Main cpp

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

blackboard w;

w.show();

return a.exec();

}

Blackboard cpp

blackboard::blackboard(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::blackboard)

{

ui->setupUi(this);

}

blackboard::~*blackboard*()

{

delete ui;

}

/\*-------------------------------------------------------

This function opens a the admin window if the admin

button is clicked. This current window will then be hid.

--------------------------------------------------------\*/

void blackboard::on\_pushButton\_admin\_clicked()

{

hide();

Landing = new landing(this);

Landing->show();

}

/\*-------------------------------------------------------

This function opens a the student window if the student

button is clicked. This current window will then be hid.

--------------------------------------------------------\*/

void blackboard::on\_pushButton\_student\_clicked()

{

hide();

Student = new student(this);

Student->show();

}

Landing page cpp

landing::landing(QWidget \*parent) :

QDialog(parent),

ui(new Ui::landing)

{

/\* This is the constructor that will pre load the dropdown menu

with all of the students names before the user is allow to.

interact-------------------------------------------------------\*/

ui->setupUi(this);

ui->stackedWidget->setCurrentIndex(0);

ui->comboBox\_students->addItem("(Choose Student)");

ui->comboBox\_students->addItem("John Witherspoon");

ui->comboBox\_students->addItem("Teresa Robers");

ui->comboBox\_students->addItem("Garrett Stephens");

ui->comboBox\_students->addItem("LeBron James");

}

landing::~*landing*()

{

delete ui;

}

void landing::on\_pushButton\_loginAdmin\_clicked()

{

QString adminPassword = ui->lineEdit\_passwordAdmin->text();

QString adminUsername = ui->lineEdit\_usernameAdmin->text();

if (adminUsername == "admin" && adminPassword == "password")

{

ui->stackedWidget->setCurrentIndex(1);

}

else

{

QMessageBox::information(this, "Error", "Incorrect Login. Try again.");

}

}

void landing::on\_pushButton\_addStudent\_clicked()

{

if (ui->comboBox\_students->currentIndex() == 0)

{

QMessageBox::information(this, "Error", "Must select a student");

}

else

{

ui->stackedWidget->setCurrentIndex(2);

}

}

/\*------------------------------------------------------------------------------------------

This function allows the admin to simply view the students record. Files are read only.

Admin must create the student file using the addStudent function before viewing the student

record or else an error will occur.

-------------------------------------------------------------------------------------------\*/

void landing::on\_pushButton\_viewStudent\_clicked()

{

if (ui->comboBox\_students->currentIndex() == 0)

{

QMessageBox::information(this, "Error", "Must select a student");

}

if (ui->comboBox\_students->currentIndex() == 1)

{

ui->stackedWidget->setCurrentIndex(3);

QFile record1("record1.txt");

if (!record1.*open*(QFile::ReadOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream in(&record1);

QString text = in.readAll();

ui->plainTextEdit\_viewRecord->setPlainText(text);

record1.*close*();

}

if (ui->comboBox\_students->currentIndex() == 2)

{

ui->stackedWidget->setCurrentIndex(3);

QFile record2("record2.txt");

if (!record2.*open*(QFile::ReadOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream in(&record2);

QString text = in.readAll();

ui->plainTextEdit\_viewRecord->setPlainText(text);

record2.*close*();

}

if (ui->comboBox\_students->currentIndex() == 3)

{

ui->stackedWidget->setCurrentIndex(3);

QFile record3("record3.txt");

if (!record3.*open*(QFile::ReadOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream in(&record3);

QString text = in.readAll();

ui->plainTextEdit\_viewRecord->setPlainText(text);

record3.*close*();

}

if (ui->comboBox\_students->currentIndex() == 4)

{

ui->stackedWidget->setCurrentIndex(3);

QFile record4("record4.txt");

if (!record4.*open*(QFile::ReadOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream in(&record4);

QString text = in.readAll();

ui->plainTextEdit\_viewRecord->setPlainText(text);

record4.*close*();

}

}

/\*---------------------------------------------------------------------------

This function allows the admin to create a brand new file for a student.

Based on which student is selected from the dropdown menu, the function

then creates a new file, and writes the desired info into the file, and

stores it so it can be viewed later. A template for the required information

is pre-loaded into the text field.

----------------------------------------------------------------------------\*/

void landing::on\_pushButton\_submitChanges\_clicked() //checks to see if student is selected, if not, error occurs.

{

if (ui->comboBox\_students->currentIndex() == 0)

{

QMessageBox::information(this, "Error", "Must select a student");

}

if (ui->comboBox\_students->currentIndex() == 1)

{

QFile record1("record1.txt");

if (!record1.*open*(QFile::WriteOnly | QFile::Text)) //checks to see if file is open, if notit creates new file.

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream out(&record1);

QString text = ui->plainTextEdit\_editRecord->toPlainText();

out << text;

record1.flush();

record1.*close*();

}

if (ui->comboBox\_students->currentIndex() == 2)

{

QFile record2("record2.txt");

if (!record2.*open*(QFile::WriteOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream out(&record2);

QString text = ui->plainTextEdit\_editRecord->toPlainText();

out << text;

record2.flush();

record2.*close*();

}

if (ui->comboBox\_students->currentIndex() == 3)

{

QFile record3("record3.txt");

if (!record3.*open*(QFile::WriteOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream out(&record3);

QString text = ui->plainTextEdit\_editRecord->toPlainText();

out << text;

record3.flush();

record3.*close*();

}

if (ui->comboBox\_students->currentIndex() == 4)

{

QFile record4("record4.txt");

if (!record4.*open*(QFile::WriteOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream out(&record4);

QString text = ui->plainTextEdit\_editRecord->toPlainText();

out << text;

record4.flush();

record4.*close*();

}

}

void landing::on\_pushButton\_back\_2\_clicked() //\*\*\*

{

ui->stackedWidget->setCurrentIndex(1); //\*\*\*\*\*\*\*\*\*\*\*

}

void landing::on\_pushButton\_back\_clicked() //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* These three functions serve as "back" buttons

{

ui->stackedWidget->setCurrentIndex(1); //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* to get back to the previous window.

}

void landing::on\_pushButton\_back\_3\_clicked() //\*\*\*\*\*\*\*\*\*\*\*

{

ui->stackedWidget->setCurrentIndex(1); //\*\*\*

}

/\*-------------------------------------------------------

This function takes the admin to the edit student window

--------------------------------------------------------\*/

void landing::on\_pushButton\_editStudent\_clicked()

{

if (ui->comboBox\_students->currentIndex() == 0)

{

QMessageBox::information(this, "Error", "Must select a student.");

}

else

{

ui->stackedWidget->setCurrentIndex(4);

}

}

/\*--------------------------------------------------------------------------------------

This function loads the files in similar to the addStudent

funtion, and this allows the admin to edit the curren

student information.

---------------------------------------------------------------------------------------\*/

void landing::on\_pushButton\_edit\_clicked()

{

if (ui->comboBox\_students->currentIndex() == 0)

{

QMessageBox::information(this, "Error", "Must select a student");

}

if (ui->comboBox\_students->currentIndex() == 1)

{

QFile record1("record1.txt");

if (!record1.*open*(QFile::WriteOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream out(&record1);

QString text = ui->plainTextEdit\_editRecord->toPlainText();

out << text;

record1.flush();

record1.*close*();

}

if (ui->comboBox\_students->currentIndex() == 2)

{

QFile record2("record2.txt");

if (!record2.*open*(QFile::WriteOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream out(&record2);

QString text = ui->plainTextEdit\_editRecord->toPlainText();

out << text;

record2.flush();

record2.*close*();

}

if (ui->comboBox\_students->currentIndex() == 3)

{

QFile record3("record3.txt");

if (!record3.*open*(QFile::WriteOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream out(&record3);

QString text = ui->plainTextEdit\_editRecord->toPlainText();

out << text;

record3.flush();

record3.*close*();

}

if (ui->comboBox\_students->currentIndex() == 4)

{

QFile record4("record4.txt");

if (!record4.*open*(QFile::WriteOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream out(&record4);

QString text = ui->plainTextEdit\_editRecord->toPlainText();

out << text;

record4.flush();

record4.*close*();

}

}

Student view cpp

student::student(QWidget \*parent) :

QDialog(parent),

ui(new Ui::student)

{

ui->setupUi(this); // This is the constructor for the stdent class.

ui->stackedWidget\_student->setCurrentIndex(0); // It loads the combo box (dropdown menu)

ui->comboBox\_studentView->addItem("(Choose Student)"); // with all of the student names up launching

ui->comboBox\_studentView->addItem("John Witherspoon"); // this window.

ui->comboBox\_studentView->addItem("Teresa Robers");

ui->comboBox\_studentView->addItem("Garrett Stephens");

ui->comboBox\_studentView->addItem("LeBron James");

}

student::~*student*()

{

delete ui;

}

/\*-------------------------------------------------------------------------

This function checks to see wheter or not a student has been selected from

the dropdown menu upon clicking the "view" button. If no student was selected,

an error message will occur. If a student has been selected, the user will

be taken to the next window. Each student has a unique login.

--------------------------------------------------------------------------\*/

void student::on\_pushButton\_viewStudent\_clicked()

{

if (ui->comboBox\_studentView->currentIndex() == 0)

{

QMessageBox::information(this, "Error", "Must select a student");

}

else

{

ui->stackedWidget\_student->setCurrentIndex(1);

}

}

/\*------------------------------------------------------------------------------------------

This function runs once the student clicks submit after entering their username and password.

The login information is verified, and upon correct entry, the student is taken to the next

window where their record file is loaded into the text field for the student to view.

All files are read only.

-------------------------------------------------------------------------------------------\*/

void student::on\_pushButton\_submitStudent\_clicked()

{

if (ui->comboBox\_studentView->currentIndex() == 1)

{

//ui->stackedWidget\_student->setCurrentIndex(1);

if (ui->lineEdit\_studentUsername->text() == "john" && ui->lineEdit\_studentPass->text() == "password")

{

ui->stackedWidget\_student->setCurrentIndex(2);

}

else

{

QMessageBox::information(this, "Error", "Incorrect login.");

}

QFile record1("record1.txt");

if (!record1.*open*(QFile::ReadOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream in(&record1);

QString text = in.readAll();

ui->plainTextEdit\_viewStudent->setPlainText(text);

record1.*close*();

}

else if (ui->comboBox\_studentView->currentIndex() == 2)

{

//ui->stackedWidget->setCurrentIndex(3);

if (ui->lineEdit\_studentUsername->text() == "teresa" && ui->lineEdit\_studentPass->text() == "password")

{

ui->stackedWidget\_student->setCurrentIndex(2);

}

else

{

QMessageBox::information(this, "Error", "Incorrect login.");

}

QFile record2("record2.txt");

if (!record2.*open*(QFile::ReadOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream in(&record2);

QString text = in.readAll();

ui->plainTextEdit\_viewStudent->setPlainText(text);

record2.*close*();

}

else if (ui->comboBox\_studentView->currentIndex() == 3)

{

//ui->stackedWidget->setCurrentIndex(3);

if (ui->lineEdit\_studentUsername->text() == "garrett" && ui->lineEdit\_studentPass->text() == "password")

{

ui->stackedWidget\_student->setCurrentIndex(2);

}

else

{

QMessageBox::information(this, "Error", "Incorrect login.");

}

QFile record3("record3.txt");

if (!record3.*open*(QFile::ReadOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream in(&record3);

QString text = in.readAll();

ui->plainTextEdit\_viewStudent->setPlainText(text);

record3.*close*();

}

else if (ui->comboBox\_studentView->currentIndex() == 4)

{

//ui->stackedWidget->setCurrentIndex(3);

if (ui->lineEdit\_studentUsername->text() == "lebron" && ui->lineEdit\_studentPass->text() == "password")

{

ui->stackedWidget\_student->setCurrentIndex(2);

}

else

{

QMessageBox::information(this, "Error", "Incorrect login.");

}

QFile record4("record4.txt");

if (!record4.*open*(QFile::ReadOnly | QFile::Text))

{

QMessageBox::information(this, "Error", "There is no data for this student.");

}

QTextStream in(&record4);

QString text = in.readAll();

ui->plainTextEdit\_viewStudent->setPlainText(text);

record4.*close*();

}

}

// This function goes back to the student view selection screen

void student::on\_pushButton\_goBack\_clicked()

{

ui->stackedWidget\_student->setCurrentIndex(0);

}

Header files

Blackboard.h

#include <QMainWindow>

#include "landing.h"

#include "student.h"

namespace Ui {

class blackboard;

}

class blackboard : public QMainWindow

{

Q\_OBJECT

public:

explicit blackboard(QWidget \*parent = 0);

~*blackboard*();

private slots:

void on\_pushButton\_admin\_clicked();

void on\_pushButton\_student\_clicked();

private:

Ui::blackboard \*ui;

landing \*Landing;

student \*Student;

};

Landing page.h

namespace Ui {

class landing;

}

class landing : public QDialog

{

Q\_OBJECT

public:

explicit landing(QWidget \*parent = 0);

~*landing*();

private slots:

void on\_pushButton\_loginAdmin\_clicked();

void on\_pushButton\_viewStudent\_clicked();

void on\_pushButton\_addStudent\_clicked();

void on\_pushButton\_back\_2\_clicked();

void on\_pushButton\_back\_clicked();

void on\_pushButton\_submitChanges\_clicked();

void on\_pushButton\_editStudent\_clicked();

void on\_pushButton\_edit\_clicked();

void on\_pushButton\_back\_3\_clicked();

private:

Ui::landing \*ui;

};

Student view.h

namespace Ui {

class student;

}

class student : public QDialog

{

Q\_OBJECT

public:

explicit student(QWidget \*parent = 0);

~*student*();

private slots:

void on\_pushButton\_viewStudent\_clicked();

void on\_pushButton\_submitStudent\_clicked();

void on\_pushButton\_goBack\_clicked();

private:

Ui::student \*ui;

};