

СОДЕРЖАНИЕ

| | |
|---|----|
| ВВЕДЕНИЕ | 4 |
| 1 Характеристика организации | 5 |
| 2 Формат прохождения практики | 6 |
| 3 Анализ используемых способов прохождения инструктажа и проверки знаний рабочих по технике безопасности на строительных объектах | 7 |
| 4 Анализ действующих систем учета прохождения инструктажа по технике безопасности | 8 |
| 5 Разработка программного обеспечения | 9 |
| 5.1 Выбор среды и языка программирования | 9 |
| 5.2 Выбор средства работы с базами данных | 10 |
| ЗАКЛЮЧЕНИЕ | 11 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 12 |
| ПРИЛОЖЕНИЕ А Исходный код ключевых функций проекта | 13 |

ВВЕДЕНИЕ

Место прохождения практики – ООО «СМОЛСТРОЙГАРАНТ».

Период прохождения практики – с 3 июля 2023 г. по 16 июля 2023 г.

Цель прохождения практики: создать единую систему учёта проведения тестирования по курсу техники безопасности на базе мессенджера Telegram.

Задачи прохождения практики:

- Выполнить анализ используемых способов проведения инструктажа и проверки знаний рабочих по технике безопасности на строительных объектах;
- Получить практические навыки работы с базами данных и Telegram BOT API, навыки разработки ПО на языке Python;
- Создать единую систему учёта проведения тестирования по курсу техники безопасности на базе мессенджера Telegram.

1 Характеристика организации

ООО «СМОЛСТРОЙГАРАНТ» — строительная компания, занимающаяся реконструкцией и ремонтом строительных объектов. С 2016 года является подрядчиком ПАО «ГМК „Норильский никель“». Производит демонтаж и переоборудование тепловых коммуникаций, а также капитальный ремонт теплоэлектростанций (ТЭЦ-1, г. Норильск).

Практика проходила в отделе технической безопасности [1], который занимается инструктажем и проверкой соблюдения техники безопасности на строительных объектах. В условиях крайнего севера, а также при работе с трехфазными масляными трансформаторами, пренебрежение техникой безопасности может привести к гибели, поэтому каждый работник в обязательном порядке проходит инструктаж и тестирование.

2 Формат прохождения практики

Строительные объекты представляют собой среду, в которой взаимодействуют разнообразные технические процессы, большое количество людей, техники, а также техногенных объектов, являющихся источниками повышенной опасности. В связи с этим, стройплощадки могут быть, а зачастую являются местами повышенной опасности для здоровья и безопасности работников. На строительных объектах придерживаются строгих норм и правил, чтобы обеспечить безопасность рабочего персонала и минимизировать риски производственных травм, несчастных случаев. Для выполнения задания, было необходимо проанализировать используемые способы проведения инструктажа и проверки знаний рабочих по технике безопасности на строительных объектах.

3 Анализ используемых способов прохождения инструктажа и проверки знаний рабочих по технике безопасности на строительных объектах

На строительных объектах безопасность играет ключевую роль, и одним из основных инструментов обеспечения безопасности является инструктаж работников по технике безопасности. Это процесс, включающий в себя не только передачу информации, но и проверку знаний и понимания сотрудниками правил и процедур безопасности.

Рассмотрим различные способы прохождения инструктажа и проверки знаний рабочих на строительных объектах:

- Индивидуальный Инструктаж. Этот метод предполагает проведение инструктажа каждого работника индивидуально или в малых группах. Преимуществом данного подхода является персональный подход: инструктор может учесть особенности каждого сотрудника. К минусам относятся низкая скорость обучения работников, высокая нагрузка на инструктора, высокие затраты на обучение большого количества работников.
- Групповой инструктаж. Такого рода инструктаж проводится для небольших групп работников. Его плюсами являются высокая эффективность при небольшом количестве обучающихся, высокая скорость обучения относительно индивидуального инструктажа при большом количестве обучающихся, меньшая нагрузка на инструкторов.
- Онлайн-Инструктаж. Метод проведения инструктажа с помощью современных технологий, таких как: Skype, Zoom. Главным преимуществом этого способа прохождения инструктажа является удобство инструкторов и работников. Инструктаж можно пройти, не тратя время на проезд до пункта проведения инструктажа, которое, в условиях крайнего севера, может занимать больше дня. К минусам относится необходимость в соответствующем оборудовании, а также стабильном интернет соединении.

Для каждого из способов проведения инструктажа предусмотрена проверка (тестирование) знаний сотрудников.

4 Анализ действующих систем учета прохождения инструктажа по технике безопасности

Данные о сотрудниках, времени и месте прохождения инструктажа и тестирования по технике безопасности хранятся в виде таблиц на бумажных носителях, в специализированных архивах. В рамках прохождения практики была поставлена задача “цифровизации” процесса хранения данных. Работать с вышеупомянутыми данными предстоит обычным офисным работникам, прорабам на строительных объектах, поэтому помимо хранения данных, необходимо реализовать интуитивно понятный интерфейс взаимодействия через смартфон.

5 Разработка программного обеспечения

5.1 Выбор среды и языка программирования

В современном мире существует огромное количество языков программирования. В целях удобства и простоты редактирования кода другим специалистом в будущем, был выбран язык Python. Python – один из наиболее популярных и многофункциональных языков программирования в мире.

Плюсы Python:

- Простота обучения и использования;
- Многозадачность и поддержка объектно-ориентированного программирования;
- Большое сообщество и экосистема;
- Кросс-платформенность.

Минусы Python:

- Низкая производительность при обработке большого объема данных;
- Сложность в разработке больших проектов в связи с динамической типизацией, потенциально приводящей к ошибкам во время выполнения.

На момент написания отчета, в компании работает порядка пятисот человек. Данное количество человек не является критичным, чтобы назвать проект большим. В связи с подобным обстоятельством, было принято решение закрыть глаза на минусы Python.

В качестве среды разработки выбрано программное обеспечение от компании Jet Brains – PyCharm [2]. PyCharm – это интегрированная среда разработки (IDE) для языка программирования Python. Он предоставляет мощные инструменты для написания, отладки и анализа кода Python, а также поддерживает множество фреймворков и библиотек. Например, асинхронный фреймворк для Telegram Bot API – Aiogram [3].

5.2 Выбор средства работы с базами данных

В качестве системы управления базами данных (СУБД) был выбран SQLite [4]. SQLite – это бесплатная, серверная, встраиваемая система управления реляционными базами данных (СУБД). Она отличается от большинства СУБД тем, что не требует отдельного сервера и запускается непосредственно в приложении, что делает ее легкой и быстрой в использовании. SQLite хранит данные в одном файле базы данных, который можно легко переносить и резервировать.

Особенности SQLite:

- Кросс-платформенность. SQLite поддерживается на различных операционных системах, включая Windows, macOS, Linux и многие другие. Это делает её идеальным выбором для кросс-платформенных приложений;
- Нет необходимости в сервере. Поскольку SQLite является встраиваемой СУБД, она не требует отдельного сервера для своей работы. Это снижает накладные расходы на обслуживание и упрощает настройку;
- Легковесность. SQLite имеет небольшой размер и низкие системные требования, что позволяет использовать её на ресурсоограниченных устройствах и в мобильных приложениях;
- Поддержка SQL. SQLite поддерживает SQL-запросы, что делает её знакомой и удобной для разработчиков, знакомых с языком SQL;
- Транзакции и ACID-совместимость. SQLite обеспечивает поддержку транзакций и соблюдает принципы ACID (Atomicity, Consistency, Isolation, Durability), обеспечивая надежность и целостность данных.

Области Применения SQLite:

- Мобильные приложения. SQLite широко используется для хранения данных в мобильных приложениях, работающих на платформах Android и iOS;
- Встроенные системы. SQLite подходит для встроенных систем, таких как медицинские устройства, автомобильные системы информации и другие.

ЗАКЛЮЧЕНИЕ

В результате прохождения учебной практики были получены следующие результаты:

- Проведён анализ используемых способов прохождения инструктажа и проверки знаний рабочих по технике безопасности на строительных объектах;
- Получены практические навыки работы с базами данных, Telegram BOT API, навыки разработки ПО на языке Python;
- Создана единая система учёта проведения тестирования по курсу техники безопасности на базе мессенджера Telegram.

Таким образом цель практики была достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ Р 12.3.050-2017 Система стандартов безопасности труда (ССБТ). Строительство. Работы на высоте. Правила безопасности. — URL: <https://docs.cntd.ru/document/1200146290>.
2. PyCharm: IDE для профессиональной разработки на Python. — URL: <https://www.jetbrains.com/ru-ru/pycharm/>.
3. Aiogram. — URL: <https://aiogram.dev>.
4. About SQLite. — URL: <https://www.sqlite.org/about.html>.

ПРИЛОЖЕНИЕ А

Исходный код ключевых функций проекта

Листинг А.1 – Исходный код

```
import sqlite3
from datetime import date
from datetime import datetime
from aiogram import Bot, Dispatcher, executor, types

# Настройкабд
conn = sqlite3.connect('TB_bot', check_same_thread=False)
cursor = conn.cursor()

def db_table_val(person_name: str, date_of_birth: str,
    person_passport: str, test_date: date, tg_nick: str ):
    cursor.execute('INSERT INTO tasks (fullname, date, passport,
        date_of_tb_test, tg_nickname) VALUES (?, ?, ?, ?, ?)',
        (person_name, date_of_birth, person_passport,
            test_date, tg_nick))
    conn.commit()

# Настройкабота
API_TOKEN = '*****'
bot = Bot(token=API_TOKEN)
dp = Dispatcher(bot)

#Обработка командпроисходитпосредствомасинхронныхфункцийвида :
@dp.message_handler(commands=['start'])
async def send_welcome(message: types.Message):
    global current_chat_id
    current_chat_id = message.from_user.id
    await message.reply("Бот СМОЛСТРОЙГАРАНТ\Управлениеи
        иадминистрированиебазойданныхсотрудников ")

@dp.message_handler(commands=['get_all_workers'])
```

```

async def get_workers_list(message: types.Message):
    global last_command
    current_user = await bot.get_chat_member(message.chat.id,
        message.from_user.id)
    last_command = 'get_all_workers'
    if current_user.status == 'administrator' or current_user.
        status == 'creator':
        all_workers = sort_and_convert_list_to_str(get_workers_list
            (False))
        answer = 'Список всехсотрудников :\n' + all_workers
        await message.answer(answer)
        answer = ''
        bad_workers = sort_and_convert_list_to_str(
            get_bad_or_good_tasks_list(True))
        answer = 'Не прошлитестирования :\n' + bad_workers + answer
        await message.answer(answer)
    else:
        await message.answer('У
            васнетправнаиспользованиеданнойкоманды .')

#Так жепредусмотреноразграничениепользователейнаадминистраторов ,
    имеющихполныйдоступкбазеданных ,
# иобычныхпользователей (38 строка)

#Добавление иудалениесотрудникаизбдпрямовтелеграмме :

@dp.message_handler(commands=['new_worker'])
async def new_worker(message: types.Message):
    global current_chat_id
    global last_command
    current_chat_id = message.chat.id
    current_user = await bot.get_chat_member(current_chat_id,
        message.from_user.id)

    if current_user.status == 'administrator' or current_user.
        status == 'creator':
        await message.reply(
            'Введите параметрывследующемвиде :\ФИОп, датарождения ,
                серияиномерпаспорта , датапрохождениятестирования ,
                телеграммия @'
            '\Примерп:\Ивановп ИванИванович , 01.01.1990 , 1111222222 ,
                10.06.23, @username')

```

```

        last_command = 'new_worker'
    else:
        await message.reply('У вас нет права использования данной команды .')

@dp.message_handler(commands=['delete_worker'])
async def delete_tasks(message: types.Message):
    global current_chat_id
    global last_command
    current_chat_id = message.chat.id

    current_user = await bot.get_chat_member(current_chat_id,
        message.from_user.id)

    if current_user.status == 'administrator' or current_user.status == 'creator':
        await message.reply('Введите @username сотрудника, которого необходимо удалить ')
        last_command = 'delete_worker'
    else:
        await message.reply('У вас нет права использования данной команды ')

#Так же осуществляется проверка корректности введенных при добавлении сотрудника данных, таких как дата :

def is_data_correct(message):
    current_text = message.split()
    flag = 0
    if len(current_text) != 3 or current_text[1].find(".") != 2:
        flag += 1
    else:
        data = current_text[1].split('.')
        try:
            day = data[0]
            month = data[1]
            year = data[2]
            if int(day) < date.today().day and int(month) < date.today().month and int(year) < date.today().year:

```

```

        flag += 1
        if int(day) > 31 or int(month) > 12 or int(year) >
2050:
            flag += 1
            if current_text[2].find("@") != 0:
                flag += 1
    except ValueError:
        flag += 1
if flag > 0:
    res = 'Неверный форматданных '
else:
    res = message

return res

# Работасбдосуществляетсяфункциямивида      :

def get_worker_id():
    try:
        sqlite_connection = sqlite3.connect('')
        cursor = sqlite_connection.cursor()
        print("Подключен к SQLite")

        sqlite_select_query = """SELECT id from workers"""
        cursor.execute(sqlite_select_query)
        all_workers_ids = cursor.fetchall()
        last_worker_id = all_workers_ids[len(all_workers_ids) - 1]
        lenth_of_str = len(last_worker_id)
        res = str(last_worker_id)[1:lenth_of_str - 3]
        cursor.close()

    except sqlite3.Error as error:
        print("Ошибка приработес    SQLite", error)
    finally:
        if sqlite_connection:
            sqlite_connection.close()
            print("Соединение с SQLite закрыто")

    return res

```