Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Марийский государственный университет»

Институт цифровых технологий

Физико-математический факультет

**Отчет**

**по производственной практике: эксплуатационной практике**

Камаева Антона Львовича

студента 3 курса направление подготовки

02.03.03 Математическое обеспечение и администрирование информационных систем

Направленность (профиль) программы

Программирование и математическая обработка информации

очной формы обучения

Йошкар-Ола

2023 г.

Содержание

[Введение 3](#_Toc147929818)

[1. Теоретическая часть 6](#_Toc147929819)

[2. Практическая часть 8](#_Toc147929820)

[Заключение 11](#_Toc147929821)

[Приложение А 12](#_Toc147929822)

# Введение

В настоящее время всё более актуальными становятся web-приложения, которые подключаются к системе и управляются с помощью базы данных.

Web-приложение – это прикладное программное обеспечение, доступ к которой пользователь получает через интернет, то есть она не требует установки на устройство.

Во время прохождения практики было получено задание о разработке web-приложения, которое бы облегчило работу предприятия.

Целями практики «Б2.В.1 Производственная практика: эксплуатационная практика» являются:

формирование профессиональной компетенции (ПК-1) способен демонстрировать базовые знания программирования и информационных технологий;

формирование профессиональной компетенции (ПК-3) способен решать задачи в области развития науки, техники и технологии с учетом нормативного правового регулирования в сфере интеллектуальной собственности;

формирование профессиональной компетенции (ПК-5) способен использовать основные методы и средства автоматизации проектирования, реализации, испытаний и оценки качества при создании конкурентоспособного программного продукта и программных комплексов, а также способен использовать методы и средства автоматизации, связанные с сопровождением, администрированием и модернизацией программных продуктов и программных комплексов;

формирование профессиональной компетенции (ПК-6) способен использовать знания направлений развития компьютеров с традиционной (нетрадиционной) архитектурой; современных системных программных средств; операционных систем, операционных и сетевых оболочек, сервисных программ; тенденции развития функций и архитектур проблемно-ориентированных программных систем и комплексов в профессиональной деятельности;

формирование профессиональной компетенции (ПК-7) способен использовать основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений;

формирование профессиональной компетенции (ПК-8) способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования;

формирование универсальной компетенции (УК-2) способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений;

формирование универсальной компетенции (УК-3) способен осуществлять социальное взаимодействие и реализовывать свою роль в команде.

Задачами практики «Б2.В.1 Производственная практика: эксплуатационная практика» являются:

изучение текущего состояния индустрии программного обеспечения;

изучение способов применения технологий и методов программирования в процессах разработки объектов программных продуктов и комплексов;

изучение способов адаптации целевых платформ к задачам разработки объектов программных продуктов и комплексов;

систематизация, закрепление и расширение теоретических знаний и практических навыков, полученных в процессе обучения;

воспитание практики применения современных методов разработки и инструментального программного обеспечения при решении профессиональных задач.

Местом прохождения практики является Акционерное общество Ордена Ленина «Марийский машиностроительный завод». Предприятие основано в августе 1941 года, завод вырос до многопрофильного универсального производства, тесно сотрудничает в разработке и производстве продукции с 20-ю ведущими НИИ страны.

Практика была пройдена в роли программиста в бюро разработки программного обеспечения во внутреннем структурном подразделении «Управление информационных технологий №53».

В должностные обязанности программиста входит:

следить за исправным состоянием оборудования;

принимать участие в разработке программного обеспечения для удовлетворения нужд предприятия;

выполняет все остальные поручения начальника бюро разработки программного обеспечения;

# Теоретическая часть

Для написания алгоритма авторизации будем использовать язык Python.

Язык программирования: [формальный язык](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA), предназначенный для записи [компьютерных программ](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0). Язык программирования определяет набор [лексических](https://ru.wikipedia.org/wiki/%D0%9B%D0%B5%D0%BA%D1%81%D0%B8%D0%BA%D0%B0), [синтаксических](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) и [семантических](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D0%BA%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно — [ЭВМ](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%BE-%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0)) под её управлением.

Python – это [высокоуровневый](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%81%D0%BE%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) язык программирования  общего назначения с [динамической](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) [строгой](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D0%BE%D0%B3%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

Задумка по реализации языка появилась в конце [1980-х годов](https://ru.wikipedia.org/wiki/1980-%D0%B5), а разработка его реализации началась в 1989 году сотрудником голландского института [CWI](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BD%D1%82%D1%80_%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B8_%D0%B8_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B8) [Гвидо ван Россумом](https://ru.wikipedia.org/wiki/%D0%93%D0%B2%D0%B8%D0%B4%D0%BE_%D0%B2%D0%B0%D0%BD_%D0%A0%D0%BE%D1%81%D1%81%D1%83%D0%BC)[[44]](https://ru.wikipedia.org/wiki/Python#cite_note-:7-44). Для распределённой операционной системы [Amoeba](https://ru.wikipedia.org/wiki/Amoeba_(%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)) требовался расширяемый [скриптовый язык](https://ru.wikipedia.org/wiki/%D0%A1%D1%86%D0%B5%D0%BD%D0%B0%D1%80%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA), и Гвидо начал разрабатывать Python на досуге, позаимствовав некоторые наработки для языка [ABC](https://ru.wikipedia.org/wiki/ABC_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) (Гвидо участвовал в разработке этого языка, ориентированного на обучение программированию). В феврале [1991 года](https://ru.wikipedia.org/wiki/1991_%D0%B3%D0%BE%D0%B4) Гвидо опубликовал исходный текст в [группе новостей](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D1%83%D0%BF%D0%BF%D0%B0_%D0%BD%D0%BE%D0%B2%D0%BE%D1%81%D1%82%D0%B5%D0%B9) alt.sources[[50]](https://ru.wikipedia.org/wiki/Python#cite_note-50). С самого начала Python проектировался как [объектно-ориентированный язык](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F).

Преимущества Python:

Язык высокого уровня;

Огромное количество пакетов и модулей;

Интерпретируемый;

Портативный и универсальный

Поддержка графического интерфейса;

# Практическая часть

Разработать алгоритм авторизации для веб-приложения учёта сертификатов прививок от COVID-19 с использованием базы данных и языка программирования Python.

Для создания базы данных мы подключим библиотеку SQlite3.

*Описание выполненного задания:*

Импортируем библиотеку SQlite3 (import sqlite3);

С помощью команды sqlite3.connect(“MMZ.db”) создаем базу данных и приравниваем ее к переменной database;

 Далее создаем переменную cursor, к которой присваиваем команду database.cursuor(), данная переменная будет отвечать за обращения к базе данных в алгоритме;

С помощью команды cursour.execute() создаем таблицу, в которой называем два столбца: login и password, которые имеют расширения text;

Для сохранения таблицы используем команду database.commit();

Вводим условие if, в котором просим пользователя ввести с клавиатуры что пользователь хочет: войти или зарегистрироваться (1/2), если пользователь ввел 1, следовательно его переносит в тело условия, где его просят ввести его логин;

Далее мы используем команду cursour.execute(), в которой будет содержаться запрос к базе данных, которая выбирает из базы данных логин, который ввел пользователь;

В условие if мы проверяем существует ли такой логин, если же он не пустой, то мы в теле условия просим пользователя ввести пароль, если условие не выполняется, то на экран пользователя выводится сообщение что такого пользователя не существует;

Так же обращаемся к базе данных с помощью команды cursour.execute(), в которой записываем запрос берущий из бд логин и пароль, где логин должен совпадать с введенным ранее пользователем;

Если пароль совпадает с логином, к которому он привязан, то на экран выводится сообщение, что пользователь вошел в систему, если же условие не выполняется, то выводится сообщение о не правильном входе;

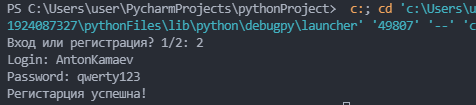
Если же пользователь хочет зарегистрироваться, то вводятся две переменные inp\_login и inp\_password, которые будут отвечать за новый логин и пароль пользователя;

В условии мы проверяем существующего пользователя по логину, если такой пользователь существует, то выводится сообщение что такой пользователь уже есть, если же такого пользователя не существует, то с помощью команды cursour.execute() и запроса вставки новых данных, мы добавляем новый логин и пароль в БД;

С помощью команды database.commit() сохраняем изменения

Результат работы можно увидеть на рисунке 1 и рисунке 2.

Далее эта база данных будет связана с основным кодом веб-приложения.

Ниже приведены скриншоты работы алгоритма авторизации (рисунок 1, рисунок 2, и рисунок 3):

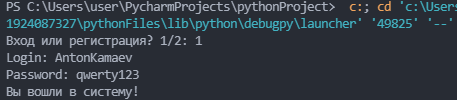
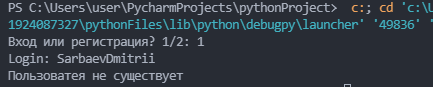
Рисунок 1 – Результат работы по созданию нового пользователя

Рисунок 2 – Результат работы входа в систему

Рисунок 3 – Ввод не существующего пользователя

# Заключение

Во время производственной практики было ознакомление с деятельностью, организационно-правовой формой и системой управления предприятия. Получены практические навыки по выбору и разработке, созданию, вводу в эксплуатацию программных продуктов.

**Составил:**

**\_\_\_\_\_\_\_\_\_** Камаев А.Л **\_\_\_\_\_\_\_**

подпись   дата

**Проверил:**

Руководитель практики

от ФГБОУ ВО «Марийский государственный университет»

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Савельева С. Ю.**\_\_\_\_\_\_\_**

      подпись                 дата

# Приложение А

import sqlite3

database = sqlite3.connect("MMZ.db")

cursor = database.cursor()

cursor.execute("""Create table if not exists users (

    login text,

    password text

)""")

database.commit()

if input("Вход или регистрация? 1/2: ") == "1":

    login = input("Login: ")

    cursor.execute(f"select login from users where login = '{login}'")

    if not cursor.fetchone() is None:

        password = input("Password: ")

        cursor.execute(f"select login, password from users where login = '{login}'")

        if cursor.fetchone()[1] == password:

            print("Вы вошли в систему!")

        else:

            print("НЕ ПРАВИЛЬНО!")

    else:

        print("Пользоватея не существует")

else:

    inp\_login = input("Login: ")

    inp\_password = input("Password: ")

    cursor.execute(f"select login from users where login ='{inp\_login}'")

    if cursor.fetchone() is None:

        cursor.execute("Insert into users values (?, ?)", (inp\_login,inp\_password))

        print("Регистарция успешна!")

        database.commit()

    else:

        print("Такой пользватель уже есть!")