### Project Report on

# E-Learning System for an Employee



## By

**Deepankar (201500093)**

**Arijeet Acharyya (201500115)**

*In partial fulfillment of requirements for the award of degree in*

Bachelor of Technology in Computer Science and Engineering

(2019)



##### Under the Project Guidance of

**Rohini N., Systems Engineer, Infosys Limited**

And

Internal Reviewer

**Mr. Suman Kalyan Kar, Assistant Professor-I, SMIT**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY

## (A constituent college of Sikkim Manipal University)

MAJITAR, RANGPO, EAST SIKKIM – 737136

**PROJECT COMPLETION CERTIFICATE**

This is to certify that the below mentioned student(s) of Sikkim Manipal Institute of Technology has / have worked under my supervision and guidance from 28th January, 2019 to 17th May, 2019 and has / have successfully completed the project entitled “E-Learning System for an Employee” in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

|  |  |  |
| --- | --- | --- |
| University Registration No | Name of Student(s) | Course |
| 201500093 | Deepankar | B.Tech (CSE) |
| 201500115 | Arijeet Acharyya | B.Tech (CSE) |

Rohini N.

Systems Engineer

Infosys Limited

Mysore

**PROJECT REVIEW CERTIFICATE**

This is to certify that the work recorded in this project report entitled “E-Learning System for an Employee” has been jointly carried out by Mr. Deepankar (Reg. 201500093), Mr. Arijeet Acharyya (Reg. 2015001115) of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering. This report has been duly reviewed by the undersigned and recommended for final submission for Major Project Viva Examination.

Mr. Suman Kalyan Kar

Assistant Professor - I

Department of Computer Science & Engineering

Sikkim Manipal Institute of Technology

Majitar, East Sikkim – 737136.

**CERTIFICATE OF ACCEPTANCE**

This is to certify that the below mentioned student(s) of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology (SMIT) has / have worked under the supervision of Ms. Rohini N. of Infosys Limited from 28th January, 2019 to 17th May, 2019 on the project entitled “E-Learning System for an Employee”

The project is hereby accepted by the Department of Computer Science & Engineering, SMIT in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

|  |  |  |
| --- | --- | --- |
| University Registration No | Name of Student(s) | Project Venue |
| 201500093 | Deepankar | Infosys, Mysore |
| 201500115 | Arijeet Acharyya | Infosys, Mysore |

Dr Kalpana Sharma

Professor & HOD

Computer Science & Engineering Department

Sikkim Manipal Institute of Technology

Majhitar, Sikkim – 737136

**DECLARATION**

We, the undersigned, hereby declare that the work recorded in this project report entitled “E-Learning System for an Employee” in partial fulfillment for the requirements of award of B.Tech (CSE) from Sikkim Manipal Institute of Technology (A constituent college of Sikkim Manipal University) is a faithful and bonafide project work carried out at Mysore under the supervision and guidance of Ms. Rohini N. of Infosys Limited, Mysore.

The results of this investigation reported in this project have so far not been reported for any other Degree / Diploma or any other Technical forum.

The assistance and help received during the course of the investigation have been duly acknowledged.

…………………

Deepankar (Reg. 201500093)

………………….

Arijeet Acharyya (Reg. 201500115)

**ACKNOWLEDGEMENT**

The success and final outcome of this project require a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I am profoundly grateful to Ms. Rohini N., Systems Engineer, Infosys Limited, Mysore my external project guide for his valuable guidance, help, suggestions and assessment on his busy schedule.

I owe my deep gratitude to my project guide Mr. Suman Kalyan Kar, who took keen interest on my project work and guided me till the completion of my project work by giving necessary information.

I express my gratitude to Dr. Kalpana Sharma, Professor and HOD Computer Science Engineering Department for her support and guidance.

I heartily thank the project coordinators for their guidance and suggestions during project work.

I am thankful and fortunate enough to get constant encouragement, support and guidance from all teaching staffs of CSE, SMIT which helped in successfully completing the project work.

Furthermore, I would like to thank my colleagues in developing the project and people who have willingly helped me out of their abilities.

…………………

Deepankar (201500093)

………………….

Arijeet Acharyya (201500115)

DOCUMENT CONTROL SHEET

|  |  |  |
| --- | --- | --- |
| 1 | Report No | CSE/Major Project/External /B.Tech/INFY 7 & 8/2019 |
| 2 | Title of the Report | E-Learning System for an Employee |
| 3 | Type of Report | Technical |
| 4 | Author(s) | Deepankar  Arijeet Acharyya |
| 5 | Organizing Unit | Infosys Limited |
| 6 | Language of the Document | English |
| 7 | Abstract | A web based application deals with online training program for employees of the company. The application basically lists out the courses available and time in which courses has to be completed. The company can assess the performance of each employee by conducting online exam for the course. |
| 8 | Security Classification | Restricted |
| 9 | Distribution Statement | Restricted |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter** | **Title** | **Page No.** |
|  |  |  |
| 0 | Abstract | 7 |
|  |  |  |
| 1 | Introduction | 8 |
|  |  |  |
|  | 1.1 General Overview of the Problem | 8 |
|  |  |  |
|  | 1.2 Literature Survey | 9 |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Name** | **Page No.** |
|  |  |  |
| 1.1 | Wireless Architecture | 8 |
|  |  |  |
| 2.1 | Link Networks | 25 |
|  |  |  |
| 2.2 | Firewall | 32 |
|  |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Table Name** | **Page No.** |
|  |  |  |
| 1.1 | Characteristic of Wireless Network | 10 |
|  |  |  |
| 6.1 | Performance chart of Cluster based routing | 45 |
|  |  |  |
|  |  |  |
|  |  |  |

**ABSTRACT**

**ABSTRACT**

E-learning refers to all electronic learning through systems that are used as part of the learning system. It essentially includes learning online through the courses that are offered on the net. E-learning allows anyone to access educational materials virtually from anywhere and anytime as per the ease of the respective user.

E-learning has come to dominate the strategic thinking of employers over the past decade in several countries. Only in recent years in India, there has been focus on e-learning, this is due to the recent development in MNC and IT concerns.

E-learning is required in IT organizations to keep the employees up-to-date on industry development which is of utmost importance. The importance of e-learning is now a given fact and it can offer an alternative that is much faster, cheaper and potentially better.

**INTRODUCTION**

A web application is a software application that runs on a remote server. In most cases, web browsers are used to access web applications, over a network, such as the internet. Some web applications are used in intranets, in companies and schools, for processing the internal works and performing specific functionalities like attendance, work sharing etc. Web applications are different from other applications because they do not need to be installed. Web applications are popular because all of the computer operating systems are compatible to the web browsers that are used to access the application.

AngularJS is a structural framework for dynamic web apps. It uses HTML as the template language and also extends HTML’s syntax to express the application components clearly and succinctly. Angular JS’s data binding and dependency injection eliminate much of the code that otherwise the user would have to write, and it all happens within the browser, making it an ideal partner with any server technology. AngularJS attempts to minimize the impedance mismatch between document centric HTML and what an application needs by creating new HTML constructs helping to build the usable application in the easiest way known.

A web based application related to this project deals with the online training program for employees of a company. As it is an intra E-learning site within an organization, only admin is authorized to register a user and then the users can login and perform different functionalities like searching and registering for a course, editing profile and viewing performance graph. A suitable database is maintained involving entities and maintaining relationships between them. The role based login and authentication is maintained to provide encapsulation for designated information accessibility for designated roles.

The data is fetched from and into the database using entity framework and web services. Angular framework is used to build the single page application to provide the view for the data. The component of the angular framework provides the view for the data that it requests to the service method of the angular application. The service method then calls the suitable API method of the web services using *HttpClient* and the interaction with the database is done via suitable data access layer method which is invoked by the API method. The data required by the angular application is fetched by the services of the angular application with the help of *HttpClient* and *Observables*. *HttpClient* helps angular applications to communicate with back end services to fetch or persist data. To increase the performance, the responses and requests are sent asynchronously with the help of *Observables.*

This application basically lists out the courses available and time in which the courses has to be completed. New courses can be added with time and the course can also be modified, updated and removed. The role based login and authentication makes it easier to separate functionalities between different sets of users. It even makes the task of the company easier to review the courses and the training done by the employees. The company can assess the performance of each employee by conducting online exam for the course.

**Problem Statement:**

* The project is to develop an E-learning system for an employee.
* In the present context providing a training and making employee project ready is a tedious task and involves time and resource.
* The process of evaluation of assessment and course review is required but is not available.

**Solution Strategy:**

* A database is maintained involving entities and relationships between them.
* The data is fetched from and into the database using entity framework and web services.
* The data is fetched from and into the database using entity framework and web .

**IMPLEMENTATION DETAILS**

The implementation details for different modules in this application are listed below

**1. Registration and Login:**

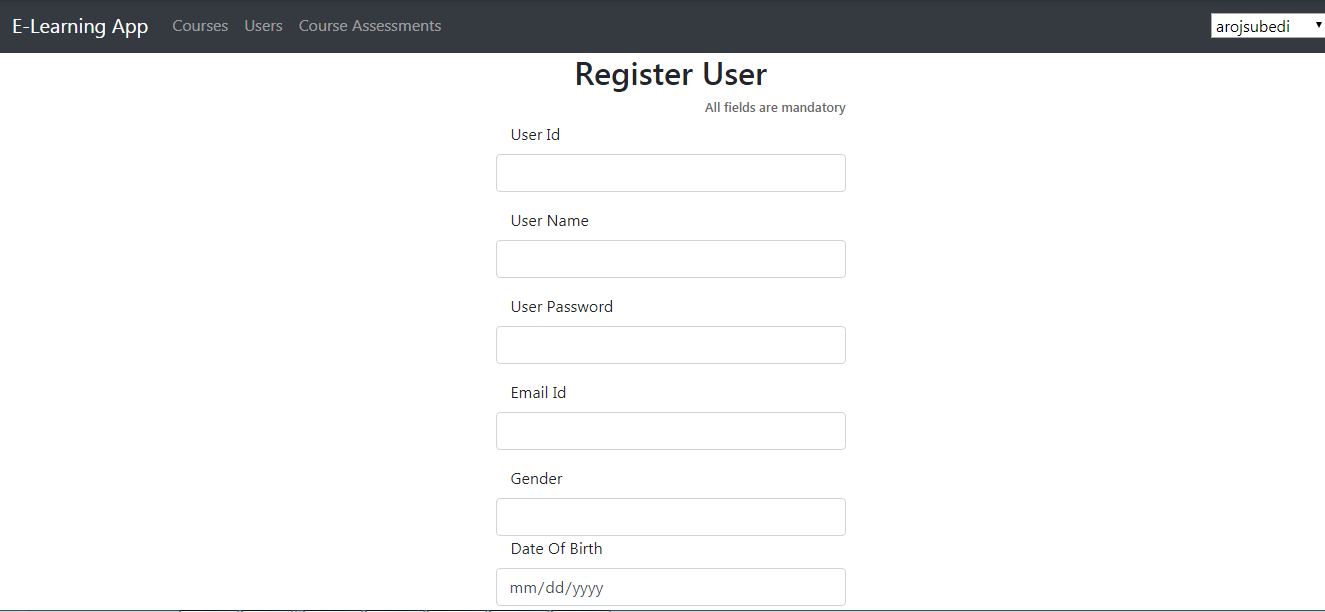
As it is an intra E-learning site within an organization, only the admin is authorized to register a user and then the users can login and update their profile. The registration form filled is posted to the database by passing the data from the service method of the angular to the API method of web service and then to the suitable register method of the data access layer. For login, the user details are fetched. The component of the angular application requests to the service method and after the service method gets the data from the web service, it passes the data to the component that has subscribed for the data. The data are then validated with the user entry to redirect to the home page based on the roles thus, role based login and authorization.

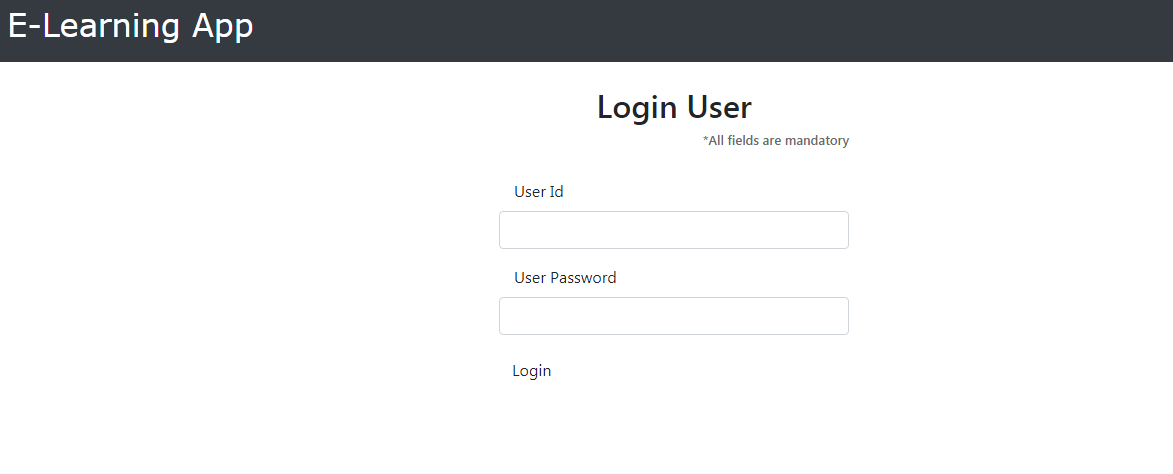
Pseudocode for Register:

1. Admin entries the suitable data in the register form.
2. The form data is passed to the *register\_user* method present in the service of the application.
3. The service method with the help of *HttpClient* makes the *post* request which is placed to the suitable url of the API method.
4. The called API method then in turn calls the suitable method of the data access layer which entries the form data to the *Users* table of the database.

Pseudocode for Login:

1. The user entries the UserId and Password in the login form.
2. The form data is passed to the Login Component.
3. The component calls the *get\_all\_user* method present in the service of the application.
4. The service method with the help of *HttpClient* makes the *get* request which is placed to the suitable url of the API method.
5. The called API method then in turn calls the suitable method of the data access layer and retrieves *Users* table data of the database and returns to the service method as *Observables*.
6. The data is then fetched to the method of the Login Component that has subscribed for the data.
7. for(i in ListOfUsers):
   1. if i[‘userId’]=form.userId and i[‘password’]=form.password and i[‘status’]=”unblocked”
      1. store the UserId in the session.
      2. Retrieve the role of i[‘roleId’]
      3. Redirect to the home page based on the role id.
   2. else
      1. Display error message



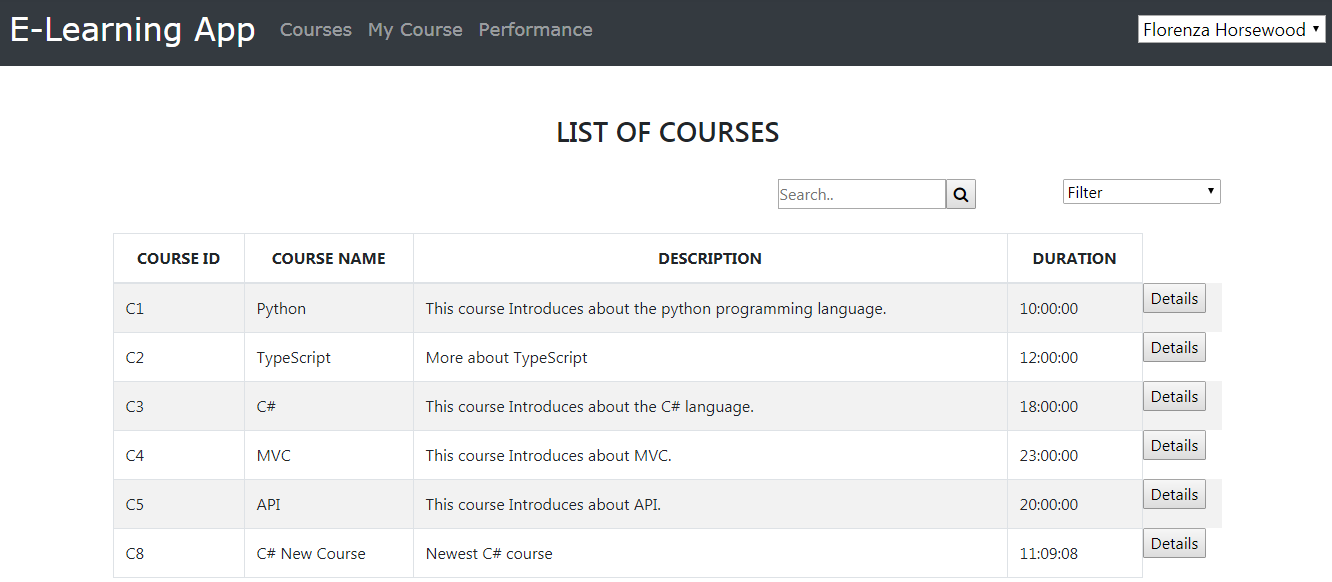


**2. List Courses and View details of a course.**

The course list can be viewed by both the admin and the users. The admin can additionally add or remove course and view the assessment details of that course while the users can view the course details and register for the course. After registering for the course, the course will be added to the ‘My Course’ section of that user.

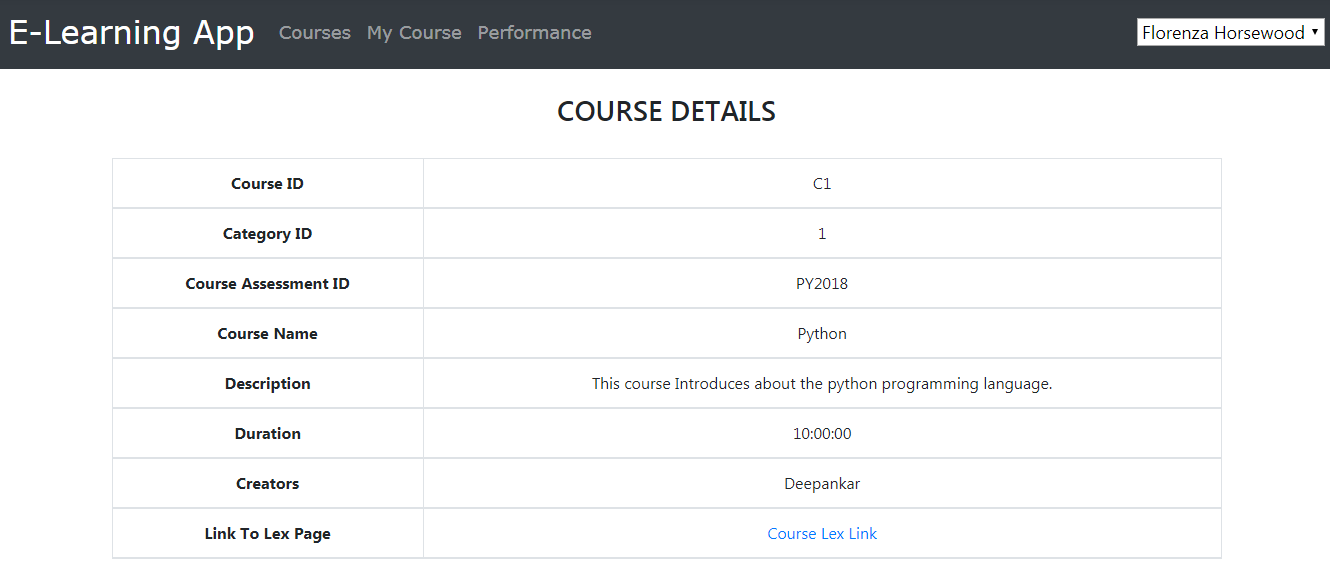
Pseudocode for listing courses:

1. The component calls the *get\_courses* method present in the service of the application.
2. The service method with the help of *HttpClient* makes the *get* request which is placed to the suitable url of the API method.
3. The called API method then in turn calls the suitable method of the data access layer and retrieves *Courses* table data of the database and returns to the service method as *Observables* and the data is passed to the view.



Pseudocode for viewing course details:

1. The component passes the *CourseId* as parameter to the *show\_course\_details* method present in the service of the application.
2. The service method with the help of *HttpClient* makes the *get* request which is placed to the suitable url of the API method.
3. The called API method then in turn calls the suitable method of the data access layer and retrieves entry matching with the *CourseId* from *Courses* table of the database and returns to the service method as *Observables* and the data is passed to the view.

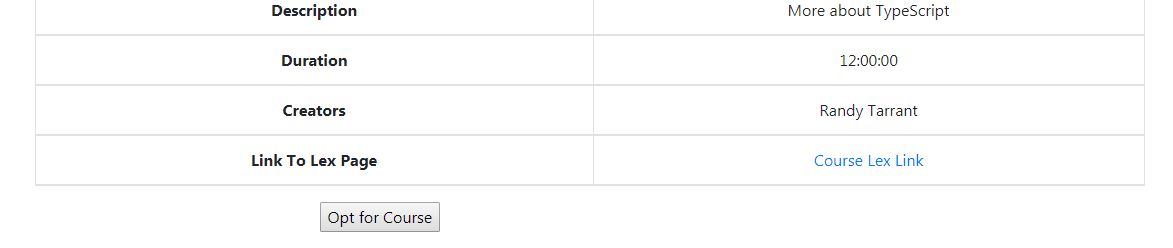


**3. Registering for a course**

Only the users are able to register for a course. Users click on ‘Opt for Course’ in the View Details section of the course.

Pseudocode for registering for a course

1. The course object to register is passed as a parameter to *RegisterCourse* method in the ListCourses component.
2. UserId is retrieved from the session storage.
3. The UserId and the course object is passed as a parameter to the *register\_cours*e method present in the service of the application.
4. The service method with the help of *HttpClient* makes the *post* request which is placed to the suitable url of the API method.
5. The called API method then in turn calls the suitable method of the data access layer which entries the passed data to the *MapUsersCourses* table of the database and sets the *Completion* column to *NotStarted* state by default.

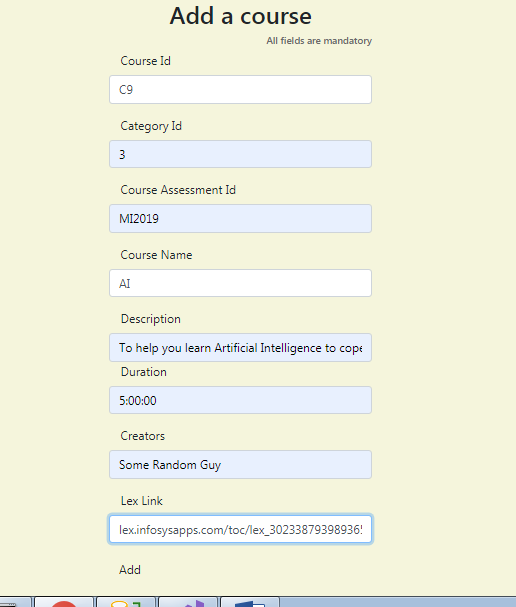


1. **Adding and removing courses**

Only admin is authorized to add and remove the courses.

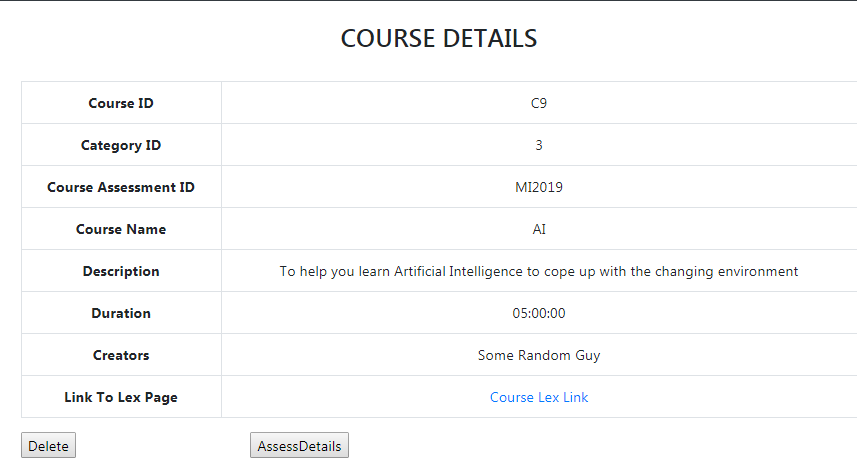
Pseudocode for adding a course

1. The admin entries the suitable data in the AddNewCourse form.
2. The form data is passed to the *add\_course* method present in the service of the application.
3. The service method with the help of *HttpClient* makes the *post* request which is placed to the suitable url of the API method.
4. The called API method then in turn calls the suitable method of the data access layer which entries the form data to the *Courses* table of the database.



Pseudocode for removing a course

1. The *course* object to remove is passed as a parameter to *DeleteCourse* method in the ListCourses component.
2. The method passes the *course* object to the *delete\_cour*se method present in the service of the application.
3. The service method with the help of *HttpClient* makes the *delete* request which is placed to the suitable url of the API method.
4. The called API method then in turn calls the suitable method of the data access layer which removes the matched entry from the *Courses* table of the database.

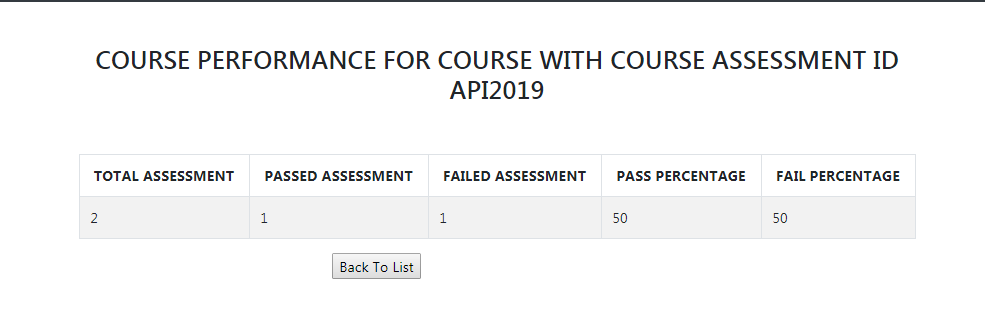


1. **Viewing assessment details of a course**

Only the admin can view the assessment details for a course. This module helps the company assess the performance of users in certain courses by looking at the pass and fail percentages.

Pseudocode for viewing assessment details of a course

1. The course object is passed as a parameter to ViewAssessmentDetails method of ListCourses component.
2. The method passes the *course* object to the *get\_cour*se\_*assessment* method present in the service of the application.
3. The service method with the help of *HttpClient* makes the *get* request which is placed to the suitable url of the API method.
4. The called API method then in turn calls the suitable method of the data access layer which fetches the matched entry from the *CourseAssessments* table of the database.
5. passPercentage => (totalCorrectAssessment / totalAssessment) \* 100;
6. failPercentage => (totalFailedAssessment / totalAssessment) \* 100;



1. **List Users and View details of a user**

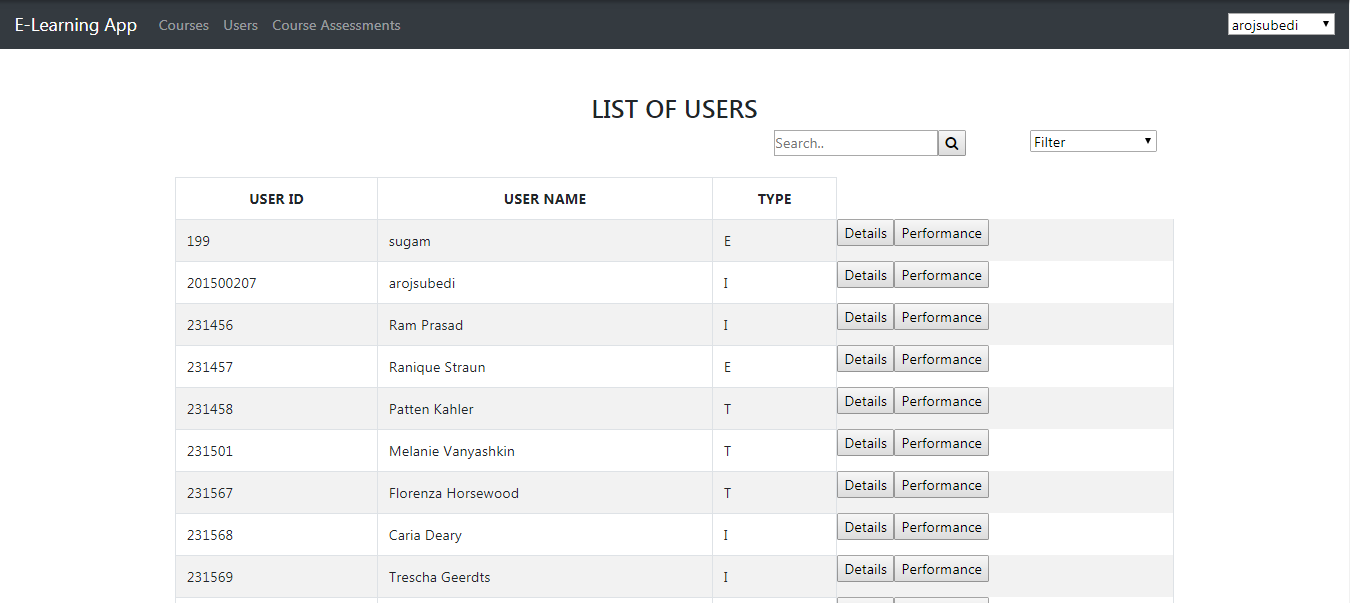
Only admin is authorized to view the list of users and details of a user. Additionally, the admin can block, unblock or delete a user.

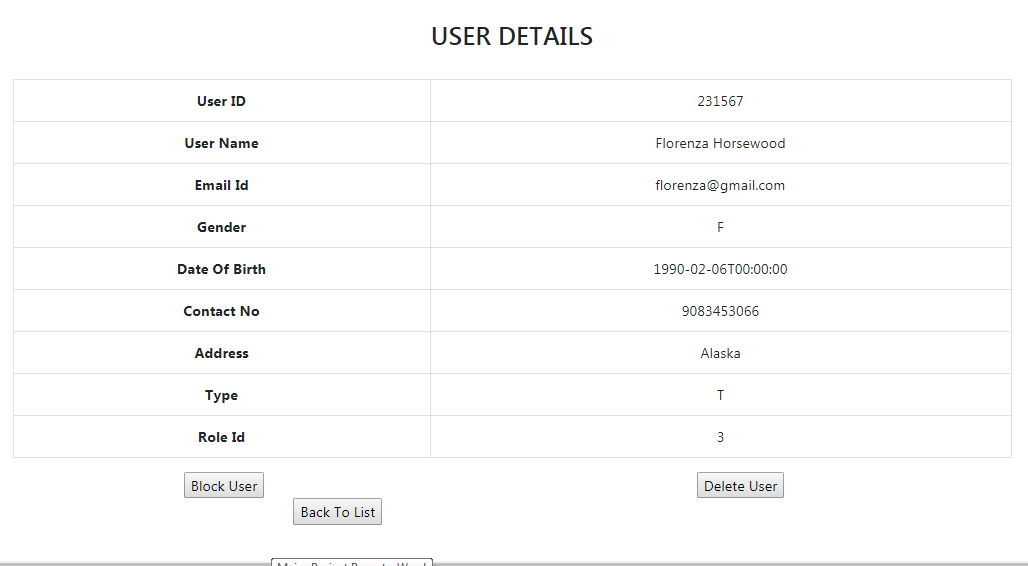
Pseudocode for listing all users

1. The component calls the *get\_users* method present in the service of the application.
2. The service method with the help of *HttpClient* makes the *get* request which is placed to the suitable url of the API method.
3. The called API method then in turn calls the suitable method of the data access layer and retrieves *Users* table data of the database and returns to the service method as *Observables* and the data is passed to the view.

Pseudocode for viewing user details

1. The component passes the *UserId* as parameter to the *show\_user\_details* method present in the service of the application.
2. The service method with the help of *HttpClient* makes the *get* request which is placed to the suitable url of the API method.
3. The called API method then in turn calls the suitable method of the data access layer and retrieves entry matching with the *UserId* from *Users* table of the database and returns to the service method as *Observables* and the data is passed to the view.



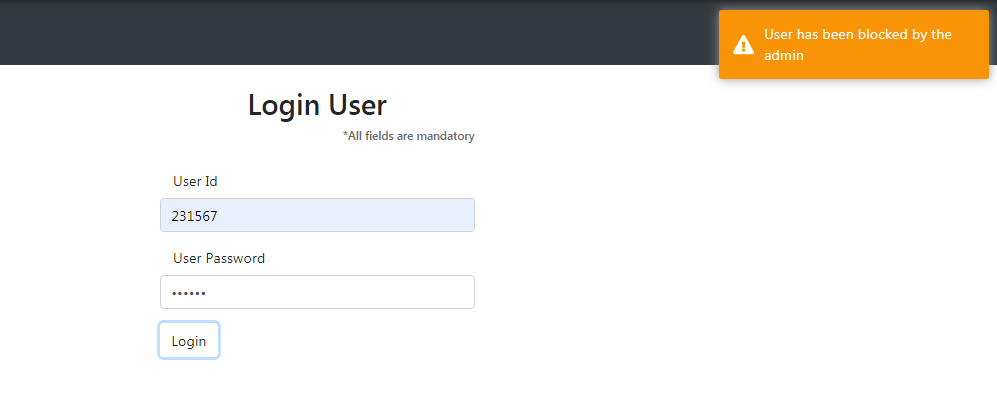


1. **Blocking, Unblocking and Deleting a user**

Only the admin is authorized to block, unblock and delete a user.

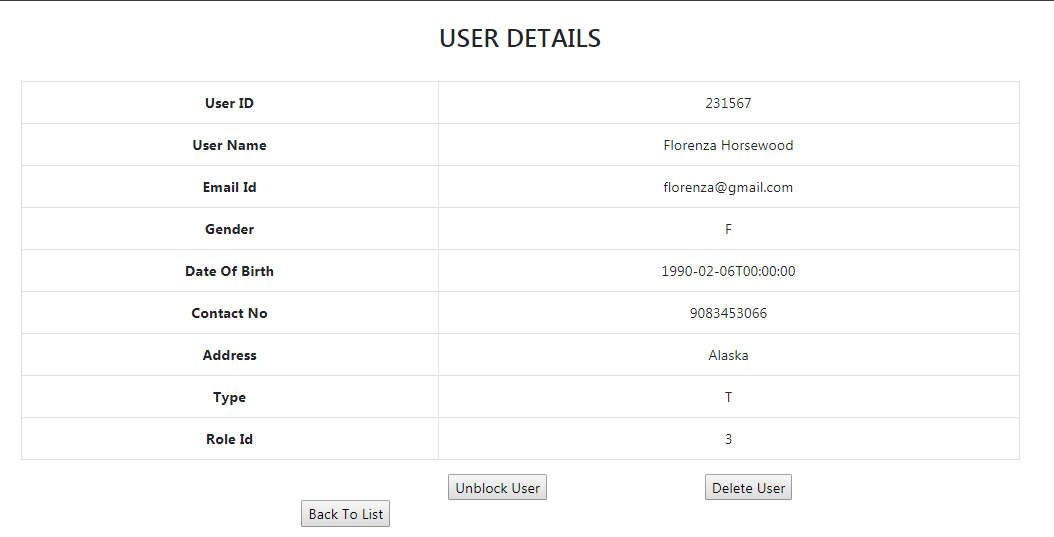
Pseudocode for blocking a user

1. The *user* object is passed as a parameter to the *BlockUser* method in the ListUsers component.
2. The *Status* attribute of the *user* object is updated to the *Blocked* state.
3. The component passes the *user* object as parameter to the *update\_user* method present in the service of the application.
4. The service method with the help of *HttpClient* makes the *put* request which is placed to the suitable url of the API method.
5. The called API method then in turn calls the suitable method of the data access layer and updates the entry matching with the *UserId* from *Users* table of the database.



Pseudocode for unblocking a user

1. The *user* object is passed as a parameter to the *UnblockUser* method in the ListUsers component.
2. The *Status* attribute of the *user* object is updated to the *Unblocked* state.
3. The component passes the *user* object as parameter to the *update\_user* method present in the service of the application.
4. The service method with the help of *HttpClient* makes the *put* request which is placed to the suitable url of the API method.
5. The called API method then in turn calls the suitable method of the data access layer and updates the entry matching with the *UserId* from *Users* table of the database.

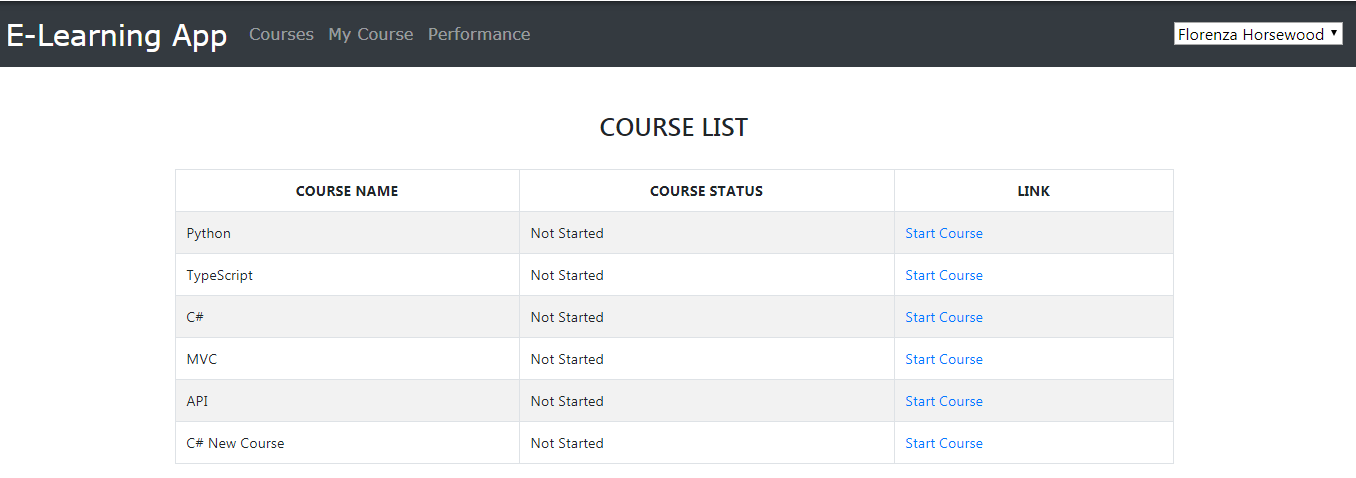
****

Pseudocode for removing a user

1. The *user* object to remove is passed as a parameter to *DeleteUser* method in the ListUsers component.
2. The method passes the *user* object to the *delete\_user* method present in the service of the application.
3. The service method with the help of *HttpClient* makes the *delete* request which is placed to the suitable url of the API method.
4. The called API method then in turn calls the suitable method of the data access layer which removes the matched entry from the *Users* table of the database.
5. **Viewing registered courses, accepting TnC and taking assessments**

Only the users are authorized to view the courses that they have registered for. Before starting the course, the user has to accept Terms and Conditions (TnC) and to take the assessment the user has to complete the course.

Pseudocode for viewing registered courses

1. Fetch the *UserId* from the session storage.
2. The *ViewRegisteredCourses* in the MyCourse component invokes the *get\_my\_course* method present in the service by passing the *UserId* as a parameter.
3. The service method with the help of *HttpClient* makes the *get* request which is placed to the suitable url of the API method.
4. The called API method then in turn calls the suitable method of the data access layer which retrieves the matched entry from the *MapUsersCourses* table of the database, returns as *Observables* to the service method and the data is passed to the view.

**Accepting TnC**

The courses in the *MapUsersCourses* are in different state of completion represented by the numeric value. The following values represents different stages of completion.

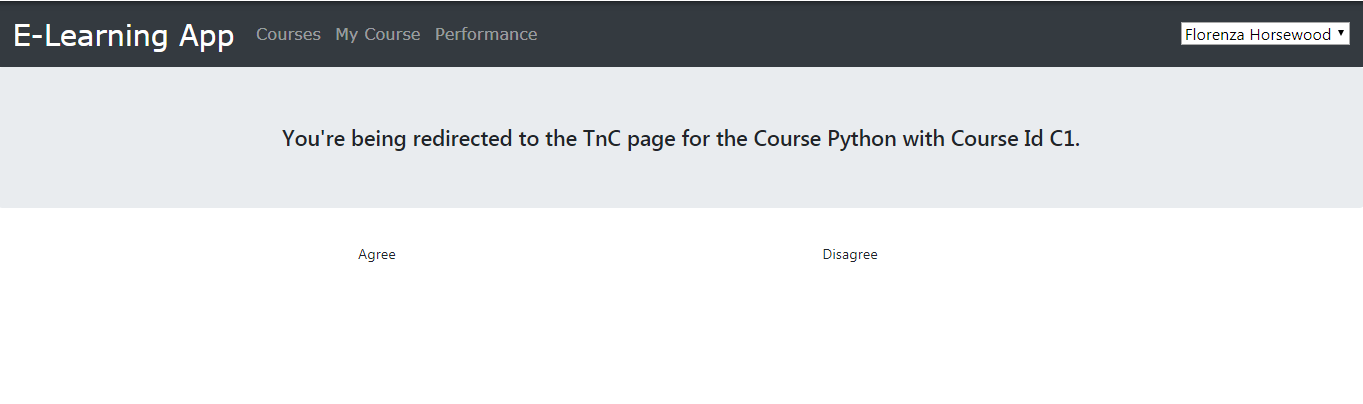
Value 0 – Not Started

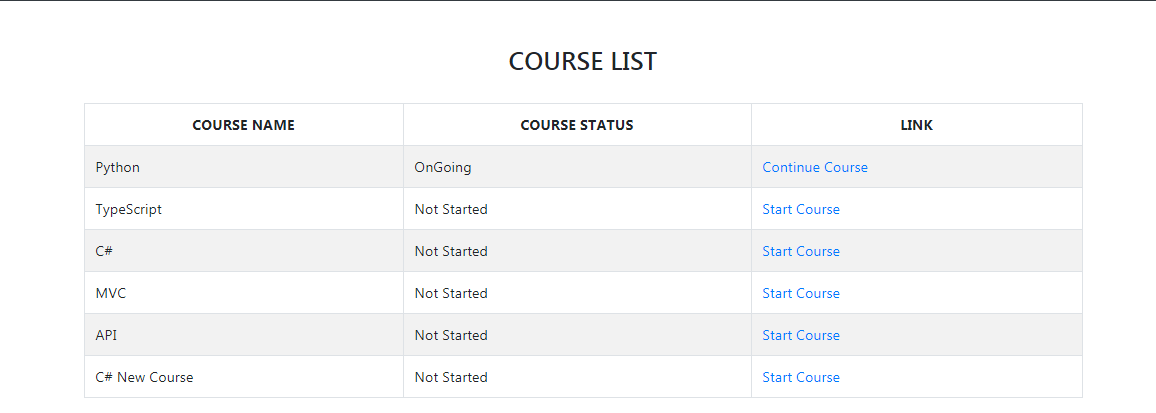
Value 1 – On Going

Value 2 – Completed and Assessment Pending

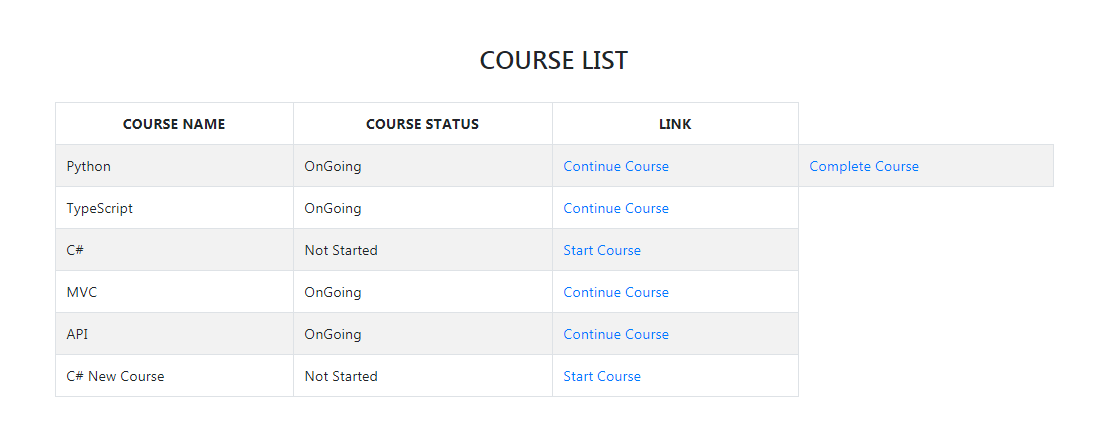
Value 3 – Passed Assessment and Course Completed

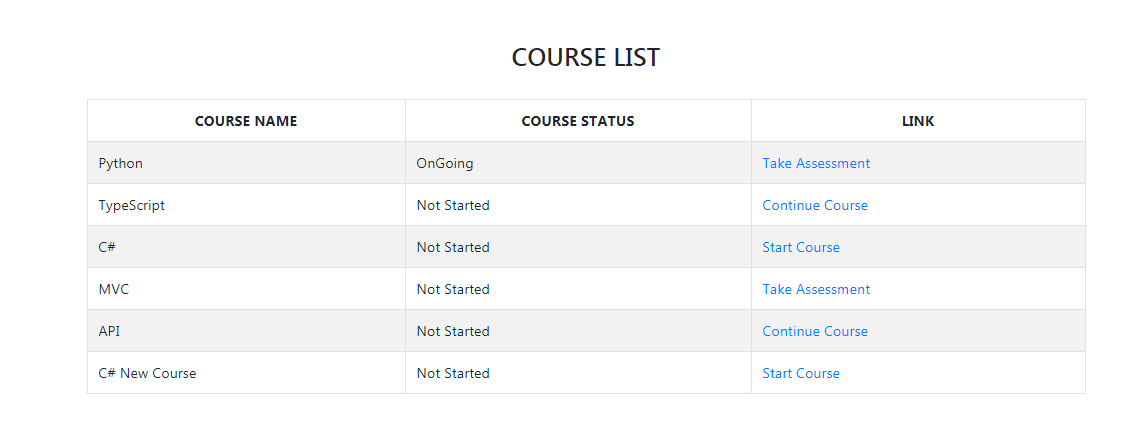
Initially, when a user registers for a course, the *Completion* column in the *MapUsersCourses* for the registered course will have value 0. After accepting/agreeing to the TnC, the value is updated to 1 and the course now is in the OnGoing stage.

****

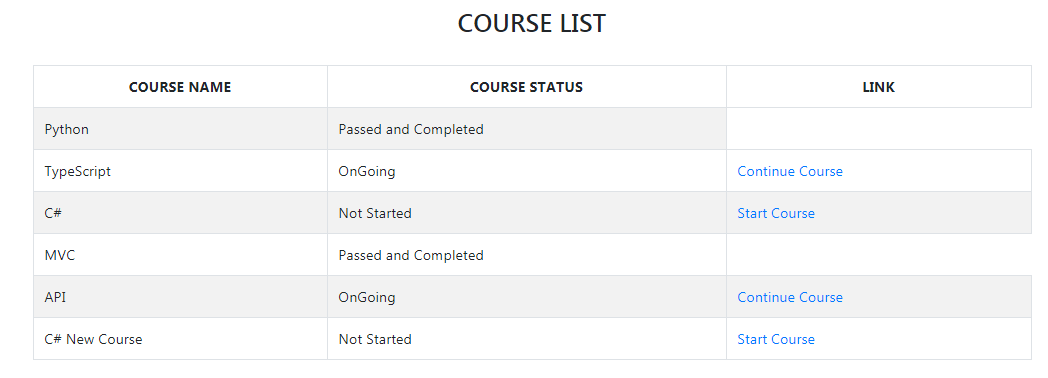
****

**Completing course and taking assessment**

After the user clicks on *Complete Course,* the User has to give an assessment related to the course to pass and complete the course.



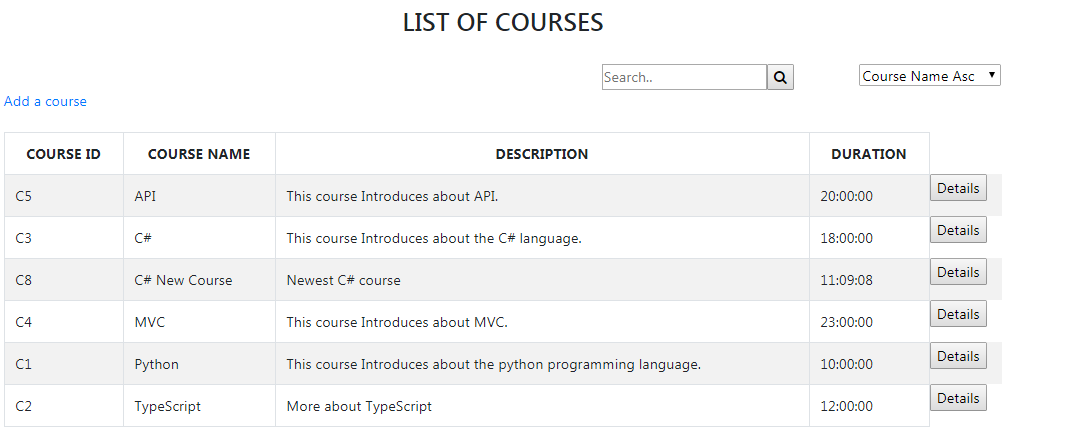
If the user passes the assessment, the state of the course in MyCourse section is changed to *Passed and Completed*.

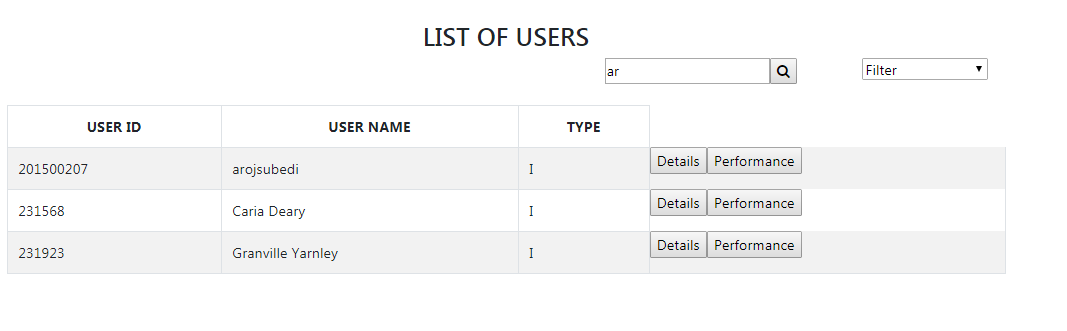
****

1. **Filtering and Searching**

Both the users and admin can filter the list to display the entries in the most appropriate way. The users can also search the entries using the keyword. The search keyword doesn’t need to be exact and matching but should be the part of the string of the searched column. The searching and filtering functionality is present in the ListUsers and as well as ListCourses module. In ListUsers module, the filtering can be done on the basis of *User Id* and *User Name* in both ascending as well as descending order and in ListCourses module, the filtering can be done on the basis of *Course Name* and *Course Id* in both ascending as well as descending order.

In ListUsers module, the search keyword can be either *User Id* or *User Name* and in ListCourses module, the search keyword can be either *Course Id* or *Course Name.* The *User Id* or *User Name* doesn’t need to be match exactly with the database entry.

****



1. **Viewing the performance graph**

The user can view their performance graph while the admin can view the performance graph of each user that have registered for one or more courses. The performance graph is displayed on the basis of completion status of the courses in the MyCourse section of the user. The performance graph is displayed in the form of Line Chart and Bar Chart.

****