

Представляю Вашему вниманию бота в Telegram «Экранные истории». Автор проекта – Белявцева Екатерина БСБО-06-22.

Представьте себе бота, который становится вашим персональным гидом в мире кинематографа и анимации. Этот бот - не просто помощник, а настоящий проводник в бескрайнем океане информации о ваших любимых фильмах, сериалах, мультфильмах и мультсериалах. Благодаря интуитивно понятной системе меню, Вы сможете легко и быстро находить самые свежие новости и принимать участие в обсуждениях по интересующим Вас темам.

Главное меню бота приветствует Вас четырьмя основными категориями: Фильмы, Сериалы, Мультфильмы и Мультсериалы. Одним нажатием кнопки Вы погружаетесь в нужную Вам категорию, где перед Вами открываются тщательно подобранные подкатегории, такие как "Кино вселенная Marvel", "Звёздные войны", "Очень странные дела" и другие.

Подменю каждой категории предлагает три ключевые опции: Новости, Обсуждения и Назад. Нажимая на кнопку "Новости", Вы сразу же получаете актуальную информацию о новинках и событиях в мире вашего любимого фильма или сериала. Если же Вам интересны мнения и дискуссии других поклонников, кнопка "Обсуждения" покажет, где уже кипят обсуждения или подскажет, что они скоро начнутся.

Возвращение в главное меню или в предыдущую категорию также просто благодаря кнопке "Назад", которая делает навигацию по боту удобной и интуитивно понятной.

Этот бот создан для того, чтобы сделать ваш опыт взаимодействия с миром кино и анимации максимально комфортным и информативным. Теперь, чтобы быть в курсе всех новостей и обсуждений, Вам достаточно всего нескольких кликов. Погружайтесь в мир развлечений с нашим ботом и наслаждайтесь каждым мгновением!

При написании бота были использованы несколько интересных приёмов:

1. Использование функции для генерации случайных новостей: это позволяет добавить некоторую случайность и уникальность в бота, предоставляя пользователям разнообразную информацию о выбранных киновселенных. В отдельном файле для каждой категории присутствует свой Список дат, персонажей, возможных событий и мест, где происходит действие, и дальше с помощью функции random генерируется новость.

```
elif self == "OSD" and message.text == "Новости":
    for i in range(5):
        msg = bot.send_message(message.chat.id, news.generate_random_news_osd())
        messages_to_delete[message.chat.id].append(msg.message_id)
```

Рисунок 1. Генерация 5 различных новостей

2. Использование словарей для хранения истории и удаления сообщений: Использование словаря 'messages_to_delete' позволяет боту отслеживать управлять удалением сообщений, что полезно для поддержания чистоты в чате. Каждое отправленное сообщение получает свой id и записывается в словарь.

```
if message.chat.id not in messages_to_delete:
    messages_to_delete[message.chat.id] = []
msg = bot.send_message(message.chat.id, text="Добро пожаловать! Выберите категорию:", reply_markup=menu.main_menu())
messages_to_delete[message.chat.id].append(message.message_id) # Сохранить ID сообщения пользователя
messages_to_delete[message.chat.id].append(msg.message_id) # Сохранить ID сообщения бота
```

Рисунок 2. Пример использования словаря 'messages_to_delete' в коде

При вызове функции удаляются все сообщения, которые попали в словарь. Функция вызывается при выходе в главное меню любым способом.

```
elif message.text == "В главное меню":
    delete_all_messages(message.chat.id)
    msg = bot.send_message(message.chat.id, text="Возвращаемся в главное меню", reply_markup=menu.main_menu())
    messages_to_delete[message.chat.id].append(msg.message_id)
```

Рисунок 3. Удаление всех сообщений

3. Использование подобию декораторов для определения того, в каком меню находится пользователь. Эта функция работает с помощью глобальной переменной 'self', которой присваивается значение в виде названия меню, которое открывает пользователь. Именно благодаря этой функции работает кнопка «Назад», которая направляет пользователя в прошлое меню.

```
elif message.text == "Симпсоны":
    msg = bot.send_message(message.chat.id, text="Выбрана вселенная Симпсонов", reply_markup=menu.content_menu_simpsons())
    messages_to_delete[message.chat.id].append(msg.message_id)
    self = "SIMPSONS"
elif self == "SIMPSONS" and message.text == "Новости":
    for i in range(5):
        msg = bot.send_message(message.chat.id, news.generate_random_news_simpsons())
        messages_to_delete[message.chat.id].append(msg.message_id)
elif message.text == "Назад" and self in ("MARVEL", "WARS", "HP"):
    self = "MOVIE_MENU"
    msg = bot.send_message(message.chat.id, text="Вы в меню выбора фильмов", reply_markup=menu.movies_menu())
    messages_to_delete[message.chat.id].append(msg.message_id)
```

Рисунок 3. Пример использования декоратора ‘self’ в коде