**Dự án CHAT VIBER**

**Trang home: Hiển thị thông tin gồm 2 tabs**

* trong file ts: khai báo 2 biến

tab1Root = ChatHomePage;

tab2Root = TestPage;

* trong file html: hiển thị thông tin 2 tabs ra

<ion-tabs>

<ion-tab [root]="tab1Root" tabTitle="ChatHomePage" tabIcon="call"></ion-tab>

<ion-tab [root]="tab2Root" tabTitle="Test" tabIcon="information-circle"></ion-tab>

</ion-tabs>

**Trang chat-home: Chức năng chat**

**Chức năng lấy danh sách rooms đã lưu trong đĩa và hiển thị ra**

* Trong file html: hiển thị giao diện gồm các slide

<ion-slides (ionSlideDidChange)="slideChanged()">

<!-- #id=0 home -->

<ion-slide>

</ion-slide>

<!-- #id=1 chatting -->

<ion-slide>

</ion-slide>

<!-- #id=2 them nhom -->

<ion-slide>

</ion-slide>

<!-- #id=3 setting -->

<ion-slide>

</ion-slide>

</ion-slides>

* Trong file ts, dùng hàm **ngOnInit** để lấy danh sách rooms đã lưu trong đĩa để hiển thị ra

ngOnInit() {

this.slides.lockSwipes(true);

this.addFromGroup = this.formBuilder.group({

room\_name: '',

image: '',

message: '',

time: ''

});

this.rooms = this.apiStorage.getUserRooms(this.user);

}

* Trong file html, hiển thị danh sách rooms ra ở slide **home**

<!-- #id=0 home -->

<ion-slide>

<ion-card>

<ion-list>

<ion-item-sliding \*ngFor="let room of rooms">

<ion-item>

<button class="list-card" ion-item (click)="goRoom(room)">

<ion-avatar item-start>

<img src="{{room.image?room.image:image\_default}}">

</ion-avatar>

<h2 class="name">{{ room?.name }}</h2>

<p>{{ room?.message }}</p>

<ion-note>{{ room?.time | date:'dd.MM hh:mm' }}</ion-note>

</button>

</ion-item>

<ion-item-options side="right">

<button ion-button color="danger" (click)="deleteRoom(room.name)">

<ion-icon name="trash" ios="ios-trash"></ion-icon>

Delete

</button>

</ion-item-options>

</ion-item-sliding>

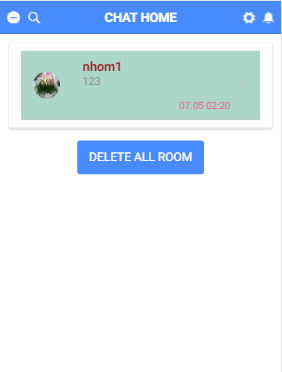
</ion-list>

</ion-card>

<button ion-button (click)="deleteAllRoom()" \*ngIf="rooms.length>0">DELETE ALL ROOM</button>

</ion-slide>

* Giao diện



**Chức năng thêm room mới**

* Trong phần header của file html, tạo button để khi kích vào thì thực hiện hàm **formAddRoom**

<button ion-button \*ngIf="slideIndex==0" (click)="formAddRoom()">

<ion-icon name="add-circle" ios="ios-add-circle" md="md-add-circle"></ion-icon>

</button>

* Trong file ts, định nghĩa hàm **formAddRoom**

formAddRoom() {

this.addFromGroup = this.formBuilder.group({

room\_name: '',

image: '',

message: '',

time: new Date().getTime()

});

this.goToSlide(2);

}

* Trong slide 2, hiện form để nhập các thông tin cho room mới

<form (ngSubmit)="onSubmit()" [formGroup]="addFromGroup" \*ngIf="!(isShowInfo)">

<ion-list>

<ion-item>

<ion-icon name="leaf" item-start></ion-icon>

<ion-label floating>Tên nhóm</ion-label>

<ion-input type="text" formControlName="room\_name"></ion-input>

<ion-icon name="add" item-end></ion-icon>

</ion-item>

<ion-item>

<ion-icon name="leaf" item-start></ion-icon>

<ion-label floating>Nhap vao duong dan hinh anh</ion-label>

<ion-input type="text" formControlName="image"></ion-input>

<ion-icon name="add" item-end></ion-icon>

</ion-item>

<ion-item>

<ion-icon name="leaf" item-start></ion-icon>

<ion-label floating>Tin nhan cuoi cung</ion-label>

<ion-input type="text" formControlName="message"></ion-input>

<ion-icon name="add" item-end></ion-icon>

</ion-item>

<ion-item>

<ion-buttons start>

<button ion-button type="submit" icon-end round>

Them nhom

<ion-icon name="share-alt"></ion-icon>

</button>

</ion-buttons>

</ion-item>

</ion-list>

</form>

* Sau khi nhập các thông tin và bấm submit thì thực hiện hàm **onSubmit,** thực hiện lấy thông tin đã nhập và lưu vào đĩa, sau đó chuyển về slide home

onSubmit() {

this.rooms.push({

name: this.addFromGroup.value.room\_name,

image: this.addFromGroup.value.image,

message: this.addFromGroup.value.message,

time: this.addFromGroup.value.time,

messages: [],

});

this.apiStorage.saveUserRooms(this.user, this.rooms);

this.goToSlide(slideSelected.home);

}

**Chức năng chat**

* Mỗi khi kích vào một room ở slide home thì thực hiện hàm goRoom
* Trong file ts, định nghĩa hàm **goRoom**, tham số truyền vào là room muốn vào, hàm này sẽ lấy lại những tin nhắn của room muốn vào, sau đó chuyển sang slide **chatting**

goRoom(room) {

this.room = room;

this.messages = this.apiStorage.getUserRoomMessages(this.user, this.room);

this.goToSlide(slideSelected.chatting);

}

* Trong slide **chatting**, hiển thị những tin nhắn của room ra trình duyệt

<ion-grid>

<ion-row \*ngFor="let message of messages">

<ion-col class="other-avatar" col-2 \*ngIf="message?.user?.username !== user?.username">

<ion-item class="other-avatar">

<ion-avatar item-start>

<img

src="{{message?.user?.image?message?.user?.image.toLowerCase().indexOf('://')>0?message?.user?.image:(authenticationServer + '/get-avatar/' + message?.user?.image + '?token=' + token):image\_default}}" />

</ion-avatar>

</ion-item>

</ion-col>

<ion-col class="message other\_message" offset-1 col-9 \*ngIf="message?.user?.username !== user?.username">

<div class="nick-name">{{ message?.user?.username }}:</div>

<br>

<div class="text-message">{{ message?.text }}</div>

<br>

<div class="time">{{message?.created | date:'dd.MM hh:mm'}}</div>

</ion-col>

<ion-col class="message my\_message" offset-1 col-9 \*ngIf="message?.user?.username === user?.username">

<div class="nick-name">{{ message?.user?.username }}:</div>

<br>

<div class="text-message">{{ message?.text }}</div>

<br>

<div class="time">{{message?.created | date:'dd.MM hh:mm'}}</div>

</ion-col>

<ion-col class="other-avatar" col-2 \*ngIf="message?.user?.username === user?.username">

<ion-item class="other-avatar">

<ion-avatar item-start>

<img

src="{{message?.user?.image?message?.user?.image.toLowerCase().indexOf('://')>0?message?.user?.image:(authenticationServer + '/get-avatar/' + message?.user?.image + '?token=' + token):image\_default}}" />

</ion-avatar>

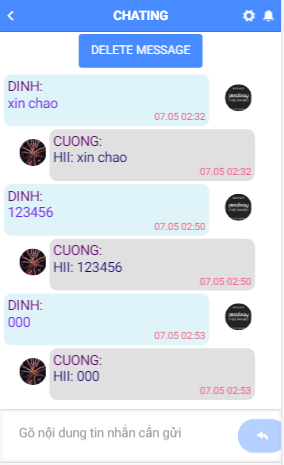
</ion-item>

</ion-col>

</ion-row>

</ion-grid>

* Giao diện:



* Chức năng chat: ở phần footer, nếu ở slide chatting (index = 1) thì hiện form để chat

<ion-toolbar \*ngIf="slideIndex==1">

<ion-row class="message\_row">

<ion-col col-10>

<ion-item no-lines>

<ion-input type="text" placeholder="Gõ nội dung tin nhắn cần gửi" [(ngModel)]="message"

(keyup.enter)="sendMessage()"></ion-input>

</ion-item>

</ion-col>

<ion-col col-2>

<button ion-button type="button" round color="primary" (click)="sendMessage()" [disabled]="message === ''">

<ion-icon name="undo"></ion-icon>

</button>

</ion-col>

</ion-row>

</ion-toolbar>

* Sau khi nhập nội dung chat và bấm enter hoặc kích button, thực hiện hàm **sendMessage,** hàm này thực hiện lấy thông tin chat, rồi thêm vào mảng thông tin chat đã có trước đó, rồi lưu xuống đĩa

sendMessage() {

this.messages.push({

user: { username: this.user.username, image: this.user.image, nickname: this.user.nickname },

text: this.message,

created: new Date().getTime()

});

this.messages.push({

user: { username: 'CUONG', image: '', nickname: 'cuongdq' },

text: 'HII: ' + this.message,

created: new Date().getTime()

});

this.room.messages = this.room.messages.concat(this.messages);

this.apiStorage.saveUserRoomMessages(this.user, this.room);

this.message = '';

this.room.messages = [];

}