## https://lh6.googleusercontent.com/0pFq3hP32XoE2MK4ubx-dw6iSMj14qLPA7m8ApXO28DseJ1hKsn1CQZBiTBqAzXqVFZZswG3s3En7NB8jgaId54riaqYe4zKvudqv4M6Sp9OCqKfsIBFdSevLvaM0xxgcpaba4trLearning Management System

**Problem Statement:** Create a online learning portal for a online training provider using NodeJS. Below are the important requirements and deliverables for the project.

1. **Business Requirement - Overview**To develop the Learning management application with features given below:
2. **Front-end Component:** There are different pages in the application that have to be created i.e. Home page ,Registration page and dashboard page etc.,.
3. **API (Application Programming Interface) Component:** 
   1. **API Component (** /api/login **)**: Implement REST endpoints POST.
   2. **API Component (** /api/register **)**: Implement REST endpoints GET, POST
   3. **API Component (** /api/createCourse **)**: Implement REST endpoints GET, POST
   4. **API Component (** /api/updateCourse **)**: Implement REST endpoints GET, POST
   5. **API Component (** /api/deleteCourse **)**: Implement REST endpoints GET, POST
   6. **API Component (** /api/editProfile **)**: Implement REST endpoints GET, POST
   7. **API Component (** /api/changeProfile **)**: Implement REST endpoints GET, POST
   8. **API Component (** /api/changePassword **)**: Implement REST endpoints GET, POST
4. **Schema Component** : Create collections **user**, **course** using MongoDB or MySql.
5. **Technical Specifications**

**2.1. Software and application details:**

|  |  |  |
| --- | --- | --- |
| Tools/Packages | Mongo  v3.2.19  Express v4.16.3  Mocha   v1.20.1  Node 8.9.0 | Install packages using either apt/npm. |
| OS Version | Ubuntu 16.04 LTS / Windows 7, 10 | 1 GB - 4 GB RAM and 1 - 4 vCPU. |
| Application port | Port 3000 is mapped port 80 | Application server runs on port 3000 in the virtual machine. This port is forwarded to port 80 publicly -- which is accessible via the live preview of the IDE (Integrated Development Environment). |

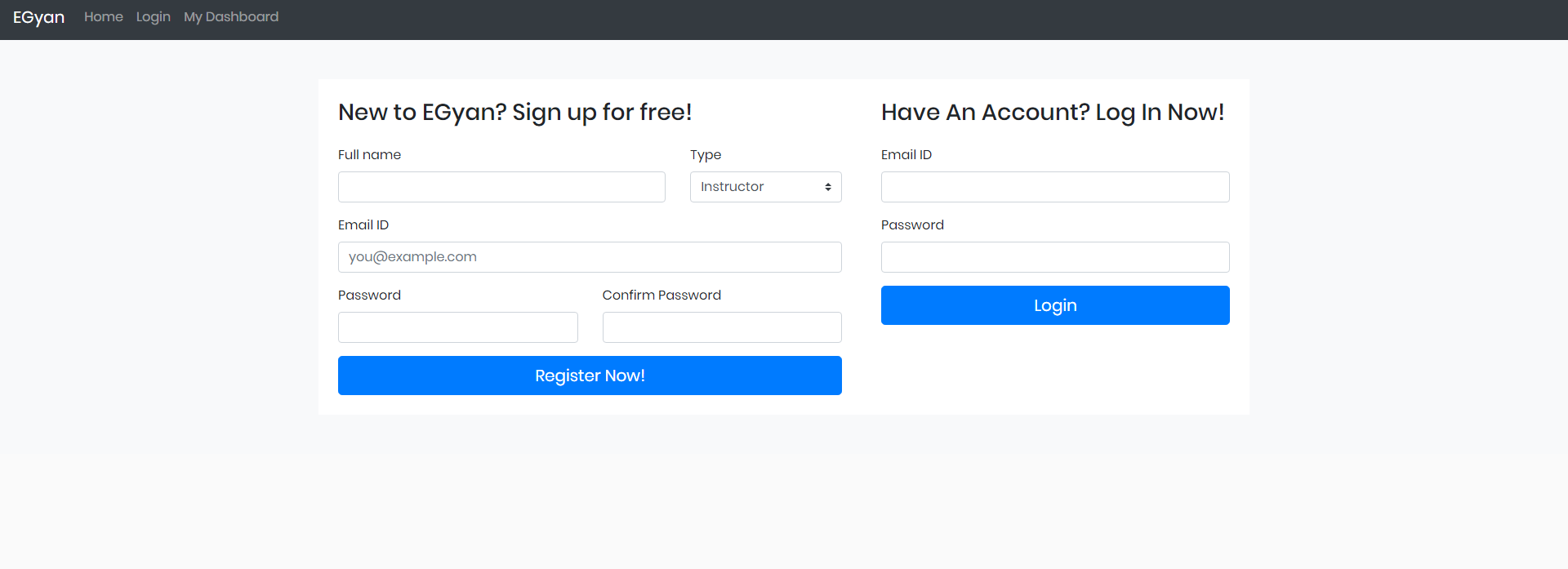
**2.2. UI (User Interface) - Design wireframes**

The UI wireframes with the required layouts are as follows:

**NOTE:** The candidate will not be evaluated based on the UI design (layout, color, formatting, and so on). The candidate is free to have a basic UI with all the required UI components (input fields, buttons, labels, and so on ).

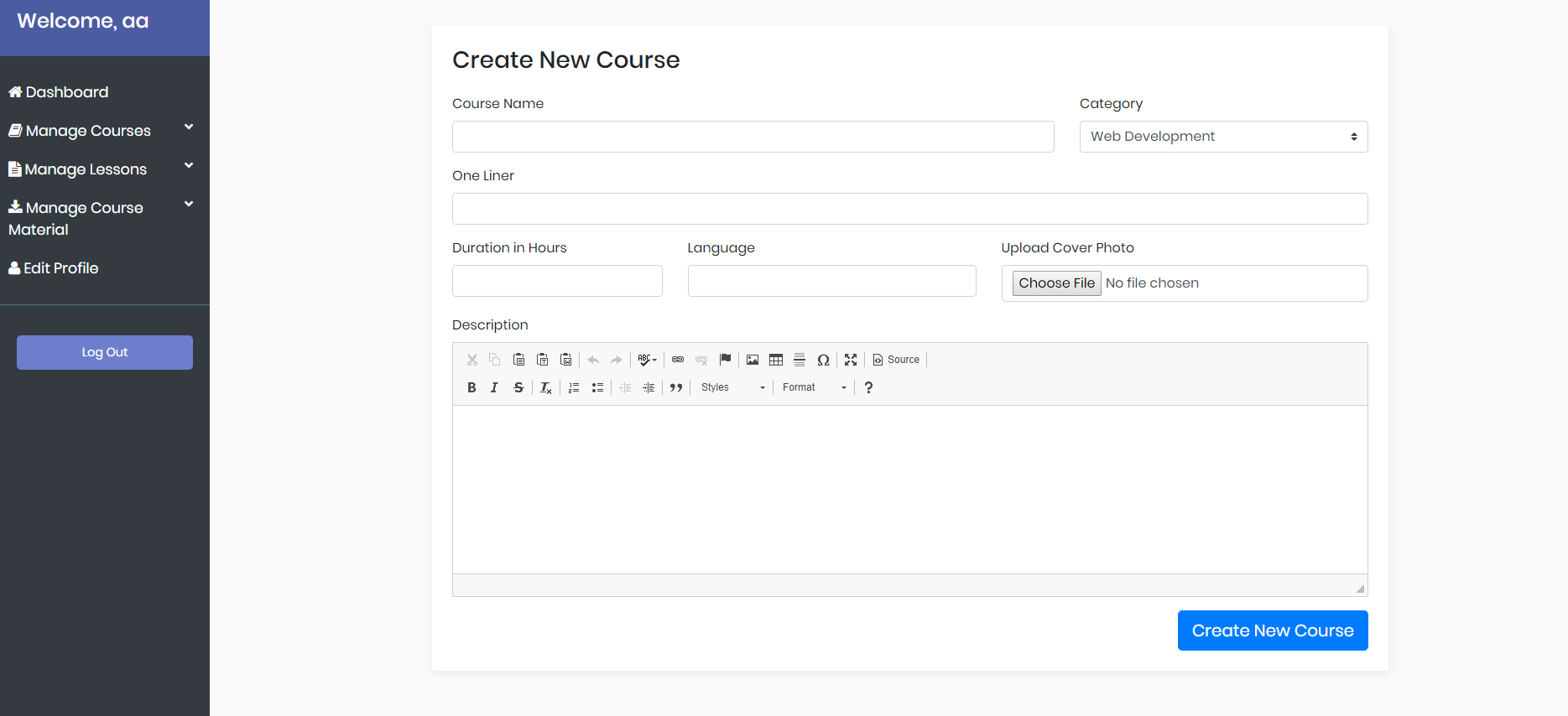
Following screenshot represents the registration page view:

* After the build of base code is completed:

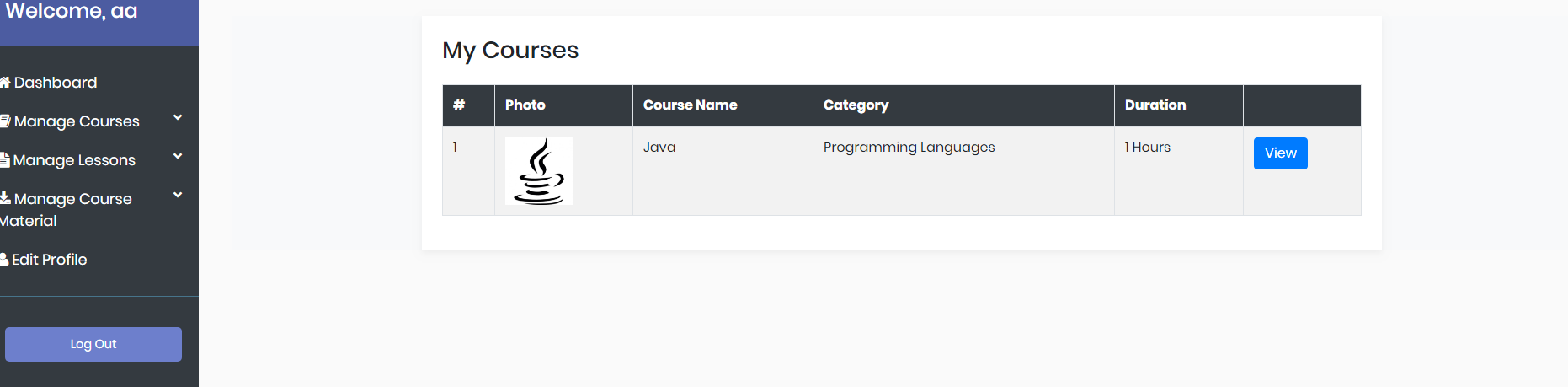


* After the features are developed, the page looks like this:

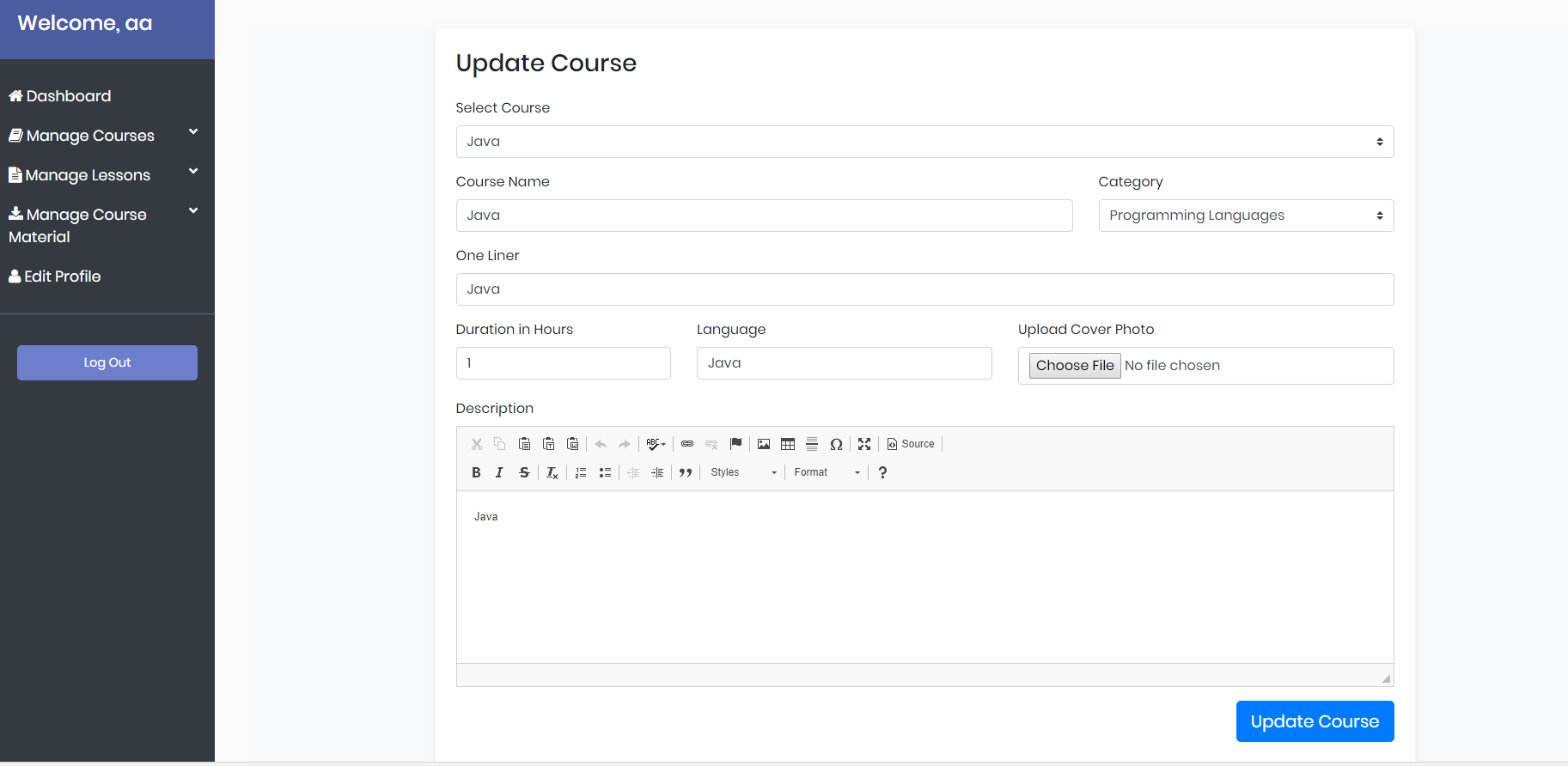
**Create New Course page**



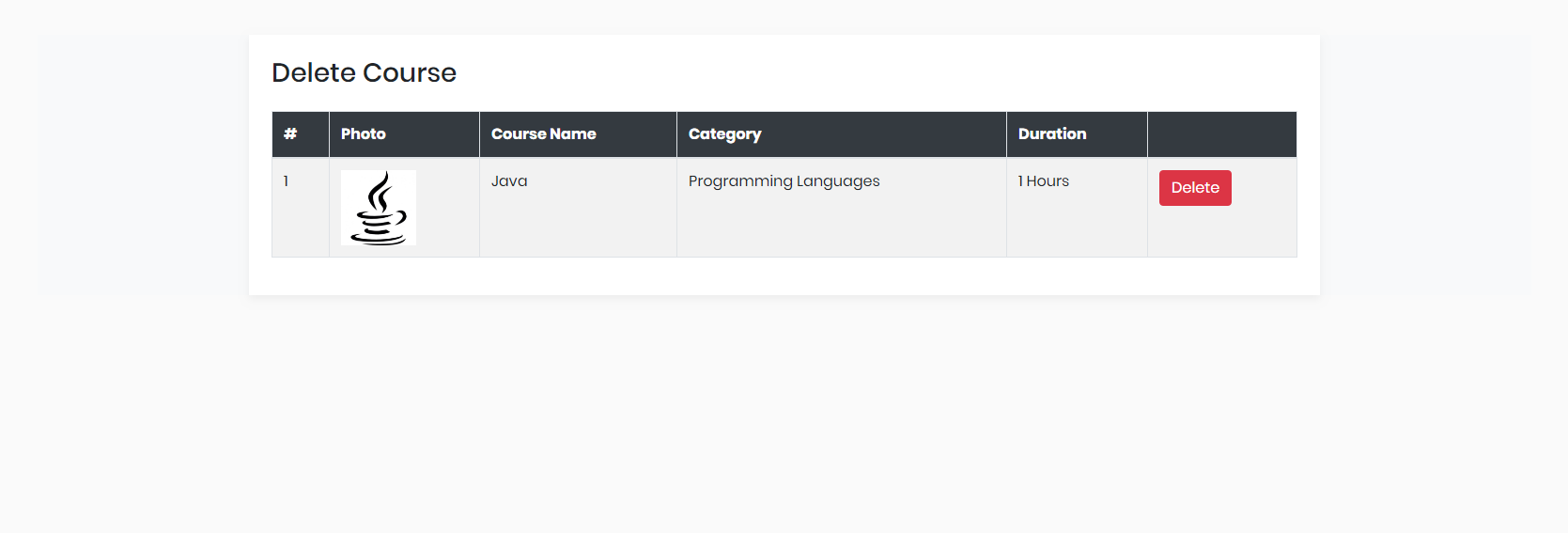
**Dashboard page**



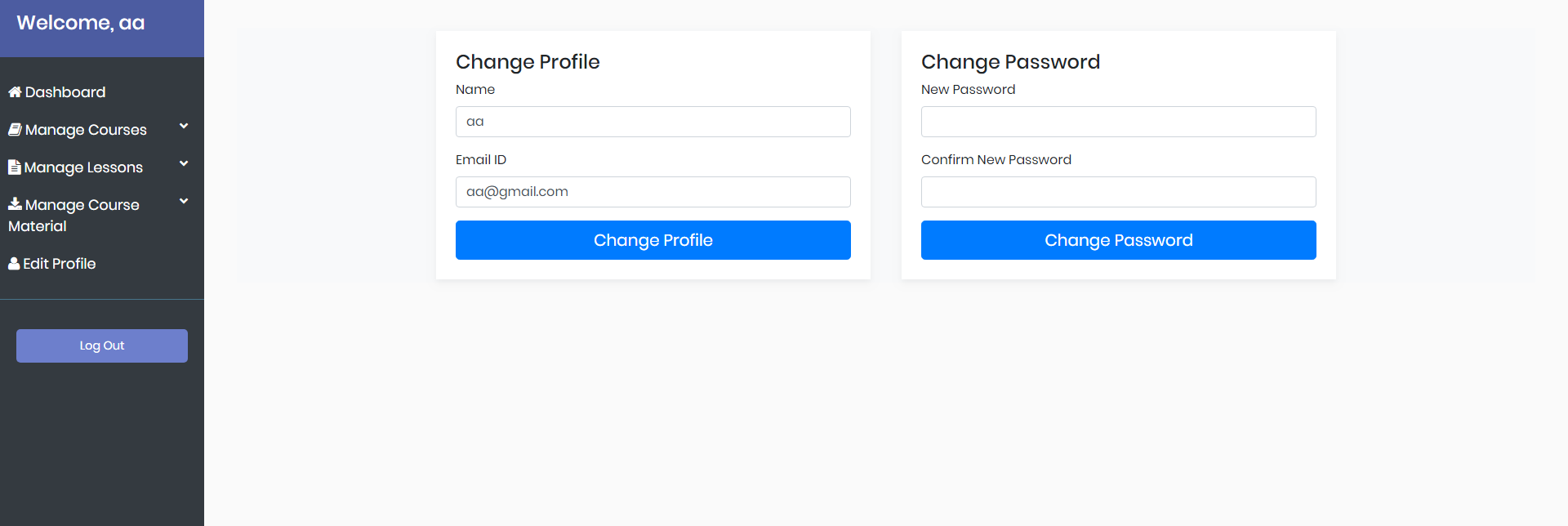
**Update Course page**



**Delete Course page**



**Edit user page**



1. **Application features**

The application should have the following set of built-in features:

**Front-end Component:**

* **Home page**:
  + - The Home page has a button where user can on click it will take user to registration page
* **Registration page**:
  + - In registration page user can register and sign in
    - The dashboard page contains following components: Manage course, Manage Lesson, Manage Course Material, Edit Profile

**API Component:**Manage Lesson, Manage Course Material.

1. **Project deliverables**

Below are the deliverables:

**Front-end Component** :HTML and CSS code for the **Registration** and **login page**,

**Manage Course** , **edit Profile** is provided ,Implement the missing logic using

JavaScript

**fileNames**:

register.ejs

createCourse.ejs

updateCourse.ejs

deleteCourse.ejs

editProfile.ejs

index.js

Make the given files to make the application work properly

The user can register in two ways:

1) As an instructor

2) As a student

Registration and login process should be working

Course creation, update, deletion should be implemented accordingly

Implement the following logics

1. Implement login using the Post method (send Email id and password)

Example request body should be in:

**{emailId: "test@gmail.com", password: "test"}**

1. User registration should be implemented (take the required details and send post request for user registration)

Example request body should be in:

**{“name”: "test", “type”: "1", “emailId”: "test@gmail.com", “password”: "aaa"}**

Type 1 for instructor:

**{“name”: "test", “type”: "2", “emailId”: "test@gmail.com", “password”: "aaa"}**

Type will be 2 for student:

1. Implement create course functionality which is a post method refer stubs for further information:

**{**

**"name": "Java",**

**"category": "Programming Languages",**

**"oneLiner": "Java",**

**"duration": "1",**

**"language": "Java",**

**"description": "%3Cp%3EJava%20..%3C%2Fp%3E%20",**

**"lessons": [],**

**“photo”:”photo blob text”**

**}**

1. Implement update course functionality which is a post method refer stubs for further information:

**{**

**"id": "5b6c5a67e8e0a03b40e584e7",**

**"name": "Java",**

**"category": "Programming Languages",**

**"oneLiner": "Java",**

**"duration": "1",**

**"language": "Java",**

**"description": "%3Cp%3EJava%3C%2Fp%3E%20",**

**"lessons": [],**

**“photo”:”blob data”**

**}**

1. Implement delete course functionality which is a post method refer stubs for further information.
2. Implement Javascript logic for editProfile.