

User Guide for CHAT APP

Introduction

CHAT APP is a user-friendly, Python-based chat application that allows users to communicate through a network via a simple graphical user interface (GUI). The application consists of a server module (`servergui.py`) and a client module (`clientgui.py`). The server module needs to be started first to listen for incoming connections from clients. Multiple clients can connect to the server simultaneously from different hosts to send and receive messages in real-time.

Prerequisites

- **Operating System:** Windows 10 or higher.
- **Python Installation:** Python 3.7 or higher must be installed.
- **Libraries:** PyQt5 for the GUI components.

Installation and Setup

Check Python Installation

1. Open Command Prompt: Press Win + R, type `cmd`, and press **Enter**.
2. Verify Python Installation by typing:

```
python --version
```

If Python is not installed, follow the official download link to install it. Ensure you tick "Add Python to PATH" during the setup.

Install Necessary Libraries

1. Install PyQt5 by running:

```
pip install PyQt5
```

PyQt5 is used for creating the graphical interface of the chat application. There is no need to install the `socket` module as it comes built-in with Python.

Running the Application

Running the Server

1. Navigate to the project folder:

```
cd path\to\your\project\folder
```

2. Launch the server by running:

```
python servergui.py
```

Click **”Start Server”** in the server GUI to begin listening for connections.

Running the Client on the First Host

1. Open another Command Prompt window and navigate to the folder containing `clientgui.py`.
2. Start the client by running:

```
python clientgui.py
```

In the client GUI, click **”Connect”** to connect to the server. Use the input box and **”Send”** button to communicate. **”Disconnect”** ends the session.

Running the Client on the Second Host

Follow the same steps as the first client to connect additional clients to the server from different hosts.

Viewing Communications

- Server and client interactions are logged in `server.log`, which can be found in the server’s directory.