

Міністерство освіти і науки України
Національний університет «Львівська політехніка»

Кафедра ЕОМ



Лабораторна робота №3

з дисципліни: «Інженерія програмного забезпечення»
на тему: “Система моніторингу рейтингу студентів та пошуку кращих.”

Виконав: ст. гр. КІ-32

Бабічок В.М.

Прийняв:

Цигилик Л.О.

Львів – 2020

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

Мета: 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.

Постановка задачі:

Розробка серверної частини.

Варіант 2 Система моніторингу рейтингу студентів та пошуку кращих.

Програма серверної частини

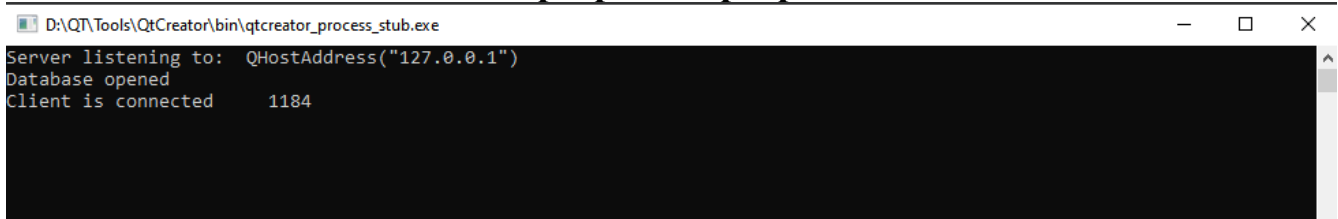


Рис 1. Вікно програми сервера

Опис API функцій сервера

Функція **LogProc**(**QTcpSocket*** socket) призначена для виконання процедури входу в систему.

Функція **RegProc**(**QTcpSocket*** socket) призначена для виконання процедури реєстрації нового користувача в системі.

Функція **GetCity**(**QTcpSocket*** socket) призначена для отримання списку областей.

Функція **GetUniver** (**QTcpSocket*** socket) призначена для отримання списку університетів.

Функція **GetSpecialty** (**QTcpSocket*** socket) призначена для отримання списку спеціальностей.

Функція **GetCourse** (**QTcpSocket*** socket) призначена для отримання списку курсів.

Функція **GetRating** (**QTcpSocket*** socket) призначена для отримання рейтингового списку студентів.

Функція **GroupSearch** (**QTcpSocket*** socket) призначена для отримання рейтингового списку студентів у потрібній користувачу групі.

Функція **NameSearch** (**QTcpSocket*** socket) призначена для отримання рейтингового списку студентів з потрібним користувачу ім'ям.

	NameCity
1	Вінницька область
2	Волинська область
3	Дніпропетровська область
4	Донецька область
5	Житомирська область
6	Закарпатська область
7	Запорізька область
8	Івано-Франківська область
9	Київська область
10	Кіровоградська область
11	Луганська область
12	Львівська область
13	Миколаївська область
14	Одеська область
15	Полтавська область
16	Рівненська область
17	Сумська область
18	Тернопільська область
19	Харківська область

Рис 2. База даних для областей

	NameUniver
1	Барська філія Глухівського національного педагогічн...
2	Вище професійне училище № 41 м. Тульчина
3	Вище професійне училище Львівського державного у...
4	Вінницька академія неперервної освіти
5	Вінницька філія Київського інституту бізнесу та техно...
6	Вінницька філія Університету сучасних знань
7	Вінницьке вище професійне училище сфери послуг
8	Вінницьке міжрегіональне вище професійне училище
9	Вінницький державний педагогічний університет імені...
10	Вінницький інститут конструювання одягу і підприємн...
11	Вінницький інститут Міжрегіональної Академії управл...
12	Вінницький кооперативний інститут
13	Вінницький навчально-науковий інститут економіки З...
14	Вінницький навчально-науковий інститут Університет...
15	Вінницький національний аграрний університет
16	Вінницький національний медичний університет ім. М....
17	Вінницький національний технічний університет
18	Вінницький соціально-економічний інститут Відкритог...
19	Вінницький торговельно-економічний інститут Київськ...

Рис 3. База даних університетів

	NameSpecialty	CourseNum	GroupNum	LastNameStudent	FirstNameStudent	RatingMark
1	105 Прикладна фізика та наноматеріали	1 курс	11	ГЕРМАН	Златодан	55.045
2	105 Прикладна фізика та наноматеріали	1 курс	11	ЄРЕМЕНКО	Добриня	42.733
3	105 Прикладна фізика та наноматеріали	1 курс	11	БІЛА	Андрій	13.303
4	105 Прикладна фізика та наноматеріали	1 курс	11	ДЕРКАЧ	Цвітан	87.778
5	105 Прикладна фізика та наноматеріали	1 курс	11	СЕМЕНОВА	Любозар	79.670
6	105 Прикладна фізика та наноматеріали	1 курс	11	КЛИМЧУК	Славута	13.904
7	105 Прикладна фізика та наноматеріали	1 курс	11	СТЕПАНЕНКО	Пор	73.964
8	105 Прикладна фізика та наноматеріали	1 курс	11	ГАЛУШКА	Магадар	24.114
9	105 Прикладна фізика та наноматеріали	1 курс	11	АНДРІЙЧУК	Святослав	64.054
10	105 Прикладна фізика та наноматеріали	1 курс	11	ЯРЕМЧУК	Дарій	62.252
11	105 Прикладна фізика та наноматеріали	1 курс	11	МАТВІЄНКО	Яромир	17.508
12	105 Прикладна фізика та наноматеріали	1 курс	11	ГОНЧАРЕНКО	Лук`ян	24.414
13	105 Прикладна фізика та наноматеріали	1 курс	11	ГУМЕНЮК	Сердитко	28.318
14	105 Прикладна фізика та наноматеріали	1 курс	11	ДОБРОВОЛЬСЬКА	Світогор	60.451
15	105 Прикладна фізика та наноматеріали	1 курс	11	КОГУТ	Надій	67.958
16	105 Прикладна фізика та наноматеріали	1 курс	11	ХОМЕНКО	Монсеррат	22.012
17	105 Прикладна фізика та наноматеріали	1 курс	11	КОЗАЧЕНКО	Олекса	53.243
18	105 Прикладна фізика та наноматеріали	1 курс	11	СКЛЯР	Собіслав	71.562
19	105 Прикладна фізика та наноматеріали	1 курс	11	ІВАЩЕНКО	Любомудр	62.553

Рис 4. База даних для рейтингових списків

	UserLog	UserPass
1	12345	12345

Рис 5. База даних зареєстрованих користувачів

Приклад коду для запису у базу даних

```
void Server::RegProc(QTcpSocket *socket )
{
    // find log in database
    query->prepare("select * from UserInfo where UserLog = :log");
    query->bindValue(":log", obj->value("log").toString());

    if (query->exec())
    {
        // if qry is not empty (if log exist)
        if (query->next())
        {
            // send user already exist password respond to client
            Data="{\"operation\":\"register\", \"resp\":\"bad\", \"err\":\"User already exist\"}";
            DataSend(socket, Data);
            socket->waitForBytesWritten(1500);
        }
        // qry is empty so login does not exist
        else
        {
            // insert login and password to database
            query->prepare("insert into UserInfo (UserLog, UserPass) values (:log, :pass)");
            query->bindValue(":log", obj->value("log").toString());
            query->bindValue(":pass", obj->value("pass").toString());

            // if registration successful
            if (query->exec())
            {
                // send good respond
                Data="{\"operation\":\"register\", \"resp\":\"ok\"}";
                DataSend(socket, Data);
                socket->waitForBytesWritten(1500);
            }
            else
            {
                // set smt went wrong respond
                Data="{\"operation\":\"register\", \"resp\":\"bad\", \"err\":\"Something went wrong when transferring data. Please check your internet connection\"}";
                DataSend(socket, Data);
                socket->waitForBytesWritten(1500);
            }
        }
    }
}
else
```

```

{
    // handle bad query execution
    Data="{\"operation\": \"register\", \"resp\": \"bad\", \"err\": \"Something went wrong when transferring data. Please check your internet connection\"}";
    DataSend(socket, Data);
    socket->waitForBytesWritten(1500);
}
}

```

Приклад коду для зчитування з бази даних

```

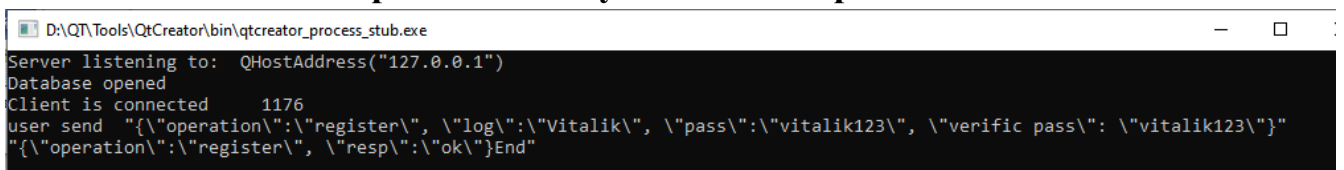
void Server::LogProc(QTcpSocket* socket)
{
    // find log in database
    query->prepare("select * from UserInfo where UserLog = :log");
    query->bindValue(":log", obj->value("log").toString());

    // if qry is not empty (if log exist)
    if (query->exec())
    {
        if (query->next())
        {
            // if password is correct
            if (query->value(1).toString() == obj->value("pass").toString())
            {
                // return validation is ok respond to client
                Data="{\"operation\": \"login\", \"resp\": \"ok\"}";
                DataSend(socket, Data);
                socket->waitForBytesWritten(1500);

            }
            // if password is not correct
            else
            {
                // send invalid password respond to client
                Data="{\"operation\": \"login\", \"resp\": \"bad\", \"err\": \"Invalid password\"}";
                DataSend(socket, Data);
                socket->waitForBytesWritten(1500);
            }
        }
        // qry is empty so login does not exist
        else
        {
            // send no such login respond to client
            Data="{\"operation\": \"login\", \"resp\": \"bad\", \"err\": \"Login doesn't exist\"}";
            DataSend(socket, Data);
            socket->waitForBytesWritten(1500);
        }
    }
    else
    {
        // handle bad query execution
        Data="{\"operation\": \"login\", \"resp\": \"bad\", \"err\": \"Something went wrong when transferring data. Please check your internet connection\"}";
        DataSend(socket, Data);
        socket->waitForBytesWritten(1500);
    }
}
}

```

Приклад пакету даних що передається



```

D:\QT\Tools\QtCreator\bin\qtcreator_process_stub.exe
Server listening to: QHostAddress("127.0.0.1")
Database opened
Client is connected    1176
user send  "{\"operation\": \"register\", \"log\": \"Vitalik\", \"pass\": \"vitalik123\", \"verific pass\": \"vitalik123\"}"
"\"operation\": \"register\", \"resp\": \"ok\"}End"

```

Усі дані передаються у форматі JSON. Це спрощує їх розшифрування, полегшує процес виявлення помилок та збільшує надійність передачі даних.

Висновок

Під час виконання цієї лабораторної роботи я створив програму-сервер, яка обробляє та виконує запити клієнтської частини шляхом взаємодії з базою даних.