

## Learning Management System

### Objectives

- Implementing a full front end using a front-end framework
  - Create multiple components with state handling
  - Implement routing
- Implementing a suitable backend for the project
  - Use a backend framework to implement a full backend
- Using Database in Web Applications

### Introduction

In this project, a customer requests creating learning management system. The customer has defined his requirements for the project.

You are required to design and develop the LMS given the customer's requirements.

### Requirements

#### 1. Users:

##### 1.1 User Types:

The Customer provided you with the following description of each user type:

##### 1- Site Administrator

The Site Administrator is an IT Engineer that manages the users' authorities. He has all other privileges.

##### 2- Learner

He/she is an end-user that had registered his personal data. He/she could enrol in a course and see his progress.

##### 3- Guest

He/she is an unregistered user or didn't login. He/she could login, register.

##### 4- Instructor

He/she is a course instructor who can add and edit courses.

##### 1.2 User Data:

Each user should have the following **personal data**:

- 1- Username (Should be unique)
- 2- Password
- 3- First Name

- 4- Last Name
- 5- Birth Date
- 6- Email (Should be unique)
- 7- Type (admin/customer)

### **1.3 User Features:**

Each user type has the following features:

- Site Administrator
  - Change User roles
    - Change a user's role from a learner to instructor and vice versa.
  - Have the same roles as instructor and learner
- Instructor
  - Create a course
  - Add activities to a course
- Learner
  - View courses and enrol
  - View activities of a course
  - View his progress and grades in a course
  - View and edit his profile
- Guest
  - Register a new account
    - The guest could register a new account
  - Login an existing account
    - The guest could login to an existing account

## **2. Courses**

A course should have a home page with information about the instructor and the syllabus. Each course should have a QA section where each question can start a thread.

## **3. Activities**

Activities can be one of the following:

- 1- Video (embedded from YouTube)
- 2- Pdf material
- 3- Quiz. Quizzes can include MCQ or true false questions only

The main webpages that must exist:

- Website Homepage (The main page for the LMS)
- Courses Page
- Login Page (The user should enter his username & password)

- For each feature decide whether it is a stand-alone page or a function embedded in another page.

## **Implementation Guidelines:**

- Implement full front-end pages with a frontend framework
  - Use the steps of design we went over. Start with a mock website.
  - Use a router to route different webpages (Components you created)
  - Use CSS and styling to design an appealing UI to your website. (Using bootstrap is optimal).
  - Use Ajax to send requests to your back-end server (Optional, you can use any other library to create your requests.)
- Implement a full backend
  - Implement an API for each needed request by your client side.
  - Implement a simple routing for each needed request.
  - Using an authentication library like OAuth will be a bonus.
- Implement a Database to store your data (MYSQL for example)
- Your frontend code must be completely separated from your backend. They should only communicate through post, get, or other REST methods.

## **Teams**

The teams should be 4-5 members in each team (not more or less). **Deadline:** Prepare and deliver a spreadsheet with the teams & names by Dec 14<sup>th</sup>, 2021.

## **Final Delivery**

**Deadline:** Jan 4<sup>th</sup>, 2021, at the section time.

- You will be required to deliver a .rar file a Google drive form (link will be shared by email). The .rar file should contain
  - A .txt file named ID.txt with your names, sections and BNs.
  - A folder named frontend containing your frontend project
  - A folder named backend containing your backend project
  - An SQL file (or any script) that does the following
    - Create all tables
    - Adds one admin account
    - Adds some dummy data
  - A .txt file named login.txt containing the login information of that admin account.

**Note: Complete Projects from the internet will get zero.**