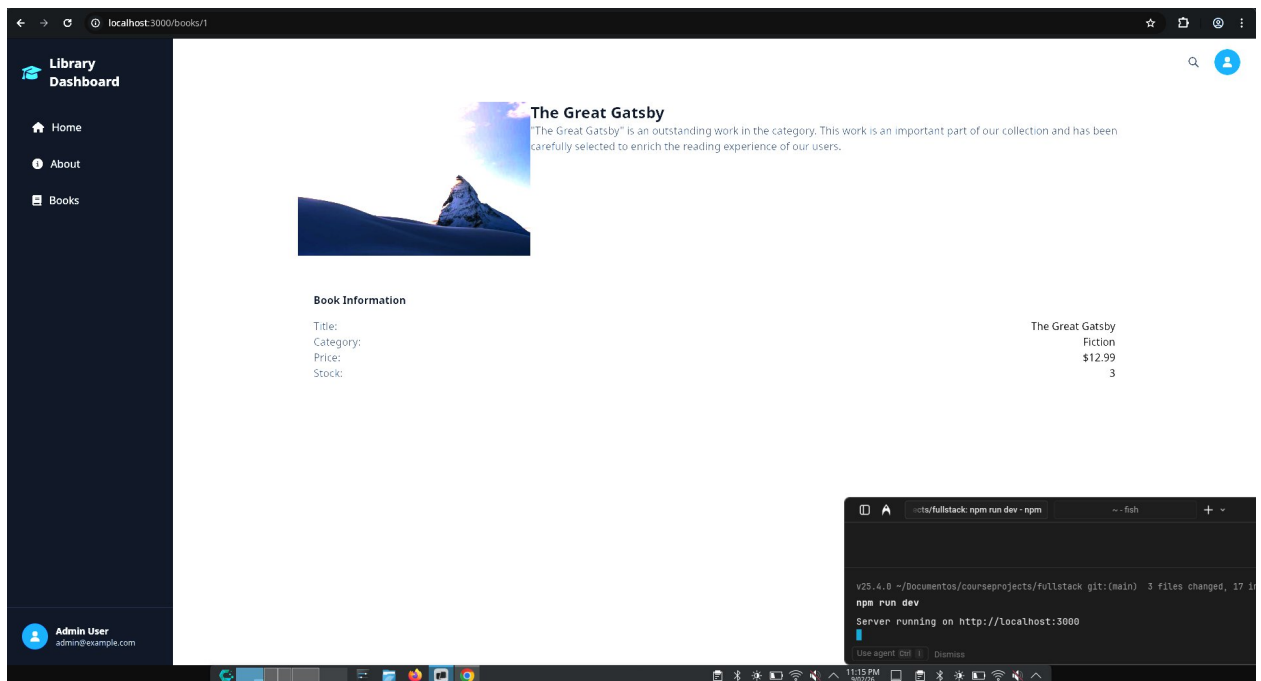
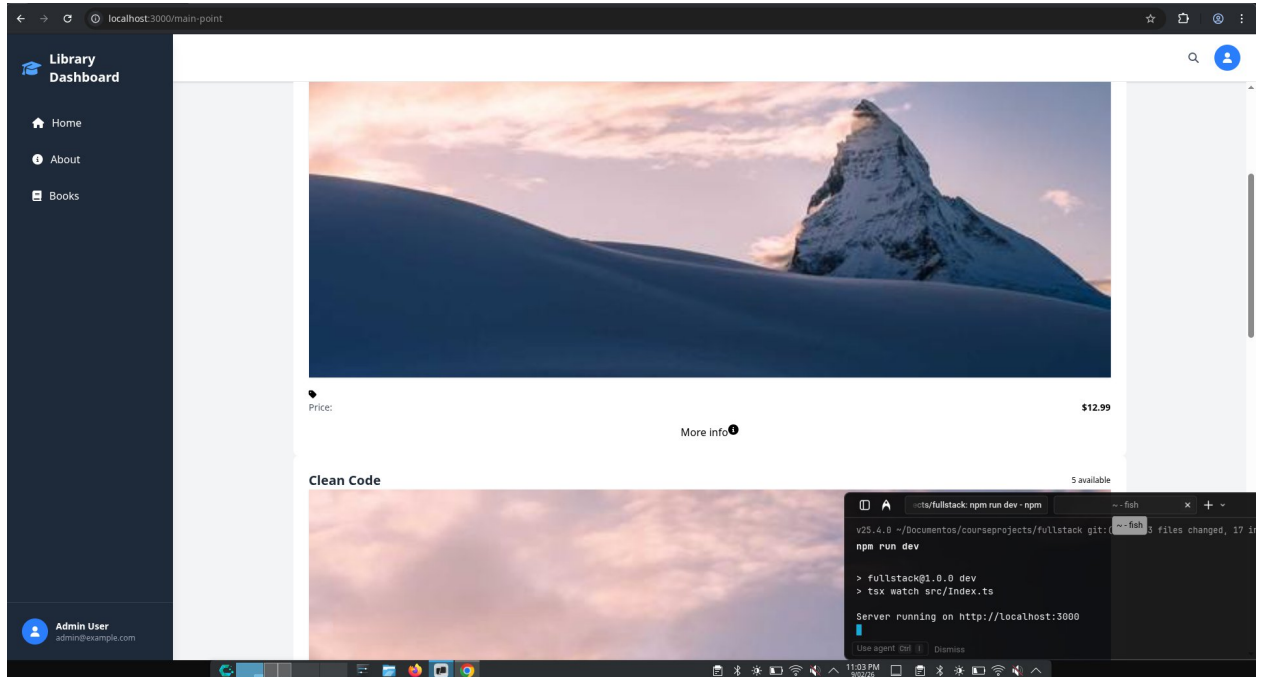


## Entrega tutorial 02

Juan José Jara Calle

Enlace GitHub: <https://github.com/JaraCalle/courseprojects>

Pantallazo de la última sección implementada corriendo local:



## Tarea

1. books.ejs el archivo utiliza `<%= book.category %>` Pero eso no va a traer nunca la categoria, ya que category es con C mayúscula
2. HomeController.ts utiliza innecesariamente el any: `static Main_Point(req: Request, res: Any)` Ahí puede ir perfectamente un Response

3. HomeController.ts hace uso de PascalCase con under score, pero en los controlares hemos definido el estandar de usar camelCase static Main\_Point(req: Request, res: Response)
4. HomeController.ts no dice información sobre aquello que va a retornar como respuesta, en este caso deberia ser :void. static mainPoint(req: Request, res: Response) {
5. HomeController.ts el viewData de mainPoint no define sus tipos correctamente y solo se establece con any, lo cual es ambiguo const viewData: any = {}; la solución seria const viewData: { [key: string]: any } = {};
6. HomeController.ts la función show también usa de forma inapropiada el any, en lugar de usar correctamente el Request, Response static show(req: any, res: any)
7. HomeController.ts en la función show, el book está mal planteado ya que está usando parseInt, pero al modificar el req por Request, como dije en el punto anterior esa función tira error, const book = Book.findById(books, parseInt(req.params.id)); por eso lo mejor es cambiarla por la funcion Number
8. show.ejs Nunca se valida si el libro realmente existe, lo cual implica que si no se pasa ningún libro. Se mostrará undefined en una vista con errores.
9. books.ejs Tampoco hace la validación de que exista el libro o la lista de libros antes de hacer todo el display. Lo cual puede genera un crash completo para el usuario final
10. app.ejs el viewData es opcional y está usando una estructura de poco clara, lo cual genera problemas en el diseño del layout
11. HomeController.ts llamamos mainPoint a una funcion, pero ese nombre no ofrece claridad a lo que hace. Sería mejor ponerle un nombre cómo lo podría ser books
12. HomeController.ts la función show, nuca valida que el book exista o devuelva algo diferente a null. Por lo cual si es null el libro la vista será un gran mensaje de error

### Propuesta de mejora

#### HomeController.ts

```
static books(req: Request, res: Response): void {  
  const viewData = {  
    title: 'Libros',  
    books: books.map(book => ({  
      ...book,  
      available: book.stock > 0,  
      formattedPrice: book.price.toLocaleString()  
    })))  
  };  
  res.render('home/books', viewData);  
}
```

```
static show(req: Request, res: Response): void {
```

```
const id = Number(req.params.id);
const book = Book.findById(books, id);

if (!book) {
  res.status(404).render('errors/404', {
    title: 'Libro no encontrado'
  });
  return;
}

const viewData = {
  title: book.title,
  book: {
    ...book,
    available: book.stock > 0,
    formattedPrice: book.price.toLocaleString()
  }
};

res.render('home/show', viewData);
}
```