**ASSIGNMENT 2 FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **BTEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | Unit 30: Application Development | | |
| **Submission date** | 04/01/2019 | **Date Received 1st submission** | 11/19/2019 |
| **Re-submission Date** |  | **Date Received 2nd submission** | 11/19/2019 |
| **Student Name** | Nguyen Trung Duc | **Student ID** | GCH17377 |
| **Class** | GCH0705 | **Assessor name** | Truong Cong Doan |
| **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** |  |

**Grading grid**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P4 | P5 | P6 | M3 | M4 | M5 | D2 | D3 |
|  |  |  |  |  |  |  |  |

# 

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

Table of Contents

[List of tables: 3](#_Toc28982284)

[List of figures: 4](#_Toc28982285)

[Introduction: 5](#_Toc28982286)

[I Peer review 5](#_Toc28982287)

[Formal questionanarie: 5](#_Toc28982288)

[Feedback analysis 7](#_Toc28982289)

[Suggestions and possible improvement 12](#_Toc28982290)

[II Application development 12](#_Toc28982291)

[2.1 Current system interface 12](#_Toc28982292)

[2.2 Test case: 25](#_Toc28982293)

[2.2.1 Admin function testing: 25](#_Toc28982294)

[2.2.2 Trainer function testing: 26](#_Toc28982295)

[2.2.3 Trainee function testing: 31](#_Toc28982296)

[III Application review 32](#_Toc28982297)

[3.1 Comparison to initial requirements 32](#_Toc28982298)

[3.1.1 Functional features: 32](#_Toc28982299)

[3.1.2 Non-functional features: 34](#_Toc28982300)

[3.2 Assessment of the system’s strength and weakness: 34](#_Toc28982301)

[3.3 Indentified risks 35](#_Toc28982302)

[Conclusion 36](#_Toc28982303)

[References: 36](#_Toc28982304)

# List of tables:

[Table 1 Admin login function test case 25](#_Toc29020553)

[Table 2 Adding trainer function test case 26](#_Toc29020554)

[Table 3Manage trainer function test case 26](#_Toc29020555)

[Table 4 Trainer Login funtion test case 27](#_Toc29020556)

[Table 5 Adding trainee function test case 28](#_Toc29020557)

[Table 6 Manage trainee functions test case 29](#_Toc29020558)

[Table 7 Search trainee funtion test case 29](#_Toc29020559)

[Table 8 Manage category and course function test case 30](#_Toc29020560)

[Table 9 Adding a new class function test case 31](#_Toc29020561)

[Table 10Trainee Login function 31](#_Toc29020562)

[Table 11Enroll for a new class funtion test case 32](#_Toc29020563)

[Table 12 Compare functions of admin with initial requirement 32](#_Toc29020564)

[Table 13 Compare functions of trainer with initial requirement 33](#_Toc29020565)

[Table 14 Compare functions of trainee with initial requirement 34](#_Toc29020566)

[Table 15 Compare non-functions with initial requirement 34](#_Toc29020567)

[Table 16 Risks to website after launch 36](#_Toc29020568)

# List of figures:

[Figure 1 result diagram of question 1 7](#_Toc29019859)

[Figure 2result diagram of question 2 8](#_Toc29019860)

[Figure 3result diagram of question 3 8](#_Toc29019861)

[Figure 4 result diagram of question 4 8](#_Toc29019862)

[Figure 5 result diagram of question 5 9](#_Toc29019863)

[Figure 6 result diagram of question 6 9](#_Toc29019864)

[Figure 7result diagram of question 7 10](#_Toc29019865)

[Figure 8result diagram of question 8 10](#_Toc29019866)

[Figure 9 result diagram of question 9 11](#_Toc29019867)

[Figure 10 result diagram of question 10 11](#_Toc29019868)

[Figure 11 general home page interface 12](#_Toc29019869)

[Figure 12 admin login form interface 13](#_Toc29019870)

[Figure 13 Home page of admin 14](#_Toc29019871)

[Figure 14 adding new trainer inteface 14](#_Toc29019872)

[Figure 15 Trainer management interface 15](#_Toc29019873)

[Figure 16 Update trainer interface 15](#_Toc29019874)

[Figure 17 Trainer login interface 16](#_Toc29019875)

[Figure 18 trainer home page interface 17](#_Toc29019876)

[Figure 19 Adding trainee form interface 17](#_Toc29019877)

[Figure 20 Trainee adding form interface 17](#_Toc29019878)

[Figure 21 detail trainee profile interface 18](#_Toc29019879)

[Figure 22trainee updating interface 19](#_Toc29019880)

[Figure 23 course management interface 19](#_Toc29019881)

[Figure 24 adding category interface 20](#_Toc29019882)

[Figure 25 Adding course interface 20](#_Toc29019883)

[Figure 26 Class management interface 21](#_Toc29019884)

[Figure 27 Detail class interface 21](#_Toc29019885)

[Figure 28 Class adding interface 22](#_Toc29019886)

[Figure 29Trainee login form interface 22](#_Toc29019887)

[Figure 30Trainee home page interface 23](#_Toc29019888)

[Figure 31 Class viewing for trainee interface 23](#_Toc29019889)

[Figure 32Detail class interface 24](#_Toc29019890)

[Figure 33 Class registering interface 24](#_Toc29019891)

# Introduction:

In assignment 1, our team discussed and made decisions about the appropriate technologies and methods that will be applied in the FPT co (CMS) system. It is a blueprint that includes sections such as requirements, existing technologies and methods for managing and developing a cms system and, ultimately, an objective evaluation of the selection, to make the most appropriate decision.

 We have built a system based on the plan given earlier. In this section, we act as testers, perform system performance tests against initial requirements, conduct a small survey by the programmers, and evaluate the strengths. weaknesses as well as giving solutions.

# I Peer review

## Formal questionanarie:

The following is a set of questions for survey research. Our target audience is IT students (They are technologically savvy people, aged 18-25). The questionnaire includes both quantitative and qualitative questions. We hope this study will gain the necessary evaluation information for our system.

1. If you are to rate the cleanliness of the code, what will it be?

a.1\*

b.2\*

c.3\*

d.4\*

e.5\*

2. On a scale of 1 to 5 how will you rate the web interface?

a.1\*

b.2\*

c.3\*

d.4\*

e.5\*

3. On a scale of 1 to 5 how will you rate the web features?

a.1\*

b.2\*

c.3\*

d.4\*

e.5\*

4. Presentation Skill

a.1\*

b.2\*

c.3\*

d.4\*

e.5\*

5. Personal recommendation

6. What system features need improvements?

7. Problems needed to changed when selecting tools?

8. Is the scope and purpose of the system reasonable?

a. Yes

b. No

9. Evaluation of the display of the group on a scale of 1 - 5.

a.1\*

b.2\*

c.3\*

d.4\*

e.5\*

10. Is the SRS Good enough?

a. Yes

b. No

c. May be

## Feedback analysis

Here are the results we have obtained after conducting survey research:

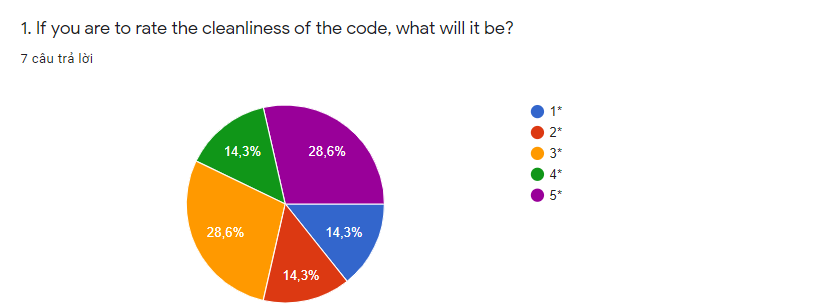


Figure result diagram of question 1

The survey results show positive feedback from the audience for the cleanliness of code when 28.6 percent voted for 5 stars and 14.3 percent for four stars. The low rating of design is only about 14.3 percent choosing 1 or 2 stars. As such, the interface of the website is relatively satisfactory for the interviewees and it is necessary to continue to make improvements to the code to be cleaner.

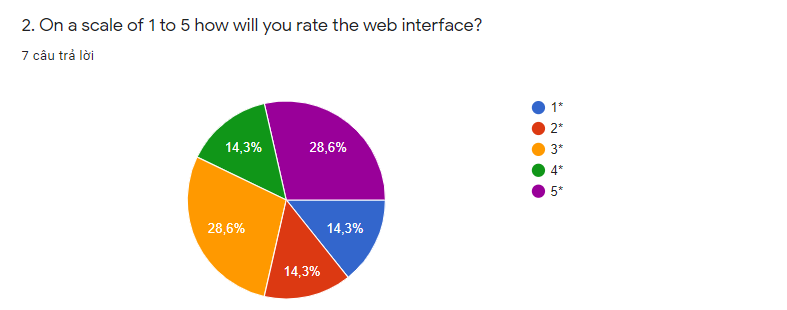


Figure result diagram of question 2

The survey's results showed positive feedback from the audience to the interface of the website when 28.6 percent voted for 5 stars and 14.3 percent for four stars. The low rating of design is only about 14.3 percent choosing 1 or 2 stars. Thus, the interface of the website was relatively pleasing to the interviewees

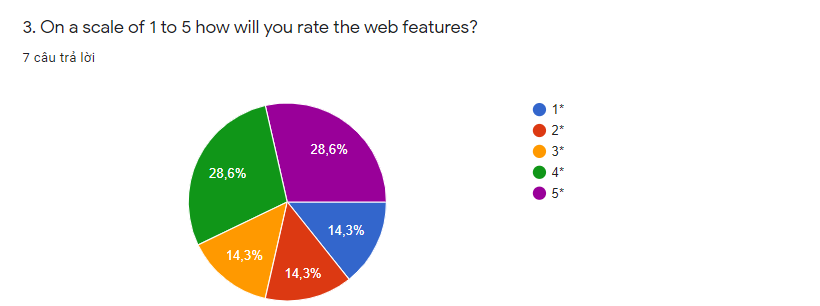


Figure result diagram of question 3

Data from the graph shows that the majority of respondents feel satisfied with the features available on the website when 28.6 percent voted 4 or 5 stars for this section. The average or low rating rate is only 14.3 percent.

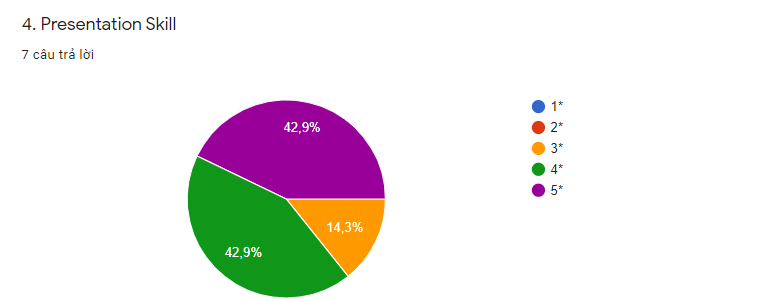


Figure result diagram of question 4

A very positive number from the respondents when they voted 4 or 5 stars for the presentation skills of the group (up to 42.9 percent voted for each type). And only 14.3 percent rated the average.

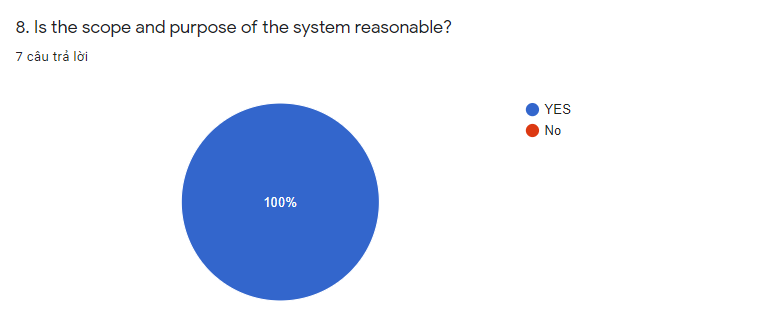


Figure result diagram of question 5

The graph above shows that 100 percent of the audience agree with the view that the scope and purpose of the system is appropriate and in the right direction.

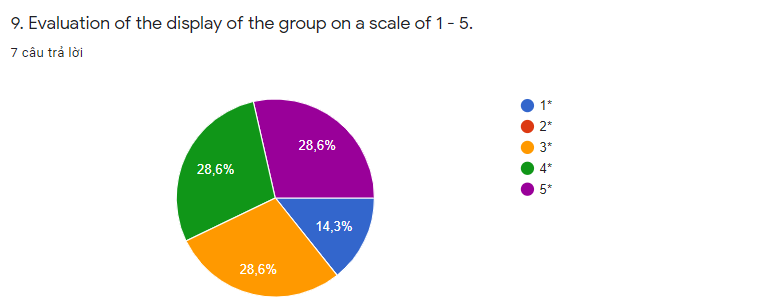


Figure result diagram of question 6

A graph showing the members' productivity shows inequality. 28.6 percent rated the effectiveness of teamwork when voting 5 stars or 4 stars. Similarly, they voted 28.6 percent for an average of 3 stars. The low rate of evaluation for teamwork efficiency is only 14.3 percent.

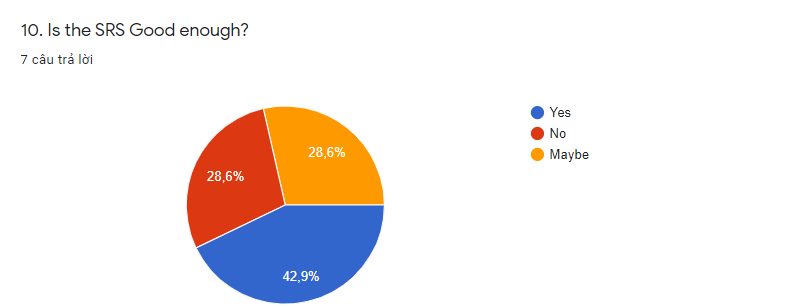


Figure result diagram of question 7

The survey results show that 42.9 percent of the audience agree that the system has fully met the initial requirements. This is considered a fairly low rate. 57.1 percent of the audience were confused or disagreed with the system's ability to meet initial requirements.



Figure result diagram of question 8

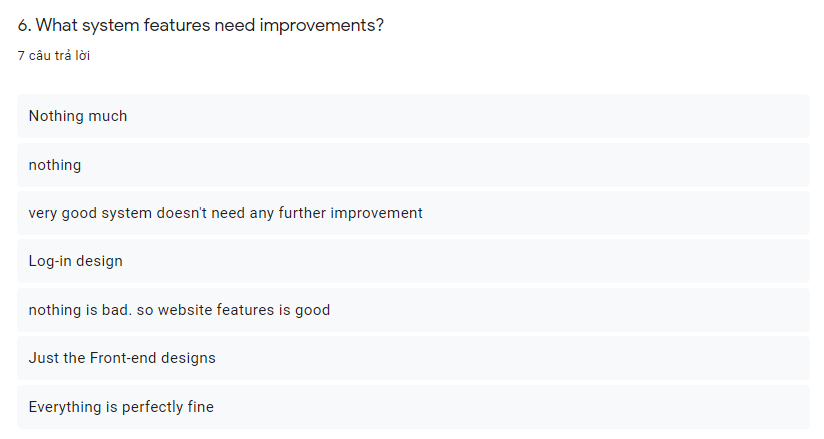


Figure result diagram of question 9

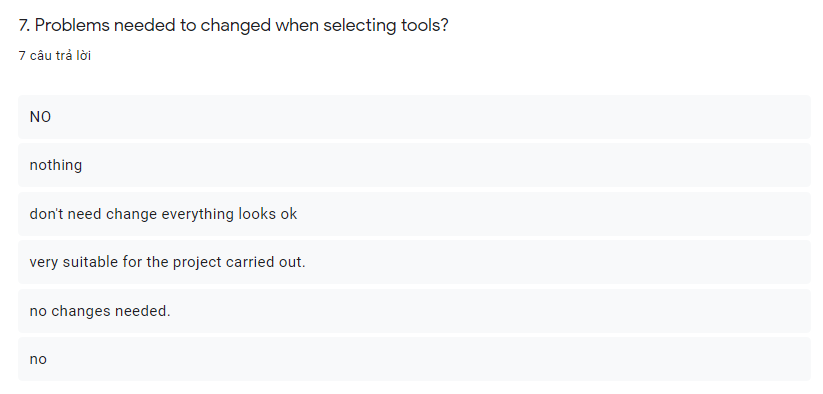


Figure result diagram of question 10

The majority of the audience agreed with the current system but did not want to change. However, some of them still want to change the look of the website.

## Suggestions and possible improvement

After performing the analysis from the results of the survey, we realized things that need to be fixed or improved in our system such as:

* Interface design requires improvements or changes to become more beautiful and easier to use.
* The function code in the system should be handled more cleanly by splits for reuse.
* Continue to add missing features compared to the initial requirements set for the system.
* There are improvements in teamwork skills to be more effective in projects.

# II Application development

## 2.1 Current system interface

The following is the educational management system for FPT employees. The interface is divided into 3 parts with specific roles: admin, trainer, trainee. To sign in with admin role, user click on admin region to go to admin login form. It also is same with trainer and trainee.



Figure general home page interface

Admin:

Login function: This is admin login interface, if you have an admin account with username and password. Please, enter username and password on right place and submit by login button to go to home page of admin.

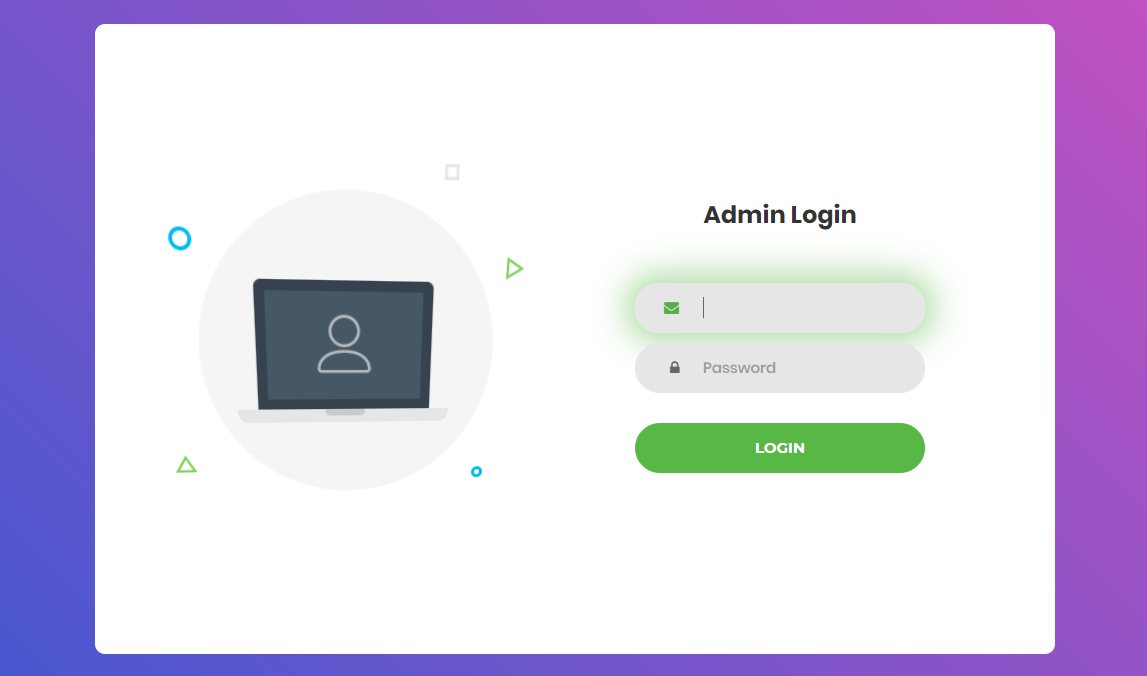


Figure admin login form interface

Home page of admin: In admin home page interface has a nav bar to control every tasks of admin. So, pressing add trainer button to go to adding trainer form. And if you want to detele or update trainer profile, please press manage trainer button. To log out press button under the name of admin.

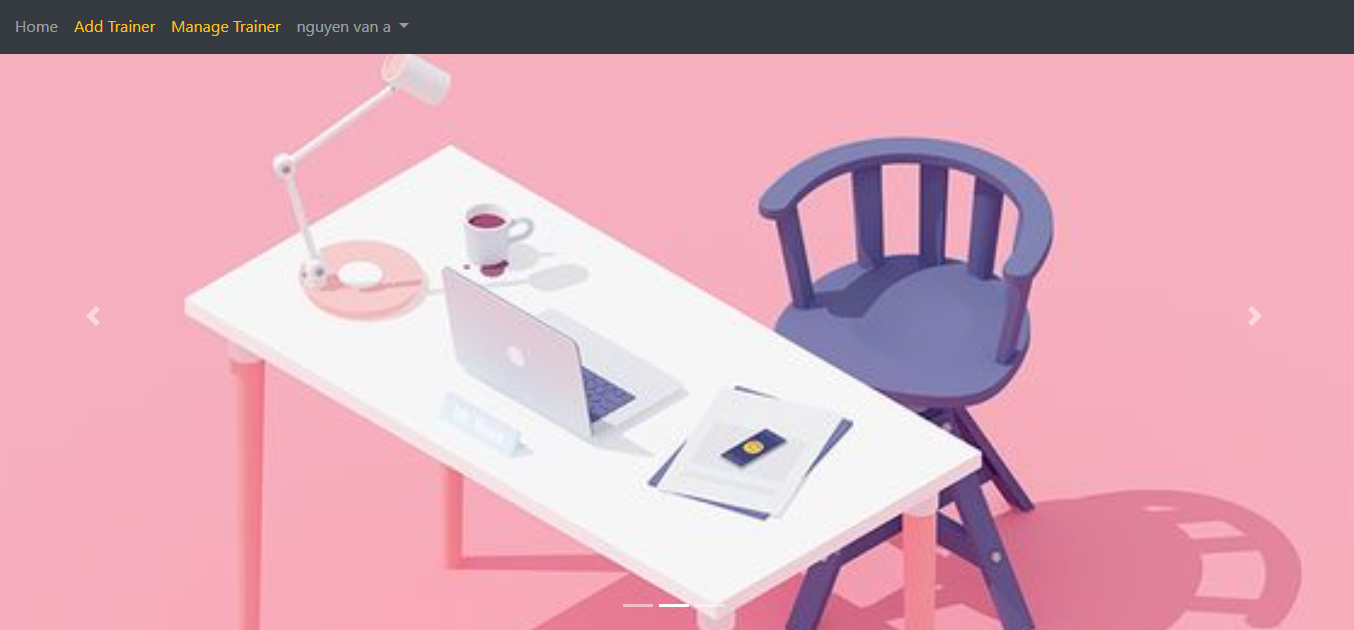


Figure Home page of admin

Adding new trainer account: the following picture is adding trainer form interface. In here, after you finish all fileds like as name of trainer, username and password, you can register a trainer account by press register button.

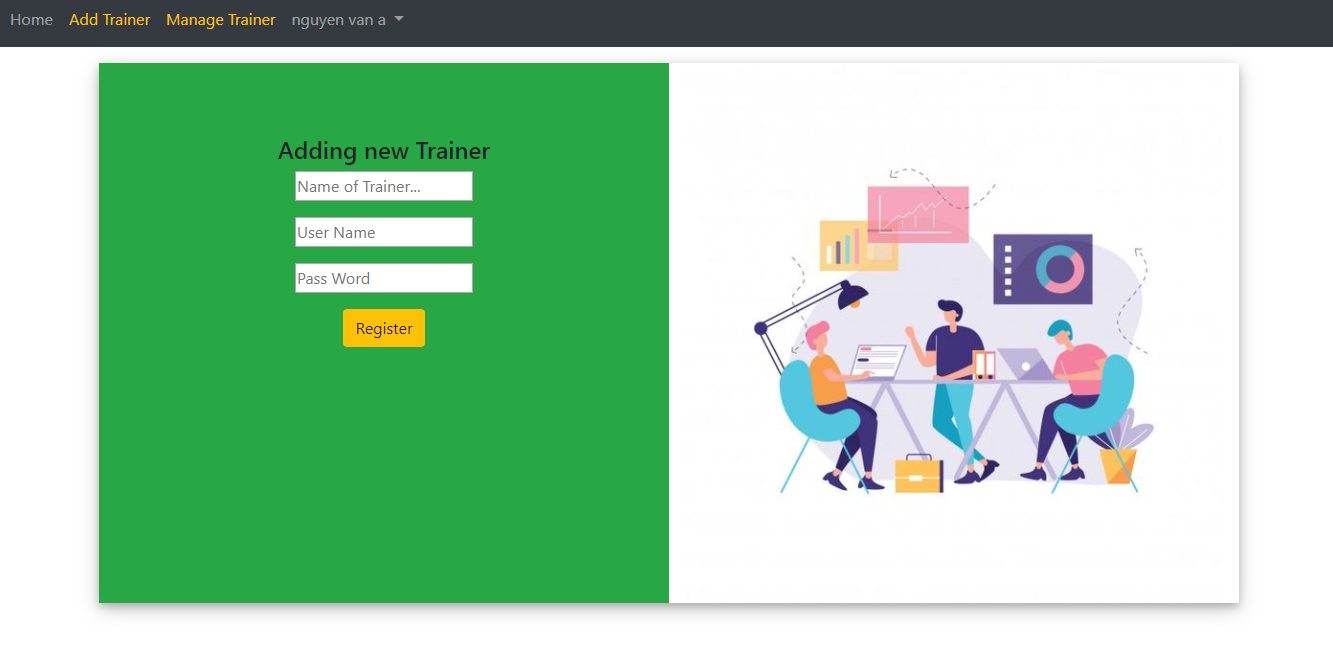


Figure adding new trainer inteface

Management Trainer account: This is the interface for manage all information of trainer. To delete a record, pressing on delete button for each record. And if you want to modify trainer profil, pressing update button.

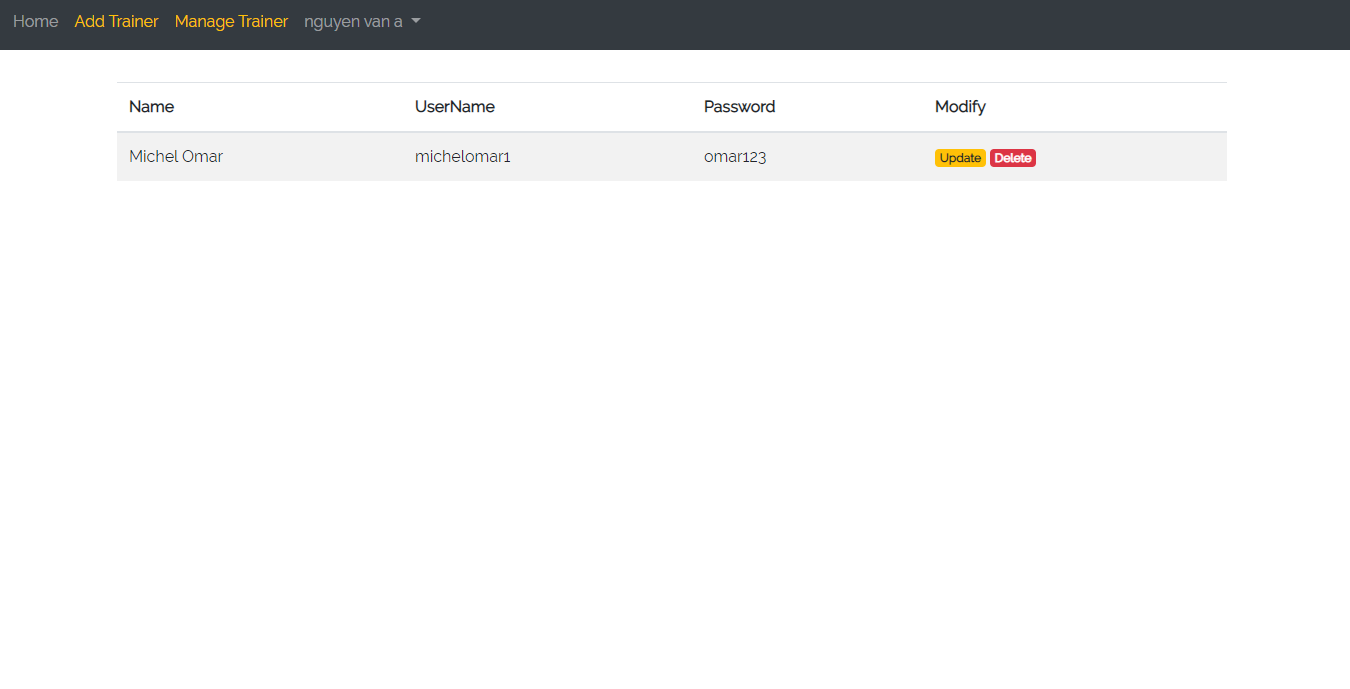


Figure Trainer management interface

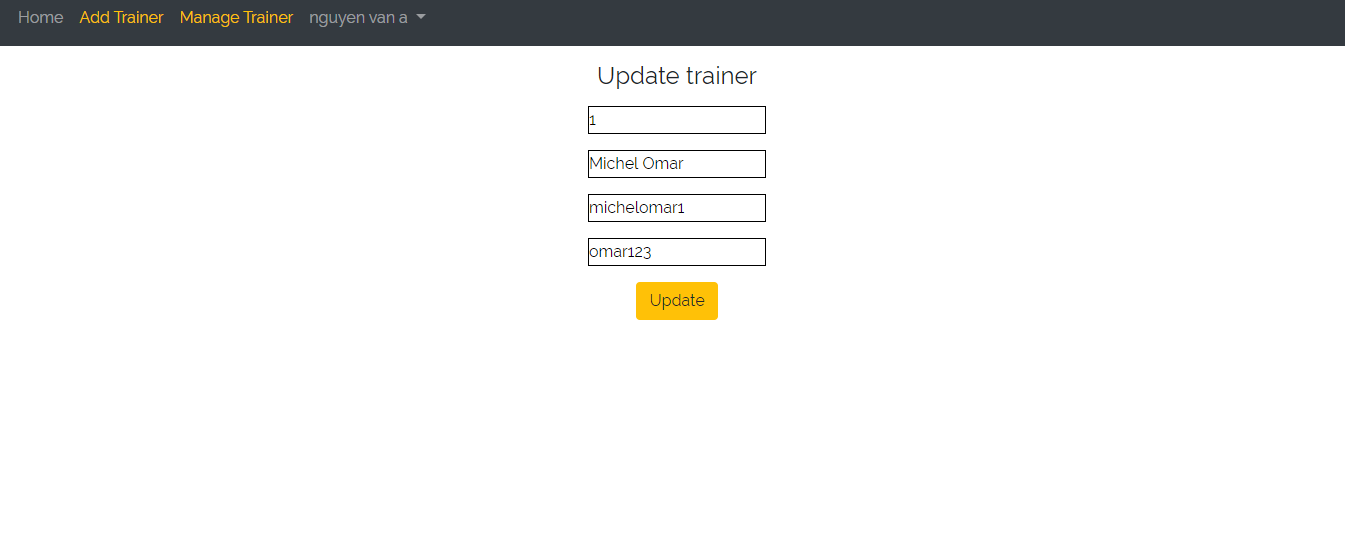


Figure Update trainer interface

After you pressed update button, the interface to update profile will appear. In here, if you want to edit some information, please remind that press update button when everything is done.

Trainer:

Login form of trainer: This is triner login interface, if you have a trainer account with username and password. Please, enter username and password on right place and submit by login button to go to home page of trainer.

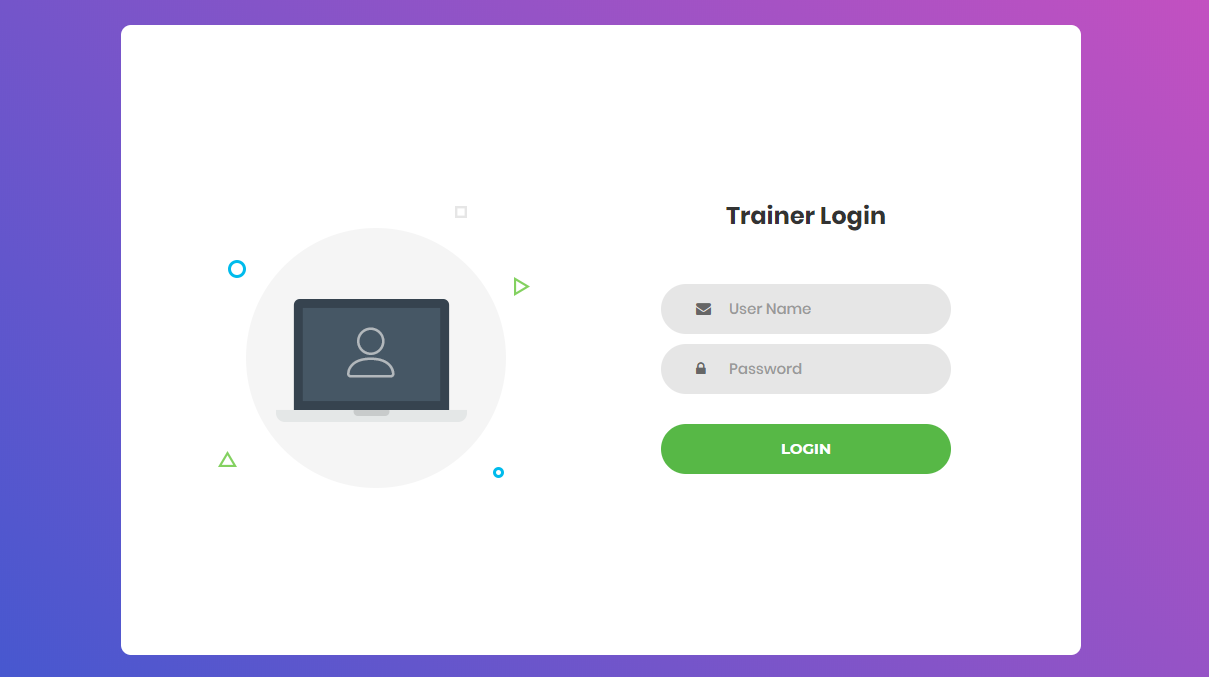


Figure Trainer login interface

Home page of trainer: In trainer home page interface has a nav bar to control every tasks of trainer. So, pressing manage trainer button to go to adding trainer form or edit profile. And if you want to add, detele or update category, course, class, please press manage course button. To log out press button under the name of admin.



Figure trainer home page interface

Adding new Trainee account page: the following picture is adding trainee form interface. In here, after you finish all fileds like as Id, name of trainer, date of birth, language, toeic, username and password, you can register a trainer account by press register button. So, please remind that Id, Name, Username and Password is required.

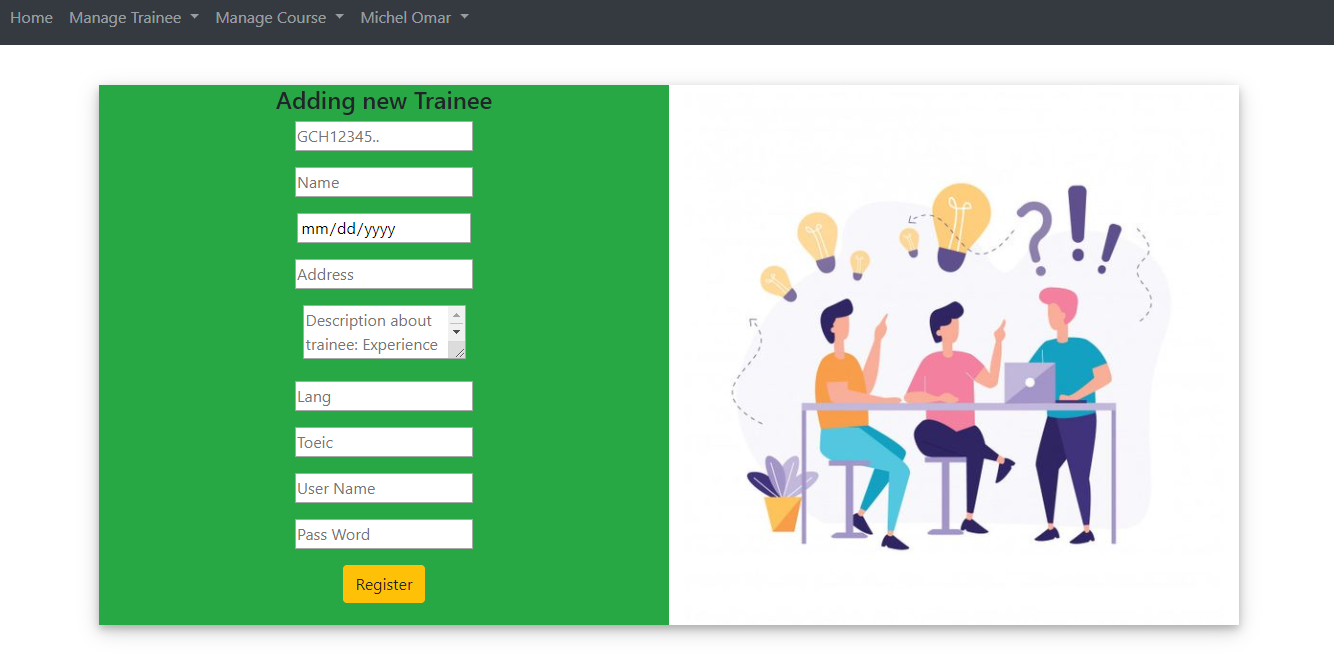


Figure Adding trainee form interface

Management Trainee page: This is the interface for manage all information of trainee. To delete a record, pressing on delete button for each record. And if you want to modify trainer profil, pressing update button. The view button to show more detail trainee profile.

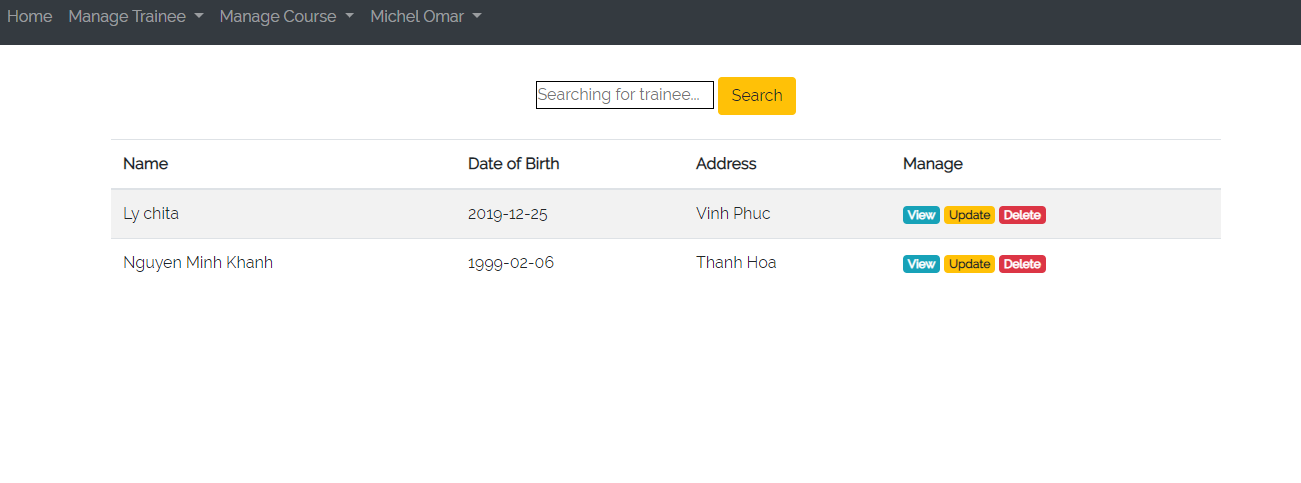


Figure Trainee adding form interface

View detail trainee profile page: This is interface of detail traiee profile. In here, you can see more detail of trainee account.

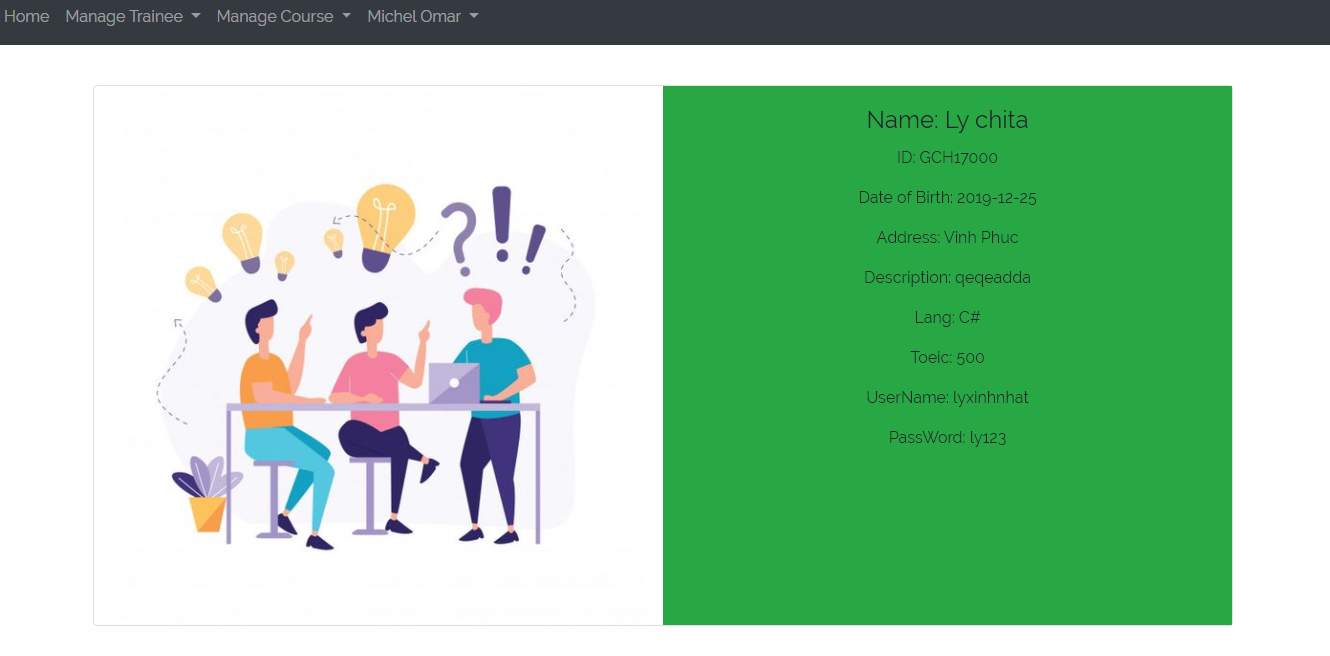


Figure detail trainee profile interface

Update trainee profile page: After you pressed update button, the interface to update profile will appear. In here, if you want to edit some information, please remind that press update button when everything is done.

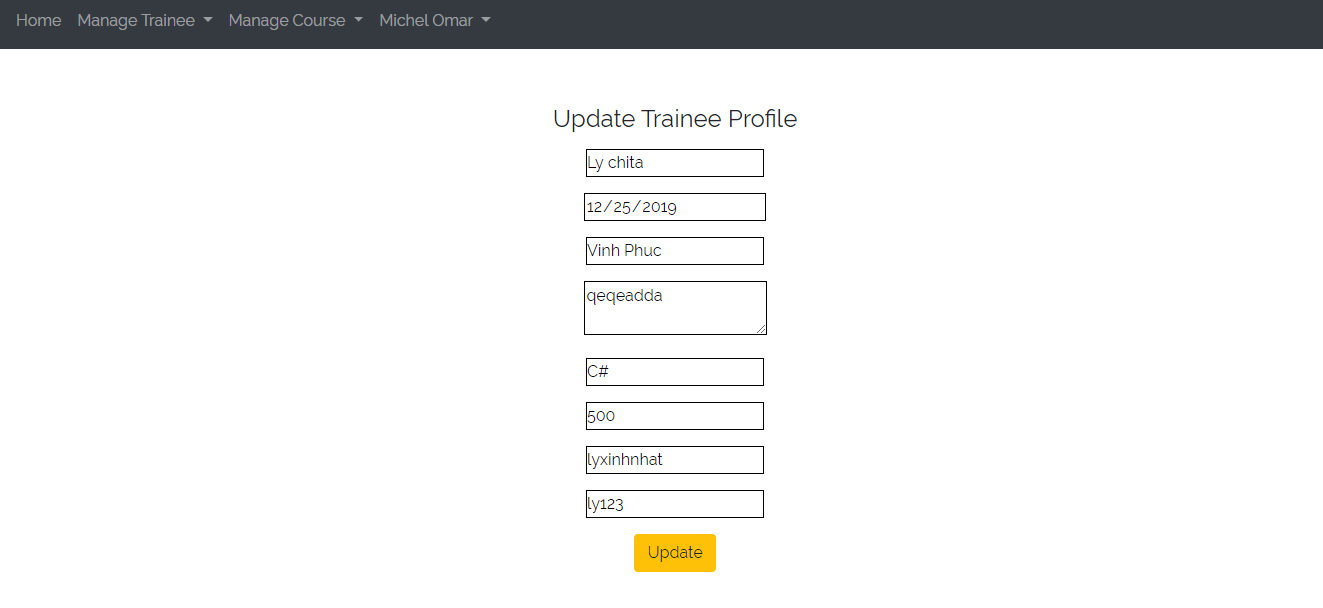


Figure trainee updating interface

Management page for categories, courses: This is interface for course management. The available categories are display on screen. If you want to adding new category, pressing add new category button. The add button on each category for adding new course in category. It’s same with delete button.

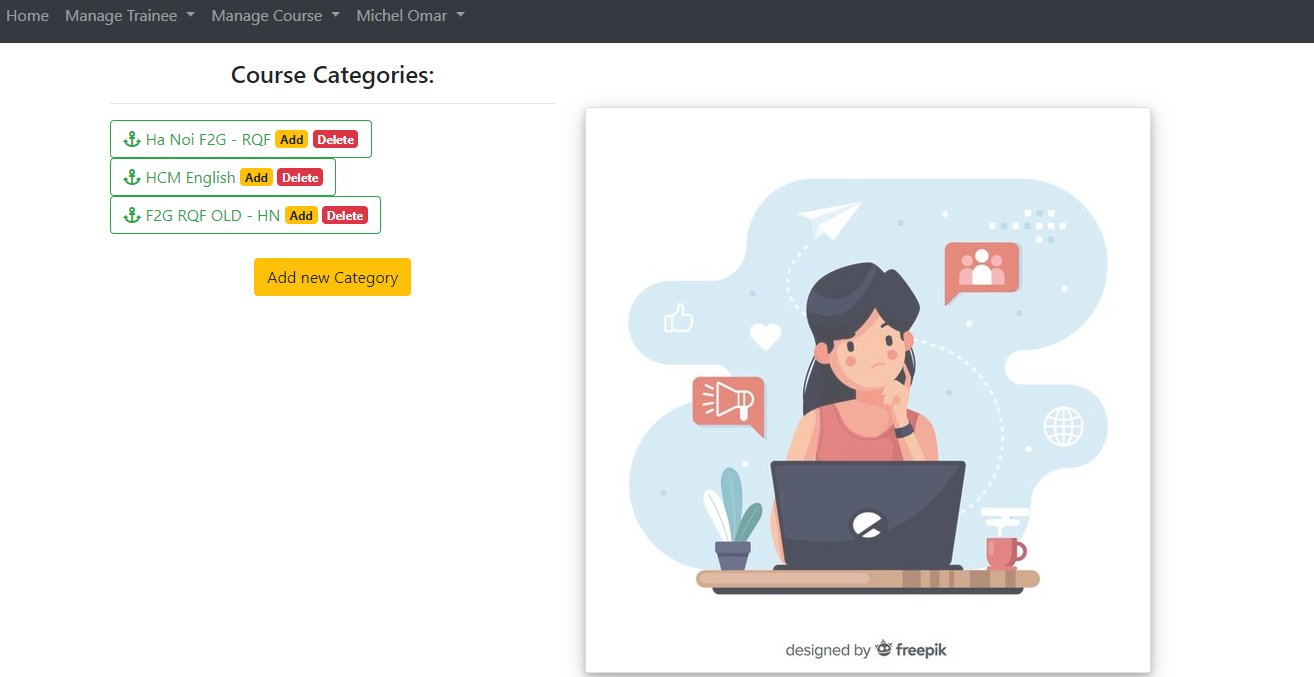


Figure course management interface

Adding new category: To add new category, fill in a category and submit on create button.

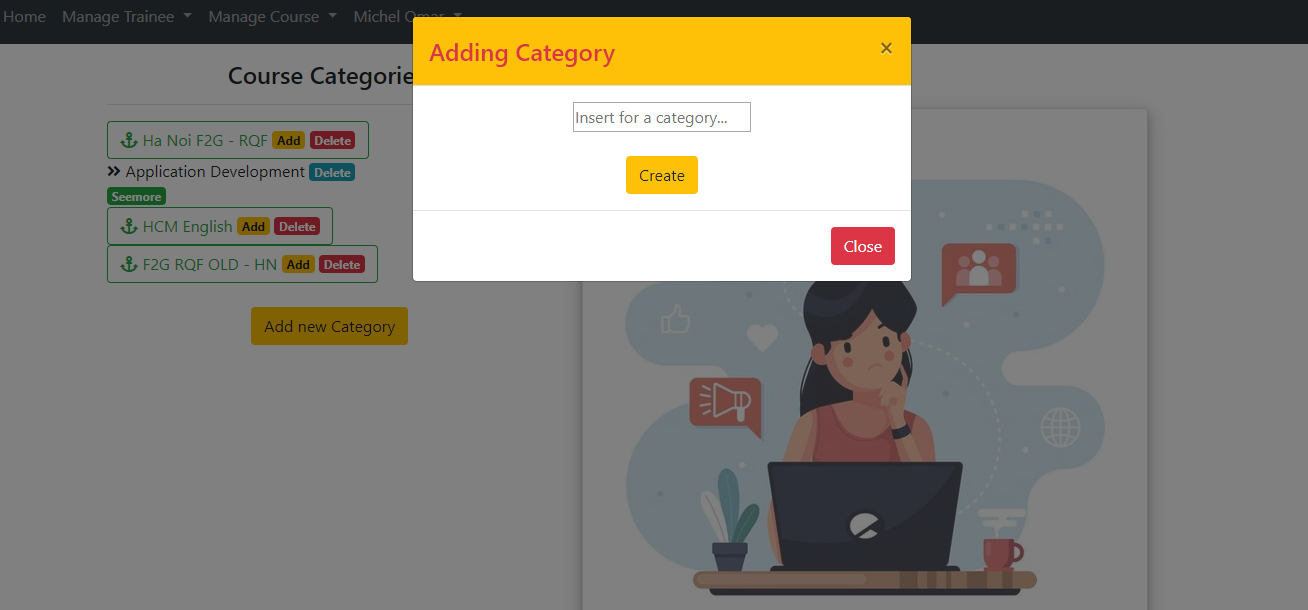


Figure adding category interface

Adding new course: To add new course in category, fill in a course name, description and submit on create button.

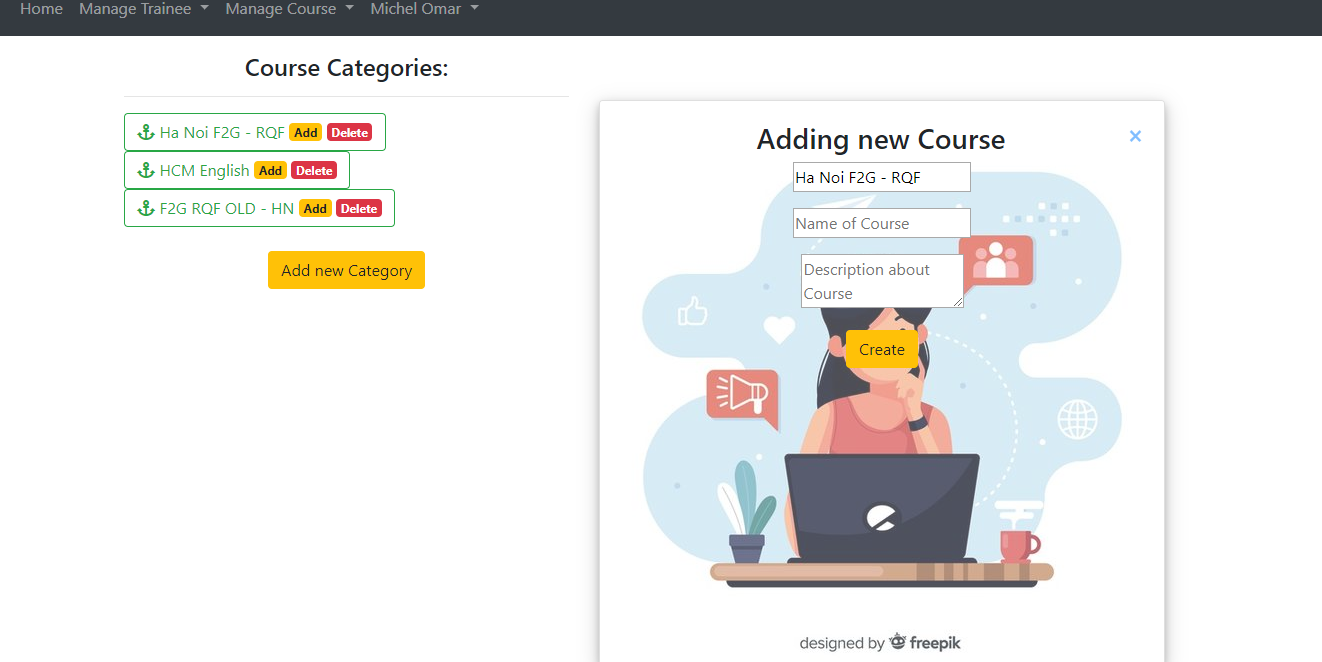


Figure Adding course interface

Management classes in course: The management class interface is place that all avaiable classes in course is showed. In here, you can view more detail or delete class.

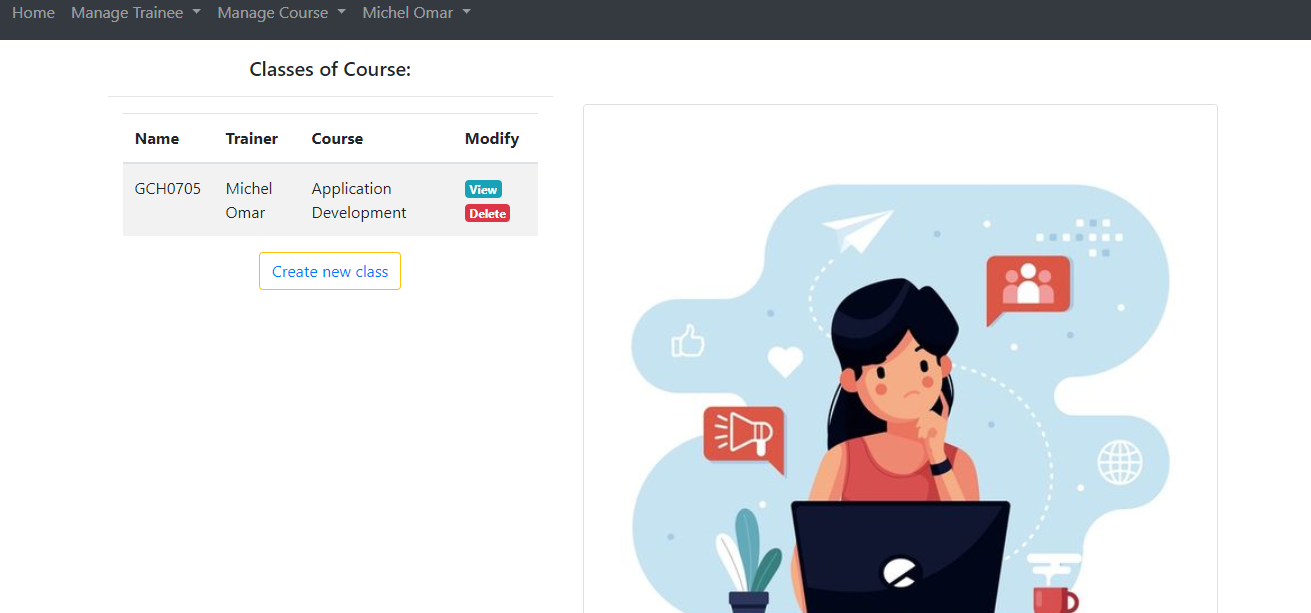


Figure Class management interface

View class detail: Every information of a class is showed in here.

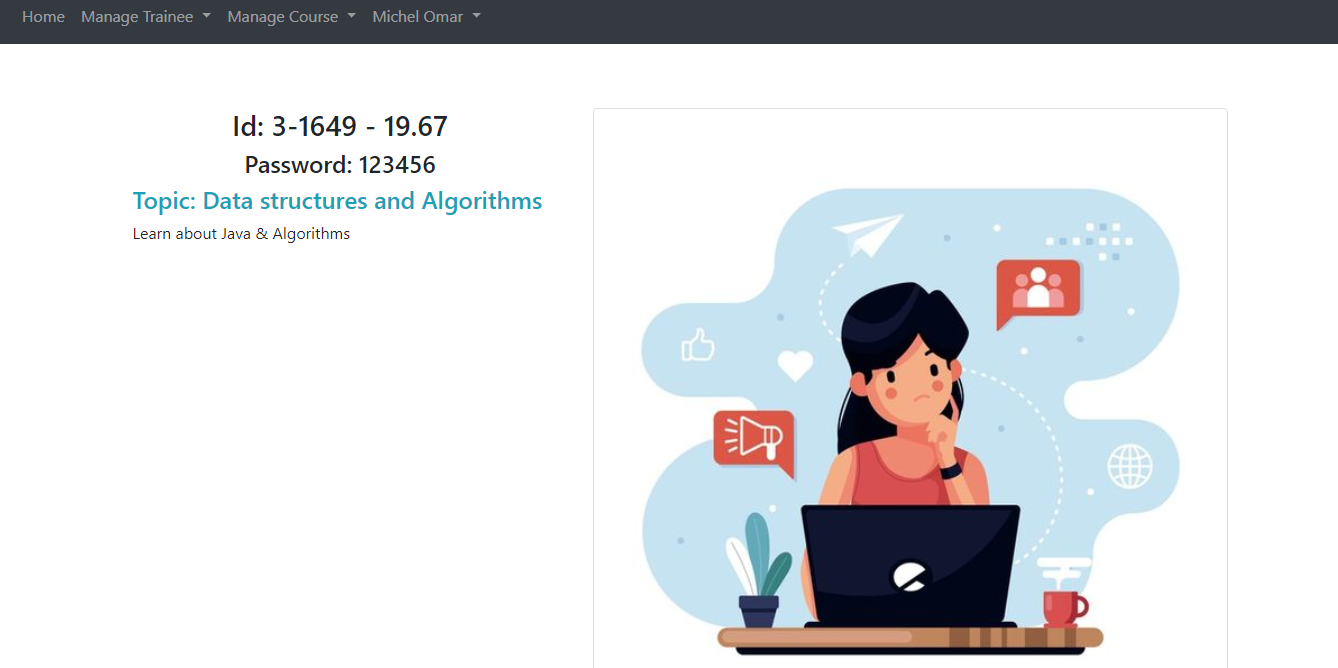


Figure Detail class interface

Adding new class: To add new class in course, fill in a class name, password, topic, description, choosing course, lecture and submit on create button.

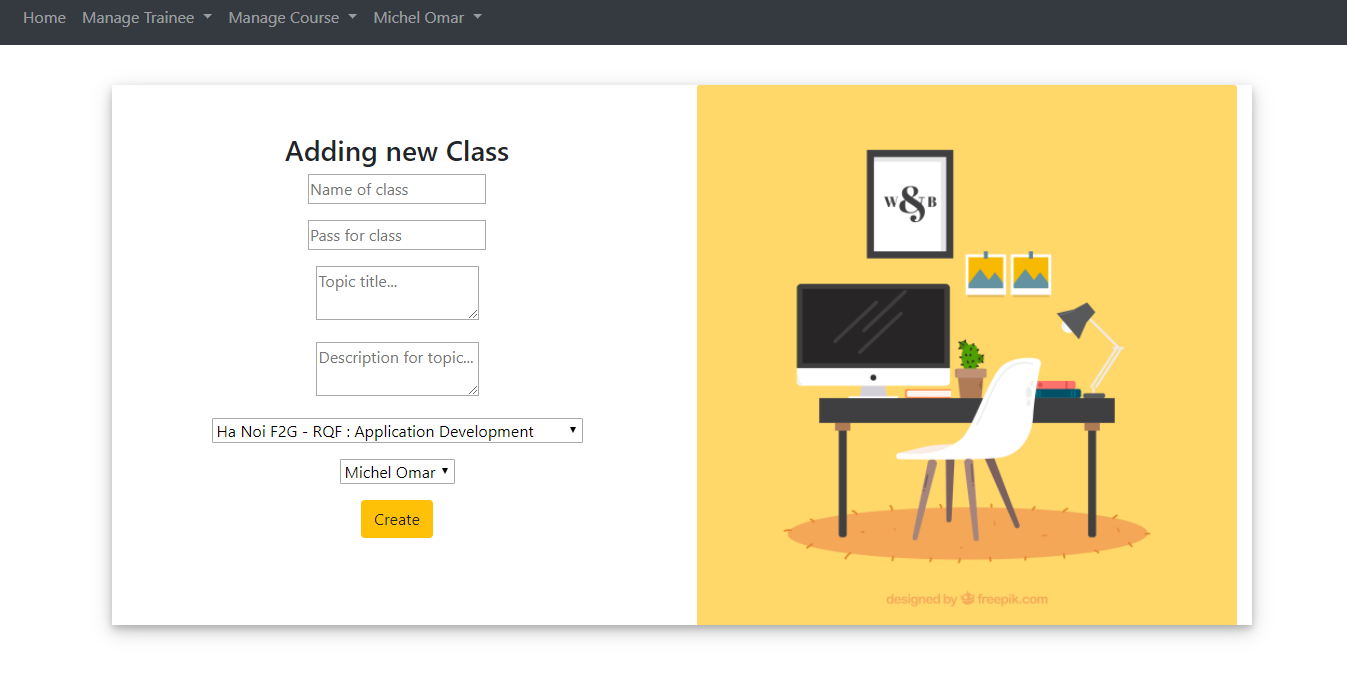


Figure Class adding interface

Trainee:

Login form of trainee: This is triner login interface, if you have a trainer account with username and password. Please, enter username and password on right place and submit by login button to go to home page of trainer.

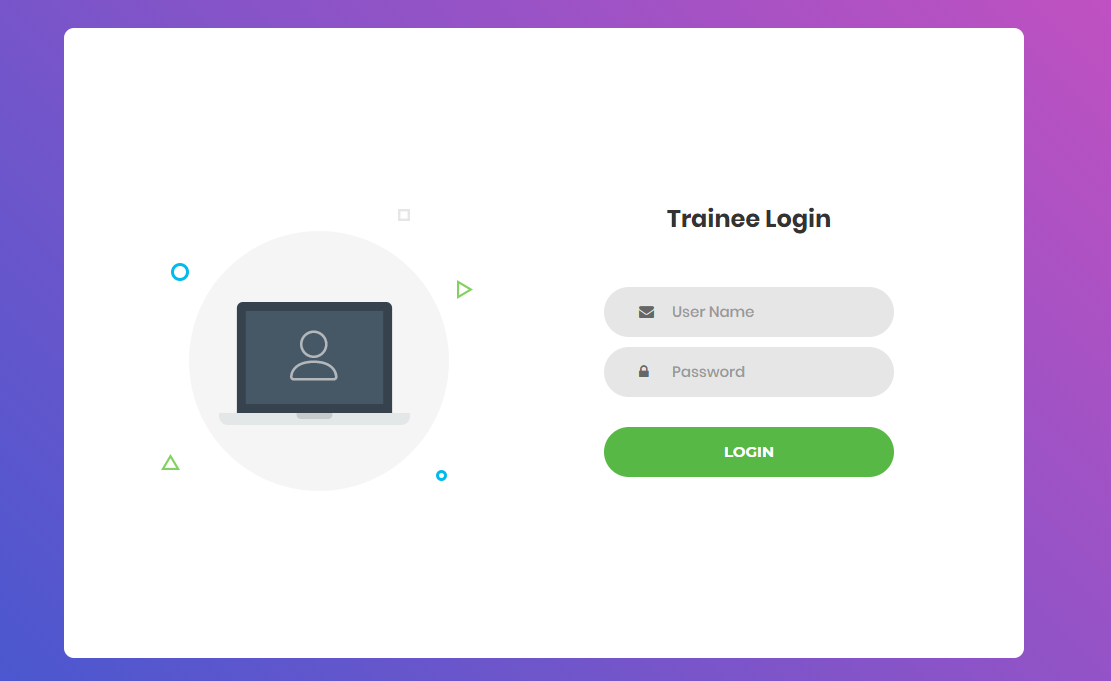


Figure Trainee login form interface

Home page of trainee: In trainee home page interface has a nav bar to control every tasks of trainee. So, pressing add View class button to go to view category, course or available class.



Figure Trainee home page interface

View categories, courses, classes of course and registered class: This interface show information like available categories and courses, class that trainee registered. To see more detail class, pressing see profile button. And if you want to register new class, find out them in category, course and enroll.

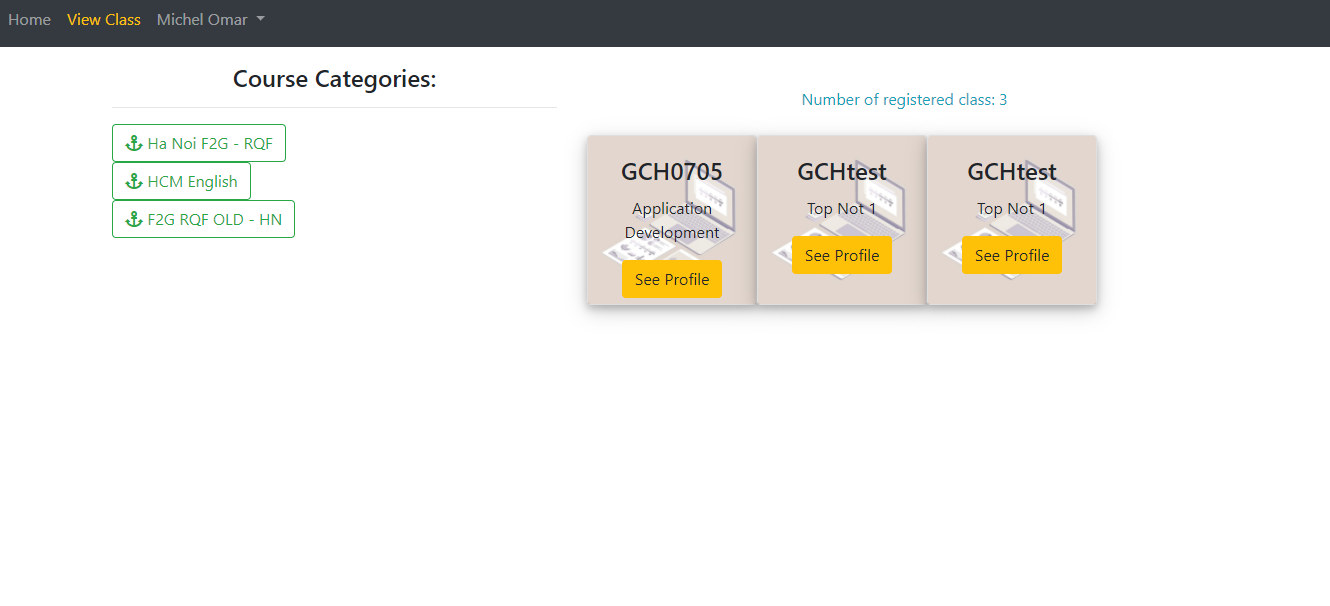


Figure Class viewing for trainee interface

View topic in class: This is interface to show more detail information of class like Name, Trainer, Topic, Description.

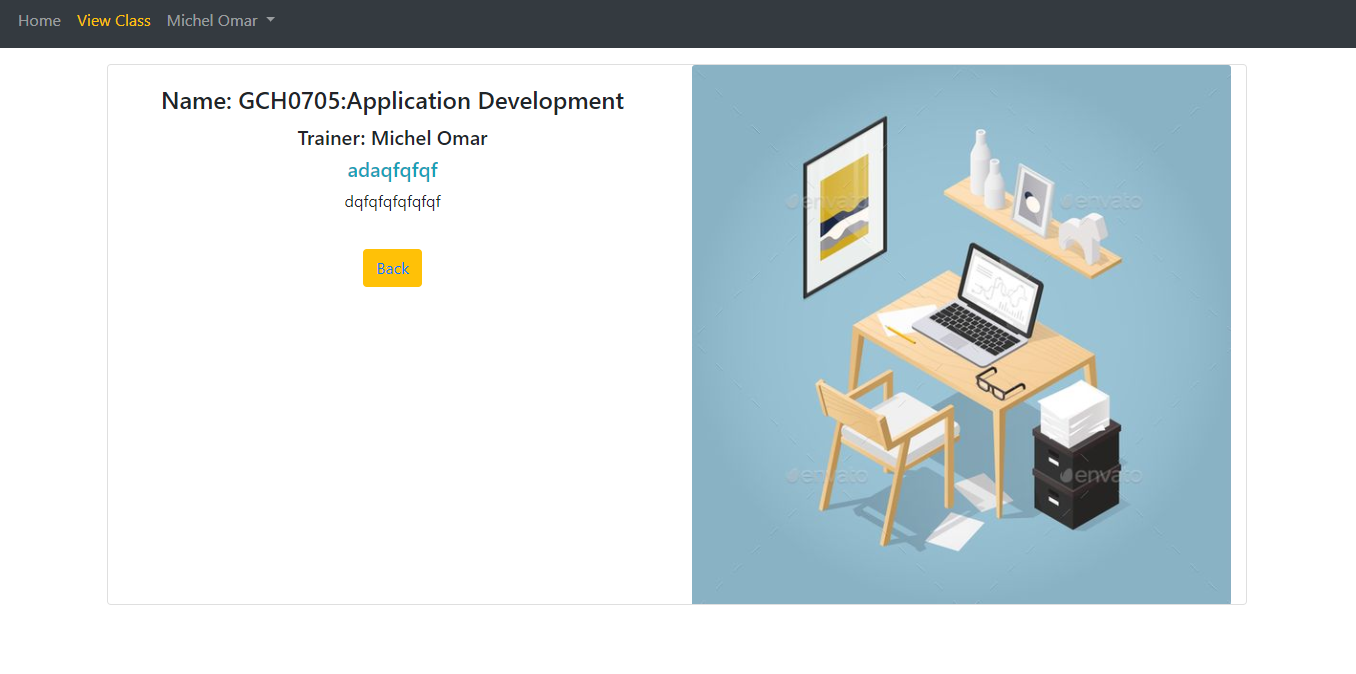


Figure Detail class interface

Register for a class: After trainees find a class that they need, entering pass word for this class and press enroll me button to register.

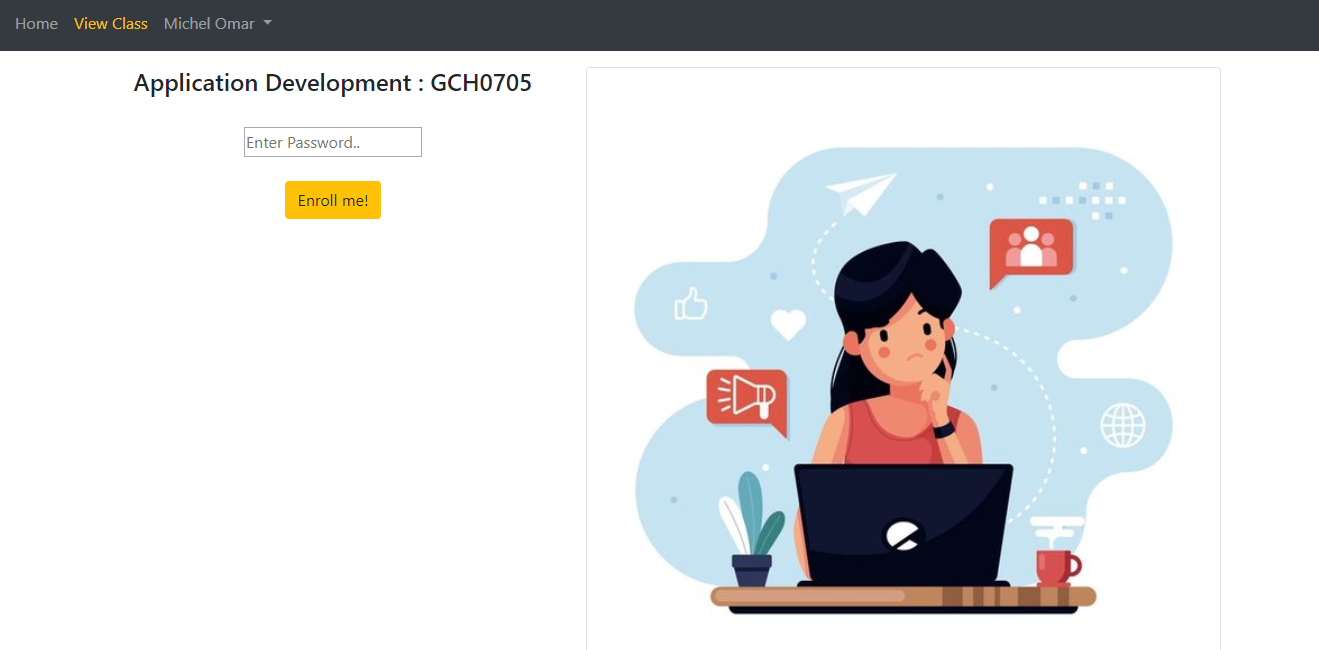


Figure Class registering interface

## 2.2 Test case:

## 2.2.1 Admin function testing:

* Login funtion:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test | What is being tested | How | Test data  used | Expected results | Time | Actual result | Action taken |
| 1 | Order of input  on data entry  screen | Enter the data following the instructions from the screen into the form with correct data | Data set:  Username: admin  Pass: admin123 | Logged in successfully | 12/31/2019 | OK | None |
| 2 | Validation of input | Performs a login without leaving any input values | Data set:  Leave the username blank | The system will issue a prompt to provide sufficient information | 12/31/2019 | OK | None |
| 3 | Sql injection | Enter values that generate sql errors | Data set:  Username:1' or '1' = '1  Password: 1' or '1' = '1 | Login unsuccessful | 12/31/2019 | OK | None |

Table Admin login function test case

* Adding trainer function:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Function | What is being tested | How | Test data  used | Expected results | Time | Actual result | Action taken |
| 1 | Validation of input | Execute register without inputting data | Data set:  Leave the username blank | The system will issue a prompt to provide sufficient information | 12/31/2019 | OK | None |
| 2 | Validation of input | Enter typical values, boundary value, values that should be rejected | Data set:  Name of trainer: %$&\*&\*@  Username: 12$#^  Pass: “””” | The system makes a request to change the input data | 12/31/2019 | Fail | Perform validate input value |
| 3 | Order of input  on data entry  screen | Enter the data following the instructions from the screen into the form with correct data | Data set:  Name: Mr teacher  Username:  Teacher123  Pass: 12345 | Notice adding successfully | 12/31/2019 | OK | None |

Table Adding trainer function test case

* Manage trainer function:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Function | What is being tested | How | Test data  used | Expected results | Time | Actual result | Action taken |
| 1 | Delete a record | Perform delete 1 trainer account | Data set:  none | The notification system deleted the object successfully | 12/31/2019 | OK | None |
| 2 | Update a record with a correct data | Perform update 1 trainer account with correct data type | Data set:  Name: Mr teacher  Username:  Teacher123  Pass: 1234578 | The notification system updated the object successfully | 12/31/2019 | OK | None |
| 3 | Update a record with values that should be rejected | Perform update 1 trainer account with values that should be rejected | Data set:  Name: ^@\*&$  Username:  &&$@123  Pass: 1234578 | The system makes a request to change the input data | 12/31/2019 | Fail | Perform validate input value |

Table Manage trainer function test case

## 2.2.2 Trainer function testing:

* Login funtion:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test | What is being tested | How | Test data  used | Expected results | Time | Actual result | Action taken |
| 1 | Order of input  on data entry  screen | Enter the data following the instructions from the screen into the form with correct data | Data set:  Username: michelomar1  Pass: omar123 | Logged in successfully | 12/31/2019 | OK | None |
| 2 | Validation of input | Performs a login without leaving any input values | Data set:  Leave the username blank | The system will issue a prompt to provide sufficient information | 12/31/2019 | OK | None |
| 3 | Sql injection | Enter values that generate sql errors | Data set:  Username:1' or '1' = '1  Password: 1' or '1' = '1 | Login unsuccessful | 12/31/2019 | OK | None |

Table Trainer Login funtion test case

* Adding trainee function:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Function | What is being tested | How | Test data  used | Expected results | Time | Actual result | Action taken |
| 1 | Order of input  on data entry  screen | Enter the data following the instructions from the screen into the form with correct data | Data set:  Id: GCH17377  Name: Nguyen Van A  Date of Birth:  01/01/1990  Address:  Ha Noi  Description:  Fpt intern  Language: english  Toeic: 800  Username: ahandsome  Password:a123 | Notice adding successfully | 12/31/2019 | OK | None |
| 2 | Validation of input | Execute register without inputting data | Data set:  Leave the username blank | The system will issue a prompt to provide sufficient information | 12/31/2019 | OK | None |
| 3 | Validation of input | Enter typical values, boundary value, values that should be rejected | Data set:  Id: GCH17377  Name: %$@^#@  Date of Birth:  01/01/1990  Address:  Ha Noi  Description:  Fpt intern  Language: english  Toeic: 800  Username: ^@#$%&#%  Password:a123 | The system makes a request to change the input data | 12/31/2019 | Fail | Perform validate input value |

Table Adding trainee function test case

* Manage trainee functions:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Function | What is being tested | How | Test data  used | Expected results | Time | Actual result | Action taken |
| 1 | Delete a record | Perform delete 1 trainee account | Data set:  none | The notification system deleted the object successfully | 12/31/2019 | OK | None |
| 2 | View a detail record | See detailed information of a trainee | Data set:  none | The trainee's information display system needs to see | 12/31/2019 | OK | None |
| 2 | Update a record with a correct data | Perform update 1 trainer account with correct data type | Id: GCH17380  Name: Nguyen Van A  Date of Birth:  01/01/1990  Address:  Ha Noi  Description:  Fpt fresher  Language: english, france  Toeic: 800  Username: ahandsome  Password:a123456 | The notification system updated the object successfully | 12/31/2019 | OK | None |
| 3 | Update a record with values that should be rejected | Perform update 1 trainer account with values that should be rejected | Id: GCH17380  Name: @@$\*&@&\*  Date of Birth:  01/01/1990  Address:  Ha Noi  Description:  Fpt fresher  Language: english, france  Toeic: 800  Username: ahandsome  Password:a123456 | The system makes a request to change the input data | 12/31/2019 | Fail | Perform validate input value |

Table Manage trainee functions test case

* Search for a trainee function:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test | What is being tested | How | Test data  used | Expected results | Time | Actual result | Action taken |
| 1 | Validation of input | Do a search using any keyword in the alphabet | Data set:  ly | Display expected results containing the character "ly" | 12/31/2019 | OK | None |
| 2 | Validation of input | Enter typical values,  boundary value, values that should be rejected | Data set:  ‘ | The system reported that no results were found | 12/31/2019 | Fail | Perform validate input value |
| 3 | Sql injection | Enter values that generate sql errors | Data set:  Username:1' or '1' = '1  Password: 1' or '1' = '1 | No results were displayed | 12/31/2019 | OK | None |

Table Search trainee funtion test case

* Manage category and course function:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Function | What is being tested | How | Test data  used | Expected results | Time | Actual result | Action taken |
| 1 | Delete a category | Perform delete 1 catelogory | Data set:  none | The notification system deleted the object successfully | 12/31/2019 | OK | None |
| 2 | Adding new category | Make a new category | Data set:  F2G Da Nang | The notification system created successfully | 12/31/2019 | OK | None |
| 3 | Adding new category | Make a new course on category | Data set: Advanced programming | The notification system created successfully | 12/31/2019 | OK | None |
| 4 | Delete a course | Perform delete 1 course in category | Data set:  none | The notification system deleted the object successfully | 12/31/2019 | OK | None |

Table Manage category and course function test case

* Adding a new class function:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test | What is being tested | How | Test data  used | Expected results | Time | Actual result | Action taken |
| 1 | Order of input  on data entry  screen | Enter the data following the instructions from the screen into the form with correct data | Data set:  Name: GCH0705  Pass:  12345  Topic: MVC  Description:  Try to manage and develop a web application | Display expected results containing the character "ly" | 12/31/2019 | OK | None |
| 2 | Validation of input | Enter typical values,  boundary value, values that should be rejected | Data set:  Name: ^@&&$  Pass:  12345  Topic: MVC  Description:  Try to manage and develop a web application | The system makes a request to change the input data | 12/31/2019 | Fail | Perform validate input value |

Table Adding a new class function test case

## 2.2.3 Trainee function testing:

* Login function:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test | What is being tested | How | Test data  used | Expected results | Time | Actual result | Action taken |
| 1 | Order of input  on data entry  screen | Enter the data following the instructions from the screen into the form with correct data | Data set:  Username: lyxinhnhat  Pass: ly123 | Logged in successfully | 12/31/2019 | OK | None |
| 2 | Validation of input | Performs a login without leaving any input values | Data set:  Leave the username blank | The system will issue a prompt to provide sufficient information | 12/31/2019 | OK | None |
| 3 | Sql injection | Enter values that generate sql errors | Data set:  Username:1' or '1' = '1  Password: 1' or '1' = '1 | Login unsuccessful | 12/31/2019 | OK | None |

Table Trainee Login function

* Enroll for a new class:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test | What is being tested | How | Test data  used | Expected results | Time | Actual result | Action taken |
| 1 | Order of input  on data entry  screen | Enter the correct password to register for a course | Data set:  Pass: 12345  (for class GCHtest) | Successful registration notification system | 12/31/2019 | OK | None |
| 2 | Validation of input | Enter the wrong password | Data set:  Pass: 713hc  (for class GCHtest) | The system notifies the wrong password and requires re-enter | 12/31/2019 | OK | None |
| 3 | Sql injection | Enter values that generate sql errors | Data set:  Password: 1' or '1' = '1 | Registration notification system failed | 12/31/2019 | OK | None |

Table Enroll for a new class funtion test case

# III Application review

3.1 Comparison to initial requirements

After the development of the first version of the system has been completed, we will continue to sit back to check that the system's functions and non functional features meet the needs of FPT. The below directly compares the current system requirements and functions.

### 3.1.1 Functional features:

Admin:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function | Initial requirement | On track | Conclusion about function of system | |
| 1 | Can login to the system through the first  page of the application | 100% | | We have designed a login interface exclusively for the admin. Here with username and password, admin can log into the system easily. In addition, the system has features that validate not allowed and prevent sql injection | |
| 2 | Can create and manage new user account for trainer (edit/delete) | 100% | | After making login. List of trainer accounts is displayed with delete or update options with information such as name, username, password. The function meets the requirement | |

Table Compare functions of admin with initial requirement

Trainer:

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Initial requirement | On Track | Conclusion about function of system |
| 1 | Can login to the system through the first  page of the application | 100% | We have designed a login interface exclusively for the Trainer. Here with username and password, trainer can log into the system easily. In addition, the system has features that validate not allowed and prevent sql injection |
| 2 | Can add new user account for trainee | 100% | After making login. Administrators can create a new trainee account with information such as trainee name, trainee accounts, age, date of birth, education, main programming language, TOEIC score, experience details, department, location, etc. on an easy web interface easy. Besides, the error values are also valid at the input. |
| 3 | Can save the information of trainee accounts in the database and from the list of trainees searchable by programming language keywords, TOEIC score etc | 80% | All information from the trainee account after being created by the trainer is saved in the database. When needed, it is easy to search by the name of the trainee. However, we still lack the invalid value validating features. |
| 4 | Can update, delete trainee accounts | 0% | Trainee cannot edit his own profile, he has to rely on the admin |
| 5 | Can manage categories such as searching, adding, updating and deleting course categories | 75% | The system provides trainers with features for managing adding and deleting categories. But there is no updating feature yet |
| 6 | Can manage courses such as searching, adding, updating and deleting courses | 75% | The system provides trainers with features for managing adding and deleting courses. But there is no updating feature yet |
| 7 | Can add topics such as topic name and topic descriptions into a course, add courses into a category | 100% | The trainer can create new topics and descriptions for a new class after performing the login session. |
| 8 | Can manage trainer profile such as adding, updating and deleting the information | 0% | The Trainer cannot directly edit his profile on our system but must rely on the admin. |
| 9 | Can assign trainer to a topic | 100% | Trainer can register teachers for specific classes on the system |
| 10 | Can assign trainee to a course | 0% | The system does not have this feature |

Table Compare functions of trainer with initial requirement

Trainee:

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Initial requirement | On track | Conclusion about function of system |
| 1 | Can login to the system through the first  page of the application | 100% | We have designed a login interface exclusively for the Trainee. Here with username and password that resigntered by trainer, trainee can log into the system easily. In addition, the system has features that validate not allowed and prevent sql injection |
| 2 | Can view courses which have a topic he is assigned to | 75% | Trainee can view the class that you have registered on the system. There is no feature registered by the trainer |
| 3 | Can update own profile | 0% | The system does not have this feature |

Table Compare functions of trainee with initial requirement

### 3.1.2 Non-functional features:

In this section we will continue to compare the existing non-funtioncal features in the system with the initial requirements in terms of performance, security, UX / UI, etc:

|  |  |  |  |
| --- | --- | --- | --- |
| Non-function features | Initial requirement | On track | Conclusion |
| UX/UI | The system needs to provide a nice user friendly interface. | 100% | Our system provides a beautiful interface, selected colors are very bright, using trendy designs such as gadient, bootrap, jquery. Besides, the interface is easy to use and not tangled, users always take the control role with the nav which is located at the top of the website. Logical user script, and there's always a navigation button to go back or show what to do next. |
| Performance | Website response time is short | 75% | Using MVC model helps the website wait time for response from 0.2 to 0.5 seconds. However, the code is not really clean, increasing the highest speed |
| Security | Ensuring information security and safety features, preventing hackers from infiltrating and stealing information. | 50% | The security features available on the site are prevention of sql injection, html and javascript code. However, the biggest minus point is that it has not done data encryption before putting into the database, which poses great risks to customer privacy. |
| Capacity and expandability | The system can cater to about 1,000 people or more | 10% | Currently the system has not provided the domain name, so it is not possible to grasp the ability to store and expand on the cloud |

Table Compare non-functions with initial requirement

## 3.2 Assessment of the system’s strength and weakness:

After completing the work of developing the learning environment management system within FPT co. Now is the time for us to make an objective assessment of what the system has done and has not done (strengths and weaknesses) in the first version of the system to make improvements in the next versions.

Advantage:

* In the front-end, the system possesses a beautiful interface but it is very simple and easy to use for users. Using technologies like bootrap, jquery makes the interface more interesting.
* The user script is simple, there are always navigation buttons that make the user feel in control of everything. Short website processing times increase the experience.
* Regarding security, the system implements features to prevent html and sql injection to avoid the risk of hackers logging directly into the system.

Disadvantage:

* The system did not fully meet the requirements put forward. The simple design is not really attractive to users; sometimes complex designs are difficult to understand.
* In the back-end, using the php language to design is somewhat outdated compared to the current trend.
* In terms of security, there are no steps to encrypt data before saving to the database, as well as decryption when using data. This poses a huge risk when hackers can use the data directly in case of theft.
* The system still lacks some validation features for the input, which makes the data not clean.
* In terms of coding, the functions are not clean, the code is still quite long, making it difficult to reuse.
* The system has not provided backup data in case of emergency, backup features to restore information have not been provided.
* The scalability of the system is limited. Only meet for more than 1000 people.

## 3.3 Indentified risks

According to Cody, L (2016, online), The launch is the crucial first step in any website or web app’s journey. The truth is not so, your website or web application is successful and growing, then you need to constant update and improve your website before the risks may occur. Here are the risks to avoid and solutions:

|  |  |  |
| --- | --- | --- |
| Number | Risks | Solution |
| 1 | **Site Security**: every day thousands of websites are under attack, especially large or new ones. Personal and financial information are most at risk of being stolen. (Rean, J, 2015, online) | Hiring an expert to check your website’s source code for potential security flaws: the latest hacks, cracks, SQL injections, and XSS vulnerabilities. “If you are using a WordPress website, I highly recommend installing WordFence Security.” (Rean, J, 2015, online) |
| 2 | **Legal Pages:** Risks when users, developers and managers do not implement or violate legal policies (information security, etc.) (Rean, J, 2015, online) | Set up pages to commit legality regarding people who use your site, from copyrights, cookies and information about your company. If you are using WordPress, you can install a plugin called Terms of Service and Automated Privacy Policy. (Rean, J, 2015, online) |
| 3 | **Website Copy:** Risks from information or content on the website that has not been proofread before launching. Mistakes in the content or information provided, however small, will take away the reputation of the entire site. | Don't launch if you don't have untested content or for important pages like About. Also provides full information about copyright, version, author (Rean, J, 2015, online) |
| 4 | **JavaScript Independence:** Some browsers may have JavaScript disabled implicitly for any reason but do not indicate how to enable it. If you do not guarantee the independence of JavaScript, your website may look like it is broken. (Rean, J, 2015, online) | “Try turning off your browser’s JavaScript and browse around your website. If it looks broken, then you need to fix that.” (Rean, J, 2015, online) |
| 5 | **Backup:** Most web hosts are not good at keeping backups of websites. This may result in loss or inability to recover data in an emergency.(Rean, J, 2015, online) | Pay them to set up recurring jobs for backups depending on your needs and save backups on a separate location independent of your server. (Rean, J, 2015, online) |
| 6 | **Unexpected risk:** Unexpected risk: Sometimes a problem in your site will occur and people will visit pages that do not exist. Sometimes, they even visit pages they shouldn't. (Rean, J, 2015, online) | “Check for all possible security risks on your site and block public access to those pages, and of course, when you block people on specific pages, you need to let them know why.” (Rean, J, 2015, online) |

Table Risks to website after launch

# Conclusion

As such, our group is under the help of lecturers and the spirit of cooperation and cooperation of our team members, our team has completed the process of developing the educational environment system within FPT co. We have really come a long way with a lot of new things in managing a project for the first time. We start with making a plan for the right tools and methods, then applying that plan to develop and manage the entire project throughout its life cycle. Finally, as testers we learn how to test and evaluate system features and offer improved solutions. Because this is the first time, we have been exposed to a large project, so it is inevitable that mistakes. If given the opportunity, we hope that there will be conditions to develop the next version of the system.

# References:

Cody, L. (2016) *4 Keys to Effectively Managing Your Website After Launch*. Avaiable from: https://modeeffect.com/4-keys-managing-website-after-launch/

[Access 25/02/2019]

Rean, J. (2015) *15 Things to Check Before Launching a New Website*. Avaiable from: https://onextrapixel.com/15-things-to-check-before-launching-a-new-website/

[Access 25/02/2019]