ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 30: Application Development		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Tran Duc Luong	**Student ID**	BH00108
Class	IT0502	**Assessor name**	Nguyen Thanh Trieu

Student declaration

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.

Student's signature	Luong

Grading grid

P4	P5	P6	M3	M4	M5	D2	D3

❒ Summative Feedback:	❒ Resubmission Feedback:	
Grade:	**Assessor Signature:**	**Date:**
Lecturer Signature:		

Nội dung

A. Introduction

In the intricate endeavor of constructing a robust business application, our team meticulously undertook a comprehensive process that encompassed not merely conceptualization and design, but also rigorous evaluation and iterative refinements. This report comprehensively details our systematic approach, chronicling the evolution from initial design concepts, as presented earlier, to the development phase, meticulously capturing the effective utilization of selected technologies and methodologies.

Throughout this endeavor, we actively sought and incorporated feedback from our esteemed peers through a formal questionnaire. This valuable input enabled us to adapt reflectively, incorporating insights into our development process. Moreover, our report critically analyzes the application's performance against pre-established specifications, delving into the intricate factors that influence its efficacy.

The culmination of our efforts is a comprehensive presentation and a fully functional application for demonstration and evaluation. In the concluding chapter, we present a reflective analysis, profoundly acknowledging the application's strengths, identifying potential weaknesses, and envisioning opportunities for future enhancements. This report serves as a testament to our unwavering commitment to delivering high-caliber business applications that empower businesses to achieve their strategic objectives.

B. P4 Create a formal questionnaire that effectively reviews your business application, problem definition statement, proposed solution and development strategy. Use this questionnaire as part of a peer-review and document any feedback given.

1. Formal questionnaire

Formal Questionnaires for Data Collection and Analysis

A formal questionnaire is a structured method of gathering feedback or information. Its standardized format ensures consistency in data collection, facilitating analysis and the extraction of meaningful insights.

Benefits of Formal Questionnaires:

Organized Data Collection: A formal questionnaire organizes responses in a systematic manner, making it easier to quantify and analyze data.

Consistent Feedback: All respondents answer the same set of questions, allowing for fair comparisons and the identification of common themes or issues.

Improved Data Analysis: The structured format enables efficient data analysis, allowing researchers to identify trends, patterns, and areas for improvement.

Enhanced User Feedback: Formal questionnaires provide a platform for users to express their experiences and suggestions, helping organizations gather valuable feedback to enhance user satisfaction and engagement.

Application in FPT Training Web Application:

Creating a formal questionnaire for the FPT Training Web Application is a valuable tool for gathering feedback and identifying areas for improvement. By distributing the questionnaire using Google Forms, I can gather insights on the following:

User experience and satisfaction levels

Perceived strengths and weaknesses of the application

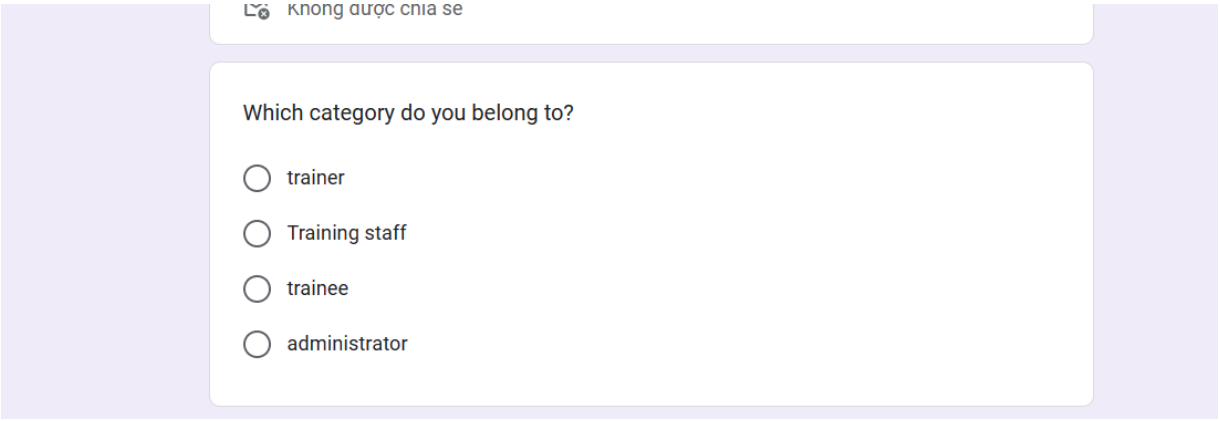
Suggestions for enhancing functionality and user interface

Best practices for implementing effective training programs

The collected data will be analyzed to inform decision-making processes and guide improvements to the FPT Training Web Application. The goal is to enhance user engagement, satisfaction, and the overall effectiveness of training programs delivered through the platform.

Link survey:

https://docs.google.com/forms/d/e/1FAIpQLSftn5EkW38eCNMaRFyMR2t8Xp8hk34q5dTuvzkNHgFSEwvfDw/viewform?usp=sharing



Survey 1

To what extent do you perceive our system? (1 - Disappointment, 5 - Excellence)

- ◯ 1
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5

Survey 2

How do you rate the flexibility and ease of use of the system? (1- Difficult to use, 5- easy to use)

- ◯ 1
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5

Survey 3

Considering the technologies employed, how adequate and suitable are they in meeting the application's objectives? (1 - Inadequate, 5 - Perfectly Suitable)

- ◯ 1
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5

Survey 4

How effectively does the system address potential risks and vulnerabilities? (1 - poor, 5 - perfect)

- 1
- 2
- 3
- 4
- 5

Survey 5

Does the system meet the needs of each trainer admintrator trainee and training staff?

- 1
- 2
- 3
- 4
- 5

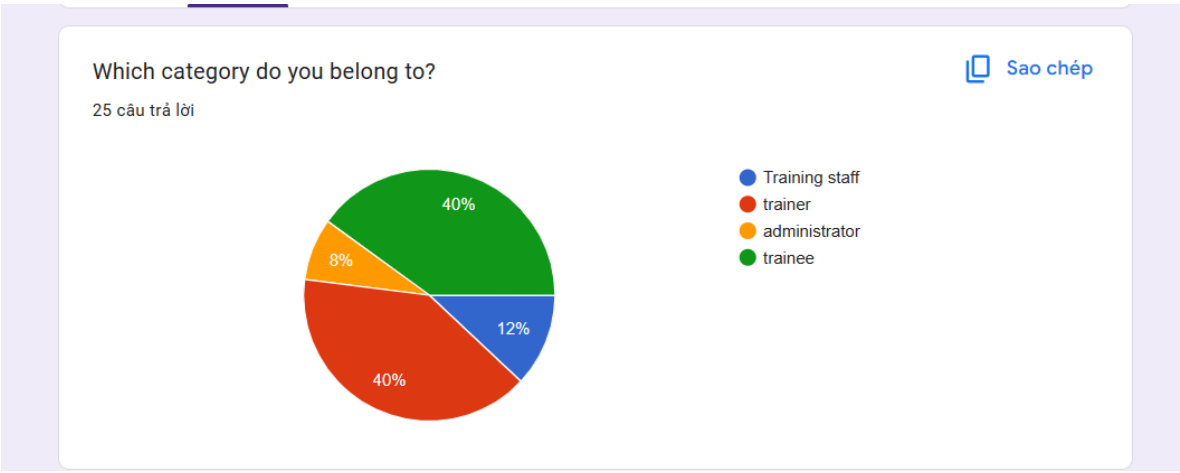
Survey 6

Please rate the comprehensiveness and clarity of the outlined development strategy. (1 - Not clear, 5 - Very clear and comprehensive)

- 1
- 2
- 3
- 4
- 5

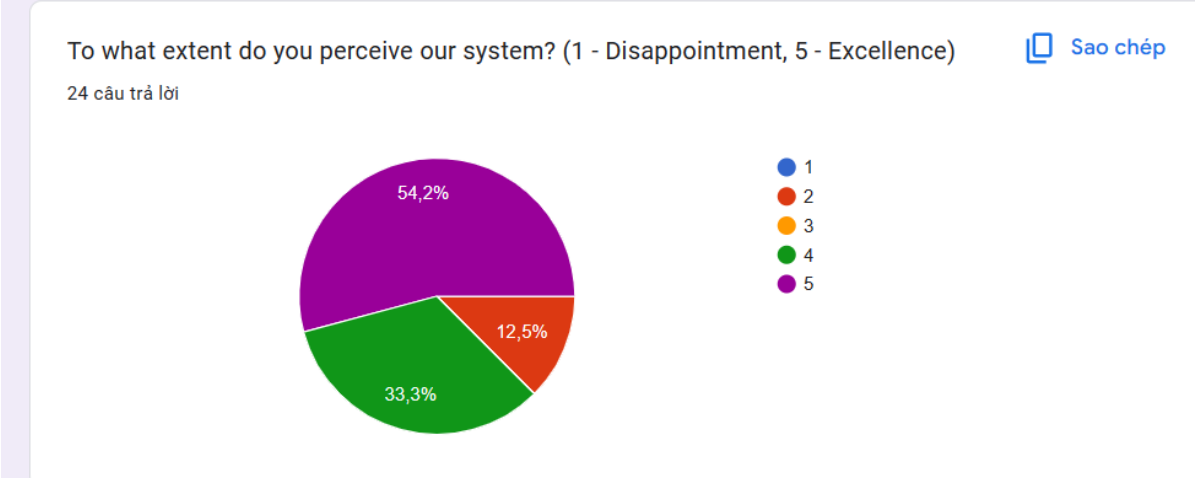
Survey 7

2. The result of survey



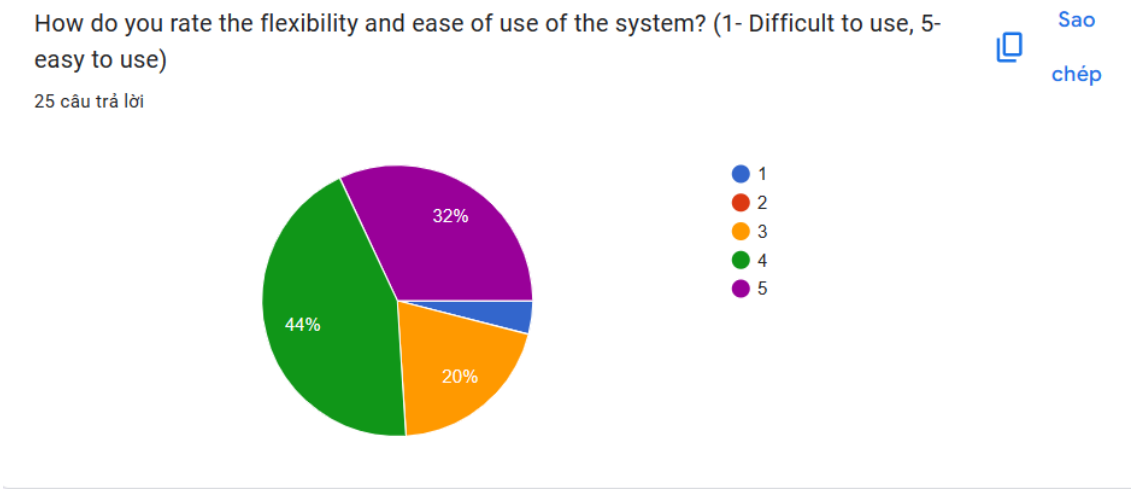
result 1

According to the above survey results, the number of people who are administrators is very small. In fact, a system only has 1 to 2 administrators. The training staff is only 3 people, accounting for a small portion. The rest are mostly trainees and trainers participating in the interview. Because these two objects account for the majority of system usage.



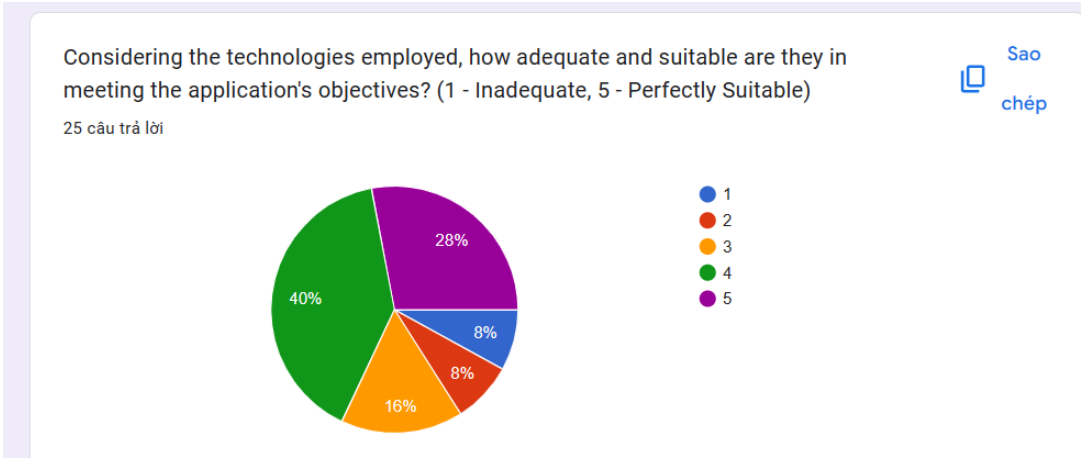
result 2

The system was rated excellent by many participants with up to 54.2%. On the contrary, only 12.5% rated the system as poor, the rest rated the system as being close to excellent. Maybe the system gets positive reviews.



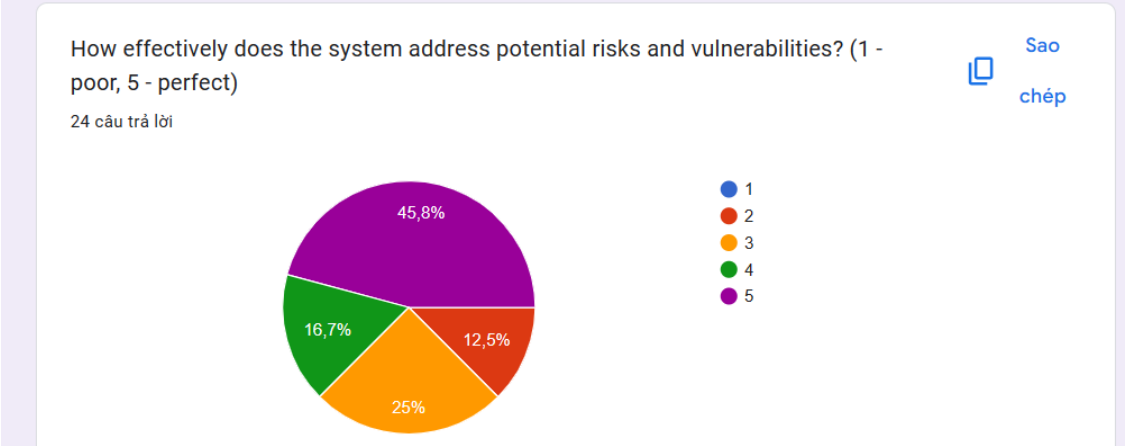
result 3

Through the assessment of the survey, from milestone 4 to 5 is a positive evaluation milestone, the system shows that many choose milestones 4 and 5. It can be seen that our system has developed with very high flexibility. 20% of participants rated the system as average. and a small portion of participants rated it at the most negative level.



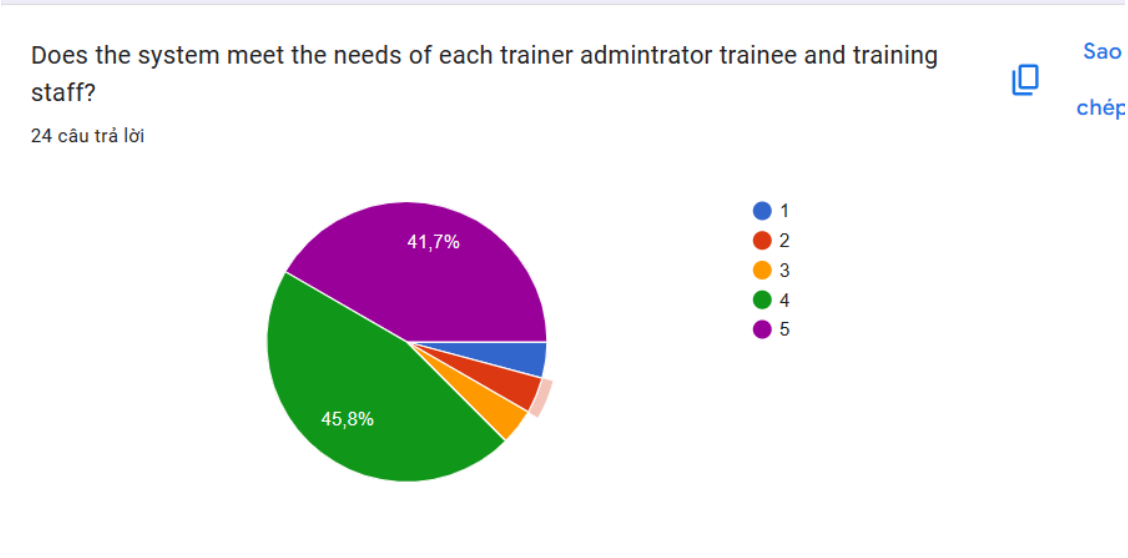
result 4

2 milestones 4 and 5 show that many people choose it, with a total of 68% of participants showing that the technologies are used at an appropriate level. The remaining milestones show a breakdown in their assessments. They seem to have little understanding of the technologies used in the system.



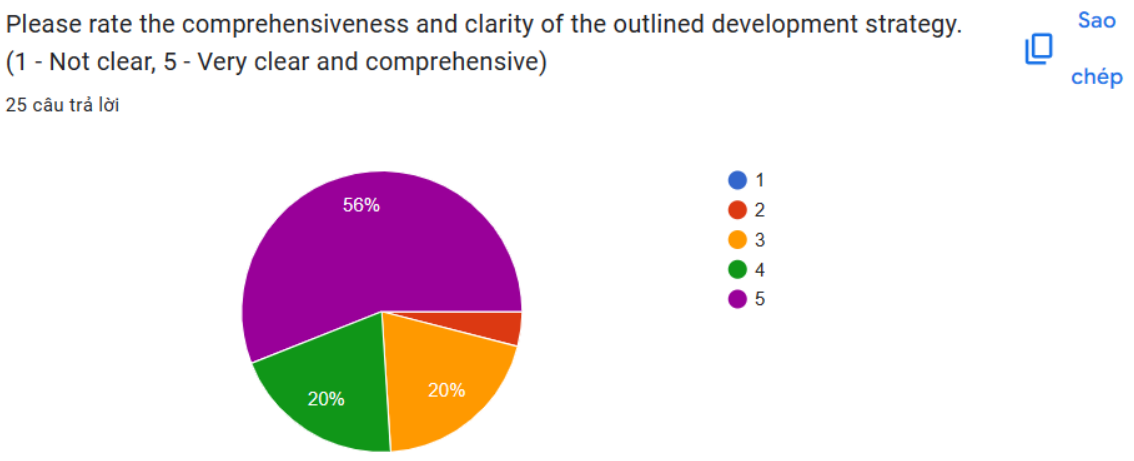
result 5

Up to 45.8% of participants felt that the system for addressing potential risks and vulnerabilities was perfect. Up to 25% of people also think that the system can only solve problems at a normal level. And a small number of participants found it less effective.



result 6

At milestones 4 and 5, many participants evaluate that the system is suitable and meets the needs of system objects. In both milestones, up to 87.5% of people accounted for the majority of participants.



result 7

Up to 56% of participants showed that the comprehensive level of development strategy was clearly and completely outlined. 20% of participants agreed that the level was only clear and complete. The remaining 20% think it's just average. Only a few think it is unclear.

3. Feedback Review

Based on the survey I conducted using the 7 comprehensive questions outlined earlier, overall feedback from users indicates moderate to high satisfaction, with most responding focuses on ratings from level 3 to level 5. These ratings, with level 3 being the lowest and level 5 being the highest, rarely include a level 2. It reflects generally positive sentiment on the aspects variety of business applications.

- I am pleased to note that the problem definition statement received moderate to high ratings, mostly hovering around level 4, indicating a clear presentation of the core challenges the application solves. decided. This indicates a positive reception of the application's suitability for the identified needs.
- Regarding the effectiveness of the proposed solution in solving the identified problem, responses tended to be at level 4, suggesting a moderate to high level of effectiveness. Although the application appears to provide solutions with a good reputation, further improvements may be needed to achieve the optimal solution level.
- Users expressed satisfaction, mainly at level 4, with the clarity and comprehensiveness of the development strategy, indicating a well-defined and well-communicated plan. However, there may be potential areas for improvement to raise it to a more exemplary level.
- Alignment between the app's design and functionality and user experience received mostly positive feedback, around a level of 4, demonstrating good alignment, although some aspects may require fine-tuning for optimal usability and experience.
- Although users found the interface to be quite intuitive and user-friendly (rated at 4 to 5), there is potential for further improvements to make the interface more user-centric and easier to navigate.
- Overall, survey responses expressed a generally positive view of the business application, acknowledging its strengths while highlighting areas for improvement. The ratings, aggregated between levels 3 and 5,

provide valuable insights that will guide me in refining and optimizing the app for a more robust and user-centric solution.

C. P5 Develop a functional business application based on a specified business problem.

1. Administrator's Role

1.1 Login by Administrator's account

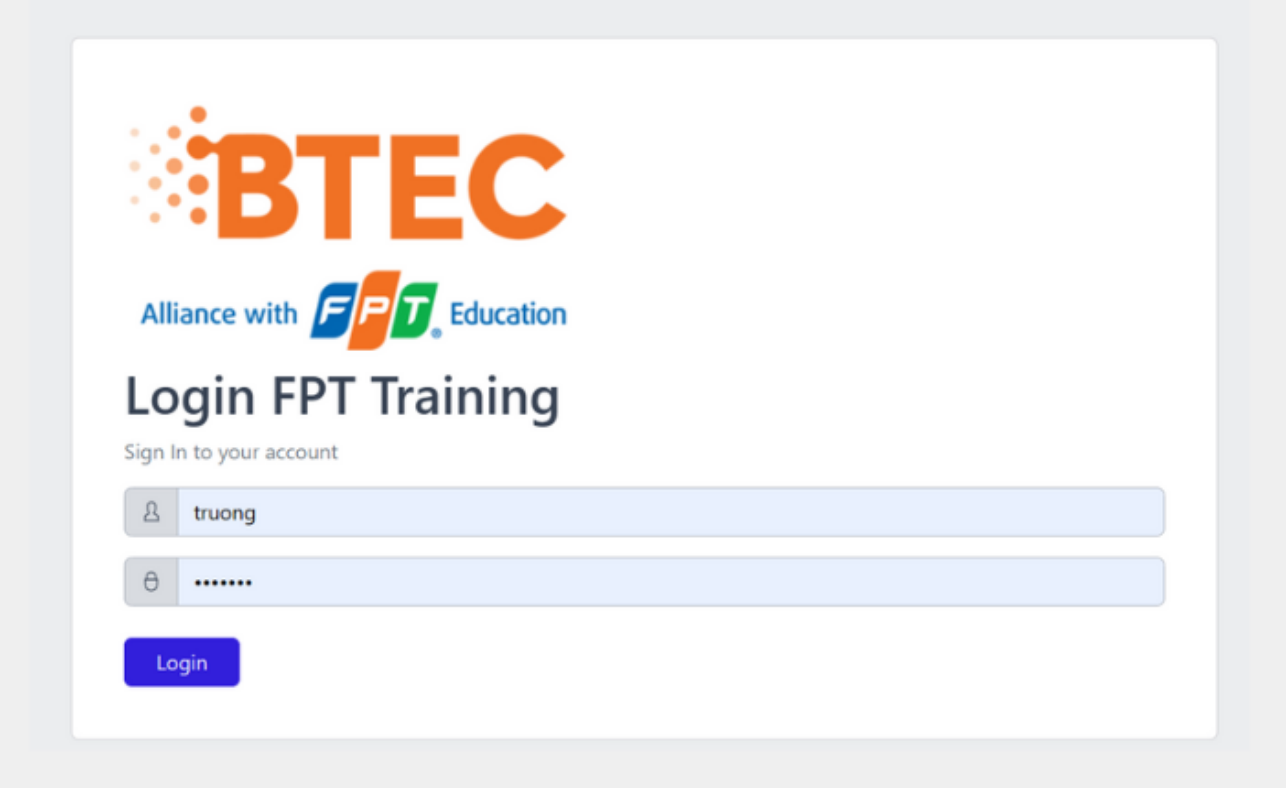
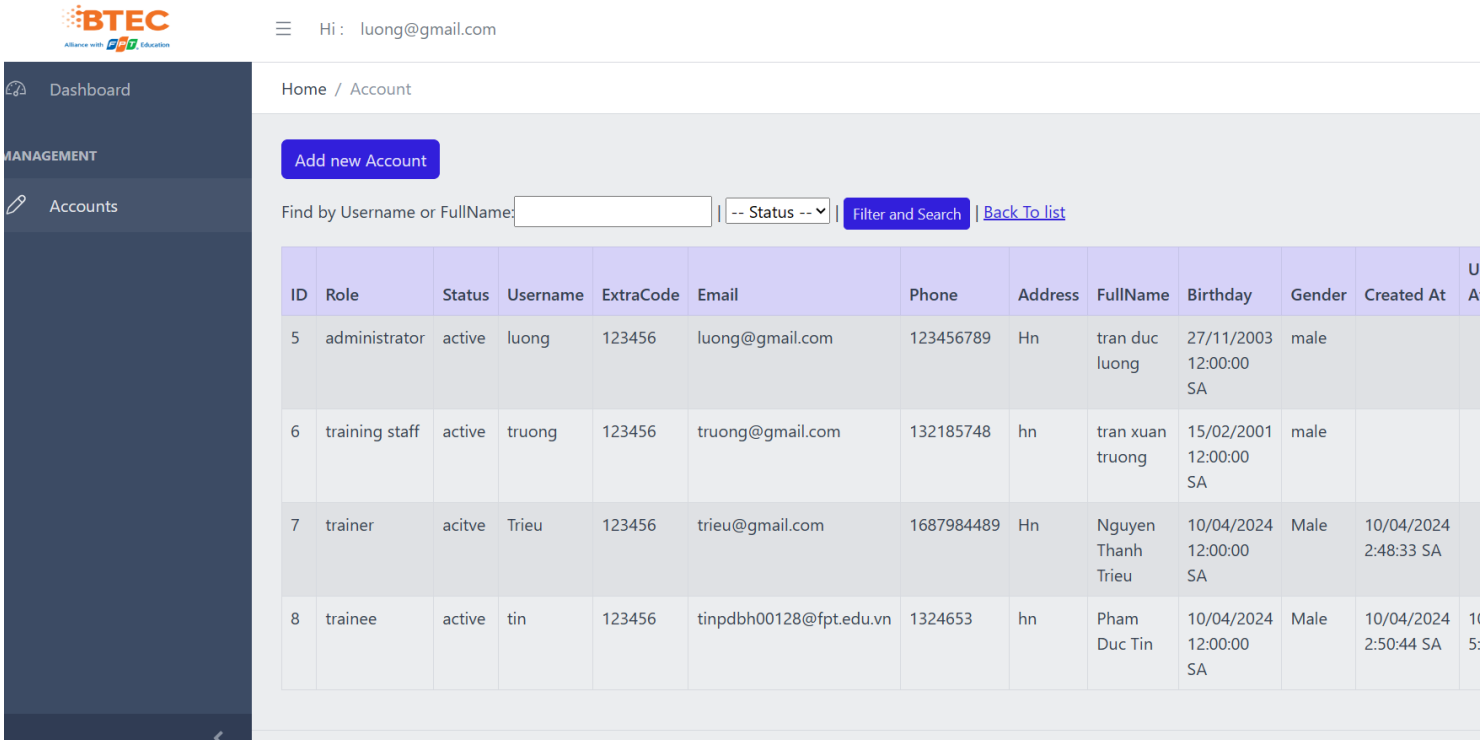


Figure 1: Login

1.2 User account management



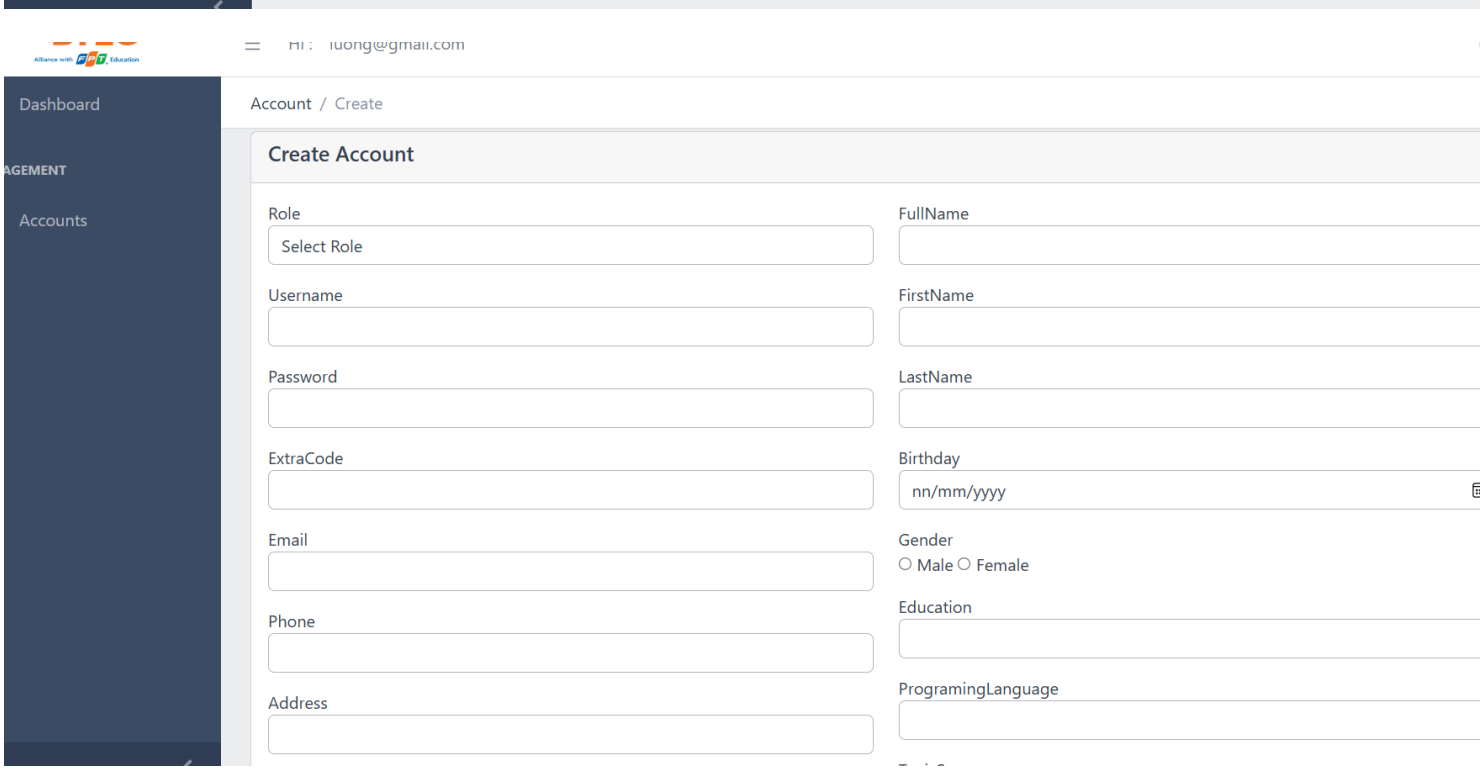


Figure 2:

We will log in to the admin, training staff and trainne accounts in this login interface. When logged in successfully, you will be redirected to the "Account" management page.

2. Training staff's role

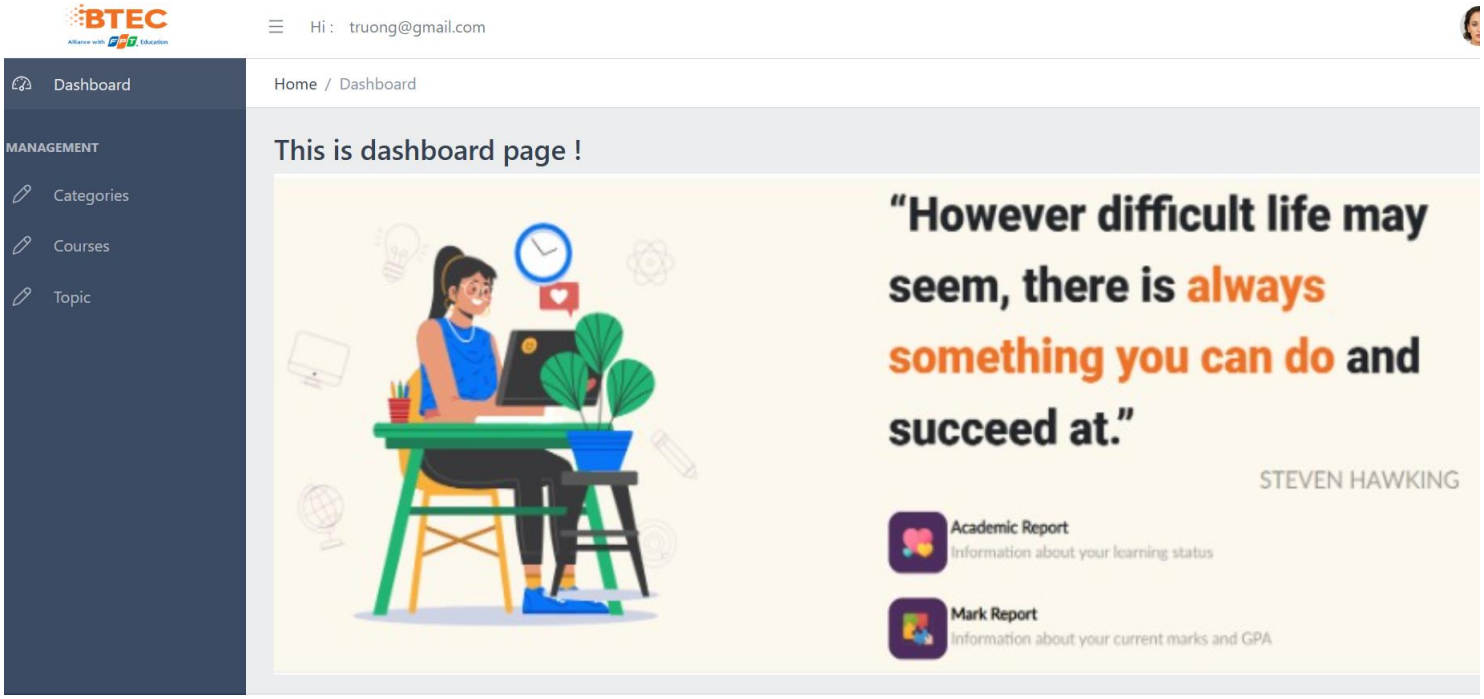
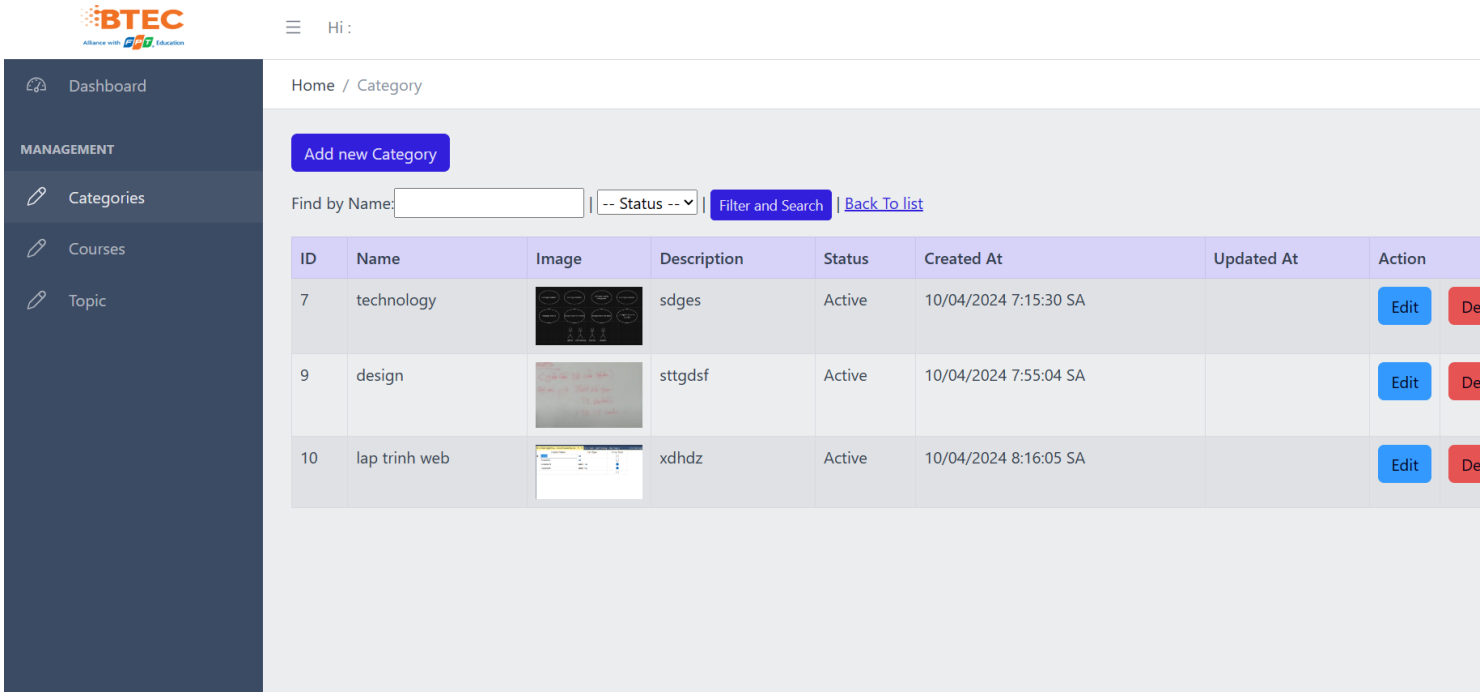


Figure 3

Log in to your account with the login name staff and when the information matches, you will

be redirected to the Menu page including Category, Course, Topic,...

2.2 User category management



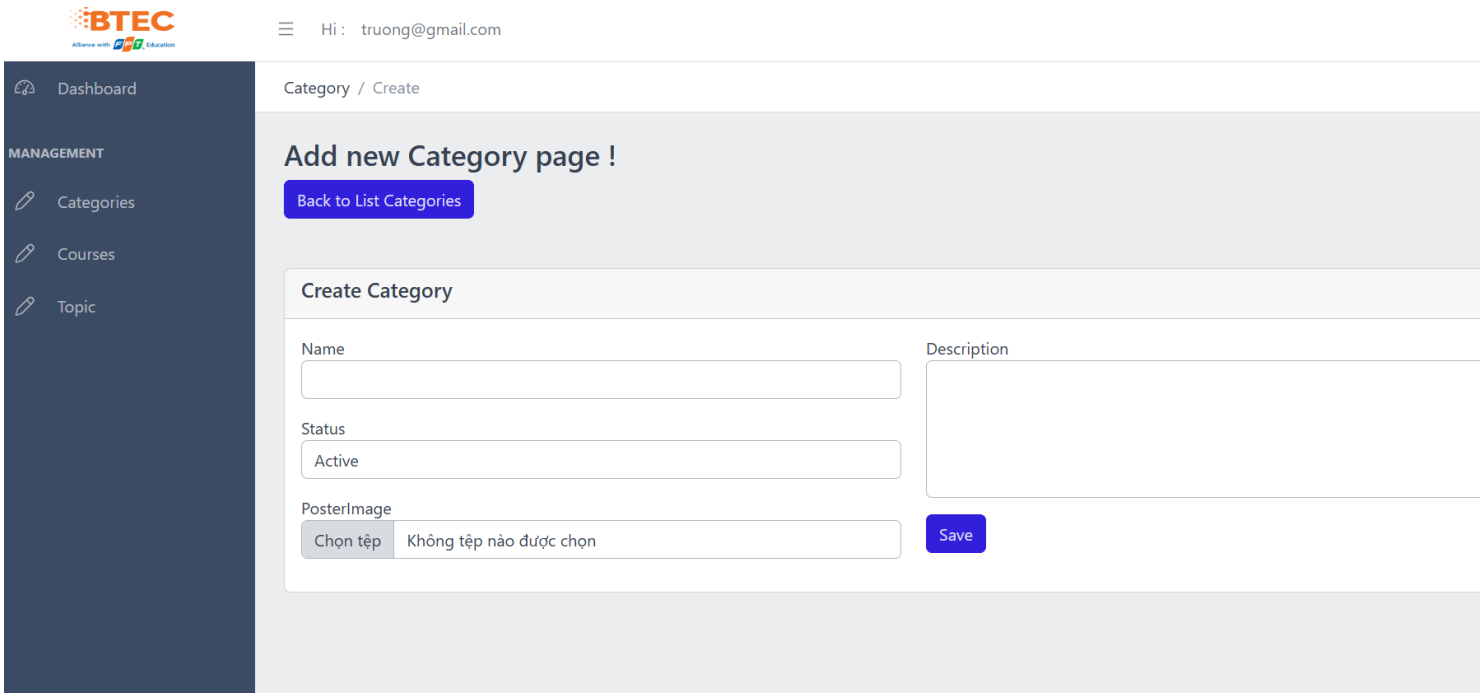


Figure 4

Training staff can update and delete student accounts and can manage course categories such as searching, adding, updating and deleting course categories.

2.3 User Course management

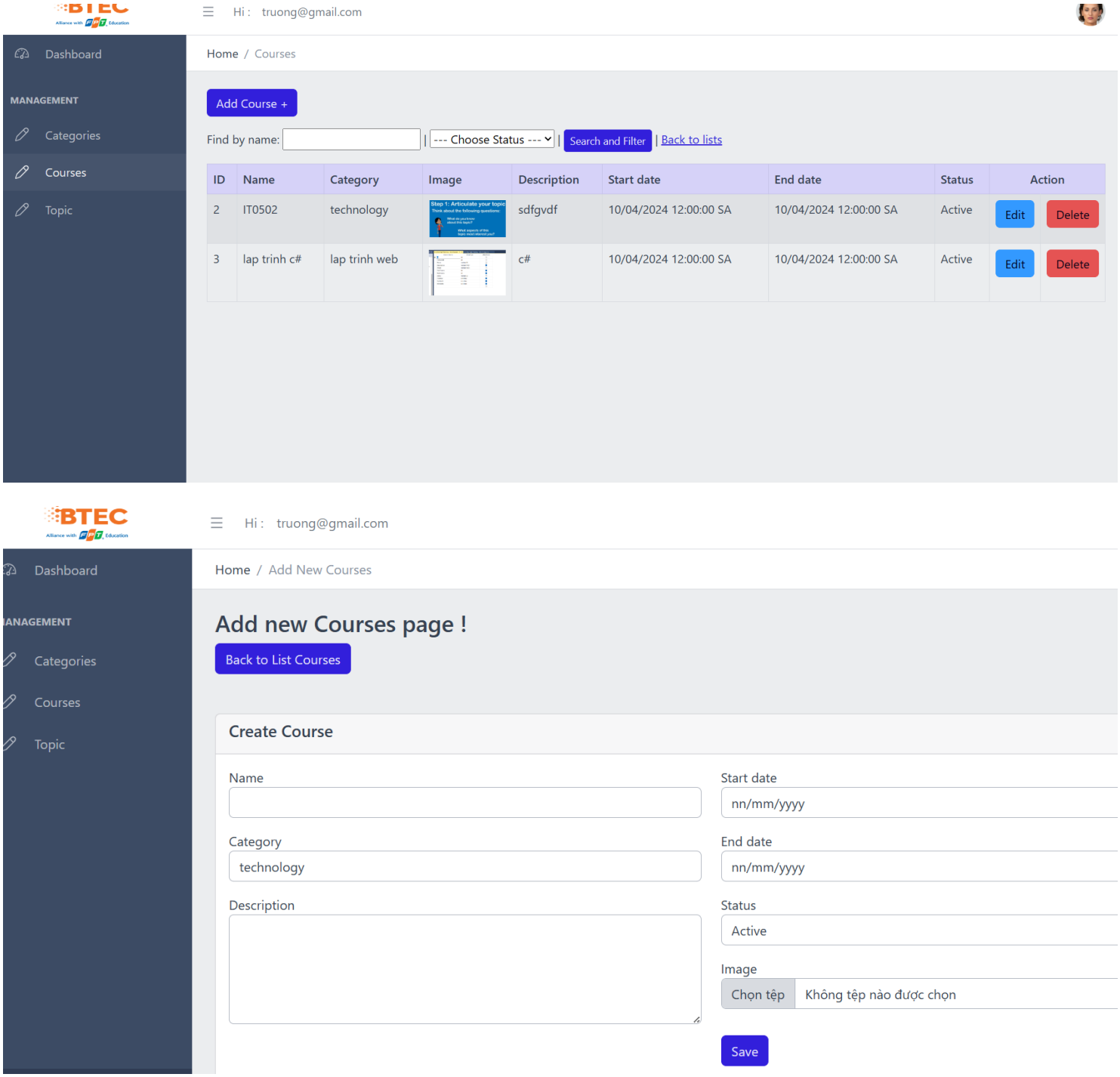


Figure 5

Training staff can manage courses such as searching, adding, updating and deleting courses. Course includes course name and description.

2.4 User Topic management

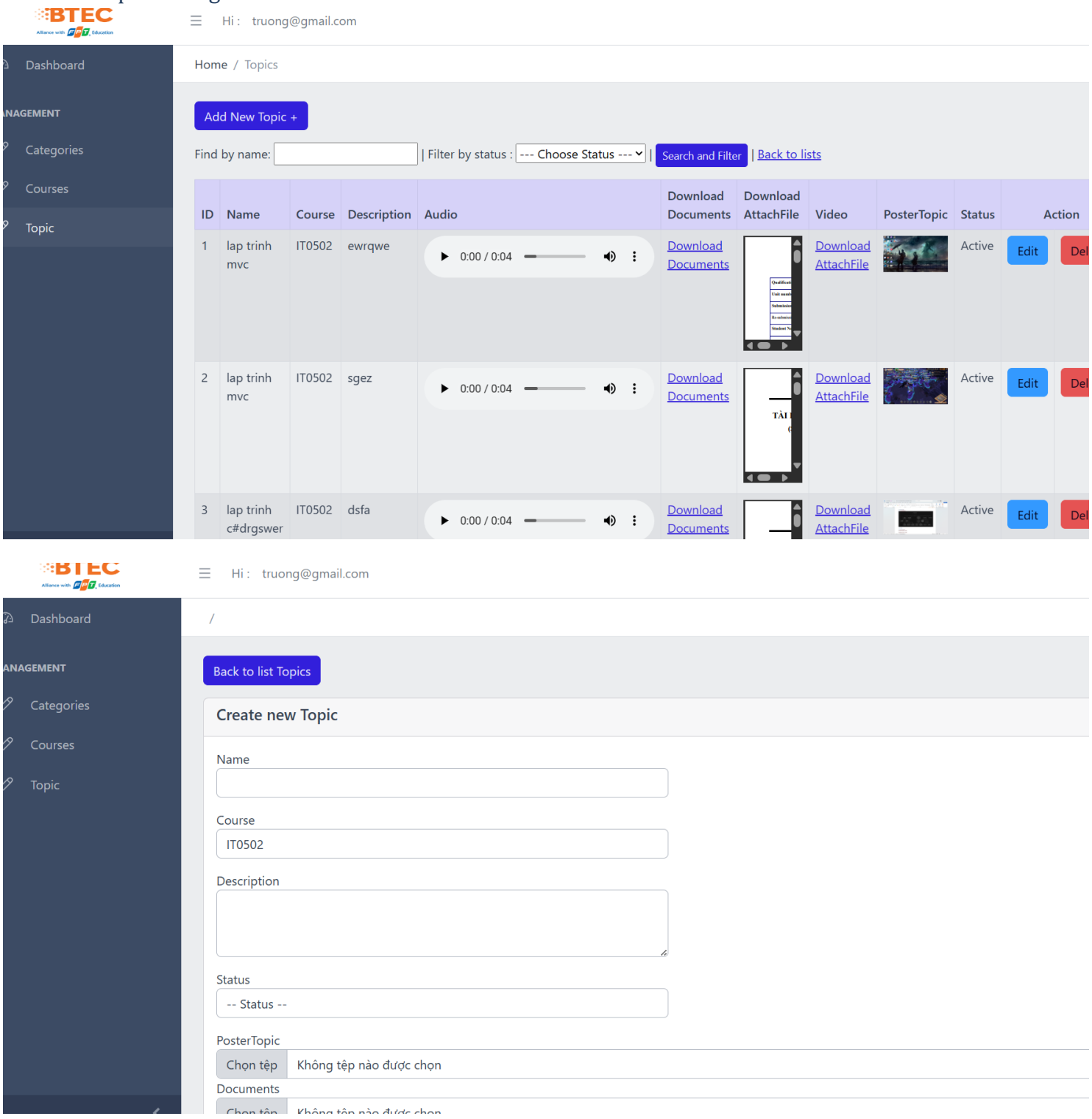


Figure 6

Topics such as topic name and topic description can be added to a course, courses can be added to a category.

3. Trainer Role

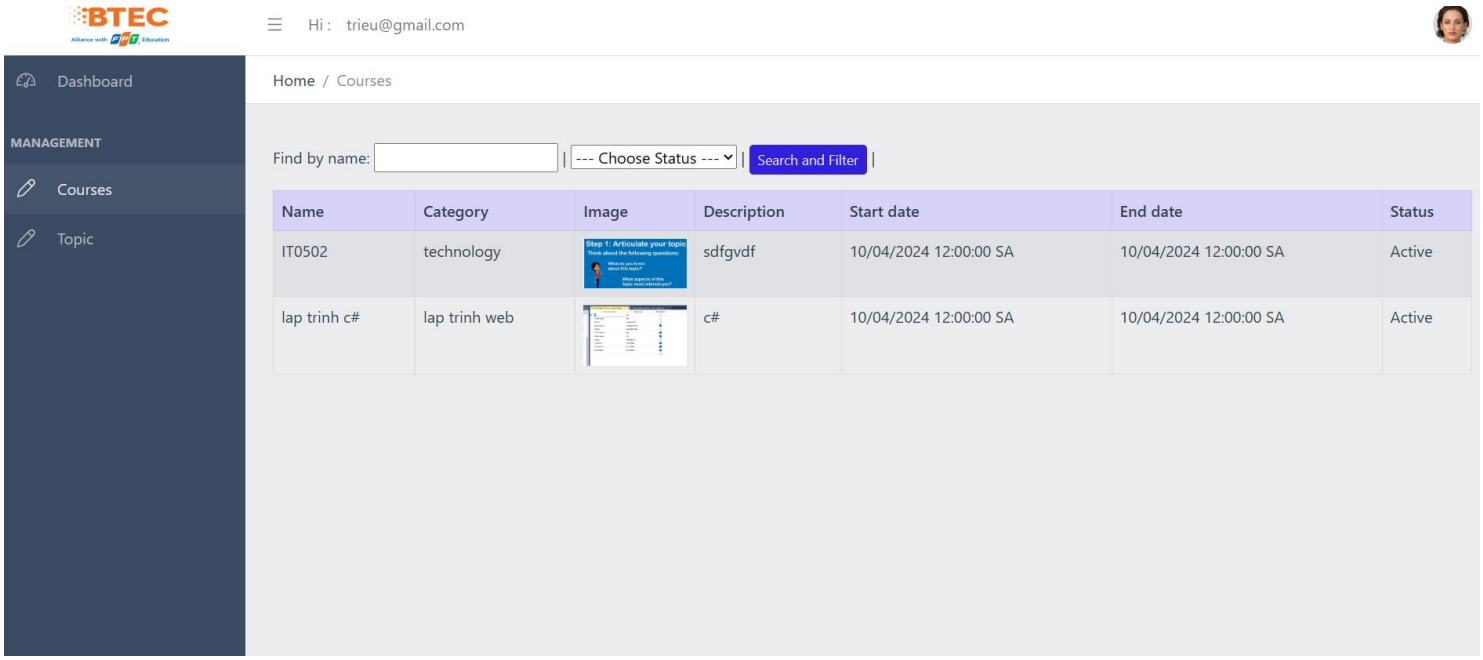


Figure 7

Trainers can view the courses they are assigned.

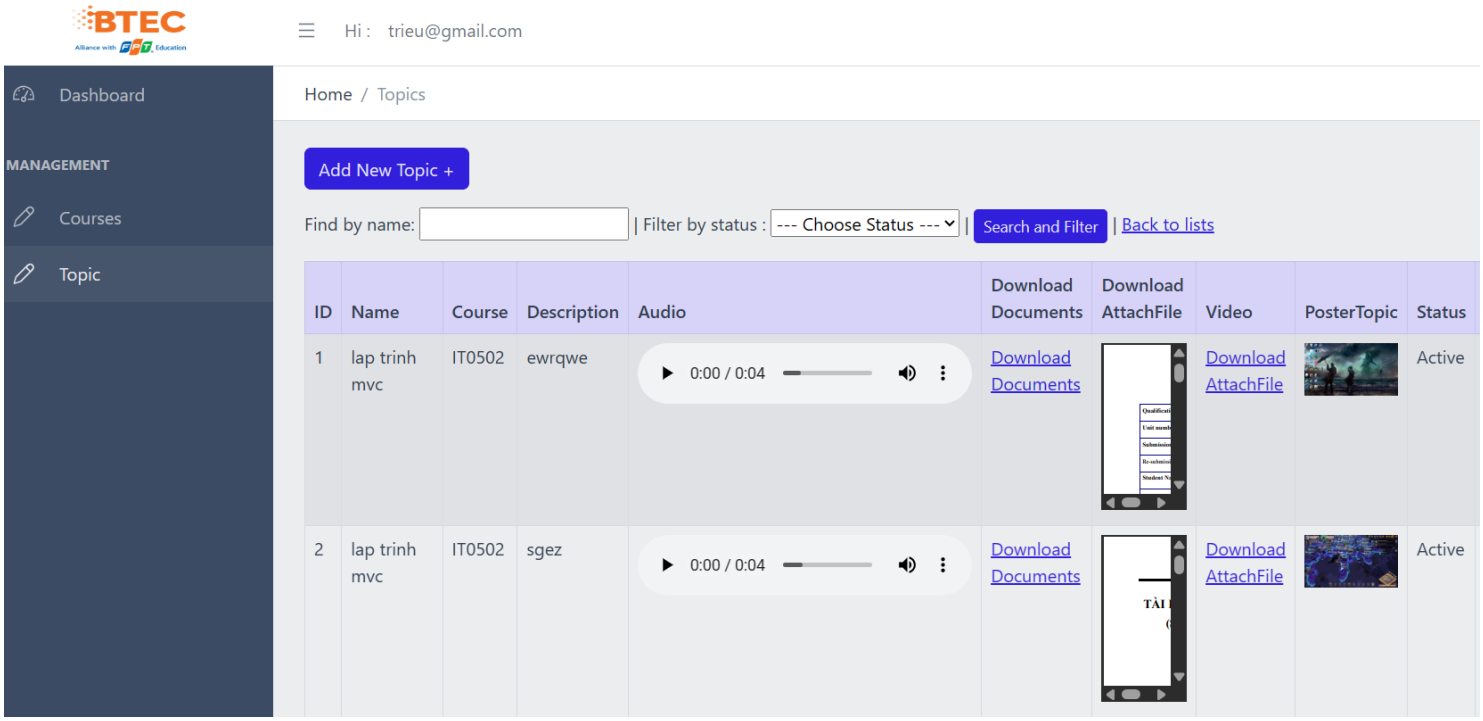


Figure 8

Trainers can view the topics they are assigned to.

4. Trainee

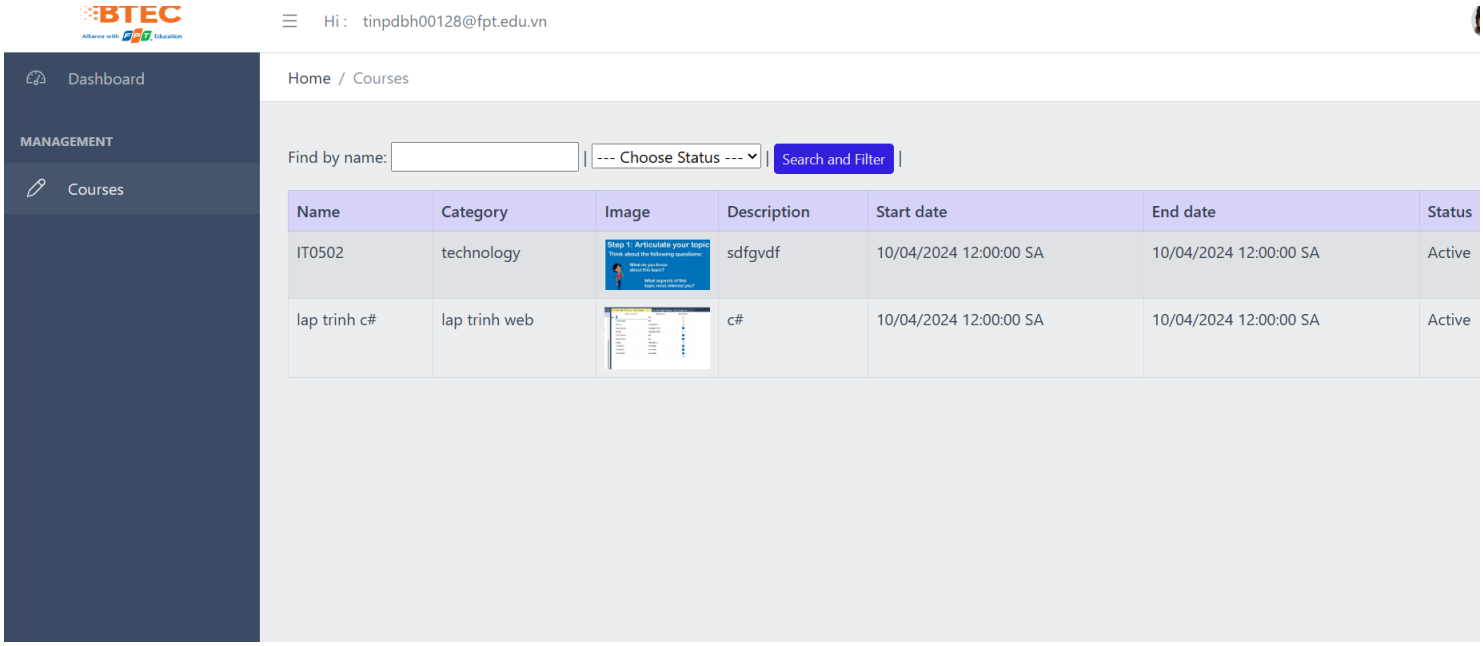


Figure 9

Trainees can only view the courses they are assigned to.

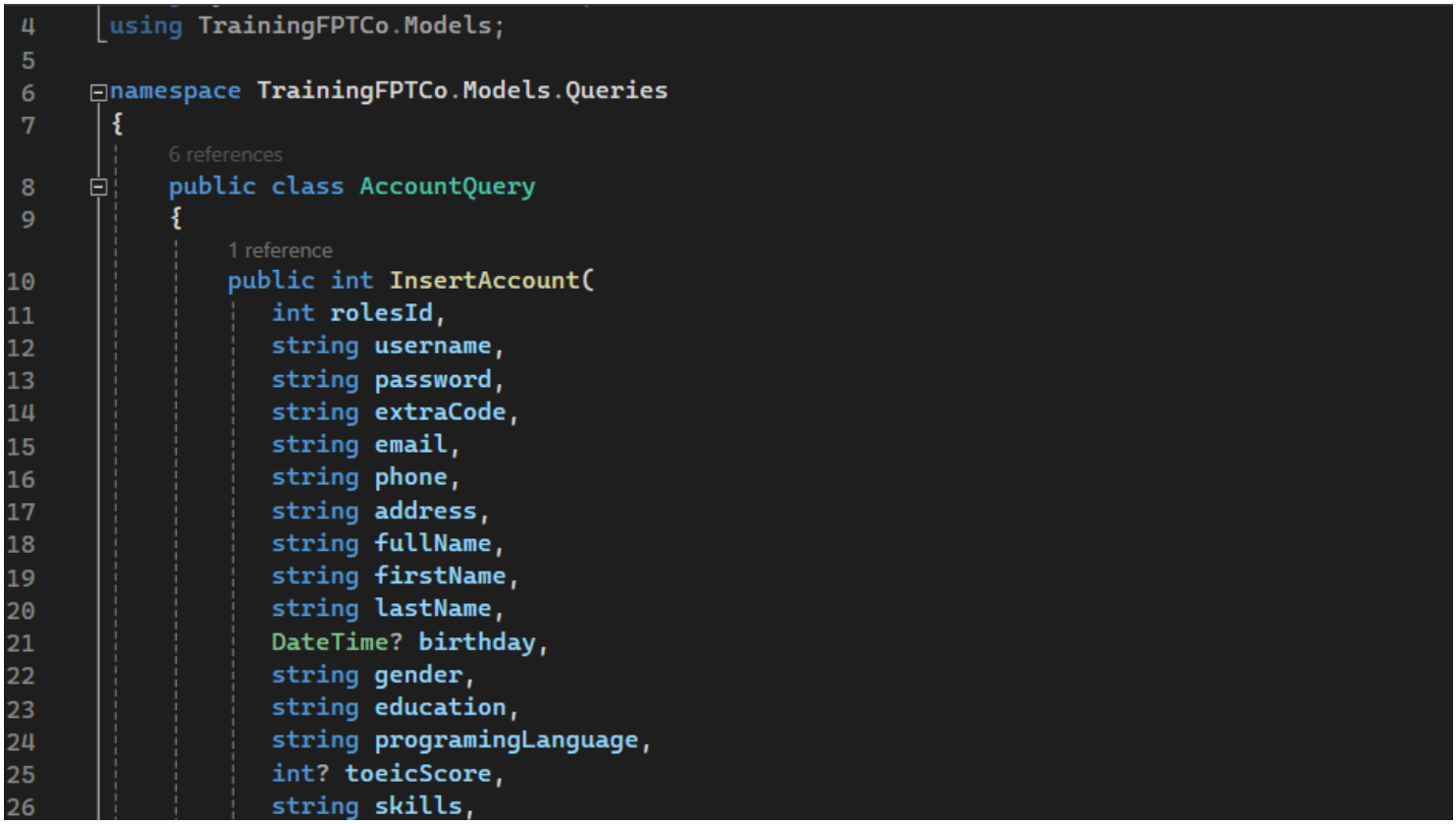
5. Code

5.1 Account

```
using Microsoft.AspNetCore.Mvc;
using TrainingFPTCo.DataDBContext;
using TrainingFPTCo.Models;
using TrainingFPTCo.Models.Queries;
using TrainingFPTCo.Helpers;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc.Rendering;

namespace TrainingFPTCo.Controllers
{
    4 references
    public class AccountController : Controller
    {
        private readonly TrainingDbContext _dbContext;

        0 references
        public AccountController(TrainingDbContext dbContext)
        {
            _dbContext = dbContext;
        }

        [HttpGet]
```

Figure 10

The "accountcontroller" class is often used to manage and control account-related behavior in software applications. It takes care of tasks like user authentication, account creation, and account deletion. The purpose of the "accountcontroller" class is to ensure that appropriate actions are taken when interacting with user accounts.



```
4   using TrainingFPTCo.Models;
5
6   namespace TrainingFPTCo.Models.Queries
7   {
        6 references
8       public class AccountQuery
9       {
            1 reference
10          public int InsertAccount(
11              int rolesId,
12              string username,
13              string password,
14              string extraCode,
15              string email,
16              string phone,
17              string address,
18              string fullName,
19              string firstName,
20              string lastName,
21              DateTime? birthday,
22              string gender,
23              string education,
24              string programingLanguage,
25              int? toeicScore,
26              string skills,
```

Figure 11

The "accountquery" class is often used to query and retrieve information about accounts. It helps perform tasks such as searching for accounts, retrieving account information, and performing account-related queries.

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace TrainingFPTCo.Models
{

    public class AccountViewModel
    {

        public List<AccountDetail> AccountDetailList { get; set; }
    }


    public class AccountDetail
    {
        8 references
        public int Id { get; set; }

        10 references
        public int RolesId { get; set; }

        7 references
        public Role Role { get; set; }

        [Required(ErrorMessage = "The Username field is required.")]
        9 references
```

Figure 12

The "accountviewmodel" class is typically used to define the data model and business logic for account-related features in the application. It helps create a set of data and corresponding processing methods to display information and interact with accounts in the user interface.

5.2 Cateogory

```
using Microsoft.AspNetCore.Mvc;
using TrainingFPTCo.DataDBContext;
using TrainingFPTCo.Models;
using TrainingFPTCo.Models.Queries;
using TrainingFPTCo.Helpers;

namespace TrainingFPTCo.Controllers
{
    4 references
    public class CategoryController : Controller
    {
        private readonly TrainingDbContext _dbContext;
        0 references
        public CategoryController(TrainingDbContext dbContext)
        {
            _dbContext = dbContext;
        }

        [HttpGet]
        3 references
        public IActionResult Index(string SearchString, string FilterStatus)
        {
            CategoryViewModel categoryModel = new CategoryViewModel();
            categoryModel.CategoryDetailList = new List<CategoryDetail>();
            var dataCategory = new CategoryQuery().GetAllCategories(SearchString,
              FilterStatus);
```

Figure 13

The "CategoryController" class is often used to manage and control behavior related to categories in software applications. It takes care of tasks like creating, editing, and deleting categories. The purpose of the "CategoryController" class is to ensure that appropriate actions are taken when interacting with categories in the application.

```
using Microsoft.Data.SqlClient;

namespace TrainingFPTCo.Models.Queries
{
    10 references
    public class CategoryQuery
    {
        1 reference
        public string GetCategoryNameById(int id)
        {
            string categoryName = null;
            using (SqlConnection connection = Database.GetSqlConnection())
            {
                string sqlQuery = "SELECT [Name] FROM [Categories] WHERE
                connection.Open();
                SqlCommand command = new SqlCommand(sqlQuery, connection);
                command.Parameters.AddWithValue("@id", id);
                using (SqlDataReader reader = command.ExecuteReader())
                {
                    if (reader.Read())
                    {
                        categoryName = reader["Name"].ToString();
                    }
                }
                connection.Close();
            }
```

class System.Str
Represents text as a

Figure 14

The "CategoryQuery" class is commonly used to query and retrieve information about categories. It helps perform tasks such as searching categories, retrieving category information, and performing category-related queries.

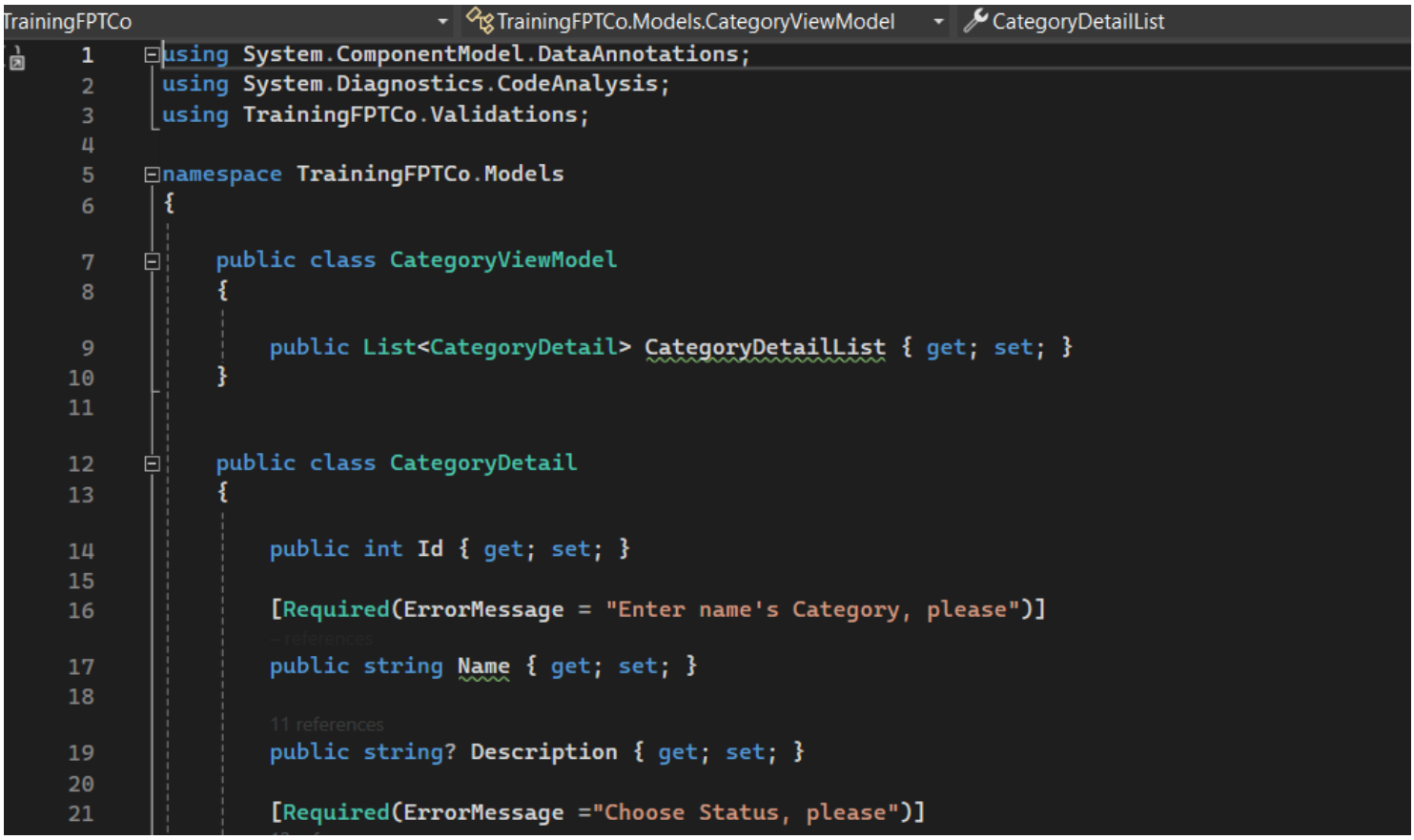


Figure 15

The "CategoryViewModel" class is typically used to define the data model and business logic for category-related features in the application. It helps create a set of data and corresponding processing methods to display information and interact with categories in the user interface.

5.3 Course

```
using TrainingFPTCo.Models;
using TrainingFPTCo.Models.Queries;

namespace TrainingFPTCo.Controllers
{
    3 references
    public class CoursesController : Controller
    {
        [HttpGet]
        3 references
        public IActionResult Index(string SearchString, string Status)
        {
            CourseViewModel course = new CourseViewModel();
            course.CourseDetailsList = new List<CourseDetail>();
            var dataCourses = new CourseQuery().GetAllDataCourses(SearchString, Status);
            var categoryQuery = new CategoryQuery();
            foreach (var data in dataCourses)
            {
                string categoryName = categoryQuery.GetCategoryNameById(data.CategoryId);
                course.CourseDetailsList.Add(new CourseDetail
                {
                    Id = data.Id,
                    Name = data.Name,
                    CategoryId = data.CategoryId,
                    Description = data.Description,
                    Status = data.Status,
```

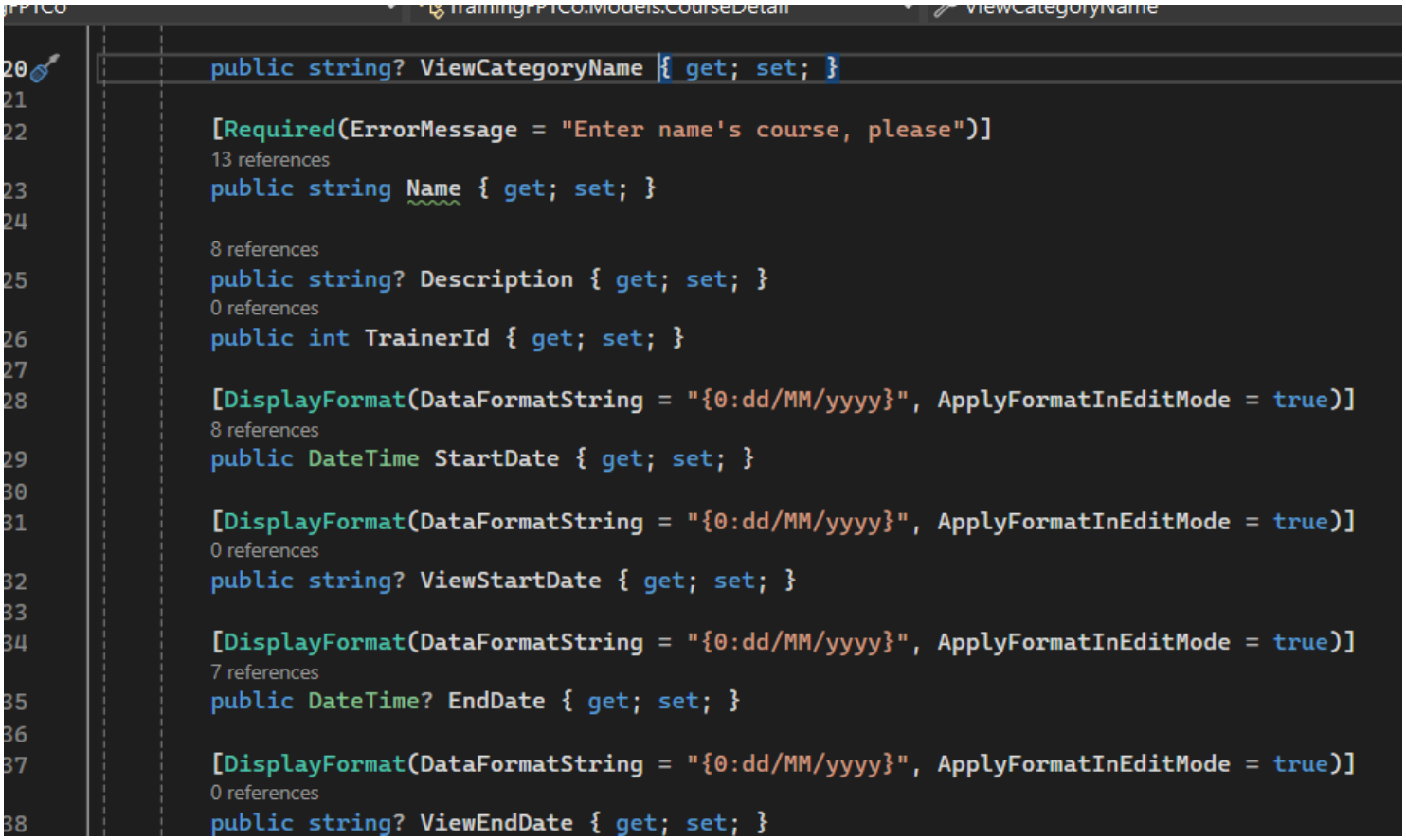
Figure 16

The "CoursesController" class is typically used to manage and control behavior related to course-related features in a software application. It takes care of tasks such as retrieving course data, updating course information, and deleting courses. The purpose of the "CoursesController" class is to ensure that appropriate actions are taken when interacting with courses in the application.

```
        string description,
        string image,
        int categoryId,
        DateTime startDate,
        DateTime? endDate,
        string status,
        int id
    )
    {
        bool checkUpdate = false;
        using (SqlConnection connection = Database.GetSqlConnection())
        {
            string sqlUpdate = "UPDATE [Courses] SET [Name] = @name, [CategoryId] =
              @Categories, [Description] = @description, [Image] = @image, [StartDate] =
              @startDate, [EndDate] = @endDate, [Status] = @status, [UpdatedAt] =
              @updatedAt WHERE [Id] = @id AND [DeletedAt] IS NULL";
            connection.Open();
            SqlCommand cmd = new SqlCommand(sqlUpdate, connection);
            cmd.Parameters.AddWithValue("@name", name ?? DBNull.Value.ToString());
            cmd.Parameters.AddWithValue("@Categories", categoryId);
            cmd.Parameters.AddWithValue("@description", description ??
              DBNull.Value.ToString());
            cmd.Parameters.AddWithValue("@image", image ?? DBNull.Value.ToString());
            cmd.Parameters.AddWithValue("@startDate", startDate);
            cmd.Parameters.AddWithValue("@endDate", endDate);
```

Figure 17

The "CourseQuery" class is commonly used to query and retrieve course information. It helps perform tasks such as searching for courses, retrieving course information, and performing course-related queries.



```
20      public string? ViewCategoryName { get; set; }
21
22      [Required(ErrorMessage = "Enter name's course, please")]
        13 references
23      public string Name { get; set; }
24
        8 references
25      public string? Description { get; set; }
        0 references
26      public int TrainerId { get; set; }
27
28      [DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]
        8 references
29      public DateTime StartDate { get; set; }
30
31      [DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]
        0 references
32      public string? ViewStartDate { get; set; }
33
34      [DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]
        7 references
35      public DateTime? EndDate { get; set; }
36
37      [DisplayFormat(DataFormatString = "{0:dd/MM/yyyy}", ApplyFormatInEditMode = true)]
        0 references
38      public string? ViewEndDate { get; set; }
```

Figure 18

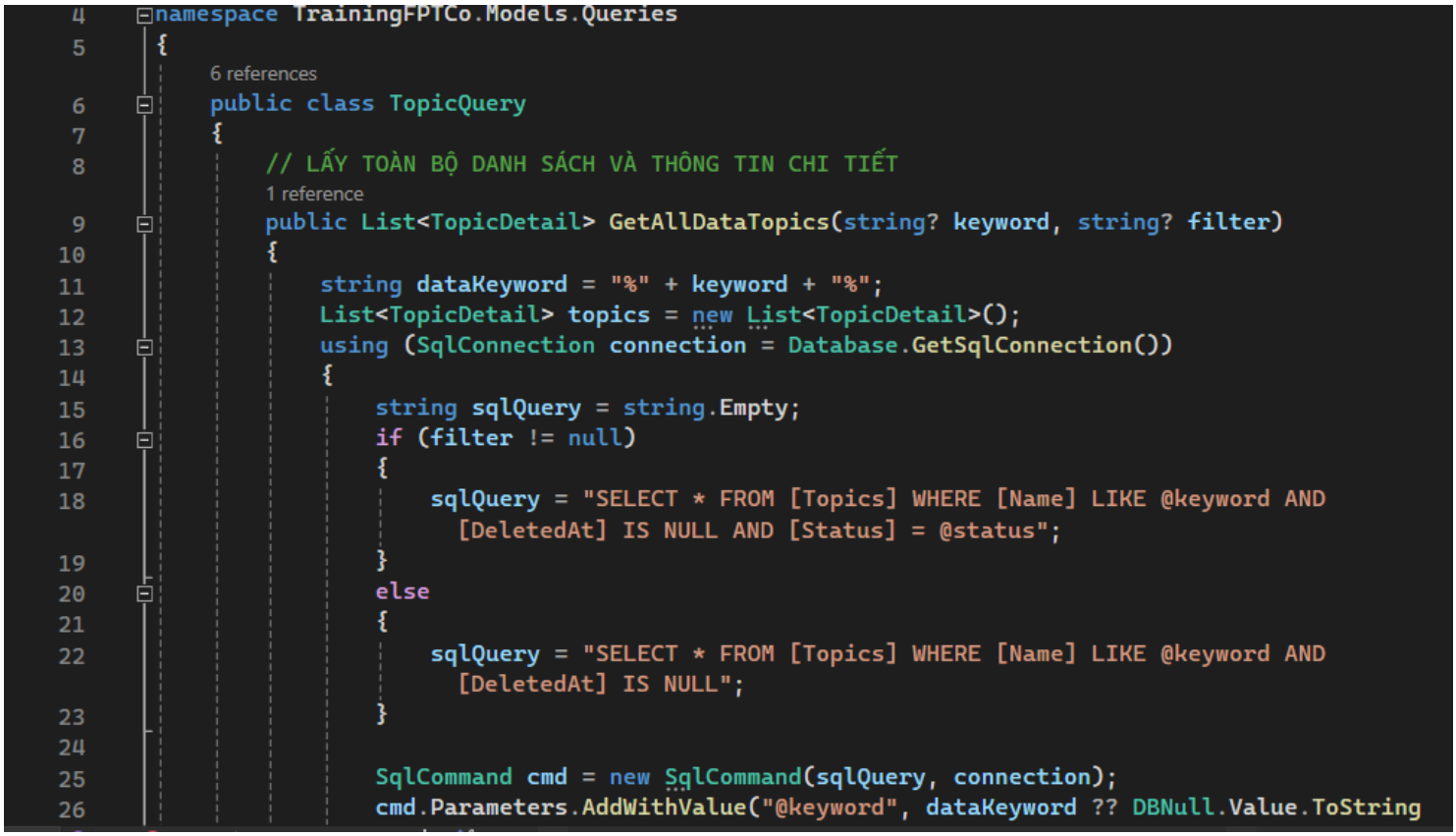
The "CourseViewModel" class is typically used to define the data model and business logic for course-related features in the application. It helps create a set of data and corresponding processing methods to display course information and interact with the course in the user interface.

5.4 Topic

```
10  {
        3 references
11      public class TopicsController : Controller
12      {
13          // INDEX
14          [HttpGet]
            3 references
15          public IActionResult Index(string SearchString, string Status)
16          {
17
18              TopicsViewModel topic = new TopicsViewModel();
19              topic.TopicDetailList = new List<TopicDetail>();
20              var dataTopics = new TopicQuery().GetAllDataTopics(SearchString, Status);
21              var courseQuery = new CourseQuery();
22              foreach (var data in dataTopics)
23              {
24                  string courseName = courseQuery.GetCourseNameById(data.CourseId);
25                  topic.TopicDetailList.Add(new TopicDetail
26                  {
27                      Id = data.Id,
28                      Name = data.Name,
29                      CourseId = data.CourseId,
30                      Description = data.Description,
31                      ViewDocuments = data.ViewDocuments,
32                      ViewAttachFile = data.ViewAttachFile,
33                      ViewPosterTopic = data.ViewPosterTopic,
                        Status = data.Status,
```

Figure 19

The "TopicController" class is often used to manage and control topic-related behavior in software applications. It takes care of tasks like creating, editing, and deleting themes. The purpose of the "TopicController" class is to ensure that appropriate actions are taken when interacting with topics in the application.



```
namespace TrainingFPTCo.Models.Queries
{
    6 references
    public class TopicQuery
    {
        // LẤY TOÀN BỘ DANH SÁCH VÀ THÔNG TIN CHI TIẾT
        1 reference
        public List<TopicDetail> GetAllDataTopics(string? keyword, string? filter)
        {
            string dataKeyword = "%" + keyword + "%";
            List<TopicDetail> topics = new List<TopicDetail>();
            using (SqlConnection connection = Database.GetSqlConnection())
            {
                string sqlQuery = string.Empty;
                if (filter != null)
                {
                    sqlQuery = "SELECT * FROM [Topics] WHERE [Name] LIKE @keyword AND
                      [DeletedAt] IS NULL AND [Status] = @status";
                }
                else
                {
                    sqlQuery = "SELECT * FROM [Topics] WHERE [Name] LIKE @keyword AND
                      [DeletedAt] IS NULL";
                }

                SqlCommand cmd = new SqlCommand(sqlQuery, connection);
                cmd.Parameters.AddWithValue("@keyword", dataKeyword ?? DBNull.Value.ToString
```

Figure 20

The "TopicQuery" class is commonly used to query and retrieve information about a topic. It helps perform tasks such as searching for topics, retrieving topic information, and performing queries related to the topic.

```
using System.ComponentModel.DataAnnotations;
using TrainingFPTCo.Migrations;
using TrainingFPTCo.Validations;

namespace TrainingFPT.Models
{
    5 references
    public class TopicsViewModel
    {
        3 references
        public List<TopicDetail> TopicDetailList { get; set; }
    }

    22 references
    public class TopicDetail
    {
        [Key]
        10 references
        public int Id { get; set; }
        [Required(ErrorMessage = "Enter Topic's name, please")]
        17 references
        public string Name { get; set; }
        [Required(ErrorMessage = "Choose Course, please")]
        11 references
        public int CourseId { get; set; }
        11 references
```

Figure 21

The "TopicViewModel" class is typically used to define the data model and business logic for theme-related features in the application. It helps create a set of data and corresponding processing methods to display information and interact with the subject in the user interface.

D.(P6)Review the performance of your business application against the Problem Definition Statement and initial requirements.

1.Function requirements

1.1 Administrator

Login

Management user account

- Create account
- Edit account
- Delete account

Logout

1.2 Training staff

Login

Management course

- Add course
- Edit course
- delete course

Management categories

- Add categories
- Edit categories
- Delete categories

Management topic

- Add topic
- Edit topic
- Delete topic

Assign trainers to topic

Assign trainee to course

Logout

1.3 Trainer

Login

View and update their profile

View course with assign topics

1.4 Trainee

Login

View their profile

View course

2. Test plan

ID	Test case name	Description	Actor
TC1	Login	Login successful	Administrator, training staff, trainer, trainee

TC2	Login	Login failed	Administrator, training staff, trainer, trainee
TC3	Account manager	Display list account	administrator
TC4	Account manager	Add account	administrator
TC5	Account manager	Edit account	administrator

TC6	Account manager	Delete account	administrator
TC7	Account manager	Search account	administrator
TC8	Category manager	Add category	Training staff
TC9	Category manager	Edit category	Training staff
TC10	Category manager	Delete category	Training staff
TC11	Category manager	Search category	Training staff
TC12	Course manager	Add course	Training staff

TC13	Course manager	Edit course	Training staff
TC14	Course manager	Delete course	Training staff
TC15	Course manager	Search course	Training staff
TC16	Topic manager	Add topic	Training staff
TC17	Topic manager	Edit topic	Training staff
TC18	Topic manager	Delete topic	Training staff

TC19	Topic manager	Search topic	Training staff
TC20	Assign trainer manager	Add assign trainer	Training staff
TC21	Assign trainer manager	Edit assign trainer	Training staff
TC22	Assign trainer manager	Delete assign trainer	Training staff
TC23	Assign trainee manager	Add assign trainee	Training staff
TC24	Assign trainee manager	Edit assign trainee	Training staff
TC25	Assign trainee manager	Delete assign trainee	Training staff

TC26	Logged in	User has session	Administrator, training staff, trainer, trainee
TC27	Logout	User has log out	Administrator, training staff, trainer, trainee

3. Test case

ID	Test case name	Description	Test data	Expected	Actual	Remarks
TC1	Login	Login successful	Username: admin, Password: admin	Login successful	Login successful	Pass
TC2	Login	Login failed	Username: admin, Password: 5	Login failed	Login failed	Pass
TC3	Account manager	Display list account		Display all data in database	Display all data in database	Pass
TC4	Account manager	Add account	Role: admin, extra_code: test, username: test, password: 1, email: test@test, Phone: 123, gender: female, Fullname: test	Add successful	Add successful	Pass

TC5	Account manager	Edit account	Role: training staff, extra_code: test2, username: test, password: 1, email: test@test,	Edit successful	Edit successful	Pass

			Phone: 123, gender: male, Fullname: test			
TC6	Account manager	Delete account	Id = 12	Delete successful	Delete successful	Pass
TC7	Account manager	Search account	Trainner	Display data of user has name: Admin	Display data of user has name: Admin	Pass
TC8	Category manager	Add category	Name: category5, Status: active	Add successful	Add successful	Pass
TC9	Category manager	Edit category	Name: category2, Status: active	Edit successful	Edit successful	Pass
TC10	Category manager	Delete category	Id = 5	Delete successful	Delete successful	Pass
TC11	Category manager	Search category	Cate8	Display data of category has name: Cate2	Display data of category has name: Cate2	Pass
TC12	Course manager	Add course	Name: course4, Category: cate003, startDate: 10/09/2023, Status: active	Add successful	Add successful	Pass

TC13	Course manager	Edit course	Name: course2, Category: cate3, startDate: 10/09/2023, Status: active	Edit successful	Edit successful	Pass
TC14	Course manager	Delete course	Id = 20	Delete successful	Delete successful	Pass
TC15	Course manager	Search course	Course10	Display data of course has name: Course1	Display data of course has name: Course1	Pass
TC16	Topic manager	Add topic	Name: topic1, Course: course2, Document: document.doc Status: active	Add successful	Add successful	Pass
TC17	Topic manager	Edit topic	Name: topic2, Course: course2, Document: document2.doc Status: active	Edit successful	Edit successful	Pass
TC18	Topic manager	Delete topic	Id = 4	Delete successful	Delete successful	Pass

TC19	Topic manager	Search topic	Topic3	Display data of topic has name: Topic1	Display data of topic has name: Topic1	Pass
TC20	Assign trainer manager	Add assign trainer	Trainer: 2, Course: 2, Status: active	Add successful	Add successful	Pass
TC21	Assign trainer manager	Edit assign trainer	Trainer: 2, Course: 1, Status: active	Edit successful	Edit successful	Pass
TC22	Assign trainer manager	Delete assign trainer	Id = 8	Delete successful	Delete successful	Pass
TC23	Assign trainee manager	Add assign trainee	Trainee: 4, Topic: 2, Status: active	Add successful	Add successful	Pass
TC24	Assign trainee manager	Edit assign trainee	Trainee: 6, Topic: 5, Status: active	Edit successful	Edit successful	Pass
TC25	Assign trainee manager	Delete assign trainee	Id = 4	Delete successful	Delete successful	Pass

TC26	Logged in	User has session		Still login until logout	Logged out after few minute	Fail
TC27	Logout	User has log out		Logout successful	Logout successful	Pass

4. Evaluation

Strength:

Clarity in Problem Definition: The project completes it with a clear problem statement and scores a 4 out of 5. This lays a solid foundation for addressing the need.

Solutions on point: Feedback shows solutions are on point, with most ratings falling in the "effective to highly effective" range. While there are still gaps, it's clear the solutions are resonating.

Clear development plan: The comprehensive development strategy scored a commendable 4, showing a clearly communicated detailed plan for implementation. This solid foundation bodes well for successful project implementation.

Appropriate design and functionality: The app's design and functionality are appropriate for the intended purpose and user experience, mainly received at level 4. While this is a solid foundation, there is still work to be done. improvements to optimize usability and overall experience.

Weaknesses:

GUI Maze: The interface could use some navigation tweaks and a more user-focused design. Ratings range from 3-4, indicating areas that need improvement. A better GUI will increase satisfaction and engagement.

Opportunities for improvement:

Refine the solution: Dive into user feedback to refine the proposed solution and improve its effectiveness. An iterative approach that incorporates user insights will tailor the app to meet their needs.

Optimize your growth strategy: Your growth strategy is solid, but you should review it closely to find areas that need improvement. Continuous improvement in development methods can improve efficiency and inspire innovation.

Enhanced interface: Prioritize interface improvements focused on intuitive navigation and user-centered design. This will take the user experience to the next level and increase overall satisfaction.

Feedback-Driven Iteration: Create a robust feedback loop, combining iterative testing and integrating flexible feedback to continuously refine the application. This ensures continuous alignment with user expectations and evolving needs.

E. Conclusion

In conclusion, this critical evaluation not only highlights the notable strengths of the project, it also acknowledges areas that could be improved. Leveraging these insights, I'm committed to adopting an iterative approach that refines the solution, optimizes development strategies, and prioritizes interface enhancements. This allows the project to further evolve in order to meet and exceed user expectations. Embracing user-centric iterations and robust testing throughout the development process ensures refined solutions with intuitive navigation features. This iterative approach includes regular feedback from users, which helps to ensure that the final product is user-friendly, effective, and highly satisfactory, always aligning with user expectations.

F.Reference

Survey:

https://docs.google.com/forms/d/e/1FAIpQLSftn5EkW38eCNMaRFyMR2t8Xp8hk34q5dTuvzkNHgFSEwvfDw/viewform?usp=sharing

Source Code: https://github.com/NEKOshirp/WEB-MVC