

```
from telegram import Update
from telegram.ext import Application, CommandHandler, MessageHandler, filters, ContextTypes
import os
```

```
# Bot Token को environment variable से सुरक्षित तरीके से लें
BOT_TOKEN = os.environ.get('BOT_TOKEN')
```

```
# /start कमांड
```

```
async def start_command(update: Update, context: ContextTypes.DEFAULT_TYPE):
```

```
    welcome_text = ""
```

```
    नमस्ते! मैं एक Anonymous Message बॉट हूँ।
```

```
    मुझे कोई भी मैसेज भेजो, और मैं उसे anonymously आपके सब्सक्राइबर्स को भेज दूंगा।
```

```
    /off - अपनी सब्सक्रिप्शन को पॉज करने के लिए
```

```
    ""
```

```
    await update.message.reply_text(welcome_text)
```

```
# /off कमांड
```

```
async def off_command(update: Update, context: ContextTypes.DEFAULT_TYPE):
```

```
    await update.message.reply_text("आपकी सब्सक्रिप्शन पॉज कर दी गई है।")
```

```
# सामान्य मैसेज handle करना
```

```
async def handle_message(update: Update, context: ContextTypes.DEFAULT_TYPE):
```

```
    user_message = update.message.text
```

```
    await update.message.reply_text(f"आपका संदेश प्राप्त हुआ और anonymously भेज दिया गया: '{user_message}'")
```

```
# Errors
```

```
async def error(update: Update, context: ContextTypes.DEFAULT_TYPE):
```

```
    print(f"Update {update} caused error {context.error}")
```

```
# मुख्य function
```

```
if __name__ == '__main__':
```

```
    print("Starting bot...")
```

```
    app = Application.builder().token(BOT_TOKEN).build()
```

```
# Commands
```

```
app.add_handler(CommandHandler('start', start_command))
```

```
app.add_handler(CommandHandler('off', off_command))
```

```
# Messages
```

```
app.add_handler(MessageHandler(filters.TEXT, handle_message))
```

```
# Errors
app.add_error_handler(error)

print("Polling...")
app.run_polling(poll_interval=3)
```