ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 30: Application Development		
Submission date	10/04/2024	**Date Received 1st submission**	10/04/2024
Re-submission Date	17/04/2024	**Date Received 2nd submission**	17/04/2024
Student Name	Đỗ Nam Khánh	**Student ID**	BH00021
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	khanh

Grading grid

P4	P5	P6	M3	M4	M5	D2	D3

□ Summative Feedback:	□ Resubmission Feedback:	
Grade:	Assessor Signature:	Date:
Lecturer Signature:		

Contents

A. Introduction.

In the meticulous endeavor of crafting a robust business application, our team embarked on a comprehensive journey that transcended mere conceptualization and design. We meticulously navigated through a process that encompassed rigorous evaluation and iterative refinements, ensuring the utmost precision and efficacy in our approach. This report serves as a comprehensive documentation of our systematic methodology, tracing the evolution from initial design concepts, as presented previously, to the culmination of the development phase. It meticulously captures the strategic utilization of cutting-edge technologies and methodologies that underpin the application's architecture and functionality.

Throughout this endeavor, we actively solicited and incorporated feedback from our esteemed peers through a structured questionnaire. This invaluable input served as a guiding light, empowering us to adapt and evolve reflectively, integrating key insights into our development process. Furthermore, our report conducts a critical analysis of the application's performance against predefined specifications, delving into the nuanced factors that shape its efficacy and functionality.

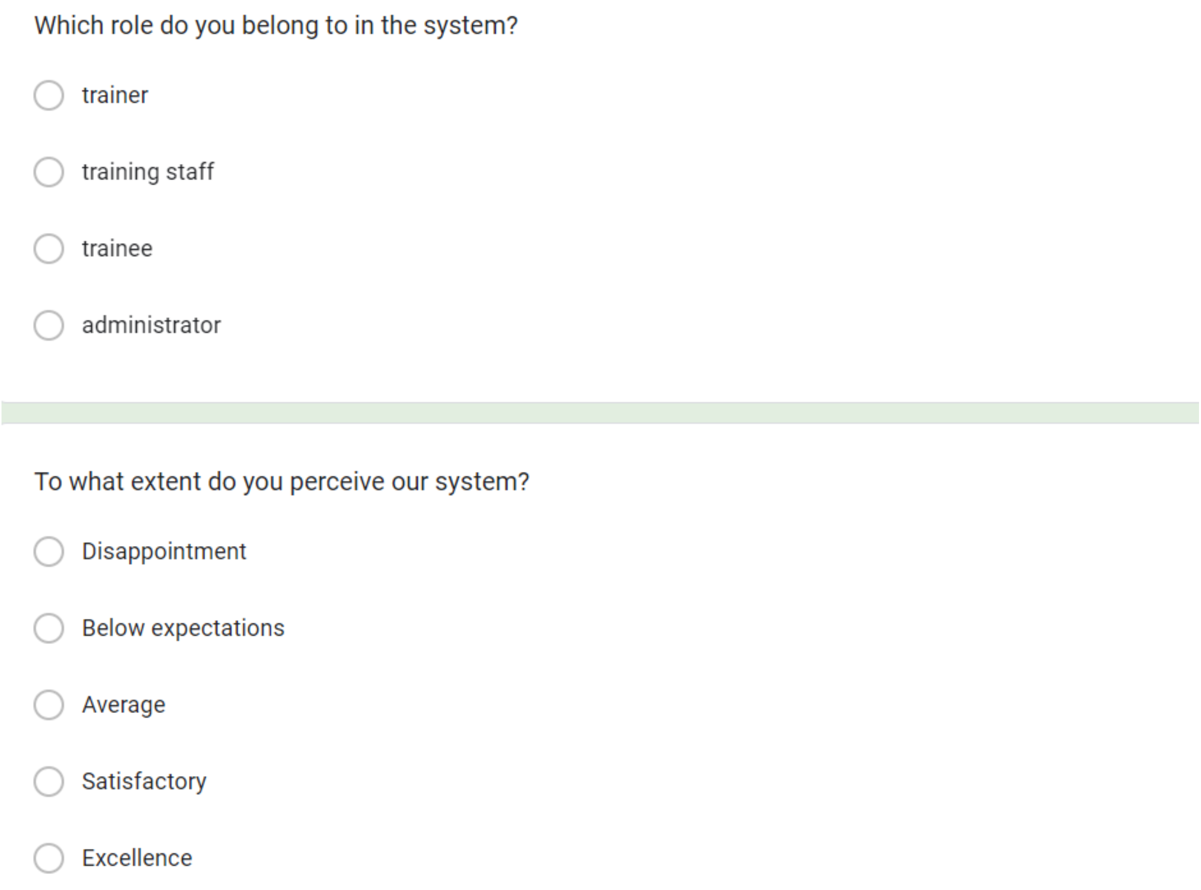
The fruition of our collective efforts is a comprehensive presentation and a fully functional application, ready for demonstration and evaluation. In the concluding chapter, we present a reflective analysis, acknowledging the application's strengths, discerning potential weaknesses, and envisioning avenues for future enhancements. This report stands as a testament to our unwavering dedication to delivering high-caliber business solutions that empower enterprises to realize their strategic objectives.

B. Content.

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:

   Link google from: https://forms.gle/Pw1VrFYtyNrQJKJn6



Which role do you belong to in the system?

○ trainer

○ training staff

○ trainee

○ administrator

To what extent do you perceive our system?

○ Disappointment

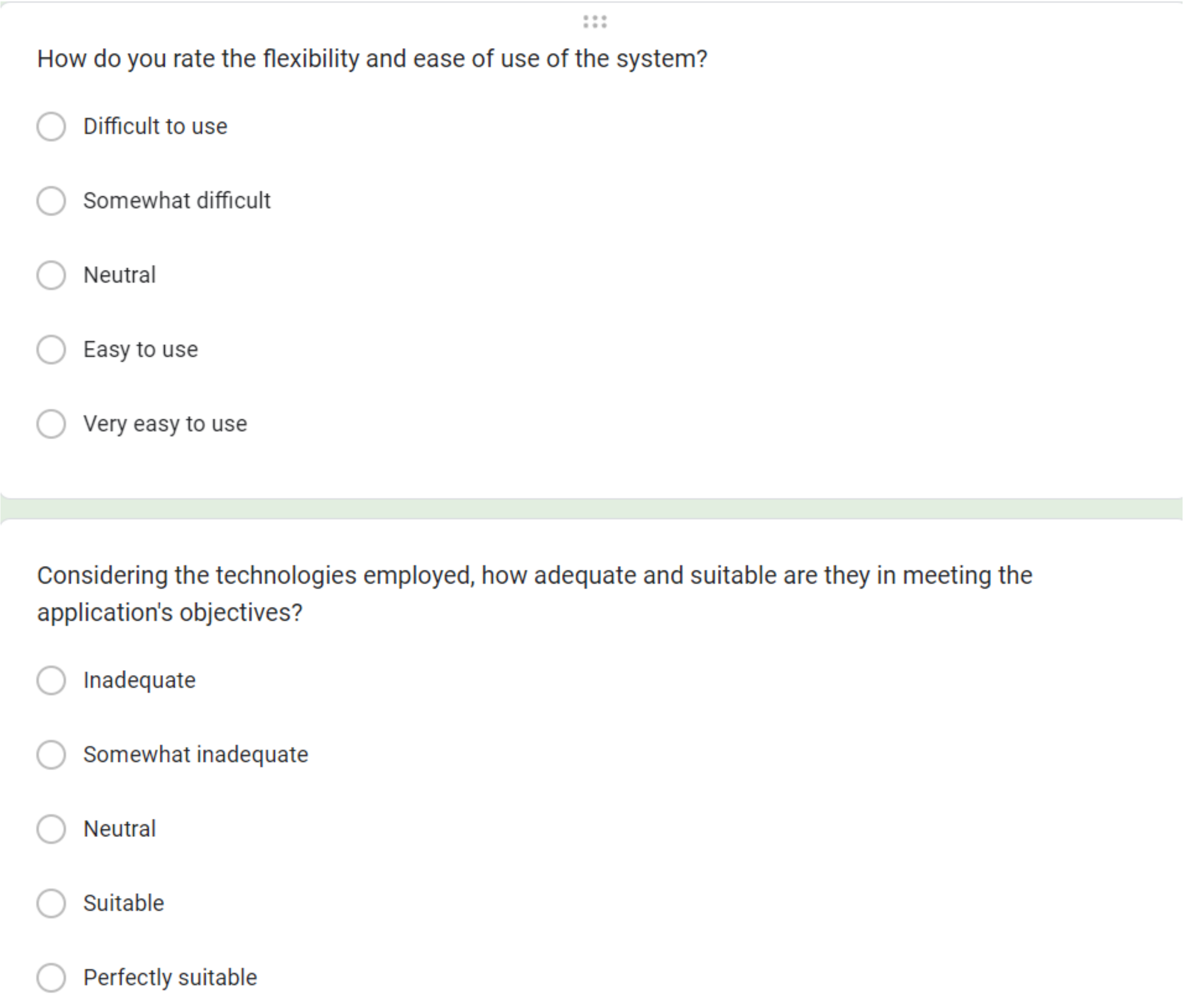
○ Below expectations

○ Average

○ Satisfactory

○ Excellence

Figure 1 Question



How do you rate the flexibility and ease of use of the system?

○ Difficult to use

○ Somewhat difficult

○ Neutral

○ Easy to use

○ Very easy to use

Considering the technologies employed, how adequate and suitable are they in meeting the application's objectives?

○ Inadequate

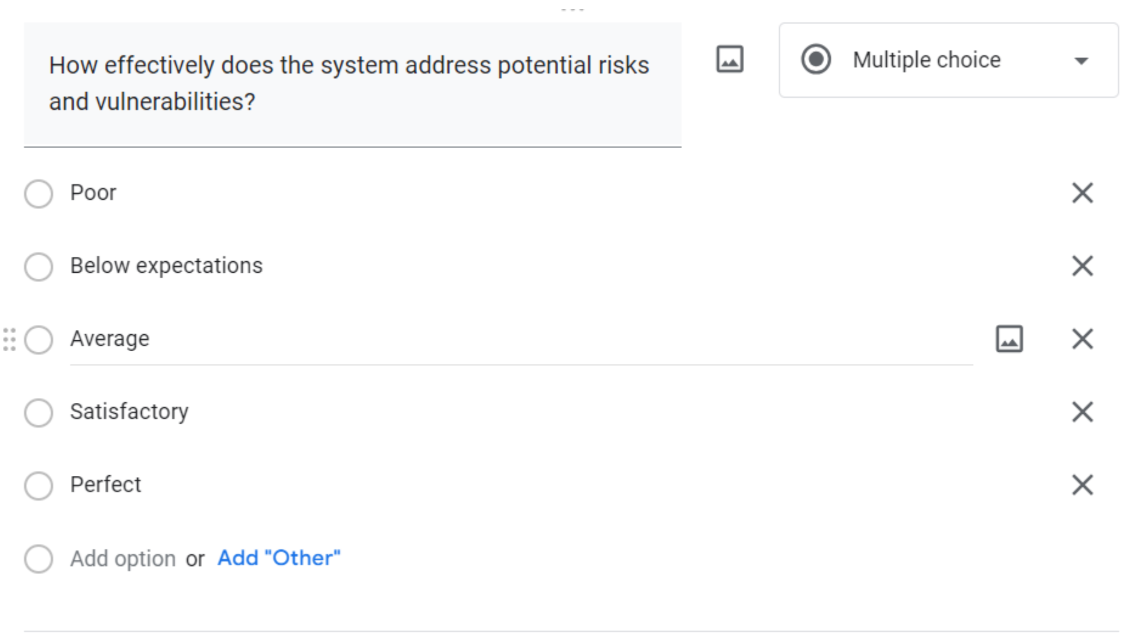
○ Somewhat inadequate

○ Neutral

○ Suitable

○ Perfectly suitable

Figure 2 Question



How effectively does the system address potential risks and vulnerabilities?

● Multiple choice

○ Poor

○ Below expectations

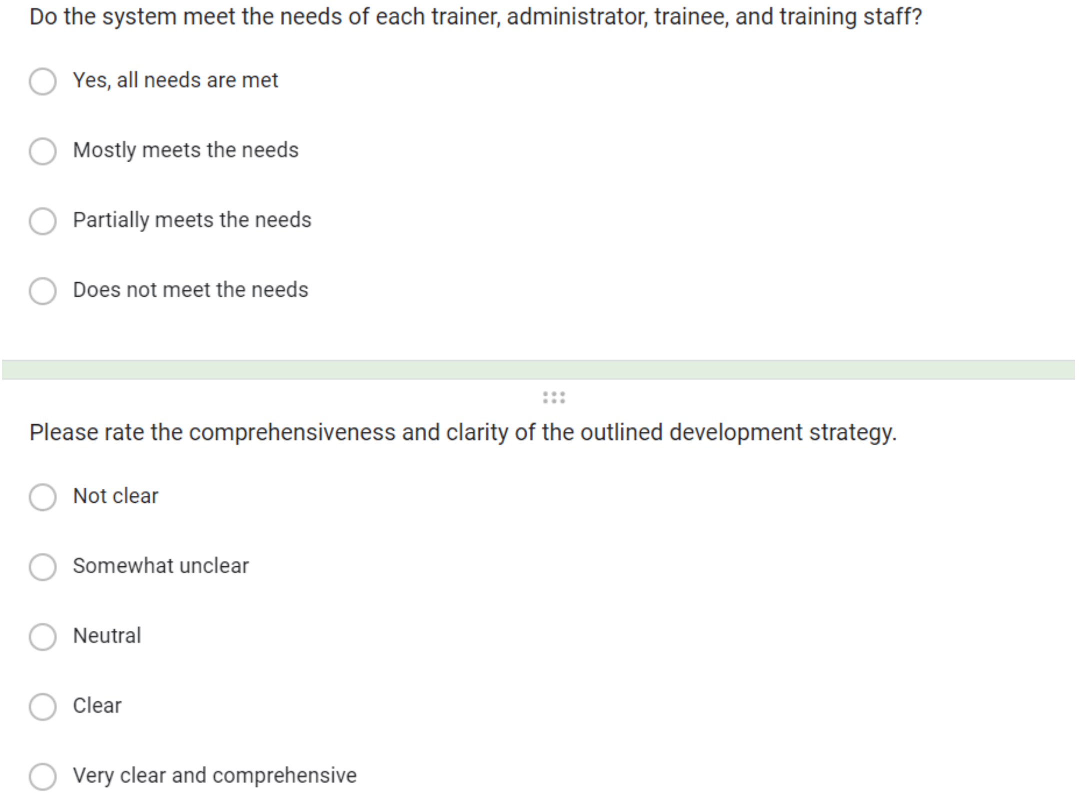
○ Average

○ Satisfactory

○ Perfect

○ Add option or Add "Other"

Figure 3 Question



Do the system meet the needs of each trainer, administrator, trainee, and training staff?

○ Yes, all needs are met

○ Mostly meets the needs

○ Partially meets the needs

○ Does not meet the needs

Please rate the comprehensiveness and clarity of the outlined development strategy.

○ Not clear

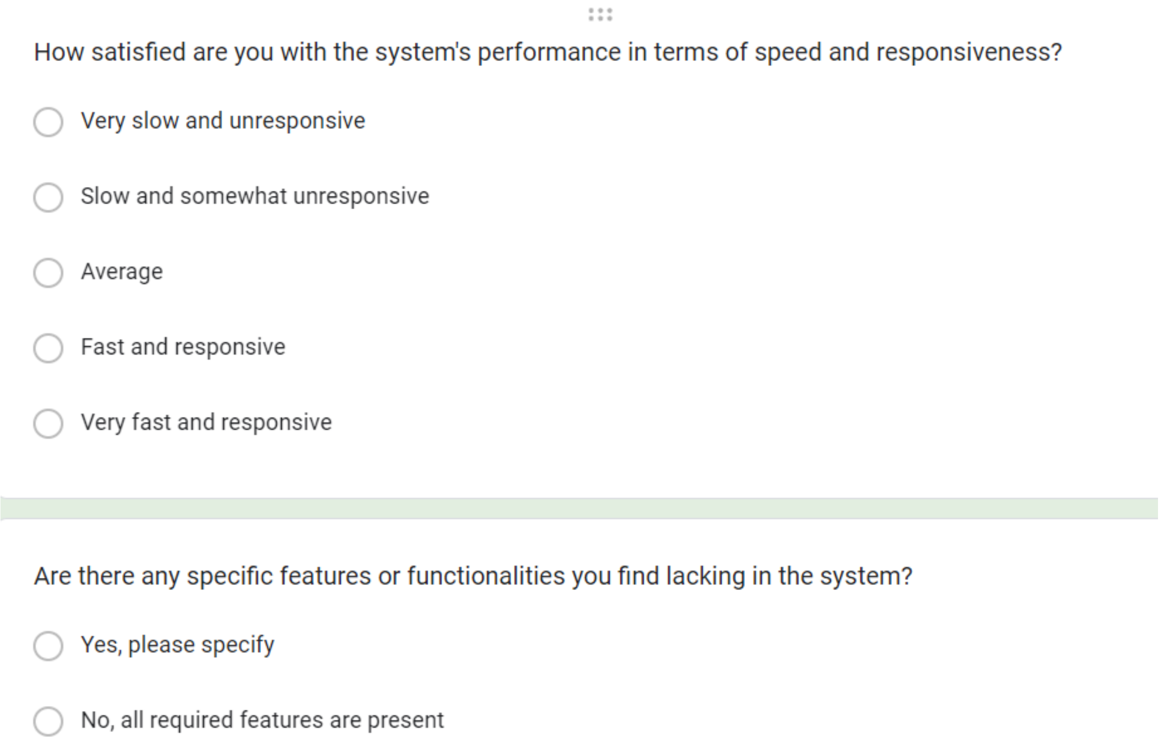
○ Somewhat unclear

○ Neutral

○ Clear

○ Very clear and comprehensive

Figure 4 Question



How satisfied are you with the system's performance in terms of speed and responsiveness?

○ Very slow and unresponsive

○ Slow and somewhat unresponsive

○ Average

○ Fast and responsive

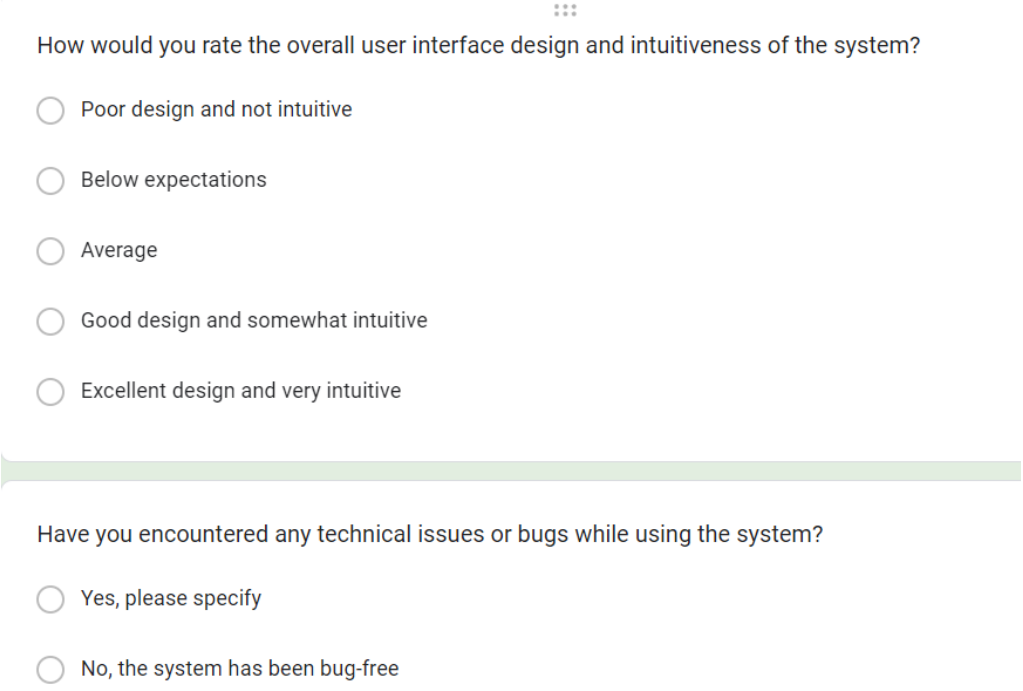
○ Very fast and responsive

Are there any specific features or functionalities you find lacking in the system?

○ Yes, please specify

○ No, all required features are present

Figure 5 Question



How would you rate the overall user interface design and intuitiveness of the system?

○ Poor design and not intuitive

○ Below expectations

○ Average

○ Good design and somewhat intuitive

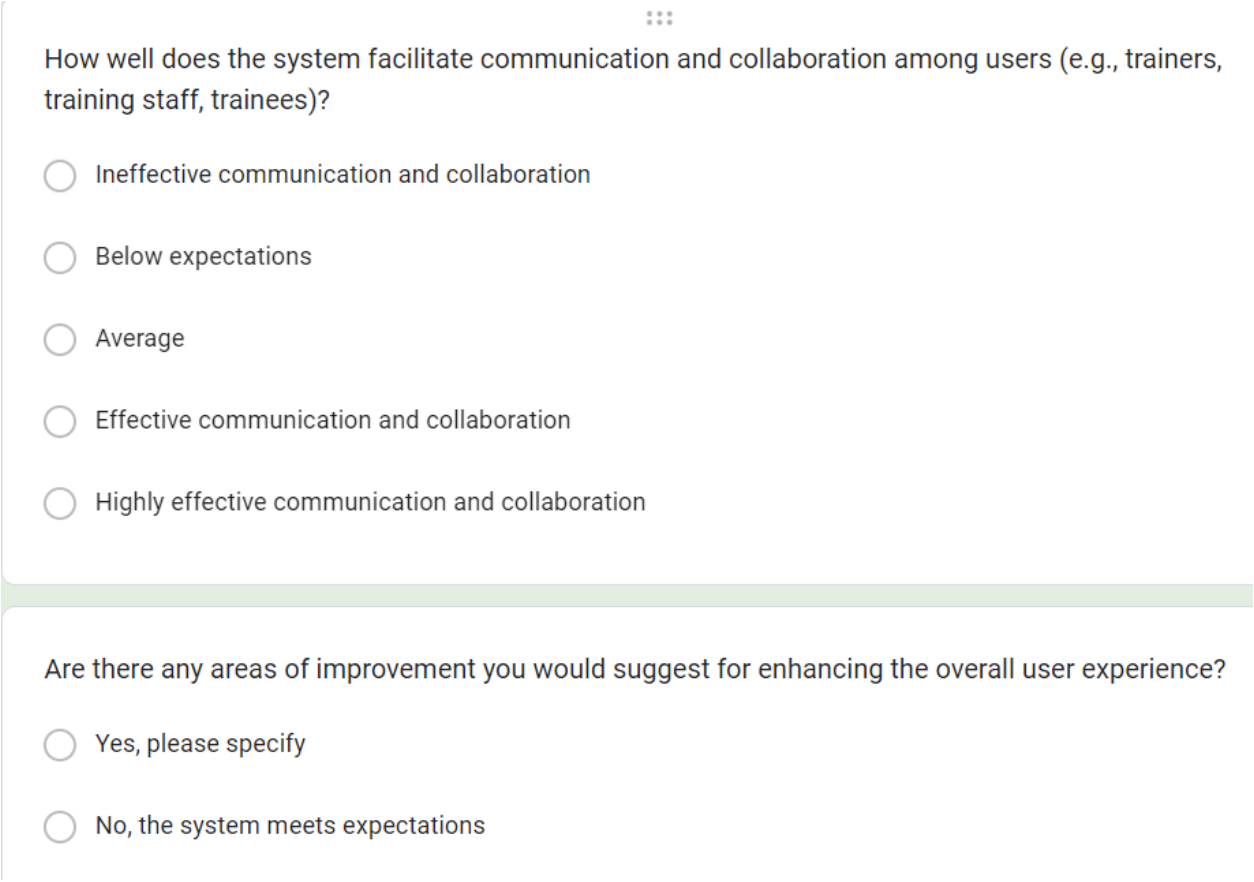
○ Excellent design and very intuitive

Have you encountered any technical issues or bugs while using the system?

○ Yes, please specify

○ No, the system has been bug-free

Figure 6 Question



How well does the system facilitate communication and collaboration among users (e.g., trainers, training staff, trainees)?

○ Ineffective communication and collaboration

○ Below expectations

○ Average

○ Effective communication and collaboration

○ Highly effective communication and collaboration

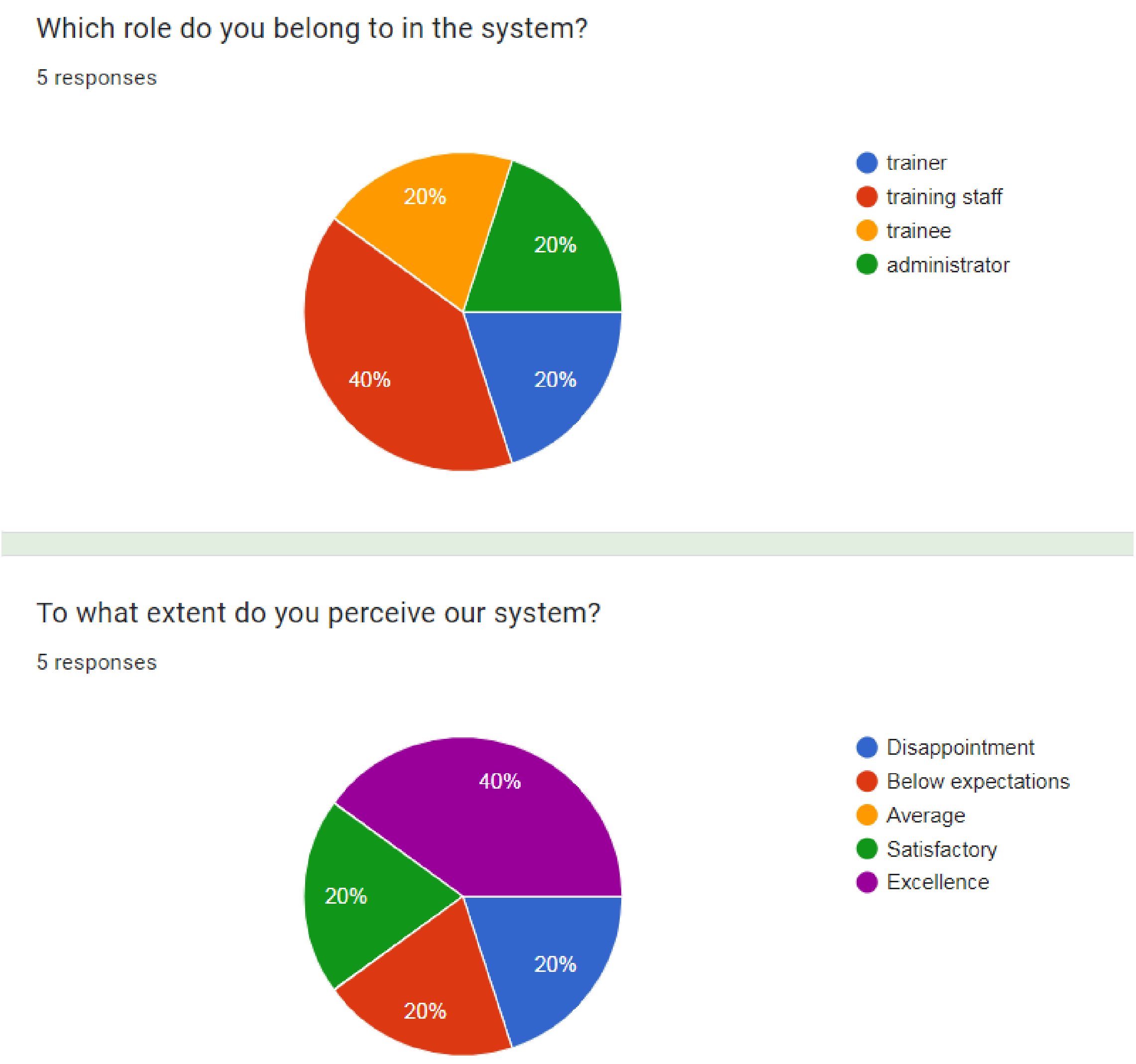
Are there any areas of improvement you would suggest for enhancing the overall user experience?

○ Yes, please specify

○ No, the system meets expectations

Figure 7 Question

2. The result of survey



Which role do you belong to in the system?

5 responses

To what extent do you perceive our system?

5 responses

Figure 8 Result

2.1.Role Distribution:

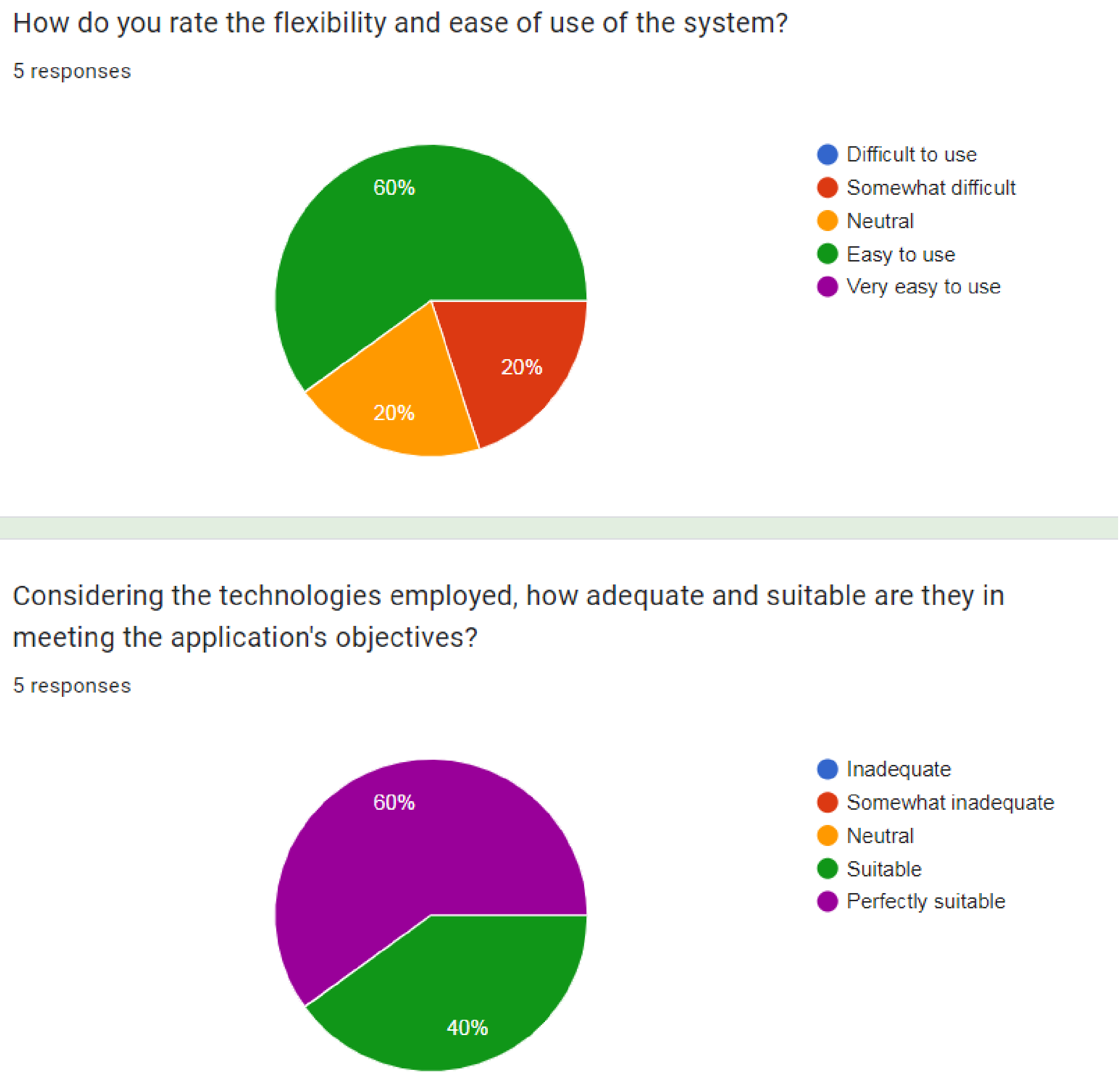
- Training staff: 40%
- Trainer, Trainee, and Administrator (combined): 20% each

Explanation: The majority of respondents (40%) identified themselves as training staff, indicating that this role is more prevalent among those interacting with the system. The remaining 20% of respondents were distributed evenly among the other roles, suggesting a relatively balanced representation across trainer, trainee, and administrator roles.

2.2.Perception of the System:

- Excellence (5): 40%
- Below expectations (2), Average (3), Satisfactory (4): 20% each

Explanation: A significant portion of respondents (40%) perceived the system as excellent, indicating a high level of satisfaction and appreciation for its functionality and features. Conversely, the remaining 60% of respondents selected various lower ratings, suggesting differing levels of satisfaction or areas for improvement in the system's performance and usability.



How do you rate the flexibility and ease of use of the system?

5 responses

Considering the technologies employed, how adequate and suitable are they in meeting the application's objectives?

5 responses

Figure 9 Result

2.3.Flexibility and Ease of Use:

- 60% of respondents chose "d) 4 - Easy to use."

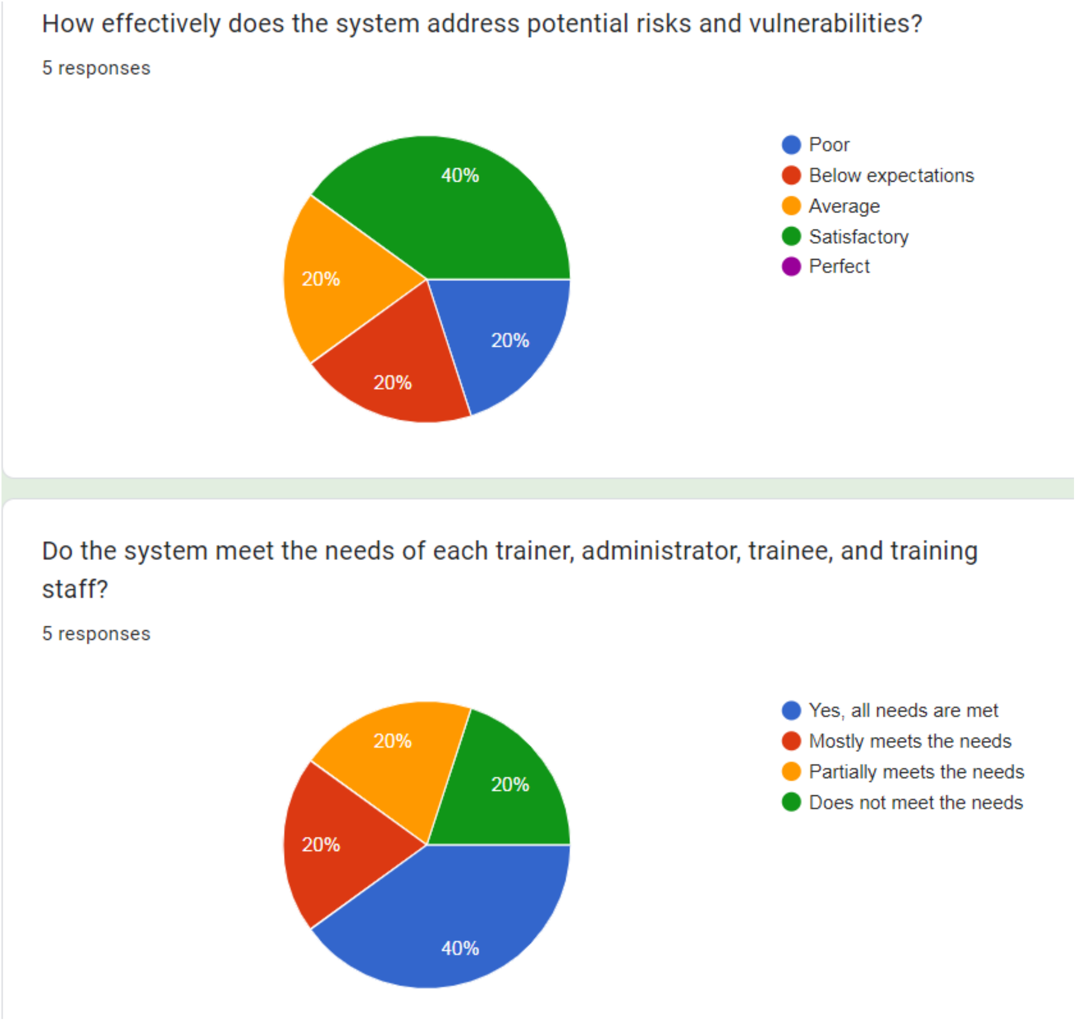
- 40% of respondents split evenly between "b) 2 - Somewhat difficult" and "c) 3 - Neutral."

Explanation: The majority of respondents (60%) find the system easy to use, indicating a high level of flexibility and user-friendliness. However, there is a significant minority (40%) who find the system somewhat difficult or neutral in terms of ease of use, suggesting that there may be areas for improvement in user interface design or instructional materials to enhance usability for all users.

2.4.Suitability of Technologies:

- 60% of respondents chose "e) 5 - Perfectly suitable."
- 40% of respondents chose "d) 4 - Suitable."

Explanation: The majority of respondents (60%) believe that the technologies employed in the system are perfectly suitable for meeting the application's objectives, indicating a high level of confidence in the technological infrastructure. Additionally, 40% of respondents consider the technologies to be suitable, further supporting the notion that the system is well-equipped to fulfill its intended purposes. Overall, the high proportion of respondents selecting "Perfectly suitable" suggests a robust technological foundation supporting the system's functionalities.



How effectively does the system address potential risks and vulnerabilities?

5 responses

Do the system meet the needs of each trainer, administrator, trainee, and training staff?

5 responses

Figure 10 Result

2.5.Effectiveness in Addressing Risks and Vulnerabilities:

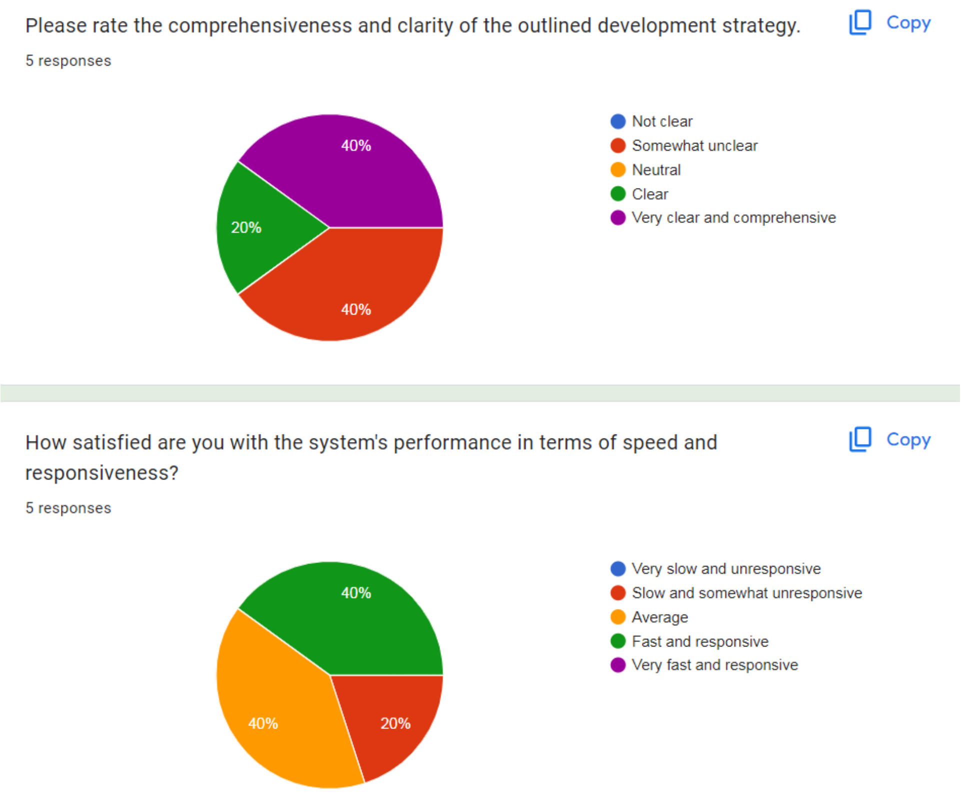
- 40% of respondents chose "d) 4 - Satisfactory."
- The remaining 60% of respondents split evenly between "e) 5 - Perfect," "c) 3 - Average," and "b) 2 - Below expectations."

Explanation: A significant portion of respondents (40%) consider the system's effectiveness in addressing potential risks and vulnerabilities to be satisfactory, indicating a moderate level of confidence in its security measures. However, the remaining 60% of respondents have varied perceptions, with equal proportions expressing high confidence (choosing "Perfect"), moderate confidence (choosing "Average"), and lower confidence (choosing "Below expectations") in the system's ability to mitigate risks and vulnerabilities.

2.6.Meeting the Needs of Users:

- 40% of respondents chose "a) Yes, all needs are met."
- The remaining 60% of respondents split evenly between "b) Mostly meets the needs," "c) Partially meets the needs," and "d) Does not meet the needs."

Explanation: A significant portion of respondents (40%) believe that the system meets the needs of all users, indicating a high level of satisfaction and effectiveness in catering to the diverse requirements of trainers, administrators, trainees, and training staff. However, the remaining 60% of respondents have differing perceptions, with equal proportions expressing varying degrees of satisfaction or dissatisfaction with the system's ability to meet user needs. This suggests that while the system may perform well for some users, there may be room for improvement to better address the needs of others.



Please rate the comprehensiveness and clarity of the outlined development strategy.

5 responses

How satisfied are you with the system's performance in terms of speed and responsiveness?

5 responses

Figure 11 Result

2.7.Comprehensiveness and Clarity of Development Strategy:

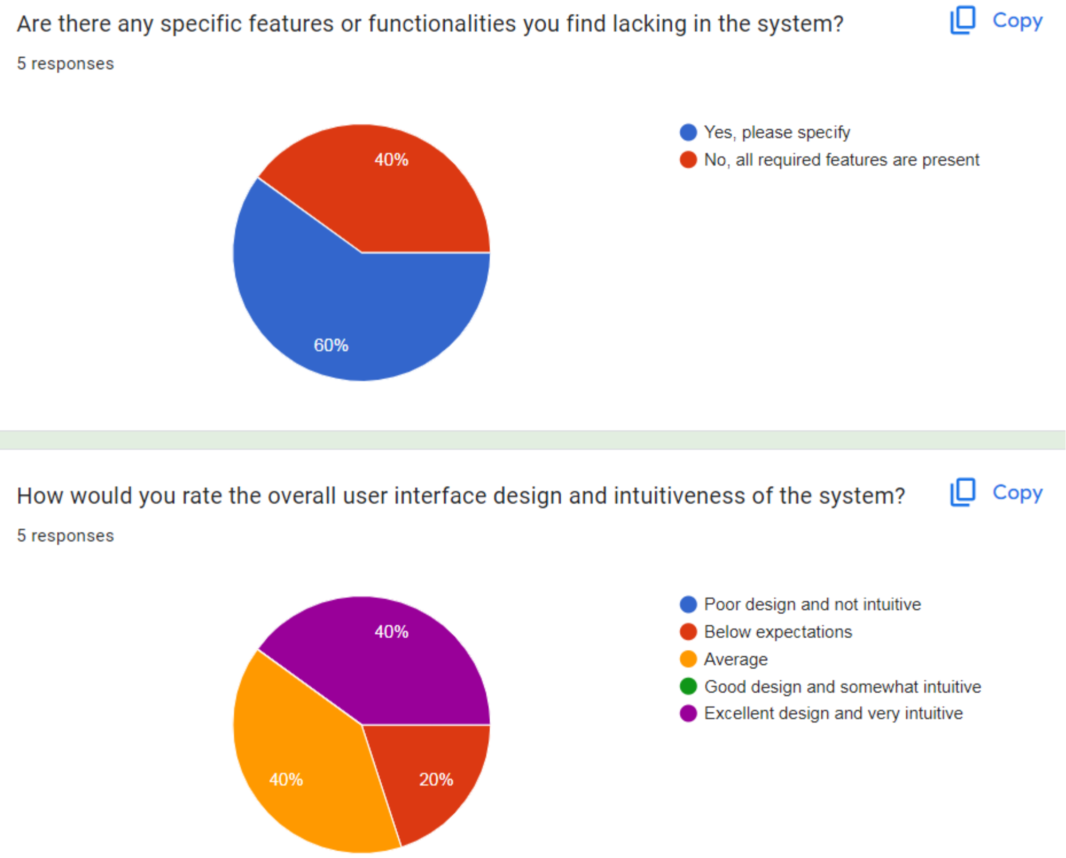
- 40% of respondents chose "b) 2 - Somewhat unclear."
- 40% of respondents chose "e) 5 - Very clear and comprehensive."
- The remaining 20% of respondents chose "d) 4 - Clear."

Explanation: A significant portion of respondents (40%) perceive the outlined development strategy as somewhat unclear, suggesting that there may be areas that require further elaboration or clarification. However, an equal proportion of respondents (40%) find the development strategy to be very clear and comprehensive, indicating a high level of satisfaction with the provided information. The remaining 20% of respondents consider the strategy to be clear, indicating a moderate level of satisfaction.

2.8.System's Performance in Speed and Responsiveness:

- 40% of respondents chose "c) 3 - Average."
- 40% of respondents chose "d) 4 - Fast and responsive."
- The remaining 20% of respondents chose "b) 2 - Slow and somewhat unresponsive."

Explanation: The majority of respondents (80%) perceive the system's performance in terms of speed and responsiveness positively, with 40% rating it as average and another 40% rating it as fast and responsive. However, there is a minority (20%) who find the system to be somewhat slow and unresponsive, indicating room for improvement in this aspect of the system's performance.



Are there any specific features or functionalities you find lacking in the system?

5 responses

How would you rate the overall user interface design and intuitiveness of the system?

5 responses

Figure 12 Result

2.9.Specific Features or Functionalities Lacking:

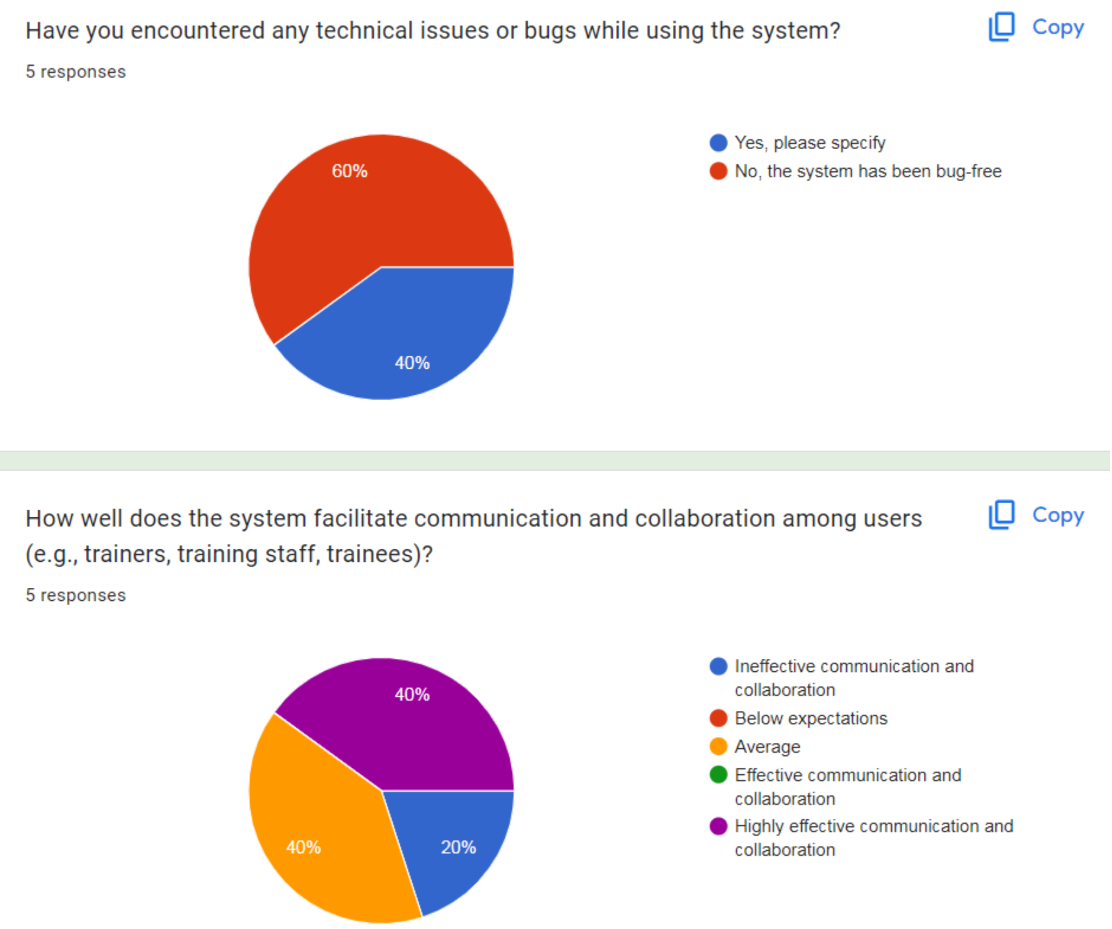
- 60% of respondents chose "a) Yes, please specify."
- The remaining 40% of respondents chose "b) No, all required features are present."

Explanation: A majority of respondents (60%) identified specific features or functionalities that they find lacking in the system, indicating areas for improvement or enhancement. However, 40% of respondents believe that all required features are present, suggesting a satisfactory level of completeness in the system's functionality from their perspective.

2.10. Overall User Interface Design and Intuitiveness:

- 40% of respondents chose "c) 3 - Average."
- 40% of respondents chose "e) 5 - Excellent design and very intuitive."
- The remaining 20% of respondents chose "b) 2 - Below expectations."

Explanation: The majority of respondents (80%) rated the overall user interface design and intuitiveness positively, with 40% considering it to be average and another 40% rating it as excellent and very intuitive. However, there is a minority (20%) who perceive the design to be below expectations, indicating potential areas for improvement in terms of user interface design and intuitiveness.



Have you encountered any technical issues or bugs while using the system?

5 responses

How well does the system facilitate communication and collaboration among users (e.g., trainers, training staff, trainees)?

5 responses

Figure 13 Result

2.11. Encountered Technical Issues or Bugs:

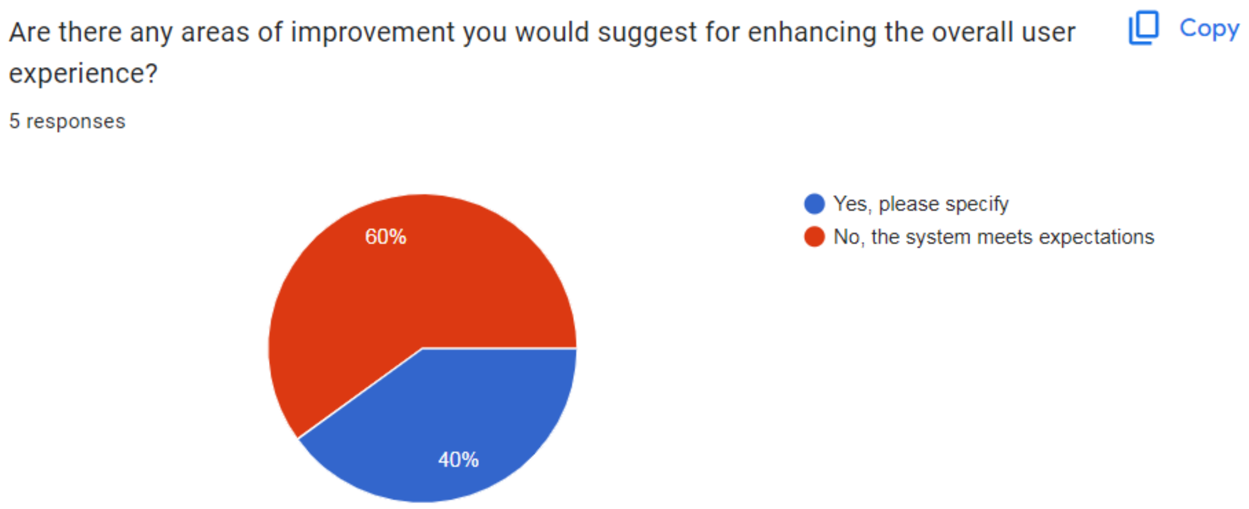
- 60% of respondents chose "a) Yes, please specify."
- The remaining 40% of respondents chose "b) No, the system has been bug-free."

Explanation: The majority of respondents (60%) reported encountering technical issues or bugs while using the system, indicating areas that require attention and improvement. However, 40% of respondents have not experienced any technical issues, suggesting a relatively stable and bug-free system for them.

2.12. Facilitation of Communication and Collaboration:

- 40% of respondents chose "c) 3 - Average."
- 40% of respondents chose "e) 5 - Highly effective communication and collaboration."
- The remaining 20% of respondents split evenly between "a) 1 - Ineffective communication and collaboration" and "b) 2 - Below expectations."

Explanation: The majority of respondents (80%) perceive the system to facilitate communication and collaboration effectively, with 40% considering it to be highly effective and another 40% rating it as average. However, there is a minority (20%) who have lower perceptions of the system's effectiveness in this rega



Are there any areas of improvement you would suggest for enhancing the overall user experience?

5 responses

Figure 14 Result

2.13. Areas of Improvement for User Experience:

- Nearly 60% of respondents chose "b) No, the system meets expectations."
- The remaining respondents (approximately 40%) chose "a) Yes, please specify."

Explanation: The majority of respondents (nearly 60%) indicated that the system meets their expectations, suggesting overall satisfaction with the user experience. However, approximately 40% of respondents identified areas for improvement, indicating specific aspects of the system that could enhance the user experience further.

3. Feedback:

- The comprehensive survey you conducted sheds light on the overall satisfaction levels of users towards the business application. It's evident from the responses that there's a prevailing sense of contentment, with ratings predominantly falling between levels 3 to 5, indicating moderate to high satisfaction. Notably, level 2 ratings were scarce, suggesting a generally positive sentiment across the board.
- One particularly encouraging aspect is the feedback on the problem definition statement. It received consistently high ratings, primarily around level 4, implying a clear and articulate presentation of the core challenges addressed by the application.

This clarity is crucial as it underpins the application's relevance and effectiveness in meeting the identified needs.

- In terms of the effectiveness of the proposed solution, the majority of responses clustered around level 4, suggesting a moderate to high level of effectiveness. While users perceive the application as providing valuable solutions, there's a consensus that further enhancements could be implemented to optimize its effectiveness and ensure it delivers the desired outcomes with utmost efficiency.

- The satisfaction expressed by users regarding the clarity and comprehensiveness of the development strategy is also noteworthy. Most responses, hovering around level 4, indicate that users perceive the strategy as well-defined and effectively communicated. However, there are areas identified for potential improvement, suggesting opportunities to elevate the strategy to an exemplary level and ensure it aligns seamlessly with the overall objectives of the application.

- Feedback on the alignment between the application's design, functionality, and user experience is largely positive, with ratings averaging around level 4. This indicates that the application effectively integrates its design elements with its functionality to create a cohesive user experience. Nevertheless, there's acknowledgment of areas where fine-tuning may be necessary to enhance usability and elevate the overall user experience to even greater heights.

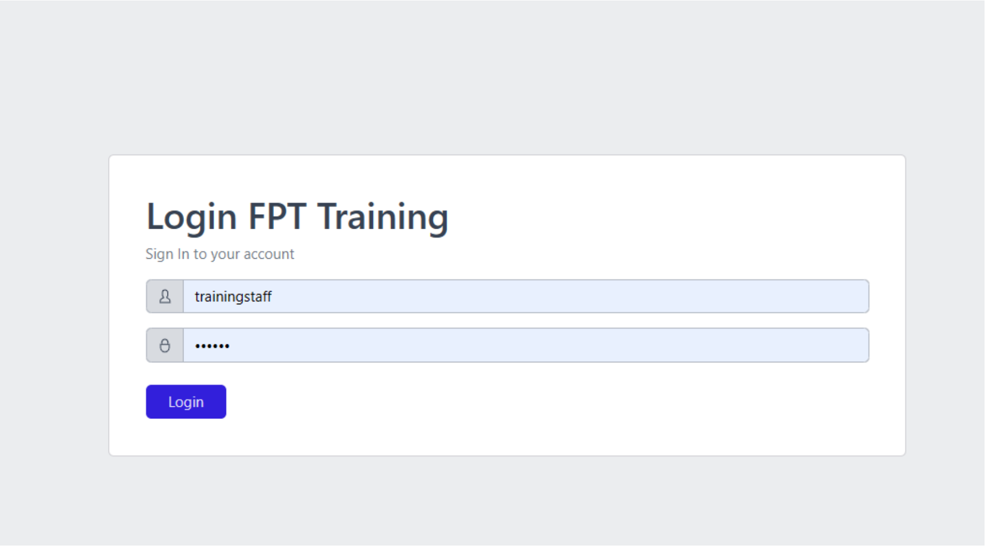
- While users generally find the interface intuitive and user-friendly, as reflected in ratings ranging from 4 to 5, there's recognition of potential areas for improvement. Enhancements aimed at making the interface more user-centric and improving navigational ease could further enhance user satisfaction and overall usability.

- In summary, the survey responses paint a favorable picture of the business application, highlighting its strengths while also identifying areas for refinement and optimization. The aggregated ratings, spanning levels 3 to 5, provide valuable insights that can serve as a roadmap for further enhancing the application's capabilities and ensuring it evolves into a truly robust and user-centric solution.

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

1. Training staff role’s .

   1.1.Login Training Staff.



Login FPT Training

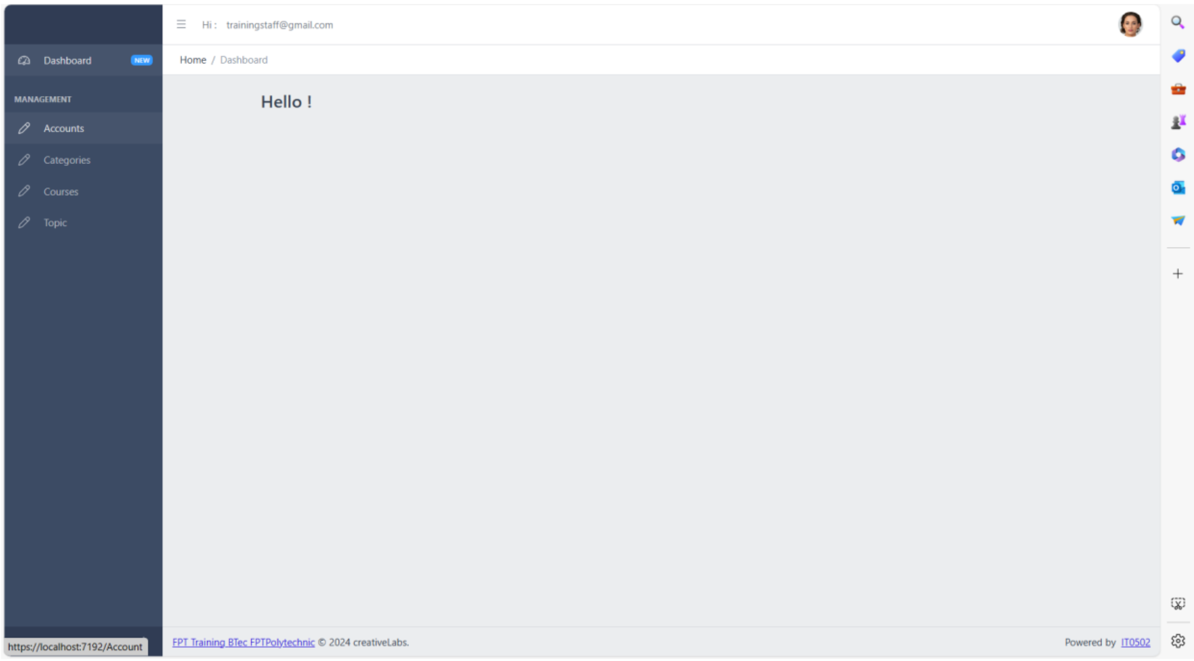
Sign In to your account

trainingstaff

••••••

Figure 15 Login

Dashboard:



Hi : trainingstaff@gmail.com

Home / Dashboard

Hello !

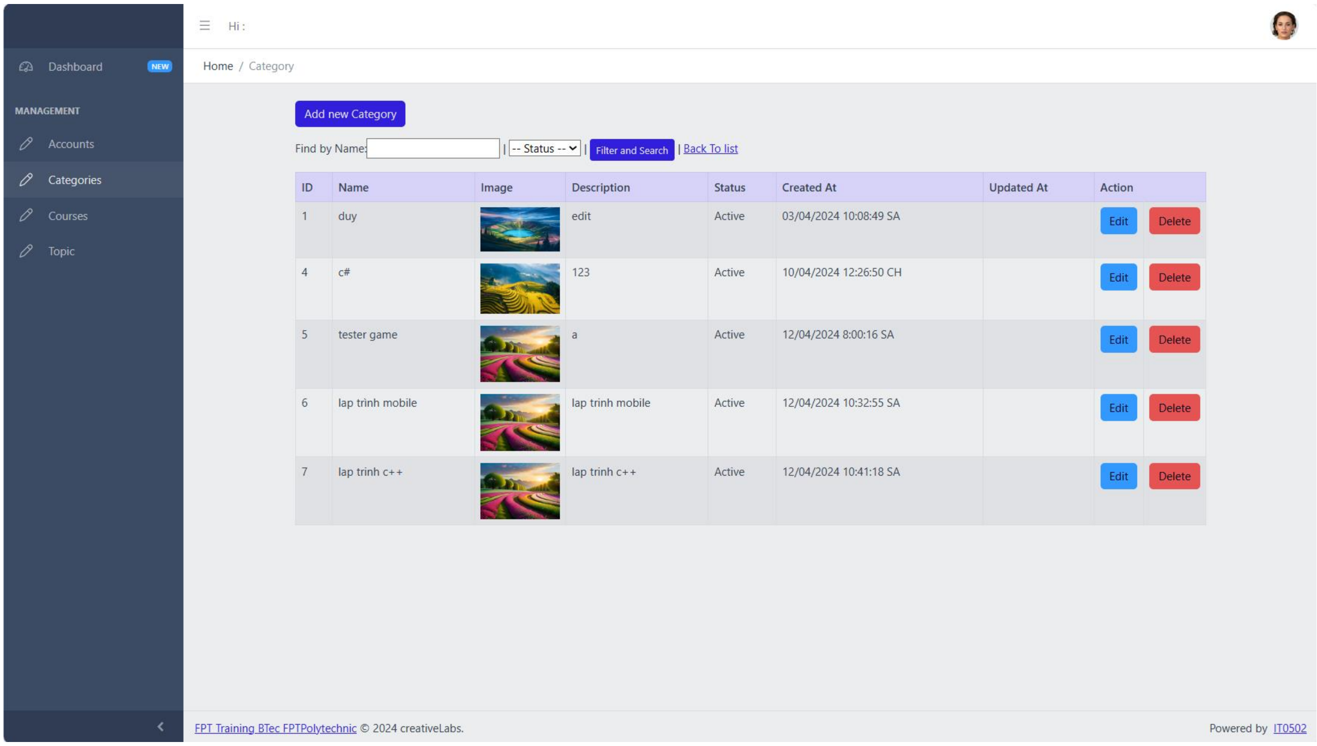
FPT Training BTec FPTPolytechnic © 2024 creativeLabs.

Powered by IT0502

https://localhost:7192/Account

Figure 16 Dashboard Training Sfatt

1.2.User Category management.

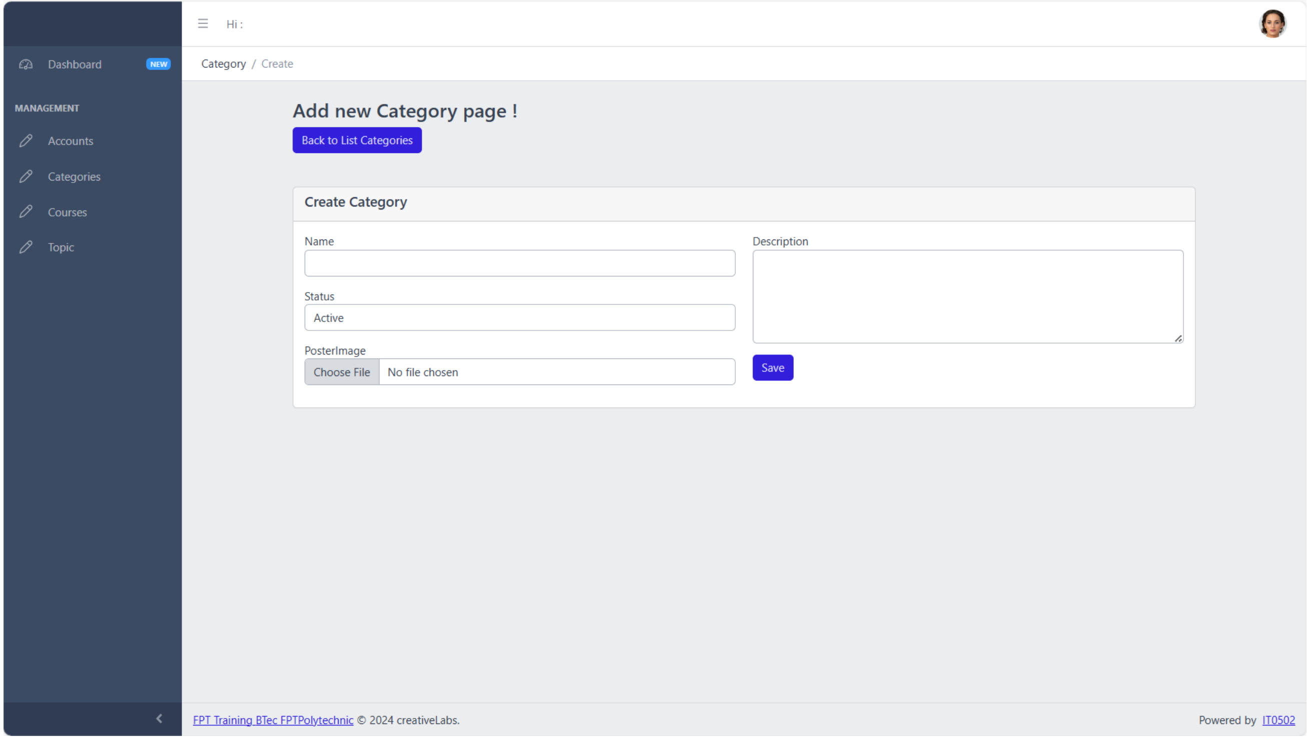


Find by Name: | -- Status -- | Filter and Search | Back To list

ID	Name	Image	Description	Status	Created At	Updated At	Action
1	duy		edit	Active	03/04/2024 10:08:49 SA		Edit Delete
4	c#		123	Active	10/04/2024 12:26:50 CH		Edit Delete
5	tester game		a	Active	12/04/2024 8:00:16 SA		Edit Delete
6	lap trình mobile		lap trinh mobile	Active	12/04/2024 10:32:55 SA		Edit Delete
7	lap trinh c++		lap trinh c++	Active	12/04/2024 10:41:18 SA		Edit Delete

FPT Training BTec FPTPolytechnic © 2024 creativeLabs. Powered by IT0502

Figure 17 Index Category



Add new Category page !

Create Category

Name:

Description:

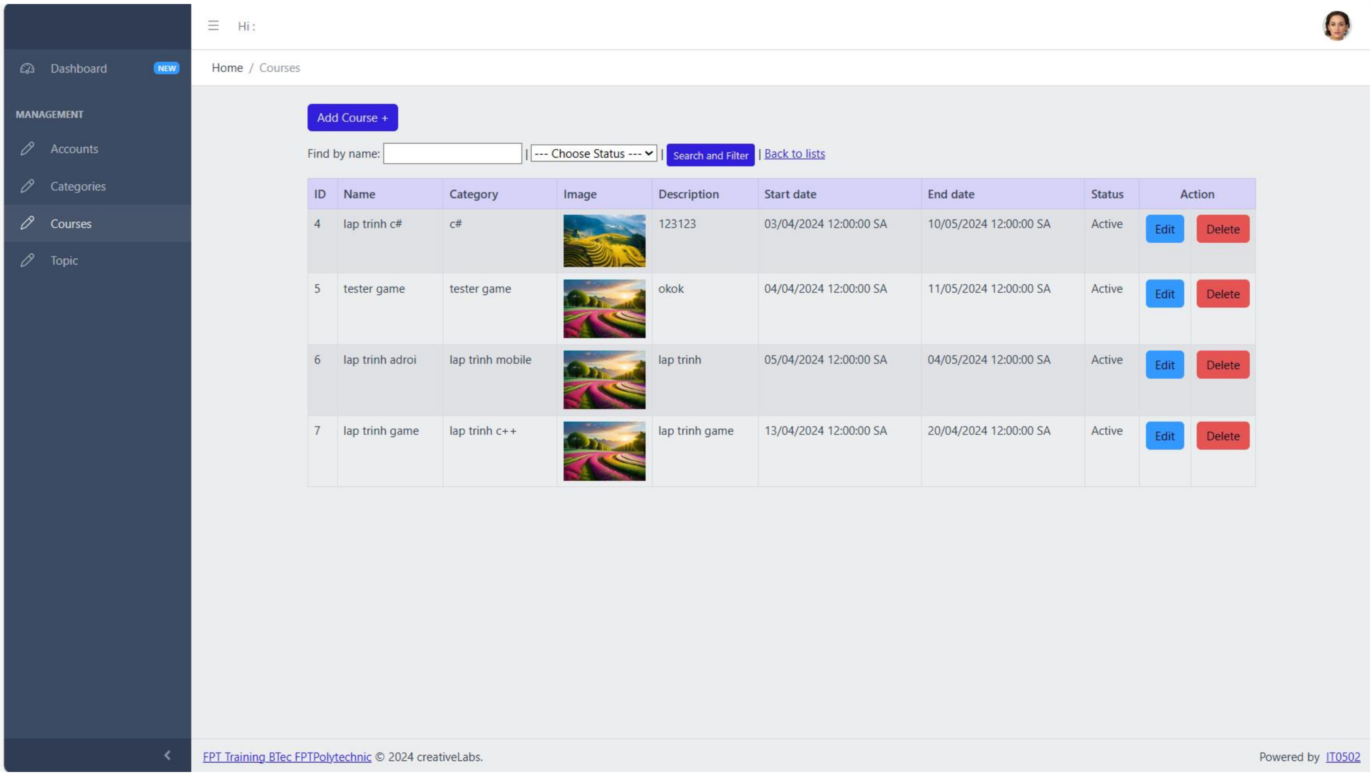
Status: Active

PosterImage: No file chosen

FPT Training BTec FPTPolytechnic © 2024 creativeLabs. Powered by IT0502

Figure 18 Add Category

1.3.User Course management.

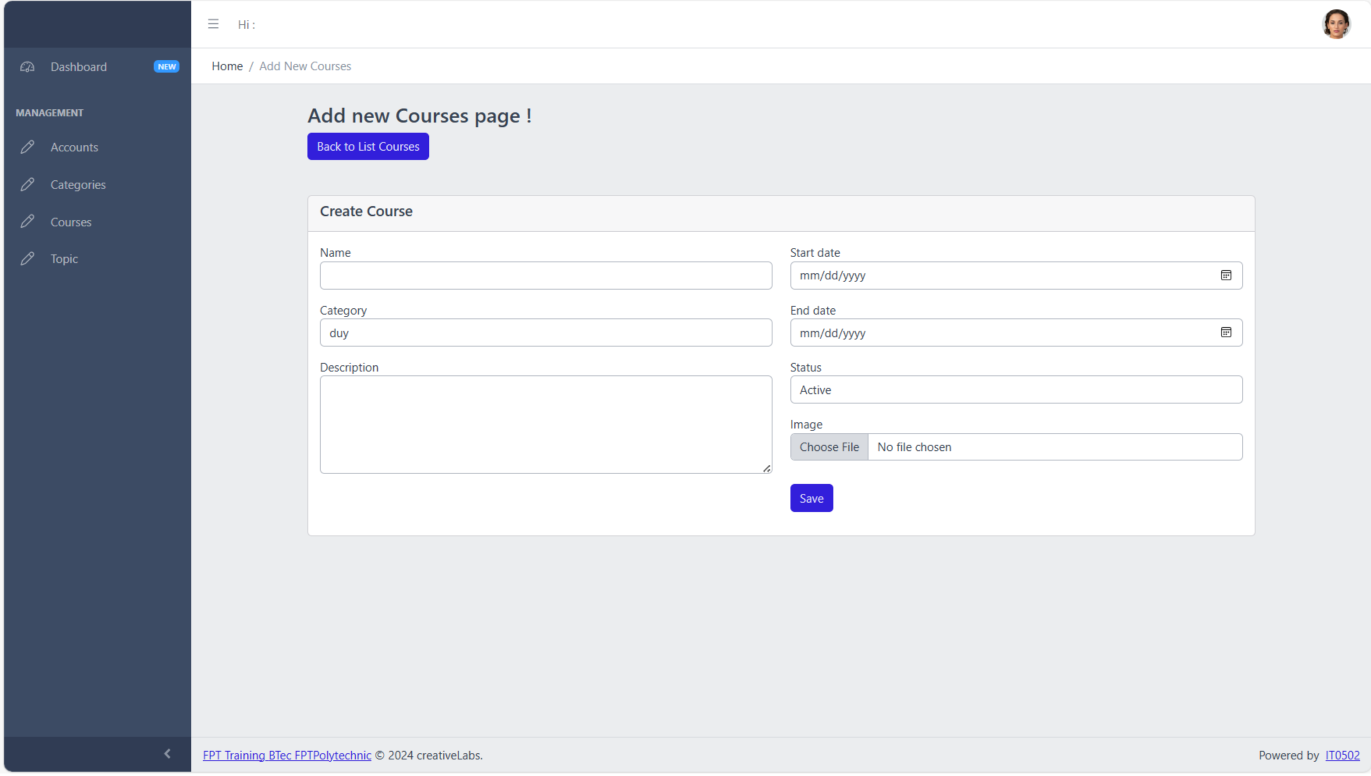


ID	Name	Category	Image	Description	Start date	End date	Status	Action
4	lap trinh c#	c#		123123	03/04/2024 12:00:00 SA	10/05/2024 12:00:00 SA	Active	Edit Delete
5	tester game	tester game		okok	04/04/2024 12:00:00 SA	11/05/2024 12:00:00 SA	Active	Edit Delete
6	lap trinh adroi	lap trinh mobile		lap trinh	05/04/2024 12:00:00 SA	04/05/2024 12:00:00 SA	Active	Edit Delete
7	lap trinh game	lap trinh c++		lap trinh game	13/04/2024 12:00:00 SA	20/04/2024 12:00:00 SA	Active	Edit Delete

FPT Training BTec FPTPolytechnic © 2024 creativeLabs.

Powered by IT0502

Figure 19 Index Course



Add new Courses page !

Create Course

Name:

Start date: mm/dd/yyyy

Category: duy

End date: mm/dd/yyyy

Description:

Status: Active

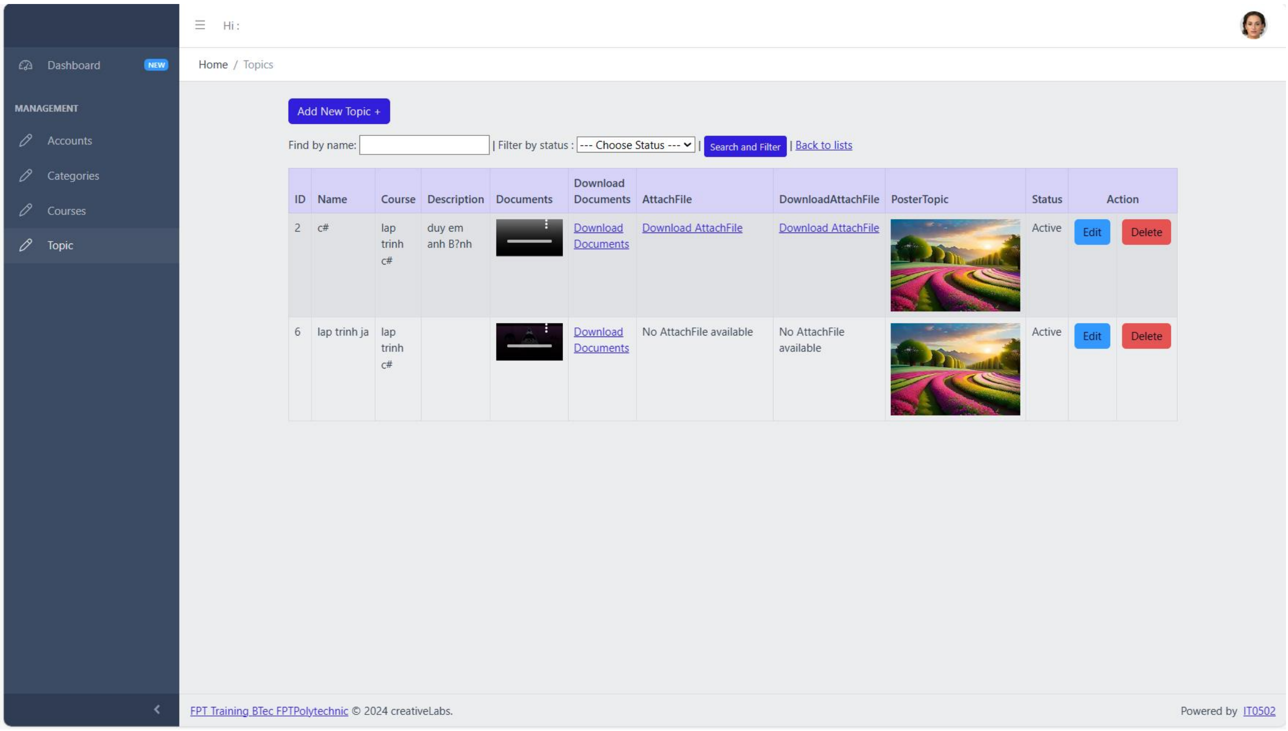
Image: No file chosen

FPT Training BTec FPTPolytechnic © 2024 creativeLabs.

Powered by IT0502

Figure 20 Add Course

1.4.User Topic management.



Home / Topics

Add New Topic +

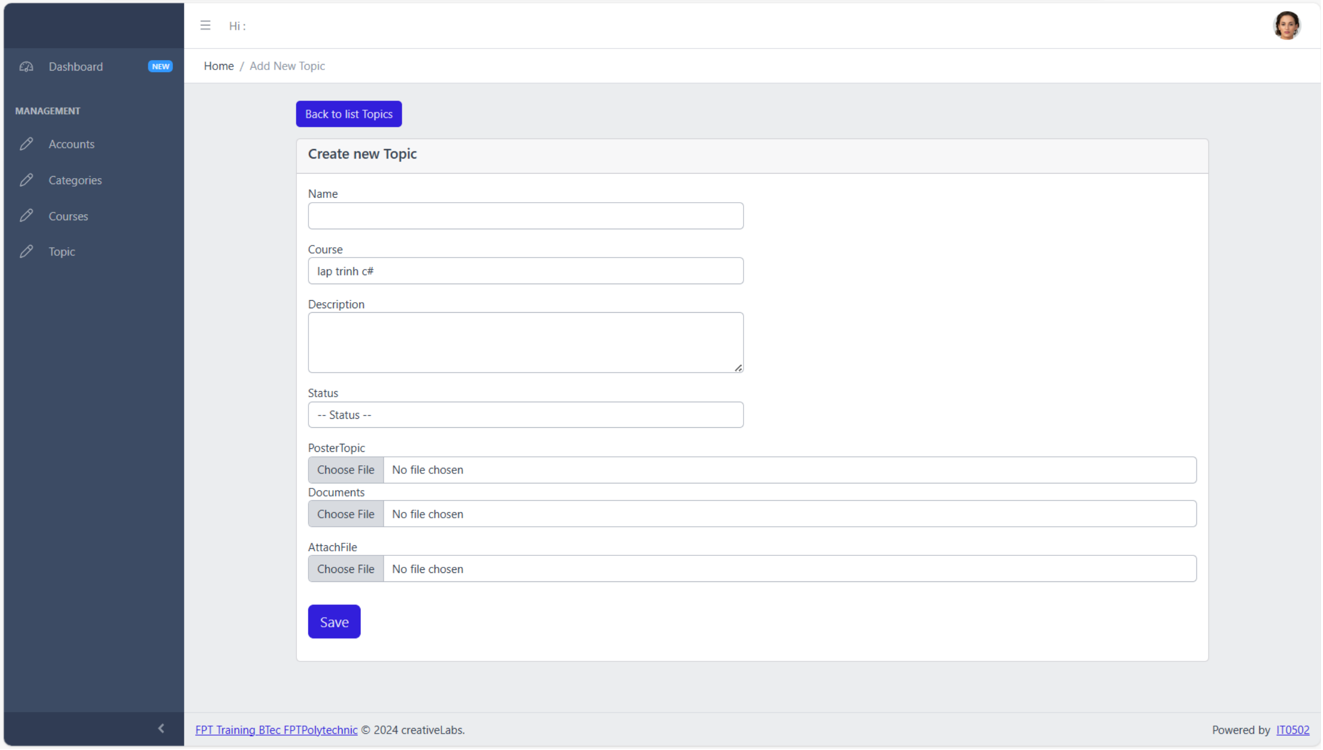
Find by name: | Filter by status : --- Choose Status --- | Search and Filter | Back to lists

ID	Name	Course	Description	Documents	Download Documents	AttachFile	DownloadAttachFile	PosterTopic	Status	Action	
2	c#	lap trinh c#	duy em anh B?nh		Download Documents	Download AttachFile	Download AttachFile		Active	Edit	Delete
6	lap trinh ja	lap trinh c#			Download Documents	No AttachFile available	No AttachFile available		Active	Edit	Delete

FPT Training BTec FPTPolytechnic © 2024 creativeLabs.

Powered by IT0502

Figure 21 Index Topic



Home / Add New Topic

Back to list Topics

Create new Topic

Name

Course: lap trinh c#

Description

Status: -- Status --

PosterTopic: Choose File No file chosen

Documents: Choose File No file chosen

AttachFile: Choose File No file chosen

Save

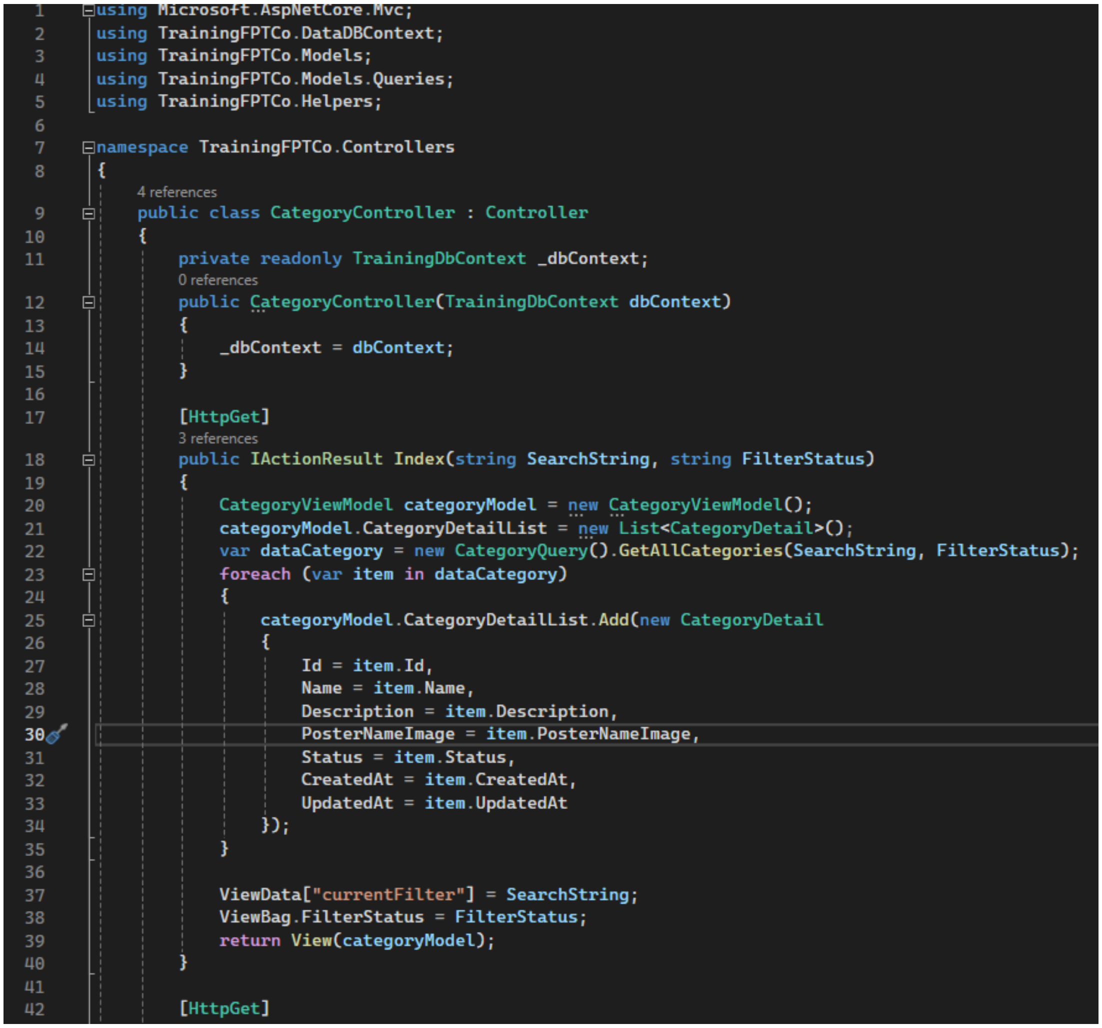
FPT Training BTec FPTPolytechnic © 2024 creativeLabs.

Powered by IT0502

Figure 22 Add Topic

2. Code.

2.1.Category.



```
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);
            foreach (var item in dataCategory)
            {
                categoryModel.CategoryDetailList.Add(new CategoryDetail
                {
                    Id = item.Id,
                    Name = item.Name,
                    Description = item.Description,
                    PosterNameImage = item.PosterNameImage,
                    Status = item.Status,
                    CreatedAt = item.CreatedAt,
                    UpdatedAt = item.UpdatedAt
                });
            }

            ViewData["currentFilter"] = SearchString;
            ViewBag.FilterStatus = FilterStatus;
            return View(categoryModel);
        }

        [HttpGet]
```

Figure 23 Category Controller

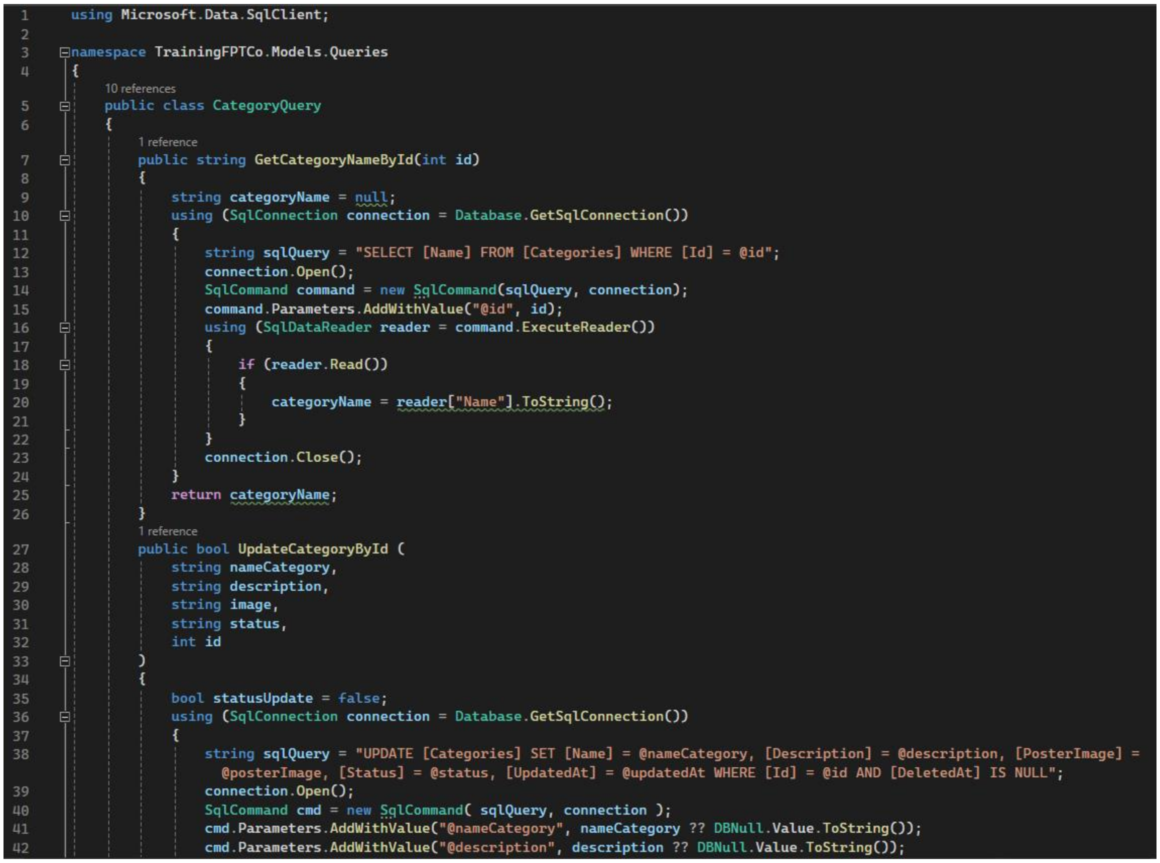
"The CategoryController class serves as a crucial component in the software application's architecture, responsible for overseeing the management and control of various operations pertaining to categories. These operations encompass a broad spectrum of tasks, including the creation, modification, and removal of categories within the application's ecosystem.

By encapsulating these functionalities within the CategoryController, the class provides a centralized point of access and coordination for all category-related activities. This architectural design fosters modularity, scalability, and maintainability, as it segregates the concerns related to category management from other components of the application.

In essence, the CategoryController acts as a guardian of category data integrity and consistency, ensuring that interactions with categories adhere to predefined rules and constraints. Through its cohesive orchestration of category-related actions, the controller plays a pivotal role in shaping the overall user experience and operational efficiency of the application.

Furthermore, the CategoryController serves as a conduit for communication between the user interface and the underlying data persistence layer. It receives input from users or external systems, processes it accordingly, and orchestrates the necessary operations to reflect changes in the category data store.

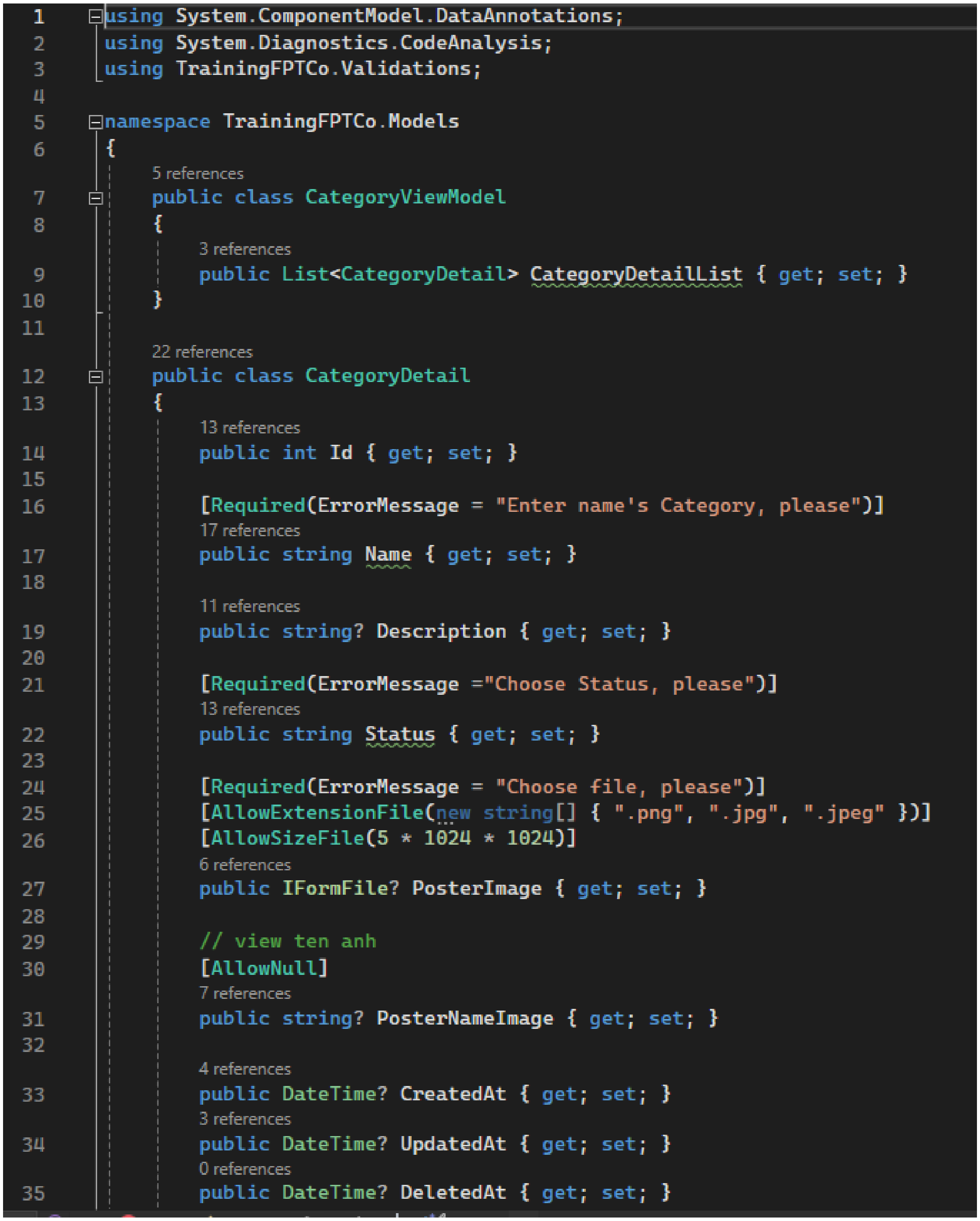
Overall, the CategoryController embodies a fundamental aspect of the application's functionality, empowering users and administrators alike to manage categories seamlessly and efficiently. Its robust design and functionality contribute significantly to the overall effectiveness and usability of the software 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 [Id] = @id";
                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();
            }
            return categoryName;
        }
        1 reference
        public bool UpdateCategoryById (
            string nameCategory,
            string description,
            string image,
            string status,
            int id
        )
        {
            bool statusUpdate = false;
            using (SqlConnection connection = Database.GetSqlConnection())
            {
                string sqlQuery = "UPDATE [Categories] SET [Name] = @nameCategory, [Description] = @description, [PosterImage] =
                  @posterImage, [Status] = @status, [UpdatedAt] = @updatedAt WHERE [Id] = @id AND [DeletedAt] IS NULL";
                connection.Open();
                SqlCommand cmd = new SqlCommand( sqlQuery, connection );
                cmd.Parameters.AddWithValue("@nameCategory", nameCategory ?? DBNull.Value.ToString());
                cmd.Parameters.AddWithValue("@description", description ?? DBNull.Value.ToString());
```

Figure 24 Category Query

The versatile 'CategoryQuery' class serves as a pivotal tool for interrogating and procuring data pertaining to categories. It facilitates an array of operations, including searching for categories, fetching detailed category information, and executing specialized category-oriented inquiries, ensuring seamless management of category-related tasks.

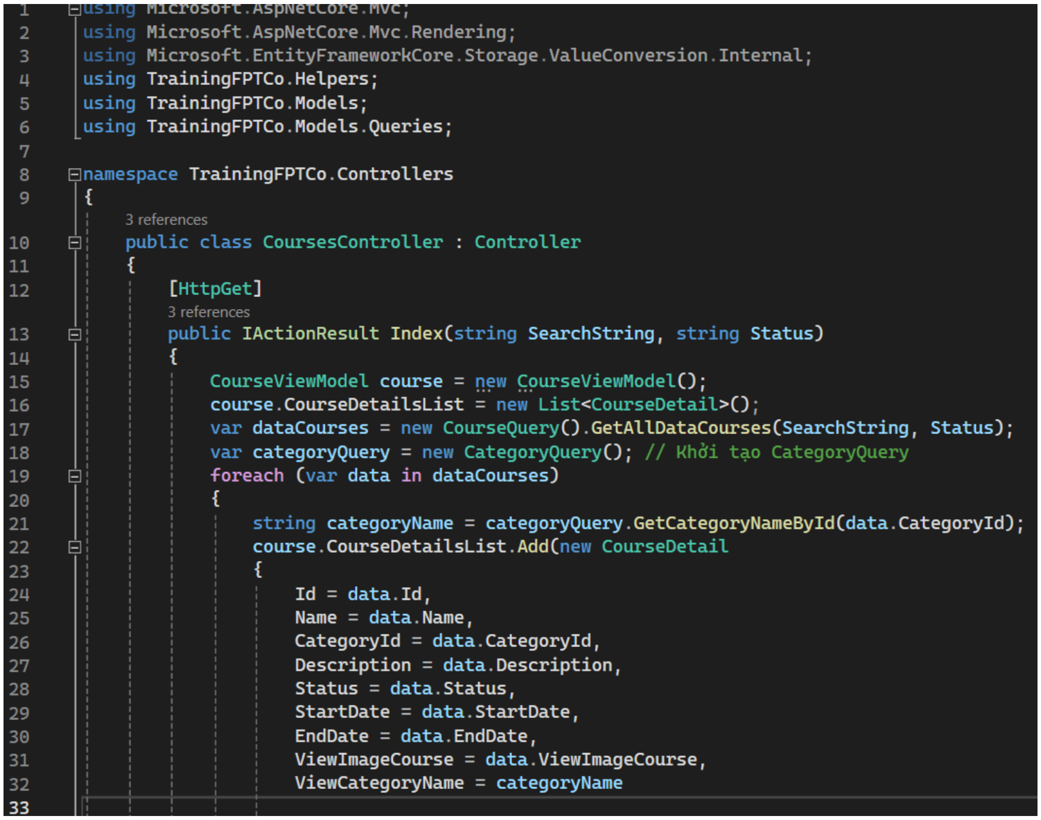


```
using System.ComponentModel.DataAnnotations;
using System.Diagnostics.CodeAnalysis;
using TrainingFPTCo.Validations;

namespace TrainingFPTCo.Models
{
    5 references
    public class CategoryViewModel
    {
        3 references
        public List<CategoryDetail> CategoryDetailList { get; set; }
    }

    22 references
    public class CategoryDetail
    {
        13 references
        public int Id { get; set; }

        [Required(ErrorMessage = "Enter name's Category, please")]
        17 references
        public string Name { get; set; }

        11 references
        public string? Description { get; set; }

        [Required(ErrorMessage ="Choose Status, please")]
        13 references
        public string Status { get; set; }

        [Required(ErrorMessage = "Choose file, please")]
        [AllowExtensionFile(new string[] { ".png", ".jpg", ".jpeg" })]
        [AllowSizeFile(5 * 1024 * 1024)]
        6 references
        public IFormFile? PosterImage { get; set; }

        // view ten anh
        [AllowNull]
        7 references
        public string? PosterNameImage { get; set; }

        4 references
        public DateTime? CreatedAt { get; set; }
        3 references
        public DateTime? UpdatedAt { get; set; }
        0 references
        public DateTime? DeletedAt { get; set; }
```

Figure 25 Category View Model

The indispensable 'CategoryViewModel' class plays a pivotal role in orchestrating the data model and business logic essential for category-centric functionalities within the application ecosystem. Acting as a cornerstone, it facilitates the creation and manipulation of a comprehensive dataset, complemented by a suite of meticulously crafted processing methods. These methods synergistically empower the user interface, facilitating seamless interaction with categories, while ensuring a rich and intuitive display of information. With its robust architecture, the 'CategoryViewModel' class encapsulates the intricacies of category management, thereby fostering an immersive and engaging user experience.

2.2.Course.



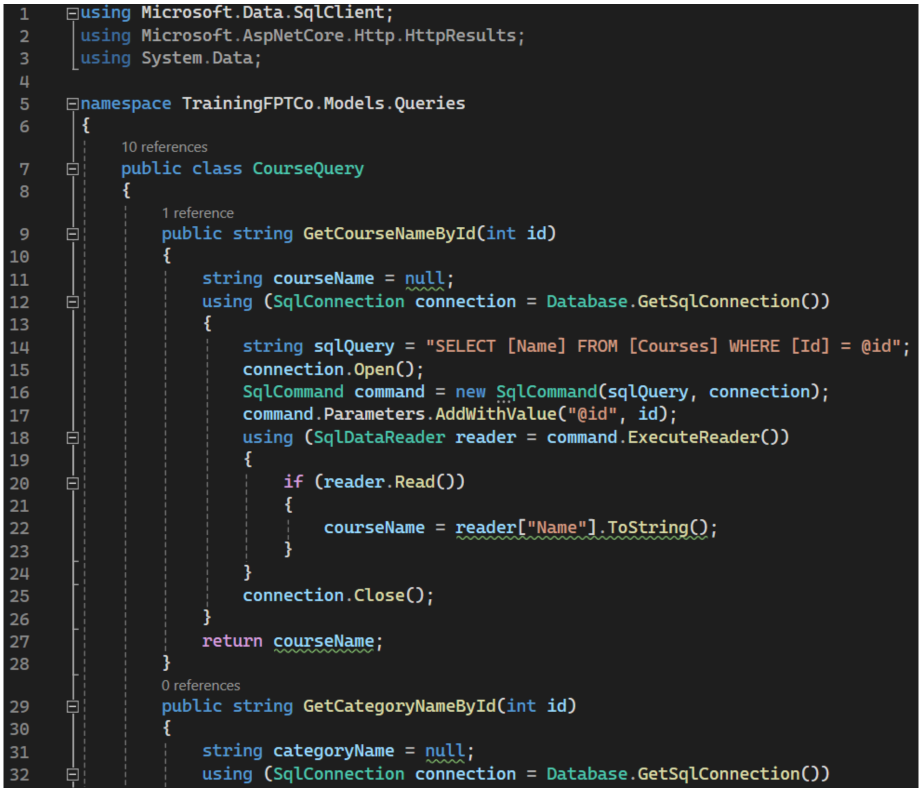
```
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion.Internal;
using TrainingFPTCo.Helpers;
using TrainingFPTCo.Models;
using TrainingFPTCo.Models.Queries;

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(); // Khởi tạo 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,
                    StartDate = data.StartDate,
                    EndDate = data.EndDate,
                    ViewImageCourse = data.ViewImageCourse,
                    ViewCategoryName = categoryName

```

Figure 26 Couse Controller

The pivotal 'CoursesController' class serves as the linchpin for orchestrating and governing the intricate web of functionalities associated with courses within the software ecosystem. It operates as the central hub, meticulously managing the gamut of tasks related to course management, including but not limited to, the retrieval, manipulation, and deletion of course data. With unwavering diligence, the 'CoursesController' class ensures the seamless execution of these operations, navigating through the complexities inherent in course-related interactions with finesse and precision.

Embodying a holistic approach to software design, the 'CoursesController' class not only encapsulates the nuts and bolts of course management but also embodies a commitment to delivering a superlative user experience. By adhering to best practices and industry standards, it fosters a cohesive and intuitive user interface, empowering users to navigate through course-related features effortlessly.

Driven by a relentless pursuit of excellence, the 'CoursesController' class stands as a testament to the software's dedication to facilitating meaningful interactions with course content. With each interaction, it ensures that every action taken aligns seamlessly with the overarching objectives of the application, thereby enriching the user's journey through the digital landscape



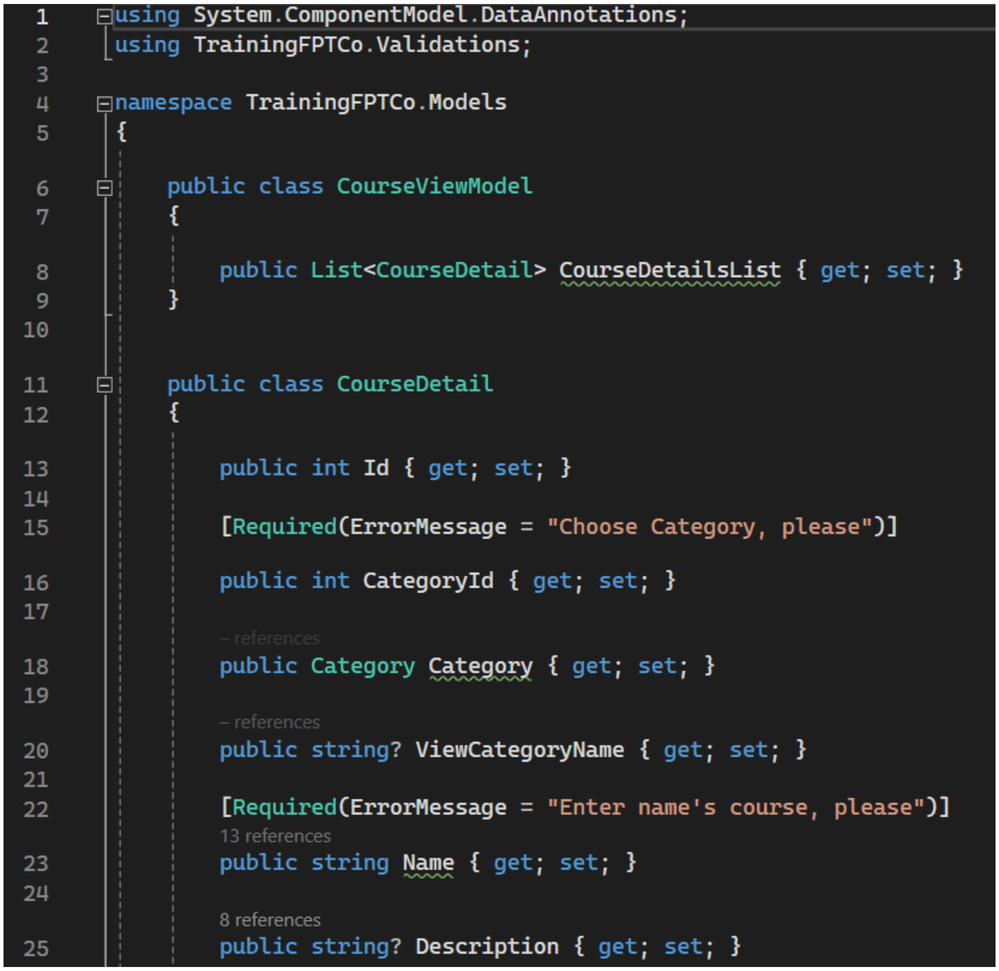
```
using Microsoft.Data.SqlClient;
using Microsoft.AspNetCore.Http.HttpResults;
using System.Data;

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

Figure 27 Course Query

The indispensable 'CourseQuery' class stands as a cornerstone in the realm of course management, dedicated to the meticulous retrieval and manipulation of course-related data within the software ecosystem. Serving as a versatile toolset, it adeptly navigates through the intricacies of course management, facilitating a myriad of essential tasks including searching for courses, extracting detailed course information, and executing specialized course-oriented queries with precision and efficiency.

At the heart of the application's functionality, the 'CourseQuery' class embodies a commitment to seamless user experience, ensuring that every interaction with course-related features is met with responsiveness and accuracy. By harnessing the power of robust querying mechanisms, it empowers users to effortlessly explore and engage with the diverse array of courses available within the system.

With its unwavering dedication to excellence, the 'CourseQuery' class encapsulates the essence of efficient data retrieval and manipulation, embodying best practices and industry standards to deliver an unparalleled user experience. Through its comprehensive suite of functionalities, it not only simplifies course management but also enriches the user's journey by providing timely and relevant access to course information, thus elevating the overall software experience to new heights.



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

namespace TrainingFPTCo.Models
{
    public class CourseViewModel
    {
        public List<CourseDetail> CourseDetailsList { get; set; }
    }

    public class CourseDetail
    {
        public int Id { get; set; }

        [Required(ErrorMessage = "Choose Category, please")]
        public int CategoryId { get; set; }

        – references
        public Category Category { get; set; }

        – references
        public string? ViewCategoryName { get; set; }

        [Required(ErrorMessage = "Enter name's course, please")]
        13 references
        public string Name { get; set; }

        8 references
        public string? Description { get; set; }
```

Figure 28 Course View Model

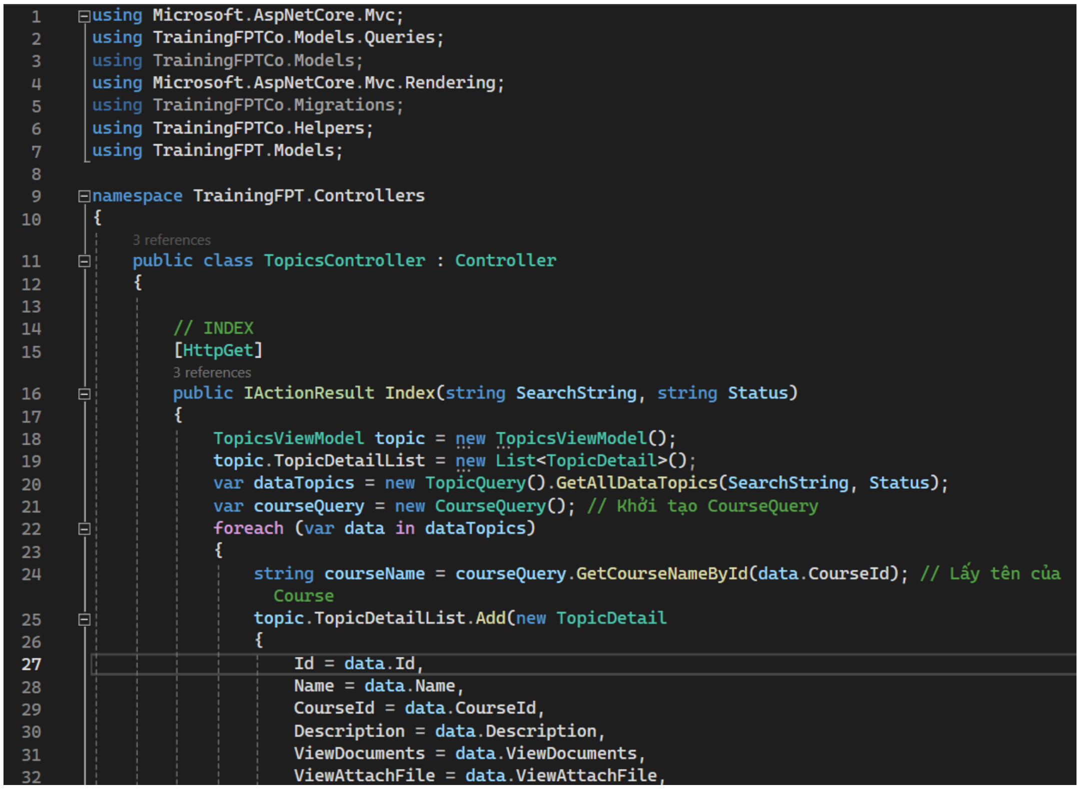
The pivotal 'CourseViewModel' class serves as the backbone of course-related functionalities within the application, embodying a robust combination of data modeling and business logic essential for seamless course management. At its core, it functions as a conduit, meticulously crafting and orchestrating a comprehensive dataset coupled with a suite of meticulously designed processing methods. These methods are strategically engineered to facilitate the seamless display of course information and empower users to interact with courses effortlessly within the user interface.

By encapsulating the intricacies of course management, the 'CourseViewModel' class ensures that every facet of the user's interaction with course-related features is meticulously orchestrated with precision and efficiency. It leverages sophisticated data modeling techniques to curate a rich and intuitive representation of course information, thereby enriching the user experience and fostering meaningful engagement with course content.

Driven by a commitment to excellence, the 'CourseViewModel' class continuously evolves to meet the dynamic demands of course management within the application landscape. Through its versatile array of processing methods and meticulously crafted data model, it empowers users to navigate through course-related functionalities with ease, ensuring a seamless and immersive experience at every turn.

In essence, the 'CourseViewModel' class stands as a testament to the application's dedication to delivering unparalleled user experience, seamlessly bridging the gap between data representation and user interaction to create a cohesive and engaging environment for exploring and engaging with course content.

2.3.Topic.



```
using Microsoft.AspNetCore.Mvc;
using TrainingFPTCo.Models.Queries;
using TrainingFPTCo.Models;
using Microsoft.AspNetCore.Mvc.Rendering;
using TrainingFPTCo.Migrations;
using TrainingFPTCo.Helpers;
using TrainingFPT.Models;

namespace TrainingFPT.Controllers
{
    3 references
    public class TopicsController : Controller
    {

        // INDEX
        [HttpGet]
        3 references
        public IActionResult Index(string SearchString, string Status)
        {
            TopicsViewModel topic = new TopicsViewModel();
            topic.TopicDetailList = new List<TopicDetail>();
            var dataTopics = new TopicQuery().GetAllDataTopics(SearchString, Status);
            var courseQuery = new CourseQuery(); // Khởi tạo CourseQuery
            foreach (var data in dataTopics)
            {
                string courseName = courseQuery.GetCourseNameById(data.CourseId); // Lấy tên của
                  Course
                topic.TopicDetailList.Add(new TopicDetail
                {
                    Id = data.Id,
                    Name = data.Name,
                    CourseId = data.CourseId,
                    Description = data.Description,
                    ViewDocuments = data.ViewDocuments,
                    ViewAttachFile = data.ViewAttachFile,
```

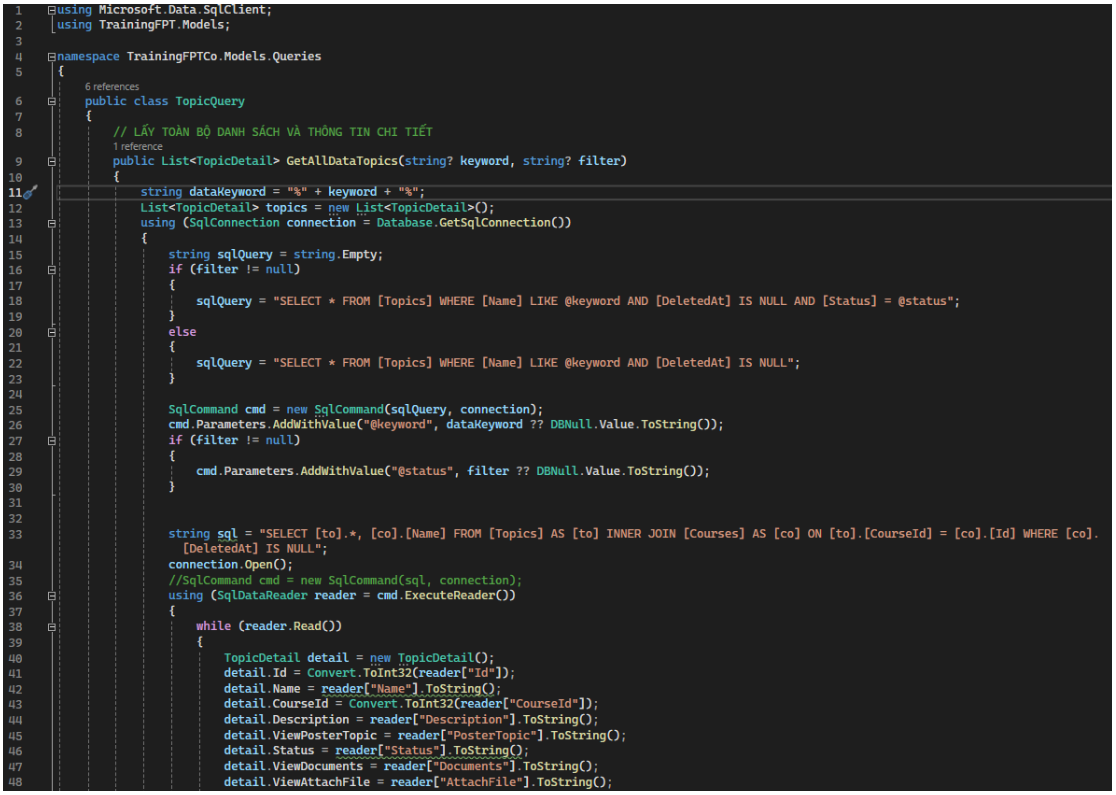
Figure 29 Topic Controller

The pivotal 'TopicController' class assumes a central role in orchestrating and governing the intricacies surrounding topic-related functionalities within software applications. As the steward of topic management, it shoulders the responsibility of creating, editing, and deleting topics with meticulous care and precision.

At its core, the 'TopicController' class serves as a linchpin, ensuring that every interaction with topics within the application landscape is met with appropriate actions and responses. By leveraging its robust set of methods and functionalities, it facilitates seamless navigation through the lifecycle of topics, from inception to modification and eventual removal.

Driven by a commitment to excellence, the 'TopicController' class embodies best practices and industry standards to deliver a superlative user experience. Through its adept management of topic-related tasks, it fosters an environment where users can engage with topics effortlessly, thereby enriching their journey through the application.

In essence, the 'TopicController' class stands as a testament to the application's dedication to effective topic management, ensuring that each interaction with topics is met with responsiveness and efficacy, ultimately contributing to a cohesive and enriching user experience.



```
using Microsoft.Data.SqlClient;
using TrainingFPT.Models;

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());
                if (filter != null)
                {
                    cmd.Parameters.AddWithValue("@status", filter ?? DBNull.Value.ToString());
                }


                string sql = "SELECT [to].*, [co].[Name] FROM [Topics] AS [to] INNER JOIN [Courses] AS [co] ON [to].[CourseId] = [co].[Id] WHERE [co].
                  [DeletedAt] IS NULL";
                connection.Open();
                //SqlCommand cmd = new SqlCommand(sql, connection);
                using (SqlDataReader reader = cmd.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        TopicDetail detail = new TopicDetail();
                        detail.Id = Convert.ToInt32(reader["Id"]);
                        detail.Name = reader["Name"].ToString();
                        detail.CourseId = Convert.ToInt32(reader["CourseId"]);
                        detail.Description = reader["Description"].ToString();
                        detail.ViewPosterTopic = reader["PosterTopic"].ToString();
                        detail.Status = reader["Status"].ToString();
                        detail.ViewDocuments = reader["Documents"].ToString();
                        detail.ViewAttachFile = reader["AttachFile"].ToString();
```

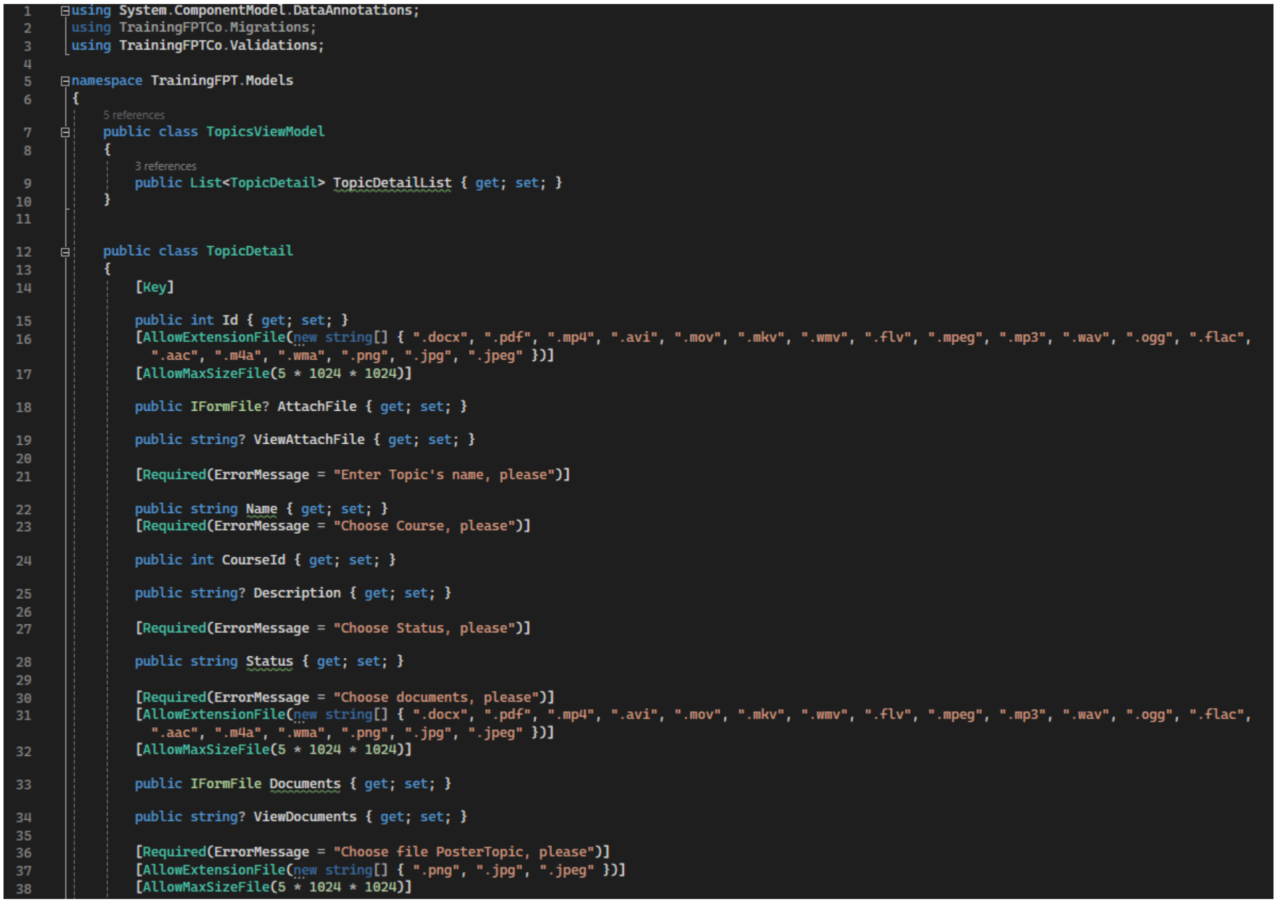
Figure 30 Topic Query

The indispensable 'TopicQuery' class serves as a foundational component in the realm of topic management within software applications. Designed to facilitate seamless access to topic-related information, it acts as a versatile toolset for querying and retrieving data pertinent to topics. Its multifaceted capabilities encompass a spectrum of essential tasks, including searching for topics, extracting detailed topic information, and executing specialized queries tailored to the intricacies of topic management.

At the heart of its functionality, the 'TopicQuery' class embodies a commitment to efficacy and precision, ensuring that every interaction with topics is met with responsiveness and accuracy. By harnessing advanced querying mechanisms, it empowers users to effortlessly explore and engage with topics, fostering a cohesive and intuitive user experience.

With its robust architecture and comprehensive feature set, the 'TopicQuery' class stands as a testament to the application's dedication to effective topic management. By encapsulating best practices and industry standards, it streamlines the process of accessing topic-related information, ultimately enriching the user's journey through the application's landscape.

In essence, the 'TopicQuery' class serves as a cornerstone in the quest for efficient topic management, empowering users to navigate through topics with ease and confidence, while ensuring that each interaction is met with timely and relevant insights.



```
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; }
    }


    public class TopicDetail
    {
        [Key]

        public int Id { get; set; }
        [AllowExtensionFile(new string[] { ".docx", ".pdf", ".mp4", ".avi", ".mov", ".mkv", ".wmv", ".flv", ".mpeg", ".mp3", ".wav", ".ogg", ".flac",
          ".aac", ".m4a", ".wma", ".png", ".jpg", ".jpeg" })]
        [AllowMaxSizeFile(5 * 1024 * 1024)]

        public IFormFile? AttachFile { get; set; }

        public string? ViewAttachFile { get; set; }

        [Required(ErrorMessage = "Enter Topic's name, please")]

        public string Name { get; set; }
        [Required(ErrorMessage = "Choose Course, please")]

        public int CourseId { get; set; }

        public string? Description { get; set; }

        [Required(ErrorMessage = "Choose Status, please")]

        public string Status { get; set; }

        [Required(ErrorMessage = "Choose documents, please")]
        [AllowExtensionFile(new string[] { ".docx", ".pdf", ".mp4", ".avi", ".mov", ".mkv", ".wmv", ".flv", ".mpeg", ".mp3", ".wav", ".ogg", ".flac",
          ".aac", ".m4a", ".wma", ".png", ".jpg", ".jpeg" })]
        [AllowMaxSizeFile(5 * 1024 * 1024)]

        public IFormFile Documents { get; set; }

        public string? ViewDocuments { get; set; }

        [Required(ErrorMessage = "Choose file PosterTopic, please")]
        [AllowExtensionFile(new string[] { ".png", ".jpg", ".jpeg" })]
        [AllowMaxSizeFile(5 * 1024 * 1024)]
```

Figure 31 Topic View Model

The pivotal 'TopicViewModel' class serves as the backbone of theme-related functionalities within the application, embodying a robust combination of data modeling and business logic essential for seamless theme management. At its core, it functions as a conduit, meticulously crafting and orchestrating a comprehensive dataset coupled with a suite of meticulously designed processing methods. These methods are strategically engineered to facilitate the seamless display of theme information and empower users to interact with themes effortlessly within the user interface.

By encapsulating the intricacies of theme management, the 'TopicViewModel' class ensures that every facet of the user's interaction with theme-related features is meticulously orchestrated with precision and efficiency. It leverages sophisticated data modeling techniques to curate a rich and intuitive representation of theme information, thereby enriching the user experience and fostering meaningful engagement with theme content.

Driven by a commitment to excellence, the 'TopicViewModel' class continuously evolves to meet the dynamic demands of theme management within the application landscape. Through its versatile array of processing methods and meticulously crafted data model, it empowers users to navigate through theme-related functionalities with ease, ensuring a seamless and immersive experience at every turn.

In essence, the 'TopicViewModel' class stands as a testament to the application's dedication to delivering unparalleled user experience, seamlessly bridging the gap between data representation and user interaction to create a cohesive and engaging environment for exploring and engaging with theme content.

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 case

Id	Test case name	Description	Actor

TC1	Login	Login successful	Administrator, training staff, trainer, trainee
TC 2	Login	Login failed	Administrator, training staff, trainer, trainee
TC 3	Category manager	Add category	Training staff
TC 4	Category manager	Edit category	Training staff
TC 5	Category manager	Delete category	Training staff
TC 6	Category manager	Search category	Training staff
TC 7	Course manager	Add course	Training staff
TC 8	Course manager	Edit course	Training staff
TC 9	Course manager	Delete course	Training staff
TC11	Course manager	Search course	Training staff
TC12	Topic manager	Add topic	Training staff
TC13	Topic manager	Edit topic	Training staff
TC14	Topic manager	Delete topic	Training staff
TC15	Assign trainer manager	Add assign trainer	Training staff
TC16	Assign trainer manager	Edit assign trainer	Training staff
TC17	Assign trainer manager	Delete assign trainer	Training staff
TC18	Assign trainee manager	Add assign trainee	Training staff

TC19	Assign trainee manager	Edit assign trainee	Training staff
TC20	Assign trainee manager	Delete assign trainee	Training staff
TC21	Logged in	User has session	Administrator, training staff, trainer, trainee
TC22	Logout	User has log out	Administrator, training staff, trainer, trainee

3. Test plan.

ID	Test case name	Description	Test data	Expected	Actual	Remarks
TC1	Login	Login successful	Username:Training staff Password: 123123	Login successful	Login successful	Pass
TC2	Login	Login failed	Username: Training Staff, Password: 5	Login failed	Login failed	Pass
TC3	Category manager	Add category	Name: category5, Status: active	Add successful	Add successful	Pass
TC4	Category manager	Edit category	Name: category2, Status: active	Edit successful	Edit successful	Pass
TC5	Category manager	Delete category	Id = 5	Delete successful	Delete successful	Pass
TC6	Category manager	Search category	Cate8	Display data of category has name: Cate2	Display data of category has name: Cate2	Pass
TC7	Course manager	Add course	Name: course	Add successful	Add successful	Pass

			4, Category: cate003, startDate: 10/09/2023, Status: active			
TC8	Course manager	Edit course	Name: course 2, Category: cate3, startDate: 10/09/2023, Status: active	Edit successful	Edit successful	Pass
TC9	Course manager	Delete course	Id = 20	Delete successful	Delete successful	Pass
TC10	Course manager	Search course	Course10	Display data of course has name: Course1	Display data of course has name: Course1	Pass
TC11	Topic manager	Add topic	Name: topic 1, Course: course2, Document: document.doc Status: active	Add successful	Add successful	Pass
TC12	Topic manager	Edit topic	Name: topic 2,	Edit successful	Edit successful	Pass

			Course: course2, Document: document2.doc Status: active			
TC13	Topic manager	Delete topic	Id = 4	Delete successful	Delete successful	Pass
TC14	Topic manager	Search topic	Topic3	Display data of topic has name: Topic1	Display data of topic has name: Topic1	Pass
TC15	Assign trainer manager	Add assign trainer	Trainer: 2, Course: 2, Status: active	Add successful	Add successful	Pass
TC16	Assign trainer manager	Edit assign trainer	Trainer: 2, Course: 1, Status: active	Edit successful	Edit successful	Pass
TC17	Assign trainer manager	Delete assign trainer	Id = 8	Delete successful	Delete successful	Pass
TC18	Assign trainee manager	Add assign trainee	Trainee: 4, Topic: 2, Status: active	Add successful	Add successful	Pass
TC19	Assign trainee manager	Edit assign trainee	Trainee: 6, Topic: 5, Status: active	Edit successful	Edit successful	Pass
TC20	Assign trainee manager	Delete assign trainee	Id = 4	Delete successful	Delete successful	Pass
TC22	Logged in	User has session		Still login until logout	Logged out after few minute	Fail
TC23	Logout	User has log out		Logout successful	Logout successful	Pass

4. Evaluation

4.1.Strengths:

Exceptional Clarity in Problem Definition: The project demonstrates an exemplary ability to articulate its problem statement, earning a stellar 4 out of 5. This robust articulation lays a solid foundation, ensuring a clear understanding of the project's objectives among stakeholders.

Highly Effective Solutions: Feedback underscores the effectiveness of the proposed solutions, with the majority falling within the "effective to highly effective" range. This validation speaks volumes about the project's ability to address the identified needs efficiently.

Thorough Development Plan: The project's development strategy garners praise for its thoroughness, scoring a commendable 4. This comprehensive plan instills confidence in successful project execution by providing stakeholders with a detailed roadmap for implementation.

Appropriate Design and Functionality: The application's design and functionality are well-suited to its intended purpose and user experience, garnering favorable level 4 ratings. While commendable, there's an opportunity to further optimize these aspects for enhanced usability and overall user satisfaction.

4.2.Weaknesses:

GUI Complexity: User feedback highlights navigational challenges and a lack of user-focused design, resulting in ratings ranging from 3-4. Addressing these areas of concern is critical to improving user satisfaction and engagement, ultimately enhancing the application's usability.

Pending Account Management and Permissions: The absence of account management and permission functionalities represents a significant gap in the project's development. Addressing this weakness is paramount to bolstering security measures and granting users greater control over their accounts.

Lack of Interface Refinement: The user interface lacks significant improvements, detracting from its visual appeal and user-friendliness. This deficiency may hinder user adoption and satisfaction, underscoring the need for interface enhancements to elevate the overall user experience.

4.3.Opportunities for Improvement:

Iterative Solution Refinement: Embrace a feedback-driven approach to refine proposed solutions iteratively, leveraging user insights to optimize effectiveness continually.

Strategic Growth Optimization: Review and refine the growth strategy to identify areas for improvement, fostering efficiency and innovation in development methods to drive sustained project success.

Enhanced Interface Design: Prioritize interface enhancements focused on intuitive navigation and user-centered design, leveraging visual elements to create an immersive and engaging user experience.

Feedback-Driven Development Iteration: Establish a robust feedback loop that integrates iterative testing and flexible feedback mechanisms, ensuring continuous alignment with evolving user expectations.

Comprehensive Account Management Integration: Incorporate robust account management and permission functionalities to enhance security and user control, addressing a critical aspect of the project's development and fortifying its integrity.

C. Conclusion.

In conclusion, our journey towards developing this business application has been marked by meticulous planning, strategic implementation, and relentless pursuit of excellence. We have meticulously navigated through the complexities of the development process, leveraging cutting-edge technologies and methodologies to bring our vision to fruition.

Through collaborative efforts and an unwavering commitment to quality, we have successfully crafted a robust and functional application that addresses the diverse needs of our stakeholders. While we celebrate our achievements, we also recognize that there is always room for improvement.

Looking ahead, we envision a future where our application continues to evolve and adapt to meet the ever-changing demands of the business landscape. We remain steadfast in our commitment to innovation, continuously seeking new opportunities to enhance our solution and deliver even greater value to our clients.

As we embark on the next phase of our journey, we do so with confidence and determination, knowing that our efforts will continue to make a meaningful impact in the world of business applications.

D. Reference

Link google from: https://forms.gle/Pw1VrFYtyNrQJKJn6

Link gitHub: https://github.com/khanh052003/TraingFPTCo.git