забезпечують більш високу продуктивність, надійність та масштабованість, а також пропонують широкий набір інструментів для управління даними, забезпечення їх цілісності та безпеки. Перехід на sql базу даних дозволить crm-системі ефективно працювати з великими обсягами даних та забезпечувати високу швидкість обробки запитів навіть при значному збільшенні навантаження.

Впровадження цих рекомендацій дозволить створити CRM-систему, яка буде не лише відповідати поточним потребам спортзалу, але й матиме потенціал для подальшого розвитку та масштабування. Багатомовність забезпечить доступність сервісу для широкої аудиторії, інтеграція з іншими системами оптимізує бізнес-процеси, а перехід на SQL базу даних забезпечить високу продуктивність та надійність при збільшенні обсягу даних та кількості користувачів.

В результаті, впровадження цих рекомендацій призведе до:

- **Підвищення ефективності роботи співробітників:** завдяки автоматизації рутинних операцій та зручному доступу до необхідної інформації.
- **Підвищення задоволеності клієнтів:** завдяки персоналізованому підходу, зручному інтерфейсу та багатомовності.
- **Збільшення прибутковості бізнесу:** завдяки оптимізації маркетингових кампаній, підвищенню лояльності клієнтів та розширенню географії бізнесу.
- **Підвищення конкурентоспроможності:** завдяки використанню сучасних технологій та інноваційних підходів до управління відносинами з клієнтами.

В цілому, впровадження запропонованих рекомендацій дозволить створити CRM-систему, яка стане потужним інструментом для розвитку та успіху спортзалу.

ВИСНОВКИ

У ході виконання дипломної роботи було успішно розроблено та впроваджено веб-сервіс для автоматизації управління відносинами з клієнтами спортзалу. Цей сервіс надає комплексний підхід до взаємодії з клієнтами, охоплюючи всі етапи їхнього життєвого циклу, від першого контакту до утримання та розвитку довгострокових відносин.

Веб-сервіс забезпечує ефективне управління клієнтською базою, дозволяючи зберігати та оновлювати інформацію про клієнтів, відстежувати їхні відвідування, уподобання та історію взаємодії зі спортзалом. Автоматизація процесів, таких як реєстрація, оплата, бронювання тренувань та розсилка повідомлень, значно спрощує роботу персоналу та підвищує ефективність обслуговування клієнтів.

Впровадження веб-сервісу дозволило спортзалу підвищити рівень задоволеності клієнтів, надаючи їм зручні інструменти для самостійного управління своїм абонементом, бронювання тренувань та отримання інформації про послуги спортзалу. Крім того, сервіс надає можливість персоналізувати комунікацію з клієнтами, пропонуючи їм індивідуальні пропозиції та акції, що сприяє підвищенню їхньої лояльності та утриманню.

Аналіз результатів впровадження веб-сервісу показав позитивний вплив на ключові показники ефективності спортзалу. Зокрема, спостерігається збільшення кількості нових клієнтів, підвищення рівня утримання клієнтів, зростання середнього чека та загального доходу спортзалу. Крім того, автоматизація процесів дозволила скоротити витрати на обслуговування клієнтів та підвищити продуктивність персоналу.

Перспективи подальшого розвитку веб-сервісу включають розширення його функціоналу, інтеграцію з іншими системами спортзалу, такими як системи відеоспостереження та контролю доступу, а також використання аналітичних

інструментів для отримання більш глибоких інсайтів щодо поведінки клієнтів та ефективності маркетингових кампаній.

Впровадження нових технологій, таких як штучний інтелект та машинне навчання, дозволить ще більше персоналізувати взаємодію з клієнтами, пропонуючи їм індивідуальні рекомендації щодо тренувань та харчування, а також прогнозувати їхні потреби та вподобання.

Важливим напрямком розвитку є також розробка мобільного додатку, який надасть клієнтам зручний доступ до функціоналу веб-сервісу з їхніх смартфонів. Це дозволить їм легко бронювати тренування, відстежувати свої результати, отримувати персоналізовані рекомендації та спілкуватися з тренерами та іншими клієнтами спортзалу.

У цілому, розроблений веб-сервіс має значний потенціал для подальшого розвитку та вдосконалення. Його використання дозволить спортзалу не лише покращити якість обслуговування клієнтів та підвищити їхню лояльність, але й відкрити нові можливості для зростання та розвитку бізнесу в умовах сучасного цифрового світу.

Таким чином, розроблений та впроваджений веб-сервіс є ефективним інструментом для автоматизації управління відносинами з клієнтами спортзалу. Він допомагає покращити якість обслуговування, підвищити лояльність клієнтів та збільшити прибутковість бізнесу. Подальший розвиток та вдосконалення сервісу, з урахуванням потреб та побажань клієнтів, дозволить спортзалу залишатися конкурентоспроможним та досягати нових успіхів на ринку.

СПИСОК ЛІТЕРАТУРИ

1. Розробка веб-сайтів: ключові етапи й аспекти. URL:<https://seo-tiko.com/articles/website-development-key-stages-and-aspects/> (дата звернення 30.01.2024)
2. Від інновацій до успіху: Ключові технології для сучасного бізнесу. URL:https://sitniks.ua/blog_post/vid-inovtziy-do-uspyhu/ (дата звернення 15.02.2024)
3. IT і бізнес: Як технології впливають на розвиток сучасних підприємств. URL:<https://lemon.school/blog/it-i-biznes-yak-tehnologiyi-vplyvayut-na-rozvytok-suchasnyh-pidpryyemstv> (дата звернення 15.02.2024)
4. XML для новачків. URL:<https://support.microsoft.com/uk-ua/office/xml-%D0%B4%D0%BB%D1%8F-%D0%BD%D0%BE%D0%B2%D0%B0%D1%87%D0%BA%D1%96%D0%B2-a87d234d-4c2e-4409-9cbc-45e4eb857d44> (дата звернення 24.02.2024)
5. Введення в XML. URL:https://w3schoolsua.github.io/xml/xml_what.html#gsc.tab=0 (дата звернення 26.02.2024)
6. Синхронізація товарів через XML. URL:<https://keepincrm.com/synchronization-xml> (дата звернення 05.04.2024)
7. Impact of Gym Management Software on the Fitness Industry. URL:<https://www.bookingninja.com/blog/impact-of-gym-management-software-on-the-fitness-industry> (дата звернення 15.04.2024)

ДОДАТКИ

Додаток А

Код AccountController.cs

```

using Power_Gym.Models;
using Power_Gym.Data;
using System.Web.Mvc;
using System.Xml;
using System;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using System.Web.Security;
using System.Xml.Linq;
using System.Linq;
using System.Collections.Generic;
using System.Web.UI.WebControls;
using System.IO;
using System.Net.Mail;
using System.Net;
using System.Globalization;

namespace Power_Gym.Controllers
{
    public class AccountController : Controller
    {
        private readonly UserManager<ApplicationUser> _userManager;
        private readonly SignInManager<ApplicationUser> _signInManager;
        private readonly ApplicationDbContext _context;
        private readonly string _xmlFilePath =
Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "data.xml");

        public AccountController(UserManager<ApplicationUser> userManager,
SignInManager<ApplicationUser> signInManager, ApplicationDbContext context)
        {
            _userManager = userManager;
            _signInManager = signInManager;
            _context = context;
        }
        public AccountController()
        {
        }
        [HttpPost]
        [ValidateAntiForgeryToken]
        //Метод реєстрації
        public ActionResult Register(RegisterViewModel model)
        {
            if (ModelState.IsValid)
            {
                bool isEmailUnique = SaveClientToXml(model);

                if (!isEmailUnique)
                {
                    TempData["Message"] = "Email address is already in use.";
                }
            }
        }
    }
}

```

```

        Session["CurrentUser"] = model.Email;
        CheckCurrentUser();
        return View(model);
    }

    }
    return View(model);
}
[HttpPost]
[ValidateAntiForgeryToken]
//Метод виходу з акаунта
public ActionResult Logout()
{
    // Очищення сесії
    Session.Clear();
    return RedirectToAction("Index", "Home"); // Перенаправлення на домашню
сторінку
}
[HttpPost]
[ValidateAntiForgeryToken]
//Метод для входу в акаунт
public ActionResult Login(LoginViewModel model, EnrollmentData
enrolledClasses, string className)
{
    XmlDocument xmlDoc = new XmlDocument();
    xmlDoc.Load(_xmlFilePath);
    if (Session["CurrentUser"] != null)
    {
        if(className != null)
        {
            DeleteClassFromXml(className, xmlFilePath);
        }
        xmlDoc.Load(_xmlFilePath);
        enrolledClasses = GetEnrolledClasses(xmlDoc,
Session["CurrentUser"].ToString());
        CheckCurrentUser();
        return View(enrolledClasses);
    }
    if (ModelState.IsValid)
    {
        bool isAuthenticated = AuthenticateUser(model.Email,
model.Password);
        if (isAuthenticated)
        {
            // Встановлення інформації про користувача у сеансі
            Session["CurrentUser"] = model.Email;
            enrolledClasses = GetEnrolledClasses(xmlDoc,
Session["CurrentUser"].ToString());
            CheckCurrentUser();
            return View(enrolledClasses);
        }
        else
        {
            TempData["Message"] = "User name or password is not correct.";
            return RedirectToAction("Index", "Home");
        }
    }
}

```

```

        return View();
    }

    public ActionResult Index()
    {
        if (Session["CurrentUser"] != null)
        {
            ViewBag.IsLoggedIn = true; // Передача інформації в перегляд
        }
        else
        {
            ViewBag.IsLoggedIn = false; // Передача інформації в перегляд
        }

        return View();
    }

    public ActionResult Enroll(string _class, string price, string
trainer_name, string image)
    {
        // Перевірка, чи користувач увійшов у систему
        if (Session["CurrentUser"] != null)
        {
            // Отримання email поточного користувача з сесії
            var userEmail = Session["CurrentUser"].ToString();

            // Отримання даних користувача з бази даних XML за його email
            var user = GetUserFromXML(userEmail);

            if (user != null)
            {
                DateTime date = new DateTime();
                date = DateTime.Now;
                date = date.AddMonths(1);

                // Створення екземпляра EnrollmentData та заповнення його
                var enrollmentData = new
EnrollmentData(_class, price, image, trainer_name, user, date);

                // Зчитуємо XML файл, де зберігаються дані про записи на
заняття

                XmlDocument xmlDoc = new XmlDocument();
                xmlDoc.Load(_xmlFilePath);
                XmlNode rootNode = xmlDoc.SelectSingleNode("users");

                if (rootNode != null)
                {
                    // Перевіряємо, чи даний користувач вже записаний на це
заняття

                    XmlNode userNode =
rootNode.SelectSingleNode("user[@email='" + enrollmentData.User.Email + "']");
                    if (userNode != null)
                    {
                        // Перевіряємо, чи користувач не має вже запису для
данного заняття

                        XmlNode classEnrollmentNode =
userNode.SelectSingleNode("classEnrollment");
                        if (classEnrollmentNode == null)
                        {

```

```

        // Створюємо елемент 'classEnrollment' для
зберігання інформації про запис на заняття
        classEnrollmentNode =
xmlDoc.CreateElement("classEnrollment");

        // Додаємо атрибути до елемента 'classEnrollment'
CreatAttribute("ClassName",
enrollmentData.ClassName, classEnrollmentNode, xmlDoc);
CreatAttribute("Price", enrollmentData.Price,
classEnrollmentNode, xmlDoc);
CreatAttribute("Trainer", enrollmentData.Trainer,
classEnrollmentNode, xmlDoc);
CreatAttribute("Image", enrollmentData.Image,
classEnrollmentNode, xmlDoc);
CreatAttribute("Date",
enrollmentData.DateEnroll.ToString(), classEnrollmentNode, xmlDoc);
// Додаємо елемент 'classEnrollment' до
відповідного 'userNode'
        userNode.AppendChild(classEnrollmentNode);

        // Зберігаємо оновлений XML-документ
xmlDoc.Save(_xmlFilePath);
    }
    else
    {
        TempData["Message"] = "Ви вже маєте абонимент";
    }
}

// Перенаправлення на головну сторінку після успішного запису
на заняття
return RedirectToAction("Index", "Home");
}

// Перенаправлення на сторінку входу, якщо користувач не увійшов у
систему або дані не вдалося отримати
return RedirectToAction("Index", "Home");
}
public ActionResult SendMessage(string Email, string Name, string Message)
{
    // Налаштування SMTP-сервера Mail.ru
    string smtpServer = "smtp.gmail.com";
    int smtpPort = 587;
    string smtpUsername = "olesyachaplyk1818@gmail.com";
    string smtpPassword = "blsi qgpj julj baip";

    // Створюємо об'єкт клієнта SMTP
    using (SmtpClient smtpClient = new SmtpClient(smtpServer, smtpPort))
    {
        // Налаштування аутентифікації
        smtpClient.Credentials = new NetworkCredential(smtpUsername,
smtpPassword);
        smtpClient.EnableSsl = true;

        using (MailMessage mailMessage = new MailMessage())
        {
            mailMessage.From = new MailAddress(smtpUsername);

```

```

        mailMessage.To.Add(smtpUsername); // Укажіть адрес отримувача
        mailMessage.Subject = Email;
        mailMessage.Body = $"Name: {Name}\nMessage: \n{Message}";
        try
        {
            // Відправляємо повідомлення
            smtpClient.Send(mailMessage);
            TempData["Message"] = "Message Send";
        }
        catch (Exception ex)
        {
            TempData["Message"] = $"Error: {ex.Message}";
        }
    }
}

return RedirectToAction("Index", "Home");
}

public void DeleteClassFromXml(string className, string xmlFilePath)
{
    // Зчитуємо XML-файл
    XmlDocument xmlDoc = new XmlDocument();
    xmlDoc.Load(xmlFilePath);
    string s = Session["CurrentUser"].ToString();
    XmlNodeList userNodes = xmlDoc.SelectNodes($"//user[@email='{s}']");

    if (userNodes.Count > 0)
    {
        // Знаходимо всі елементи "classEnrollment", що мають атрибут
        // ClassName зі значенням, рівним className
        XmlNodeList classEnrollmentNodes =
            xmlDoc.SelectNodes($"//classEnrollment[@ClassName='{className}']");

        // Видаляємо кожен знайдений елемент "classEnrollment" з XML
        foreach (XmlNode node in classEnrollmentNodes)
        {
            node.ParentNode.RemoveChild(node);
        }
        // Зберігаємо зміни у XML-файлі
        xmlDoc.Save(xmlFilePath);
    }
}

private void CreatAttribute(string attribut, string enrollmentData, XmlNode
classEnrollmentNode, XmlDocument xmlDoc)
{
    XmlAttribute classNameAttr = xmlDoc.CreateAttribute(attribut);
    classNameAttr.Value = enrollmentData;
    classEnrollmentNode.Attributes.Append(classNameAttr);
}

private List<string> GetEnrolledUserEmails(string className)
{
    List<string> enrolledEmails = new List<string>();

    // Зчитуємо XML файл, де зберігаються дані про записи на заняття
    XmlDocument xmlDoc = new XmlDocument();
    xmlDoc.Load(_xmlFilePath);
}

```

```

        // Отримуємо вузол заняття зі списком записаних користувачів
        XmlNode classNode =
xmlDoc.SelectSingleNode($"enrollmentData[ClassName='{className}']");

        if (classNode != null)
        {
            // Отримуємо список елементів "email" з вузла заняття
            XmlNodeList emailNodes =
classNode.SelectNodes("ClientEmails/email");

            if (emailNodes != null)
            {
                foreach (XmlNode emailNode in emailNodes)
                {
                    enrolledEmails.Add(emailNode.InnerText); // Додаємо емейл
до списку
                }
            }

            return enrolledEmails;
        }
        //Метод для перевірки введених даних при вході
        private bool AuthenticateUser(string email, string password)
        {
            XmlDocument xmlDoc = new XmlDocument();
            xmlDoc.Load(_xmlFilePath);

            XDocument xdoc = XDocument.Load(new XmlNodeReader(xmlDoc));

            var userNode = xdoc.Descendants("user")
                .FirstOrDefault(u => (string)u.Attribute("email") == email
&& (string)u.Attribute("password") == password);

            return userNode != null;
        }
        // Метод для перевірки унікальності електронної адреси та збереження
користувача
        private bool SaveClientToXml(RegisterViewModel model)
        {
            XmlDocument xmlDoc = new XmlDocument();
            xmlDoc.Load(_xmlFilePath);

            XmlNode rootNode = xmlDoc.SelectSingleNode("users");
            if (rootNode == null)
            {
                rootNode = xmlDoc.CreateElement("users");
                xmlDoc.AppendChild(rootNode);
            }

            // Перевірка на унікальність електронної адреси
            bool isEmailUnique = IsEmailUnique(xmlDoc, model.Email);
            if (!isEmailUnique)
            {
                return false; // Повертаємо false, якщо електронна адреса вже
використовується
            }

            Client client = new Client

```

```

    {
        FullName = model.FullName,
        UserName = model.UserName,
        Email = model.Email,
        PhoneNumber = model.PhoneNumber,
        Password = model.Password
    };
    XmlNode userNode = client.ToXmlNode(xmlDoc);
    rootNode.AppendChild(userNode);

    xmlDoc.Save(_xmlFilePath);
    return true; // Повертаємо true, якщо користувач успішно збережений
}
private bool IsEmailUnique(XmlDocument xmlDoc, string email)
{
    XmlNodeList userNodes = xmlDoc.SelectNodes("//user[@email='" + email +
    "'"]);
    return userNodes.Count == 0;
}

private Client GetUserFromXML(string email)
{
    Client user = null;

    try
    {
        // Логіка для читання даних з XML-файлу і знаходження користувача
        за email
        XmlDocument xmlDoc = XmlDocument.Load(_xmlFilePath);

        var query = xmlDoc.Descendants("user")
            .FirstOrDefault(u => u.Attribute("email")?.Value
            == email);

        if (query != null)
        {
            // Знайдено користувача, створюємо об'єкт User з відповідними
            даними
            user = new Client
            {
                UserName = query.Attribute("userName")?.Value,
                Email = query.Attribute("email")?.Value,
                PhoneNumber = query.Attribute("phoneNumber")?.Value,
                FullName = query.Attribute("fullName")?.Value,
                Password = query.Attribute("password")?.Value
            };
        }
    }
    catch (Exception ex)
    {
        // Обробка помилок читання XML або інших винятків
        Console.WriteLine("Error reading XML data: " + ex.Message);
    }

    return user;
}
private EnrollmentData GetEnrolledClasses(XmlDocument xmlDoc, string
userEmail)
{

```

```

        EnrollmentData enrollment= new EnrollmentData ();

        // Знаходимо користувача за його email
        XmlNode userNode = xmlDoc.SelectSingleNode("//user[@email='" +
userEmail + "']");
        if (userNode != null)
        {
            // Отримуємо список занять, на які записаний користувач
            foreach (XmlNode enrollmentNode in
userNode.SelectNodes("classEnrollment"))
            {

                string className =
enrollmentNode.Attributes["ClassName"].Value;
                string Date = enrollmentNode.Attributes["Date"].Value;
                string Price = enrollmentNode.Attributes["Price"].Value;
                string Image = enrollmentNode.Attributes["Image"].Value;
                string Trainer = enrollmentNode.Attributes["Trainer"].Value;
                enrollment = new EnrollmentData(className,Price,Image,Trainer,
GetUserFromXML(userEmail), DateTime.Parse(Date));
            }
        }
        else
        {
            TempData["Message"] = "Email not found";
        }

        return enrollment;
    }
    private void CheckCurrentUser()
    {
        if (Session["CurrentUser"] != null)
        {
            ViewBag.IsLoggedIn = true; // Передача інформації в перегляд
        }
        else
        {
            ViewBag.IsLoggedIn = false; // Передача інформації в перегляд
        }
    }
}
}

```

Додаток Б

Код Client.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Xml;
using System.Xml.Serialization;

namespace Power_Gym.Models
{
    public class Client:ApplicationUser
    {
        [XmlElement("FullName")]
        public string FullName { get; set; }

        [XmlElement("UserName")]
        public string UserName { get; set; }

        [XmlElement("Email")]
        public string Email { get; set; }

        [XmlElement("PhoneNumber")]
        public string PhoneNumber { get; set; }

        [XmlIgnore]
        public string Password { get; set; }

        public XmlNode ToXmlNode(XmlDocument xmlDoc)
        {
            XmlNode userNode = xmlDoc.CreateElement("user");

            XmlAttribute attribute = xmlDoc.CreateAttribute("userName");
            attribute.Value = UserName;
            userNode.Attributes.Append(attribute);

            attribute = xmlDoc.CreateAttribute("email");
            attribute.Value = Email;
            userNode.Attributes.Append(attribute);

            attribute = xmlDoc.CreateAttribute("phoneNumber");
            attribute.Value = PhoneNumber;
            userNode.Attributes.Append(attribute);

            attribute = xmlDoc.CreateAttribute("password");
            attribute.Value = Password;
            userNode.Attributes.Append(attribute);

            attribute = xmlDoc.CreateAttribute("fullName");
            attribute.Value = FullName;
            userNode.Attributes.Append(attribute);

            return userNode;
        }
    }
}
```

Додаток В

Код LoginViewModel.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;

namespace Power_Gym.Models
{
    public class LoginViewModel
    {
        [Required]
        [EmailAddress]
        [Display(Name = "Електронна пошта")]
        public string Email { get; set; }

        [Required]
        [DataType(DataType.Password)]
        [Display(Name = "Пароль")]
        public string Password { get; set; }

        [Display(Name = "Запам'ятати мене")]
        public bool RememberMe { get; set; }

        public List<string> listClass { get; set; }
    }
}
```

Додаток Д

Код Register.cshtml

```

@model Power_Gym.Models.RegisterViewModel

@if (TempData["Message"] != null)
{
    <script>
        alert("@TempData["Message"]");
    </script>
}

@{
    ViewBag.Title = "Register";
}

<section class="class section" id="class">
    <div class="container">
        <div class="row">
            <!-- Yoga Class -->
            <div class="col-lg-4 col-md-6 col-12" data-aos="fade-up" data-aos-
delay="400">
                <div class="test">
                    
                    <div class="class-info">
                        <h3 class="mb-1">All Time</h3>
                        <span><strong>з 7:00 до 21:00</strong></span><br />
                        <span class="class-price">1200 &#8372</span>
                        <p class="mt-3">Тренажерний зал</p>
                        <p class="mt-3">Фітнес тренування</p>
                        <p class="mt-3">Sky Jumping</p>

                        @if (ViewBag.IsLoggedIn != null && ViewBag.IsLoggedIn)
                        {
                            <form action="/Account/Enroll" method="post">
                                @Html.AntiForgeryToken()
                                <input type="hidden" name="_class" value="All
Time">
                                <input type="hidden" name="price" value="1200">
                                <input type="hidden" name="trainer_name"
value="Bella">
                                <input type="hidden" name="image"
value="images/class/allTime-class.jpg">
                                <input type="hidden" name="date" value=" 7:00 AM -
9:00 AM">
                                <button type="submit" class="btn custom-btn bg-
color nav-link">Підписатись</button>
                            </form>
                        }
                        else
                        {
                            <button type="button" class="btn custom-btn bg-color
nav-link" data-toggle="modal" data-target="#registerForm">Вхід</button>
                        }
                    }
                }
            }
        }
    }

```

```

        </div>
    </div>
</div>
<div class="col-lg-4 col-md-6 col-12" data-aos="fade-up" data-aos-
delay="400">
    <div class="class-thumb">
        
        <div class="class-info">
            <h3 class="mb-1">GymPro</h3>
            <span><strong>з 7:00 до 21:00</strong></span><br />
            <span class="class-price">1000 &#8372</span>
            <p class="mt-3">Тренажерний зал</p>
            <p class="mt-3">Фітнес тренування</p>
            <p class="mt-3" style="visibility: hidden">as</p>

            @if (ViewBag.IsLoggedIn != null && ViewBag.IsLoggedIn)
            {
                <form action="/Account/Enroll" method="post">
                    @Html.AntiForgeryToken()
                    <input type="hidden" name="_class" value="GymPro">
                    <input type="hidden" name="price" value="1000">
                    <input type="hidden" name="trainer_name"
value="Yuliya">
                    <input type="hidden" name="image"
value="images/class/gympro-class.jpg">
                    <input type="hidden" name="date" value="7:00 AM -
9:00 AM">
                    <button type="submit" class="btn custom-btn bg-
color nav-link">Підписатись</button>
                </form>
            }
            else
            {
                <button type="button" class="btn custom-btn bg-color
nav-link" data-toggle="modal" data-target="#registerForm">Вхід</button>
            }
        </div>
    </div>
</div>
<div class="col-lg-4 col-md-6 col-12" data-aos="fade-up" data-aos-
delay="400">
    <div class="class-thumb">
        
        <div class="class-info">
            <h3 class="mb-1">Gym</h3>
            <span><strong>з 7:00 до 21:00</strong></span><br />

            <span class="class-price">900 &#8372</span>
            <p class="mt-3">Тренажерний зал</p>
            <p class="mt-3" style="visibility:hidden">as</p>
            <p class="mt-3" style="visibility: hidden">as</p>

            @if (ViewBag.IsLoggedIn != null && ViewBag.IsLoggedIn)
            {
                <form action="/Account/Enroll" method="post">
                    @Html.AntiForgeryToken()
                    <input type="hidden" name="_class" value="Gym">

```

```

        <input type="hidden" name="price" value="900">
        <input type="hidden" name="trainer_name"
value="Kolya">
        <input type="hidden" name="image"
value="images/class/gym-class.jpg">
        <input type="hidden" name="date" value="7:00 AM -
9:00 AM">
        <button type="submit" class="btn custom-btn bg-
color nav-link">Підписатись</button>
    </form>
    }
    else
    {
        <button type="button" class="btn custom-btn bg-color
nav-link" data-toggle="modal" data-target="#registerForm">Вхід</button>
    }
</div>
</div>
</div>
<div class="row mt-4">
    <div class="col-lg-4 col-md-6 col-12" data-aos="fade-up" data-aos-
delay="400">
        <div class="class-thumb">
            
            <div class="class-info">
                <h3 class="mb-1">GymDay</h3>
                <span><strong>з 7:00 до 16:00</strong></span><br />
                <span class="class-price">800 &#8372</span>
                <p class="mt-3">Тренажерний зал</p>
                <p class="mt-3" style="visibility: hidden">as</p>
                <p class="mt-3" style="visibility: hidden">as</p>
                @if (ViewBag.IsLoggedIn != null && ViewBag.IsLoggedIn)
                {
                    <form action="/Account/Enroll" method="post">
                        @Html.AntiForgeryToken()
                        <input type="hidden" name="_class" value="GymDay">
                        <input type="hidden" name="price" value="800">
                        <input type="hidden" name="trainer_name"
value="Ivanna">
                        <input type="hidden" name="image"
value="images/class/gymday-class.jpg">
                        <input type="hidden" name="date" value="з 7:00 до
16:00">
                        <button type="submit" class="btn custom-btn bg-
color nav-link">Підписатись</button>
                    </form>
                }
                else
                {
                    <button type="button" class="btn custom-btn bg-color
nav-link" data-toggle="modal" data-target="#registerForm">Вхід</button>
                }
            </div>
        </div>
    </div>
</div>
<div class="col-lg-4 col-md-6 col-12" data-aos="fade-up" data-aos-
delay="400">

```

```

        <div class="class-thumb">
            
            <div class="class-info">
                <h3 class="mb-1">Fitness</h3>
                <span><strong>За попереднім записом</strong></span><br />
                <span class="class-price">700 &#8372</span>
                <p class="mt-3">Груповий фітнес</p>
                <p class="mt-3">Sky Jumping</p>
                <p class="mt-3" style="visibility: hidden;">Sky Jumping</p>
                @if (ViewBag.IsLoggedIn != null && ViewBag.IsLoggedIn)
                {
                    <form action="/Account/Enroll" method="post">
                        @Html.AntiForgeryToken()
                        <input type="hidden" name="_class" value="Fitness">
                        <input type="hidden" name="price" value="50$">
                        <input type="hidden" name="trainer_name"
value="Bella">
                        <input type="hidden" name="image"
value="images/class/fitness-class.jpg">
                        <input type="hidden" name="date" value="За
попереднім записом">
                        <button type="submit" class="btn custom-btn bg-
color nav-link">Підписатись</button>
                    </form>
                }
                else
                {
                    <button type="button" class="btn custom-btn bg-color
nav-link" data-toggle="modal" data-target="#registerForm">Вхід</button>
                }
            </div>
        </div>
    </div>
    <div class="col-lg-4 col-md-6 col-12" data-aos="fade-up" data-aos-
delay="400">
        <div class="class-thumb">
            
            <div class="class-info">
                <h3 class="mb-1">Weekend</h3>
                <span><strong>Субота - Неділя</strong></span><br />
                <span class="class-price">500 &#8372</span>
                <p class="mt-3">Тренажерний зал</p>
                <p class="mt-3" style="visibility: hidden;">as</p>
                <p class="mt-3" style="visibility: hidden;">as</p>
                @if (ViewBag.IsLoggedIn != null && ViewBag.IsLoggedIn)
                {
                    <form action="/Account/Enroll" method="post">
                        @Html.AntiForgeryToken()
                        <input type="hidden" name="_class" value="Weekend">
                        <input type="hidden" name="price" value="50$">
                        <input type="hidden" name="trainer_name"
value="Bella">
                        <input type="hidden" name="image"
value="images/class/weekend-class.jpg">
                        <input type="hidden" name="date" value="Субота-
Неділя">

```

```
                <button type="submit" class="btn custom-btn bg-
color nav-link">Підписатись</button>
            </form>
        }
        else
        {
            <button type="button" class="btn custom-btn bg-color
nav-link" data-toggle="modal" data-target="#registerForm">Вхід</button>
        }
    </div>
</div>
</div>
<!-- Aerobic Class -->
<!-- Аналогічно для інших класів -->
<!-- Cardio Class -->
<!-- Аналогічно для інших класів -->

</div>
</div>
</section>
```



```

        <div id="minutes" class="time">0</div>
        <div class="label">Minutes</div>
    </div>
    <div class="time-box">
        <div id="seconds" class="time">0</div>
        <div class="label">Seconds</div>
    </div>
</div>
</div>
<h3 class="bordered" style="color:#F13A11; border-
color:white; border-radius:10px;" data-aos="fade-up" data-aos-delay="300">
    @Model.Price €
</h3>
<div class="form-container">
    <form action="/Account/Login" method="post"
class="">
        @Html.AntiForgeryToken()
        <input type="hidden" name="className"
value="@Model.ClassName">
        <button type="submit" class="btn custom-btn
bg-color nav-link">Відписатись</button>
    </form>
</div>
</div>
</div>
</div>
</div>
</section>
if (Model.ClassName == null)
{
    <section class="hero d-flex flex-column justify-content-center
align-items-center" id="home" style="background-image:
url('../Content/@Model.Image') ">

        <div class="bg-overlay"></div>

        <div class="container">
            <div class="row">

                <div class="col-lg-8 col-md-10 mx-auto col-12">
                    <div class="hero-text mt-5 text-center">
                        <!-- Додавання назви абонементу -->
                        <h1 class="text-white" data-aos="fade-up" data-
aos-delay="500"><b style="color:#F13A11">Користувач не зареєстрований на жодне
заняття</b></h1>

                        <!-- Додавання ціни абонементу -->

                        <p class="subscription-price"></p>
                        <!-- Додавання імені тренера -->
                        <h3 class="text-white" data-aos="fade-up" data-
aos-delay="500">Тренер: &nbsp;&nbsp;&nbsp;<b
style="color:#F13A11">@Model.Trainer</b></h3>

                        <h4 class="text-white" data-aos="fade-up" data-
aos-delay="500"><b style="color:#F13A11">Абонемент дійсний ще :</b></h4>
                        <div class="countdown-body">
                            <div class="countdown-container">

```

```

class="time">0</div>
class="time">0</div>
class="time">0</div>
border-color:white; border-radius:10px;" data-aos="fade-up" data-aos-
delay="300">
    @Model.Price @
</h3>
<div class="form-container">
    <button type="submit"
onclick="scrollToSection('#about')" href="/#about" class="btn custom-btn bg-
color nav-link">Підписатись</button>
</div>
</div>
</div>
</div>
</div>
</div>
</section>
}
else
{
    <section class="hero d-flex flex-column justify-content-center align-
items-center" id="home" style="background-image: url('../Content/@Model.Image')
">
    <div class="bg-overlay"></div>
    <div class="container">
        <div class="row">
            <div class="col-lg-8 col-md-10 mx-auto col-12">
                <div class="hero-text mt-5 text-center">
                    <!-- Додавання назви абонементу -->
                    <h1 class="text-white" data-aos="fade-up" data-aos-
delay="500"><b style="color:#F13A11">Користувач не зареєстрований на жодне
заняття</b></h1>
                    <!-- Додавання ціни абонементу -->

```

```

        <p class="subscription-price"></p>
        <!-- Додавання імені тренера -->
        <h3 class="text-white" data-aos="fade-up" data-aos-
delay="500">Тренер: &nbsp;&nbsp;&nbsp;<b style="color:#F13A11">@Model.Trainer</b></h3>

        <h4 class="text-white" data-aos="fade-up" data-aos-
delay="500"><b style="color:#F13A11">Абонемент дійсний ще :</b></h4>
        <div class="countdown-body">
            <div class="countdown-container">
                <div class="time-box">
                    <div id="days" class="time">0</div>
                    <div class="label">Days</div>
                </div>
                <div class="time-box">
                    <div id="hours" class="time">0</div>
                    <div class="label">Hours</div>
                </div>
                <div class="time-box">
                    <div id="minutes" class="time">0</div>
                    <div class="label">Minutes</div>
                </div>
                <div class="time-box">
                    <div id="seconds" class="time">0</div>
                    <div class="label">Seconds</div>
                </div>
            </div>
            </div>
            <h3 class="bordered" style="color:#F13A11; border-
color:white; border-radius:10px;" data-aos="fade-up" data-aos-delay="300">
                @Model.Price €
            </h3>
            <div class="form-container">

                <a onclick="scrollToSection('#class') "
href="/#class" class="btn custom-btn bg-color nav-link
smoothScroll">Абонементи</a>
            </div>
        </div>
    </div>
</div>
</section>
}

```

```

<section class="schedule section" id="schedule">
    <div class="container">
        <div class="row">

            <div class="col-lg-12 col-12 text-center">
                <h2 class="text-white" data-aos="fade-up" data-aos-
delay="200">Розклад</h2>
            </div>

```

```

<div class="col-lg-12 py-5 col-md-12 col-12">
  <table class="table table-bordered table-responsive
schedule-table" data-aos="fade-up" data-aos-delay="300">

    <thead class="thead-light">
      <th><i class="fa fa-calendar"></i></th>
      <th>ПН</th>
      <th>Вт</th>
      <th>Ср</th>
      <th>Чт</th>
      <th>Пт</th>
      <th>Сб</th>
    </thead>

    <tbody>
      <tr>
        <tr>
          <td><small>7:00 </small></td>
          <td>
            <strong>Фітнес</strong>
            <span>7:00 - 7:45 </span>
          </td>
          <td>
            <strong>Кардіо</strong>
            <span>7:00 - 7:45 </span>
          </td>
          <td>
            <strong>Фітнес</strong>
            <span>7:00 - 7:45 </span>
          </td>
          <td>
            <strong>Кардіо</strong>
            <span>7:00 - 7:45 </span>
          </td>
          <td>
            <strong>Фітнес</strong>
            <span>7:00 - 7:45 </span>
          </td>
        </tr>

      <tr>
        <td><small>8:00 </small></td>
        <td>
          <strong>Sky jumping</strong>
          <span>8:00 - 8:45</span>
        </td>
        <td>
          <strong>Фітнес</strong>
          <span>8:00 - 8:45 </span>
        </td>
        <td>
          <strong>Кросфіт</strong>
          <span>8:00 - 8:45 </span>
        </td>
        <td>
          <strong>Sky jumping</strong>
          <span>8:00 - 8:45</span>
        </td>
        <td>

```

```

        <strong>Кросфіт</strong>
        <span>8:00 - 8:45 </span>
    </td>
    <td>
        <strong>Кросфіт</strong>
        <span>8:00 - 8:45 </span>
    </td>
</tr>

<tr>
    <td><small>9:00</small></td>
    <td></td>
    <td>
        <strong>Sky jumping</strong>
        <span>9:00 - 9:45</span>
    </td>
    <td>
    </td>
    <td>
        <strong>Sky jumping</strong>
        <span>9:00 - 9:45</span>
    </td>
    <td>
    </td>
    <td>
        <strong>Фітнес</strong>
        <span>9:00 - 9:45 </span>
    </td>
</tr>

<tr>
    <td><small>18:00</small></td>
    <td>
        <strong>Кардіо</strong>
        <span>18:00 - 18:45</span>
    </td>
    <td>
        <strong>Sky jumping</strong>
        <span>18:00 - 18:45</span>
    </td>
    <td></td>
    <td>
        <strong>Sky jumping</strong>
        <span>18:00 - 18:45</span>
    </td>
    <td>
        <strong>Фітнес</strong>
        <span>18:00 - 18:45</span>
    </td>
    <td>
    </td>
</tr>

<tr>
    <td><small>19:00</small></td>
    <td>
        <strong>Фітнес</strong>
        <span>19:00 - 19:45 pm</span>
    </td>
    <td>

```

```

        <strong>Кросфіт</strong>
        <span>19:00 - 19:45</span>
    </td>
    <td>
        <strong>Кардіо</strong>
        <span>19:00 - 19:45</span>
    </td>
    <td>
        <strong>Кросфіт</strong>
        <span>19:00 - 19:45</span>
    </td>
    <td>
        <strong>Кардіо</strong>
        <span>19:00 - 19:45</span>
    </td>
    <td>
        <strong>Кардіо</strong>
        <span>19:00 - 19:45</span>
    </td>
</tr>
<tr>
    <td><small>20:00</small></td>
    <td>
        <strong>Sky jumping</strong>
        <span>20:00 - 20:45 </span>
    </td>
    <td>
        <strong>Фітнес</strong>
        <span>20:00 - 20:45</span>
    </td>
    <td>
        <strong>Кросфіт</strong>
        <span>20:00 - 20:45</span>
    </td>
    <td>
        <strong>Фітнес</strong>
        <span>20:00 - 20:45</span>
    </td>
    <td>
        <strong>Sky jumping</strong>
        <span>20:00 - 20:45</span>
    </td>
    <td>
        <strong>Кросфіт</strong>
        <span>20:00 - 20:45</span>
    </td>
</tr>
</tbody>
</table>
</div>

</div>
</div>
</section>

<script>
    // Set the date we're counting down to using Razor syntax to pass
    Model.Date

```

```

    var countdownDate = new Date("@Model.DateEnroll.ToString("MMM d, yyyy
HH:mm:ss", CultureInfo.InvariantCulture)").getTime();

    // Update the count down every 1 second
    var x = setInterval(function() {
        // Get today's date and time
        var now = new Date().getTime();

        // Find the distance between now and the count down date
        var distance = countdownDate - now;

        // Time calculations for days, hours, minutes and seconds
        var days = Math.floor(distance / (1000 * 60 * 60 * 24));
        var hours = Math.floor((distance % (1000 * 60 * 60 * 24)) / (1000 *
60 * 60));
        var minutes = Math.floor((distance % (1000 * 60 * 60)) / (1000 *
60));
        var seconds = Math.floor((distance % (1000 * 60)) / 1000);

        document.getElementById("days").innerHTML = days;
        document.getElementById("hours").innerHTML = hours;
        document.getElementById("minutes").innerHTML = minutes;
        document.getElementById("seconds").innerHTML = seconds;

        // If the count down is finished, write some text
        if (distance < 0) {
            clearInterval(x);
            document.getElementById("days").innerHTML = '0';
            document.getElementById("hours").innerHTML = '0';
            document.getElementById("minutes").innerHTML = '0';
            document.getElementById("seconds").innerHTML = '0';
        }
    }, 1000);
</script>
}

```