

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Розробка CRM-системи управління бізнес-процесами»

Студента 2 курсу, 6м групи,
спеціальності 121 «Інженерія
програмного забезпечення»,
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

Вороневича Ростислава
Олександровича

Науковий керівник
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Рзаєва Світлана
Леонідівна

Гарант освітньої програми
доктор технічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Криворучко Олена
Володимирівна

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

8 листопада 2019 р.

Завдання на випускний кваліфікаційний проект студентів

Вороневичу Ростиславу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту «Розробка CRM-системи
управління бізнес-процесами»

Затверджена наказом ректора від "13" грудня 2019 р. № 4304

2. Строк здачі студентом закінченого проекту 01 грудня 2020 р.

3. Цільова установка та вихідні дані до проекту

Мета проекту вдосконалення системи продажів шляхом створення
прототипу інформаційної системи.

Об'єкт дослідження процес розробки CRM-системи управління бізнес-
процесами

Предмет дослідження розробка системи для подальшого впровадження на
підприємства.

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ОБГРУНТУВАННЯ ВИБОРУ ПРОГРАМНОЇ ПЛАТФОРМИ

CRM-СИСТЕМ

1.1. Функціональні характеристики CRM-систем

1.2. Мова програмування C#

1.3. Висновки до розділу

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ

2.1. Аргументування вибору CRM

2.2. Опис CRM-системи, платформ та використаних технологій

2.2.1. ASP.NET

2.2.2. SQL Server

2.2.3. AJAX

2.3. Висновки до розділу 2

РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ

3.1. Опис інтерфейсу користувача

3.1.1. Робочий простір менеджера

3.1.2. Робочий простір техніка

3.1.3. Робочий простір різноробочого

3.1.4. Робочий простір адміністратора

3.2. Розробка і опис програми

3.3. Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання проекту

№ пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів проекту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>	20.09.2019	20.09.2019
2.	<i>Розробка та затвердження завдання на проект магістра</i>	08.11.2019	08.11.2019
3.	<i>Вступ та перелік літературних джерел</i>	24.01.2020	24.01.2020
4.	<i>Наукова стаття</i>	01.09.2020	01.09.2020
5.	<i>Технічне завдання</i>	28.02.2020	28.02.2020
6.	<i>Розділ 1. Обґрунтування вибору програмної платформи CRM-систем</i>	25.06.2020	25.06.2020
7.	<i>Розділ 2. Проектування системи</i>	07.09.2020	07.09.2020
8.	<i>Розділ 3. Реалізація системи</i>	19.10.2020	19.10.2020
9.	<i>Програма та методика тестування</i>	21.10.2020	21.10.2020
10.	<i>Керівництво користувача</i>	23.10.2020	23.10.2020
11.	<i>Висновки та пропозиції</i>	30.10.2020	30.10.2020
12.	<i>Здача випускного кваліфікаційного проекту на кафедрі (перша перевірка)</i>	05.11.2020	05.11.2020
13.	<i>Підготовка автореферату та презентації доповіді</i>	05.11.2020	05.11.2020
14.	<i>Попередній захист випускного кваліфікаційного проекту</i>	25.11.2020- 27.11.2020	25.11.2020 -27.11.2020
15.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	01.12.2020	01.12.2020
16.	<i>Підготовка до публічного захисту випускної кваліфікаційної роботи</i>	08.12.2020- 09.12.2020	

7. Дата видачі завдання «8» листопада 2019 р.

8. Науковий керівник випускного кваліфікаційного проекту Рзаєва С. Л.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Криворучко О. В.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Вороневич Р. О.

(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

Відмітка про попередній захист Рзаєва С. Л.
(ПІБ, підпис, дата)

Випускний кваліфікаційний проект студента Вороневича Р. О.
(прізвище, ініціали)

Гарант освітньої програми Криворучко О. В .
(прізвище, ініціали, підпис)

« » 20 p.

АНОТАЦІЯ

Відштовхуючись від мети дослідження, в роботі було виділено увагу вдосконаленню системи продажів шляхом створення прототипу інформаційної системи, що в свою чергу дозволить спростити роботу працівникам та автоматизувати певні процеси.

В результаті дослідження було виявлено оптимальний варіант для покращення роботи підприємств та обрано певні програмні продукти для реалізації.

Визначена роль CRM-систем в сучасних реаліях та мета їх впровадження на підприємства різної форми власності.

Розробка програмного продукту виконана у середовищі Visual Studio 2019. C# виступив в ролі обраною мови програмування. Окремі елементи програми розроблені із використанням HTML.

ABSTRACT

Based on the purpose of the study, the paper focused on improving the sales system by creating a prototype information system, which in turn will simplify the work of employees and automate certain processes.

As a result of the research, the optimal option for improving the work of enterprises was identified and certain software products were selected for implementation.

The role of CRM-systems in modern realities and the purpose of their implementation in enterprises of different forms of ownership are determined.

Software development was performed in Visual Studio 2019. C # acted as the selected programming language. Some elements of the program are designed using HTML.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІС – Інформаційна Система
 БД – База Даних
 ASP – Active Server Page
 MVC – Model - View - Controller
 T-SQL – Transact-SQL
 TDS – Tabular Data Stream
 ODBC – Open Database Connectivity
 SQL – Structured Query Language
 AJAX – Asynchronous JavaScript and XML
 EF Core – Entity Framework Core
 ORM – Object-Relational Mapping
 VS – Visual Studio

					КНТЕУ 121 06-03.МР			
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка CRM-системи управління бізнес-процесами	Стадія	Аркуш	Аркуші
Зав. каф.		Криворучко О.В.		19.10.20		ПС	2	51
Керівник		Рзаєва С.Л.		19.10.20		Факультет інформаційних технологій 2м курс, 6 група		
Гарант		Криворучко О.В.		19.10.20				
Розробив		Вороневич Р.О.		19.10.20	Перелік умовних скорочень			

Зміст

ВСТУП.....	4
РОЗДІЛ 1. ОБГРУНТУВАННЯ ВИБОРУ ПРОГРАМНОЇ ПЛАТФОРМИ CRM-СИСТЕМ.....	6
1.1. Функціональні характеристики CRM-систем.....	6
1.2. Мова програмування C#.....	10
1.3. Висновки до розділу	12
РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ	14
2.1. Аргументування вибору CRM.....	14
2.2. Опис CRM-системи, платформ та використаних технологій	22
2.2.1. ASP.NET.....	22
2.2.2. SQL Server.....	24
2.2.3. AJAX.....	26
2.2.4. Entity Framework Core (EF Core).....	28
2.3 Висновки до розділу	31
РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ.....	32
3.1. Опис інтерфейсу користувача	32
3.1.1 Робочий простір менеджера	34
3.1.2. Робочий простір техніка	37
3.1.3. Робочий простір різноробочого	38
3.1.4. Робочий простір адміністратора	40
3.2 Розробка і опис програми	44
3.3 Висновки до розділу 3	45
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	48
ТЕХНІЧНЕ ЗАВДАННЯ.....	52
ДОДАТКИ.....	54

					КНТЕУ 121 06-03.МР			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		19.10.20	Розробка CRM-системи управління бізнес-процесами	Стадія	Аркуш	Аркуші
Керівник		Рзаєва С.Л.		19.10.20		Зміст	3	51
Гарант		Криворучко О.В.		19.10.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив		Вороневич Р.О.		19.10.20				
					Зміст			

ВСТУП

Актуальність теми Запорука успішного ведення бізнесу – постійне самовдосконалення. Але, в поточній реальності, вдосконалення неможливо без застосування сучасних інформаційних технологій. Без належного контролю і управлінських впливів, навіть дуже хороші менеджери відчутно недопрацьовують. Без якісних інструментів керівники вимірюють ефективність співробітників на дотик. В результаті ситуація наступна: керівництво приблизно знає вхід (клієнтів) і вихід (продажу), а що робиться всередині і що можна ще поліпшити не знають і не вимірюють. Правильно організована інформаційна система може принести бізнесу значні переваги.

Питання управління взаємовідносинами з клієнтами все частіше піднімаються в якості пріоритетних у багатьох компаніях. Компанії, що працюють в самих різних областях, розглядають CRM як частину корпоративної стратегії з двох причин:

1. новітні технології дозволяють компаніям точніше націлювати пропозиції на обрані сегменти ринку;
2. традиційний маркетинг значно програє в порівнянні з потенціалом стратегій, орієнтованих на клієнта.

Концепція взаємин з клієнтами за допомогою об'єднання сучасних технологій з новим ринковим мисленням дає можливість створювати, розвивати і зміцнювати довгострокові відносини з ретельно обраними клієнтами, а також збільшити вигоди клієнта, значно підвищити прибуток компанії.

					<i>КНТЕУ 121 06-03.МР</i>		
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
Зав. каф.	Криворучко О.В.			21.01.20	Розробка CRM-системи управління бізнес-процесами	<i>Стадія</i>	<i>Аркуш</i>
Керівник	Рзаєва С.Л.			21.01.20		<i>В</i>	<i>4</i>
Гарант	Криворучко О.В.			21.01.20		<i>Факультет інформаційних технологій</i>	
Розробив	Вороневіч Р.О.			21.01.20			
					<i>Вступ</i>	<i>2м курс, 6 група</i>	

Мета дослідження вдосконалення системи продажів шляхом створення прототипу інформаційної системи.

Об'єкт дослідження CRM-система управління бізнес процесами.

Предмет дослідження розробка CRM-системи для подальшого впровадження на підприємства.

Новизна дослідження вдосконалення, порівняно з існуючими системами, концепції роботи з клієнтами та реалізація власного бачення CRM-системи.

Методи дослідження аналіз діяльності підприємств з залученням інформаційних систем, обґрунтування вибору інформаційної системи.

Завдання дослідження

1. Визначення доцільності використання інформаційних систем на підприємствах.
2. Аргументувати вибір саме CRM-системи в залученні її на підприємство.
3. Обґрунтувати вибір обраного ПЗ для реалізації процесу створення прототипу CRM-системи.
4. Протестувати та усунути можливі недоліки прототипу.

Практичне значення альтернативне рішення існуючим CRM-системам власним прототипом для подальшого впровадження на різні типи підприємства.

					КНТЕУ 121 06-03.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		5

РОЗДІЛ 1

ОБГРУНТУВАННЯ ВИБОРУ ПРОГРАМНОЇ ПЛАТФОРМИ CRM-СИСТЕМ

1.1. Функціональні характеристики CRM-систем

В основі успіху діяльності будь якої компанії першочерговою місією є орієнтування фірми на клієнта. Це означає, що на задоволення потреб клієнта повинна бути направлена діяльність всіх підрозділів компанії, а не лише відділу продажу, спеціалісти якого напряму спілкуються з клієнтом. Всі співробітники повинні бути зацікавлені в задоволенні потреб споживачів, знають та тверезо оцінюють можливості конкурентів, та повинні своєчасно знаходити оптимальне рішення для залучення та утримання клієнтів.

Строга орієнтація на клієнта значно підвищує шанси компанії на виживання в довгостроковій перспективі, збільшує короткостроковий прибуток від стимуляції збуту.

Для досягнення строгої орієнтації на клієнта компанії необхідно розуміти зміни, які відбуваються на ринку, ще й до цього своєчасно на них реагувати, тобто компанія в цілому має бути максимально гнучкою.

Компанії з строгою орієнтацією на ринок відрізняє те, що вона постійно аналізує запити споживачів, стратегію конкурентів, освоює нові технології, а також шукає рішення для своїх цільових споживачів.

Компанія, строго орієнтована на ринок, не тільки задовольняє запити споживачів, а й отримує значні прибутки протягом короткого проміжку часу, створює високу споживчу вартість і управляє споживчою лояльністю, що позитивно позначається на положенні компанії в довгостроковій перспективі.

					<i>КНТЕУ 121 06-03.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Розробка CRM-системи управління бізнес- процесами	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.	Криворучко О.В.			25.06.20		<i>P1</i>	6	51
Керівник	Рзаєва С.Л.			25.06.20		Факультет інформаційних технологій 2м курс, 6 група		
Гарант	Криворучко О.В.			25.06.20				
Розробив	Вороневич Р.О.			25.06.20	<i>Обґрунтування вибору програмної платформи crm- систем</i>			

Компанія зі слабкою орієнтацією на споживача має поверхневі уявлення про потреби споживачів і про конкуренцію, що призводить до значного зниження прихильності клієнтів і, як наслідок, їх переходу до конкурентів. Такій компанії доводиться нести значні витрати по запобіганню відтоку клієнтів, масштаби подібних витрат можна порівняти з витратами на залучення нових клієнтів. У сукупності низький рівень споживчої лояльності і високі маркетингові витрати призводять до більш низького, ніж очікувалося, прибутку.

Високий рівень купівельної стабільності є головною перевагою суворої орієнтованості на клієнта. Саме тому збереження клієнтів є одним із важливих завдань клієнт-орієнтованої компанії.

Компанія, строго орієнтована на клієнта, розробляє і впроваджує стратегію, що забезпечує високий рівень задоволення клієнта і створення лояльності. Саме задоволені, постійні клієнти забезпечують прибутковість компанії.

Задоволення покупця і збереження клієнтської бази є важливими показниками маркетингової діяльності компанії, але для створення лояльності потрібен високий рівень довіри клієнта. Якщо споживач рекомендує продукт компанії своїм знайомим, це говорить про високу формі емоційної підтримки і означає, що клієнт довіряє обраній компанії.

Забезпечити ефективну роботу маркетингу відносин дозволяє маркетингова інформаційно-аналітична система. Впровадження такої системи, яка є інструментом управління маркетинговою інформацією, передбачає здійснення змін в системі управління маркетинговою діяльністю і направленою на задоволення потреб користувачів в інформації, необхідній для прийняття управлінських рішень, а також на створення організаційних умов для ефективного функціонування і розвитку компанії в цілому.

					КНТЕУ 121 06-03.МР	Аркуш
						7
Зм.	Аркуш	№ докум	Підпис	Дата		

Компанія, в якій краще організована діяльність по збору зовнішньої інформації та її внутрішнього розподілу, може найбільш точно прогнозувати динаміку ринкових тенденцій і діяти без зволікання, більш обґрунтовано приймати управлінські рішення.

Впровадження системи інформаційно-аналітичного забезпечення маркетингової діяльності дозволяє компанії успішно розвиватися і управляти взаємовідносинами з клієнтами, що вимагає прийняття концепції клієнтоорієнтованості, або концепції CRM (Customer Relationship Management).

Найбільш повно суть даної концепції відображає таке визначення: «система управління взаємодією з клієнтами - корпоративна інформаційна система, призначена для автоматизації CRM-стратегії компанії, зокрема, для підвищення рівня продажів, оптимізації маркетингу і поліпшення обслуговування клієнтів шляхом збереження інформації про клієнтів і історії взаємин з ними, встановлення і поліпшення бізнес-процедур і подальшого аналізу результатів».

Основні принципи CRM-системи:

1. Наявність єдиного сховища інформації, звідки в будь-який момент можна отримати всі відомості про попередні і заплановані взаємодії з клієнтами.
2. Використання всіх каналів взаємодії (телефонні дзвінки, електронна пошта, рекламні розсилки і т.д.).
3. Постійний аналіз зібраної інформації про клієнтів і підготовка даних для прийняття відповідних організаційних рішень (наприклад, сегментація клієнтів на основі їх значущості для компанії)».

Впровадження CRM-системи дозволяє домогтися таких результатів, як зниження витрат на службу підтримки, поліпшення якості сервісу, підвищення задоволеності клієнта, збільшення кількості повторних покупок.

					КНТЕУ 121 06-03.МР	Аркуш
						8
Зм.	Аркуш	№ докум	Підпис	Дата		

Ідеологія CRM передбачає перехід від стратегії мас-маркетингу і масових продажів до індивідуальних продажів або послуг, налаштованим або допрацьованим так, щоб вони відповідали персональним вимогам клієнтів. У підсумку це зробить відносини компанії і клієнта більш комфортними і ефективними, розрахованими на перспективу.

CRM-система передбачає автоматизацію бізнес-процесів маркетингу, заснованих на персональній роботі з кожним з клієнтів (доведення контакту до укладання угоди, а також взаємини з клієнтом після операції), а також обслуговуванні і підтримці клієнтів (важливо дізнатися, що не влаштовує клієнта і чого йому бракує, щоб саме це йому і запропонувати в рамках майбутньої маркетингової діяльності). CRM-система забезпечує оперативний доступ до всієї наявної інформації, реалізуючи різну форму її подання для різних користувачів.

CRM-система дозволяє впорядкувати і накопичувати інформацію про клієнтів, розвивати програми лояльності. Фахівці компанії можуть працювати в єдиному інформаційному просторі, а керівництво і аналітики - вчасно отримувати оперативну інформацію і приймати найбільш ефективні рішення. За допомогою цієї системи здійснюється управління маркетингом, проводиться аналіз ефективності рекламних кампаній.

Сучасні CRM-системи розділені на три ключових напрямки:

1. Оперативний CRM - це рівень автоматизації оперативних процедур виконавців і оперативних керівників різного рівня. В Україні саме інформаційні системи цього рівня найчастіше називають CRM-системами. До даного напрямку відносяться системи автоматизації маркетингу, продажів і сервісу, облікові системи, що зберігають і обробляють фінансову інформацію про клієнтів. Споживачами результатів автоматизації даного рівня є безпосередні виконавці - співробітники відділів, які працюють з клієнтами: маркетологи,

					КНТЕУ 121 06-03.МР	Аркуш
						9
Зм.	Аркуш	№ докум	Підпис	Дата		

менеджери з продажу, секретарі, керівники відповідних лінійних підрозділів, служба сервісної підтримки.

2. Аналітичний CRM - це рівень автоматизації управлінських процесів, пов'язаних з глибоким аналізом даних про клієнтську базу компанії. Відповідно, інформаційні системи, що входять в це коло, - це системи організації сховищ даних, системи оперативного і статистичного аналізу клієнтської бази. Споживачами результатів автоматизації даного рівня є найчастіше керівники, які займаються стратегічним управлінням компанії.

У число завдань, що вирішуються за допомогою аналітичного CRM, входять синхронізація розрізнених масивів даних і пошук статистичних закономірностей в цих даних для вироблення найбільш ефективної стратегії маркетингу, продажів, обслуговування клієнтів і т.д. Вимагає хорошої інтеграції систем, великого обсягу напрацьованих статистичних даних, якісного аналітичного інструментарію.

3. Колабораційний CRM - це рівень автоматизації контактів з клієнтами по різних каналах зв'язку (телефон, особистий контакт, електронні канали). Споживачами результату автоматизації цього рівня є ті ж співробітники, що і в колі оперативного CRM.

1.2. Мова програмування C#

В якості середовища розробки було обрано Visual Studio 2019, який як для студента майже ідеальний варіант, дивлячись на його доступність та відносну простоту в освоєнні та подальшій праці.

Обраною мовою для написання CRM-системи – C #. Мова проста, сучасна та об'єктно-орієнтована.

					КНТЕУ 121 06-03.МР	Аркуш
						10
Зм.	Аркуш	№ докум	Підпис	Дата		

C # подібний синтаксисом та відноситься до сім'ї мов C, з цієї сім'ї синтаксис найбільш подібний до двох мов C ++ та Java. Мова має можливість обробляти дані різних типів, має статичну типізацію, атрибути, властивості, події і змінні, перевантаження операторів (в тому числі операторів явного і неявного приведення типу), анонімні функції з підтримкою замикань, LINQ, виключення, коментарі в форматі XML.

Мова C# створена для розробки різноманітних додатків, які використовуються в середовищі .NET Framework, де можна розробляти який завгодно додаток майже без обмежень. Можна створювати за бажанням динамічні та статичні веб-сторінки, веб служби, компоненти для роботи з базами даних, різні додатки та програми.

Особливості C #:

1. Інтегрована підтримка автоматичного генерування XML-документації. Автоматично звільнюється динамічно розподілена пам'ять.
2. Можливість позначати класи та методи атрибутами, які визначені користувачем. В деяких випадках має певну корисність при документуванні та має можливість впливати на процес компіляції.
3. Абсолютний доступ до бібліотеки базових класів .NET, а ще легкий доступ до Windows-API.
4. Вказівки та прямий доступ до пам'яті, звісно якщо в них є необхідність. Втім мова розроблена певним чином, що в домінуючій більшості випадків можна обійтися і без цього.
5. Підтримка певних властивостей та подій в стилі VB. Нескладна зміна ключів компіляції. Дає можливість отримувати виконувані файли чи бібліотеки компонентів .NET, які мають всі шанси бути викликані іншим кодом.

					КНТЕУ 121 06-03.МР	Аркуш
						11
Зм.	Аркуш	№ докум	Підпис	Дата		

6. Є змога використання мови C # для написання динамічних web-сторінок ASP.NET. Однією з деяких областей, для яких не призначена ця мова, є критичні за часом та високопродуктивні програми, коли дійсно має значення, займати виконання циклу 1000 чи 1050 машинних циклів, і звільнювати ресурси потрібно відразу. C ++ залишається в даній області кращою з мов низького рівня. У C # відсутні певні головні моменти, які в свою чергу потрібні для створення додатків з високою продуктивністю, виконання яких гарантується в конкретних точках коду.

Для стабільної та спокійної роботи була обрана операційна система Windows 10, на даний момент вона є найстабільнішою, досі підтримується компанією Microsoft та є найпоширенішою серед існуючих станом на 2020 рік. Вона підтримує та стабільно запускає сучасні програми та компоненти, які в подальшому будуть використані в створенні CRM-системи.

1.3. Висновки до розділу

Таким чином, прийняття компанією концепції з управління взаємовідносин з клієнтами та впровадження CRM-системи дозволяє забезпечити високу ефективність роботи з клієнтами. Результат впровадження CRM – працююча в компанії ідеологія клієнтоорієнтованості, підтримувана засобами автоматизації.

CRM-система забезпечує впорядкованість інформації про клієнтів, розвиток програм лояльності, можливість роботи в єдиному інформаційному полі, а також оперативність даних для прийняття найбільш ефективних управлінських рішень. CRM-система дозволяє управляти маркетингом, а також проводити аналіз ефективності рекламних кампаній, відстежувати історію взаємин з клієнтами, розвиваючи її.

					КНТЕУ 121 06-03.МР	Аркуш
						12
Зм.	Аркуш	№ докум	Підпис	Дата		

Для ефективного створення програмного забезпечення вдалим вибором буде використання об'єктно-орієнтованої мови програмування C#, оскільки вона являє собою універсальний інструмент для досягнення поставлених цілей.

					КНТЕУ 121 06-03.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		13

РОЗДІЛ 2

ПРОЕКТУВАННЯ СИСТЕМИ

2.1. Аргументування вибору CRM

Для вирішення низки проблем та автоматизації певних процесів, потрібно впровадити CRM-систему. Початковий етап роботи – це розробка самого проекту програми: того, що потрібно від програми і яким функціоналом вона повинна володіти. Від того наскільки якісно буде опрацьований проект системи буде залежати майбутня віддача від його реалізації.

Впровадження інновацій, в наш час, має доволі велике значення не лише для загального підвищення конкурентоспроможності компаній, а й для розвитку результативних клієнтських взаємовідносин, що забезпечують, в власну чергу, прибутковість та продуктивність компанії.

Саме тому, стратегія рішення тільки проблеми оптимізації виробництва, не є запорукою подальшого виживання компанії. У тих сферах, де збільшення конкуренції йде достатньо активно поряд з швидким оновленням інформаційних технологій, включається ще один зовнішній фактор – зовнішні інвестиції. Для виведення нових товарів та послуг на масовий ринок компаніям вже не вистачає власних коштів. Залучення ж зовнішніх інвестицій зазвичай супроводжується вимогою якнайскорішого їх повернення, що прямо дає визначення в необхідній швидкості нарощування доходів, що в свою чергу веде до нарощування клієнтської бази.

На сьогоднішній день споживач став об'єктом та головною ціллю всіх зусиль виробників та надавачів послуг, а їх задоволеність відносинами з

					<i>КНТЕУ 121 06-03.МР</i>		
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
Зав. каф.		Криворучко О.В.		07.09.20	Розробка CRM-системи управління бізнес-процесами	<i>Стадія</i>	<i>Аркуш</i>
Керівник		Рзаєва С.Л.		07.09.20		<i>P2</i>	<i>14</i>
Гарант		Криворучко О.В.		07.09.20		<i>Факультет інформаційних технологій</i>	
Розробив		Вороневич Р.О.		07.09.20			
					<i>Проектування системи</i>	<i>2м курс, 6 група</i>	

постачальником товарів і послуг – ключовим фактором явного успіху компанії. Боротьба за клієнта стає все більш і більш запеклою. Разом з тим, деякі керівники досі вважають, що місце інноваційних технологій в боротьбі за клієнта - в розряді діяльності допоміжних служб. Однак, як показує досвід, ті керівники, які розглядають передові технології як важливий інструмент управління бізнесом, швидше приходять до успіху. Для компаній, що бажають бути успішними, настав момент, коли не можна далі відкладати приведення в порядок своєї клієнтської бази. При цьому не має значення, чи працює компанія на корпоративному або роздрібному ринку. Будь-якому керівнику в поточній ситуації необхідно приймати рішення, які вимагають доброго розуміння клієнтів.

У ситуації, що склалася особливо актуальним є скорочення витрат, точне планування, активна робота з наявними клієнтами та їх утриманням. Не маючи єдиної вивіреної бази з даними про клієнтів, компанія втрачає істотні кошти через недостатньо ефективної роботи відділів маркетингу, продажів і обслуговування.

Звісно, на допомогу прийдуть перевірені часом та тисячами компаній, добре зарекомендовані CRM-системи по типу: «Sales Creatio», «Битрикс24»,

«OneBox», «1С:CRM», і т.д. Насправді, їх неймовірна кількість, але це не нашій варіант. Для забезпечення потреб різних підприємств буде створена власна, універсальна CRM-система, яка допоможе керувати бізнес процесами, сприятиме спрощенню роботи якнайбільшій кількості людей шляхом автоматизації більшості робочих моментів з накопиченням та подальшим використанням отриманих даних в роботі з клієнтами різної складності.

Створення власної CRM-системи буде спрямоване на її універсальність та гнучкість в роботі та однакової ефективності на більшості типах зайнятості підприємств.

					КНТЕУ 121 06-03.МР	Аркуш
						15
Зм.	Аркуш	№ докум	Підпис	Дата		

Основною метою створення CRM-системи – йти в ногу з часом, оскільки відштовхуючись від минулого коли жодної системи взаємини з клієнтами ще не існувало, продажі в переважній більшості компаній вибудовувались за певною схемою.

Існує офіс, в ньому декілька менеджерів та стаціонарних телефонів. Менеджери самі телефонують та приймають вхідні дзвінки, якимось чином продають, якимось інформують клієнтів, якимось відповідають на їх різні запитання. Більш того, чимала кількість компаній по такій схемі працюють до сьогоднішнього дня. При такій системі, як можна зрозуміти, результати відділу продажів великим чином залежали, в першу чергу, від порядності та індивідуальних якостей кожного з менеджерів. В той час і з'явився культ всемогутнього продавця, а більшість компаній активно розпочали розвивати навички комунікації своїх менеджерів. Питання побудови системи та конкретних правил в роботі з клієнтами перейшов на другий план.

У ті часи компанії цілим скопом відправляли свої відділи продажів на підвищення кваліфікації, надіючись, що саме магічні інструменти та певні техніки зроблять неймовірний ривок для збільшення продажів їх бізнесу. У рівних умовах мали значну перевагу ті, кому вдавалося найняти до себе на роботу супер-професіоналів свого діла.

З одного боку, якщо компанії пощастило з кваліфікованим менеджером, то він не втратить клієнта, вчасно йому зателефонує і в результаті доведе до угоди. Однак дивлячись на інший бік, в такій ситуації компанія повною мірою залежить від найнятого менеджера. Сьогодні знайшов професіонала - компанія в великому плюсі. Завтра професіонал пішов до конкурентів (разом з всією накопиченою базою клієнтів минулої компанії), і компанія повністю розорена. Таких випадків зустрічалось багато і серед досить вагомих та великих компаній, коли уходили цілі команди - і компанія, що мала великий та стабільний оборот, залишалася лише з боргами.

					КНТЕУ 121 06-03.МР	Аркуш
						16
Зм.	Аркуш	№ докум	Підпис	Дата		

Рік за роком почали появлятися сучасні технології, які в свою чергу принесли з собою абсолютно нові та небачені раніше інструменти роботи з різними клієнтами. З ходом часу стало відомо, що для успіху продажів важливо знати все про покупця: як його звуть і скільки йому років, де він проживає, чи має власну сім'ю, чому надає перевагу в купівлі, де і як часто він це робить. Використовуючи інноваційні автоматизовані інформаційні системи, call-центри та аналітичні бази даних, з'явилась можливість і рентабельність працювати з кожним клієнтом так, немов він в них єдиний.

Людина егоїстична, дуже сильно любить увагу до власної персони. Клієнт завжди буде розчарованим "холодним" прийомом в банку, недостатньою увагою в магазині, довгою та нудною чергою в поліклініці. Тому будь-яку увагу, навіть незначну, до своїх потреб він залишає в своїй пам'яті. Компанія, що сконцентровує увагу до кожного з покупців, стає улюбленою для переважної більшості клієнтів. Клієнт звертатиметься до послуг обраної компанії знову і знову, тим самим певною мірою збільшуючи загальний обсяг продажів компанії без великих витрат на просування товару. За статистикою, задоволений клієнт розповість про вдачу покупку в середньому п'яти знайомим, які в свою чергу з великою вірогідністю звернуться за необхідності в перевірену фірму.

Існуючі клієнти витрачають на 67% більше, ніж нові клієнти, а нові клієнти, залучені постійними, на 25% більш цінні, ніж будь-які інші. Не доводиться сумніватися в тому, чому підприємства інвестують в лояльність і рекомендації. Це основа стратегії. Якщо клієнти лояльні до бренду, то з імовірністю 60% вони розкажуть про нього своїм друзям і родині (рис. 2.1). Безкоштовний сарафанний маркетинг.

					КНТЕУ 121 06-03.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		17

For brands they are loyal to, consumers are willing to:

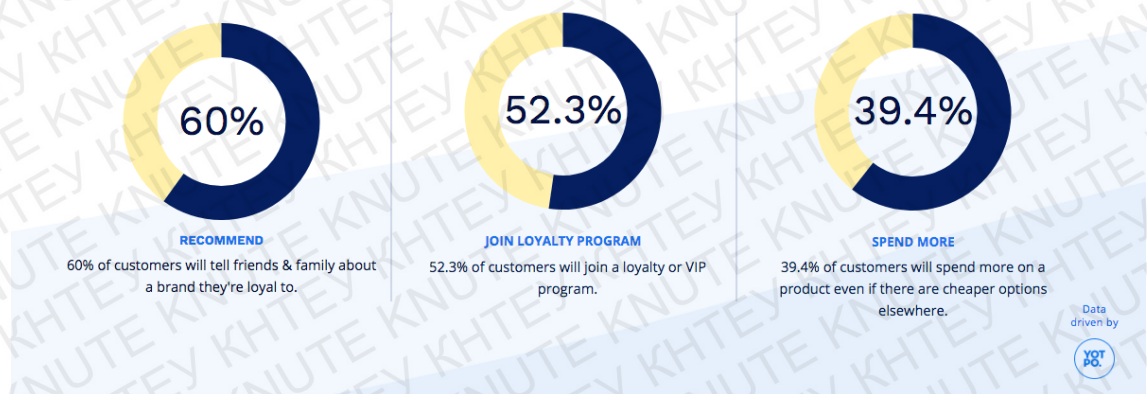


Рис. 2.1. Діаграма в відсотковому співвідношенні візуалізації лояльності покупців

Щоб завоювати лояльність клієнтів і поліпшити якість обслуговування, компанії наймають менеджерів, купують CRM-системи, навчають співробітників, і думають, що це допоможе продажам.

Існує чимало плюсів, які приносить з собою CRM-система в компанію:

1. Безпечне місце для зберігання даних. Завдяки системі CRM більше не доведеться думати і турбуватися про зберігання цінних даних про клієнтів. Співробітники відділу продажів спокійно вводять всю інформацію, вона залишається в базі даних, до якої вони мають доступ, і права на цей доступ визначає адміністратор, який має права редагувати і видаляти, на відміну від користувачів, які в змозі лише переглядати. Все залежить від рівня довіри до конкретного працівника. Вся інформація зберігається в одному місці і не може зникнути безслідно, завжди можна побачити зміни, внесені тим чи іншим менеджером. Працівники можуть приходити і уходити, а інформація буде на місці.
2. Менеджери з продажу зможуть заздалегідь зрозуміти, чого хоче покупець. Оскільки вся інформація зберігається в одному місці, менеджер має доступ до повної історії операцій з тією або іншою

					KНTEУ 121 06-03.MP	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		18

особою, що дозволяє йому заздалегідь аналізувати потреби клієнта. Коли спілкування з клієнтом будується на рівні довіри це підвищує його лояльність і довіру до компанії.

3. Ефективне управління часом. Менеджери зможуть правильно розподіляти навантаження та час. Завдяки цьому працівники можуть планувати свої завдання на конкретний день таким чином, щоб вони могли точно зателефонувати всім, писати та відсилати документи. Все це для того, щоб покупець в якийсь момент почувався непотрібним або забутим. Ефективний розподіл часу прискорює всі стадії угоди, а це означає, що можна обробити більшу кількість замовлень.
4. Можливість легко і швидко створювати звіти. Завдяки CRM-системі час, який менеджери витрачають на підготовку типових щотижневих та щомісячних звітів може піти в минуле. Менеджери можуть формувати звіти безпосередньо з системи та негайно надсилати їх керівнику за кілька кліків. Сам звіт стає простим, прозорим, а головне автоматизованим.
5. Скорочення часу на однотипні завдання. З CRM-системою менеджеру не доводиться витрачати стільки часу на адміністративні завдання. Це спрощує і видаляє рутинні дії. Всі деталі по продукту зберігаються прямо в CRM, також зникає необхідність постійно записувати щось в календар, а в наслідку судорожно шукати, на яке число він призначив зустріч. Тепер можна просто поставити завдання, а CRM сама тобі про них нагадає.
6. Економія на «людському факторі». Після інтеграції CRM-системи, значно зменшиться кількість помилок створених з вини співробітника. Всі ці історії з помилкою в замовленні, ціною, адресою доставки будуть значно зменшені, а значить можна заощадити гроші, які в іншому випадку пішли б на відстеження та виправлення таких помилок.

					КНТЕУ 121 06-03.МР	Аркуш
						19
Зм.	Аркуш	№ докум	Підпис	Дата		

7. Автоматична аналітика. Система сама збирає дані по всіх операціях і готує аналітику. Можна дивитися, на якому етапі угоди відбувається більша кількість невдач, і виходячи з цього знайти розуміння, з чим це пов'язано. У разі якщо недостатньо базової аналітики, завжди можна додати віджети або персонально розроблені рішення.

Для кращого розуміння роботи можна розглянути декілька моделей CRM-систем.

Модель IDIC розробили Peppers and Rogers Group як шаблонний проект для впровадження CRM в різних ситуаціях. IDIC позначає чотири етапи впровадження CRM: ідентифікація, диференціація, взаємодія і настройка (рис. 2.2).

1. Ідентифікація

Перший крок – визначити клієнтів, яких можна добути збором інформації, такої як ім'я, адреса і історія покупок, в кожній точці контакту. Мета – зібрати якомога більше даних про кожного клієнта, щоб краще зрозуміти його потреби, бажання і поведінку при покупці.

2. Диференціація

Наступний крок – сегментування клієнтів на основі поточної і прогнозованої тривалості життя. Потрібно пам'ятати, не всі клієнти мають однакову цінність. Потрібно розрізняти клієнтів в залежності від їх цінності. Можна розставити пріоритети в своїх відносинах з клієнтами. Потрібно відповідати очікуванням кожного сегмента для оптимального прибутку.

3. Взаємодія

Третій етап – застосувати CRM-плани для взаємодії з клієнтами. Після аналізу та сегментації клієнтів, можна розробити індивідуальні взаємодії. Наприклад, для цінних клієнтів запропонувати пільги або бонуси за лояльність, стимулюючи постійні витрати.

					КНТЕУ 121 06-03.МР	Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата		

4. Налаштування

Після документації взаємодій, можна проаналізувати їх і розробити більш індивідуальний сервіс. Мета – гарантувати, що потреби і очікування клієнтів будуть задоволені, за правилами сегмента або точкового підходу.

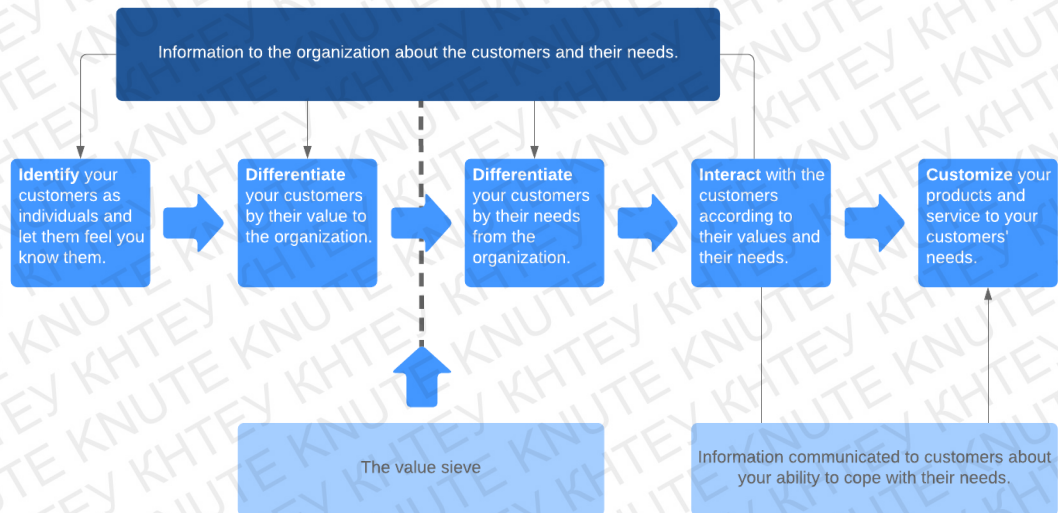


Рис. 2.2. Модель IDIC

Наступна модель QCI

Це скоріше модель управління клієнтами. Модель індексу конкурентоспроможності якості фокусується на трьох напрямках: залучення, утримання та проникнення (рис. 2.3).

Модель QCI починається з зовнішнього середовища клієнта - больові точки, бізнес-цілі і інші фактори. Вони будуть впливати на готовність купувати або взаємодіяти з командою продажів. Це, в свою чергу, впливає на якість обслуговування. Досвід клієнта впливає на пропозицію (то, що пропонуєте клієнту) і дії з маніпуляцією клієнтом. Зі схеми видно, для залучення і утримання клієнтів задіяні багато дій.

Модель QCI також враховує людей і технології, пов'язані з підтримкою роботи цієї системи. Ця модель як починається так і закінчується людьми.

					КНТЕУ 121 06-03.МР	Аркуш
						21
Зм.	Аркуш	№ докум	Підпис	Дата		

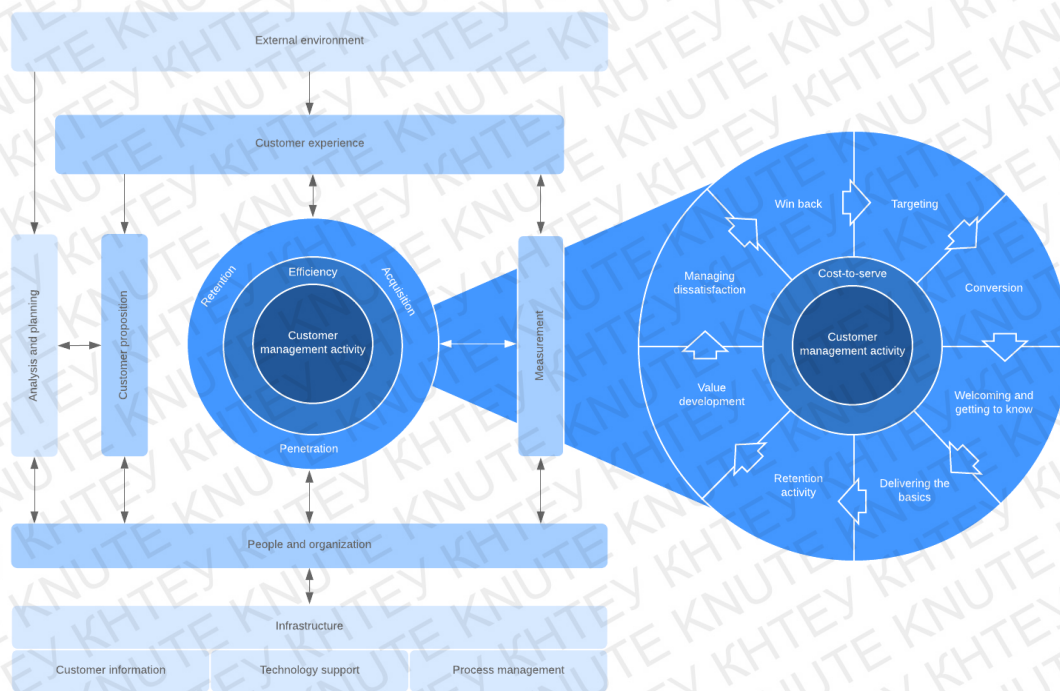


Рис. 2.3. Модель QCI

Таким чином компанію без CRM-системи і компанію з CRM-системою можна порівняти з двома поїздами. Тільки перший поїзд - стара повільна електричка, а другий - сучасний високо комфортний поїзд-експрес, який доїжджає до кінцевого пункту в два рази швидше. Звичайно, за умови грамотного машиніста.

2.2. Опис CRM-системи, платформ та використаних технологій

2.2.1. ASP.NET

Для створення CRM – системи була обрана платформа ASP.NET. Для відповіді обґрунтування вибору відзначаються відмінні риси обраної платформи. Технологія ASP.NET є розвитком Active Server Page (ASP). Дана технологія являє собою універсальну платформу для розробки веб-додатків

					<i>KHTEY 121 06-03.MP</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		22

корпоративного рівня. ASP.NET пропонує нову модель програмування і інфраструктуру, які дозволяють розробляти захищені і масштабовані рішення.

Рішення реалізовано за допомогою паттерна MVC 5. Концепція паттерна (шаблону) MVC (model – view – controller) передбачає поділ додатка на три компоненти:

Контролер (controller) представляє клас, що забезпечує зв'язок між користувачем і системою, поданням і сховищем даних. Він отримує введені користувачем дані і обробляє їх. І в залежності від результатів обробки відправляє користувачеві певний вивід, наприклад, у вигляді подання.

Подання (view) – це власне візуальна частина або призначений для користувача інтерфейс програми. Як правило, html сторінка, яку користувач бачить, зайшовши на сайт.

Модель (model) представляє клас, що описує логіку використовуваних даних.

При такому підході модель є незалежним компонентом - будь-які зміни контролера або подання не зачіпають модель. Контролер і подання є відносно незалежними компонентами, і нерідко їх можна змінювати незалежно один від одного.

Завдяки цьому реалізується концепція поділ відповідальності, в зв'язку з чим легше побудувати роботу над окремими компонентами. Крім того, внаслідок цього додаток має більшу податливість до тестування. І якщо важлива візуальна частина або фронтенд, то можна тестувати подання незалежно від контролера. Або ми можемо зосередитися на бекенді і тестувати контролер.

Конкретні реалізації та визначення даного патерну можуть відрізнятися, але в силу своєї гнучкості і простоти він став дуже популярним останнім часом, особливо в сфері веб-розробки.

ASP.NET має наступні функціональні можливості:

					КНТЕУ 121 06-03.МР	Аркуш
						23
Зм.	Аркуш	№ докум	Підпис	Дата		

1. Простота розгортання. Розгортання ASP.NET додатків виконується шляхом копіювання файлів програми в спеціальну папку на web сервері. Перезапуск web-сервера не потрібно;
2. Засоби безпеки. Розробник ASP.NET може використовувати в своєму додатку будь-яку із запропонованих типових схем авторизації і аутентифікації користувачів;
3. Підтримка національних мов. ASP.NET використовує Unicode і розробники мають можливість застосовувати в своїх проектах національні алфавіти;
4. Висока продуктивність. ASP.NET має справу зі скомпільованим кодом. Завдяки цьому ASP.NET отримує можливість ефективно використовувати різні механізми оптимізації коду;
5. Підтримка мобільних пристроїв. ASP.NET підтримується будь-яким браузером, запущеним на будь-якому пристрої (як заявляє Microsoft);
6. Можливості налагодження. ASP.NET забезпечує можливість трасування і налагодження коду додатків;
7. Інтеграція з .NET Framework. ASP.NET є частиною платформи .NET Framework. Розробники можуть використовувати можливості, що надаються цією платформою при створенні додатків;

ASP.NET містить безліч готових елементів управління, застосовуючи які, можна швидко створювати інтерактивні web-додатки. Загалом, можливості ASP.NET обмежені тільки уявою.

2.2.2. SQL Server

Microsoft SQL Server в якості мови запитів використовує версію SQL, що отримала назву Transact-SQL (скорочено T-SQL), що є реалізацією SQL-92 (стандарт ISO для SQL) з множинними розширеннями. TSQL дозволяє використовувати додатковий синтаксис для збережених процедур і забезпечує

					<i>KHTEY 121 06-03.MP</i>	Аркуш
						24
Зм.	Аркуш	№ докум	Підпис	Дата		

підтримку транзакцій (взаємодія бази даних з керуючим додатком). Microsoft SQL Server і Sybase ASE для взаємодії з мережею використовують протокол рівня додатка під назвою Tabular Data Stream (TDS, протокол передачі табличних даних). Протокол TDS також був реалізований в проєкті FreeTDS з метою забезпечити різним додаткам можливість взаємодії з базами даних Microsoft SQL Server і Sybase.

Microsoft SQL Server також підтримує Open Database Connectivity (ODBC) – інтерфейс взаємодії додатків з СУБД. Версія SQL Server 2014 забезпечує можливість підключення користувачів через веб-сервіси, що використовують протокол SOAP. Це дозволяє клієнтським програмам, не призначеним для Windows, кроссплатформенно з'єднуватися з SQL Server. Microsoft також випустила сертифікований драйвер JDBC, що дозволяє додаткам під керуванням Java (таким як BEA і IBM WebSphere) з'єднуватися з Microsoft SQL Server.

SQL Server підтримує віддзеркалення і кластеризації баз даних. Кластер сервера SQL – це сукупність однаково конфігурованих серверів; така схема допомагає розподілити робоче навантаження між декількома серверами. Всі сервера мають одне віртуальне ім'я, і дані розподіляються по IP-адресам машин кластеру протягом робочого циклу. Також в разі відмови або збою на одному з серверів кластера доступний автоматичний перенос навантаження на інший сервер.

У SQL Server вбудована підтримка .NET Framework. Завдяки цьому, збережені процедури БД можуть бути написані на будь-якій мові платформи .NET, використовуючи повний набір бібліотек, доступних для .NET Framework, включаючи Common Type System (система поводження з типами даних в Microsoft .NET Framework). Однак, на відміну від інших процесів, .NET Framework, будучи базисною системою для SQL Server, виділяє додаткову пам'ять і вибудовує засоби управління SQL Server замість того, щоб

					KHTEY 121 06-03.MP	Аркуш
						25
Зм.	Аркуш	№ докум	Підпис	Дата		

використовувати вбудовані засоби Windows. Це підвищує продуктивність в порівнянні з загальними алгоритмами Windows, так як алгоритми розподілу ресурсів спеціально налаштовані для використання в структурах SQL Server.

2.2.3. AJAX

Коли існуючих можливостей стало мало, а вдосконалювати існуюче вже нікуди, тоді і відбувається технологічний прорив. Таким проривом і є AJAX (Asynchronous JavaScript and XML) - підхід до побудови призначених для користувача інтерфейсів веб-додатків, при якому web-сторінка, що не перезавантажується, сама довантажує потрібні користувачу дані. AJAX - один з компонентів концепції DHTML.

Що ж дає ця технологія. В даний час розробка WEB додатків прагне до розмежування клієнтської частини і серверної. При розробці складних проектів виникає необхідність в структурованості і легкості читання коду. Не слід засмічувати код програміста кодом верстальника, а код верстальника - правками дизайнера, і так далі.

Виникає необхідність в розмежуванні роботи. Так, наприклад, дизайнер буде робити свою роботу, верстальник свою, програміст свою, і при цьому ніхто один одному заважати не буде. У підсумку кожному учаснику проекту досить буде знати тільки ті дані, з якими йому доведеться працювати. В такому випадку продуктивність групи і якість проекту підвищується в рази.

Отже, AJAX - це ідея, яка базується на двох основних принципах:

1. Використання DHTML для динамічної зміни змісту сторінки;
2. Використання XMLHttpRequest для звернення до сервера "на льоту".

Використання цих двох підходів дозволяє створювати набагато зручніші WEB-інтерфейси користувача на тих сторінках сайтів, де необхідна активна взаємодія з користувачем. Використання Ајах стало найбільш популярне після того, як компанія Google почала активно використовувати його при створенні

					КНТЕУ 121 06-03.МР	Аркуш
						26
Зм.	Аркуш	№ докум	Підпис	Дата		

своїх сайтів, таких як Gmail, Google maps і Google suggest. Створення цих сайтів підтвердило ефективність використання даного підходу.

Отже докладніше: якщо взяти класичну модель WEB-додатки(рис. 2.4)

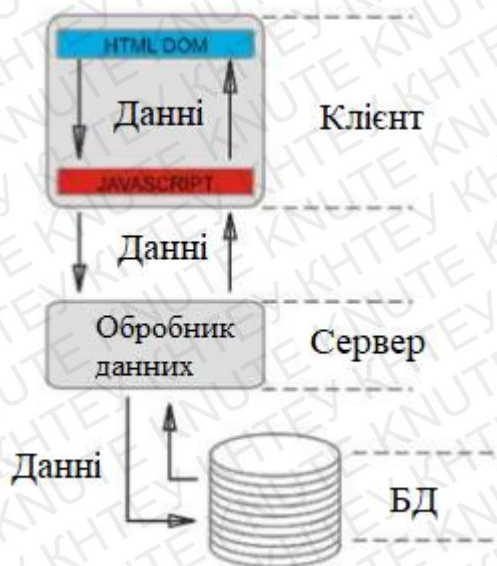


Рис. 2.4. Класична модель Web-додатку

Клієнт, набираючи в рядку пошуку адресу ресурсу який його цікавить, потрапляючи на сервер, робить до нього запит. Сервер робить обчислення за результатами запитів, звертається до бази даних і так далі, після чого отримані дані йдуть клієнтові і, в разі необхідності підставляються в шаблони і обробляються браузером. Результатом є сторінка, яку ми бачимо, і яку 80% населення країни які перебувають в WEB називають Інтернетом. Це класична модель, яка встигла себе зарекомендувати і заслужити собі почесне місце під сонцем. Це найпростіша модель взаємодії і, як наслідок, найпоширеніша.

При зверненні до сервера, генерується сторінка, яка буде відображатися користувачеві, і пропонувати йому зробити потрібну йому послідовність дій. При свідомому (хоча і не обов'язково) виборі клієнта, його запит буде звертатися до AJAX модулю, який і буде виробляти всі потрібні йому обчислення і роботу з сервером як таким. Основна відмінність в тому що цей

метод дає нам можливість динамічно звертатись до сервера і виконувати цікаві дії. Наприклад, потрібно виконати звернення до бази даних і отримати цікаві дані, які потім можна використовувати.

2.2.4.Entity Framework Core (EF Core)

Entity Framework Core (EF Core) являє собою об'єктно-орієнтовану, неважку і розширювану технологію від компанії Microsoft яка дає доступ до даних. EF Core являється ORM-інструментом (object-relational mapping - відображає дані на реальних об'єктах). Тобто EF Core дає можливість працювати базами даних, але являє собою більш високий рівень абстракції: EF Core дозволяє абстрагуватися від самої бази даних та її таблиць і працювати з даними незалежно від використовуваного типу сховища. Якщо на фізичному рівні можна оперувати таблицями, індексами, первинними і зовнішніми ключами, але на концептуальному рівні, який пропонує Entity Framework, йде робота з об'єктами.

Entity Framework Core підтримує чимало різноманітних систем баз даних. Таким чином, можна через EF Core працювати з будь-якою СУБД, якщо для неї є потрібний провайдер.

Як технологія доступу до даних Entity Framework Core може використовуватись на різних платформах стека .NET. Це та стандартні платформи типу Windows Forms, консольні додатки, WPF, UWP і ASP.NET Core. При цьому кроссплатформенна природа EF Core дозволяє задіяти її не тільки на ОС Windows, але і на Linux і Mac OS X.

Платформу Entity Framework Core можна застосовувати в різних технологіях стека .NET. В даному випадку можна розглянути базові моменти платформи на прикладі консольних додатків, як найбільш простих і не містять ніякого зайвого коду.

					<i>КНТЕУ 121 06-03.МР</i>	Аркуш
						28
Зм.	Аркуш	№ докум	Підпис	Дата		

Для створення додатку з використанням Entity Framework Core потрібно створити в Visual Studio 2019 новий проект по типу Console App (.NET Core) (порядок дій зображено на рис. 2.5, рис. 2.6):

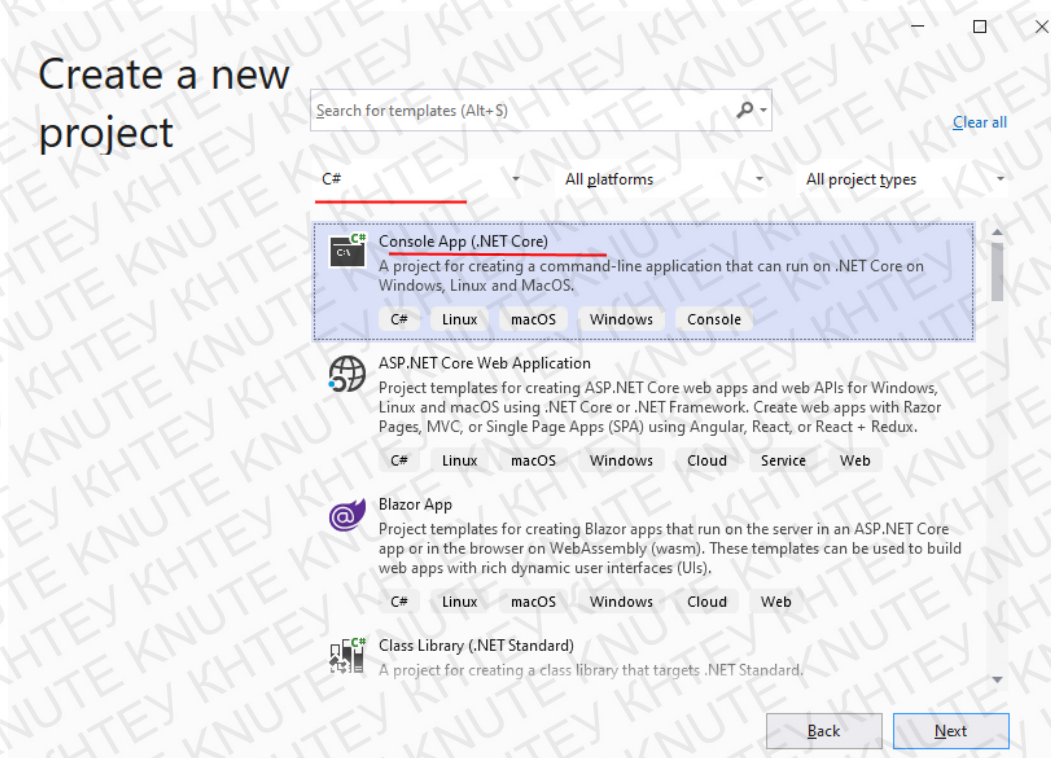


Рис. 2.5. Створення нового проекту

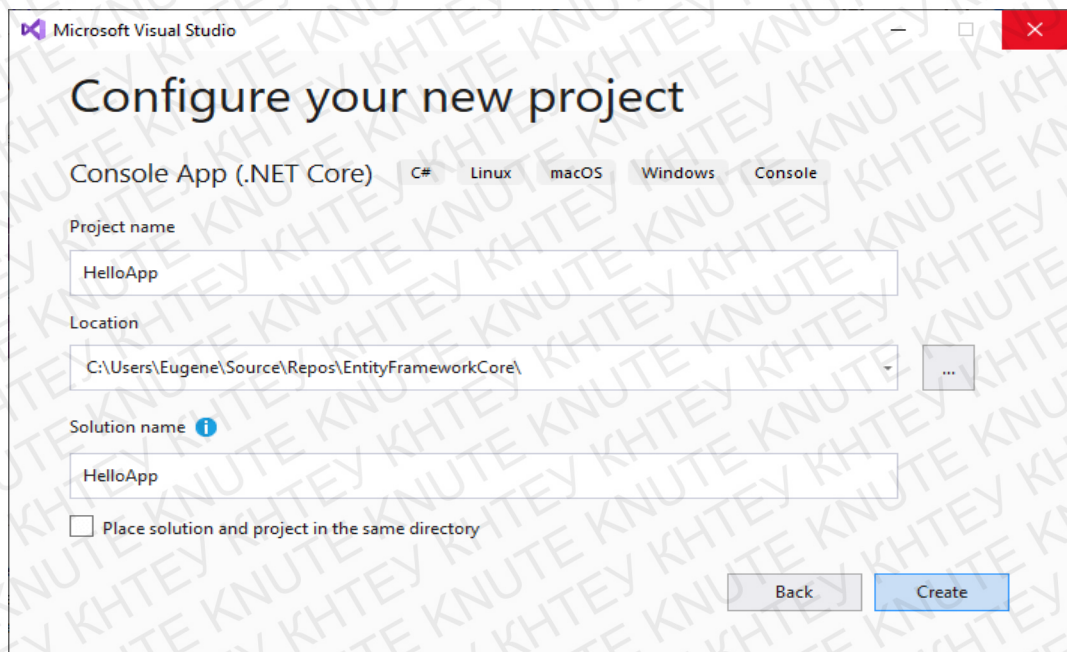


Рис. 2.6. Конфігуратор нового проекту

					КНТЕУ 121 06-03.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		29

Отже, Visual Studio створює проект з базової примітивною структурою, де можна виділити власне файл логіки додатка - файл Program.cs. І щоб почати працювати з EntityFramework Core, необхідно спочатку додати в проект пакет EntityFramework Core. Для цього потрібно перейти в проекті до пакетного менеджера NuGet.

Однак тут потрібно шукати не загальний пакет для Entity Framework Core, а пакет для конкретної СУБД. Так, в даному випадку буде використовуватись MS SQL Server в якості СУБД, тому треба шукати пакет Microsoft.EntityFrameworkCore.SqlServer (рис. 2.7):

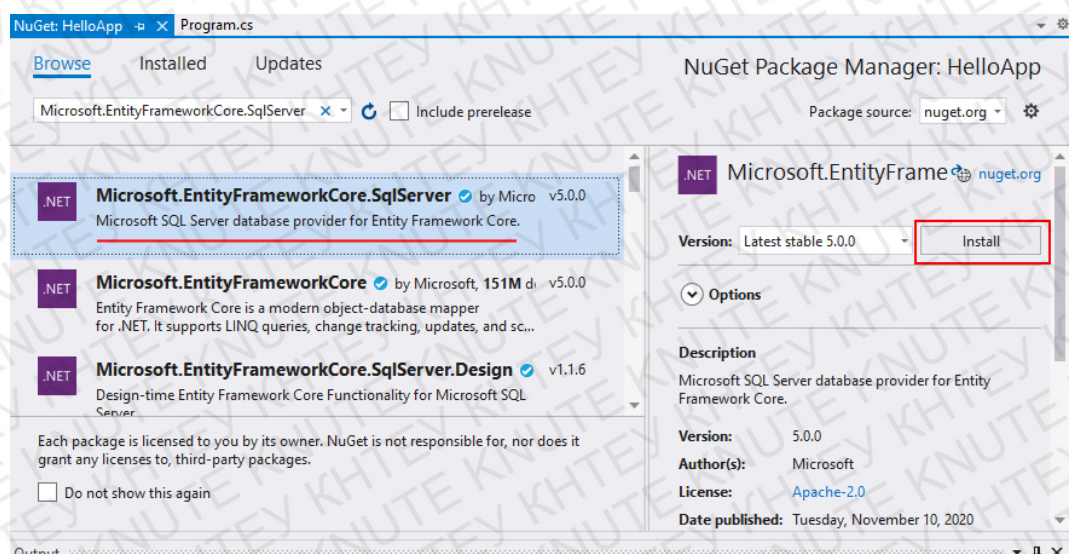


Рис. 2.7. Додавання пакету EntityFramework

В якості альтернативи для додавання пакетів можна використовувати Package Manager Console. Для цього в меню Visual Studio потрібно перейти до пункту Tools -> NuGet Package Manager -> Package Manager Console

У відкритому внизу в Visual Studio вікні Package Manager Console ввести команду:

Install-Package Microsoft.EntityFrameworkCore.SqlServer

Після виконання цієї команди потрібно виконати другу команду:

Install-Package Microsoft.EntityFrameworkCore.Tools

Все, необхідні пакети додані. Тепер можна їх використовувати.

					KHTEY 121 06-03.MP	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		30

2.3 Висновки до розділу

Таким чином для отримання максимальної ефективності в роботі підприємства з клієнтами, найкращим шляхом до досягнення змін в кращу сторону буде прийняття на озброєння CRM-системи, які допоможуть автоматизувати чималу кількість робочих процесів, спростити та покращити роботу з значною кількістю клієнтів, яка в свою чергу підніме рівень лояльності споживачів до обраного підприємства, зміцнить відносини між клієнтом та підприємством та налагодить стабільну та довгострокову між ними співпрацю.

Для реалізації поставленої мети буде створено власну CRM-систему, яка буде створена на операційній системі Windows 10, використовуючи середовище розробки Visual Studio 2019 на мові C# з використанням SQL платформ та EF Core технологій.

					<i>KHTEU 121 06-03.MP</i>	Аркуш
						31
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 3

РЕАЛІЗАЦІЯ СИСТЕМИ

3.1. Опис інтерфейсу користувача

Інтерфейс користувача – інтерфейс, за допомогою якого користувач в силах керувати програмним забезпеченням чи апаратним оснащенням. Інтерфейс користувача повинен бути максимально комфортним в користуванні та інтуїтивно зрозумілим для будь якої людини, яка ним користується.

Після запуску локального хостингу та запуску створеної CRM-системи, можна отримати результат, який зображений на рис. 3.1.

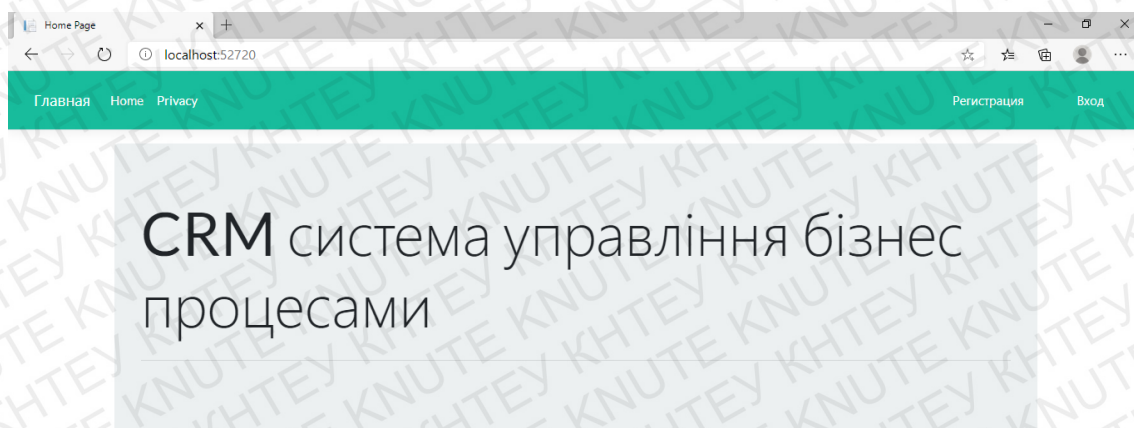


Рис. 3.1. Головне вікно програми

Інтерфейс не багатий, але це і не потрібно в роботі. Головне, щоб нічого не відволікало від робочого процесу.

Далі існує два шляхи:

1. Реєстрація нового користувача
2. Вхід у вже існуючий обліковий запис

					<i>КНТЕУ 121 06-03.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка CRM-системи управління бізнес-процесами	Стадія	Аркуш	Аркушів
Зав. каф.		Криворучко О.В.		19.10.20		РЗ	32	51
Керівник		Рзаєва С.Л.		19.10.20		Факультет інформаційних технологій 2м курс, 6 група		
Гарант		Криворучко О.В.		19.10.20				
Розробив		Вороневич Р.О.		19.10.20	Реалізація системи			

Для демонстрації реєстрації можна її пройти, що в наслідок цих дій доавить нового робітника в загальну базу (рис. 3.2).

Для цього потрібно ввести:

1. ПІБ
2. Власну пошту
3. Вигадати пароль
4. Підтвердити вигаданий пароль
5. Ввести свій номер телефону

Далі обрати робочу роль і натиснути «Зарегистрироваться».

Реєстрація завершена.

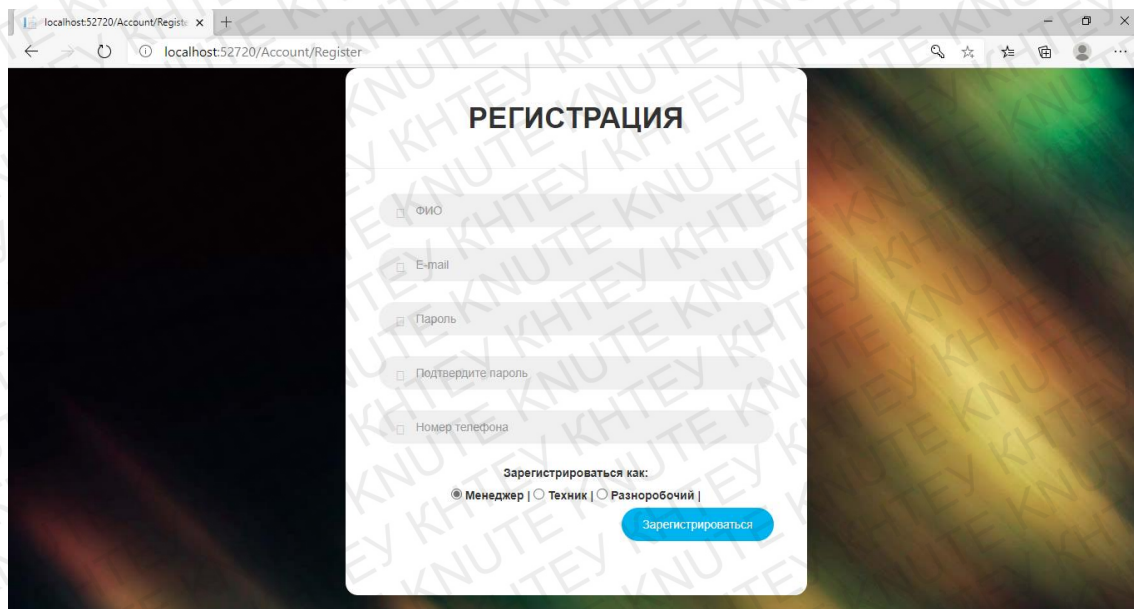


Рис. 3.2. Вікно реєстрації

Якщо обліковий запис вже існує, то достатньо лише авторизуватись, щоб розпочати роботу.

В програмі визначено чотири типу користувачів: Адміністратор, Менеджер, Технік, Різнорабочий.

Менеджер – відповідає за роботу з клієнтами, замовленнями та створенням задач для інших користувачів.

Технік – відповідає за всі нюанси, які пов’язані з замовленням.

					КНТЕУ 121 06-03.МР	Аркуш
						33
Зм.	Аркуш	№ докум	Підпис	Дата		

Різноробочий – власне виконує задачі поставлені менеджером та затверджені техніком, додаток використовує лише для нагадування.

Адміністратор – відповідає за загальну роботу всіх робітників, може керувати правами доступу користувачів, налаштуваннями для додатку, слідує за роботою та графіками.

3.1.1 Робочий простір менеджера

Зайшовши на аккаунт менеджера та перейшовши на його робочий простір, можна отримати вікно, зображене на рис.3.3

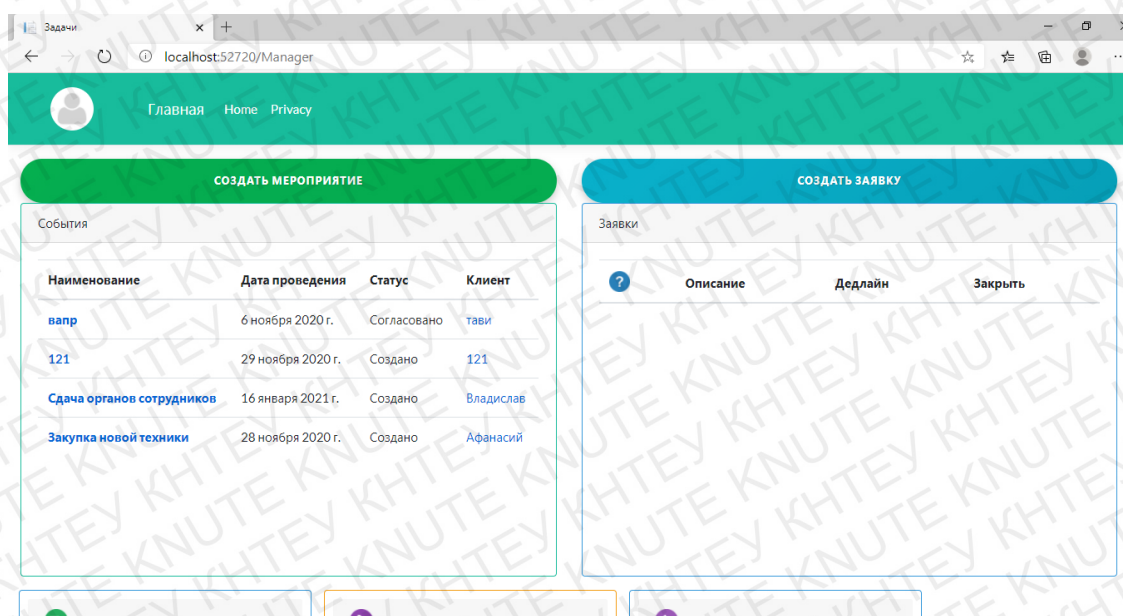


Рис. 3.3. Робоче вікно менеджера

В його силах створювати заходи нові заявки. Якщо прогорнути трішки вниз, можна побачити його завдання та час, який відведено для закінчення цих операцій (рис. 3.4)

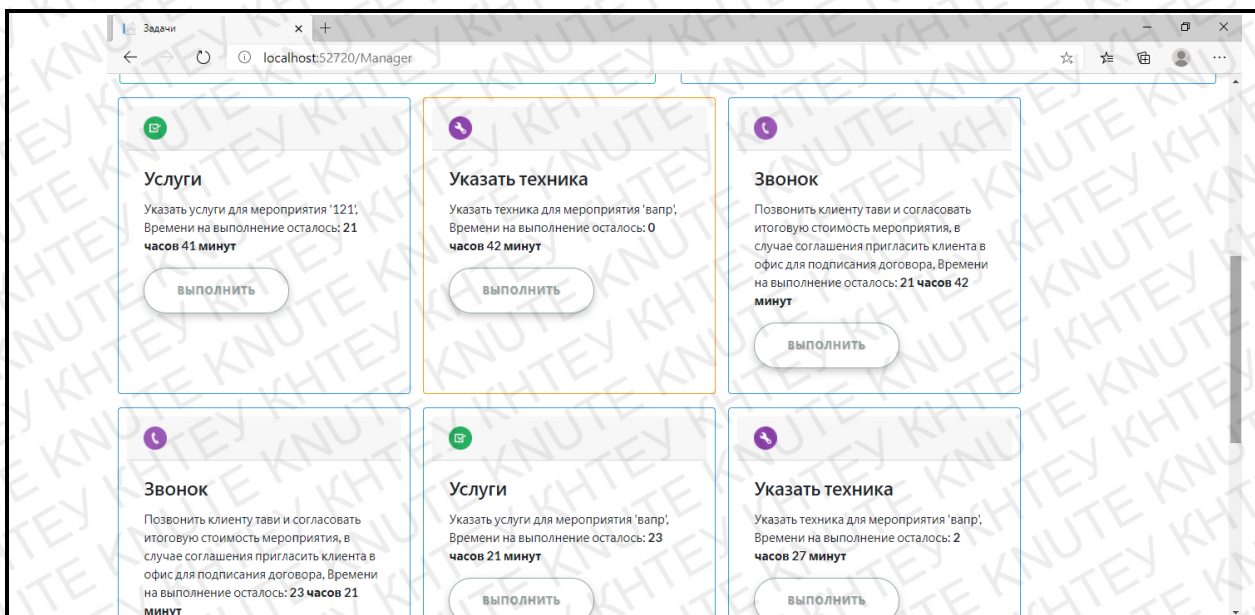


Рис. 3.4. Сповіщення про завдання менеджера

Для створення нових заходів потрібно натиснути «Создать мероприятие» та в новому вікні, де можна ввести всю необхідну інформацію заповнити необхідні поля (рис. 3.5)

Рис. 3.5. Вікно створення нового заходу

Далі отримана інформація додається до бази і далі можна з нею більш детально працювати.

Відкривши захід натисненням на його назву можна отримати детальну інформацію про захід, додати певні послуги, які створюються та корегуються адміністратором, створити для подальшої роздруківлі договір та призначити відповідальних осіб (рис. 3.6)

Название:	Закупка новой техники						
Описание:	Покупка игровых ковриков						
Статус:	Создано						
Тип мероприятия:	Массовое						
Место проведения:	Ближайший магазин						
Дата проведения:	28 ноября 2020 г.						
Время начала:	15:58						
Дата поступления:	21.11.2020						
Услуги:	<table border="1"> <thead> <tr> <th>Наименование</th> <th>Кол-во</th> <th>Сумма</th> </tr> </thead> <tbody> <tr> <td>Добавить</td> <td></td> <td></td> </tr> </tbody> </table>	Наименование	Кол-во	Сумма	Добавить		
Наименование	Кол-во	Сумма					
Добавить							
Договор:	Создать						
Менеджер:	Менеджер						
Клиент:	Афанасий						
Техник:	Указать						

Рис. 3.6 Робоча область по конкретному заходу

Для максимізації ефективності в роботі з клієнтом, менеджер може відкрити детальну інформацію про клієнта(натиснувши на його ім'я відкриється нове вікно), а саме його ім'я, номер телефону, його пошту, ім'я менеджера, що його оформив, детальні данні про нього та захід який з ним буде обговорюватись, щоб з більшою вірогідність провести успішну угоду (рис. 3.7).

Клиент ID=5	
Имя	Афанасий
Телефон	324236543343
Email	24325@gmail.com
Менеджер	Менеджер
Описание	Любит котиков
Мероприятия	Закупка новой техники

Рис. 3.7. Візуалізація існуючої інформації про клієнта

3.1.2. Робочий простір техніка

При переході на профіль техніка та натиснувши зліва на кнопку «Уведомления», вискакує вікно з новими завданнями для конкретного робітника (рис. 3.8).

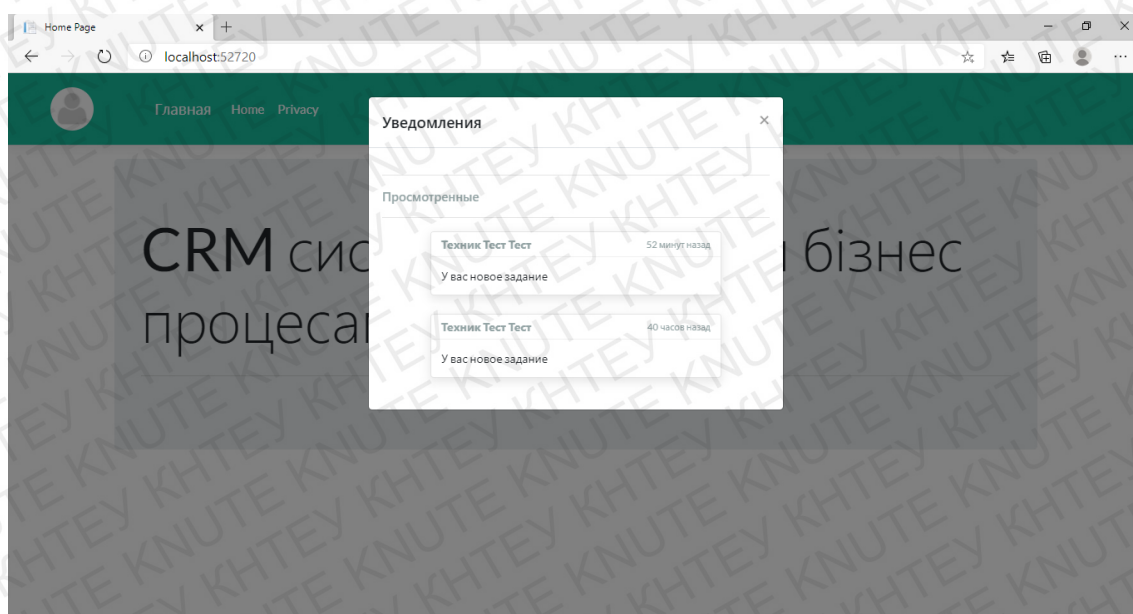


Рис. 3.8. Вікно сповіщень

Перейшовши в робочу зону техніка буде доступно перелік завдань, які в скорому часі потрібно виконати (рис. 3.9).

					<i>КНТЕУ 121 06-03.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		37

Технік може бачити конкретне завдання, та має доступ до повної інформації про клієнта та заходу, який за ним закріплений.

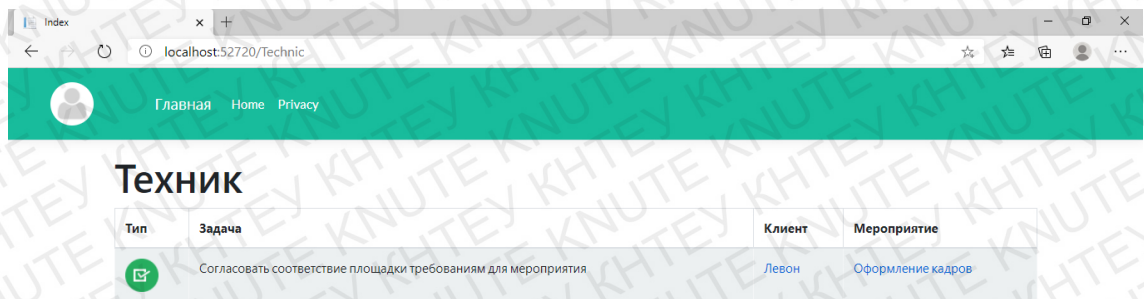


Рис. 3.9. Робоча зона техніка

Після виконання підготовчих робіт, технік, відштовхуючись від отриманої інформації може відписати менеджеру та зробити вибір підтвердити, або відмовитись від проведення запланованих заходів (рис. 3.10).

Название:	Закупка новой техники		
Услуги:			
Сообщение для менеджера	Самі цим займайтесь ПОДТВЕРДИТЬ ОТКЛОНИТЬ		

Рис. 3.10. Результат роботи техніка

3.1.3. Робочий простір різноробочого

Перейшовши на профіль різноробочого, як завжди, його зустрічають сповіщення, які накопились за останній час (рис. 3.11).

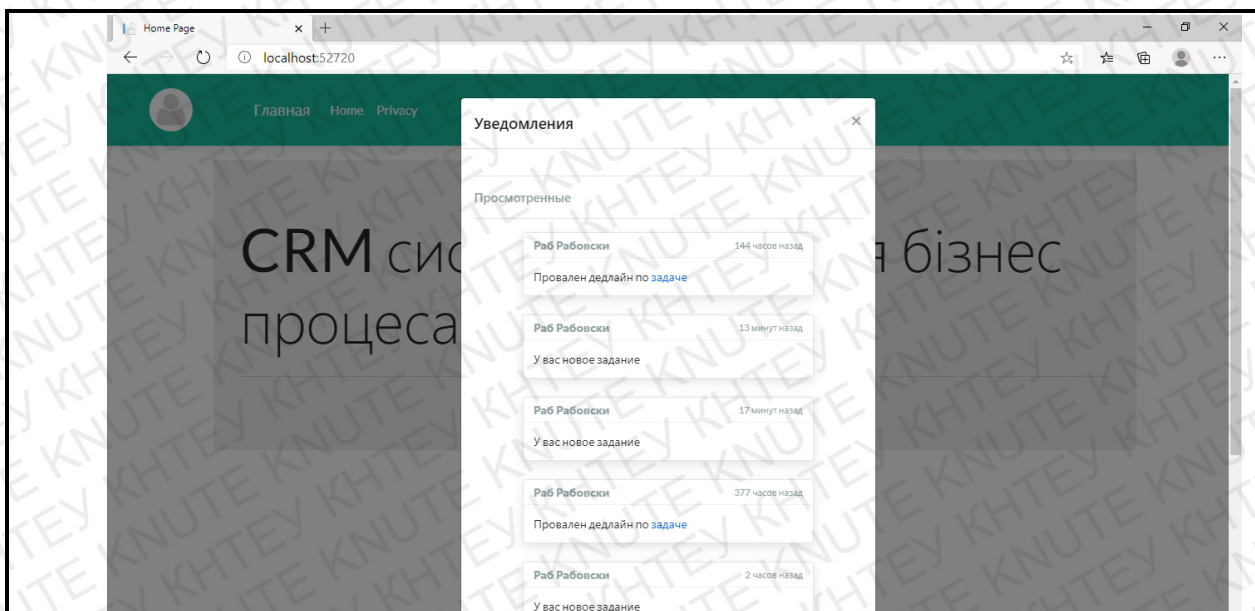


Рис. 3.11. Вікно сповіщень різноробочого

Як оговорювалось раніше, робоча область в різноробочого не велика. Додаток використовується лише для нагадування та отримання оповіщень. Він може лише переглядати заявки, які чекають підтвердження техніка та вже підтверджені заходи, звісно, теж маючи доступ до інформації про захід, який його очікує (рис. 3.12).

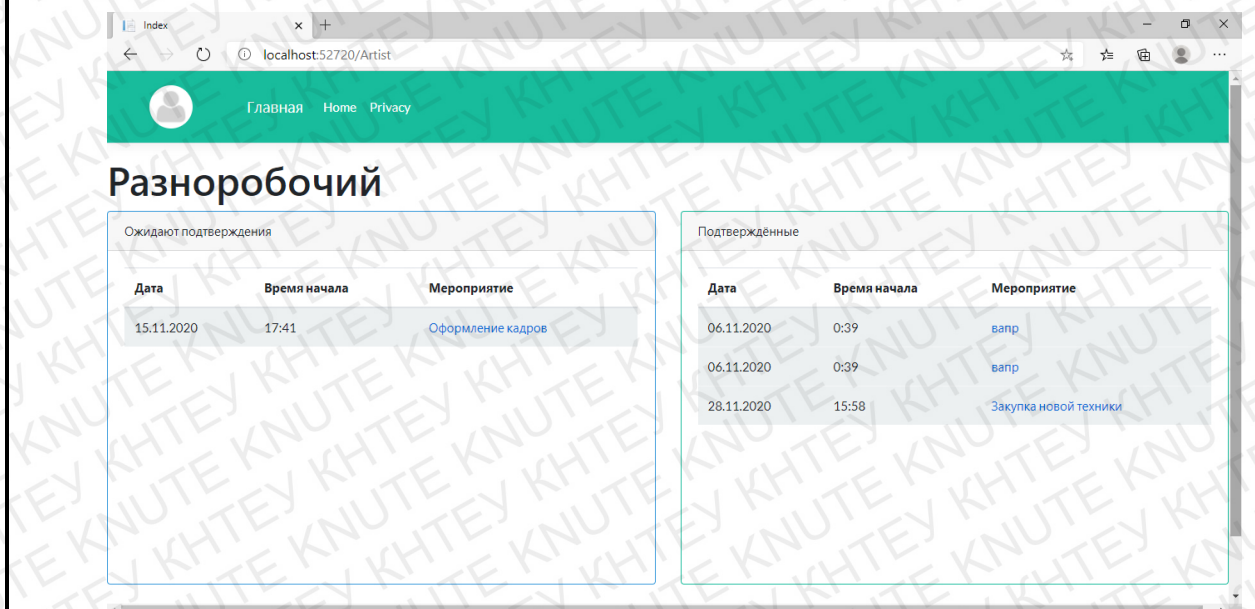


Рис. 3.12. Робоча область різноробочого

3.1.4. Робочий простір адміністратора

Перейшовши на профіль адміністратора, зустрічає можливість переглядати оповіщення всіх користувачів, яка доступна лише адміністратору (рис. 3.13).

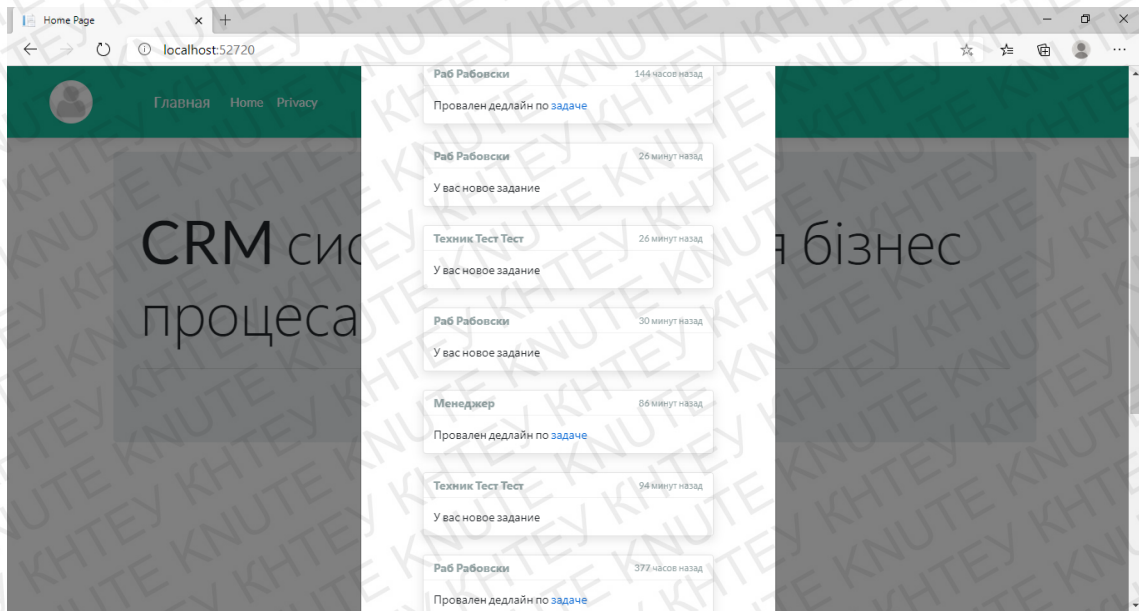


Рис. 3.13. Оповіщення адміністратора

Натиснувши на логотип чоловічка, який знаходиться зліва зверху, можна перейти до керування ролями, отримавши наступне вікно, де можна створювати нові ролі, переглядати список вже існуючих користувачів та змінювати їх права доступу (рис. 3.14).

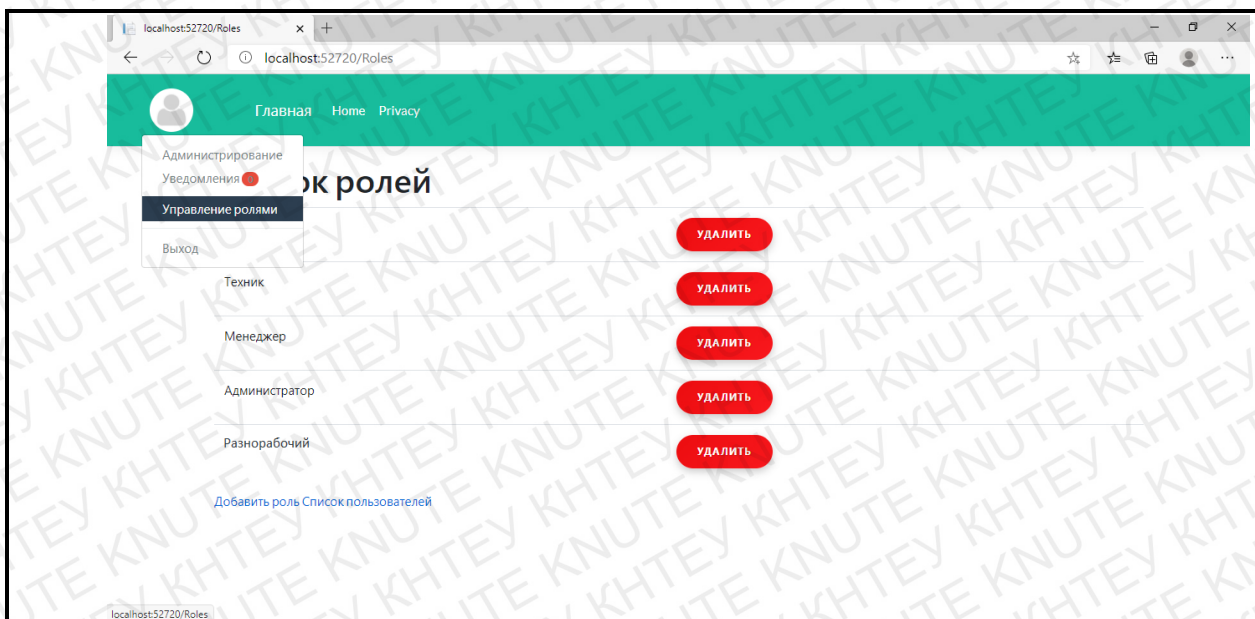


Рис. 3.14. Вікно керування ролями

Натиснувши по вкладці адміністрування переходить на робочий простір адміністратора, в якого є детальна статистика представлена в графіках, де можна побачити в наслідок яких причин в більшості випадків виходять відмови, які послуги користуються найбільшою популярністю, заходи, які найчастіше замовляють та успішність менеджерів в укладанні вдалих договорів (рис. 3.15). Може переглядати графік, де зображено на яку суму було оформлено договори по днях (рис. 3.16).

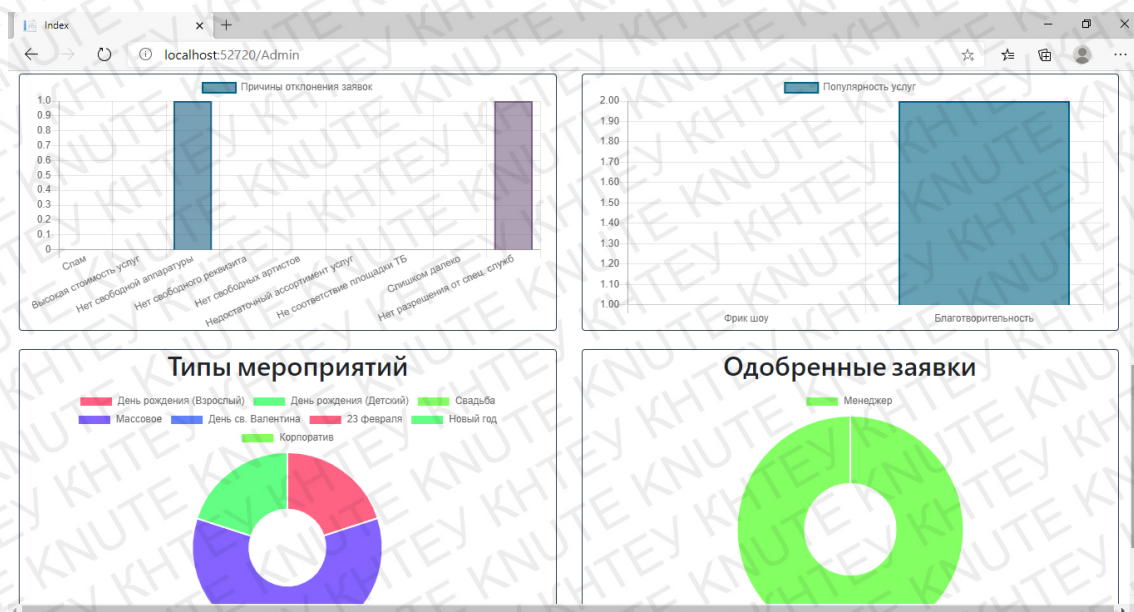


Рис. 3.15. Детальна статистика в графіках

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	KHTEY 121 06-03.MP	41

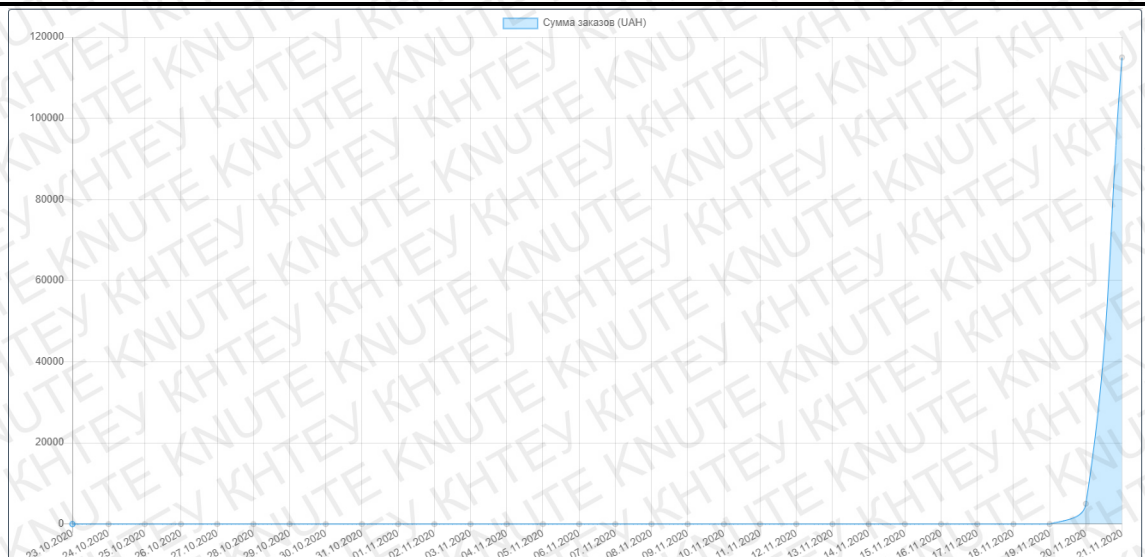


Рис. 3.16. Графік ефективності роботи

В користуванні адміністратора в головному вікні присутні чотири кнопки з різним наповненням (рис. 3.17).

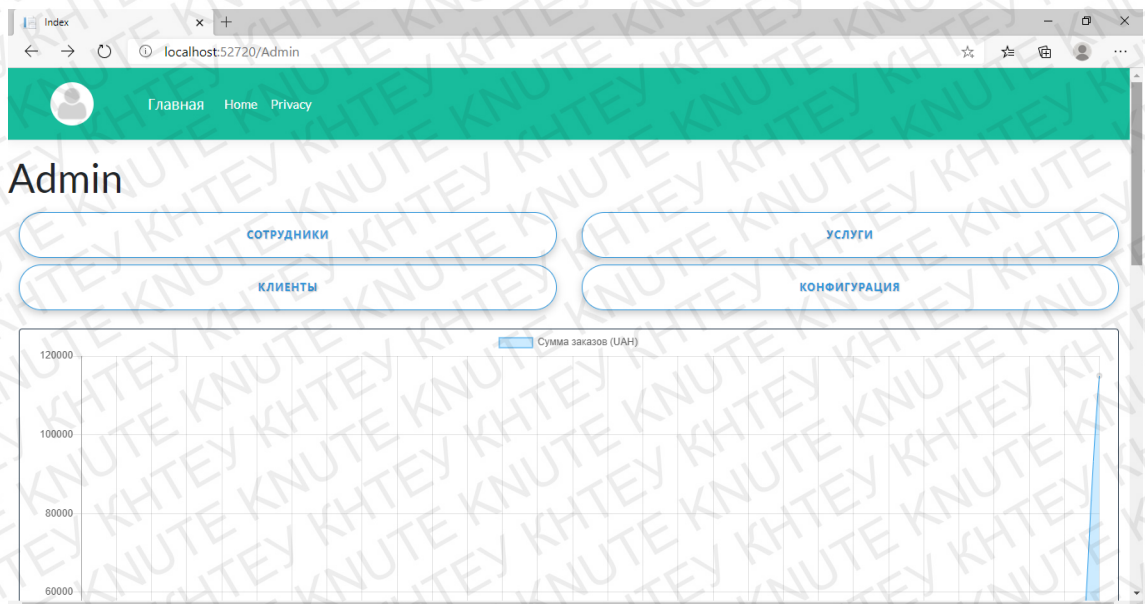


Рис. 3.17. Робоча область адміністратора

Перейшовши по першій кнопці можна отримати результат, де зображені всі працівники, та надана вся потрібна інформація про них (рис. 3.18).

					КНТЕУ 121 06-03.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		42

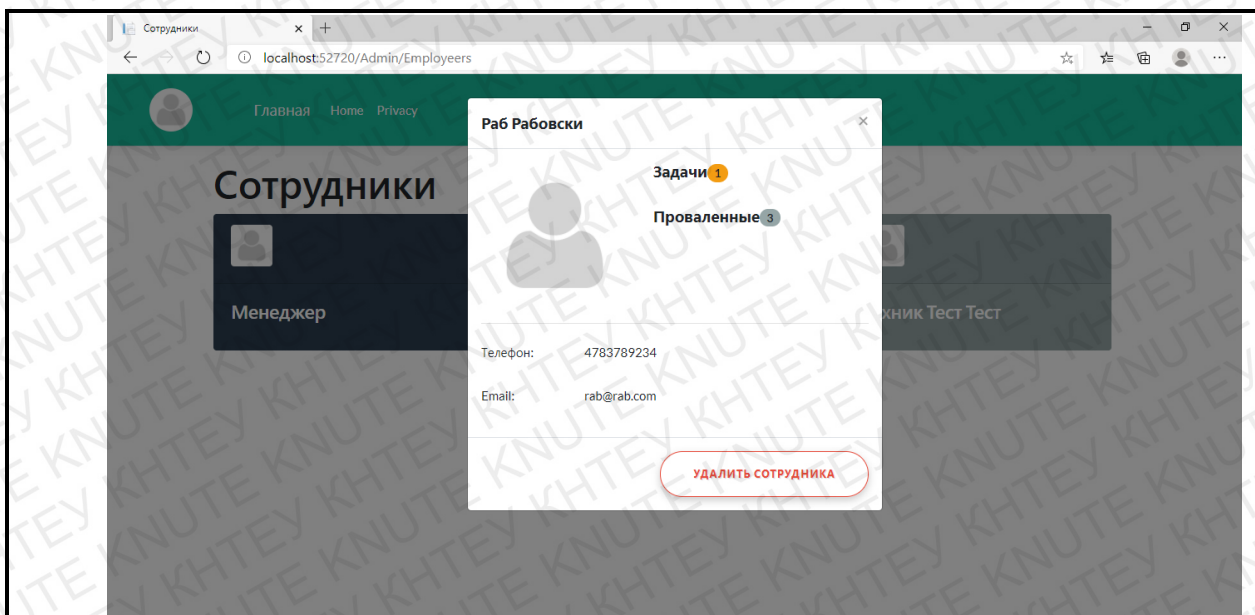


Рис. 3.18. Вікно «Сотрудники»

Натиснувши кнопку «Услуги» перейде у вікно, де є можливість керувати послугами. Можна як створювати нові послуги з детальним описом та створенням ціни, так і видаляти існуючі (рис. 3.19).

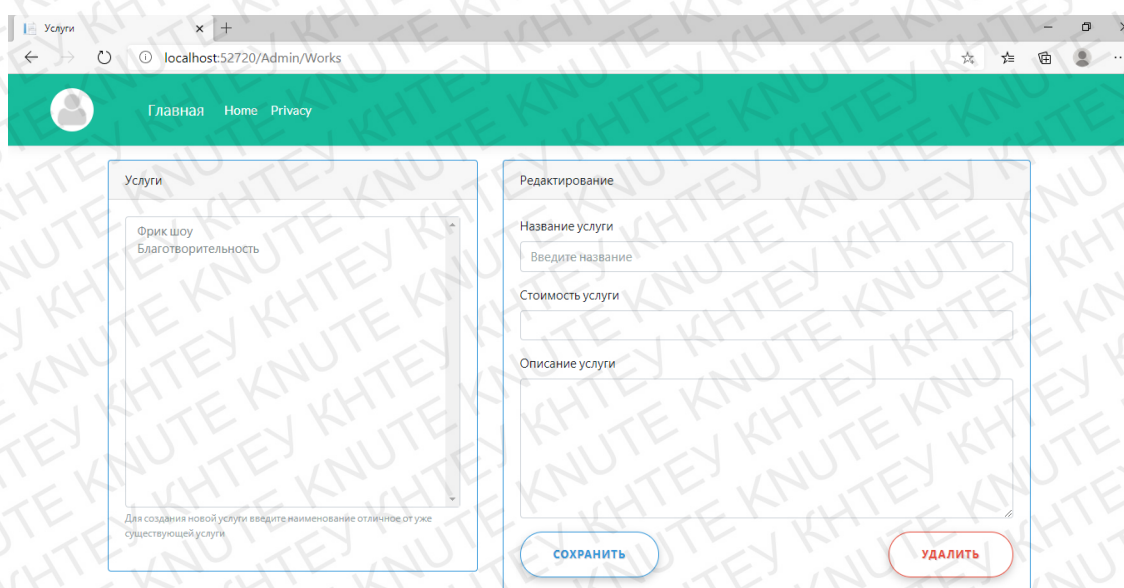


Рис. 3.19. Вікно управління послугами

					КНТЕУ 121 06-03.МР		Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			43

По вкладці «Клієнти», відкривається база клієнтів з накопиченою інформацією про них, а за вкладкою «Конфігурація» переходить до вікна де можна створювати та змінювати шаблони договорів.

3.2 Розробка і опис програми

Для запуску проекту на локальному сервері необхідно встановити MS SQL Server. База даних автоматично ініціалізується при першому запуску.

Для асинхронного отримання інформації з серверу без перезавантаження сторінки використовувалась функція WebAPI, де в додатку визначений AJAXController (рис. 3.20).

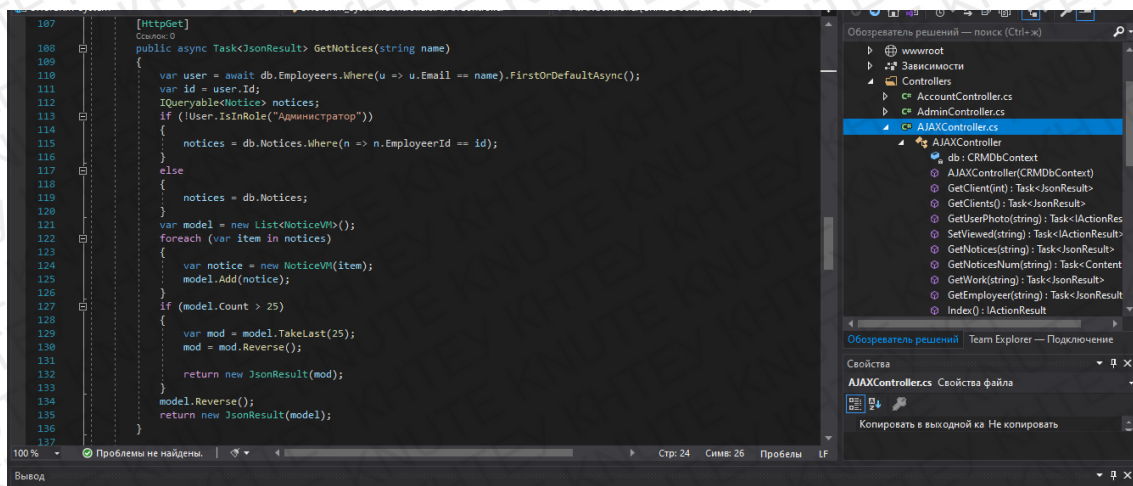


Рис. 3.20. AJAXController

Робота з базою даних здійснюється через EF Core, контекст БД визначений в класі CRMDbContext, який знаходиться в папці Data (рис. 3.21).

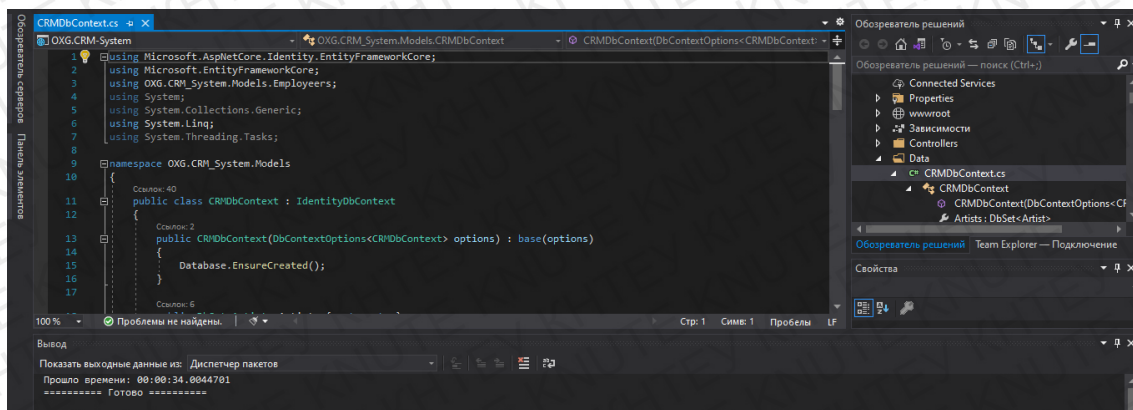


Рис. 3.21. CRMDbContext

Клієнтська частина розроблена з використанням мови HTML (рис. 3.22).

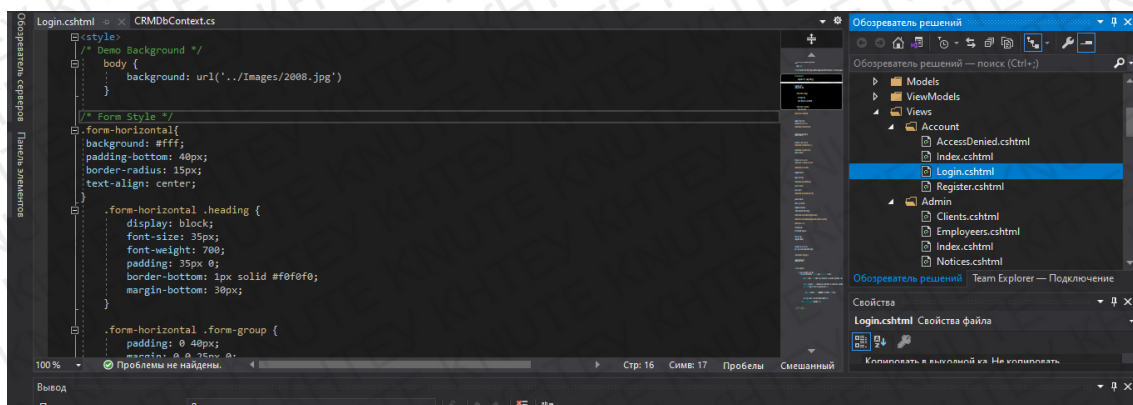


Рис. 3.22. Приклад коду з використанням HTML

Для роботи з додатком був використаний NuGet пакет, який дає можливість переносити дану CRM-систему на інші комп'ютери, для подальшого її використання та можливості редагування (рис. 3.23).

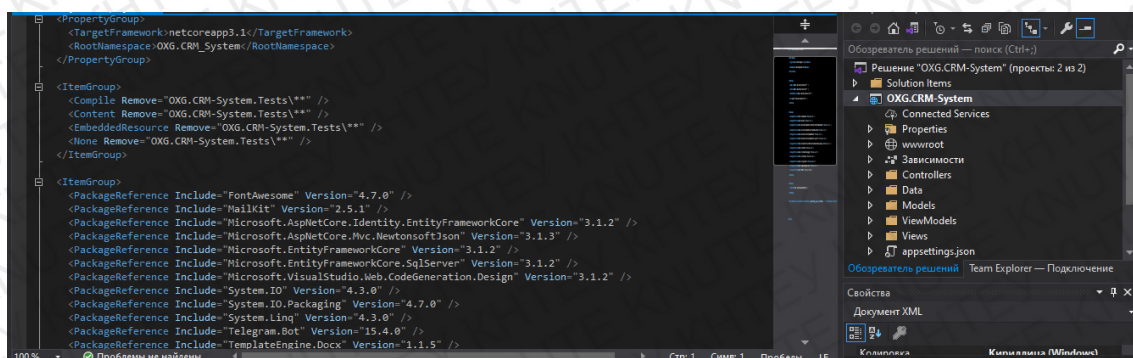


Рис. 3.23. Набір NuGet пакету

3.3 Висновки до розділу 3

Таким чином використання певних новітніх платформ та технологій дало змогу створити якісний прототип CRM-системи управління бізнес процесами з мінімалістичним(невідволікаючим) зовнішнім виглядом, з можливістю розмежування доступу до системи цілих чотирьох видів облікових записів та можливістю їх подальшого збільшення в разі

					Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06-03.МР
					45

необхідності, різним за змістом на насиченістю інтерфейсом для кожного з користувачів, можливість детально та якісно аналізувати всі робочі процеси в реальному часі та за потреби корегувати робочий процес з будь якого місця та пристрою маючи доступ до мережі інтернет.

Незважаючи на зовнішній вигляд, прототип CRM-системи в результаті отримав доволі широкий функціонал, можливість подальшого розширення шляхом додавання нових технологій та здатність до модернізування для задоволення потреб всіх підприємств, на якому він буде використовуватись.

					<i>КНТЕУ 121 06-03.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		46

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Проведено детальний аналіз доцільності використання та подальшого впровадження інформаційних систем на підприємства будь якого напрямку, лідером в ефективності роботи з клієнтами для максимізації ефективності є CRM-система.

Було виявлено, що CRM-система забезпечує доступ до впорядкованої та детальної інформації про клієнта, з яким в конкретний момент часу йде робота, розвиває програму лояльності. Надає можливість працювати в єдиному для всіх інформаційному полі, та забезпечує оперативність даних для забезпечення прийняття найефективніших управлінських рішень. Дозволяє переглядати історію взаємодії з клієнтом та розвивати їх в подальшому для отримання надійних та довгострокових відносин з кожним з клієнтів конкретного підприємства.

Для реалізації програмного продукту було прийняте рішення в використанні Visual Studio 2019, який працює на операційній системі Windows 10, вдалим рішенням в створенні був вибір мови C#, та використання MS SQL Server платформ та EF Core технологій.

В процесі виконання роботи було створено власний прототип CRM-системи, в якому реалізовано розмежований доступ для різних користувачів, індивідуальна робоча зона та зручний інтерфейс.

Великим плюсом створеного прототипу – є можливість змінювати його та підганяти під конкретне підприємство для забезпечення максимальної ефективності в використанні цього продукту.

В даного прототипу є великий потенціал, який, звісно, потрібно реалізовувати, оскільки CRM-системи на даний момент часу дуже затребувані,

					<i>КНТЕУ 121 06-03.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.	Криворучко О.В.			30.10.20	Розробка CRM-системи управління бізнес-процесами	Стадія	Аркуш	Аркушів
Керівник	Рзаєва С.Л.			30.10.20		ВП	47	51
Гарант	Криворучко О.В.			30.10.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив	Вороневич Р.О.			30.10.20				
					Висновки та пропозиції			

дивлячись на ситуацію зараз дуже складно вести бази в паперовому вигляді відштовхуючись від їх масивності та незручності в використанні, не кажучи вже про доступність для всіх робітників підприємства. Найближчими роками CRM-системи набиратимуть оберти та розповсюджуватимуться з неймовірною швидкістю. Саме тому була створена CRM-система з можливістю модернізації, для її більшої універсальності та можливістю захоплення більшого ринку потенційних користувачів в вигляді підприємств різного типу зайнятості та розміру.

					<i>КНТЕУ 121 06-03.МР</i>	Аркуш
						48
Зм.	Аркуш	№ докум	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Что такое Entity Framework Core [Електронний ресурс]. – Режим доступу: <https://metanit.com/sharp/entityframeworkcore/1.1.php>
2. Установка клиентских средств NuGet [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/ru-ru/nuget/install-nuget-client-tools>
3. Внедрение CRM [Електронний ресурс]. – Режим доступу: <http://it-capital.info/crm/>
4. Робота з crm системою, що це – ЦРМ розшифровка [Електронний ресурс]. – Режим доступу: <https://vzyatinfo.biz.ua/roboata-z-crm-sistemoju-shho-ce-crm-rozshifrovka/>
5. Подход к выбору crm [Електронний ресурс]. – Режим доступу: <http://www.mashportal.ru/Default.aspx?tabid=122&newsType=ArticleView&ArticleID=154&PageID=165>
6. Краткое руководство. Установка и использование пакета в Visual Studio [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/ru-ru/nuget/quickstart/install-and-use-a-package-in-visual-studio>
7. Что такое ASP.NET. Инсталляция и тестовый проект. [Електронний ресурс]. – Режим доступу: <https://www.internet-technologies.ru/articles/lekciya-1-chto-takoe-asp-net-installyaciya-i-testovyy-proekt.html>
8. История языка программирования C# [Електронний ресурс]. – Режим доступу: <https://www.interestprograms.ru/article-history-csharp>
9. Краткий обзор языка C# [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/>
10. Краткое руководство. Установка и использование пакета с помощью dotnet CLI [Електронний ресурс]. – Режим доступу:

					<i>КНТЕУ 121 06-03.МР</i>	Аркуш
						49
Зм.	Аркуш	№ докум	Підпис	Дата		

<https://docs.microsoft.com/ru-ru/nuget/quickstart/install-and-use-a-package-using-the-dotnet-cli>

11. Чому вашому відділу продажів потрібна CRM? [Електронний ресурс]. – Режим доступу: <https://www.0432.ua/list/263850>
12. Основи програмування на C # [Електронний ресурс]. – Режим доступу: <http://jak.koshachek.com/articles/osnovi-programuvannja-na-c.html>
13. CRM-модели, которые повысят лояльность клиентов [Електронний ресурс]. – Режим доступу: <http://gadgetick.com/newsnew-4605.html>
14. Выборка и проекция из базы данных [Електронний ресурс]. – Режим доступу: <https://metanit.com/sharp/entityframework/4.2.php>
15. Web API [Електронний ресурс]. – Режим доступу: <https://metanit.com/sharp/mvc/12.1.php>
16. AJAX [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/AJAX>
17. Краткое руководство. Создание веб-приложения ASP.NET Core в Azure [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/ru-ru/azure/app-service/quickstart-dotnetcore?pivots=platform-windows#launch-the-publish-wizard?utm_source=aspnet-start-page&utm_campaign=vside
18. Створення додатку типу Windows Forms Application у Microsoft Visual Studio [Електронний ресурс]. – Режим доступу: <https://www.bestprog.net/uk/2015/12/17/001-%D1%81%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F-%D0%B4%D0%BE%D0%B4%D0%B0%D1%82%D0%BA%D1%83-%D1%82%D0%B8%D0%BF%D1%83-windows-forms-application-%D1%83-microsoft-visual-studio-2010/>

					КНТЕУ 121 06-03.МР	Аркуш
						50
Зм.	Аркуш	№ докум	Підпис	Дата		

19. Особливості мови програмування C # та бази даних [Електронний ресурс]. – Режим доступу: https://studwood.ru/1618507/informatika/osoblivosti_movi_programuvannya_bazi_danih
20. Новые возможности C# 8.0 [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/ru-ru/dotnet/csharp/whats-new/csharp-8>
21. Hypertext Transfer Protocol - HTTP [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/ru/docs/Web/HTTP>

					<i>КНТЕУ 121 06-03.МР</i>	Аркуш
						51
Зм.	Аркуш	№ докум	Підпис	Дата		

ТЕХНІЧНЕ ЗАВДАННЯ

Однією з найголовніших заporук відповідності вимогам та високої якості розробленого програмного рішення є грамотно спроектоване до початку розробки технічне завдання. Отже, далі будуть розкриті положення технічного завдання до розробки CRM-системи.

1. Загальні відомості

Назва проекту: «CRM-система».

Планові строки розробки: серпень 2019 – вересень 2019.

Потенційні користувачі: робітники підприємства на чолі з адміністратором.

Призначення програмного рішення: спростити роботу працівникам на різних рівнях та автоматизувати деякі процеси.

Мета створення програмного рішення: розвинути свої навички програмування на C#.

2. Структура проекту

Структура проекту а саме самої системи вказана в пункті «Структура системи». Вона має такий вигляд:



					<i>КНТЕУ 121 06-03.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка CRM-системи управління бізнес-процесами	Стадія	Аркуш	Аркушів
Зав. каф.		Криворучко О.В.		28.02.20		ТЗ	52	51
Керівник		Рзаєва С.Л.		28.02.20		Факультет інформаційних технологій 2м курс, 6 група		
Гарант		Криворучко О.В.		28.02.20				
Розробив		Вороневич Р.О.		28.02.20				
					Технічне завдання			

3. Функціональні вимоги до додатку

Перелік функціональних можливостей визначений базовими діями, що виконує CRM:

- Створення клієнтської бази;
- Фіксування певних подій;
- Розподіл робочих процесів;
- Збереження всіх подій.

4. Вимоги до програмного забезпечення

Розробка серверної та клієнтської частини програмного рішення здійснюватиметься із використанням мови програмування C#, доступом до баз даних за допомогою MS SQL Setver та розробки в середовищі Visual Studio 2019.

5. Вимоги до технічного забезпечення

Найважливішою вимогою до задіяних пристроїв є наявність комп'ютера з операційною системою Windows не нижче Windows 7.

Серверна частина для ПК має виконуватися за наступних мінімальних конфігурацій:

- Microsoft Windows 10/8/7.
- RAM: 4 Гб і більше.
- Processor Intel Core i3 та краще.
- Наявність доступу до мережі інтернет, наявність монітору, клавіатури та мишки.

					<i>КНТЕУ 121 06-03.МР</i>	Аркуш
						53
Зм.	Аркуш	№ докум	Підпис	Дата		

ДОДАТКИ

Додаток А

Код класу AJAX

```
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using OXG.CRM_System.Models;
using OXG.CRM_System.ViewModels;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace OXG.CRM_System.Controllers
{
    [Route("api/[controller]/[action]")]
    [ApiController]
    [Authorize]
    public class AJAXController : ControllerBase
    {
        private readonly CRMDbContext db;
        public AJAXController(CRMDbContext context)
        {
            db = context;
        }

        /// </summary>
        /// <param name="id">id клиента</param>
        /// <returns></returns>
        [HttpGet]
        public async Task<JsonResult> GetClient(int id)
        {
            var client = await db.Clients.Where(c => c.Id == id).FirstOrDefaultAsync();
            return new JsonResult(client);
        }

        /// <returns></returns>
        [HttpGet]
        public async Task<JsonResult> GetClients()
        {
            var clients = db.Clients;
            var client = await db.Clients.Where(c => c.Name ==
"Temp").FirstOrDefaultAsync();
            clients.Remove(client);
            return new JsonResult(clients);
        }

        /// <param name="name">Email пользователя</param>
        /// <returns></returns>
        [HttpGet]
        public async Task<IActionResult> GetUserPhoto(string name)
        {
            var user = await db.Employeeers.Where(u => u.Email ==
name).FirstOrDefaultAsync();
            var t = user.Photo;
            if (!string.IsNullOrEmpty(t))
            {
                return Content(t);
            }
            else
            {
                return Ok();
            }
        }
    }
}
```

```

    }
}

/// <param name="name">UserName(Email) Пользователя</param>
/// <returns></returns>
[HttpGet]
public async Task<ActionResult> SetViewed(string name)
{
    var user = await db.Employeeers.Where(u => u.Email ==
name).FirstOrDefaultAsync();
    var id = user.Id;
    IQueryable<Notice> notices;
    if (!User.IsInRole("Администратор"))
    {
        notices = db.Notices.Where(n => n.EmployeeId == id && !n.IsViewed);
    }
    else
    {
        notices = db.Notices.Where(n => !n.IsViewed);
    }
    foreach (var item in notices)
    {
        item.IsViewed = true;
    }
    await db.SaveChangesAsync();
    return new OkResult();
}

/// <param name="name">Email текущего пользователя</param>
/// <returns></returns>
[HttpGet]
public async Task<JsonResult> GetNotices(string name)
{
    var user = await db.Employeeers.Where(u => u.Email ==
name).FirstOrDefaultAsync();
    var id = user.Id;
    IQueryable<Notice> notices;
    if (!User.IsInRole("Администратор"))
    {
        notices = db.Notices.Where(n => n.EmployeeId == id);
    }
    else
    {
        notices = db.Notices;
    }
    var model = new List<NoticeVM>();
    foreach (var item in notices)
    {
        var notice = new NoticeVM(item);
        model.Add(notice);
    }
    if (model.Count > 25)
    {
        var mod = model.TakeLast(25);
        mod = mod.Reverse();

        return new JsonResult(mod);
    }
    model.Reverse();
    return new JsonResult(model);
}

/// <param name="name">Email пользователя</param>
/// <returns></returns>

```

```

[HttpGet]
public async Task<ContentResult> GetNoticesNum(string name)
{
    var user = await db.Employeeers.Where(u => u.Email ==
name).FirstOrDefaultAsync();
    var id = user.Id;
    var notices = 0;
    if (User != null)
    {
        if (!User.IsInRole("Администратор"))
        {
            notices = await db.Notices.Where(n => n.EmployeeId == id &&
!n.IsViewed).CountAsync();
        }
        else
        {
            notices = await db.Notices.Where(n => !n.IsViewed).CountAsync();
        }
    }
    return Content(notices.ToString());
}

/// <param name="name">Наименование</param>
/// <returns></returns>
[HttpGet]
public async Task<JsonResult> GetWork(string name)
{
    var work = await db.Works.Where(u => u.Name == name).FirstOrDefaultAsync();
    return new JsonResult(work);
}

/// <param name="name">Имя сотрудника</param>
/// <returns></returns>
[HttpGet]
public async Task<JsonResult> GetEmployee(string name)
{
    var emp = await db.Employeeers.Include(u => u.Missions).Where(u => u.Name ==
name).FirstOrDefaultAsync();
    return new JsonResult(new AdminEmployeeVM(emp));
}

/// <returns></returns>
[HttpGet]
public IActionResult Index()
{
    return Content("Ok");
}
}
}

```

Код класу CRMDbContext

```

using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;
using OXG.CRM_System.Models.Employeers;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace OXG.CRM_System.Models
{
    public class CRMDbContext : IdentityDbContext
    {
        public CRMDbContext(DbContextOptions<CRMDbContext> options) : base(options)
        {
            Database.EnsureCreated();
        }

        public DbSet<Artist> Artists { get; set; }
        public DbSet<Manager> Managers { get; set; }
        public DbSet<Technic> Technics { get; set; }
        public DbSet<Certificate> Certificates { get; set; }
        public DbSet<Contract> Contracts { get; set; }
        public DbSet<Client> Clients { get; set; }
        public DbSet<ClientVK> ClientsVK { get; set; }
        public DbSet<Costume> Costumes { get; set; }
        public DbSet<Equipment> Equipment { get; set; }
        public DbSet<Event> Events { get; set; }
        public DbSet<EventWork> EventWorks { get; set; }
        public DbSet<Mission> Missions { get; set; }
        public DbSet<Notice> Notices { get; set; }
        public DbSet<User> Employeers { get; set; }
        public DbSet<Work> Works { get; set; }
        public DbSet<Requisite> Requisites { get; set; }

        public Manager GetMustFreedomManager()
        {
            var managers = Managers.Include(m => m.Missions);
            var min = 999999;
            var id = "";
            foreach (var item in managers)
            {
                if (item.Missions.Count < min)
                {
                    id = item.Id;
                }
            }
        }
    }
}

```

```
    }  
    }  
    var manager = managers.Where(m => m.Id == id).FirstOrDefault();  
    return manager;  
  }  
}
```

Код NuGet пакеты

```
<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>netcoreapp3.1</TargetFramework>
    <RootNamespace>OXG.CRM_System</RootNamespace>
  </PropertyGroup>

  <ItemGroup>
    <Compile Remove="OXG.CRM-System.Tests\*" />
    <Content Remove="OXG.CRM-System.Tests\*" />
    <EmbeddedResource Remove="OXG.CRM-System.Tests\*" />
    <None Remove="OXG.CRM-System.Tests\*" />
  </ItemGroup>

  <ItemGroup>
    <PackageReference Include="FontAwesome" Version="4.7.0" />
    <PackageReference Include="MailKit" Version="2.5.1" />
    <PackageReference Include="Microsoft.AspNetCore.Identity.EntityFrameworkCore"
Version="3.1.2" />
    <PackageReference Include="Microsoft.AspNetCore.Mvc.NewtonsoftJson" Version="3.1.3"
/>
    <PackageReference Include="Microsoft.EntityFrameworkCore" Version="3.1.2" />
    <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="3.1.2"
/>
    <PackageReference Include="Microsoft.VisualStudio.Web.CodeGeneration.Design"
Version="3.1.2" />
    <PackageReference Include="System.IO" Version="4.3.0" />
    <PackageReference Include="System.IO.Packaging" Version="4.7.0" />
    <PackageReference Include="System.Linq" Version="4.3.0" />
    <PackageReference Include="Telegram.Bot" Version="15.4.0" />
    <PackageReference Include="TemplateEngine.Docx" Version="1.1.5" />
    <PackageReference Include="VkNet" Version="1.50.0" />
  </ItemGroup>

  <ItemGroup>
    <Folder Include="wwwroot\AccountPhotos\" />
  </ItemGroup>

  <ProjectExtensions><VisualStudio><UserProperties appsettings_1json__JsonSchema=""
/></VisualStudio></ProjectExtensions>

</Project>
```

Код рабочего поля администратора

```

@model OXG.CRM_System.ViewModels.AdminIndexVM
@{
    ViewData["Title"] = "Index";
}
<script src="~/js/Chart.min.js"></script>
<h1>Admin</h1>
<div style="margin:1%">
    <div class="row">
        <div class="col-lg-6">
            <a asp-action="Employeeers" class="btn btn-outline-info btn-block">Сотрудники</a>
            <a asp-action="Clients" class="btn btn-outline-info btn-block">Клиенты</a>
        </div>
        <div class="col-lg-6">
            <a asp-action="Works" class="btn btn-outline-info btn-block">Услуги</a>
            <a asp-action="Setting" class="btn btn-outline-info btn-block">Конфигурация</a>
        </div>
    </div>
    <br />
    <div class="row">
        <div class="col-lg-12">
            <div class="card border-primary">
                <div class="container-fluid">
                    <canvas id="eventSumChart" @*width="420" height="220"*@></canvas>
                    @*График доходов 2 месяцев, линейный, 30 дней*@
                </div>
            </div>
        </div>
    </div>
    <br />
    <div class="row">
        <div class="col-lg-6">
            <div class="card border-primary">
                <div class="container-fluid">
                    <canvas id="requestCloseTypesChart"></canvas> @*График причин отклонения заявок, столбы, всё время*@
                </div>
            </div>
        </div>
        <div class="col-lg-6">
            <div class="card border-primary">
                <div class="container-fluid">
                    <canvas id="numWorkPopularChart"></canvas> @*График популярности различных услуг, столбы, 30 дней или всё время*@
                </div>
            </div>
        </div>
    </div>
    <br />
    <div class="row">
        <div class="col-lg-6">
            <div class="card border-primary">
                <h3 align="center">Типы мероприятий</h3>
                <div class="container-fluid">
                    <canvas id="eventTypesChart"></canvas> @*График типов мероприятий, круговой, всё время*@
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
    </div>
    <div class="col-lg-6">
        <div class="card border-primary">
            <h3 align="center">Одобренные заявки</h3>
            <div class="container-fluid">
                <canvas id="ManagerGoodRequestChart" ></canvas> @*График одобренных
заявок менеджеров, круговой, 30 дней или всё время*@
            </div>
        </div>
    </div>
</div>
</div>

```

```

<script>
    var labls = @Json.Serialize(Model.Last30Days);
    var datas = @Json.Serialize(Model.EventsSum);
    var rejectCauses = @Json.Serialize(Model.RejectCauses2);
    var rejectNums = @Json.Serialize(Model.RejectNum);
    var worksName = @Json.Serialize(Model.WorksName);
    var worksCount = @Json.Serialize(Model.WorksNum);
    var typeName = @Json.Serialize(Model.TypesName);
    var typesCount = @Json.Serialize(Model.TypesCount);
    var managersName = @Json.Serialize(Model.ManagerName);
    var requestCount = @Json.Serialize(Model.ManagerRequestCount);

```

```

    var ctx = document.getElementById("eventSumChart");
    var ct2 = document.getElementById("requestCloseTypesChart");
    var ct3 = document.getElementById("numWorkPopularChart");
    var ct4 = document.getElementById("eventTypesChart");
    var ct5 = document.getElementById("ManagerGoodRequestChart");

```

```

        var myChart = new Chart(ctx, {
            type: 'line',
            data: {
                labels:labls,
                datasets: [{
                    label: 'Сумма заказов (UAH)',
                    data: datas,
                    backgroundColor: [
                        'rgba(0, 157, 255, 0.2)'
                    ],
                    borderColor: [
                        'rgba(54, 162, 235, 1)'
                    ],
                    borderWidth: 1
                }]
            },
            options: {
                scales: {
                    yAxes: [{
                        ticks: {
                            beginAtZero: true
                        }
                    }]
                }
            }
        });

```

```

        var barChart = new Chart(ct2, {
            type: 'bar',

```

```

        data: {
            labels: rejectCauses,
            datasets: [{
                label: 'Причины отклонения заявок',
                data: rejectNums,
                backgroundColor: [
                    'rgba(0, 99, 132, 0.6)',
                    'rgba(0, 99, 132, 0.6)',
                    'rgba(30, 99, 132, 0.6)',
                    'rgba(30, 99, 132, 0.6)',
                    'rgba(60, 99, 132, 0.6)',
                    'rgba(60, 99, 132, 0.6)',
                    'rgba(90, 99, 132, 0.6)',
                    'rgba(90, 99, 132, 0.6)',
                    'rgba(120, 99, 132, 0.6)',
                    'rgba(120, 99, 132, 0.6)',
                    'rgba(150, 99, 132, 0.6)',
                    'rgba(150, 99, 132, 0.6)',
                    'rgba(180, 99, 132, 0.6)',
                    'rgba(180, 99, 132, 0.6)',
                    'rgba(210, 99, 132, 0.6)',
                    'rgba(210, 99, 132, 0.6)',
                    'rgba(240, 99, 132, 0.6)',
                    'rgba(240, 99, 132, 0.6)'
                ],
                borderColor: [
                    'rgba(0, 99, 132, 1)',
                    'rgba(0, 99, 132, 1)',
                    'rgba(30, 99, 132, 1)',
                    'rgba(30, 99, 132, 1)',
                    'rgba(60, 99, 132, 1)',
                    'rgba(60, 99, 132, 1)',
                    'rgba(90, 99, 132, 1)',
                    'rgba(90, 99, 132, 1)',
                    'rgba(120, 99, 132, 1)',
                    'rgba(120, 99, 132, 1)',
                    'rgba(150, 99, 132, 1)',
                    'rgba(150, 99, 132, 1)',
                    'rgba(180, 99, 132, 1)',
                    'rgba(180, 99, 132, 1)',
                    'rgba(210, 99, 132, 1)',
                    'rgba(210, 99, 132, 1)',
                    'rgba(240, 99, 132, 1)',
                    'rgba(240, 99, 132, 1)'
                ],
                borderWidth: 2,
                hoverBorderWidth: 0
            }]
        }
    });

```

```

    var barChart3 = new Chart(ct3, {
        type: 'bar',
        data: {
            labels: worksName,
            datasets: [{
                label: 'Популярность услуг',
                data: worksCount,
                backgroundColor: [
                    'rgba(0, 99, 132, 0.6)',
                    'rgba(0, 99, 132, 0.6)',
                    'rgba(30, 99, 132, 0.6)',
                    'rgba(30, 99, 132, 0.6)',
                    'rgba(60, 99, 132, 0.6)',

```

```

        'rgba(60, 99, 132, 0.6)',
        'rgba(90, 99, 132, 0.6)',
        'rgba(90, 99, 132, 0.6)',
        'rgba(120, 99, 132, 0.6)',
        'rgba(120, 99, 132, 0.6)',
        'rgba(150, 99, 132, 0.6)',
        'rgba(150, 99, 132, 0.6)',
        'rgba(180, 99, 132, 0.6)',
        'rgba(180, 99, 132, 0.6)',
        'rgba(210, 99, 132, 0.6)',
        'rgba(210, 99, 132, 0.6)',
        'rgba(240, 99, 132, 0.6)',
        'rgba(240, 99, 132, 0.6)'
    ],
    borderColor: [
        'rgba(0, 99, 132, 1)',
        'rgba(0, 99, 132, 1)',
        'rgba(30, 99, 132, 1)',
        'rgba(30, 99, 132, 1)',
        'rgba(60, 99, 132, 1)',
        'rgba(60, 99, 132, 1)',
        'rgba(90, 99, 132, 1)',
        'rgba(90, 99, 132, 1)',
        'rgba(120, 99, 132, 1)',
        'rgba(120, 99, 132, 1)',
        'rgba(150, 99, 132, 1)',
        'rgba(150, 99, 132, 1)',
        'rgba(180, 99, 132, 1)',
        'rgba(180, 99, 132, 1)',
        'rgba(210, 99, 132, 1)',
        'rgba(210, 99, 132, 1)',
        'rgba(240, 99, 132, 1)',
        'rgba(240, 99, 132, 1)'
    ],
    borderWidth: 2,
    hoverBorderWidth: 0
    });

    var roundChart = new Chart(ct4, {
        type: 'pie',
        data: {
            labels: typesName,
            datasets: [{
                label: 'Типы мероприятий',
                data: typesCount,
                backgroundColor: [
                    "#FF6384",
                    "#63FF84",
                    "#84FF63",
                    "#8463FF",
                    "#6384FF",
                    "#FF6384",
                    "#63FF84",
                    "#84FF63",
                    "#8463FF",
                    "#6384FF"
                ]
            }],
        },
        options: {cutoutPercentage: 40}
    });

```

```

var roundChart2 = new Chart(ct5, {
  type: 'pie',
  data: {
    labels: managersName,
    datasets: [{
      label: 'Одобрённые заявки',
      data: requestCount,
      backgroundColor: [
        "#84FF63",
        "#6384FF",
        "#FF6384",
        "#84FF63",
        "#8463FF",
        "#6384FF",
        "#63FF84",
        "#FF6384",
        "#63FF84",
        "#8463FF"
      ],
    }],
  },
  options: {cutoutPercentage: 40}
});
</script> @*Скрипт графиков*@

<script>
var container = document.getElementById('LayoutContainer')
container.classList.remove('container')
</script>

```

Код рабочего поля менеджера

```

@model OXG.CRM_System.Models.Employeeers.Manager
@{
    ViewData["Title"] = "Задачи";
}

<style>
    #line_block {
        float: left;
        margin: 0 15px 15px 0;
        padding: 10px;
    }
</style>

<div class="row" style="margin-left:0px; margin-right:0px;">
    <div class="col-lg-6">
        <a asp-action="New" asp-controller="Events" type="button" class="btn btn-success
btn-lg btn-block">Создать мероприятие</a>
        <div class="card border-success mb-3 ">
            <div class="card-header">События</div>
            <div class="card-body table-responsive" style="height:400px">
                <table class="table table-hover">
                    <thead>
                        <tr>
                            <th scope="col">Наименование</th>
                            <th scope="col">Дата проведения</th>
                            <th scope="col">Статус</th>
                            <th scope="col">Клиент</th>
                        </tr>
                    </thead>
                    <tbody>
                        @if (Model.Events != null && Model.Clients != null)
                        {
                            @foreach (var item in Model.Events)
                            {
                                <tr>
                                    <th scope="row"><a asp-action="View" asp-
controller="Events" asp-route-id="@item.Id">@item.Name</a></th>
                                    <td>@item.DeadLine.ToLongDateString()</td>
                                    <td>@item.Status</td>
                                    <td><a asp-action="Client" asp-controller="Sight"
asp-route-id="@item.Client.Id">@item.Client.Name</a></td>
                                </tr>
                            }
                        }
                    </tbody>
                </table>
            </div>
        </div>
        <div class="col-lg-6">
            <button type="button" class="btn btn-info btn-lg btn-block" data-toggle="modal"
data-target="#ModalCenter">Создать заявку</button>
            <div class="card border-info mb-3 ">
                <div class="card-header">Заявки</div>
                <div class="card-body table-responsive" style="height:400px">
                    <table class="table table-hover">
                        <thead>
                            <tr>

```

```

        <th scope="col"></th>
        <th scope="col">Описание</th>
        <th scope="col">Дедлайн</th>
        <th scope="col">Заккрыть</th>
    </tr>
</thead>
<tbody>
    @if (Model.Missions != null)
    {
        @foreach (var item in Model.Missions)
        {
            @if (item.Status != "Заккрыто" && (item.MissionType ==
"Заявка" || item.MissionType == "Автоматическая заявка"))
            {
                <tr>
                    <td></td>
                    <td>@Html.Raw(item.MissionText)</td>
                    <td>@item.DeadLine.ToShortDateString(),
@item.DeadLine.ToShortTimeString()</td>
                    <td><a asp-action="Close" asp-
controller="Mission" asp-route-id="@item.Id">Заккрыть</a></td>
                </tr>
            }
        }
    }
</tbody>
</table>
</div>
</div>
</div>
</div>

<div class="row" style="margin-left: 1%; margin-right:0px;">
    @foreach (var item in Model.Missions)
    {
        if (item.MissionType != "Заявка" && item.MissionType != "Автоматическая заявка"
&& item.Status != "Заккрыто")
        {
            <div class="card border-info mb-3" style="max-width: 22rem;"
title="@item.Status" id="line_block">
                <div class="card-header"> </div>
                <div class="card-body">
                    <h4 class="card-title">@item.MissionType</h4>
                    <p class="card-text">@item.MissionText, Времени на выполнение
осталось: <strong>@item.LeftTime.Hours часов @item.LeftTime.Minutes минут</strong> </p>
                    <div><a class="btn btn-outline-secondary" asp-action="Close" asp-
controller="Mission" asp-route-id="@item.Id">Выполнить</a></div>
                </div>
            </div>
        }
    }
</div>

<script>
    var container = document.getElementById('LayoutContainer')
    container.classList.remove('container')
</script>

```

```

@*<script>
    onload = function () { setTimeout('location.reload (true)', 60000) }
</script>*@

<div class="modal fade" id="ModalCenter" tabindex="-1" role="dialog" aria-
labelledby="exampleModalCenterTitle" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <form asp-action="NewRequest" asp-controller="Mission" method="post">
            <div class="modal-content" style="width:550px">
                <div class="modal-header">
                    <h5 class="modal-title" id="exampleModalLongTitle">Новая заявка</h5>
                    <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
                        <span aria-hidden="true">&times;</span>
                    </button>
                </div>
                <div class="modal-body">
                    <div class="form-group">
                        <label for="exampleInputEmail1">Текст заявки</label>
                        <textarea name="RequestText" class="form-control"
id="exampleInputEmail1" required></textarea>
                        <small id="emailHelp" class="form-text text-muted">Введите текст
заявки.</small>
                    </div>
                </div>
                <div class="modal-footer">
                    <input type="submit" class="btn btn-success" value="Создать" />
                </div>
            </div>
        </form>
    </div>
</div>

<script src="~/js/site.js" asp-append-version="true"></script>

```

Код робочого поля техніка

```

@model OXG.CRM_System.Models.Employeers.Technic

@{
    ViewData["Title"] = "Index";
}

<h1>Техник</h1>

<table class="table table-bordered table-hover table-responsive-lg">
    <thead>
        <tr>
            <th scope="col">Тип</th>
            <th scope="col">Задача</th>
            <th scope="col">Клиент</th>
            <th scope="col">Мероприятие</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model.Missions)
        {
            @if (item.Status != "Закрито")
            {
                <tr class="table-light">
                    <th><a asp-action="ConfirmEvent" asp-route-id="@item.Event.Id"></a></th>
                    <td>@item.MissionText</td>
                    <td><a asp-controller="Sight" asp-action="Client" asp-route-
id="@item.Event.Client.Id" id="client">@item.Event.Client.Name</a></td>
                    <td><a asp-controller="Events" asp-action="View" asp-route-
id="@item.Event.Id">@item.Event.Name</a></td>
                </tr>
            }
        }
    </tbody>
</table>

```

Код робочого поля різноробочого

```

@model OXG.CRM_System.Models.Employeeers.Artist

@{
    ViewData["Title"] = "Index";
}

<h1>Разноробочий</h1>

<div class="row">
    <div class="col-lg-6">
        <div class="card border-info mb-3 ">
            <div class="card-header">Ожидают подтверждения</div>
            <div class="card-body table-responsive" style="height:400px">
                <table class="table table-hover">
                    <thead>
                        <tr>
                            <th scope="col">Дата</th>
                            <th scope="col">Время начала</th>
                            <th scope="col">Мероприятие</th>
                        </tr>
                    </thead>
                    <tbody>
                        @foreach (var item in Model.Missions)
                        {
                            if (item.Event.Status.Contains("Ожидает согласования"))
                            {
                                <tr class="table-light">
                                    <td>@item.DeadLine.ToShortDateString()</td>
                                    <td>@item.Event.DeadLine.ToShortTimeString()</td>
                                    <td><a asp-controller="Events" asp-action="View" asp-
route-id="@item.Event.Id">@item.Event.Name</a></td>
                                </tr>
                            }
                        }
                    </tbody>
                </table>
            </div>
        </div>
    <div class="col-lg-6">
        <div class="card border-success mb-3 ">
            <div class="card-header">Подтверждённые</div>
            <div class="card-body table-responsive" style="height:400px">
                <table class="table table-hover">
                    <thead>
                        <tr>
                            <th scope="col">Дата</th>
                            <th scope="col">Время начала</th>
                            <th scope="col">Мероприятие</th>
                        </tr>
                    </thead>
                    <tbody>
                        @foreach (var item in Model.Missions)
                        {
                            if (item.Event.Status.Contains("Согласовано"))
                            {
                                <tr class="table-light">
                                    <td>@item.DeadLine.ToShortDateString()</td>
                                    <td>@item.Event.DeadLine.ToShortTimeString()</td>
                                </tr>
                            }
                        }
                    </tbody>
                </table>
            </div>
        </div>
    </div>
</div>

```

```

        <td><a asp-controller="Events" asp-action="View" asp-
route-id="@item.Event.Id">@item.Event.Name</a></td>
    </tr>
    }
}
</tbody>
</table>
</div>
</div>
</div>
</div>

<script>
    var container = document.getElementById('LayoutContainer')
    container.classList.remove('container')
</script>

```

Код вікна авторизації

```

@model OXG.CRM_System.ViewModels.LoginViewModel
@{
    ViewData["Title"] = "Login";
    Layout = null;
}

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.min.css" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css" />

<style>
/* Demo Background */
body {
    background: url('../Images/2008.jpg');
}

/* Form Style */
.form-horizontal{
    background: #fff;
    padding-bottom: 40px;
    border-radius: 15px;
    text-align: center;
}

    .form-horizontal .heading {
        display: block;
        font-size: 35px;
        font-weight: 700;
        padding: 35px 0;
        border-bottom: 1px solid #f0f0f0;
        margin-bottom: 30px;
    }

    .form-horizontal .form-group {
        padding: 0 40px;
        margin: 0 0 25px 0;
        position: relative;
    }

    .form-horizontal .form-control{
        background: #f0f0f0;
        border: none;
        border-radius: 20px;
        box-shadow: none;
        padding: 0 20px 0 45px;
        height: 40px;
        transition: all 0.3s ease 0s;
    }

    .form-horizontal .form-control:focus{
        background: #e0e0e0;
        box-shadow: none;
        outline: 0 none;
    }

    .form-horizontal .form-group i{
        position: absolute;
        top: 12px;
        left: 60px;
        font-size: 17px;
        color: #c8c8c8;
        transition : all 0.5s ease 0s;
    }

```

```

.form-horizontal .form-control:focus + i{
  color: #00b4ef;
}
.form-horizontal .fa-question-circle{
  display: inline-block;
  position: absolute;
  top: 12px;
  right: 60px;
  font-size: 20px;
  color: #808080;
  transition: all 0.5s ease 0s;
}
.form-horizontal .fa-question-circle:hover{
  color: #000;
}
.form-horizontal .main-checkbox{
  float: left;
  width: 20px;
  height: 20px;
  background: #11a3fc;
  border-radius: 50%;
  position: relative;
  margin: 5px 0 0 5px;
  border: 1px solid #11a3fc;
}
.form-horizontal .main-checkbox label{
  width: 20px;
  height: 20px;
  position: absolute;
  top: 0;
  left: 0;
  cursor: pointer;
}
.form-horizontal .main-checkbox label:after{
  content: "";
  width: 10px;
  height: 5px;
  position: absolute;
  top: 5px;
  left: 4px;
  border: 3px solid #fff;
  border-top: none;
  border-right: none;
  background: transparent;
  opacity: 0;
  -webkit-transform: rotate(-45deg);
  transform: rotate(-45deg);
}
.form-horizontal .main-checkbox input[type=checkbox]{
  visibility: hidden;
}
.form-horizontal .main-checkbox input[type=checkbox]:checked + label:after{
  opacity: 1;
}
.form-horizontal .text{
  float: left;
  margin-left: 7px;
  line-height: 20px;
  padding-top: 5px;
  text-transform: capitalize;
}
.form-horizontal .btn{
  float: right;
  font-size: 14px;
}

```

```

color: #fff;
background: #00b4ef;
border-radius: 30px;
padding: 10px 25px;
border: none;
text-transform: capitalize;
transition: all 0.5s ease 0s;
}
@media only screen and (max-width: 479px){
.form-horizontal .form-group{
padding: 0 25px;
}
.form-horizontal .form-group i{
left: 45px;
}
.form-horizontal .btn{
padding: 10px 20px;
}
}
</style>

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

    <div class="col-md-offset-3 col-md-6">
      <form class="form-horizontal" asp-action="Login" asp-controller="Account">
        <span class="heading">АВТОРИЗАЦИЯ</span>
        <div class="form-group">
          <input type="email" asp-for="Email" class="form-control"
id="inputEmail" placeholder="E-mail">
          <i class="fa fa-user"></i>
        </div>
        <div class="form-group help">
          <input type="password" asp-for="Password" class="form-control"
id="inputPassword" placeholder="Пароль">
          <i class="fa fa-lock"></i>
          <a asp-action="Register" class="fa fa-question-circle"></a>
        </div>
        <div class="form-group">
          <div class="main-checkbox">
            <input type="checkbox" asp-for="RememberMe" id="checkbox1"
name="check" />
            <label for="checkbox1"></label>
          </div>
          <span class="text">Запомнить</span>
          <button type="submit" class="btn btn-default">ВХОД</button>
        </div>
        <div asp-validation-summary="ModelOnly"></div>
      </form>
    </div>

  </div><!-- /.row -->
</div><!-- /.container -->

```

Код вікна реєстрації

```

@model OXG.CRM_System.ViewModels.RegisterViewModel
@{
    ViewData["Title"] = "Регистрация";
    Layout = null;
}

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.min.css" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css" />

<style>
    /* Demo Background 'https://source.unsplash.com/collection/9693967/1600x900'*/
    body {
        background: url('../Images/2008.jpg');
    }

    /* Form Style */
    .form-horizontal {
        background: #fff;
        padding-bottom: 40px;
        border-radius: 15px;
        text-align: center;
    }

    .form-horizontal .heading {
        display: block;
        font-size: 35px;
        font-weight: 700;
        padding: 35px 0;
        border-bottom: 1px solid #f0f0f0;
        margin-bottom: 30px;
    }

    .form-horizontal .form-group {
        padding: 0 40px;
        margin: 0 0 25px 0;
        position: relative;
    }

    .form-horizontal .form-control {
        background: #f0f0f0;
        border: none;
        border-radius: 20px;
        box-shadow: none;
        padding: 0 20px 0 45px;
        height: 40px;
        transition: all 0.3s ease 0s;
    }

    .form-horizontal .form-control:focus {
        background: #e0e0e0;
        box-shadow: none;
        outline: 0 none;
    }

```

```

.form-horizontal .form-group i {
    position: absolute;
    top: 12px;
    left: 60px;
    font-size: 17px;
    color: #c8c8c8;
    transition: all 0.5s ease 0s;
}

.form-horizontal .form-control:focus + i {
    color: #00b4ef;
}

.form-horizontal .fa-question-circle {
    display: inline-block;
    position: absolute;
    top: 12px;
    right: 60px;
    font-size: 20px;
    color: #808080;
    transition: all 0.5s ease 0s;
}

.form-horizontal .fa-question-circle:hover {
    color: #000;
}

.form-horizontal .main-checkbox {
    float: left;
    width: 20px;
    height: 20px;
    background: #11a3fc;
    border-radius: 50%;
    position: relative;
    margin: 5px 0 0 5px;
    border: 1px solid #11a3fc;
}

.form-horizontal .main-checkbox label {
    width: 20px;
    height: 20px;
    position: absolute;
    top: 0;
    left: 0;
    cursor: pointer;
}

.form-horizontal .main-checkbox label:after {
    content: "";
    width: 10px;
    height: 5px;
    position: absolute;
    top: 5px;
    left: 4px;
    border: 3px solid #fff;
    border-top: none;
    border-right: none;
    background: transparent;
    opacity: 0;
    -webkit-transform: rotate(-45deg);
    transform: rotate(-45deg);
}

.form-horizontal .main-checkbox input[type=checkbox] {

```

```

        visibility: hidden;
    }

    .form-horizontal .main-checkbox input[type=checkbox]:checked +
label:after {
        opacity: 1;
    }

    .form-horizontal .text {
        float: left;
        margin-left: 7px;
        line-height: 20px;
        padding-top: 5px;
        text-transform: capitalize;
    }

    .form-horizontal .btn {
        float: right;
        font-size: 14px;
        color: #fff;
        background: #00b4ef;
        border-radius: 30px;
        padding: 10px 25px;
        border: none;
        text-transform: capitalize;
        transition: all 0.5s ease 0s;
    }

    @@media only screen and (max-width: 479px) {
        .form-horizontal .form-group {
            padding: 0 25px;
        }

        .form-horizontal .form-group i {
            left: 45px;
        }

        .form-horizontal .btn {
            padding: 10px 20px;
        }
    }
</style>

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

        <div class="col-md-offset-3 col-md-6">
            <form class="form-horizontal" asp-action="Register" asp-controller="Account">
                <span class="heading">РЕГИСТРАЦИЯ</span>
                <div class="form-group help">
                    <input asp-for="Name" type="text" class="form-control" id="FI0"
placeholder="ФИО">
                    <i class="fa-address-card"></i>
                    <span asp-validation-for="Name"></span>
                </div>
                <div class="form-group">
                    <input asp-for="Email" type="email" class="form-control"
id="inputEmail" placeholder="E-mail">
                    <i class="fa-user"></i>
                </div>
                <div class="form-group help">
                    <input asp-for="Password" type="password" class="form-control"
id="inputPassword" placeholder="Пароль">
                    <i class="fa-lock"></i>

```

```

        <span asp-validation-for="Password"></span>
    </div>
    <div class="form-group help">
        <input asp-for="PasswordConfirm" type="password" class="form-control"
id="ConfirmPassword" placeholder="Подтвердите пароль">
        <i class="fa-lock"></i>
        <span asp-validation-for="PasswordConfirm"></span>
    </div>
    <div class="form-group help">
        <input asp-for="PhoneNumber" type="text" class="form-control"
id="PhoneNumber" placeholder="Номер телефона">
        <i class="fa-phone"></i>
        <span asp-validation-for="PhoneNumber"></span>
    </div>

    <div asp-validation-summary="ModelOnly"></div>

    <label for="Radios">Зарегистрироваться как:</label>
    <div id="Radios" class="row">

        @*<div class="custom-control custom-radio">*@
            <input type="radio" id="customRadio1" asp-for="UserType"
value="Менеджер" class="custom-control-input" checked="">
            <label class="custom-control-label" for="customRadio1">Менеджер
|</label>
        @*</div>*@

        @*<div class="custom-control custom-radio">*@
            <input type="radio" id="customRadio2" asp-for="UserType"
value="Техник" class="custom-control-input">
            <label class="custom-control-label" for="customRadio2">Техник
|</label>
        @*</div>*@

        @*<div class="custom-control custom-radio">*@
            <input type="radio" id="customRadio3" asp-for="UserType"
value="Разнорабочий" class="custom-control-input">
            <label class="custom-control-label"
for="customRadio3">Разнорабочий |</label>
        @*</div>*@
    </div>

    <div class="form-group">
        <input type="submit" class="btn btn-default"
value="Зарегистрироваться" />
    </div>

</form>
</div>
</div><!-- /.row -->
</div><!-- /.container -->

```