**CHAPTER 1**

**INTRODUCTION**

* 1. **Problem And Motivation**

People who move to unknown environment face many issues, mostly being unfamiliar to the local language and culture, I myself have faced it and felt that an application should be developed to solve this issue of at least being able to converse with them.

* 1. **Purpose Of the Project**

Since I have personally experienced this problem, I believe a program should be made to at least be able to communicate with them. People who migrate to a foreign location confront several challenges, chief among them being unfamiliar with the local language and culture. This was the sole purpose of me choosing this topic as the motivation.

* 1. **Objectives Of the Project**

In summation, I am constructing a web application that will enable quick and easy communication amongst persons with various linguistic backgrounds. This will lessen the likelihood of mistakes and misunderstandings leading to misunderstandings among the people.

* 1. **Scope Of the Project**

The scope of the project is to create a clean and easy to use user friendly web application that is mainly being focused towards users ranging from students away from their homes to office workers in a foreign location for a meeting to a tourist trying to a conversation with a local man.

**CHAPTER** **2**

**LITERATURE SURVEY**

**2.1 Introduction**

The chat application is software by which messages can be transmitted to a remote location

or anywhere and can communicate through the internet at the same time.

The chat application is software by which messages can be transmitted to a remote location

or anywhere and can communicate through the internet at the same time.

The chat application is software by which messages can be transmitted to a remote location

or anywhere and can communicate through the internet at the same time.

The chat application is software by which messages can be transmitted to a remote location

or anywhere and can communicate through the internet at the same time.

The chat application is software by which messages can be transmitted to a remote location

or anywhere and can communicate through the internet at the same time.

The chat application is software by which messages can be transmitted to a remote location

or anywhere and can communicate through the internet at the same time.

The chat application is software by which messages can be transmitted to a remote location

or anywhere and can communicate through the internet at the same time.

The chat application is software by which messages can be transmitted to a remote location

or anywhere and can communicate through the internet at the same time.

The chat application is software by which messages can be transmitted to a remote location

or anywhere and can communicate through the internet at the same time.

The chat application is software by which messages can be transmitted to a remote location

or anywhere and can communicate through the internet at the same time.

The chat application is software by which messages can be transmitted to a remote location

or anywhere and can communicate through the internet at the same time.

The chat application is software by which messages can be transmitted to a remote location

or anywhere and can communicate through the internet at the same time.

The chat application is software by which messages can be transmitted to a remote location or anywhere and can communicate through the internet at the same time.

Chat messages are simply short with a specific end goal to empower different members to react rapidly. Subsequently, an inclination like a chat conversation is made, which recognizes chatting from other content based online correspondence structures, for example, Internet forums and email.

The chat translator app is programmed software which is used to convert or translate languages from one language to others. The use of translator application is to translate the messages or chats. Users copy the text from the chat application or type the text into the translator app and translate into user’s desired language.

There are thousands of languages using people in the world but all people are not good at more than few languages. Sometimes people need to communicate or interact with different languages that are not understandable all the speech or vocabulary.

For the purpose of being understandable an un-known language users use the translator application software.

But the current chat translation process is manual. For the chat translation users still are using copy and paste process. Auto translation of chat is the union of chat system and translation system.

In this system chats can translate automatically without the use of copy and paste of the text because traditionally the chat systems and the translation systems are different software. If user needs to chat only with another user, he can use only chat application.

**CHAPTER 3**

**SOFTWARE REQUIREMENT SPECIFICATION**

**3.1 Introduction**

A software requirement Specification (SRS) comprehensive description of the intended purpose and environment for development. The SRS fully describes what the software will do and how it expected to perform[2]. Software requirements specification is a perfect detailed description of the behavior of the system to be developed. Current program has been justified in the following parameters of SRS that are: -

**Software Requirements, Hardware Requirements, Non-Functional Requirements, Functional Requirements, Feasibility Study**

The following are all the parameters used in the program.

**3.2 System requirement**

**3.2.1 Software requirements (Used for Developing)**

Operating System : Windows 10 Pro

Software Tools : Adobe Dreamweaver

Web Technologies : HTML, JavaScript, CSS, Python, Perl

Web Server : GitHub

Database : Firebase

**3.2.1 Software requirements (Used for Running)**

Operating System : Windows 7 or above

Software : Any Browser(Latest one Preferred)

Other : Stable Internet Support

|  |  |
| --- | --- |
| Processor | Intel Core i3-8220 or above |
| RAM | 8 GB or above |
| Storage | 126 GB or above |
| Other | Net access |

**3.2.2 Hardware requirements (Used for Developing)**

**3.3 Functional and non-functional requirements**

**3.3.1 Functional requirements**

The functional requirements for a system describe the functionality or the services that the system is expected to provide.

* Provides a user-friendly UI/UX
* Translates the Text efficiently
* On closing all data is deleted for better privacy

**3.3.2 Non-functional requirements**

These are those requirements which are not directly concerned with the specific functions delivered by the system.

* **Reliability:**

Uses GitHub and Google Firebase servers as their servers are powerful and don’t go down.

* **Usability**:

Easy to use interface, so anyone with zero experience can use it without trouble

* **Maintainability:**

The program can be easily corrected as it is made in HTML and JScript.

* **Portability:**

The ease of transporting a program from one hardware configuration to another is there.

**3.4 Feasibility Study**

**3.4.1 Technical Feasibility**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* **Does the necessary technology exist to do what is suggested?**

Yes, there are Multiple softwares that can be used to translate the given text into the desired language.

* **Do the proposed equipment have the technical capacity to hold the data required to use the new system?**

Yes, Firebase servers are virtually more than enough to store the current data.

* **Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?**

Currently it is being capped to limited number of users for demonstration purpose but it has to potential to serve large number of users simultaneously.

* **Can the system be upgraded if developed?**

Yes, depending on the level of skills possessed by the developer, it can be made into a full-scale application.

**3.4.2 Economic Feasibility**

A system can be developed technically and that will be used if installed must still be a good investment for the organization.

In my project, there is no additional cost enquired as it is using GitHub’s free webserver. The system is economically feasible. It does not require any additional hardware or software.

**3.4.3 Operational Feasibility**

Proposed system does not require any additional admins to monitor and manage the system as it is a client-to-client based interface.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

**CHAPTER 4**

**SYSTEM ANALYSIS**

**4.1 System Description**

The Polyglot is a simple software that helps in communicating with people who are of different linguistic background, this web-application consists only the user module as it is a client-to-client chat interface

Although the program is a simple chat interface with one module, it is split into three parts: -

* Home page
* Join page
* Chat page

**4.2 Existing System**

## Currently there are a few android-based applications that help in translation during conversations but none of the are refined, and there are no web-based applications that can be used in both PC and Smartphones.

**4.3 Proposed System**

Polyglot is been proposed to overcome all the shortcomings of the existing products in the market today. This software is supported to eliminate and, in some cases, reduce the hardships faced by people.

This product will fix all the issues form a bad and inconsiderate UI design to its bug ridden back-end code and functionality.

This project can be easily implemented and enhanced by others for their own specific requirements.

**CHAPTER 5**

**SYSTEM DESIGN**

**5.1 Architecture**

The database structure in my project is designed in the Google’s Firebase database which is a free and reliable database server for projects to full scale applications that require a fast and responsive database to manage operations.

**5.2 Database Design**

The database structure in my project is designed into two tables. The detailed structure of the database is as follows;

|  |  |  |
| --- | --- | --- |
| Room ID | Number | Primary Key |
| Creator | Text | Not null |
| Joiner | Text | Not null |
| clang | Text | Not null |
| Jlang | Text | Not null |

**Table 5.2.1 Login Table**

|  |  |  |
| --- | --- | --- |
| Room ID | Number | Foreign Key |
| User | Text | Not null |
| Message | Text | Not null |

**Table 5.2.2 Data Table**

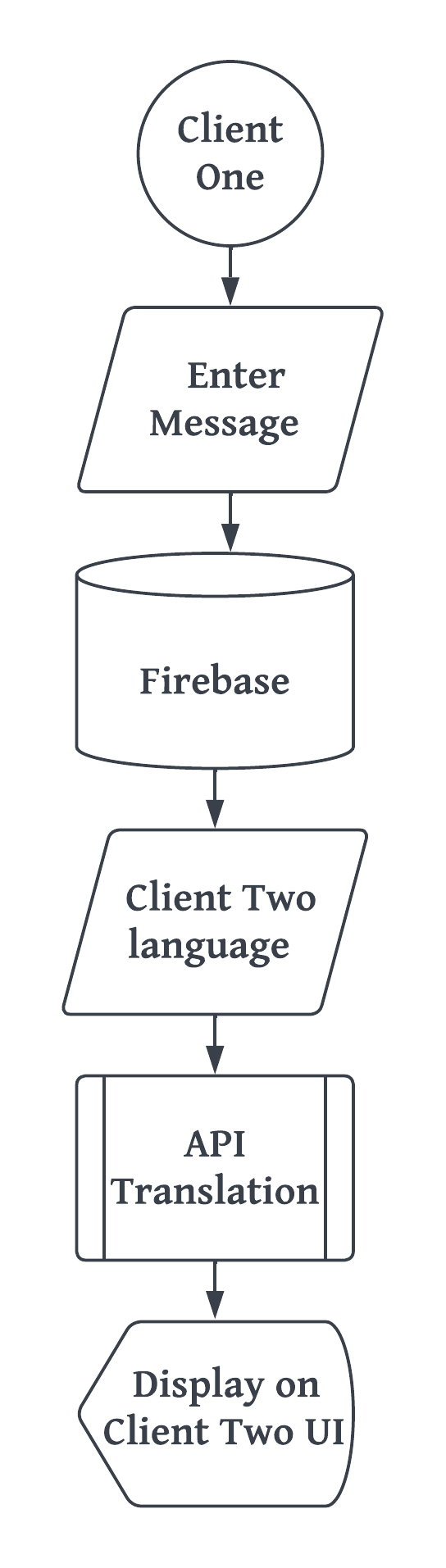
**CHAPTER 6**

**DETAILED DESIGN**

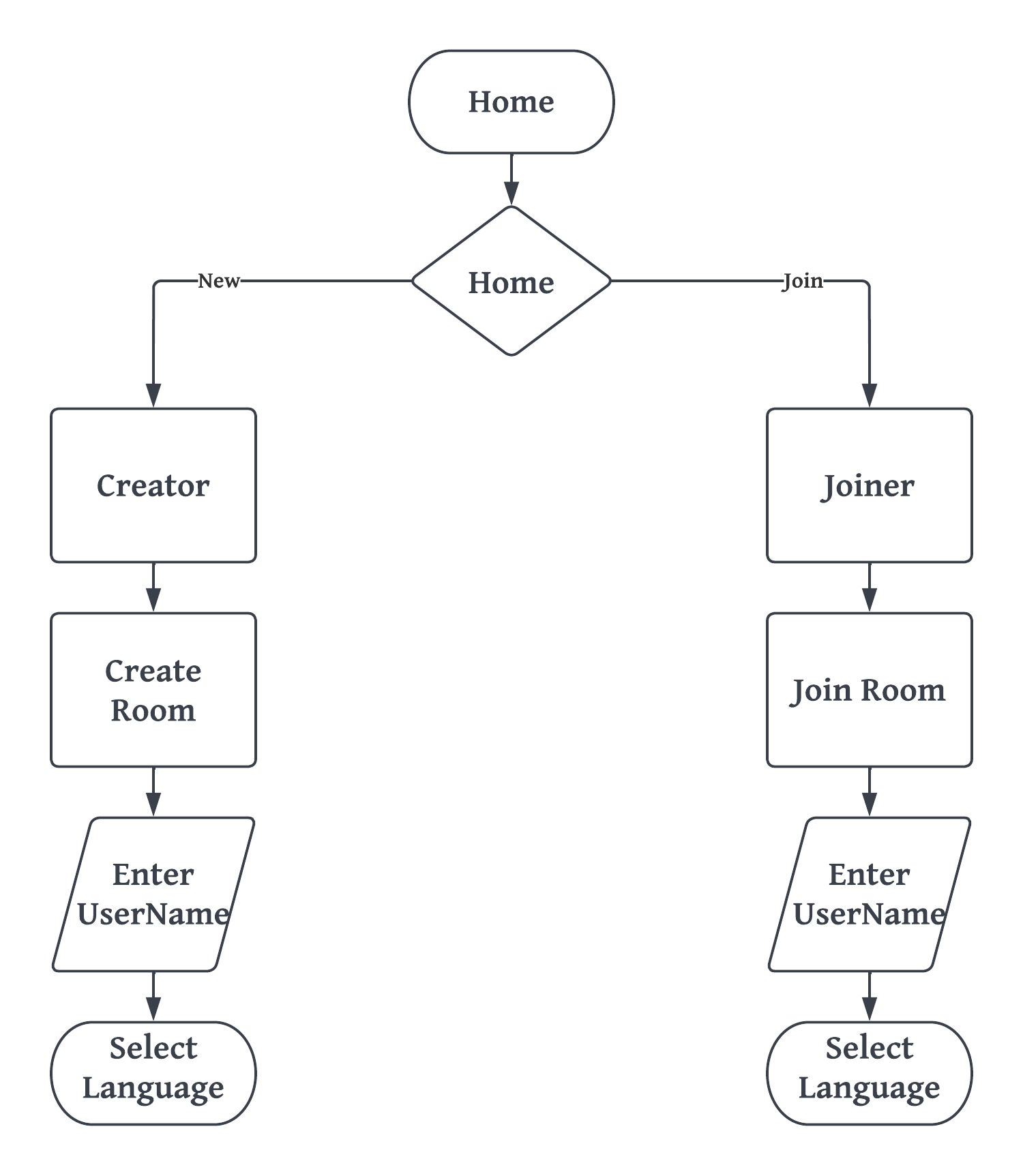
**6.1 Introduction**

In my project detailed design starts after the system design phase is completed. The goal of this phase is to develop the internal logic of each of the modules identified during system design.

**6.2 DFD**



**Fig 6.2.1 DFD Of Data**



**Fig 6.2.2 DFD of Login**

**CHAPTER 7**

**IMPLEMENTATION**

**7.1 Coding**

**Index.html (Home Page)**

<!DOCTYPE html>

<html>

<head>

<title>Polyglot</title>

<link rel="icon" type="image/png" href="images/cs logo.png">

<linkrel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/reset.min.css">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font awesome.min.css">

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">

<div class="container">

<p> <a class="navbar-brand" href="index.html">POLYGLOT</a></p>

<p><a class="navbar-brand" href="index.html">&nbsp;</a> </p>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

</div>

</nav>

<br>

<br>

<br>

<div id="app">

<h2 class="joincht">Chating with multilingual friends made easy</h2>

<br><br><br><br>

<div><a href="Join.html"><b><button class="button button4">Chat Now</button></b></a></div>

</div>

<br>

<br>

<br>

<div class="foot">

<footer class="py-5 bg-dark" style="position:fixed;bottom:0%;left:0%;width:100%">

<div class="container">

<p class="m-0 text-center text-white">POLYGLOT cc.</p>

</div>

</footer>

</div>

</body>

<script src="/socket.io/socket.io.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/vue/2.5.17/vue.js"></script>

<script src="https://unpkg.com/axios/dist/axios.min.js"></script>

</html>

**Join.html (Login Page)**

<!DOCTYPE html>

<html>

<head>

<title>Polyglot</title>

<link rel="icon" type="image/png" href="images/cs logo.png">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/reset.min.css">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font awesome.min.css">

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">

<div class="container">

<p> <a class="navbar-brand" href="index.html">POLYGLOT</a></p>

<p><a class="navbar-brand" href="index.html">&nbsp;</a> </p>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<!--<div class="collapse navbar-collapse" id="navbarResponsive">

<ul class="navbar-nav ml-auto">

<li class="nav-item active">

<a class="nav-link" href="index.html">Main</a>

<span class="sr-only">(current)</span>

</li>

<li class="nav-item">

<a class="nav-link" href="about.html">About us</a>

</li>

<li class="nav-item">

<a class="nav-link" href="code.html">Code</a>

</li>

</ul>

</div>-->

</div>

</nav>

<div id="app" style="position: center">

<div>

<div v-if="!ready">

<h2 class="joincht">Join A Chatroom</h2>

<div class="containerInputs">

<form @submit.prevent="addName" id="joinchat">

<div class="form-group formJoin" style="background:#D3D3D3;border-radius:7px">

<input type="text" class="form-control" v-model="name" style="border:0px;line-height:2" placeholder="Enter name" id="name"><br>

<input type="text" class="form-control " v-model="rom" style="border:0px;line-height:2" placeholder="Enter room" id="room"><br>

<select id="lang" class="form-control" style="border:0px;line-height:2" >

<option selected disabled>-Select Language-</option>

<option value="en">English</option>

<option value="hi">Hindi</option>

<option value="ml">Malayalam</option>

<option value="ta">Tamil</option>

<option value="te">Telugu</option>

<option value="kn">Kannada</option>

</select>

<br><br>

<div style="margin-left:40%;">

<input type="submit" value="Join" style="border:0px; padding:5%;background-color:#444753; color:#fff;border-radius:7px;width:35%">

</div>

</div>

</form>

</div>

</div>

</div>

</div>

<div class="foot" >

<footer class="py-5 bg-dark" style="position:fixed;bottom:0%;left:0%;width:100%">

<div class="container">

<p class="m-0 text-center text-white">POLYGLOT cc.</p>

</div>

</footer>

</div>

</body>

<script type="module">

import { initializeApp } from "https://www.gstatic.com/firebasejs/9.9.4/firebase-app.js";

import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.9.4/firebase-analytics.js";

const firebaseConfig = {

apiKey: "AIzaSyB5B-7WM0EX2YM\_kQDQXNaixejpYEevlMY",

authDomain: "mapgo-f01f5.firebaseapp.com",

databaseURL: "https://mapgo-f01f5-default-rtdb.firebaseio.com",

projectId: "mapgo-f01f5",

storageBucket: "mapgo-f01f5.appspot.com",

messagingSenderId: "496899577881",

appId: "1:496899577881:web:82694b7eb8487149357288",

measurementId: "G-8EZLBGQ1T8"

};

import {getDatabase,ref,set,child,update,remove,get,onValue} from "https://cdnjs.cloudflare.com/ajax/libs/firebase/9.9.4/firebase-database.min.js";

const app = initializeApp(firebaseConfig);

const analytics = getAnalytics(app);

const db = getDatabase();

document.getElementById('joinchat').addEventListener('submit',adddata);

function getval(id)

{

return document.getElementById(id).value;

}

function adddata()

{

var name=getval('name');

var room=getval('room');

var lang=getval('lang');

db.on('login', function(snapshot) {

var check = snapshot.val();

if (check.room != room)

{

set(ref(db,"login/"),{

creator:name,

room:room,

clang:lang

})

}

else if(check.joiner=="" && check.room==room)

{

if(check.creator != name)

{

alert("Name Exists, Select a New Name");

}

else

{

set(ref(db,"login/"),{

joiner:name,

jlang:lang

})

}

}

else

{

alert("Room Exists,Create a New Room");

}

}

</script>

<script src="/socket.io/socket.io.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/vue/2.5.17/vue.js"></script>

<script src="https://unpkg.com/axios/dist/axios.min.js"></script>

</html>

**Chat.html (Chat Page)**

<!DOCTYPE html>

<html>

<head>

<title>Polyglot</title>

<link rel="icon" type="image/png" href="images/cs logo.png">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/reset.min.css">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font awesome.min.css">

<link rel="stylesheet" href="https://pyscript.net/alpha/pyscript.css" />

<script defer src="https://pyscript.net/alpha/pyscript.js"></script>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">

<div class="container">

<p> <a class="navbar-brand" href="index.html">POLYGLOT</a></p>

<p><a class="navbar-brand" href="index.html">&nbsp;</a> </p>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarResponsive">

<ul class="navbar-nav ml-auto">

<li class="nav-item active">

<a class="nav-link" href="index.html">Main</a>

<span class="sr-only">(current)</span>

</li>

<li class="nav-item">

<a class="nav-link" href="about.html">About us</a>

</li>

<li class="nav-item">

<a class="nav-link" href="code.html">Code</a>

</li>

</ul>

</div>

</nav>

<div id="app">

<div>

<div class="clearfix containerChat" v-else>

<div class="people-list" id="people-list" >

<div class="onlineUsrs">

Online Users

</div>

<ul class="list">

<li class="clearfix" v-for="friend in friends">

<div class="about">

<div class="name">{{friend.name}}</div>

<div class="status">

<i class="fa fa-circle online"></i> {{friend.type}}

</div>

</div>

</li>

</ul>

</div>

<div class="chat">

<div class="chat-header clearfix" style="background: #AFAFAF">

<div class="chat-about" >

{{roomname}}

</div>

</div> <!-- end chat-header -->

<div class="chat-history scroll scroll1">

<ul>

<li class="clearfix" v-for="message in messages">

<div v-if="message.type == 0">

<div class="message-data align-right">

<span class="message-data-name" >{{message.by}}</span> <i class="fa fa-circle me"></i>

</div>

<div class="message other-message float-right">

{{message.message}}

</div>

</div>

<div v-else>

<div class="message-data" >

<span class="message-data-name"><i class="fa fa-circle online"></i> {{message.by}}</span>

</div>

<div class="message my-message">

{{message.message}}

</div>

</div>

</li>

</ul>

</div> <!-- end chat-history -->

<form @submit.prevent="send">

<div class="chat-message clearfix form-group">

<small v-if="typing" class="text-gray float-left">{{typing}} is typing...</small>

<input type="text" v-on:keyup.13="submit" name="message-to-send" id="message-to-send" v-model="newmessage" placeholder ="Type your message" rows="3"></input>

</div> <!-- end chat-message -->

</form>

</div> <!-- end chat -->

</div>

</div>

</div>

<br>

<br>

<br>

<div class="foot">

<footer class="py-5 bg-dark" style="position:fixed;bottom:0%;left:0%;width:100%">

<div class="container">

<p class="m-0 text-center text-white"><b>POLYGLOT cc.</b></p>

</div>

</footer>

</div>

</body>

<script type="module">

import { initializeApp } from "https://www.gstatic.com/firebasejs/9.9.4/firebase-app.js";

import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.9.4/firebase-analytics.js";

const firebaseConfig = {

apiKey: "AIzaSyB5B-7WM0EX2YM\_kQDQXNaixejpYEevlMY",

authDomain: "mapgo-f01f5.firebaseapp.com",

databaseURL: "https://mapgo-f01f5-default-rtdb.firebaseio.com",

projectId: "mapgo-f01f5",

storageBucket: "mapgo-f01f5.appspot.com",

messagingSenderId: "496899577881",

appId: "1:496899577881:web:82694b7eb8487149357288",

measurementId: "G-8EZLBGQ1T8"

};

import { getDatabase,ref,set,child,update,remove ,get,onValue} from "https://cdnjs.cloudflare.com/ajax/libs/firebase/9.9.4/firebase-database.min.js";

const app = initializeApp(firebaseConfig);

const analytics = getAnalytics(app);

const db = getDatabase();

const dbref = ref(db,"data/");

onValue(dbref,(snapshot)=>{

var message = snapshot.val().message;

})

</script>

<script src="/socket.io/socket.io.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/vue/2.5.17/vue.js"></script>

<script src="https://unpkg.com/axios/dist/axios.min.js"></script>

</html>

**Style.css(General Styling)**

<style type="text/css">

@import url(https://fonts.googleapis.com/css?family=Lato:400,700);

\*, \*:before, \*:after {

box-sizing: border-box;

}

body {

font: 14px/20px "Lato", Arial, sans-serif;

padding: 40px 0;

color: white;

}

.containerChat {

margin: 0 auto;

width: 780px;

background: #444753;

border-radius: 5px;

}

.containerInputs{

margin-left: 40%;

margin-top: 5%;

width: 20%;

background: #F2F5F8;

border-radius: 10px;

}

.scroll {

width: 515px;

height: 40px;

overflow: auto;

float: left;

}

.scroll1::-webkit-scrollbar {

width: 5px;

}

.scroll1::-webkit-scrollbar-track {

background: #ddd;

}

.scroll1::-webkit-scrollbar-thumb {

background: #666;

}

.people-list {

width: 260px;

float: left;

}

.people-list .search {

padding: 20px;

}

.people-list input {

border-radius: 3px;

border: none;

padding: 14px;

color: white;

background: #6A6C75;

width: 90%;

font-size: 14px;

}

.people-list .fa-search {

position: relative;

left: -25px;

}

.people-list ul {

padding: 20px;

height: 600px;

}

.people-list ul li {

padding-bottom: 20px;

}

.people-list img {

float: left;

}

.people-list .about {

float: left;

margin-top: 8px;

}

.people-list .about {

padding-left: 8px;

}

.people-list .status {

color: #92959E;

}

.chat {

width: 520px;

float: left;

background: #F2F5F8;

border-top-right-radius: 5px;

border-bottom-right-radius: 5px;

color: #434651;

}

.chat .chat-header {

padding: 20px;

border-bottom: 2px solid white;

}

.chat .chat-header img {

float: left;

}

.chat .chat-header .chat-about {

float: left;

padding-left: 10px;

margin-top: 6px;

}

.chat .chat-header .chat-with {

font-weight: bold;

font-size: 16px;

}

.chat .chat-header .chat-num-messages {

color: #92959E;

}

.chat .chat-header .fa-star {

float: right;

color: #D8DADF;

font-size: 20px;

margin-top: 12px;

}

.chat .chat-history {

padding: 30px 30px 20px;

border-bottom: 2px solid white;

overflow-y: scroll;

height: 575px;

}

.chat .chat-history .message-data {

margin-bottom: 15px;

}

.chat .chat-history .message-data-time {

color: #a8aab1;

padding-left: 6px;

}

.chat .chat-history .message {

color: white;

padding: 18px 20px;

line-height: 26px;

font-size: 16px;

border-radius: 7px;

margin-bottom: 30px;

width: 90%;

position: relative;

}

.chat .chat-history .message:after {

bottom: 100%;

left: 7%;

border: solid transparent;

content: " ";

height: 0;

width: 0;

position: absolute;

pointer-events: none;

border-bottom-color: #4AB19D;

border-width: 10px;

margin-left: -10px;

}

.chat .chat-history .my-message {

background: #4AB19D;

}

.chat .chat-history .other-message {

background: #94C2ED;

}

.chat .chat-history .other-message:after {

border-bottom-color: #94C2ED;

left: 93%;

}

.chat .chat-message {

padding: 30px;

}

.chat .chat-message input {

width: 100%;

border: none;

padding: 10px 20px;

font: 14px/22px "Lato", Arial, sans-serif;

margin-top: 20px;

border-radius: 5px;

resize: none;

}

.chat .chat-message .fa-file-o, .chat .chat-message .fa-file-image-o {

font-size: 16px;

color: gray;

cursor: pointer;

}

.chat .chat-message button {

float: right;

color: #94C2ED;

font-size: 16px;

text-transform: uppercase;

border: none;

cursor: pointer;

font-weight: bold;

background: #F2F5F8;

}

.chat .chat-message button:hover {

color: #75b1e8;

}

.online, .offline, .me {

margin-right: 3px;

font-size: 10px;

}

.online {

color: #4AB19D;

}

.onlineUsrs{

font-family: Lato;

text-align: center;

font-size: 1.1em;

padding: 25px;

}

.offline {

color: #E38968;

}

.me {

color: #94C2ED;

}

.align-left {

text-align: left;

}

.align-right {

text-align: right;

}

.float-right {

float: right;

}

.formJoin{

padding: 3%;

}

.clearfix:after {

visibility: hidden;

display: block;

font-size: 0;

content: " ";

clear: both;

height: 0;

}

.joincht{

font-family: Lato;

text-align: center;

color: #444753;

font-size: 3em;

margin-top: 10%;

}

.foot{

bottom: 1%;

}

</style>

**CHAPTER 8**

**SOFTWARE TESTING**

**8.1 Introduction**

System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error. The purpose of testing is to identify and correct bugs in the developed system. Nothing is complete without testing. Testing is the vital to the success of the system. Testing and validation are the most important steps after the implementation of the developed system[2].

**8.2 System Testing**

In this Project the system testing is performed to ensure that there are no errors in the implemented system. And we find that there the system is working perfectly. The purpose of testing that we used is to identify and correct bugs in the developed system.

**8.2.1 Test Plan**

A test is a sequential step to execute a test operating on a set of predefined inputs to produce certain expected outputs. There are two types of test cases; manual and automated. Mannered test case is executed manually while an automated test case is executed using automated system. In this project we have used both the processes[2].

The features we have tested are:

**•** Verify that the entries are of the correct format.

**•** No duplicate entries should be allowed.

**•** All links take the user to the correct page.

So, in this application I have opted for two type of testing which is Unit Testing and the System Testing.

The basic units in Unit testing are:

**•** Validating the user request

**•** Validating the input given by the user

**•**  Exception handling

The basic units in System testing are:

Integration of all programs is correct or not

**•** Checking whether the entire system after integrating is working as expected.

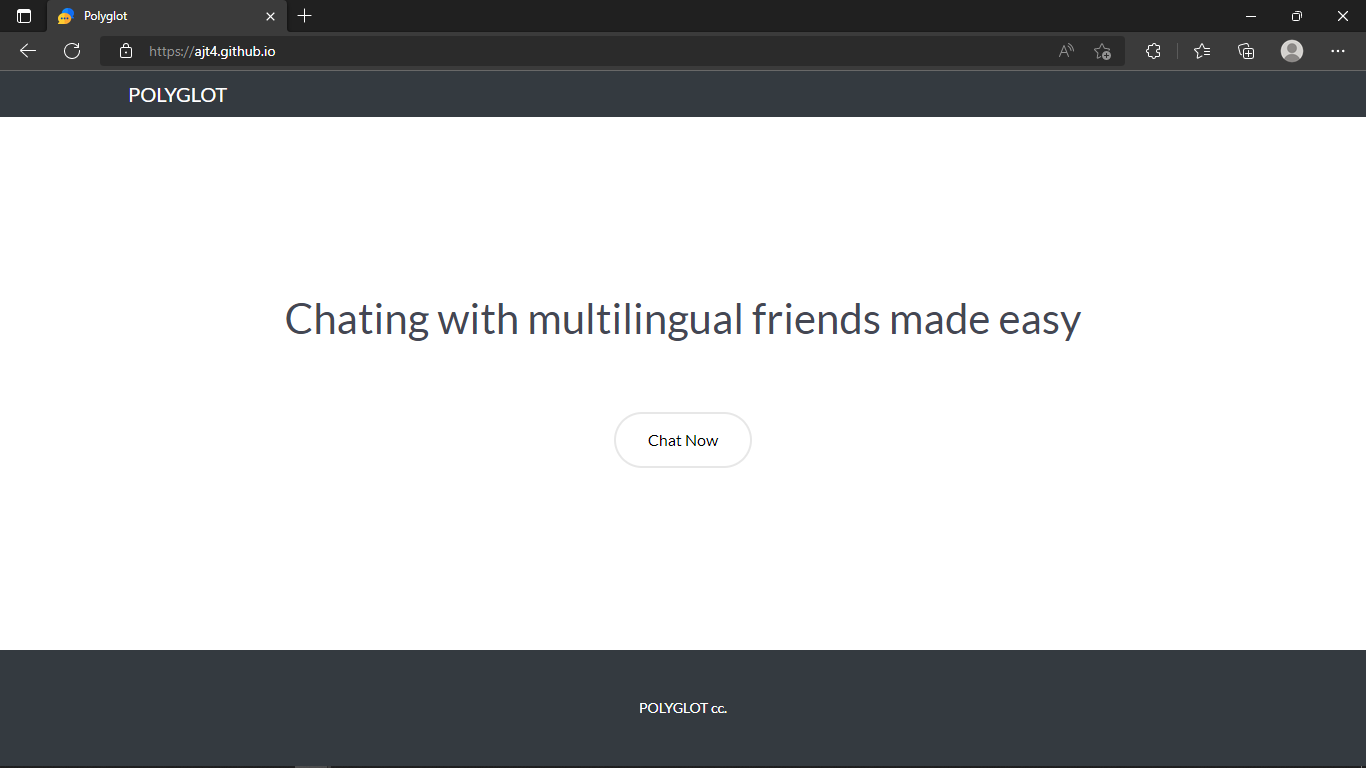
**•** The system is tested as whole after the unit testing.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Number** | **Testing Scenario** | **Expected Result** | **Result** |
| TC-01 | Display Home Page | “Home page should appear on accessing link” | pass |
| TC-02 | UI orientation remains the same in any device | “Design and orientation should remain the same” | pass |
| TC-03 | Proceed to the JOIN page on click of the button | “Join page can be accessed from the home page” | pass |
| TC-04 | Check if any input options are empty | “All fields must be filled or a prompt alert” | pass |
| TC-05 | Same name cannot be entered more than once | “Only one entry per name” | pass |
| TC-06 | Home button navigates to HOME page | “Screen should navigate to the HOME page” | pass |
| TC-07 | When chat button is clicked, the screen should navigate to CHAT SCREEN page | “Screen should navigate to the CHAT SCREEN page” | Pass |
| TC-08 | Chat should translate according to the user’s requirement | “Chat should be translated” | Pass |
| TC-09 | Joiner and creator names should be different | “Prompt message should be displayed” | Pass |

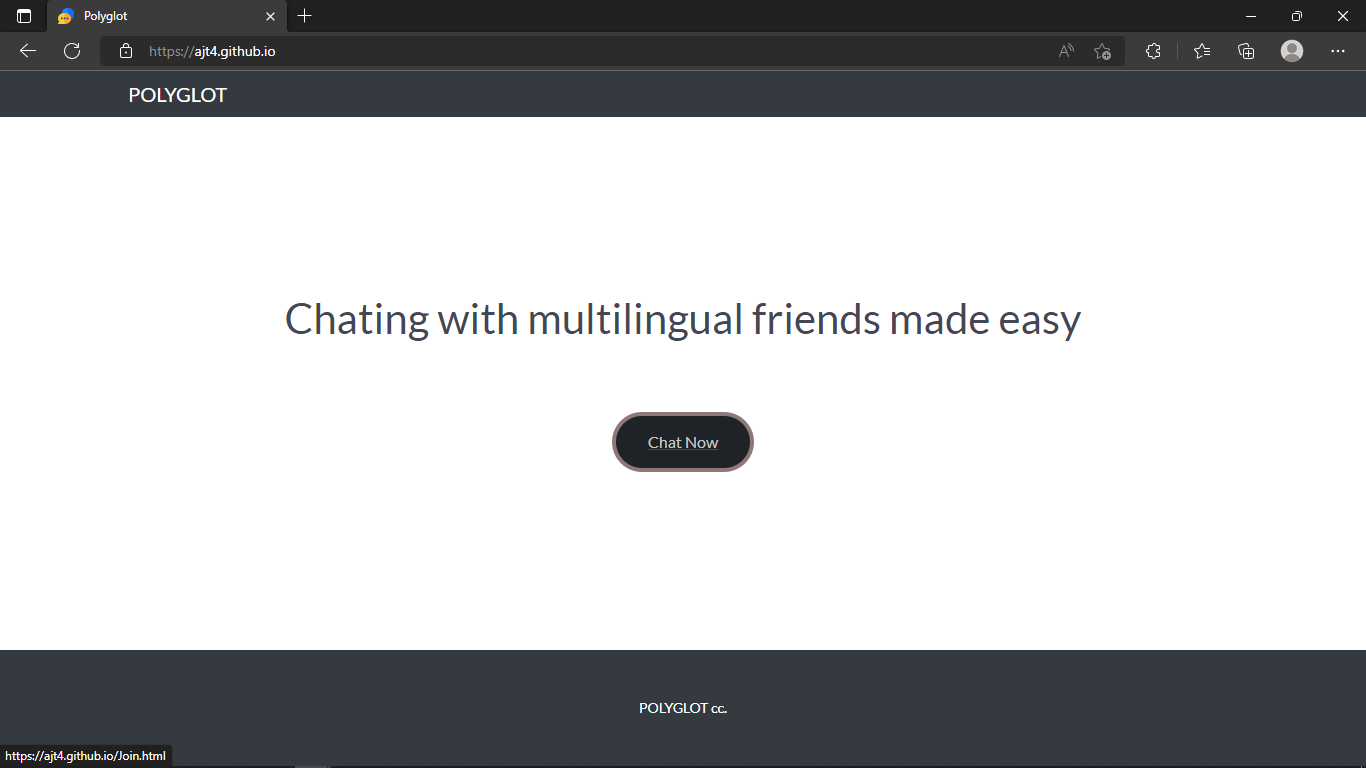
**Table 8.2.1 Testing Table**

**CHAPTER 9**

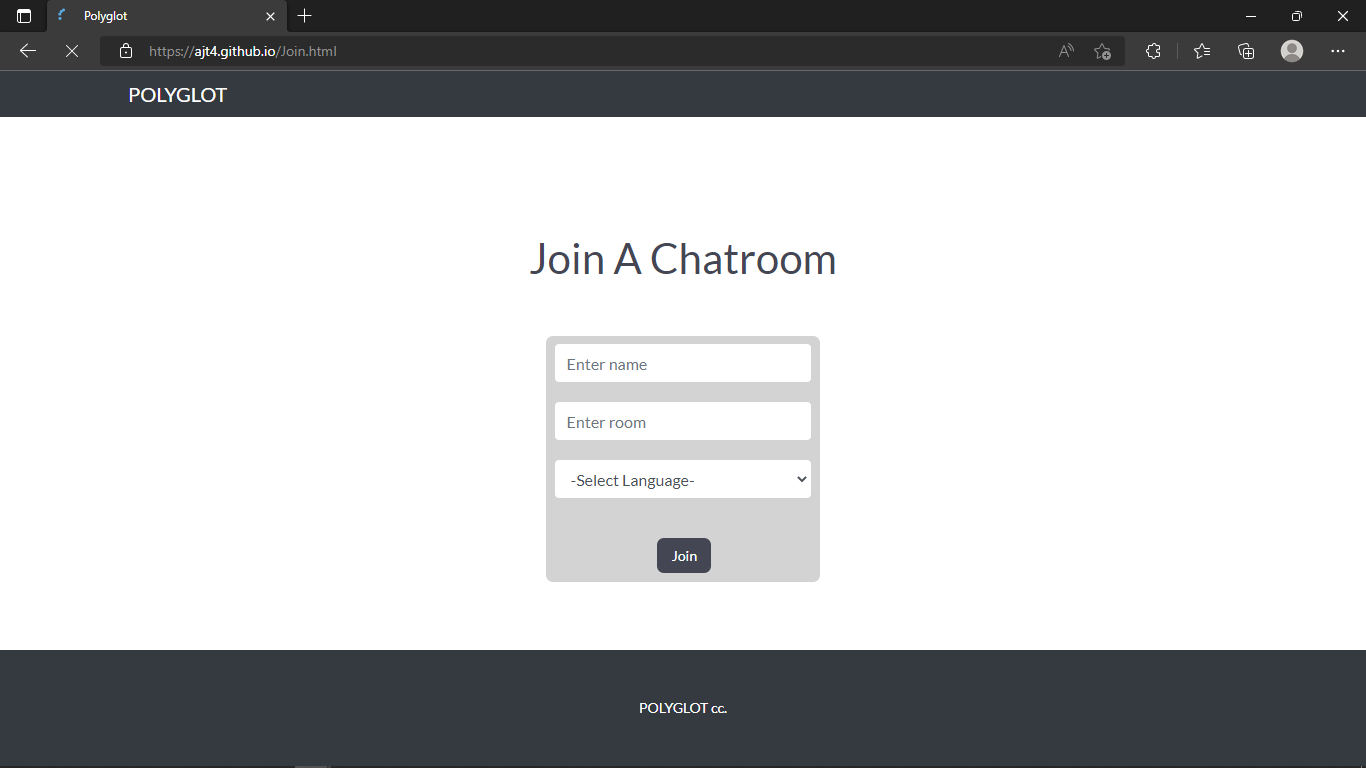
**SNAPSHOTS**



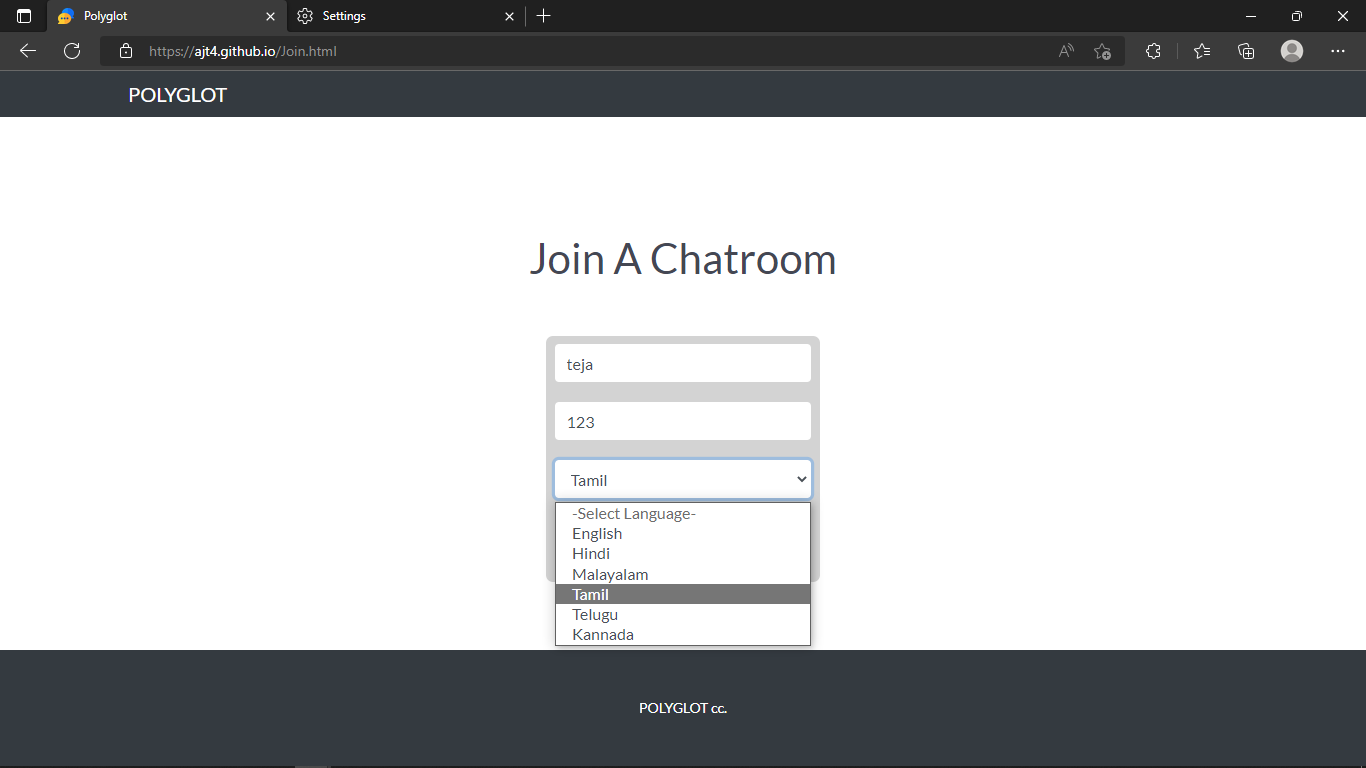
**Fig 9.1 Home Page**



**Fig 9.2 Home Page (Chat Button)**



**Fig 9.3 Join Page**



**Fig 9.4 Join Now (Detailed)**

**CHAPTER 10**

**SYSTEM MAINTENANCE**

Maintenance is the process of making adaptation of the software for external changes (requirements changes or enhancements) and internal changes (fixing bugs)[2]. When changes are made during the maintenance phase all preceding steps of the model must be revisited in this project. In this, we have used two types of maintenance.

**10.1 Corrective maintenance**

In this phase we found the bugs and corrected accordingly.

* API not working →Use of another API
* Server not Responding →Use of Firebase Server
* Repetitive entries were allowed →Added validation condition
* UI not responding →Optimized UI
* Buttons not working as expected →Reprogrammed them according to specification

**10.2 Adaptive Maintenance**

In this, the software undergoes changes that are upgraded and enhanced to include more features and services. According to the environment changes the software undergoes updating processes.

* Adding new languages over time according to demand
* Substituting unnecessary complex code work for easier implementations
* Enabling Current market leading features
* Upgrading UI overtime

**CHAPTER 11**

**CONCLUSION**

Conclusion, in short, I am making a web-application that will help people with different lingual backgrounds to communicate effortlessly in a jiffy. This will reduce the possible errors and misinterpretation that will cause confusion between the individuals.

**CHAPTER 12**

**FUTURE ENHANCEMENT**

# This Since this platform is built in a way to easy to update, we can add any feature that are necessary for the application to make user experience easier as well as to keep application with the trend of market. In future this can be turned into a full-fledged application that compete with the top market competition. We can add more features like chat, like and all the premium features to the user.

**1. Translating voice calls and video calls.**

**2. Encrypted messages.**

**3. Faster server.**

**4. Wider variety of languages.**

**5. Support of group chat.**

**REFERENCES**

**TEXT REFERENCES**

[1] Srikanth S -Web Programming by Skyward Publication.

# [2] Ian Sommerville-Software Engineering by Pearson

**WEB REFERENCES**

[3] [www.youtube.com](http://www.youtube.com)

[4] [www.w3schools.com](http://www.w3schools.coml)

[5] [www.github.com](http://www.github.com)

[6] [www.stackoverflow.com](http://www.stackoverflow.coml)

[7] [www.geeksforgeeks.com](http://www.geeksforgeeks.com)

[8] [www.quora.com](http://www.quora.com)