

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина» (УрФУ)

Институт фундаментального образования

Кафедра «Технической физики»

Оценка

Руководитель курсового

проектирования

Члены комиссии

Дата защиты

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

по теме: Создание телеграмм−бота для обучения сотрудников в сфере онлайн-продаж

по дисциплине: Информатика

Студент: Бузаньяров Эльдар Салаватович

(Подпись)

Группа: ФТ−140007

Екатеринбург

2025



Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина» (УрФУ)

Институт фундаментального образования

Кафедра «Технической физики»

**Задание на курсовую работу**

Студент Бузаньяров Эльдар Салаватович

Группа ФТ−140007

специальность/направление подготовки 09.03.02 Информационные системы и технологии

1.Тема курсовой работы: Создание телеграмм−бота для обучения сотрудников в сфере онлайн–продаж.

2.Содержание работы, в том числе состав графических работ и расчетов: описание и реализация алгоритмов, блок−схемы алгоритмов, таблицы, листинг кода.

3.Дополнительные сведения: работа выполнена на языке программирования python и библиотеки pyTelegramBotAPI.

4. План выполнения курсовой работы

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование элементов  проектной работы | Сроки | Примечания | Отметка о выполнении |
| Изучение методов и библиотек | 6.01-8.01 |  |  |
| Знакомство со сферой онлайн−продаж, составление плана курса | 9.01 |  |  |
| Планирование | 10.01 |  |  |
| Реализация алгоритма | 10.01-13.01 |  |  |
| Предварительная проверка | 13.01 |  |  |
| Оформление пояснительной записки | 14.01-17.01 |  |  |
| Защита работы | 18.01 |  |  |

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/М.Н. Ятченко

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc187785214)

[**1 Методология создания алгоритма** 5](#_Toc187785215)

[**1.1 Онлайн-продажи** 5](#_Toc187785216)

[**1.2 Создание бота** 6](#_Toc187785217)

[**1.2.1 Создание и обозначение блок**−**схем** 6](#_Toc187785218)

[**1.2.2 Приветствие** 7](#_Toc187785219)

[**1.2.3 Ввод имени** 7](#_Toc187785220)

[**1.2.4 Главное меню бота** 8](#_Toc187785221)

[**1.2.5 Аккаунт** 9](#_Toc187785222)

[**1.2.6 Выбор темы** 10](#_Toc187785223)

[**1.2.7 Урок** 11](#_Toc187785224)

[**1.2.8 Прогресс** 12](#_Toc187785225)

[**1.3 Средства и методы создания системы** 13](#_Toc187785226)

[**1.3.1 Библиотеки и доп. средства** 13](#_Toc187785227)

[**1.3.2 Методы создания программным путем** 13](#_Toc187785228)

[**1.3.3 Реализация отдельных объектов** 14](#_Toc187785229)

[**1.3.4 Реализация приветственного сообщения** 15](#_Toc187785230)

[**1.3.5 Реализация создания аккаунта** 16](#_Toc187785231)

[**1.3.6 Реализация главного меню** 17](#_Toc187785232)

[**1.3.7 Реализация аккаунта** 17](#_Toc187785233)

[**1.3.8 Реализация навигации между темами и уроками** 18](#_Toc187785234)

[**1.3.9 Реализация уроков** 19](#_Toc187785235)

[**1.3.10 Реализация прогресса** 20](#_Toc187785236)

[**2 Реализация алгоритма** 21](#_Toc187785237)

[**2.1 Состав информационной системы** 21](#_Toc187785238)

[**2.2 Страница навигации** 22](#_Toc187785239)

[**2.3 Личный кабинет неавторизованного пользователя** 24](#_Toc187785240)

[**2.4 Личный кабинет авторизованного пользователя** 25](#_Toc187785241)

[**2.5 Бонусы и акции** 26](#_Toc187785242)

[**2.6 История транзакций** 27](#_Toc187785243)

[**2.7 Добавление транзакции** 27](#_Toc187785244)

[ЗАКЛЮЧЕНИЕ 29](#_Toc187785245)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 30](#_Toc187785246)

[ПРИЛОЖЕНИЕ 31](#_Toc187785247)

# ВВЕДЕНИЕ

В условиях стремительного роста электронной коммерции и усиления конкуренции на рынке онлайн-продаж, обучение персонала приобретает первостепенное значение. Данный проект представляет собой реализацию телеграмм−бота, предоставляющего доступ к курсу по подготовке и повышению квалификации специалистов в этой области. Бот позволяет пользователям проходить обучение в интерактивном режиме, получая текстовую и мультимедийную информацию, а также взаимодействовать с системой посредством кнопок и обратных запросов. Также предусмотрен функционал для отслеживания личного прогресса, что мотивирует пользователей и позволяет им контролировать процесс обучения.

Проект способствует повышению профессиональной компетенции сотрудников, что, в свою очередь, ведет к увеличению эффективности продаж и улучшению качества обслуживания клиентов. Данная разработка может быть использована в любой организации, занимающейся онлайн-торговлей, и стать неотъемлемой частью процесса адаптации и развития персонала.

Целью данной работы является разработка телеграмм−бота, который предоставляет доступ к онлайн-курсу для обучения персонала в сфере онлайн-продаж.

Для достижения поставленной цели необходимо решить следующие задачи:

− Изучить сферу онлайн-продаж;

− Изучить инструменты и методы для создания телеграмм-ботов;

− Разработать телеграмм−бота;

− Реализовать телеграмм−бота;

− Протестировать телеграмм−бота.

# **1 Методология создания алгоритма**

## **1.1 Онлайн-продажи**

Онлайн-продажи представляют собой мощную и постоянно развивающуюся сферу, которая кардинально изменила принципы ведения бизнеса и потребительского поведения. За последние годы они превратились из нишевого явления в доминирующую силу в мировой экономике. Этот переход обусловлен рядом факторов, включая глобализацию, технологический прогресс и растущую популярность интернет-покупок. Сегодня для многих компаний онлайн-продажи являются не просто дополнительным каналом сбыта, а основным источником дохода и стратегическим направлением развития.

Преимущества онлайн-продаж очевидны: они позволяют бизнесу преодолевать географические ограничения и получать доступ к широкой, часто международной аудитории. Покупатели, в свою очередь, ценят удобство и доступность покупок в любое время и из любой точки мира. Развитие мобильных технологий еще больше усилило этот тренд, делая онлайн-шопинг еще более простым и доступным. Кроме того, электронная коммерция позволяет компаниям собирать и анализировать данные о поведении клиентов, что дает возможность персонализировать предложения и улучшать качество обслуживания.

Однако, стоит отметить, что динамичная природа онлайн-продаж также создает ряд вызовов. Конкуренция на этом рынке очень высока, и для успеха необходимо постоянное развитие и совершенствование. Успех в онлайн-продажах определяется не только качеством продукции и привлекательными ценами, но и профессионализмом персонала, который работает с клиентами в цифровой среде. Менеджеры по продажам должны обладать не только отличными навыками общения, но и глубоким пониманием потребностей клиентов, а также умением адаптироваться к постоянно меняющимся технологиям.

Работа в онлайн-продажах требует гибкости, креативности и постоянного обучения. Менеджеры должны уметь устанавливать контакт в цифровой среде, выявлять потребности и мотивации клиентов, представлять продукт таким образом, чтобы он был максимально привлекательным и решать возникающие вопросы и возражения. Постпродажное обслуживание также играет важную роль, поскольку от него зависит лояльность клиентов и их готовность к повторным покупкам. В конечном счете, онлайн-продажи — это не только про транзакции, но и про выстраивание долгосрочных отношений с клиентами, что требует постоянного совершенствования и развития навыков персонала.

## **1.2 Создание бота**

### **1.2.1 Создание и обозначение блок**−**схем**

В данной работе для описания отдельных процессов и алгоритмов взаимодействия с графическим интерфейсом будут использоваться блок−схемы, построенные с использованием приложения draw.io в соответствии с обозначениями, представленными в таблице 1.

Таблица 1 – Обозначение блок−схем

|  |  |
| --- | --- |
| Блок | Значение |
|  | Начало/конец алгоритма,  переход в следующий/предыдущий раздел |
|  | Процесс/Действие |
|  | Ячейка данных |
|  | Ввод/вывод данных |
|  | Показывают направление потока выполнения алгоритма |

### **1.2.2 Приветствие**

При запуске бота пользователь получает приветственное сообщение, содержащее имя его аккаунта в мессенджере. Так же появляется возможность нажать кнопку для перехода к следующему этапу. Схема взаимодействия представлена на рисунке 1.



Рисунок 1 – Приветствие

### **1.2.3 Ввод имени**

После приветствия бот задает вопрос о том, как зовут пользователя. После получения ответа бот записывает имя пользователя в отдельный словарь и предоставляет возможность продолжить работу, нажав кнопку. После нажатия пользователя переносит в главное меню телеграмм бота. Схема взаимодействия представлена на рисунке 2.

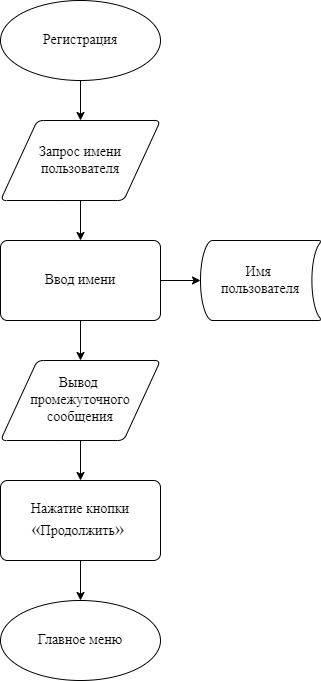


Рисунок 2 –Ввод имени

### **1.2.4 Главное меню бота**

В главном меню пользователю предоставляется возможность перейти в три раздела бота, а именно: аккаунт, выбор темы и прогресс. Переход в разделы осуществляется с помощью кнопок, с названиями соответствующих разделов. Схема взаимодействия представлена на рисунке 3.

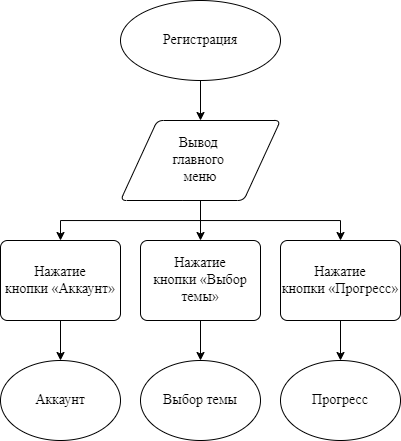


Рисунок 3 – Главное меню

### **1.2.5 Аккаунт**

В данном разделе пользователь получает данные об аккаунте, такие как имя пользователя и текущие курсы пользователя. Также пользователь может по нажатию кнопки вернуться в главное меню. Схема взаимодействия представлена на рисунке 4.

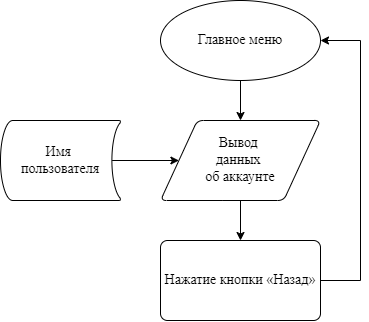


Рисунок 4 – Аккаунт

### **1.2.6 Выбор темы**

Курс имеет 4 главные темы, каждая тема имеет 3 урока. В данном разделе пользователь имеет возможность перейти в любую тему, а из него в любой раздел выбранной темы и попасть на соответствующий информационный. Также система кнопок даёт возможность возвращаться на уровень выше, позволяя пользователю выбрать другую тему или урок. Схема взаимодействия представлена на рисунке 5.

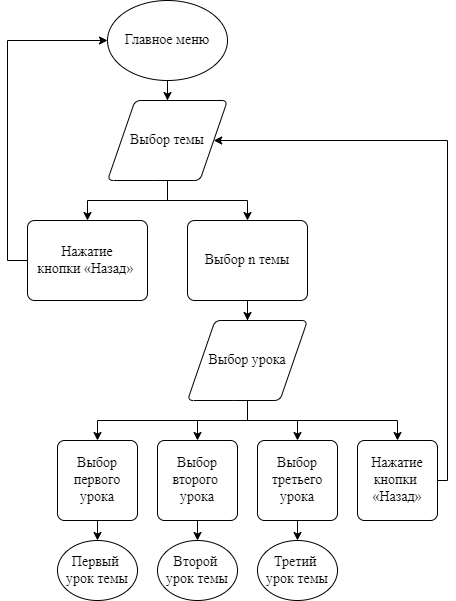


Рисунок 5 – Выбор темы

### **1.2.7 Урок**

После выбора определенного урока бот предоставляет пользователю информационные материалы по соответствующей теме, такие как:

– Текстовые материалы;

– Схемы;

– Изображения;

– Ссылки на статьи;

– Ссылки на видеоматериалы.

Так же можно отметить урок как пройденный, двигая прогресс прохождения курса или выйти и выбрать другой урок. Схема взаимодействия представлена на рисунке 6.

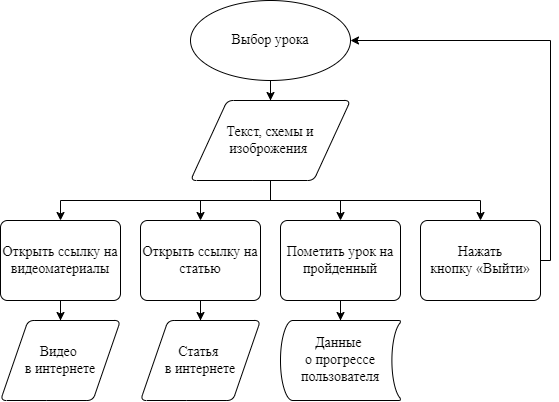


Рисунок 6 – Урок

### **1.2.8 Прогресс**

Данный раздел позволяет пользователю отслеживать свой прогресс прохождения курса и выйти обратно в меню. Здесь отображается следующая информация: количество пройденных уроков, общее количество уроков и их соотношение в процентах.

Данные о прогрессе прохождения курса хранятся в отдельном словаре. Схема взаимодействия представлена на рисунке 7.

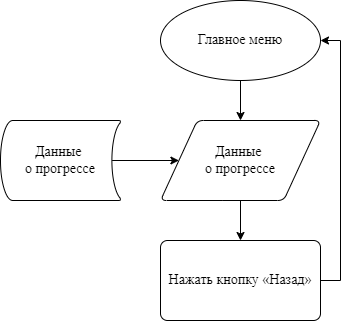


Рисунок 7 – Урок

## **1.3 Средства и методы создания системы**

### **1.3.1 Библиотеки и доп. средства**

Разработка телеграмм–бота для обучения сотрудников в сфере онлайн–продаж осуществляется с использованием языка программирования Python и библиотеки pyTelegramBotAPI.

pyTelegramBotAPI – библиотека Python для создания телеграмм–ботов. Она предоставляет возможность создания элементов интерфейса бота. В данной работе с помощью данной библиотеки созданы:

− Текстовые сообщения;

− Сообщения с изображениями;

− Кнопки с ссылками.

Регистрация бота и привязка API-ключа осуществляется с помощью телеграмм-бота BotFather, позволяющего создавать ботов и управлять ими. Он так же предоставляет возможности для настройки имени и фотографии профиля ботаю.

### **1.3.2 Методы создания программным путем**

Телеграмм–бот для обучения сотрудников в сфере онлайн–продаж реализуется программным методом с использованием линейных и разветвляющихся алгоритмов. Полный код программы представлен в приложении А.

Для создания клавиатуры и кнопок, отправки и удаления сообщений и картинок используются соответствующие классы и методы. Для обработки запросов, отправляемых при нажатии кнопок применяются декораторы, они позволяет вызывать разные функции при разных событиях, происходящих в чате. Пример − отправка сообщений, нажатие кнопок.

### **1.3.3 Реализация отдельных объектов**

Для создания самого бота используется класс TeleBot. При создании объекта бота в класс передается API-ключ, полученный с помощью BotFather:

from telebot import \*  
token = '7398398254:AAFa27\_AjW45tXGjZbXSlMou8YH0cnsAx8I'  
bot = telebot.TeleBot(token)

Для реализации кнопок необходимо создать клавиатуру и объекты с классом кнопки. После создания кнопок их необходимо добавить в клавиатуру:

goback\_keyboard = types.InlineKeyboardMarkup()  
goback\_keyboard.row\_width = 2  
article = types.InlineKeyboardButton('Статья', url=article\_link)  
video = types.InlineKeyboardButton('Видео', url=video\_link)  
complete = types.InlineKeyboardButton('Просмотрено', callback\_data='progress\_plus\_1b')  
goback = types.InlineKeyboardButton('Выйти', callback\_data='theme1')  
goback\_keyboard.add(article, video, complete, goback)

При нажатии кнопок может быть осуществлен переход по интернет-ссылке или отправлен запрос, который будет обработан ботом с помощью декоратора callback\_query\_handler:

@bot.callback\_query\_handler(func = lambda call: True)  
def answer(call):  
 option = call.data  
 umessage = call.message  
 match option:  
 case 'start':  
 bot.send\_message(umessage.chat.id, text='Как вас зовут')  
 bot.register\_next\_step\_handler(umessage, registration)

Для отправки сообщений используются методы send\_message и send\_photo в зависимости от того, что необходимо отправить. Метод send\_message позволяет отправить текст и привязать к сообщению заранее созданную клавиатуру с кнопками. Метод send\_photo позволяет отправить изображение с подписью. При отправке сообщения, предыдущее удаляется, улучшая восприятие интерфейса. Удаление происходит по индексу сообщения, которое было отправлено перед текущим:

message\_id = bot.send\_photo(umessage.chat.id, image, caption=text, reply\_markup=goback\_keyboard).message\_id  
bot.delete\_message(umessage.chat.id, message\_id - 1)

### **1.3.4 Реализация приветственного сообщения**

Бот обрабатывает команду start от пользователя и запускает функцию, которая создает текст сообщения, включающий имя аккаунта пользователя. Для продолжения создается клавиатура и кнопка с соответствующим текстом. Далее бот отправляет приветственный текст с клавиатурой, привязанной к сообщению. После нажатия кнопки бот получает запрос start :

@bot.message\_handler(commands=['start'])  
def welcome\_message(message):  
 welcome\_message = (f'Здравствуйте, {message.from\_user.first\_name}! \n'  
 f'Приветствуем вас на платформе для обучения сотрудников в сфере онлайн-продаж')  
 start\_button\_keyboard = types.InlineKeyboardMarkup()  
 start\_button = types.InlineKeyboardButton('Начать', callback\_data='start')  
 start\_button\_keyboard.add(start\_button)  
 bot.send\_message(message.chat.id, welcome\_message, reply\_markup=start\_button\_keyboard)

### **1.3.5 Реализация создания аккаунта**

Заранее создаются словари, хранящие данные об именах пользователях и их прогрессе:

user\_data = {}

progress\_data = {}

Так же создается функция, сохраняющая данные о введенном имени и устанавливающая нулевой прогресс у нового пользователя. При нажатии кнопки «Продолжить» бот получает запрос main:

def registration(message):  
 user\_id = message.chat.id  
 user\_name = message.text  
 user\_data[user\_id] = user\_name  
 progress\_data[user\_id] = []  
 continue\_keyboard = types.InlineKeyboardMarkup()  
 continue\_button = types.InlineKeyboardButton('Продолжить', callback\_data='main')  
 continue\_keyboard.add(continue\_button)  
 bot.send\_message(message.chat.id, f'Приятно познакомиться, {user\_name}!', reply\_markup=continue\_keyboard)

При получении запроса start бот просит ввести имя пользователя и вызывает ранее созданную функцию, обрабатывающую введенные данные:

@bot.callback\_query\_handler(func = lambda call: True)  
def answer(call):  
 option = call.data  
 umessage = call.message  
 match option:  
 case 'start':  
 bot.send\_message(umessage.chat.id, text='Как вас зовут')  
 bot.register\_next\_step\_handler(umessage, registration)

### **1.3.6 Реализация главного меню**

При получении запроса main бот отправляет сообщение с клавиатурой, которая содержит кнопки, позволяющие перейти в разделы бота. Каждая кнопка отправляет разные запросы, отвечающие за свои разделы:

case 'main':  
 choose\_topic = types.InlineKeyboardMarkup()  
 account\_b = types.InlineKeyboardButton('Аккаунт', callback\_data='account')  
 theme\_b = types.InlineKeyboardButton('Выбор темы', callback\_data='course')  
 progress\_b = types.InlineKeyboardButton('Прогресс', callback\_data='progress')  
 choose\_topic.row\_width = 1  
 choose\_topic.add(account\_b, theme\_b, progress\_b)  
 message\_id = bot.send\_message(umessage.chat.id, 'Курс "Эффективные продажи в онлайн-магазине"', reply\_markup=choose\_topic).message\_id  
 bot.delete\_message(umessage.chat.id, message\_id - 1)

### **1.3.7 Реализация аккаунта**

При получении запроса account бот отправляет сообщение с данными об аккаунте, в частности имя пользователя, которое было получено при регистрации, и кнопку, позволяющую вернуться в главное меню:

case 'account':  
 chatid = umessage.chat.id  
 username = user\_data[chatid]  
 account\_message = (f'{username} \n'  
 f'\n'  
 f'Курсы:\n'  
 f' "Эффективные продажи в онлайн-магазине"')  
 account\_keyboard = types.InlineKeyboardMarkup()  
 account\_button = types.InlineKeyboardButton('Назад', callback\_data='main')  
 account\_keyboard.add(account\_button)  
 message\_id = bot.send\_message(umessage.chat.id, account\_message, reply\_markup=account\_keyboard).message\_id  
 bot.delete\_message(umessage.chat.id, message\_id - 1)

### **1.3.8 Реализация навигации между темами и уроками**

При получении запроса course бот отправляет сообщение с клавиатурой, которая содержит кнопки, позволяющие перейти в одну из глав курса, и кнопку, позволяющую вернуться в главное меню. Каждая кнопка отправляет разные запросы, отвечающие за свои главы:

case 'course':  
 themes\_list = types.InlineKeyboardMarkup()  
 themes\_list.row\_width = 1  
 theme1 = types.InlineKeyboardButton('1 | Введение в онлайн-продажи', callback\_data='theme1')  
 theme2 = types.InlineKeyboardButton('2 | Работа с клиентами', callback\_data='theme2')  
 theme3 = types.InlineKeyboardButton('3 | Презентация товара и работа с возражениями', callback\_data='theme3')  
 theme4 = types.InlineKeyboardButton('4 | Завершение сделки и постпродажное обслуживание', callback\_data='theme4')  
 goback = types.InlineKeyboardButton('Назад', callback\_data='main')  
 themes\_list.add(theme1, theme2, theme3, theme4, goback)  
 message\_id = bot.send\_message(umessage.chat.id, 'Выберите тему:', reply\_markup=themes\_list).message\_id  
 bot.delete\_message(umessage.chat.id, message\_id-1)

При получении запроса, соответствующего какой-либо главе бот отправляет сообщение с клавиатурой, которая содержит кнопки, позволяющие перейти в один из уроков главы, и кнопку, позволяющую вернуться к выбору главы. Каждая кнопка отправляет разные запросы, отвечающие за свои уроки:

case 'theme1':  
 themes\_list = types.InlineKeyboardMarkup()  
 themes\_list.row\_width = 1  
 theme\_a = types.InlineKeyboardButton('Особенности онлайн-продаж и их отличие от офлайн-продаж', callback\_data='theme1a')  
 theme\_b = types.InlineKeyboardButton('Основные понятия: конверсия, средний чек, лиды', callback\_data='theme1b')  
 theme\_c = types.InlineKeyboardButton('Роль менеджера по продажам в онлайн-магазине', callback\_data='theme1c')  
 goback = types.InlineKeyboardButton('Назад', callback\_data='course')  
 themes\_list.add(theme\_a, theme\_b, theme\_c, goback)  
 message\_id = bot.send\_message(umessage.chat.id, 'Выберите урок:', reply\_markup=themes\_list).message\_id  
 bot.delete\_message(umessage.chat.id, message\_id - 1)

### **1.3.9 Реализация уроков**

При получении запроса, соответствующего какому-либо уроку бот отправляет сообщение с материалами по данному уроку и клавиатурой, которая содержит кнопки, позволяющие открыть видео или статью, и кнопки, позволяющие отметить урок как пройденный или вернуться к выбору урока:

case 'theme1a':  
 image = open('difference.png', 'rb')  
 text = ('Добро пожаловать в увлекательный мир онлайн-продаж! '  
 'Этот раздел заложит фундамент вашего понимания специфики этого канала. '  
 'В отличие от традиционных офлайн-магазинов, где взаимодействие происходит лицом к лицу, онлайн-продажи полагаются на цифровые инструменты и коммуникацию. '  
 'Главное преимущество — это скорость и доступность: клиенты могут совершать покупки в любое время дня и ночи, из любой точки мира, имея лишь доступ к интернету. '  
 'Также важна персонализация: онлайн позволяет собирать данные о клиентах и настраивать предложения под их конкретные потребности и предпочтения. '  
 'Это может быть история покупок, поисковые запросы или поведение на сайте. '  
 'Но, в отличие от офлайна, здесь нет прямого физического контакта. '  
 'Поэтому, мы должны уметь создавать доверительную атмосферу и эффективно общаться с клиентами в цифровом формате, используя текстовые сообщения, чаты, электронную почту и другие средства связи.')  
 article\_link = 'https://softservice-group.ru/online-vs-ofline/'  
 video\_link = 'https://www.youtube.com/watch?v=lLdMnMTYaf0'  
 goback\_keyboard = types.InlineKeyboardMarkup()  
 goback\_keyboard.row\_width = 2  
 article = types.InlineKeyboardButton('Статья', url=article\_link)  
 video = types.InlineKeyboardButton('Видео', url=video\_link)  
 complete = types.InlineKeyboardButton('Просмотрено', callback\_data='progress\_plus\_1a')  
 goback = types.InlineKeyboardButton('Выйти', callback\_data='theme1')  
 goback\_keyboard.add(article, video, complete, goback)  
 message\_id = bot.send\_photo(umessage.chat.id, image, caption=text, reply\_markup=goback\_keyboard).message\_id  
 bot.delete\_message(umessage.chat.id, message\_id - 1)

### **1.3.10 Реализация прогресса**

При прохождении урока, пользователь может отметить урока как пройденный, нажав на кнопку. Она вызывает функцию, передавая данные о пользователе и об уроке, который необходимо учесть как пройденный:

case 'progress\_plus\_1a':  
 progress\_plus(umessage, '1a')

Функция получает данные о пользователе, который пометил урок как пройденный, и индекс этого урока. Эти данные записываются в ранее созданный словарь. Так же производится проверка того, не пройден ли данный урок ранее:

def progress\_plus(message, theme):  
 user\_id = message.chat.id  
 completed\_themes = progress\_data[user\_id]  
 completed\_themes.append(theme)  
 progress\_data[user\_id] = list(set(completed\_themes))

При получении запроса progress из главного меню, бот отправляет сообщение, содержащее данные о прогрессе пользователя и кнопку, позволяющую вернуться в главное меню:

case 'progress':  
 progress\_id = umessage.chat.id  
 completed\_lessons = progress\_data[progress\_id]  
 persents = round(len(completed\_lessons)/12, 2)\*100  
 progress\_message = (f'Ваш прогресс\n'  
 f'\n'  
 f'Курсы:\n'  
 f'"Эффективные продажи в онлайн-магазине"\n'  
 f'{len(completed\_lessons)}/12 ({persents}%)')  
 progress\_keyboard = types.InlineKeyboardMarkup()  
 goback = types.InlineKeyboardButton('Назад', callback\_data='main')  
 progress\_keyboard.add(goback)  
 message\_id = bot.send\_message(umessage.chat.id, progress\_message, reply\_markup=progress\_keyboard).message\_id  
 bot.delete\_message(umessage.chat.id, message\_id - 1)

# **2 Реализация алгоритма**

## **2.1 Состав информационной системы**

Разработанный телеграмм-бот предоставляет доступ к онлайн-курсу и включает в себя:

− Приветственное сообщение;

− Создание аккаунта;

− Главное меню;

− 4 главы;

− 12 уроков;

− Отслеживание прогресса.

## **2.2 Страница навигации**

Страница навигации представляет собой имитацию внешнего вида главной страницы сайта компании «ЛУКОИЛ». По умолчанию применена светлая тема с соблюдением корпоративной цветовой схемы. Для пользователя предоставляется возможность поменять тему, пролистать баннеры и перейти в личный кабинет. Смена темы применяет другую тему во всех окнах с соблюдением корпоративного стиля. Перелистывание баннеров производится с помощью кнопок, отвечающих за каждый из них. Переход в личный кабинет открывает новое окно с соответствующим функционалом Для интерактивности и улучшения пользовательского опыта реализованы функции смены цвета шрифта при наведении на кнопки. В светлой теме используются белый, черный, розовый и красный цвета. Внешний вид страницы навигации в светлой теме представлен на рисунке 8.

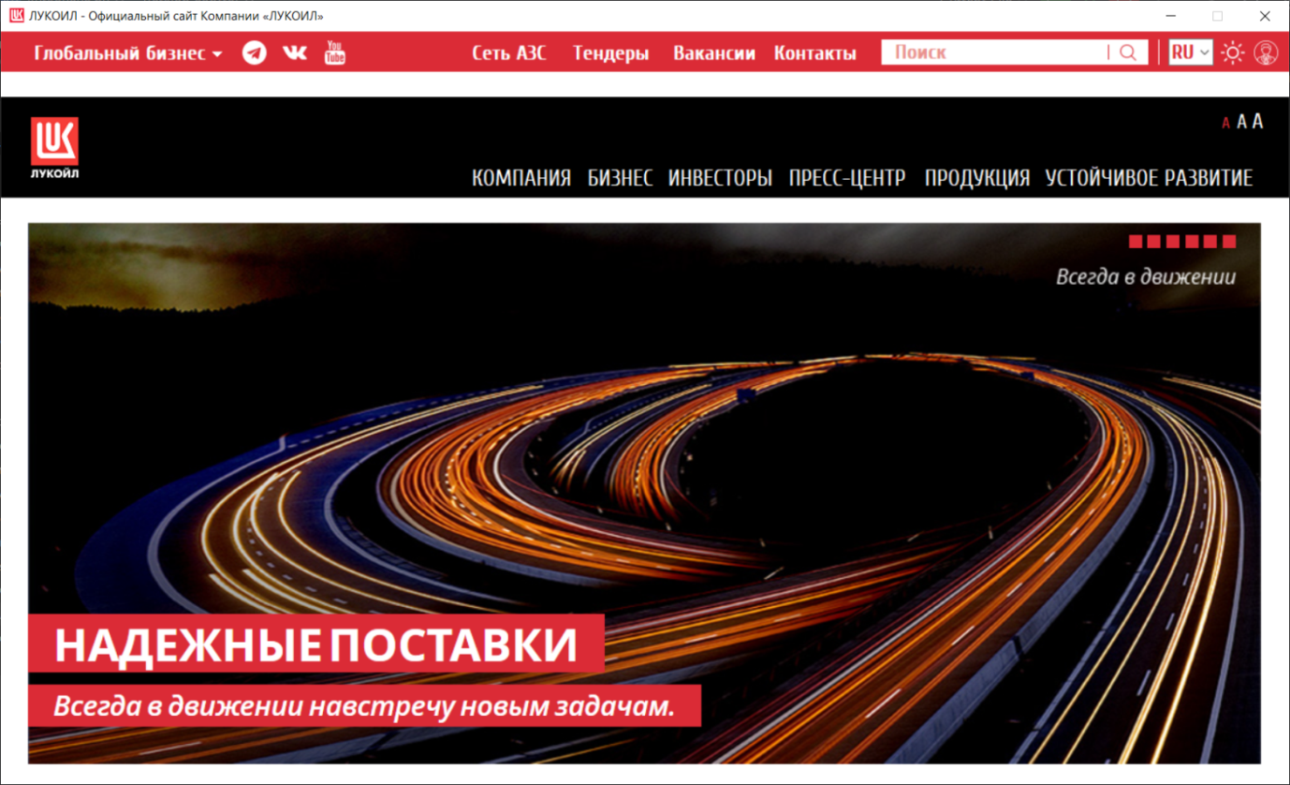


Рисунок 8 − Страница навигации в светлой теме

В светлой теме используются белый, черный, серый, темно−серый, розовый и темно−красный цвета. Внешний вид страницы навигации в темной теме представлен на рисунке 9.

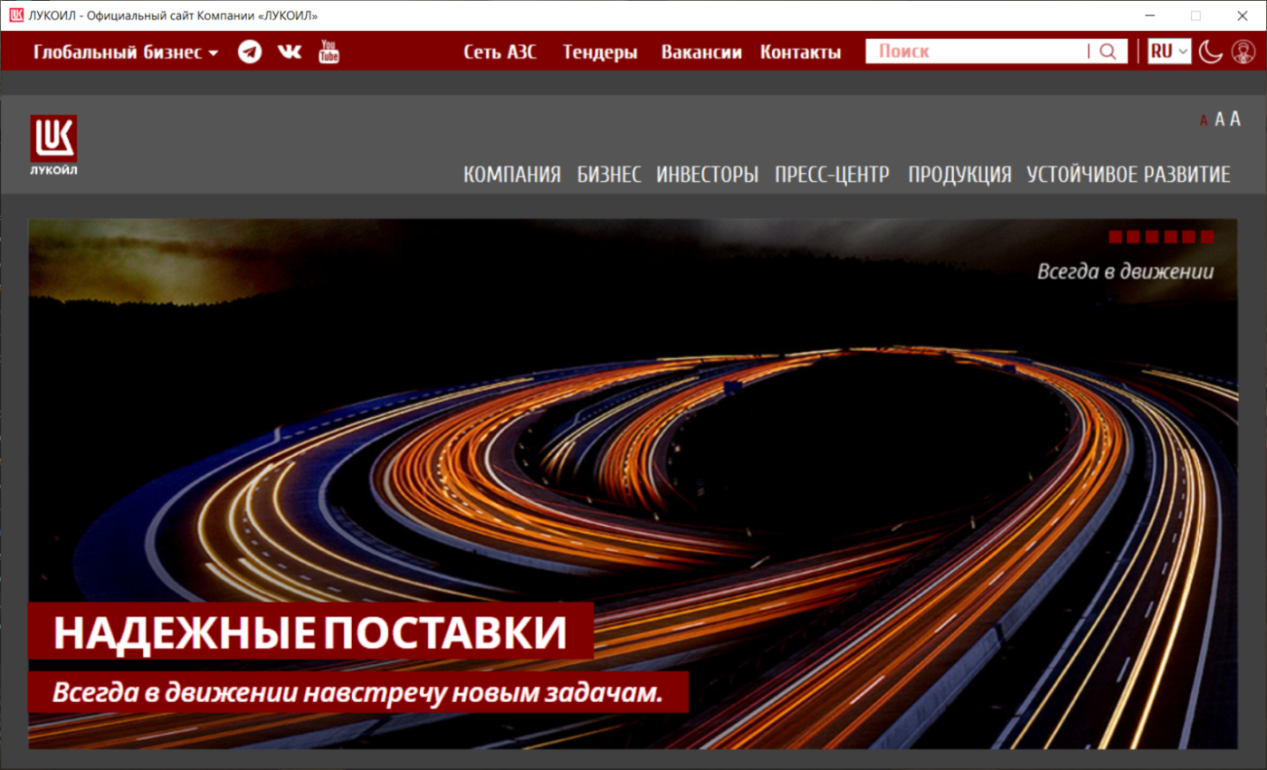


Рисунок 9 − Страница навигации в темной теме

## **2.3 Личный кабинет неавторизованного пользователя**

Личный кабинет по умолчанию предлагает пользователю зарегистрироваться, но также предоставляет возможность выполнить вход в уже существующий аккаунт. При неверно введённых данных пользователь получает оповещение в виде надписи на кнопке авторизации. Внешний вид личного кабинета неавторизованного пользователя при регистрации в обеих темах представлен на рисунке 10.

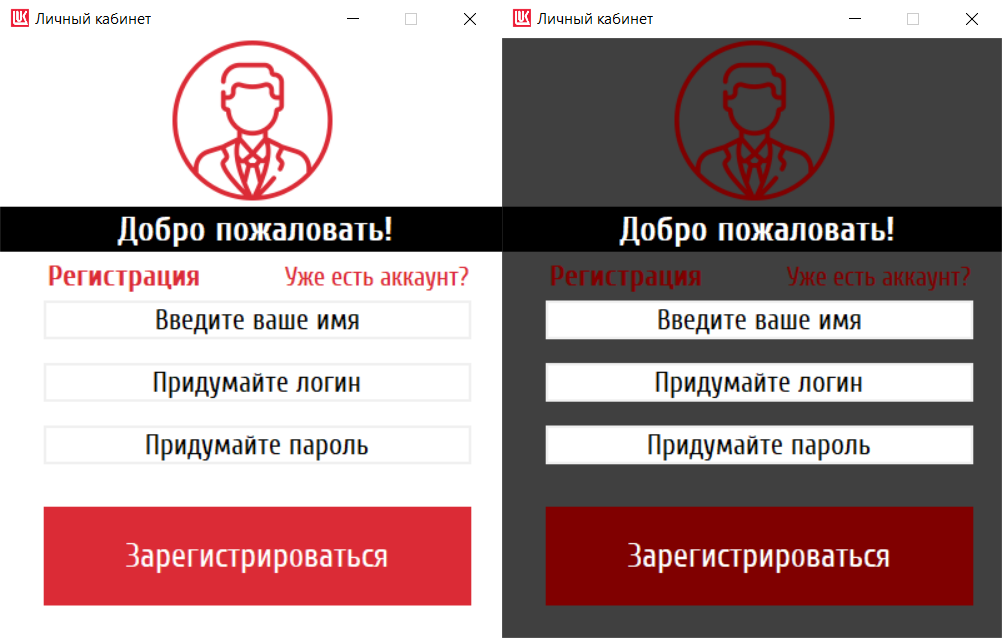


Рисунок 10 – Регистрация

Внешний вид личного кабинета неавторизованного пользователя при входе в обеих темах представлен на рисунке 11.

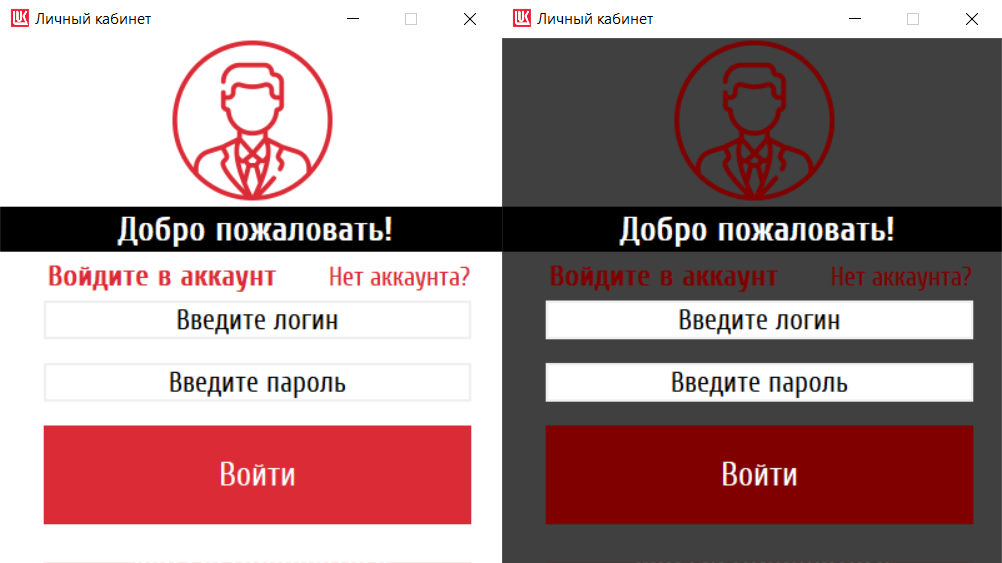


Рисунок 11 – Вход

## **2.4 Личный кабинет авторизованного пользователя**

Личный кабинет при успешной авторизации предоставляет пользователю возможность открыть историю транзакций, баланс бонусных баллов и выйти из аккаунта. При этом все данные об аккаунте, транзакциях и бонусах сохраняются. Внешний вид личного кабинета авторизованного пользователя в обеих темах представлен на рисунке 12.

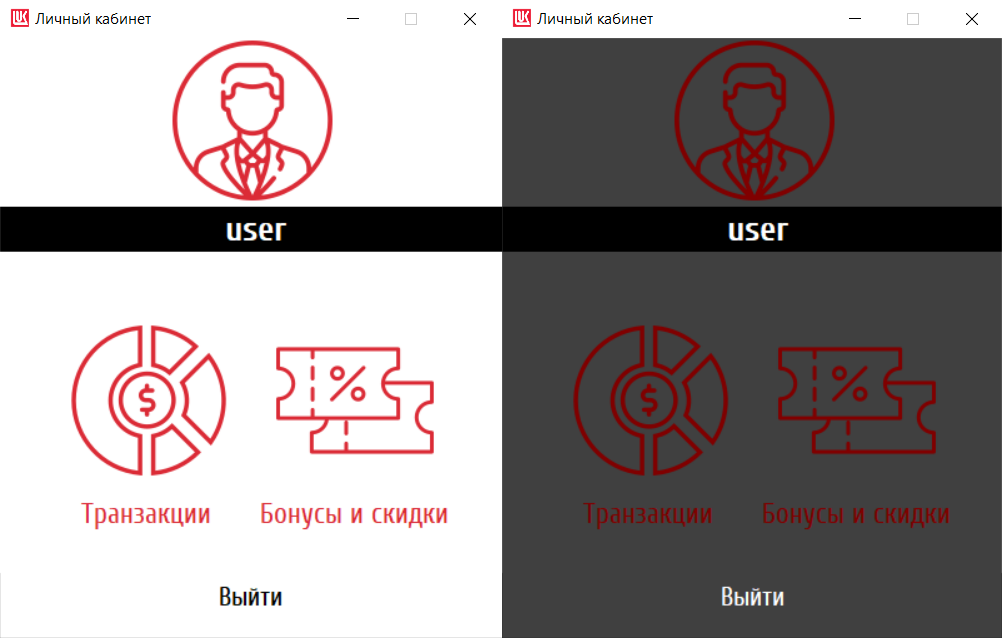


Рисунок 12 – Личный кабинет

## **2.5 Бонусы и акции**

Данный раздел предоставляет пользователю возможность получить информацию о текущем балансе бонусных баллов, а также получить информацию о промокодах. Внешний вид раздела бонусов и скидок в обеих темах представлен на рисунке 13.



Рисунок 13 – Бонусы и скидки

## **2.6 История транзакций**

Данный раздел предоставляет пользователю информацию о транзакциях, такую как:

− Дата транзакции;

− Количество приобретенных литров;

− Тип приобретенного бензина;

− Общая сумма транзакции;

− Начисленные бонусы.

Также пользователь имеет возможность добавить новую транзакции, нажав соответствующую кнопку, открывающую специальное окно. Внешний вид раздела истории транзакций представлен на рисунке 14.

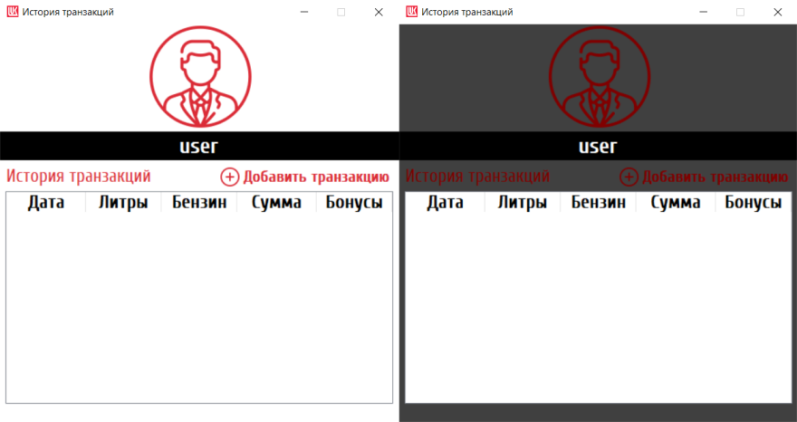


Рисунок 14 – История транзакций

## **2.7 Добавление транзакции**

Данный раздел предоставляет пользователю возможность добавить новую транзакцию в историю, позволяя выбрать дату в интерактивном интерфейсе календаря, ввести количество купленных литров и выбрать тип топлива в выпадающем списке. При неверно введённых данных пользователь получает оповещение в виде надписи на кнопке добавления транзакции. Внешний вид раздела добавления транзакции представлен на рисунке 15.

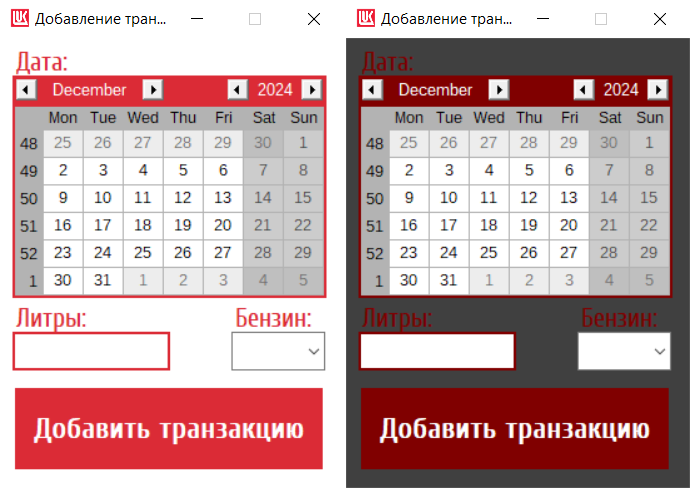


Рисунок 15 – Добавление транзакции

# ЗАКЛЮЧЕНИЕ

В заключение следует отметить, что в рамках данной работы был разработан прототип пользовательского интерфейса, имитирующий главную страницу и личный кабинет веб−сайта ПАО «ЛУКОЙЛ». Целью работы являлась разработка прототипа интуитивно понятного интерфейса личного кабинета, обеспечивающего удобный доступ к истории транзакций и бонусам.

Для достижения поставленной цели были успешно решены следующие задачи:

− Проведен анализ функционала сайта заказчика, что позволило определить ключевые элементы, необходимые для реализации в прототипе;

− Определена потребность заказчика в добавлении функционала личного кабинета;

− Изучены программные методы и средства для реализации системы;

− Разработана и реализована информационная система в виде прототипа, включающего смену цветовой темы и интерфейс личного кабинета с отображением истории транзакций и бонусной программой;

− Проведено тестирование информационной системы, что позволило убедиться в ее работоспособности и выявить потенциальные области для улучшения.

Таким образом, все поставленные задачи были выполнены, и цель работы была достигнута. Несмотря на то, что разработанный прототип не является полноценным веб−сайтом и не включает в себя весь спектр функций реального веб−приложения, он демонстрирует успешную реализацию ключевых аспектов пользовательского опыта. Прототип может служить в качестве основы для дальнейшей разработки и доработки, демонстрируя потенциал использования Tkinter для создания привлекательных и эргономичных пользовательских интерфейсов.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

<https://habr.com/ru/articles/580408>

<https://pypi.org/project/pyTelegramBotAPI>

<https://stackoverflow.com/questions/72830807/telebot-telegram-message-with-buttons>

<https://stackoverflow.com/questions/71021892/telebot-delete-message-pytelegrambotapi>

<https://ru.stackoverflow.com/questions/1578908>

<https://www.synapse-studio.ru/blog/skhodstva-i-razlichiya-offlayn-i-onlayn-torgovli>

# ПРИЛОЖЕНИЕ

**ПРИЛОЖЕНИЕ А**