

Maintenance Applicative

Objectif: Améliorer une application faite en Angular doublée d'une API Flask en python

Taches Accomplis :

- Voir uniquement les attractions portant la mention visible
- Attribuer une critique à un attraction
- Remise en forme du de la page d'accueil des attractions

Mise en œuvre :

Critique visible uniquement :

J'ai mi une condition dans ma boucle qui affiche les critique l'incitant à n'afficher que ceux dont le champ visible est à 1

```
<ng-container *ngIf="attraction.visible == true">
  <div class="attraction">
    <span>{{attraction.nom}}</span>
    <span>{{attraction.description}}</span>
    <span>{{attraction.difficulte}} / 5</span>
  </div>
```

Système de critique :

- Rajout d'une table critique contenant une clé étrangère (l'ID de l'attraction)

```
CREATE TABLE critique (
  critique_id INT AUTO_INCREMENT,
  name VARCHAR(255) DEFAULT NULL,
  prenom VARCHAR(255) DEFAULT NULL,
  texte VARCHAR(255) NOT NULL,
  note INT CHECK (note BETWEEN 0 AND 5),
  attraction_id INT NOT NULL,
  PRIMARY KEY (critique_id),
  FOREIGN KEY (attraction_id) REFERENCES attraction(attraction_id) ON DELETE CASCADE
);
```

- J'ai créé une pop up dans laquelle j'ai mis un formulaire ou j'entre les données (nom , prenom , commentaire , note) de celui qui critique

```
<h2> Welcome {{firstName}} </h2>

<div class="pcritique" [formGroup]="critiqueFormGroup">
  <mat-form-field class="champ" appearance="fill">
    <mat-label>Nom</mat-label>
    <input matInput formControlName="name">
  </mat-form-field>

  <mat-form-field class="champ" appearance="fill">
    <mat-label>Prénom</mat-label>
    <input matInput formControlName="prenom">
  </mat-form-field>

  <mat-form-field class="champ" appearance="fill">
    <mat-label>Commentaire</mat-label>
    <textarea matInput formControlName="texte"></textarea>
  </mat-form-field>

  <mat-form-field class="champ" appearance="fill">
    <mat-label>Note</mat-label>
    <input matInput formControlName="note" >
  </mat-form-field>

  <button mat-button class="champ">Annuler</button>
  <button mat-button class="champ" (click)="onSubmit(critiqueFormGroup)" >Enregistrer</button>
</div>

<ng-container >
```

- La critique est toujours associé à la bonne attraction car dans la boucle qui gère l'affichage des attractions je passe au bouton ' ajouter une critique ' l'id de l'attraction qui sera envoyer dans la requête pour effectuer le lien entre les deux tables

```

export class PopUpComponent implements OnInit {

  firstName: string;
  id: number;

  constructor(@Inject(MAT_DIALOG_DATA) public data, public attractionService: AttractionService) {
    this.firstName = data.name;
    this.id = data.id;
  }

  ngOnInit(): void {
    this.critiqueFormGroup = this.fc.group({
      name: [],
      prenom: [],
      texte: ['', Validators.required],
      note: [, Validators.required],
      attraction_id: [this.id, Validators.required]
    });
  }

  public fc = inject(FormBuilder);

  public onSubmit(critiqueFormulaire: FormGroup) {
    console.log(critiqueFormulaire)
    this.attractionService.postCritiqueAttraction(critiqueFormulaire.getRawValue()).subscribe(result => {
      critiqueFormulaire.patchValue({critique_id: result.result});
    });
  }
}

```

On filtre alors les critiques de cette manière

```

export interface CritiqueInterface {
  critique_id: number | null,
  name: string,
  prenom: string,
  texte: string,
  note: number | null ,
  attraction_id: number
}

```

```

<ng-container *ngFor="let critique of critiques | async;">
  <ng-container *ngIf="attraction.attraction_id == critique.attraction_id">
    <div class="critique">
      <span>{{critique.name}}</span><br/>
      <span>{{critique.texte}}</span><br/>
      <span>{{critique.note}} / 5</span><br/>
    </div>
  </ng-container>

```