**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**



**З В І Т**

**до лабораторної роботи №3**

з курсу: «Інженерія програмного забезпечення»

на тему: «Розробка серверної частини. Розробка комунікації за протоколом TCP. Підключення серверного модуля до БД.»

Варіант № 26

***Виконав:***

*Студент групи КІ-301*

*Солтисяк О.Р.*

***Прийняв:***

*Іванов Ю.С.*

Львів – 2024

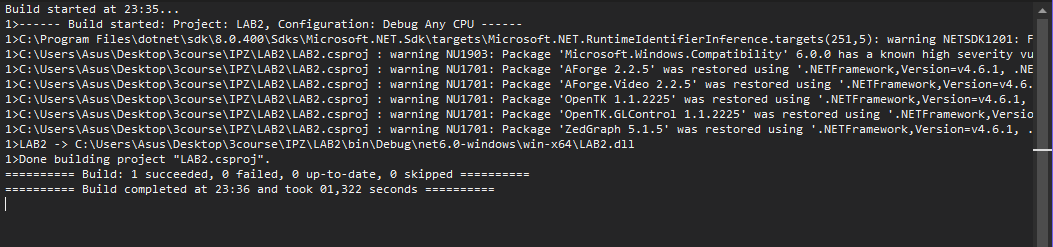
***Тема:*** Розробка серверної частини. Розробка комунікації за протоколом TCP. Підключення серверного модуля до БД.

***Мета роботи:***  Розробити консольну аплікацію що буде підтримувати зв’язок по протоколу TCP/IP, отримувати дані та записувати у БД. Також, згідно деякої команди, вичитувати з БД необхідну інформацію та передавати по TCP протоколу на клієнтську частину.

**Завдання**

|  |  |
| --- | --- |
| № п/п | Завдання |
| 26 | Розробити консольну аплікацію що буде підтримувати зв’язок по протоколу TCP/IP, отримувати дані та записувати у БД. Також, згідно деякої команди, вичитувати з БД необхідну інформацію та передавати по TCP протоколу на клієнтську частину.  ВАРІАНТ ЗАВДАННЯ:  **Комплексна система охорони множини об’єктів (видеоспостереження, аналіз та сповіщення про загрозу).** |

**Виконання**

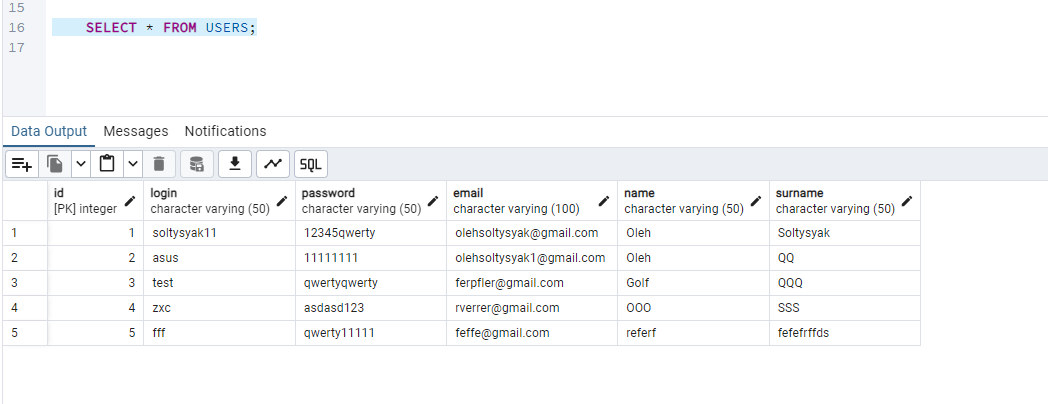


*Рисунок 1.1. Вікно програми*

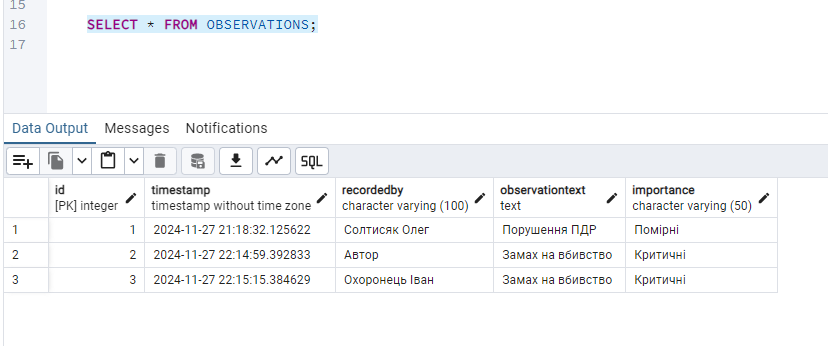
Пропоную розглянути кожну функцію серверної частини окремо:

1. **RegisterForm.cs / RegisterForm.Designer.cs**: Ці файли відповідають за форму реєстрації.
   * *RegisterForm.cs*: Тут написано логіку поведінки форми — як обробляти дані при реєстрації, як взаємодіяти з сервером, перевірки тощо.
   * *RegisterForm.Designer.cs*: Цей файл генерується автоматично і відповідає за розміщення елементів управління на формі, таких як кнопки, текстові поля, тощо. Він містить лише візуальну частину форми.
2. **LoginForm.cs / LoginForm.Designer.cs**:
   * *LoginForm.cs*: Логіка авторизації користувача, перевірка введених даних та взаємодія з сервером для авторизації.
   * *LoginForm.Designer.cs*: Візуальний дизайн форми авторизації, розміщення текстових полів, кнопок, тощо.
3. **VideoMonitoringForm.cs / VideoMonitoringForm.Designer.cs**:
   * *VideoMonitoringForm.cs*: Логіка для форми відео-моніторингу, зокрема, запуск відео з камери, обробка кадрів, взаємодія з базою даних для збереження записів спостережень.
   * *VideoMonitoringForm.Designer.cs*: Відповідає за розміщення елементів на формі для відео-моніторингу — це може бути екран для показу відео, кнопки для запуску/зупинки моніторингу, текстові поля для запису даних.
4. **Server.cs**: Цей файл містить серверну частину, яка приймає підключення від клієнтів, обробляє запити, взаємодіє з базою даних для авторизації, реєстрації та запису даних.
5. **DataBaseInfo.cs**: Цей файл містить функціонал для роботи з базою даних, наприклад, з'єднання з базою, виконання SQL запитів і закриття з'єднань.
6. **Program.cs**: Це основний файл програми, який відповідає за запуск додатку. Він може містити Main() метод, який ініціалізує форми, запуск сервера, або основні налаштування для всього проєкту.

Ці файли разом реалізують взаємодію між клієнтським інтерфейсом (реєстрація, авторизація, відео-моніторинг) та серверною частиною, яка працює з базою даних і зберігає дані спостережень чи інформацію про користувачів.

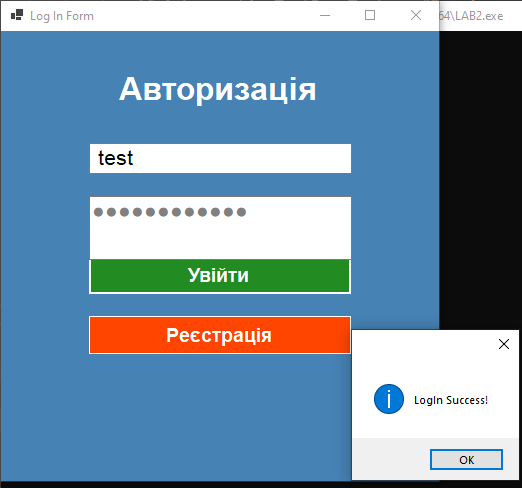


*Рисунок 1.2. База даних зареєстрованих користувачів (users)*

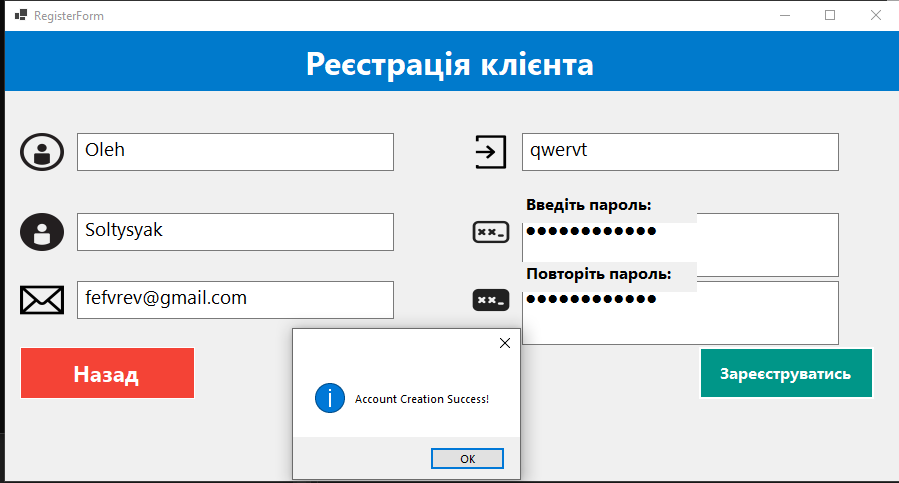


*Рисунок 1.3. База даних, яка містить у собі інформацію про спостереження та авторів*

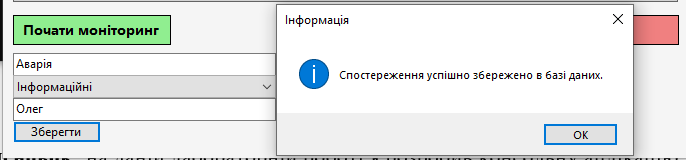
1. **Процес авторизації:**



1. **Процес реєстрації: (клієнтська та серверна частина)**



1. **Процес зчитування та запису у базу даних: (клієнтська та серверна частина)**



**Висновок:** на даній лабораторній роботі я розробив консольну аплікацію, яка підтримує зв’язок по протоколу TCP/IP, отримує дані та записує їх у БД. Також, згідно деяких команд, вона вичитує з БД необхідну інформацію та передає її по TCP протоколу на клієнтську частину.