**Museum Test**

**Backend:**

* Technologies: .NET 6, Entity Framework Core, SQL Server 2019.
* Architecture: N-layer architecture -> Onion Architecture.
* Layers:

1. **MuseumTest.Domain**: It is the center part of the architecture. It holds all application domain objects. This layer holds POCO classes (Code First) with entities. These domain entities don't have any dependencies.
2. **MuseumTest.Repository**: The layer is intended to create an abstraction layer between the Domain entities layer and the Business Logic layer of the application. It is a data access pattern that prompts a more loosely coupled approach to data access. We create a generic repository, which queries the data source for the data, maps the data from the data source to a business entity, and persists changes in the business entity to the data source.
3. **MuseumTest.Services**: The layer holds interfaces which are used to communicate between the UI layer and repository layer. It holds business logic for an entity so it’s called the business logic layer as well.
4. **MuseumTest:** It’s the most external layer. It is a Web API. This layer has an implementation of the Dependency Inversion Principle so that the application builds a loosely coupled application. It communicates to the internal layer via interfaces and communicates to the client with controllers.

**Frontend:**

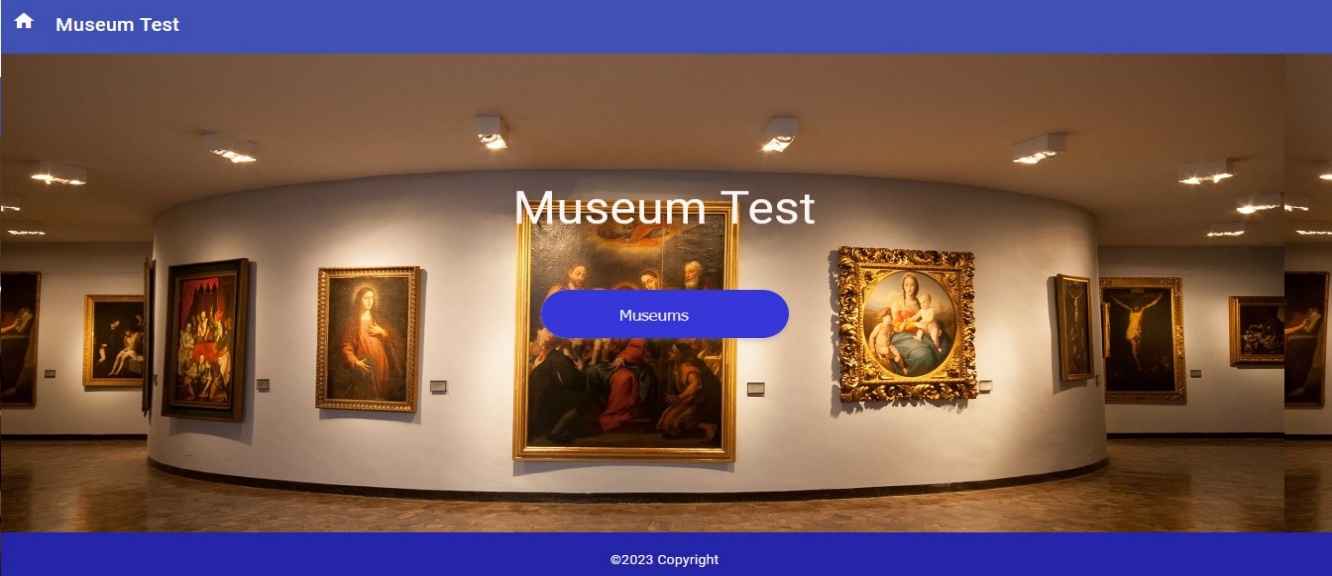
* Technologies: Angular version 15.0.4.
* The application includes components-based scalable framework for building the UI.
* The services contain methods that maintain data throughout the life of the application, data is available all the time. They contain also functions to connect with the backend.
* The models are using for modeling the data used in the view and handling user interactions such as clicking on buttons, scrolling, or causing other changes in the view.

**User Manual**

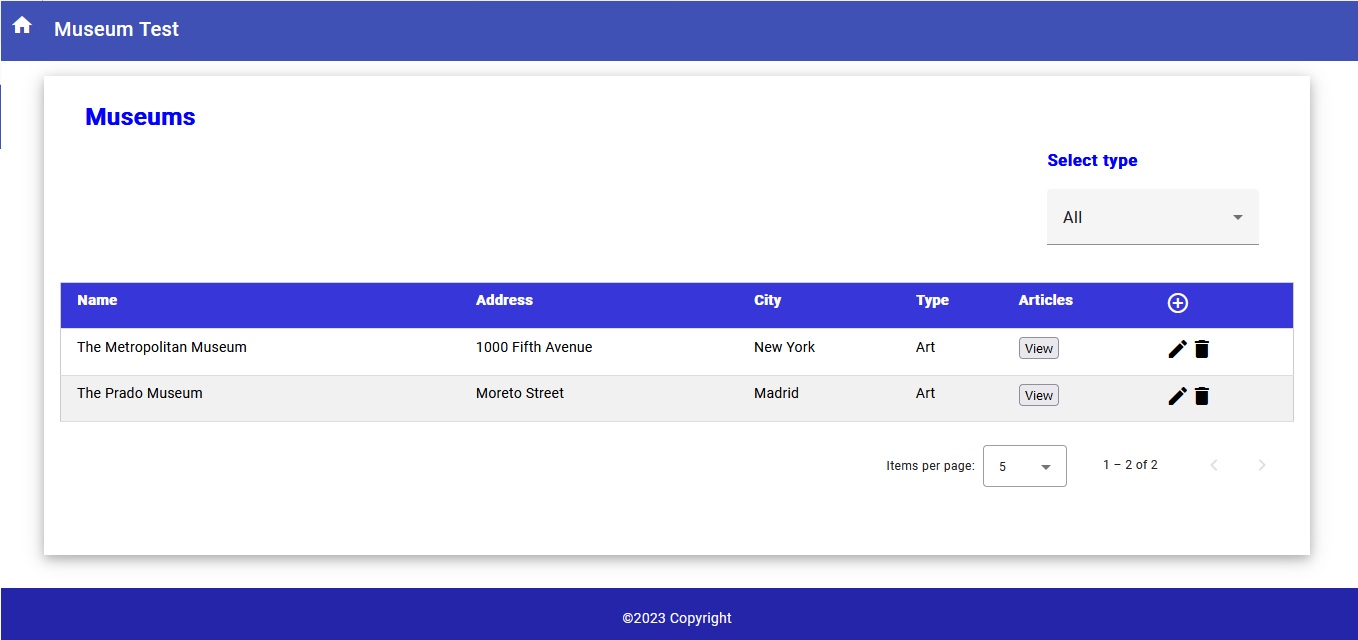
* **Execution:**

1. Open Visual Studio 2022 in MuseumTest route.
2. Build the solution.
3. In *package manager console* execute the command *add-migration <mig-name>.*
4. In *package manager console* execute the command *update-database.*
5. Ctrl+F5.
6. In MuseumTestUI open a console and run the command *npm install.*
7. In the console run the command *ng-serve –o.*

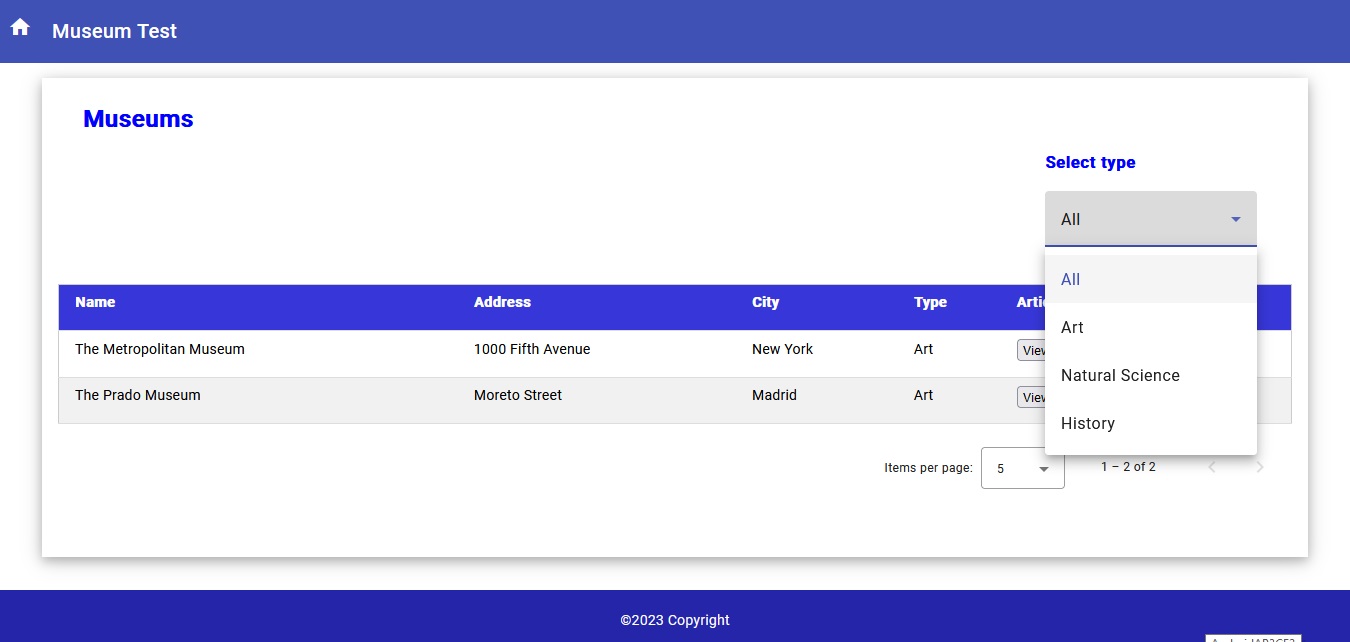
* Work with application.
* Presentation.



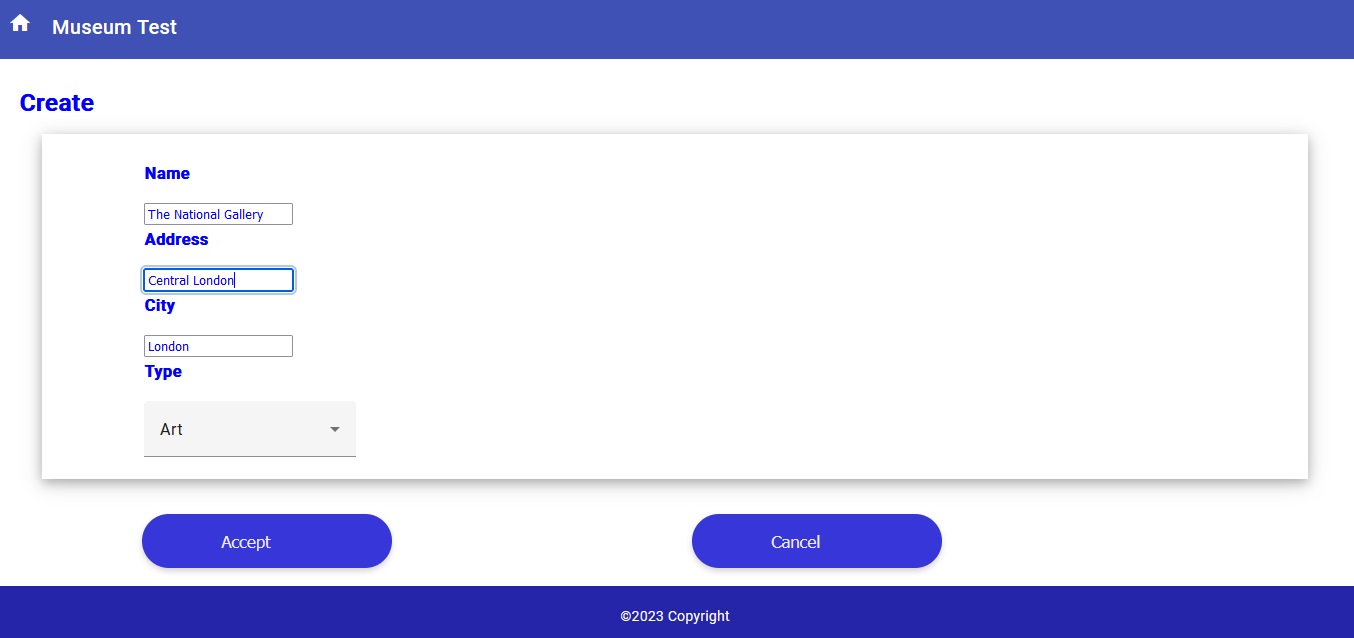
* List of museums (pagination).

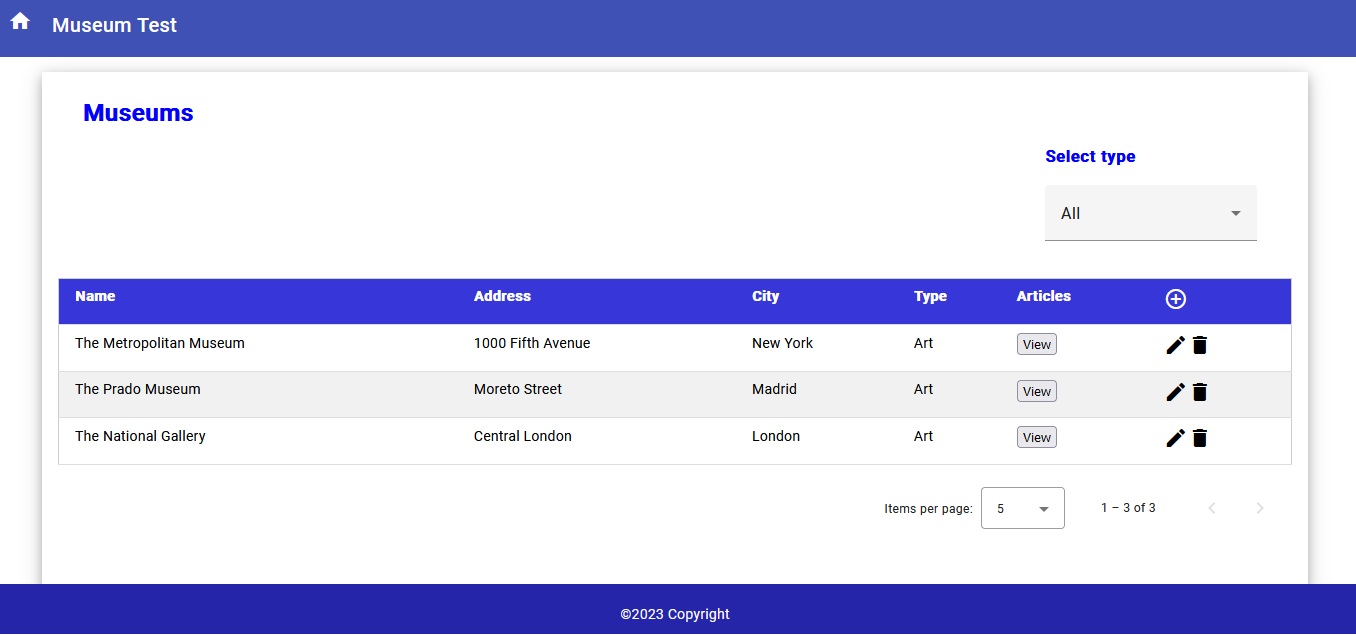


* Select museums by theme.

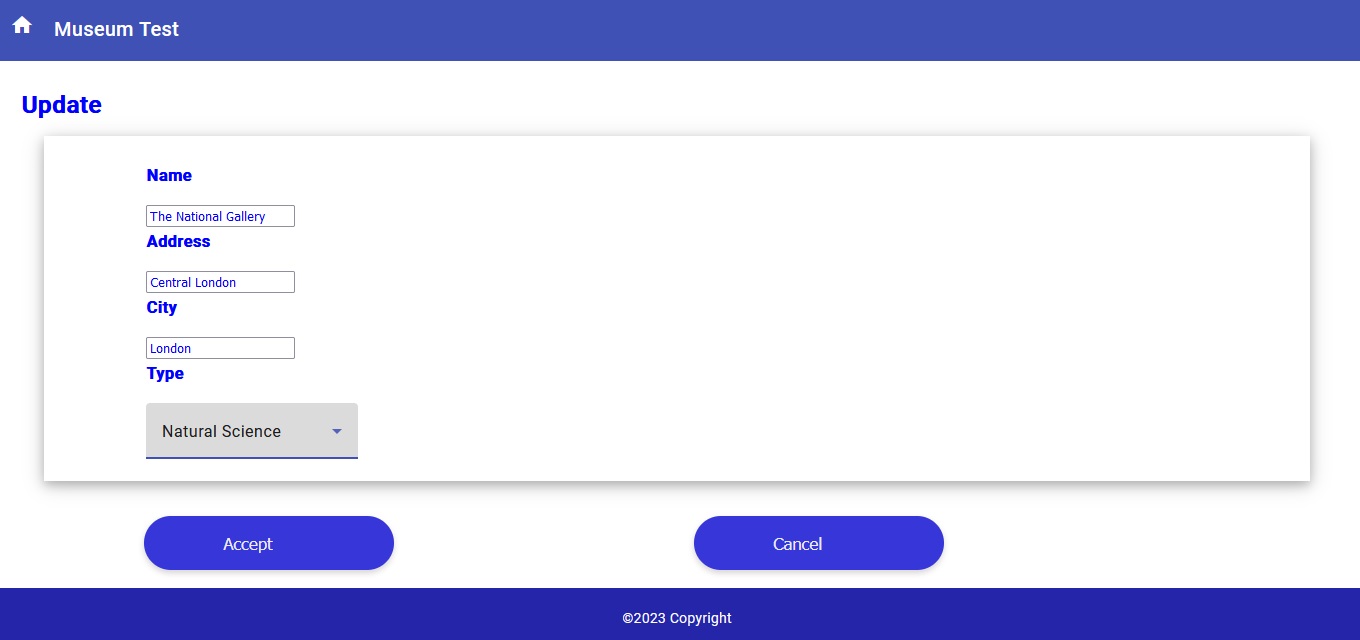


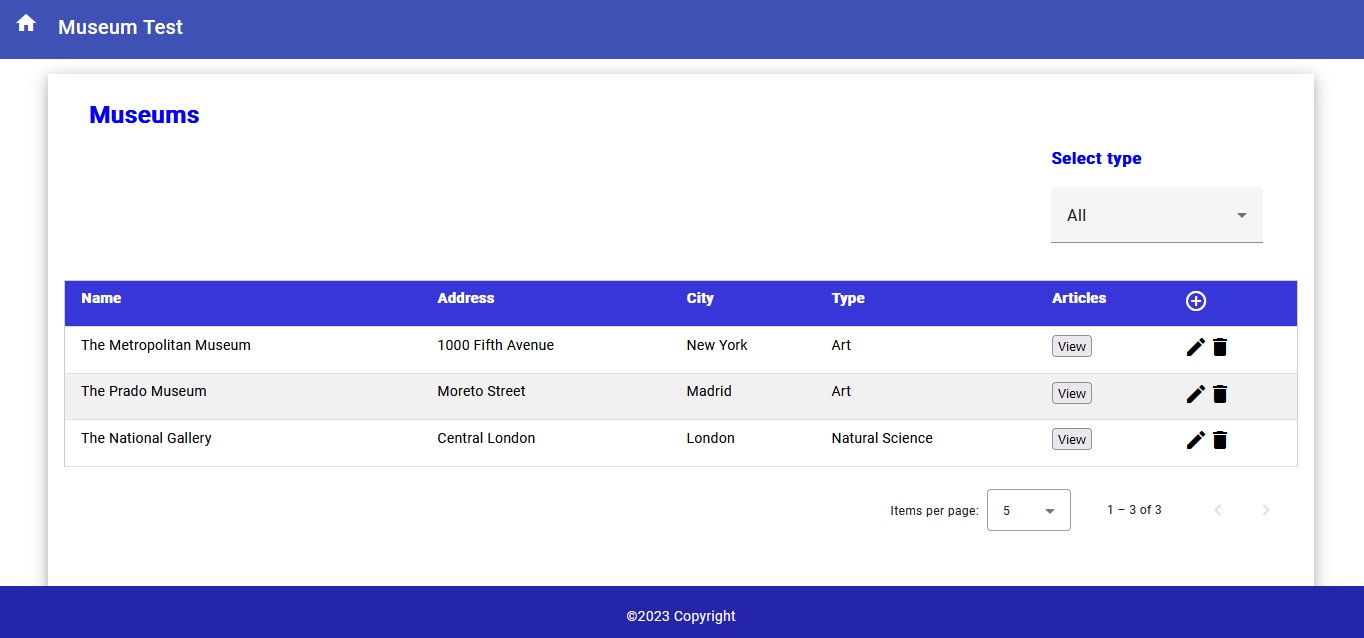
* Create a museum.



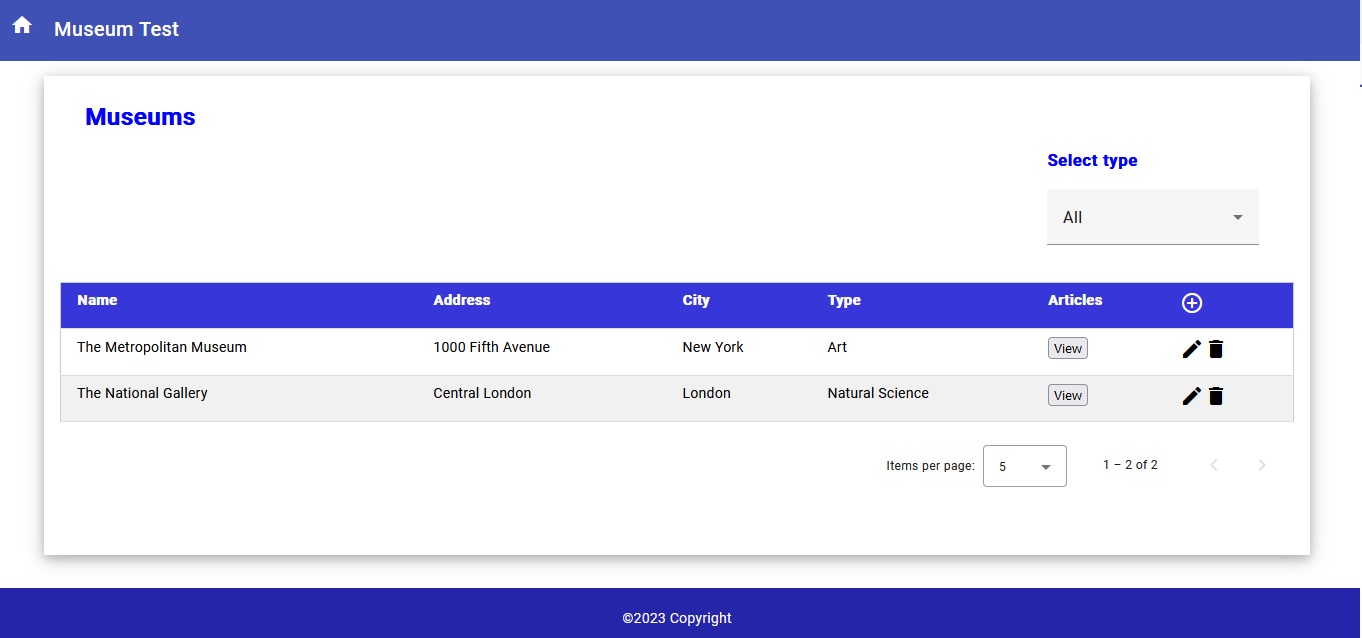


* Edit a museum.

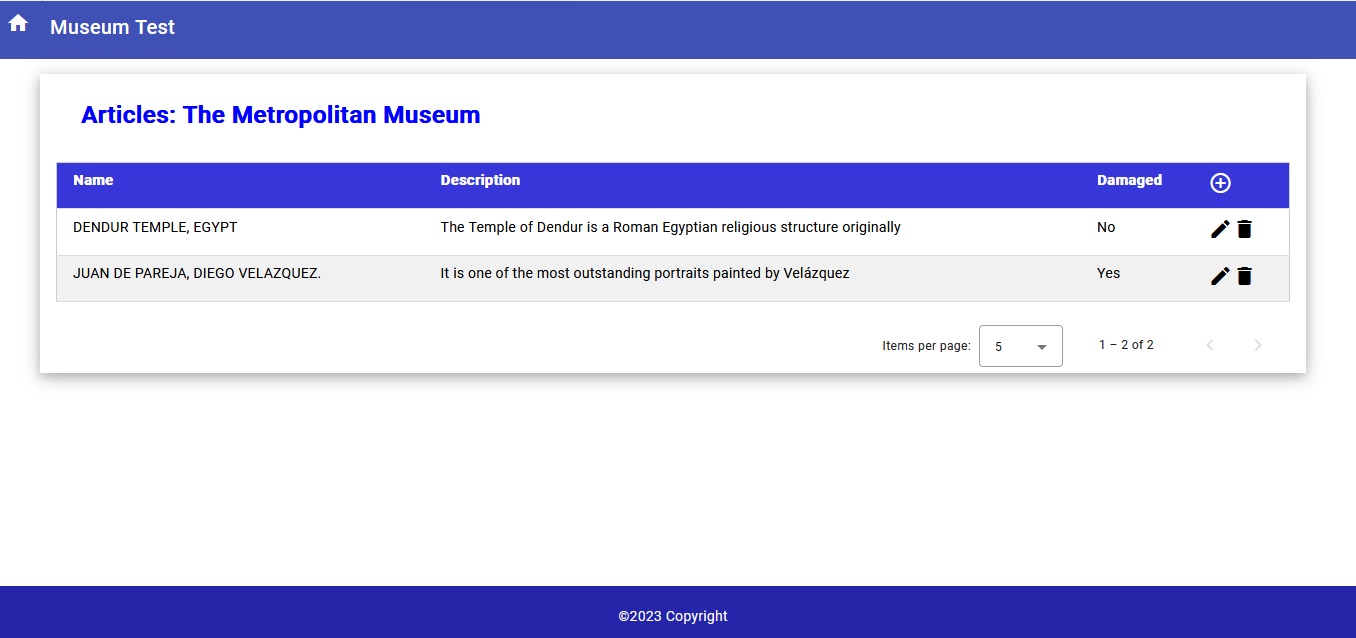




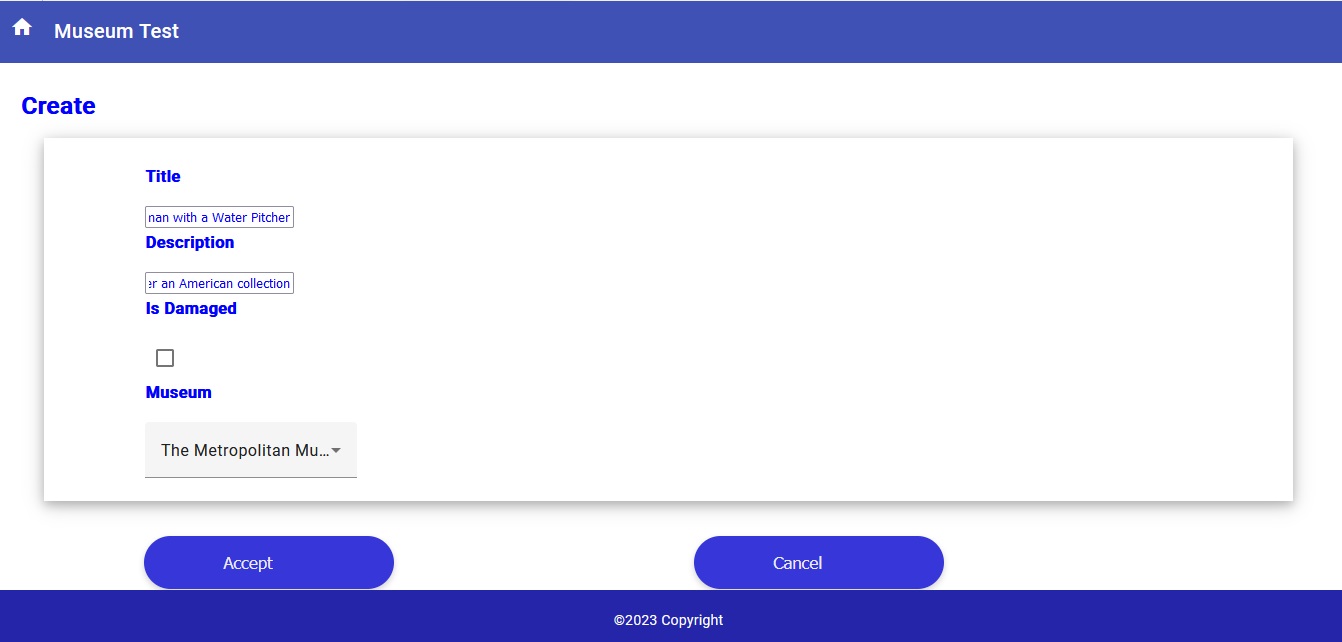
* Delete a museum.

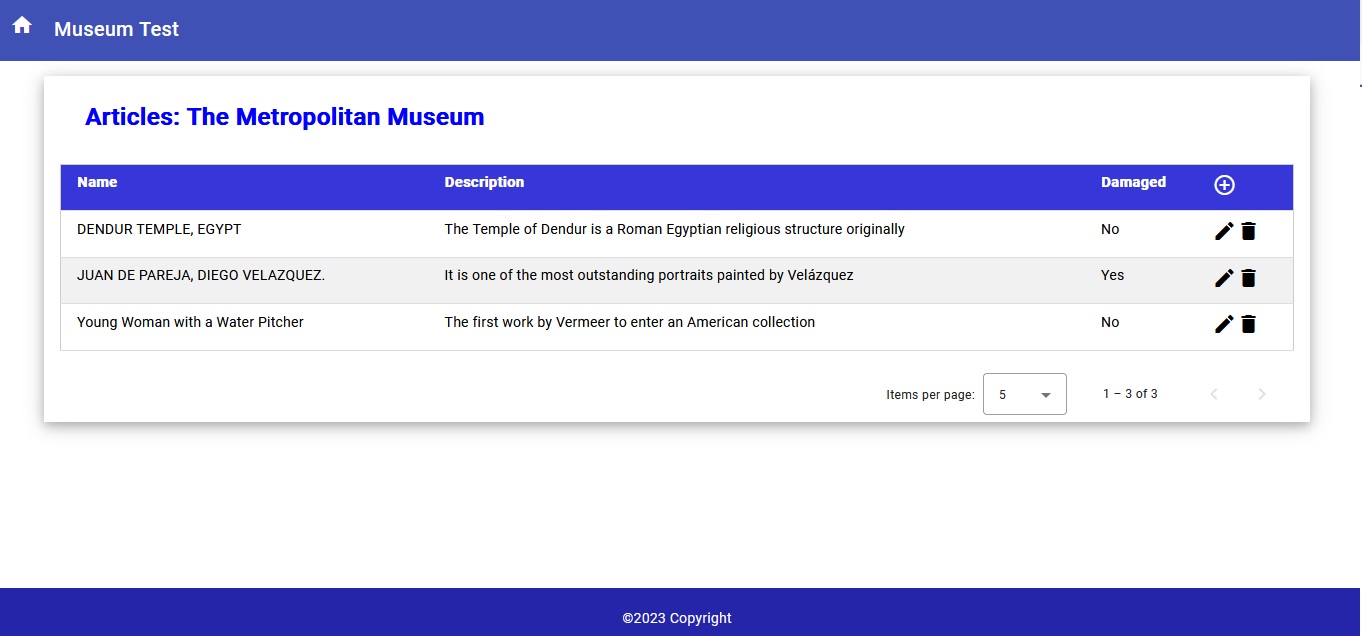


* List article from the museum(pagination).

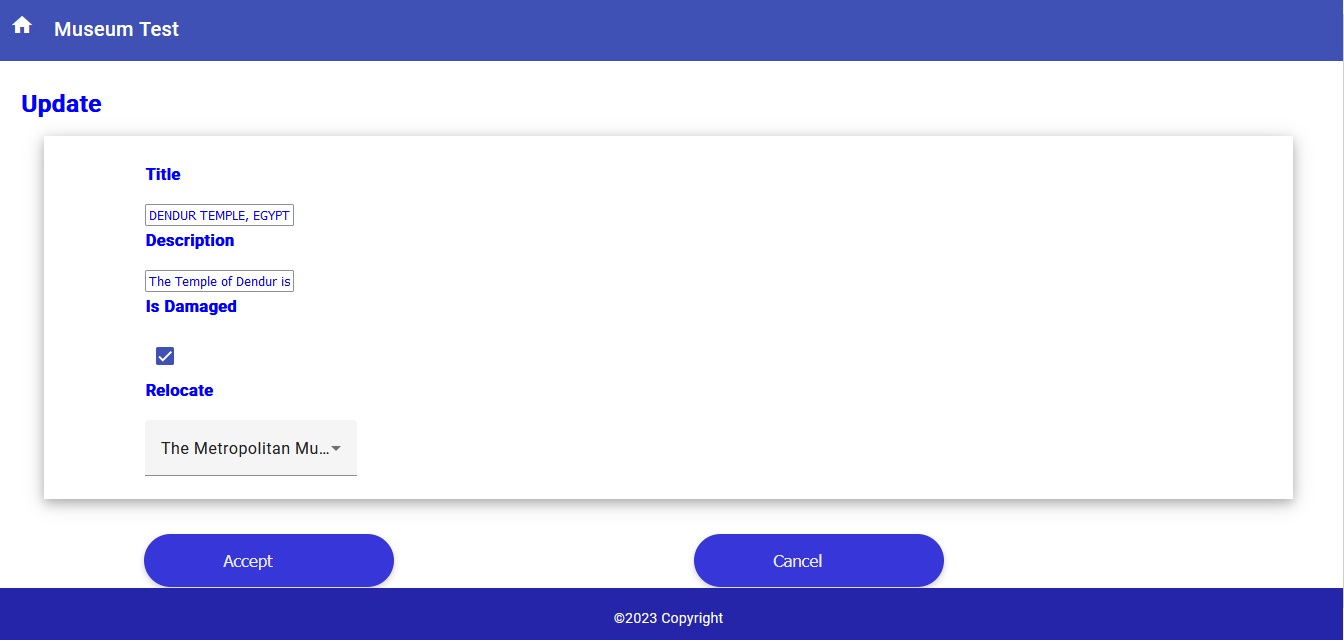


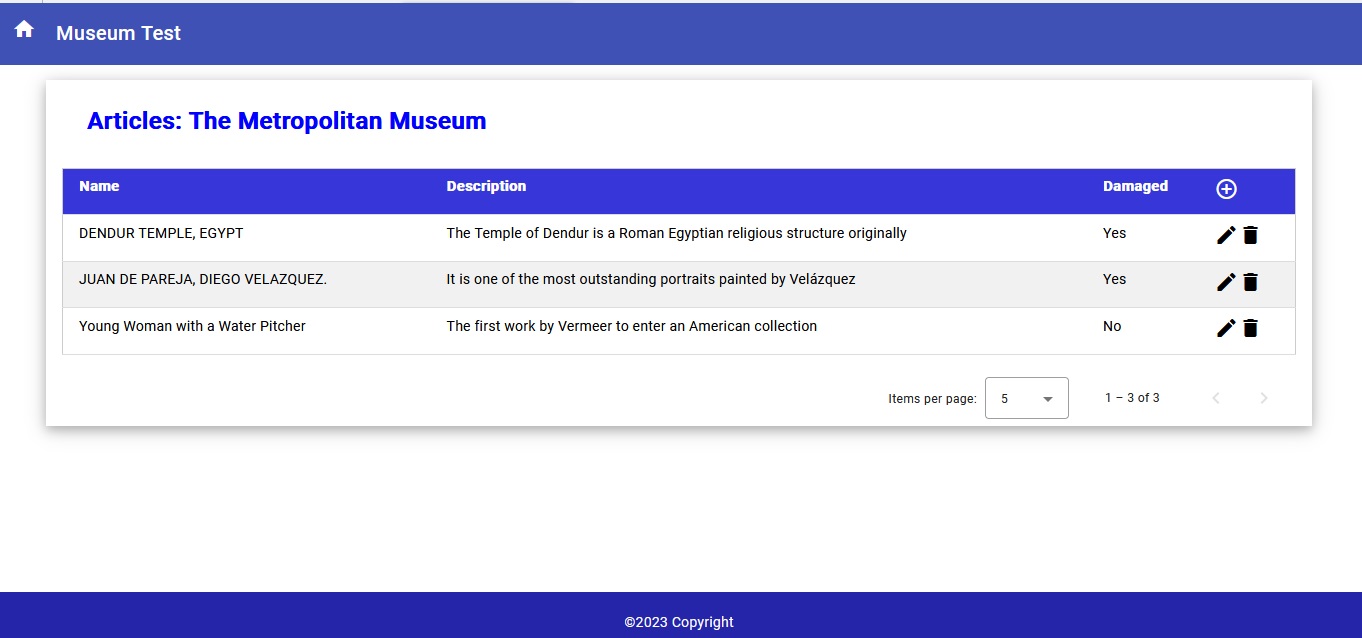
* Create Article.





* Edit Article (Relocate, change description, damaged status and other properties).





* Delete Article.

