# Express.js Retake Exam - Video Tutorials

## Exam rules:

* You have 6 hours **from 09:00 to 15:00**
* When you are ready, delete the ***node\_modules***folder, make sure all dependencies are listed in the ***package.json*** file and submit your archived project at: [SoftUni Judge](https://judge.softuni.bg/Contests/1652/ExpressJS-19-April-2019)
* You are provided with **HTML & CSS** resources, which you are allowed to **modify**
* Use **Express.js** as a back-end framework
* Use **MongoDB** as a database with **mongoose**
* You can use whatever **view engine** you like (Handlebars, EJS, Pug etc.…)

## Application Overview

Get familiar with the provided **HTML & CSS** and create an application for **video tutorials**, **courses** and **course lectures.**

## Functional Requirements

The Functionality Requirements describe the functionality that the **Application** must support.

### Guest (not logged in)

The **application** should provide Guest (not logged in) users with the functionality to login, register and **view** the Home page.

### Users (logged in)

The **application** should provide **Users** (logged in) with the functionality to **view all the courses which are listed**, **course details page of all listed courses** and **they should be able to enroll in courses**. **Enrolled** **in** users should be able to see **lectures** in that course and the **Play Video** section. Courses can be **public** or **not.** Users can access **only** **public courses.**

### Admin (logged in)

The **application** should provide Admin (logged in) with the functionality to logout, create courses**, edit courses, create lectures, delete lectures**.

The admin should be **added to the database when the application starts**. A **registration of a new admin** user should **not be possible**!

## Database Models (15 pts)

The **Database** of the Tutorials application needs to support **3 entities**

### User (5 pts)

* Username - string (required), unique
* Password - string (required)
* **Enrolled Courses - a collection of** Courses
* **Roles - array with roles ("User", "Admin")**

### Course (5 pts)

* **Title - string (required), unique**
* **Description - string (required), max length of 50 symbols,**
* **Image Url - string (required),**
* **Is Public - boolean, default - false,**
* **Lectures - a collection of Lectures,**
* **Users Enrolled - a collection of Users**

### Lecture (5 pts)

* **Title - string (required)**
* **Video Url - string (required)**
* **Course - a single Course**

Implement the entities with the **correct datatypes**.

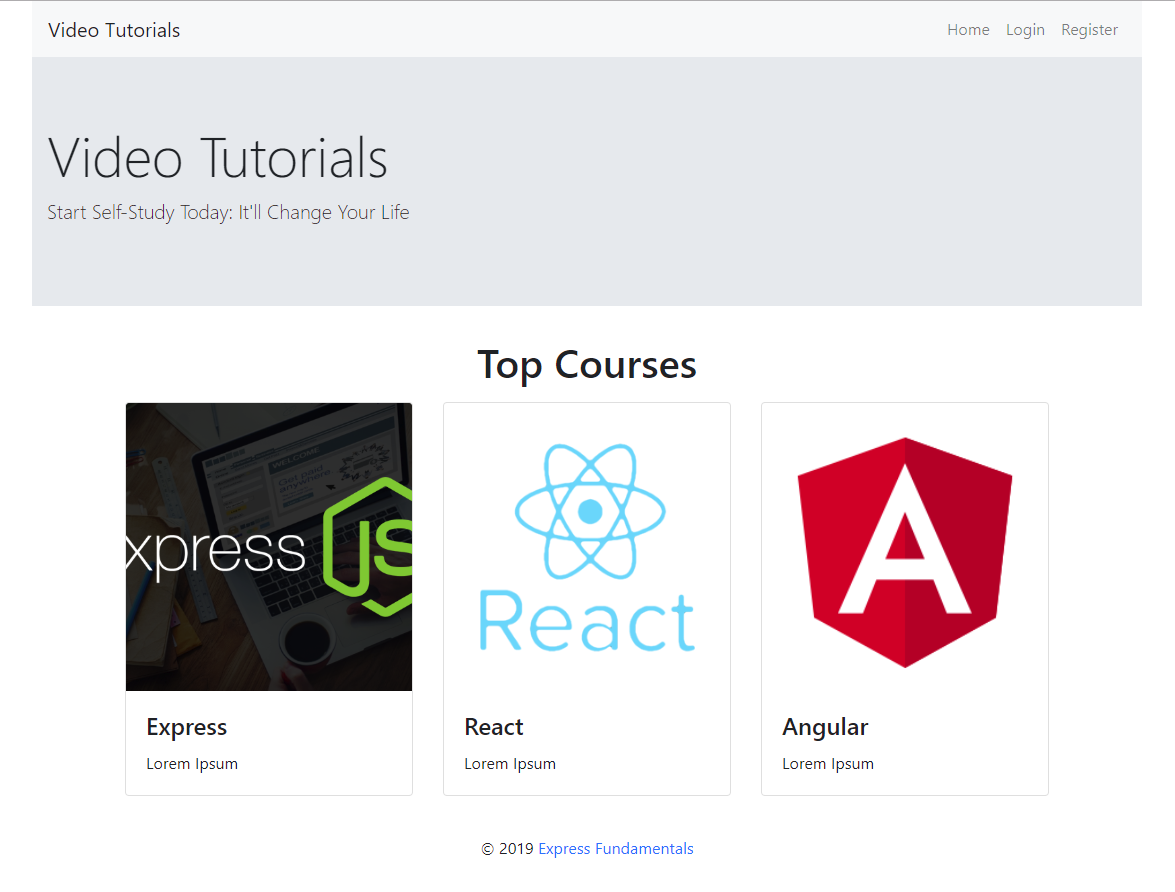
## Application Pages (80 pts)

### Guest Pages (10 pts)

These are the pages and functionalities, accessible by **Guests (logged out users).**

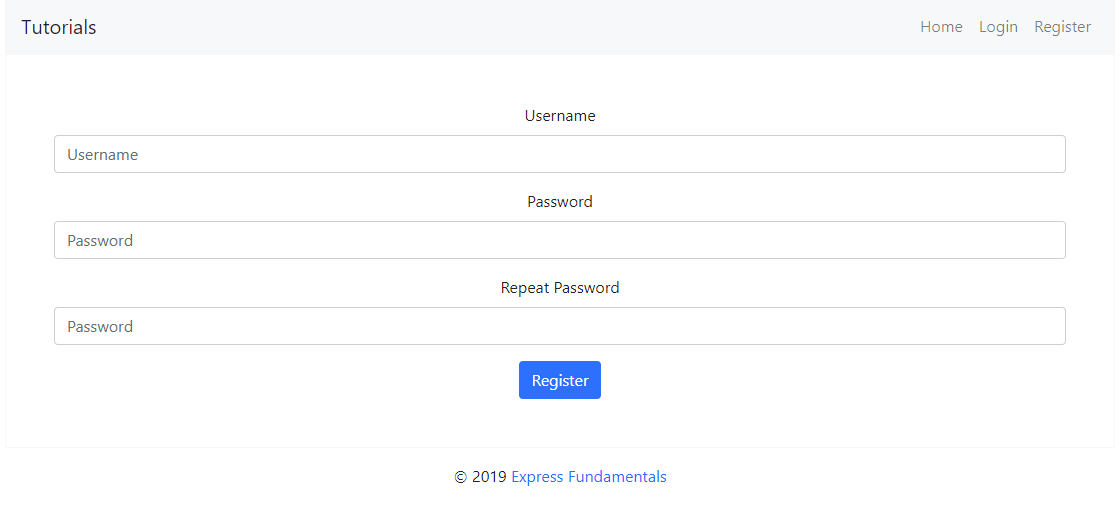
#### Home Page (logged out user)

List the **top three (3) public courses ordered** by the **count** of enrolled in users in **descending** order.

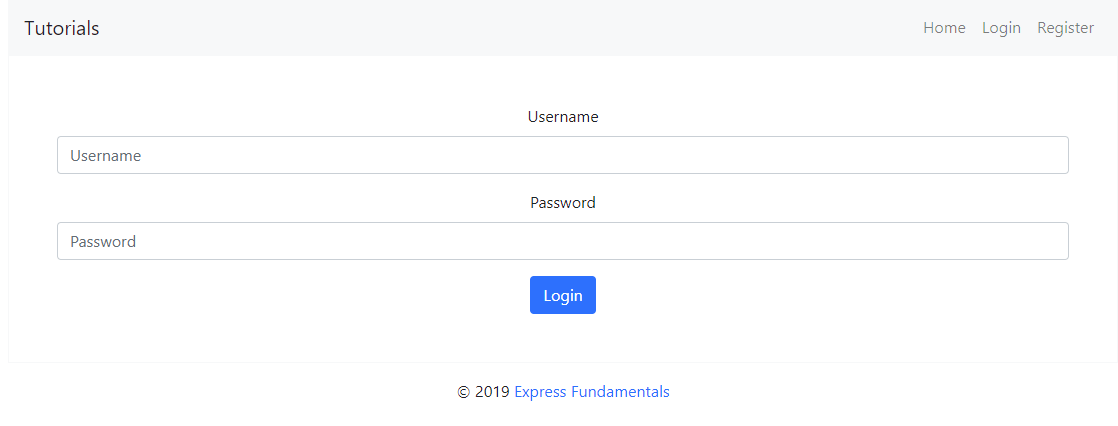


#### Register Page (logged out user)

Register a user inside the database with **username** and **password**. Both **passwords** must **match**!



#### Login Page (logged out user)

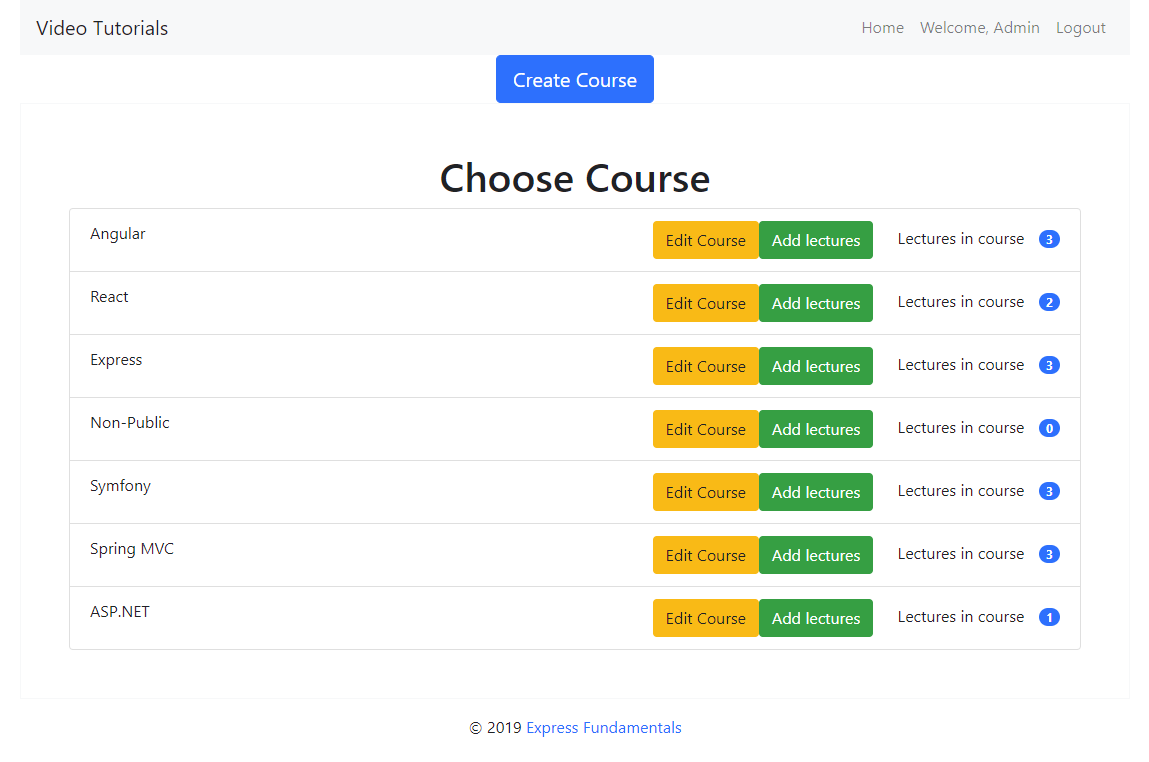


### Admin Pages (40 pts)

These are the **pages** and **functionalities**, accessible by Admin

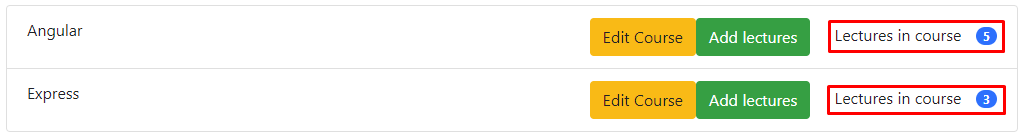
#### Home Page (logged in Admin)

This is the **home** admin page. An admin can **create**, **edit** **courses** and **create** the **lectures** in them.

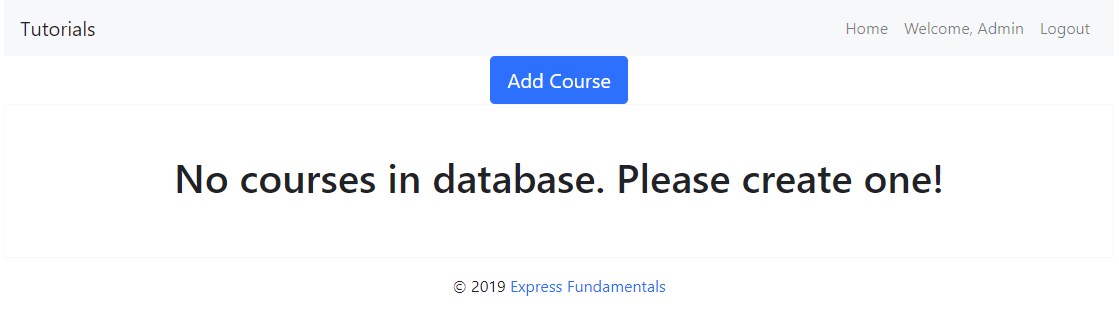


#### List All Courses (logged in Admin) (10 pts)

List **all courses** from the database and the **count** of lectures in them.

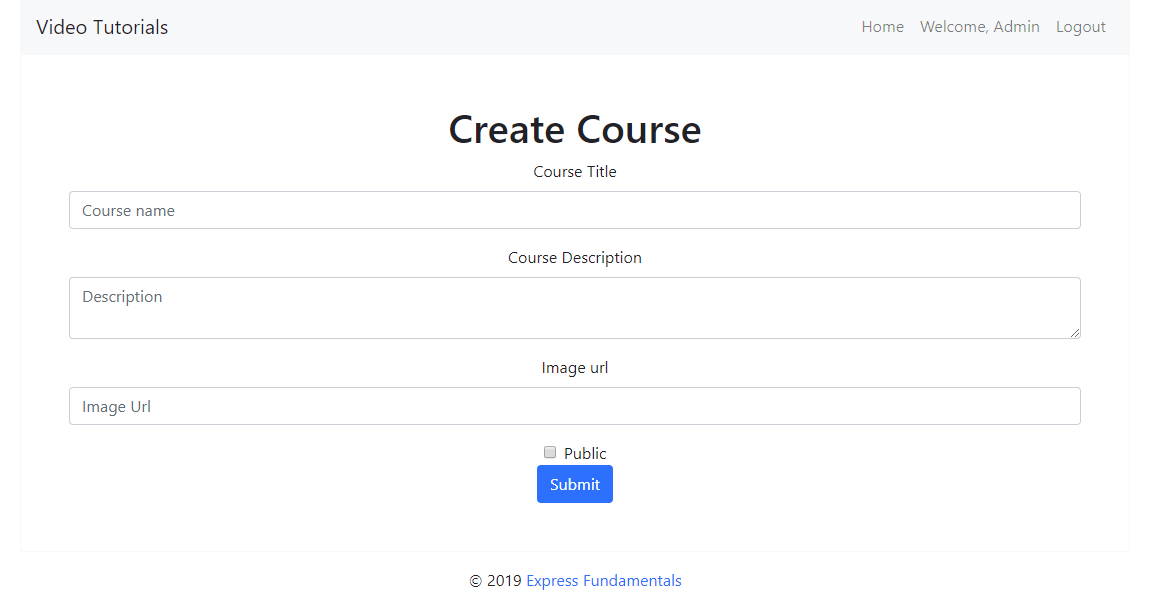


If there are no courses in the database yet, display "**No courses in database. Please create one!**"



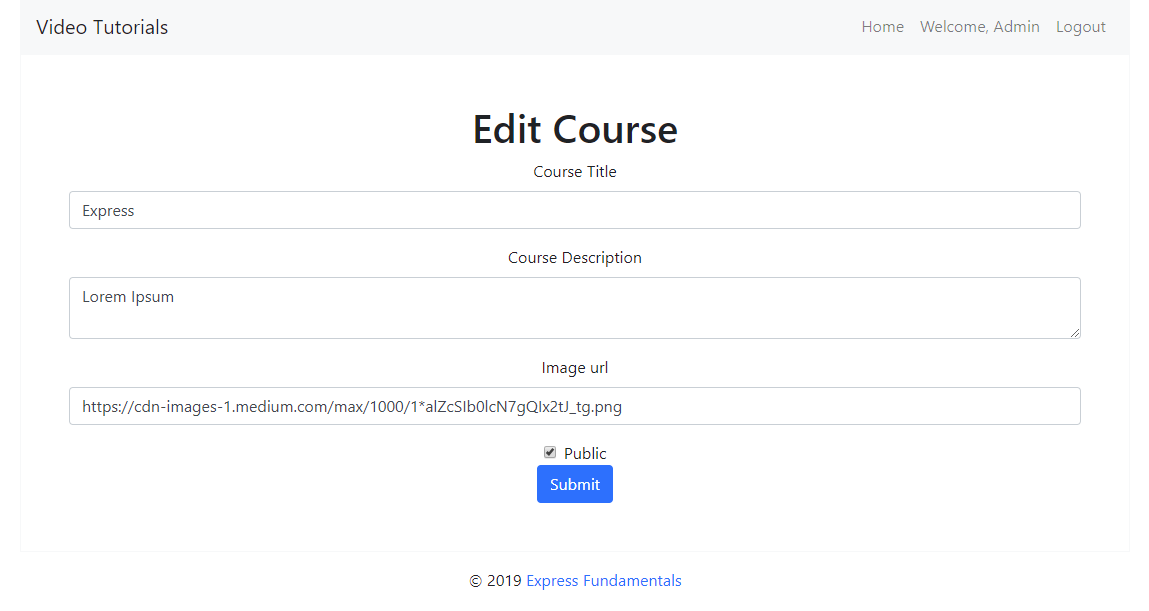
#### Create Course (logged in Admin) (5 pts)

Enter course **title**, **description**, **image URL** and choose if the course will be **public** or not. You will receive the value of the checkbox as a string **"on"** if it's checked or **undefined** if it's not. You have to convert the received value as Boolean **true** or **false** and save the course in the database.



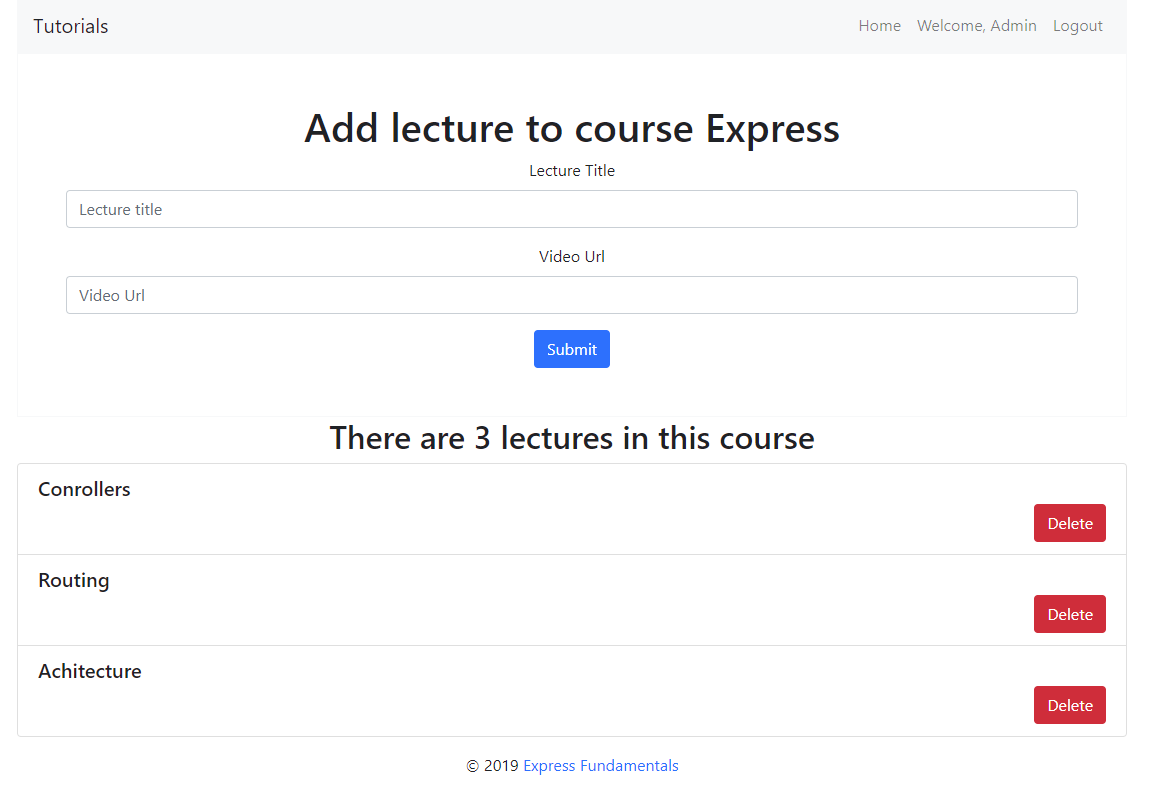
#### Edit Course (logged in Admin) (10 pts)

All form fields should be **filled** with the corresponding information of the selected course.

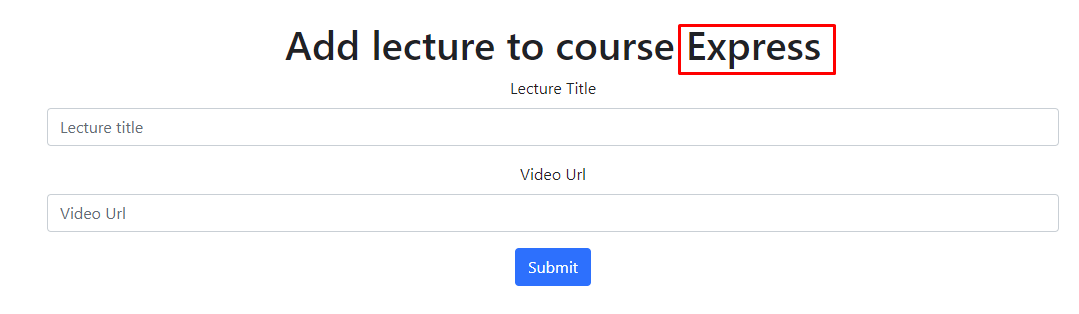


#### Lecture Panel (logged in Admin) (15 pts)

This view should include the **create form** with the **course title** and **all** of the **lectures** in that course with their **count**.

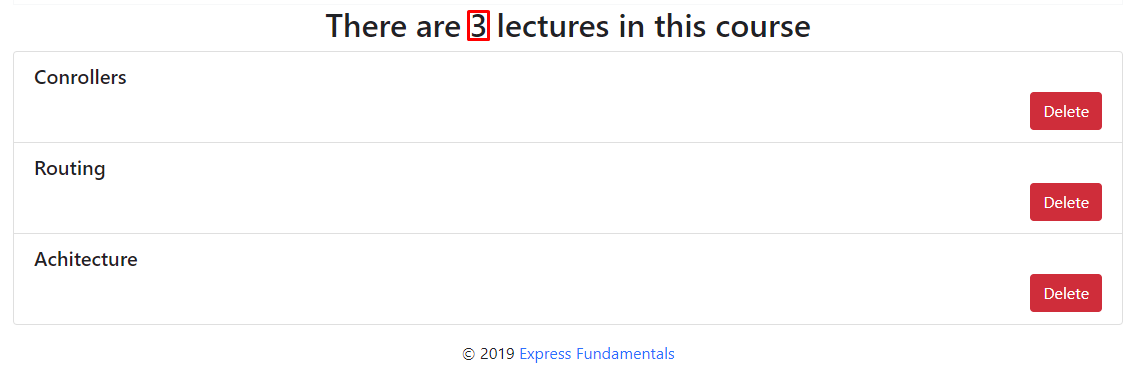


##### Create Lecture (logged in Admin) (5 pts)



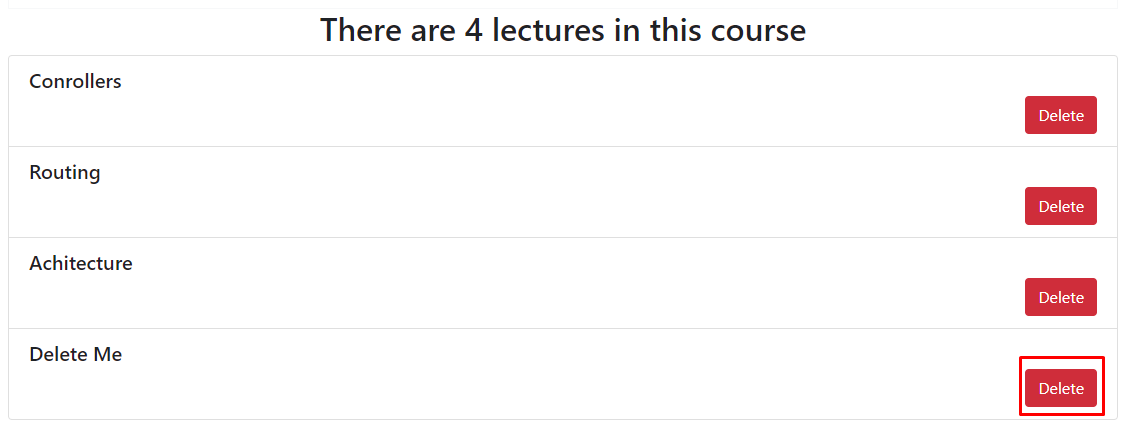
##### List Course Lectures (logged in Admin) (5 pts)

Below the **Add lecture** form, you should be able to see all **lectures** in the course. After adding a lecture, it should be instantly listed below the form.

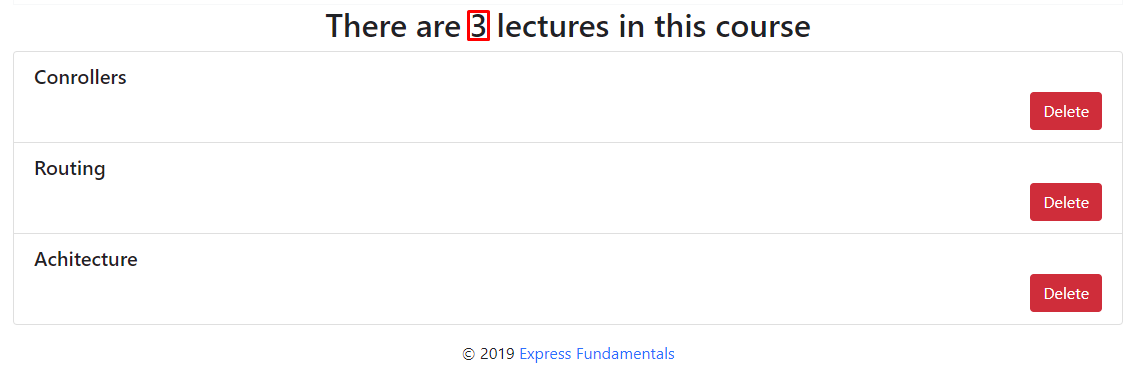


##### Delete Lecture (logged in Admin) (5 pts)

Upon deleting a lecture, you should be **redirected to the same page** only this time one lecture less should be shown.



After deletion:

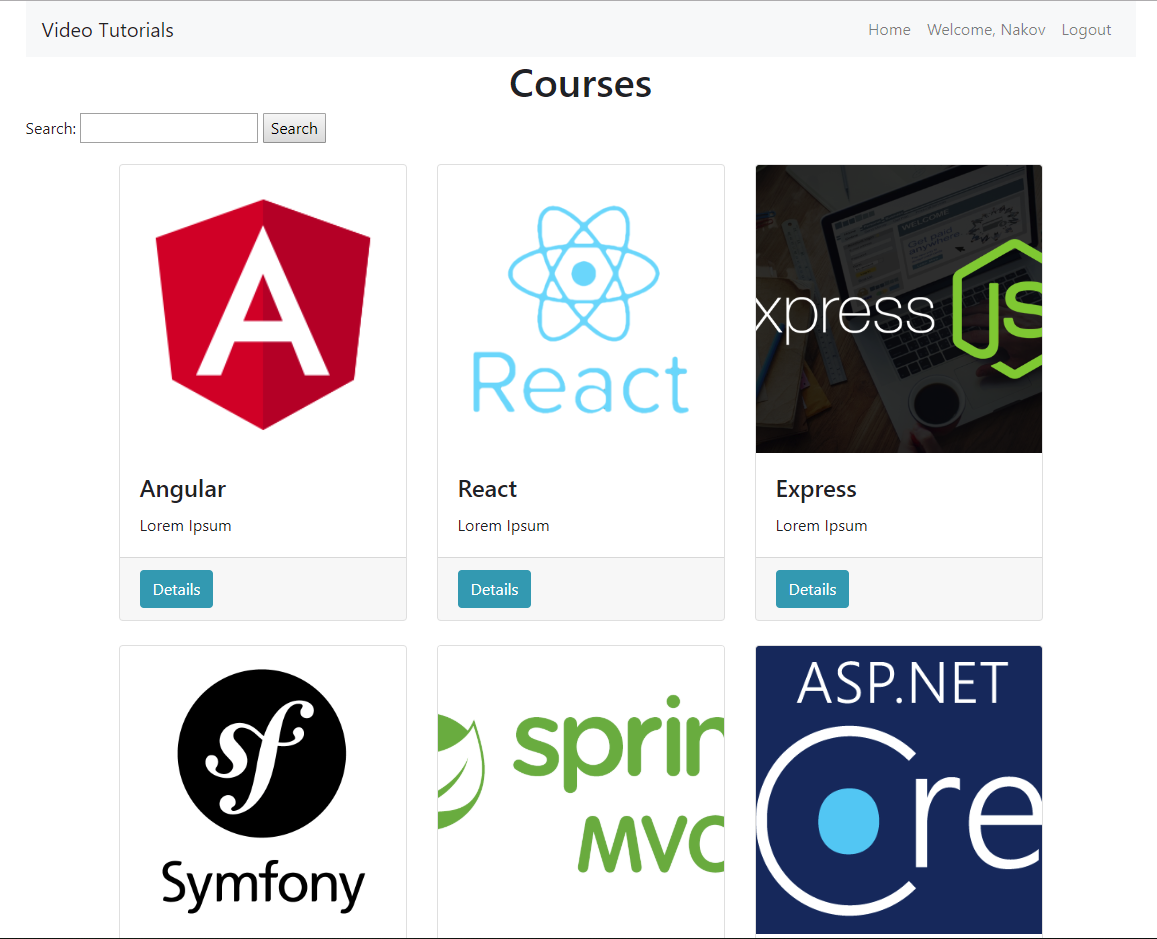


### User Pages (30 pts)

These are the **templates** and **functionalities**, accessible by Users (**logged in** users).

#### Home Page (logged in user) (5 pts)

List **all public** courses with a button **Details** which leads to a **course details page**.



#### Logged in Details Page (logged in user) (10 pts)

In the Details of a **public** course section, you should load course **name** and **description** and the **list** of lectures **if** the user is enrolled in the course.

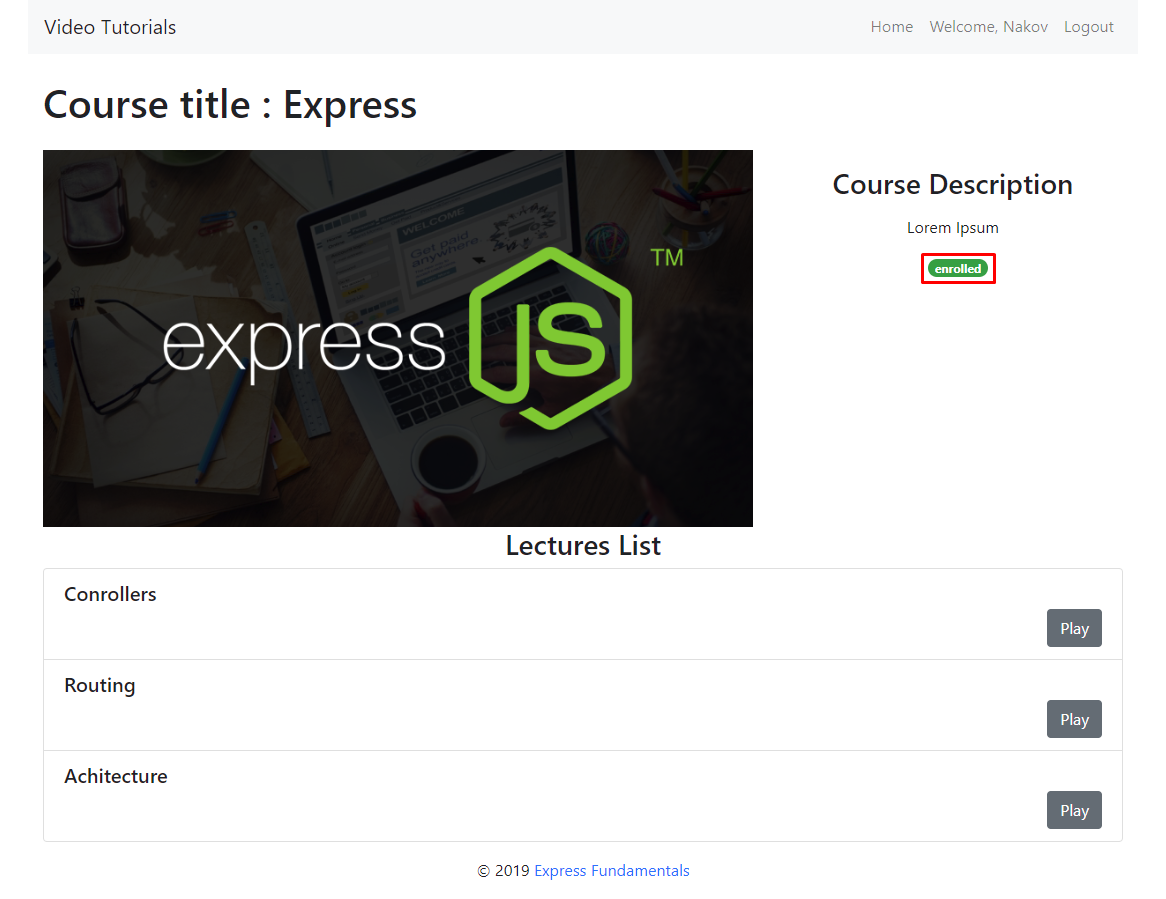
##### Not Enrolled in Course (logged in user) (5 pts)

If the user is **not enrolled** in the course you should display **Enroll button.** In this scenario the **lectures list** should be **hidden** and the following message: "**To get full access please enroll the course**" should be displayed.

****

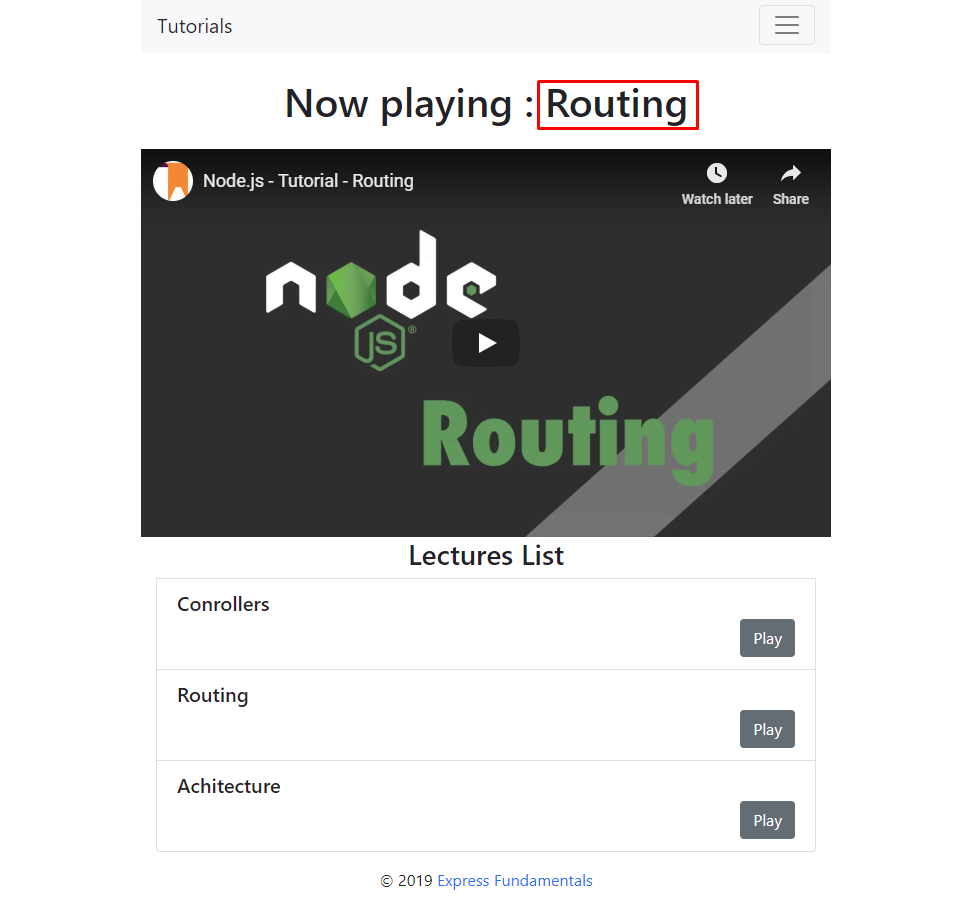
##### Enrolled in Course (logged in user) (5 pts)

**After** the user has enrolled in the course, the **Enroll button** should be **hidden** and all lectures in that course should be **listed** with a **Play** button which leads to a **Play video** page



#### Logged in Play video (logged in user) (10 pts)

You should display the current lecture **video** and the lecture **title** in **Now Playing: {{title}}.** This page is accessible only for **public** courses and users who are **enrolled** in that course. Below it you should **list all the lectures** in that course with the **Play** button.



## Security Requirements (5 pts)

The Security Requirements are mainly **access** requirements. Configurations about which users can access specific functionalities and pages.

* Guest (not logged in) users can access Home page and functionality.
* Guest (not logged in) users can access Login page and functionality.
* Guest (not logged in) users can access Register page and functionality.
* Users (logged in) can access All course page and functionality.
* Users (logged in) can access **Course** Details page and functionality.
* Users (logged in and enrolled in the course) can access **Play Video** page and functionality.
* Users (logged in) can access Logout functionality.
* Admins (logged in) can access **Course Panel (Admin home page)**.
* Admins (logged in) can access Create **Course** page and functionality.
* Admins (logged in) can access Edit **Course** page and functionality.
* Admins (logged in) can access the Lecture Panel page and functionality.
* Admins (logged in) can access the Logout page and functionality.

## \* Bonus (10 pts)

**Search in courses.** Filter only the **courses names** that **include the searched string** (**case-insensitive**).

