Mini Project -(2021 - 2022)

SYNOPSIS

ON

I – Chat App

(Web Application)

Submitted By :

1- Name – Shubham Bhardwaj
Branch/Sec - B.Tech(CSE) - C

University Roll No – 191500797

2- Name – Haresh Singh

Branch/Sec - B.Tech(CSE) - H

University Roll No – 191500311

3- Name – Tanmay Gaur

Branch/Sec - B.Tech(CSE) - G

University Roll No – 191500844

4- Name – Umang Chauhan

Branch/Sec - B.Tech(CSE) - C

University Roll No – 191500869

5- Name – Shubhransh Singh

Branch/Sec - B.Tech(CSE) - J University Roll No - 191500807

Submitted To:

Mandeep Sir

PROJECT NAME: "I- CHAT"

Index

INTRODUCTION

We are creating “A Web-browser based application” named “**I-chat**”.

This messaging app is used for chatting purposes. It make communication easy with respect to such as instant messaging.

It is a web-Browser based application that runs within the web browser via Internet without accessing the operating system of any individual computer. The advantage of browser-based applications is that they can run in Windows, Mac, or Linux machine.

USE OF THE PROJECT

This app is used for chatting purposes. They make communication easy with respect to such as instant messaging and voice calls. This app can be installed on a smartphone regardless of the smartphone operating system.

In today’s era, we all are aware of the need to communicate with each other. However we the people are so busy in our own lives, we forget and sometimes ignore that *the world beyond us exists.*

One solution to this is **Virtual Communication**.

Communicating virtually with people is when individuals interact without being face-to-face but through words and texts that are sent at a higher speed than letters. We can share information, comment, and ask questions, develop social etiquette.

Communicating virtually make us elevate from being digitally literate to digitally fluent, collaborate using appropriate virtual tools and system and they are cost-effective too.

So putting our efforts into this application, we are going to create I-chat.

FUNCTIONAL SPECIFICATION

In that, we will be using Socket.io (to establish a connection between server and client i.e. two-way connection) and many more technologies that have been discussed below.

Socket.io:-

Socket.io is the two-way connection between the server and the client.

As we know Http, Https protocol is a one-way connection between server and client. So in this, we are going to use ws(web socket), wss(web socket secure) protocol to establish the connection between the server and client too.

As we know a webpage consists of a front end as well as a backend.

Front-end:-

It is the practice of converting data to a graphical interface, through the use of HTML, CSS, and JavaScript, so the users can view and interact with that data and they are explained below.

CSS:

CSS is used for styling the Webpages. CSS is of three types i.e. internal CSS, external CSS, inline CSS. It describes how Html elements are to be displayed on the screen. It involves many designing for the bare Mark-up to create it attractive.

HTML :-

The first version of Html was written by Tim Berners-Lee in 1993. Since then, there have been many different versions of Html. We are using Html 5 in our project.

Html is a programming language used to describe the structure of information on a webpage.

Front end developer uses Html as to make the structure of their ideas or we can say a raw website. In that, design our idea by using CSS, JS.

JavaScript:-

JavaScript was created by Brendan Eich in 1995. It is the programming, Scripting language that allows implementing complex features on web pages. JS helps the web developer to make a dynamic and interactive webpage by implementing a custom client-side script.

Another part is a back-end.

Back-end:-

Back-end development refers to server-side development. It focuses on databases, scripting, and architecture of websites.

Code written by back-end developers helps to communicate the database information to the browser.

So, in this project, we will use node.js as a backend, explained here

NodeJs:

NodeJs was developed by many developers but the original author was Ryan Dahl in 2009.

NodeJs is an open-source, cross-platform that executes JavaScript code outside a web browser.

It also works as a backend. And we are going to use it as a backend.

-

EXISTING SYSTEM

Yes, this Real-Time Chat Application has already been created and implemented in today's world and almost every one of us is using it. One such example is What's App, We Chat and, many more.

According to **our idea of creating a Real-Time Chatting Application named "I-chat" to establish productively and trusting virtual working relationships.**

We have used the basic mark-up language (for creating a webpage) that is HTML (HyperText Markup Language) and the basic styling requiring with CSS (Cascading Styling Sheets) to give it a gesture of a real-time application.

Some of the features are:-

1. Scrolling bar at the right:-
   Whenever there are a lot of messages on the screen, to see them all, a scrolling bar on the right-hand side will appear on its own.
2. Lots of designing by using CSS:-
   We will be using a nice lot of RGB colors, and many more CSS to enhance our webpage.
3. Message notification sound:-

   With every message that arrives, there is a notification alert with a minute ringtone noise.

This is all we will be doing in this project, and apart from using the basic CSS, we will put all our efforts to add more features to this application.

Future/ Extension Scope

In the extension of the “i-chat”, we would be adding some modules into it.

Some of the modules are enlisted below:

- **Emoji:** As we all know in today’s world emotions play a vital role, but virtually we cannot show or present our felling. So to nullify these cons, emoji is there so that people can present their expression by texting some emoji. So we will be providing a module for emoji/emotions in the future.
- **Grammarly:** In this era, “**Formal** C**hatting**” is also an important aspect, which requires proper English and vocabulary. So the module for the proper vocab and grammar would be included in the extension of the project.
- **Different languages:** By default, the chatting language would be English. But if the user wants to text in other landguage like “French”, “Hindi”,” Spanish” etc, they can do with the help of some imports.
- **A box of users:** We would be adding a box in which the name of users chatting in that server at a time with simple HTML and an amount of CSS to enhance it.
  And many more**.**

.

Software Requirement Specification(SRS)

Hardware requirements

In hardware requirement, we require all those components which will provide us the platform for the development of the project. The minimum hardware required for the development of this project is as follows-

RAM - a minimum of 128 MB

Hard disk- minimum 4GB

Processor - core i3

These all are the minimum hardware required for our project. We want to make our project to be used in any type of computer therefore, we have taken a minimum configuration to a large extent. 128 MB ram is used so that we can execute our project in the least possible ram. 500 GB hard disk is used because the project takes less space to be executed or stored. Therefore minimum hard disk is used.

Software requirements

The software can be defined as a program that runs on our computer. It acts as petrol in the vehicle. It provides the relationship between the human and computer. Various software is needed in this project for its development.

Which are as follows-

Operating system- windows7

Others- visual studio

Software architecture

Socket overview

A socket is an object that represents a low-level access point to the IP stack. This socket can be opened or closed or one set of the number of an intermediate state. A socket can send and receive data down disconnection. Data is generally sent in blocks of few kilobytes at a time for efficiency, each of these blocks is called a packet.

All packets that travel on the internet must use the Internet protocol. This means that the source IP address, the destination address must be included in the packet, the most packet also contains a port number. A port is simply a number between 1 and 65,535 that is used to differentiate higher protocols. Ports are important when It comes to programming your network applications because no two applications can use the same port.

Packets that contain port numbers come in two flavors: UDP and TCP/IP. UDP has lower latency than TCP/IP, especially on startup. Where data integrity is not of the utmost concern. UDP can prove easier to use than TCP, but it should never be used where data integrity is more important than performance; however, data send by UDP can sometimes arrive in the wrong order and effectively useless to the user. TCP/IP is more complex than UDP and has longer latencies.

UDP Ports

The User Datagram Protocol(UDP) is an unreliable, connectionless oriented protocol that uses an IP address for the destination host and a port number to identify the destination application.

Software Process Model

The Software Process Model used is in the spiral model. The choice for these models in the light of enhancement that we foresee for the future. The enhancements would be in the area of networks being introduced in the software.

Process Modules

The functionalities and responsibilities of the system were partitioned and then assigned to subsystems or components as described below.

Graphical User Interface

The user interface that the software provides to the user is interactive. It provides two different forms, one for a list of systems and the other for the actual text chatting.

Resolving Names

This module handles the code that is necessary to view the different aspects of the network connections in a system.

Connections

This module deals with the establishment of a connection between the host system and the system selected from the list. The connection is made by the help of sockets that uses ports to send and receive packets from one system to another. The message sent or received is coded and is encoded or decode respectively.

Conclusion

Our overview has given a broad picture of the design of the software in terms of the different modules used. It also gives us an idea about the degree to which each module performs related tasks. We also get an idea about the independence among the modules.