|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ | | | | | | | | | | |
|  | | | |  | | |  | | | |
| ФГБОУ ВПО «Пермский государственный национальный исследовательский университет» | | | | | | | | | | |
|  | |  | | | | | | |  | |
|  | | ОТЧЕТ  по лабораторной работе «Регистрация и аутентификация» по дисциплине «Учебная практика по базам данных» | | | | | | |  | |
|  | | |  | | | | |  | | |
|  | Работу выполнил  студент гр. ПМИ-1,2  Васильевых М.Г. \_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 | | | |  | Проверил  ассистент кафедры МОВС  Турова И.А. \_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 | | | |  |
|  | |  | | | | | | |  | |
|  | | Пермь 2021 | | | | | | |  | |

Содержание

[1 Постановка задачи 3](#_Toc83312391)

[2 Описание предметной области и базы данных 4](#_Toc83312392)

[3 Описание функционала разработанной программы 5](#_Toc83312393)

[3.1. Внешний вид и функционал приложения 5](#_Toc83312394)

[3.2 Общий алгоритм решения задачи 7](#_Toc83312395)

[4 Реализация программы 8](#_Toc83312396)

[5 Тестирование программы 13](#_Toc83312397)

# 1 Постановка задачи

Написать (доработать) программу, которая на графическом интерфейсе пользователя позволяет зарегистрировать нового пользователя с логином и паролем (т.е. занести соответствующую запись в базу данных). Для хранения хэша пароля выбрать подходящий тип (ограничить длину). Отметка времени регистрации должна сохраняться в базу данных посредством указания значения по умолчанию для столбца. Пароль должен сохраняться в виде хеш-кода, зависящего от сгенерированного для этого пользователя модификатора (соли) либо сгенерированного с помощью одной из адаптивных хэш-функций BCrypt, SCrypt, etc. Создание пользователя с существующим логином запрещено. Регистрация пользователей с пустым логином запрещена. Регистрация должна разрешаться только в случае, если используется сложный пароль, т.е. пароль, содержащий буквы разного регистра, цифры и спец. символы.

Написать программу, которая на графическом интерфейсе пользователя позволяет аутентифицировать ранее зарегистрированного пользователя (проверить логин и пароль).

# 2 Описание предметной области и базы данных

Для работы приложения нам таблица из базы данных «Пользователи» - рисунок 1.1

В базе данных хранится информация о зарегистрированных пользователях, логин, пароль (в виде хеш строки), дата регистрации пользователя. Представленная база данных реляционного типа.

Id имеет тип integer – целочисленный

Login имеет тип text – строка без ограничений по размеру

Password имеет тип char(60) – символьный массив длинной 60 символов так как алгоритм хэширования подразумевает хэш блинной строго 60 символов

Time имеет тип timestap – с установленным значение по умолчанию дата и время сейчас

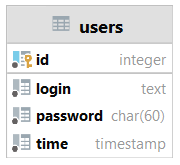


Рисунок 1.1 – Используемая БД

# 3 Описание функционала разработанной программы

3.1. Внешний вид и функционал приложения

При запуске приложения регистрации отображается главная форма. На ней расположены элементы для ввода значения(textbox) логина и пароля нового пользователя и кнопка(button) для регистрации пользователя, над ней текстовое поле(label) с информацией о результате регистрации.

Если пользователь ввёл логин, который уже существует в базе или пустой логин, то приложение выводит в текстовое поле соответствующее сообщение. Если пользователь ввёл пароль не соответствующий принятым условиям, то в текстовое поле выведется соответствующее сообщение. Внешний вид приложения представлен на рис. 1.

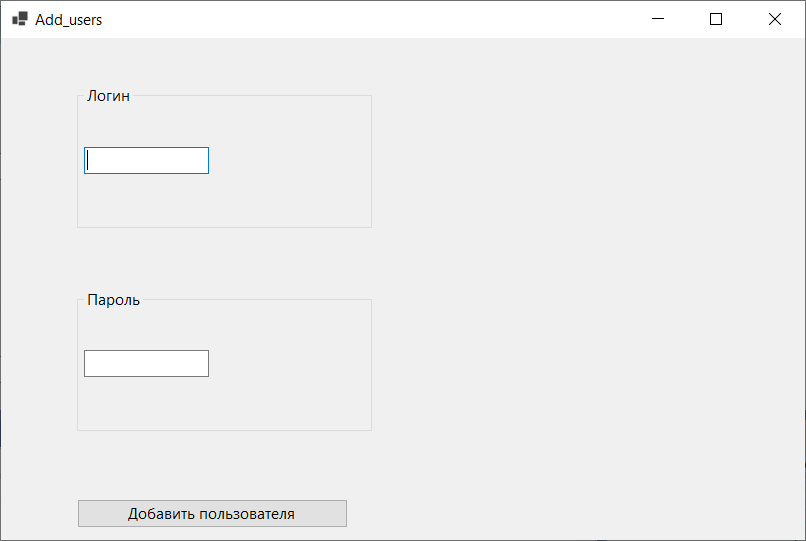


Рис. 1. Главная форма приложения регистрации

При запуске приложения аутентификации отображается главная форма. На ней расположены элементы для ввода значения(textbox) логина и пароля нового пользователя и кнопка(button) для регистрации пользователя, над ней текстовое поле(label) с информацией о результате регистрации.

Если пользователь ввёл логин, которого не существует в базе, то приложение выводит в текстовое поле соответствующее сообщение. Если пользователь ввёл пароль не соответствующий данному логину, то в текстовое поле выведется соответствующее сообщение. Внешний вид приложения представлен на рис. 1.

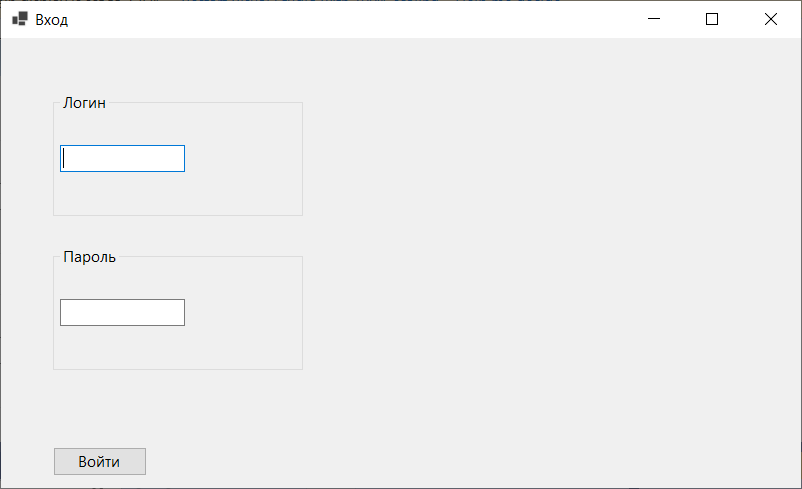


Рис. 2. Главная форма приложения аутентификации

## 3.2 Общий алгоритм решения задачи

Данную задачу можно разбить на 2 основные подзадачи: регистрация и аутентификация пользователя.

1)Регистрация пользователя

Сначала получил логин и пароль нового пользователя. С использование таблицы подготовим выборку в соответствии логинам идентичным введенным пользователем.

Если при проверке в таблице не оказалось таких логинов, переходим к проверке соответствия пароля заданным стандартам безопасности, проверяя наличие хотя бы одного символа из каждого необходимого множества при помощи операции пересечения множеств.

Если оба условия выполнены, выполняем шифрование пароля, и передаём значение о новом пользователе в таблицу, время регистрации устанавливается по умолчанию в БД.

2) Аутентификация пользователя

Сначала получил логин и пароль нового пользователя. С использование таблицы подготовим выборку в соответствии логинам идентичным введенным пользователем и соответствующим им хэш строкам пароля.

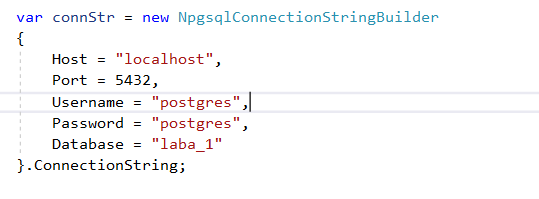
Сравним полученный пароль с вычисленной хэш строкой.

Если пароль прошёл проверку то выдаём сообщение об аутентификации, в ином случае о неверном пароле или о не существовании данного логина в базе.

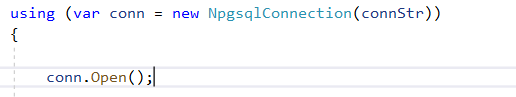
# 4 Реализация программы

В первую очередь, создается строка подключения к БД. Для этого

необходимо создать экземпляр класса NpgsqlConnectionStringBuilder в который будет передана информацию о БД, а именно: адрес сервера, инициализируемый каталог (каталог, содержащий БД) и данные для аутентификации, в качестве которого, в нашем случае, выступает проверка подлинности Windows.

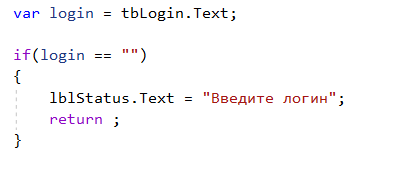


Затем открываем соединение с БД, сначала создав экземпляр объекта

NpgsqlConnection, а затем применив к нему метод Open. 

Эти действия общие для обоих программ.

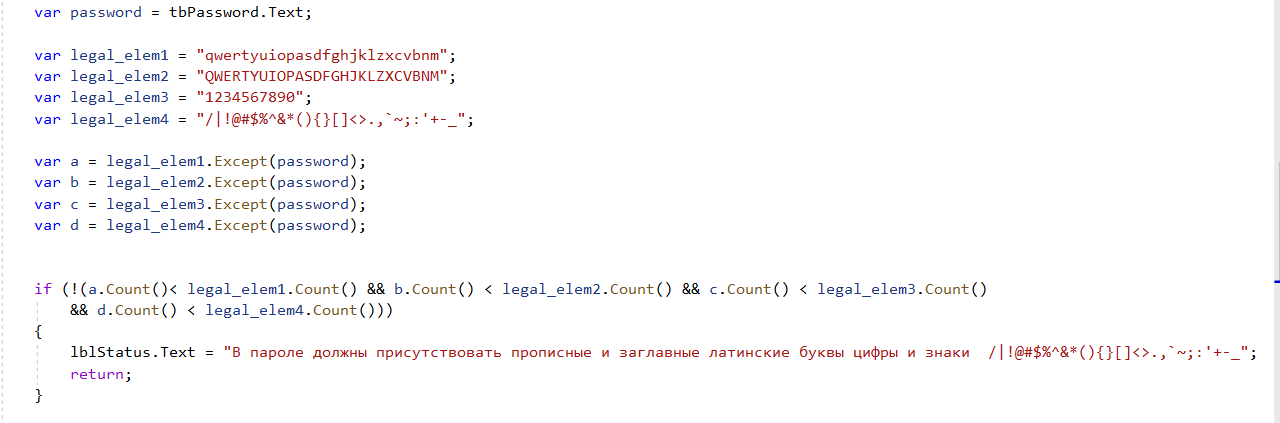
С начало разберем работу программы регистрации.



Получаем логин.



Формируем запрос, о количестве пользователей с таким же логином и отправляем его в БД при использовании параметра для защиты от инъекций в запрос. С помощью него проверяем наличие такого логина (если запрос вернул 1 то логин есть если 0 то нет) в базе, если он есть выдаём соответствующее сообщение, если нет то продолжаем работу дальше.

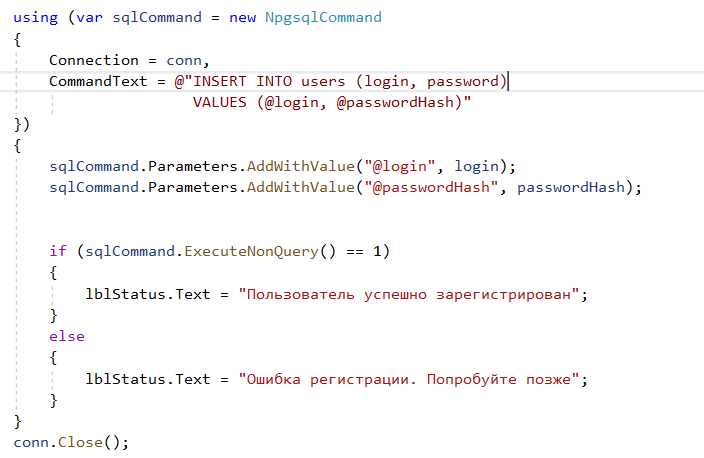


Формируем множество элементов необходимых в пароле, и проверяем, есть ли они в ведённом пароле при помощи операций разности строк.

То есть из каждого множества вычитаем множества пароля, а затем проверяем, изменился ли размер множеств если да то в пароле есть легальные все легальные элементы.

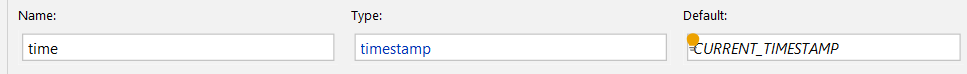


Кодируем пароль при помощи алгоритма BCrypt.



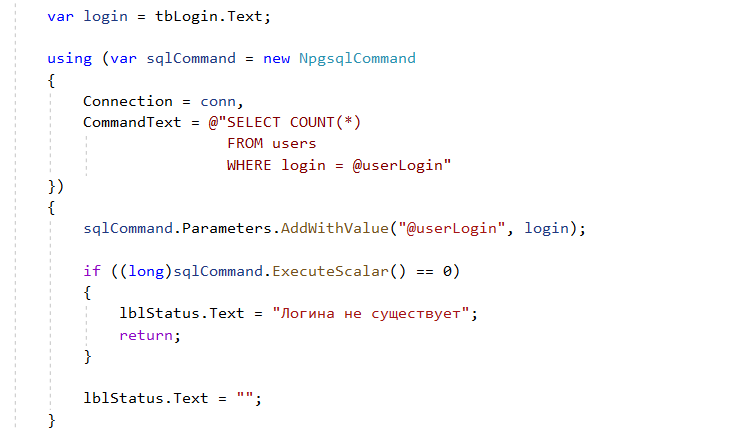
Создаем новый запрос на внесение новой записи, о пользователе используя для этого запрос вставки передавая в него параметры пароля и логина для защиты от инъекций в запрос (Внедрение в код произвольного sql кода для изменения, получения информации, удаления БД).

Затем делаем проверку об удачности внесения изменений в БД.



Для внесения данных о времени регистрации пользователя использовалось установка времени по умолчанию.

Теперь разберём работу программы аутентификации.



Получаем логин и при помощи запроса в БД, с использованием параметра для защиты от иньекций, получаем сведение о количестве таких же логинах в базе (по условию задачи запрос может вывести или 1 или 0). Если такого логина нет в базе, то отправляем соответствующее сообщение в информационное поле, если же такой логин нашелся, продолжаем работу.



Затем проверяем введённый пароль, формируя запрос на поиск пользователя с данным и логином и возвращение в приложение хэш строки его пароля.

При помощи встроенной функции в библиотеку BCrypt сравниваем пароль и хэш-строку (так как напрямую мы не можем сравнить, а хэш строка вычисляется каждый раз по-разному из за соли).

Если проверка прошла успешно выводим соответствующее сообщение, иначе сообщение о неверном пароле.

# 5 Тестирование программы

Тестирование программы можно разделить на 9 случаев:

1. Регистрация с пустыми полями.
2. Регистрация с существующим логином.
3. Регистрация с «простым» паролем.
4. Регистрация в соответствии со всеми правилами.
5. Аутентификация с пустыми полями.
6. Аутентификация с несуществующим логином.
7. Аутентификация с неправильным паролем.
8. Аутентификация в соответствии со всеми правилами.
9. Ввод sql инъекции в поле логина для регистрации.
10. Ввод sql инъекции в поле логин для аутентификации.

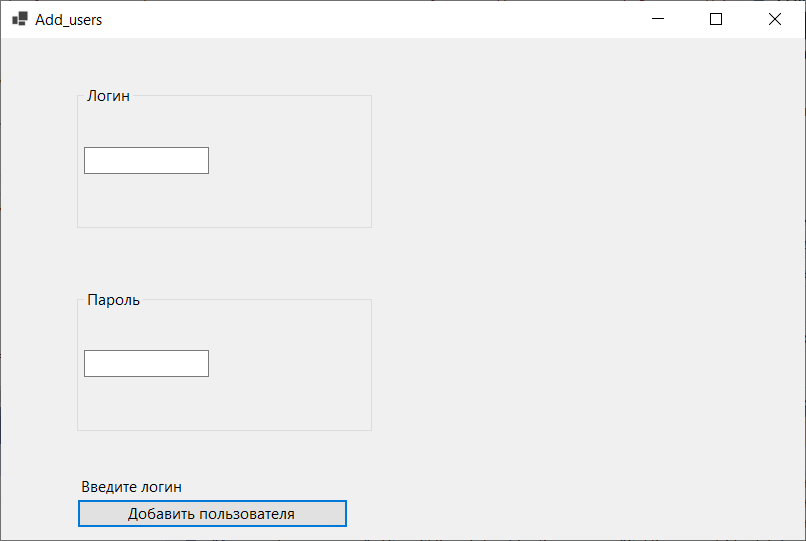
Тест 1(тип 1)

Входные данные:  
Логин:

Пароль:

Ожидаемый результат: сообщение об отсутствии логина

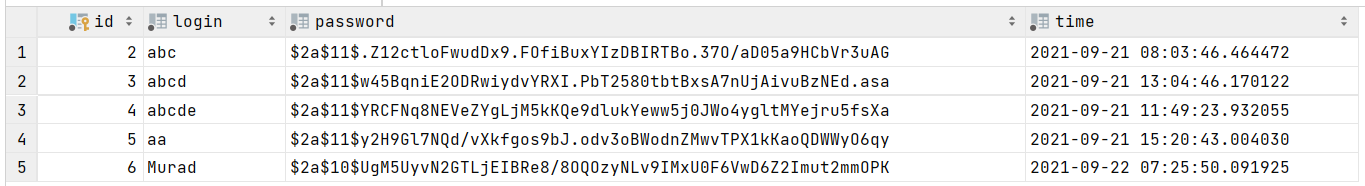
Выходной результат:



Тест 2(тип 2)  
Входные данные:

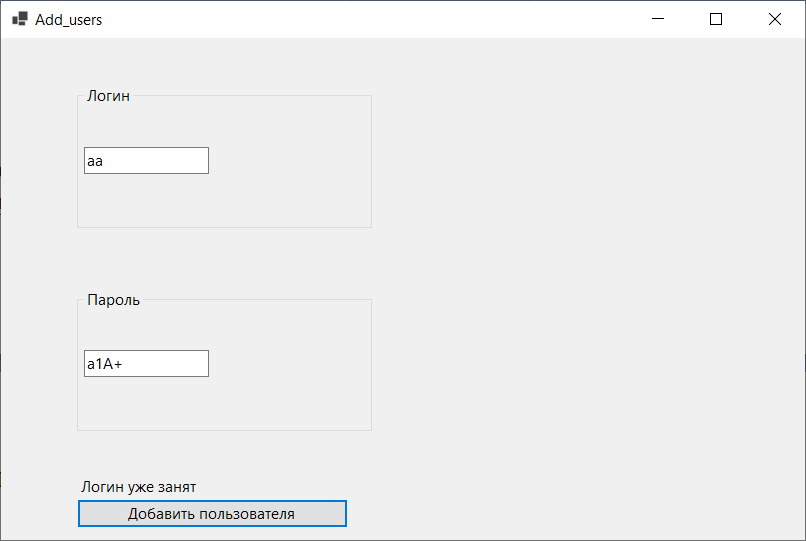
Логин:aa

Пароль:a1A+



Ожидаемый результат: сообщение о наличии такого логина в БД.

Выходной результат:

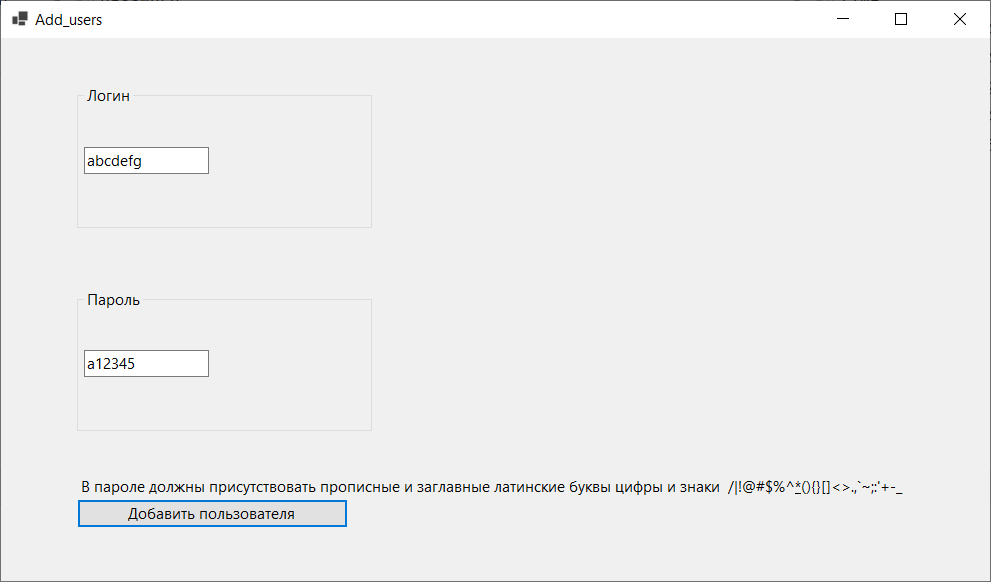


Тест 3(тип 3)  
Входные данные:

Логин:abcdefg

Пароль:a12345

Ожидаемый результат: сообщение о несоответствии введенного пароля правилам безопасности.

Выходной результат: 

Тест 4(тип 4)

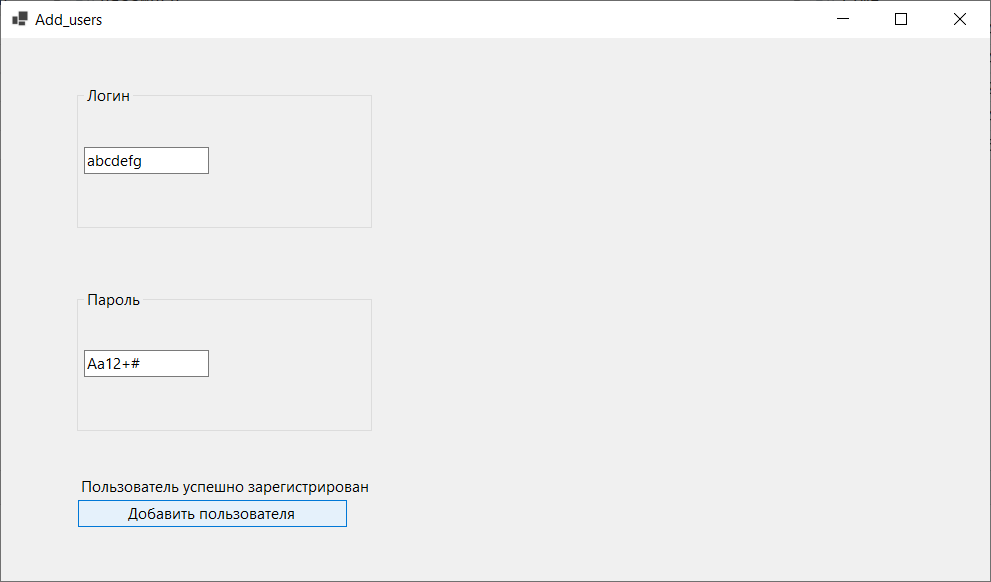
Входные данные:

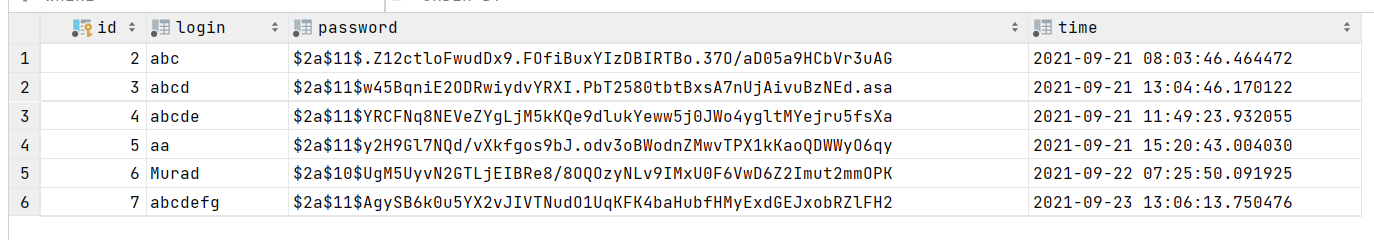
Логин:abcdefg

Пароль:Aa12+#

Ожидаемый результат: успешная регистрация пользователя.

Выходной результат:



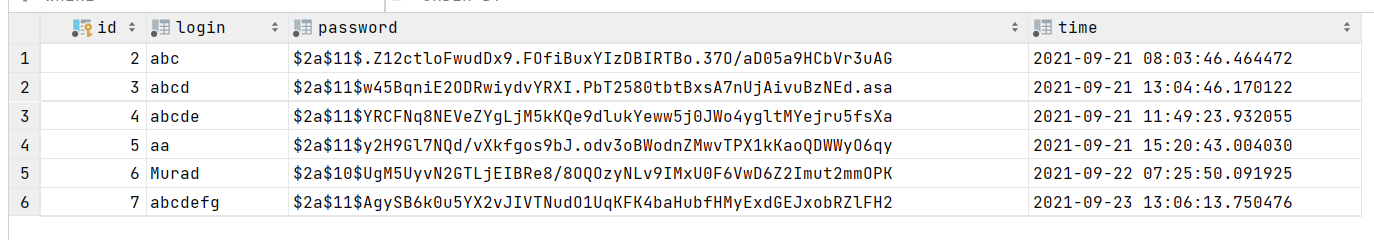


Тест 5(тип 5)

Входные данные:

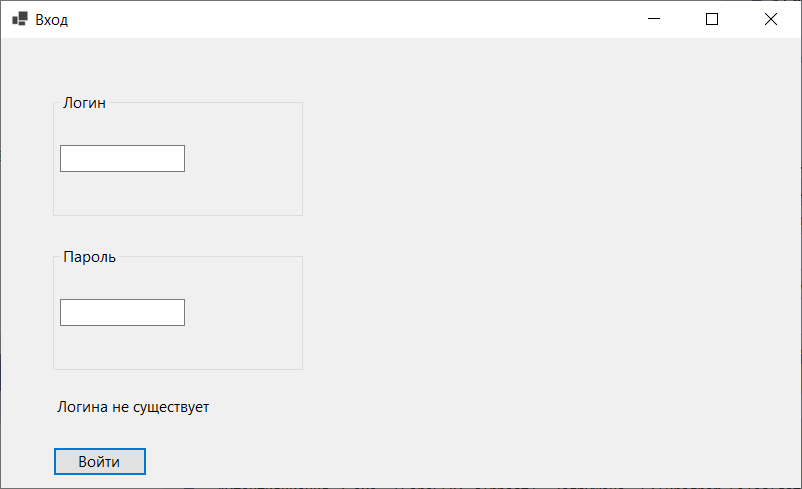
Логин:

Пароль:



Ожидаемый результат: сообщение об отсутствии данного логина в БД.

Выходной результат:

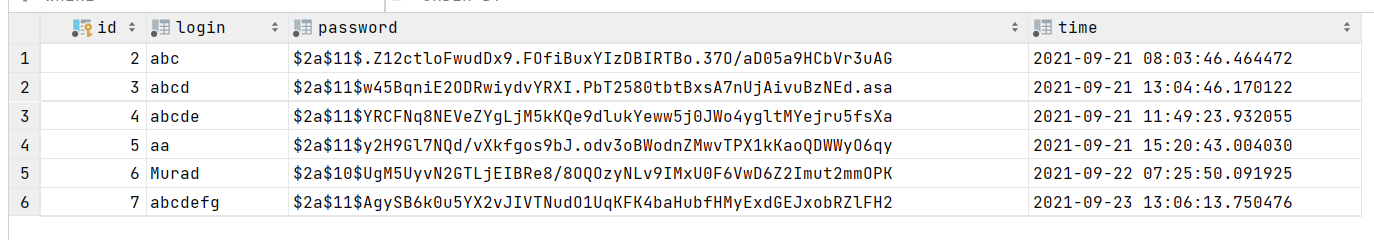


Тест 6(тип 6)

Входные данные:

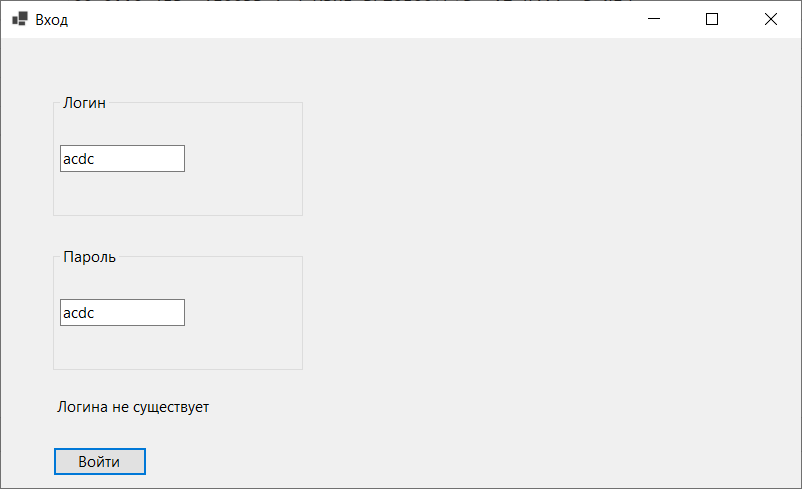
Логин:acdc

Пароль:acdc



Ожидаемый результат: сообщение об отсутствии данного логина в БД.

Выходной результат:

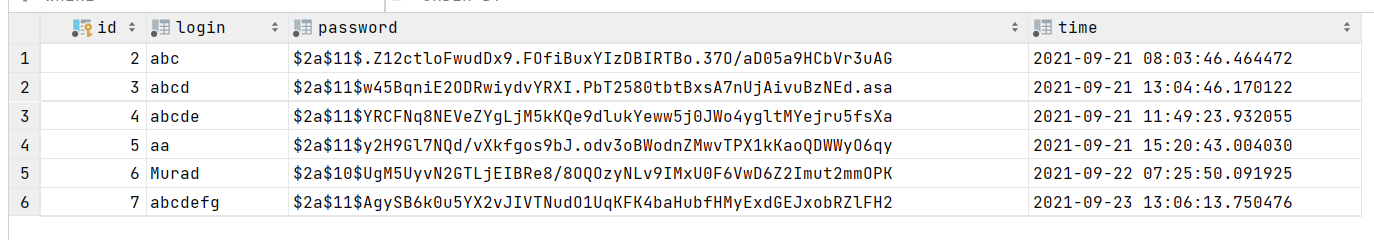


Тест 7(тип 7)

Входные данные:

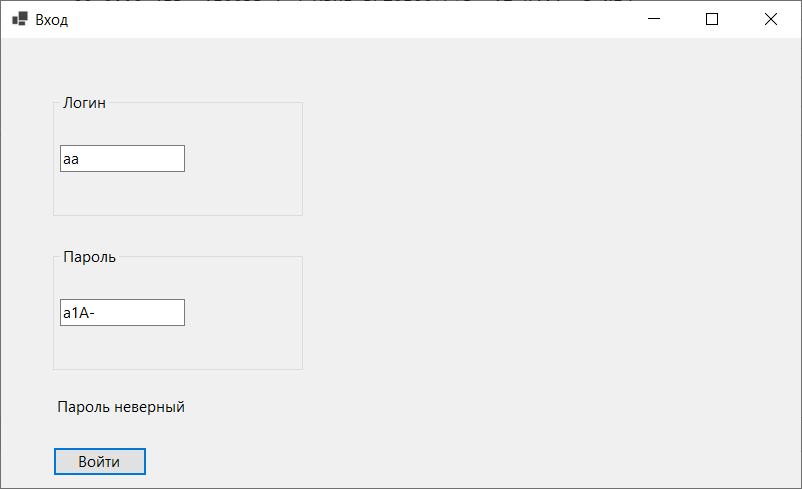
Логин:aa

Пароль:a1A-



Ожидаемый результат: сообщение о неправильно введённом пароле.

Выходной результат:

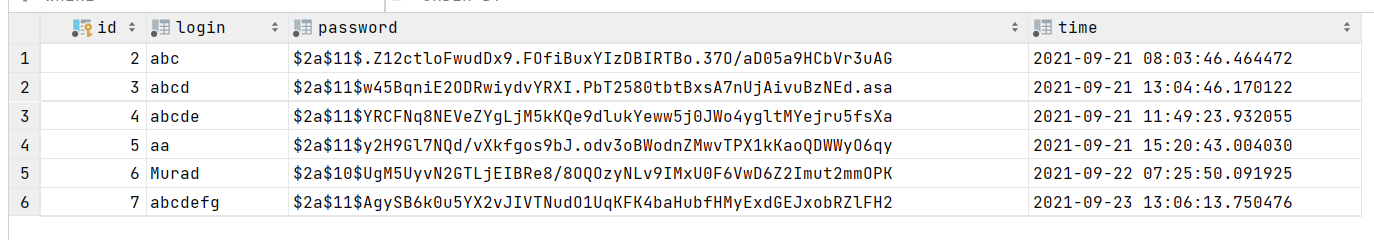


Тест 8(тип 8)

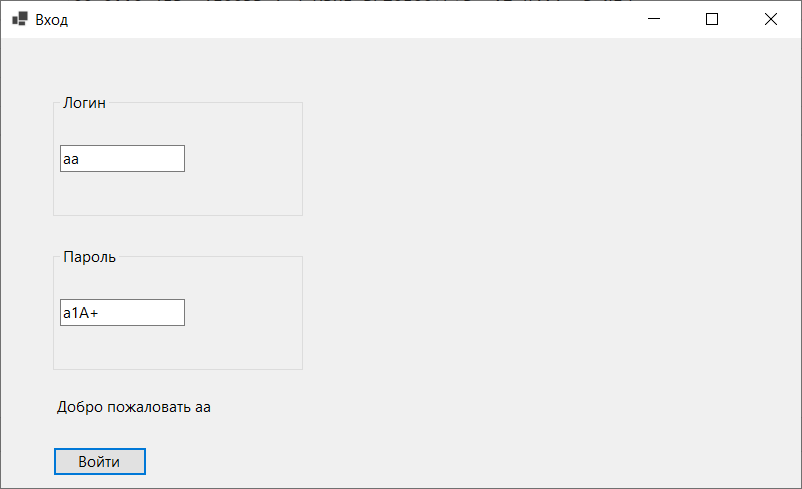
Входные данные:

Логин:aa

Пароль:a1A+



Ожидаемый результат: сообщение об успешной аутентификации.

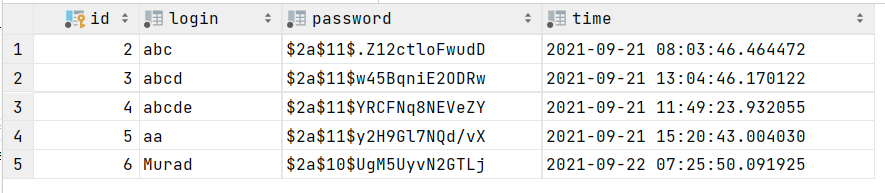
Выходной результат: 

Тест 9(тип 9)

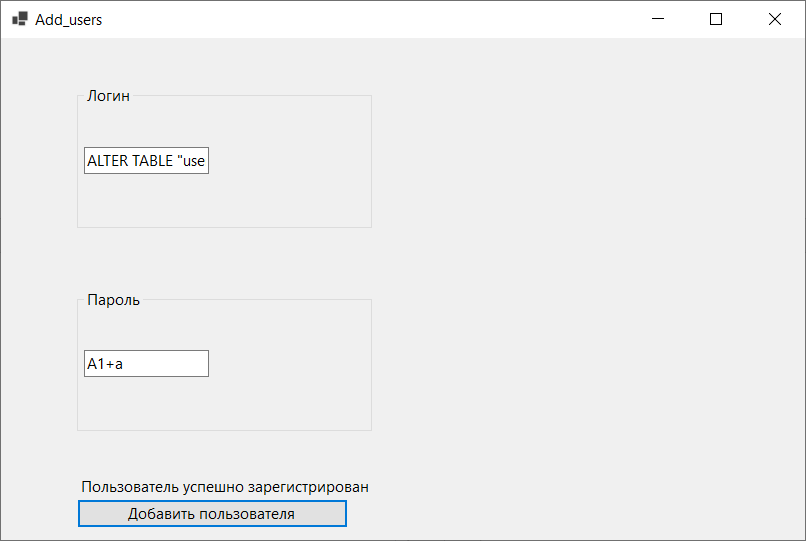
Входные данные:

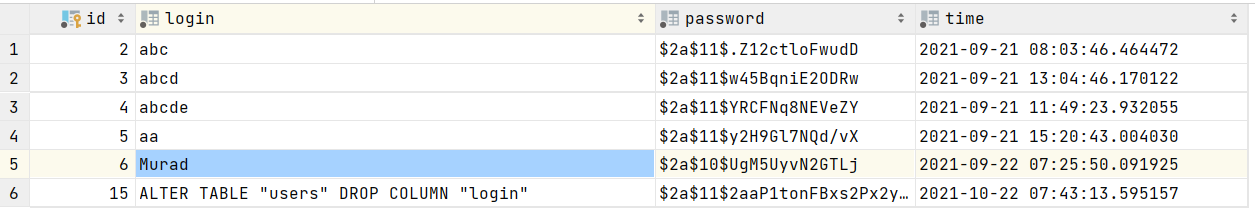
Логин: ALTER TABLE "users" DROP COLUMN "login"

Пароль:a1A+



Ожидаемый результат: сообщение об успешной регистрации

Выходной результат: 

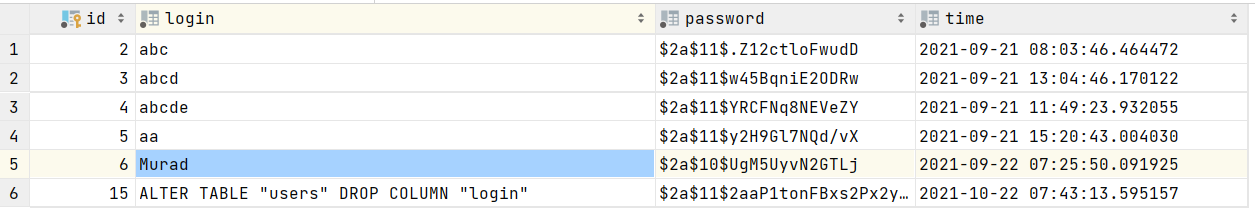


Тест 9(тип 9)

Входные данные:

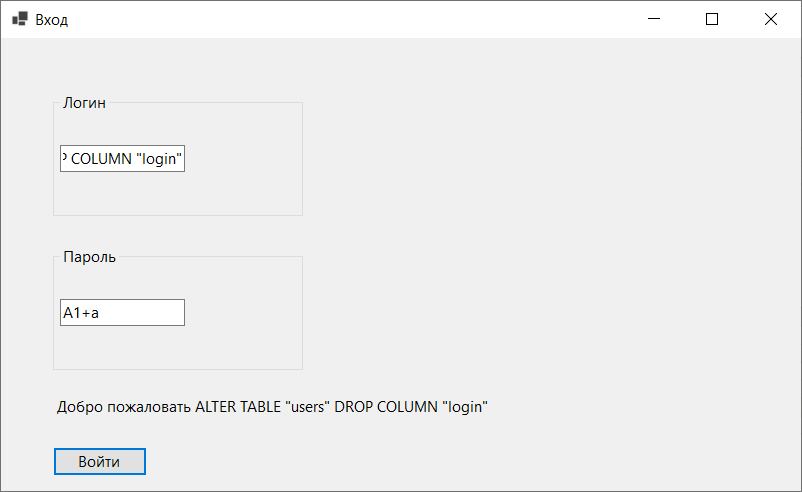
Логин: ALTER TABLE "users" DROP COLUMN "login"

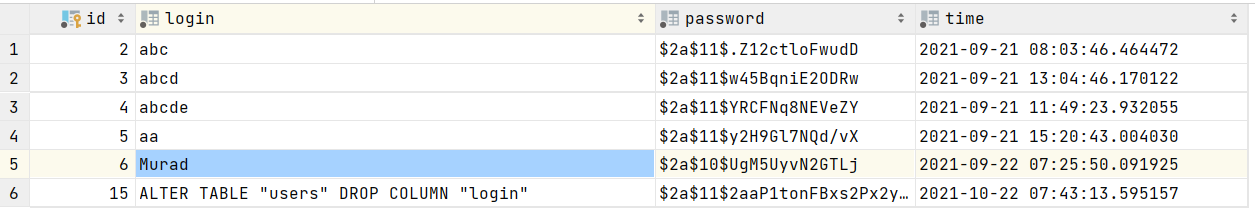
Пароль:a1A+



Ожидаемый результат: сообщение об успешной аутентификации.

Выходной результат:





Вывод: Программа успешно прошла тестирование, работает корректно.