









Index.js คือไฟล์ server

เข้าหน้า login โดยจะเข้าผ่าน localhost:5000

หลังจากนนั้นทำการใส่ชื่อและอีเมลเข้าไปแล้วกดปุ่ม enter chat

จะเข้าไปที่หน้า home.js จะเห็นช่องแชทและคนที่กำลังออนไลน์อยู่ในขณะนั้น

และเมื่อพิมพ์ข้อความเข้าไปมันจะรับ/ส่งไปที่ index.js

1.หน้า login เป็นหน้าที่จะมีการให้กรอกชื่อและอีเมล

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<link *rel*="stylesheet" *href*="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

<meta *charset*="UTF-8">

<meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">

<meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

<title>Login</title>

<style>

body {

background: linear-gradient(120*deg*, rgb(60, 241, 238), rgb(20, 16, 230));

overflow: hidden;

}

*.container*{

display: flex;

justify-content: center;

align-items: center;

flex-direction: column;

height: 100*vh*;

width: 100*vw*;

color: rgb(0, 0, 0);

}

div{

border: 2*px* solid #dedede;

color: rgb(0, 0, 0);

border-radius: 5*px*;

padding: 10*px*;

margin: 10*px* 0;

background-attachment: scroll;

}

h1 {

font-size: 3.5*em*;

padding: 0.5*em*;

margin: 0;

font-weight: 300;

text-align: center;

}

form {

background: linear-gradient(120*deg*, rgb(60, 241, 238), rgb(20, 16, 230));

}

*.text-muted* {

color: #000000*!important*;

text-shadow: 3*px* 3*px* 5*px* rgb(255, 255, 255);

}

*.btn-primary* {

background-color :red;

}

</style>

</head>

<body>

<div *class*="container p-5">

<h1 *class*="text-center mb-5 text-uppercase text-muted">Real Time Chat</h1>

<div *class*="text-uppercase center">

<form *action*="" *method*="POST">

<div *class*="form-group">

<label *for*="\_email">Your Name</label>

<input *type*="text" *name*="user\_name" *id*="\_rname" *class*="form-control" *placeholder*="Name" *rnodeequired*>

</div>

<div *class*="form-group">

<label *for*="\_email">Email</label>

<input *type*="text" *name*="user\_email" *id*="\_email" *class*="form-control" *placeholder*="Email" *rnodeequired*>

</div>

***<***% if (locals.login\_errors) {

login\_errors.forEach(function(error\_msg){ %>

<div *class*= "alert alert-danger" *role*="alert">***<***%= error\_msg %></div>

***<***% });

} %>

<button *type*="submit" *class*="btn btn-primary">Enter Chat</button>

</form>

</div>

</div>

</body>

</html>

2.หน้า home หน้านี้จะเป็นการแสดงช่องแชทและคนที่กำลังออนไลน์อยู่

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<link *rel*="stylesheet" *href*="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

<meta *charset*="UTF-8" />

<link *rel*="stylesheet" *href*="/css/style.css">

<meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

<title>Real time Chat App</title>

</head>

<body>

<h1><***center***>REAL TIME CHAT</***center***></h1>

<div *class*="container">

<div *class*="inbox">

<div *class*="inbox\_\_people-list">

<h4>UserOnlines</h4>

<div *class*="inbox\_\_people"></div>

<a *href*="/logout" *class*="btn btn-danger inbox\_logout">Logout</a>

</div>

<div *class*="charnum">

<p *id*="charNum">100 characters remaining</p>

</div>

<div *class*= "inbox\_\_messages">

<div *class*="messages\_\_history"></div>

<div *class*="fallback"></div>

</div>

</div>

<form *class*="message\_form inbox\_input" >

<input *type*="text" *class*="message\_form\_\_input" *maxlength*="100" *placeholder*="Type a message" *onkeyup*="*countChars*(*this*);" />

</form>

</div>

<script *src*="https://cdn.socket.io/3.1.3/socket.io.min.js" *integrity*="sha384-cPwlPLvBTa3sKAgddT6krw0cJat7egBga3DJepJyrLl4Q9/5WLra3rrnMcyTyOnh" *crossorigin*="anonymous"></script>

<script>

*function* *countChars*(obj) {

*var* maxLength *= 100*;

*var* strLength *= obj.value.length*;

*var* charRemain *= (*maxLength *-* strLength*)*;

*document*.*getElementById*("charNum").*innerHTML* *=* charRemain *+* ' characters remaining';

}

*const socket = io()*;

*const inboxPeople = document.querySelector(*".inbox\_\_people"*)*;

*const inputField = document.querySelector(*".message\_form\_\_input"*)*;

*const messageForm = document.querySelector(*".message\_form"*)*;

*const messageBox = document.querySelector(*".messages\_\_history"*)*;

*const fallback = document.querySelector(*".fallback"*)*;

*let* userName *=* "";

*const newUserConnected = (*user*) => {*

userName *=* user *||* "<%=name%>"*;*

*socket.emit(*"new user"*,* userName*);*

*addToUsersBox(*userName*);*

*}*;

*const addToUsersBox = (*userName*) => {*

*if (!!document.querySelector(`.${*userName*}-userlist`)) {*

*return;*

*}*

*const userBox = `*

*<div class="chat\_ib ${*userName*}-userlist">*

*<h5>${*userName*}</h5>*

*</div>*

*`;*

*inboxPeople.innerHTML +=* userBox*;*

*}*;

*const addNewMessage = ({* user*,* message *}) => {*

*const time = new Date();*

*const formattedTime = time.toLocaleString(*"en-US"*, { hour:* "numeric"*, minute:* "numeric"*, day:*"numeric"*, year:*"2-digit"*,month:*"2-digit" *});*

*const receivedMsg = `*

*<div class="incoming\_\_message">*

*<div class="received\_\_message">*

*<p>${*message*}</p>*

*<div class="message\_\_info">*

*<span class="message\_\_author">${*user*}</span>*

*<span class="time\_date">${*formattedTime*}</span>*

*</div>*

*</div>*

*</div>`;*

*const myMsg = `*

*<div class="outgoing\_\_message">*

*<div class="sent\_\_message">*

*<p>${*message*}</p>*

*<div class="message\_\_info">*

*<span class="time\_date">${*formattedTime*}</span>*

*</div>*

*</div>*

*</div>`;*

*messageBox.innerHTML +=* user *===* userName *?* myMsg *:* receivedMsg*;*

*}*;

*// new user is created so we generate nickname and emit event*

*newUserConnected*();

*messageForm*.*addEventListener*("submit", (e) *=>* {

*e*.*preventDefault*();

*if* (*!inputField*.*value*) {

*return*;

}

*socket*.*emit*("chat message", {

message: *inputField*.*value*,

nick: userName,

});

*inputField*.*value* *=* "";

});

*inputField*.*addEventListener*("keyup", () *=>* {

*socket*.*emit*("typing", {

isTyping: *inputField*.*value*.length *>* 0,

nick: userName,

});

});

*socket*.*on*("new user", *function* (data) {

*data*.*map*((user) *=>* *addToUsersBox*(user));

});

*socket*.*on*("user disconnected", *function* (userName) {

*document*.*querySelector*(`.${userName}-userlist`).*remove*();

});

*socket*.*on*("chat message", *function* (data) {

*addNewMessage*({ user: *data*.*nick*, message: *data*.*message* });

});

*socket*.*on*("typing", *function* (data) {

*const { isTyping, nick } =* data;

*if* (*!*isTyping) {

*fallback*.*innerHTML* *=* "";

*return*;

}

*fallback*.*innerHTML* *=* `<p>${nick} is typing...</p>`;

});

</script>

</body>

</html>

3.Style.css จะเป็นการตกแต่งหน้า UI ต่างๆในเว็บ

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Nunito', sans-serif;

}

html, body {

background: linear-gradient(120*deg*, rgb(10, 6, 233), rgb(6, 223, 223));

overflow: hidden;

}

*.container*{

display: flex;

justify-content: center;

align-items: center;

flex-direction: column;

height: 100*vh*;

width: 100*vw*;

}

*.inbox* {

width: 600*px*;

height: 72*%*;

border-radius: 0.2*em*;

position: relative;

box-shadow: 1*px* 1*px* 12*px* rgba(255, 255, 255, 0.877);

}

form {

background: linear-gradient(120*deg*, rgb(60, 241, 238), rgb(20, 16, 230));

}

*.inbox\_input* {

width: 600*px*;

border-radius: 0.1*em*;

position: relative;

}

*.inbox\_logout* {

margin: 40*px*;

bottom: -20*px*;

position: absolute;

right: 0;

}

*.message\_form\_\_input*{

background: rgba(68, 4, 247, 0.486);

position: absolute;

bottom: 0;

left: 0;

border: none;

width: 73*%*;

padding: 1.2*em*;

outline: none;

color: rgba(255, 255, 255, 0.9);

font-weight: 300;

}

*.chat\_ib*{

float: left;

text-align: left;

margin: -0.25*em* 2*em*;

font-size: 0.7*em*;

font-weight: 300;

color: rgb(255, 255, 255);

width: 200*px*;

}

*.inbox\_\_people-list-bot*{

border-radius: 0.2*em*;

position: relative;

}

*.inbox\_\_people-list* {

background: rgba(47, 4, 206, 0.774);

width: 25*%*;

height: 100*%*;

float: right;

border-top-right-radius: 0.2*em*;

border-bottom-right-radius: 0.2*em*;

}

*.inbox\_\_people-list* h4 {

background: rgb(42, 26, 185);

color: rgba(255, 255, 255, 0.9);

font-size: 0.9*em*;

padding: 1*em*;

margin: 0;

font-weight: 300;

text-align: center;

}

*.inbox\_\_messages::-webkit-scrollbar-track*

{

-webkit-box-shadow: inset 0 0 6*px* rgba(0,0,0,0.3);

background-color: #F5F5F5;

}

*.inbox\_\_messages::-webkit-scrollbar*

{

width: 10*px*;

background-color: #3e01cc;

}

*.inbox\_\_messages::-webkit-scrollbar-thumb*

{

background-color: rgb(87, 1, 226);

background-image: ***-webkit-gradient***(linear, 0 0, 0 100*%*,

***color-stop***(.5, rgba(255, 255, 255, .2)),

***color-stop***(.5, transparent), ***to***(transparent));

}

*.inbox\_\_messages*{

overflow-y: scroll;

overflow-x: hidden;

width: 450*px*;

background: rgba(0, 0, 0, 0.1);

height: 91*%*;

border-top-right-radius: 0.2*em*;

border-bottom-right-radius: 0.2*em*;

}

*.outgoing\_\_message* , *.incoming\_\_message* {

background: linear-gradient(120*deg*, rgb(95, 8, 235), rgb(63, 225, 236));

padding: 1*em* 0;

height: auto;

width: 60*%*;

border-radius: 5*px*;

margin: 2*em* 1*em*;

}

*.outgoing\_\_message* {

float: right;

}

*.incoming\_\_message* {

float: left;

}

*.sent\_\_message* {

color: rgb(255, 255, 255);

font-size: 1*em*;

margin: 0 1*em*;

}

*.received\_\_message* {

color: rgb(255, 255, 255);

font-size: 1*em*;

margin: 0 1*em*;

}

*.message\_\_info*{

margin: 0 2*em*;

}

*.message\_\_author*,*.time\_date*{

color: rgb(255, 255, 255);

font-size: 0.5*em*;

margin: 1 0*em*;

}

h1{

color: rgb(0, 0, 0);

text-shadow: 3*px* 3*px* 5*px* rgb(255, 255, 255);

}

*.btn*{

top: 320*px*;

font-family: "proxima-nova", sans-serif;

font-weight: 800;

font-size: 13*px*;

text-transform: uppercase*!important*;

letter-spacing: 2*px*;

color: rgb(255, 255, 255);

cursor: hand;

text-align: center;

text-transform: capitalize;

border: 1.5*px* solid rgb(255, 247, 247);

position: relative;

overflow: hidden*!important*;

-webkit-transition: all .3*s* ease-in-out;

-moz-transition: all .3*s* ease-in-out;

-o-transition: all .3*s* ease-in-out;

transition: all .3*s* ease-in-out;

background: transparent*!important*;

z-index:10;

}

*.btn:hover*{

border: 1*px* solid #ffffff;

color: #ffffff*!important*;

}

*.btn::before* {

content: '';

width: 0*%*;

height: 100*%*;

display: block;

background: #ff0404;

position: absolute;

-ms-transform: skewX(-20*deg*);

-webkit-transform: skewX(-20*deg*);

transform: skewX(-20*deg*);

left: -10*%*;

opacity: 1;

top: 0;

z-index: -12;

-moz-transition: all .7*s* cubic-bezier(0.77, 0, 0.175, 1);

-o-transition: all .7*s* cubic-bezier(0.77, 0, 0.175, 1);

-webkit-transition: all .7*s* cubic-bezier(0.77, 0, 0.175, 1);

transition: all .7*s* cubic-bezier(0.77, 0, 0.175, 1);

box-shadow:2*px* 0*px* 14*px* rgba(0,0,0,.6);

}

*.btn::after* {

content: '';

width: 0*%*;

height: 100*%*;

display: block;

background: #1bb9e9ab;

position: absolute;

-ms-transform: skewX(-20*deg*);

-webkit-transform: skewX(-20*deg*);

transform: skewX(-20*deg*);

left: -10*%*;

opacity: 0;

top: 0;

z-index: -15;

-webkit-transition: all .94*s* cubic-bezier(.2,.95,.57,.99);

-moz-transition: all .4*s* cubic-bezier(.2,.95,.57,.99);

-o-transition: all .4*s* cubic-bezier(.2,.95,.57,.99);

transition: all .4*s* cubic-bezier(.2,.95,.57,.99);

box-shadow: 2*px* 0*px* 14*px* rgba(0,0,0,.6);

}

*.btn:hover::before*, *.btn1O:hover::before*{

opacity:1;

width: 116*%*;

}

*.btn:hover::after*, *.btn1O:hover::after*{

opacity:1;

width: 120*%*;

}

4.ER

