# Customizable Chat App - Code Explanation

## 1. Importing Modules

import os  
from datetime import datetime

* os: Interact with the operating system (used for clearing console screen)
* datetime: Provides current date and time to timestamp messages

## 2. Clear Screen Function

def clear\_screen():  
 os.system('cls' if os.name == 'nt' else 'clear')

* Clears console screen based on operating system (Windows/Linux/Mac)

## 3. Save Chat Function

def save\_chat(chat\_history, filename):  
 with open(filename, 'a', encoding='utf-8') as file:  
 for line in chat\_history:  
 file.write(line + '\n')  
 print(f"\U0001F4BE Chat saved to {filename}\n")

* Saves chat history to a text file (appends without overwriting)

## 4. Get Timestamp Function

def get\_timestamp():  
 return datetime.now().strftime("%Y-%m-%d %H:%M:%S")

* Returns current date and time formatted for messages

## 5. Main Chat App Function

def chat\_app():  
 print("\U0001F4AC Welcome to the Customizable Chat App!")  
 print("Choose mode:")  
 print("1. Personal Chat \U0001F495")  
 print("2. Business Chat \U0001F4BC")  
 mode = input("Enter your choice (1 or 2): ")

* Displays welcome message and asks user to choose chat mode

## 6. Set Chat Mode

if mode == "1":  
 chat\_type = "Personal"  
 filename = "personal\_chat.txt"  
elif mode == "2":  
 chat\_type = "Business"  
 filename = "business\_chat.txt"  
else:  
 chat\_type = "Personal"  
 filename = "personal\_chat.txt"

* Determines chat type and file name based on user input

## 7. Get Usernames

user1 = input("Enter username for User 1: ")  
user2 = input("Enter username for User 2: ")

* Collects usernames for the two users

## 8. Chat Instructions

print("\nType your messages below! (Type 'exit' to quit, 'clear' to clear screen, 'save' to save chat)\n")  
chat\_history = []

* Explains special commands and initializes chat history list

## 9. Chat Loop

turn = 0  
while True:  
 current\_user = user1 if turn == 0 else user2  
 message = input(f"{current\_user}: ")

* Alternates turns between users

## 10. Check Commands

if message.lower() == 'exit':  
 break  
elif message.lower() == 'clear':  
 clear\_screen()  
 continue  
elif message.lower() == 'save':  
 save\_chat(chat\_history, filename)  
 continue

* Handles special commands: exit, clear, save

## 11. Format and Display Message

timestamp = get\_timestamp()  
if chat\_type == "Personal":  
 formatted\_message = f"[\U0001F495 {current\_user} at {timestamp}]: {message}"  
else:  
 formatted\_message = f"[{timestamp} | {current\_user}]: {message}"  
  
print(formatted\_message)  
chat\_history.append(formatted\_message)

* Adds timestamp and formats message depending on chat type

## 12. Switch Turns

turn = 1 - turn

* Alternates between User 1 and User 2

## 13. Save Chat on Exit

save\_chat(chat\_history, filename)

* Automatically saves chat at the end of the session

## 14. Run the App

if \_\_name\_\_ == "\_\_main\_\_":  
 chat\_app()

* Ensures the chat app runs only if script is executed directly

## Summary Table

| Section | Purpose |
| --- | --- |
| Modules | Clear screen, timestamp messages |
| Functions | Reusable code for commands and timestamps |
| chat\_app() | Controls main chat logic |
| Chat loop | Alternates users, displays messages, checks commands |
| Commands | exit, clear, save functionality |
| File saving | Records chat history |

This explanation is **presentation-friendly** and ready to include as a reference or handout.