Міністерство науки і освіти України

Житомирський державний технологічний університет

Кафедра програмного забезпечення систем

КУРСОВИЙ ПРОЕКТ

з дисципліни «Бази даних»

на тему: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студента (ки) \_\_\_\_\_ курсу \_\_\_\_\_\_ групи

спеціальності 6.05010301 «Програмна інженерія»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_Оцінка: ECTS \_\_\_\_\_

Члени комісії: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ І.І. Сугоняк

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.O. Данильченко

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Н. Ковальчук

м. Житомир – 2017 рік

**РЕЦЕНЗІЯ НА КУРСОВУ РОБОТУ З ДИСЦИПЛІНИ   
„БАЗИ ДАНИХ”**

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

2

Розроб.

Ксенич Є.Г.

Перевір.

Керівник

Н. Контр.

Затверд.

Пояснювальна записка

Літ.

Аркушів

ФІКТ, гр. ПІ-52

студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ групи\_\_\_\_\_\_\_\_\_\_\_\_\_

на тему\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЗМІСТ РЕЦЕНЗІЇ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| ***№*** | ***Критерії оцінювання*** | ***Кількість балів*** |
| 1. | Повнота дослідження і викладу матеріалу (по кожному розділу): |  |
|  | ВСТУП |  |
|  | РОЗДІЛ 1 |  |
|  | РОЗДІЛ 2 |  |
|  | РОЗДІЛ 3 |  |
|  | РОЗДІЛ 4 |  |
|  | ВИСНОВКИ І ПРОПОЗИЦІЇ |  |
| 2. | Врахування вимог до оформлення |  |
| 3. | Відповідність ТЗ та складність реалізованого програмного продукту |  |

**Висновок викладача**: В цілому робота відповідає/ не відповідає вимогам вищої школи і може/ не може бути допущеною до захисту.

Науковий керівник: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ „\_\_\_\_” \_\_\_\_\_\_\_\_ 20\_\_ р

**ПІДСУМКИ ЗАХИСТУ КУРСОВОЇ РОБОТИ**

|  |  |  |
| --- | --- | --- |
| ***№*** | ***Критерії оцінювання*** | ***Кількість балів*** |
| 1. | Повнота та рівень підготовки доповіді |  |
| 2. | Обгрунтування технічного рішення та якість реалізації |  |
| 3. | Відповіді на запитання членів комісії |  |

**Рейтингова підсумкова оцінка курсової роботи**

|  |  |  |
| --- | --- | --- |
| **Кількість балів** | **Національна шкала** | **Шкала ECTS** |
|  |  |  |

Члени комісії:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата захисту курсової роботи „\_\_\_\_” \_\_\_\_\_\_\_\_\_ 20\_\_\_ р

# 

# ЗМІСТ

ВСТУП………………………………………………………………………….7

РОЗДІЛ 1 ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ….…………..8

1.1 Аналіз інформаційних потреб та визначення предметної області дослідження……………………………….…………………………………….....8

1.2 Обґрунтування вибору засобів реалізації …….………………..................9

РОЗДІЛ 2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ…..……………………………...12

2.1 Аналіз інформаційних процесів …………….…………………………...12

2.2 Проектування структури бази даних ……………………...………..…...14

2.3 Розробка математичної моделі та алгоритмів обробки даних в системи…………………………………………………………………………...18

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОБРОБКИ ДАНИХ...….……….….19

3.1 Проектування інтерфейсу обробки даних ………………………….…..19

3.2 Реалізація операцій обробки даних в БД ………......…………………....20

3.3 Організація звітності системи…………………………………………....23

РОЗДІЛ 4 АДМІНІСТРУВАННЯ БАЗ ДАНИХ…………………………....25

4.1. Розробка заходів захисту інформації в БД………………………...…....25

4.2. Налаштування параметрів роботи SQL-сервера……………………….28

ВИСНОВКИ………………………………………………………………….......32

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ…………………………………..……33

ДОДАТКИ…………………………………………………………………..…....34

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ORM - Object-relational mapping - технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних».

EF - Entity Framework – технологія доступу до даних, що базується на принципах ORM.

LINQ - Language Integrated Query – мова запитів, що нагадує SQL, та реалізована у .Net.

ВСТУП

Метою курсового проекту – є дослідження особливостей проектування та реалізації бази даних виконаних робіт з ландшафтного дизайну.

Актуальність дослідження полягає у тому, щоб визначити найоптимальніші шляхи вирішення поставленої проблеми. І загальну необхідність у вирішенні даної проблеми.

Тема на курсовий проект – це розробка бази даних для виконаних робіт з ландшафтного дизайну.

До інформаційних потреб даної предметної області можна віднести: можливості взаємодії з інформацією у базі даних.

Предметом дослідження є можливості застосування концепції БД та СУБД для забезпечення інформаційних потреб предметної області.

Об’єкт дослідження - методи, засоби проектування баз даних для заданої предметної області, та можливість застосування концепції бази даних для повного забезпечення потреб обраної предметної області.

РОЗДІЛ 1 ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

**1.1 Аналіз інформаційних потреб та визначення предметної області дослідження**

Основною метою аналізу інформаційних потреб, та визначення предметної області – є збір інформації, що допоможе створити програмний продукт, що задовольнить усі поставлені перед ним задачі, буде швидко оперувати даними, та буде легко розширюваним.

Для правильного, та повного визначення предметної області потрібно зібрати велику кількість різнопланової інформації, що у повній мірі характеризує усі аспекти даної предметної області, та дає розуміння усім інформаційних процесів, які використовуються у рамках даної предметно області.

Для забезпечення повноцінного функціонування програмного продукту, адміністратор повинен мати доступ до взаємодії з такою інформацією:

* Дата початку замовлення.
* Дата кінця замовлення.
* Назва обробленої землі.
* Розмір обробленої землі.
* Адрес обробленої землі.
* Тип виконуваної роботи для певного замовлення.
* Кількість виконуваної роботи для певного замовлення.
* Ціна виконуваної роботи для певного замовлення.

**1.2 Обґрунтування вибору засобів реалізації**

Аналіз та вибір засобів реалізації проводиться з урахуванням того, що доступ до даних повинен бути ефективним і число користувачів, що будуть отримувати доступ до даних у один час невелике.

Також потрібно зазначити, що сервер повинен забезпечувати високий рівень надійності.

Безпосередньо сервери будуть працювати на ОС Windows.

У таблицях нижче наведено порівняльні характеристики СУБД.

Таблиця 1.1

Функціональне порівняння СУБД

|  |  |  |
| --- | --- | --- |
| Характеристика | Microsoft SQL Server | Oracle |
| Адміністративне керування | Добре | Відмінно |
| Графічні інструменти | Відмінно | Добре |
| Простота обслуговування | Відмінно | Відмінно |
| Механизм данних | Добре | Відмінно |
| Робота с декількома ЦП | Задовільно | Відмінно |
| Функції з’єднання і вибір індексів | Відмінно | Відмінно |
| Одночасний доступ декількох користувачів | Добре | Відмінно |
| Обробка даних мультимедіа | Плохо | Відмінно |
| Подключення к Web | Задовільно | Відмінно |
| Повнотекстовий пошук | Добре | Відмінно |
| Функціональна сумісність | Добре | Добре |
| Інтеграція з іншими СУБД | Добре | Добре |
| Єдина реєстрація | Добре | Добре |
| Робота під керівництвом ОС | Задовільно | Добре |
| Можливості програмування | Задовільно | Відмінно |
| Процедури, що зберігаються та тригери | Добре | Відмінно |
| Вбудована мова програмування | Задовільно | Відмінно |
| Побудова БД | Добре | Відмінно |
| Мова SQL | Відмінно | Відмінно |
| Підтримка об’єктно-орієнтованої парадигми | Задовільно | Відмінно |
| Робота у режимі віддаленого доступу | Відмінно | Відмінно |
| Тиражування | Відмінно | Відмінно |
| Розподілена обробка транзакцій | Відмінно | Відмінно |
| Дистанційне адміністрування | Добре | Відмінно |
| Організація сховищ даних і підготовка звітів | Відмінно | Відмінно |
| Засоби завантаження | Відмінно | Відмінно |
| Засоби аналізу | Відмінно | Відмінно |

Таблиця 1.2

Порівняння за вимогами ТЗ

|  |  |  |
| --- | --- | --- |
| Необходимые требования | MS SQL Server | Oracle |
| Локалізація інтерфейсу користувача, можливість побудови і сортування полів БД, що містять символи кирилиці | + | + |
| Підтримка структури відносних даних | + | + |
| Підтримка технології клієнт/сервер | + | + |
| Підтримка багатопроцесорної архітектури | + | + |
| Підтримка кластерної архітектури | + | + |
| Наявність засобів для створення індексів і кластерів для підвищення ефективності використання даних | + | + |
| Відновлення баз даних із використанням журнала трансакцій | + | + |
| Механізм блокування трансакцій під час запису або на рівні сторінок | + | + |
| підтримка ANSI SQL | + | + |
| Підтримка стандарту SQL-3 (новое название – SQL99) | + - | + |
| підтримка ODBC | + | + |
| Контроль цілісності БД | + | + |
| Підтримка утиліт резервування БД | + | + |
| Імпорт/експорт таблиць БД | + | + |
| Сумісність з ОС модулів користувача та сервера | + | + |
| Підтримка визначених мережевих протоколів | + | + |
| Наявність графічного інтерфейсу для адміністраторів БД | + | + |
| Контроль доступу до даних. Аутентифікація засобами СУБД | + | + |
| Централізоване керування користувачами | + | + |
| Наявність оптимізатора запитів для оптимізації планів виконання | + | + |
| Підтримка великих двійкових об’єктів (BLOB) | + | + |
| Підтримка OLAP технологій, спеціалізованих засобів OLAP-аналізу | + | + |
| Підтримка протоколів VI SAN (Virtual Interface System Area Network) | + | + |
| Відлагоджений механізм реплікації даних | + | + |
| Підтримка служби єдиного каталогу | + | + |

Таблиця 1.3

Експертна оцінка багатокористувацьких СУБД

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СУБД | Продук-тивність | Конкурентний доступ | Кількість користувачів | Великі БД | Готовність |
| Microsoft SQL Server | 6 | 7 | 5 | 5 | 7 |
| Oracle | 6 | 9 | 8 | 8 | 8 |

Керуючись даними, які описані у таблицях вище, для виконання поставленої задачі я обрав Microsoft SQL Server, що є оптимальним вибором для сучасних додатків, адже дозволяє розробляти бази даних для додатків будь-якого розміру, легко масштабується, та у повній мірі задовольняє вимоги до безпеки і дозволяє зберегти важливі дані, що цікавить нас не в останню чергу.

Дане рішення у повній мірі дозволяє реалізувати усі поставлені вимоги до готового програмного продукту.

2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ

**2.1 Аналіз інформаційних процесів**

Модуль аутентифікації

Розділ замовлень

Розділ замовників

База даних

Ядро системи (бізнес-логіка)

Інтерфейс керування рахунками

Статистичні звіти

Розділ виконавців

Розділ

землі

Розділ

звітів

Накопичувач інформації про послуги

Модуль контролю доступа

Інтерфейс

адміністратора

Рис. 2.1. – Структура інформаційної системи бази даних виконаних робіт з ландшафтного дизайну.

Інформаційна система

Ідентифікація користувачів

Ведення нормативно - правової бази

Збереження даних

Підтримка бази даних

Документаційне забезпечення

Рис. 2.2. Функціональна схема бази даних виконаних робіт з ландшафтного дизайну.

Ідентифікація користувачів

Ввід інформації про користувачів

Ввід даних

Правильність вводу

Збереження інформації

Огляд даних

Обробка інформації

Пошук інформаці

Завдання умов пошуку

Виконання умов пошуку

Перегляд результату пошуку

Сортування інформації

Завдання умов сортування

Виконання умов сортування

Перегляд результату сортування

Рис. 2.3. Структурна схема функції системи <*Ідентифікація користувача*>

**2.2 Проектування структури бази даних**

Відповідно до поставлених вимог, та особливостей предметної області що були отримані у ході розроблення минулого розділу була спроектована база даних, що спроможна у повній мірі забезпечити поставлені до готового програмного рішення вимоги.

Основою для даної структури бази даних стала інформація зібрана у попередніх розділах, та наявність таких вимог, як правильність обробки інформації, та можливість створювати, видаляти, та оновляти інформацію.

Даний етап являє собою однину з найважливіших частин виконання курсового проекту, адже на основі даного розділу буде базуватися уся наступна робота і безпосередньо сам програмний продукт і саме тому для виконання цього розділу потрібно надзвичайно продумано виконувати усі завдання .

Внаслідок проектування бази даних для програмного продукту було створені наступні таблиці:

* zakaz
* land
* customer
* work
* designer
* zakaz\_difficulties
* difficulties

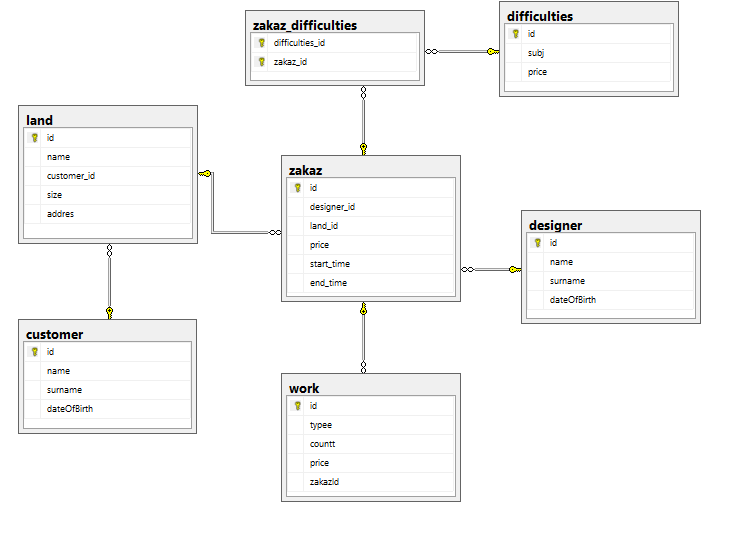


Рис. 2.4. – ERD діаграма бази даних

Основною таблицею, навколо якої відбувається основна взаємодія, це таблиця zakaz. В дану таблицю вводиться основна інформація про замовлення.

Таблиця 2.1

Структура таблиці "zakaz"

| Назва | Тип даних | ПК | ЗК | Опис поля |
| --- | --- | --- | --- | --- |
| id | int identity | + | - | Код замовлення |
| designer\_id | int | - | + | Код дизайнера |
| land\_id | int | - | + | Код землі |
| price | int | - | - | Ціна замовлення |
| start\_time | Date | - | - | Дата початку робіт |
| end\_time | Date | - | - | Дата кінця робіт |

Таблиця land містить у собі дані про певну територію, та код замовника, по якому можна знайти замовника.

Таблиця 2.2

Структура таблиці "land"

| Назва | Тип даних | ПК | ЗК | Опис поля |
| --- | --- | --- | --- | --- |
| id | int identity | + | - | Код території |
| name | int | - | - | Назва території |
| customer\_id | int | - | + | Код замовника |
| size | int | - | - | Розмір землі |
| addres | varcar(36) | - | - | Адреса землі |

Таблиця customer містить основну інформацію про замовника, це – ім'я, прізвище, дана народження.

Таблиця 2.3

Структура таблиці "customer"

| Назва | Тип даних | ПК | ЗК | Опис поля |
| --- | --- | --- | --- | --- |
| id | int identity | + | - | Код замовника |
| name | varcar(36) | - | - | Ім’я замовника |
| surname | varcar(36) | - | - | Прізвище замовника |
| dateOfBirth | Date | - | - | Дата народження замовника |

Таблиця work містить основну інформацію про певну одиницю роботи, яка була виконана на певній терирорії.

Таблиця 2.4

Структура таблиці " work "

| Назва | Тип даних | ПК | ЗК | Опис поля |
| --- | --- | --- | --- | --- |
| id | int identity | + | - | Код виконаної роботи |
| typee | varcar(36) | - | - | Тип виконаної роботи |
| countt | int | - | - | Кількість виконаної роботи |
| price | int | - | - | Ціна виконаної роботи |
| zakazId | int | - | + | Код замовлення |

Таблиця designer містить основну інформацію про замовника.

Таблиця 2.5

Структура таблиці " designer "

| Назва | Тип даних | ПК | ЗК | Опис поля |
| --- | --- | --- | --- | --- |
| id | int identity | + | - | Код дизайнера |
| name | varcar(36) | - | - | Ім’я дизайнера |
| surname | varcar(36) | - | - | Прізвище дизайнера |
| dateOfBirth | Date | - | - | Дата народження дизайнера |

Таблиця zakaz\_difficulties містить навігаційну інформацію стосовно замовлення, та труднощів що виникали під час виконання роботи.

Таблиця 2.6

Структура таблиці " zakaz \_difficulties "

| Назва | Тип даних | ПК | ЗК | Опис поля |
| --- | --- | --- | --- | --- |
| difficulties\_id | int | + | + | Код дизайнера |
| zakaz\_id | int | + | + | Ім’я дизайнера |

Таблиця difficulties містить інформацію стосовно труднощів що виникали під час виконання певної роботи

Таблиця 2.7

Структура таблиці " zakaz \_difficulties "

| Назва | Тип даних | ПК | ЗК | Опис поля |
| --- | --- | --- | --- | --- |
| difficulties\_id | int | + | + | Код дизайнера |
| zakaz\_id | int | + | + | Ім’я дизайнера |

**2.3 Розробка математичної моделі та алгоритмів обробки даних в системи**

Формула розрахунку вартості замовлення:

S = LS \* 200 + ∑DF + ∑(WС \* WP)

Де S – вартість замовлення.

DF – ціна трудності певного замовлення.

WС – кількість одиниць виконаних робіт певної роботи.

WP – ціна за виконання певної роботи.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОБРОБКИ ДАНИХ

**3.1 Проектування інтерфейсу обробки даних**

БД

Інтерфейс

Запити

(процедури)

Форми

Звіти

По замовленню

По замовленням

Ордер по розміру землі і кількості робіт

Реєстрації

Авторизації

Створення/редагування замовлень

CRUD запити для отримання, редагування та збереження даних

Пошук замовників за ім’ям та прізвищем.

Пошук виконавців за ім’ям та прізвищем.

Створення/редагування виконавців

Створення/редагування замовників

Створення/редагування робочих ділянок

Рис. 3.1Структура інтерфуйсу бази даних виконаних робіт з ландшафтного дизайну.

Результати усіх запитів відображаються у вигляді web-сторінок.

Усі звіти формуються безпосередньо у ядрі додатку і повертаються у вигляді pdf документу.

**3.2 Реалізація операцій обробки даних в БД.**

Для забезпечення роботи інтерфейсу було використано:

* JavaScript
* jQuery
* ASP.NET MVC Framework
* Entity Framework

У ролі сервера для даного курсового проекту було прийнято рішення використовувати SQL Server.

Доступ до даних у даному веб-проекті здійснюється за допомогою об’єктно-орієнтованої технології доступу до даних Entity Framework, що надає можливість взаємодії з об’єктами як засобами LINQ у вигляді LINQ to Entities, так і засобами  Entity SQL.

Для встановлення зв’язку з базою даних використовувався підхід Database First, що за готовою базою даних створює відповідну модель, яка згодом буде використовуватися для підключення до бази даних.

У результаті проробленої роботи ми отримали базу даних виконаних робіт з ландшафтного дизайну, у якій ми можемо зберігати усю необхідну інформацію про колекцію виконаних робіт з ландшафтного дизайну.

Також була отримана можливість додавати, редагувати та видаляти замовлення.

Для демонстраційного прикладу роботи вибірки даних з бази даних продемонструємо роботу вибірки списку замовлень.

using (LD\_kursEntities db = new LD\_kursEntities())

{

foreach (var item in zakaz)

{

InfoList.Add(new ZakazInfo()

{

zakaz = Mapper.Map<Zakaz>(item),

land = Mapper.Map<Land>(item.land),

customer = Mapper.Map<Customer>(item.land.customer),

designer = Mapper.Map<Designer>(item.designer),

difficults = Mapper.Map<List<Difficulties>>(item.difficulties),

work = Mapper.Map<List<Work>>(item.work)

});

}

}

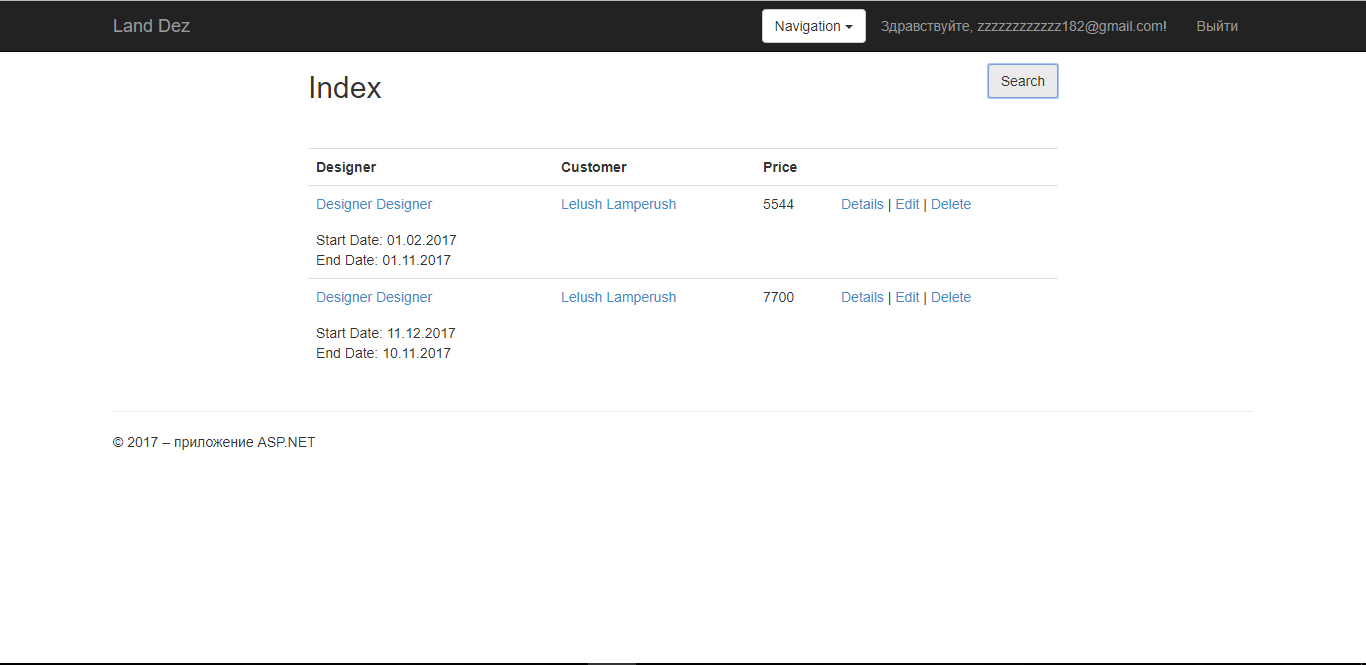


Рис. 3.2. – Список виконаних робіт.

Прикладом для демонстрації функції модифікації даних служить наступний програмний код.

using (LD\_kursEntities db = new LD\_kursEntities())

{

db.Entry(Mapper.Map<customer>(\_customer)).State = EntityState.Modified;

return db.SaveChanges();

}

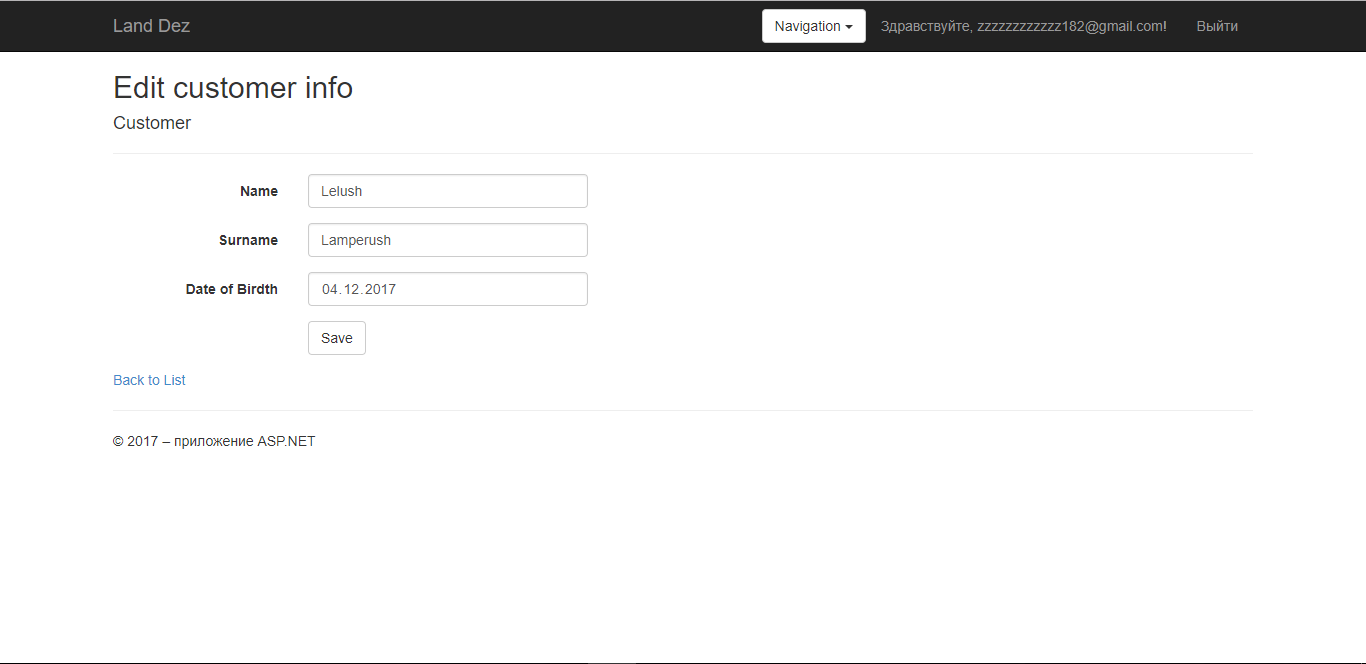


Рис. 3.3. – Редагування даних замовника.

Прикладом для демонстрації функції видалення даних служить наступний програмний код видалення замовлення.

using (var transaction = db.Database.BeginTransaction()){

try

{

foreach (var item in zakaz.difficulties.ToList())

{

db.Entry(item).State = EntityState.Deleted;

}

foreach (var item in zakaz.work.ToList())

{

db.Entry(item).State = EntityState.Deleted;

}

db.Entry(zakaz).State = EntityState.Deleted;

db.SaveChanges();

transaction.Commit();

}

catch (Exception ex)

{

transaction.Rollback();

}

}

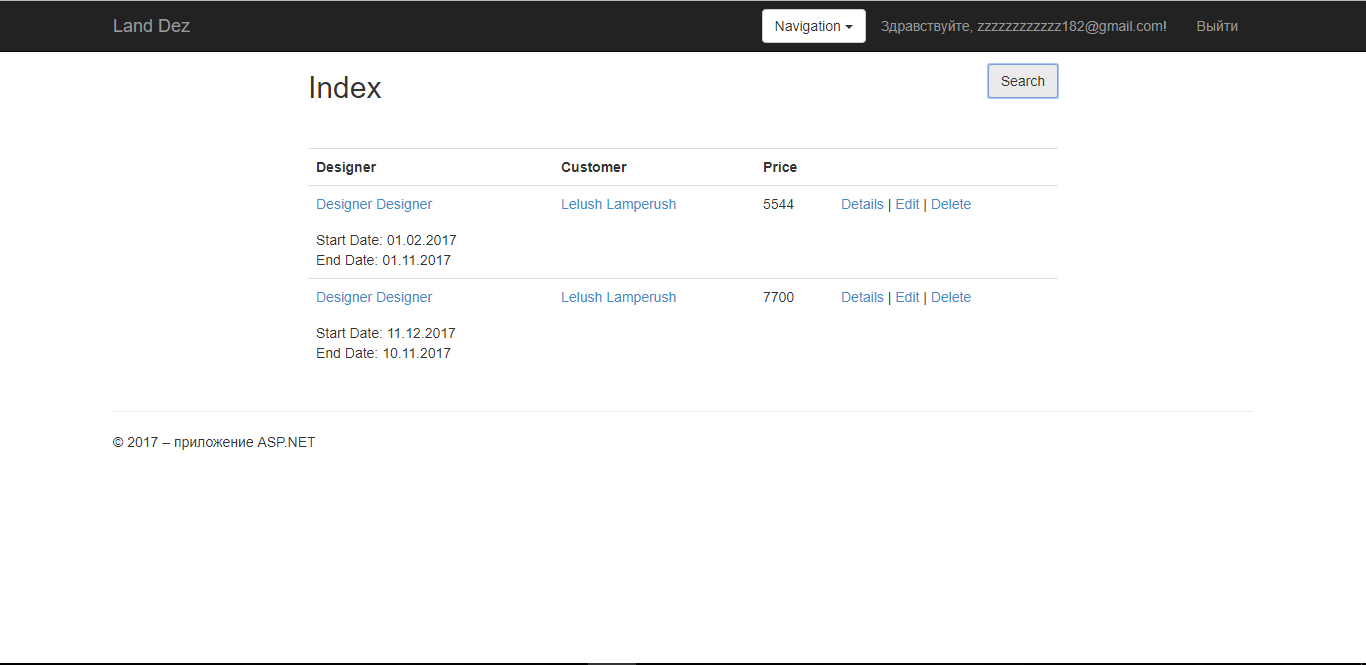


Рис. 3.4. – Видалення виконаних робіт.

**3.3 Організація звітності системи**

У даному курсовому проектові мною були реалізовані такі звіти:

* Звіт по окремому замовленню.
* Звіт по розміру обробленої земельної ділянки, та кількості робіт у замовленнях з можливістю фільтрації по даті виконання замовлення відповідно за періоди: за весь час, за рік та за 1 місяць.
* Звіт по розмірам обробленої земельної ділянки за критеріями дати та мінімального розміру земельної ділянки.

Звіти по окремому замовленню можна отримати на сторінці того замовлення, звіт по якому потрібно отримати.

Для загальної звітності передбачена окрема сторінка. Вигляд сторінки для відображення звітів показано на рисунку 3.5.

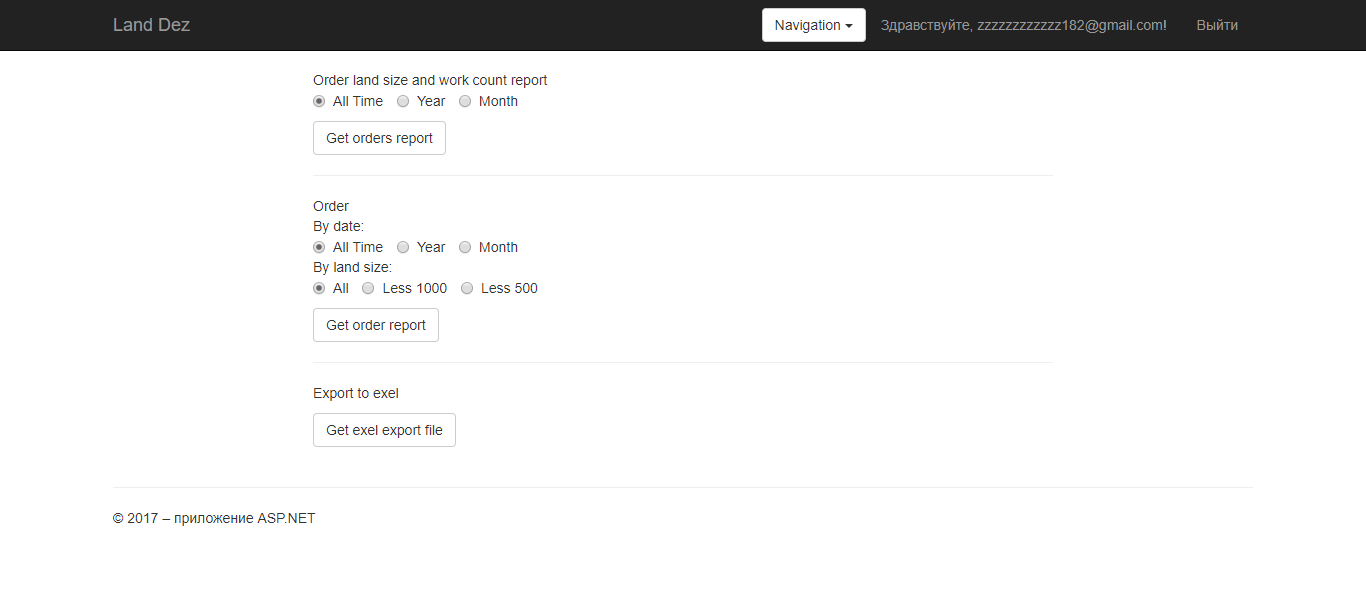


Рис. 3.5. – Сторінка звітності.

РОЗДІЛ 4 АДМІНІСТРУВАННЯ БАЗ ДАНИХ

**4.1. Розробка заходів захисту інформації в БД**

Методи захисту інформації регламентуються на рівні додатку. Для підключення до бази даних на рівні додатку використовується методика реєстрації/авторизації користувачів. Безпосередньо сам захист виконується на рівні встановлення ролей користувача.

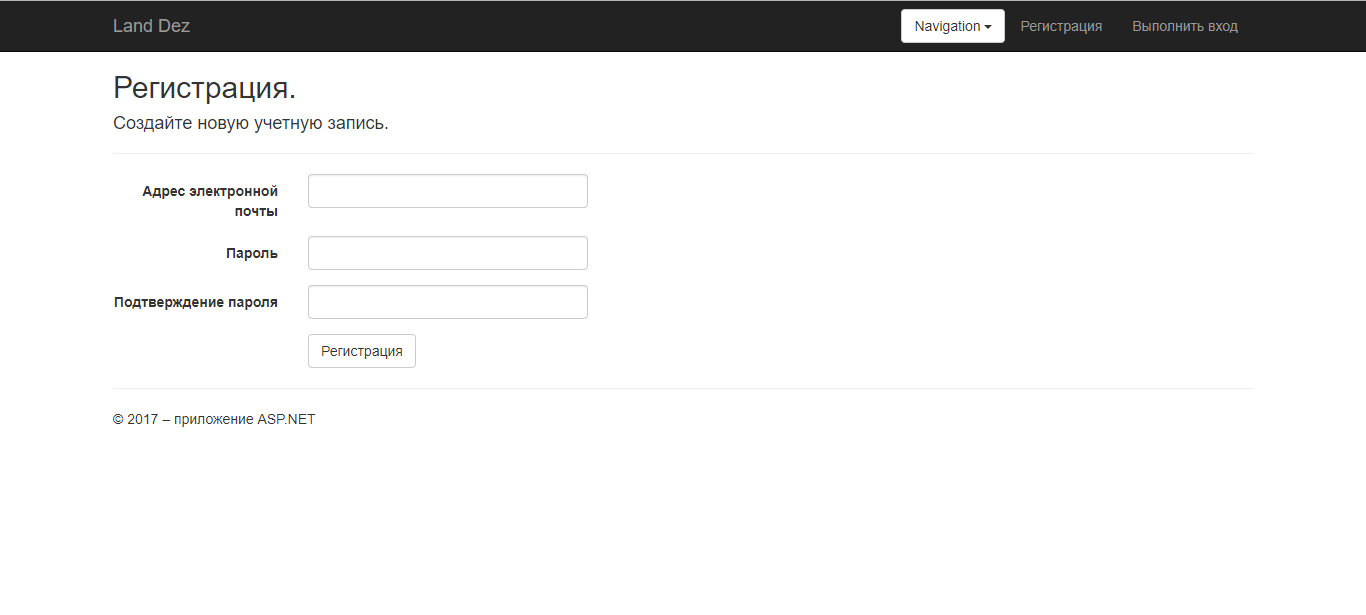


Рис. 3.5. – Форма реєстрації.

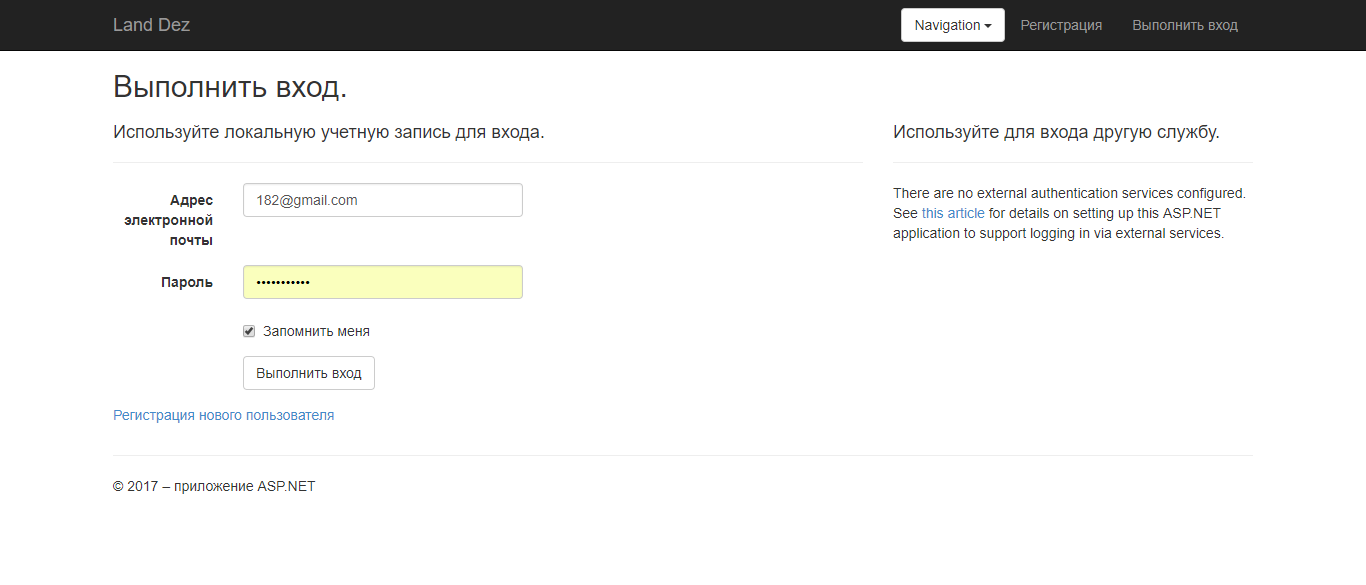


Рис. 3.6. – Форма авторизації.

Детально про об’єкти бази даних, доступ до яких надається ролям «Користувач», та «Адміністратор» наведено у таблиці 4.1. На перетині рядків і стовпців зазначено дії, які може виконувати користувач даного типу ролі: 0 – немає доступу, 1 – перегляд даних, 2 – редагування, 3 – видалення, 4 – вставка даних, 5 - повний доступ(виконання процедури).

Таблиця 4.1

Матриця доступу до об’єктів бази даних

|  |  |  |
| --- | --- | --- |
|  | Користувач | Адміністратор |
| Замовлення | 1 | 5 |
| Робота | 1 | 5 |
| Замовлення-робота | 1 | 5 |
| Дизайнер | 1 | 5 |
| Земля | 1 | 5 |
| Труднощі | 1 | 5 |
| Замовник | 1 | 5 |

Усі запити повністю захищені від користувачів, що не мають необхідного рівня доступу. Користувачам без відповідного рівня доступу навіть не показуються можливі посилання на ресурси до яких немає доступу.

При спробі ручного переходу на сторінку, на яку у користувача немає доступу, йому буде показано відповідна сторінка, на якій йому буде запропоновано зайти з іншого акаунту, адже даний акаунт не має достатніх прав на взаємодію з даним ресурсом.

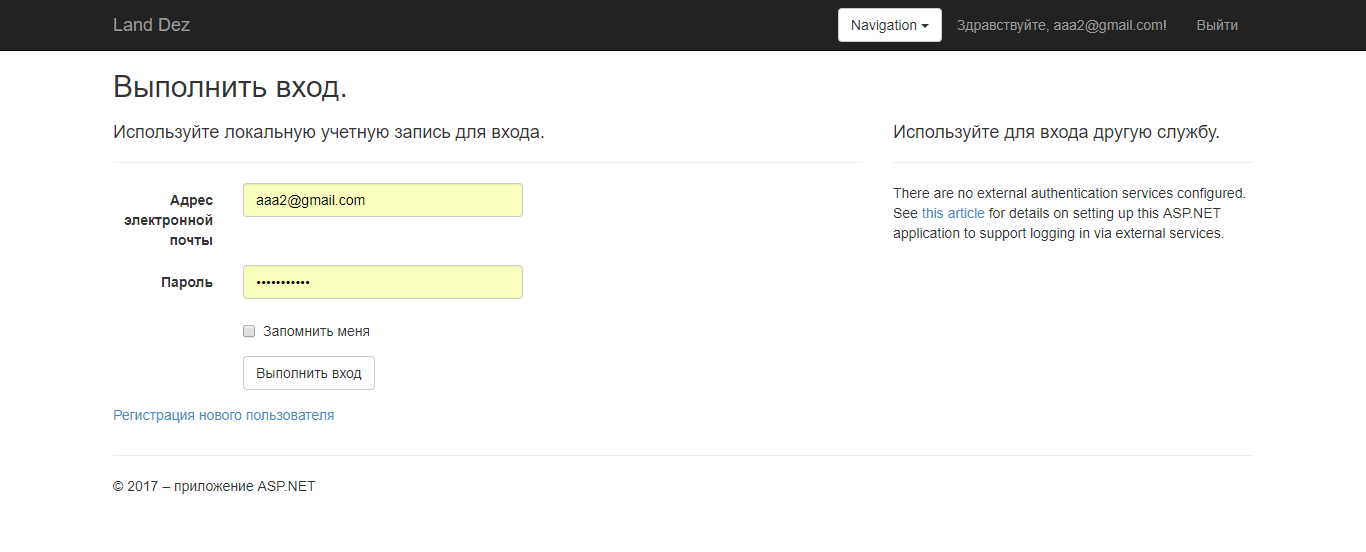


Рис. 3.7. – Форма авторизації при спробі перейти на ресурс до якого немає доступу.

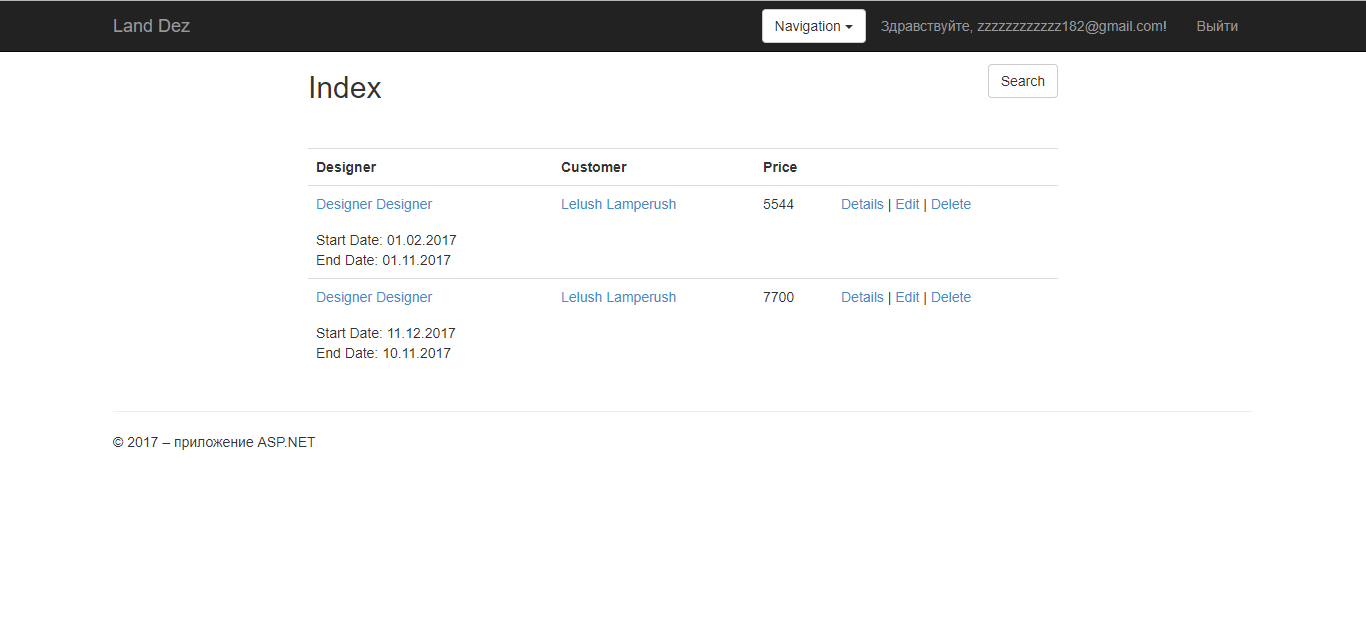


Рис. 3.7. – Вигляд Списку замовлень для користувача з правами на доступ до даних.

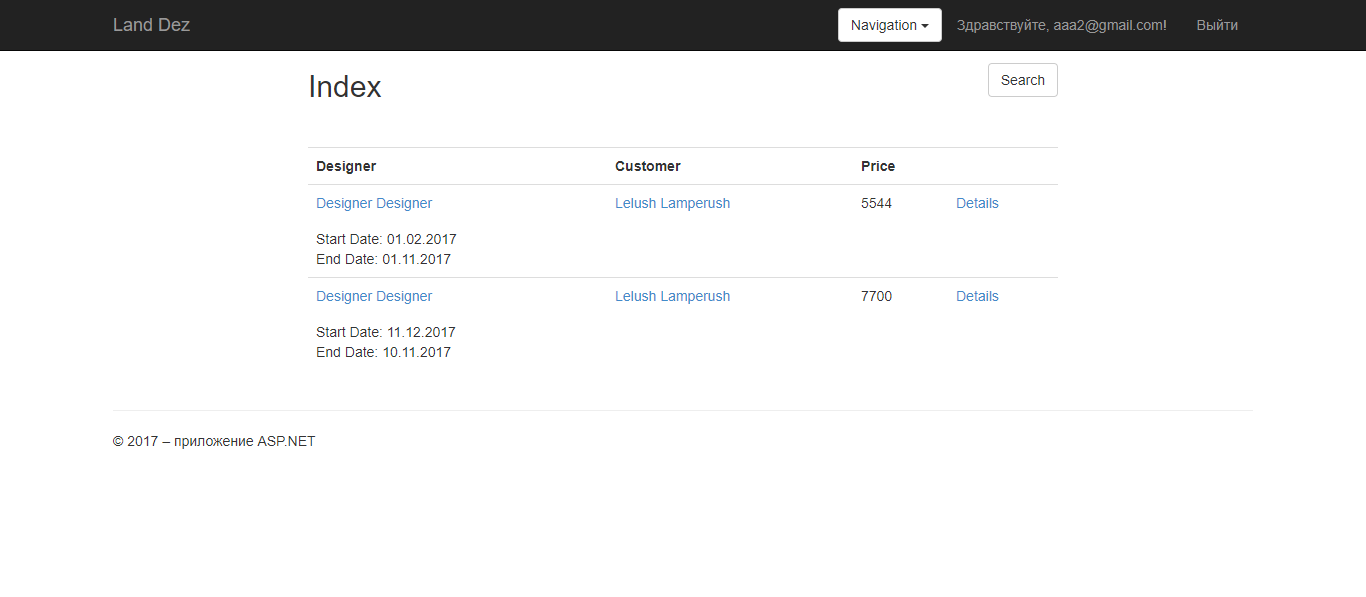


Рис. 3.7. – Вигляд Списку замовлень для користувача без прав на доступ до даних.

Функція фільтрації доступу і заборони доступу тих, у кого немає достатнього рівня доступу покладається на параметр Authorize, приклад використання: [Authorize(Roles = "admin")]

А для забезпечення динамічної зміни контенту відповідно до прав доступу на web-сторінках використовується вираз User.IsInRole, приклад використання:

@if (User.IsInRole("admin")){}

**4.2. Налаштування параметрів роботи SQL-сервера**

У попередніх пунктах ми уже визначили, що для роботи з базою даних готового програмного продукту буде використовуватися Microsoft SQL Server, а саме SQL Server 2014.

Для встановлення Microsoft SQL Server 2014 потрібно скачати файл встановлення. Далі необхідно запустити файл встановлення.

При запускові файлу установки, у меню зліва потрібно обрати пункт «Встановлення», після чого у центральній області вікна з’явиться вікно вибору, в ньому потрібно обрати пункт «Новий екземпляр SQL сервера».

Після чого будуть розпаковані усі необхідні для інсталяції файли.

У вікні що з’явилося потрібно ввести ключ продукту, або обрати варіант безкоштовної редакції програмного продукту.

Далі у вікні потрібно обрати варіант установки серед представлених:

* SQL Server Feature Installation – налаштування роблять вручну.
* SQL Server PowerPoint for SharePoint – додатково буде встановлено і сконфігуровано файл PowerPoint.
* All Features With Default – Будуть поставлені усі необхідні компоненти, з можливістю пост модеріції.

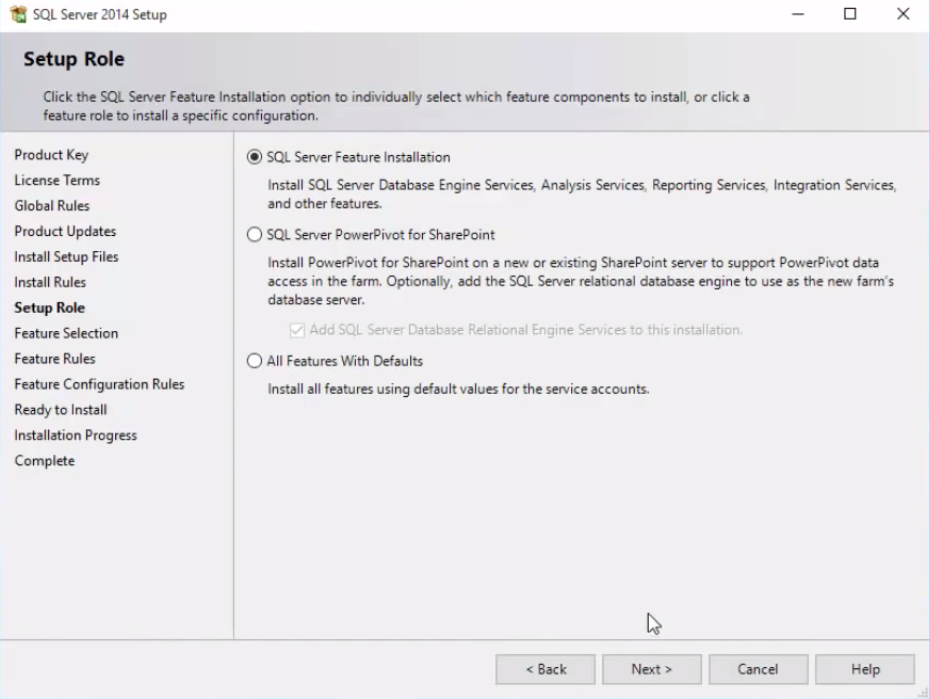


Рис. 4.1. – Вибір варіанта установки.

У наступному вікні необхідно буде обрати необхідні вам компоненти.

Перелік елементі, які можна вибрати для установки:

- Database Engine Services - SQL Server.

- SQL Server Replication - компоненти реплікації SQL Server використовуються для синхронізації баз даних.

- Full - Text Search - компонент повнотекстового пошуку дозволяє організувати ефективний пошук по текстових полям бази з обліків різних мов і різних форм слова.

- Analysis Services - дозволяє будувати багатовимірні ( OLAP ) сховища даних і DataMining моделі для проведення аналізу та побудови прогнозів.

- Reporting Services - сервіси та інструменти для побудови та управління звітами.

- Shared Features (вони ставляться 1 раз, і будуть доступні всім екземплярам, які встановлені на машині).

- Business Intelligence Development Studio - якщо стоїть Visual Studio, то до неї додаються нові типи проектів для розробки рішень Analysis Services, Reporting Services і Integration Services. Якщо Visual Studio немає, то ставиться «міні» Visual Studio, в якій доступні тільки ці, перераховані вище типи проектів.

- Client Tools Connectivity - провайдери для з'єднання клієнтів з сервером. - Integration Services - сервіси, що дозволяють організувати отримання, перетворення і перенесення даних з різних джерел.

- Client Tools Backwards Compatibility - SQL Distributed Management Objects (SQL - DMO ), Decision Support Objects ( DSO ), Data Transformation Services ( DTS ).

- Client Tools SDK - SDK для розробників.

- SQL Server Books Online - документація по SQL Server.

- Management Tools - Basic - базовий варіант Management Studio, SQLCMD і SQL Server PowerShell provider. - Management Tools

- Complete - повноцінна Management Studio (підтримка Analysis Serveices, Integration Services, Reporting Services ), Profiler, Database Engine Tuning Advisor, SQL Server Utility.

- SQL Client Tools Connectivity SDK - на Microsoft Connect є баг щодо опису цього елемента

- SQL Client Connectivity SDK and Client Tools SDK DOCUMENTATION.

- Microsoft Sync Framework - багатофункціональна платформа синхронізації, що дозволяє інтегрувати будь-який додаток з будь-якими даними з будь-якого сховища, за будь- якого протоколу і в будь-якій мережі.

Після чого відбудеться безпосередньо встановлення обраних компонентів SQL сервера.

ВИСНОВКИ

У процесі роботи над курсовим проектом було здійснено аналіз представленої предметної області. Було складено технічне завдання, на основі якого було спроектовано програмний продукт.

Спочатку була розроблена база даних виконаних робіт з ландшафтного дизайну. У результаті аналізу предметної області було виділено основну інформацію, яка необхідна для розроблення програмного продукту.

Далі для реалізації програмного продуту було обрано засоби реалізації, у вигляді ORM Entity Framework. Що являє собою зручний інструмент роботи з різними серверами баз даних. У ролі сервера баз даних було обрано Microsoft SQL Server. Що у сумі надає нам набір інструментів, що дозволяють у повній мірі дозволяє виконати усі поставлені задачі.

Проектування структури бази даних дозволяє побачити усю повноту цілей, які повинен виконувати програмний продукт, що дозволяє визначити вимоги до бази даних.

У результаті усієї проробленої роботи ми отримали повнофункціональну базу даних, у якій ми може зберігати інформацію про колекцію виконаних робіт з ландшафтного дизайну.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] ADO.NET Entity Framework [Електронний ресурс] – Доступ за посиланням: <https://ru.wikipedia.org/wiki/ADO.NET_Entity_Framework> [Дата звернення 16.12.2017.]

[2] Language Integrated Query [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Language_Integrated_Query> Дата звернення 16.12.2017.]

[3] Nick H. ASP.NET MVC Security / Harrison Nick. – Morrisville, 2015. – 85 с.

[4] Object-relational mapping [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Object-relational_mapping> [Дата звернення 16.12.2017.]

[5] Руководство по Entity Framework [Електронний ресурс] – Режим доступу до ресурсу: <https://metanit.com/sharp/entityframework/> [Дата звернення 16.12.2017.]

ДОДАТКИ