From telegram import Bot, Update, InlineKeyboardButton, InlineKeyboardMarkup

From telegram.ext import Updater, CommandHandler, CallbackQueryHandler, MessageHandler, Filters

Import random

Users = {}

Photos = {}

Likes = {}

Def start(update: Update, context):

User\_id = update.message.from\_user.id

If user\_id not in users:

Users[user\_id] = {‘photo’: None, ‘seen’: []}

Reply\_markup = InlineKeyboardMarkup([

[InlineKeyboardButton(“Регистрация”, callback\_data=’register’)]

])

Update.message.reply\_text(“Привет, пуссак/пуссачиха! Покажи рыло!88982492”, reply\_markup=reply\_markup)

Def register(update: Update, context):

User\_id = update.callback\_query.from\_user.id

Update.callback\_query.message.reply\_text(“Отправь мне свою фотографию!”)

Def photo\_received(update: Update, context):

User\_id = update.message.from\_user.id

Photo\_id = update.message.photo[-1].file\_id

Users[user\_id][‘photo’] = photo\_id

Photos[user\_id] = photo\_id

Reply\_markup = InlineKeyboardMarkup([

[InlineKeyboardButton(“Знакомиться”, callback\_data=’browse’)],

[InlineKeyboardButton(“Хватит”, callback\_data=’stop’)]

])

Update.message.reply\_text(“Отлично! Теперь выбери действие:”, reply\_markup=reply\_markup)

Def browse\_photos(update: Update, context):

User\_id = update.callback\_query.from\_user.id

Unseen\_users = [u for u in photos if u != user\_id and u not in users[user\_id][‘seen’]]

If unseen\_users:

Random\_user = random.choice(unseen\_users)

Users[user\_id][‘seen’].append(random\_user)

Reply\_markup = InlineKeyboardMarkup([

[InlineKeyboardButton(“Лайк ❤️”, callback\_data=f’like\_{random\_user}’)],

[InlineKeyboardButton(“Дизлайк 🚫”, callback\_data=f’dislike\_{random\_user}’)],

])

Context.bot.send\_photo(chat\_id=user\_id, photo=photos[random\_user], reply\_markup=reply\_markup)

Else:

Update.callback\_query.message.reply\_text(“Вы просмотрели все фотографии 😅. Ждите новых участников!”)

Def like(update: Update, context):

User\_id = update.callback\_query.from\_user.id

Liked\_user = int(update.callback\_query.data.split(‘\_’)[1])

If liked\_user not in likes:

Likes[liked\_user] = []

Likes[liked\_user].append(user\_id)

If user\_id in likes and liked\_user in likes[user\_id]:

Context.bot.send\_message(chat\_id=user\_id, text=f”Ура! 🎉 Взаимный лайк с {liked\_user}. Начните общение!”)

Context.bot.send\_message(chat\_id=liked\_user, text=f”Ура! 🎉 Взаимный лайк с {user\_id}. Начните общение!”)

Else:

Update.callback\_query.answer(“Лайк учтен! Давайте продолжим.”)

Def main():

Updater = Updater(“YOUR\_TOKEN”, use\_context=True)

Dp = updater.dispatcher

Dp.add\_handler(CommandHandler(“start”, start))

Dp.add\_handler(CallbackQueryHandler(register, pattern=’^register$’))

Dp.add\_handler(MessageHandler(Filters.photo, photo\_received))

Dp.add\_handler(CallbackQueryHandler(browse\_photos, pattern=’^browse$’))

Dp.add\_handler(CallbackQueryHandler(like, pattern=’^like\_\d+$’))

Updater.start\_polling()

Updater.idle()

If \_\_name\_\_ == “\_\_main\_\_”:

Main()