**Project Proposal**

Computer Networks – CSE351

**Submitted to**

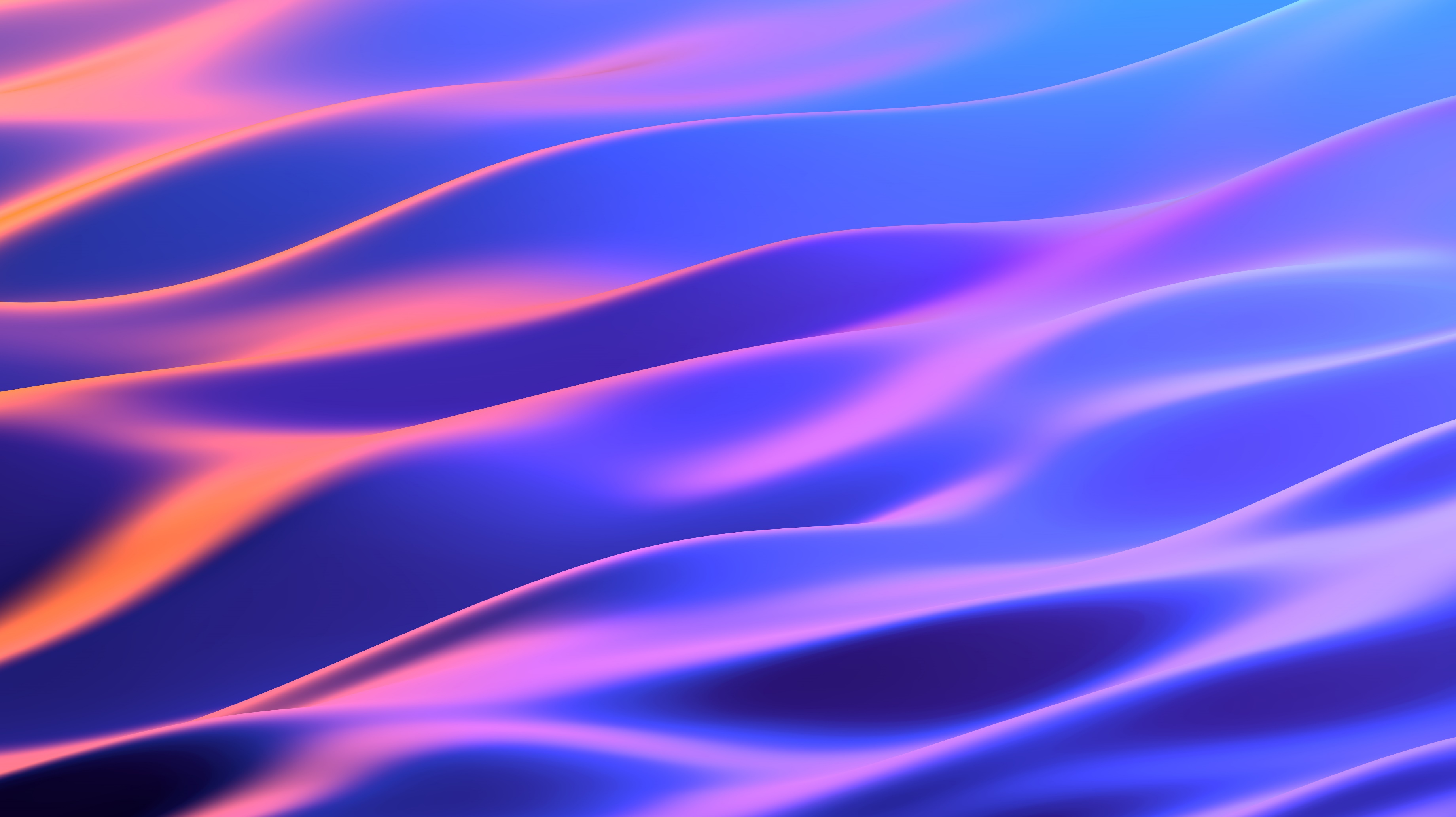
Prof. Ayman Bahaa El-dine

Eng. Noha Wahdan

**Group: 20**

|  |  |
| --- | --- |
| Ahmed Wael Ibrahim Mohamed | 20P3343 |
| Donia Sameh Farouk Mohamed | 20P3424 |
| Joliana Emad Kamal Naguib | 20P3292 |
| Mohamed Amr El-Mallah | 20P3485 |





Contents

[Executive Summary 4](#_Toc154072216)

[Focus 4](#_Toc154072217)

[Project Overview 5](#_Toc154072218)

[Hardware and Software Requirements 5](#_Toc154072219)

[Operating System 5](#_Toc154072220)

[Programming Language 5](#_Toc154072221)

[Operating System 5](#_Toc154072222)

[Additional Software Products 5](#_Toc154072223)

[Project Scope 6](#_Toc154072224)

[Project Objectives 6](#_Toc154072225)

[Phases of Implementation 6](#_Toc154072226)

[Constraints and Assumptions 6](#_Toc154072227)

[Key Functionalities 6](#_Toc154072228)

Project Proposal

# Executive Summary

The modern era, shaped by globalization and technological advancements, has significantly altered human life, notably through the emergence of social networking. As an integral facet of contemporary existence, this evolution necessitates innovative technologies and software to facilitate seamless human interaction. Our project responds to this need by developing a sophisticated peer-to-peer messaging application, offering users worldwide the ability to communicate effortlessly via a user-friendly command-line interface.

Leveraging the capabilities of Python and Socket APIs alongside the Computer Networks protocol stack, our application seeks to enable global communication. Users will access private and group chat functionalities within chat rooms, empowering them to exchange messages seamlessly. Furthermore, the application offers the added feature of sending formatted text, encompassing options such as Bold, Italic, and more.

This project's primary goal is to create an intuitive, robust platform that harnesses contemporary technologies to foster enhanced global communication while prioritizing user convenience and versatility in communication styles.

# Focus

After a thorough but efficient preparation period, our team will create the Peer-to-Peer Messaging Application.

We will build this with a focus on **Ability to authenticate users**, **Ability to send/receive messages** both in private and in chat rooms and **Ability to send formatted messages** through a user-friendly command-line interface.

The resulting software will achieve:

* Sophisticated yet user-friendly user interface.
* Secure and reliable connection between clients.
* Secure and sustainable user authentication protocols.
* Efficient User communication protocols.
* Error handling and resilience.

# Project Overview

We are committed to developing and delivering a product that fulfils the provided requirement specifications in the best possible methods. As such, we have put together a plan to develop the Peer-to-Peer Messaging Application. The software solution will fulfil the requirements mentioned in the **Key Functionalities** section.

## Hardware and Software Requirements

This section will discuss the operational requirements for developing and operating the finished software solution.

|  |  |
| --- | --- |
| Operating System  * OS that supports running Python Scripts * Windows – MacOs - Linux | Programming Language  * Python Programming 3.9 or later * Soket Programming APIs |
| Operating System  * Desktop Computers * Personal Computers (Laptops) * Computing devices supporting Python Scripts | Additional Software Products  * Any further products will be discussed later when needed. |

# Project Scope

## Project Objectives

The project aims to develop a robust Peer-to-Peer Multi-User Chatting Application using Python programming language and socket programming APIs. The project focuses on text-based communication while utilizing the peer-to-peer network topology, TCP and UDP protocols. The project progresses through three technical phases that will be mentioned in a later section.

## Phases of Implementation

|  |
| --- |
| **Phase (1) – Basic Client-Server Setup**  Implement client-side and server-side software capable of communicating together using Python and Socket API. |
| **Phase (2) – Peer-to-Peer Architecture and Chat Rooms**  Integrate peer-to-peer architecture and implement the ability to create chat rooms. |
| **Phase (3) – One-to-One Chatting and Protocol Optimization**  Implement one-to-one chatting feature between the user and another selected user. In addition to protocol optimizations and performance enhancements. |

## Constraints and Assumptions

* Both server and client software run on windows machines with python packages and interpreter installed.
* Both server and client software will run on a pre-specified port number.
* Users will interact through the system using Command-line interface (CLI).

## Key Functionalities

1. **User Authentication**

Users can authenticate the user’s credentials (username and password).

1. **Chat Rooms**

Users can create chat rooms where that other users can join or join a pre-created chat room.

1. **Group Messaging in Chat Rooms**

Users can send or read messages sent in a chat room. All messages are visible to all the users.

1. **One-to-One Chatting**

Users can participate in a private chat with another user. The messages are visible to the participants only and no other user can see the messages.

1. **Message Formatting**

Users can send formatted messages (i.e., bold, italic).

1. **User Interface**

Users can interact with the software using a user-friendly command-line interface (CLI).