**ТАШКЕНТСКИЙ МЕЖДУНАРОДНЫЙ УНИВЕРСИТЕТ**

**ОБРАЗОВАНИЯ**



**КУРСОВАЯ РАБОТА**

**по дисциплине разработка программного продукта**

**на тему: Автоматизация оптики**

|  |
| --- |
| ВЫПОЛНИЛ: |
| Джамалитдинов Ислом |
| Студент 1 курса группы: RI-103  Направления информационные технологии |

Ташкент 2022 г

2025 г.

**Общая структура технического задания на информационную систему**

**Содержание:**

1. **Общие сведения о проекте :**
   1. **Полное и краткое наименования информационной системы:**

* 1. **Структура проекта (вплоть до мелочей):**
  2. **Сроки сдачи проекта:**
  3. **Шифр темы или номера:**

1. **Цели и задачи проекта :**
   1. **Распределение работ между Исполнителем и Заказчиком:**

**2.2 Заказчик предоставляет информацию:**

**2.3 Исполнитель выполняет работы:**

1. **Дизайн:**
   1. **Заполненный заказчиком бриф лист:**
   2. **Функциональные прототипы страниц:**
   3. **Логика следования покупателя по приложению с различных точек входа на приложению:**
2. **Требования:**

* 1. **Требования к системе в целом:**
  2. **К администратору сайта:**

**4.3К хостингу:**

**4.4 Требования к Мобильному приложению.**

**4.5 Требования к качеству оказываемых услуг**

**4.6 Требования к безопасности оказания услуг и безопасности результатов работ**

**4.7 Порядок сдачи – приемки оказанных услуг**

**4.8 Требования по передаче заказчику технических и иных документовпо**

**завершению и сдаче оказанных услуг:**

**4.9 Требования по объему гарантий качества оказываемых услуг**

**4.10Требования к траспортабелности**

**4.11Требования к эксплуатации техническому обслуживанию ремонту и хранению компонентов системы**

**4.12Требования к защите информации от несанкционированного доступа**

**4.13Требования по сохранности информации при авариях**

**4.14Требования к защите от влияния внешнего воздействия**

**4.15Требования к патентной и лицензионной чистоте состав и содержание работ по созданию системы**

1. **Состав и содержание работ по созданию систем**

1. **Порядок приёма и заказа:**

**7.Структура каталога товаров :**

**8. Контент:**

**9. Между народный ГОСТ:**

**1.1 Полное и краткое наименования информационной системы**

**Салон оптики «Одежда для глаз»**

**1.2Структура проекта (вплоть до мелочей).**

**Необходимо изложить ВСЮ структуру проекта. Начиная от главной страницы и заканчивая самыми мелкими страницами. Соблюдение этого правила позволит ВСЕМ участникам проекта (дизайнеры, программисты, менеджеры и пр.) всегда понимать объем работы и иметь перед глазами структуру объекта 1.3 Сроки сдачи проекта:**

**Техническое задание следует разделить на этапы, чтобы не думать о той далекой дате, когда будет завершен весь проект, а работать этапами и более четко понимать текущий статус работ и сроки перехода на следующий этап разработки сайта согласно техническому заданию. Если подразумевается работа как заказчика (подготовка контента, например), так и исполнителя, то следует указать, что из-за несвоевременности подачи информации заказчиком, исполнитель имеет право сдвинуть сроки сдачи проекта на соответствующее время.**

* 1. **Шифр темы или номера:**

**2. Цели и задачи проекта :**

**Здесь следует разместить определенные цели и задачи проекта. Важно это сделать не пространно, например, “Создание интернет магазина с возможностью онлайн заказа товаров”, а подробно описать то, для чего создается ваш интернет магазин и какие задачи он должен выполнять.**

## **2.1** **Распределение работ между Исполнителем и Заказчиком.**

**Здесь все зависит от условий договора. Обычно мы размещаем всю информацию, которую готовим сами, а заказчик готовит каталог товаров, размещает информацию и фотографии товаров в системе администрирования. Однако возможен и вариант когда Заказчик предоставляет информацию, а Исполнитель выполняет все работы по её размещению.**

* 1. **Распределение работ между Исполнителем и Заказчиком:**

**Размещение материалов происходит через систему администрирования. Материалы, подготовленные Заказчиком размещаются силами Заказчика. Материалы, подготовленные Исполнителем, размещаются силами Исполнителя.**

**3. Дизайн:**

* 1. **Заполненный заказчиком бриф лист.**

**Бриф лист следует заполнить максимально подробно, так как это сразу поможет понять примерный объем работы. Вы сможете скачать бриф лист в конце статьи в составе технического задания**.

* 1. **Функциональные прототипы страниц:**
  2. **Очень важный пункт, к которому стоит отнестись с должным вниманием. Мы все понимаем,**

**что точек входа на приложение может быть несколько. Пользователь может прийти к вам сразу на страницу товара и не увидеть, что доставка у вас бесплатная, а за заказ от 7 вы еще и подарок получите. А эти вещи, как вы понимаете, сильно мотивирую потенциального покупателя совершить заказ. Так вот суть этого пункта в том, что вы должны четко понимать как максимально эффективно склонить покупателя к покупке вне зависимости от того, какая точка входа была использована.**

**4 Требования:**

**Пункт, в котором прописываются все требования, которые должны быть исполнены для работы сайта и к человеку, который должен заниматься его администрированием**

**4.2Требования к администратору приложения.**

**Эти требования прописываются для того, чтобы Заказчики понимали, что работа с приложению требует определенных навыков и не так проста как кажется.**

**Как вариант:  
– Базовое знание HTML;  
– Обработка фотографий (Photoshop/…);  
– Умение работать с базой данных, административным функционалом приложении;  
– Понимание основ маркетинга и SEO.**

**4.3Системные требования к хостингу.**

**Основные требования к хостингу могут выглядеть следующим образом:  
PHP 5.0.5—5.2.x  
MySQL 4.1 и выше, 5.x  
Необходимые расширения PHP:  
4.4 Требования к Мобильному приложению.**

**Для работы Приложения требуется смартфон или компьютер с установленной операционной системой Android 5.0.1 (или более новой).**

**Для установки приложения требуется не менее 20 Мб свободного места в постоянной памяти устройства .**

**Для установки и обновления Приложения необходимо подключение к сети Интернет.  
Для пользование приложением потребуется одно разовая регистрация**

**прохождения регистрации потребуется вести Номер телефона,email,   
Ф.И.О заказчика**

**Для пользования системой оплаты потребуется привязка банковской карты**

**Главная страница мобильного приложения должна содержать графическую часть и меню сайта (у юзера появляются списки меню в боковой части приложения)**

**После регистрации у юзера формируется личный кабинет. У юзера списки меню в нижней части приложения и в боковой части, который открывается с помощью слайдера пальцем**

Ремонт

очков

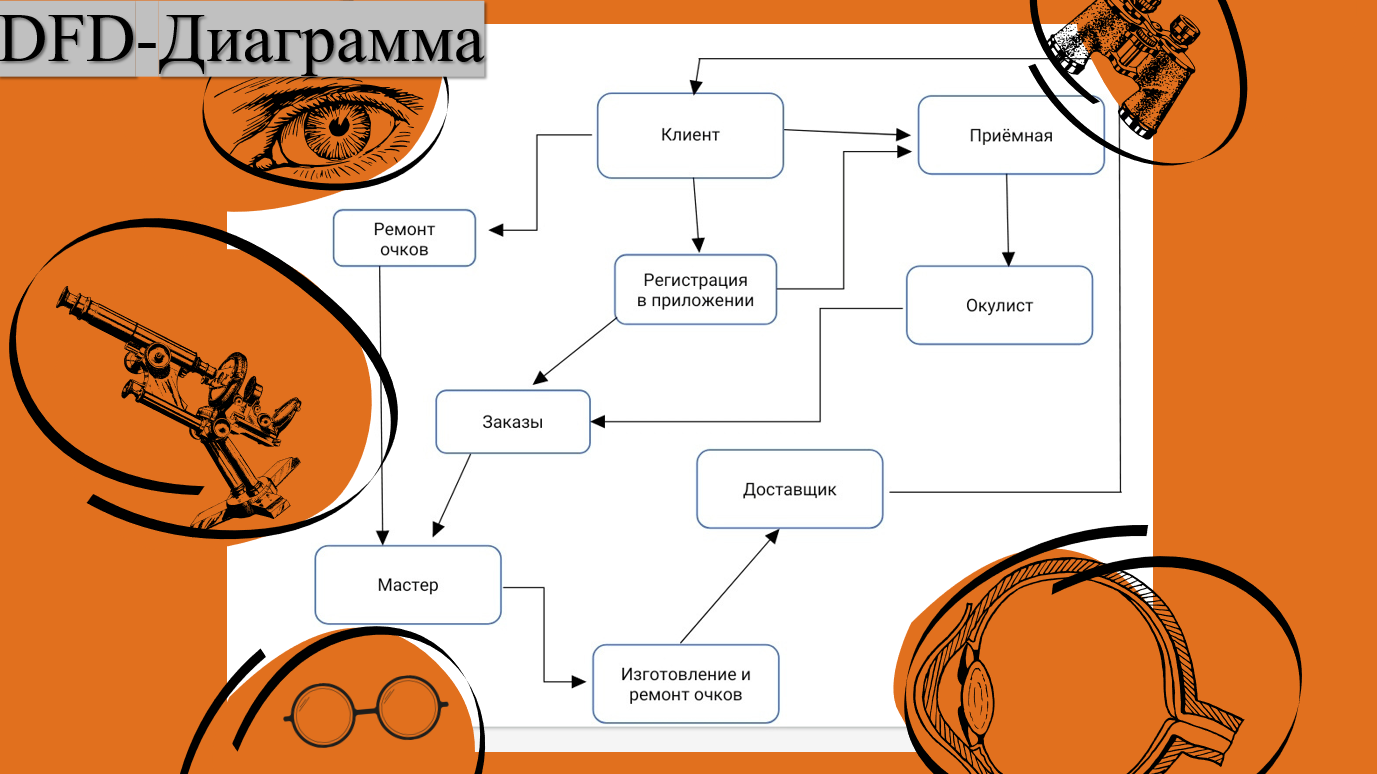
Регистрация

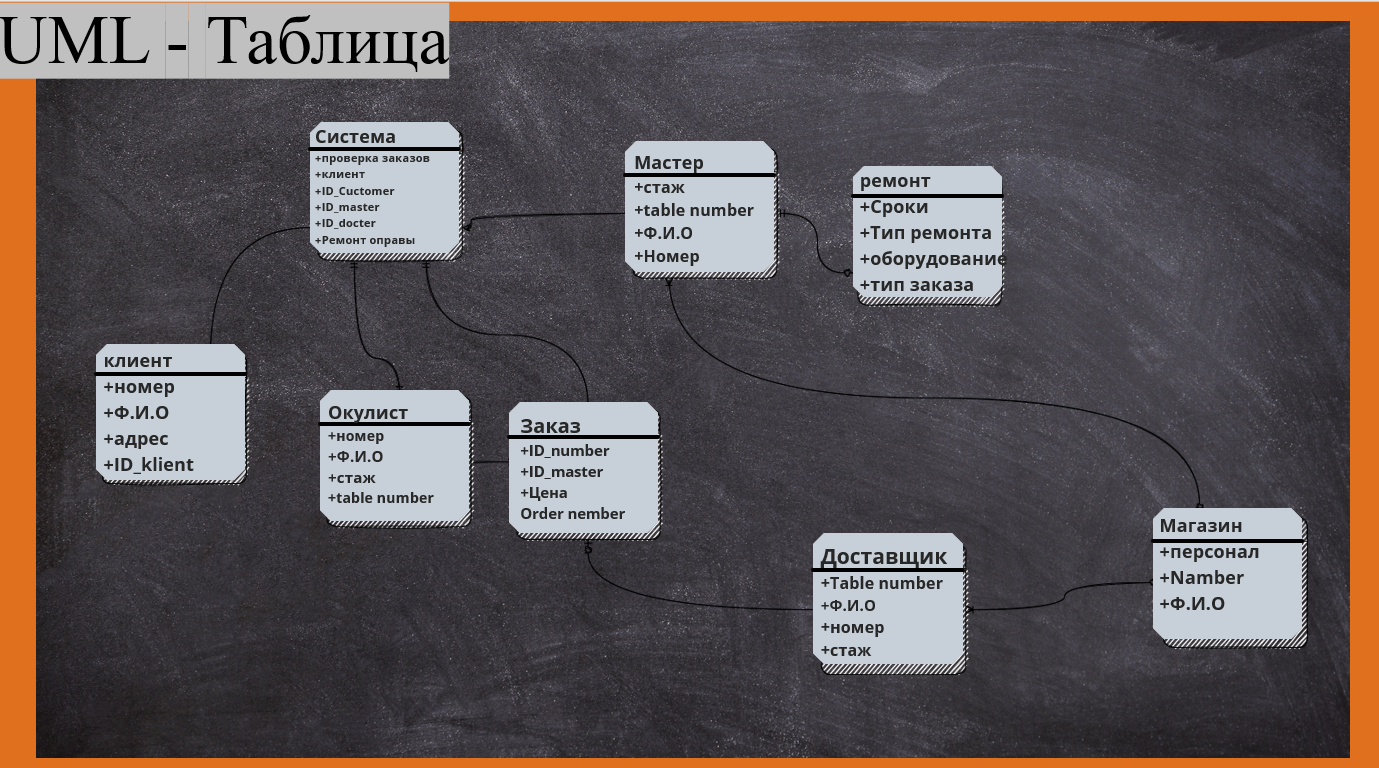
в приложении

Приёмная

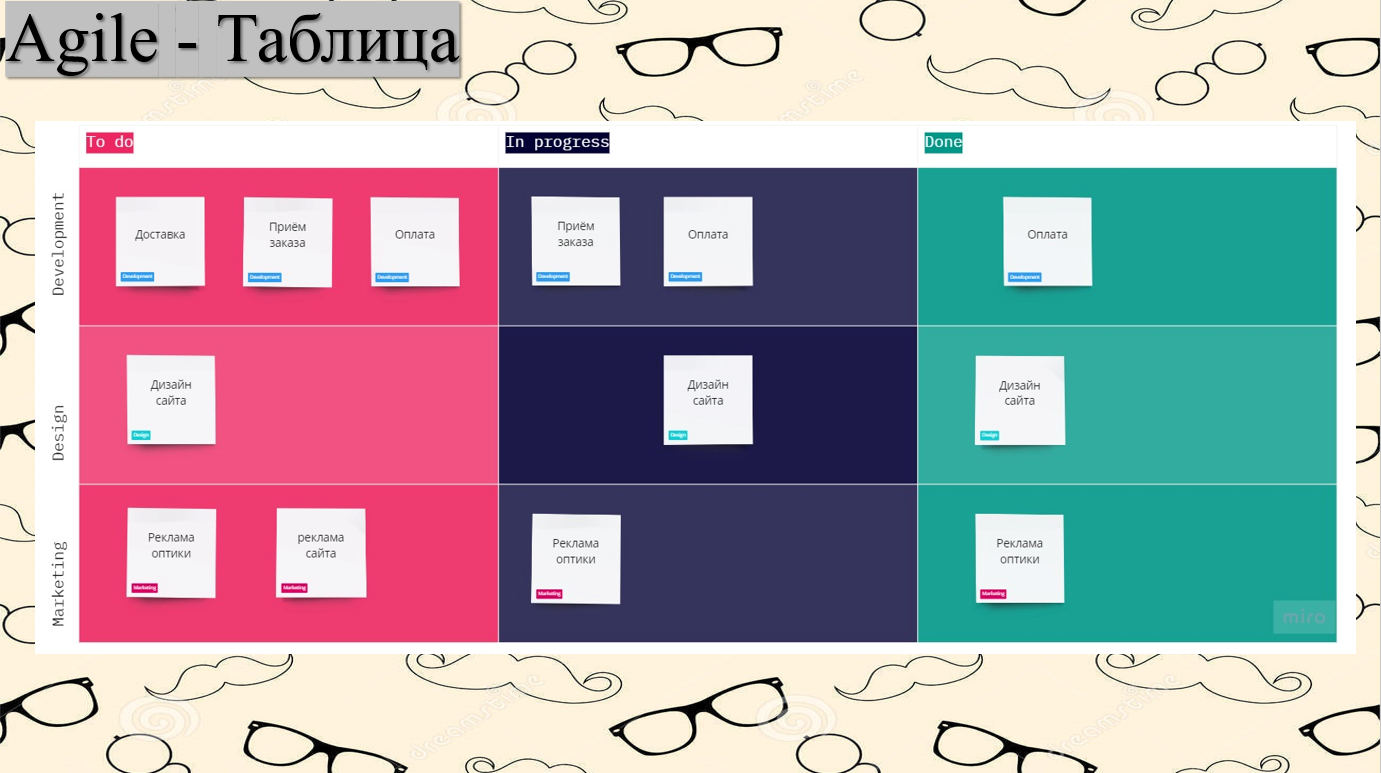
Клиент

Окулист









from tkinter import \*

import tkinter as tk

class SampleAPP(tk.Tk):

def \_\_init\_\_(self, \*arg, \*\*kwargs):

tk.Tk.\_\_init\_\_(self, \*arg, \*\*kwargs)

container = tk.Frame(self)

container.pack(side="top", fill="both", expand=True)

container.grid\_rowconfigure(0, weight=1)

container.grid\_columnconfigure(0, weight=1)

self.frames = {}

for F in (MenuPage, StartPage, LoginPage, RePage):

page\_name = F.\_\_name\_\_

frame = F(parent=container, controller=self)

self.frames[page\_name] = frame

frame.grid(row=0, column=0, sticky="nsew")

self.show\_frame("MenuPage")

def show\_frame(self, page\_name):

frame = self.frames[page\_name]

frame.tkraise()

class StartPage(tk.Frame):

def \_\_init\_\_(self, parent, controller):

tk.Frame.\_\_init\_\_(self, parent)

self.controller = controller

self.controller.title("")

self.backGroundImage = PhotoImage(file=r"isl.png")

self.backGroundImageLabel = Label(self, image=self.backGroundImage)

self.backGroundImageLabel.place(x=0, y=0)

def show\_frame(self, page\_name):

frame = self.frames[page\_name]

frame.tkraise()

def register\_user():

username\_info = username\_entry.get()

password\_info = password\_entry.get()

file = open(username\_info + ".txt", "w")

file.write(username\_info + "\n")

file.write(password\_info)

file.close()

username\_entry.delete(0, "end")

password\_entry.delete(0, "end")

tk.Label(self,

text="Вы зарегистрированы",

fg="green",

font=("Arial", 11)).pack()

def return\_page():

controller.show\_frame('RePage')

tk.Label(self,

text="Введите свои данные",

height="2",

font=("Arial", 20)).pack()

tk.Label(self, text="Имя (клиента) \* ").pack()

username\_entry = tk.Entry(self)

username\_entry.pack()

tk.Label(self, text="Ваш email \* ").pack()

password\_entry = tk.Entry(self)

password\_entry.pack()

tk.Button(self,

text="Зарегистрироваться",

width=20,

height=1,

command=return\_page).pack()

tk.Button(self, text="Выход", width=20, height=1,

command=return\_page).pack()

class LoginPage(tk.Frame):

def \_\_init\_\_(self, parent, controller):

tk.Frame.\_\_init\_\_(self, parent)

self.controller = controller

self.backGroundImage = PhotoImage(file=r"isl.png")

self.backGroundImageLabel = Label(self, image=self.backGroundImage)

self.backGroundImageLabel.place(x=0, y=0)

def return\_page():

controller.show\_frame('StartPage')

tk.Label(self,

text="Введите ваш логин",

height="2",

font=("Arial", 20)).pack()

tk.Label(self, text="").pack()

tk.Label(self, text="Имя", bg="white").pack()

username\_entry = tk.Entry(self)

username\_entry.pack()

tk.Button(self, text="Войти", width=20, height=1,

command=return\_page).pack()

tk.Button(self, text="Выход", width=20, height=1,

command=return\_page).pack()

class RePage(tk.Frame):

def \_\_init\_\_(self, parent, controller):

tk.Frame.\_\_init\_\_(self, parent)

self.controller = controller

self.backGroundImage = PhotoImage(file=r"isl.png")

self.backGroundImageLabel = Label(self, image=self.backGroundImage)

self.backGroundImageLabel.place(x=0, y=0)

def return\_page():

controller.show\_frame('StartPage')

tk.Button(self,

text="Главная",

width=20,

height=1,

command=return\_page).pack()

tk.Button(self, text="Услуги", width=20, height=1,

command=return\_page).pack()

tk.Button(self,

text="окулист",

width=20,

height=1,

command=return\_page).pack()

tk.Button(self,

text="мастера",

width=20,

height=1,

command=return\_page).pack()

tk.Button(self,

text="Доставка",

width=20,

height=1,

command=return\_page).pack()

tk.Button(self,

text="Личный кабинет",

width=20,

height=1,

command=return\_page).pack()

tk.Button(self, text="О нас", width=20, height=1,

command=return\_page).pack()

class MenuPage(tk.Frame):

def \_\_init\_\_(self, parent, controller):

tk.Frame.\_\_init\_\_(self, parent)

self.controller = controller

self.backGroundImage = PhotoImage(file=r"isl.png")

self.backGroundImageLabel = Label(self, image=self.backGroundImage)

self.backGroundImageLabel.place(x=0, y=0)

def return\_page():

controller.show\_frame('StartPage')

def return\_login():

controller.show\_frame('LoginPage')

label5 = tk.Label(self,

text="Добро пожаловать!",

width="300",

height="2",

font=("Arial", 20)).pack(pady=30)

tk.Button(self,

text="Логин",

height="2",

width="30",

command=return\_login).pack(pady=30)

tk.Button(self,

text="Регистрация",

height="2",

width="30",

command=return\_page).pack(pady=30)

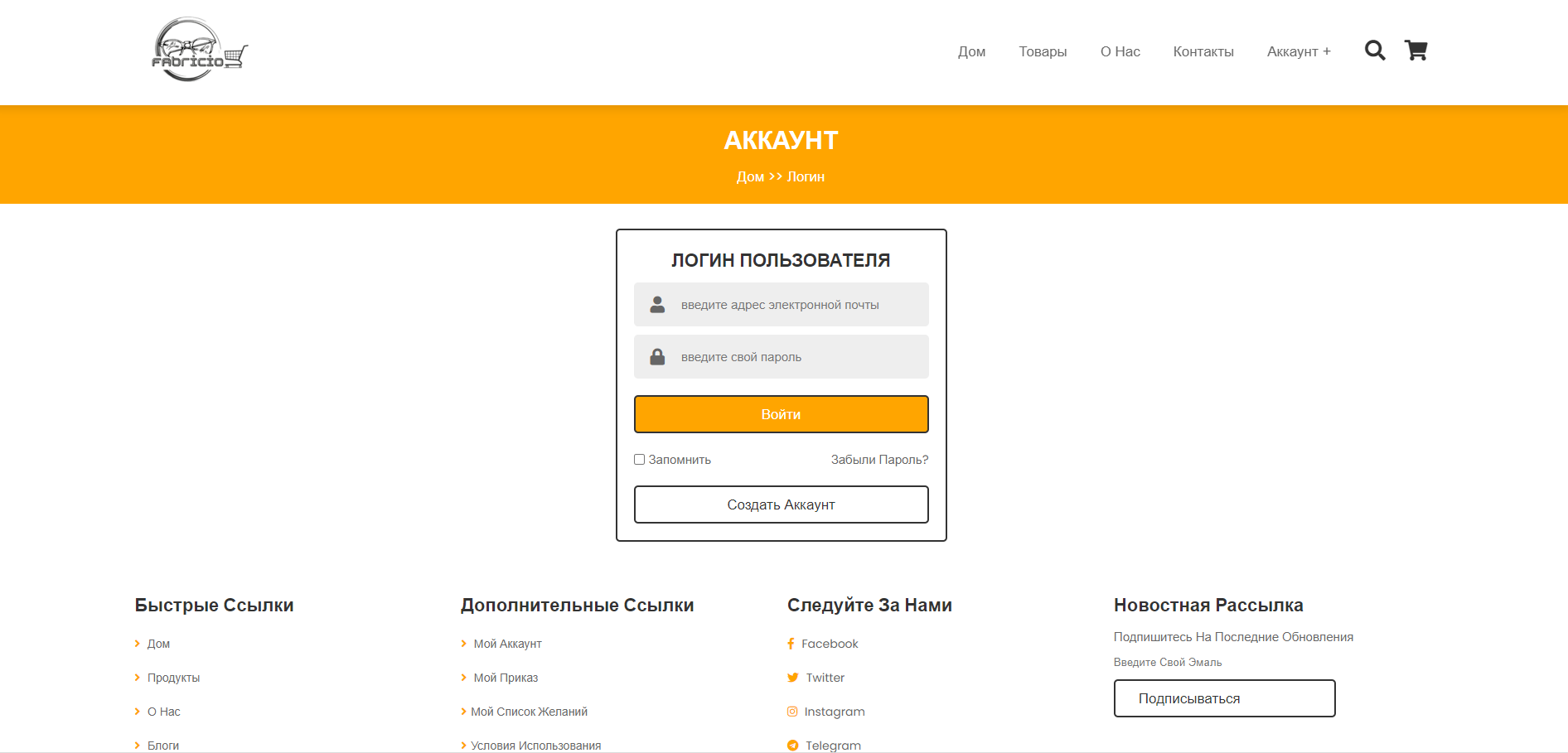
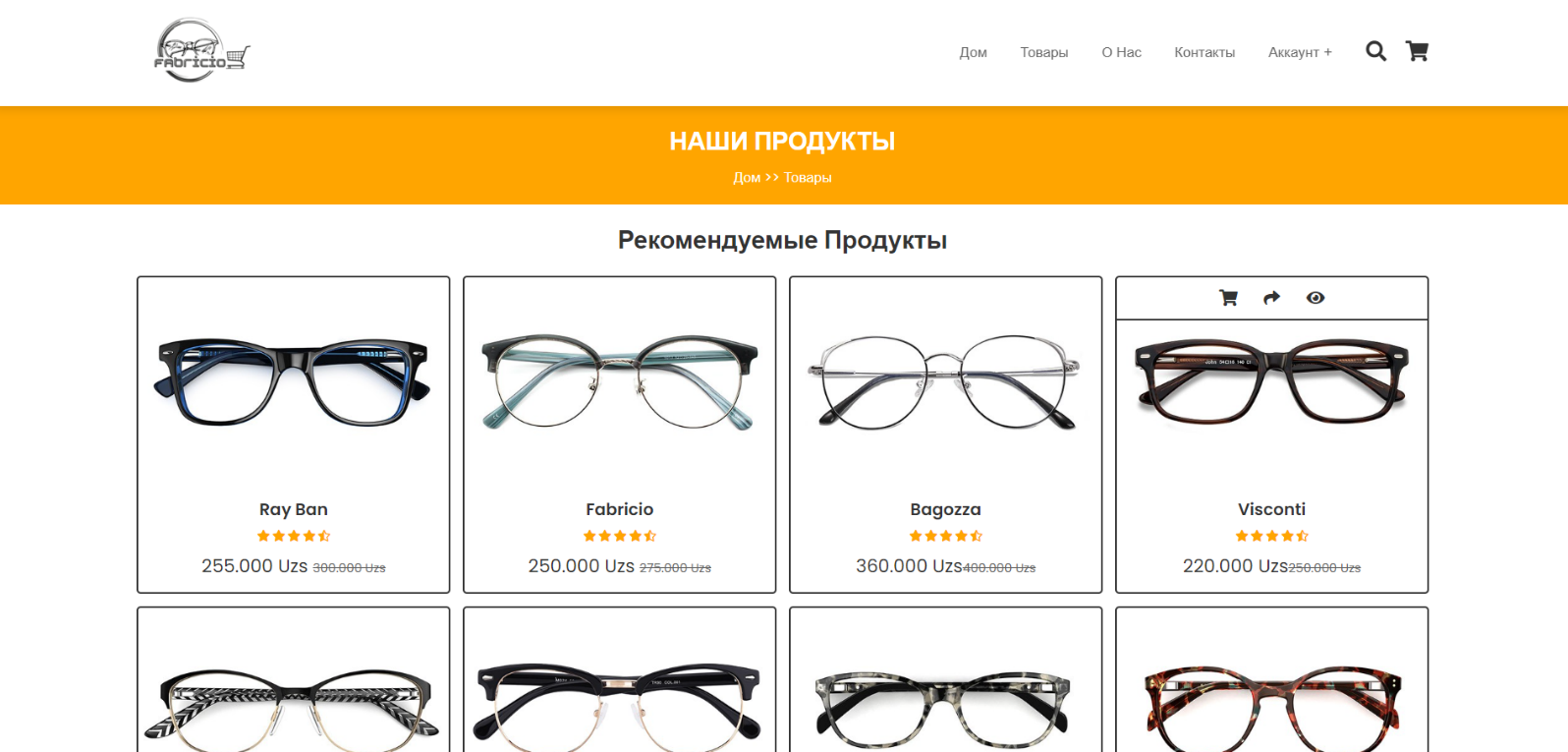
# функция Регистрацмя

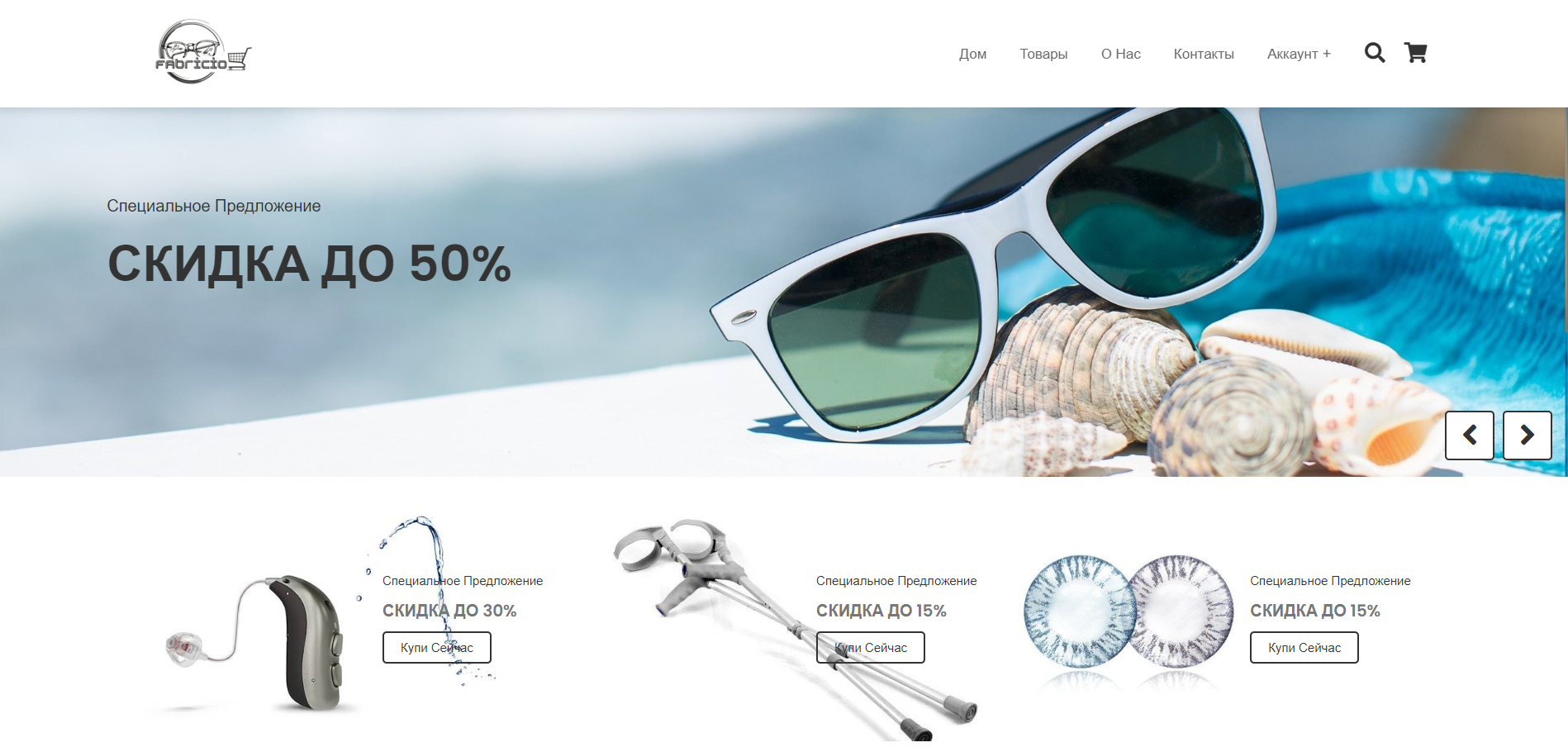
if \_\_name\_\_ == "\_\_main\_\_":

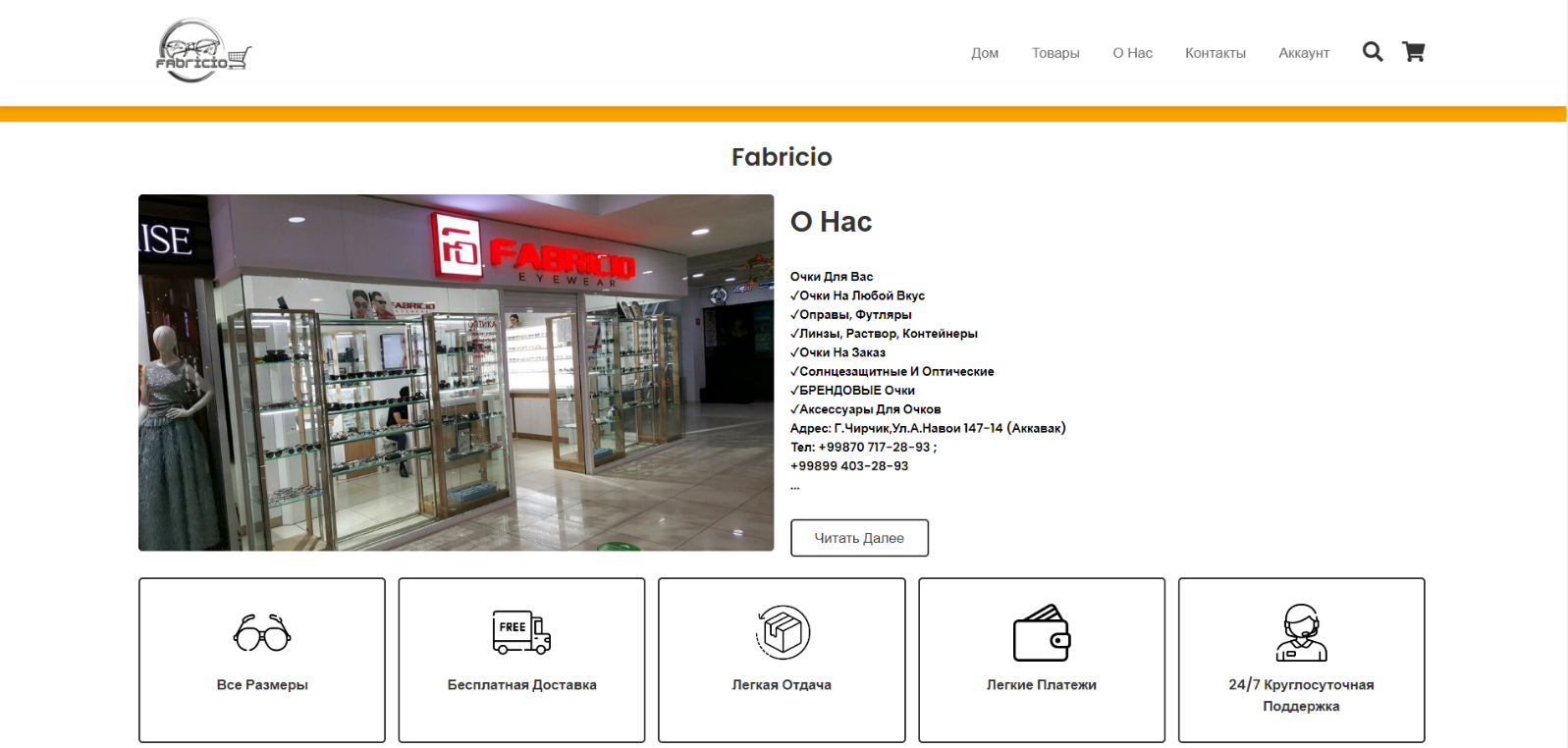
app = SampleAPP()

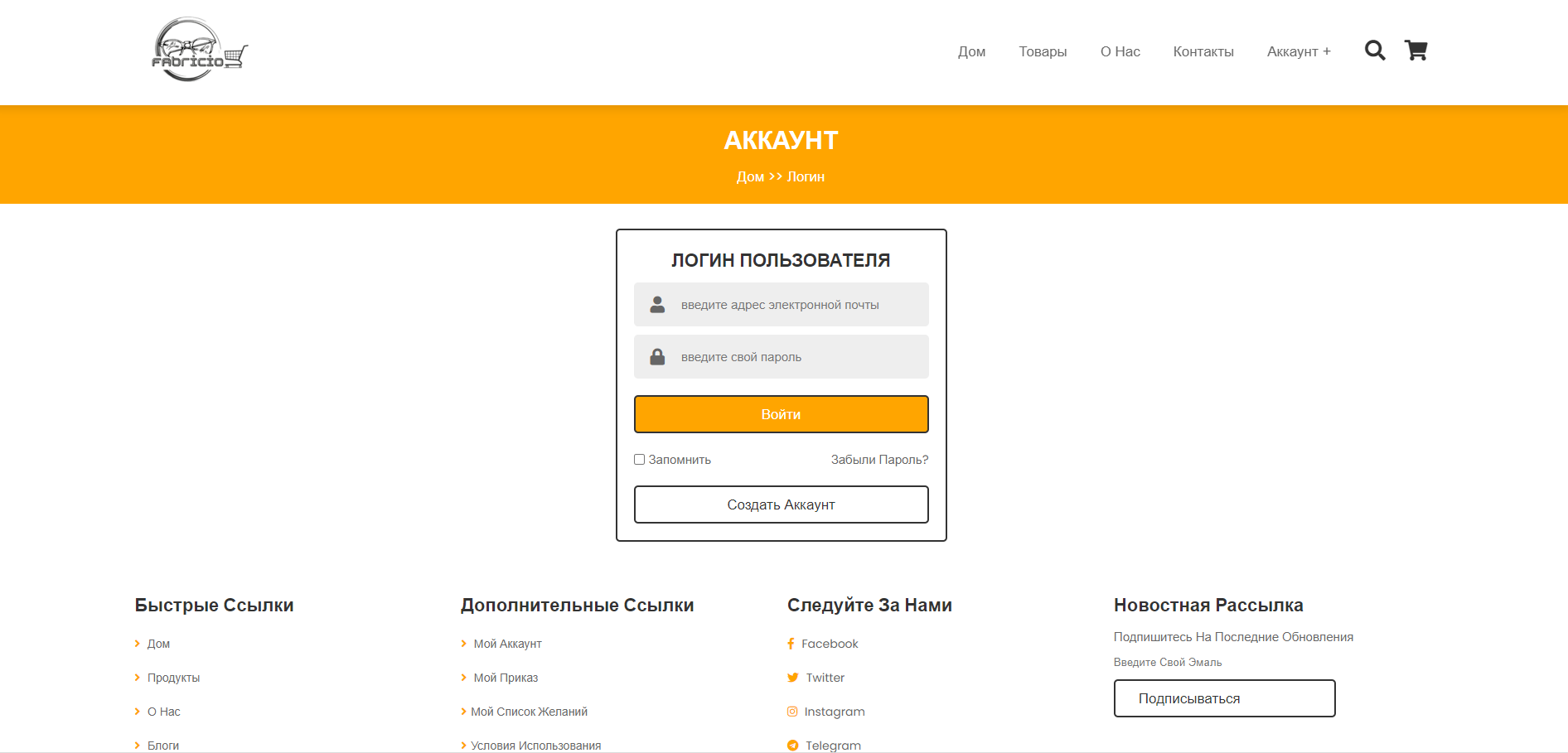
app.geometry('600x500')

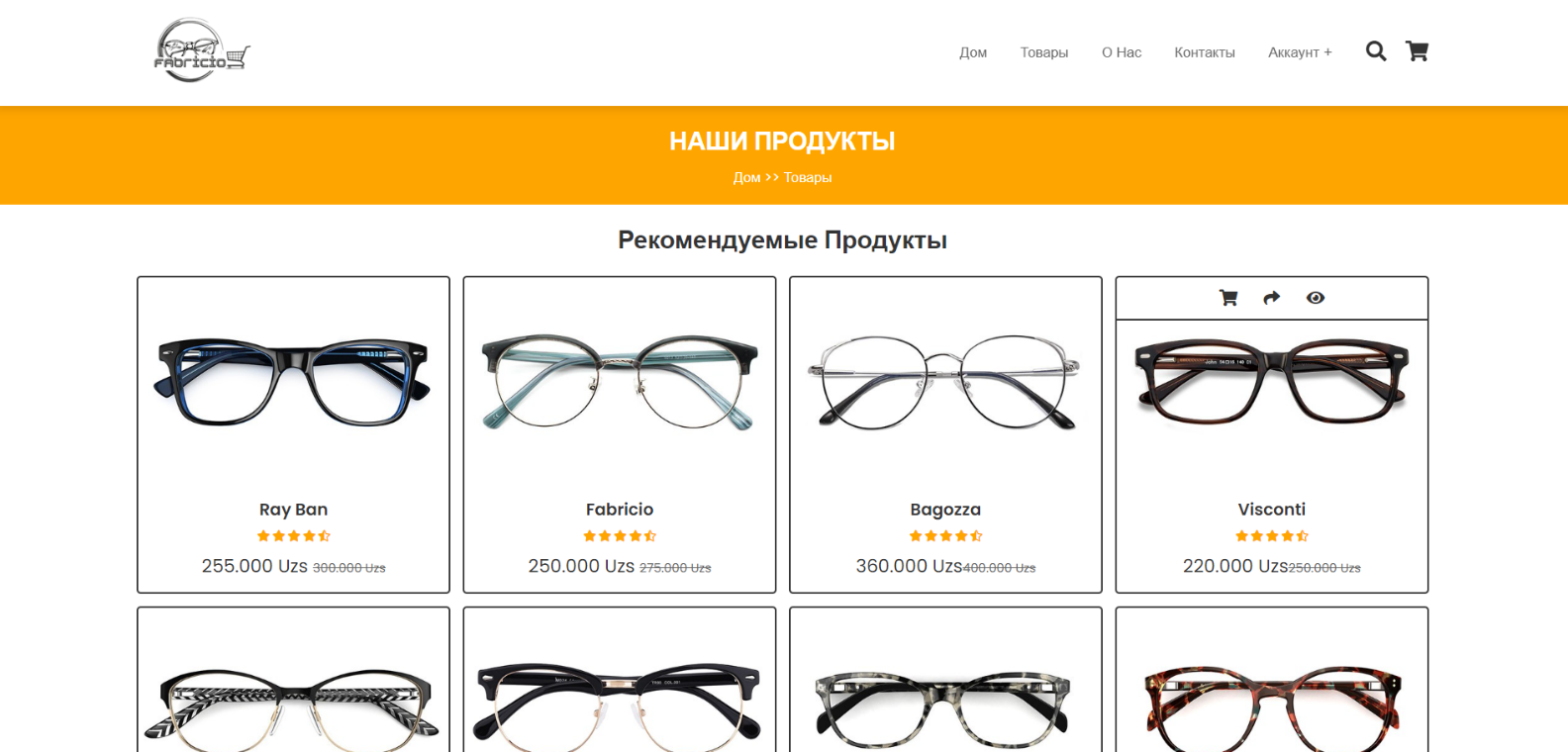
app.mainloop()











.YOU TUBE

