# 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](add_url_here) // url here
* You are required to **edit** the **provided HTML** to make it functional and **may extend** it as needed.
* 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 the **lectures** which belongs to them. The **user** should be able to **view courses,** and should be able to **enroll in. After** the user is enrolled in course, he/she is able to **view all the lectures** in that course. The **admin** should be able to **create courses, list or unlist them and add lectures**. He should also be able to **edit courses and delete lectures**.

**Note: The points described in each of the sections include all the functionalities that the given task should cover!**

## Functional Requirements

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

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

The **application** should provide Users (logged in) with the functionality to logout, **view** the Home page, view all courses**, and enroll to them. Lecture list is visible only to users which are enrolled in that course.**

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

The admin should be **seeded with the starting of the application**. A **registration of a new admin** user should **not be possible**!

### Users

The user should be able to **view all the courses which are listed**, **course details page of all listed courses** and **they should be able to enroll in**. **Enrolled** **in** users should be able to see **lectures** in that course and the **Play Video** section.

The user should **not be able** to **edit** or **delete** **courses** **nor the lectures**

### Courses and Lectures

Courses can be **listed** or **unlisted.** User can see **only** **listed courses.**

### Admin

The admin **should be able** only to create, edit courses and add, delete lectures in them.

## 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 Listed- 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**.

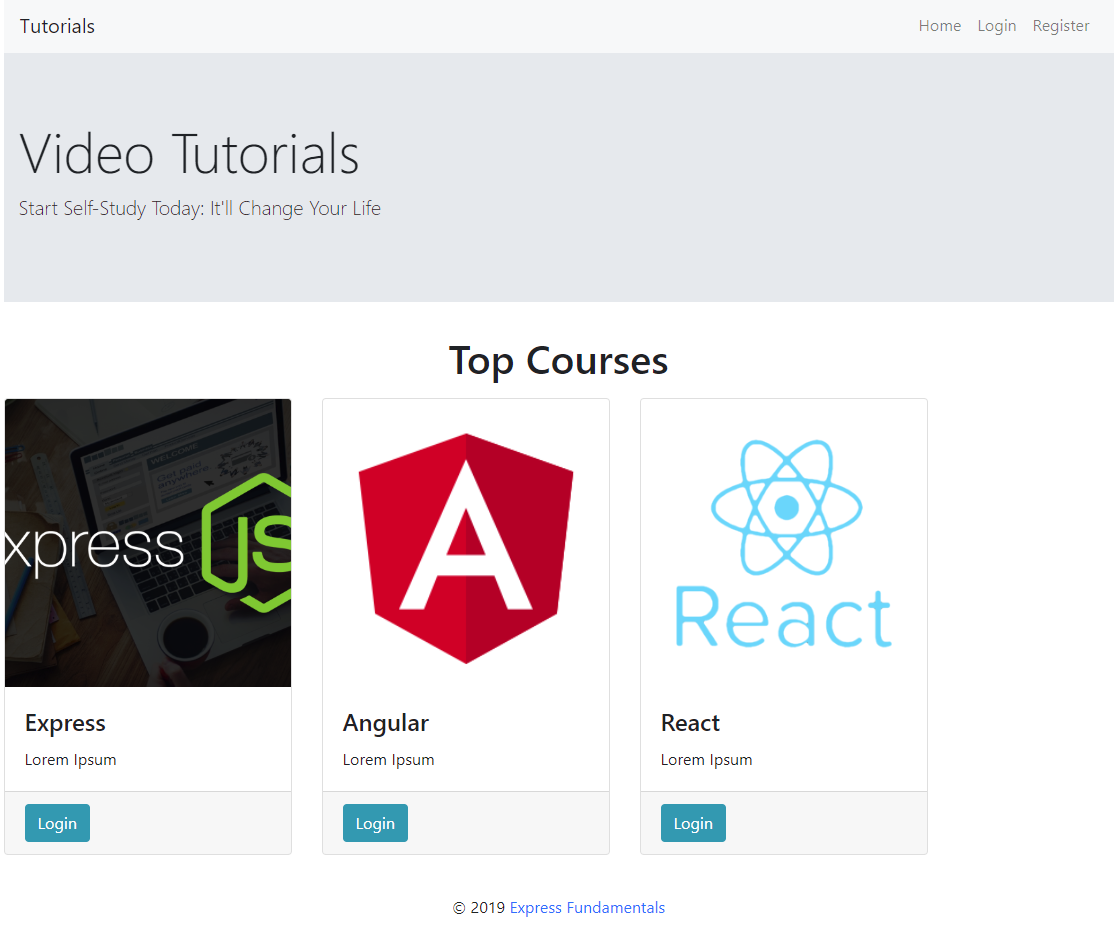
## Application Pages ( pts)

### Guest Pages ( pts)

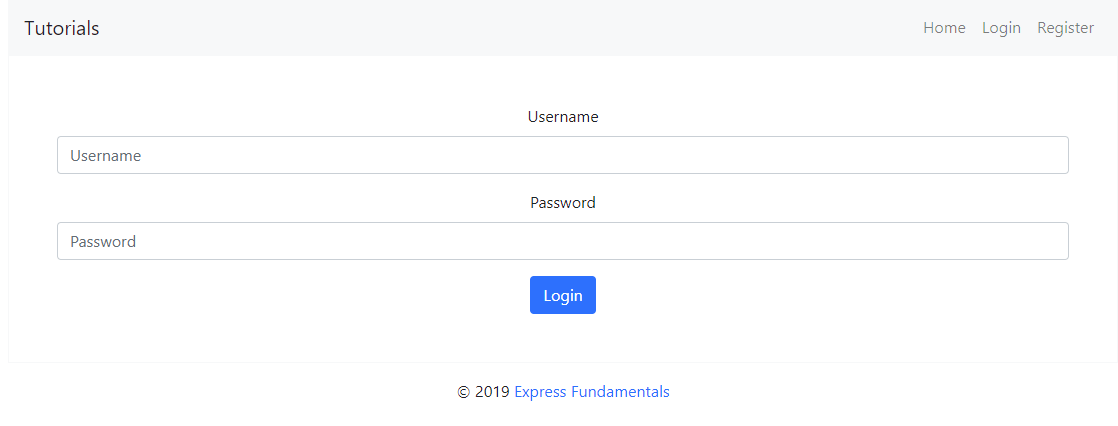
These are the **pages** and **functionalities**, accessible by Guests (**logged out** users).

#### Index Page (logged out user)

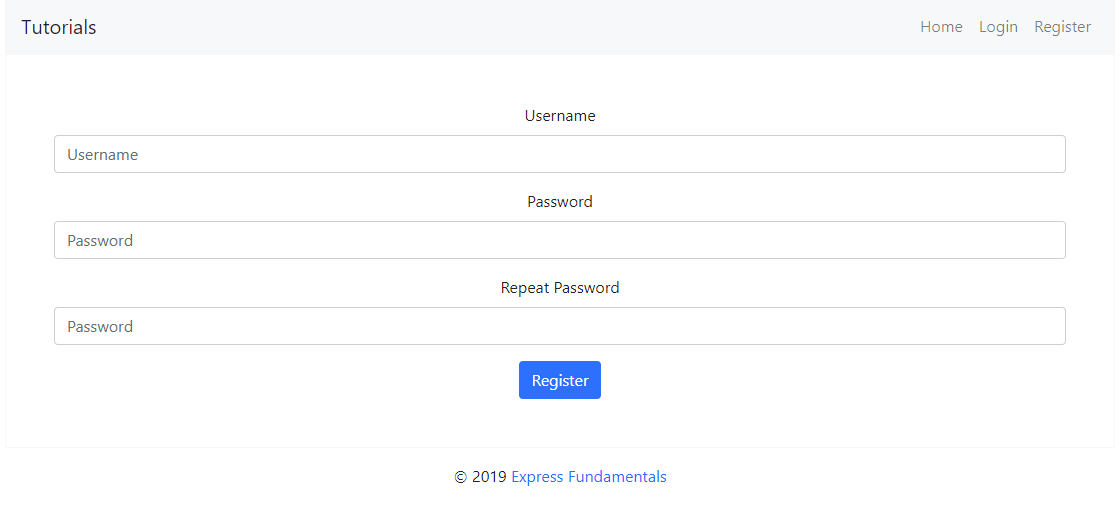
You should list the **top three (3) courses ordered** by the **count** of enrolled in users in **descending** order. In the bottom of every course card should be **Login** button.



#### Login Page (logged out user) ( pts)



#### Register Page (logged out user) ( pts)

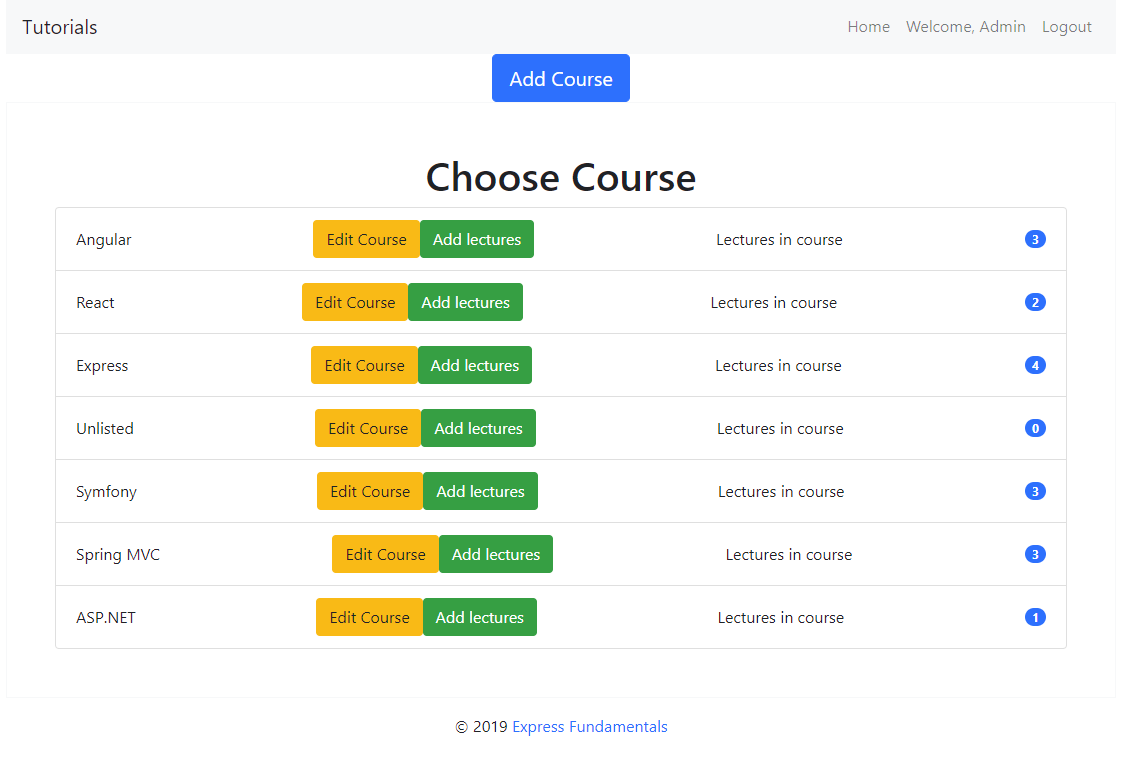


### Admin Pages ( pts)

These are the **templates** and **functionalities**, accessible by Admins

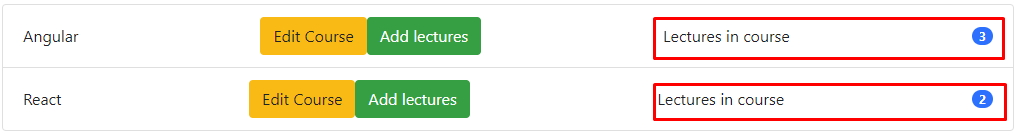
#### Admin Course Panel (logged in Admin) ( pts)

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

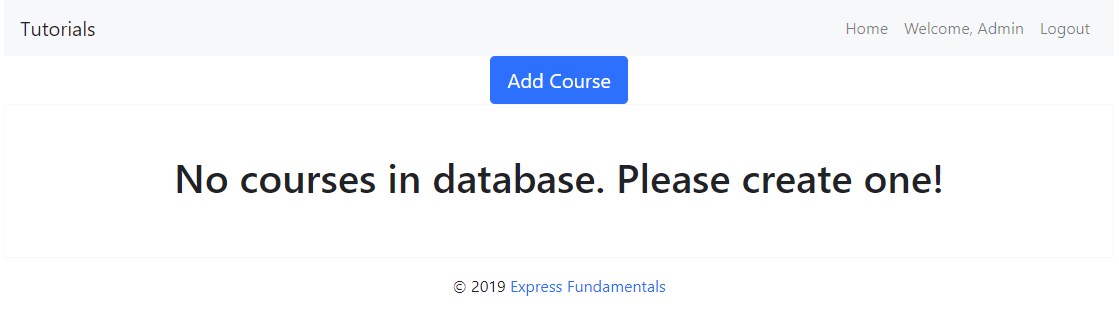


#### Admin List all courses in Course Panel (logged in Admin) ( pts)

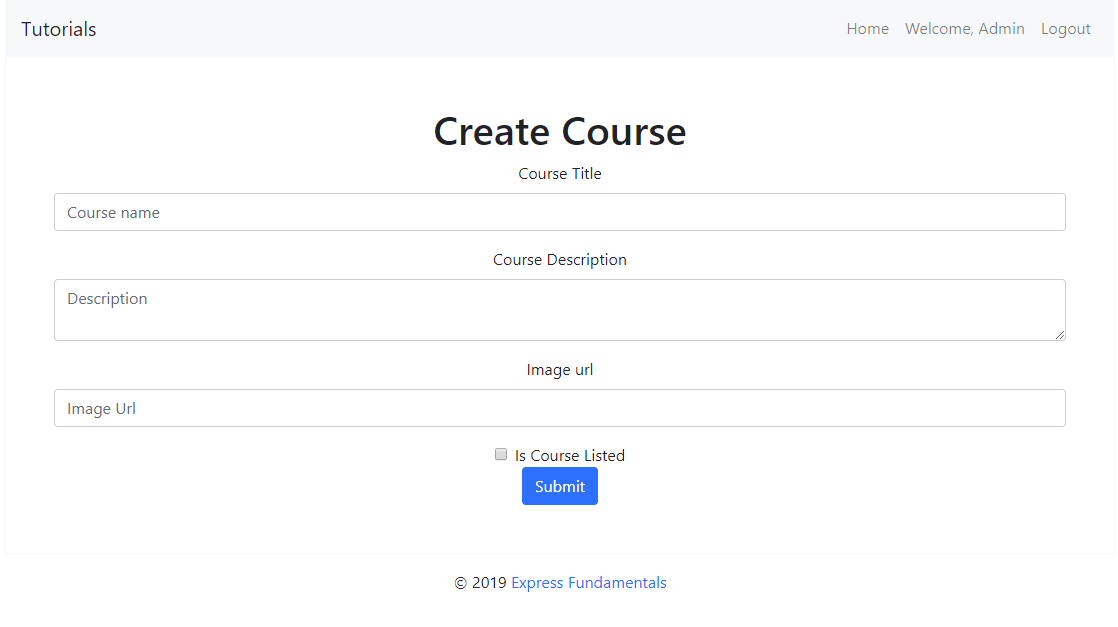
You should list **all courses** from the database and the **count** of the lectures in them.



If there is no courses in database yet display “**No courses in database. Please create one!**”

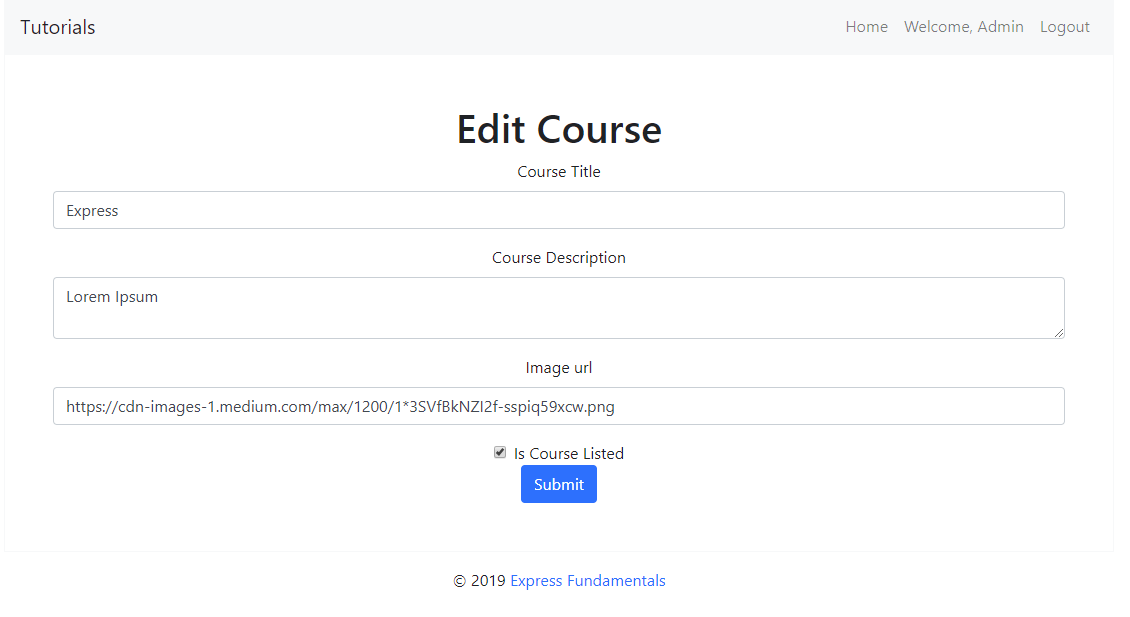


#### Admin Create Course (logged in Admin) ( pts)



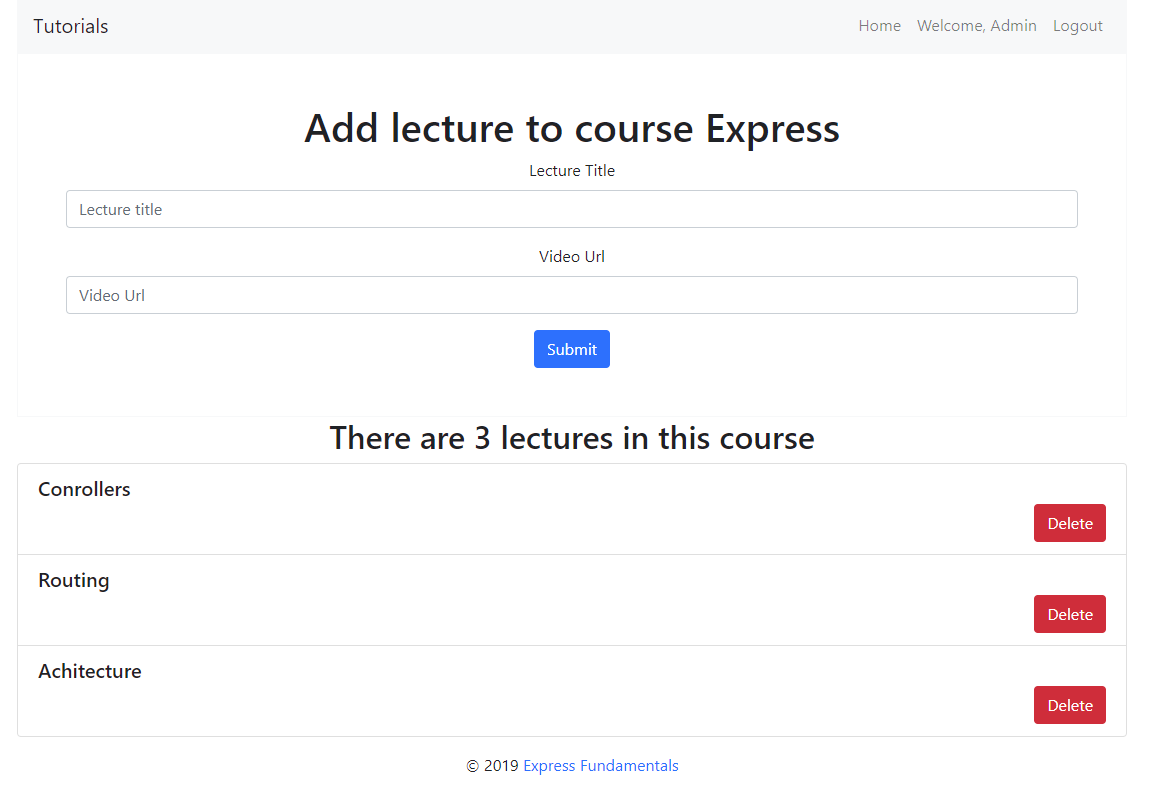
#### Admin Edit Course (logged in Admin) ( pts)

All form fields should be **filled** with the current information.

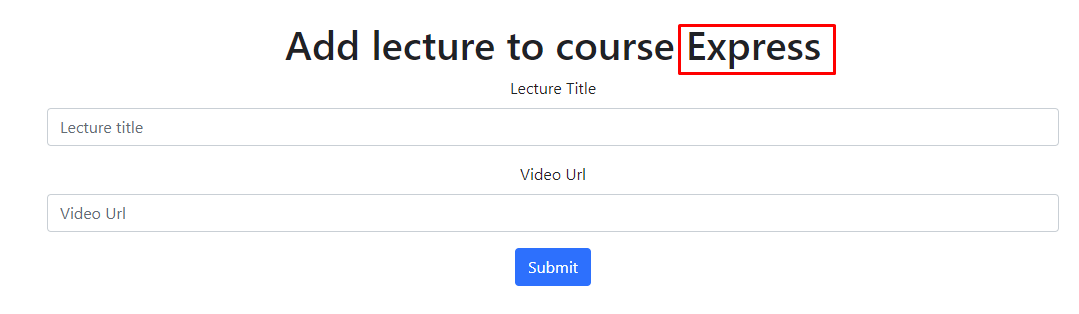


#### Admin Create Lecture (logged in Admin) ( pts)

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

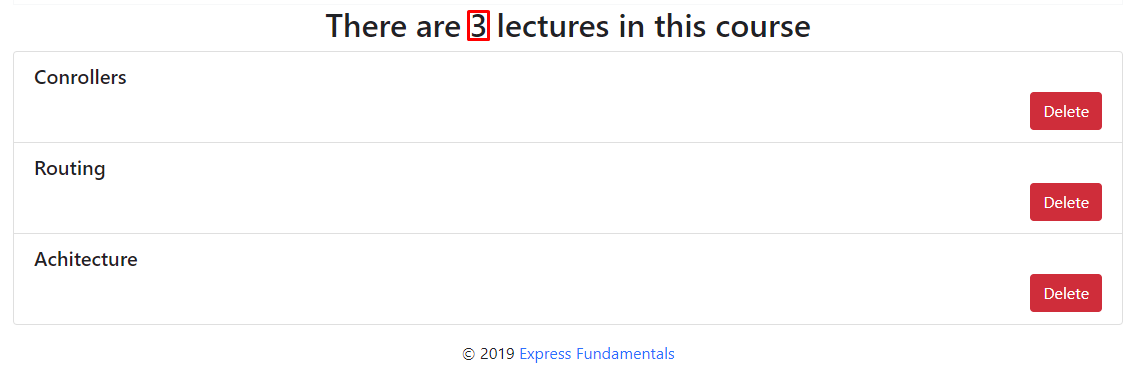


#### Admin Lecture Create form with the Course title (logged in Admin) ( pts)

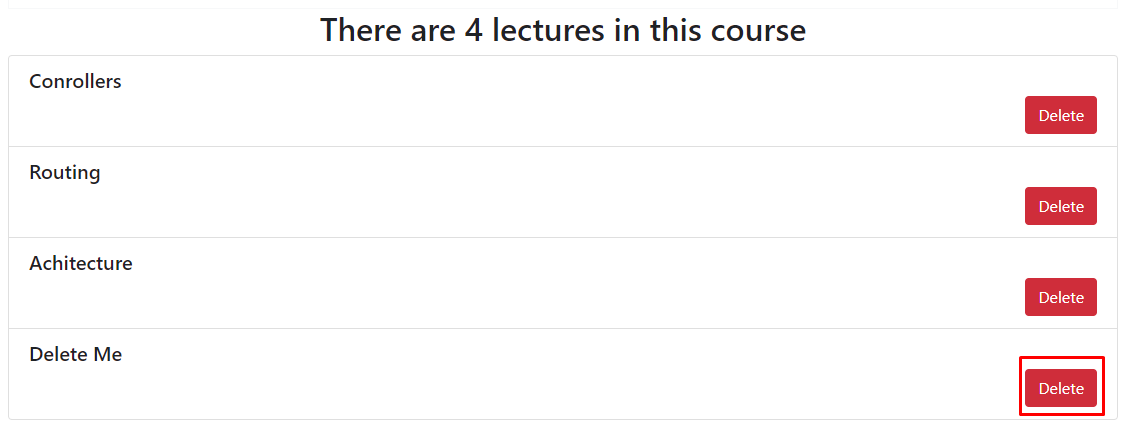


#### Admin List Lectures List in this Course (logged in Admin) ( pts)

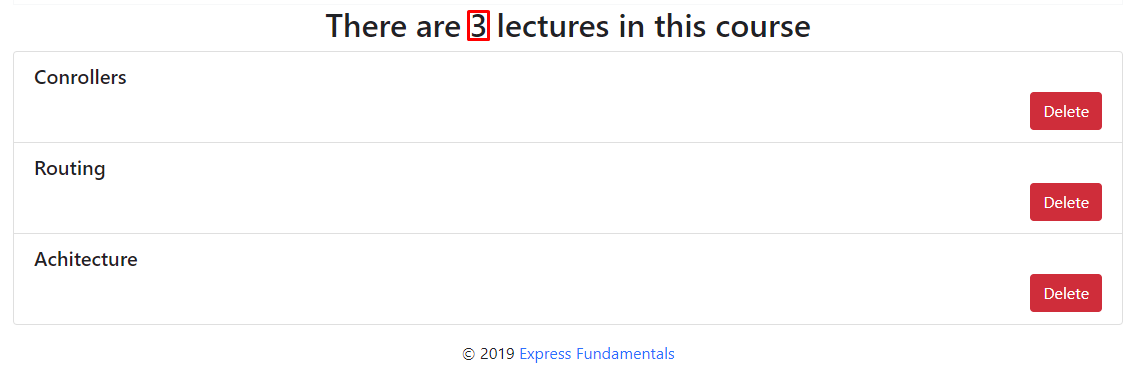
Below the add lecture form you should be able to see all **lectures** in the course. After adding, the new added lecture must be append to the list



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



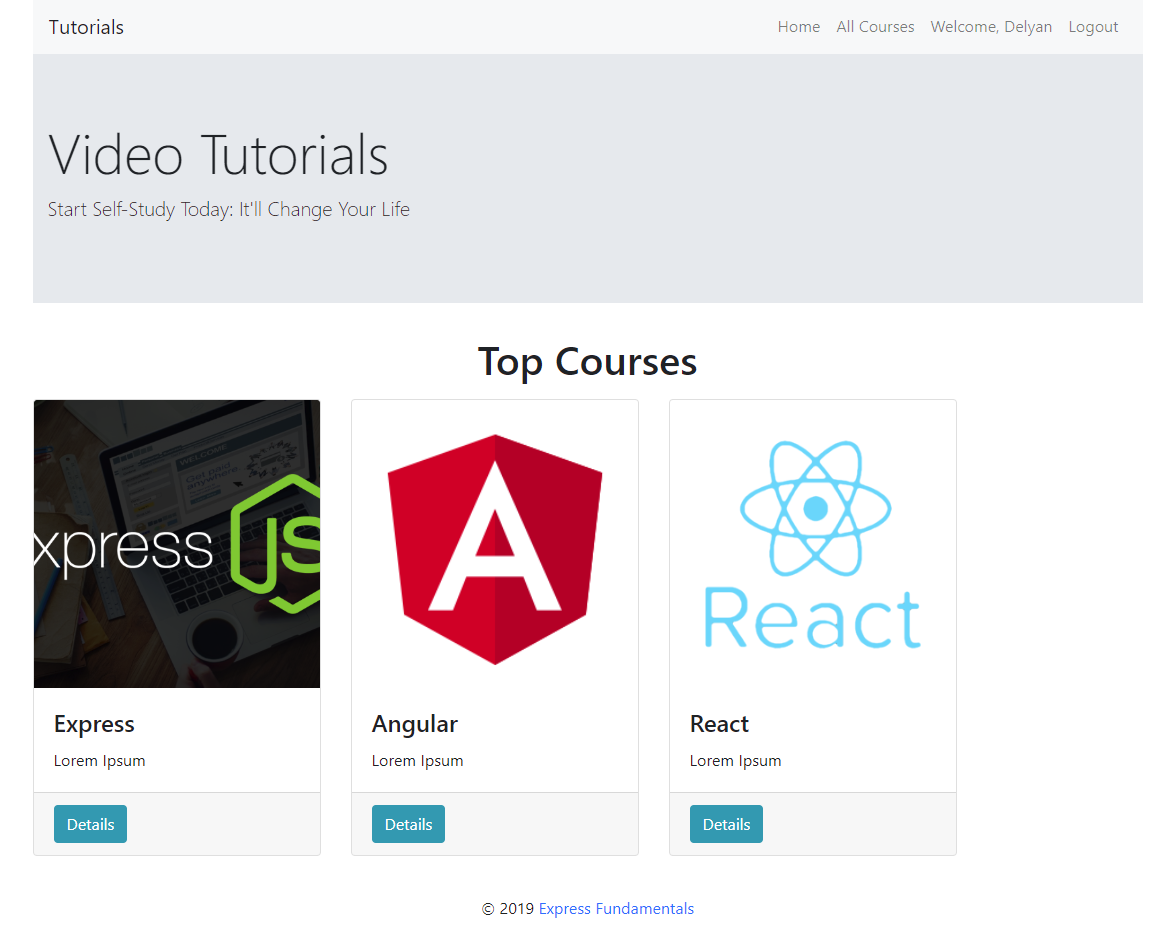
And after delete.



### User Pages ( pts)

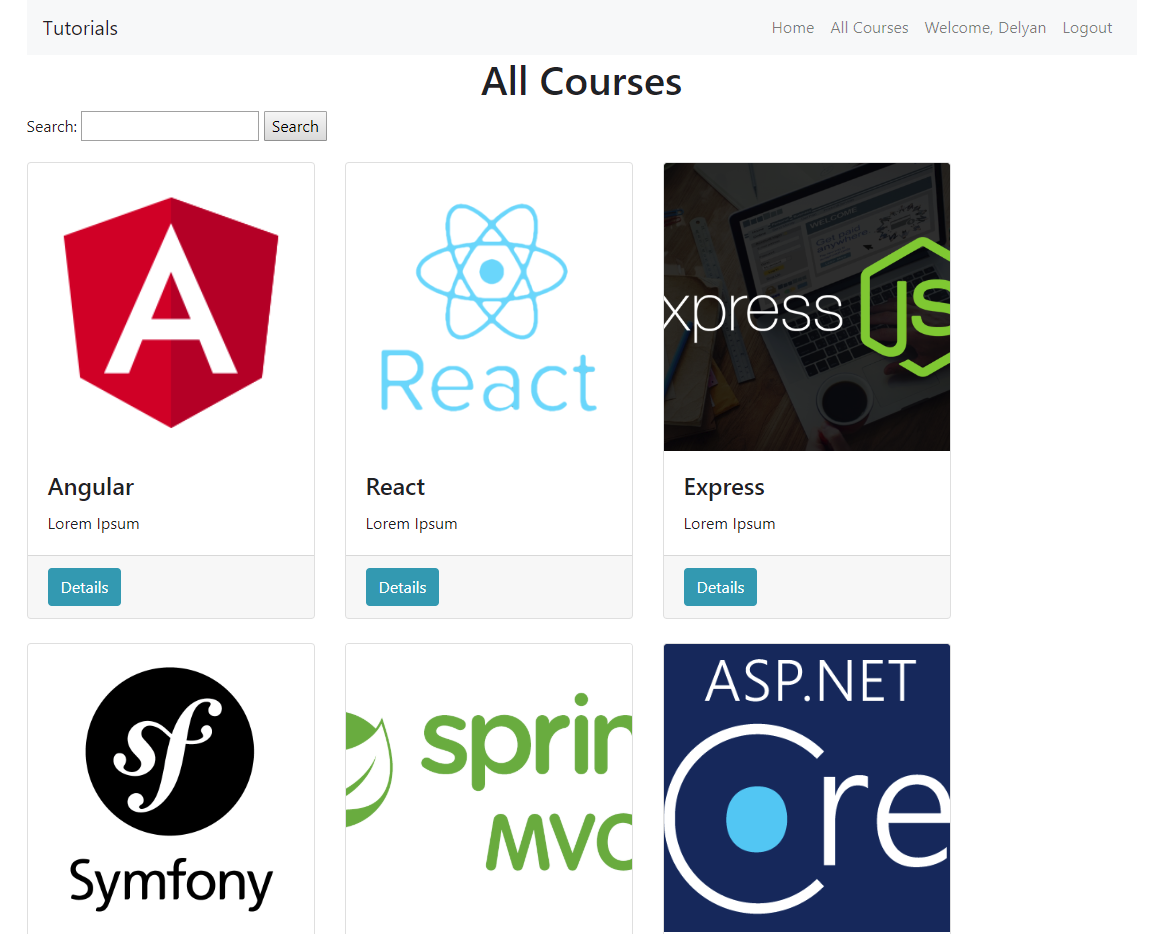
These are the **templates** and **functionalities**, accessible by Users (**logged in** users).

#### Logged In Home Page (logged in user) ( pts)



#### Logged In All Courses (logged in user) ( pts)

You should load **all listed** courses with button **Details** which leads to course details page.



#### Logged in Details page (logged in user) ( pts)

In the Details section you should load course name and description.

#### Logged, not enrolled in user Details page (logged in user) ( pts)

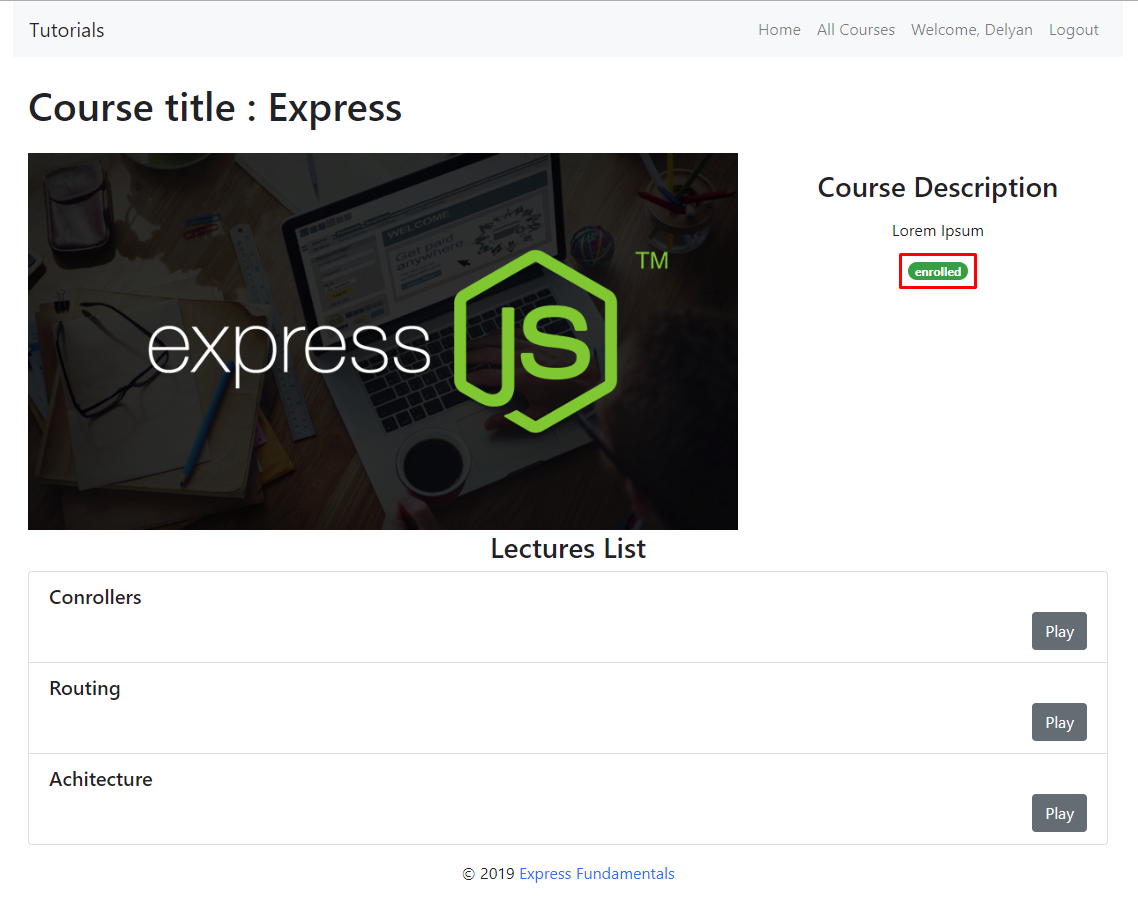
If the user is **not enrolled** in the course you should display **Enroll button.** In this scenario the **lectures list** should be **hidden**.

If the uses is **not enrolled** in are, a message should be displayed bellow the description **"To get full access please enroll the course "**.

****

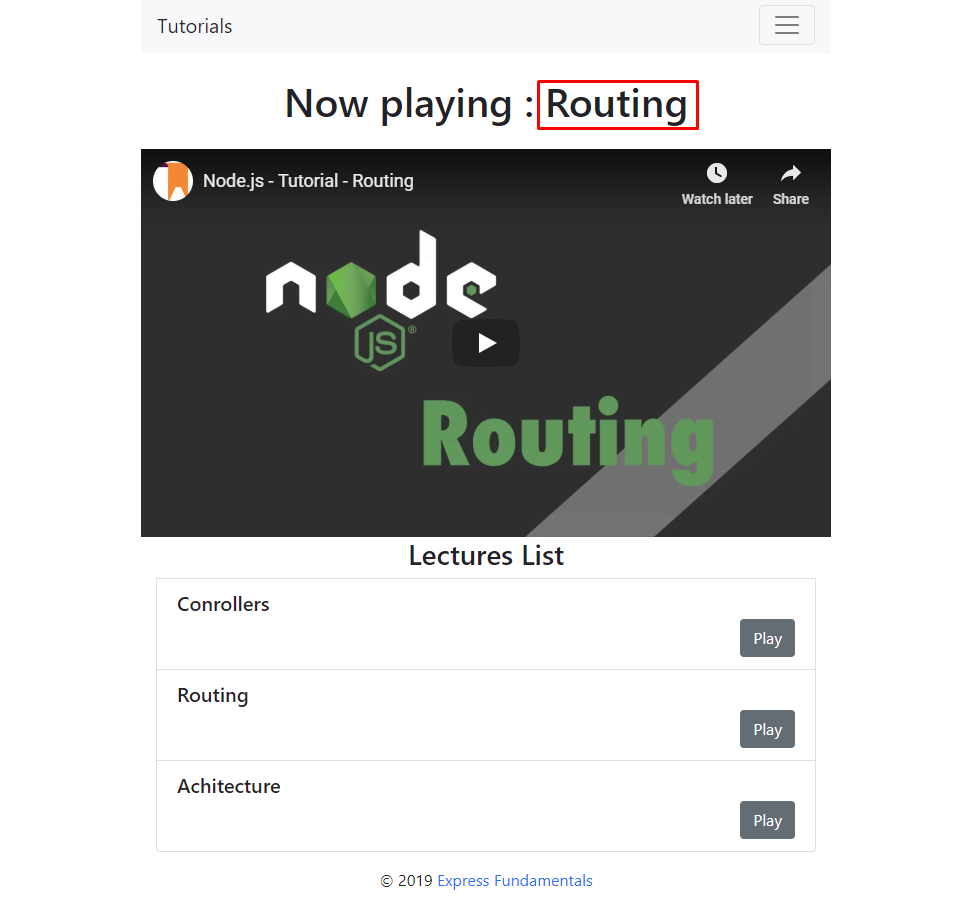
#### Logged and enrolled in user Details page (logged in user) ( pts)

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



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

You should display the current lecture **video** and the lecture **title** in **Now Playing title.** Below it you should list all lectures in that course with the **Play** button.



## Security Requirements (10 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 **Edit Course**.
* Admins (logged in) can access **Add lectures**.
* Admins (logged in) can access Create **Course** page and functionality.
* Admins (logged in) can access the Create Lecture page and functionality.
* Admins (logged in) can access the Logout page and functionality.

## Bonus (10 pts)

When a **user searches in courses**, filter the **courses names** that **include the searched string** (**case-insensitive**).

