import streamlit as st

# --- НОВАЯ, БОЛЕЕ НАДЁЖНАЯ ФУНКЦИЯ ДЛЯ КОПИРОВАНИЯ ---

# Страница Входа

# ИСПРАВЛЕННАЯ Страница Входа

def login\_page():

st.header("Вход или Регистрация")

# Поле для ввода имени (логина). Оно теперь единственное.

username = st.text\_input("Введите ваше имя (Например, Константинов Ярослав)")

# Добавляем кнопку "Войти"

if st.button("Войти / Зарегистрироваться"):

# Новая проверка: просто смотрим, ввел ли пользователь имя.

if username: # Если поле username не пустое

st.session\_state['logged\_in'] = True

st.session\_state['username'] = username

# st.rerun() перезагрузит страницу и покажет калькулятор.

# Ваша функция get\_user\_state() сама создаст "личный блокнот", если нужно.

st.rerun()

else:

# Если пользователь ничего не ввел, вежливо просим его это сделать.

st.warning("Пожалуйста, введите имя, чтобы продолжить.")

# --- Данные о продуктах ---

PRODUCTS\_DK = [

"ДК", "Акт", "Трз", "Комбо/Кросс КК", "ЦП", "Смарт", "Кешбек", "ЖКУ", "БС",

"Инвесткопилка", "БС со Стратегией", "Токенизация", "Накопительный Счет",

"Вклад", "Детская Кросс", "Стикер Кросс", "Сим-Карта",

"Селфи ДК", "Селфи КК"

]

PRODUCTS\_KK = [

"КК", "Акт", "Трз", "Кросс ДК", "ЦП", "Смарт", "Кешбек", "ЖКУ", "БС",

"Инвесткопилка", "БС со Стратегией", "Токенизация", "Накопительный Счет",

"Вклад", "Детская Кросс", "Стикер Кросс", "Сим-Карта",

"Селфи ДК"

]

PRODUCTS\_MP = [

"МП", "ЦП", "Смарт", "Кешбек", "ЖКУ", "БС", "Инвесткопилка",

"БС со Стратегией", "Токенизация", "Накопительный Счет", "Вклад",

"Детская Кросс", "Стикер Кросс", "Сим-Карта", "Кросс ДК",

"Селфи ДК", "Селфи КК"

]

PRODUCT\_LISTS = {

"ДК": PRODUCTS\_DK,

"КК": PRODUCTS\_KK,

"МП": PRODUCTS\_MP

}

# --- Стили ---

# ИСПРАВЛЕННАЯ ВЕРСИЯ ФУНКЦИИ. СКОПИРУЙТЕ И ЗАМЕНИТЕ ЕЮ СТАРУЮ.

# ИСПРАВЛЕННАЯ ВЕРСИЯ ФУНКЦИИ СТИЛЕЙ. СКОПИРУЙТЕ И ЗАМЕНИТЕ ЕЮ СТАРУЮ.

def apply\_styles():

st.markdown("""

<style>

.main { background-color: #FFFFFF; }

/\* --- НОВЫЕ ПРАВИЛА ДЛЯ "КОРОБКИ" С КНОПКАМИ --- \*/

/\* 2. Ставим "коробку" ровно по центру с помощью магии auto-отступов \*/

margin-left: auto;

margin-right: auto;

margin-top: 20px;

/\* 3. Ограничиваем максимальную ширину на очень больших экранах,

чтобы кнопки не были гигантскими. \*/

max-width: 500px;

}

/\* --- НОВЫЕ ПРАВИЛА ДЛЯ САМИХ КНОПОК ВНУТРИ "КОРОБКИ" --- \*/

/\* --- ПРАВИЛА ДЛЯ МАЛЕНЬКИХ ЭКРАНОВ (ТЕЛЕФОНОВ) --- \*/

/\* Эта инструкция сработает, только если ширина экрана 600px или меньше \*/

@media (max-width: 600px) {

.main-menu-container {

/\* На маленьких экранах делаем "коробку" чуть шире для удобства \*/

width: 95%;

}

}

/\* --- Остальные стили для других элементов оставляем без изменений --- \*/

div.stButton > button {

height: 50px;

border: 1px solid #CCCCCC;

border-radius: 8px;

background-color: #FFFFFF;

color: #000000;

font-family: 'Calibri', sans-serif;

font-size: 16px;

text-align: center;

}

div.stButton > button:hover {

background-color: [jg:пароль\_(regexp)\_150]

border-color: #AAAAAA;

}

.stToggle { font-family: 'Calibri', sans-serif; color: #000000; }

.report-text {

font-family: 'Calibri', sans-serif;

font-size: 16px;

line-height: 1.8;

border: 1px solid #DDDDDD;

padding: 15px;

border-radius: 8px;

white-space: pre-wrap;

background-color: #FAFAFA;

}

</style>

""", unsafe\_allow\_html=True)

# --- Логика состояний (сессии) для МНОГИХ пользователей ---

def initialize\_global\_state():

# Это создает "полку" для хранения блокнотов всех пользователей

if 'logged\_in' not in st.session\_state:

st.session\_state['logged\_in'] = False

if 'username' not in st.session\_state:

st.session\_state['username'] = None

if 'user\_data' not in st.session\_state:

st.session\_state['user\_data'] = {}

def get\_user\_state():

# Эта функция находит на "полке" личный блокнот текущего сотрудника

username = st.session\_state['username']

if username not in st.session\_state['user\_data']:

st.session\_state['user\_data'][username] = {

'page': 'main',

'toggles': {},

'report\_text': ""

}

return st.session\_state['user\_data'][username]

def logout():

# Эта функция "закрывает сессию" - отправляет пользователя обратно к экрану входа

st.session\_state['logged\_in'] = False

st.session\_state['username'] = None

st.rerun()

# --- Функции для переключения страниц (НОВЫЕ ВЕРСИИ) ---

def go\_to\_page(page\_name):

# Теперь эта функция меняет страницу в ЛИЧНОМ блокноте пользователя

user\_state = get\_user\_state()

user\_state['toggles'] = {}

user\_state['page'] = page\_name

def go\_to\_main():

# То же самое для возврата на главную

user\_state = get\_user\_state()

user\_state['toggles'] = {}

user\_state['page'] = 'main'

def reset\_all():

# И сброс работает только для текущего пользователя

user\_state = get\_user\_state()

user\_state['page'] = 'main'

user\_state['toggles'] = {}

user\_state['report\_text'] = ""

# --- Логика генерации отчета ---

def generate\_report\_text(main\_product, toggles):

product\_list = PRODUCT\_LISTS.get(main\_product.upper())

if not product\_list:

return ""

report\_lines = [f"{product} {'+' if toggles.get(product, False) else '-'}" for product in product\_list]

return "\n".join(report\_lines)

# --- Страницы приложения ---

# НОВАЯ, НАДЕЖНАЯ ВЕРСИЯ ФУНКЦИИ main\_page

def main\_page():

st.header("Выберите основной продукт")

# 1. Создаем наш "умный стеллаж" из трёх колонок.

# Мы делим ширину в пропорции 1:4:1.

# Это значит, что центральная колонка будет в 4 раза шире боковых.

# Боковые колонки будут пустыми "распорками".

left\_space, main\_content, right\_space = st.columns([1, 4, 1])

# 2. Теперь мы говорим: "Всё, что дальше, клади в центральную колонку".

with main\_content:

# 3. Создаем наши кнопки.

# Ключевой параметр use\_container\_width=True заставляет кнопку

# растянуться на ВСЮ ширину своей колонки.

st.button(

"ДК",

on\_click=go\_to\_page,

args=('dk',),

use\_container\_width=True

)

st.button(

"КК",

on\_click=go\_to\_page,

args=('kk',),

use\_container\_width=True

)

st.button(

"МП",

on\_click=go\_to\_page,

args=('mp',),

use\_container\_width=True

)

# Правильная версия страницы с продуктами

def product\_submenu\_page(product\_type, product\_list):

# 1. Сначала получаем записную книжку текущего пользователя

user\_state = get\_user\_state()

st.header(f"Дополнительные продукты для «{product\_type}»")

for product in product\_list:

# 2. Работаем с галочками из ЕГО книжки

user\_state['toggles'][product] = st.toggle(

product,

value=user\_state['toggles'].get(product, False),

key=f"{st.session\_state.username}\_{product\_type}\_{product}" # Уникальный ключ для каждого пользователя!

)

st.divider()

col1, col2 = st.columns(2)

with col1:

if st.button("Сформировать отчет"):

# 3. Генерируем отчет на основе ЕГО галочек

user\_state['report\_text'] = generate\_report\_text(product\_type, user\_state['toggles'])

user\_state['page'] = 'report'

st.rerun()

with col2:

st.button("Вернуться", on\_click=go\_to\_main)

# Правильная версия страницы отчета

# ИСПРАВЛЕННАЯ ВЕРСИЯ страницы отчета

# ИСПРАВЛЕННАЯ И ПРОСТАЯ ВЕРСИЯ СТРАНИЦЫ ОТЧЕТА

def report\_page():

# 1. Сначала получаем записную книжку текущего пользователя

user\_state = get\_user\_state()

st.header("Сформированный отчет")

# 2. Берем текст отчета из ЕГО книжки

report\_text = user\_state.get('report\_text', "Отчет пуст.")

# 3. ВОТ ГЛАВНОЕ ИЗМЕНЕНИЕ!

# Мы используем стандартный элемент Streamlit "текстовое поле".

# Он идеально подходит для отображения и копирования текста.

st.text\_area(

label="Отчет для копирования:",

value=report\_text,

height=300, # Можете подобрать удобную высоту

help="Нажмите на текст, затем Ctrl+C (или Cmd+C), чтобы скопировать"

)

# 4. Кнопку "Сбросить" оставляем, но ей больше не нужны колонки.

st.button("Сбросить", on\_click=reset\_all)

# --- Главная функция приложения ---

def main():

apply\_styles()

initialize\_global\_state()

# Сначала проверяем, вошел ли пользователь в систему

if not st.session\_state.get('logged\_in'):

login\_page() # Если не вошел, показываем страницу входа

else:

# Если вошел, то показываем калькулятор

# Добавим сбоку имя пользователя и кнопку "Выйти" для удобства

st.sidebar.success(f"Вы вошли как: {st.session\_state.username}")

st.sidebar.button("Выйти", on\_click=logout)

# Получаем личную "записную книжку" текущего пользователя

user\_state = get\_user\_state()

# А теперь показываем нужную страницу калькулятора

if user\_state['page'] == 'main':

main\_page()

elif user\_state['page'] == 'dk':

product\_submenu\_page("ДК", PRODUCTS\_DK)

elif user\_state['page'] == 'kk':

product\_submenu\_page("КК", PRODUCTS\_KK)

elif user\_state['page'] == 'mp':

product\_submenu\_page("МП", PRODUCTS\_MP)

elif user\_state['page'] == 'report':

report\_page()

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

main()