<https://udemy-certificate.s3.amazonaws.com/image/UC-a4fd27c6-68a1-459b-92df-6e017b8f5d1c.jpg?v=1666958400000>

<https://myvisit.card.uralsk-07.kz/>

пошаговая инструкция для выполнения лабораторной работы №12, включающая создание базы данных, разработку приложения и интеграцию модулей:

**1. Создание базы данных в MS SQL Server**

1. **Подключение к серверу:**
   * Откройте **SQL Server Management Studio (SSMS)**.
   * Подключитесь к серверу, указав:
     + Имя сервера.
     + Тип аутентификации (Windows или SQL Server Authentication).
   * Нажмите **Connect**.
2. **Создание базы данных:**
   * В панели **Object Explorer** щелкните правой кнопкой на **Databases** → **New Database**.
   * Укажите имя базы данных (например, UserManagementDB) и нажмите **OK**.
3. **Создание таблиц:**
   * Выберите созданную базу данных.
   * Щелкните правой кнопкой на **Tables** → **New Table**.
   * Создайте таблицу Users с такими полями:
     + UserID (int, Primary Key, Identity: начальное значение = 1, шаг = 1).
     + UserName (nvarchar(50), Not Null).
     + PasswordHash (nvarchar(64), Not Null).
     + Email (nvarchar(100), Not Null).
     + SecretQuestionID (int, Foreign Key).
   * Создайте таблицу SecretQuestions:
     + SecretQuestionID (int, Primary Key, Identity).
     + Question (nvarchar(255), Not Null).
   * Сохраните таблицы.
4. **Связь таблиц:**
   * Откройте **Database Diagrams** (щелкните правой кнопкой и выберите **New Diagram**).
   * Перетащите таблицы Users и SecretQuestions.
   * Свяжите поле SecretQuestionID из Users с SecretQuestionID в SecretQuestions.
   * Сохраните диаграмму.

**2. Создание приложения с окном авторизации**

1. **Создание проекта:**
   * В Visual Studio выберите **Create a new project**.
   * Выберите шаблон **Windows Forms App (.NET Framework)**.
   * Укажите имя проекта, например, UserAuthApp.
2. **Разработка окна авторизации:**
   * В дизайнере формы создайте интерфейс:
     + Поле для ввода имени пользователя (txtUsername).
     + Поле для ввода пароля (txtPassword, свойство UseSystemPasswordChar = True).
     + Кнопка **Войти** (btnLogin).
     + Метка для ошибок авторизации (lblError, свойство Visible = False).
   * Добавьте обработчик событий для кнопки **Войти**.

**3. Создание модели данных**

1. **Добавление Entity Framework:**
   * В меню **Tools** → **NuGet Package Manager** → **Manage NuGet Packages for Solution**.
   * Установите пакет **EntityFramework**.
2. **Создание модели данных:**
   * Щелкните правой кнопкой на проекте → **Add** → **New Item** → **ADO.NET Entity Data Model**.
   * Выберите вариант **EF Designer from Database**.
   * Укажите подключение к вашей базе данных.
   * Выберите таблицы Users и SecretQuestions для импорта.
   * Сохраните модель.

**4. Сохранение данных в базу**

1. **Добавление данных в таблицу SecretQuestions:**
   * В SSMS выполните скрипт:

sql

Копировать код

INSERT INTO SecretQuestions (Question)

VALUES

('What is your mother\'s maiden name?'),

('What was the name of your first pet?'),

('What is your favorite color?');

1. **Код для сохранения пользователей:**
   * В проекте создайте класс UserService для работы с базой:

csharp

Копировать код

using System;

using System.Linq;

public class UserService

{

private readonly UserManagementDBEntities \_db;

public UserService()

{

\_db = new UserManagementDBEntities();

}

public bool Login(string username, string password)

{

string hashedPassword = HashPassword(password); // Реализуйте метод хэширования

var user = \_db.Users.FirstOrDefault(u => u.UserName == username && u.PasswordHash == hashedPassword);

return user != null;

}

public void Register(string username, string password, string email, int secretQuestionId)

{

string hashedPassword = HashPassword(password);

var newUser = new User

{

UserName = username,

PasswordHash = hashedPassword,

Email = email,

SecretQuestionID = secretQuestionId

};

\_db.Users.Add(newUser);

\_db.SaveChanges();

}

private string HashPassword(string password)

{

using (var sha256 = System.Security.Cryptography.SHA256.Create())

{

var hashedBytes = sha256.ComputeHash(System.Text.Encoding.UTF8.GetBytes(password));

return BitConverter.ToString(hashedBytes).Replace("-", "").ToLower();

}

}

}

**5. Интеграция и тестирование**

1. **Обработка авторизации:**
   * Добавьте в форму код для обработки кнопки **Войти**:

csharp

Копировать код

private void btnLogin\_Click(object sender, EventArgs e)

{

UserService userService = new UserService();

bool isValid = userService.Login(txtUsername.Text, txtPassword.Text);

if (isValid)

{

MessageBox.Show("Login successful!");

}

else

{

lblError.Text = "Invalid username or password.";

lblError.Visible = true;

}

}

1. **Регистрация пользователя:**
   * Реализуйте отдельную форму для регистрации, чтобы пользователь мог добавить свои данные.
2. **Проверка работы:**
   * Проверьте авторизацию с созданными пользователями.
   * Убедитесь, что данные сохраняются корректно в базе данных.

**6. Отчет**

1. **Название работы:**
   * Лабораторная работа №12. Разработка и интеграция модулей проекта.
2. **Цель работы:**
   * Закрепить навыки работы с Visual Studio 2019, MS SQL Server и Entity Framework.
3. **Ход работы:**
   * Опишите шаги выполнения работы (создание базы данных, разработка приложения, интеграция).
4. **Вывод:**
   * Сделайте выводы о проделанной работе.
5. **Контрольные вопросы:**
   * Ответьте на вопросы:
     1. **Что такое интеграция?** Интеграция — процесс объединения отдельных модулей системы для обеспечения их совместной работы.
     2. **Как реализовывается создание базы данных?** Создание базы данных выполняется в MS SQL Server через Management Studio, где создаются таблицы, связи и ограничения.
     3. **Что такое модель данных?** Модель данных — это объектное представление данных и их связей, используемое для взаимодействия с базой данных.
     4. **Как передаются данные в MSSQL Server?** Данные передаются через запросы SQL, либо с использованием ORM, таких как Entity Framework.