




AUGUST 26, 2020

# APPLICATION DEVELOPMENT ASSIGNMENT 2

HUY TRAN  
STUDENT ID: GCD18457  
Class: GCD0606:

---



## 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>		<b>Date Received 1st submission</b>	
<b>Re-submission Date</b>		<b>Date Received 2nd submission</b>	
<b>Student Name</b>	Tran Quang Huy	<b>Student ID</b>	GCD18457
<b>Class</b>	GCD0606	<b>Assessor name</b>	Hoang Nhu Vinh
<b>Student declaration</b> 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.			
		<b>Student's signature</b>	Huy

### Grading grid

P4	P5	P6	M3	M4	M5	D2	D3

☐ **Summative Feedback:**

☐ **Resubmission Feedback:**

**Grade:**

**Assessor Signature:**

**Date:**

**Lecturer Signature:**

## Contents

<b>Introduction.....</b>	<b>1</b>
<b>1. Peer review.....</b>	<b>2</b>
<b>1.1. Formal questionnaire.....</b>	<b>2</b>
<b>1.2. Feedback analysis.....</b>	<b>6</b>
1.2.1. Improve login .....	6
1.2.2. Lock account when the login fails 5 times .....	8
1.2.3. Change permission to view dashboard.....	9
1.2.4. Improve Training Staff page.....	9
1.2.5. Improve button with icon .....	10
<b>2. Application development.....</b>	<b>11</b>
<b>2.1. Development application.....</b>	<b>11</b>
2.1.1. System Workflow .....	11
2.1.2. Database .....	12
2.1.3. System structure .....	13
2.1.4. Implement code and result.....	19
<b>2.2. Tools, techniques, and methodologies.....</b>	<b>35</b>
<b>2.3. Ideas/Potential to improve the system.....</b>	<b>54</b>
<b>3. Application review .....</b>	<b>60</b>
<b>3.1. Comparison to initial requirements.....</b>	<b>60</b>
<b>3.2. Analyse the factors.....</b>	<b>75</b>
3.2.1. The factors influence the performance of the business application .....	75
3.2.2. Critical review the design development and testing application .....	76
<b>3.3. Assessment of the system's strength and weakness and improve system .....</b>	<b>86</b>
3.3.1. System's strength and weakness .....	86
3.3.2. Improvement system .....	92
<b>Conclusion .....</b>	<b>95</b>

---

## Table of Figures

Figure 1. Overview system workflow .....	11
Figure 2. Entity-relationship model .....	12
Figure 3. MVC Architecture.....	13
Figure 4. The design of API in ASP.NET Core.....	57

## Table of Pictures

Picture 1. Login interface .....	6
Picture 2. Forgot password interface.....	7
Picture 3. Login fails 5 times interface .....	8
Picture 4. Administrator unlock user .....	8
Picture 5. Existing Course interface .....	9
Picture 6. Existing Topic interface.....	9
Picture 7. Training staff user management interface .....	10
Picture 8. FPT Learning System structure .....	13
Picture 9. UnAuthenticated - View - Home – Index.....	19
Picture 10. Login interface .....	20
Picture 11. Register interface.....	21
Picture 12. Authenticated Index .....	22
Picture 13. Category index interface .....	23
Picture 14. Category create interface .....	24
Picture 15. Category details interface.....	24
Picture 16. Category edit interface .....	25
Picture 17. Courses index interface .....	26
Picture 18. Course create interface .....	27
Picture 19. Course Details interface .....	27
Picture 20. Course edit interface .....	28
Picture 21. Topic index interface .....	29
Picture 22. Topic create interface.....	29
Picture 23. Topic details interface .....	30
Picture 24. Training staff topic edit interface. ....	30
Picture 25. Training Staff Assign Trainee into Course interface .....	31
Picture 26. TraineeCourses create interface.....	31
Picture 27. TrainerTopicsController index interface.....	32
Picture 28. TrainerTopics Create interface .....	32
Picture 29. UsersController index interface with role TrainingStaff.....	33
Picture 30. UsersController index interface with role Administrator.....	33
Picture 31. UsersController Edit interface .....	34
Picture 32. Training Staff view Course with Balsamiq mockups 3 .....	35
Picture 33. Balsamiq browser container.....	36
Picture 34. Balsamiq Image.....	36
Picture 35. Balsamiq Label .....	36
Picture 36. Balsamiq Rectangle.....	37

---

Picture 37. Balsamiq Text Input .....	37
Picture 38. Balsamiq Icon.....	37
Picture 39. Balsamiq Button .....	37
Picture 40. Balsamiq Combobox .....	37
Picture 41. Training Staff view Course with bootstrap studio .....	38
Picture 42. Bootstrap studio Select template .....	38
Picture 43. Bootstrap studio selects the bootstrap version .....	39
Picture 44. Bootstrap Studio modify the template.....	39
Picture 45. Bootstrap Studio preview template .....	40
Picture 46. Bootstrap Studio export .....	40
Picture 47. Training Staff view Course with lucid Chart.....	41
Picture 48. Insert content .....	41
Picture 49. LucidChart content container .....	42
Picture 50. LucidChart Actor .....	43
Picture 51. Lucidchart Use Case.....	43
Picture 52. Lucidchart relationship .....	43
Picture 53. Training Staff view Course with Visual Studio .....	44
Picture 54. Visual Studio create Model.....	45
Picture 55. Visual studio DbSet .....	46
Picture 56. Visual Studio Add-Migration and Update-Database .....	46
Picture 57. Visual Studio create ViewModel.....	47
Picture 58. Visual Studio creates Controller .....	47
Picture 59. Visual Studio creates the view.....	48
Picture 60. Training Staff view Course with SQL Server.....	49
Picture 61. SSMS Edit .....	50
Picture 62. SSMS Design .....	51
Picture 63. SSMS Select.....	51
Picture 64. Training Staff view Course with Google Chrome.....	52
Picture 65. Google Chrome with localhost:44363 .....	52
Picture 66. Google Chrome with google chrome developer tool .....	53
Picture 67. Performance login .....	81
Picture 68. Performance dashboard page .....	81
Picture 69. Performance view user.....	82
Picture 70. Performance create user .....	82
Picture 71. Performance create category .....	83
Picture 72. Performance creates course.....	83
Picture 73. Performance Create Topic.....	84
Picture 74. Performance Assign Trainer to Topic .....	84
Picture 75. Performance Assign Trainee to Course .....	85
Picture 76. Font use in the system.....	87
Picture 77. Pictures on the index page .....	87
Picture 78. Content Options .....	88
Picture 79. Main content .....	88
Picture 80. Redirect HTTP to HTTPS.....	89

---

Picture 81. Unnecessary CSS and JavaScript.....	91
Picture 82. Implement notification.....	94
Picture 83. Implement chatbox.....	94

## Table of tables

Table 1. Formal questionnaire for peer review .....	2
---	---

## Introduction

In this report, we will demonstrate how to deploy the application based on the requirements and SRS that has been deployed previously.

From the Peer-review feedback, the exploitation problems of SRS have been corrected and added to the requirements of the system to develop optimally to deliver to customers.

Besides, the final product, FPT Learning System, will be deployed to the cloud computing service of Azure.

The strengths and weaknesses of the system will be analyzed for development for the next version



## 1. Peer review

### 1.1. Formal questionnaire

No.	Module / Screen	Description	Creator	Create date	Answer	Answer date	Status
1	Login	In the SRS document, page 48, 61, 82, 85, there is not the "Forgot Password" button on the Login screen. I understand that it is better to add this button. Do I understand correctly?	Tester	22/08/20	Yes, in the Login screen need a "Forgot Password" button. If the user forgot their password, they will need a feature to get back their password through their email.	23/08/20	Closed: Answered
2	Login	In the SRS document, page 48,61, 82,85 if the user login with the wrong password or email 5 times. The account will be locked. Do I understand correctly?	Tester	22/08/20	Yes, the system needs to improve this feature for security. In this way, I think when the password detections programs will hard to find out the pass of the user, especially is admin or training staff.	23/08/20	Closed: Answered
3	Admin Dashboard page	In the SRS document, page 16, there is a summary table of the system such as total user, total course, topic, category. I understand that is unnecessary to have a summary table there. Do I understand correctly?	Tester	22/08/20	Yes, the system requirement did not include the summary table. If the summary table will be existing, just only Training Staff will need it.	23/08/20	Closed: Answered

Table 1. Formal questionnaire for peer review

4	Training staff add a course	In the SRS document, page 68, there is not show up the course exists in the database, so training staff will not input the same name of course by looking at the sub-table. I understand that is better to have a sub-table to show the name of course exists. Do I understand it correctly?	Tester	22/08/20	Yes, the requirement document didn't mention this feature. But in this case, this is a good feature to improve User Experience. It will make training staff easier to Add a new course.	24/08/20	Closed: Answered
5	Training staff add a topic	In the SRS document, page 71, there is not show up the topic exists in the database, so training staff will not input the same name of the topic by looking at the sub-table. I understand that is better to have a sub-table to show the name of the topic exists. Do I understand it correctly?	Tester	22/08/20	Yes, the requirement document didn't mention this feature. Similar to the add a course feature. The system needs it to improve UX.	24/08/20	Closed: Answered
6	Trainee view list of courses	In SRS document, page 86, Trainee only view list of courses especially is course name and can't view courses details and topic in that course. I understand that is better to give permission to the trainee to view the topic and course details. Do I understand it correctly?	Tester	22/08/20	No, the requirement document didn't mention that. So, the system does need to implement this feature.	24/08/20	Closed: Answered

7	Training staff delete trainer account	In SRS document, page 79, Training staff can delete the trainer account through the user management page. I understand that is the wrong permission for training staff. Do I understand correctly?	Tester	22/08/20	Yes, the SRS document has a mistake to define the permission of the Training staff that Training staff can't delete the trainer account. Base on the requirement. Training staff only delete the information of the trainer	24/08/20	Closed: Answered
8	Training Staff add the category	In the SRS document, page 72, the category name is unique. I understand that it is necessary to have this business rule. Do I understand it correctly?	Tester	22/08/20	Yes, the name of the category must be unique same with the course name and topic name. That will not confuse the user	24/08/20	Closed: Answered
9	Edit, delete, details button (table button)	In SRS document, page 17, 22,32, 27, 38, 40, 42, 47 the edit/delete/details button just show the name such as edit, delete, details. I understand that is necessary to have an icon for these buttons. Do I understand it correctly?	Tester	22/08/20	Yes, to improve user experience, the system needs to have an icon for these buttons by add using font Awesome. Font Awesome allows the font-end developer to make the website look more friendly by adding a script into _customlayout.cshtml and which HTML that FE developer wants to system.	24/08/20	Closed: Answered

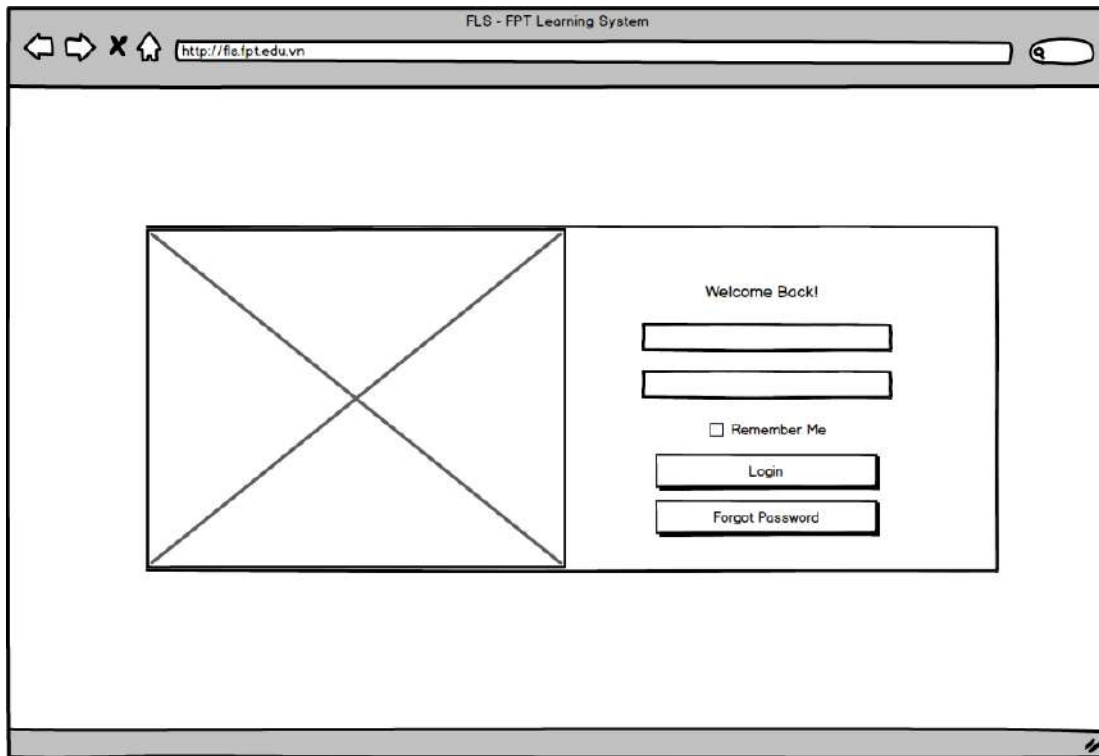
10	Trainee view list of courses	<p>In the SRS document, page 85 there is a table show list of courses but not have information about topics assigned in courses. I suggest 2 options:</p> <p>Option 1: The button “Details” will redirect trainee to the Topic Details page.</p> <p>Option 2: Give the information about Topic in the table</p> <p>Please confirm your selection.</p>	Tester	26/08/20	<p>I prefer to select option 2 better than option 1. In this case, the trainee only needs to see the limited information about the course and topic. Option 2 is better than option 1. That developer can give the specific information that the system want trainee to see in the table. And this will make the system more security and each role have clear permission.</p>	26/08/20	Closed: Answered
----	------------------------------	---	--------	----------	--	----------	------------------

## 1.2. Feedback analysis

Base on peer-review with the questionnaire, there are some serious problems and problems that may develop in the future.

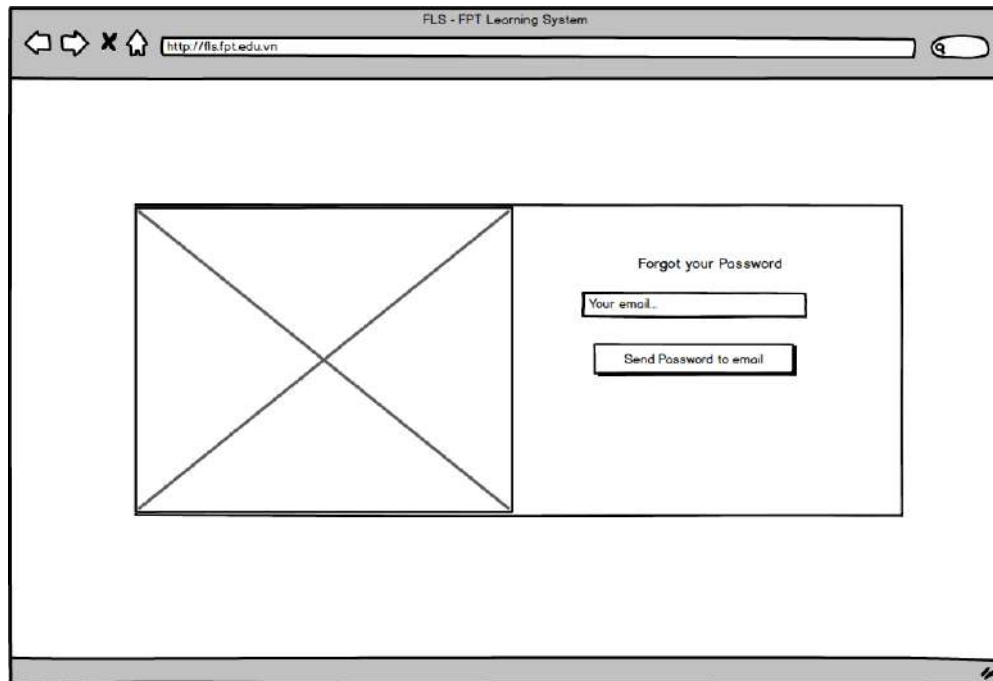
### 1.2.1. Improve login

Base on the system's requirement the "Forgot Password" Not necessary to deploy, but can be considered for future development depending on requirements.



Picture 1. Login interface

When the user enters his/her email which email has existed in the database. The system will send an email with a secure link to change the password.



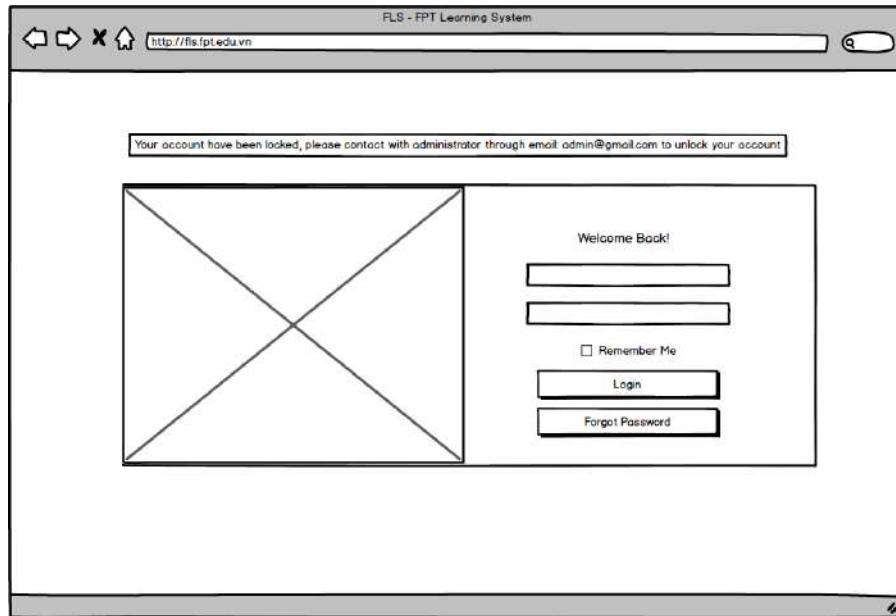
The image shows a web browser window with the title "FLS - FPT Learning System". The address bar displays "http://fls.fpt.edu.vn". The main content area contains a "Forgot your Password" form. On the left side of the form is a large square placeholder with a diagonal 'X' inside. To the right of this placeholder, the text "Forgot your Password" is displayed. Below this text is a text input field labeled "Your email:". Underneath the input field is a button labeled "Send Password to email".

*Picture 2. Forgot password interface*

### 1.2.2. Lock account when the login fails 5 times

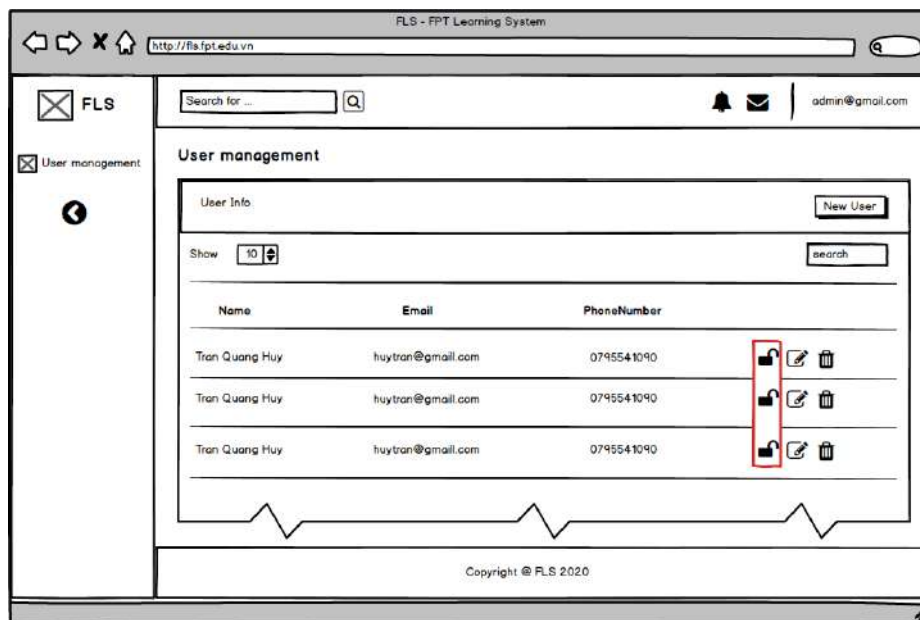
In order to increase the security of the system, as well as prevent outside hackers from harming users, this is a serious issue that needs to be developed.

When the user input email or password to login into system fail 5 times, the account status will be locked, and the only administrator can unlock it.



Picture 3. Login fails 5 times interface

With the role Administrator in the "User management" page, the Administrator has the ability to unlock users by click on the "unlock" button.



Picture 4. Administrator unlock user

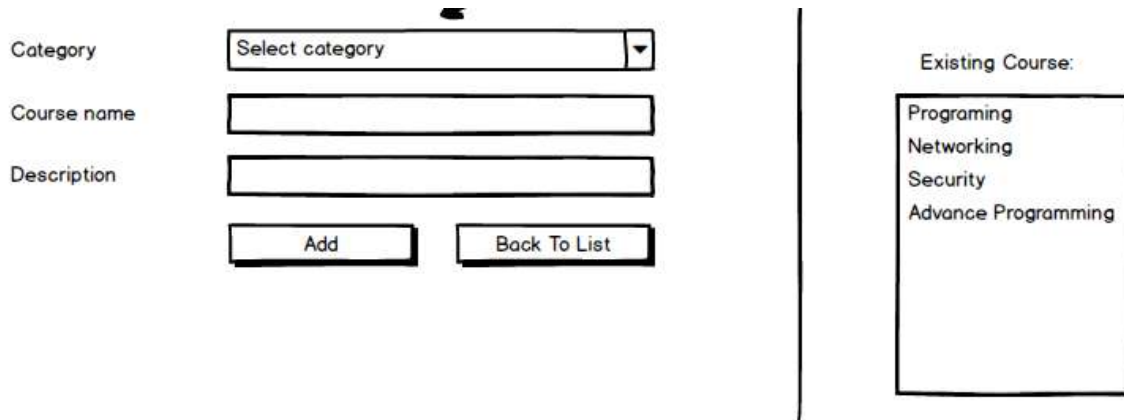
### 1.2.3. Change permission to view dashboard

Based on peer-review, only the Training Staff will need his system's statistics page, so other roles like Administrator, Trainer, Trainee will not be able to see the dashboard.

### 1.2.4. Improve Training Staff page

Base on peer-review, Training staff page will need to change with adding existing Course or topic table:

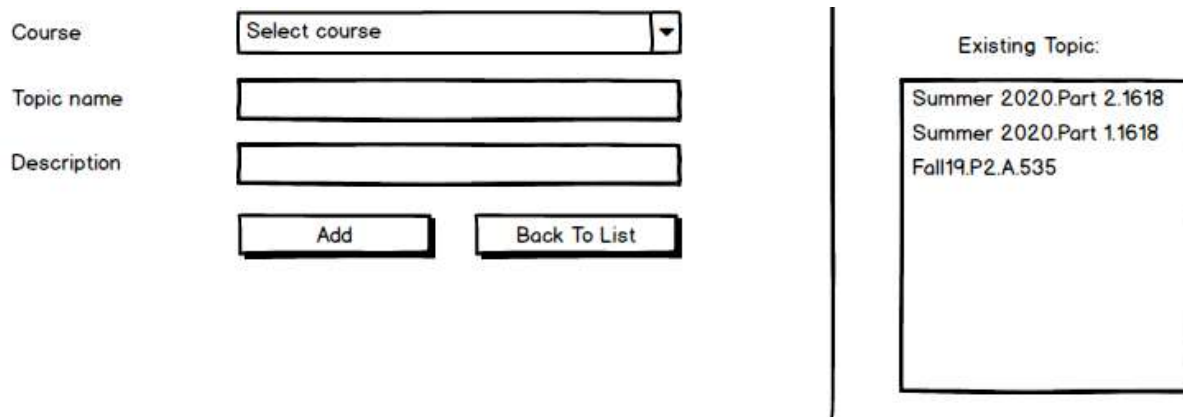
- **Sub-table to show Course, Topic exists in the database:**



The diagram shows a form for adding a new course and a table of existing courses. The form on the left has fields for Category (a dropdown menu), Course name, and Description, followed by 'Add' and 'Back To List' buttons. To the right, a table titled 'Existing Course:' lists four items: Programing, Networking, Security, and Advance Programming.

Existing Course:
Programing
Networking
Security
Advance Programming

Picture 5. Existing Course interface



The diagram shows a form for adding a new topic and a table of existing topics. The form on the left has fields for Course (a dropdown menu), Topic name, and Description, followed by 'Add' and 'Back To List' buttons. To the right, a table titled 'Existing Topic:' lists three items: Summer 2020.Part 2.1618, Summer 2020.Part 1.1618, and Fall19.P2.A.535.

Existing Topic:
Summer 2020.Part 2.1618
Summer 2020.Part 1.1618
Fall19.P2.A.535

Picture 6. Existing Topic interface



- Training staff can't delete trainer account:

### User management

User Info

New User

Show

10

search

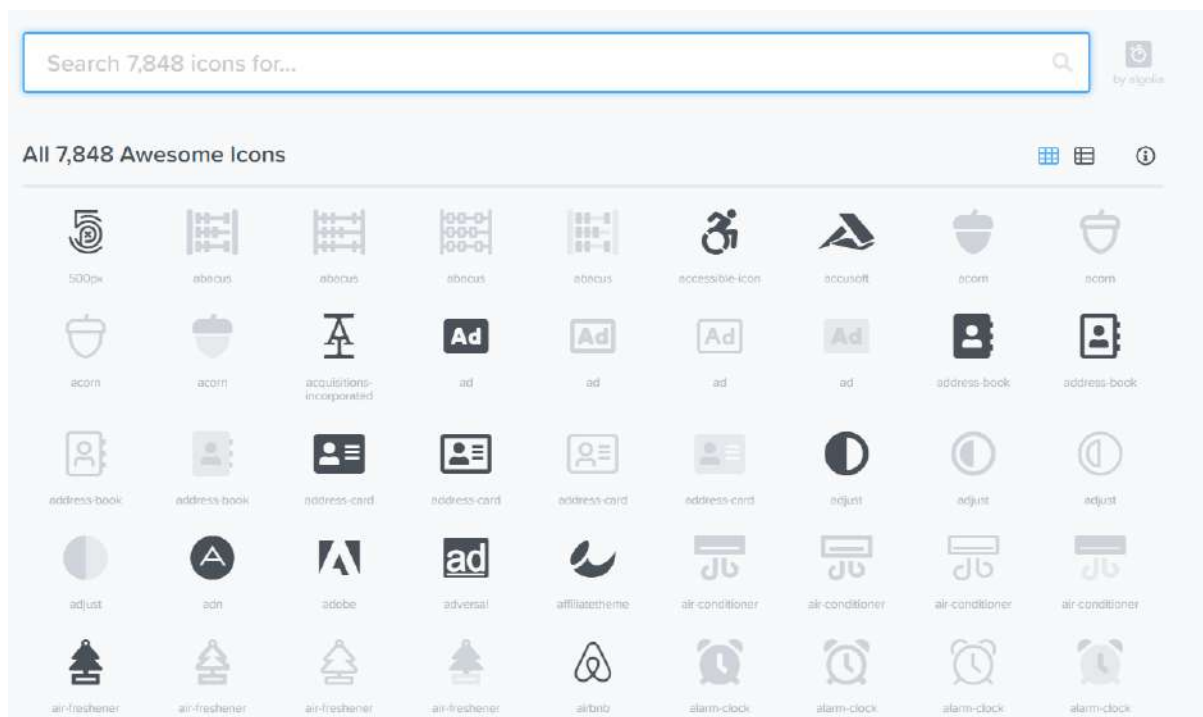
Name	Email	PhoneNumber	
Tran Quang Huy	huytran@gmail.com	0795541090	
Tran Quang Huy	huytran@gmail.com	0795541090	
Tran Quang Huy	huytran@gmail.com	0795541090	

Picture 7. Training staff user management interface

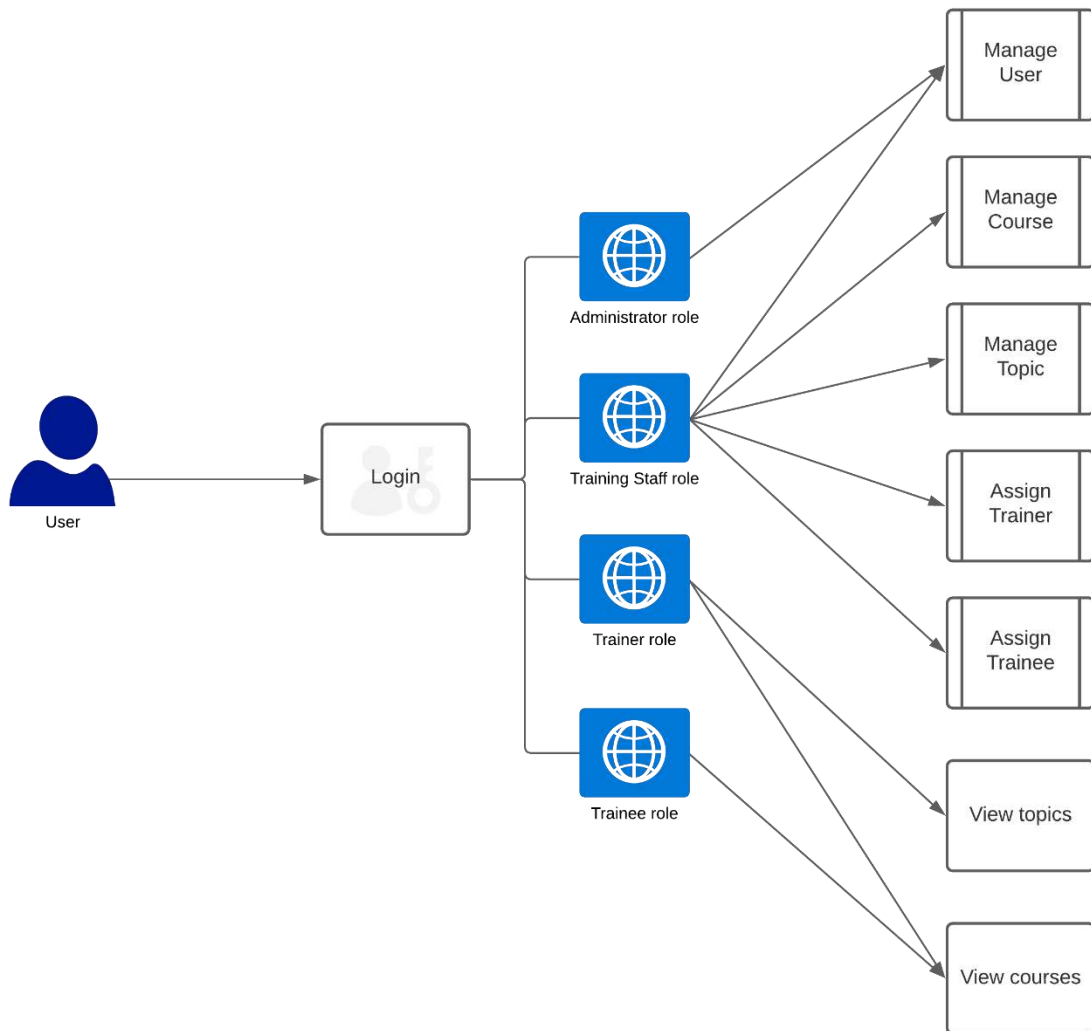
#### 1.2.5. Improve button with icon

The delete, edit, details buttons should be updated to increase the user experience.

Icons are available at <https://fontawesome.com/>.



**2. Application development**  
**2.1. Development application**  
**2.1.1. System Workflow**



*Figure 1. Overview system workflow*

## 2.1.2. Database

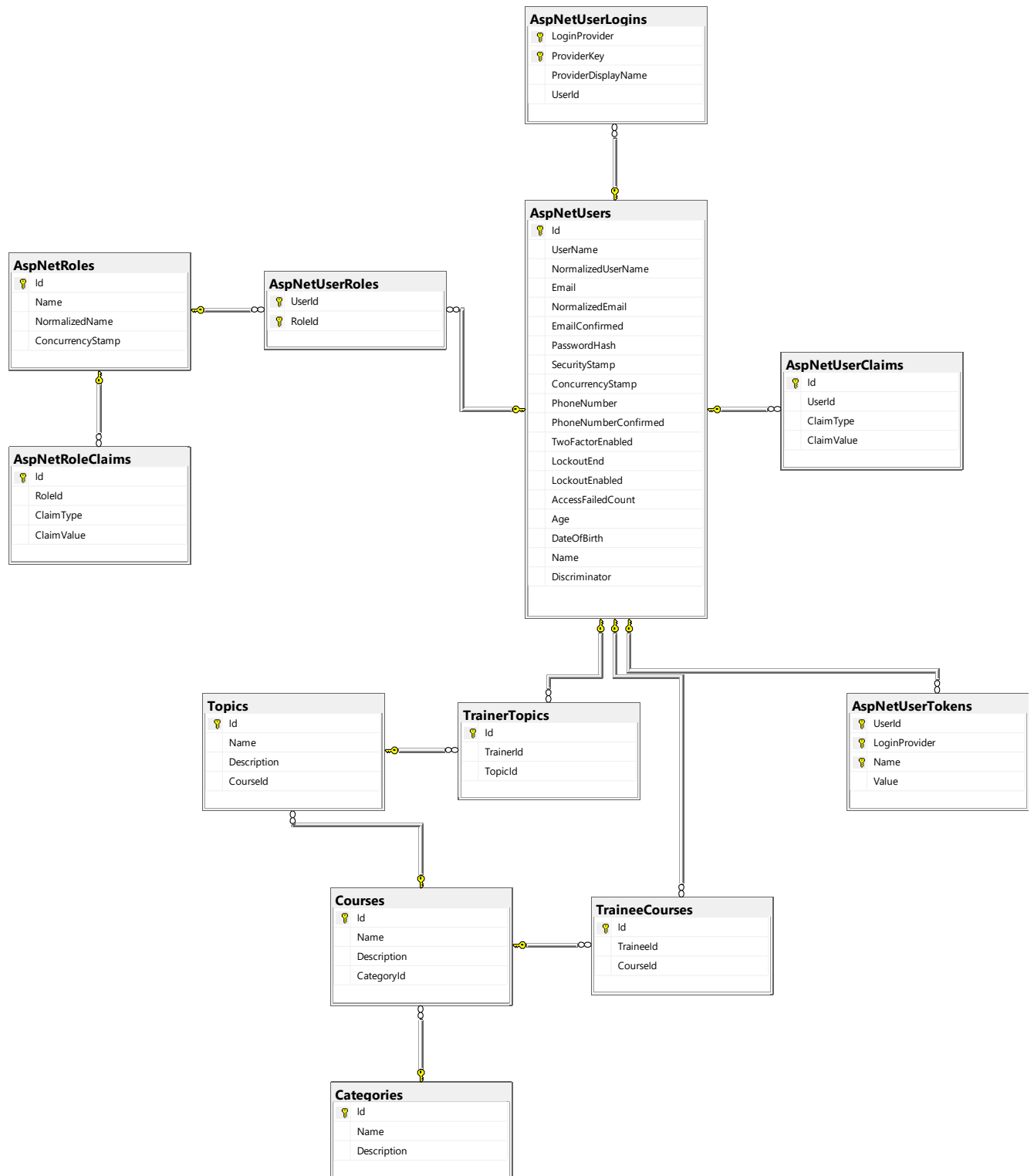
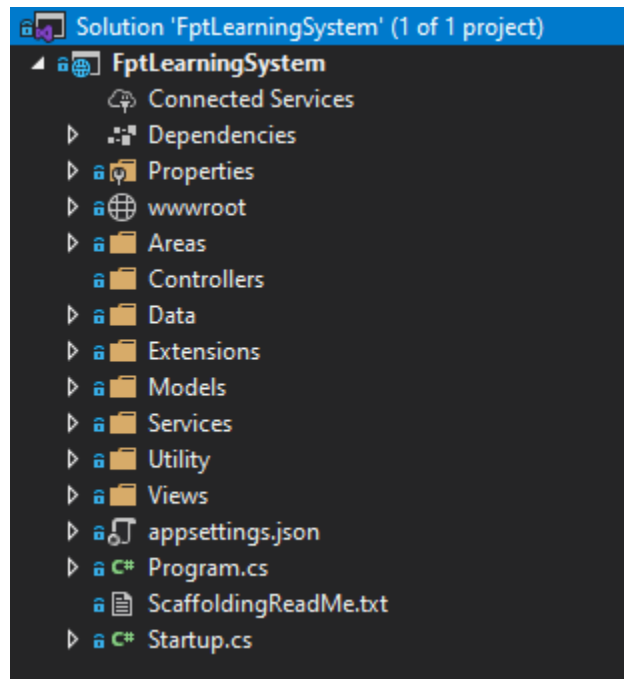


Figure 2. Entity-relationship model

### 2.1.3. System structure



Picture 8. FPT Learning System structure

- **MVC Architecture:**

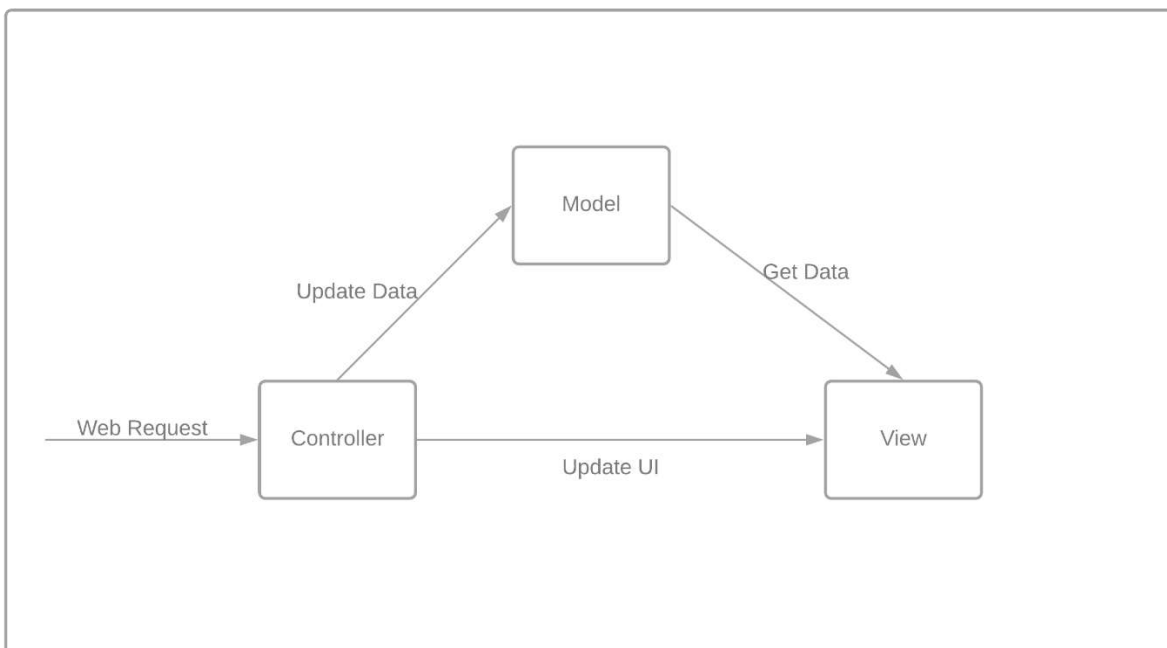


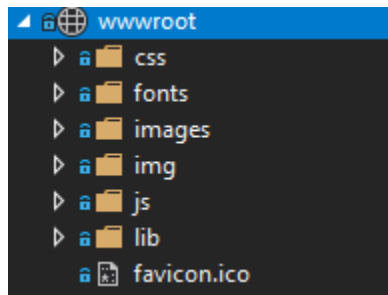
Figure 3. MVC Architecture

- Properties – launchSetting.json

```
{
  "iisSettings": {
    "windowsAuthentication": false,
    "anonymousAuthentication": true,
    "iisExpress": {
      "applicationUrl": "http://localhost:51097",
      "sslPort": 44363
    }
  },
  "profiles": {
    "IIS Express": {
      "commandName": "IISExpress",
      "launchBrowser": true,
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    },
    "FptLearningSystem": {
      "commandName": "Project",
      "launchBrowser": true,
      "applicationUrl": "https://localhost:5001;http://localhost:5000",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    }
  }
}
```

**launchSetting.json** Store application profile, developers also set the environment variables

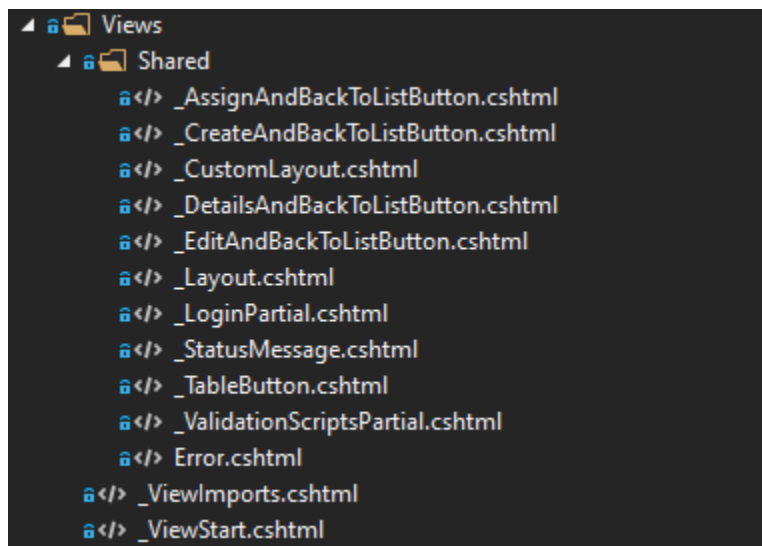
- **wwwroot:**



This is a new thing that has been introduced with ASP.NET core and this has been created automatically in this project.

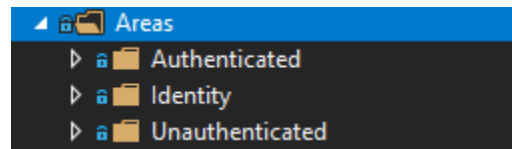
This folder will be placed on all the static files such as images, CSS, JavaScript, font, etc.

- **Razor page:**



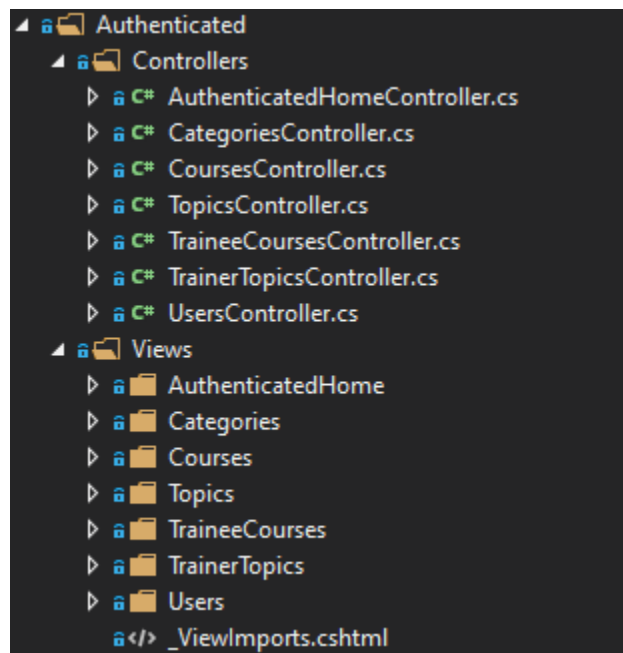
This folder store the default view of the system, besides that the `_layout.cshtml` will be reused in another view. Besides, the partial view will be stored here.

## Areas:



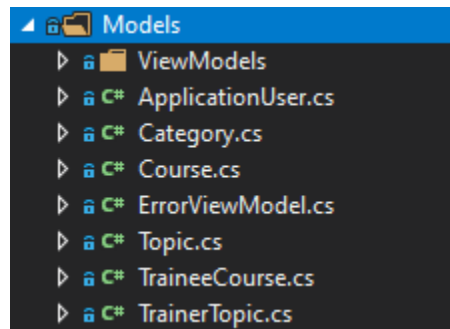
In ASP.NET core, there are areas to divide the application into multiple crucial components.

In each Areas has 2 mains folder such as Controllers and Views



The controllers moved in the corresponding area so that will as a separation. The controller goes with specific View to render the page on the website

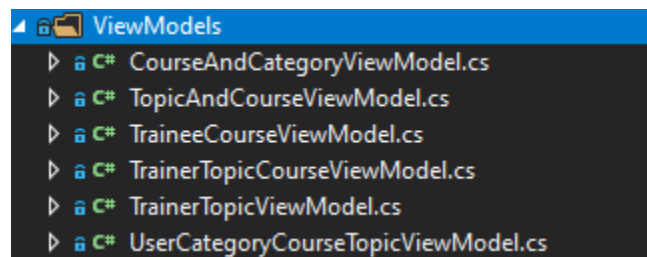
- **Model:**



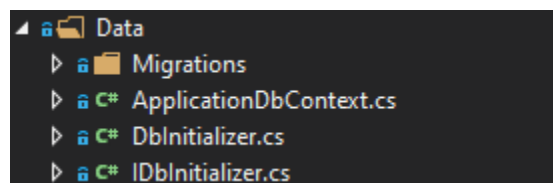
In the model, there are 2 part: Model and ViewModels

The model will be created base on ERD, each model presents the type of value in the database and the connection between Primary Key and Foreign Key.

The ViewModels can know that the Sub-Model can present for the model, and will be used in the special case in the controller



- **Data:**

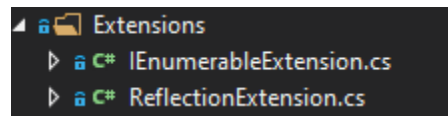


There is 3 main part of the system:

- **Migrations folder:** store the migration of the model, can be reused easily in another computer or cloud.
- **ApplicationDbContext:** In the database structures, the Dbset has a mission is to connect and work with the database, so the ApplicationDbContext like the gate to connect between application and database.
- **DbInitializer:** When uploading to the cloud to make more security to create Administrator users and initiating the Database, this is an important part to seed the database.

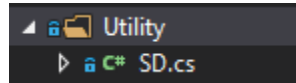


- **Extensions folder:**



This folder will store the Extensions that ASP.NET core didn't have. In this project, the Extensions will make a list for dropdown the ID of item and present the text base on the Item name.

- **Utility folder:**



This folder will help the system work with some special value. In this project, the Utility will store the variable role and name of the role. With this folder, developers can re-use in a lot of cases.

#### 2.1.4. Implement code and result

- Area: Unauthenticated

**Controller - HomeController:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Unauthenticated/Controllers/HomeController.cs>

**View – Home - Index:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Unauthenticated/Views/Home/Index.cshtml>

**Result – Home - Index:**



*Picture 9. UnAuthenticated - View - Home – Index*

- **Area: Identity**

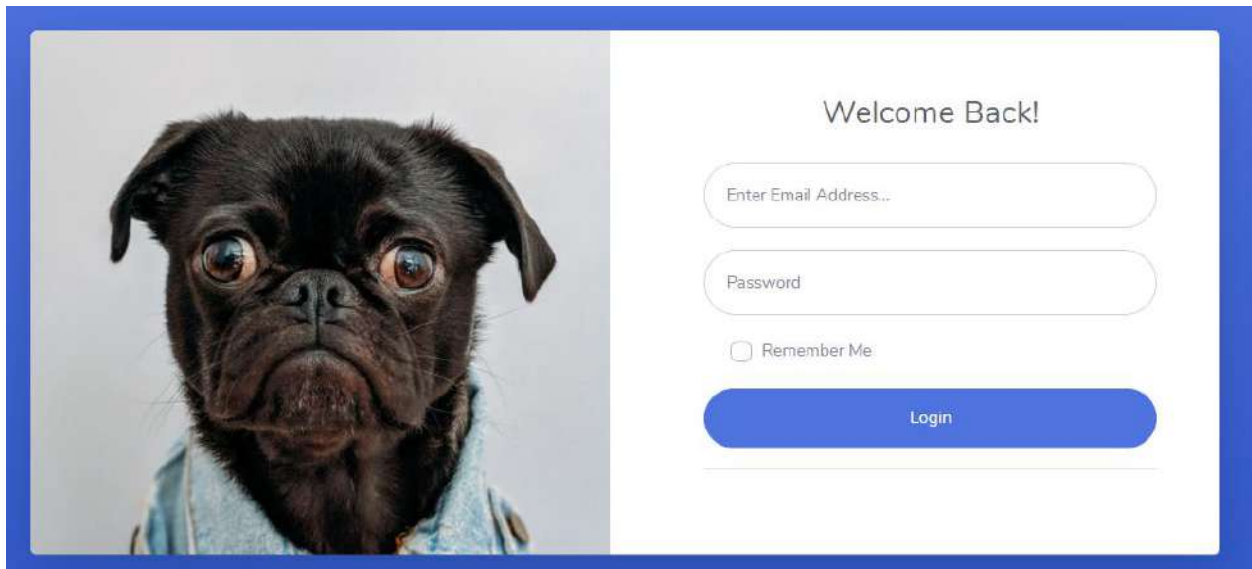
**Controller - Login:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Identity/Pages/Account/Login.cshtml.cs>

**View - login:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Identity/Pages/Account/Login.cshtml>

**The result – login:**



*Picture 10. Login interface*

**Controller – Register:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Identity/Pages/Account/Register.cshtml.cs>

**View – register:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Identity/Pages/Account/Register.cshtml>


**The result – Register:**

Create a New Account

Name

Email

Phone Number

Phone Number  

Age

Password

Confirm password

Role ☐ Trainer ☒ Training Staff

Create Account

Picture 11. Register interface

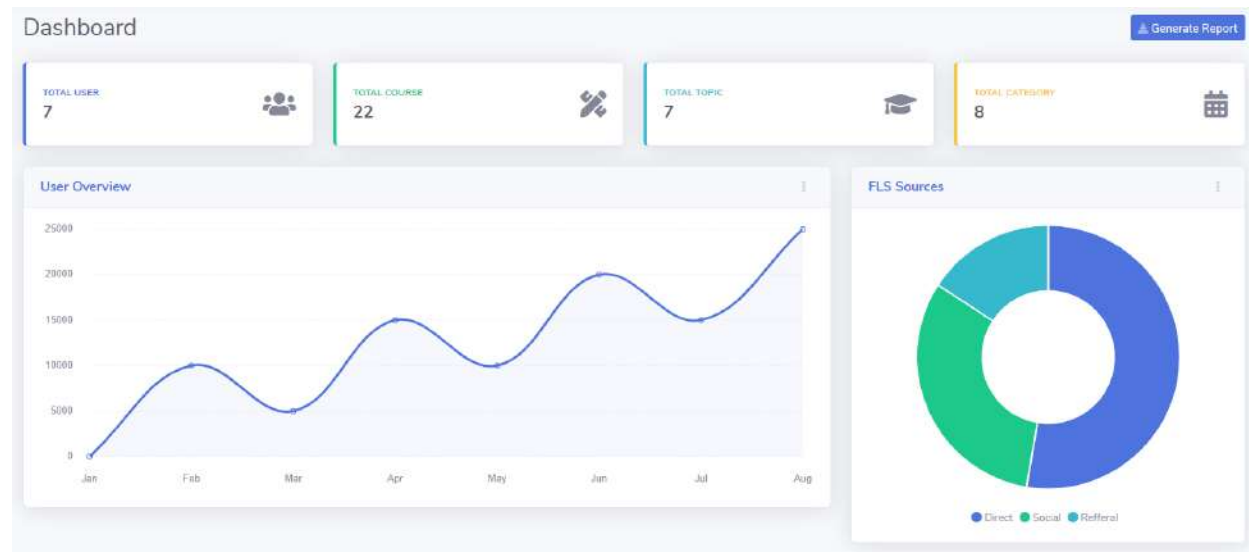
- **Area: Authenticated**

**Controller - AuthenticatedHomeController:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Controllers/AuthenticatedHomeController.cs>

**Result AuthenticatedHomeController – View – Index:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/AuthenticatedHome/Index.cshtml>



Picture 12. Authenticated Index

## Controller - CategoriesController:

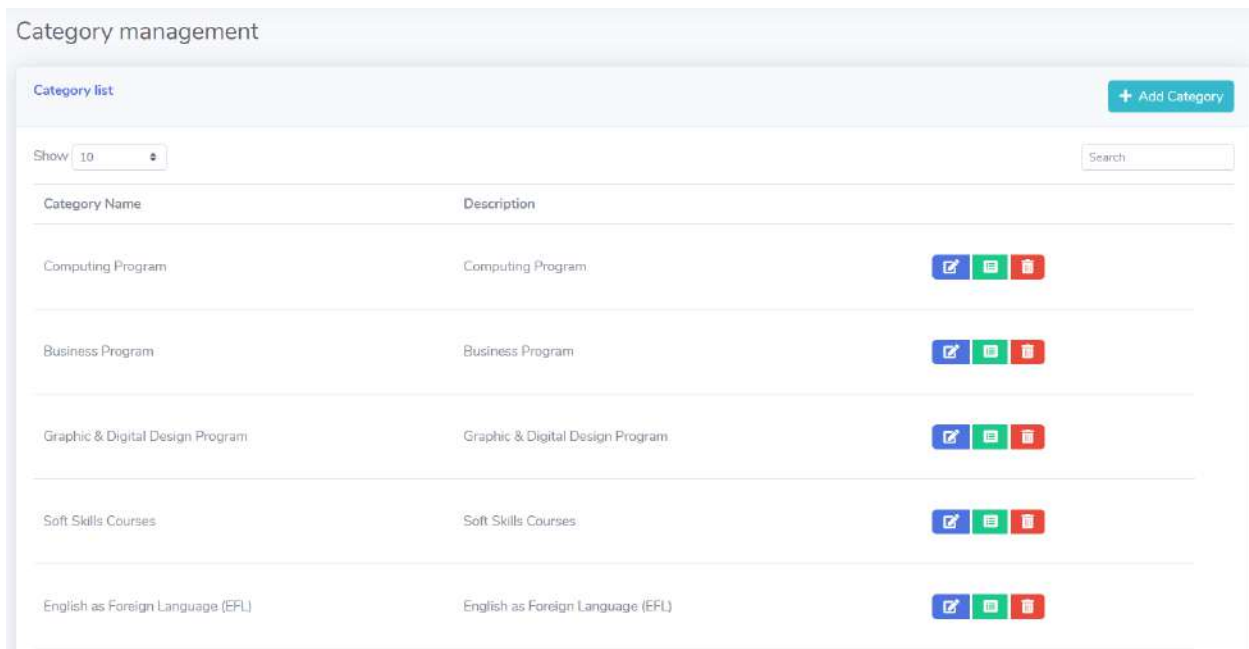
<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Controllers/CategoriesController.cs>

## View – CategoriesController:

### Index:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/Categories/Index.cshtml>

### Result view Categories Index:

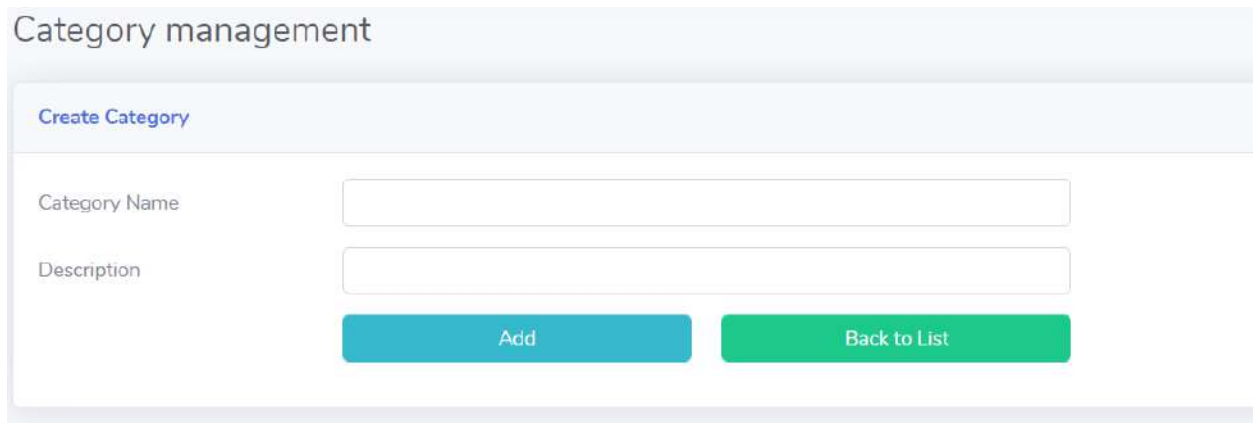


Picture 13. Category index interface

**Create:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/Categories/Create.cshtml>

**Result view Categories Create:**



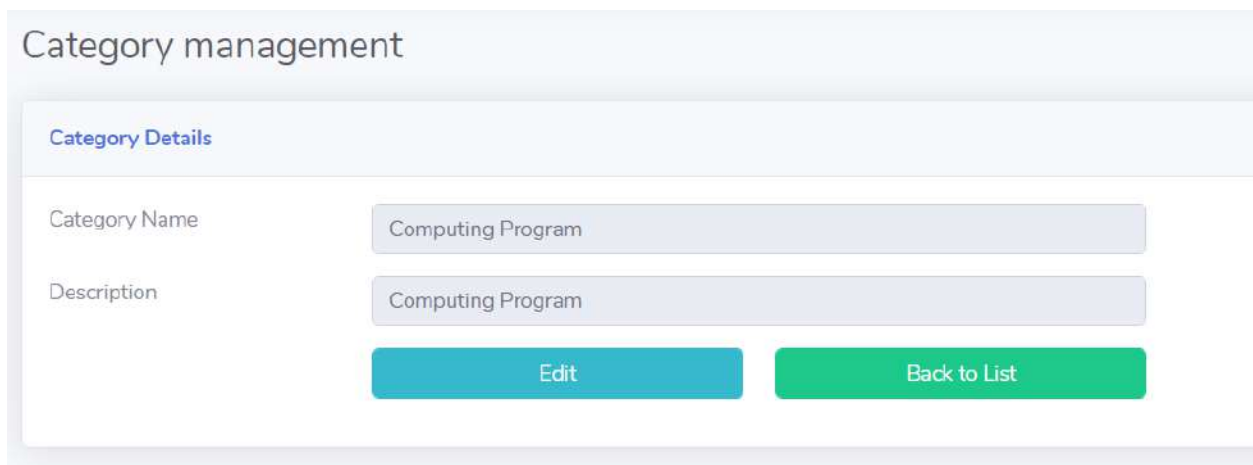
The screenshot shows a web interface titled "Category management". Below the title is a section labeled "Create Category". This section contains two input fields: "Category Name" and "Description". Below these fields are two buttons: a blue "Add" button and a green "Back to List" button.

Picture 14. Category create interface

**Details:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/Categories/Details.cshtml>

**Result categories Details:**



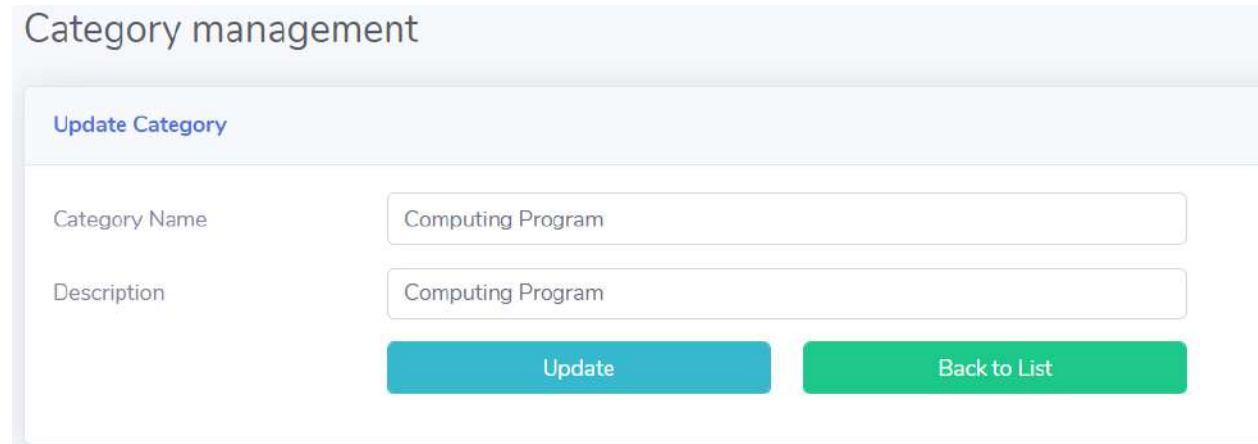
The screenshot shows a web interface titled "Category management". Below the title is a section labeled "Category Details". This section displays the details for a category named "Computing Program". The "Category Name" and "Description" fields both show "Computing Program". Below these fields are two buttons: a blue "Edit" button and a green "Back to List" button.

Picture 15. Category details interface

**Edit:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/Categories/Edit.cshtml>

**Result categories Edit:**



The screenshot shows a web interface for editing a category. At the top, there is a header "Category management". Below it, a sub-header "Update Category" is displayed. The form contains two input fields: "Category Name" and "Description", both containing the text "Computing Program". At the bottom of the form, there are two buttons: a blue "Update" button and a green "Back to List" button.

*Picture 16. Category edit interface*



### Controller – CoursesController:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Controllers/CoursesController.cs>

### View – CoursesController:

#### Index:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/Courses/Index.cshtml>

#### Result Courses index:

Course management

Course list			<a href="#">+ Add Course</a>
Show 10	Search		
Course Name	Category Name	Description	
Programming	Computing Program	Programming	<a href="#">Edit</a> <a href="#">View</a> <a href="#">Delete</a>
Networking	Computing Program	Networking	<a href="#">Edit</a> <a href="#">View</a> <a href="#">Delete</a>
Professional Practice	Computing Program	Professional Practice	<a href="#">Edit</a> <a href="#">View</a> <a href="#">Delete</a>
Database Design and Development	Computing Program	Database Design and Development	<a href="#">Edit</a> <a href="#">View</a> <a href="#">Delete</a>

Picture 17. Courses index interface

**Create:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/Courses/Create.cshtml>

**Result Course create:**

Course management

Create Course

Category:

Course Name:

Description:

Existing Course:

Programming
Networking
Professional Practice
Database Design and Development
Security
Software Development Life Cycle (SDLC)
Advance Programming
Application Development

Picture 18. Course create interface

**Details:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/Courses/Details.cshtml>

**Result Course Details:**

Course management

Course Details

Course Name:

Category Name:

Description:

Picture 19. Course Details interface

**Edit:**

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authentication/Views/Courses/Edit.cshtml>

**Result course edit:**

Course management

Update Course

Category: Computing Program

Course Name: Programming

Description: Programming

Update Back to List

Existing Course:

Programming
Networking
Professional Practice
Database Design and Development
Security
Software Development Life Cycle (SDLC)
Advance Programming
Application Development

Picture 20. Course edit interface

### Controller – TopicsController:

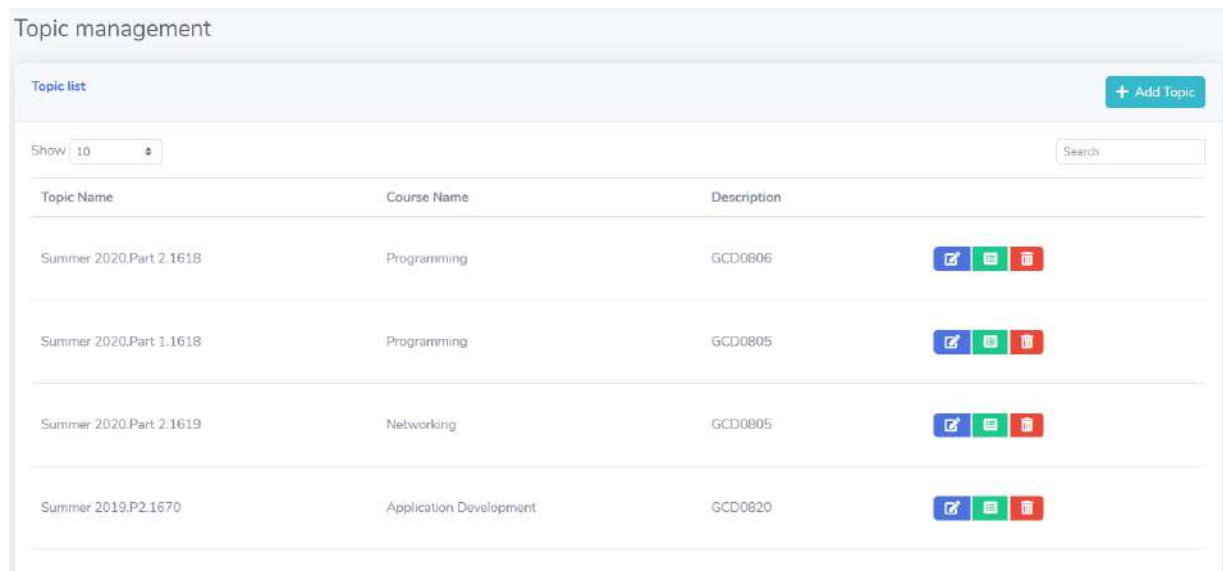
<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Controllers/TopicsController.cs>

### View – TopicsController:













#### Index:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/Topics/Index.cshtml>

#### Result topic index:



The screenshot shows a web interface titled "Topic management". It features a "Topic list" section with a "+ Add Topic" button in the top right corner. Below the button is a "Show" dropdown menu set to "10" and a "Search" input field. The main content is a table with three columns: "Topic Name", "Course Name", and "Description". There are four rows of data, each with a set of three action icons (edit, view, delete) on the right.

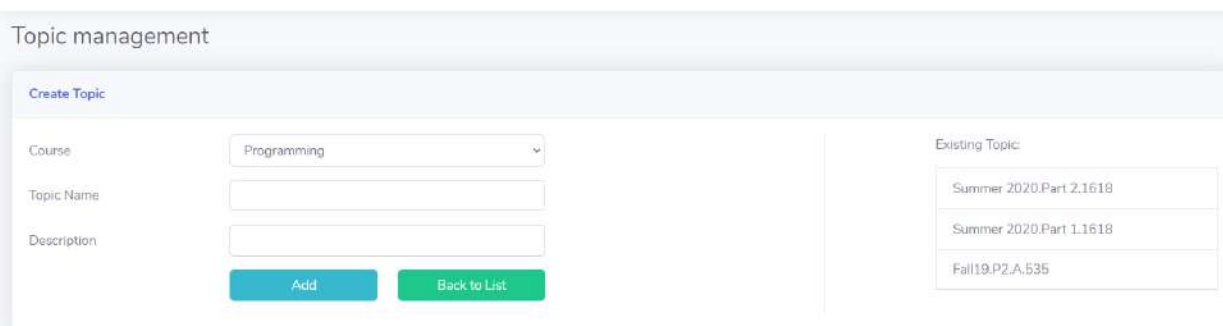
Topic Name	Course Name	Description	
Summer 2020.Part 2.1618	Programming	GCD0806	  
Summer 2020.Part 1.1618	Programming	GCD0805	  
Summer 2020.Part 2.1619	Networking	GCD0805	  
Summer 2019.P2.1670	Application Development	GCD0820	  

Picture 21. Topic index interface

#### Create:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/Topics/Create.cshtml>

#### Result Topic create:



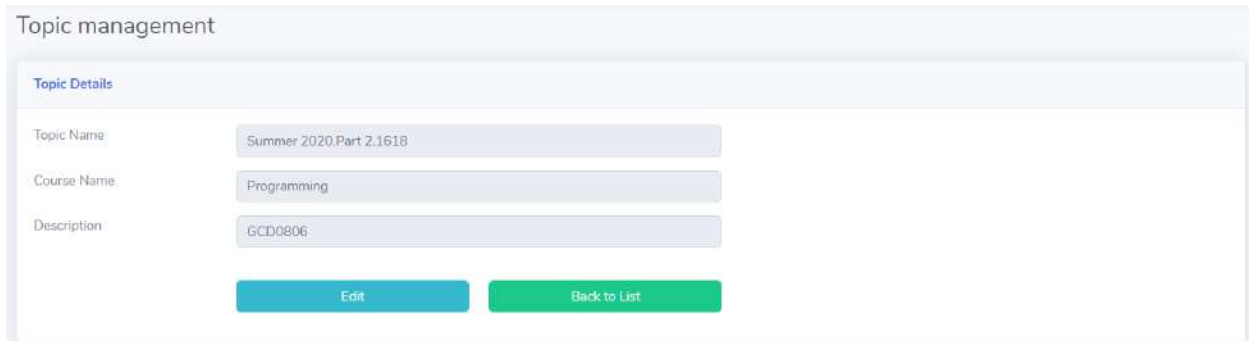
The screenshot shows a web interface titled "Topic management". It features a "Create Topic" section with a form for creating a new topic. The form has three input fields: "Course" (a dropdown menu with "Programming" selected), "Topic Name", and "Description". Below the fields are two buttons: "Add" and "Back to List". To the right of the form is a section titled "Existing Topic:" with a list of three topics: "Summer 2020.Part 2.1618", "Summer 2020.Part 1.1618", and "Fall19.P2.A.535".

Picture 22. Topic create interface

### Details:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/Topics/Details.cshtml>

### Result topic details:



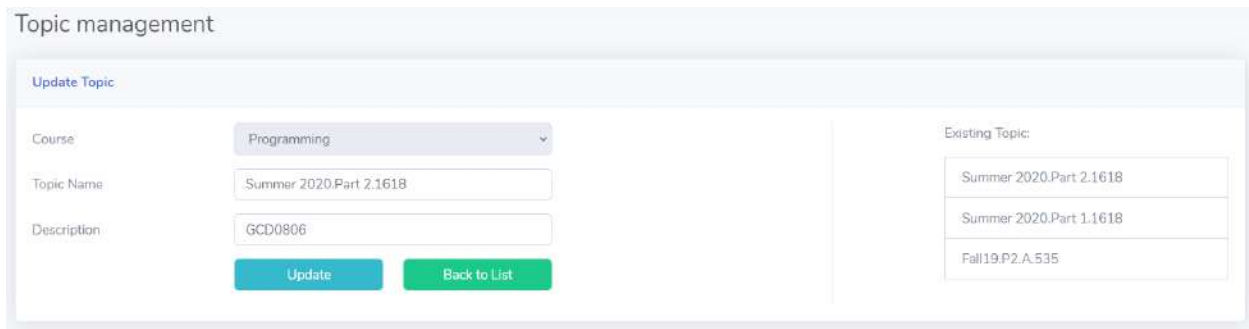
The screenshot shows a web interface titled "Topic management". Below the title is a section labeled "Topic Details". This section contains three text input fields: "Topic Name" with the value "Summer 2020.Part 2.1618", "Course Name" with the value "Programming", and "Description" with the value "GCD0806". Below these fields are two buttons: a blue "Edit" button and a green "Back to List" button.

Picture 23. Topic details interface

### Edit:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/Topics/Edit.cshtml>

### Result topic edit:



The screenshot shows a web interface titled "Topic management". Below the title is a section labeled "Update Topic". This section is divided into two main areas. On the left, there are three input fields: "Course" (a dropdown menu showing "Programming"), "Topic Name" (containing "Summer 2020.Part 2.1618"), and "Description" (containing "GCD0806"). Below these fields are two buttons: a blue "Update" button and a green "Back to List" button. On the right, there is a section titled "Existing Topic:" which contains a table with three rows of existing topics:

Summer 2020.Part 2.1618
Summer 2020.Part 1.1618
Fall19.P2.A.535

Picture 24. Training staff topic edit interface.

### Controller – TraineeCoursesController:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Controllers/TraineeCoursesController.cs>

### View – TraineeCoursesController:

#### Index:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/TraineeCourses/Index.cshtml>

#### Result TraineeCourses index:



The screenshot shows a web interface titled "Assign Trainee". It features a table with the heading "Assigned Trainee List". In the top right corner of the table area, there is a button labeled "+ Assign Trainee To Course". The table has two columns: "Trainee" and "Course". It contains two rows of data:

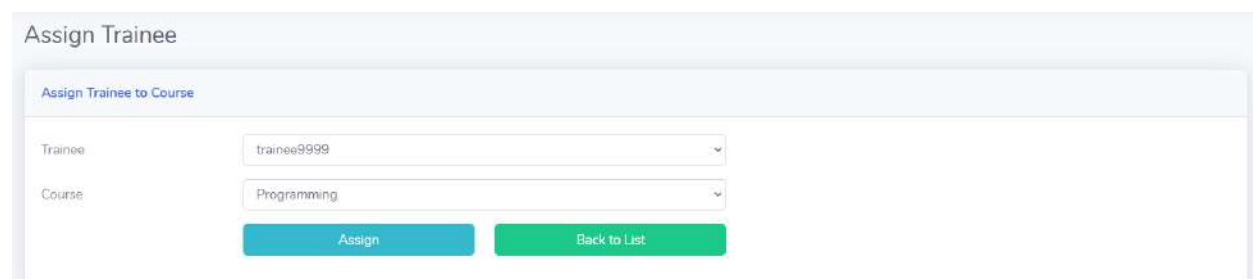
Trainee	Course
trainee9999	Professional Practice
trainee9999	Programming

Picture 25. Training Staff Assign Trainee into Course interface

#### Create:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/TraineeCourses/Create.cshtml>

#### Result TraineeCourses create:



The screenshot shows a web interface titled "Assign Trainee". It features a form with the heading "Assign Trainee to Course". The form has two dropdown menus: "Trainee" with the value "trainee9999" and "Course" with the value "Programming". Below the dropdowns are two buttons: "Assign" (blue) and "Back to List" (green).

Picture 26. TraineeCourses create interface

### Controller – TrainerTopicsController:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Controllers/TrainerTopicsController.cs>

### View – TrainerTopicsController:

#### Index:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/TrainerTopics/Index.cshtml>

#### Result TrainerTopicsController index:

Assign Trainer

Assigned Trainer List

+ Assign Trainer To Topic

Trainer	Topic	
trainer	Summer 2020.Part 1.1618	
trainer1	Summer 2020.Part 2.1618	
trainer1	Summer 2020.Part 2.1619	
trainer	Spring20.P2.538	
trainer	Summer 2020.Part 2.1618	

Picture 27. TrainerTopicsController index interface

#### Create:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/TrainerTopics/Create.cshtml>

#### Result TrainerTopics Create:

Assign Trainer

Assign Trainer To Topic

Trainer

Topic

Picture 28. TrainerTopics Create interface

## Controller – UsersController:

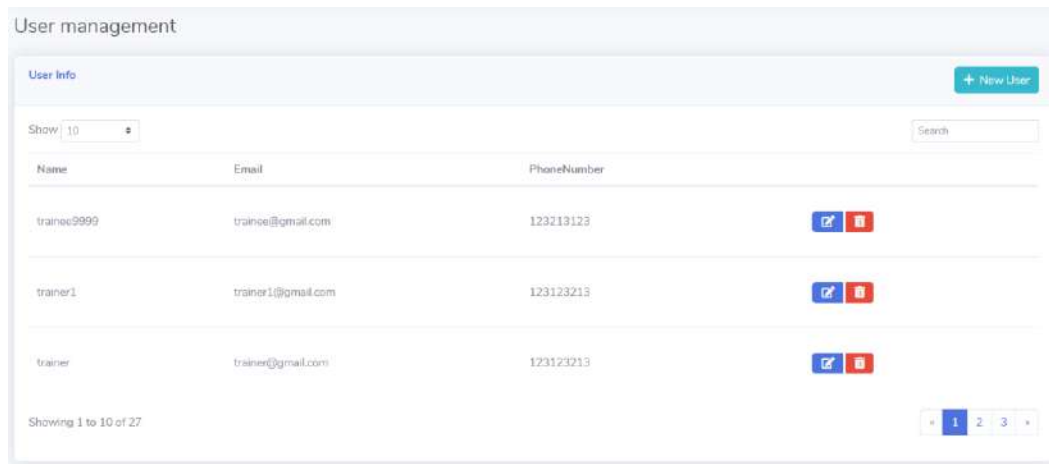
<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Controllers/UsersController.cs>

## View UsersController:

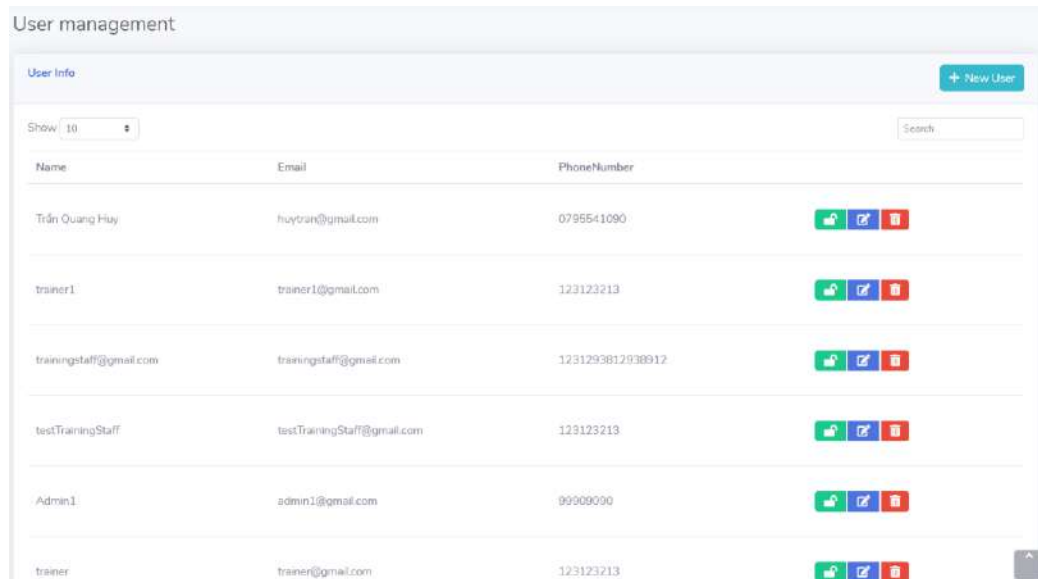
### Index:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/Users/Index.cshtml>

## Result UsersController index:



Picture 29. UsersController index interface with role TrainingStaff



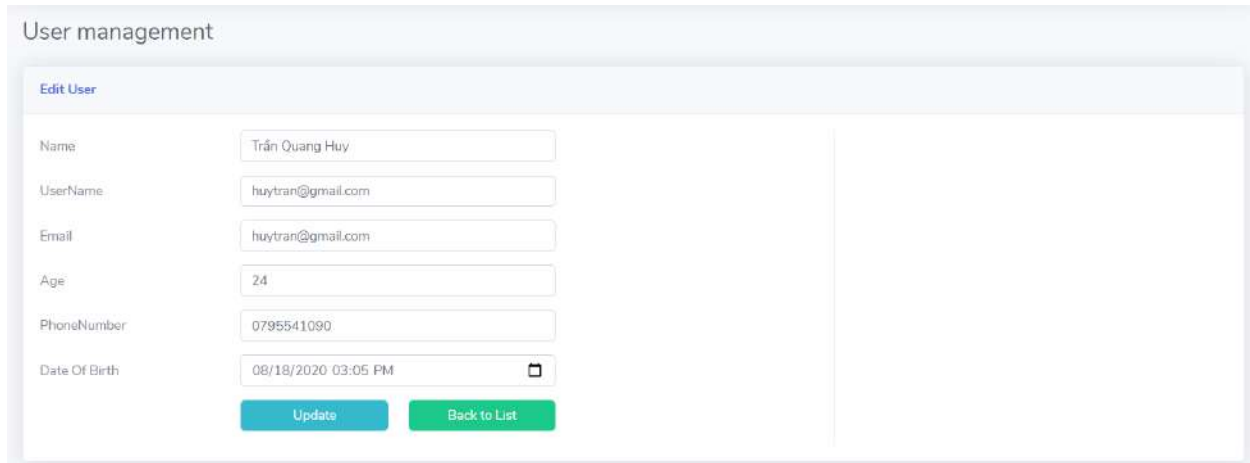
Picture 30. UsersController index interface with role Administrator



## Edit:

<https://github.com/SuperMido/FptLearningSystem/blob/master/FptLearningSystem/Areas/Authenticated/Views/Users/Edit.cshtml>

## Result UsersController Edit:



User management

Edit User

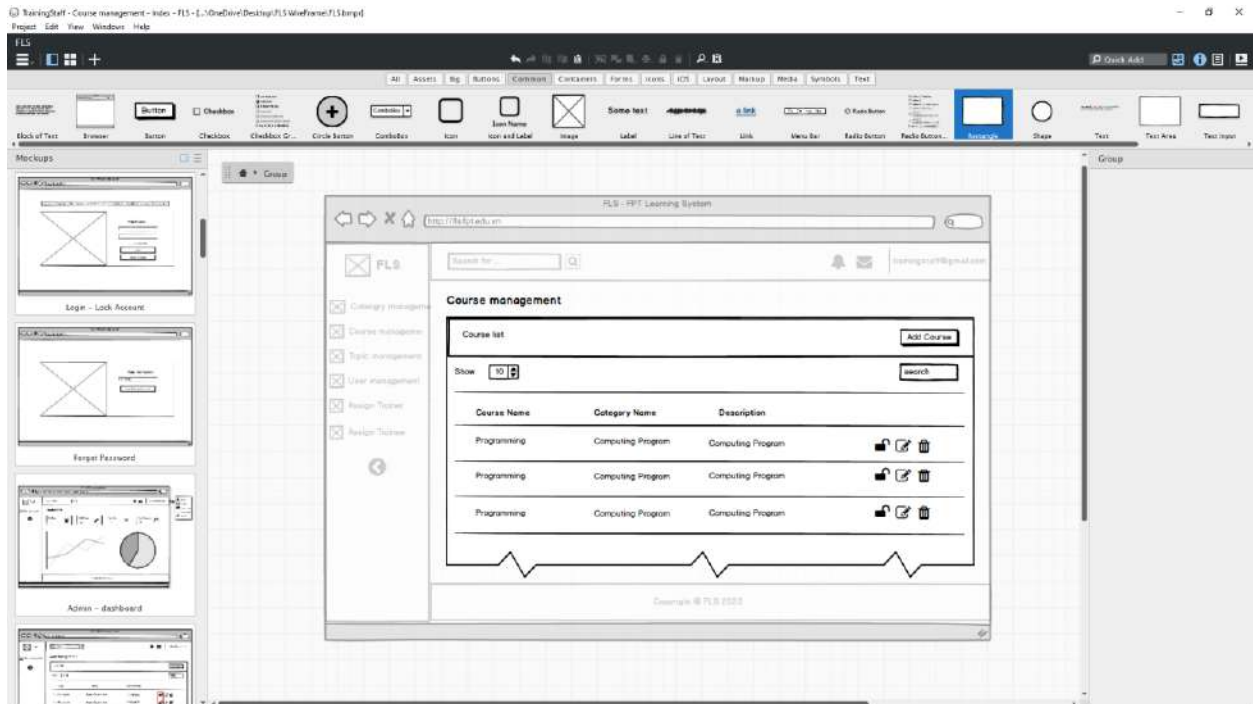
Name	<input type="text" value="Trần Quang Huy"/>
UserName	<input type="text" value="huytran@gmail.com"/>
Email	<input type="text" value="huytran@gmail.com"/>
Age	<input type="text" value="24"/>
PhoneNumber	<input type="text" value="0795541090"/>
Date Of Birth	<input type="text" value="08/18/2020 03:05 PM"/>

Picture 31. UsersController Edit interface

## 2.2. Tools, techniques, and methodologies

Example using tools, techniques, and methodologies for creating “Training Staff view Course” function:

- Balsamiq mockups 3:



Picture 32. Training Staff view Course with Balsamiq mockups 3

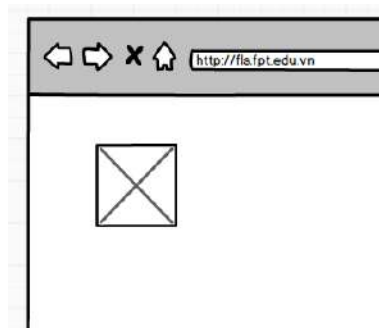
In Balsamiq mockup 3 the Browser container, image, Label, Rectangle, text input, Icon, button, and ComboBox were used.

- **Browser container:**



Picture 33. Balsamiq browser container

- **Image:**



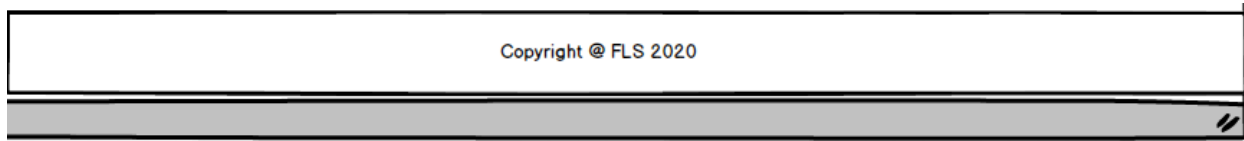
Picture 34. Balsamiq Image

- **Label**

Course Name	Category Name	Description
Programming	Computing Program	Computing Program
Programming	Computing Program	Computing Program
Programming	Computing Program	Computing Program

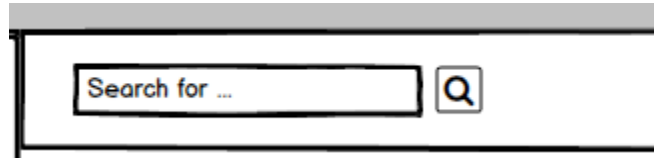
Picture 35. Balsamiq Label

- **Rectangle:**



Picture 36. Balsamiq Rectangle

- **Text Input:**



Picture 37. Balsamiq Text Input

- **Icon:**



Picture 38. Balsamiq Icon

- **Button:**



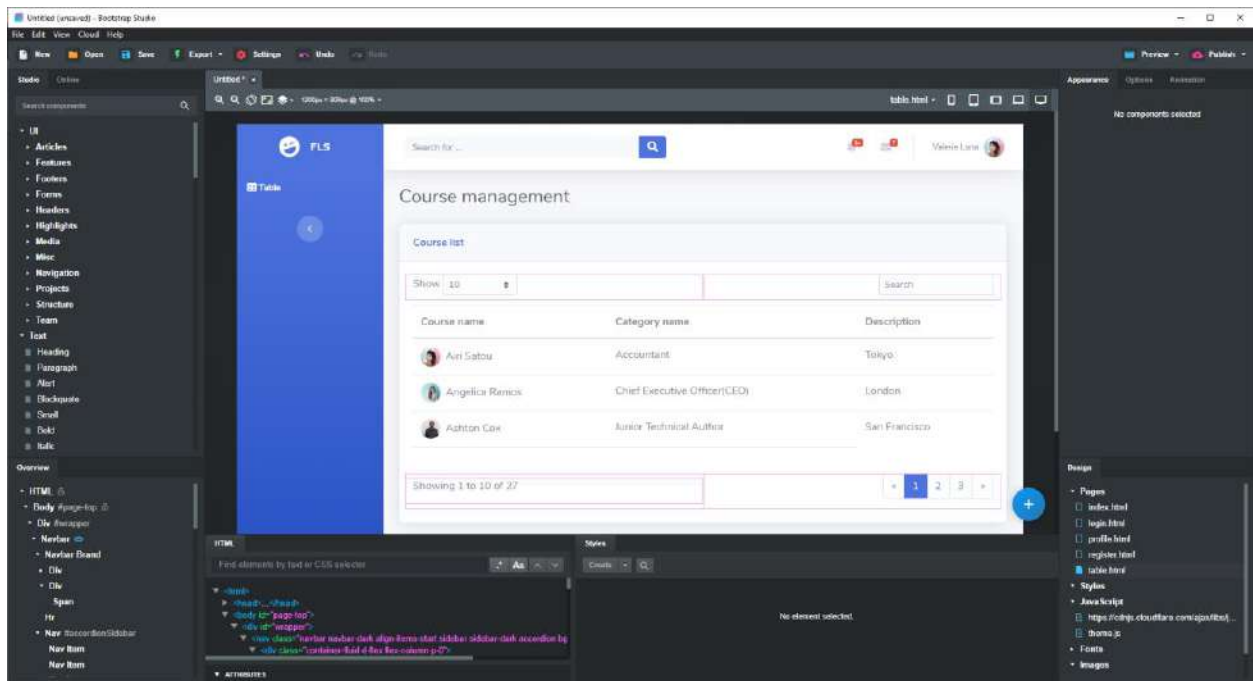
Picture 39. Balsamiq Button

- **ComboBox:**



Picture 40. Balsamiq Combobox

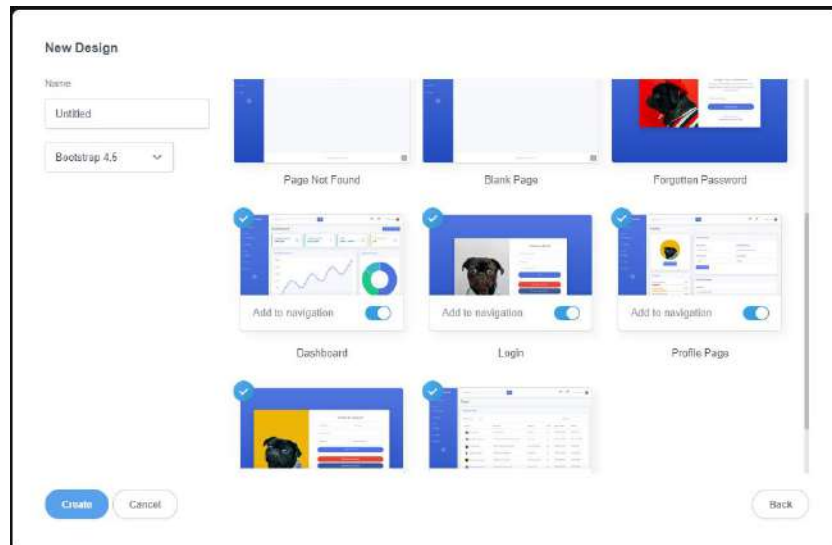
- **Bootstrap studio:**



Picture 41. Training Staff view Course with bootstrap studio

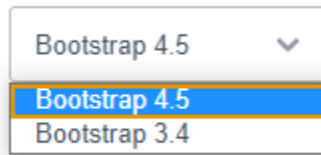
Based on the template provided by Bootstrap Studio, div tags and tags are added. Besides, the JavaScript libraries are added in header and footer tags.

- **Select Template:**



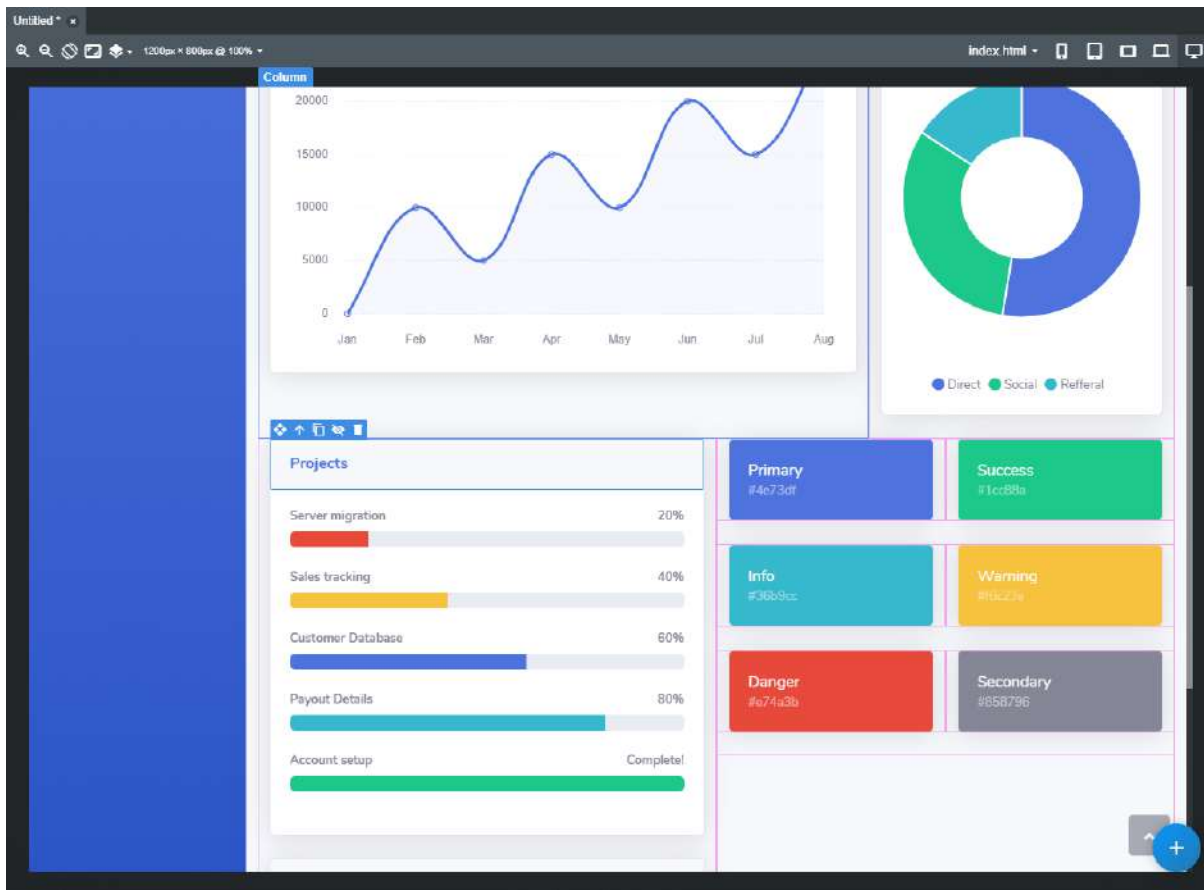
Picture 42. Bootstrap studio Select template

- **Select the Bootstrap version:**



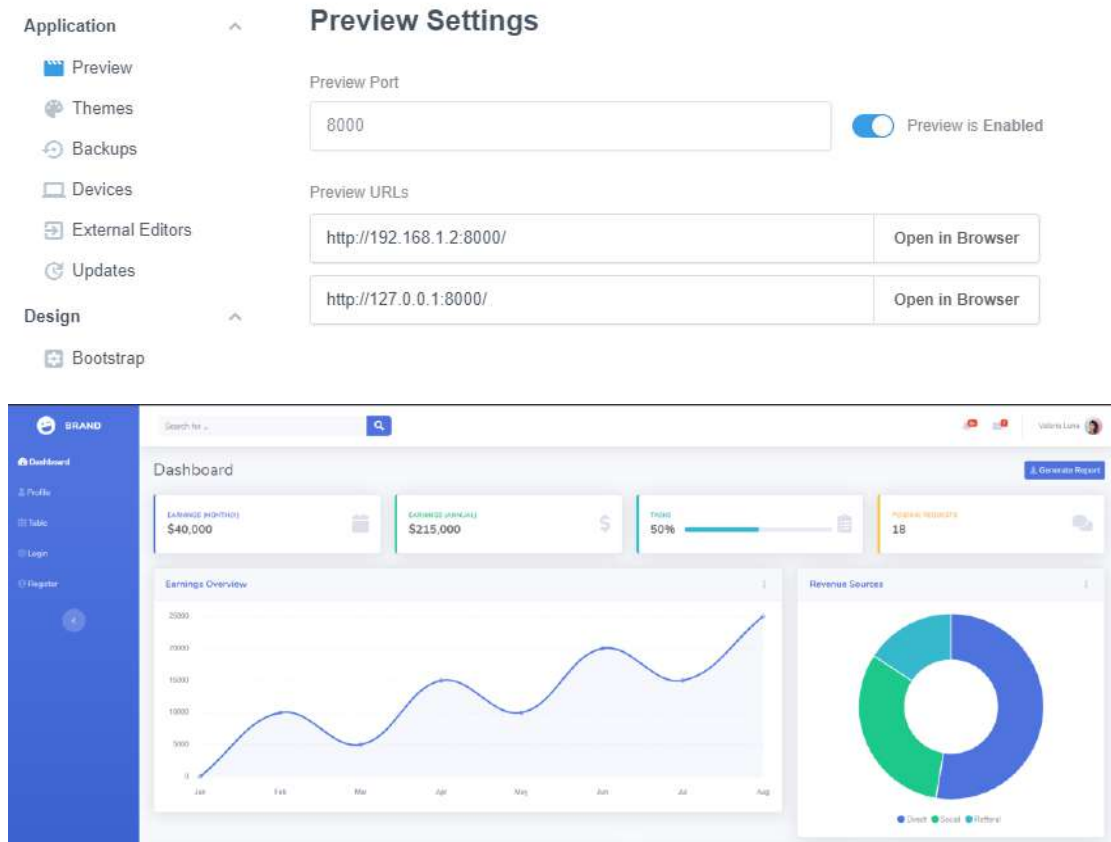
Picture 43. Bootstrap studio selects the bootstrap version

- **Modify template:**



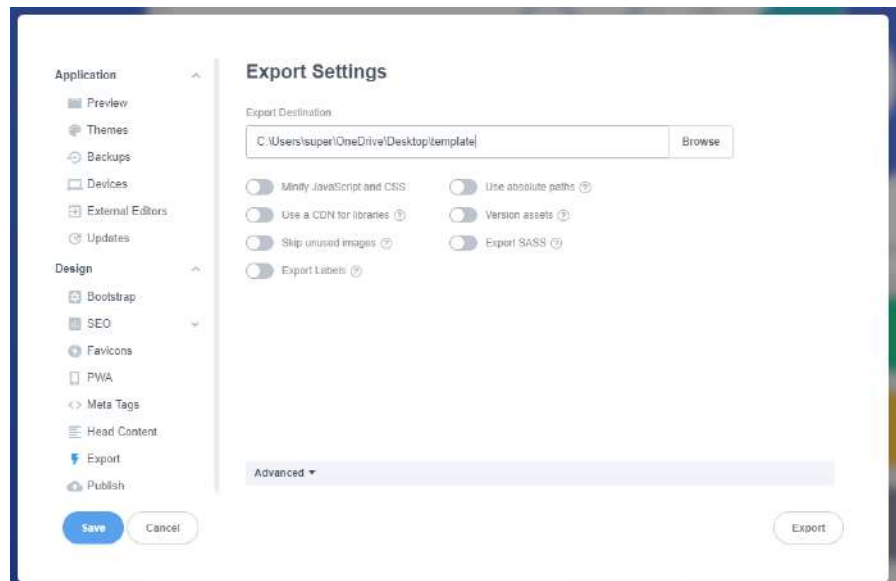
Picture 44. Bootstrap Studio modify the template

- **Preview template on server local with 800 port:**



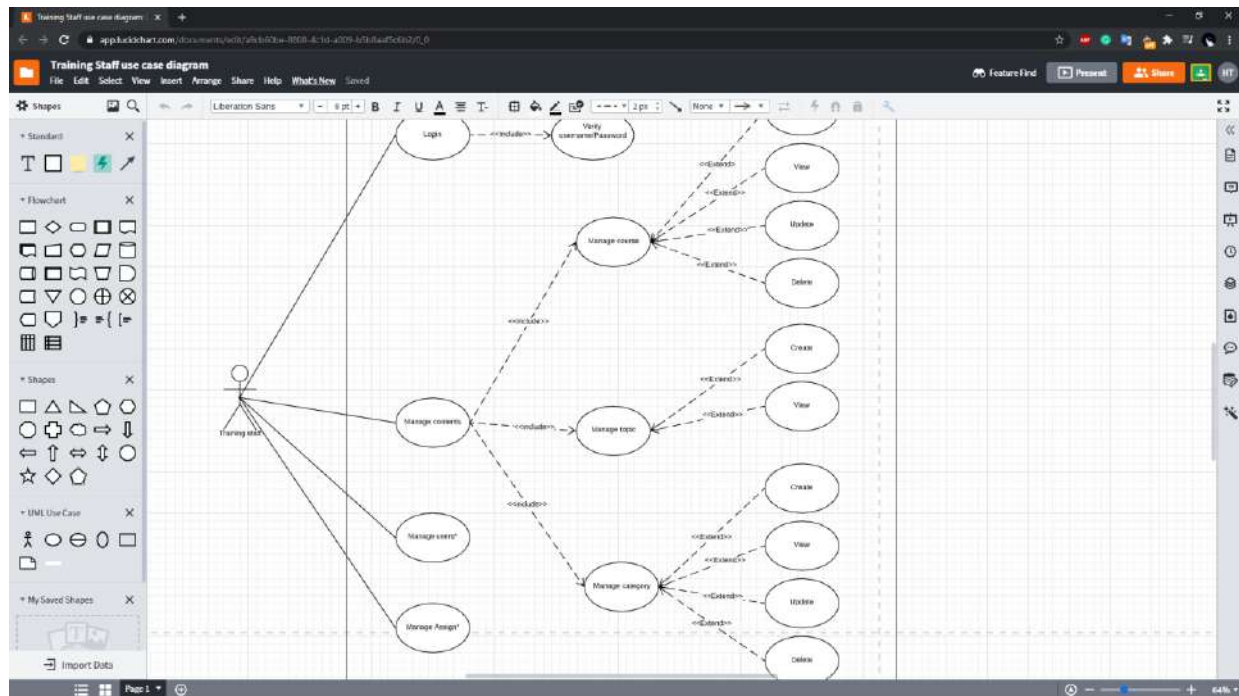
Picture 45. Bootstrap Studio preview template

- **Export Bootstrap:**



Picture 46. Bootstrap Studio export

- **Lucid Chart:**

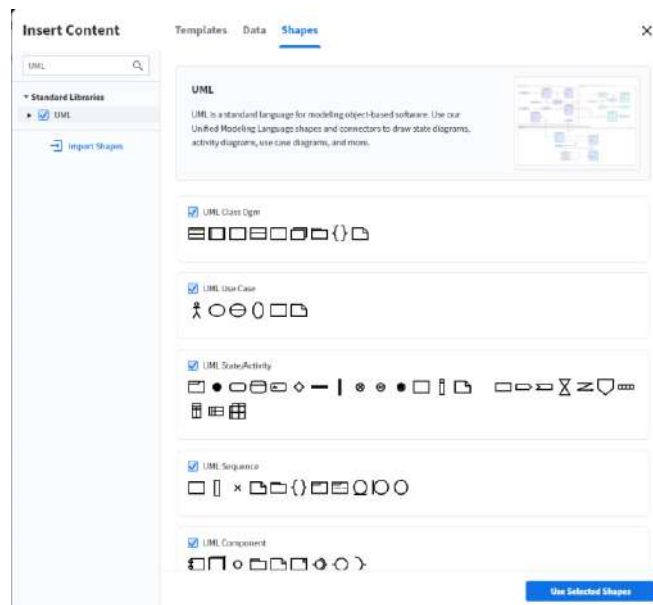


Picture 47. Training Staff view Course with lucid Chart

By using Lucid Chart, the Use Case is important for this project to help developers know what Training staff can do which specific action.

In lucid Chart, the UML Use Case was used to develop it.

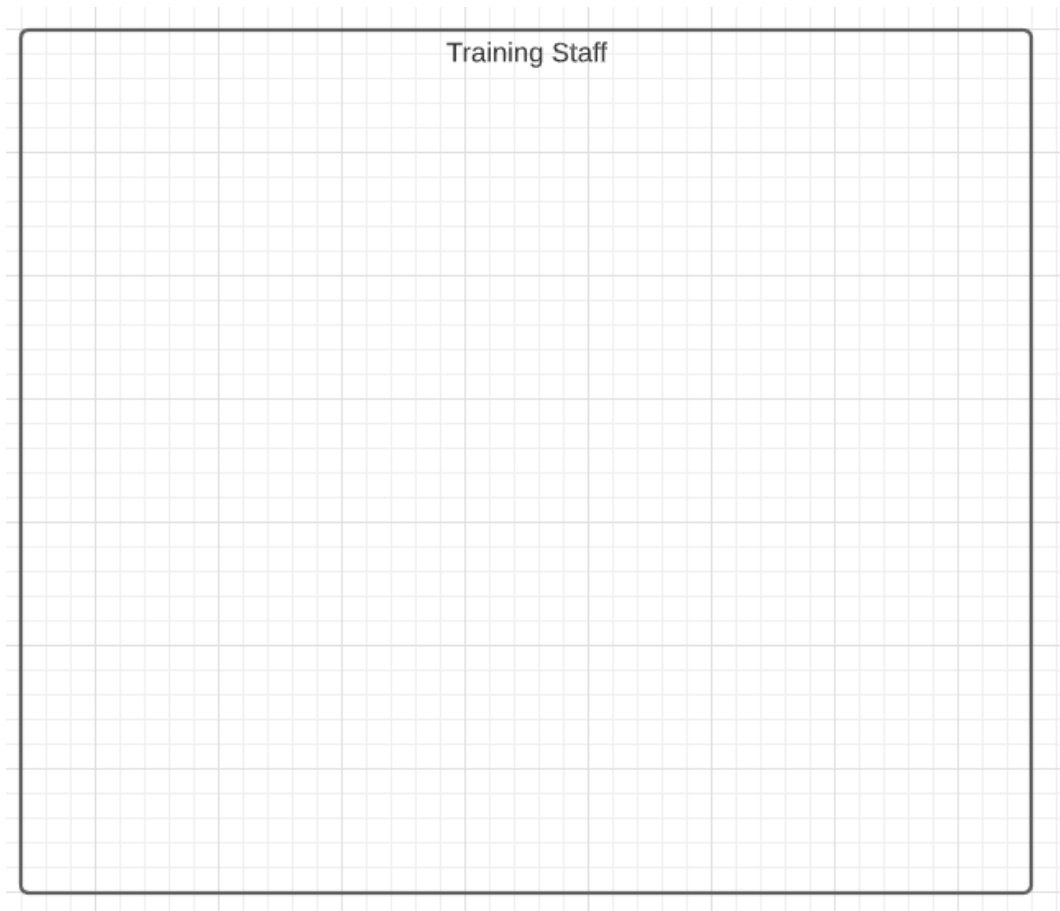
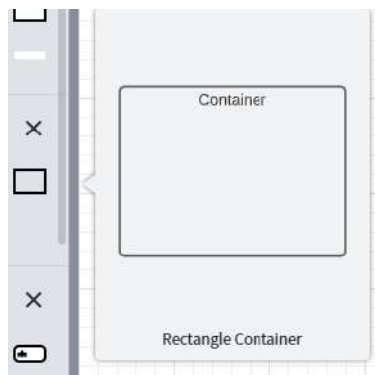
- **Insert content:**



Picture 48. Insert content

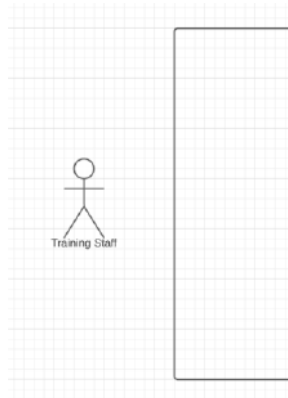


- **Using the content container:**



*Picture 49. LucidChart content container*

- **Using Actor:**



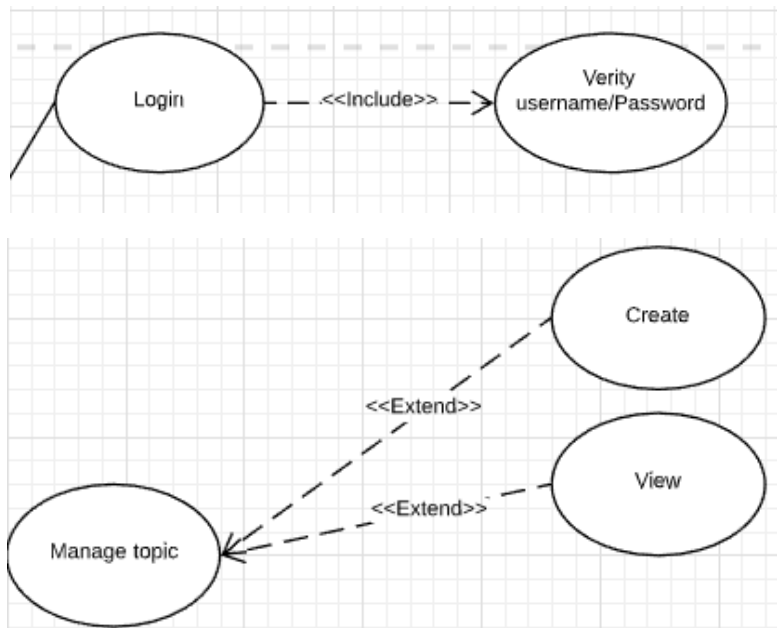
Picture 50. LucidChart Actor

- **Create Use Case:**



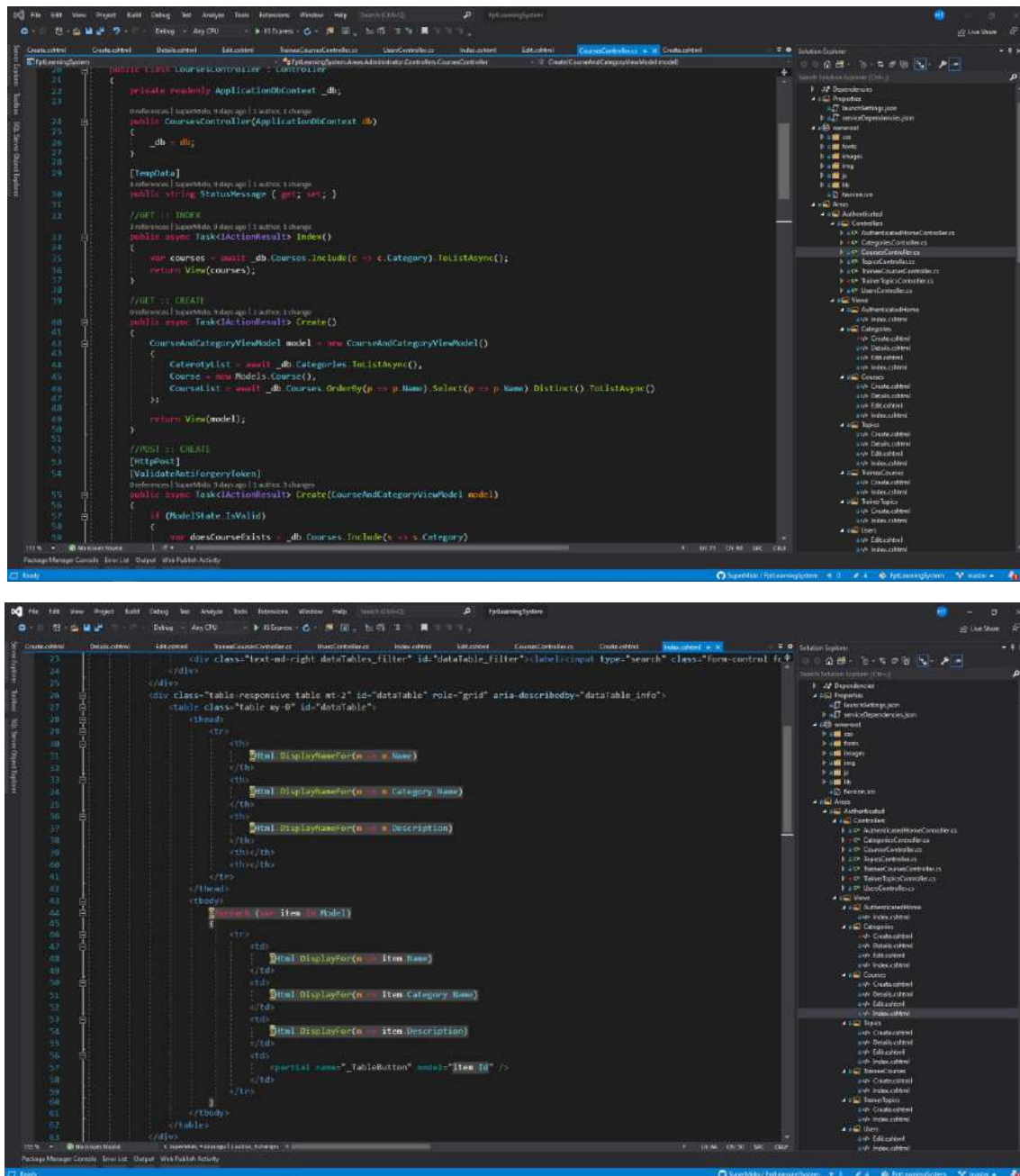
Picture 51. Lucidchart Use Case

- **Create a relationship:**



Picture 52. Lucidchart relationship

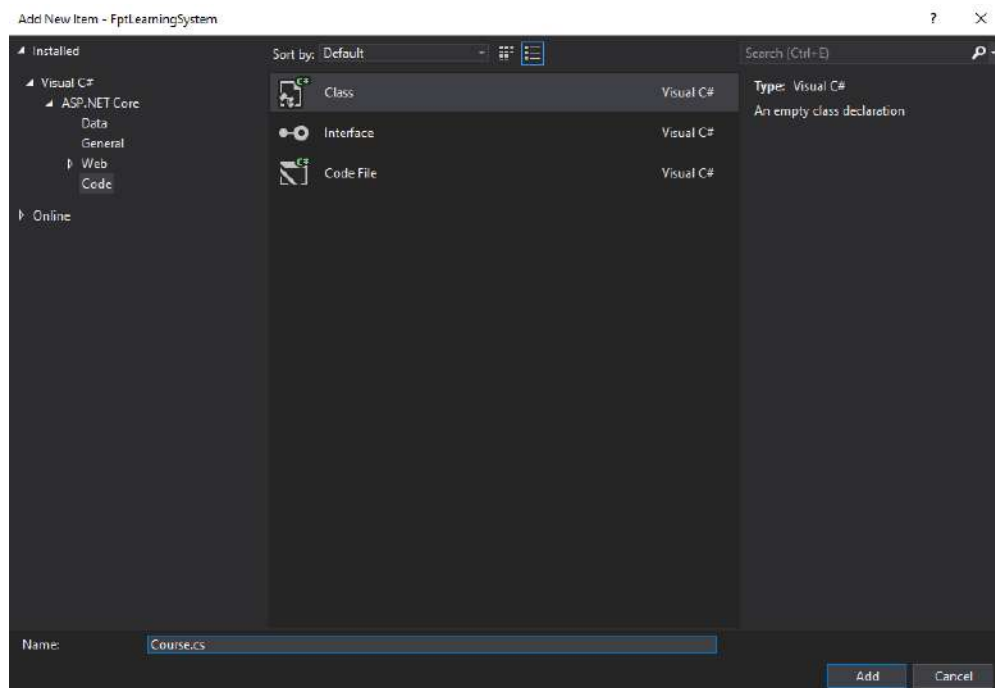
- Visual Studio 2019:



Picture 53. Training Staff view Course with Visual Studio

There is 4 main part to develop Training Staff View Course in Visual Studio:

- Create Model Course and add DbSet



```
public class Course
{
    [Key]
    6 references | SuperMido, 14 days ago | 1 author, 1 change
    public int Id { get; set; }
    [Required]
    [Display(Name = "Course Name")]
    19 references | SuperMido, 14 days ago | 1 author, 1 change
    public string Name { get; set; }
    2 references | SuperMido, 14 days ago | 1 author, 1 change
    public string Description { get; set; }

    [Required]
    [Display(Name = "Category")]
    3 references | SuperMido, 14 days ago | 1 author, 1 change
    public int CategoryId { get; set; }

    [ForeignKey("CategoryId")]
    8 references | SuperMido, 14 days ago | 1 author, 1 change
    public virtual Category Category { get; set; }
}
```

Picture 54. Visual Studio create Model

**Dbset:**

```
22 references | Supermido, 14 days ago | 1 author, 1 change  
public DbSet<Course> Courses { get; set; }
```

Picture 55. Visual studio DbSet

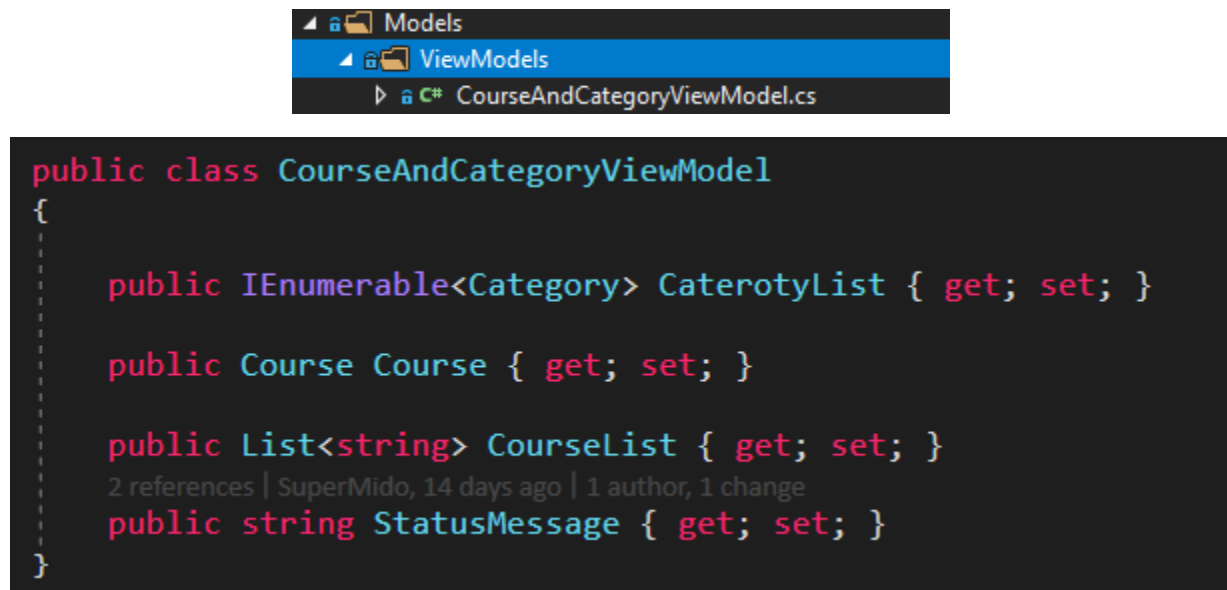
**Add-Migration and Update-Database:**

```
Package Manager Console  
Package source: All [v] [g] Default project: FptLearningSystem  
Each package is licensed to you by its owner. NuGet is not responsible for  
licenses. Follow the package source (feed) URL to determine an appropriate  
Package Manager Console Host Version 5.6.0.6591  
Type 'get-help NuGet' to see all available NuGet commands.  
PM> Add-Migration AddCourseIntoDatabase
```

```
Package Manager Console  
Package source: All [v] [g] Default project: FptLearningSystem  
Each package is licensed to you by its owner. NuGet is not responsible for  
licenses. Follow the package source (feed) URL to determine an appropriate  
Package Manager Console Host Version 5.6.0.6591  
Type 'get-help NuGet' to see all available NuGet commands.  
PM> Update-Database
```

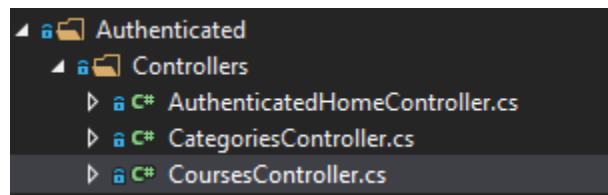
Picture 56. Visual Studio Add-Migration and Update-Database

- Create ViewModel for Course and Category



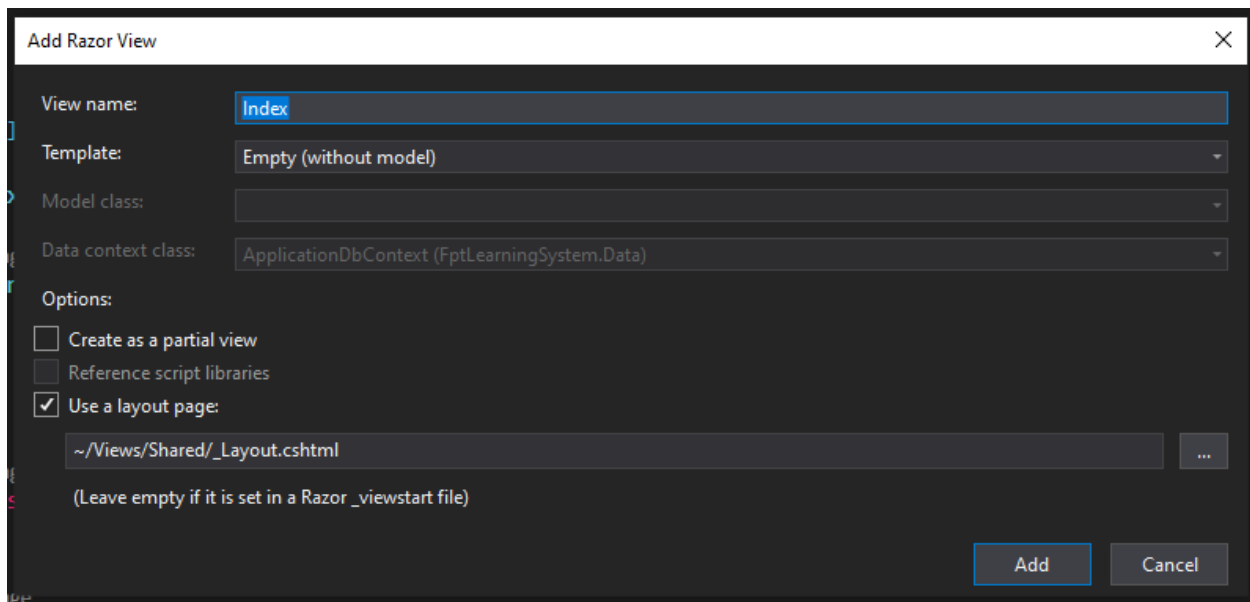
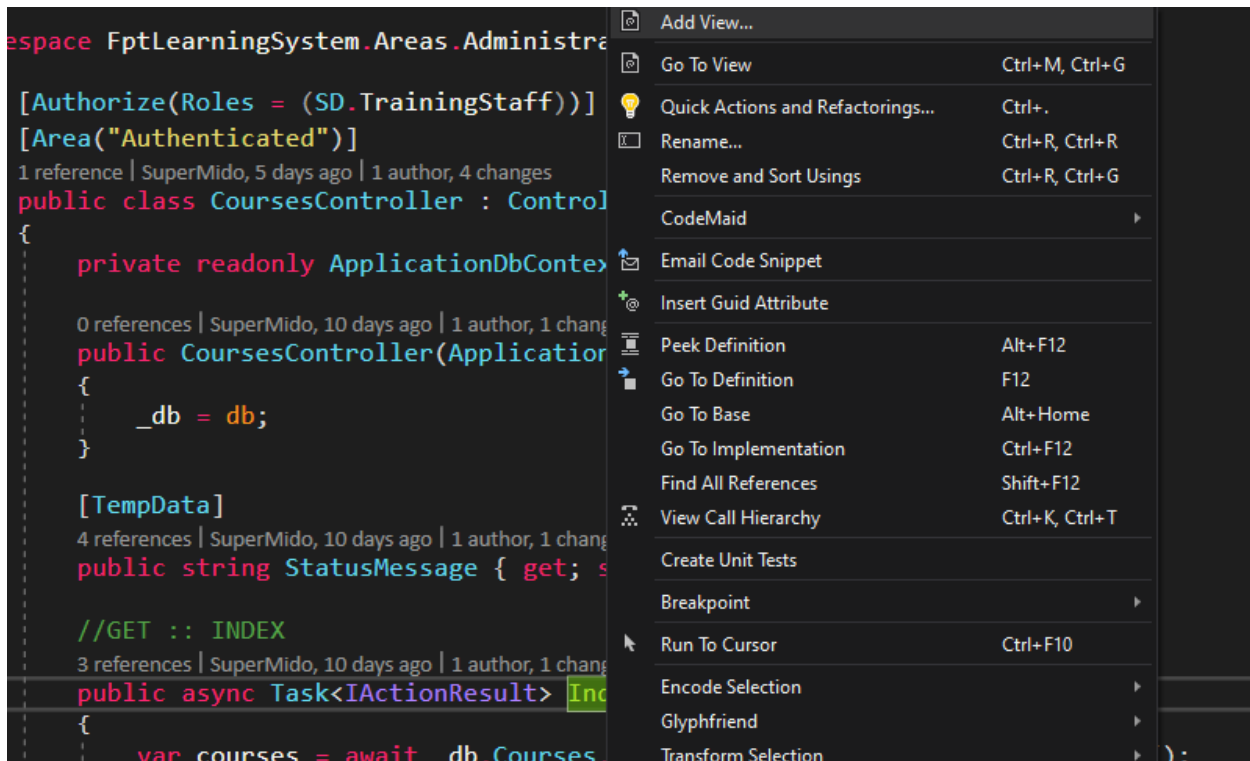
Picture 57. Visual Studio create ViewModel

- Create Controller in Area Authenticated



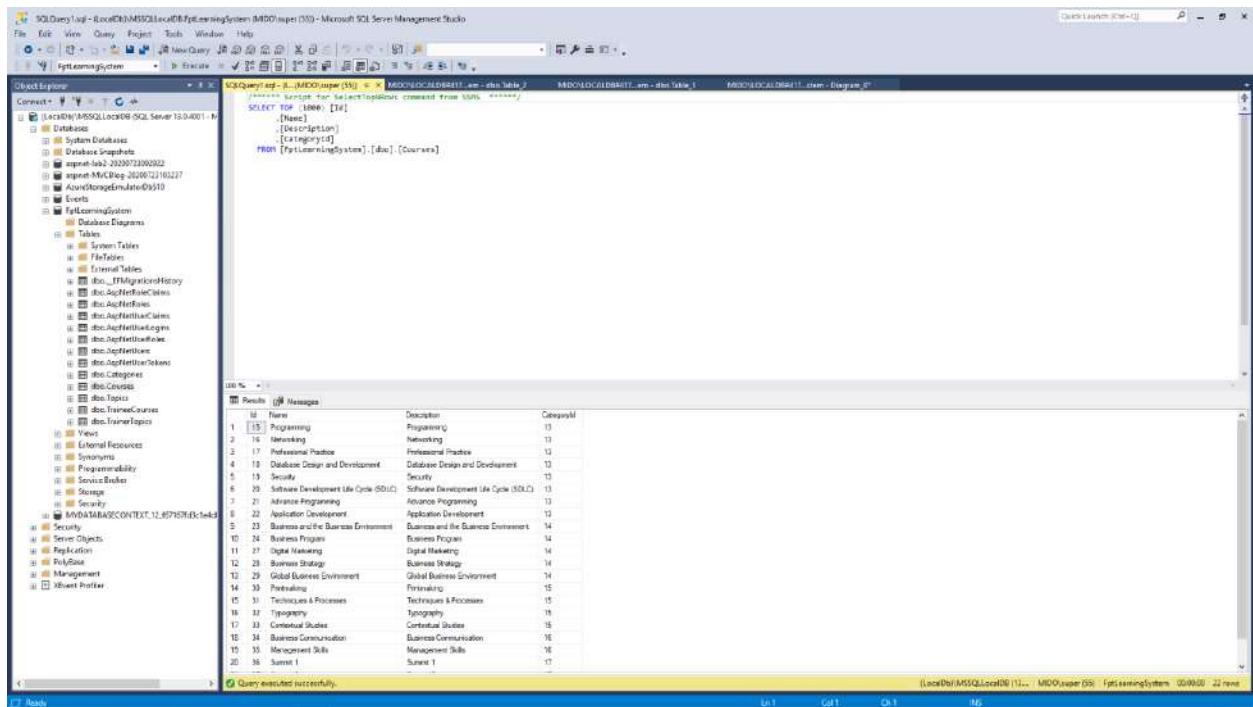
Picture 58. Visual Studio creates Controller

- With each controller in CoursesController will create a view



Picture 59. Visual Studio creates the view

- **SQL server:**

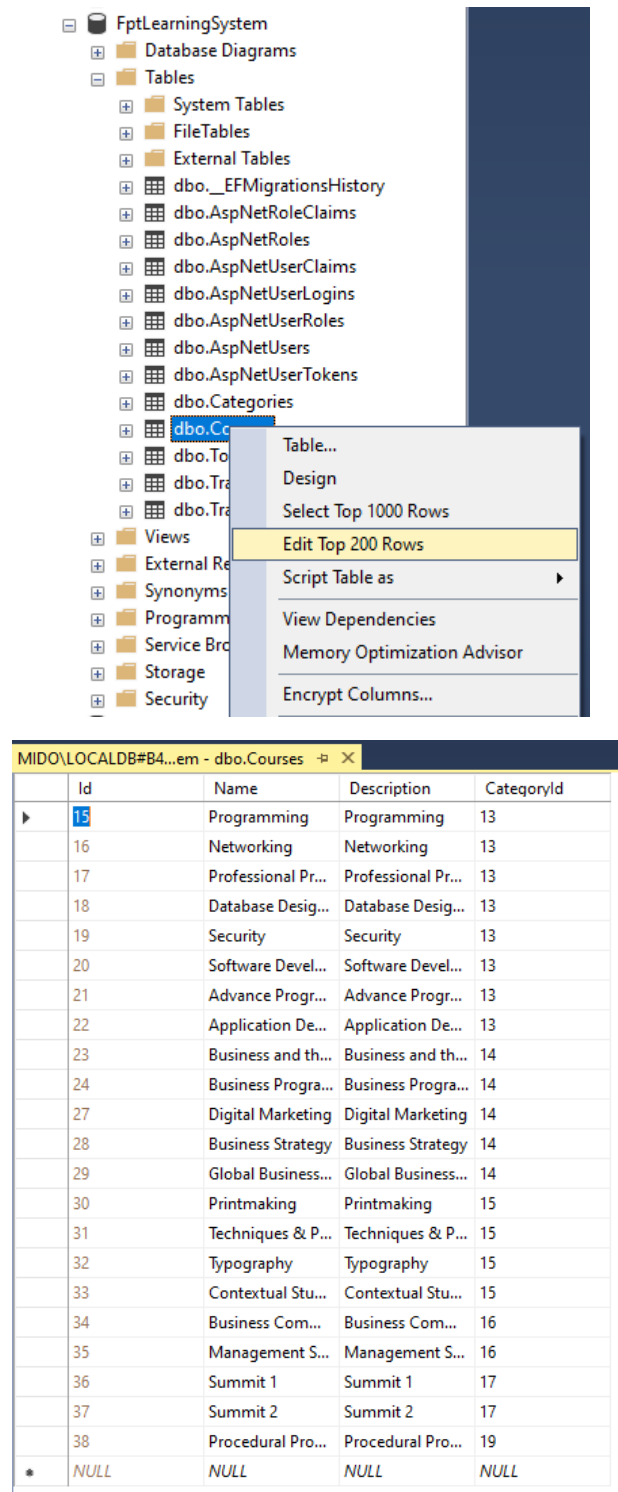


Picture 60. Training Staff view Course with SQL Server

In SQL Server, the Developer can view the list of tables in the system. Besides, the Developer can also check the data in each table and add data to the table.



- **Edit Course:**



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'FptLearningSystem' database is expanded, showing the 'Tables' folder. The 'dbo.Courses' table is selected, and a context menu is open. The 'Edit Top 200 Rows' option is highlighted. Below the menu, the 'dbo.Courses' table is displayed in a grid view. The table has the following columns: Id, Name, Description, and CategoryId. The data is as follows:

Id	Name	Description	CategoryId
15	Programming	Programming	13
16	Networking	Networking	13
17	Professional Pr...	Professional Pr...	13
18	Database Desig...	Database Desig...	13
19	Security	Security	13
20	Software Devel...	Software Devel...	13
21	Advance Progr...	Advance Progr...	13
22	Application De...	Application De...	13
23	Business and th...	Business and th...	14
24	Business Progra...	Business Progra...	14
27	Digital Marketing	Digital Marketing	14
28	Business Strategy	Business Strategy	14
29	Global Business...	Global Business...	14
30	Printmaking	Printmaking	15
31	Techniques & P...	Techniques & P...	15
32	Typography	Typography	15
33	Contextual Stu...	Contextual Stu...	15
34	Business Com...	Business Com...	16
35	Management S...	Management S...	16
36	Summit 1	Summit 1	17
37	Summit 2	Summit 2	17
38	Procedural Pro...	Procedural Pro...	19
NULL	NULL	NULL	NULL

Picture 61. SSMS Edit

- Design:

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Name	nvarchar(MAX)	<input type="checkbox"/>
Description	nvarchar(MAX)	<input checked="" type="checkbox"/>
CategoryId	int	<input type="checkbox"/>

Picture 62. SSMS Design

- Select:

```

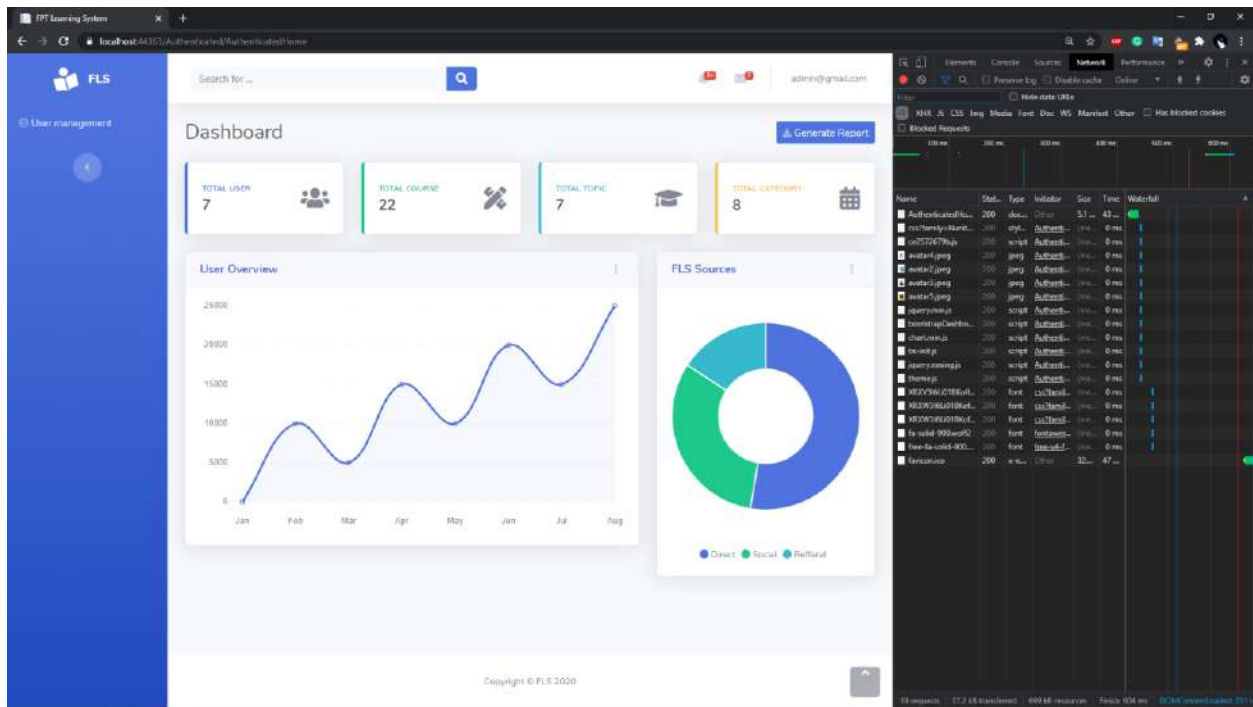
/***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [Id]
      ,[Name]
      ,[Description]
      ,[CategoryId]
FROM [FptLearningSystem].[dbo].[Courses]
  
```

	Id	Name	Description	CategoryId
1	15	Programming	Programming	13
2	16	Networking	Networking	13
3	17	Professional Practice	Professional Practice	13
4	18	Database Design and Development	Database Design and Development	13
5	19	Security	Security	13
6	20	Software Development Life Cycle (SDLC)	Software Development Life Cycle (SDLC)	13
7	21	Advance Programming	Advance Programming	13
8	22	Application Development	Application Development	13
9	23	Business and the Business Environment	Business and the Business Environment	14
10	24	Business Program	Business Program	14
11	27	Digital Marketing	Digital Marketing	14
12	28	Business Strategy	Business Strategy	14
13	29	Global Business Environment	Global Business Environment	14
14	30	Printmaking	Printmaking	15
15	31	Techniques & Processes	Techniques & Processes	15
16	32	Typography	Typography	15
17	33	Contextual Studies	Contextual Studies	15
18	34	Business Communication	Business Communication	16
19	35	Management Skills	Management Skills	16
20	36	Summit 1	Summit 1	17

Query executed successfully.

Picture 63. SSMS Select

- **Google Chrome:**



Picture 64. Training Staff view Course with Google Chrome

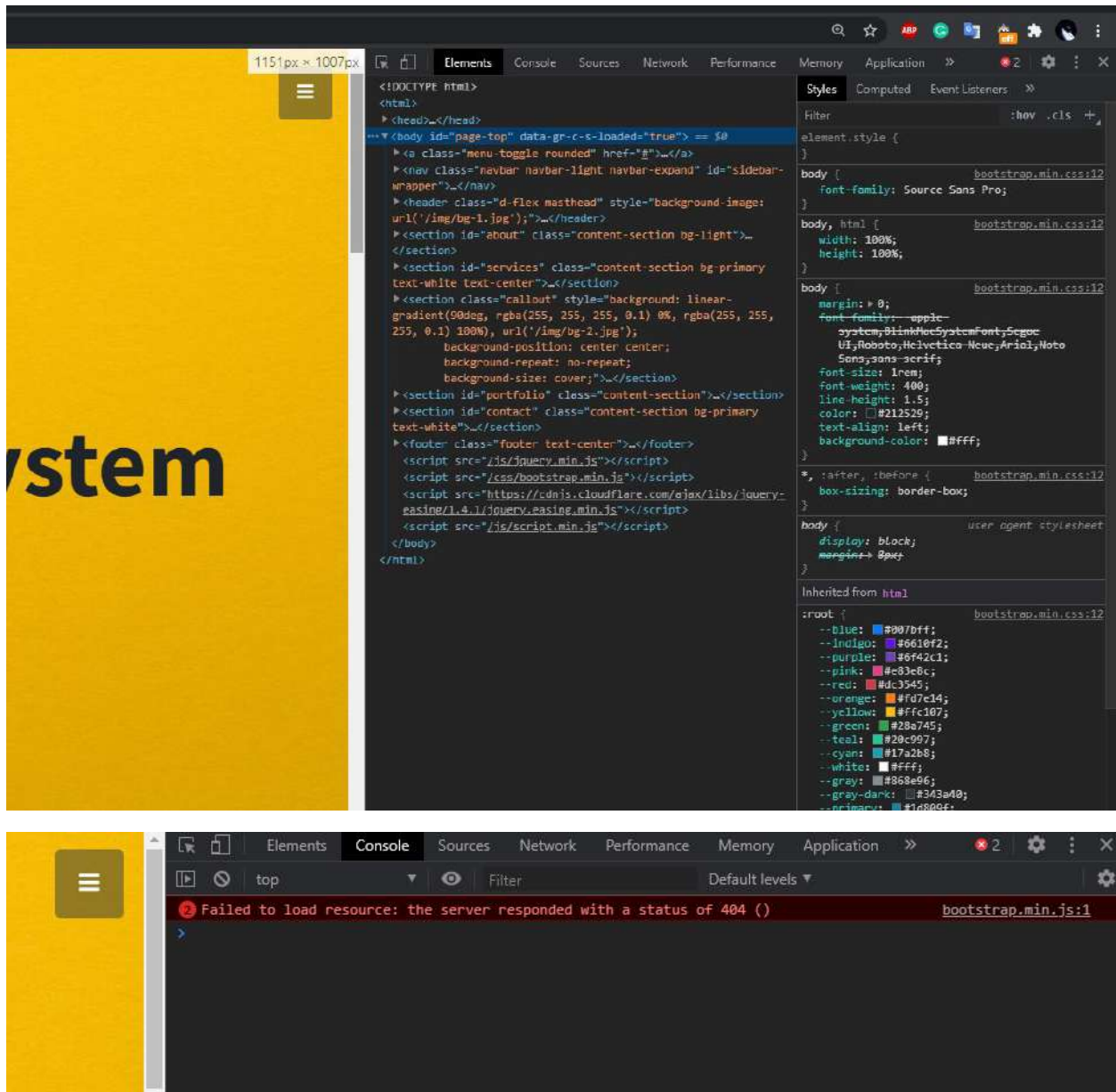
In Google Chrome there are 2 main things that developer used to develop this FLS system:

- **Google Chrome:** With IIS Server will create a server run on localhost:44363, so the developer will be using google chrome to access the website



Picture 65. Google Chrome with localhost:44363

- **Google Chrome developer tool:** This tool will help developers debug the UI in Elements tag, console for debugging JavaScript code, etc.



Picture 66. Google Chrome with google chrome developer tool

### 2.3. Ideas/Potential to improve the system

- **View development:**

In view that developers have created in the system, there are Header and Footer and Body in each view.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
6 <title>Table - Brand</title>
7 <link rel="stylesheet" href="assets/bootstrap/css/bootstrap.min.css">
8 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Munito:200,200i,300,300i,400,400i,600,600i,700,700i,800,800i,900">
9 <link rel="stylesheet" href="assets/fonts/fontawesome-all.min.css">
10 </head>
11 <body id="page-top">
12 <div id="wrapper">
13 <nav class="navbar navbar-dark align-items-start sidebar sidebar-dark accordion bg-gradient-primary p-0">
14 <div class="container-fluid d-flex flex-column p-0">
15 <a class="navbar-brand d-flex justify-content-center align-items-center sidebar-brand m-0" href="#">
16 <div class="sidebar-brand-icon rotate-n-15"><i class="fas fa-laugh-wink"></i></div>
17 <div class="sidebar-brand-text mx-3"><span>Brand</span></div>
18 </a>
19 <div class="sidebar-divider my-0"></div>
20 <ul class="nav navbar-nav text-light id="accordionSidebar">
21 <li class="nav-item role="presentation"><a class="nav-link" href="index.html"><i class="fas fa-tachometer-alt"></i></li>
22 <li class="nav-item role="presentation"><a class="nav-link" href="profile.html"><i class="fas fa-user"></i><span>Pro</span></li>
23 <li class="nav-item role="presentation"><a class="nav-link active" href="table.html"><i class="fas fa-table"></i><span>Pro</span></li>
24 <li class="nav-item role="presentation"><a class="nav-link" href="login.html"><i class="fas fa-user-circle"></i><span>Pro</span></li>
25 <li class="nav-item role="presentation"><a class="nav-link" href="register.html"><i class="fas fa-user-circle"></i><span>Pro</span></li>
26 </ul>
27 <div class="text-center d-none d-md-inline"><button class="btn rounded-circle border-0" id="sidebarToggle" type="button">
28 </div>
29 </div>
30 </nav>
31 <div class="d-flex flex-column" id="content-wrapper">
32 <div id="content">
33 <nav class="navbar navbar-light navbar-expand bg-white shadow mb-4 topbar static-top">
34 <div class="container-fluid">
35 <button class="btn btn-link d-md-none rounded-circle mr-3" id="sidebarToggleTop" type="button"><i class="fas fa-l<
36 <form class="form-inline d-none d-sm-inline-block mr-auto ml-md-3 my-2 my-md-0 mw-100 navbar-search">
37 <div class="input-group">
38 <input class="bg-light form-control border-0 small" type="text" placeholder="Search for ...">
39 <div class="input-group-append"><button class="btn btn-primary py-0" type="button"><i class="fas fa-search"></i></div>
40 </div>
41 </div>
```

With layout views, maintaining the consistent look and feel across all the views becomes much easier, as developers have only one layout file to modify, should there be any change. The change will then be immediately reflected across all the views in the entire application.

- **Logo:**

The FPT Learning System did not have a logo. In a current version, the logo is using the icon from Font Awesome.

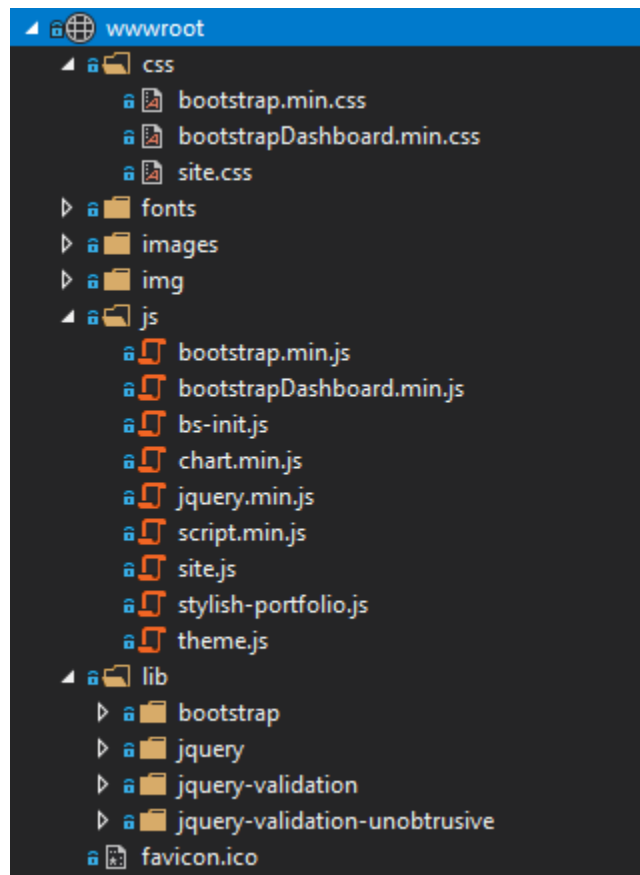


If the FPT Learning System has a professional logo design that will:

- Makes a great first impression
- Show the concept and strategy
- Gives business an identity

- **Minify CSS, HTML, JavaScript:**

Removing unnecessary or redundant data without affecting how the page is processed by the browser

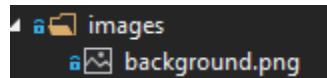


Outside of surveying FLS's source code manually, **Google recommends** trying the following resources:

- The **HTML minified** to minify HTML.
- **CSSNano** and **CSS** to minify CSS.
- **UglifyJS2** and the **Closure Compiler** to minify JavaScript.

- **Optimize images:**

Graphic elements such as favicons, logos, and product images can easily comprise up to two-thirds of a page's total weight that make the effect on page load time, and specifically, the conversion rate is huge.



Fortunately, optimizing images is easy. The replacement of a **PNG** with a **JPEG** image file can easily save your page size and load time.

- **Speed up server response time:**

**Server response time:** the time it takes for your server to begin loading a page's content for a user.

In the FLS system controller, there are too many database queries. This problem will make the system slow down. When a lot of users using this application at the same time, the application may be very slow and sometimes crash.

```
public async Task<ActionResult> Create(TraineeCourseViewModel model)
{
    var traineeRoleId = await _db.Roles.Where(t => t.Name == SD.Trainee).Select(t => t.Id).FirstAsync();

    var listTraineeId = await _db.UserRoles
        .Where(t => t.RoleId == traineeRoleId)
        .Select(t => t.UserId)
        .ToArrayAsync();

    List<ApplicationUser> traineeUsers = await _db.ApplicationUsers
        .Where(t => listTraineeId
            .Any(name => name.Equals(t.Id)))
        .ToListAsync();

    if (ModelState.IsValid)
    {
        var doesTraineeCourseExists = _db.TraineeCourses.Include(t => t.Course).Include(t => t.User)
            .Where(t => t.Course.Id == model.TraineeCourse.CourseId && t.User.Id == model.TraineeCourse.TraineeId);

        if (doesTraineeCourseExists.Count() > 0)
        {
            //Error
            StatusMessage = "Error: The Trainee has bene assigned under " + doesTraineeCourseExists.First().Course.Name;
        }
        else
        {
            _db.TraineeCourses.Add(model.TraineeCourse);
            await _db.SaveChangesAsync();
            return RedirectToAction(nameof(Index));
        }
    }

    TraineeCourseViewModel modelVM = new TraineeCourseViewModel()
    {
        Courses = await _db.Courses.ToListAsync(),
        TraineeCourse = model.TraineeCourse,
        StatusMessage = StatusMessage,
        Users = traineeUsers
    };
    return View(modelVM);
}
```



- **Create API with ASP.NET Core in the future:**

Web APIs are a great framework for building HTTP services and exposing the same over an arsenal of clients including browsers, mobiles, and tablets.

Benefits ASP.NET Web APIs:

- Empower the developers by handing over control over the way HTTP protocol messages are sent and responded to.
- ASP.NET Web APIs provide a great level of abstraction with which developers can create the Web APIs that may encapsulate **HttpMessageHandler**.
- The architecture of web APIs is very light, which makes them a perfect alternative for the developers when they want to build applications for devices with limited bandwidth.
- Reusable across other platforms and frameworks like **ReactJs**.

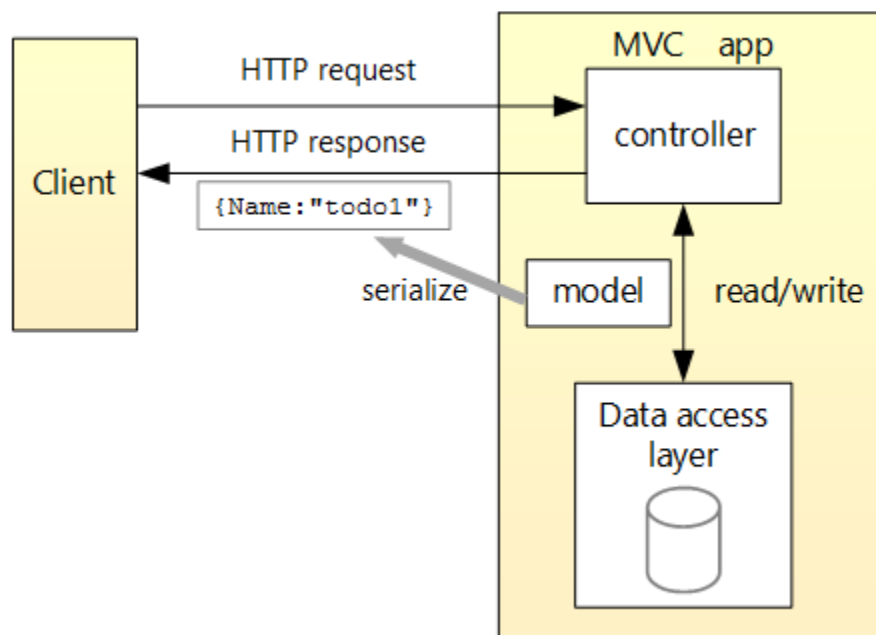


Figure 4. The design of API in ASP.NET Core



- **Security in ASP.NET Core**

In ASP.NET Core, authentication is handled by the **IAuthenticationService**, which is used by authentication middleware. The authentication service uses registered authentication handlers to complete authentication-related actions.

Authentication-related actions include:

- Authenticating a user.
- Responding when an unauthenticated user tries to access a restricted resource.

In the Controller code, there are some critical security issues, the code is including the RoleId from the database. This problem will make the system's not secure and the data will be a leak.

```
var listTraineeId = await _db.UserRoles
    .Where(t => t.RoleId == "cdd4e45c-5394-4924-a900-6537dcc09996")
    .Select(t => t.UserId)
    .ToArrayAsync();
```

The system needs to implement another query to get RoleId then pass the variable replaces the Id string.

In Training Staff role that Training staff just only create accounts for trainees. But in Google Chrome Tool, it's will easy for training staff to create Administrator, Training Staff, or trainer account by change value in the input tag.

```
<div class="col-5">
  <input type="radio" name="rdUserRole" value="Trainee" checked> ==
  " Trainee
  "

  <input type="radio" name="rdUserRole" value="TrainingStaff"
  checked>
  " Trainee
```

The system needs to implement the condition to create users in this case.

- **Using Cloud in Viet Nam**

The system was using Azure Cloud with location Southeast Asia for application and database, which will make the connection between the user and website slow in Vietnam.

Resource group ([change](#)) : [FPTLearningSystem](#)

Status : Running

Location : Southeast Asia

Subscription ([change](#)) : [Azure for Students](#)

Subscription ID : 7026c62c-4603-4cf2-bccf-a8a2b232f417

Tags ([change](#)) : [Click here to add tags](#)

Resource group ([change](#)) : [FPTLearningSystem](#)

Status : Online

Location : Southeast Asia

Subscription ([change](#)) : [Azure for Students](#)

Subscription ID : 7026c62c-4603-4cf2-bccf-a8a2b232f417

Tags ([change](#)) : [Click here to add tags](#)

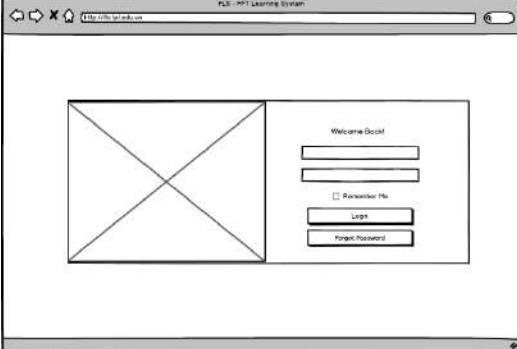
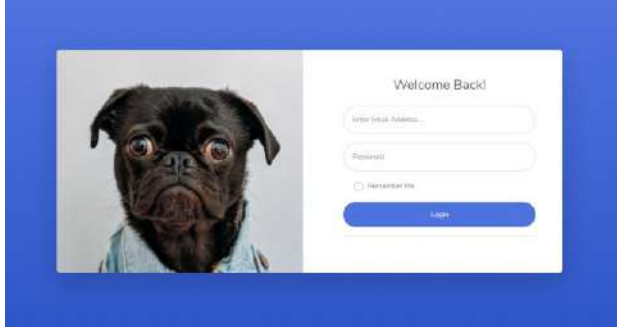
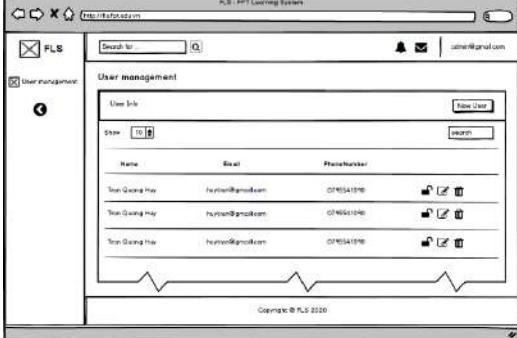
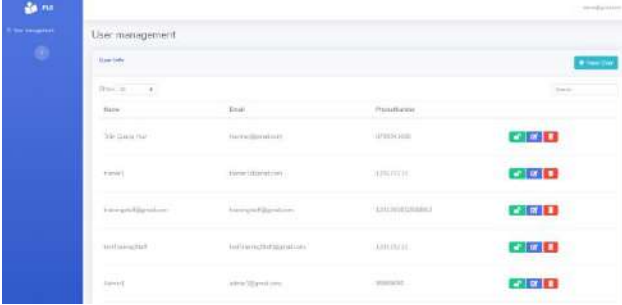
In the future, the service needs to develop more, especially hiring a cloud service in Vietnam. This will significantly speed up your website and avoid many potential problems.

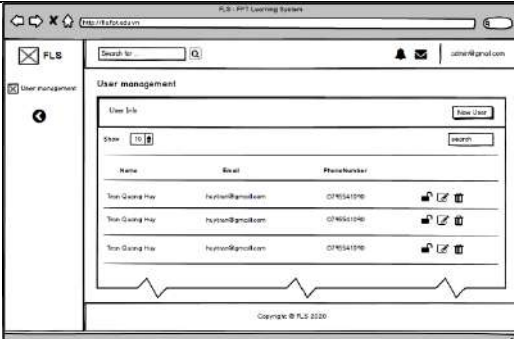
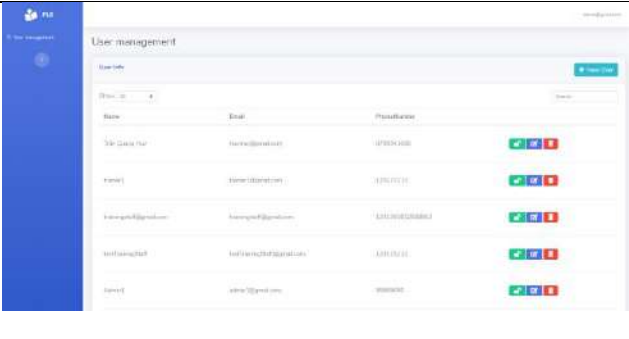
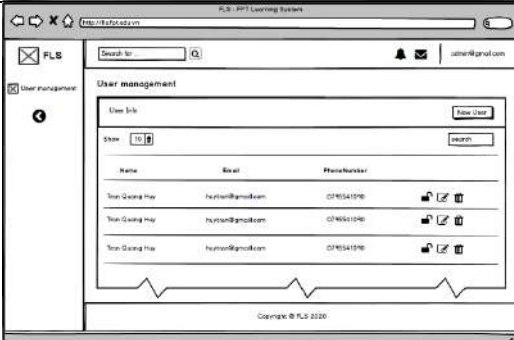
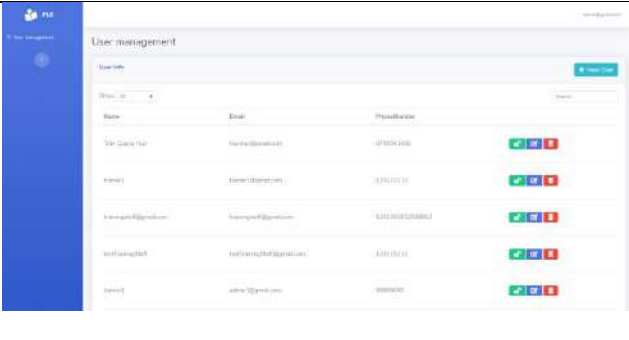
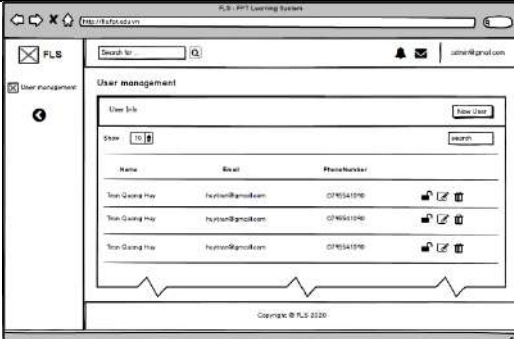
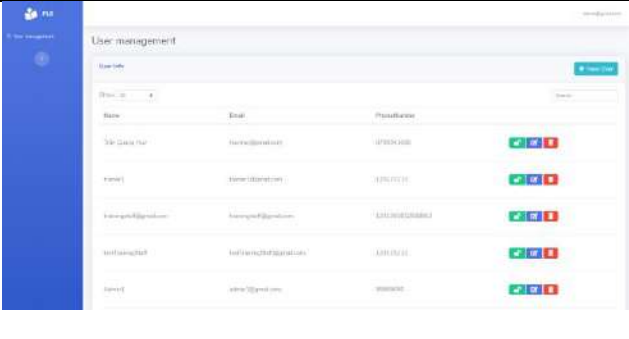
- **Notification function:**


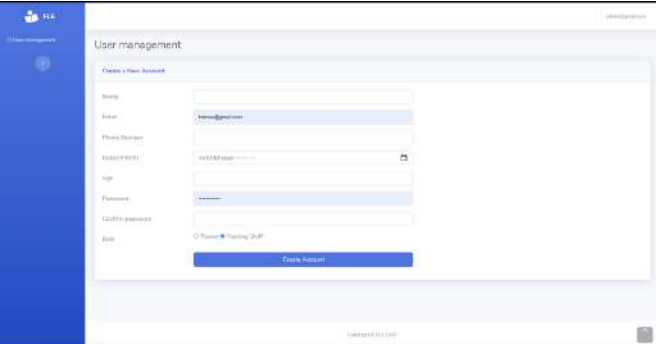
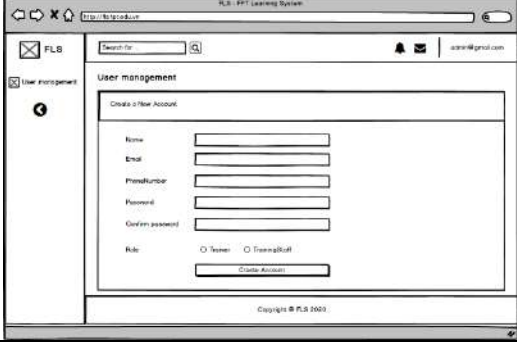
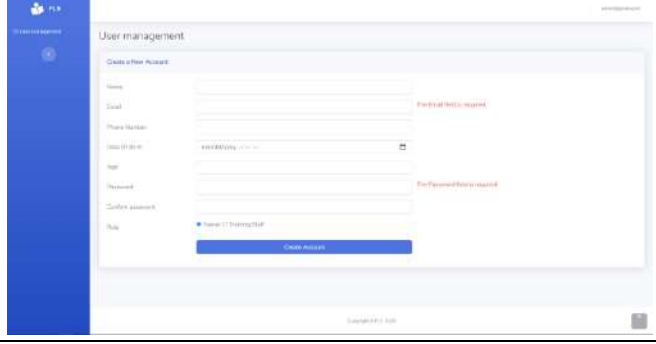
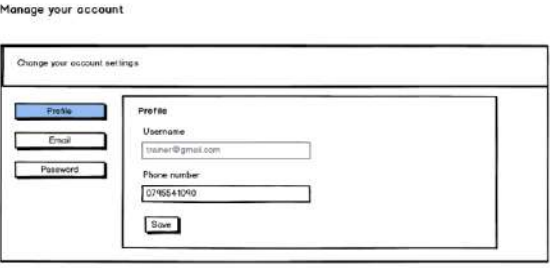
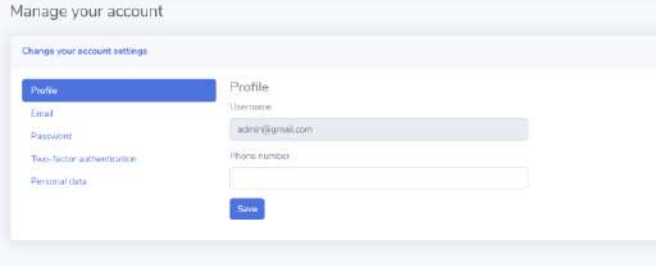
To increase interaction for users, especially trainees, the system needs to have the function to notify that user has been added to any course or message from the Trainer or Training Staff.

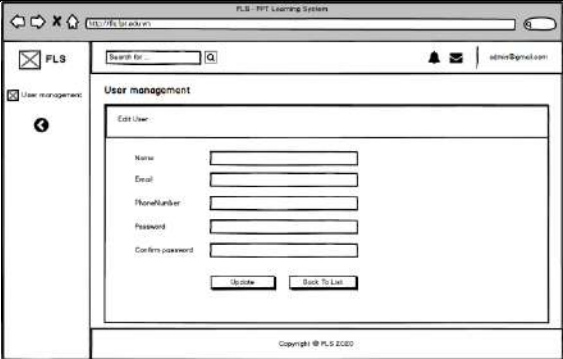
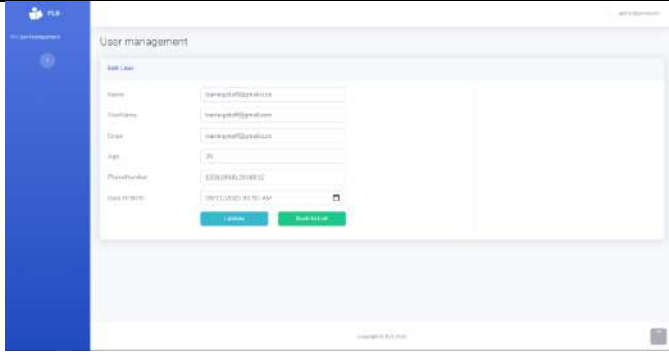

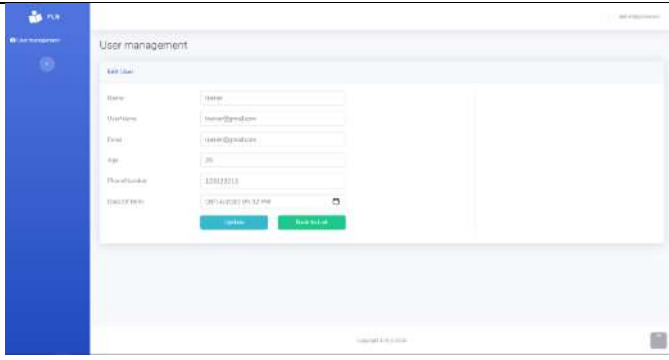

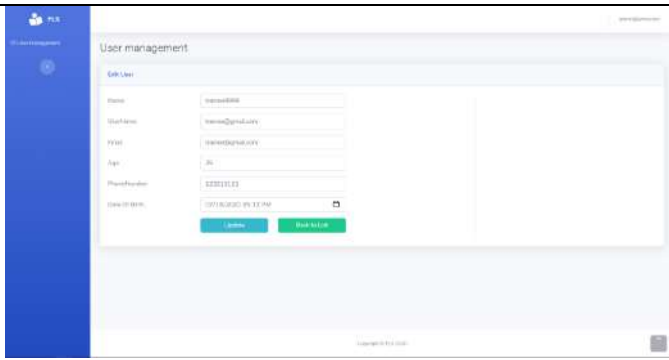
### 3. Application review

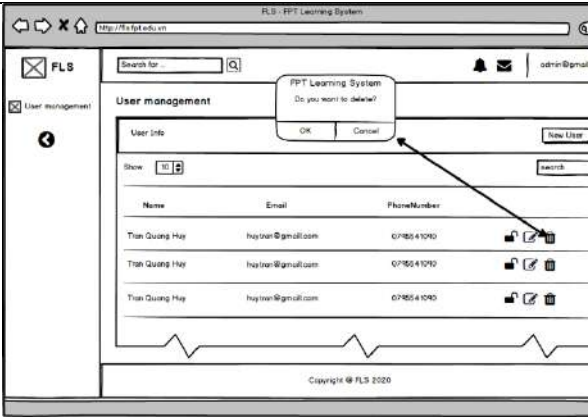
#### 3.1. Comparison to initial requirements

No.	Use Case	Describe	Mockup	Result	Completed (%)	Status
Administrator						
1	UC_A D01	Administrator login			100	Finished
2	UC_A D02	Administrator view Administrator profile			100	Finished

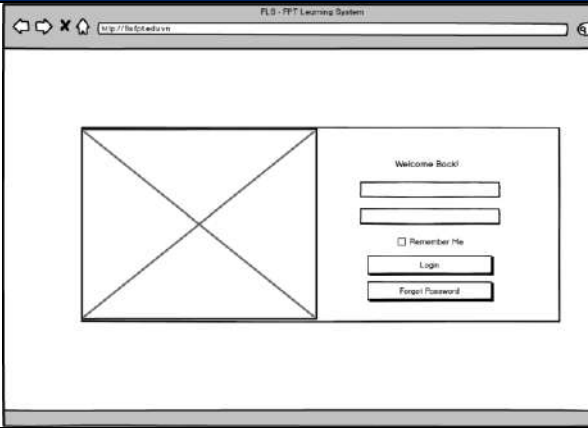
3	UC_A D03	Administrator view Training staff profile			100	Finished
4	UC_A D04	Administrator view Trainer profile			100	Finished
5	UC_A D05	Administrator view Trainee profile			100	Finished

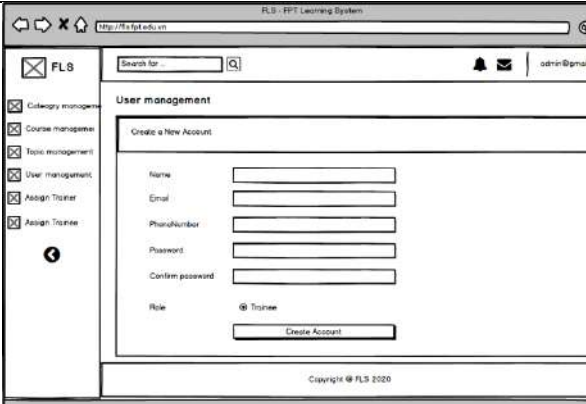
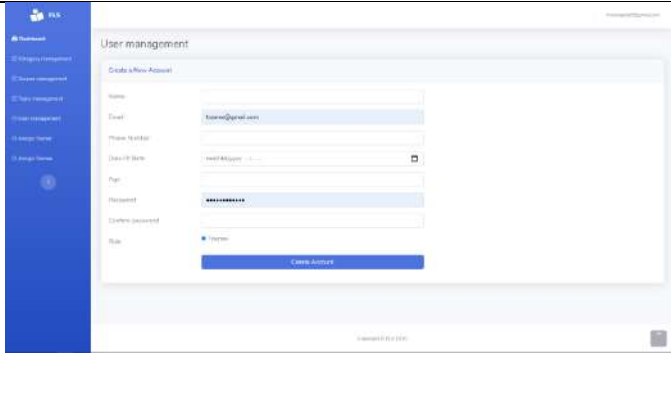
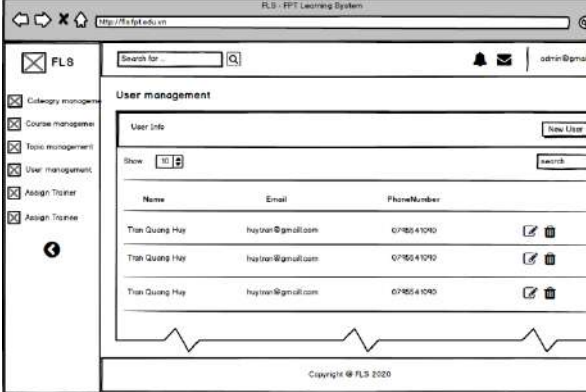
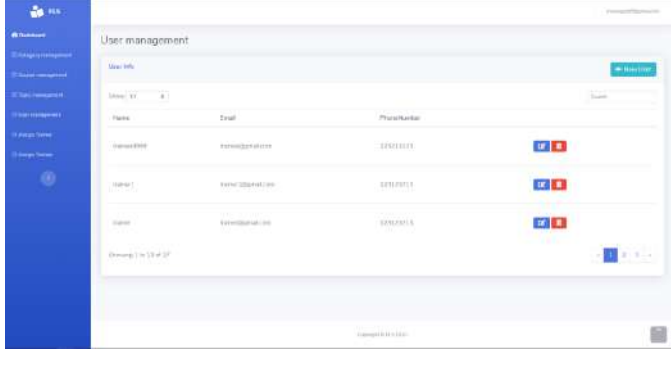
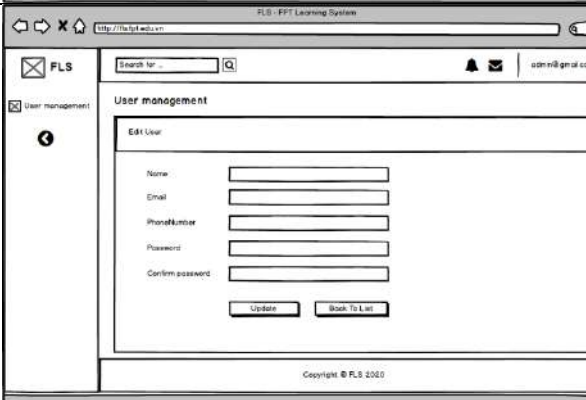
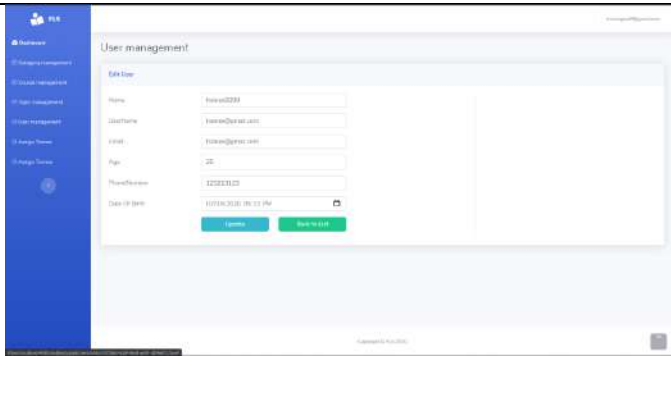
6	UC_A D06	Administrator creates Training staff account			100	Finished
7	UC_A D07	Administrator creates Trainer account			100	Finished
8	UC_A D08	Administrator creates Trainee account	-	-	0%	Unnecessary
9	UC_A D09	Administrator edit Administrator account			100	Finished

10	UC_A D10	Administrator edit Training staff account			100	Finished
11	UC_A D11	Administrator edit Trainer account			100	Finished
12	UC_A D12	Administrator edit Trainee account			100	Finished

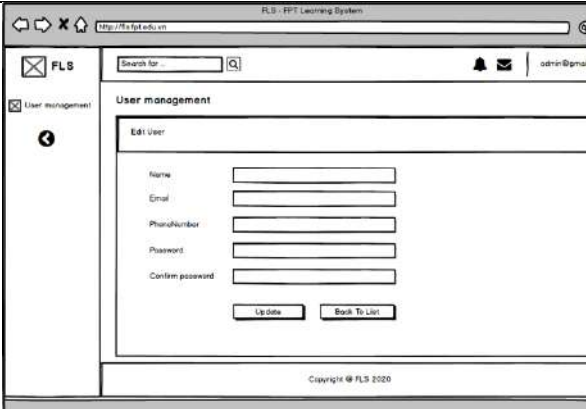
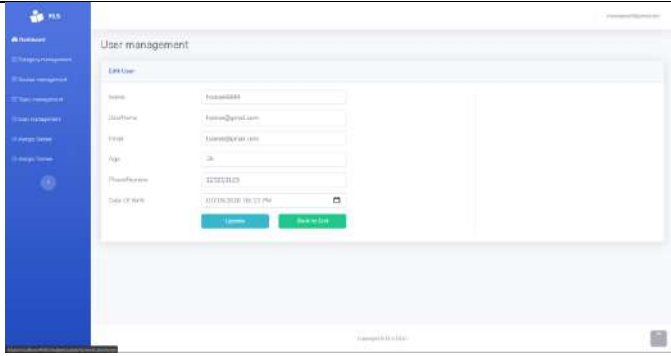
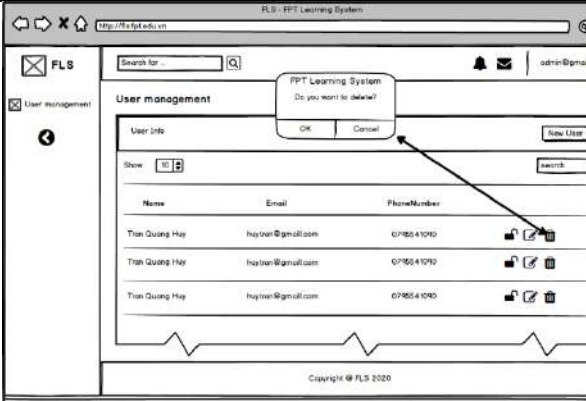
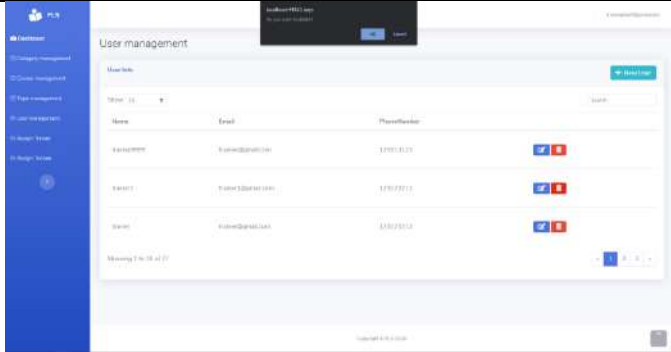
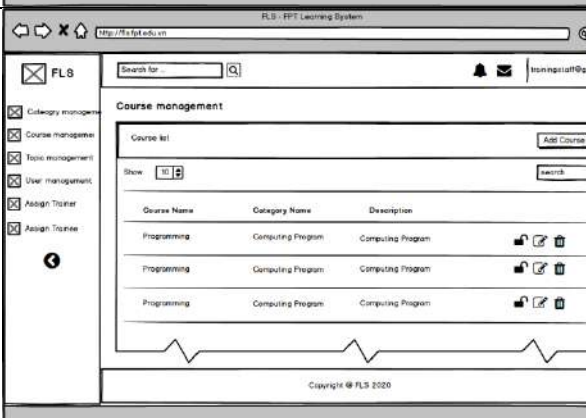
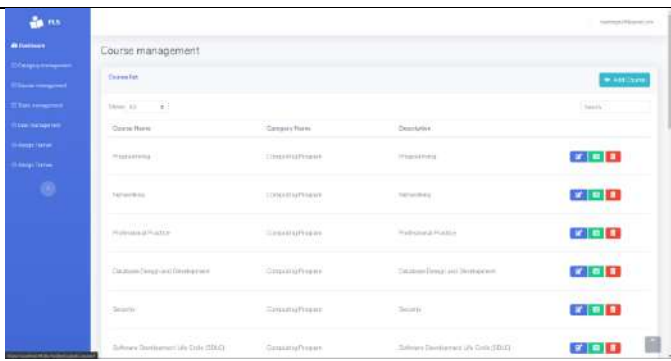
13	UC_A D13	Administrator delete exist account		100	Finished
----	-------------	--	--	-----	----------

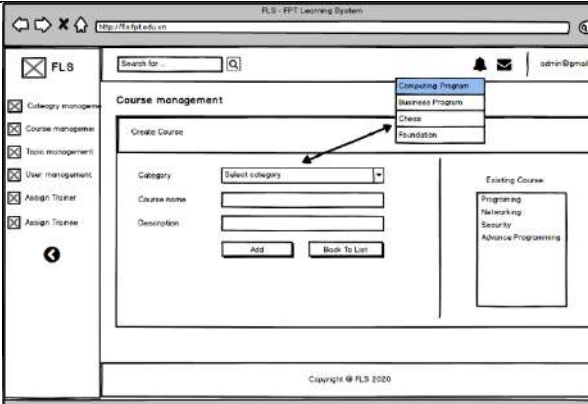
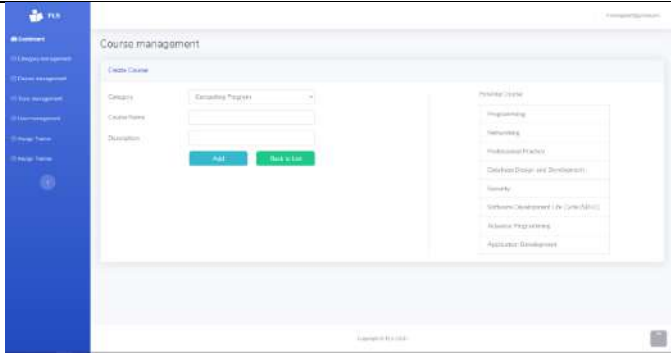
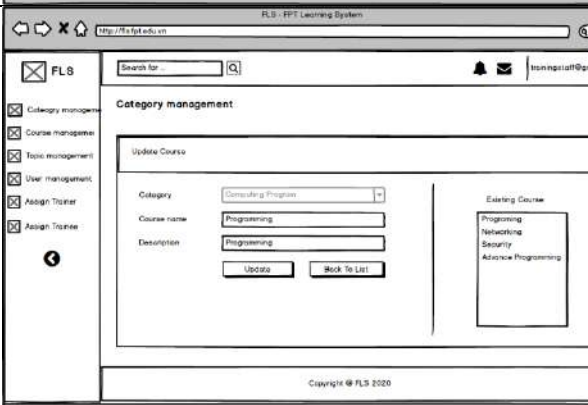
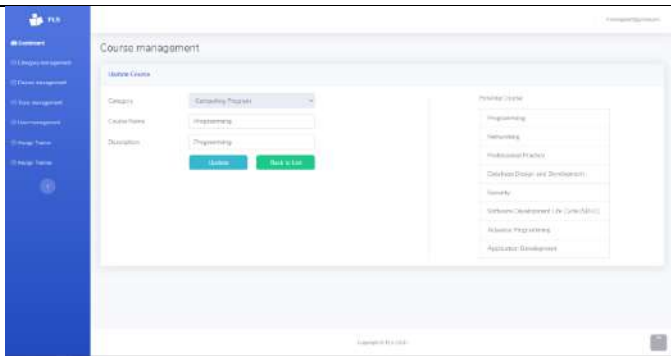
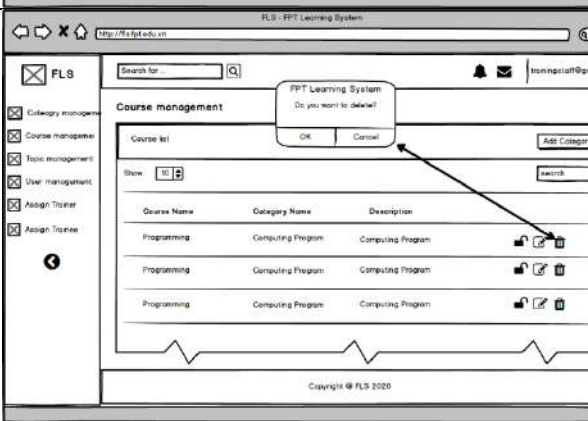
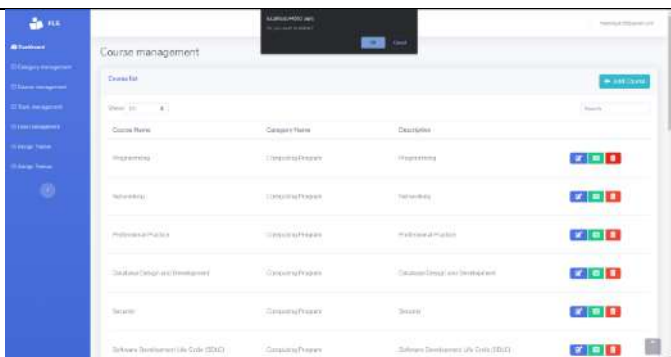
Training Staff

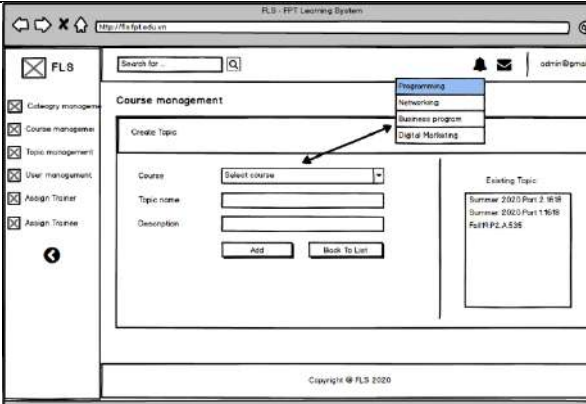
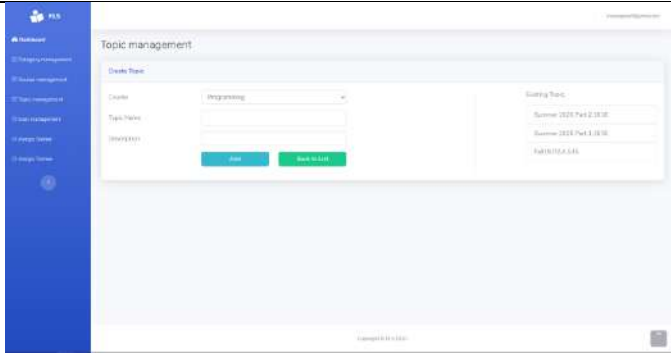
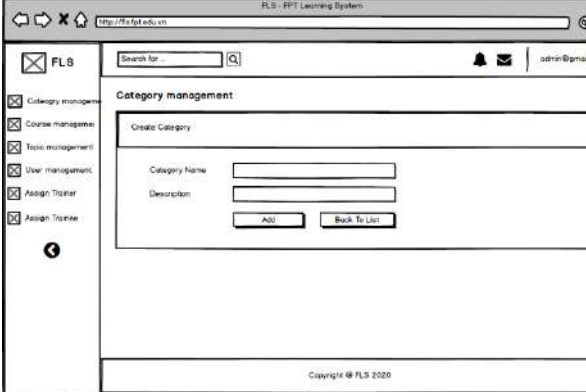
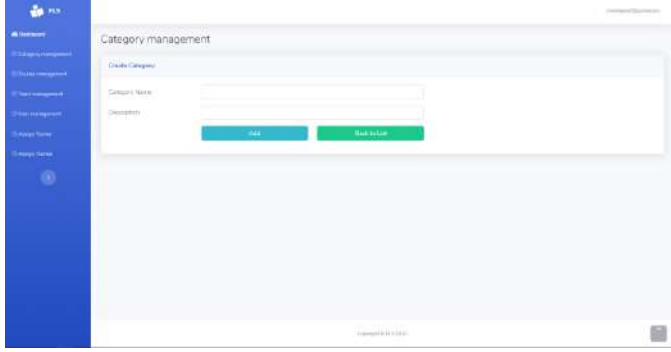
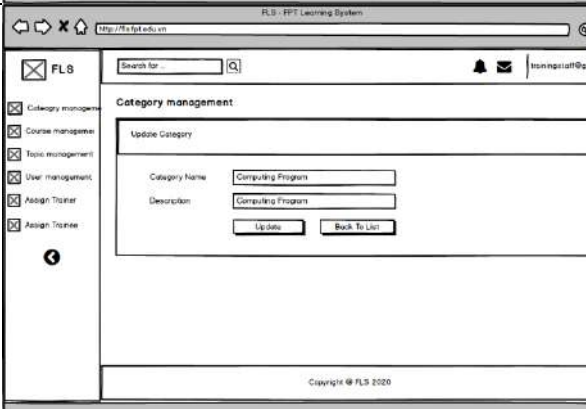
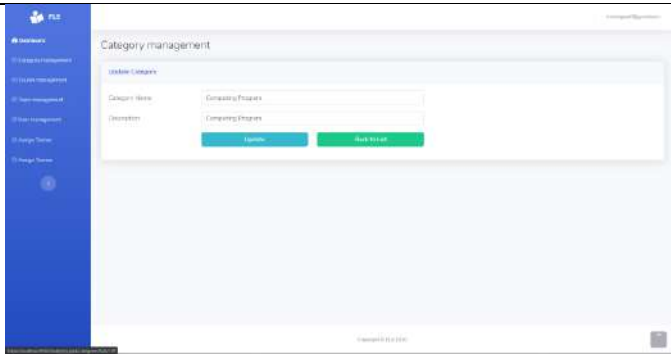
1	UC_T S01	Training staff login		100	Finished
---	-------------	-------------------------	---	-----	----------

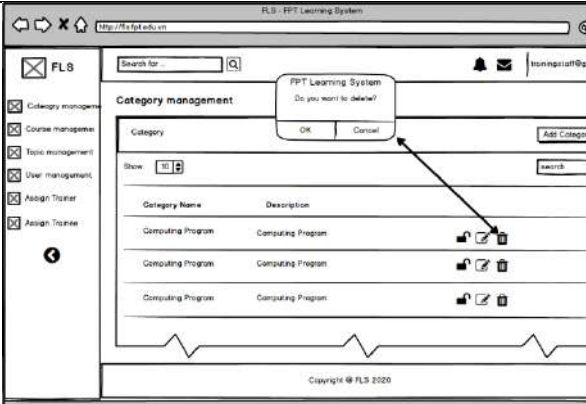
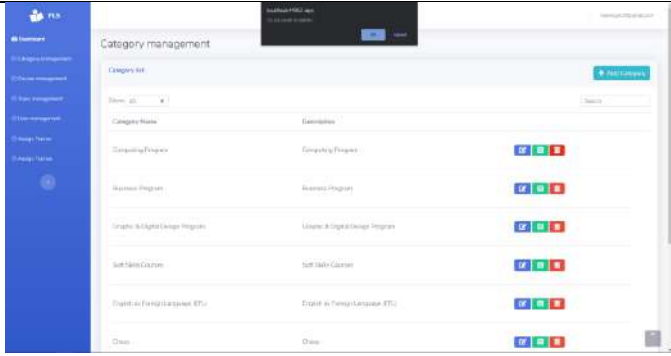
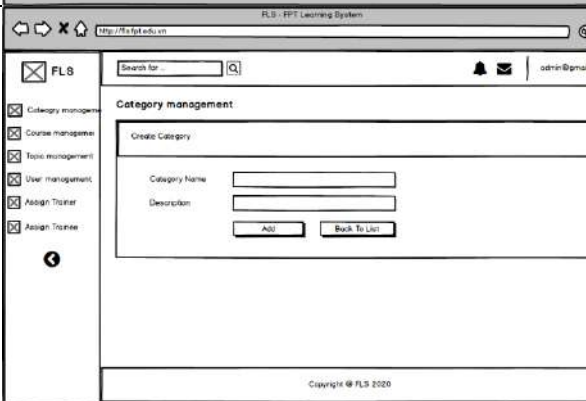
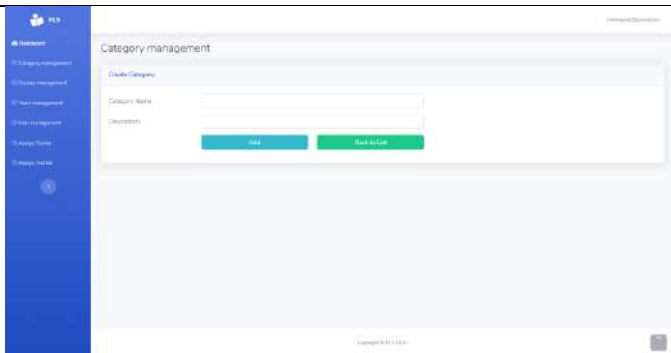
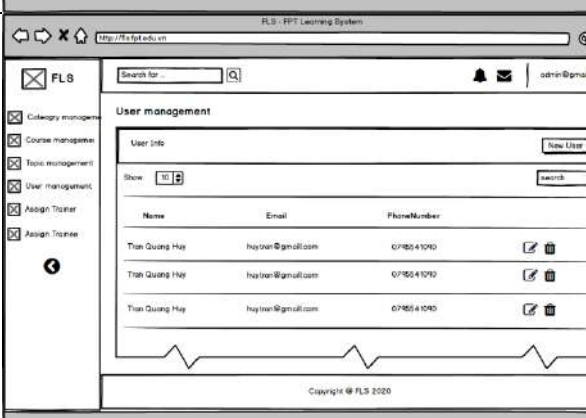
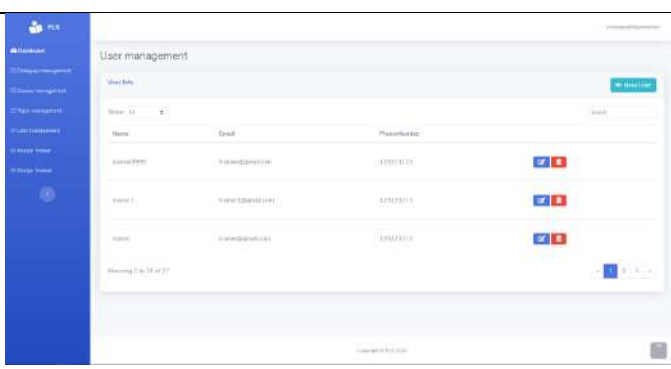
2	UC_T S02	Training staff creates Trainee account			100	Finished
3	UC_TS 03	Training staff view list of trainees			100	Finished
4	UC_TS 04	Training staff view trainee profile			100	Finished

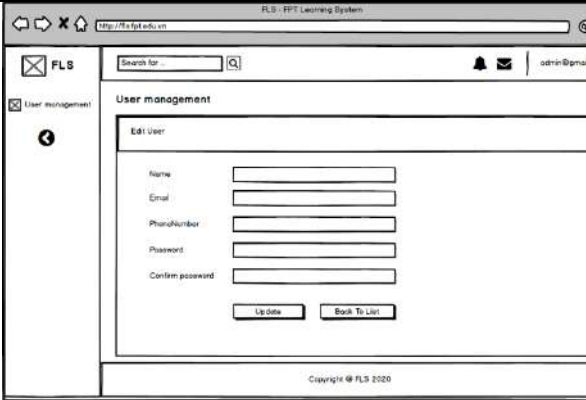
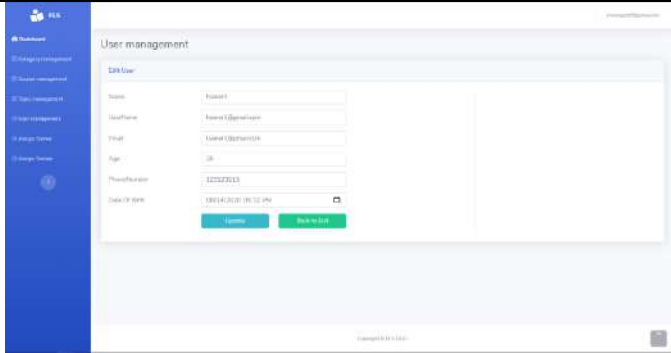
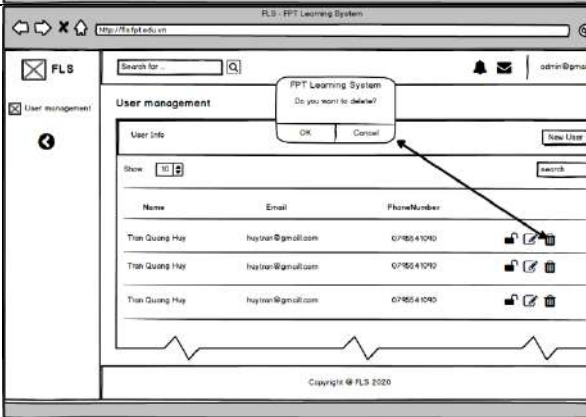
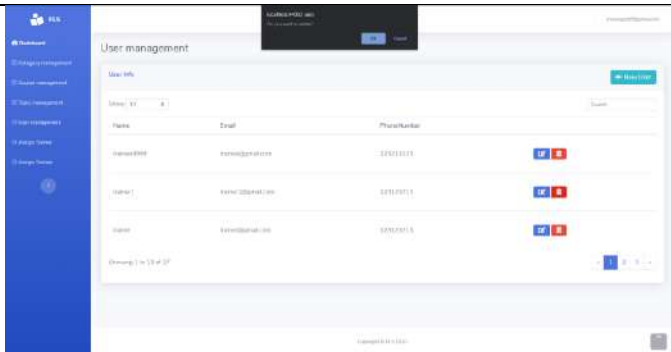


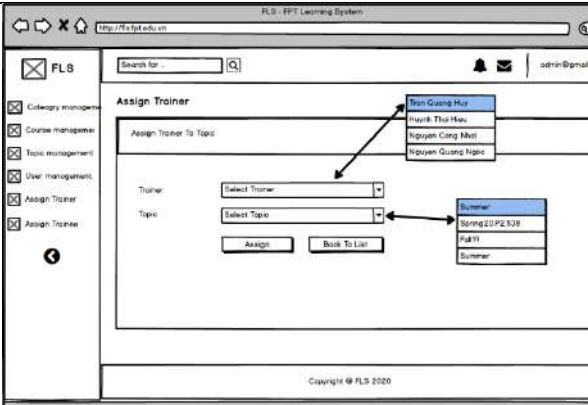
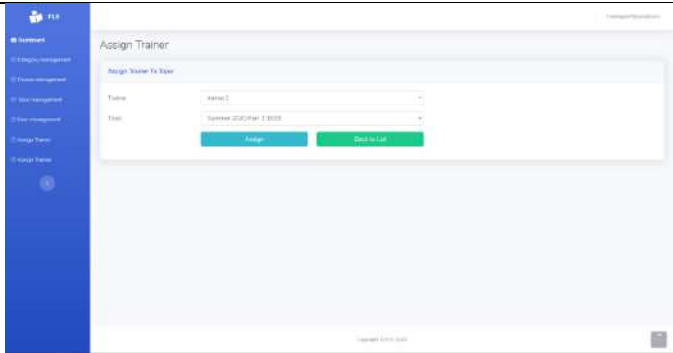
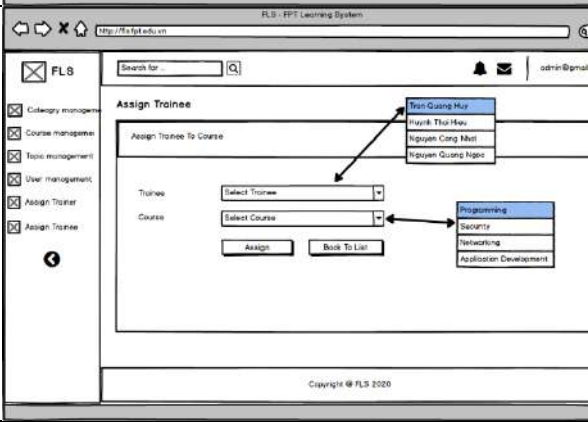
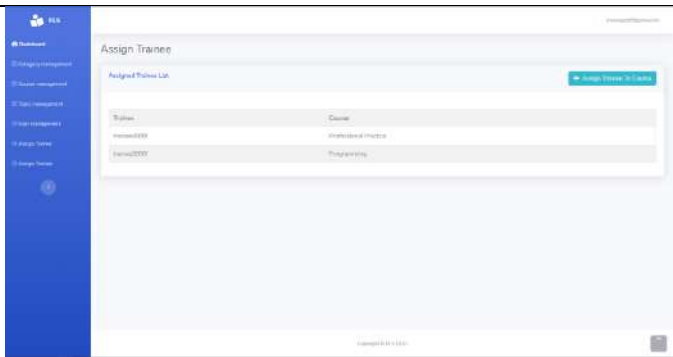
5	UC_TS 05	Training staff update trainee profile	 	100	Finished
6	UC_TS 06	Training staff delete exist account	 	100	Finished
7	UC_TS 07	Training staff view list of courses	 	100	Finished

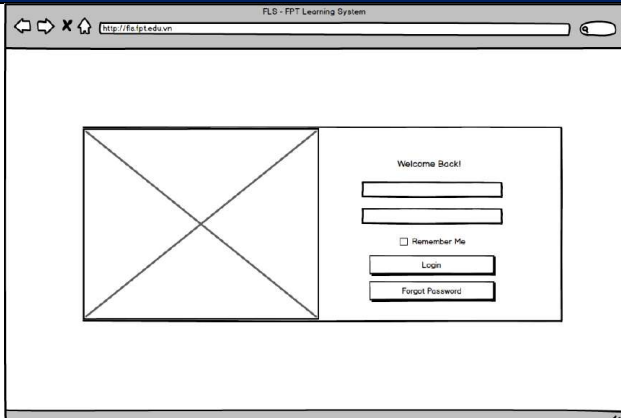
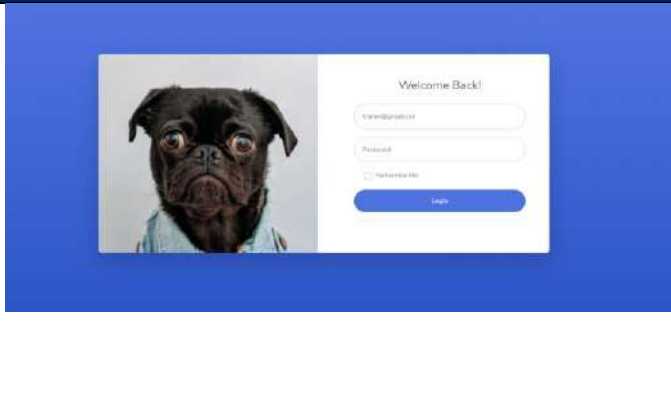
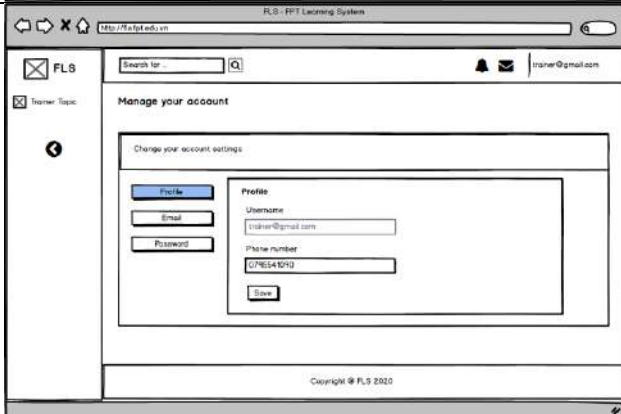
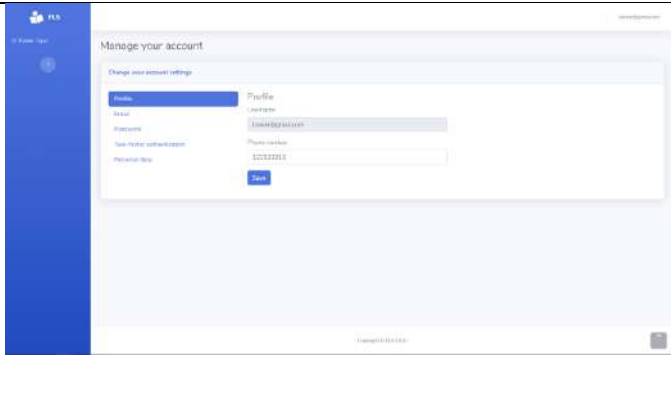
8	UC_TS 08	Training staff add course			100	Finished
9	UC_TS 09	Training staff update course			100	Finished
10	UC_TS 10	Training staff delete course			100	Finished

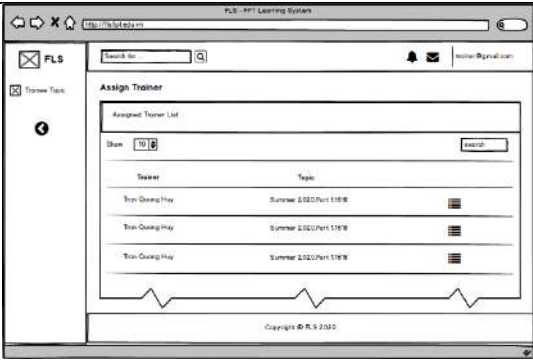
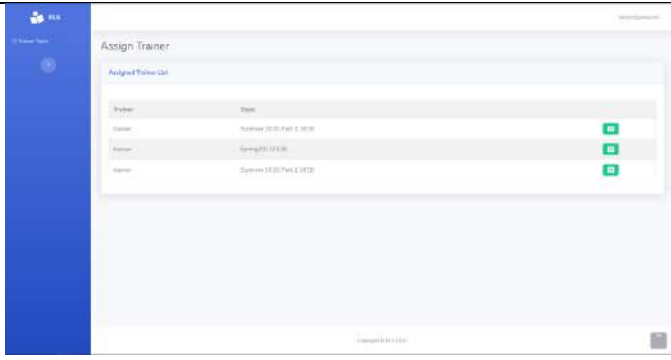
11	UC_TS 11	Training staff Add topic into course			100	Finished
12	UC_TS 12	Training staff add category			100	Finished
13	UC_TS 13	Training staff update category			100	Finished

14	UC_TS 14	Training staff delete Category			100	Finished
15	UC_TS 15	Training staff add course to category			100	Finished
16	UC_TS 16	Training staff view trainer profile			100	Finished

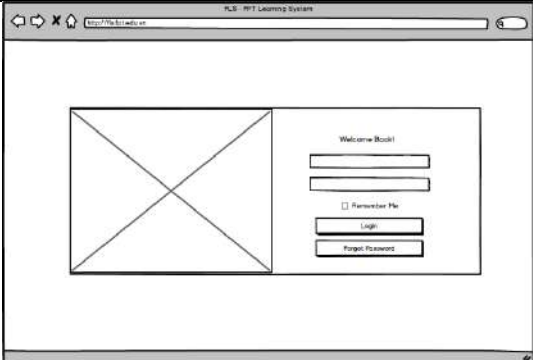
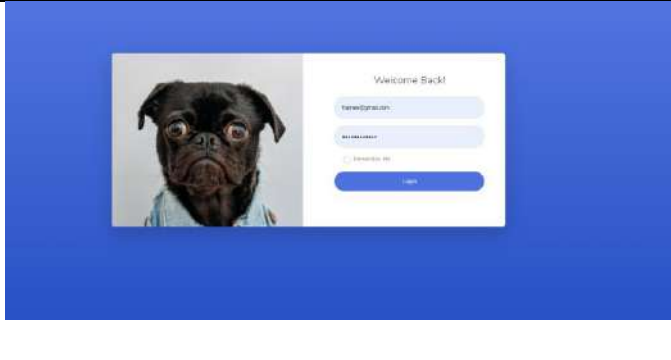
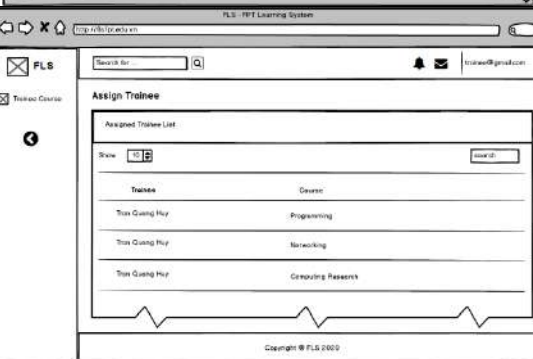
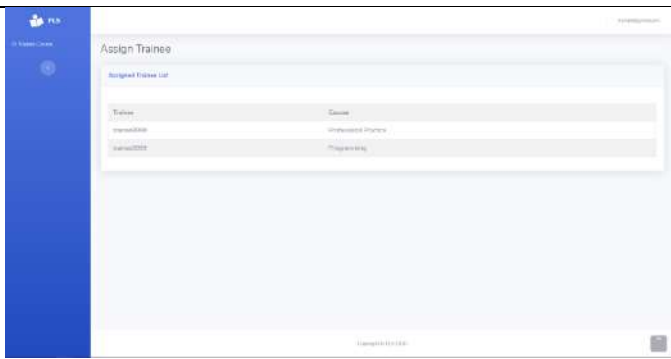
17	UC_TS 17	Training staff creates Trainer account	-	-	0%	Unnecessary
18	UC_TS 18	Training staff update trainer profile			100	Finished
19	UC_TS 19	Training staff delete exist trainer			100	Finished

20	UC_TS 20	Training staff assign a trainer to topic			100	Finished
21	UC_TS 21	Training staff assign trainee to course			100	Finished

Trainer						
1	UC_T N01	Trainer login			100	Finished
2	UC_T N02	Trainer update current Trainer profile			100	Finished

3	UC_T N03	Trainer view list of Topic	 	100	Finished
---	----------	----------------------------	---	-----	----------

Trainee

1	UC_T E01	Trainee login	 	100	Finished
2	UC_T E02	Trainee view list of courses	 	100	Finished



Most of the functions stated in the SRS are performed (95%). Based on the project's peer-review and requirement, several functions were added as well as removed, such as:

- Add a sub-table for show a Course, Topic exists in the database
- Dashboard in Training Staff index to analyze the data in the system.
- Remove permission of Training Staff to create trainer account
- Remove permission of Administrator to create trainee account
- The administrator can lock the user account
- Using icon from Font Awesome to improve UI

### **3.2. Analyze the factors**

#### **3.2.1. The factors influence the performance of the business application**

- **The network:**

Inconsistent bandwidth, variable contention, and increased latency can all have significantly negative effects on application performance.

The variability of connectivity over the network is one of the major performance risks for most businesses: if the developer can't control how information is being sent to the client, that can't control the speed at which it arrives. This makes it extremely hard to know exactly how the developer should develop FPT Learning System.

- **Third-Party Components**

Third-party web components are among the greatest factors affecting web application performance. If the FLS system depends on third-party data and that third party becomes unavailable, the system could fail. Likewise, poor performance by the third party can affect on system's performance.

- **Application Design**

Inefficient code algorithms at the application layer, suboptimal SQL queries, and poorly configured network infrastructure are all frequently major factors affecting application performance.

- **Mobile Environment**

Due to the increase in the number of users engaging with software via mobile devices like smartphones and tablets, mobile apps (or mobile versions of desktop apps) need to be designed with those platforms in mind for the best possible results.

- **Database**

Nearly 90 percent of application performance issues are related to the database layer (Data from Gleanster Research). This is due to several factors - databases are at the heart of applications, they are complex, and they are difficult to scale horizontally. When an application experiences rapid growth in demand, the database may indeed become a genuine bottleneck. Most other parts of the infrastructure can scale more easily than the database. Scaling out (where the application makes use of multiple database servers simultaneously for scaling reads) is difficult at the application layer.

- **The web browser**

The Web browser is the key blind spot for gaining true end-to-end visibility into application performance. With new approaches to application design and the increased usage of Web Services, the ability to monitor the processing that takes place within the browser has become one of the key requirements for full visibility into application performance.

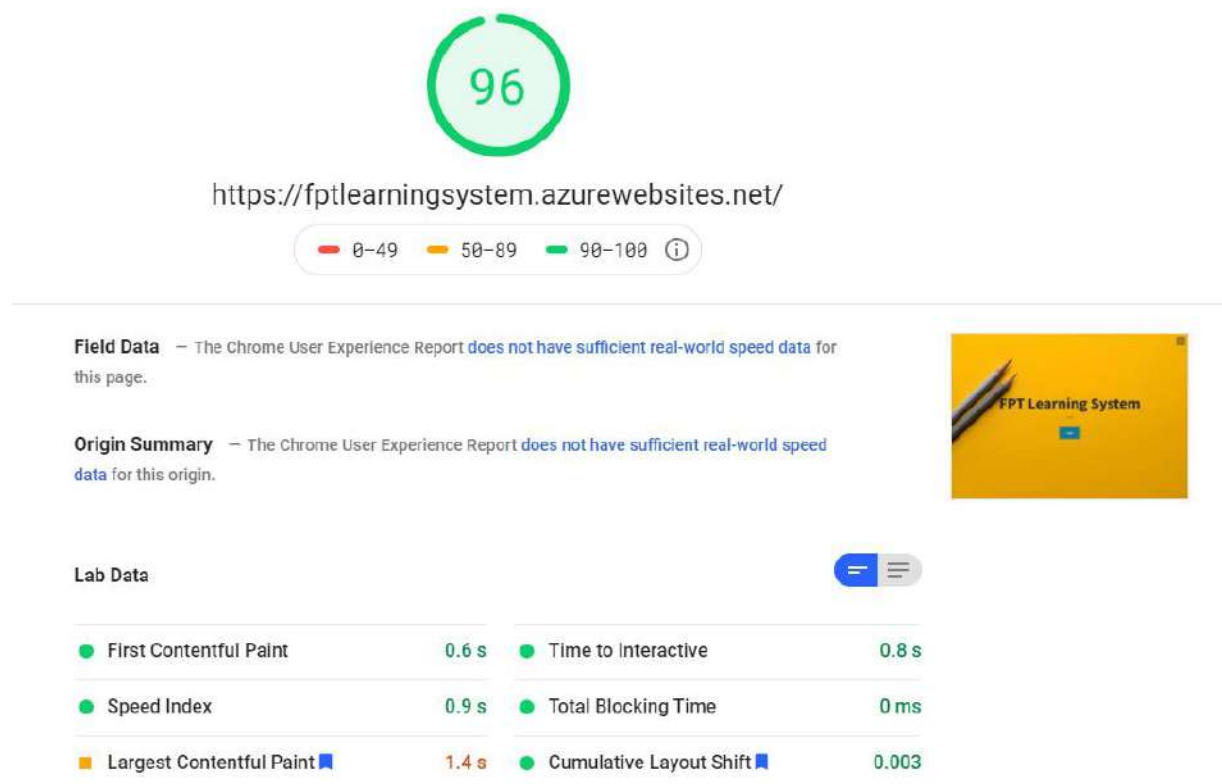
### 3.2.2. Critical review the design development and testing application

Base on Google PageSpeed Insights:

<https://developers.google.com/speed/pagespeed/insights/?url=https%3A%2F%2Ffptlearningsystem.azurewebsites.net%2F&tab=desktop>

There is the FPT Learning System result of performance:

- **Desktop:**



- **First Contentful Paint (0.6 s):** First Contentful Paint marks the time at which the first text or image is painted
- **Time to Interactive (0.8 s):** Time to interactive is the amount of time it takes for the page to become fully interactive
- **Speed Index (0.7 s):** Speed Index shows how quickly the contents of a page are visibly populated
- **Total Blocking Time (0 ms):** Sum of all periods between FCP and Time to Interactive, when task length exceeded 50ms, expressed in milliseconds
- **Largest Contentful Paint (1.4 s):** Largest Contentful Paint marks the time at which the largest text or image is painted
- **Cumulative Layout Shift: (0.003):** Cumulative Layout Shift measures the movement of visible elements within the viewport

Opportunity	Estimated Savings
■ Serve images in next-gen formats	 0.52 s ▾
■ Remove unused CSS	 0.16 s ▾
▲ Reduce initial server response time	 0.1 s ▾

**Diagnostics** — More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

▲ Ensure text remains visible during webfont load	▾
▲ Serve static assets with an efficient cache policy — 13 resources found	▾
● Avoid chaining critical requests — 20 chains found	▾
● Keep request counts low and transfer sizes small — 28 requests • 1,471 KiB	▾
● Largest Contentful Paint element — 1 element found	▾
● Avoid large layout shifts — 2 elements found	▾

The system passed a lot of audits condition. There are 2 audits that the system did not pass such as User timing Marks and measures and Avoid long main-thread tasks.

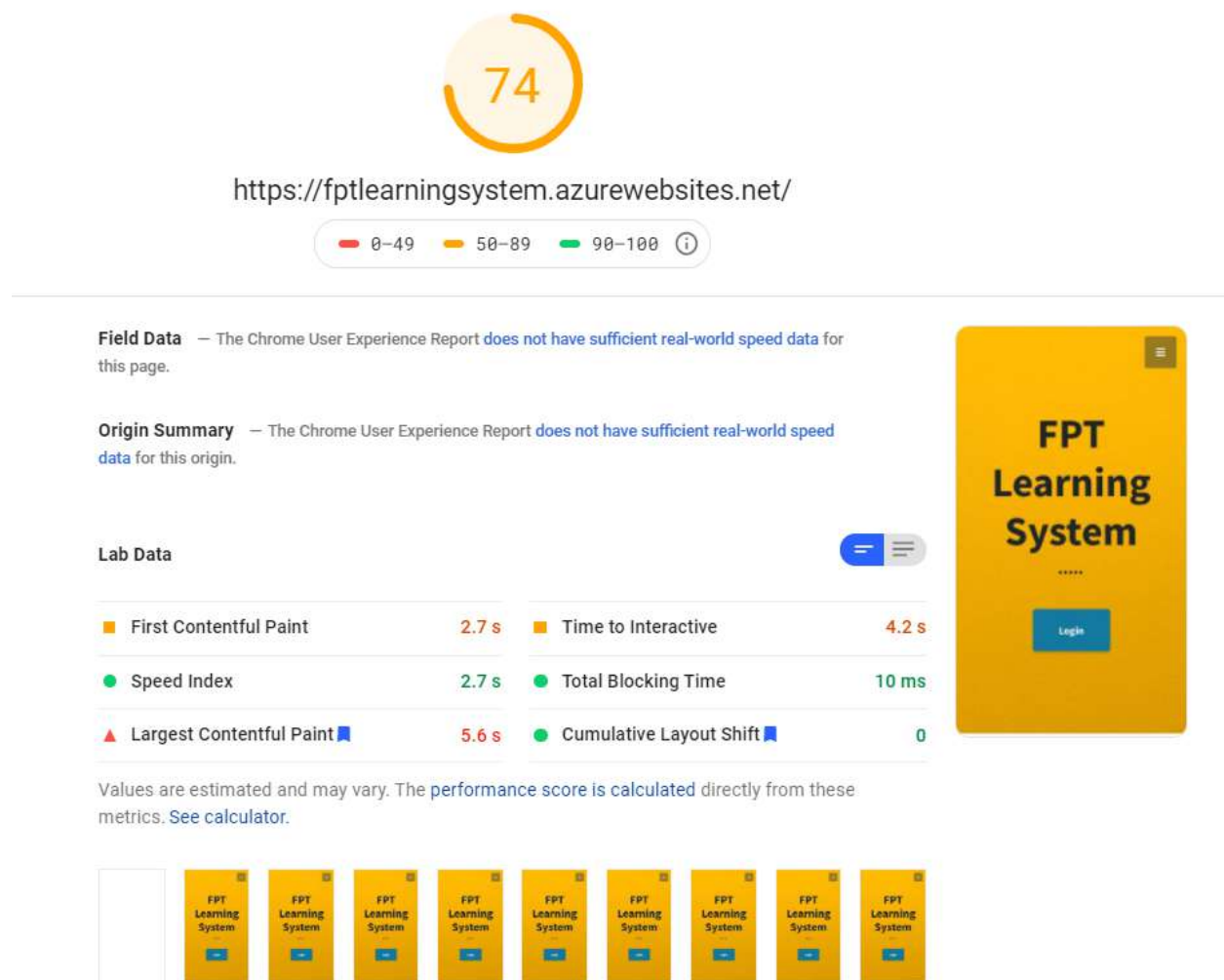
Passed audits (21)	^
● Eliminate render-blocking resources — Potential savings of 70 ms	▼
● Properly size images — Potential savings of 164 KiB	▼
● Defer offscreen images	▼
● Minify CSS	▼
● Minify JavaScript	▼
● Remove unused JavaScript — Potential savings of 28 KiB	▼
● Efficiently encode images — Potential savings of 177 KiB	▼
● Enable text compression	▼
● Preconnect to required origins	▼
● Avoid multiple page redirects	▼
● Preload key requests	▼
● Use video formats for animated content	▼
● Avoids enormous network payloads — Total size was 1,471 KiB	▼
● Avoids an excessive DOM size — 128 elements	▼
● User Timing marks and measures	▼
● JavaScript execution time — 0.0 s	▼
● Minimizes main-thread work — 0.2 s	▼
● Minimize third-party usage — Third-party code blocked the main thread for 0 ms	▼
● Uses passive listeners to improve scrolling performance	▼
● Avoids <code>document.write()</code>	▼
● Avoid long main-thread tasks	▼

## Mobile:

The application running in mobile will be slower than desktop, that because when developer develops this application just aim to build an application on Desktop and did not consider for mobile application.

The application in the next gen in the future will be considered to develop to run better on mobile.

In this time the result is acceptable.



**Opportunities** — These suggestions can help your page load faster. They don't **directly affect** the Performance score.

Opportunity	Estimated Savings
▲ Serve images in next-gen formats	3.3 s ▼
▲ Efficiently encode images	0.9 s ▼
■ Eliminate render-blocking resources	0.48 s ▼
■ Remove unused CSS	0.3 s ▼
■ Remove unused JavaScript	0.15 s ▼

**Diagnostics** — More information about the performance of your application. These numbers don't **directly affect** the Performance score.

▲ Ensure text remains visible during webfont load	▼
▲ Serve static assets with an efficient cache policy — 13 resources found	▼
● Avoid chaining critical requests — 19 chains found	▼
● Keep request counts low and transfer sizes small — 27 requests • 1,395 KiB	▼
● Largest Contentful Paint element — 1 element found	▼
● Avoid long main-thread tasks — 2 long tasks found	▼

Your speed results for `ftplearningsystem.azurewebsites.net`

**Your mobile page speed is 3.2 seconds on a 4G connection.**

RATING

**Poor**

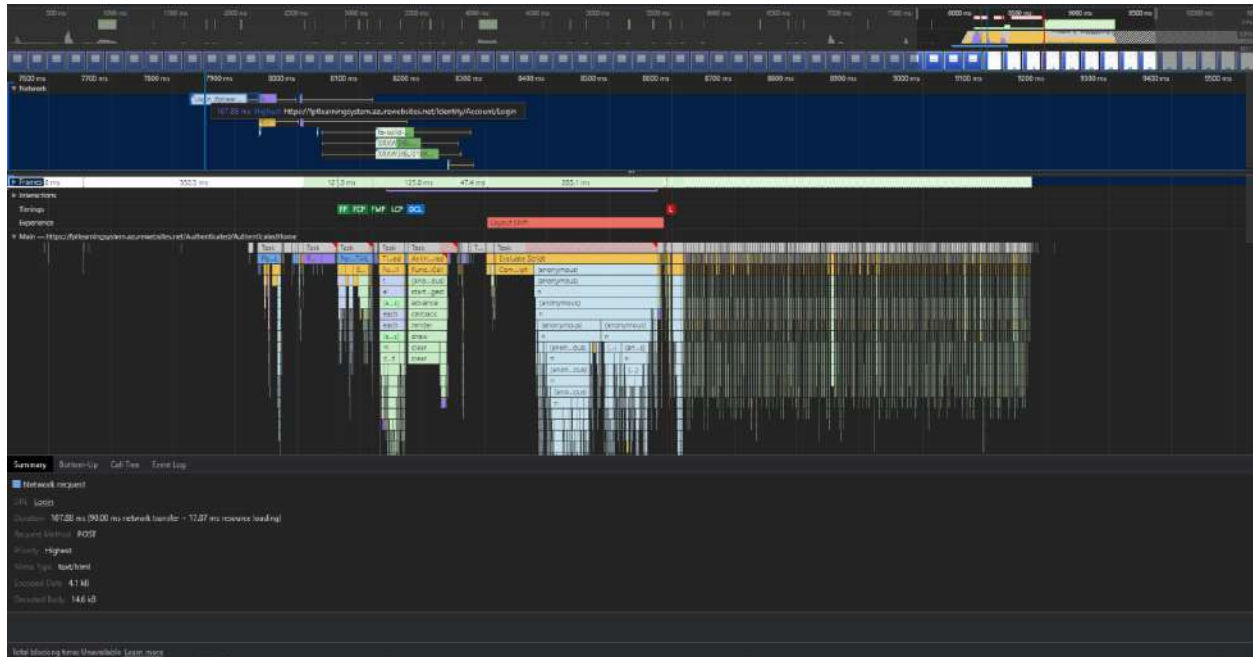
Poor sites start to load in over 2.5 seconds.  
[Learn more](#)



Testing application:

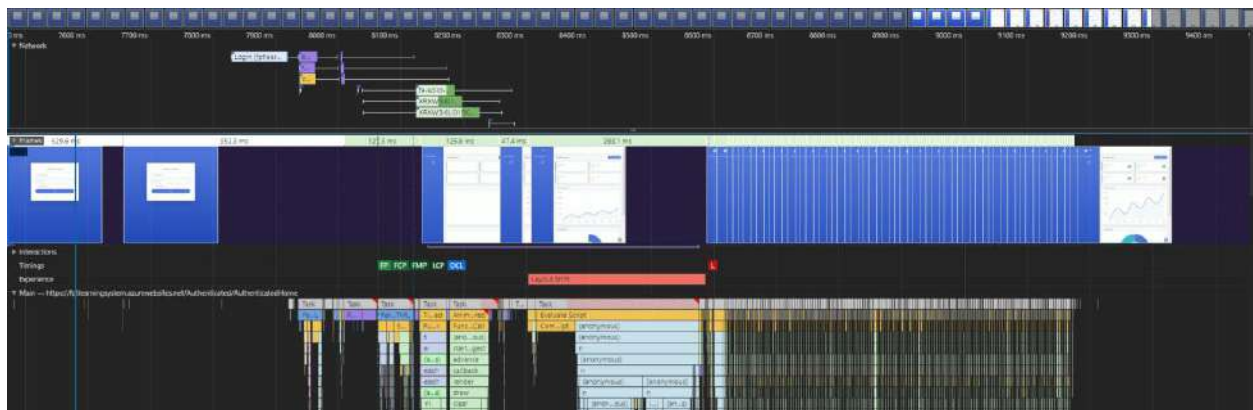
Function login:

The system costs 107.88 ms to login into the system.



Picture 67. Performance login

If login successfully, the FLS will move to the dashboard page. The system cost 121.3ms to redirect from login to dashboard page with Administrator role.

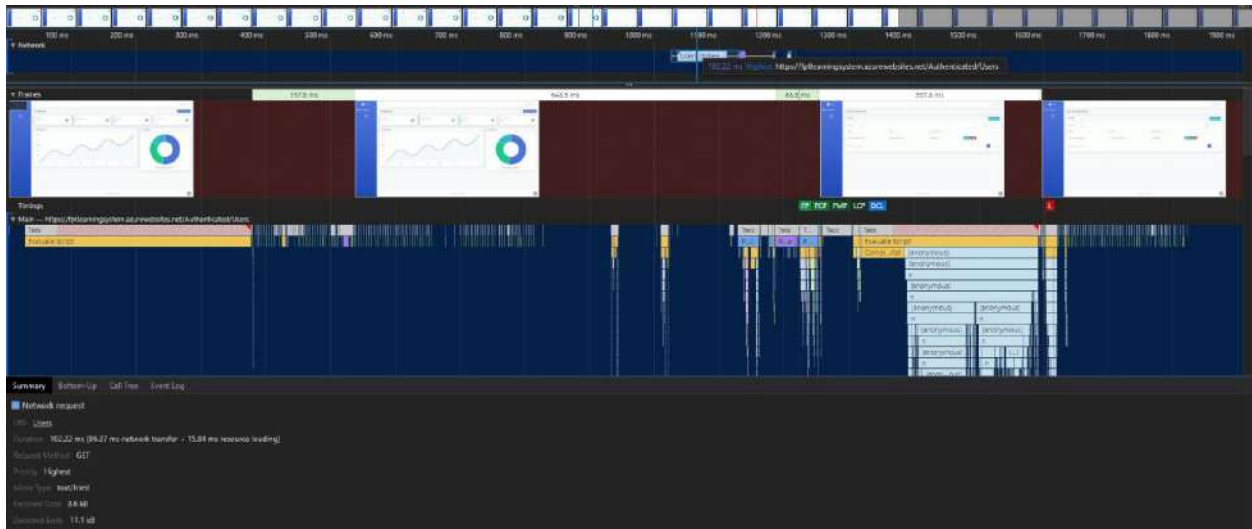


Picture 68. Performance dashboard page



### View user:

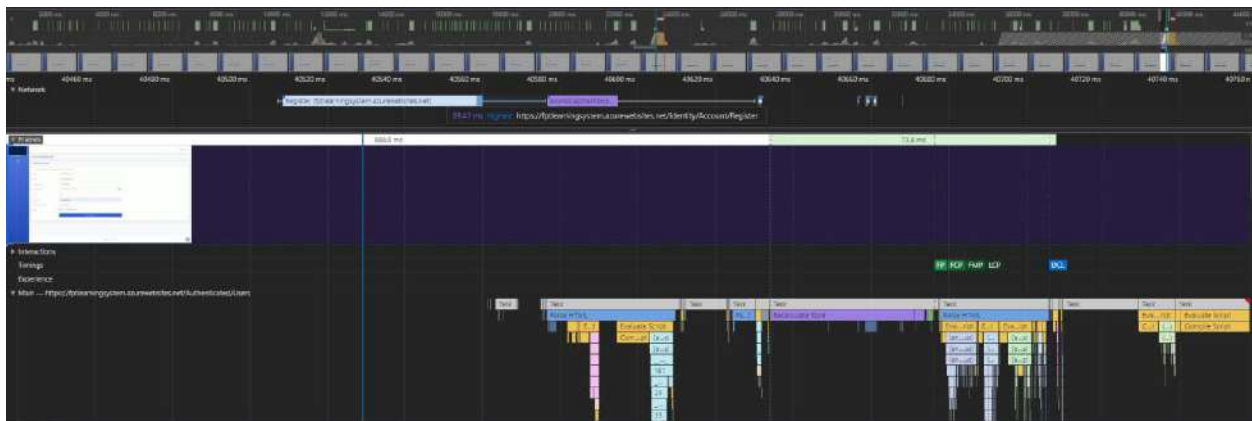
The FLS system costs 102.22 ms to get data from the database and render the user management view.



Picture 69. Performance view user

### Create user:

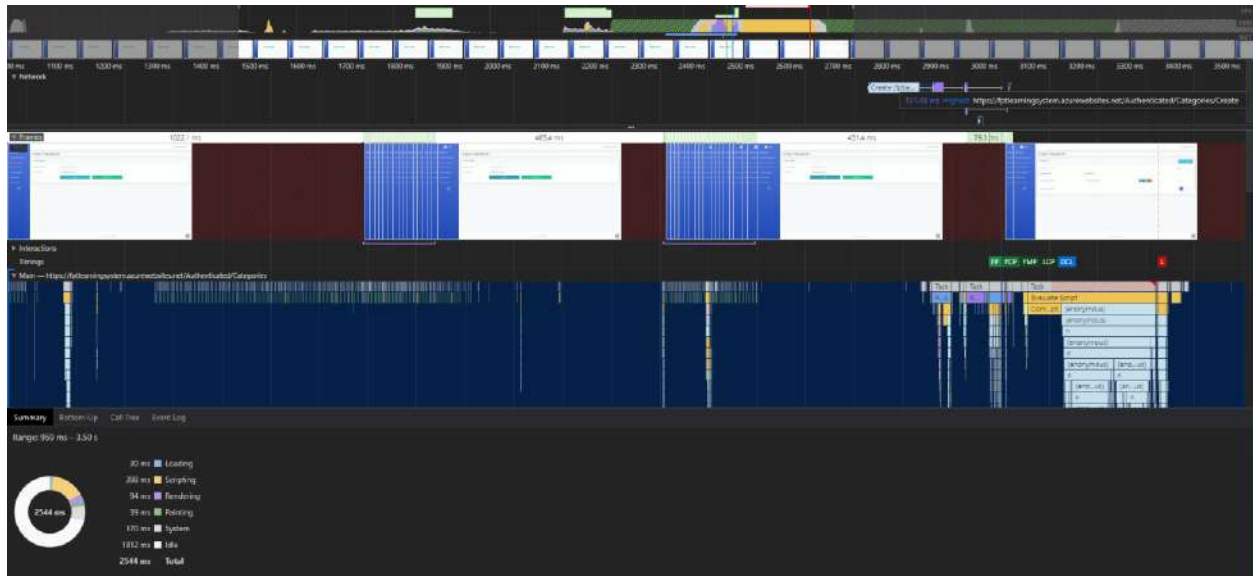
The FLS System costs 69.43 ms to create a new user and 53.52 to redirect to the index page.



Picture 70. Performance create user

### Create category:

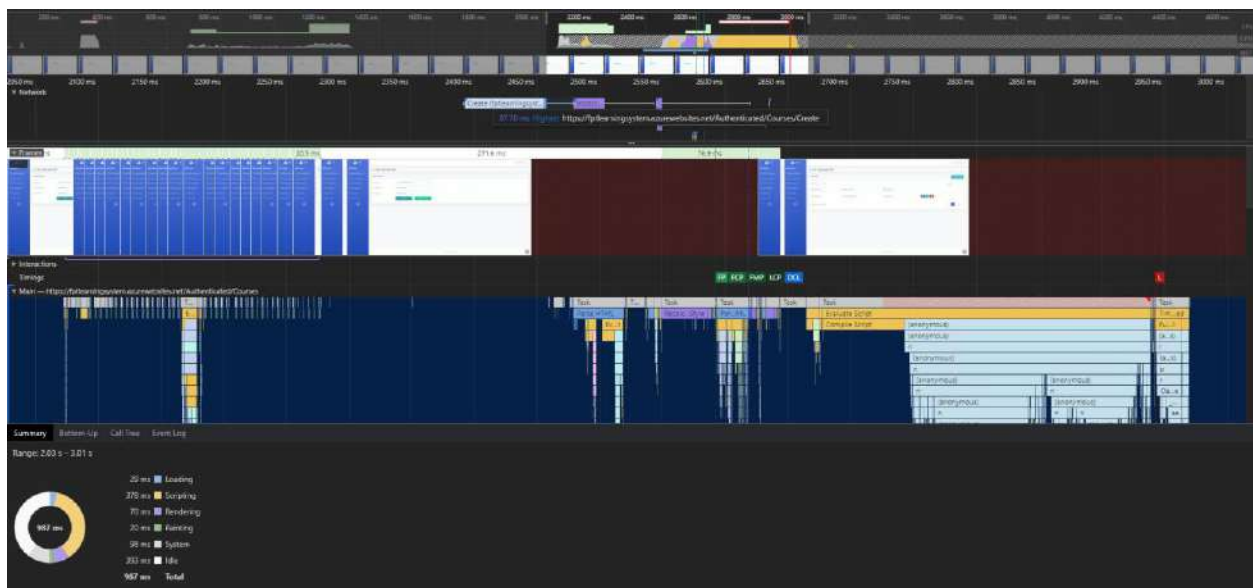
With TrainingStaff Role, the system will cost 131.03 ms to create a category and 63.99 ms to redirect to the index page.



Picture 71. Performance create category

### Create a Course:

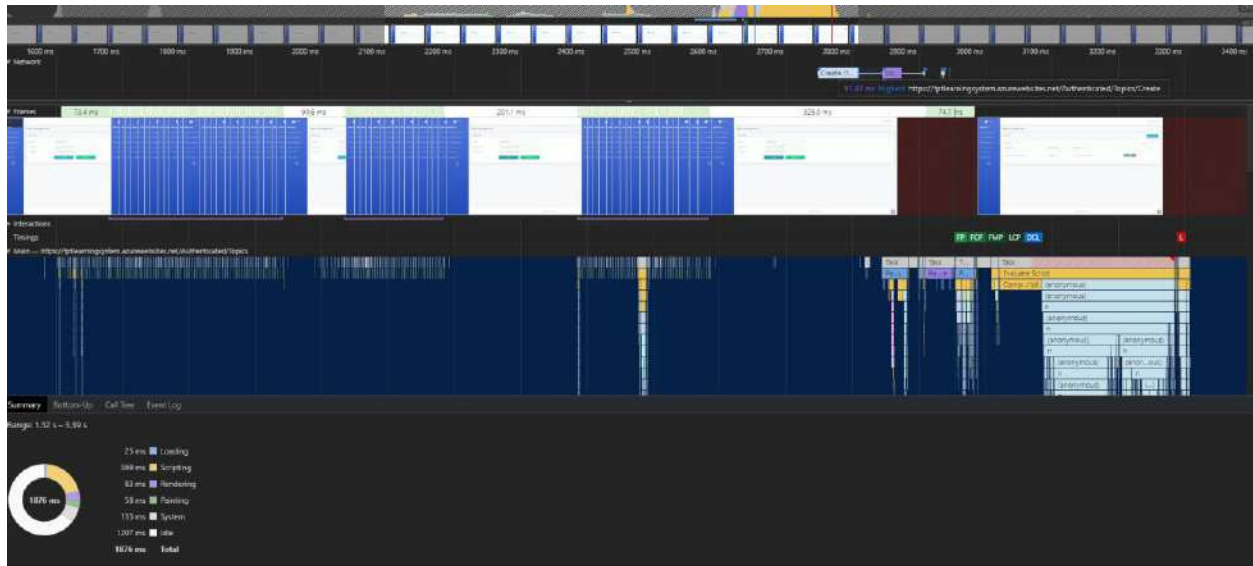
With TrainingStaff Role, the system will cost 87.70 ms to create a course and 64.63 ms to redirect to the index page.



Picture 72. Performance creates course

## Create topic:

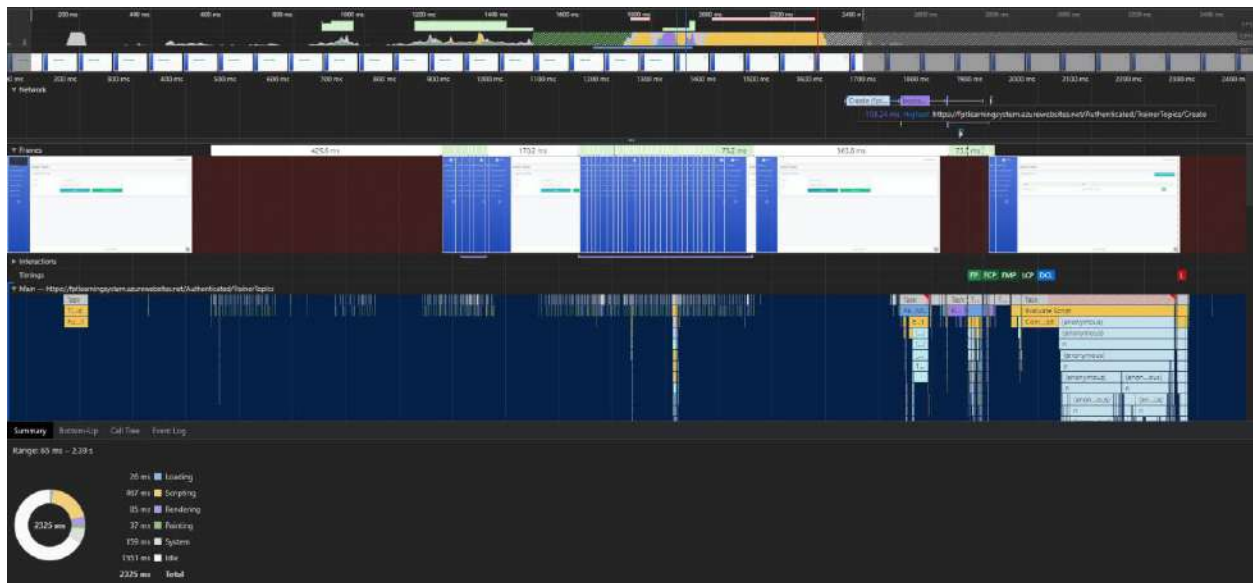
With TrainingStaff Role, the system will cost 97.37 ms to create a topic and 60.56 ms to redirect to the index page.



Picture 73. Performance Create Topic

## Assign Trainer to topic:

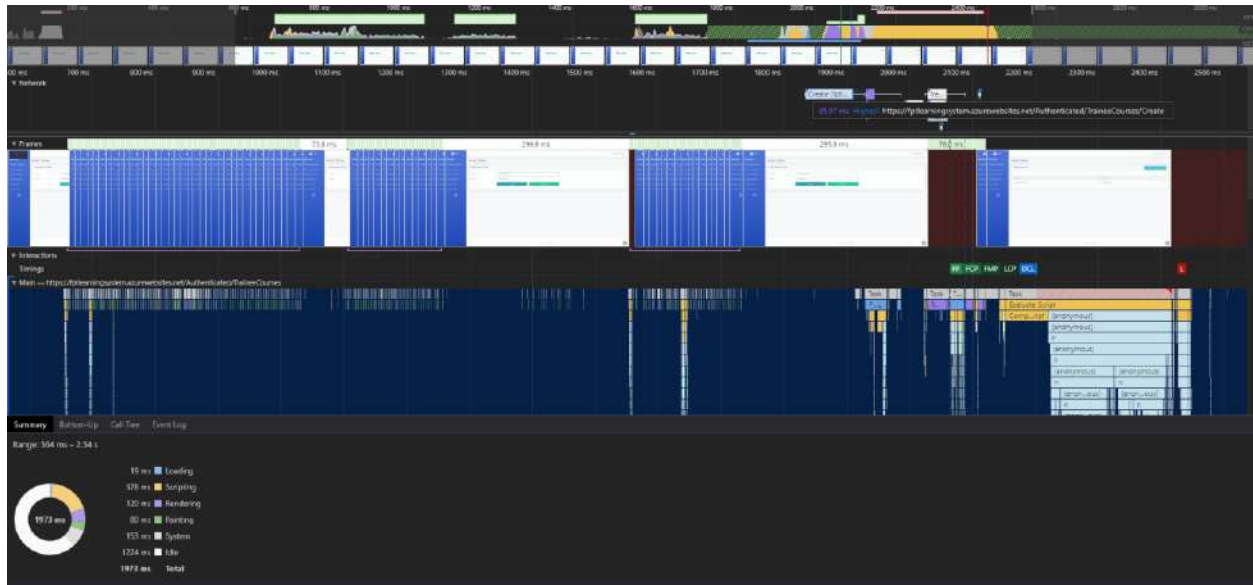
With TrainingStaff Role, the system will cost 103.24 ms to assign Trainer to the topic and 86.26 ms to redirect to the index page.



Picture 74. Performance Assign Trainer to Topic

### Assign Trainee to Course:

With TrainingStaff Role, the system will cost 95.97 ms to assign Trainee to the course and 56.02 ms to redirect to the index page.



Picture 75. Performance Assign Trainee to Course

Based on the testing results, all system functions are successful and executed immediately (<100ms). The user experience with the system is very good. The parameters also indicated that the system functionality met the requirements.

The problems that the results show are mainly in the rendering of the UI, heavy in JavaScript, and Bootstrap files.

### **3.3. Assessment of the system's strength and weakness and improve system**

#### **3.3.1. System's strength and weakness**

- **System's strength**

- **Loading website:**

Base on the result of Google PageSpeed Insights the application get 96 point and green color. That shows the system is good to run in Desktop. And the content in the website running fast enough. In general, only metrics contribute to your Lighthouse Performance score, not the results of Opportunities or Diagnostics. That said, improving the opportunities and diagnostics likely improves the metric values.

- The website has a very fast page load time (less than 1s).
    - The login page or management of user information, course code management, topic management, management of categories, assign a trainer to the topic and assign trainee to course all load very fast and there is no delay.

- **UI Design:**

By using templates from Bootstrap studio, the system has a more user-friendly UI. Users can easily see the necessary information when using this Application.

- **User-focused on interface**

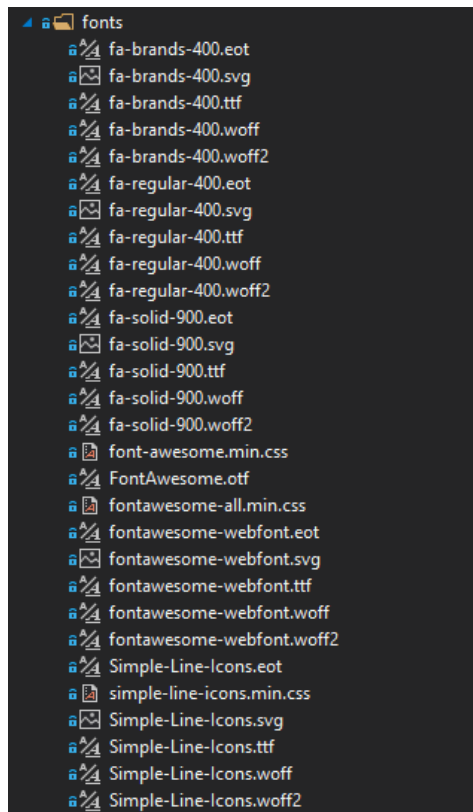
The buttons as well as the content are arranged in order, besides, the redundant content is discarded to avoid unnecessary distraction to the user.

The information about the user, the courses, and the topics displayed is clear and detailed.

Function buttons are arranged according to the original design, clear and complete.

- **Typography and pictures**

Fonts used on the system from Font Awesome are easy to read and see



Picture 76. Font use in the system

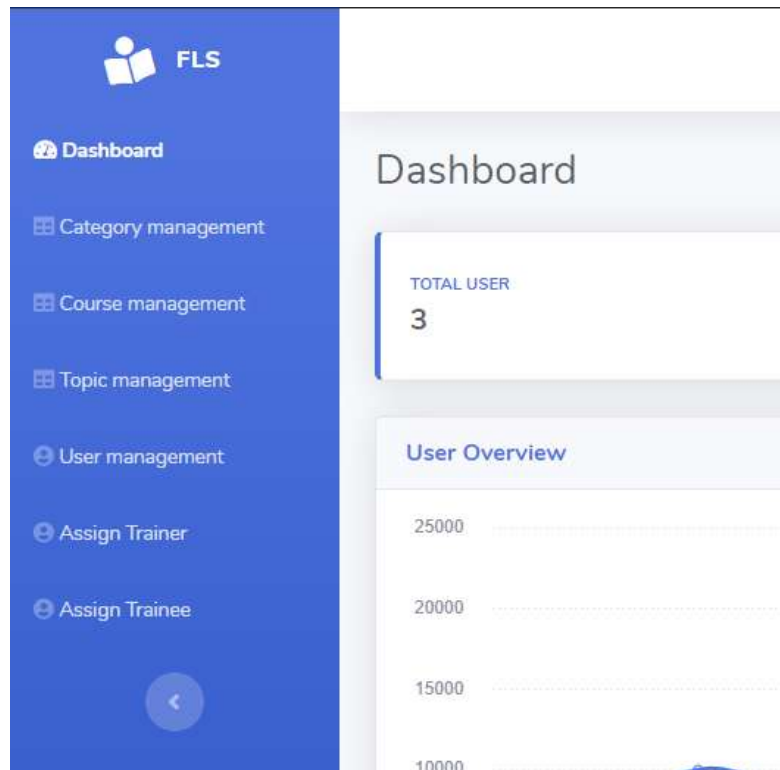
Default picture is colorful and meaning in the index page when the user using the system:



Picture 77. Pictures on the index page

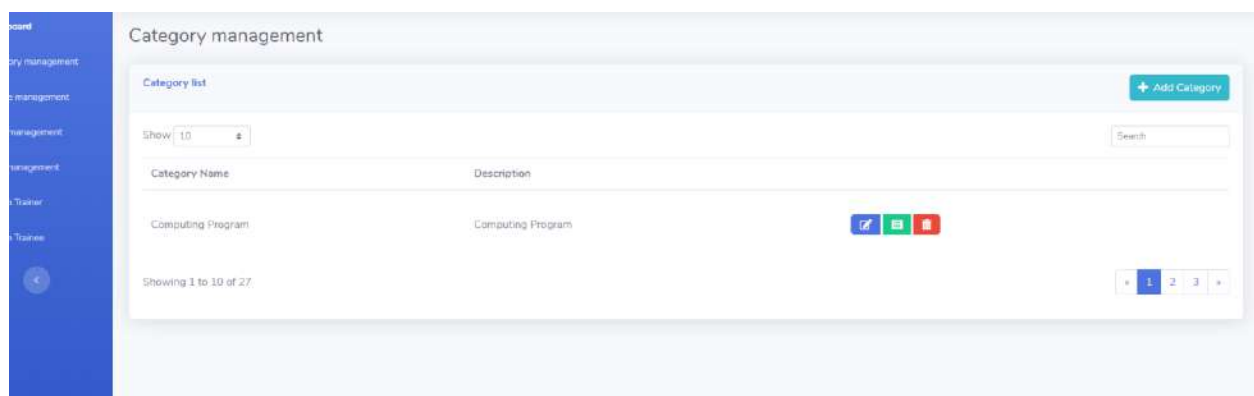
- **Content structure**

The option to choose contents of the Roles are shown on the right side of the screen, users are easy to use and manage what features they are using.



Picture 78. Content Options

The main contents such as data tables, creating such as editing and deleting information are designed in the middle of the screen.

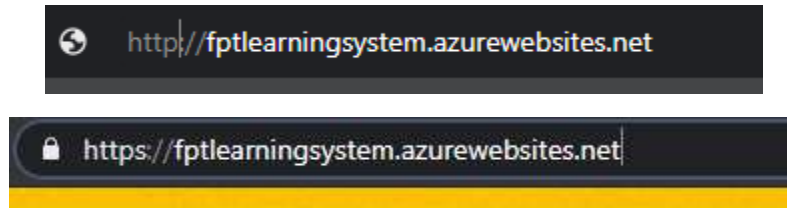


Picture 79. Main content



- **Redirects HTTP traffic to HTTPS**

When a user using **HTTP**, the FLS system will auto redirect to **HTTPS**, that will make the system more security



Picture 80. Redirect HTTP to HTTPS

- **Function**

The functions on the website are easy to use. such as add user functions, editing functions, delete functions, detailed, assign.

The way to use the functions is so easy which the button easy to find.

The set functions are not redundant, very focused on user needs, very suitable for the user to have less knowledge about using the website.

- **Security**

The user's password is encrypted before being put into the database and reverse-encrypted when the login is used.

Functions are decentralized. Each user can only use the correct functions that the system allows. All bypass methods are not allowed on the system, the system will notify users who are not allowed to use that feature.

Token after login of the user will be encrypted and saved in the cookie.

Application										
Filter										
Only show cookies with an issue										
Name	Value	Do...	Path	Expir...	Size	Http...	Secu...	Sam...	Prior...	
.AspNetCore.Identity.A...	CfDJ8lMo_fi03kJBrqksWfItgHrKMa...	fptle...	/	Sessi...	784	✓	✓	Lax	Med...	
.AspNetCore.Antiforge...	CfDJ8lMo_fi03kJBrqksWfItgHrdleYo...	fptle...	/	Sessi...	190	✓		Strict	Med...	
ARRAffinity	df29efd2b2059a0fc5b5912902d95a...	.fptl...	/	Sessi...	75	✓			Med...	
1P_JAR	2020-08-25-13	.gsta...	/	2020...	19		✓	None	Med...	



- **System's weakness**
  - **UI/UX Design:**

The current page does not respond with a 200 when offline **start\_url** does not respond with a 200 when offline.

Because of using the regular version Font Awesome, the system is limited to the number of icons used. Therefore, some icons have to use equivalent icons which leads to the icon not matching the buttons or functions mentioned by the system.

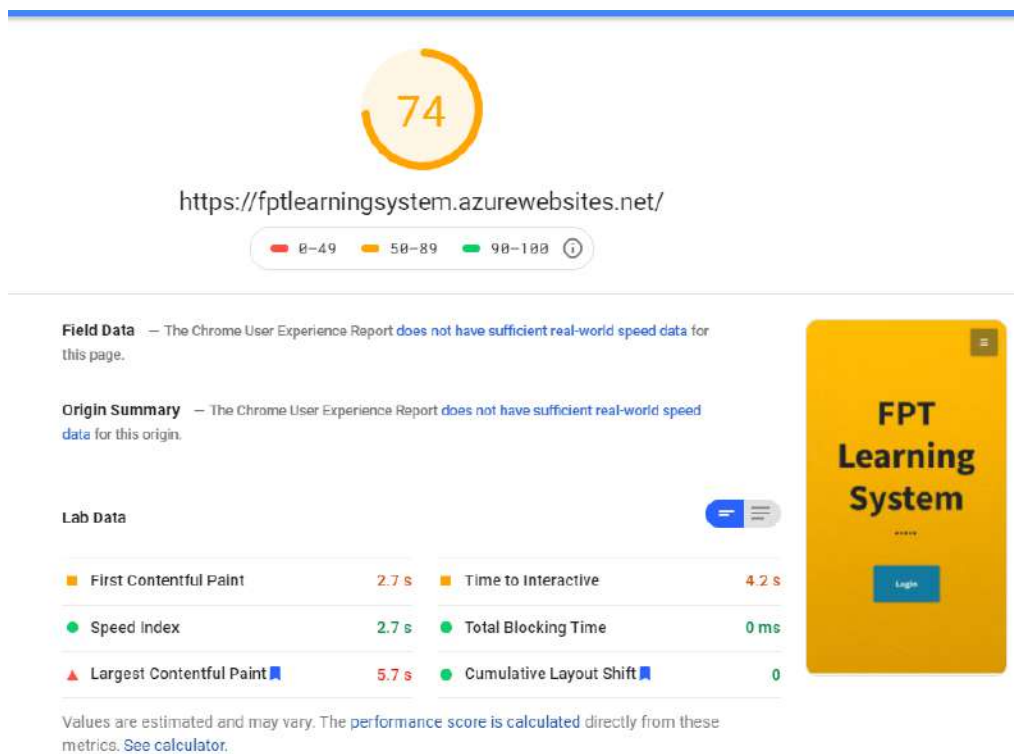
- **Interactive**

There is no notification function for the user, especially the trainee, to hold information about the assigned courses.

There is no chat box for communicating between Trainer, Training Staff, and Trainee.

- **The system runs on mobile**

The performance of the FLS system to run on mobile is lower than desktop



- **Security**

The system did not implement environment key so the information not secure especially is the role

- **Implement code:**

Too much query to get data from the database

```
public async Task<IActionResult> Create()
{
    var traineeRoleId = await _db.Roles.Where(t => t.Name == SD.Trainee).Select(t => t.Id).FirstAsync();

    var listTraineeId = await _db.UserRoles
        .Where(t => t.RoleId == traineeRoleId)
        .Select(t => t.UserId)
        .ToArrayAsync();

    List<ApplicationUser> traineeUsers = await _db.ApplicationUsers
        .Where(t => listTraineeId
            .Any(name => name.Equals(t.Id)))
        .ToListAsync();

    TraineeCourseVM = new TraineeCourseViewModel()
    {
        Courses = await _db.Courses.ToListAsync(),
        Users = traineeUsers,
        TraineeCourse = new Models.TraineeCourse(),
    };
}
```

- **Unnecessary CSS and JavaScript**

Unnecessary code and minify all CSS and JavaScript.

Remove unused CSS0.3 s ^			
Remove dead rules from stylesheets and defer the loading of CSS not used for above-the-fold content to reduce unnecessary bytes consumed by network activity. <a href="#">Learn more.</a>			
URL	Transfer Size	Potential Savings	
/css/bootstrap.min.css (fptlearningsystem.azurewebsites.net)	37.5 KiB	35.1 KiB	
...css/free.min.css (kit-free.fontawesome.com)	13.8 KiB	13.7 KiB	

Remove unused JavaScript0.15 s ^			
Remove unused JavaScript to reduce bytes consumed by network activity. <a href="#">Learn more.</a>			
URL	Transfer Size	Potential Savings	
/js/jquery.min.js (fptlearningsystem.azurewebsites.net)	38.9 KiB	27.9 KiB	

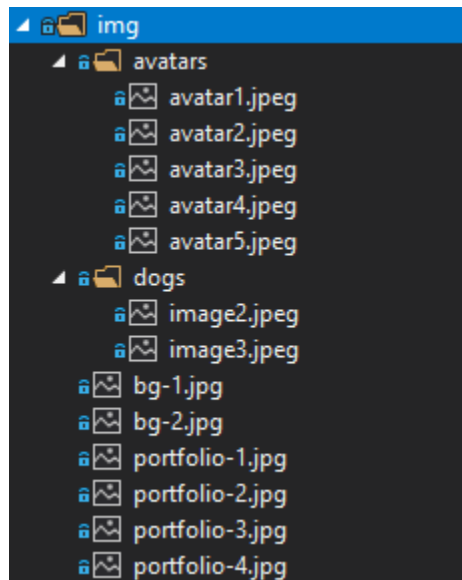
Picture 81. Unnecessary CSS and JavaScript

### 3.3.2. Improvement system

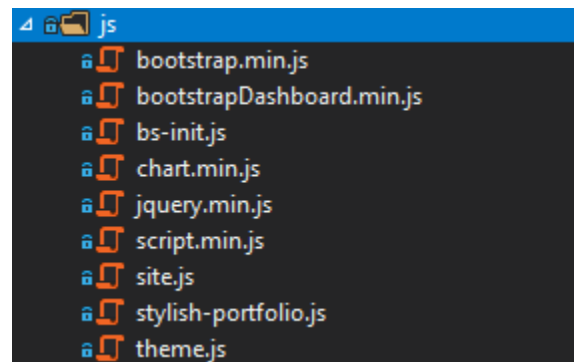
#### Boost speed:

Customers are impatient, so the speed of your site has a big impact on conversion and bounce rates. Speed up the site to make the best first impression on the customers.

- **Update images:** Reduce the size of image files to loading faster. Additionally, load images offscreen to increase page speed. The replacement of a PNG with a JPEG image file can easily save your page size and load time.



- **Clean up your code:** Remove unnecessary code and minify all CSS and JavaScript to make sure the site loads and runs as quickly as possible.



- **Load site faster with Progressive Web Apps (PWA)**

## Improve UX design:

- **Make the system seamless:**
  - **Seamless sign-in** Allow users to sign in with their Google account and use Autofill to automatically populate their saved information for faster login.
  - **Enhance the website experience:** Progressive Web Apps (PWAs) deliver native-like capabilities, reliability, and installability while reaching anyone, anywhere, on any device.
- **Security code:**

By using the RoleId copy from the database, the developer will use another query to get an ID and store it into the variable and reuse it in another function in the controller.

```
var listTraineeId = await _db.UserRoles
    .Where(t => t.RoleId == "cdd4e45c-5394-4924-a900-6537dcc09996")
    .Select(t => t.UserId)
    .ToArrayAsync();
```

```
var traineeRoleId = await _db.Roles.Where(t => t.Name == SD.Trainee).Select(t => t.Id).FirstAsync();

var listTraineeId = await _db.UserRoles
    .Where(t => t.RoleId == traineeRoleId)
    .Select(t => t.UserId)
    .ToArrayAsync();
```

```
var result = await _userManager.CreateAsync(user, Input.Password);
if (result.Succeeded)
{
    if (role == SD.Trainee && (User.IsInRole(SD.Administrator) || User.IsInRole(SD.TrainingStaff)))
    {
        await _userManager.AddToRoleAsync(user, SD.Trainee);
    }
    else
    {
        if (role == SD.Trainer && (User.IsInRole(SD.Administrator) || User.IsInRole(SD.TrainingStaff)))
        {
            await _userManager.AddToRoleAsync(user, SD.Trainer);
        }
        else
        {
            if (role == SD.TrainingStaff && User.IsInRole(SD.Administrator))
            {
                await _userManager.AddToRoleAsync(user, SD.TrainingStaff);
            }
            else
            {
                await _userManager.DeleteAsync(user);
            }
        }
    }
}
```

- **Serve images in next-gen formats:**

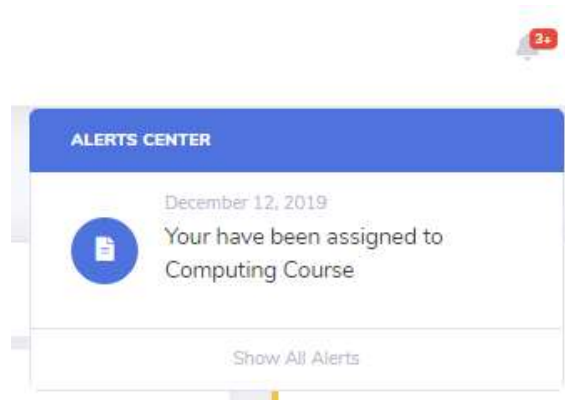
Image formats like JPEG 2000, JPEG XR, and WebP often provide better compression than PNG or JPEG, which means faster downloads and less data consumption

WebP is supported in Chrome and Opera and provides better lossy and lossless compression for images on the web

A developer can create WebP Images with the command Line from <https://web.dev/codelab-serve-images-webp/>

- **Implement notification:**

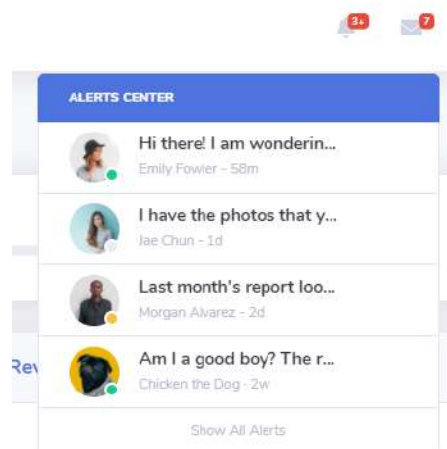
Notification function will help the user know what happened on the website especially is the trainee can know what course that he/she has been assigned or the trainer can know what topic that he/she has been assigned.



Picture 82. Implement notification

- **Implement chatbox:**

Communication is important for this application. Example: Trainee needs to ask Training Staff about assign he/her in the wrong course.



Picture 83. Implement chatbox

## Conclusion

From the requirements of the system and based on SRS, FPT Learning System has been developed and developed. The application was developed from a simple local system and then deployed to the cloud service of Azure.

The system functions have been reevaluated and changed to suit the software and better interact with users.

FPT Learning System in the report is in the first version and needs many improvements to better serve the school environment.

**Live demo:** <https://fptlearningsystem.azurewebsites.net/>

- **Administrator:** [admin@gmail.com](mailto:admin@gmail.com) – Password@123
- **Training Staff:** [trainingstaff@gmail.com](mailto:trainingstaff@gmail.com) – Password@123
- **Trainer:** [trainer@gmail.com](mailto:trainer@gmail.com) – Password@123
- **Trainee:** [trainee@gmail.com](mailto:trainee@gmail.com) – Password@123

**Source code:** <https://github.com/SuperMido/FptLearningSystem>