МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ПЕРСПЕКТИВНОЙ ИНЖЕНЕРИИ

**Кафедра прикладной информатики**

Лабораторная работа №1-4

ПО ДИСЦИПЛИНЕ «Программирование мобильных устройств»

**Выполнил:**

студент группы ПИН-Б-З-22-1

Гадиян Сергей Гариевич

Ставрополь, 2025 г.

Ход работы.

Модуль 1, Задание 1 выполнено согласно лабораторной работе(рис.1,2).

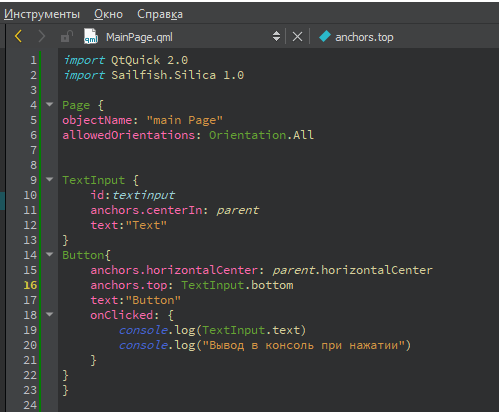


Рисунок 1 – Код задания 1

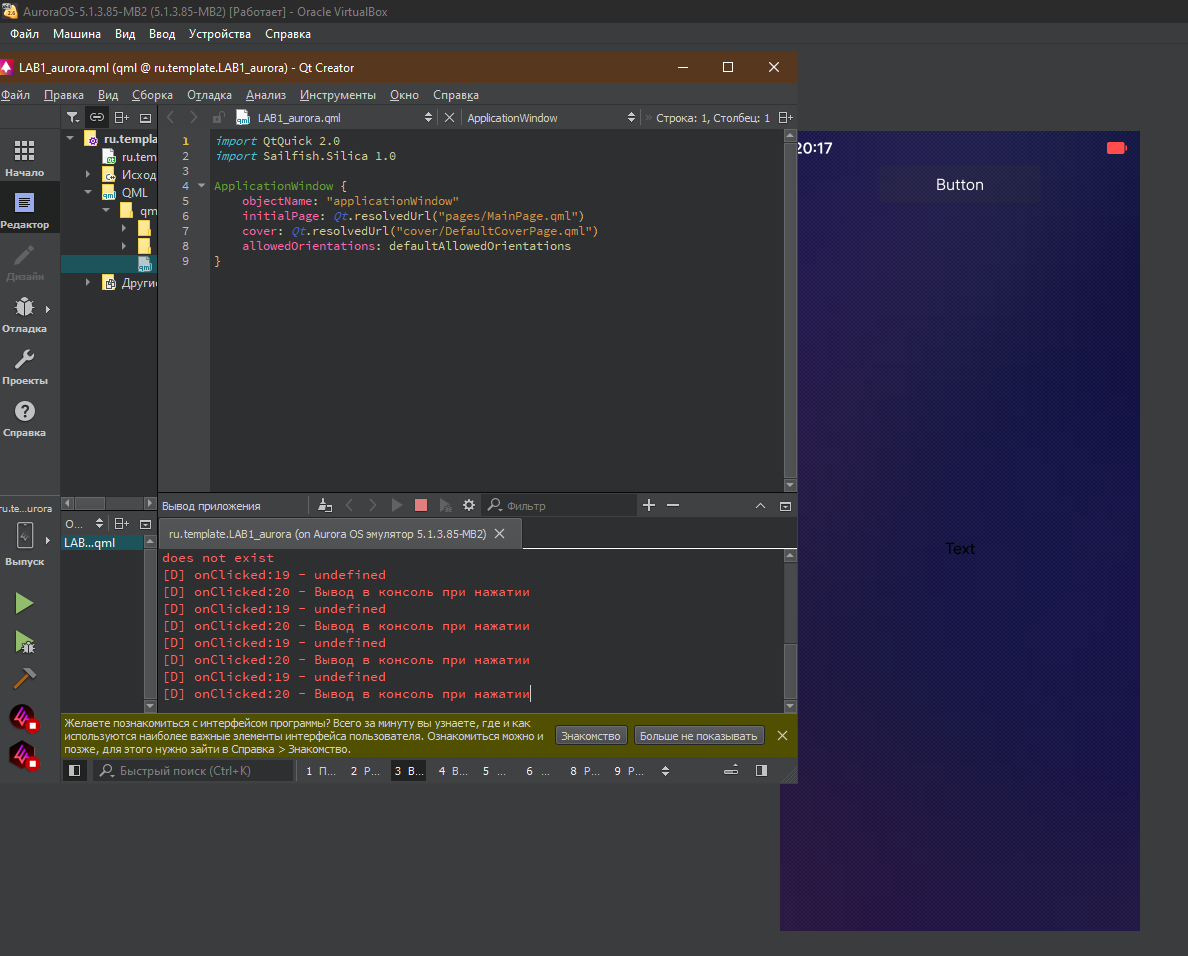


Рисунок 2 – Вывод результата выполнения приложения в консоль

Модуль 1, Задание 2 выполнено согласно лабораторной работе(рис.3,4).

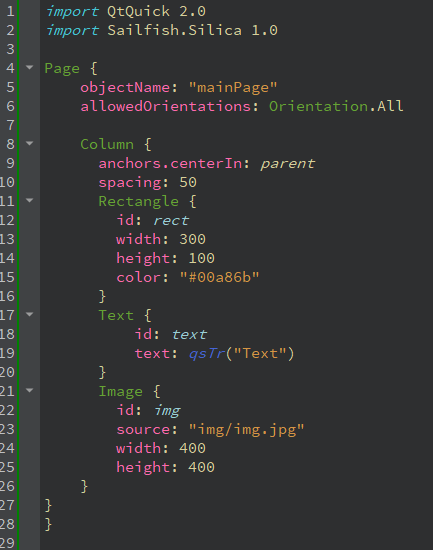


Рисунок 3 – Программный код

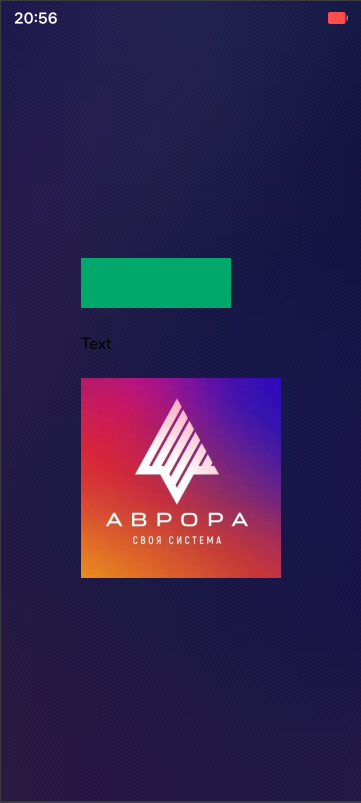


Рисунок 4 – Результат

Модуль 1 Задание 3 выполнено согласно лабораторной работе(рис.5,6).

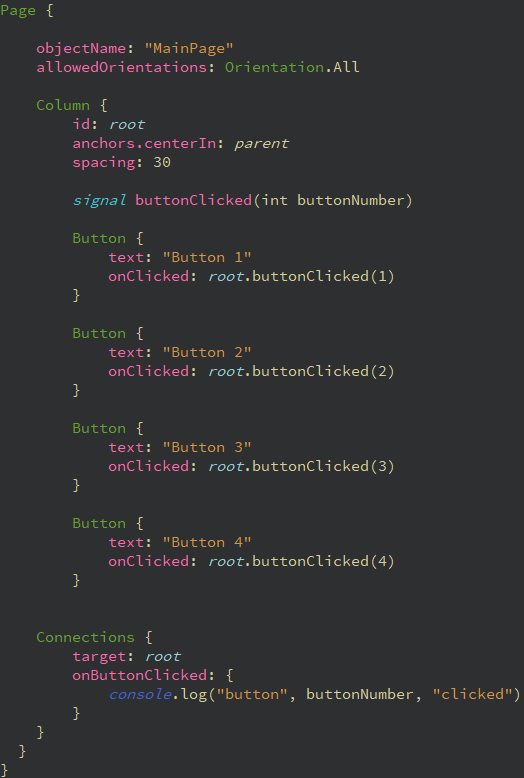


Рисунок 5 – Программный код

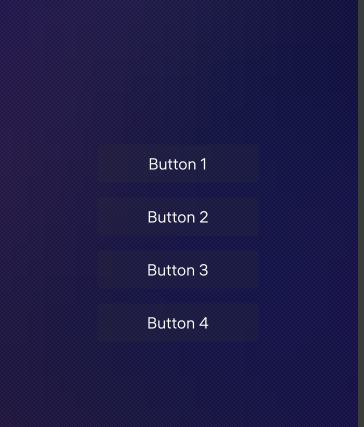


Рисунок 6 – Результат

Модуль 2 Задание 1 выполнено согласно лабораторной работе(рис.7,8,9).



Рисунок 7 – Результат выполнения



Рисунок 8 – Перемещение прямоугольника при нажатии на него



Рисунок 9 – Программный код

Модуль 2 Задание 2 выполнено согласно лабораторной работе(рис.10,11,).

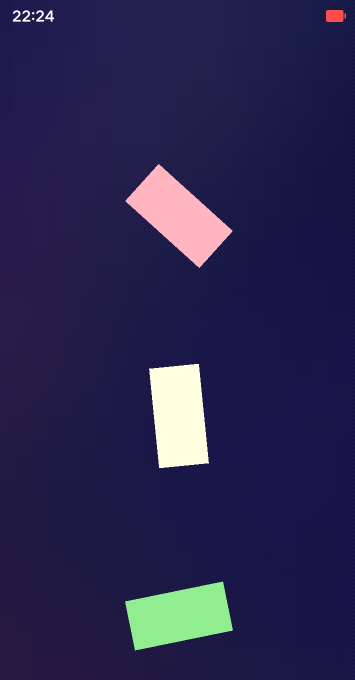


Рисунок 10 - Результат



Рисунок 11 – Программный код

Модуль 2 Задание 3 выполнено согласно лабораторной работе(рис.12,13).

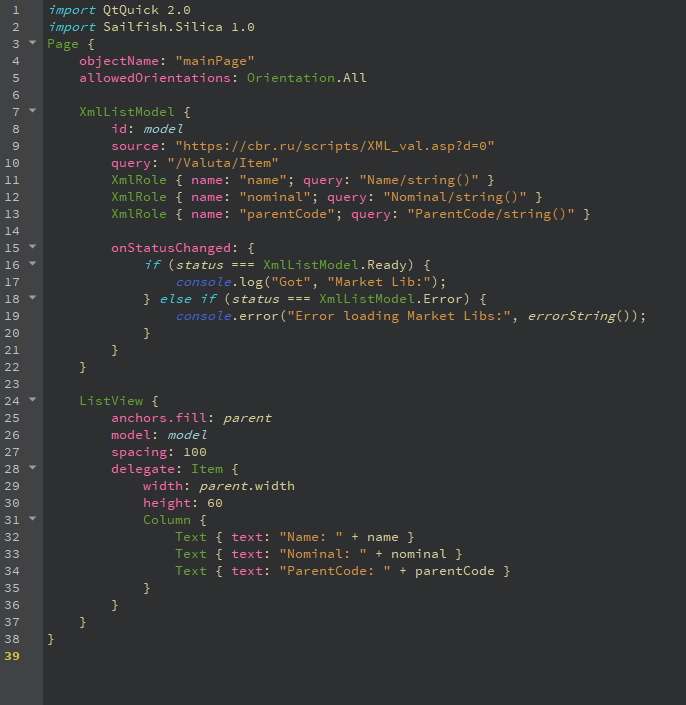


Рисунок 12 – Код

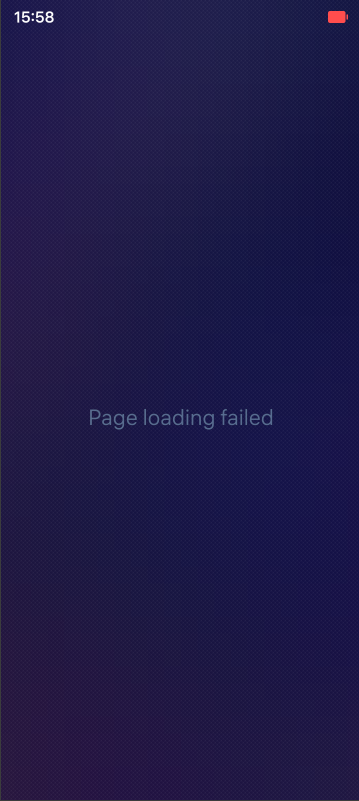


Рисунок 13 – Результат

Модуль 3, задание 1 выполнено согласно лабораторной работе (рис.13,14,15).



Рисунок 13 – Код

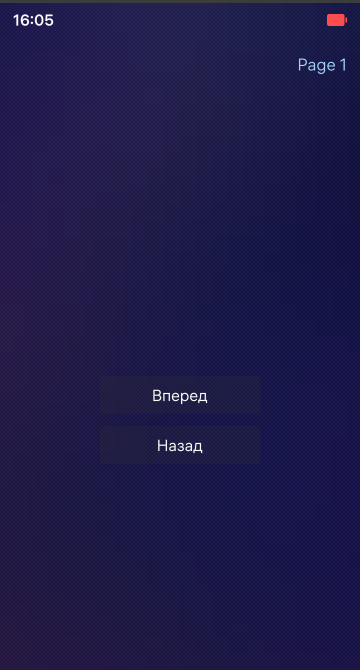


Рисунок 14 – Результат

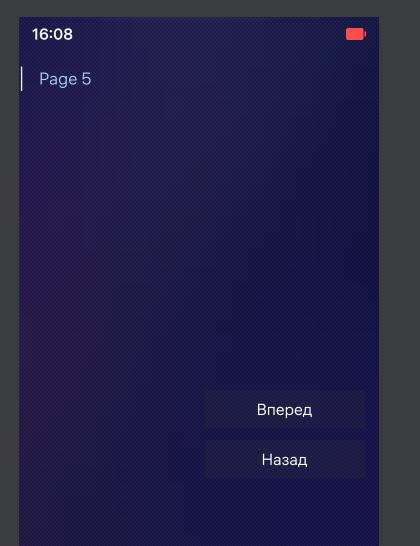


Рисунок 15 – Результат

В результате все кнопки функционируют, страницы листаются.

Модуль 3, задание 2 выполнено согласно лабораторной работе (рис.16,17,18).

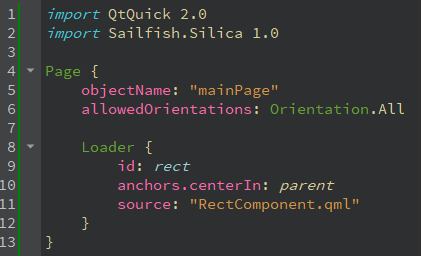


Рисунок 16 – MainPage.qml

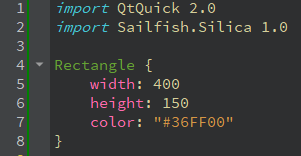


Рисунок 17 – RectComponent.qml

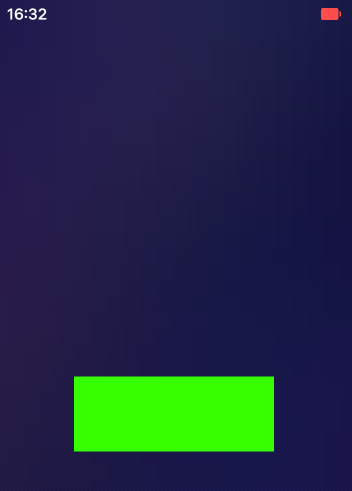


Рисунок 18 – Результат

Модуль 3, задание 2 выполнено согласно лабораторной работе (рис.19,20).



Рисунок 19 – Код

Результат – работающий слайдер.

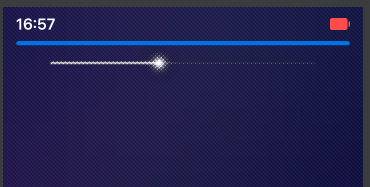


Рисунок 20 – Результат

**Модуль 4. Задание 1.**

Создаем два новых файла (Рис.21)

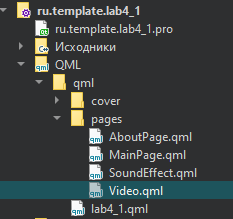


Рисунок 21 – Новые файлы

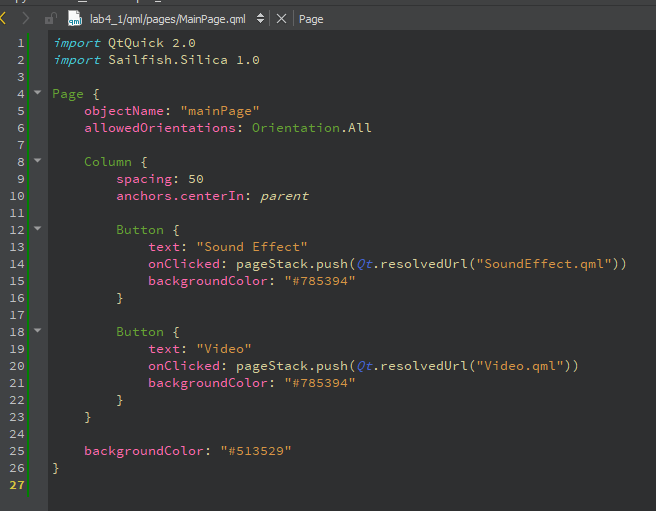


Рисунок 22 – Содержимое MainPage.qml

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

*import* QtMultimedia 5.6

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

SoundEffect {

id: *sound*

source: "audio.wav"

}

Column {

anchors.centerIn: *parent*

width: *parent*.width

Button {

id: *btn*

text: "Play Sound Effect"

anchors.horizontalCenter: *parent*.horizontalCenter

onClicked: *sound*.play()

backgroundColor: "#585394"

}

}

Slider {

id: *volumeSlider*

label: "Звук"

value: 1.0

valueText: *value*

minimumValue: 0.0

maximumValue: 1.0

stepSize: 0.1

width: *parent*.width - Theme.paddingLarge \* 2

onValueChanged: *sound*.volume = *value*

color: "#585394"

backgroundColor: "#585394"

}

Slider {

id: *repeatSlider*

label: "Количество повторений"

value: 1.0

valueText: *value*

minimumValue: 0

maximumValue: 10

stepSize: 1

width: *parent*.width - Theme.paddingLarge \* 2

onValueChanged: *sound*.loops = *value*

color: "#585394"

backgroundColor: "#585394"

}

}

Листинг – Содержимое SoundEffect.qml

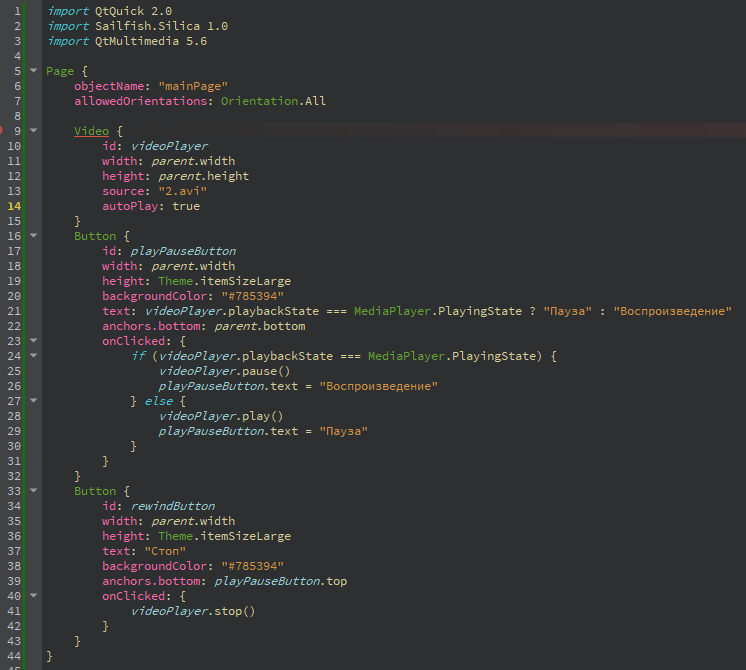


Рисунок 23 – Содержимое Video.qml

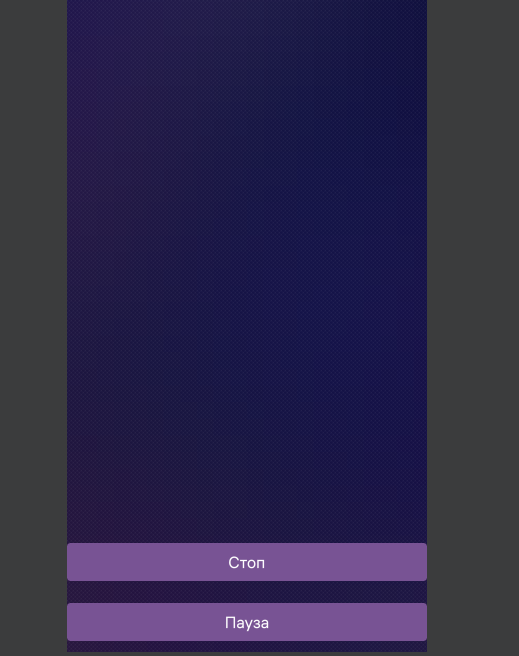


Рисунок 24 – Результат

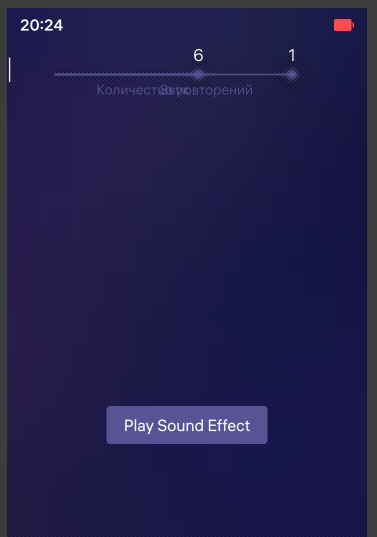


Рисунок 25 – Результат

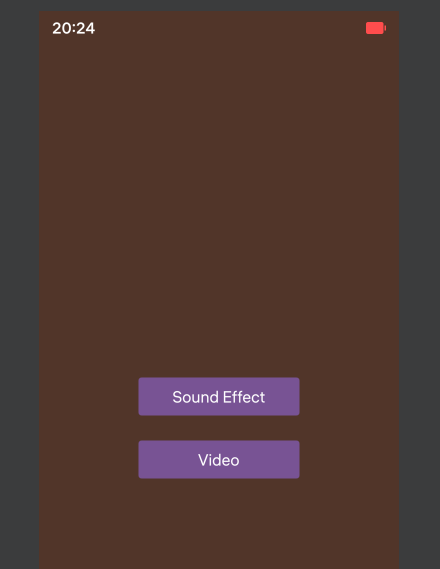


Рисунок 26 – Результат

**Модуль 4. Задание 2.**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: OrientationAll

backgroundColor: "#ffebcd"

ListModel {

id: *jsonModel*

}

ListView {

anchors.fill: *parent*

model: *jsonModel*

spacing: 130

delegate: Item {

Row {

spacing: 30

x: 30

Text {

text: "ID: " + model.id

color: "#ff5800"

font.pointSize: 30

font.Family: "monospace"

}

Text {

text: "Completed: " + model.completed

color: "#390000"

font.pointSize: 30

font.Family: "monospace"

}

}

}

}

*function* *processData*(data) {

*for* (*var* *i* = 0; *i* < *data*.length; *i*++) {

*jsonModel*.append({

"id": *data*[*i*].id,

"completed": *data*[*i*].completed

});

}

}

*function* *processData*(data) {

*for* (*var* *i* = 0; *i* < *data*.length; *i*++) {

*jsonModel*.append({

"id": *data*[*i*].id,

"completed": *data*[*i*].completed

});

}

}

Component.onCompleted: {

*var* *request* = *new* *XMLHttpRequest*();

*request*.onreadystatechange = *function*() {

*if* (*request*.readyState === *XMLHttpRequest*.DONE) {

*if* (*request*.status === 200) {

*var* *response* = *JSON*.parse(*request*.responseText);

*processData*(*response*);

} *else* {

*console*.error("Failed to fetch JSON data", *request*.status, *request*.statusText);

}

}

};

*request*.open("GET", "https://jsonplaceholder.typicode.com/todos", true);

*request*.send();

}

}

Листинг – Программный код MainPage.qml

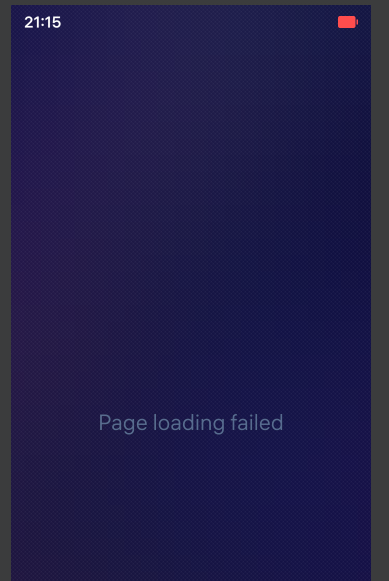


Рисунок 27 – Результат

В результате запуска приложения ничего не вывелось, мне не удалось исправить

**Модуль 4. Задание 3.**

*import* QtQuick 2.0

*import* Sailfish.Silica 1.0

*import* QtQuick.LocalStorage 2.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

backgroundColor: "#f8f8ff"

*property* *string* surname: ""

*property* *string* name: ""

*property* *string* secondName: ""

*property* *var* noteModel: ListModel {

ListElement { note: "Киндрук Екатерина Александровна" }

ListElement { note: "Дубовицкая Юлия Александровна" }

}

*function* *addNote*() {

*if* (*surname* !== "" && *name* !== "" && *secondName* !== "") {

*noteModel*.append({"note": *surname* + " " + *name* + " " + *secondName*})

}

}

*function* *deleteNote*(index) {

*noteModel*.remove(*index*)

}

*function* *initializeDatabase*() {

*var* *db* = LocalStorage.openDatabaseSync("notesDB", "1.0", "Notes Database", 1000000)

*db*.transaction(*function*(tx) {

*tx*.executeSql('CREATE TABLE IF NOT EXISTS notes(note TEXT)')

*console*.log("Таблица создана")

})

}

Component.onCompleted: {

*initializeDatabase*()

}

SilicaFlickable {

anchors.fill: *parent*

Column {

id: *column*

y: 30

width: *parent*.width

TextField {

id: *noteSurname*

placeholderText: "Фамилия"

width: *parent*.width

onTextChanged: *surname* = *text*

EnterKey.onClicked: *noteName*.focus = true

placeholderColor: *noteSurname*.highlighted ? "#423189" : Theme.darkSecondaryColor

}

TextField {

id: *noteName*

placeholderText: "Имя"

onTextChanged: *name* = *text*

EnterKey.onClicked: *noteSecondName*.focus = true

placeholderColor: *noteName*.highlighted ? "#423189" : Theme.darkSecondaryColor

}

TextField {

id: *noteSecondName*

placeholderText: "Отчество"

onTextChanged: *secondName* = *text*

EnterKey.onClicked: *focus* = false

placeholderColor: *noteSecondName*.highlighted ? "#423189" : Theme.darkSecondaryColor

}

Button {

id: *addButton*

text: "Добавить"

anchors.horizontalCenter: *parent*.horizontalCenter

color: "#423189"

highlightBackgroundColor: "#b4a8e0"

highlightColor: "#b69eff"

onClicked: *addNote*()

}

}

ListView {

id: *noteListView*

width: *parent*.width

height: *parent*.height / 2

model: *noteModel*

anchors.top: *column*.bottom

delegate: Item {

width: *parent*.width

height: Theme.itemSizeMedium

Row {

x: 20

Label {

text: model.note

wrapMode: Text.Wrap

width: 600

anchors.verticalCenter: *parent*.verticalCenter

color: Theme.primaryColor

font.family: "Liberation Mono"

font.pixelSize: Theme.fontSizeSmall

}

IconButton {

id: *btnDelete*

icon.source: "delete.png"

onClicked: *deleteNote*(index)

}

}

}

}

}

}

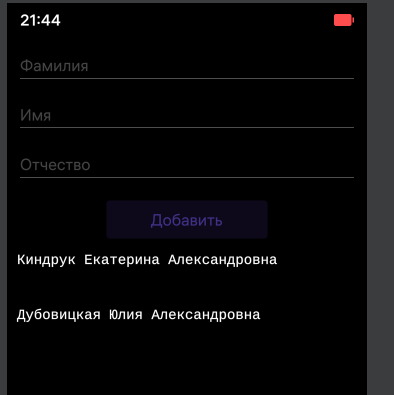


Рисунок 28 – Результат

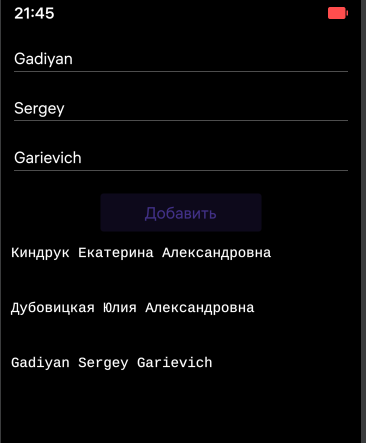


Рисунок 29 – Результат добавления данных