Book Project

\*\*\*\*\*\*\*\*\*add-book.component.html\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<div class="container">

    <span><h1>{{ title }}</h1></span>

    <div class="form">

        <form [formGroup]="bookForm" (ngSubmit) = "onSubmit()">

            <p>

                <label for="bookName">Book Name</label>

                <input type="text" id="bookName" name="bookName" formControlName="bookName">

            </p>

            <p>

                <label for="bookAuthor">Author</label>

                <input type="text" id="bookAuthor" name="bookAuthor" formControlName="bookAuthor">

            </p>

            <p>

                <label for="bookPrice">Price </label>

                <input type="text" id="bookPrice" name="bookPrice" formControlName="bookPrice">

            </p>

            <p>

                <label for="bookPublisher">Publisher </label>

                <input type="text" id="bookPublisher" name="bookPublisher" formControlName="bookPublisher">

            </p>

            <p>

                <label for="bookGenre">Genre</label>

                <input type="radio" value="fiction" id="bookGenre" name="bookGenre"  formControlName="bookGenre"> Fiction

                <input type="radio" value="science" id="bookGenre" name="bookGenre"  formControlName="bookGenre"> Science

                <input type="radio" value="novel" id="bookGenre" name="bookGenre"  formControlName="bookGenre"> Novel

                <input type="radio" value="self help" id="bookGenre" name="bookGenre"  formControlName="bookGenre"> Sefl Help

                <input type="radio" value="autobiography" id="bookGenre" name="bookGenre"  formControlName="bookGenre"> Autobiography

                <input type="radio" value="technology" id="bookGenre" name="bookGenre"  formControlName="bookGenre"> Technology

            </p>

            <p>

                <button type="submit">Submit</button>

            </p>

        </form>

    </div>

</div>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* add-book.component.ts \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

export class AddBookComponent implements OnInit {

  title: string = 'Add Book';

  bookForm: FormGroup;

  book: Book = new Book();

  submitted = false;

  constructor(

    private fb: FormBuilder,

    private bookService: BookService,

    private router: Router) {

    this.bookForm = this.fb.group({

      bookName: ['', [Validators.required]],

      bookAuthor: ['',[Validators.required]],

      bookPrice: ['',[Validators.required]],

      bookGenre: ['',[Validators.required]],

      bookPublisher: ['',[Validators.required]]

      });

   }

  ngOnInit(): void {

  }

  onSubmit(){

    this.book = new Book();

    this.book.bookName = this.form.bookName.value;

    this.book.bookAuthor = this.form.bookAuthor.value;

    this.book.bookPublisher = this.form.bookPublisher.value;

    this.book.bookGenre = this.form.bookGenre.value;

    this.book.bookPrice = this.form.bookPrice.value;

    this.submitted = true;

    this.saveBook();

  }

  saveBook(){

    this.bookService.addBook(this.book)

      .subscribe(data => {

        console.log(data);

        this.goToBookList();

      },

       error => console.log(error));

    // this.book = new Book();

  }

  get form(){

    return this.bookForm.controls

  }

  goToBookList(){

    this.router.navigate(['/BookList']);

  }

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\* book-list.component.html \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<div class = "row">

    <h2> Book List</h2>

</div>

<table class = "table table-striped table-bordered">

    <thead>

        <tr>

            <th> Book Name</th>

            <th> Author </th>

            <th> Publisher </th>

            <th> Genre </th>

            <th> Price </th>

            <th> Actions </th>

        </tr>

    </thead>

    <tbody>

        <tr \*ngFor = "let book of books" >

            <td> {{ book.bookName }} </td>

            <td> {{ book.bookAuthor }} </td>

            <td> {{ book.bookPublisher }} </td>

            <td> {{ book.bookGenre }} </td>

            <td> {{ book.bookPrice }} </td>

            <td>

                <button (click) = "updateBook(book.bookId)" class = "btn btn-primary"> Update</button>

                <button (click) = "deleteBook(book.bookId)" class = "btn btn-danger" style="margin-left: 10px"> Delete</button>

                <button (click) = "bookDetail(book.bookId)" class = "btn btn-primary" style="margin-left: 10px"> View</button>

            </td>

        </tr>

    </tbody>

</table>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* book-list.component.ts \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

export class BookListComponent implements OnInit {

  books!: Book[];

  constructor(private bookService: BookService,

    private router: Router) { }

  ngOnInit(): void {

    console.log("inside book list");

    this.getBooks();

  }

  private getBooks() {

    this.bookService.getBookList().subscribe(data =>{

      this.books = data;

    })

  }

  bookDetail(bookId: number){

    this.router.navigate(['BookDetail', bookId]);

  }

  updateBook(bookId: number){

    this.router.navigate(['BookUpdate', bookId]);

  }

  deleteBook(bookId: number){

    this.bookService.deleteBook(bookId).subscribe(data => {

      console.log(data);

      this.getBooks();

    })

  }

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*book-details.component.html \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<h2>Book Details</h2>

<hr/>

<div \*ngIf="book">

  <div>

    <label><b>Book Name: </b></label> {{book.bookName}}

  </div>

  <div>

    <label><b>Author: </b></label> {{book.bookAuthor}}

  </div>

  <div>

    <label><b>Publisher: </b></label> {{book.bookPublisher}}

  </div>

  <div>

    <label><b>Genre: </b></label> {{book.bookGenre}}

  </div>

  <div>

    <label><b>Price: </b></label> {{book.bookPrice}}

  </div>

</div>

<br>

<br>

<button (click)="list()" class="btn btn-primary">Back to Book List</button><br>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*book-details.component.ts \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

export class BookDetailsComponent implements OnInit {

  bookId!: number;

  book!: Book;

  constructor(

    private route: ActivatedRoute,

    private router: Router,

    private bookService: BookService

  ) { }

  ngOnInit(): void {

    this.book = new Book();

    this.bookId = this.route.snapshot.params['bookId'];

    this.bookService.getBookById(this.bookId)

      .subscribe(data =>{

        console.log(data)

        this.book = data;

      }, err => console.log(err));

  }

  list(){

    this.router.navigate(['BookList']);

  }

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* book-update.component.html \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<h3>Update Book</h3>

<div style="width: 400px;">

  <form (ngSubmit)="onSubmit()">

    <div class="form-group">

      <label for="name">Book Name</label>

      <input type="text" class="form-control" id="bookName" required [(ngModel)]="book.bookName" name="bookName">

    </div>

    <div class="form-group">

      <label for="name">Author</label>

      <input type="text" class="form-control" id="bookAuthor" required [(ngModel)]="book.bookAuthor" name="bookAuthor">

    </div>

    <div class="form-group">

      <label for="name">Publisher</label>

      <input type="text" class="form-control" id="bookPublisher" required [(ngModel)]="book.bookPublisher" name="bookPublisher">

    </div>

    <div class="form-group">

        <label for="name">Genre</label><br>

        <!-- <input type="text" class="form-control" id="bookPublisher" required [(ngModel)]="book.bookPublisher" name="bookPublisher"> -->

        <input type="radio" value="fiction" id="bookGenre" [(ngModel)]="book.bookGenre" name="bookGenre"> Fiction

        <input type="radio" value="science" id="bookGenre" [(ngModel)]="book.bookGenre" name="bookGenre"> Science

        <input type="radio" value="novel" id="bookGenre" [(ngModel)]="book.bookGenre" name="bookGenre"> Novel

        <input type="radio" value="self help" id="bookGenre" [(ngModel)]="book.bookGenre" name="bookGenre"> Sefl Help

        <input type="radio" value="autobiography" id="bookGenre" [(ngModel)]="book.bookGenre" name="bookGenre"> Autobiography

        <input type="radio" value="technology" id="bookGenre" [(ngModel)]="book.bookGenre" name="bookGenre"> Technology

      </div>

      <div class="form-group">

        <label for="name">Price</label>

        <input type="text" class="form-control" id="bookPrice" required [(ngModel)]="book.bookPrice" name="bookPrice">

      </div>

    <button type="submit" class="btn btn-success">Submit</button>

  </form>

</div>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*book-update.component.ts \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

export class BookUpdateComponent implements OnInit {

  bookId!: number;

  book!: Book;

  constructor(

    private route: ActivatedRoute,

    private router: Router,

    private bookService: BookService

  ) { }

  ngOnInit(): void {

    this.book = new Book();

    this.bookId = this.route.snapshot.params['bookId'];

    this.bookService.getBookById(this.bookId)

      .subscribe(data =>{

        console.log(data)

        this.book = data;

      }, err => console.log(err)

      );

  }

  updateBook(){

    this.bookService.updateBook(this.bookId, this.book)

    .subscribe(data =>{

      console.log(data);

      this.book = new Book();

      this.goToBookList();

    }, err => console.log(err));

  }

  onSubmit(){

    this.updateBook();

  }

  goToBookList(){

    this.router.navigate(['/BookList']);

  }

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* book.ts. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

export class Book {

    bookId!: number;

    bookName!: string;

    bookAuthor!: string;

    bookPublisher!: string;

    bookPrice!: number;

    bookGenre!: string;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* book.service.ts \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

export class BookService {

  private baseURL = "http://localhost:8080/books/";

  constructor(private httpClient: HttpClient) { }

  getBookList(): Observable<Book[]>{

    return this.httpClient.get<Book[]>(`${this.baseURL}` +'getall');

  }

  getBookById(bookId: number): Observable<Book>{

    return this.httpClient.get<Book>(`${this.baseURL}`+'get/'+`${bookId}`);

  }

  addBook(book: Book): Observable<Object>{

    return this.httpClient.post(`${this.baseURL}`+'save', book);

  }

  updateBook(bookId: number, book: Book): Observable<Object>{

    return this.httpClient.put(`${this.baseURL}`+'update/'+`${bookId}`, book);

  }

  deleteBook(bookId: number): Observable<Object>{

    return this.httpClient.delete(`${this.baseURL}`+'delete/'+`${bookId}`);

  }

}

\*\*\*\*\*\*\*\*\*\*\*\*\* app.component.html \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<nav class="navbar navbar-expand-sm bg-primary navbar-dark">

  <!-- Links -->

  <ul class="navbar-nav">

    <li class="nav-item">

        <a routerLink="BookList" class="nav-link" routerLinkActive="active">Book List</a>

    </li>

    <li class="nav-item">

        <a routerLink="/AddBook" class="nav-link" routerLinkActive="active">Add Book</a>

    </li>

  </ul>

</nav>

<div class="container">

  <br>

  <h2 style="text-align: center;">{{title}}</h2>

  <hr>

  <div class="card">

    <div class="card-body">

  <router-outlet></router-outlet>

    </div>

  </div>

</div>

<footer class="footer">

  <div class="container">

      <span>All Rights Reserved 2022 @STG Infotech LLP</span>

  </div>

</footer>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* app.component.ts \*\*\*\*\*\*\*\*\*\*\*\*\*

export class AppComponent {

  title = 'Book Project';

}

\*\*\*\*\*\*\*\*\*\*\*\*\* style.css \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

@import '~bootstrap/dist/css/bootstrap.min.css';

.footer {

    position: absolute;

    bottom: 0;

    width:100%;

    height: 70px;

    background-color: blue;

    text-align: center;

    color: white;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* app.routing.module \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

const routes: Routes = [

  { path: '', redirectTo: 'BookList', pathMatch: 'full' },

  { path: 'BookList', component: BookListComponent },

  { path: 'AddBook', component: AddBookComponent },

  { path: 'BookUpdate/:bookId', component: BookUpdateComponent },

  { path: 'BookDetail/:bookId', component: BookDetailsComponent },

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* app.module.ts \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

imports: [

    BrowserModule,

    AppRoutingModule,

    ReactiveFormsModule,

    FormsModule,

    HttpClientModule

  ],