# Exercise: Databases – EF Core

# IRunes

Problems for exercises and homework for the [“C# Web Basics” course @ SoftUni](https://softuni.bg/courses/csharp-web-development-basics). Yoy can submit your solution in the course web page.

You have been tasked to implement a simple application, using the Web Server. The application imitates a **store** for **Music Albums** and **Music Tracks**. You will see the functionality – described below.

## Database Requirements

The first thing you need to do is implement the Database entities. Use Entity Framework Core, and implement the following entities:

### User

* Id – a **string** (**GuID**).
* Username – a **string**.
* Password – a **string** (**encoded** in the database).
* Email – a **string**.

### Album

* Id – a **string** (**GuID**).
* Name – a **string**.
* Cover – a **string** (a **link** to an **image**).
* Price – a **decimal** (**sum** of all Tracks’ **prices**, **reduced** by **13%**).
* Tracks – a **collection** of Tracks.

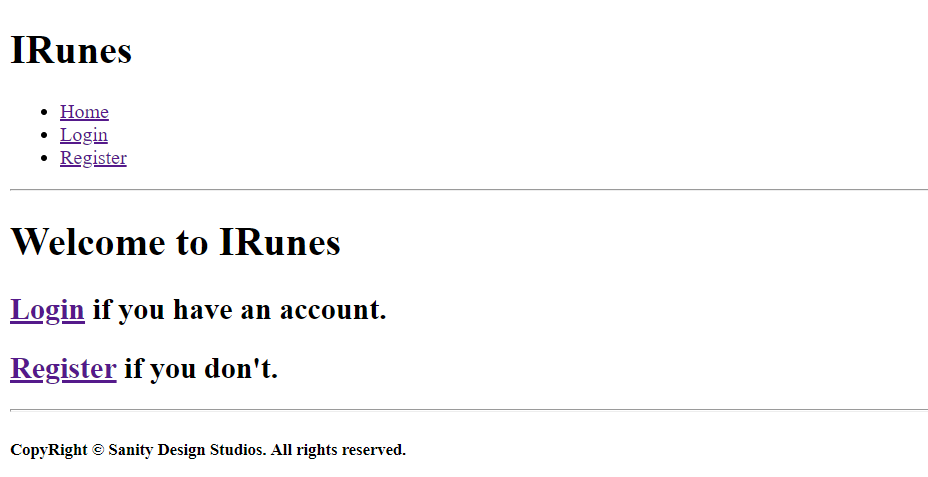
### Track

* Id – a **string** (**GuID**).
* Name – a **string**.
* Link – a **string** (a **link** to a **video**).
* Price – a **decimal**.

## Template Requirements

Here you will find your template requirements. This is a simple application, but it will work with HTML pages, so get ready to write a lot of HTML. 😉 Each HTML Page comes with a Route on which you should return it.

### Index (guest, logged-out) (route=”/Home/Index”, route=”/”)



[**Login**]links **relocate** you to “/Users/Login”.

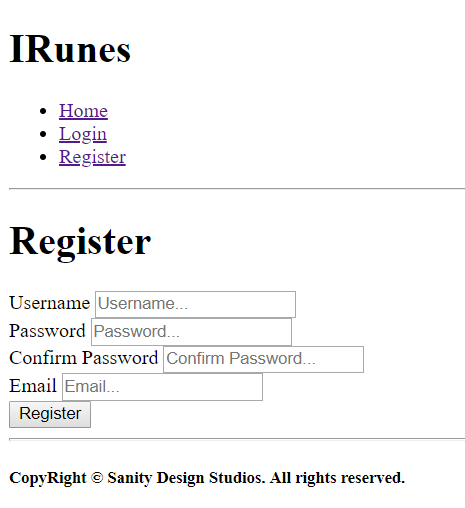
[**Register**]links **relocate** you to “/Users/Register”.

### Login (guest, logged-out) (route=”/Users/Login”)



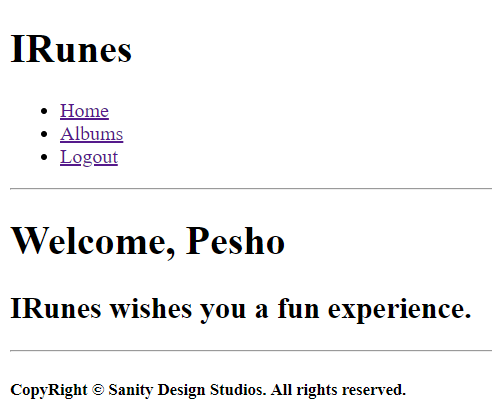
The [**Login**] button **submits** the **form** witha **POST** **request** to “/Users/Login”.

### Register (guest, logged-out) (route=”/Users/Register”)

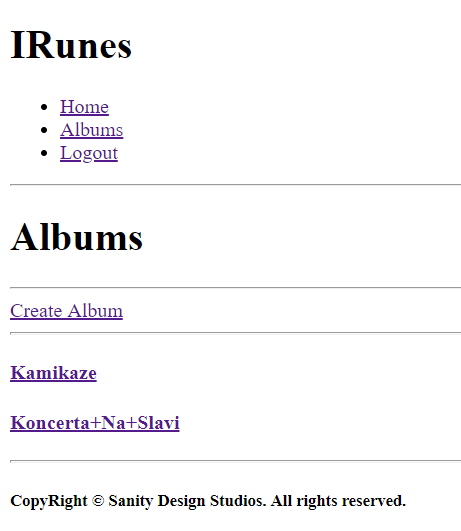


The [**Register**] button **submits** the **form** witha **POST** **request** to “/Users/Register”.

### Index (user, logged-in) (route=”/Home/Index”, route=”/”)

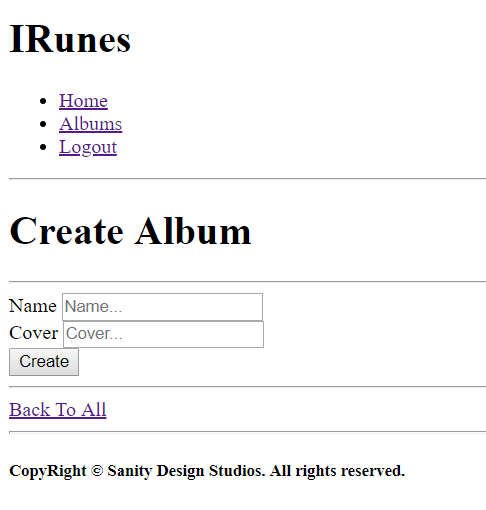


### All Albums (user, logged-in) (route=”/Albums/All”)



[**Create** **Album**] links **relocate** you to “/Albums/Create”.

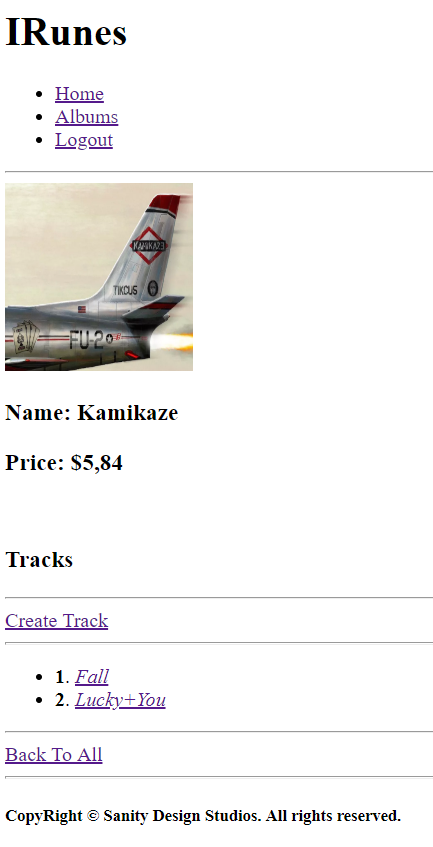
### Album Create (user, logged-in) (route=”/Albums/Create”)



The [**Create**] button **submits** the **form** witha **POST** **request** to “/Albums/Create”.

[**Back To All**] links **relocate** you to “/Albums/All”.

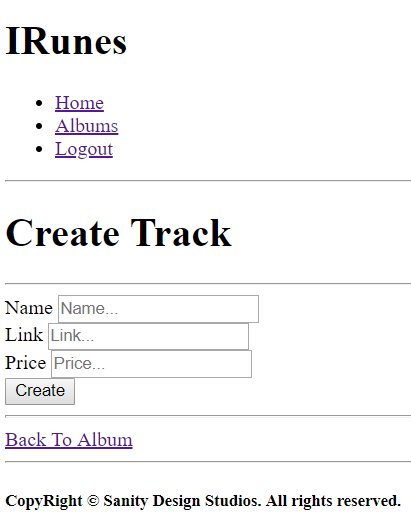
### Album Details (user, logged-in) (route=”/Albums/Details?id={albumId}”)



[**Create** **Track**] links **relocate** you to “/Tracks/Create?albumId={albumId}”.

[**Back To All**] links **relocate** you to “/Albums/All”.

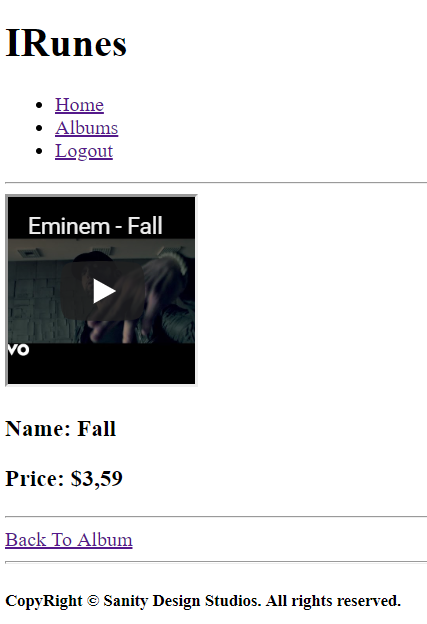
### Track Create (user, logged-in) (route=”/Tracks/Create?albumId={albumId}”)



The [**Create**] button **submits** the **form** witha **POST** **request** to “/Tracks/Create?albumId={albumId}”.

[**Back To Album**] links **relocate** you to “/Album/Details?id={albumId}”. (the Album from which this page was accessed).

### Track Details (user, logged-in) (route=”/Tracks/Details?albumId={albumId}&trackId={trackId}”)



[**Back To Album**] links **relocate** you to “/Album/Details?id={albumId}”. (the Album from which this page was accessed).

## Functional Requirements

The functional requirements are quite simple.

### Users

The application should provide guests (logged-out) with the **functionality** to access:

* The Guest Index Page
* The Login Page and Functionality
* The Register Page and Functionality

The application should provide users (logged-in) with the **functionality** to access:

* The User Index Page
* The All Albums Page and Functionality
* The Album Create Page and Functionality
* The Album Details Page and Functionality
* The Track Create Page and Functionality
* The Track Details Page and Functionality

### Albums

The Albums are **created** and **presented** on the All Albums Page, in a **listed format** with only their names as elements. Each **album** **name** should be a **link** which **leads** to the corresponding Album’s Details Page.

If there **are no Albums currently** in the Database, a message “There are currently no albums.” should be printed.

On the Album’s Details Page, its tracks should be **listed**, in an **indexed list**, **starting** from **1**. The **order** of **data** is **not mandatory**.

Each **track name** should be a link which leads to the corresponding Track’s Details Page.

### Tracks

The Tracks are **created** and **presented** on their Album’s Details Page. Tracks are created using the Album’s id which is passed through the **query parameters**.

When you **create** a Track, you can pass it the **iframe-ready url**, in order to make your work easier.

## Notes

**Note**: **No data validation** is **required**. If an **invalid** **form** is **sent**, just **redirect back** to the form.

**Note**: If the **2 passwords** of the **Registration** of **Users** do **NOT match**, just **redirect back** to the form.