

```

import { Component, inject, OnInit } from '@angular/core';
import { UsersService } from '../Services/customerService/users.service';
import { Users } from '../Models/Users';
import { CommonModule } from '@angular/common';
import { ChatService } from '../Services/MessageService/chat.service';
import { Chat } from '../Models/chat.model';
import { UserLocalStorageService } from '../Services/userLocalStorage.service';
import { use } from 'echarts';
import { FormsModule } from '@angular/forms';

```

```

@Component({
  selector: 'app-chat',
  standalone: true,
  imports: [CommonModule, FormsModule],
  templateUrl: './chat.component.html',
  styleUrls: ['./chat.component.css'],
})
export class ChatComponent {
  // userService = inject(UsersService);
  // chatService = inject(ChatService);
  // users: Users[] = [];
  // chatList: Chat[] = [];
  // message: Chat;
  // userLocalStorageService = inject(UserLocalStorageService);
  // loggedUser: any;
  // clickedUser: any;
  // isClicked = false;

  // constructor() {
  //   this.message = new Chat();
  // }

  // ngOnInit(): void {
  //   this.getAllUsers();
  //   this.userLocalStorageService.user$.subscribe((user) => {
  //     this.loggedUser = user;
  //   });
  //   this.isClicked = false;
  // }

  // getAllUsers() {
  //   this.userService.getUsers().subscribe((res: any) => {
  //     this.users = res;
  //   });
  // }

  // GetUserChat(user: any) {
  //   this.isClicked = true;
  //   this.clickedUser = user;
  //   debugger;
  //   this.chatService
  //     .getChat(user.userId, this.loggedUser.user.userId)
  //     .subscribe((res: any) => {
  //       if (res.success) {
  //         this.chatList = res.res;
  //       }
  //     });
  // }

```

```

//    }
//  });
//}

// onSubmit(event: Event): void {
//  debugger;
//  event.preventDefault(); // Prevent the default form submission
//  if (this.message.message !== '') {
//    this.message.userId = this.clickedUser.userId;
//    this.message.senderId = this.loggedUser.user.userId;
//    this.chatService.AddMessage(this.message).subscribe((res: any) => {
//      if (res.success) {
//        this.GetUserChat(this.clickedUser);
//        this.message = new Chat();
//      }
//    });
//  }
//}

// getTimeDifference(inputTime: Date | undefined): string {
//  if (!inputTime) {
//    return 'No time provided'; // Handle undefined case
//  }

//  const inputDate = new Date(inputTime); // Convert to Date object
//  const currentDate = new Date(); // Get the current time

//  const timeDiff = currentDate.getTime() - inputDate.getTime(); // Difference in milliseconds

//  // Convert the difference to a readable format
//  const seconds = Math.floor(timeDiff / 1000);
//  const minutes = Math.floor(seconds / 60);
//  const hours = Math.floor(minutes / 60);
//  const days = Math.floor(hours / 24);

//  if (days > 0) {
//    return `${days} day(s) ago`;
//  } else if (hours > 0) {
//    return `${hours} hour(s) ago`;
//  } else if (minutes > 0) {
//    return `${minutes} minute(s) ago`;
//  } else {
//    return `${seconds} second(s) ago`;
//  }
//}
}

```

```

<!-- <div class="container-fluid">
<div class="row clearfix">
  <div class="col-lg-12 minHeight">
    <div class="card chat-app">
      <div id="plist" class="people-list">
        <div class="input-group">
          <div class="input-group-prepend"></div>
          <input type="text" class="form-control" placeholder="Search..." />
        </div>
        <div class="overflow">
          <ul
            class="list-unstyled chat-list mt-2 mb-0"
            *ngFor="let user of users"
          >
            <li class="clearfix">
              <a (click)="GetUserChat(user)">
                <img [src]="user.imagePath" alt="avatar" />
                <div class="about">
                  <div class="name">{{ user.username }}</div>
                  <div class="status">Last message 7 mins ago</div>
                </div></a>
              </li>
            </ul>
          </div>
        <div class="chat minHeightChat">
          <div class="chat-header clearfix">
            <div class="row">
              <div class="col-lg-6">
                <a
                  href="javascript:void(0);"
                  data-toggle="modal"
                  data-target="#view_info"
                >
                  
                  
                </a>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

        <div class="chat-about" *ngIf="isClicked">
            <h6 class="m-b-0">{{ clickedUser.username }}</h6>
            <small>Last message: 2 hours ago</small>
        </div>
    </div>
</div>
<div
    class="chat-history minHeightChathistory"
    style="
        background-image: url('../assets/img/chat-with-user-image-removebg.png');
    "
>
    <span
        *ngIf="!isClicked"
        class="h-100 p-0 w-100 message container-image"
    >
    <div class="centered">
        </div></span>
    >
    <ul class="m-b-0" *ngFor="let chat of chatList">
        <li
            class="clearfix"
            *ngIf="loggedUser.user.userId != chat.userId"
        >
            <div class="float-right">
                <div class="message-data">
                    <span class="message-data-time mb-2 me-2">{{
                        getTimeDifference(chat.timestamp)
                    }}</span>
                </div>
                <div class="message other-message">
                    {{ chat.message }}
                </div>
            </div>
        </li>
        <br />
        <li
            class="clearfix"
            *ngIf="loggedUser.user.userId == chat.userId"
        >
            <div class="message-data">
                <span class="message-data-time">{{
                    getTimeDifference(chat.timestamp)
                }}</span>
            </div>
            <div class="message my-message">{{ chat.message }}</div>
        </li>
    </ul>
</div>
<div class="chat-message clearfix">
    <div class="input-group mb-0">
        <form (submit)="onSubmit($event)">

```

```

<div class="row">
  <div class="col-2">
    <div class="input-group-prepend me-1">
      <button type="submit" class="input-group-text">
        <i
          class="fa fa-paper-plane"
          aria-hidden="true"
          style="font-size: 22px"
        ></i>
      </button>
    </div>
  </div>
  <div class="col-10">
    <input
      type="text"
      class="form-control"
      [(ngModel)]="message.message"
      name="message"
      placeholder="Enter text here..."
    />
  </div>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div> -->

```

```

.container {
  background-color: #f4f7f6;
  margin-top: 20px;
}
.container-image {
  position: relative;
  text-align: center;
  color: white;
}
.centered {
  font-size: 40px;
  position: absolute;
  top: 45%;
  left: 50%;
  transform: translate(-50%, -50%);
}
.card {
  background: #fff;
  transition: 0.5s;
  border: 0;
  margin-bottom: 30px;
}

```

```
border-radius: 0.55rem;
position: relative;
width: 100%;
box-shadow: 0 1px 2px 0 rgb(0 0 0 / 10%);
}

.chat-app .people-list {
width: 320px;
position: absolute;
left: 0;
top: 0;
padding: 20px;
z-index: 7;
}

.chat-app .chat {
margin-left: 320px;
border-left: 1px solid #eaeaea;
}

.people-list {
-moz-transition: 0.5s;
-o-transition: 0.5s;
-webkit-transition: 0.5s;
transition: 0.5s;
}

.people-list .chat-list li {
padding: 10px 15px;
list-style: none;
border-radius: 3px;
}

.people-list .chat-list li:hover {
background: #efefef;
cursor: pointer;
}

.people-list .chat-list li.active {
background: #efefef;
}

.people-list .chat-list li .name {
font-size: 15px;
}

.people-list .chat-list img {
width: 45px;
border-radius: 50%;
}

.people-list img {
float: left;
border-radius: 50%;
}
```

```
.people-list .about {
  float: left;
  padding-left: 8px;
}

.people-list .status {
  color: #999;
  font-size: 13px;
}

.chat .chat-header {
  padding: 15px 20px;
  border-bottom: 2px solid #f4f7f6;
}

.chat .chat-header img {
  float: left;
  border-radius: 40px;
  width: 40px;
}

.chat .chat-header .chat-about {
  float: left;
  padding-left: 10px;
}

.chat .chat-history {
  padding: 20px;
  border-bottom: 2px solid #fff;
}

.chat .chat-history ul {
  padding: 0;
}

.chat .chat-history ul li {
  list-style: none;
  margin-bottom: 30px;
}

.chat .chat-history ul li:last-child {
  margin-bottom: 0px;
}

.chat .chat-history .message-data {
  margin-bottom: 15px;
}

.chat .chat-history .message-data img {
  border-radius: 40px;
  width: 40px;
}

.chat .chat-history .message-data-time {
  color: #434651;
}
```

```
padding-left: 6px;  
font-size: 13px;  
}
```

```
.chat .chat-history .message {  
  color: #444;  
  padding: 10px 12px;  
  line-height: 16px;  
  font-size: 14px;  
  border-radius: 7px;  
  display: inline-block;  
  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: #fff;  
  border-width: 10px;  
  margin-left: -10px;  
}
```

```
.chat .chat-history .my-message {  
  background: #efefef;  
}
```

```
.chat .chat-history .my-message:after {  
  bottom: 100%;  
  left: 30px;  
  border: solid transparent;  
  content: " ";  
  height: 0;  
  width: 0;  
  position: absolute;  
  pointer-events: none;  
  border-bottom-color: #efefef;  
  border-width: 10px;  
  margin-left: -10px;  
}
```

```
.chat .chat-history .other-message {  
  background: #e8f1f3;  
  text-align: right;  
}
```

```
.chat .chat-history .other-message:after {  
  border-bottom-color: #e8f1f3;  
  left: 93%;  
}
```



```
.chat .chat-message {
  padding: 20px;
}

.online,
.offline,
.me {
  margin-right: 2px;
  font-size: 8px;
  vertical-align: middle;
}

.online {
  color: #86c541;
}

.offline {
  color: #e47297;
}

.me {
  color: #1d8ecd;
}

.float-right {
  float: right;
}

.clearfix:after {
  visibility: hidden;
  display: block;
  font-size: 0;
  content: " ";
  clear: both;
  height: 0;
}

.overflow {
  max-height: 700px !important;
  overflow: auto;
}

.minHeightChat {
  height: 800px;
}

.minHeightChathistory {
  height: 620px;
  overflow: auto;
}

@media only screen and (max-width: 767px) {
  .chat-app .people-list {
    height: 465px;
    width: 100%;
  }
}
```

```
    overflow-x: auto;
    background: #fff;
    left: -400px;
    display: none;
}
.chat-app .people-list.open {
    left: 0;
}
.chat-app .chat {
    margin: 0;
}
.chat-app .chat .chat-header {
    border-radius: 0.55rem 0.55rem 0 0;
}
.chat-app .chat-history {
    height: 300px;
    overflow-x: auto;
}
}
```

```
@media only screen and (min-width: 768px) and (max-width: 992px) {
    .chat-app .chat-list {
        height: 650px;
        overflow-x: auto;
    }
    .chat-app .chat-history {
        height: 600px;
        overflow-x: auto;
    }
}
```

```
@media only screen and (min-device-width: 768px) and (max-device-width: 1024px) and (orientation:
landscape) and (-webkit-min-device-pixel-ratio: 1) {
    .chat-app .chat-list {
        height: 480px;
        overflow-x: auto;
    }
    .chat-app .chat-history {
        height: calc(100vh - 350px);
        overflow-x: auto;
    }
}
```