# SoftUni Tutorials - Workshop - Part 2

This application is created as a workshop for the [VueJS Course](https://softuni.bg/trainings/2820/vuejs-march-2020) @ [Software University](https://softuni.bg)

# Task Requirements

So far, we have created **one** **dynamic** **page**. The next step is to split this dynamic into **multiple** **pages** and **route** **between** **them**. You will be provided with **REST API** that provide **user service** (login, register and logout) and more. You will find more information below.

## Guest Users

Every **not logged in** user (guest user) **should have access to** this pages:

* Each technology
  + C#, JAVASCRIPT, JAVA and PYTHON
* And each technology subjects / tutorials
* User page (Login and Register)

### Technologies



Each technology should be the first part of the **url**. After that the currently selected **subject name** / **tutorial name** should follow. Here's some examples:

* ***/javascript/object-composition***
* ***/javascript/asynchronous-programming***
* ***/java/generics***
* ***/java/stacks-and-queues***
* ***/csharp/stacks-and-queues***
* ***/csharp/multidimensional-arrays***
* ***/python/comprehension***
* ***/python/typles-and-sets***
* ***and so on…***

*Use the provided "****GET Access****" from API to:*

* [**http://localhost:9999/api/tutorials**](http://localhost:9999/api/tutorials) **- get all tutorials**
* [**http://localhost:9999/api/tutorials?technology={technology}**](http://localhost:9999/api/tutorials?technology=%7btechnology%7d)- **get all tutorials by given technology**
* [**http://localhost:9999/api/tutorials/{id}**](http://localhost:9999/api/tutorials/%7bid%7d)- **get single tutorial by id**

When some tutorial is selected, you should **display** the current content on that tutorial in that specific technology.

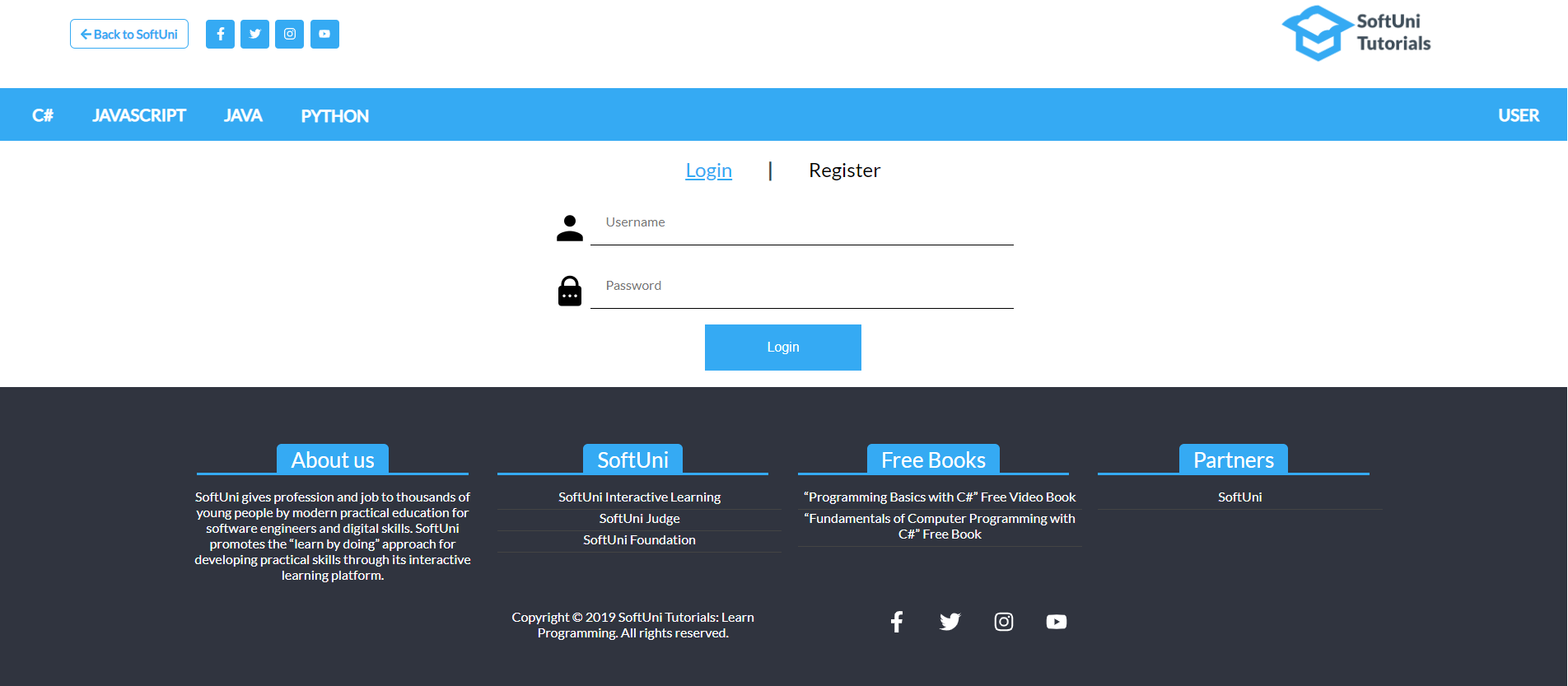
For example, if the **url** is "***localhost:8080/javascript/object-composition***" the result should be the picture below:



### User

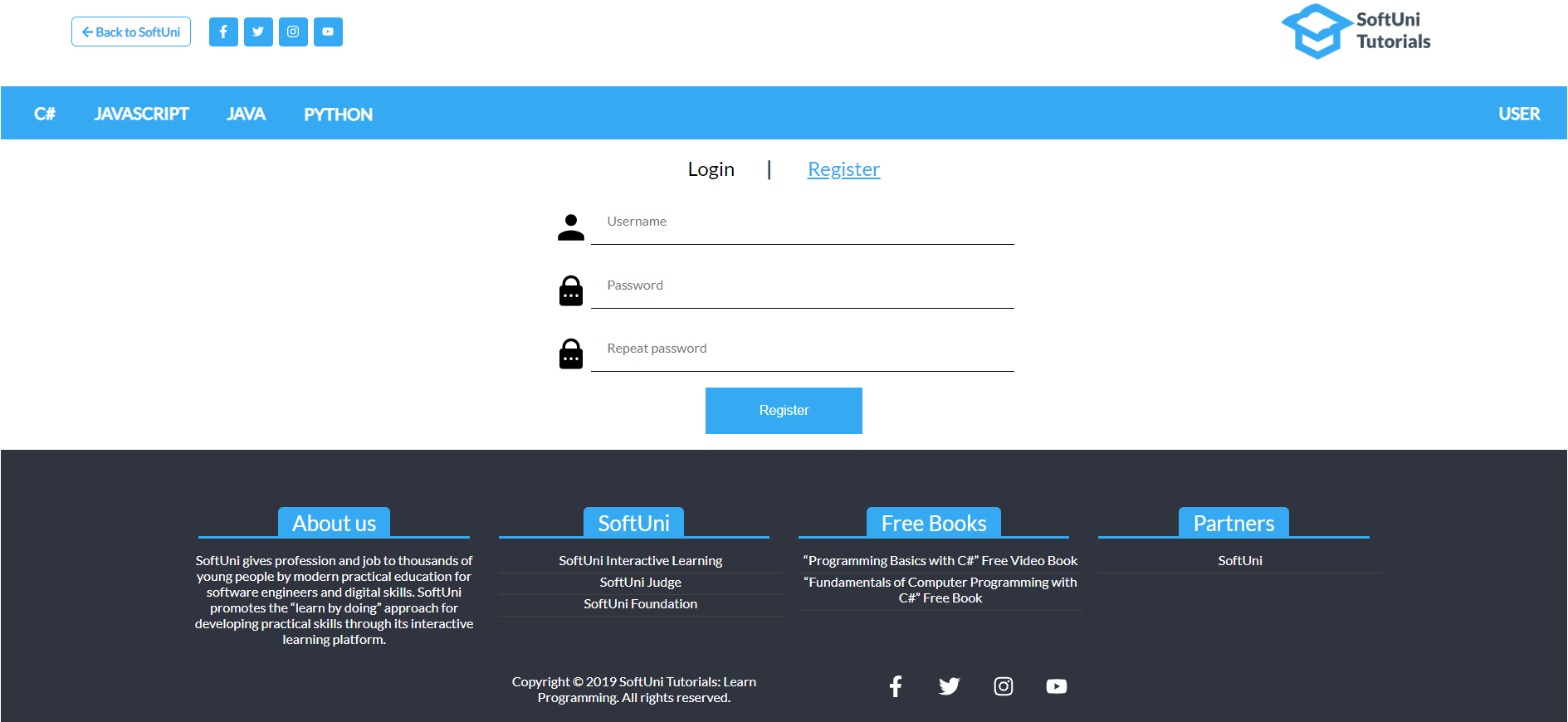
[**USER**] button on the right corner should lead to the login page ("***localhost:8080/user/login***")

**POST Request** - [**http://localhost:9999/api/user/login**](http://localhost:9999/api/user/login) The API expect **email** and **password**. After successful login the API return cookie (JWT info).



[**Register**] button inside that page should lead to the register page ("***localhost:8080/user/register***")

**POST Request** - [**http://localhost:9999/api/user/register**](http://localhost:9999/api/user/register). After successful registration, the user should be redirected to the login page, because the **auth** **cookie** is provided only after successful login. The API expect **email** **and** **password.**



## Logged-in User

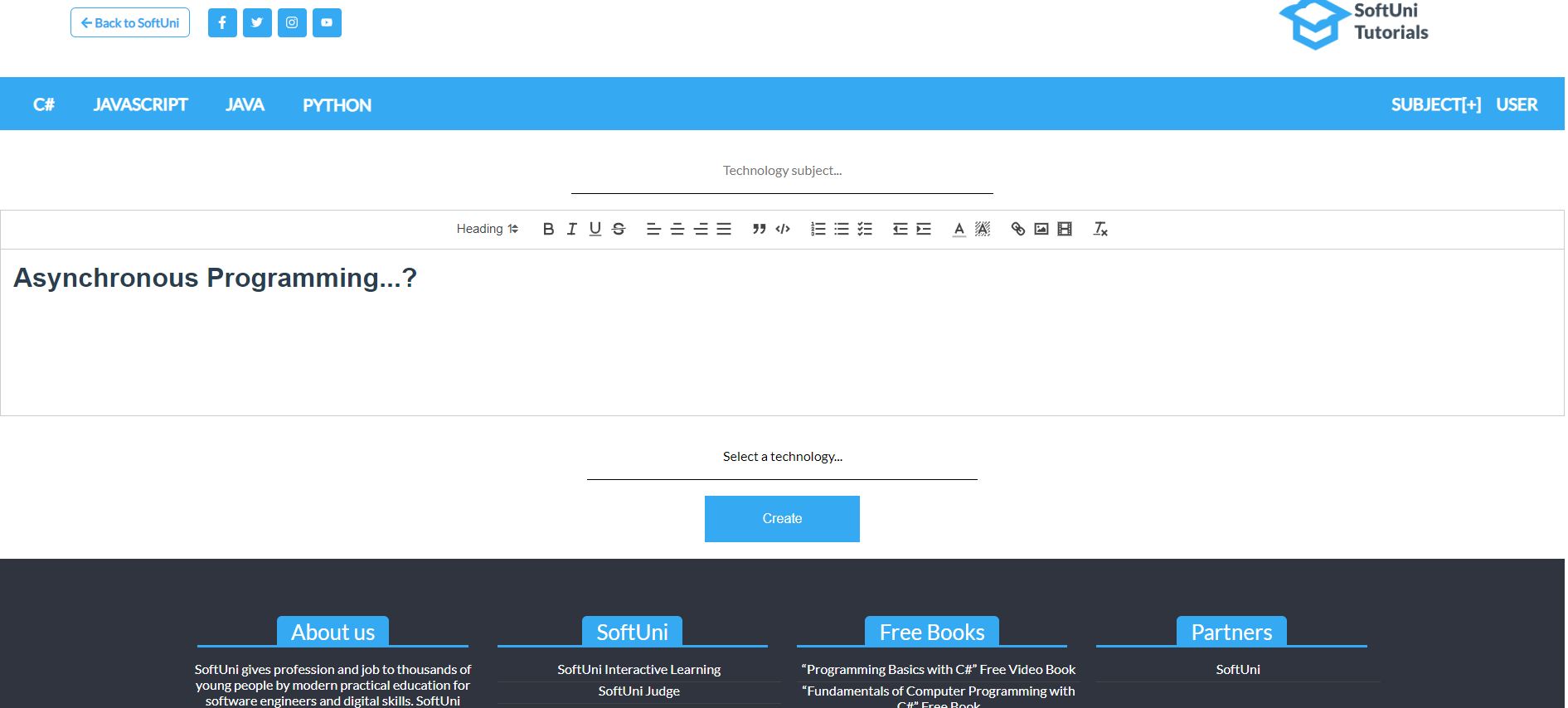
Every logged-in user also should have access to all the technology routes. There are 2 additional routes only accessible for the logged-in user: Create tutorial (**SUBJECT [+]**) and Profile page (**USER**)

Be sure this two pages have **route** **guards** and can be accessed **only** by **authenticated** users.

### SUBJECT [+]

**POST Request -** [**http://localhost:9999/api/tutorials**](http://localhost:9999/api/tutorials)

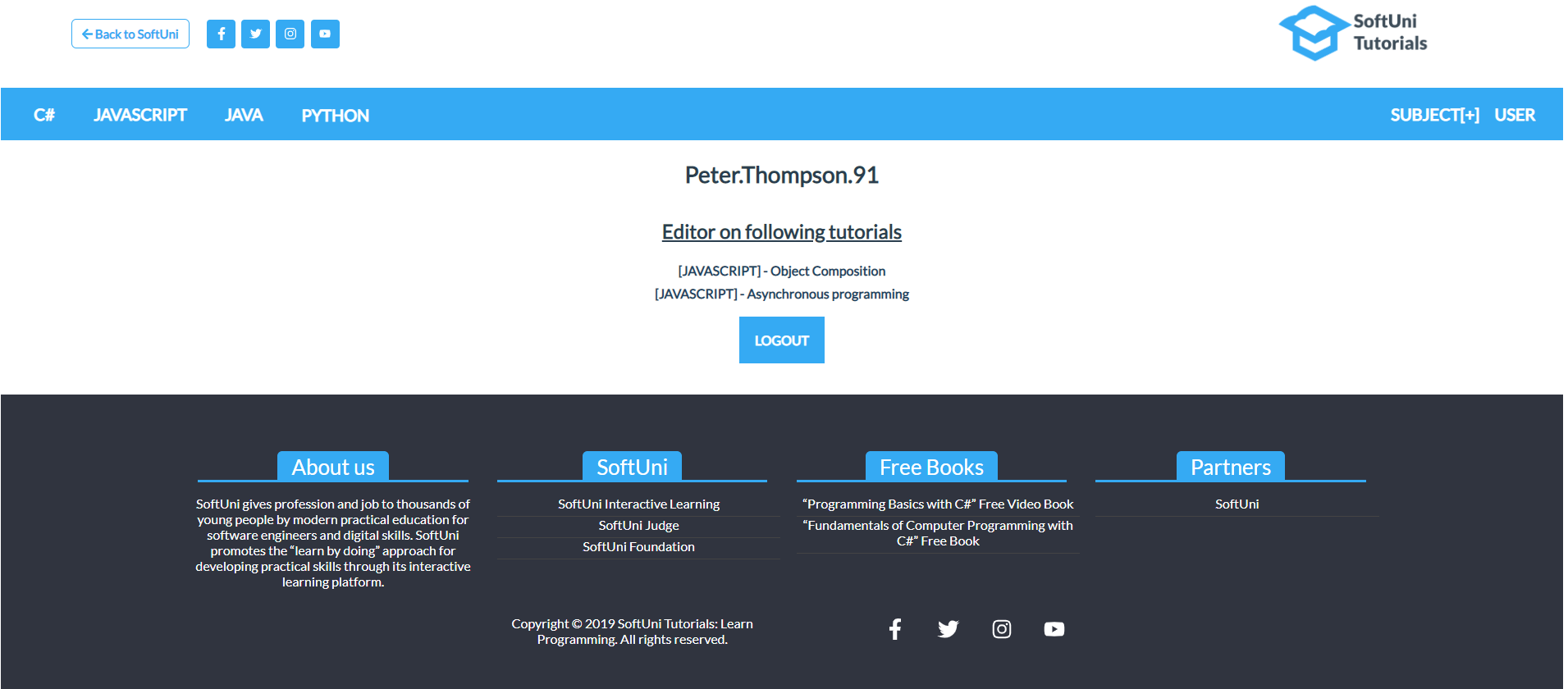
The API expect: **name (technology subject)**, **content (article) and technology**



### User (Profile)

**GET Request -** [**http://localhost:9999/api/auth**](http://localhost:9999/api/auth).

The "profile" page should display the email/username on the user and his **editing history**



Clicking over [**LOGOUT**] button, the user should be logged-out, clearing all of his authentication info on both sides.

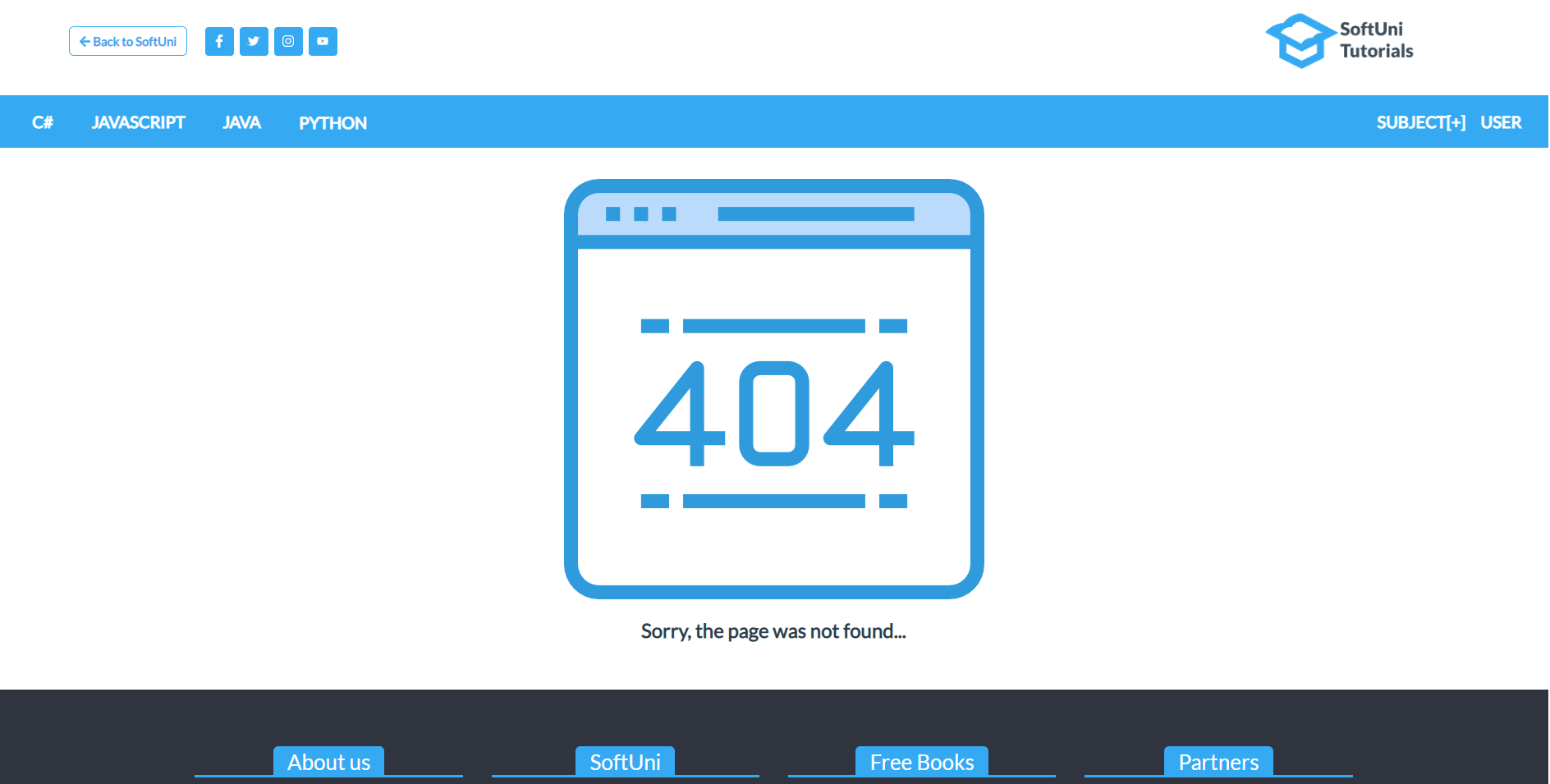
**POST Request -** [**http://localhost:9999/api/user/logout**](http://localhost:9999/api/user/logout)

**Each listed tutorial** that the current user is created should be **link** to the "**edit page**" **on that tutorial**. You can reuse the **SUBJECT [+]** page and functionality to do that or create additional similar to it.

**PUT Request -** [**http://localhost:9999/api/tutorials/{tutorialId}**](http://localhost:9999/api/tutorials/%7btutorialId%7d)

## Other

Use this template like **WildCard** for all invalid routes



***GOOD LUCK* 😊**