

## ASSIGNMENT 2 FRONT SHEET

<b>Qualification</b>	<b>BTEC Level 5 HND Diploma in Computing</b>		
<b>Unit number and title</b>	Unit 30: Application Development		
<b>Submission date</b>	24/02/2021	<b>Date Received 1st submission</b>	
<b>Re-submission Date</b>		<b>Date Received 2nd submission</b>	
<b>Student Name</b>	Truong Vi Kiet	<b>Student ID</b>	GCD191129
<b>Class</b>	GCD0704	<b>Assessor name</b>	Hoang Nhu Vinh
<b>Student declaration</b> <p>I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.</p>			
		<b>Student's signature</b>	

### Grading grid

P4	P5	P6	M3	M4	M5	D2	D3

☐ **Summative Feedback:**

☐ **Resubmission Feedback:**

**Grade:**

**Assessor Signature:**

**Date:**

**Lecturer Signature:**

## Table of Contents

I. Introduction .....	3
II. Peer Review and Feedback Analysis (P4) .....	3
1. Peer Review: .....	3
2. Feedback Analysis: .....	4
III. Application Development based on specified business problem (P5) .....	5
1. Login and Dashboard .....	5
2. Trainer Management .....	6
3. Trainee Management .....	8
4. Course Management .....	11
5. Course Category Management .....	13
IV. Application Evaluation (P6) .....	15

## Table of Figures

Figure 1 Login Page .....	5
Figure 2 Training Staff dashboard .....	6
Figure 3 All Trainer page.....	6
Figure 4 Edit Trainer .....	7
Figure 5 All Course of Trainer page .....	7
Figure 6 Assign Course to Trainer .....	8
Figure 7 Reassign Trainer to a Different Course .....	8
Figure 8 All Trainee page .....	9
Figure 9 Create Trainee .....	9
Figure 10 Edit Trainee .....	10
Figure 11 All Course of Trainee.....	10
Figure 12 Assign Course to Trainee.....	11
Figure 13 Reassign Trainee to Different Course .....	11
Figure 14 All Course page .....	12
Figure 15 Create Course .....	12
Figure 16 Edit Course .....	13
Figure 17 All Category page.....	13
Figure 18 Create Course Category .....	14
Figure 19 Edit Course Category .....	14
Figure 20 Application's Google Lighthouse score .....	15
Figure 21 Login function performance test .....	16
Figure 22 View All Trainer performance test.....	17
Figure 23 Create Trainer performance test .....	18
Figure 24 Edit Trainer performance test .....	19
Figure 25 Delete Trainer performance test .....	20
Figure 26 Logout performance test .....	21

## I. Introduction

In the previous documents, we have analyzed the user requirements for Academic Portal Application (APA) in order to have a clear idea about how we should develop the web application. Furthermore, we have already discussed about various technologies and tools, and identified what we have used in the process of designing and developing APA. In this document, we will go through the application development process and review the application in various criteria.

## II. Peer Review and Feedback Analysis (P4)

### 1. Peer Review:

Peer reviews are needed in order to have a more thorough analysis on the Application Development. Insights and ideas from other developers and testers may give us an opportunity to discover other features and improvements for versions to come. Therefore, we have designed a questionnaire for in order to collect the peer reviews from other developers in the IT department. Below are ideas and feedbacks that we have collected in the form of questions, combining with our own self reflection on these questions as the answer and we have a summary table as follows:

Problem	Question	Answer
<b>High-level Role Authorization</b>	Why we have two high level roles or “Superuser” (Admin, Training Staff) that have permission to interact to other roles (Trainer, Trainee) in application? Can we merge these roles into one superuser role for convenience?	Because each role has their own management work to carry, we can’t put those into one “superuser”. The supposed workload would be too heavy. It is recommended to split the high level-role’s authorization based on the requirements.
<b>Login attempt restriction</b>	Why were there no restriction on users’ amount of login attempt? That means a user or other people can guess for an account password eventually.	This is something we have realized after reading this peer review. We will set a limitation for the login attempts. We can avoid a brute force password attack as you have mentioned.
<b>Selective criteria search, filter, sort</b>	I see that the Training Staff can only search by Name, what about searching using different criteria, or filter, or sort?	We have realized that this could be a very convenient feature for the application. We will implement that in the later versions.

<b>Trainer/Trainee request for teach a Course</b>	The Trainer/Trainee can only request the Training Staff at the HR department for joining a course. How about letting them request through the app?	We also realized that this would be a very convenient features for both the Training Staff and Trainer/Trainee. We will consider implementing this feature at a later version if we have the capability to.
<b>Application performance</b>	Why the performance of application when deploying is slower than developing section.	The web app's loading time affected by many factors (Internet connection, server connection, etc.). Therefore, it is much slower compared to the local testing process. We will work on ways to improve on our performance.
<b>Application UI</b>	The UI is quite simple and easy to read, but can you improve it a bit cleaner? Some of the alignments still doesn't feel right	Because there are not much about the UI requirements, we have chosen to go with the more simplified UI build utilizing what we have in Visual Studio. We will try to tinker with what we have and improve the UI at the later versions.

## 2. Feedback Analysis:

Through the summary table above, we have found out various ideas that we can use to improve the Academic Portal Application. Those ideas are listed as follows:

- Login attempts limit: we can implement a limit on the amount of attempts each user can login and impose a lock on that account for a certain amount of time. This will prove effective on preventing password brute force attack.
- Request to Join Course from Trainer/Trainee: Trainer/Trainee can send a request through the APA system and Training Staff can approve. This is a very convenient and effective feature. However, judging by the skill of the current team members, this function is hard to implement. The team will ask for advice for a more skilled senior in the department and come up with a solution at a later version
- Selective criteria search, filter and sort: these features may become a very effective tool for Admins and Trainers in their management work. The necessity for this function will grow along with the growth of APA's amount system users. This function will eventually be implemented in versions to come.
- Application performance improvements: We realized that we can find a hosting service from a nearer location to us and improve the APA's solution structure and merge some of the views in the MVC may improve the loading time and performance of the application.

- UI improvements: the current basic UI can be considered satisfactory according to the UI requirements. However, the UI can be further cleaned and improved, along with possibility of developing responsive UI to adapt to various devices aside from PCs and Laptop.

### III. Application Development based on specified business problem (P5)

Based on the Software Requirement Specification, we had developed the Academic Portal Application (APA) for assisting in managing FPT Co.'s internal training programs. The development team have split the work for two members that is in charge of the project. This document will only cover the application's Training Staff functions. The report on Admin, Trainer and Trainee role will be covered by the other member. In this section, we will present the features a Training Staff account has when using Academic Portal App. There will be explanations coupled with the screenshots of the app's UI in the following paragraph.

#### 1. Login and Dashboard

First, we have the login screen, which is a function shared by all 4 roles in the APA's system.

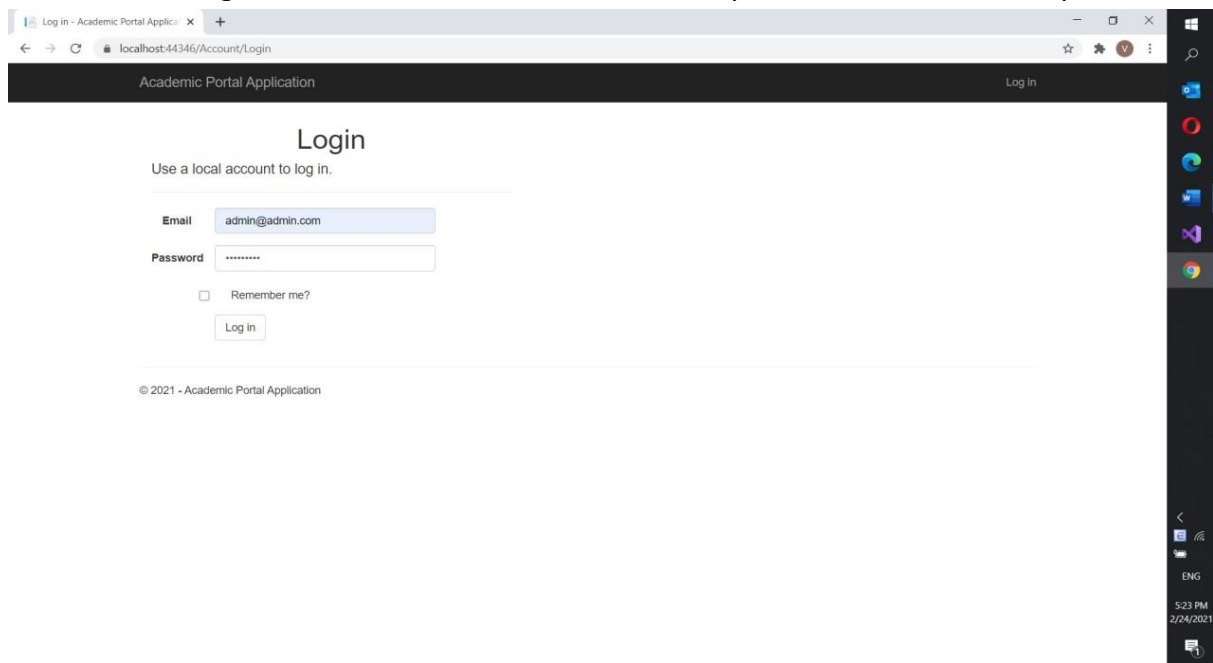


Figure 1 Login Page

After Training Staff has logged in, he/she will be redirected to the Dashboard page. Currently, there's not much aside from a welcome sentence, more function of the dashboard will be developed at the later versions. Furthermore, Training Staff can log out whenever he/she want by clicking Logoff

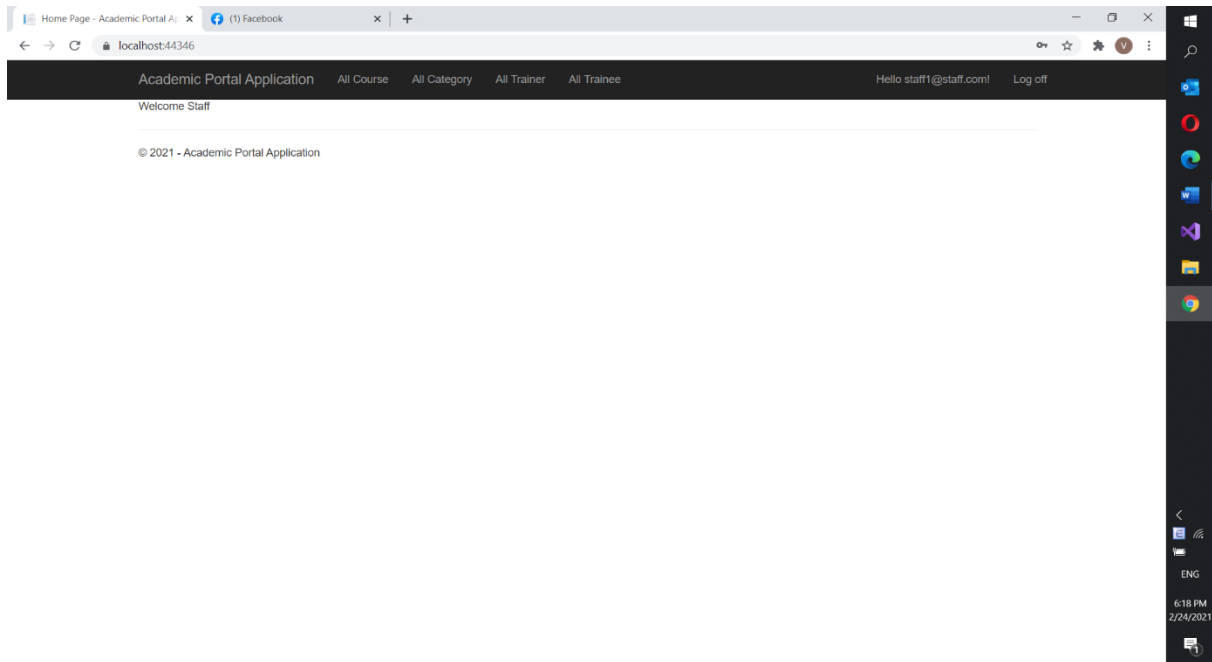


Figure 2 Training Staff dashboard

## 2. Trainer Management

Training Staff can open All Trainer page by clicking All Trainer on the navbar. This page will list the Trainer accounts that are currently registered into the system on a table, coupled with action buttons on each row of account the table. Training staff can change a Trainer account information and see Courses that are related to a Trainer through using the Edit and All Trainer Related Course button.

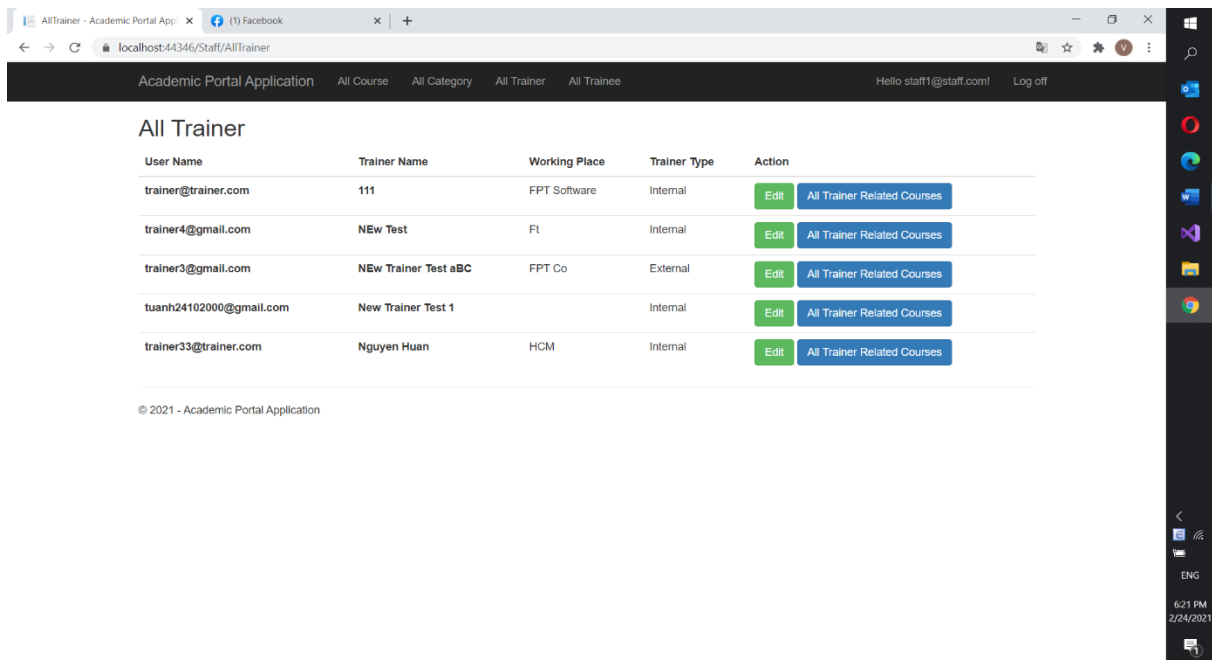


Figure 3 All Trainer page



Academic Portal Application | All Course | All Category | All Trainer | All Trainee | Hello staff1@staff.com! | Log off

## Edit Trainer

Edit Trainer

Email:

WorkingPlace:

TrainerName:

TypeId:

© 2021 - Academic Portal Application

Figure 4 Edit Trainer

Academic Portal Application | All Course | All Category | All Trainer | All Trainee | Hello staff1@staff.com! | Log off

## All Course Of Trainer

Trainer Id	Course Name	Course Description	Action
trainer@trainer.com	JS	...	<input type="button" value="Reassign"/> <input type="button" value="Remove"/>

© 2021 - Academic Portal Application

Figure 5 All Course of Trainer page

Furthermore, the Training Staff can also manage and organize a Trainer's teaching program by using Assign, Reassign, Remove buttons on All Course of Trainer page to make changes to a Trainer's list of Assigned Courses.

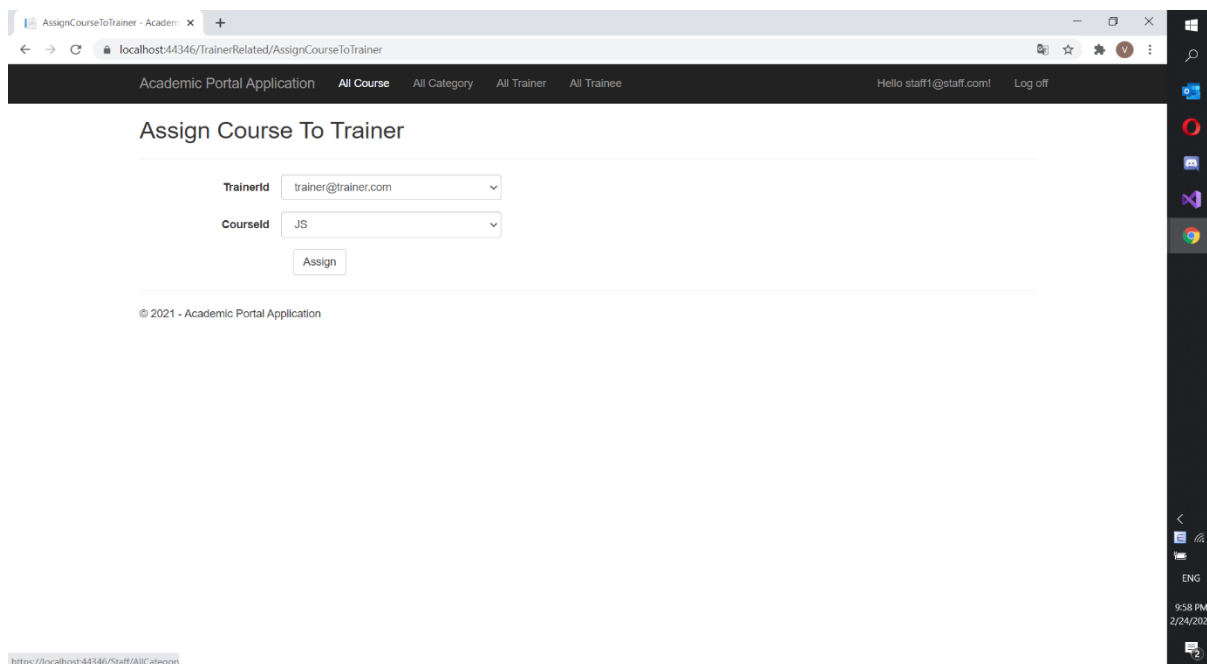


Figure 6 Assign Course to Trainer

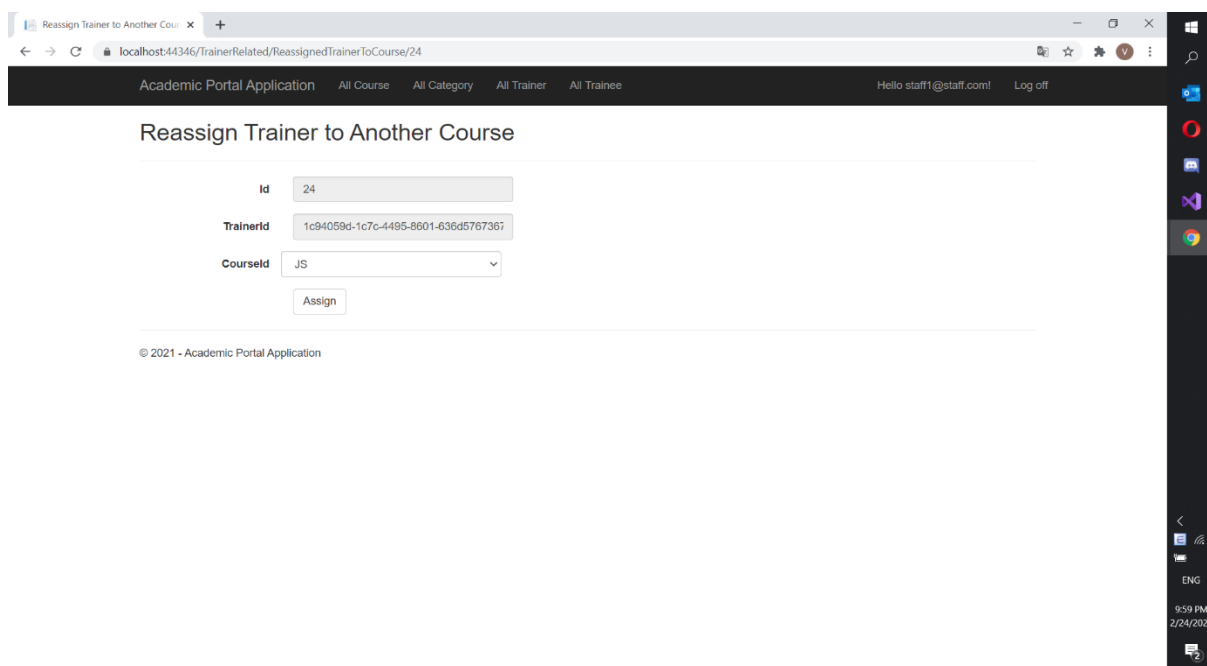


Figure 7 Reassign Trainer to a Different Course

### 3. Trainee Management

Training Staff can open All Trainee page by clicking All Trainee on the navbar. This page will list the registered Trainee accounts that are currently registered into the system on a table with action buttons on each row of account the table. Training Staff is allowed to have better degree of control over Trainee Account compared to Trainer Account. Training Staff can perform CRUD operations on a Trainee

Account by using the Create, Edit, Delete buttons. Moreover, Search function is also implemented so that Training Staff can search for a Trainee account by name.

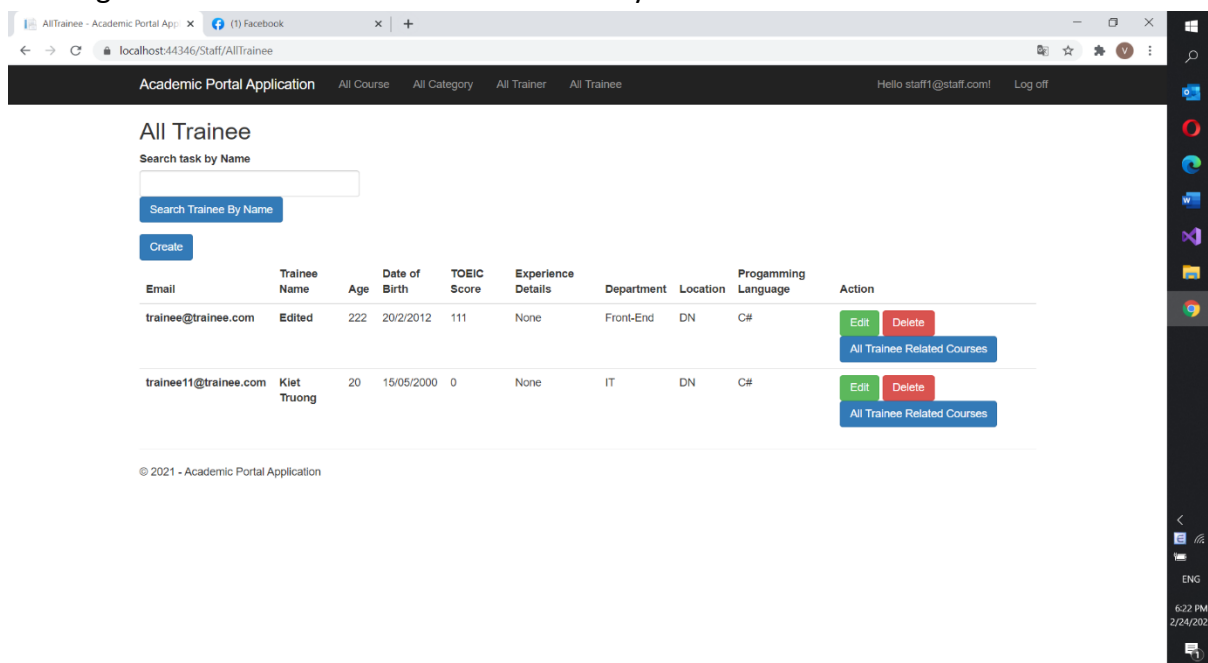


Figure 8 All Trainee page

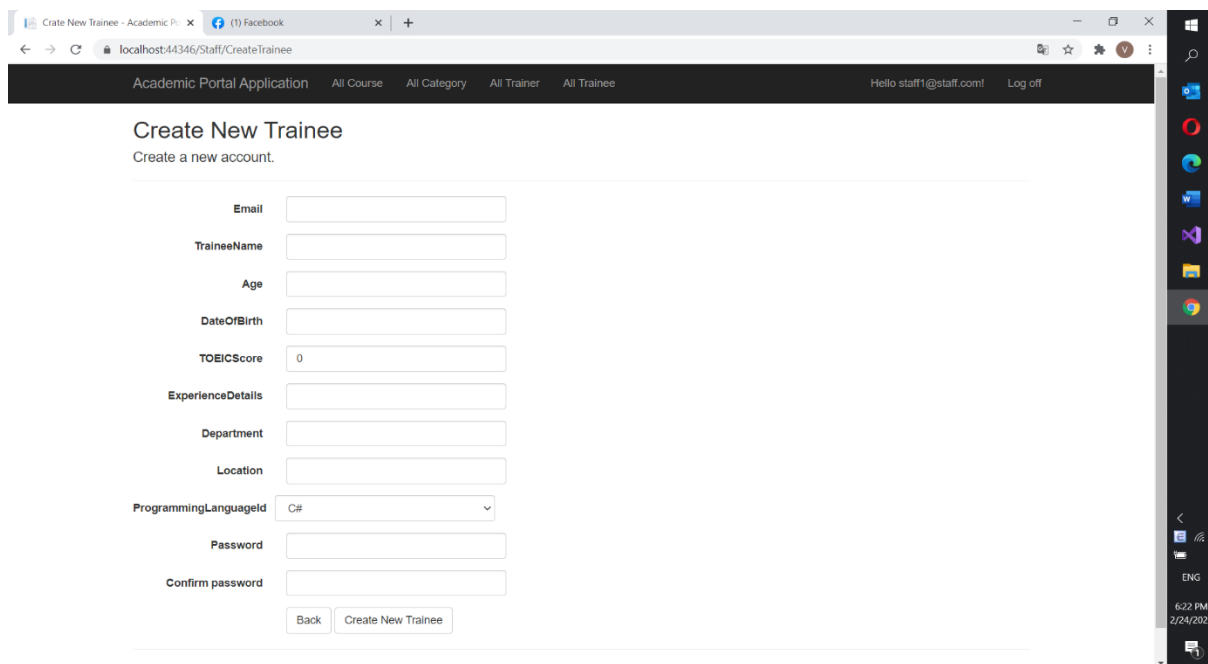
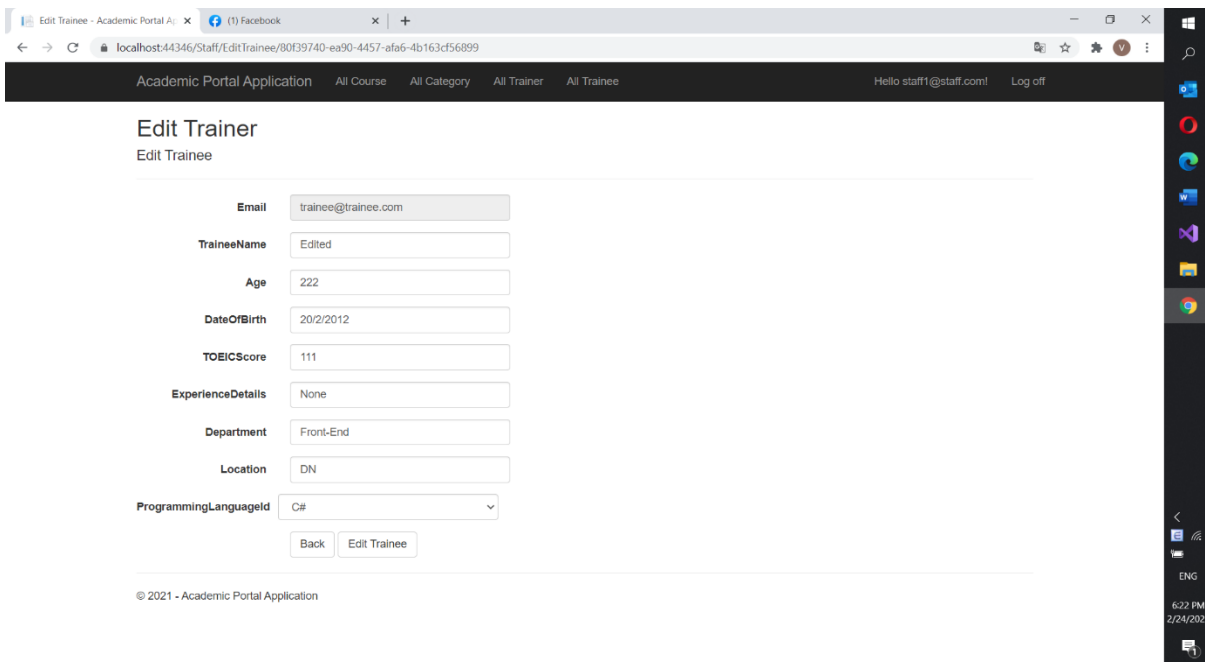


Figure 9 Create Trainee



Academic Portal Application | All Course | All Category | All Trainer | All Trainee | Hello staff1@staff.com! | Log off

### Edit Trainer

Edit Trainee

Email:

TraineeName:

Age:

DateOfBirth:

TOEICScore:

ExperienceDetails:

Department:

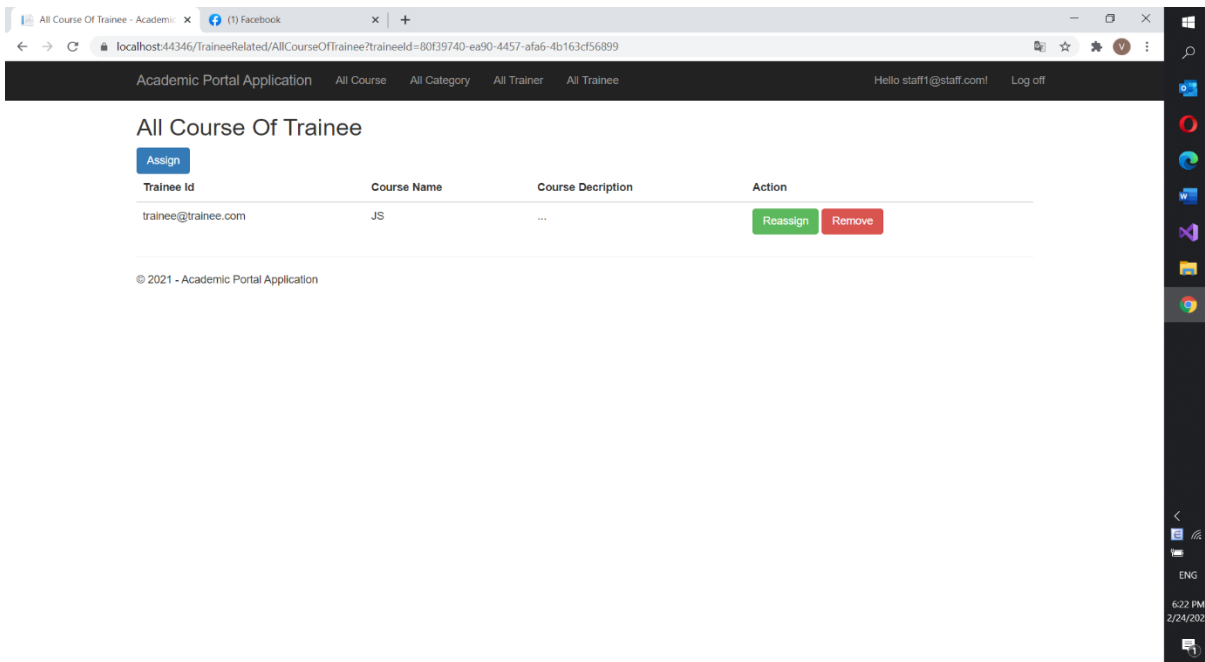
Location:

ProgrammingLanguageId:

© 2021 - Academic Portal Application

Figure 10 Edit Trainee

Furthermore, the Training Staff can also manage and organize a Trainee's training program by using Assign, Reassign, Remove buttons on All Course of Trainee page to change selected Trainee's list of Assigned Courses.



Academic Portal Application | All Course | All Category | All Trainer | All Trainee | Hello staff1@staff.com! | Log off

### All Course Of Trainee

Trainee Id	Course Name	Course Description	Action
trainee@trainee.com	JS	...	<input type="button" value="Reassign"/> <input type="button" value="Remove"/>

© 2021 - Academic Portal Application

Figure 11 All Course of Trainee

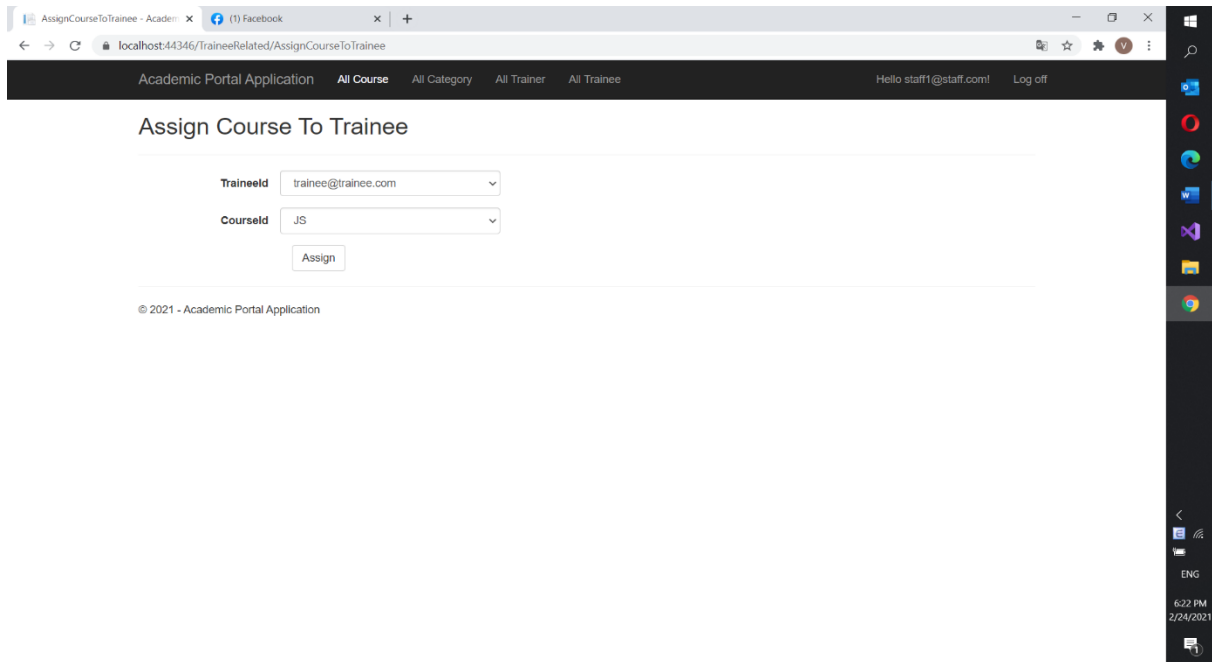


Figure 12 Assign Course to Trainee

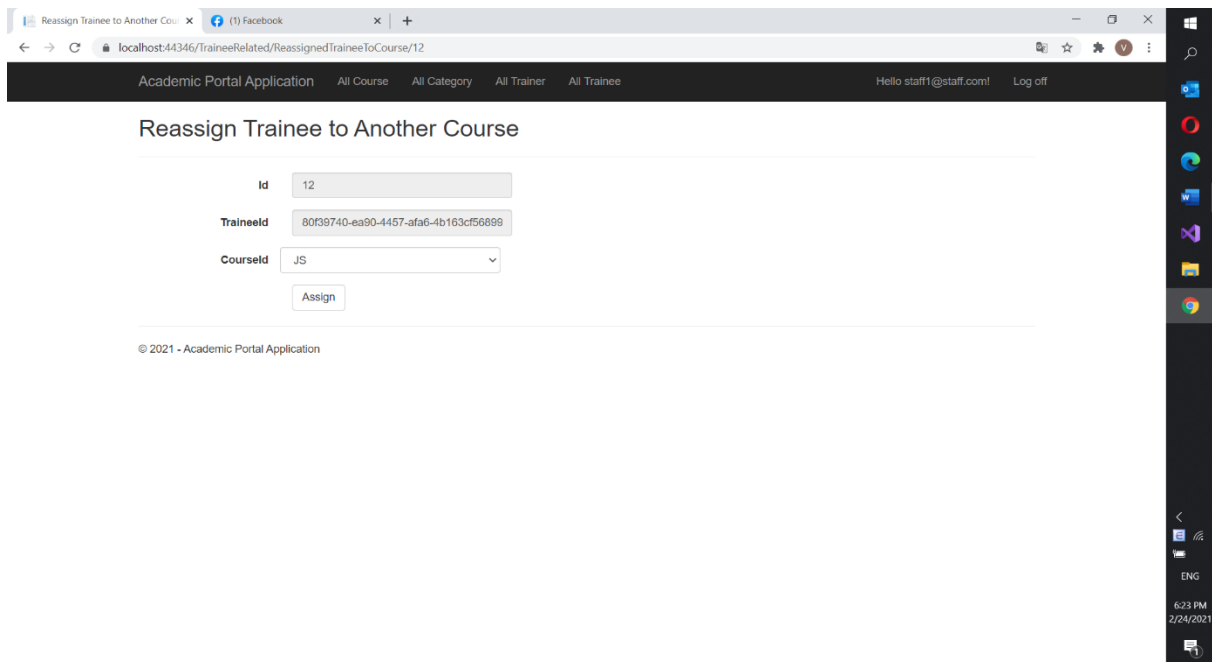


Figure 13 Reassign Trainee to Different Course

#### 4. Course Management

Training Staff can open All Course page by clicking All Course on the navbar. This page shows all available Courses in the system on a table with action buttons on each row of Course. Training Staff can perform CRUD operations on the list of Course by using the Create, Edit, Delete. A Search function is also implemented, so Training Staff can also search a course by its name.

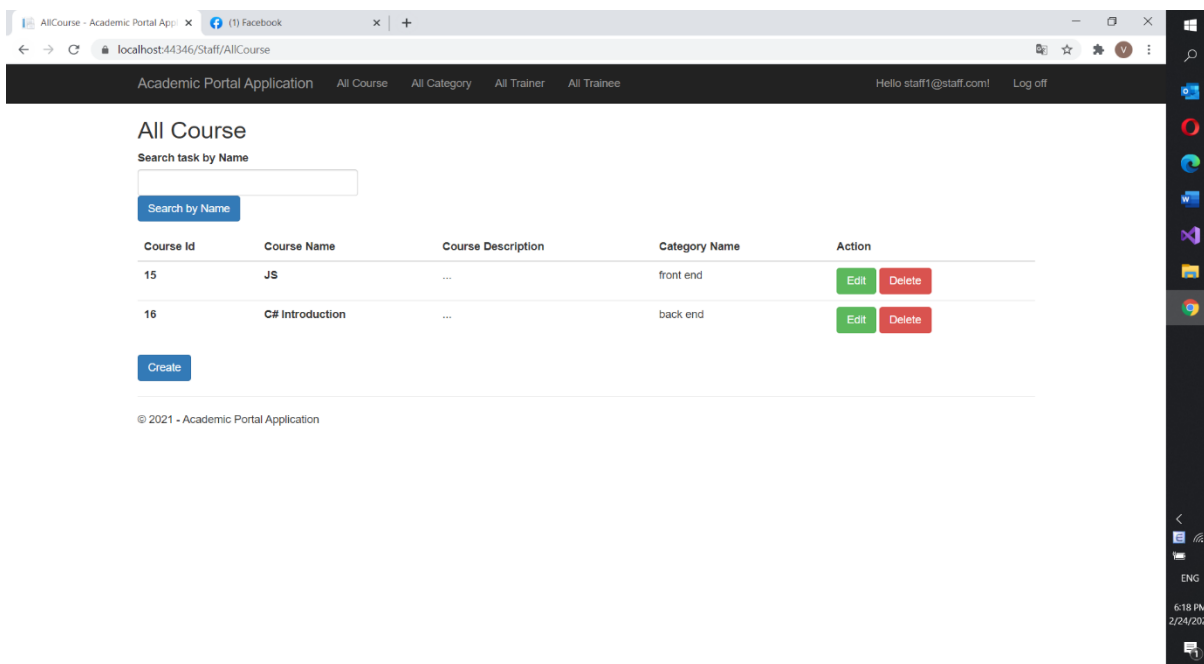


Figure 14 All Course page

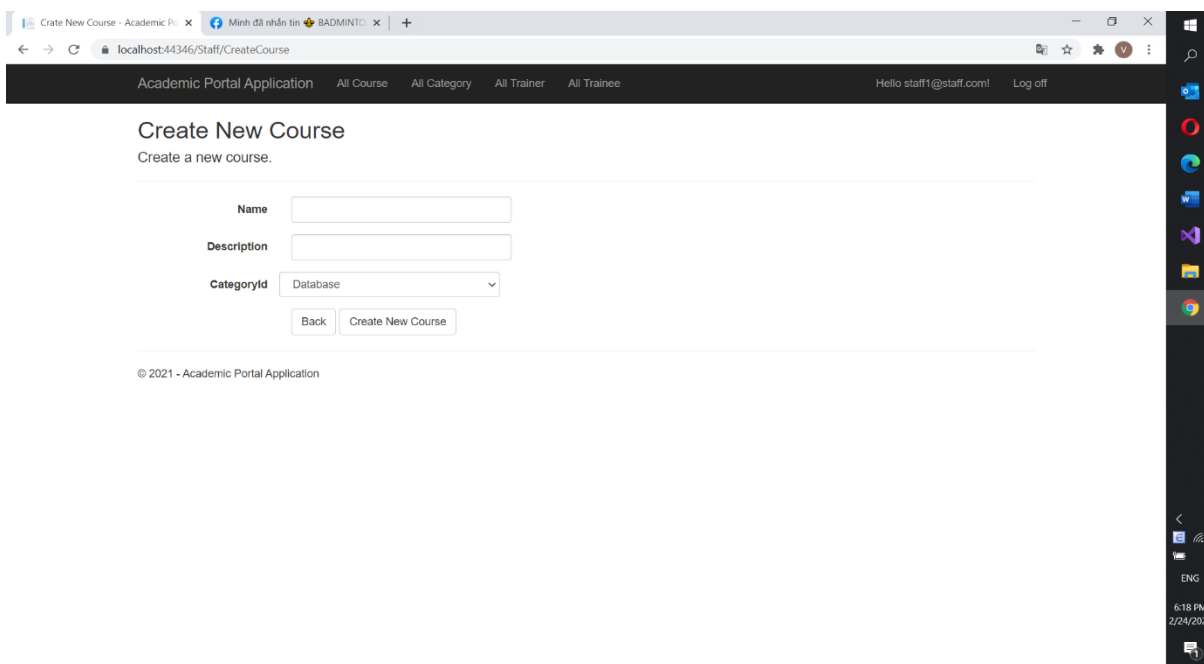
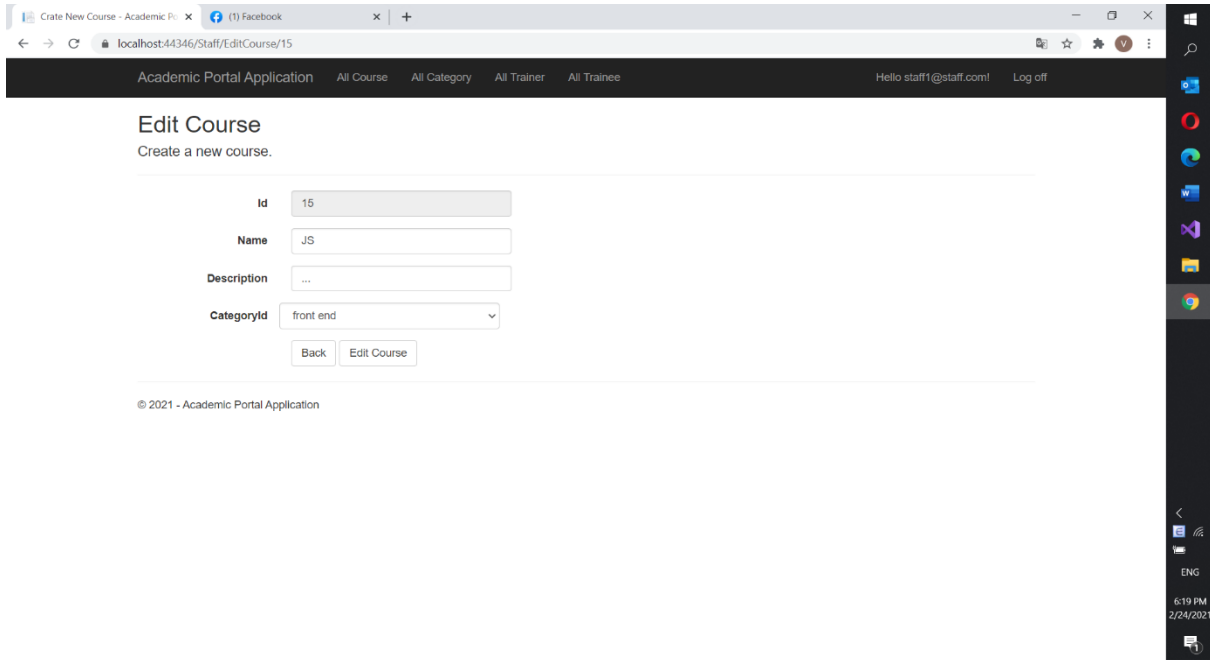


Figure 15 Create Course



Academic Portal Application | All Course | All Category | All Trainer | All Trainee | Hello staff1@staff.com! | Log off

### Edit Course

Create a new course.

Id: 15

Name: JS

Description: ...

CategoryId: front end

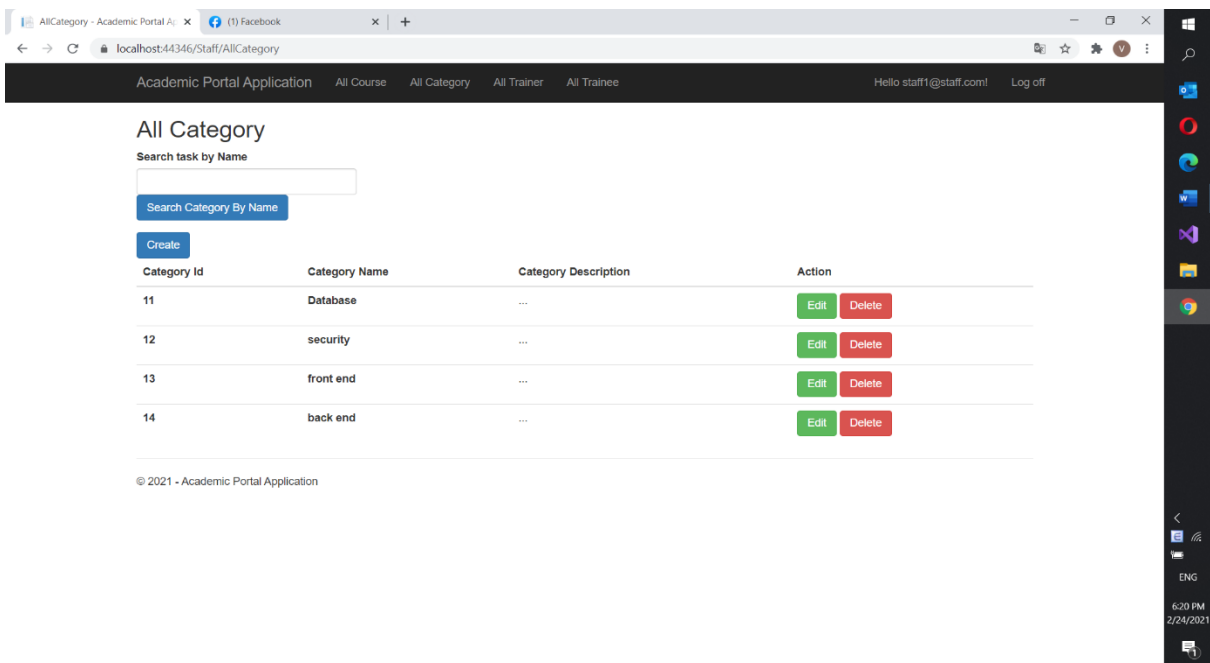
Back | Edit Course

© 2021 - Academic Portal Application

Figure 16 Edit Course

## 5. Course Category Management

Training Staff can open All Category page by clicking All Category on the navbar. This page shows all available Course Categories in the system on a table with action buttons on each row of Course Category. Training Staff can perform CRUD operations on the list of Course by using the Create, Edit, Delete. A Search function is also implemented, so Training Staff can also search a Course Category by its name.



Academic Portal Application | All Course | All Category | All Trainer | All Trainee | Hello staff1@staff.com! | Log off

### All Category

Search task by Name

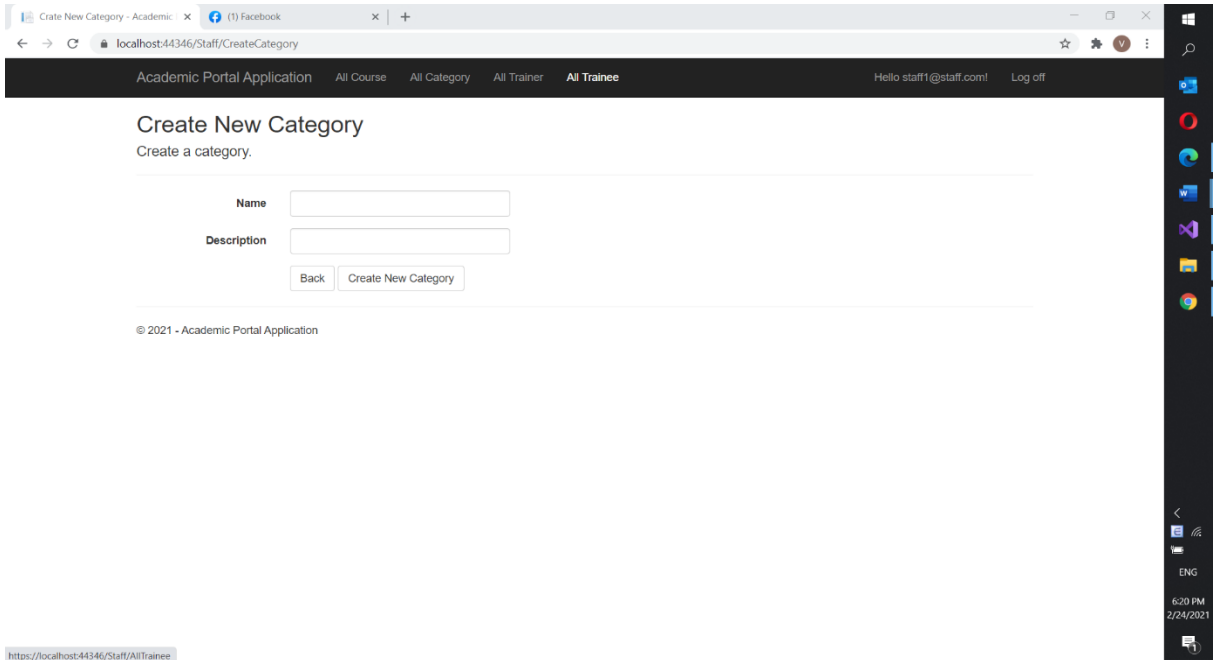
Search Category By Name

Create

Category Id	Category Name	Category Description	Action
11	Database	...	<a href="#">Edit</a> <a href="#">Delete</a>
12	security	...	<a href="#">Edit</a> <a href="#">Delete</a>
13	front end	...	<a href="#">Edit</a> <a href="#">Delete</a>
14	back end	...	<a href="#">Edit</a> <a href="#">Delete</a>

© 2021 - Academic Portal Application

Figure 17 All Category page



Academic Portal Application | All Course | All Category | All Trainer | **All Trainee** | Hello staff1@staff.com! | Log off

## Create New Category

Create a category.

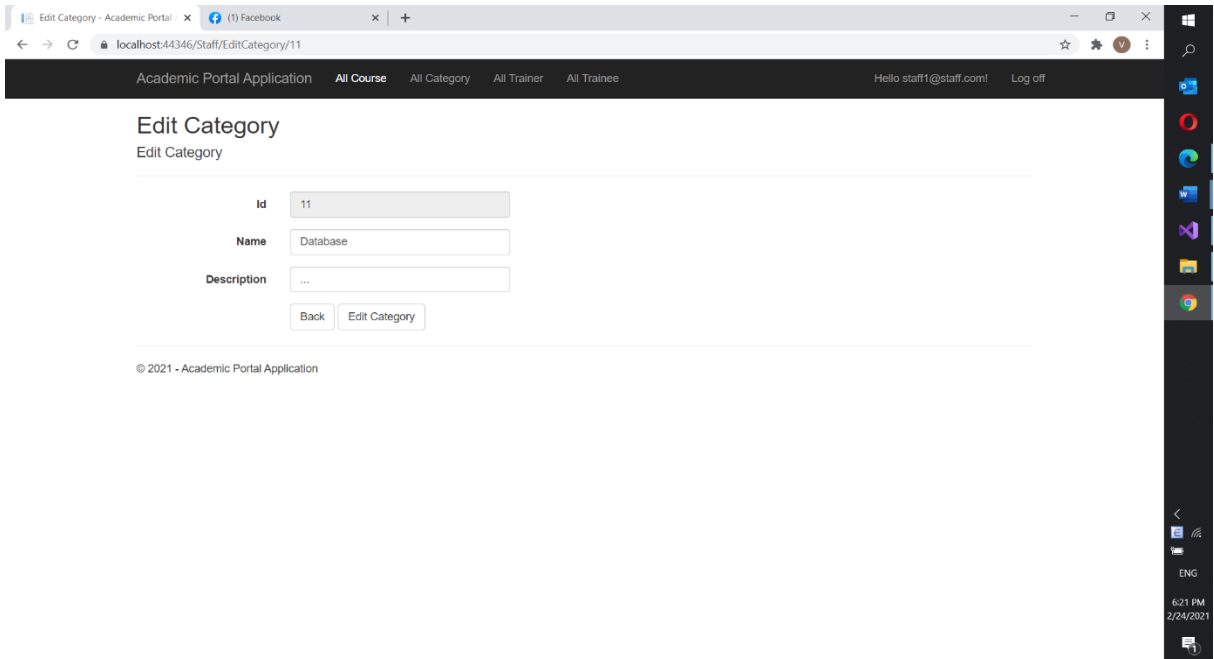
**Name**

**Description**

© 2021 - Academic Portal Application

https://localhost:44346/Staff/AllTrainee

Figure 18 Create Course Category



Academic Portal Application | **All Course** | All Category | All Trainer | All Trainee | Hello staff1@staff.com! | Log off

## Edit Category

Edit Category

**Id**

**Name**

**Description**

© 2021 - Academic Portal Application

localhost:44346/Staff/EditCategory/11

Figure 19 Edit Course Category



## IV. Application Evaluation (P6)

During the development process, the local host only serves as a testing and debugging environment. The performance of an application can only be review when they are deployed into the internet, being hosted in a server. There are two main criteria used for determining the performance of a web application.

- **Loading Speed:** Since we've deployed both front-end and back-end at the same root, our connection speed between client-side and server-side is the same as that of the localhost in the development environment. The only problem is the method of access to the database. We used Microsoft Azure to store our data, so the server would take quite an amount time to retrieve the data from the database.
- **Data rendering:** We used the client-side rendering system, which means that the data rendering process would rely on the client computer. However, since our project doesn't use too many static files like image, video, etc. So, it's almost a computer that can render the data with the best performance.

We can use the Lighthouse, one of Google Chrome's Developer Tools to measure the performance of the application, the criteria are measured and then turned in to scores to rate our application's performance like picture below.

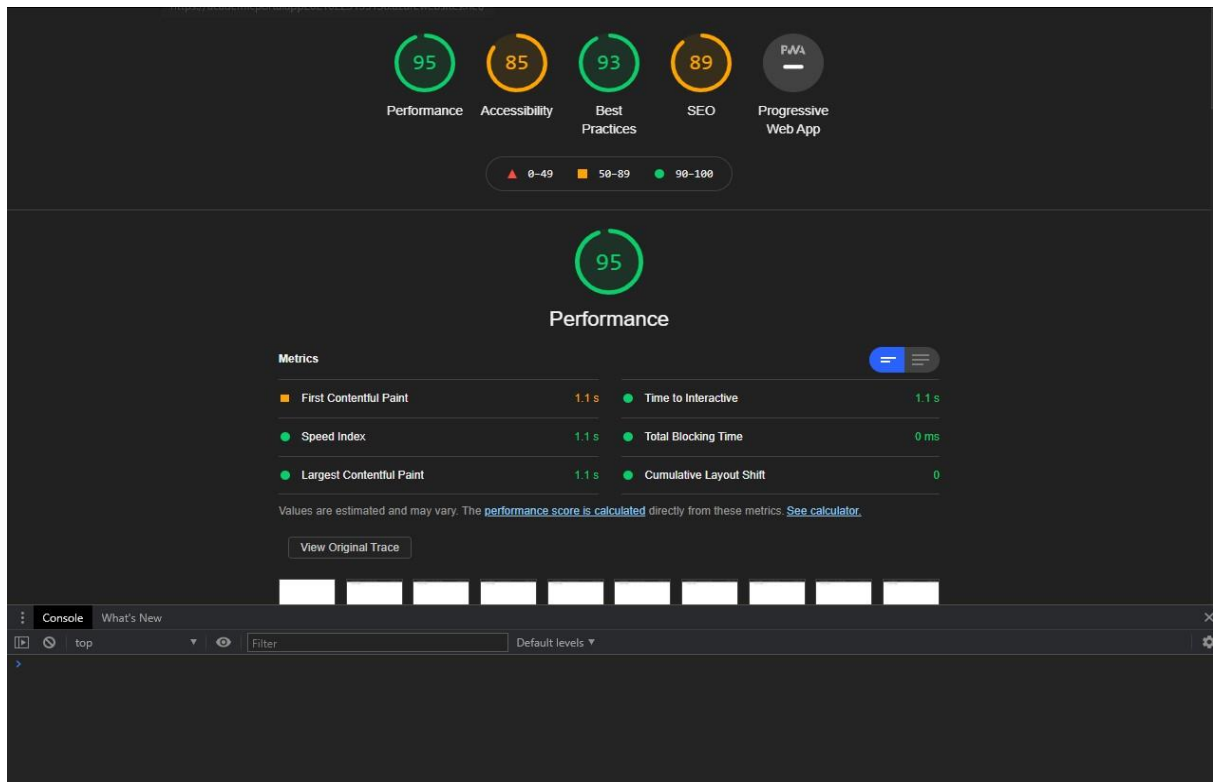


Figure 20 Application's Google Lighthouse score

We can see the application's performance score rated in Performance, Accessibility, Best Practices and SEO (Search Engine Optimization).

Below is a performance test on a sequence of functions: Login, All Trainers, Create, Edit, Delete Trainer, Logout.

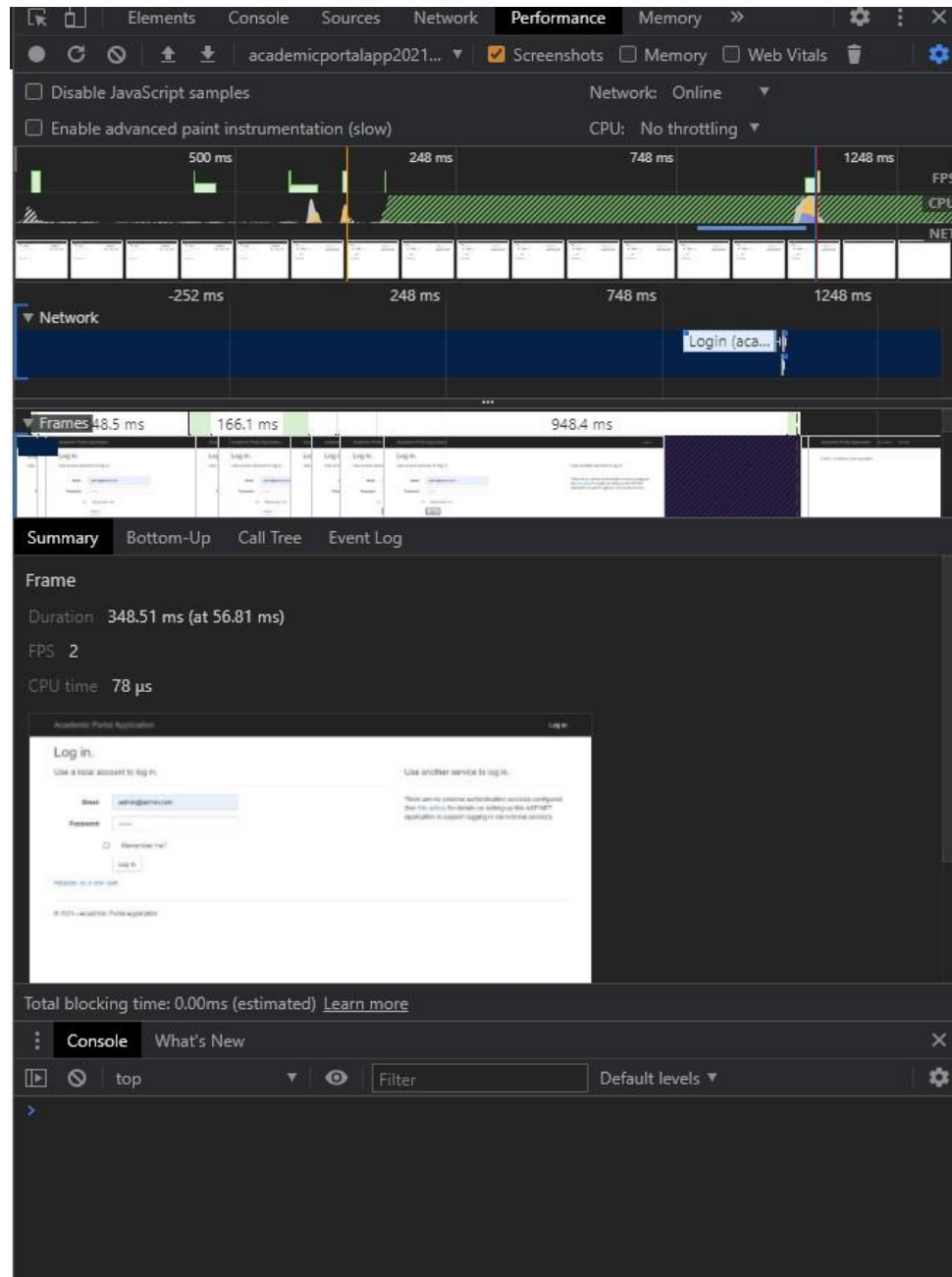


Figure 21 Login function performance test

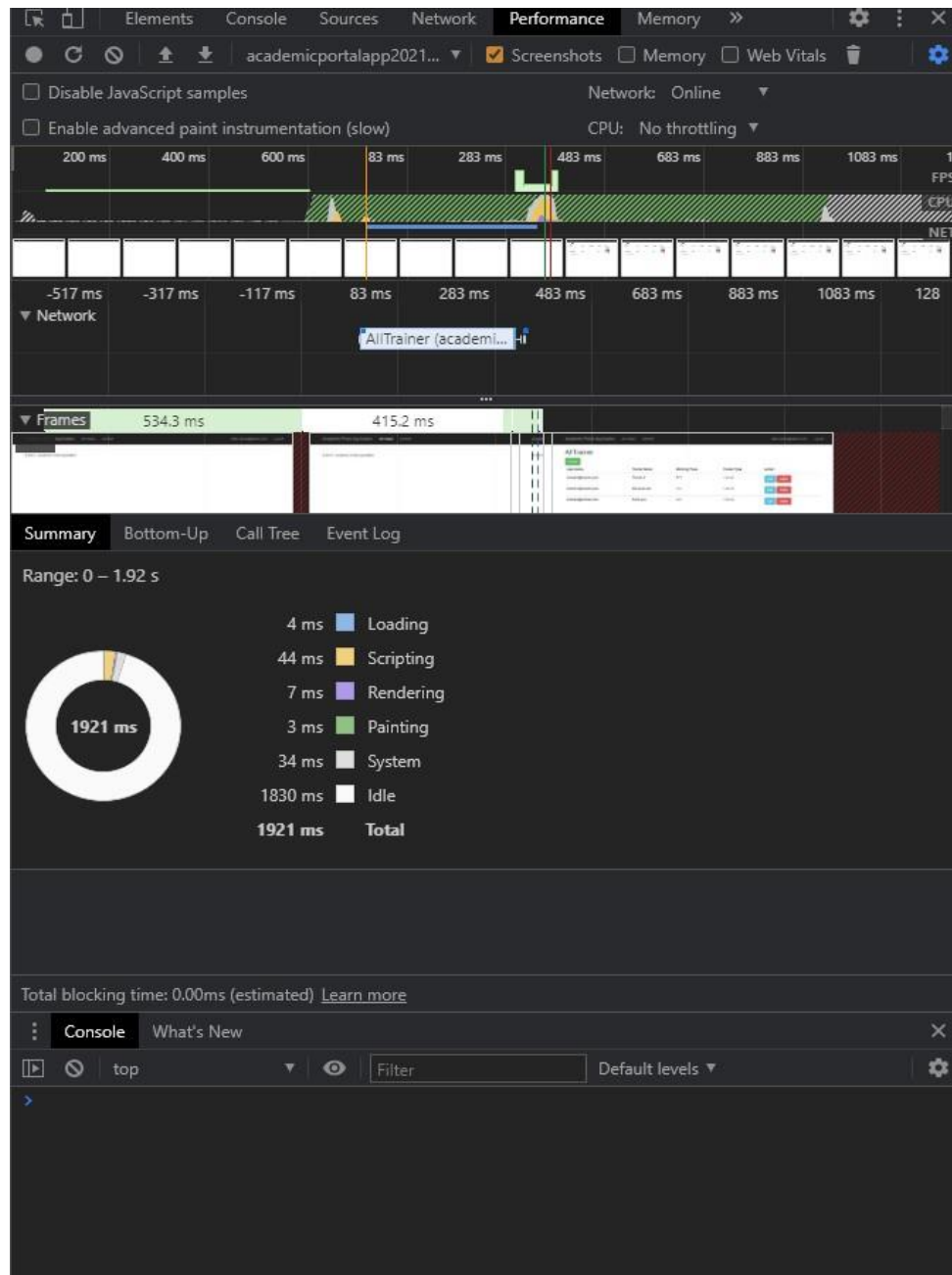


Figure 22 View All Trainer performance test

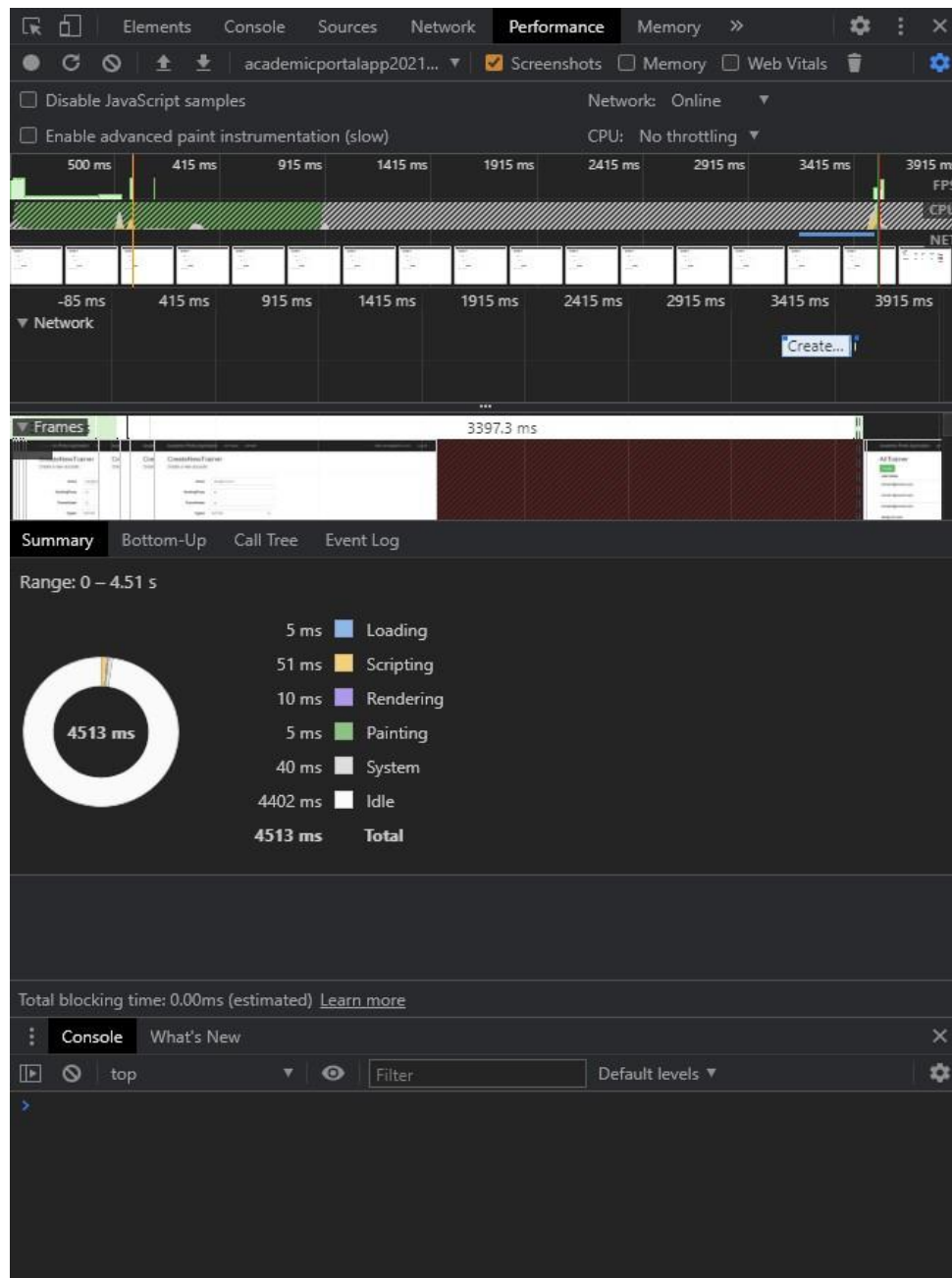


Figure 23 Create Trainer performance test

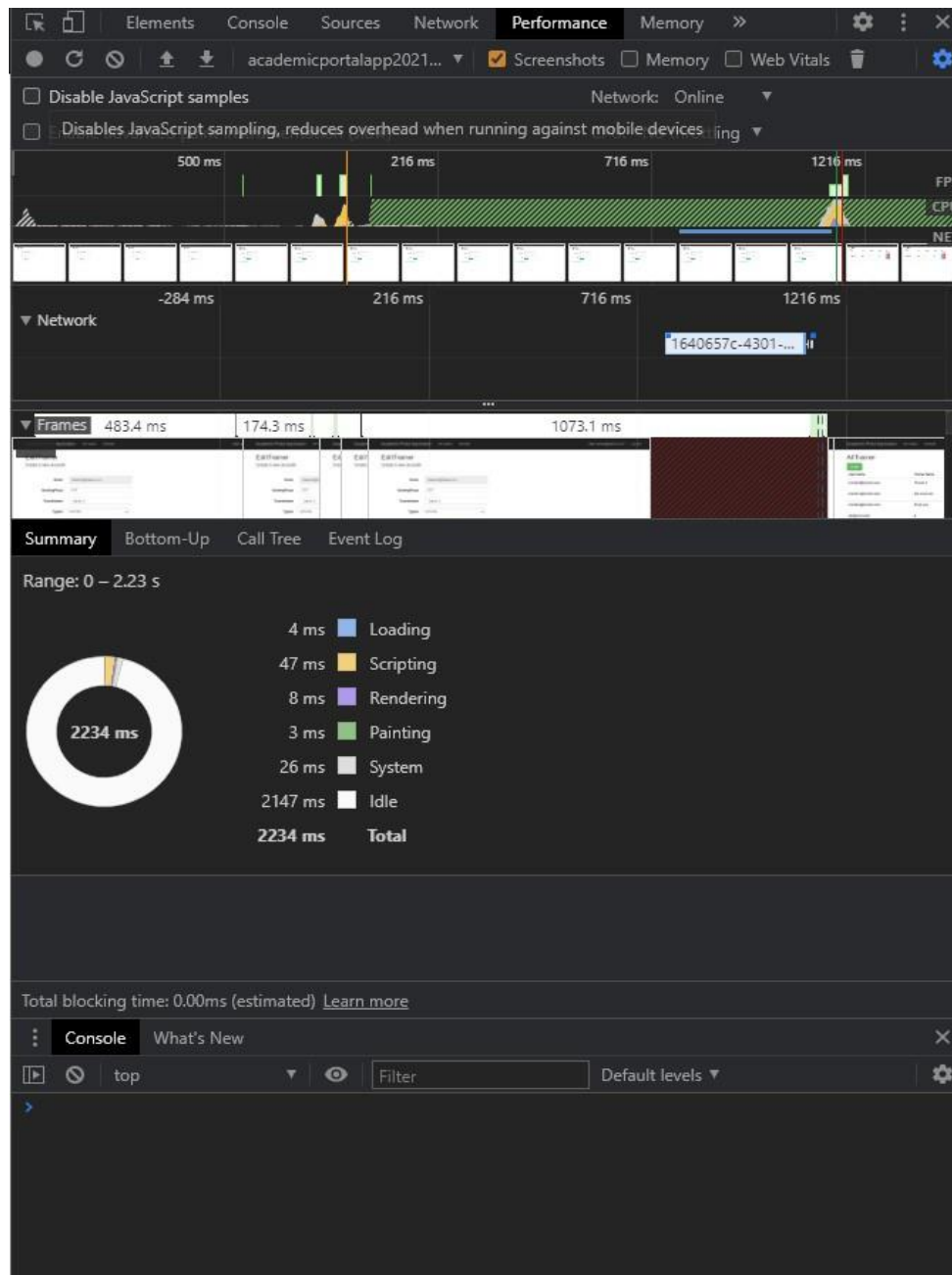


Figure 24 Edit Trainer performance test

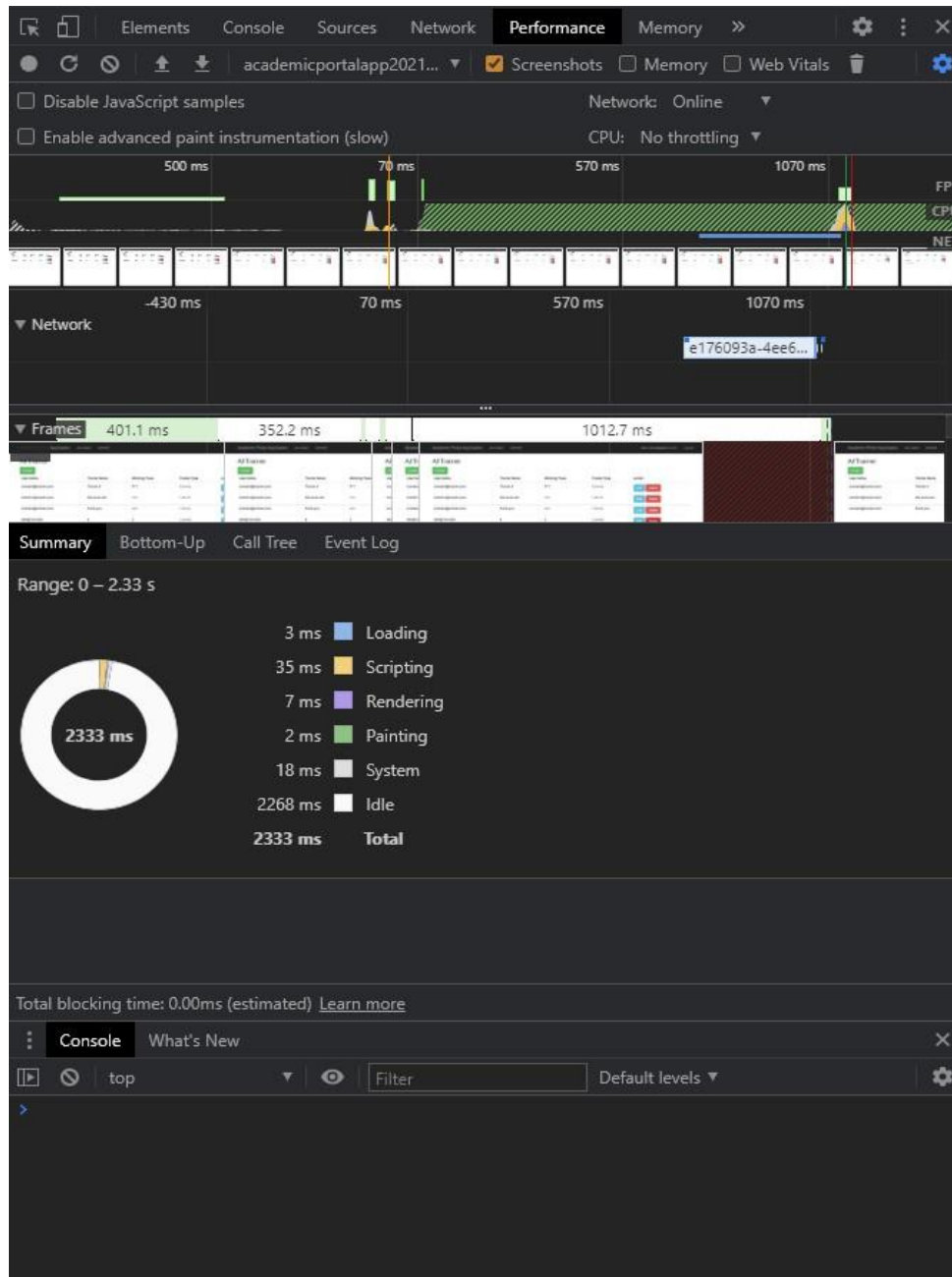


Figure 25 Delete Trainer performance test

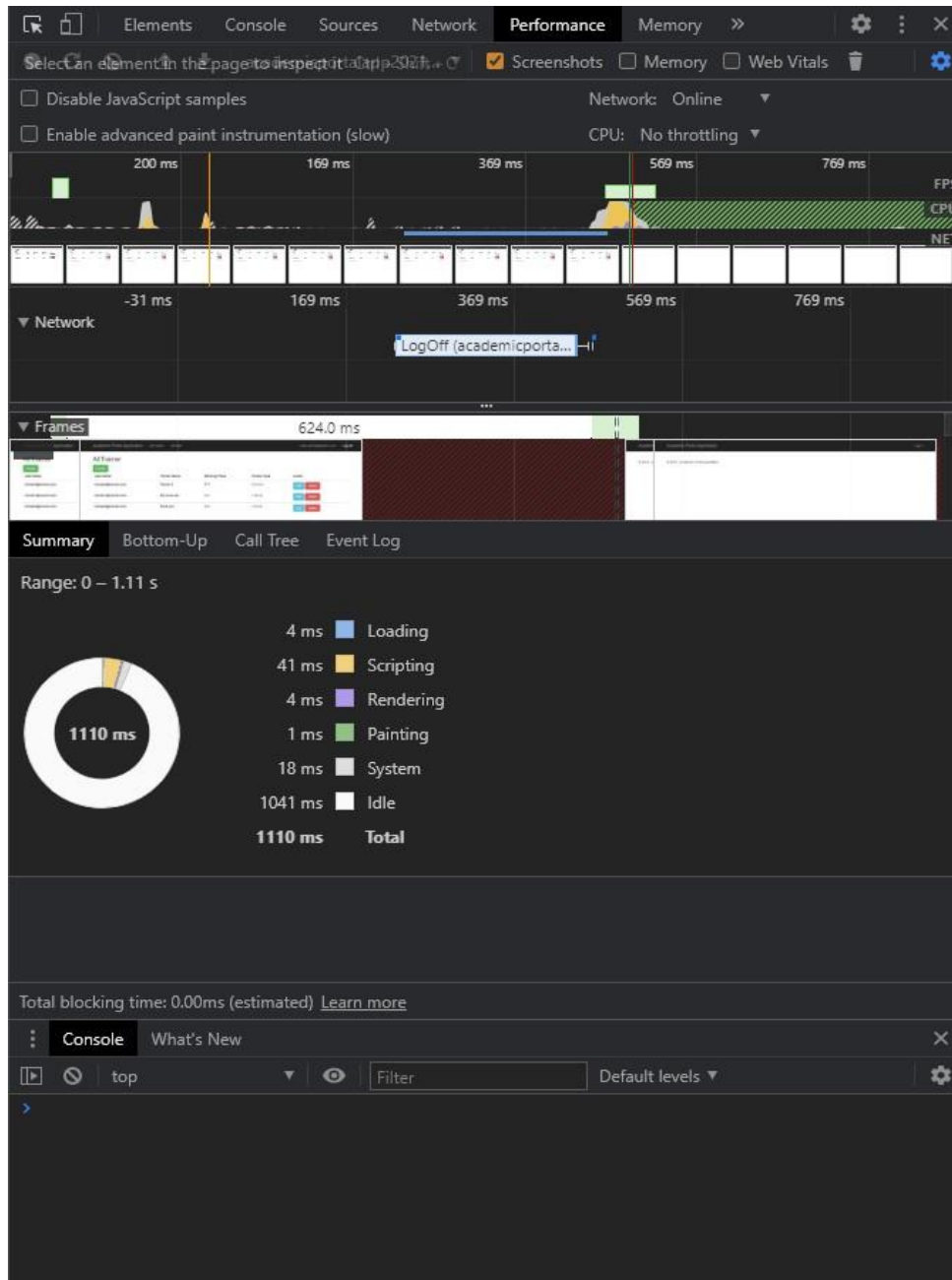


Figure 26 Logout performance test

The loading time results of basic functions such as Login and Logout took 0.3s and 1.1s respectively. As for the functions that requires fetching and retrieving a larger amount of data took 2~4 seconds in average. This is a relatively decent loading speed, combined with the good Lighthouse performance score proves that the performance of the application code is satisfactory. We can also confirm that the long loading time during the deployment process is due to the method of access to Azure's database. The team will try to resolve this function at a later date.