

МИНОБРНАУКИ РОССИИ  
Федеральное государственное автономное образовательное  
учреждение высшего образования  
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт компьютерных технологий и информационной безопасности

Направление подготовки (шифр, название): 09.03.04 «Программная инженерия»

## Отчёт по практике

Обучающегося 1 курса

Фамилия: Жалнин

Имя: Дмитрий

Отчество: Игоревич

Обучающийся



Жалнин Д.И.

подпись

расшифровка подписи

Место практики: Научно-исследовательская лаборатория "Лаборатория мобильной и веб-разработки" ООО НПИЦИТ "Иносфера"

наименование профильной организации

с

Исправить титульный по образцу! С таким же форматированием. Отчет должен включать 4 описание 4х модулей.

Вид практики: Учебная практика

Тип практики: Ознакомительная практика, распределённая

Способ проведения практики: Стационарная

Форма проведения практики: Дискретная (по видам и периодам практик)

Сроки прохождения практики с 10.02.2020 по 21.06.2020

Руководитель практики  
от структурного подразделения ЮФУ

Пирская Любовь Владимировна

ФИО, подпись

## Содержание

1	Android-Разработка .....	4
1.1	Постановка задачи .....	4
1.2	Разработка приложения .....	4
1.3	MainActivity .....	4
1.3.1	Алгоритм работы activity .....	4
1.3.2	Работа с ресурсами activity .....	6
1.4	WebActivity .....	7
1.4.1	Алгоритм работы activity .....	7
1.5	Описание интерфейса приложения .....	10
1.5.1	“MainActivity” .....	10
1.5.2	“WebActivity” .....	12
1.5.3	Видео работы приложения.....	13
2	Администрирование .....	14
2.1	Постановка задачи .....	14
2.2	Описание выполненных работ .....	14
2.2.1	Установка технологии контроля версий на базе Git.....	14
2.2.2	Создание локального репозитория .....	15
2.3	Создание и добавление удаленного репозитория .....	16
2.4	Создание ветвлений.....	18
2.5	Слияние ветвлений .....	20
2.6	Создание запроса на слияние .....	21
2.7	Разрешение конфликтов локально.....	23
2.8	Разрешение конфликтов на GitLab .....	26
2.9	Конфигурация сервера по умолчанию .....	27
2.10	Конфигурация виртуального сервера.....	28
2.11	Ссылки .....	29
2.12	Заключение.....	29
3	Веб-разработка .....	31
3.1	Постановка задачи .....	31
3.2	Описание разработки сайта .....	31
3.2.1	Выбор цветовой палитры .....	31

3.2.2	Создание макета сайта .....	32
3.2.3	Подготовка графического материала для сайта.....	32
3.2.4	Создание Header страницы.....	32
3.2.5	Создание Footer страницы.....	34
3.3	Вёрстка основной части страницы .....	35
3.4	Адаптация сайта под мобильные устройства. ....	35
3.5	Именованние URL страниц .....	36
3.6	Заключение.....	36
4	Знакомство с 1С .....	37
4.1	Постановка задачи .....	37
4.2	Создание справочников .....	37
4.3	Создание накладной .....	39
4.4	Создание процедур и связанных формулой реквизитов .....	40
4.5	Заключение.....	40
	Приложение .....	42

# **1 Android-Разработка**

## **1.1 Постановка задачи**

Необходимо разработать стабильно работающее мобильное приложение без «падений», «крашей», exception для ОС Android в AnroidStudio, включающее 2 activity:

1. Первая activity содержит авторизацию. Пользователь вводит 2 поля (Имя пользователя, пароль). Список логинов и паролей хранится в ресурсах. Предусмотреть обработку неверного логина и пароля с информированием пользователя. При успешной авторизации пользователь попадает на вторую activity, где в шапке приложения отображается: название приложения и имя пользователя.

2. На второй Activity открывается веб-сайт в приложении с использованием WebView. В зависимости от того, под каким логином и паролем «зашел» пользователь в приложение, открывается соответствующий данному пользователю сайт. Адреса веб-сайта можно также хранить в ресурсах или задать прямо в коде приложения, выбор на усмотрение студента. Приложение должно поддерживать вертикальную и горизонтальную ориентацию. Тематика, дизайн, расположение элементов выбирается студентом самостоятельно.

## **1.2 Разработка приложения**

Разработанное Android-приложение состоит из 2х активити: MainActivity и web.

## **1.3 MainActivity**

MainActivity – является первым activity, на которое попадает пользователь при запуске приложения. Интерфейс activity описан в п.3.1.

### **1.3.1 Алгоритм работы activity**

При вводе данных и нажатии кнопки происходит сверка текста в полях и в локальной “базе данных”. Если введенная пара “логин-пароль” совпадают с одной из хранящихся пар, пользователь попадает на следующую activity WebActivity. Иначе пользователь увидит появляющуюся надпись с текстом

«Неверный логин или пароль, повторите попытку». В случае, если при нажатии кнопки поля пусты, сообщение о неправильности введенных данных будет так же показано.

Локальная “база данных” представляет собой 3 расположенных (в ресурсах) в файле arrays.xml массива строк с доступными именами пользователя, паролями и уникальными для каждого имени адресами сайтов, которые передаются в следующую activity для загрузки:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name = 'Passwords'>
        <item>251100</item>
        <item>123456</item>
        <item>qwerty</item>
    </string-array>

    <string-array name = 'Sites'>
        <item>https://vseinstrumenti.ru</item>
        <item>https://chipdip.ru</item>
        <item>https://citilink.ru</item>
    </string-array>

    <string-array name = "Logins">
        <item>Petr</item>
        <item>Danil</item>
        <item>Dima</item>
    </string-array>
</resources>
```

Алгоритм нажатия кнопки выполняется в функции “click()”, которая присоединена к кнопке в XML-layout файле в качестве слушателя. В случае правильного ввода данных функция “click()” создаёт экземпляр “Intent” для перехода к “WebActivity”, помещает туда необходимые данные (имя пользователя и индекс нужного сайта в ) с помощью команды “putExtra()” и запускает новую activity функцией “startActivity()”, при неправильном вводе на экран выводится “TextView” с оповещением о неправильно введенном пароле.

```
public void LoginButton(View view) {
    Login = LoginText.getText().toString();
    Password = PasswordText.getText().toString();

    // Проверка на совпадение с одним из пароль/логинов
    int CountOfUsers = 3;
    String[] LoginsList = getResources().getStringArray(R.array.Logins);
```

```

String[] PasswordsList =
getResources().getStringArray(R.array.Passwords);
int IndexOfCoincidence = -100,i;

for (i = 0;i < CountOfUsers; i++)
    if(LoginsList[i].equals(Login))break;

if(i < CountOfUsers)IndexOfCoincidence = i;

if(IndexOfCoincidence == -100)
{
    Log.i(Tag, "Неверный логин или пароль!");
    ErrorText.setVisibility(View.VISIBLE);
}
else
{
    if (PasswordsList[IndexOfCoincidence].equals>Password))
    {
        ErrorText.setVisibility(View.INVISIBLE);
        Log.i(Tag, "Выполнен вход! Запускаю следующий активити");
        //Intent intent = new Intent(MainActivity.this, web.class);
        Intent intent = new Intent(MainActivity.this, WebActivity.class);
        intent.putExtra("Index", IndexOfCoincidence);
        startActivity(intent);
        ErrorText.setVisibility(View.INVISIBLE);
        LoginText.setText("");
        PasswordText.setText("");
    }
    else
    {
        Log.i(Tag, "Неверный логин или пароль!");
        ErrorText.setVisibility(View.VISIBLE);
    }
}
}
}

```

### 1.3.2 Работа с ресурсами activity

Дизайн MainActivity прописан в двух XML-layout файлах (см п.4.1 и п.4.2), хранящихся в папке “./res/layouts”. Один из них предназначен для вертикальной ориентации устройства, второй – для горизонтальной. Присоединение файлов разметки и кода приложение происходит с помощью функции “findViewById(R.id. ...)”:

```

...
private TextView ErrorText,LoginText>PasswordText;pet;

```

```

...
@Override
protected void onCreate(Bundle savedInstanceState)
{
...
    EditText = findViewById(R.id.ErrorMessage);
    LoginText = findViewById(R.id.LoginText);
    PasswordText = findViewById(R.id.PasswordText);
...
}

```

## 1.4 WebActivity

WebActivity – второе activity, на которое попадает пользователь после успешной авторизации. Интерфейс activity описан в п.3.2.

### 1.4.1 Алгоритм работы activity

Получение данных из “MainActivity” в “WebActivity” осуществляется из объекта “Intent”, созданного при переходе из первого ко второму activity, с помощью команды “getIntent().getSerializableExtra()”:

```

@Override
protected void onCreate(Bundle savedInstanceState)
{
...
    int txtName = getIntent().getIntExtra("Index", -1);
...
}

```

Передача индекса пользователя в “шапку” приложения происходит с помощью отправки в ActionBar текста из переменной “Logins[txtName]”. Установка текста в ActionBar производится с помощью функции setTitle("DIY helper:" + Logins[txtName]);

```

@Override
protected void onCreate(Bundle savedInstanceState)
{
...
    // Принимаем Индекс и по нему вызываем нужный сайт
    int txtName = getIntent().getIntExtra("Index", -1);
    String Sites[] = getResources().getStringArray(R.array.Sites);
    Log.i(Tag, Sites[txtName]);
    String Logins[] = getResources().getStringArray(R.array.Logins);
    setTitle("DIY helper:" + Logins[txtName]);
...
}

```

```
}
```

Алгоритм настройки “WebView” – встроенного браузера который загружает уникальную страницу для каждого пользователя :

```
private                                WebView                                webView;  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
...  
webView = findViewById(R.id.Site);  
webView.setWebViewClient(new                                MyWebViewClient());  
wv.getSettings().setLoadsImagesAutomatically(true);  
...  
}
```

При пересоздании activity, страница, на которой находился пользователь будет сохранена и снова открыта с помощью процесса сохранения, описанного в методе “saveInstanceState()”. Если же activity была загружена впервые, будет открыта страница, которая передавалась из “MainActivity” в объекте “Intent”:

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
...  
if (savedInstanceState==null)  
{  
    webView.loadUrl(Sites[txtName]);  
    Log.i(Tag,Sites[txtName]);  
}  
                                else                                {  
    webView.loadUrl(savedInstanceState.getString("Url"));  
    Log.i(Tag,CurrentUrl);  
}  
...  
}  
@Override  
protected void onSaveInstanceState(@NonNull Bundle outState)  
{  
    super.onSaveInstanceState(outState);  
    outState.putString("Url", webView.getUrl());  
}
```

Также к “WebView” подключена возможность пролистывания страниц по нажатию кнопки “back”:

```
@Override  
public void onBackPressed()  
{
```



```
if (webView.canGoBack()) webView.goBack();  
else super.onBackPressed();  
}
```

## 1.5 Описание интерфейса приложения

### 1.5.1 “MainActivity”

Интерфейс “MainActivity” демонстрируется на рисунках 1 и 3. Содержит 2 поля для ввода (EditText) логина и пароля, кнопку (Button) “Войти”, одно поле с текстом (TextView): поле ошибки (изначально невидимо) и одно ImageView поле, выполняющее функцию дизайна. Элементы activity расположены в контейнере “ConstrainLayout”. При неправильном вводе выводится сообщение, показанное на рисунке 2. Листинг XML-файлов горизонтальной и вертикальной ориентаций находится в разделах 4.1 и 4.2.

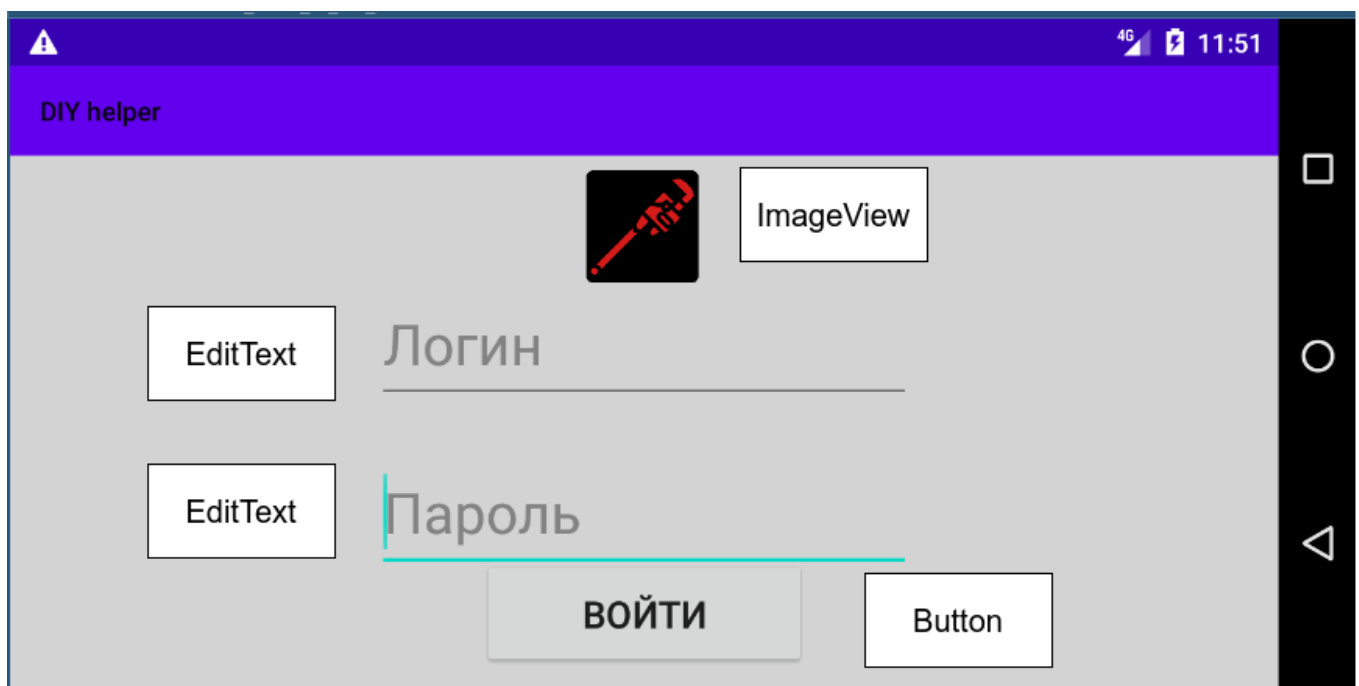


Рисунок 1 – “MainActivity” (горизонтальная ориентация)

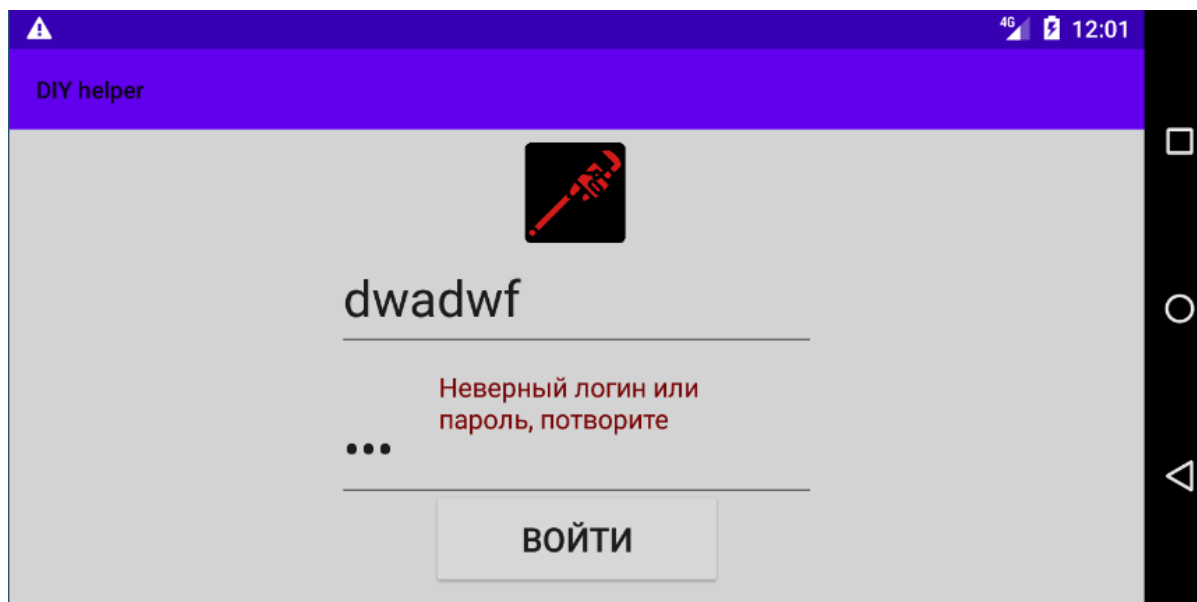


Рисунок 2 – “MainActivity” сообщение о неправильно введенных данных

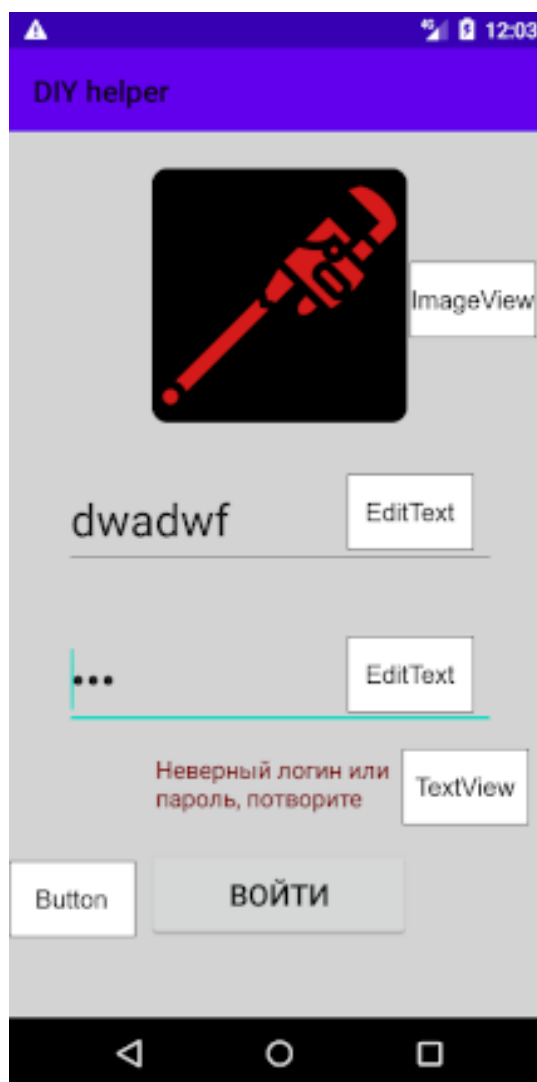


Рисунок 3 – “MainActivity” (вертикальная ориентация)

### 1.5.2 “WebActivity”

Интерфейс activity демонстрируется на рисунках 4 и 5. Содержит в себе поле “WebView”, веб-страница которая загружается в зависимости от имени пользователя. Элементы “WebActivity” расположены в “ConstarinLayout”. Листинг XML-файла разметки находится в разделе 4.3.

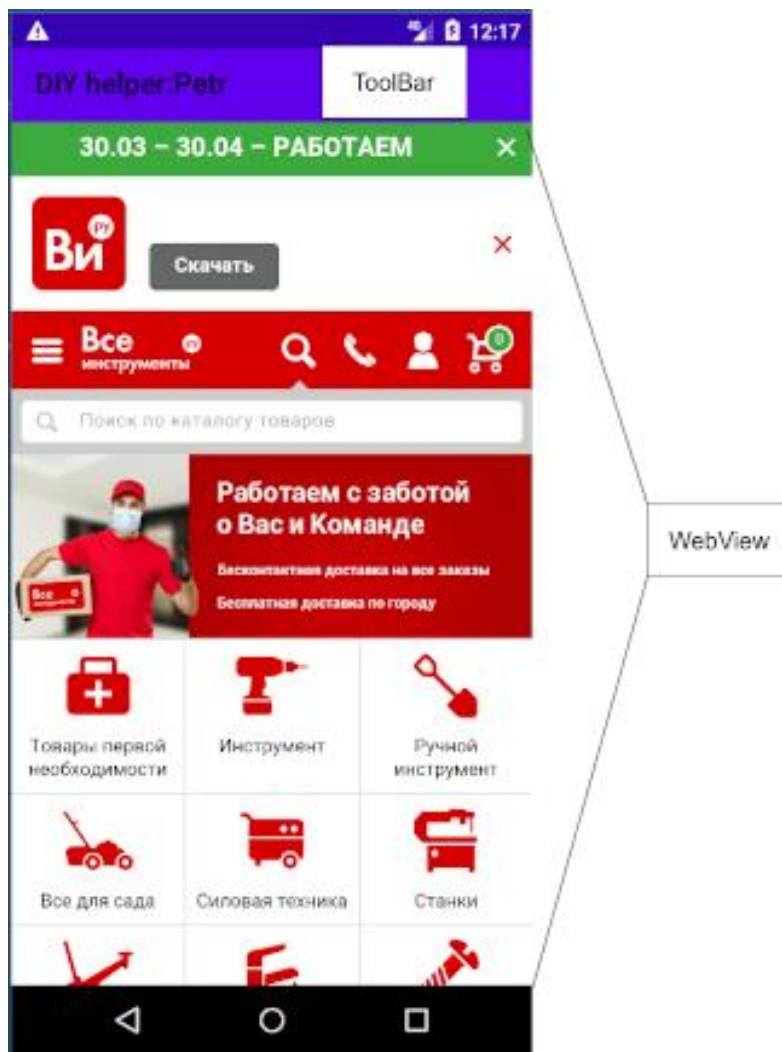


Рисунок 4 – “WebActivity” (вертикальная ориентация)

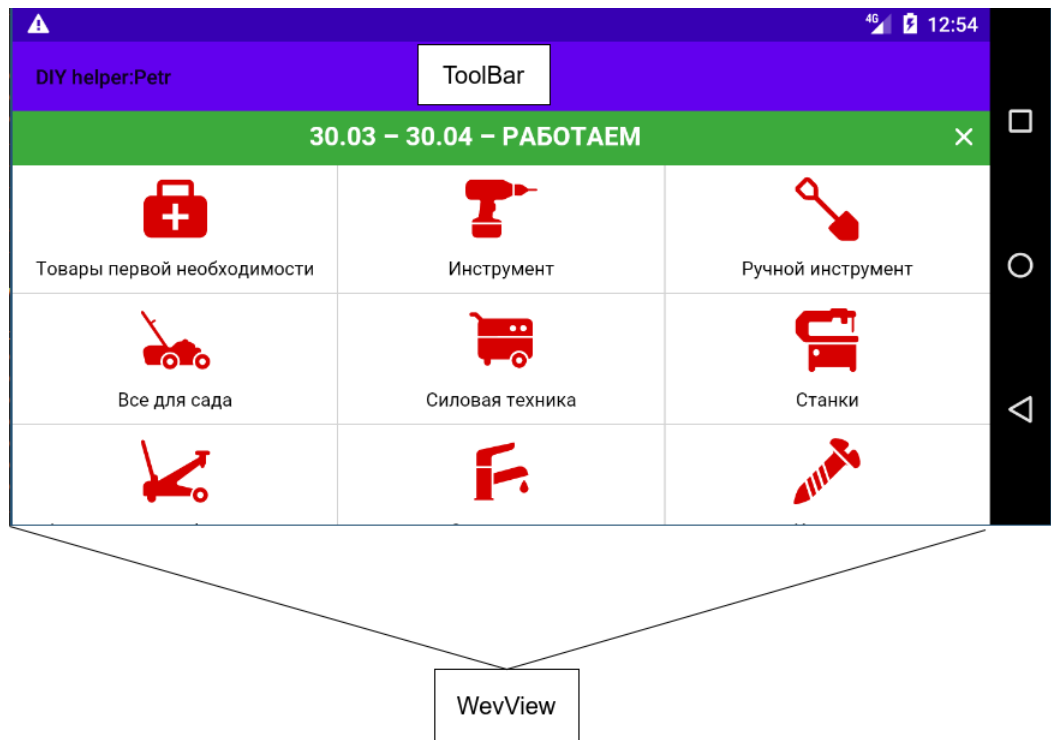


Рисунок 5 – “WebActivity” (горизонтальная ориентация)

### 1.5.3 Видео работы приложения

На видео демонстрируется работа всех функций программы - [https://youtu.be/FpwM\\_aSKoUM](https://youtu.be/FpwM_aSKoUM)

## **2 Администрирование**

### **2.1 Постановка задачи**

Необходимо изучить и внедрить в работу технологию конфигурационного контроля на базе Git:

- a. Установить git локально на рабочей станции.
- b. Перенести в git один из предыдущих проектов
- c. Создать учетную запись в GitLab
- d. Связать локальный репозиторий с проектом в Gitlab.
- e. Выполнить локально и отразить в gitlab: ветвление, слияние, создание запросов на слияние, принятие/отклонение запросов на слияние, разрешение конфликтов (локальные и gitlab скриншоты).

Необходимо изучить конфигурирование и подготовить платформу nginx для следующего проекта:

- a. Установить nginx
- b. Сконфигурировать сервер по умолчанию
- c. Сконфигурировать виртуальный сервер

1.3 Факультативно: Подготовить локальный Dockerfile для сборки окружения nginx в контейнер, продемонстрировать его запуск.

### **2.2 Описание выполненных работ**

#### **2.2.1 Установка технологии контроля версий на базе Git**

Установить Git можно на современные Unix-подобные операционные систем, такие как Windows, Mac OS, Linux. Пример установки системы контроля версий на OS Linux можно видеть на Рисунке 1.

```

[frizik@frizik ~]$ sudo pacman -S git
[sudo] пароль для frizik:
разрешение зависимостей...
проверка конфликтов...

Пакеты (1) git-2.26.2-1

Будет загружено: 6.22 MiB
Будет установлено: 38.36 MiB
Изменение размера: 0.93 MiB

:: Приступить к установке? [Y/n]
:: Получение пакетов...
   git-2.26.2-1-x86_64             6.2 MiB 1825 KiB/s 00:03 [#####] 100%
(1/1) проверка ключей [#####] 100%
(1/1) проверка целостности пакета [#####] 100%
(1/1) загрузка файлов пакетов [#####] 100%
(1/1) проверка конфликтов файлов [#####] 100%
(1/1) проверка доступного места [#####] 100%
:: Обработка изменений пакета...
(1/1) обновление git [#####] 100%
:: Запуск post-transaction hooks...
(1/3) Creating system user accounts...
(2/3) Reloading system manager configuration...
(3/3) Arming ConditionNeedsUpdate...
[frizik@frizik ~]$

```

Рисунок 1 — Установка системы контроля версий

## 2.2.2 Создание локального репозитория

Создание локального репозитория Git происходит командой.

```
$ git init
```

После чего, добавляем все файлы проекта на отслеживание в Git следующей командой (в папке находятся исходники практики по разработке андроид приложения).

```
$ git add .
```

Далее, делаем самый первый commit – фиксацию последних изменений в файлах. В нашем случае происходит закрепление начального состояния файлов. Пример выполнения этой операции можем видеть на Рисунке 2.

```

[frizik@frizik git_prac]$ git add .
[frizik@frizik git_prac]$ git commit -m "first_test"
[master (корневой коммит) 6661885] first_test
43 files changed, 1448 insertions(+)
create mode 100644 Practice.iml
create mode 100644 app/.gitignore
create mode 100644 app/app.iml
create mode 100644 app/build.gradle
create mode 100644 app/proguard-rules.pro
create mode 100644 app/src/androidTest/java/com/example/practice/ExampleInstrumentedTest.java
create mode 100644 app/src/main/AndroidManifest.xml
create mode 100644 app/src/main/java/com/example/practice/MainActivity.java
create mode 100644 app/src/main/java/com/example/practice/MyWebViewClient.java
create mode 100644 app/src/main/java/com/example/practice/web.java
create mode 100644 app/src/main/res/drawable-v24/ic_launcher_foreground.xml
create mode 100644 app/src/main/res/drawable/ic_launcher_background.xml
create mode 100644 app/src/main/res/drawable/monkey_wrench__1_.png
create mode 100644 app/src/main/res/layout-land/activity_main.xml
create mode 100644 app/src/main/res/layout-land/web.xml
create mode 100644 app/src/main/res/layout/activity_main.xml
create mode 100644 app/src/main/res/layout/web.xml
create mode 100644 app/src/main/res/mipmap-anydpi-v26/ic_launcher.xml
create mode 100644 app/src/main/res/mipmap-anydpi-v26/ic_launcher_round.xml
create mode 100644 app/src/main/res/mipmap-hdpi/ic_launcher.png
create mode 100644 app/src/main/res/mipmap-hdpi/ic_launcher_round.png
create mode 100644 app/src/main/res/mipmap-mdpi/ic_launcher.png
create mode 100644 app/src/main/res/mipmap-mdpi/ic_launcher_round.png
create mode 100644 app/src/main/res/mipmap-xhdpi/ic_launcher.png
create mode 100644 app/src/main/res/mipmap-xhdpi/ic_launcher_round.png
create mode 100644 app/src/main/res/mipmap-xxhdpi/ic_launcher.png
create mode 100644 app/src/main/res/mipmap-xxhdpi/ic_launcher_round.png
create mode 100644 app/src/main/res/mipmap-xxxhdpi/ic_launcher.png
create mode 100644 app/src/main/res/mipmap-xxxhdpi/ic_launcher_round.png
create mode 100644 app/src/main/res/values/arrays.xml
create mode 100644 app/src/main/res/values/colors.xml
create mode 100644 app/src/main/res/values/save.xml
create mode 100644 app/src/main/res/values/strings.xml
create mode 100644 app/src/main/res/values/styles.xml
create mode 100644 app/src/test/java/com/example/practice/ExampleUnitTest.java
create mode 100644 build.gradle
create mode 100644 gradle.properties
create mode 100644 gradle/wrapper/gradle-wrapper.jar
create mode 100644 gradle/wrapper/gradle-wrapper.properties
create mode 100755 gradlew
create mode 100644 gradlew.bat
create mode 100644 local.properties
create mode 100644 settings.gradle
[frizik@frizik git_prac]$

```

Рисунок 2 — Фиксация первоначальных файлов в репозитории

## 2.3 Создание и добавление удаленного репозитория

Создание удаленного репозитория происходит с помощью веб-инструмента GitLab. Процесс создания можно видеть на Рисунке 3.



**Blank project**   Create from template   Import project   CI/CD for external repo

**Project name**  
Praktika

**Project URL**   **Project slug**  
https://gitlab.com/FriZIk/   praktika

Want to house several dependent projects under the same namespace? [Create a group.](#)

**Project description (optional)**  
Description format

**Visibility Level** ⓘ

☒ **Private**  
Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.

☐ **Public**  
The project can be accessed without any authentication.

☐ **Initialize repository with a README**  
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Create project   Cancel

Рисунок 3 — Создание удаленного репозитория

Добавим удаленный репозиторий в наш локальный репозиторий и загрузим последние изменения на удаленный репозиторий. Сделаем это с помощью следующих команд:

```
$ git remote add origin https://gitlab.com/FriZIk/praktika.git  
$ git push origin master
```

Результат выполнения данных операций можно видеть на Рисунке 4 и на Рисунке 5.

```
create new repository  
[frizik@frizik git_prac]$ git remote add origin https://gitlab.com/FriZIk/praktika  
[frizik@frizik git_prac]$ git push origin master  
Username for 'https://gitlab.com': FriZIk  
Password for 'https://FriZIk@gitlab.com':  
warning: переадресация на https://gitlab.com/FriZIk/praktika.git/  
Перечисление объектов: 75, готово.  
Подсчет объектов: 100% (75/75), готово.  
При сжатии изменений используется до 4 потоков  
Сжатие объектов: 100% (60/60), готово.  
Запись объектов: 100% (75/75), 152.38 KiB | 6.35 MiB/s, готово.  
Total 75 (delta 5), reused 0 (delta 0), pack-reused 0  
To https://gitlab.com/FriZIk/praktika  
* [new branch]      master -> master  
[frizik@frizik git_prac]$
```

Рисунок 4 — Добавление удаленного репозитория с последующей загрузкой файлов на него

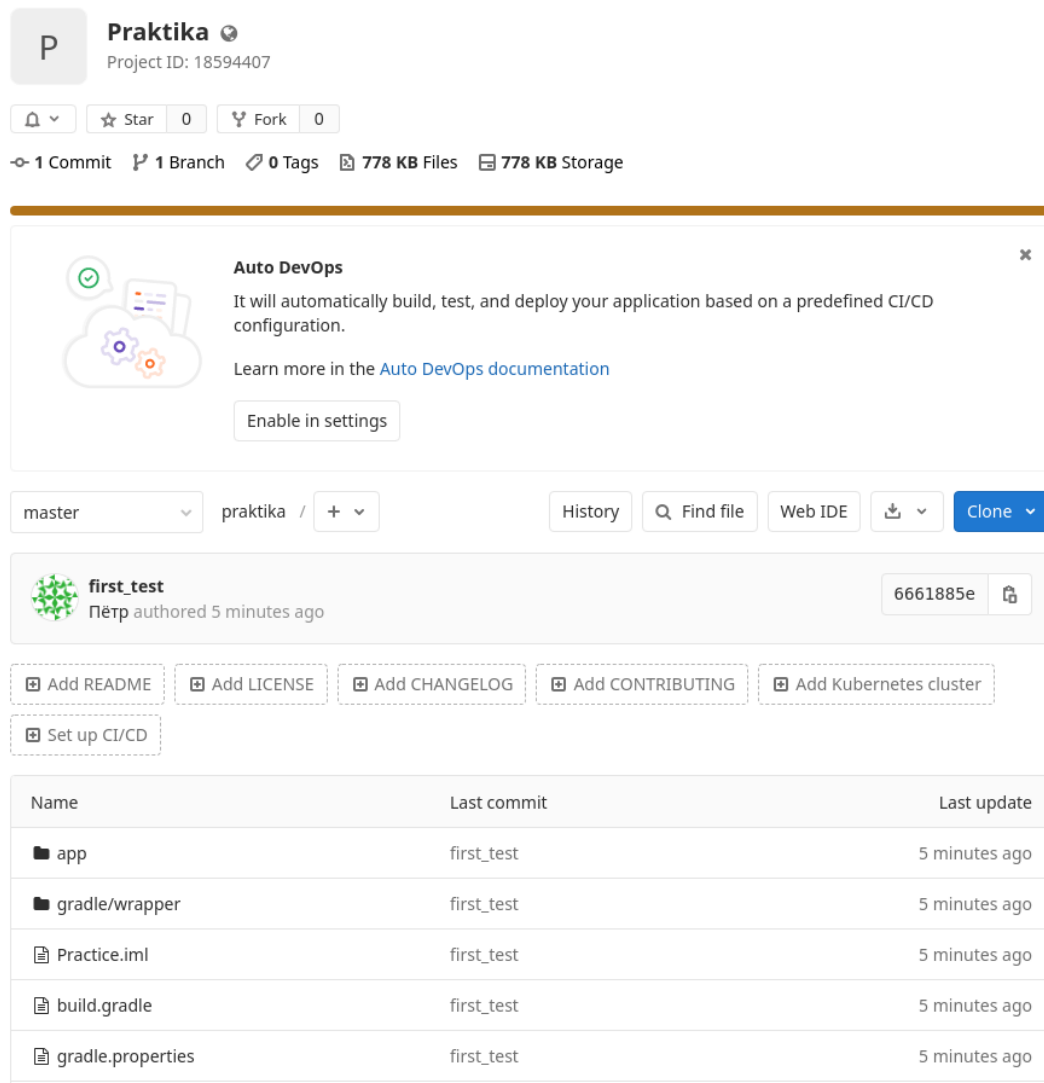


Рисунок 5 — Наш удаленный репозиторий после загрузки

## 2.4 Создание ветвлений

Дополнительные ветки проекта нужны для того, чтобы программисты могли вести совместную разработку проекта и при этом не мешать друг другу. При создании проекта, по умолчанию основной веткой является ветвь master.

Для создания новой ветки следует выполнить команду:

```
$ git branch test_branch(название для примера)
```

Для вывода списка доступных ветвлений следует выполнить команду:

```
$ git branch
```

Результат выполнения данных команд можно увидеть на Рисунке 6.

```
[frizik@frizik git_prac]$ git branch test_branch
[frizik@frizik git_prac]$ git branch
* master
  test_branch
[frizik@frizik git_prac]$
```

Рисунок 6 — Создание новой ветви

Для внесения изменений в новую ветвь, следует её выбрать. Это делается следующей командой.

```
$ git checkout test_branch(вместо test_branch следует вписать название
созданной ветки, если она называется иначе)
```

Далее, после изменения содержимого файлов проекта, мы можем посмотреть, какие были внесены изменения в эти файлы, относительно последней версии проекта. Делается это следующей командой.

```
$ git diff
```

Результат выполнения данной команды можно видеть на Рисунке 7.



```
$ git merge test_branch
```

Пример выполнения данной команды можно видеть на Рисунке 9.

```
[frizik@frizik PracticGit]$ git branch
* master
  test_branch
[frizik@frizik PracticGit]$ git merge test_branch
Обновление f505213..7b58afd
Fast-forward
 app/src/main/java/com/example/practice/MainActivity.java | 97 +-----
 1 file changed, 1 insertion(+), 96 deletions(-)
[frizik@frizik PracticGit]$
```

Рисунок 9 — Слияние ветви

В итоге можем видеть, что в ветке master появился новый commit, который ранее был в другой ветке (см. Рисунок 10, Рисунок 11).

```
[frizik@frizik PracticGit]$ git log
commit 7b58afd3819f27888461141f8e60a7c0a110e77a (HEAD -> master, origin/test_branch, test_branch)
Author: frizik <tester251100@gmail.com>
Date: Tue May 5 16:20:35 2020 +0300

    clean comments

commit f5052130f26229de2486da91e83ac5f4094a1062 (origin/master)
Author: frizik <tester251100@gmail.com>
Date: Tue May 5 15:59:40 2020 +0300

    frist test
[frizik@frizik PracticGit]$
```

Рисунок 10 — Результат слияния ветвления

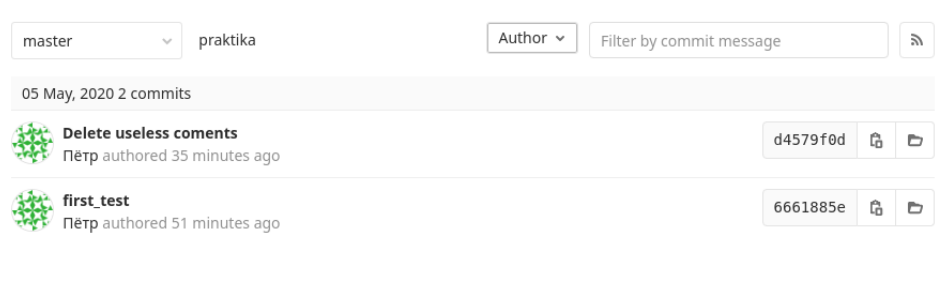


Рисунок 11 — Результат слияния, отраженный в GitLab

## 2.6 Создание запроса на слияние

Создание запроса на слияние происходит на странице репозитория в GitLab в разделе ветвлений (см. Рисунок 12 и Рисунок 13).

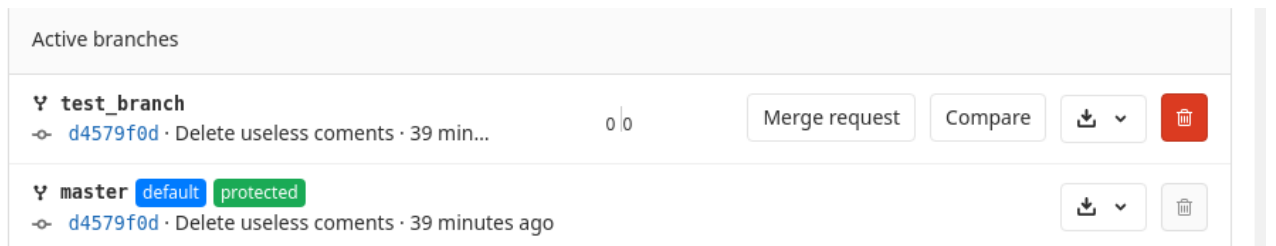


Рисунок 12 — Раздел ветвлений репозитория

**Edit Merge Request !2**

From `test` into `master` ▾

---

**Title**

Start the title with **WIP:** to prevent a **Work In Progress** merge request from being merged before it's ready.  
Add [description templates](#) to help your contributors communicate effectively!

**Description**

**Write** Preview

Describe the goal of the changes and what reviewers should be aware of.

Markdown is supported [Attach a file](#)

---

**Assignee**  [Assign to me](#)

**Milestone**

**Labels**

Рисунок 13 — Запрос на слияние

Принятие или отклонение запроса на слияние происходит в разделе Merge Requests (см. Рисунок 14).

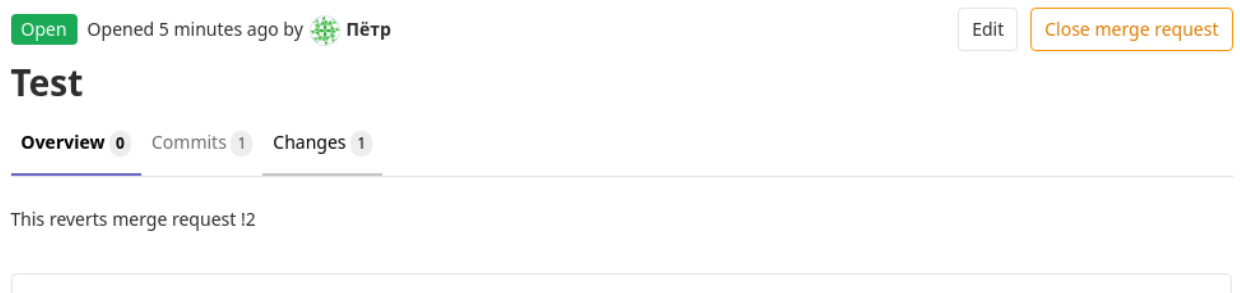


Рисунок 14 — Интерфейс принятия и отклонения запросов на слияние

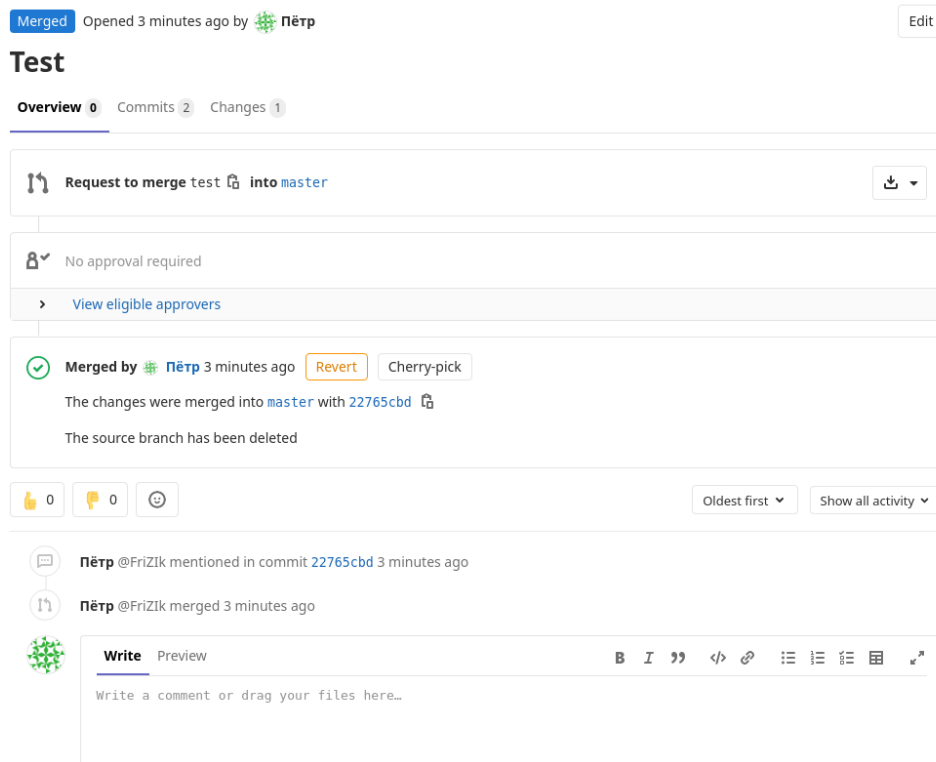


Рисунок 15 — Принятие запроса на слияние

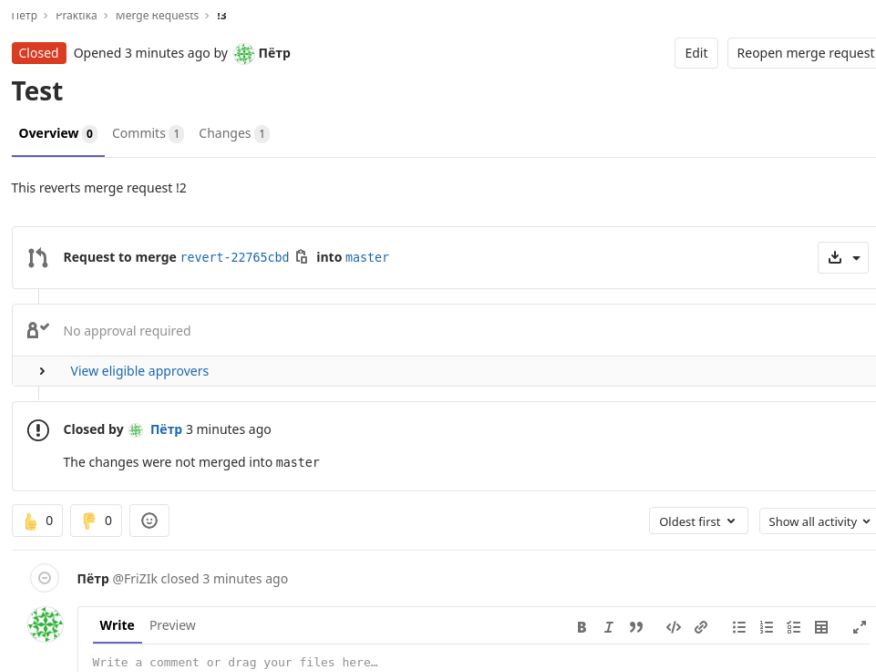


Рисунок 16 — Отклонение запроса на слияние (возврат)

## 2.7 Разрешение конфликтов локально

Конфликты происходят при объединении ветвлений, если в них было произведено изменения одного и того же файла.

Делаем новую ветвь, в которой вносим следующие изменения (см. Рисунок 17).

```
[frizik@frizik git_prac]$ git branch
* conflict
  master
  test
  test_branch
[frizik@frizik git_prac]$ git status
На ветке conflict
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>..», чтобы добавить файл в индекс)
  (use "git restore <file>.." to discard changes in working directory)
    изменено:   app/src/main/java/com/example/practice/MainActivity.java

нет изменений добавленных для коммита
(используйте «git add» и/или «git commit -a»)
[frizik@frizik git_prac]$ git add .
[frizik@frizik git_prac]$ git commit -m "konfliktnaya situation"
[conflict 6fc481f] konfliktnaya situation
 1 file changed, 1 insertion(+), 2 deletions(-)
[frizik@frizik git_prac]$ git branch
* conflict
  master
  test
  test_branch
[frizik@frizik git_prac]$
```

Рисунок 17 — Изменения в новой ветви

После чего вносим дополнительных изменений в основную ветвь master (см. Рисунок 18).

```
[frizik@frizik git_prac]$ git checkout master
Переключено на ветку «master»
[frizik@frizik git_prac]$ git status
На ветке master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>..», чтобы добавить файл в индекс)
  (use "git restore <file>.." to discard changes in working directory)
    изменено:   app/src/main/java/com/example/practice/MainActivity.java

нет изменений добавленных для коммита
(используйте «git add» и/или «git commit -a»)
[frizik@frizik git_prac]$ git add .
[frizik@frizik git_prac]$ git commit -m "oy, sey4as vse break"
[master c219cce] oy, sey4as vse break
 1 file changed, 1 insertion(+), 1 deletion(-)
[frizik@frizik git_prac]$
```

Рисунок 18 — Изменения в основной ветви master

При попытке выполнить слияние, появится сообщение о конфликте (см. Рисунок 19).



```
[frizik@frizik git_prac]$ git branch
conflict
* master
test
test_branch
[frizik@frizik git_prac]$ git merge conflict
Автослияние app/src/main/java/com/example/practice/MainActivity.java
КОНФЛИКТ (содержимое): Конфликт слияния в app/src/main/java/com/example/practice/MainActivity.java
Не удалось провести автоматическое слияние; исправьте конфликты и сделайте коммит результата.
[frizik@frizik git_prac]$
```

Рисунок 19 — Неудачная попытка провести слияние

В итоге выполнения данной операции, в наш изменяемый файл внесутся изменения из двух конфликтующих ветвлений следующим образом (см. Рисунок 20).

```

    if(IndexOfCoincidence == -100)
    {
        Log.i(Tag, "Неверный логин или пароль!");
        ErrorText.setVisibility(View.VISIBLE);
    }
    else
    {
        if (PasswordsList[IndexOfCoincidence].equals(Password))
        {
            ErrorText.setVisibility(View.INVISIBLE);
            Log.i(Tag, "Выполнен вход! Запускаю следующий активити");
            //Intent intent = new Intent(MainActivity.this, web.class);
            Intent intent = new Intent(MainActivity.this, web.class);
            intent.putExtra("Index", IndexOfCoincidence);
            startActivity(intent);
            ErrorText.setVisibility(View.INVISIBLE);
            LoginText.setText("");
            PasswordText.setText("");
        }
        else
        {
            Log.i(Tag, "Неверный логин или пароль!");
            ErrorText.setVisibility(View.VISIBLE);
        }
    }
}

Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<<< HEAD (Current Change)
}
/*BekBeKbEK */
=====
}
>>>>>>> conflict (Incoming Change)

```

Рисунок 20 — Конфликты ветвлений в файле программы

Далее следует исправить файлы и сделать заключительный коммит, закрепляющий результат слияния ветвлений (см. Рисунок 21).

```

[frizik@frizik git_prac]$ git status
На ветке master
У вас есть не слитые пути.
(разрешите конфликты, затем запустите «git commit»)
(используйте «git merge --abort», чтобы остановить операцию слияния)

Не слитые пути:
(используйте «git add <файл>», чтобы пометить разрешение конфликта)
оба изменены:  app/src/main/java/com/example/practice/MainActivity.java

нет изменений добавленных для коммита
(используйте «git add» и/или «git commit -a»)
[frizik@frizik git_prac]$ git add -A
[frizik@frizik git_prac]$ git commit -a
[master 1697521] Merge branch 'conflict'
[frizik@frizik git_prac]$ git branch
conflict
* master
test
test_branch
[frizik@frizik git_prac]$ git log
commit 1697521528c15c9adff3977fac76afdd3521843c (HEAD -> master)
Merge: c219cce 6fc481f
Author: frizik <tester251100@gmail.com>
Date: Tue May 5 23:40:58 2020 +0300

    Merge branch 'conflict'

commit c219cce14a2cd9659109bf783d091e3123455bd7
Author: frizik <tester251100@gmail.com>
Date: Tue May 5 23:34:46 2020 +0300

    oy, sey4as vse break

commit 6fc481f6250e585d0b8316c908fb168d34d45a4e (conflict)
Author: frizik <tester251100@gmail.com>
Date: Tue May 5 19:06:49 2020 +0300

    konfliktnaya situation

```

Рисунок 21 —Результат слияния ветвлений

## 2.8 Разрешение конфликтов на GitLab

Конфликт может произойти после принятия запроса на слияние (см. Рисунок 22).

Open

Opened 3 minutes ago by Пётр

EditClose merge request

InlineSide-by-side

app/src/main/AndroidManifest.xml

Interactive modeEdit inlineView file @b95071f

24	24	@@ -24,27 +24,28 @@
25	25	</application>
26	26	</manifest>
		HEAD//our changes
27		/*Здравствуйте !!!*
	27	/* Здравствуйте !!!*/
		origin//their changes

### Resolve conflicts on source branch

You can resolve the merge conflict using either the Interactive mode, by choosing Use Ours or Use Theirs buttons, or by editing the files directly. Commit these changes into `test_merge_gitlab`

### Commit message

Merge branch 'master' into 'test\_merge\_gitlab'

# Conflicts:  
# app/src/main/AndroidManifest.xml

Commit to source branchCancel

Рисунок 22 — Разрешение конфликта на GitLab

## 2.9 Конфигурация сервера по умолчанию

После скачивания Nginx, его можно запустить с помощью команды:

```
$ sudo systemctl start nginx
```

После этого действия запустится сервер по умолчанию, сконфигурированный в файле `/etc/nginx/nginx.conf` (Приложение А.1). При обращении к локальному IP адресу (127.0.0.1, localhost) будет доступна вступительная страница Nginx (см. Рисунок 23).

# Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

Рисунок 23 — Страница дефолтного сервера

## 2.10 Конфигурация виртуального сервера

Конфигурация виртуального сервера производится путем написания конфигурационного файла в директории `/etc/nginx/nginx.conf`. Прописываем в нём поля: `server_name` для обозначения имени сервера, `listen 80` для указания какой порт будем слушать, `root` для указания адреса директории с файлами сайта(корневой раздел - `root`) , `index` для определения html страницы сервера.

В случае правильности синтаксиса перезапускаем `nginx` следующей командой:

```
$ sudo systemctl reload nginx (либо restart, но не рекомендуется)
```

После указанных действий, добавляем имя нашего сервера в список серверов, который описан в файле `/etc/hosts` (см. Приложение А.2). Теперь, мы можем подключаться к виртуальному серверу (я разместил туда свой старый проект из школы кода) (см. Рисунок 24).

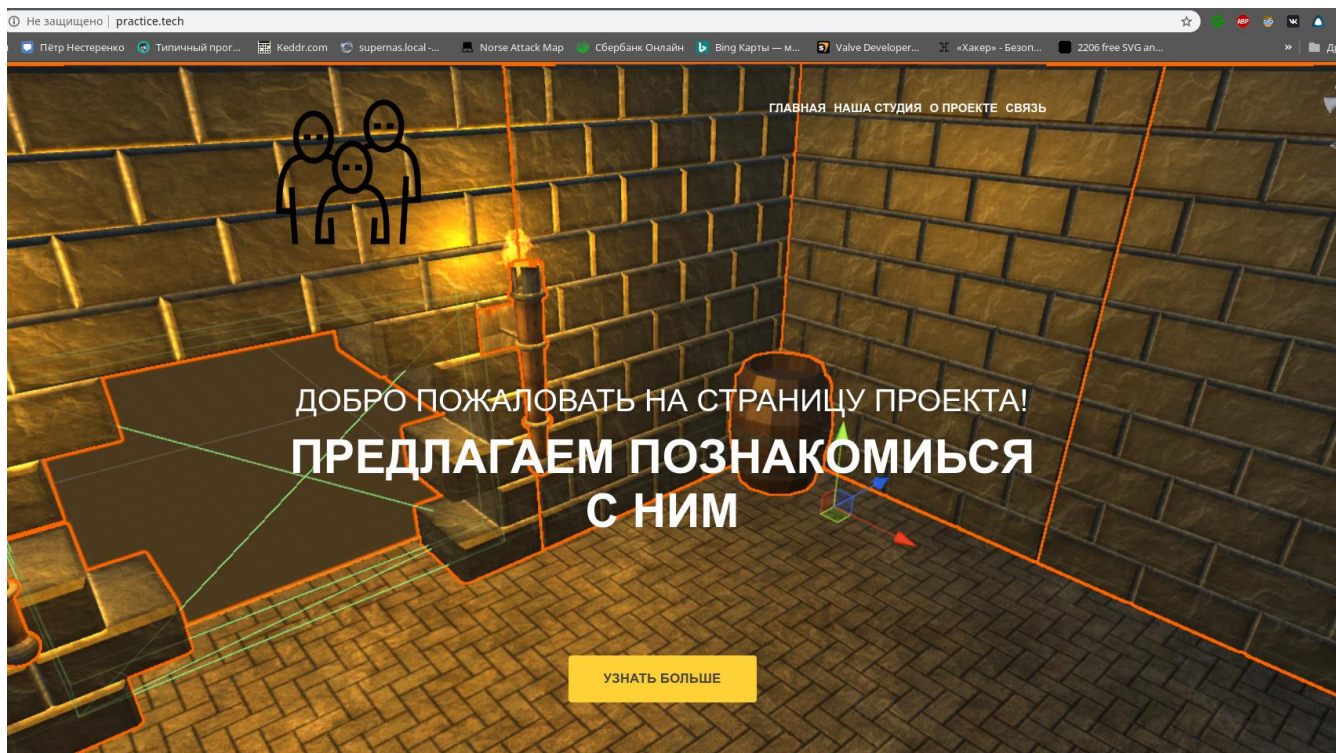


Рисунок 24 — Подключение к виртуальному серверу

## 2.11 Ссылки

Ссылка на репозиторий:

<https://gitlab.com/FriZIk/praktika>

Ссылка на видео с настройкой и тестами nginx:

<https://youtu.be/jlyDIm8cs9Q>

## 2.12 Заключение

В ходе практики была реализована установка системы контроля git и работа с ней, установка и настройка виртуального сервера nginx.

Для выполнения поставленной задачи использовалась операционная система GNU/Linux, система контроля версий Git, веб-сервер nginx.

В ходе выполнения задания:

- Узнал как использовать систему контроля версий Git
- Научился ветвлению проектов в Git при разработке ПО

- Познакомились со методами разрешения конфликтов, связанных со слиянием нескольких веток

- Узнал как настраивать виртуальный сервер на базе nginx

- Получил базовые навыки работы с Linux

## 3 Веб-разработка

### 3.1 Постановка задачи

Необходимо разработать одностраничный сайт визитку. Сайт должен содержать 3 блока, а именно: «Обо мне», «Портфолио», «Контакты». Должна быть соблюдена цветовая гамма. Сайт должен быть адаптивным и кроссбраузерным. Должно быть реализовано меню.

1. Первый блок сайта «Обо мне» должен содержать общую информацию о владельце сайта.

2. Второй блок сайта «Портфолио» должен содержать расширенную информацию о интересах, умениях и тд.

3. Третья страница «Контакты» должна содержать ссылки на социальные сети для связи с владельцем.

Тематика, дизайн, расположение элементов выбирается студентом самостоятельно и не должно повторяться.

### 3.2 Описание разработки сайта

#### 3.2.1 Выбор цветовой палитры

Выбор цветовой палитры был осуществлён с помощью сервиса [colorscheme.ru](https://colorscheme.ru). Выбранную палитру цветов можно увидеть на Рисунке 1.



Рисунок 22 — Цветовая палитра сайта



### 3.2.2 Создание макета сайта

Создание макетов страниц было осуществлено с помощью сервиса Figma (Рисунок 2).

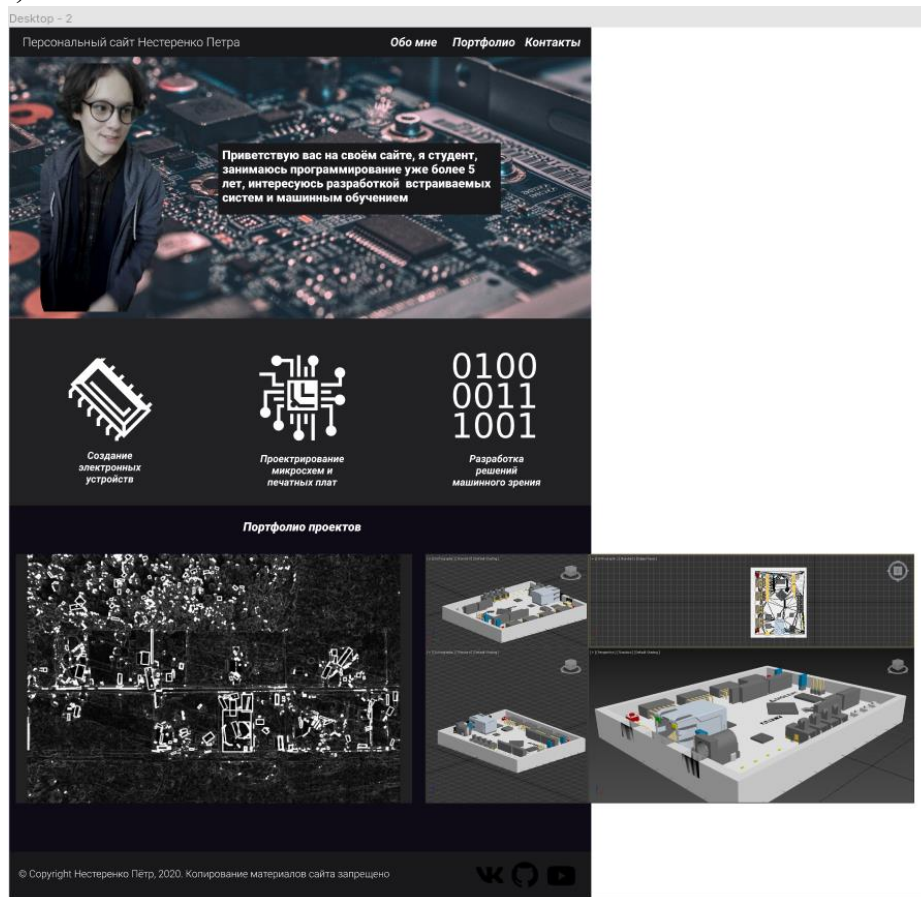


Рисунок 2 — Макет страницы сайта

### 3.2.3 Подготовка графического материала для сайта

Для использования графического материала, такого как изображения, фон, на сайте их первым делом нужно подготовить. Это производится сжатием изображения или конвертирование его формат JPG.

Для данного сайта все изображения были сжаты до минимально комфортных размеров, а также фон сайта был конвертирован в формат JPG.

### 3.2.4 Создание Header страницы.

Header страницы содержит в себе информацию о владельце и меню с помощью которого можно перемещаться между блоками сайта. (см. Рисунок 3).



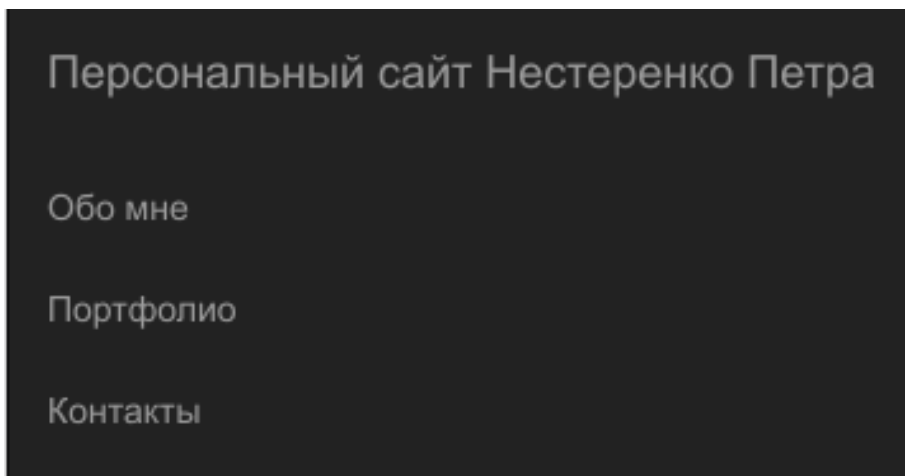
### Рисунок 3 – Header страницы

Название было выполнено с использованием библиотеки bootstrap и класса navbar.

```
<div class="navbar-header">  
  <a class="navbar-brand" href="#">Персональный сайт  
  Нестеренко Петра</a>  
</div>
```

Меню было также реализовано с помощью класса `<nav>`. а также тега `<ul>` служащего для создания маркированных списков, в нашем случае списка меню. Вёрстка адаптивна, внешний вид меню меняется в зависимости от разрешения (см. Рисунок 4).

```
<nav id="menu">  
  <ul class="nav navbar-nav navbar-right">  
    <li class="active"><a href="#main">Главная</a></li>  
    <li><a href="#aboutme">Обо мне</a></li>  
    <li><a href="#Portfolio">Портфолио</a></li>  
    <li><a href="#Contact">Контакты</a></li>  
  </ul>  
</nav>
```



### Рисунок 4 – Мобильное меню

Цвет верхней панели был сделан с помощью следующих CSS атрибутов:

```
.TopBar
{
    margin: 0;
    background-color:#19191A;
    border-radius:0;
    margin:0;
    border-radius: 0;
}
```

### 3.2.5 Создание Footer страницы

В Footer страницы описана основная контактная информация клуба, а также ссылки на другие страницы сайта (см. Рисунок 4).

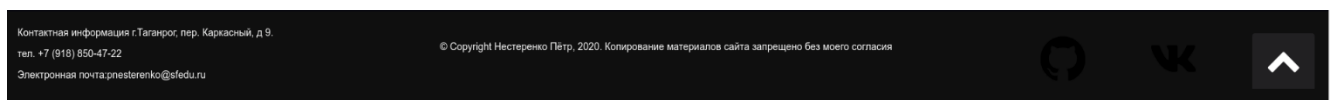


Рисунок 5 – Footer страницы

Основной текст с информацией выполнен с помощью тега абзаца `<p>`

```
<div class="col-md-3" style="padding:20px">
<p>Контактная информация г.Таганрог, пер. Каркасный, д 9.</p>
<p>тел. +7 (918) 850-47-22</p>
<p>Электронная почта:pnesterenko@sfedu.ru</p>
</div>
```

Цвет Footer был придан при помощи стилей:

```
#Contact
{
    background-color:#101010;
    width: 100%;
    width: 100%;
    position:relative;
}
```

Подробную реализацию Footer можно увидеть в **Приложении Б.1** и в **Приложении В.1**.

### 3.3 Вёрстка основной части страницы

Основная часть страницы выполнена в несколько блоков разных цветов, иногда фоном выступает изображение. Каждый такой блок был сделан при помощи тега <div>, а цвет им был придан при помощи стилей header.css.

Ниже приведён пример блока сайта.

```
<div id="Info" class="jumbotron jumbotron-fluid row"
style="color:#FFFFFF">
    <div class="col-md-2"></div>
    <div class="col-md-3" style="margin:0">
        
    </div>
    <div class="col-md-6" style="margin-top: 100px;background-color:
#222123;text-align: center;margin-bottom: 0 auto;">
        <h1 class="display-4" >Приветствую вас на своём сайте</h1>
        <p style="text-align:center"> Изучаю программирование и IT уже 5 лет,
интересуюсь разработкой встраиваемых систем и машинным обучением. Сейчас
обучаюсь на кафедре математического обеспечения и применения ЭВМ, ИКТИБ
ЮФУ (бывший ТРТУ/ТРТИ) . Люблю решать сложные задачи и учиться новому.</p>
    </div>
</div>
```

Скриншоты страниц сайта можно видеть в Приложении А.1, А.2, А.3,А4.

### 3.4 Адаптация сайта под мобильные устройства.

Адаптация проводится средствами библиотеки Bootstrap. Содержимое сайта подстраивается при изменении разрешения экрана.

Пример адаптации сайта приведён на Рисунке 6.

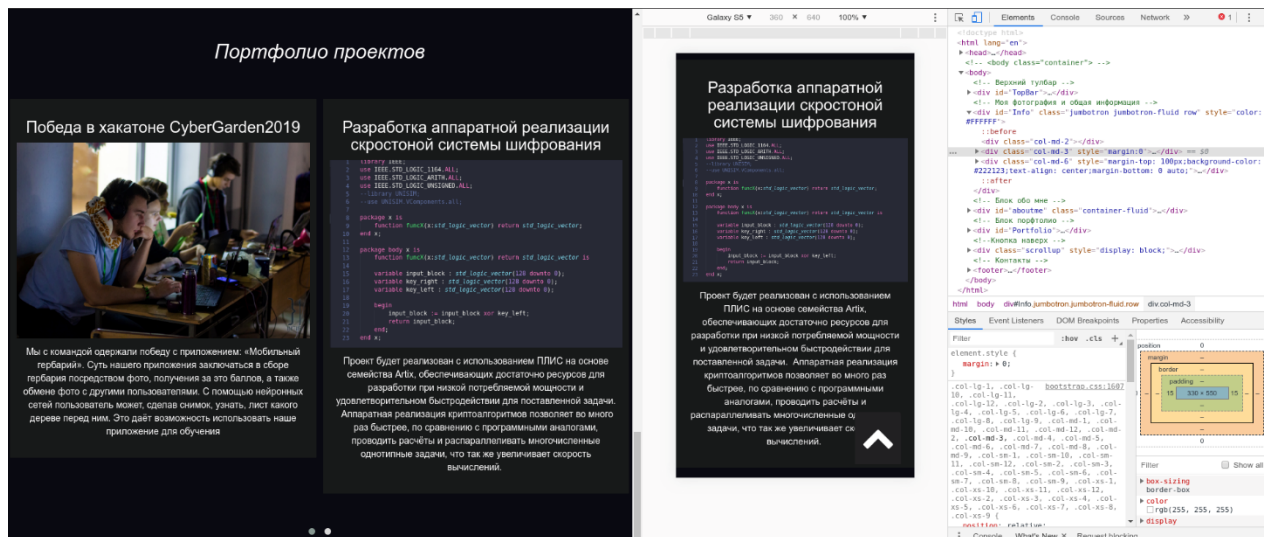


Рисунок 6 – Адаптация сайта для мобильных устройств

### 3.5 Именованное URL страниц

Для подключения к сайту через URL был сконфигурирован локальный сервер nginx. URL основной страницы: <http://practice.tech/> Кроссбраузерность. Скриншоты сайта, сделанные с разных браузеров, приведены в Приложении Г.

Все javascript-скрипты приведены в приложении Г.1, Г.2, Г.3. Сертификат о прохождении курса на Stepik приведён в приложении Д1.

### 3.6 Заключение

В ходе практики были выполнены все требования к разрабатываемому веб-сайту. Оно соответствует всем аспектам технического задания.

Для выполнения поставленной задачи использовался сервер nginx, текстовый редактор VisualStudio Code. Проект написан на языке разметки гипертекста HTML, стилей CSS и языка программирования JavaScript с использованием фреймворков bootstrap и OwlCarousel.

В ходе выполнения задания научился:

- Разработке веб-сайтов с помощью HTML и CSS
- Адаптивной и кроссбраузерной вёрстке.

## 4 Знакомство с 1С

### 4.1 Постановка задачи

В ходе практики необходимо:

1. Изучить видео инструкции.
2. Придумать тематику системы, 2 справочника и 1 документ, отражающий учетное событие с объектами справочника (если событие окажется не учетным - отправлю работу на переделку). Один справочник - иерархический, второй - нет.
3. Создать в конфигураторе 2 справочника и документ. В каждом справочнике должно быть не менее 1 дополнительного реквизита (поля). В иерархическом справочнике - как минимум 2 уровня, 6 групп, в каждой группе по 2 элемента. В одноуровневом справочнике - не менее 3 элементов. Документ должен содержать простой реквизит и табличную часть с реквизитами. В табличной части должно быть не менее 3 связанных формулой реквизитов. При изменении 2 реквизитов пересчитывается значение третьего. Ввести не менее 3 документов и заполнить их данными.

### 4.2 Создание справочников

Создали подсистему для нашего предприятия - Рисунок 1.

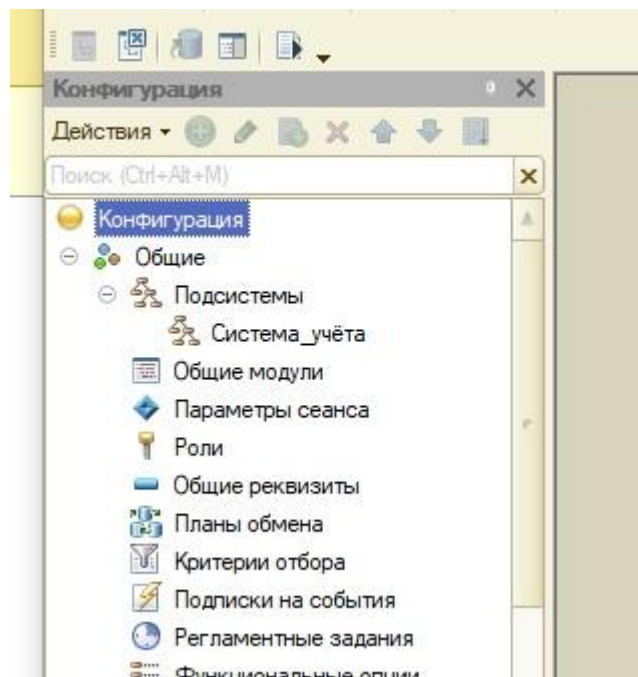


Рисунок 1 – Иерархия проекта

В результате проведённых действий внешний вид “предприятия изменился”, что видно на Рисунке 2.

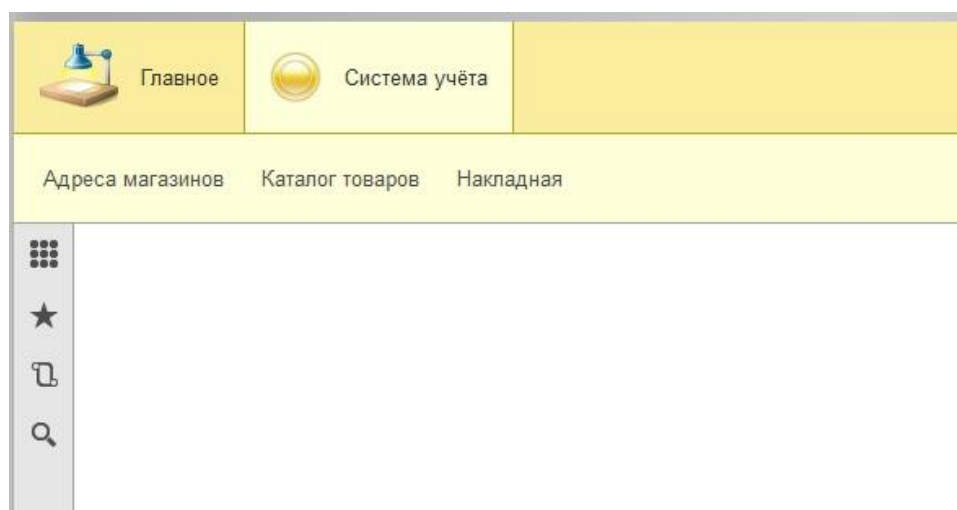


Рисунок 2 – Изменения GUI

Создаем дополнительные атрибуты, такие как адрес магазина Рисунок 3. Далее заполняем адреса данными Рисунок 4.

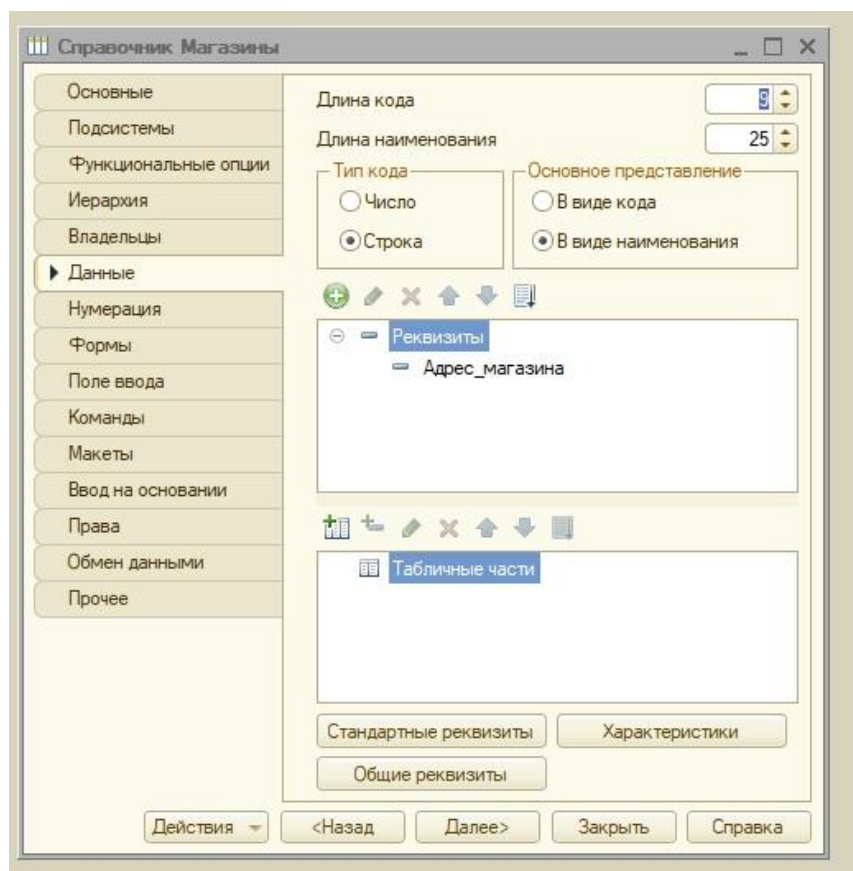


Рисунок 3 – Создание реквизита с адресом магазина

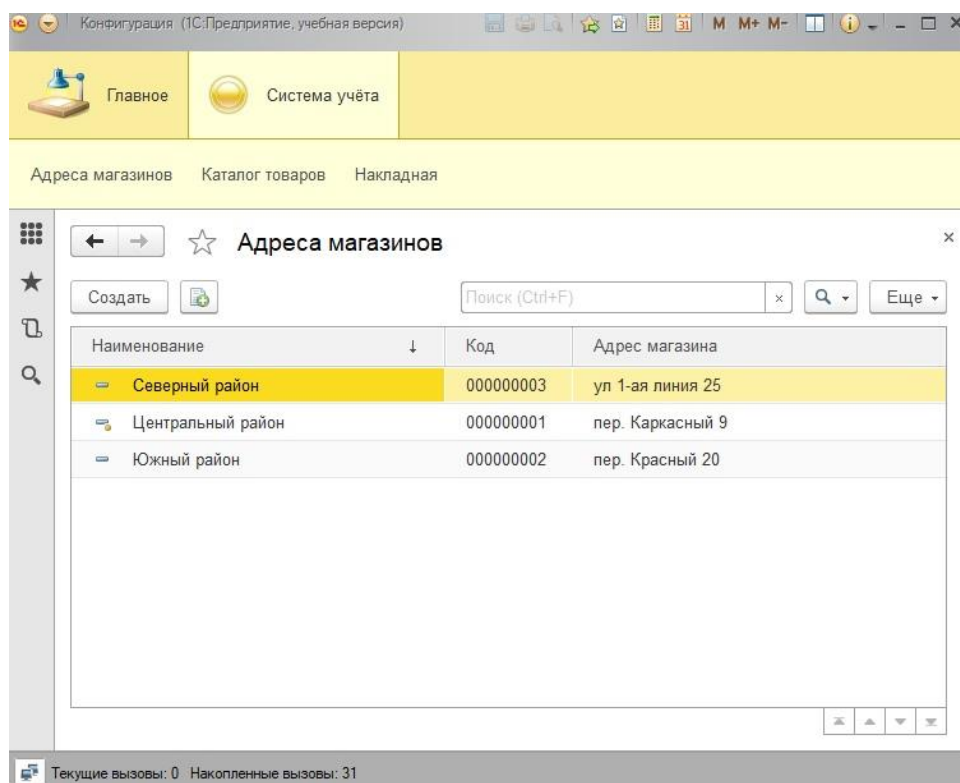


Рисунок 4 – Добавление данных в созданные поля

### 4.3 Создание накладной

После создания иерархического справочника содержащего названия инструментов и их цену, а так же штрих код, заполняем поля данными, и создаем накладную. Процесс создания представлен на рисунке 5.

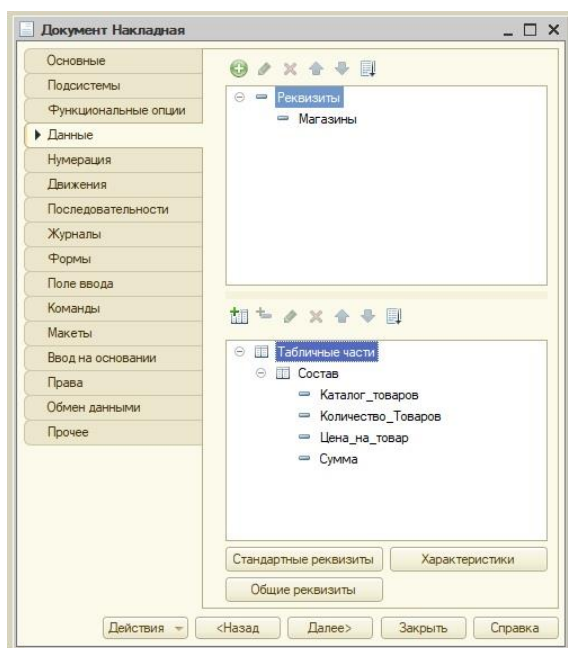


Рисунок 5 – Создание накладной

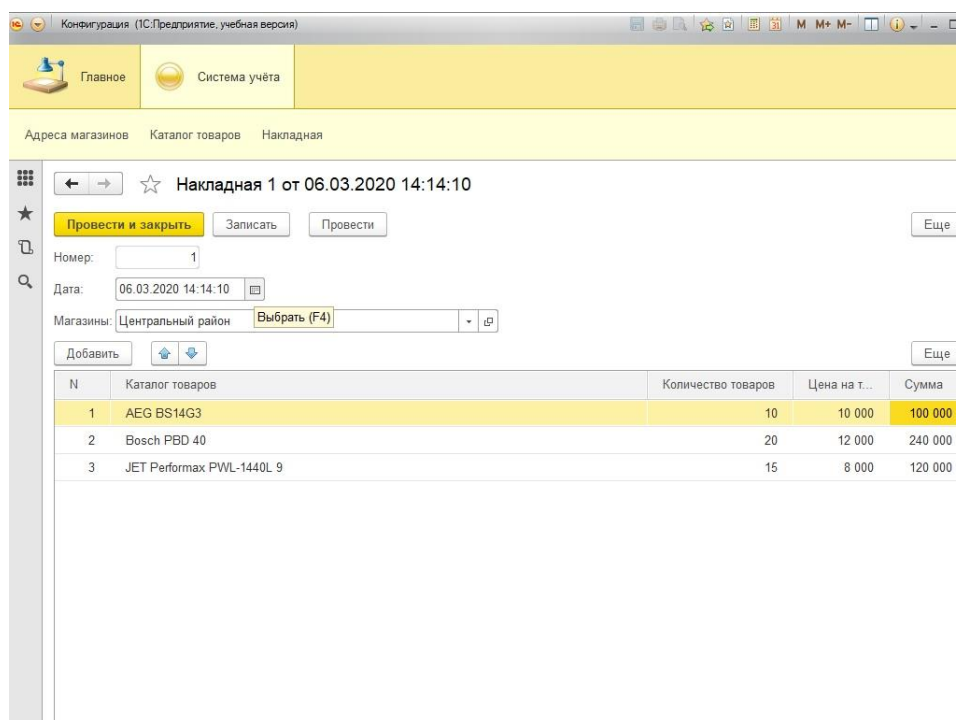
#### 4.4 Создание процедур и связанных формул реквизитов

Для того чтобы связать компоненты справочников и таблиц друг с другом, необходимо описать специальные программы-процедуры. Процесс создания подобной формулы представлен на рисунке 6.

```
«НаКлиенте»
Процедура СоставПриИзменении(Элемент)
    ТекущаяСтрока = Элементы.Состав.ТекущиеДанные;
    ТекущаяСтрока.Сумма = ТекущаяСтрока.Количество_Товаров * ТекущаяСтрока.Цена_на_товар;
КонецПроцедуры
```

Рисунок 6 – Листинг процедуры

Данная процедура высчитывает суммарную стоимость товаров, результат работы данной программы представлен на рисунке 7.



N	Каталог товаров	Количество товаров	Цена на т...	Сумма
1	AEG BS14G3	10	10 000	100 000
2	Bosch PBD 40	20	12 000	240 000
3	JET Performax PWL-1440L 9	15	8 000	120 000

Рисунок 6 – Листинг процедуры

#### 4.5 Заключение

В ходе выполнения задачи модуля “Знакомство с 1С”, задача была успешно выполнена. Мы научились создавать различные справочники(иерархические и нет) и таблицы в системе 1С, познакомились с языком описания формул 1С и успешно написали процедуру для нахождения общей стоимости товаров.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Официальная документация по Nginx [Электронный Ресурс] URL: <https://nginx.org/ru/docs/>
- 2 Документация Git [Электронный Ресурс] URL: <https://git-scm.com/doc>
- 3 HTML справочник [Электронный Ресурс] URL: <http://htmlbook.ru/>
- 4 Веб-разработка для начинающих: HTML и CSS [Электронный Курс]  
URL: <https://stepik.org/course/38218>
- 5 Introduction Bootstrap. [Электронный ресурс]. <https://bootstrap-4.ru>

## Приложение

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#D3D3D3"
tools:context=".MainActivity">

<Button
    android:id="@+id/LoginButton"
    android:layout_width="175dp"
    android:layout_height="61dp"
    android:layout_marginStart="92dp"
    android:layout_marginEnd="93dp"
    android:layout_marginBottom="12dp"
    android:onClick="LoginButton"
    android:text="Войти"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.501"
    app:layout_constraintStart_toStartOf="parent" />

<EditText
    android:id="@+id/PasswordText"
    android:layout_width="287dp"
    android:layout_height="66dp"
    android:layout_marginStart="37dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="36dp"
    android:ems="10"
    android:hint="Пароль"
    android:inputType="textPassword"
    android:textSize="30sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/LoginText" />

<EditText
    android:id="@+id/LoginText"
    android:layout_width="287dp"
    android:layout_height="66dp"
    android:layout_marginStart="37dp"
    android:layout_marginEnd="36dp"
    android:ems="10"
    android:hint="Логин"
    android:inputType="textEmailAddress"
```

```
android:textSize="30sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.498"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/Emblem" />
```

```
<ImageView
    android:id="@+id/Emblem"
    android:layout_width="97dp"
    android:layout_height="60dp"
    android:layout_marginStart="95dp"
    android:layout_marginTop="56dp"
    android:layout_marginEnd="95dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/monkey_wrench__1_" />
```

```
<TextView
    android:id="@+id/ErrorMessage"
    android:layout_width="165dp"
    android:layout_height="40dp"
    android:layout_marginStart="151dp"
    android:layout_marginTop="9dp"
    android:layout_marginEnd="151dp"
    android:layout_marginBottom="100dp"
    android:text="Неверный логин или пароль, потворите попытку"
    android:textColor="#780404"
    android:textSize="16sp"
    android:visibility="invisible"
    app:layout_constraintBottom_toTopOf="@+id/LoginButton"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/LoginText"
    app:layout_constraintVertical_bias="0.0" />
```

```
<androidx.constraintlayout.widget.Barrier
    android:id="@+id/barrier"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:barrierDirection="top"
    tools:layout_editor_absoluteY="658dp" />
```

```
<androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar_main"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:background="?attr/colorPrimary"
    android:minHeight="?attr/actionBarSize"
    android:theme="?attr/actionBarTheme"
    app:layout_constraintEnd_toEndOf="parent"
```

```

        app:layout_constraintHorizontal_bias="0.47"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="1dp" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

## activity\_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#D3D3D3"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/LoginButton"
        android:layout_width="175dp"
        android:layout_height="61dp"
        android:layout_marginStart="92dp"
        android:layout_marginEnd="93dp"
        android:layout_marginBottom="52dp"
        android:onClick="LoginButton"
        android:text="Войти"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

3
    <EditText
        android:id="@+id/PasswordText"
        android:layout_width="287dp"
        android:layout_height="66dp"
        android:layout_marginStart="37dp"
        android:layout_marginTop="41dp"
        android:layout_marginEnd="36dp"
        android:ems="10"
        android:hint="Пароль"
        android:inputType="textPassword"
        android:textSize="30sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/LoginText" />

    <EditText
        android:id="@+id/LoginText"
        android:layout_width="287dp"
        android:layout_height="66dp"
        android:layout_marginStart="37dp"
        android:layout_marginTop="32dp"

```

```
android:layout_marginEnd="36dp"
android:ems="10"
android:hint="Логин"
android:inputType="textEmailAddress"
android:textSize="30sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.509"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/Emblem" />
```

```
<ImageView
    android:id="@+id/Emblem"
    android:layout_width="170dp"
    android:layout_height="170dp"
    android:layout_marginStart="95dp"
    android:layout_marginTop="80dp"
    android:layout_marginEnd="95dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.49"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/monkey_wrench__1_" />
```

```
<TextView
    android:id="@+id/ErrorMessage"
    android:layout_width="165dp"
    android:layout_height="40dp"
    android:layout_marginStart="151dp"
    android:layout_marginTop="52dp"
    android:layout_marginEnd="151dp"
    android:layout_marginBottom="60dp"
    android:text="Неверный логин или пароль, потворите попытку"
    android:textColor="#780404"
    android:textSize="16sp"
    android:visibility="invisible"
    app:layout_constraintBottom_toTopOf="@+id/LoginButton"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/PasswordText" />
```

```
<androidx.constraintlayout.widget.Barrier
    android:id="@+id/barrier"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:barrierDirection="top"
    tools:layout_editor_absoluteY="658dp" />
```

```
<androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar_main"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:background="?attr/colorPrimary"
```

```

        android:minHeight="?attr/actionBarSize"
        android:theme="?attr/actionBarTheme"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.47"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="1dp" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

## web.xml (land)

```

<?xml version="1.0" encoding="utf-8"?>
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/ConstLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <WebView
        android:id="@+id/Site"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/toolbar_web"
        app:layout_constraintVertical_bias="1.0">

    </WebView>

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar_web"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?attr/colorPrimary"
        android:minHeight="?attr/actionBarSize"
        android:theme="?attr/actionBarTheme"
        android:visibility="visible"
        app:layout_constraintBottom_toTopOf="@+id/Site"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

## web.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/ConstLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <WebView
        android:id="@+id/Site"
        android:layout_width="match_parent"
        android:layout_height="603dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/toolbar_web">

    </WebView>

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar_web"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?attr/colorPrimary"
        android:minHeight="?attr/actionBarSize"
        android:theme="?attr/actionBarTheme"
        android:visibility="visible"
        app:layout_constraintBottom_toTopOf="@+id/Site"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

## MainActivity

```

package com.example.practice;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.content.res.Resources;
import android.nfc.Tag;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.TextView;
import androidx.appcompat.widget.Toolbar;

public class MainActivity extends AppCompatActivity
{
    private TextView ErrorText,LoginText,PasswordText;

```

```

private static final String Tag = "Практика";
private String Login, Password;
private Toolbar toolbar;

@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    ErrorText = findViewById(R.id.ErrorMessage);
    LoginText = findViewById(R.id.LoginText);
    PasswordText = findViewById(R.id.PasswordText);

    toolbar = findViewById(R.id.toolbar_main);
    setSupportActionBar(toolbar);
    setTitle("DIY helper");
}

@Override
protected void onSaveInstanceState(Bundle outState)
{
    super.onSaveInstanceState(outState);
    outState.putString("Login", LoginText.getText().toString());
    outState.putString("Password", PasswordText.getText().toString());
    Log.i(Tag, LoginText.getText().toString());
}

@Override
protected void onRestoreInstanceState(Bundle savedInstanceState)
{
    super.onRestoreInstanceState(savedInstanceState);
    LoginText.setText(savedInstanceState.getString("Login"));
    PasswordText.setText(savedInstanceState.getString("Password"));
}

@Override
public void onBackPressed()
{
    super.onBackPressed();
}

public void LoginButton(View view) {
    Login = LoginText.getText().toString();
    Password = PasswordText.getText().toString();

    // Проверка на совпадение с одним из паролей/логинов
    int CountOfUsers = 3;
    String[] LoginsList = getResources().getStringArray(R.array.Logins);
    String[] PasswordsList = getResources().getStringArray(R.array.Passwords);
    int IndexOfCoincidence = -100, i;

```



```

for (i = 0; i < CountOfUsers; i++)
    if(LoginsList[i].equals(Login))break;

if(i < CountOfUsers)IndexOfCoincidence = i;

if(IndexOfCoincidence == -100)
{
    Log.i(Tag, "Неверный логин или пароль!");
    ErrorText.setVisibility(View.VISIBLE);
}
else
{
    if (PasswordsList[IndexOfCoincidence].equals>Password))
    {
        ErrorText.setVisibility(View.INVISIBLE);
        Log.i(Tag, "Выполнен вход! Запускаю следующий активити");
        //Intent intent = new Intent(MainActivity.this, web.class);
        Intent intent = new Intent(MainActivity.this, WebActivity.class);
        intent.putExtra("Index", IndexOfCoincidence);
        startActivity(intent);
        ErrorText.setVisibility(View.INVISIBLE);
        LoginText.setText("");
        PasswordText.setText("");
    }
    else
    {
        Log.i(Tag, "Неверный логин или пароль!");
        ErrorText.setVisibility(View.VISIBLE);
    }
}
}
}

```

## WebActivity

```

package com.example.practice;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.webkit.WebResourceRequest;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

public class WebActivity extends AppCompatActivity
{
    private static final String Tag = "Практика";

```

```

private WebView webView;
private Toolbar toolbar;
private String CurrentUrl = "";

@Override
protected void onCreate(@Nullable Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.web);
    toolbar = findViewById(R.id.toolbar_web);
    setSupportActionBar(toolbar);

    webView = findViewById(R.id.Site);
    webView.setWebViewClient(new WebViewClient());

    // Принимаем Индекс и по нему вызываем нужный сайт
    int txtName = getIntent().getIntExtra("Index", -1);
    String Sites[] = getResources().getStringArray(R.array.Sites);
    Log.i(Tag, Sites[txtName]);
    String Logins[] = getResources().getStringArray(R.array.Logins);
    setTitle("DIY helper:" + Logins[txtName]);
    webView.getSettings().setJavaScriptEnabled(true);

    if(savedInstanceState == null)
    {
        webView.loadUrl(Sites[txtName]);
        Log.i(Tag, Sites[txtName]);
    }
    else {
        webView.loadUrl(savedInstanceState.getString("Url"));
        Log.i(Tag, CurrentUrl);
    }
}

@Override
protected void onSaveInstanceState(Bundle outState)
{
    {
        outState.putString("Url", webView.getUrl());
        super.onSaveInstanceState(outState);
    }
}

@Override
public void onBackPressed()
{
    {
        if (webView.canGoBack()) webView.goBack();
        else super.onBackPressed();
    }
}
}

```

## arrays.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name = 'Passwords'>
        <item>251100</item>
        <item>123456</item>
        <item>qwerty</item>
    </string-array>

    <string-array name = 'Sites'>
        <item>https://vseinstrumenti.ru</item>
        <item>https://chipdip.ru</item>
        <item>https://citilink.ru</item>
    </string-array>

    <string-array name = "Logins">
        <item>Petr</item>
        <item>Danil</item>
        <item>Dima</item>
    </string-array>
</resources>
```

## colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#6200EE</color>
    <color name="colorPrimaryDark">#3700B3</color>
    <color name="colorAccent">#03DAC5</color>
</resources>
```

## AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.practice">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
            </intent-filter>
        </activity>
    </application>
```

```
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
    android:name=".WebActivity">
    </activity>
</application>

</manifest>
```

## ПРИЛОЖЕНИЕ АДМИНИСТРИРОВАНИЕ “А”

### **A.1    /etc/nginx/sites-available/default (изменённая часть)**

```
# Default server configuration
#
user frizik http;
...
http
{
...
    server
    {
        listen 80;
        server_name practice.tech;
        Root /home/frizik/Site_For_Dungeon-master;
        Index index.html;
    }
...

```

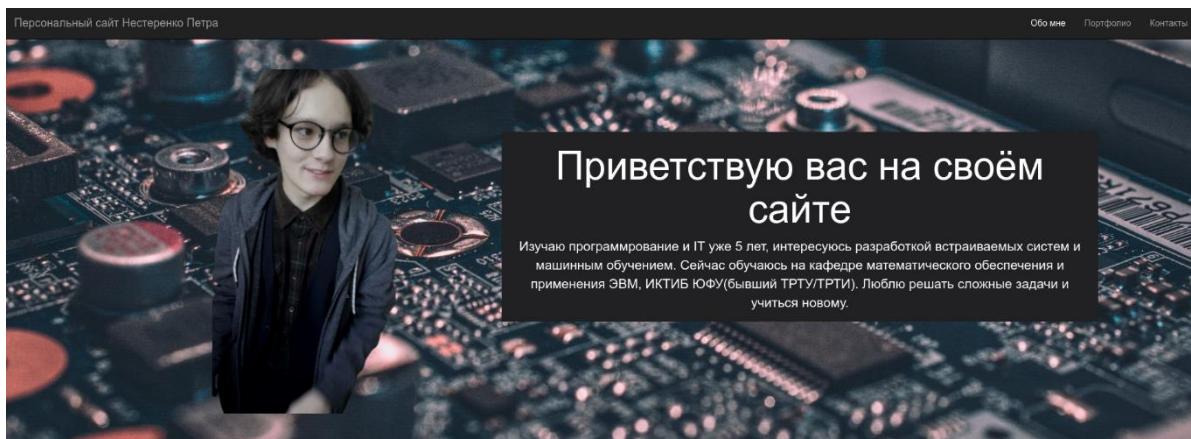
### **A.2    /etc/hosts**

```
127.0.0.1 localhost
127.0.0.1 practice.tech
127.0.1.1 frizik.localdomain frizik

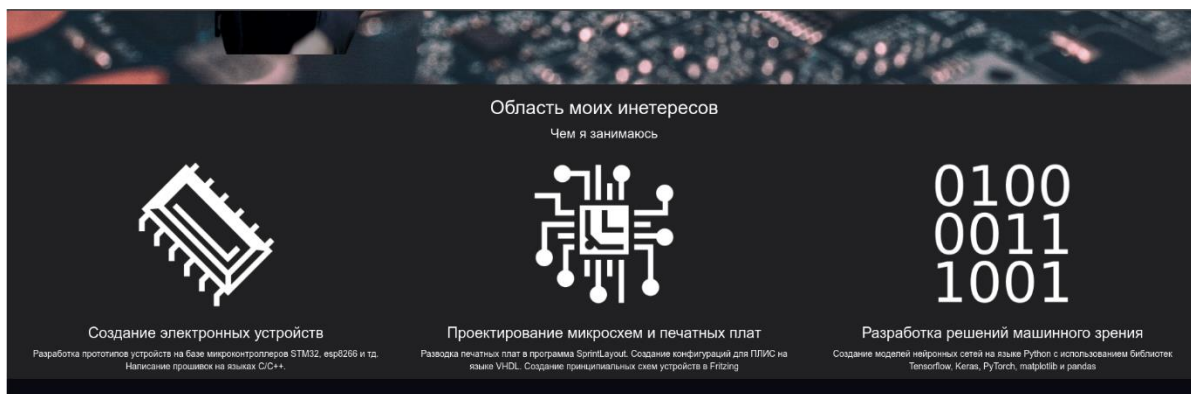
```

# ПРИЛОЖЕНИЕ WEB “А”

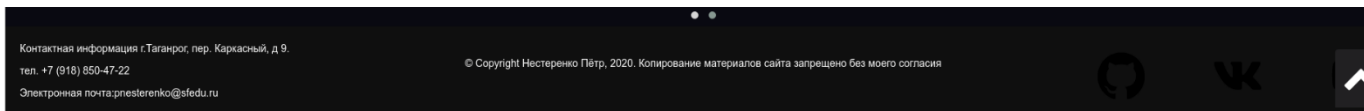
## А.1 Основной блок страницы



## А.2 Блок “Обо мне”



## А.3 Блок “Контакты(footer)”



## ПРИЛОЖЕНИЕ WEB “Б”

### Б.1 iINDEX.HTML

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
    <link rel="stylesheet" href="styles/header.css">
    <link rel="scripts" href="scripts/scrollup.js">

    <!-- Scrollup надо бы сделать -->
    <script src="scripts/PageUp.js" type="text/javascript"></script>

    <!-- Font Awesome -->
    <link rel="stylesheet" href="libs/font-awesome-4.7.0/css/font-
awesome.min.css">

    <!-- Совиная корусель -->
    <link rel="stylesheet"
href="libs/owlcarousel/assets/owl.carousel.min.css"/>
    <link rel="stylesheet"
href="libs/owlcarousel/assets/owl.theme.default.css"/>
    <script src="libs/owlcarousel/owl.carousel.min.js"></script>
    <script src="scripts/Owl_Init.js" type="text/javascript"></script>

    <!-- Для появления элментов при прокрутке (не работает) -->
    <link rel="stylesheet" href="libs/aos-master/dist/aos.css">
    <script src="libs/aos-master/dist/aos.js"
type="text/javascript"></script>
    <script src="scripts/Init_AOS.js" type="text/javascript"></script>

    <!-- Плавный переход между якорными ссылками -->
    <script src="scripts/plavniy.js" type="text/javascript"></script>

    <!-- Заголовое и иконка сайта -->
    <title>Персональный сайт Нестеренко Пётра</title>
    <link rel="shortcut icon" href="images/processor.ico" type="image/x-
icon">

  </head>
  <!-- <body class="container"> -->
  <body>
    <!-- Верхний тулбар -->
    <div id="TopBar">
      <nav class="navbar navbar-inverse" style="margin: 0;border-radius:0">
        <div class="container-fluid">
          <div class="navbar-header">
            <a class="navbar-brand" href="#">Персональный сайт Нестеренко
Петра</a>
```

```

</div>
<nav id="menu">
  <ul class="nav navbar-nav navbar-right">

    <!-- <li class="active"><a href="#TopBar">Главная</a></li> --
>
    <li><a href="#aboutme">Обо мне</a></li>
    <li><a href="#Portfolio">Портфолио</a></li>
    <li><a href="#Contact">Контакты</a></li>
  </ul>
</nav>
</div>
</nav>
</div>

<!-- Моя фотография и общая информация -->
<div id="Info" class="jumbotron jumbotron-fluid row"
style="color:#FFFFFF">
  <div class="col-md-2"></div>

  <div class="col-md-3" style="margin:0">
    
  </div>

  <div class="col-md-6" style="margin-top: 100px;background-color:
#222123;text-align: center;margin-bottom: 0 auto;">
    <h1 class="display-4">Приветствую вас на своём сайте</h1>
    <p style="text-align:center"> Изучаю программирование и IT уже
5 лет, интересуюсь разработкой встраиваемых систем и машинным обучением. Сейчас
обучаюсь на кафедре математического обеспечения и применения ЭВМ, ИКТИБ ЮФУ(бывший
ТРТУ/ТРТИ). Люблю решать сложные задачи и учиться новому.</p>
  </div>
</div>

<!-- Блок обо мне -->
<div id="aboutme" class="container-fluid">
  <div class="row">
    <h2 style="color:#FFFFFF">Область моих инетересов</h2>
    <p class="lead" style="color:#FFFFFF">Чем я занимаюсь</p>
    <div class="col-md-4" style="color:#FFFFFF">
      
      <h3>Создание электронных устройств</h3>
      <p>Разработка прототипов устройств на базе микроконтроллеров
STM32, esp8266 и тд. Написание прошивок на языках C/C++.</p>
    </div>
    <div class="col-md-4" style="color:#FFFFFF">
      
      <h3>Проектирование микросхем и печатных плат</h3>
      <p>Разводка печатных плат в программа SprintLayout. Создание
конфигураций для ПЛИС на языке VHDL. Создание принципиальных схем устройств в
Fritzing</p>
    </div>
    <div class="col-md-4" style="color:#FFFFFF">
      
      <h3>Разработка решений машинного зрения</h3>
      <p>Создание моделей нейронных сетей на языке Python с
использованием библиотек Tensorflow, Keras, PyTorch, matplotlib и pandas </p>
    </div>
  </div>
</div>
</div>

```



```

<!-- Блок портфолио -->
<div id="Portfolio">
<div class="home-demo" style="background-color:#0C0A13;margin: 0;">
  <h2 style="color:#FFFFFF;text-align: center">Портфолио проектов</h2>
  <div class="owl-carousel home-slider owl-theme" style="text-align:center;color:#FFFFFF">
    <div class="item">
      <h3>Победа в хакатоне CyberGarden2019</h3>
      
      <p style="padding:10px">Мы с командой одержали победу с приложением: «Мобильный гербарий». Суть нашего приложения заключаться в сборе гербария посредством фото, получения за это баллов, а также обмене фото с другими пользователями. С помощью нейронных сетей пользователь может, сделав снимок, узнать, лист какого дерева перед ним. Это даёт возможность использовать наше приложение для обучения</p>
    </div>
    <div class="item">
      <h3>Разработка аппаратной реализации скростонной системы шифрования</h3>
      
      <p style="padding:10px">Проект будет реализован с использованием ПЛИС на основе семейства Artix, обеспечивающих достаточно ресурсов для разработки при низкой потребляемой мощности и удовлетворительном быстродействии для поставленной задачи. Аппаратная реализация криптоалгоритмов позволяет во много раз быстрее, по сравнению с программными аналогами, проводить расчёты и распараллеливать многочисленные однотипные задачи, что так же увеличивает скорость вычислений. </p>
    </div>
    <div class="item">
      <h3>Победа в хакатоне ХакниCOVID-19, проводимым администрацией Ростовской области</h3>
      
      <p style="padding:10px">Наше "Приложение дополненной реальности для парковых зон с информационным насыщением" заняло первое место. Многие из нас при посещении музеев, парков и иных памятников культуры обращают внимание на информационные таблицы, рассказывающие об истории этого объекта. Приложение первокурсников добавляет в процесс чтения интерактивные элементы, что даёт возможность рассмотреть с разных сторон 3D модель предмета на своём смартфоне, прочитать информацию на виртуальных информационных досках, не ограничиваясь информацией, изложенной на одной реальной таблице.</p>
    </div>
  </div>
</div>

<!-- Кнопка вверх -->
<div class="scrollup">
  <i class="fa fa-chevron-up"></i>
</div>

<!-- Контакты -->
<footer>
<div id="Contact">
<div class="container-fluid">
  <div class="row" style="margin-bottom: auto;color:#FFFFFF">
    <div class="col-md-3" style="padding:20px">
      <p>Контактная информация г.Таганрог, пер. Каркасный, д 9.</p>
      <p>тел. +7 (918) 850-47-22</p>
      <p>Электронная почта:pnesterenko@sfedu.ru</p>
    </div>
  </div>
</div>

```

```
<div class="col-md-6" class="text-align" style="padding:40px;text-align: center;">  
    <p>© Copyright Нестеренко Пётр, 2020. Копирование материалов сайта  
запрещено без моего согласия</p>  
</div>  
<div class="col-md-1 navbar-right" style="text-align:center;margin-top: 20px">  
    <p><a  
href="https://www.youtube.com/channel/UC8cA0IOgP0ZJDOWGMQ6PABA?view_as=subscriber">  
          
    </a></p>  
</div>  
<div class="col-md-1 navbar-right" style="text-align:center;margin-top: 20px">  
    <p><a href="https://vk.com/petr_nesterenko">  
          
    </a></p>  
</div>  
<div class="col-md-1 navbar-right" style="text-align:center;margin-top: 20px">  
    <p><a href="https://github.com/FriZIK">  
          
    </a></p>  
</div>  
  
<div class="clearfix visible-lg"></div>  
</div>  
</div>  
</div>  
</footer>  
</html>
```

## ПРИЛОЖЕНИЕ WEB “В”

### B.1 header.css

```
}
#Info
{
  background-image: url('../images/background.jpg');
  background-size: cover;
  background-repeat: no-repeat;
  background-position: 50% 50%;
  margin:0;
}

.TopBar
{
  margin: 0;
  background-color:#19191A;
  border-radius:0;
  margin:0;
  border-radius: 0;
}

#aboutme
{
  margin : 0;
  text-align: center;
  background-color:#222123;
}

#Contact
{
  background-color:#101010;
  width: 100%;
  width: 100%;
  position:relative"
}

html
{
  overflow-x: hidden;
}

/* Блок для появления корусели */
.home-demo .item {
  background: #19191A;
  flex: 1 1 100px;
  padding: 10px;
}
.home-demo h2 {
  color: #19191A;
  text-align: center;
  padding: 5rem 0;
  margin: 0;
  font-style: italic;
  font-weight: 300;
}
.home-slider {
  display: flex;
  justify-content: space-between;
```

```

    flex-wrap: wrap;
}
@media (min-width: 601px) {
    .home-demo .item {
        margin: 5px;
    }
}

/* .owl-prev{
    background-image: url("<i class='fa fa-chevron-left'></i>");
    background-size: cover;
    width: 100px;
    height: 100px;
}
.owl-next{
    background-image: url("<i class='fa fa-chevron-right'></i>");
    background-size: cover;
    width: 100px;
    height: 100px;
} */

/* Для кнопки вверх*/
div.scrollup {
    position: fixed;
    color: #fff;
    background-color: #286090;
    right: 20px;
    bottom: 20px;
    padding: 4px 10px;
    font-size: 50px;
    border-top-left-radius: 4px;
    border-top-right-radius: 4px;
    cursor: pointer;
    display: none;
    text-align: center;
    z-index: 99999;
    background-color: #222123;
}
div.scrollup:hover {
    background-color: #000;
}

```

## ПРИЛОЖЕНИЕ WEB “Г”

### Г.1 PageUp.js

```
$(function() {  
    // при нажатии на кнопку scrollup  
    $('.scrollup').click(function() {  
        // переместиться в верхнюю часть страницы  
        $("html, body").animate({  
            scrollTop:0  
        },1000);  
    })  
})  
// при прокрутке окна (window)  
$(window).scroll(function() {  
    // если пользователь прокрутил страницу более чем на 200px  
    if ($(this).scrollTop()>200) {  
        // то сделать кнопку scrollup видимой  
        $('.scrollup').fadeIn();  
    }  
    // иначе скрыть кнопку scrollup  
    else {  
        $('.scrollup').fadeOut();  
    }  
});
```

### Г.2 Owl\_Init.js

```
$(document).ready(function()  
{  
    $('.owl-carousel').owlCarousel(  
        {  
            items : 2,  
            loop: true,  
            autoplay: true,  
            smartSpeed: 1000,  
            autoplayTimeout: 3500,  
            // nav:true,  
            dots: true,  
            responsiveClass:true,  
            responsive:{  
                0:{  
                    items:1,  
                    nav:false,  
                    loop: true  
                },  
                600:{  
                    items:2,  
                    nav:false,  
                    loop: true  
                }  
            }  
        }  
    );  
});
```

```

    },
    1000:{
        items:2,
        nav:false,
        loop: true
    }
}
}
}
);
});

```

## Д.2 Plavniy.js

```

$(document).ready(function(){
    $("#menu").on("click","a", function (event) {
        event.preventDefault();
        var id = $(this).attr('href'),
            top = $(id).offset().top;
        $('body,html').animate({scrollTop: top}, 1500);
    });
});

```

## ПРИЛОЖЕНИЕ WEB “Д”

### Д.1 Сертификат Stepik.

