SAMARQAND DAVLAT UNIVERSITETI

RAQAMLI TEXNOLOGIYALAR FAKULTETI

203-GURUH TALABASI ESANOV OTABEKNING

DASTURLASH ASOSLARI FANIDAN

KURS ISHI

Mavzu: MVC texnologiyasiga asoslanib QT muhitida “Virtual Darslar” platformasini tuzish

Tekshirdi: MELIYEV FARHOD

SAMARQAND 2021

Reja:

1. MVC – Model view controller haqida
2. Loyihamiz tuzilmasi
3. Loyiha kodlari
4. Loyiha natijasi
5. Xulosa
6. Foydalanilgan adabiyotlar

1. MVC – Model view controller haqida

Tarix

Grafik foydalanuvchi interfeysini dastlabki ishlab chiqishda muhim tushunchalardan biri bo'lgan,MVC dasturiy ta'minot konstruksiyalarini ularning vazifalariga ko'ra amalga oshirish va ularni tavsiflashdagi ilk yondashuvlardan biri hisoblanadi.

Trygve Reenskaug, 1970-yil Xerox Palo Alto Tadqiqotlar Markazi(PARC)da SmallTalk-79 dasturlash tilida MVC'ni tanishtirdi. 1980-yillarda Jim Altthoff va boshqalar Smalltalk-80 klass kutubxonasi uchun MVC'ning birinchi versiyasini ishlab chiqishdi. Faqat keyinroq, 1988-yilda "The Journal of Object Technology"(Obyekt texnologiyalar jurnali) da MVC, umumiy konseptsiya sifatida maqola chop etildi.

MVC andozasi(ingliz tilida pattern, rus tilida шаблон) keyinchalik riovjlandi, shuningdek MVC'ni turli xil kontekstlarga moslashtiradigan iyerarxik model-view-controller(HMVC), model-view-adapter(MVA), model-view-presenter(MVP), model-view-viewmodel(MVVM) va boshqa turli xil variantlari paydo bo'ldi.

MVC'dan foydalanishdan maqsad

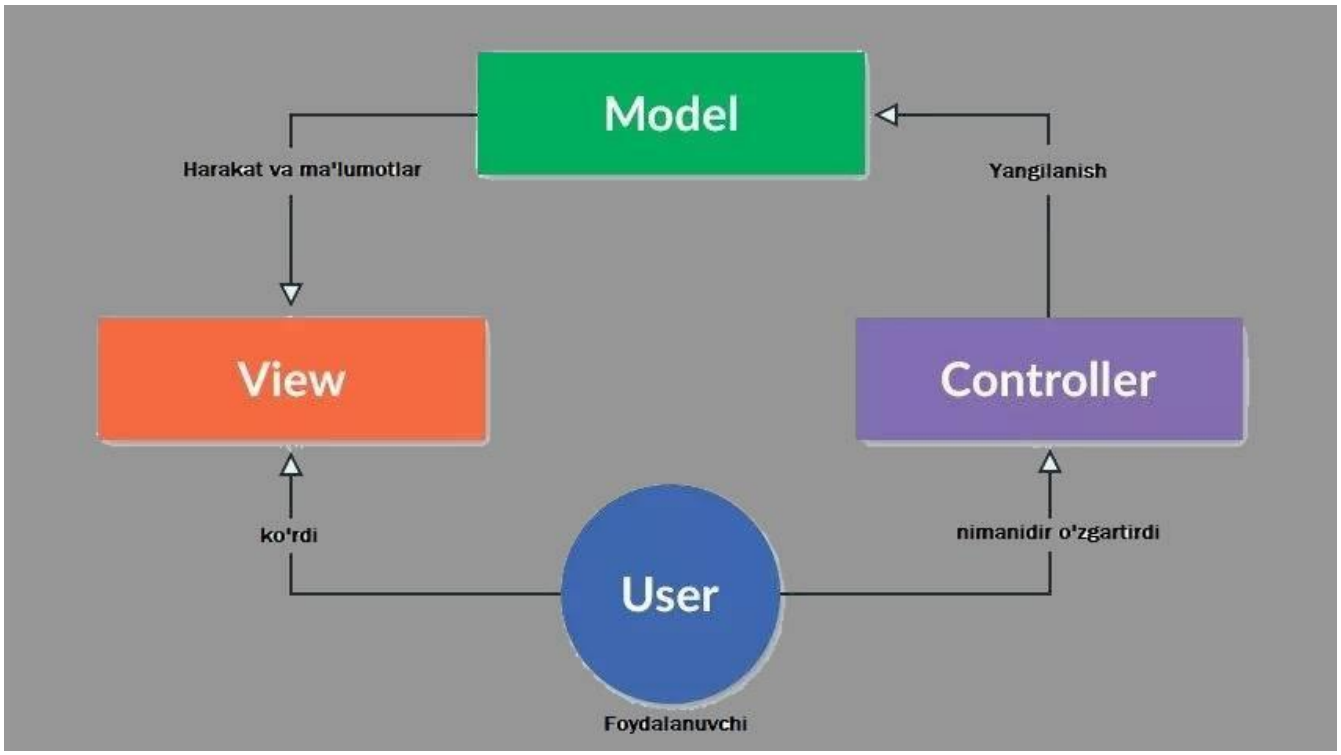
MVC, ilovaning turli xil komponentalarini ajratib ishlatishni talab qilgani uchun, dasturchilar bir-biriga halaqit yoki ishini bo'lib turmasdan, turli xil komponentlari ustida paralell ravishda ishlash qobiliyatiga ega bo'lishiadi. Masalan, bir jamoa o'zining dasturchilarini front-end va back-end qismlari uchun ma'sul qilib qo'yishlari mumkin. Back-end dasturchilar ma'lumot strukturasini tuzishsa va aksincha front-end dasturchilar ma'lumot strukturasi mavjud bo'lgandan keyingi ilovaning ko'rinishini tuzishlari mumkin bo'ladi.

Koddan qayta foydalanish mumkinligi

MVC arxitekturasi tamoyillari qayta foydalanib bo'ladigan kod yozishni ta'minlaydi.

Afzalliklari

- Bir vaqtni o'zida ishlab chiqish
- Bitta o'zgartirish butun ilovaga ta'sir qilmaydi
- Bitta model uchun bir va undan ortiq shablonlar(views)
- Katta o'lchamdagi veb-ilovalar uchun ideal tanlov
- Hamjihatlikda ishlash



1-rasm. MVC ishlash prinsipi

1. Model

Model - ilovadagi ma'lumotlarni boshqarishga ma'sul. U view'dan kelgan so'rovga hamda o'zini yangilash uchun controller'dagi ko'rsatmaga javob beradi.

2. View

View - ilova ichida obyektlarni namoyish etishni anglatadi. Aniqroq aytadigan bo'lsak, u foydalanuvchi ko'radigan har qanday komponenta hisoblanadi. Oddiygina shablon deb ataymiz.

3. Controller

Controller - ikkala model va views'dagi o'zgarishlarni yangilaydi. U kiritish(input) qabul qiladi va tegishli update(o'zgarish, yangilanish)ni bajaradi. Misol uchun, controller - view orqali kiritilgan ma'lumotni qabul qiladi va keyin model yordamida ma'lumotni qayta ishlab, yana qayta view'ga jo'natadi.

1. Loyihamiz tuzilmasi

```
39 lines (32 sloc)  587 Bytes

QT       += core gui webkit webkitwidgets
QT       +=sql

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

TARGET = Lifedu-x
TEMPLATE = app

SOURCES += main.cpp\
        mainwindow.cpp \
    home.cpp \
    login.cpp \
    about.cpp \
    kurs.cpp \
    dars.cpp \
    admin.cpp

HEADERS  += mainwindow.h \
    home.h \
    login.h \
    db.h \
    about.h \
    kurs.h \
    dars.h \
    admin.h

FORMS    += mainwindow.ui \
    home.ui \
    login.ui \
    about.ui \
    kurs.ui \
    dars.ui \
    admin.ui

DISTFILES += \
    ../../Desktop/l.png

RESOURCES += \
    assets/files.qrc
```

Webview va Sql kutubxonalarini qo'shish

Sources bo'limi:
MVC dagi Controllerlar hisoblanadi

Bu yerda DB.h faylimizda 2 ta model bor:
- Session Class
-DBHelper Class

FORMS ui - fayllari:
VIEW hisoblanadi

2- rasm. Loyiha tuzilmasi

Loyiha kodlari

- Tizim uchun 2 ta model tuzilgan. Modellar db.h faylida saqlanadi:

  Db.h fayli githubda: https://github.com/EsanovOtabek/qt-lifedu/blob/main/db.h

  1. Session Class – bu klassda dasturni ochgan foydalanuvchini local ma'lumotlari saqlanadigan baza bilan ishlaydi (3-rasm).
  2. DBHelper Class – bu klassda dasturning ma'lumotlar bazasi bilan ishlaydi (4-rasm). Ushbu class C++ dasturlash tilining STL (Standart Template Library) kutubxonasidagi <map> sinfining imkoniyatlaridan foydlanilgan

```
class Session
{
public:
    QString id;
    QString fio;
    QString email;
    QString echo;

private:

    QSqlDatabase db;
    bool insert(){
        QSqlQuery query;
        QString T=QDateTime::currentDateTime().toString("d.MM.yyyy hh:mm:ss");
        QString sql="INSERT INTO session (user_id,fio,email,utime) VALUES("+
                this->id+",'"+this->fio+"','"+this->email+"','"+T+"')";
        query.prepare(sql);
        return query.exec();
    }
    bool delet(){
        QSqlQuery query;
        QString sql="DELETE FROM session";
        query.prepare(sql);
        return query.exec();
    }
    bool count(){
        QSqlQuery query;
        QString sql="SELECT * FROM session";
        query.prepare(sql);
        query.exec();
        while (query.next()){
            this->id=query.value(1).toString();
            this->fio=query.value(2).toString();
            this->email=query.value(3).toString();
            return 1;
        }
        return 0;
    }

public:
    Session(){
        this->db = QSqlDatabase::addDatabase("QSQLITE");
        this->db.setDatabaseName("C:/Users/beoo/Documents/LifeduTest/assets/user.db");
        if(db.open()){
            this->echo="DB connect!";
        }
        else{
            this->echo="DB connection error!";
        }
    }

    bool login(){
        return this->insert();
    }
    bool isActive(){
        return this->count();
    }
    bool logout(){
        return this->delet();
    }
};
```

3-rasm. Session sinfi

```
class DBHelper{

    QSqlDatabase db;
    QString tablename;
    void table_key(map <int,QString> &key, QString tablename){
        key[0]="id";
        if(tablename=="users"){
            key[1]="fio";
            key[2]="email";
            key[3]="password";
        }else if(tablename=="courses"){
            key[1]="name";
            key[2]="title";
            key[3]="image";
        }else if(tablename=="lessons"){
            key[1]="course_id";
            key[2]="title";
            key[3]="content";
            key[4]="video";
        }else if(tablename=="user_courses"){
            key[1]="user_id";
            key[2]="course_id";
            key[3]="vaqt";
        }
    }

public:
    QString echo;
    DBHelper(){
        this->db = QSqlDatabase::addDatabase("QSQLITE");
        this-
>db.setDatabaseName("C:/Users/beoo/Documents/LifeduTest/assets/lifedu.db");
        if(db.open()){
            this->echo="DB connect!";
        }
        else{
            this->echo="DB connection error!";
        }
    }

    int count(QString sql){
        QSqlQuery query;
        query.prepare(sql);
        query.exec();
        int k=0;
        while(query.next()){
            k++;
        }
        return k;
    }

    void select(QString sql, map< int, map<QString,QString> > &mp, QString tablename){
        int k=0;
        map<int,QString> key;
        table_key(key,tablename);
        QSqlQuery query;
        query.prepare(sql);
        query.exec();
        while(query.next()){
            for(int i=0;i<key.size();i++){
                mp[k][key[i]]=query.value(i).toString();
            }
            k++;
        }
    }

    bool inupde(QString sql){
        QSqlQuery query;
        query.prepare(sql);
        return query.exec();
    }
};
```

4-rasm. DBHelper sinfi

- **.h kutubxona fayllar**

about.h - https://github.com/EsanovOtabek/qt-lifedu/blob/main/about.h

admin.h - https://github.com/EsanovOtabek/qt-lifedu/blob/main/admin.h

dars.h - https://github.com/EsanovOtabek/qt-lifedu/blob/main/dars.h

db.h - https://github.com/EsanovOtabek/qt-lifedu/blob/main/db.h

home.h - https://github.com/EsanovOtabek/qt-lifedu/blob/main/home.h

kurs.h - https://github.com/EsanovOtabek/qt-lifedu/blob/main/kurs.h

login.h - https://github.com/EsanovOtabek/qt-lifedu/blob/main/login.h

mainwindow.h - https://github.com/EsanovOtabek/qt-lifedu/blob/main/mainwindow.h

- **.cpp source files. Controllerlar**

about.cpp - https://github.com/EsanovOtabek/qt-lifedu/blob/main/about.cpp

admin.cpp - https://github.com/EsanovOtabek/qt-lifedu/blob/main/admin.cpp

dars.cpp - https://github.com/EsanovOtabek/qt-lifedu/blob/main/dars.cpp

main.cpp - https://github.com/EsanovOtabek/qt-lifedu/blob/main/db.cpp

home.cpp - https://github.com/EsanovOtabek/qt-lifedu/blob/main/home.cpp

kurs.cpp - https://github.com/EsanovOtabek/qt-lifedu/blob/main/kurs.cpp

login.cpp - https://github.com/EsanovOtabek/qt-lifedu/blob/main/login.cpp

mainwindow.cpp - https://github.com/EsanovOtabek/qt-lifedu/blob/main/mainwindow.cpp

- **.ui View files. VIEW – Ko'rinishlar**

about.ui - https://github.com/EsanovOtabek/qt-lifedu/blob/main/about.ui

admin.ui - https://github.com/EsanovOtabek/qt-lifedu/blob/main/admin.ui

dars.ui - https://github.com/EsanovOtabek/qt-lifedu/blob/main/dars.ui

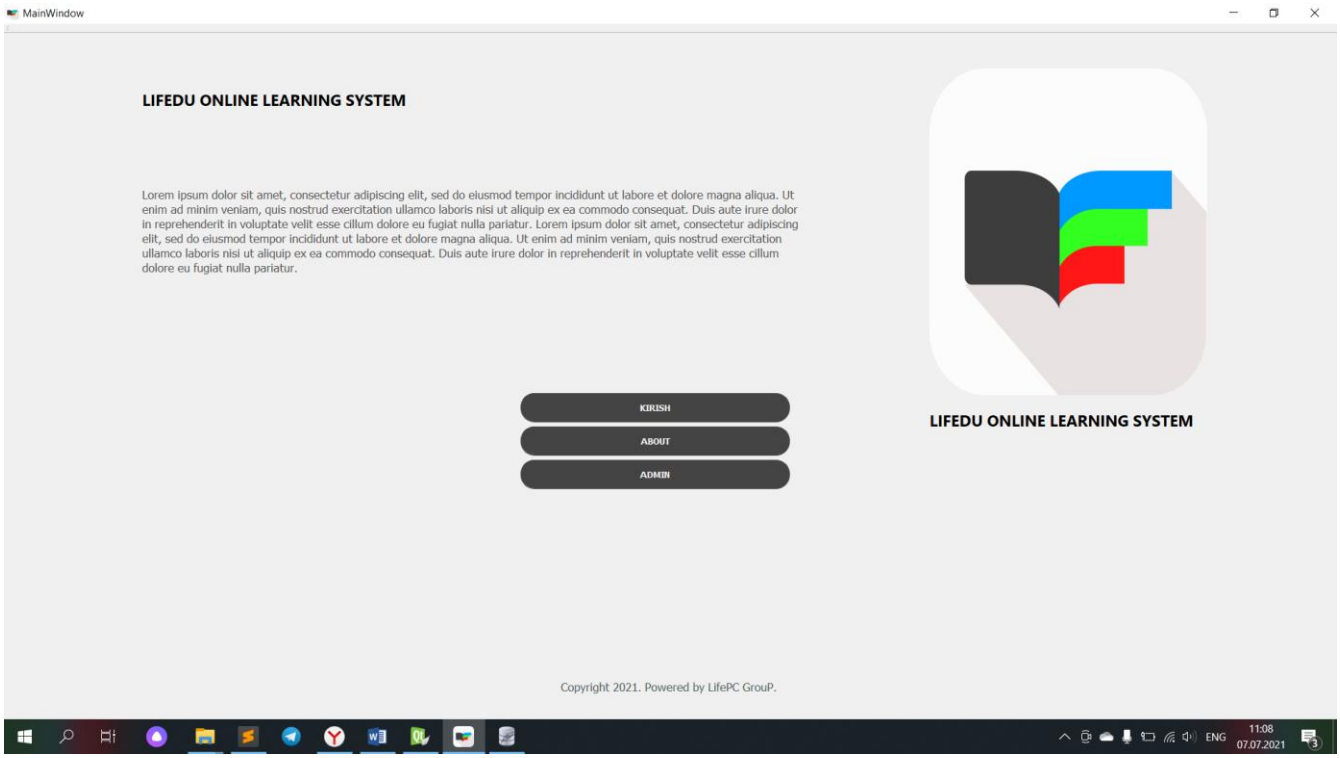
home.ui - https://github.com/EsanovOtabek/qt-lifedu/blob/main/home.ui

kurs.ui - https://github.com/EsanovOtabek/qt-lifedu/blob/main/kurs.ui

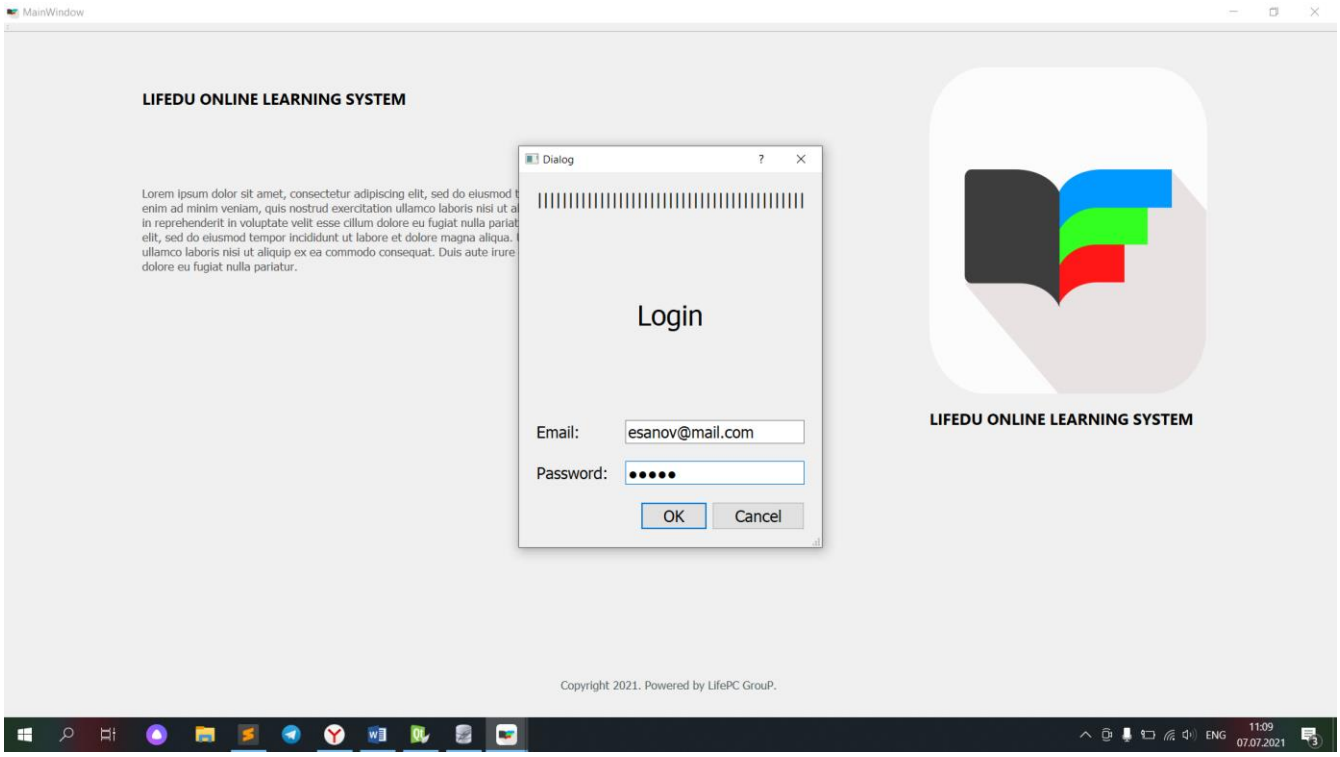
login.ui - https://github.com/EsanovOtabek/qt-lifedu/blob/main/login.ui

mainwindow.ui - https://github.com/EsanovOtabek/qt-lifedu/blob/main/mainwindow.ui

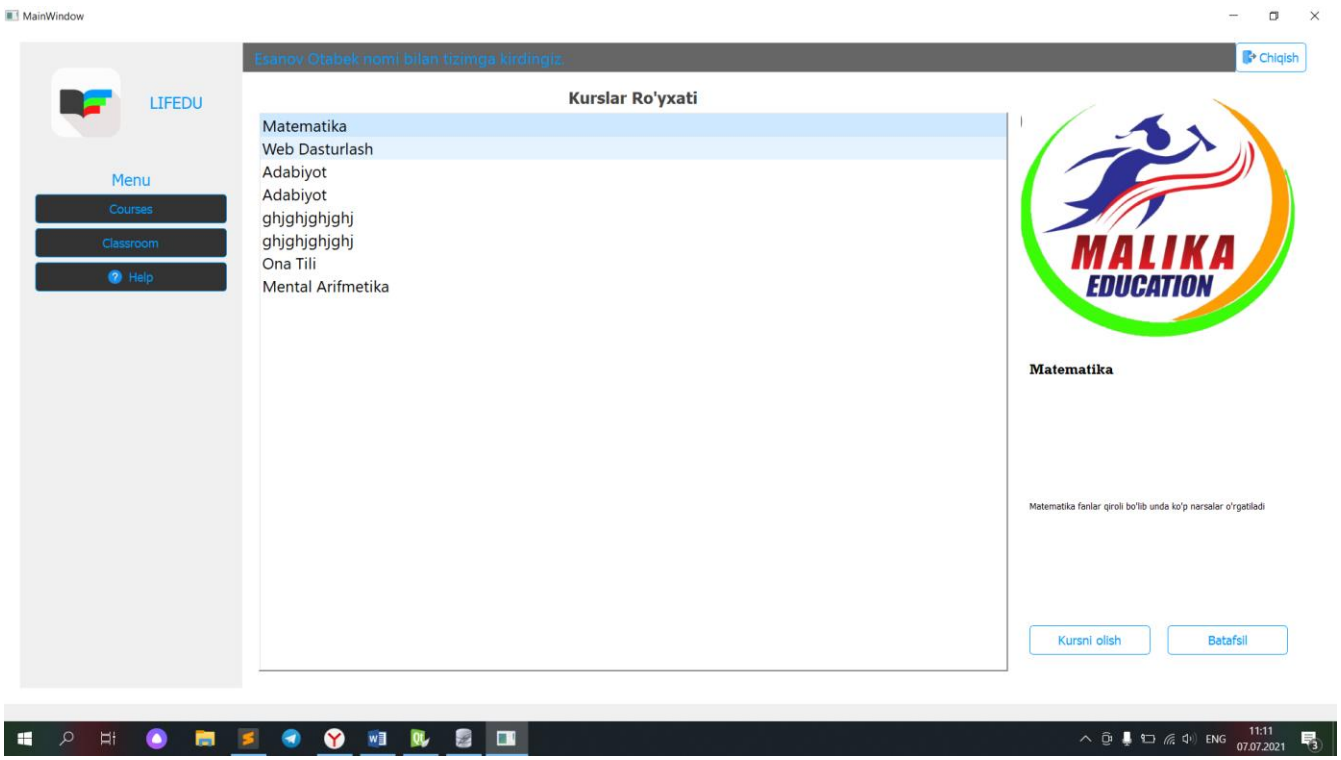
Loyiha natijasi



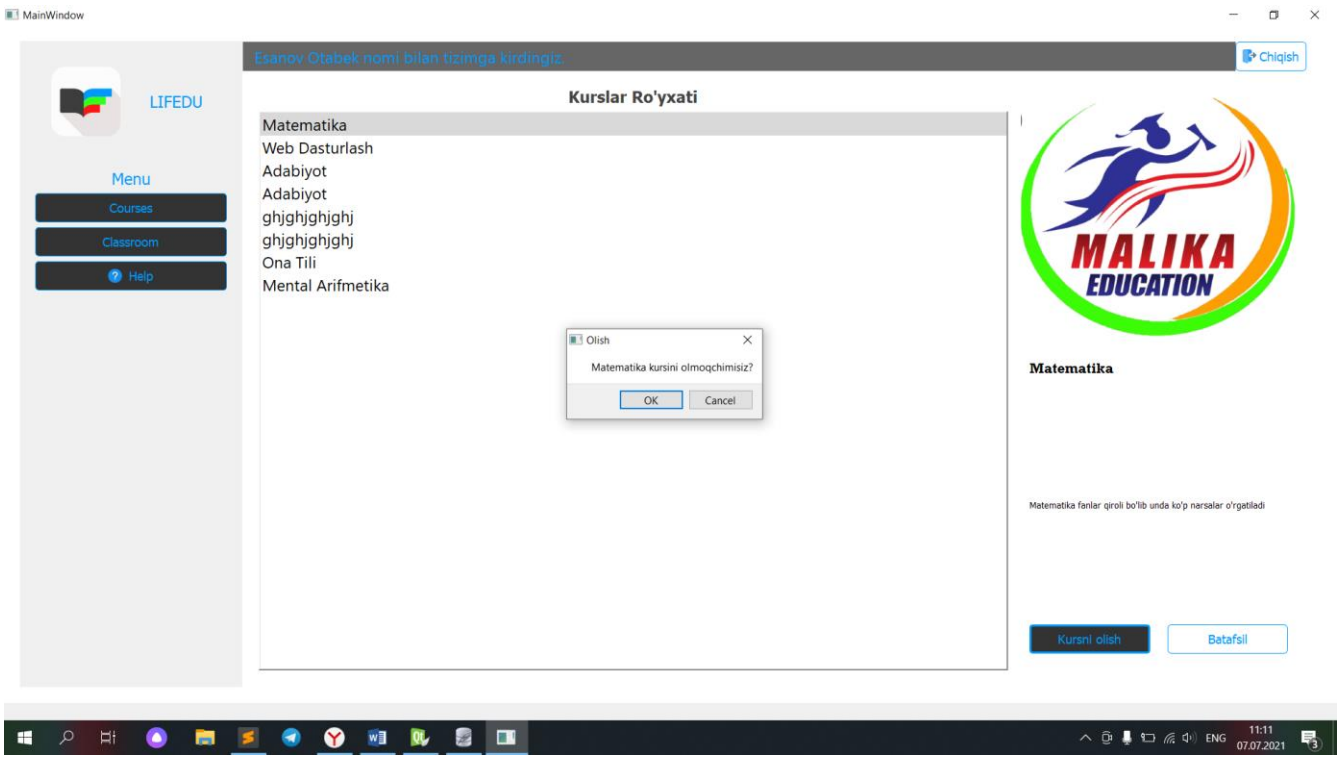
5-rasm. Bosh sahifa



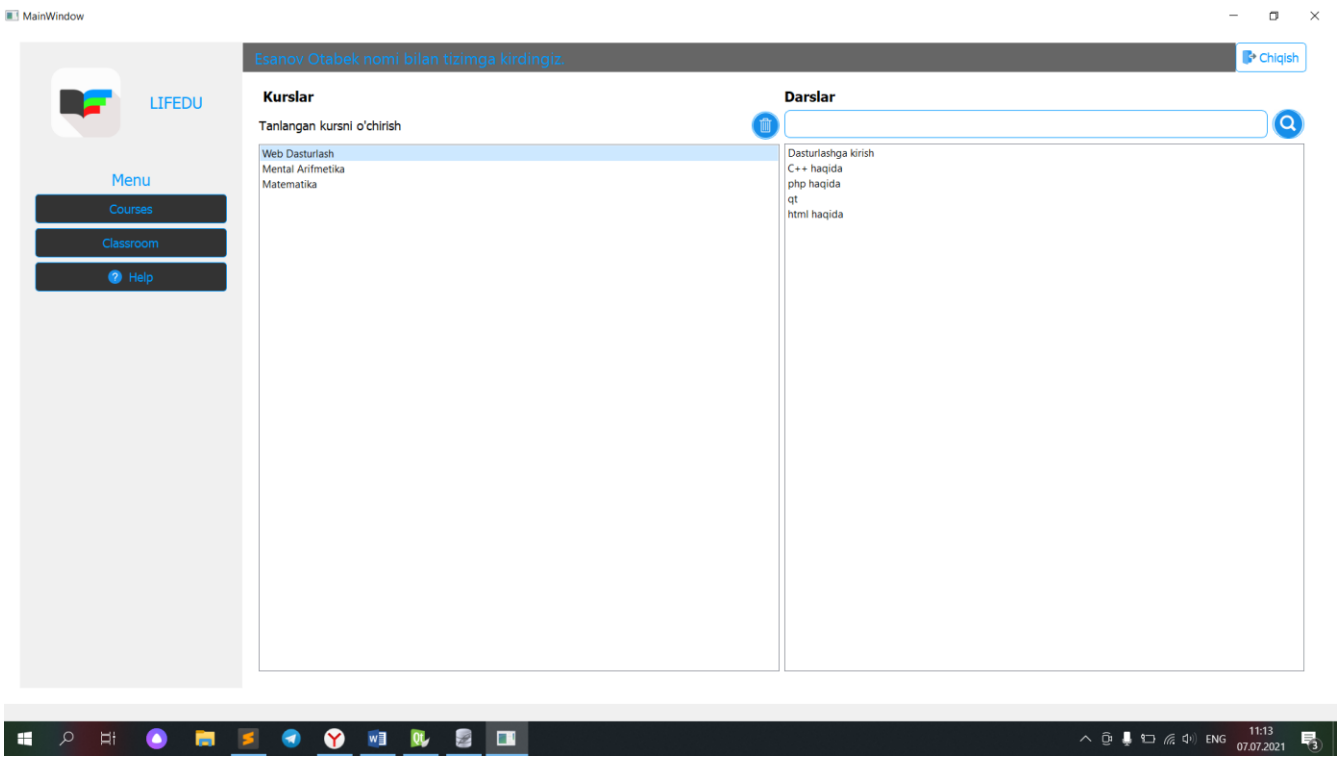
6-rasm. Login



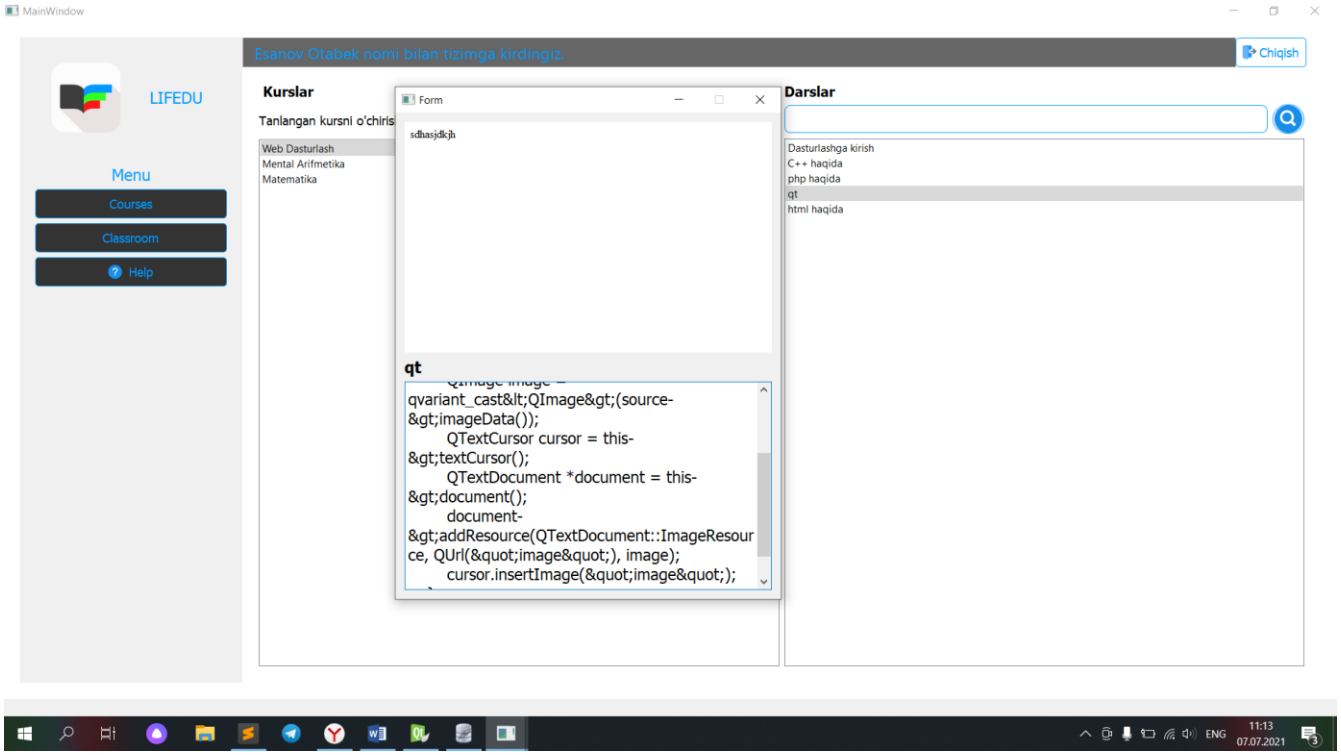
7-rasm. Kurslar



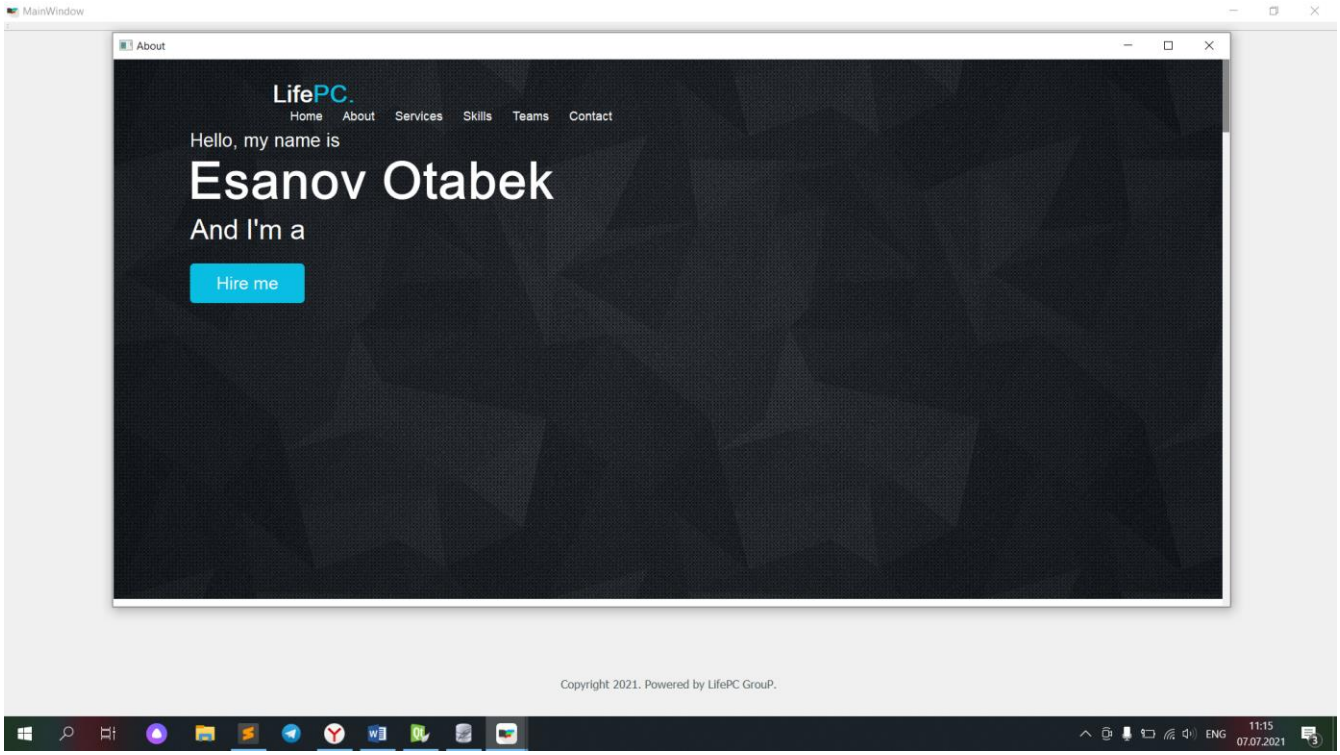
8-rasm. Kursni olish.



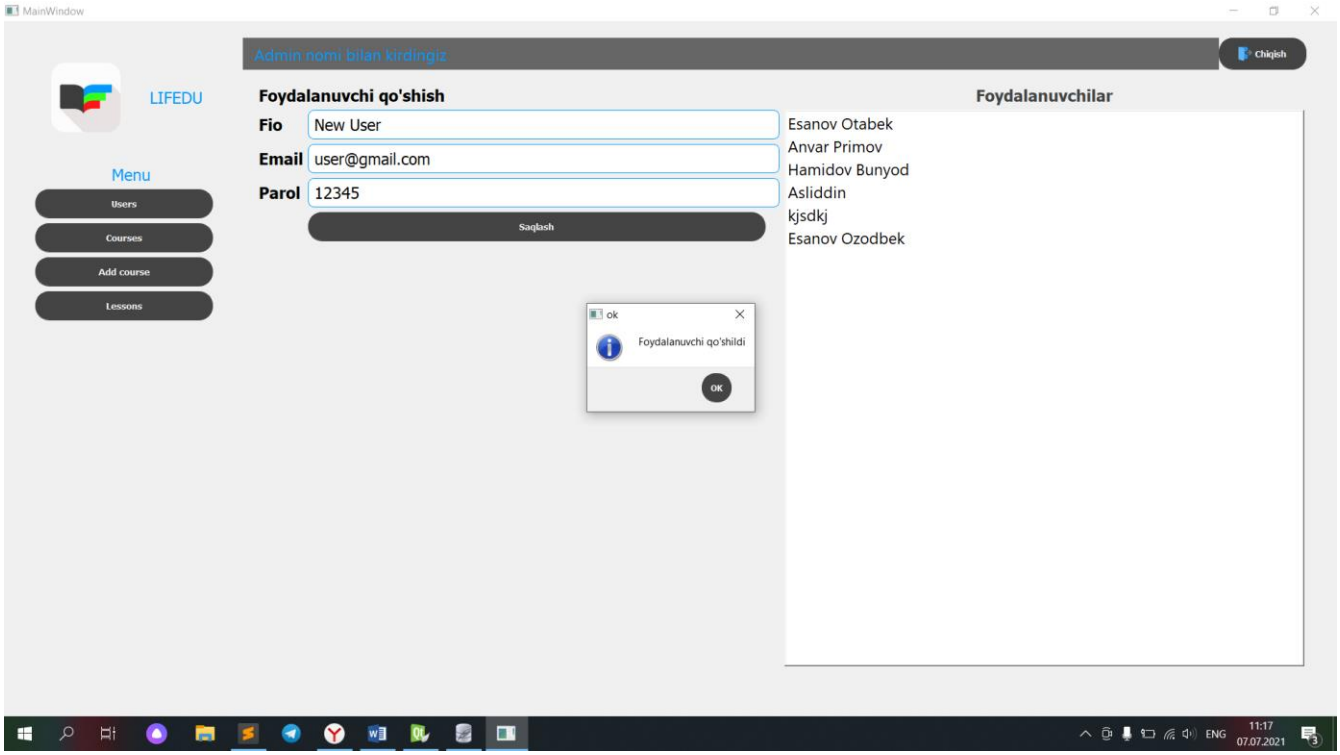
9-rasm. Sinfxona



10-rasm. Darsni ustiga 2 marta bossa dars ochiladi.

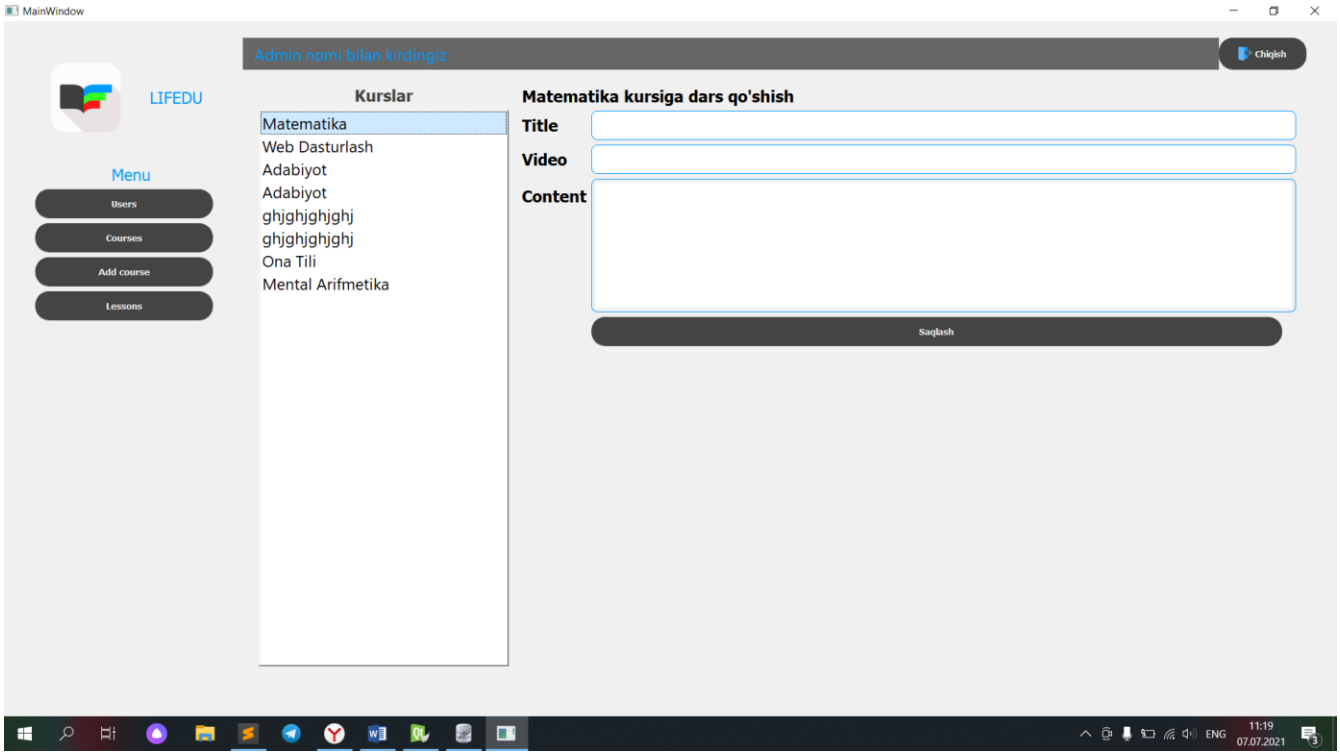


11-rasm. About sahifasi. Buning uchun Webview ishlatilgan

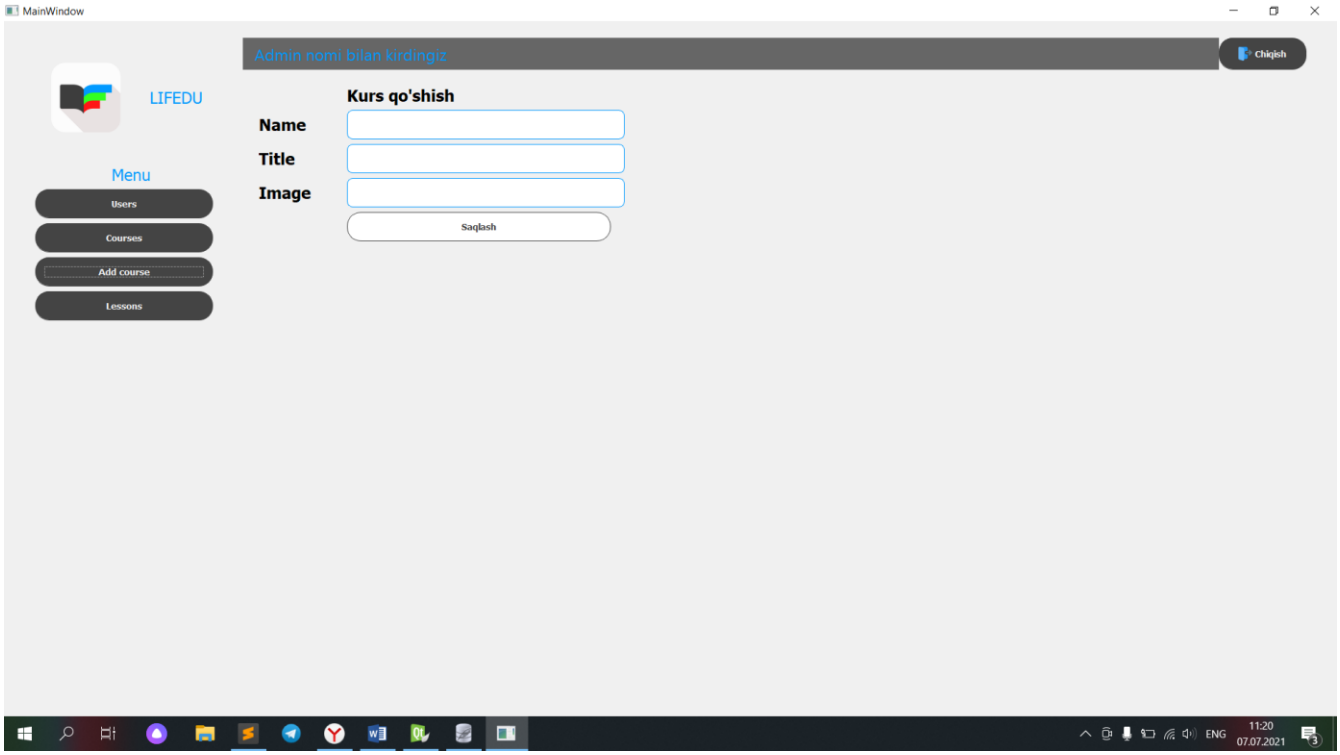


12-rasm Admin uchun foydalanuvchilarni qo’shish qayta ishlash sahifasi.

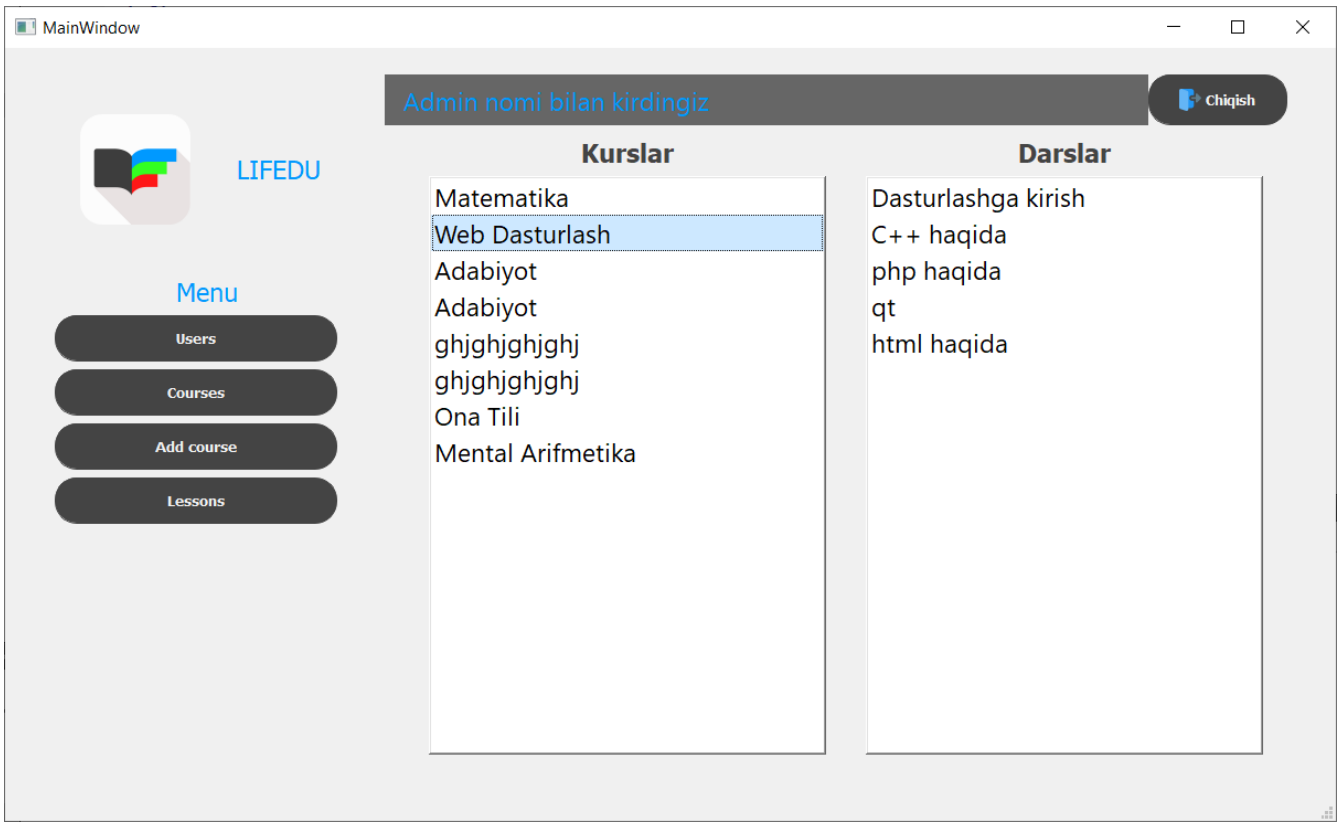
Foydalanuvchini ustiga 2 marta bossa Foydananuvchini ma’lumotlarini ko’radi.



13-rasm. Admin – kurslarga yangi dars qo’shish sahifasi



14-rasm. Kurs qo’shish sahifasi



15-rasm. Admin – kurslar va ularning darslarini nazorat qilish bo’limi

16-rasm. Loyiha logotipi.

Xulosa:

Ushbu kurs ishida QT creator muhiti, SqLite3 Ma’lumotlar bazasi va C++ dasturlash tilining imkoniyatlaridan foydalanib tayyorlandi. Bunda Ma’lumotlar bazasi va u bilan ishlashni tahlil qilib ko’rildi. Shuni aytish kerakki Loyihada MVC texnologiyasidan foydalanish tizimni yanada optimallashtirishga yordam beradi va tizimni ishlash yaxshi bo’ladi. C++ map(STL) imkoniyatlaridan foydalanib MB dagi ma’lumotlarni map bilan olib, u bilan qayta ishlandi. Sqlite3 MBning imkoniyatlaridan foydalanildi.

Foydalanilgan adabiyotlar:

1. https://doc.qt.io
2. https://www.geeksforgeeks.org
3. https://tutorials.uz