#include "boursewindoww.h"

#include "ui\_boursewindoww.h"

#include<QInputDialog>

#include<QMessageBox>

#include<QDebug>

#include<QRegExp>

#include<QListView>

#include"analyze.h"

#include<QListWidget>

#include<Qt>

#include<QDesktopServices>

#include<QUrl>

boursewindoww::boursewindoww(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::boursewindoww)

{

ui->setupUi(this);

db = QSqlDatabase::addDatabase("QSQLITE");

db.setDatabaseName("exchange.db");

db.open();

if(db.isOpen())

{

QSqlQuery query(db);

query.prepare("CREATE TABLE IF NOT EXISTS tab\_exchange(list varchar(100))");

if(!query.exec())

{

qDebug() << "Failed to create Table tab\_exchange";

}

}

ui->label\_3->setText(QString("%1").arg(ui->listbourse->count()));

connect(ui->search\_text,&QLineEdit::textChanged,this,&boursewindoww::findBourse);

}

boursewindoww::~boursewindoww()

{

delete ui;

}

void boursewindoww::on\_Select\_clicked()

{

QListWidgetItem \*item = ui->listbourse->currentItem();

if ((item->isSelected() and ui->Select->isEnabled()))

{

resultwind = new analyze(this);

resultwind->setWindowTitle("Results");

resultwind->show();

}

}

void boursewindoww::on\_ADD\_new\_clicked()

{

QString c\_text = QInputDialog::getText(this,"Item","Enter new item");

QString s\_text = c\_text.simplified();

if(!s\_text.isEmpty())

{

ui->listbourse->addItem(s\_text);

int r = ui->listbourse->count() -1;

ui-> listbourse->setCurrentRow(r);

QListWidgetItem \*item = ui->listbourse->currentItem();

ui->listbourse->sortItems( Qt::SortOrder::AscendingOrder );

ui->label\_3->setText(QString("%1").arg(ui->listbourse->count()));

// ui->listbourse->update();

// QSqlQuery query;

// query.prepare("INSERT INTO tab\_exchange (list) "

// "VALUES (:listbourse)");

// query.bindValue(":id", listbourse);

// query.exec();

QString itemEdit = ui->listbourse->currentItem()->text();

db.open();

if(db.isOpen())

{

qDebug() << "Database opened successfully";

QSqlQuery query(db);

query.prepare("SELECT list FROM tab\_exchange");

if (!query.exec()) {

qDebug() <<\_\_func\_\_ << " "<<\_\_LINE\_\_<< " "<< query.lastError().text();

return;

}

QSqlRecord record = query.record();

int list;

if((list = record.indexOf("userName")) == -1)

qDebug() << "Column list not in Database";

while(query.next())

{

QString listFound = query.value(list).toString();

if(listFound == list)

{

QMessageBox::critical(this,"Error","The stock exchange " + listFound + " already exists");

return;

}

}

//----------------------------------------------------------------------------------------------------------

query.prepare(" INSERT INTO tab\_exchange (itemEdit) VALUES (:list)");

query.bindValue(":id", itemEdit);

if (!query.exec()) {

qDebug() <<\_\_func\_\_ << " "<<\_\_LINE\_\_<< " "<< query.lastError().text();

QMessageBox::critical(this,"Error","Due to an error your request could not be completed");

return;

}else{

QMessageBox::information(this," sucessfull","Your addition has been sucessfull");

}

}

else

qDebug() << "Failed to open database because "<< db.lastError().text();

db.close();

this->close();

}

//save\_to\_database(listbourse);

//exch parcouriha mbaad tzid tab if not exists mbaad kol modif f listexc tetsajjl f db

}

void boursewindoww::save\_to\_database(QStringList& ai\_string)

{

if(!db.isOpen())

db.open();

if(db.isOpen())

{

qDebug() << "database opened successfully";

QSqlQuery query(db);

query.prepare("SELECT \* FROM tab\_exchange");

bool ret = query.exec();

if(ret)

qDebug() << "Query executed successfully";

else

qDebug() << "Query error "<< query.lastError().text();

// check if entry already in database

QSqlRecord record = query.record();

int list;

bool databaseError = false;

if((list = record.indexOf("list")) == -1)

qDebug() << "Column list not in Database";

while(query.next())

{

QString listFound = query.value(list).toString();

if(listFound == ai\_string[1])

{

qDebug() << "Entry is already in database";

databaseError = true;

break;

}

}

if(databaseError)

return;

// If not in database add it

query.prepare("INSERT INTO tab\_exchange(list)"

"VALUES(:el)");

// query.bindValue(":td", ai\_string[1].toInt());

query.bindValue(":el", ai\_string[1]);

// query.bindValue(":de", ai\_string[3]);

ret = query.exec();

if(ret)

qDebug() << "Query executed successfully";

else

qDebug() << "Query error "<< query.lastError().text();

// query.next();

// qDebug()<<query.value(0)<<" "<< query.value(1) <<" "<< query.value(2);

if(!query.isActive())

{

query.finish();

db.close();

}

}

else

qDebug() << "Database error " <<db.lastError().text();

}

void boursewindoww::on\_QUIT\_clicked()

{

int ret= QMessageBox::warning(0,"quit","Are you sure you want to quit",

QMessageBox::No|QMessageBox::Yes);

if (ret == QMessageBox::Yes)

close();

}

void boursewindoww::deleteitemfromdatabase(QString& ai\_string)

{

}

void boursewindoww::on\_Remove\_clicked()

{

int r = ui->listbourse->currentRow();

if (r!=-1) {

QListWidgetItem \*item = ui->listbourse->takeItem(r);

delete item;

}

ui->label\_3->setText(QString("%1").arg(ui->listbourse->count()));

// QSqlQuery dQuery(db);

// dQuery.prepare("DELETE FROM tab\_exchange WHERE rank != 1 AND list != \"B3" + this-> + "\"");

// if(dQuery.exec())

// {

// qDebug <<"Query executed!";

// }

}

void boursewindoww::findBourse(const QString arg)

{

QListWidgetItem \*item = nullptr;

for(int i = 0 ; i < ui->listbourse->count() ; i++)

{

item = ui->listbourse->item(i);

item->setHidden(false);

}

// QRegExp regExp (arg1,Qt::CaseInsensitive,QRegExp::Wildcard);

QList<QListWidgetItem \*> widgetList = ui->listbourse->findItems(arg, Qt::MatchContains);

if(widgetList.isEmpty())

{

qDebug() << "No match";

return;

}

QStringList itemsLabel;

for(QListWidgetItem \* listWidget : widgetList)

{

itemsLabel << listWidget->text();

qDebug() << listWidget->text();

}

ui->listbourse->selectAll();

item = nullptr;

for(int i = 0 ; i < ui->listbourse->count() ; i++)

{

item = ui->listbourse->item(i);

if(!itemsLabel.contains(item->text()))

{

// qDebug() << item->text() << "was hidden";

item->setHidden(true);

ui->label\_3->setText(QString("%1").arg(ui->listbourse->count()));

}

else if (itemsLabel.isEmpty() and ui->find->isEnabled()) {

QMessageBox::information(this, tr("Empty Search Field"),

tr("The search field is empty. "

"Please enter a word and click Find."));

}

}

ui->listbourse->repaint();

}

void boursewindoww::on\_About\_clicked()

{

QString link="https://markets.businessinsider.com/indices/world-stock-markets";

QDesktopServices::openUrl(QUrl (link));

QString link2="https://tradebrains.in/10-largest-stock-exchanges-in-the-world/";

QDesktopServices::openUrl(QUrl (link2));

}