1. **XAMPP**
2. Download and install *XAMPP*. You can download *XAMPP* from [here](https://www.apachefriends.org/download.html). Please choose the appropriate installer with your operating system.

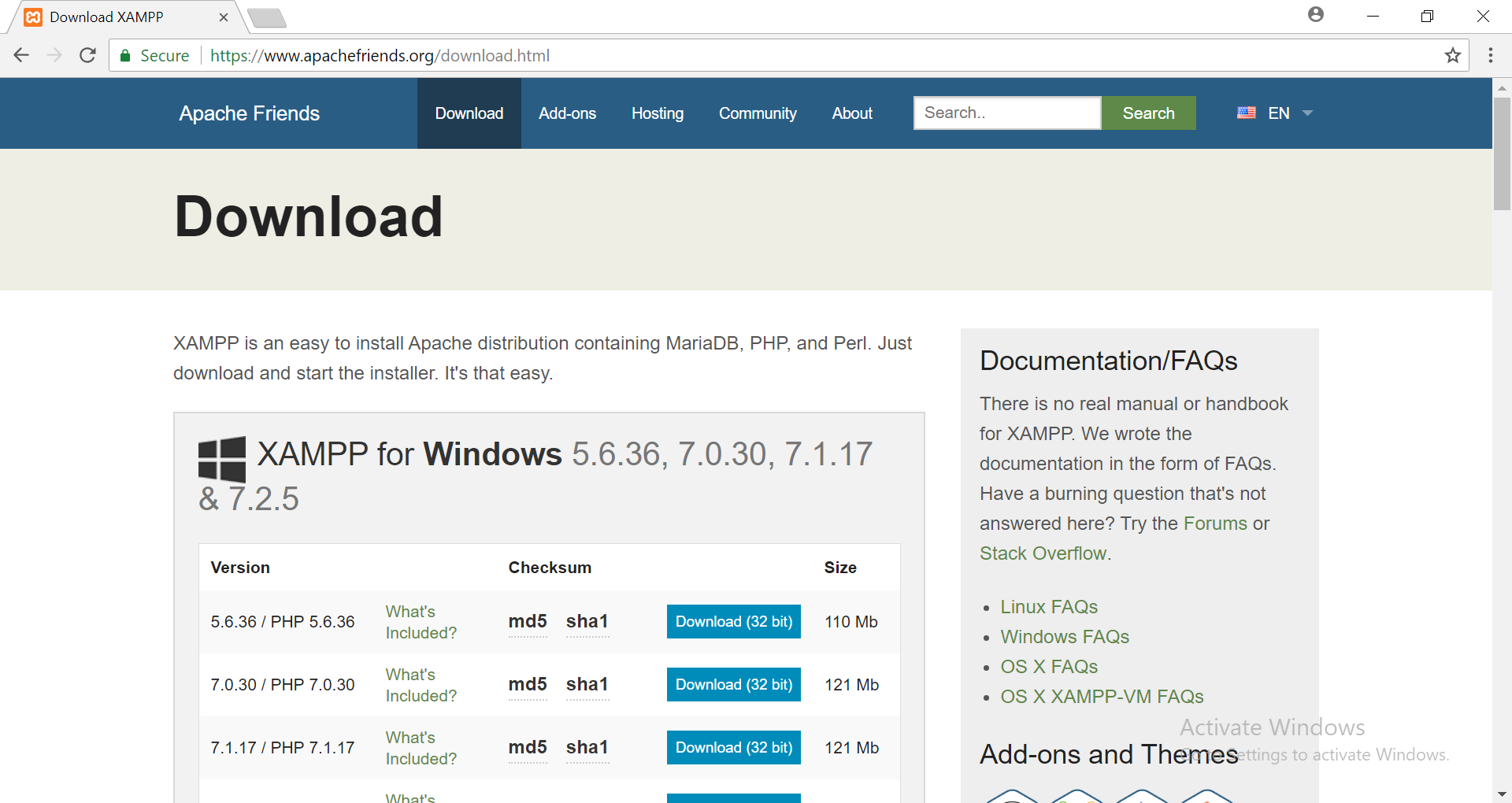


Figure 1: XAMPP's official download page

1. Open *XAMPP*

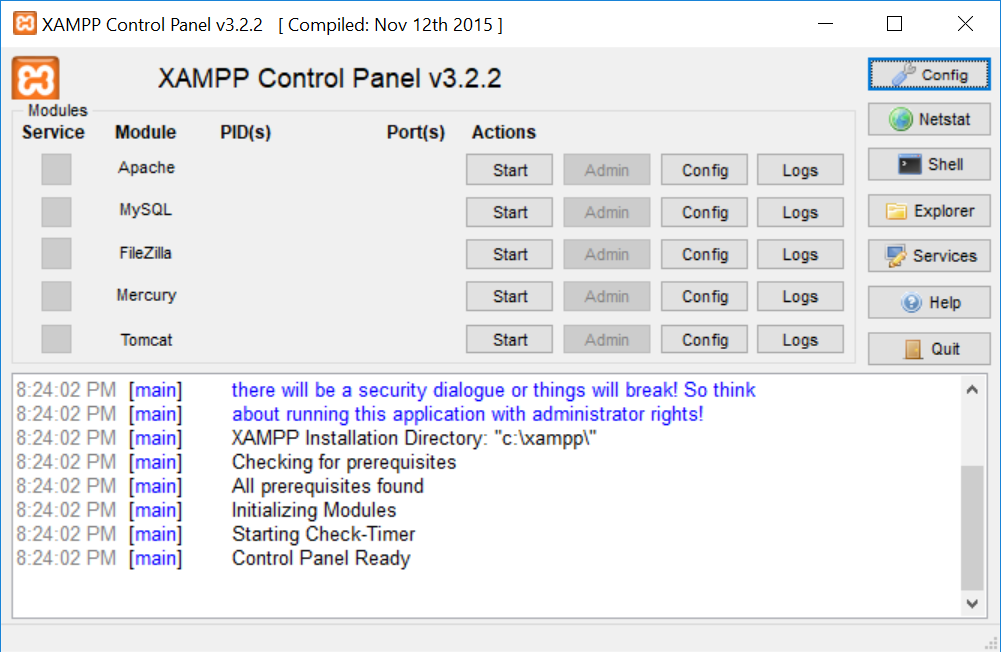


Figure 2: XAMPP's main interface

1. Click on *Config* of Module **Apache** and select *PHP (php.ini)*

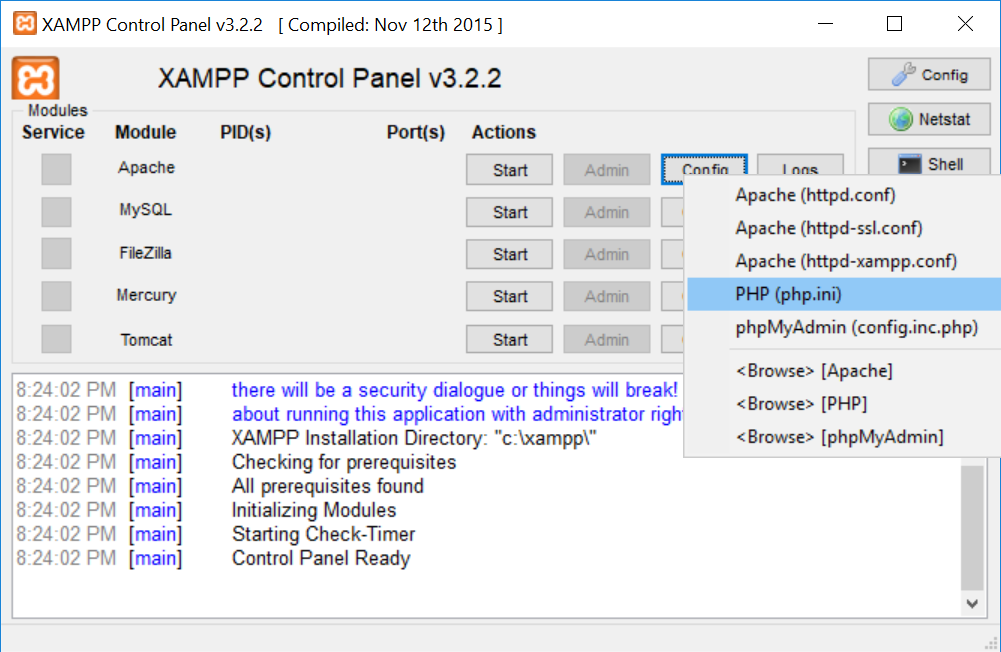


Figure 3: PHP (php.ini) on XAMPP's interface

1. At File Uploads section, set *file\_uploads=On* to allow HTTP file uploads, and *upload\_max\_filesize=30M* (Maximum allowed size for uploaded files is 30MB)

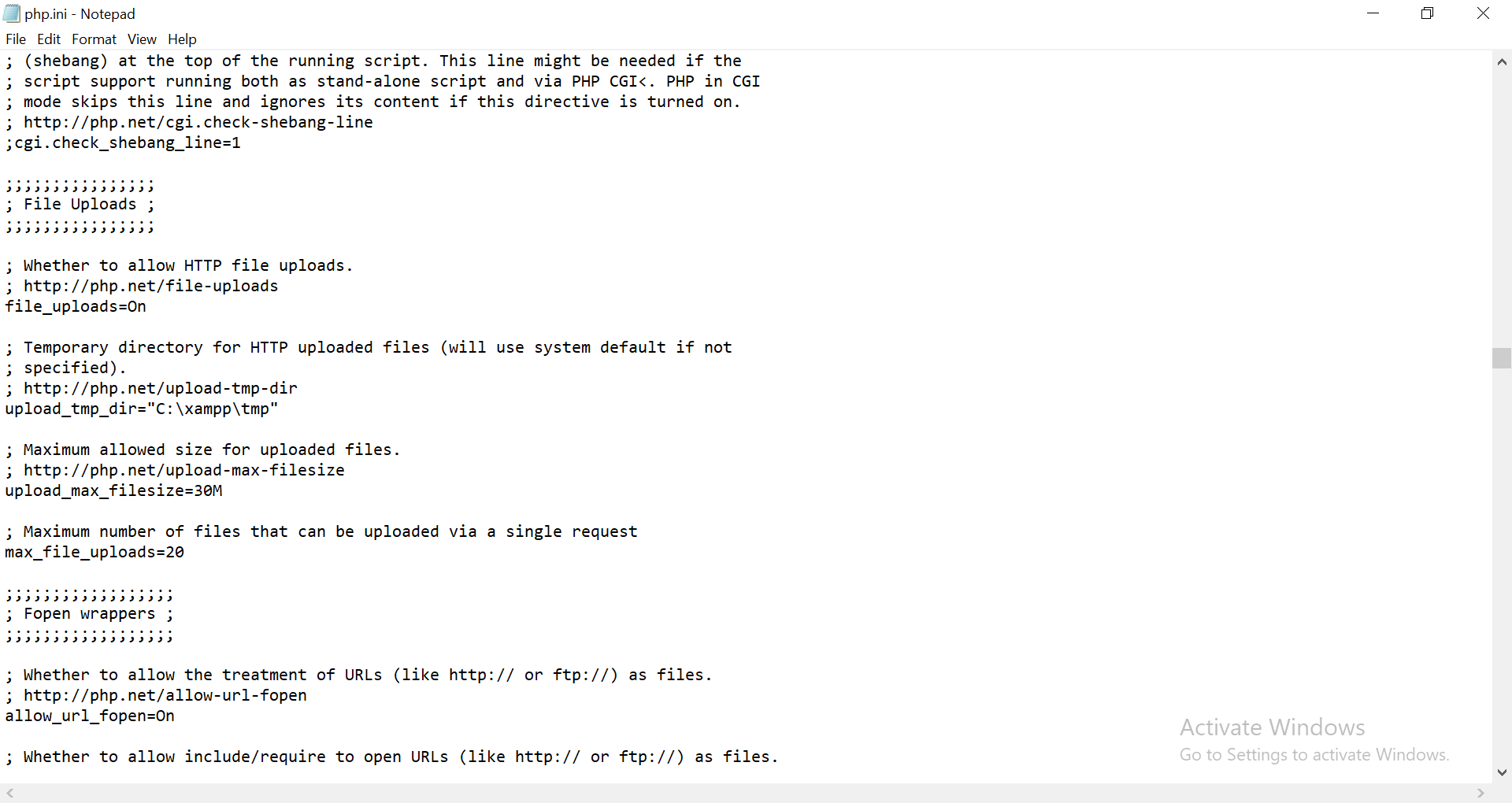


Figure 4: Our PHP.ini config

1. Click on *Config* of Module **Apache** and select *Apache (httpd.conf)*

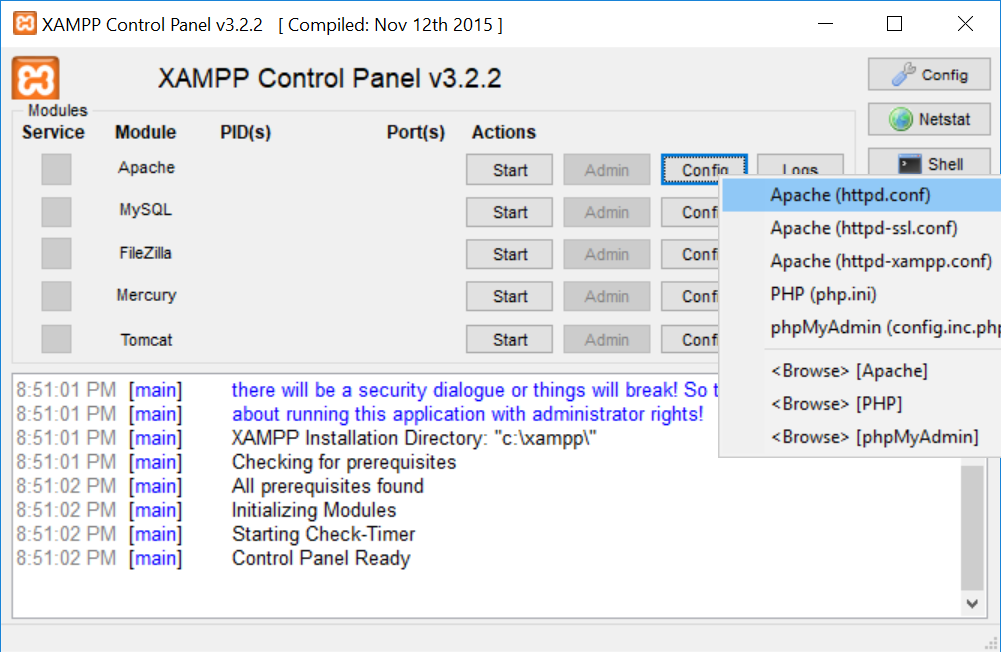


Figure 5: Apache (httpd.conf) on XAMPP's interface

1. Change *DocumentRoot* from the default value to the path of *QuizMaker* on your computer

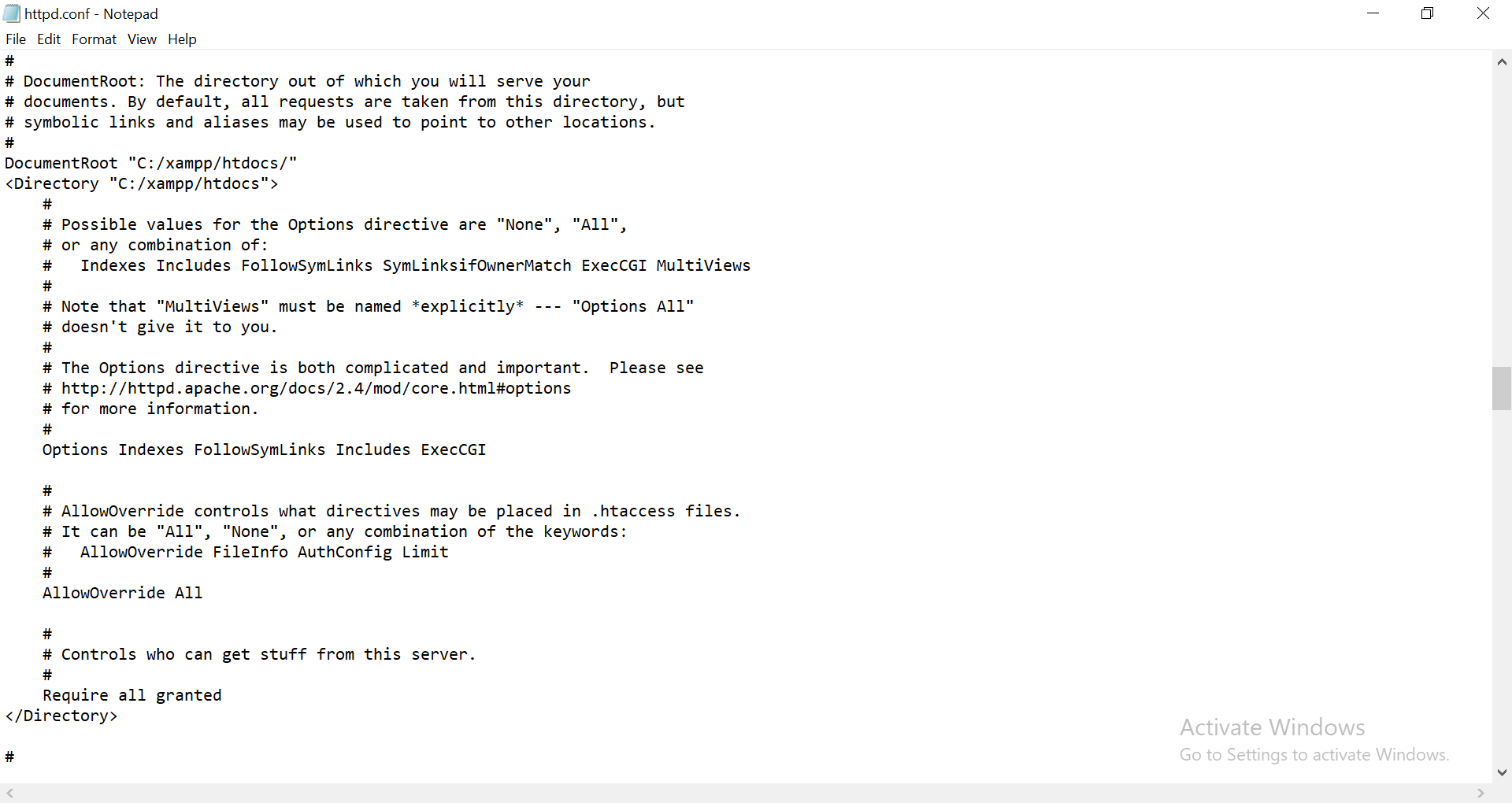


Figure 6: Default DocumentRoot of httpd.conf

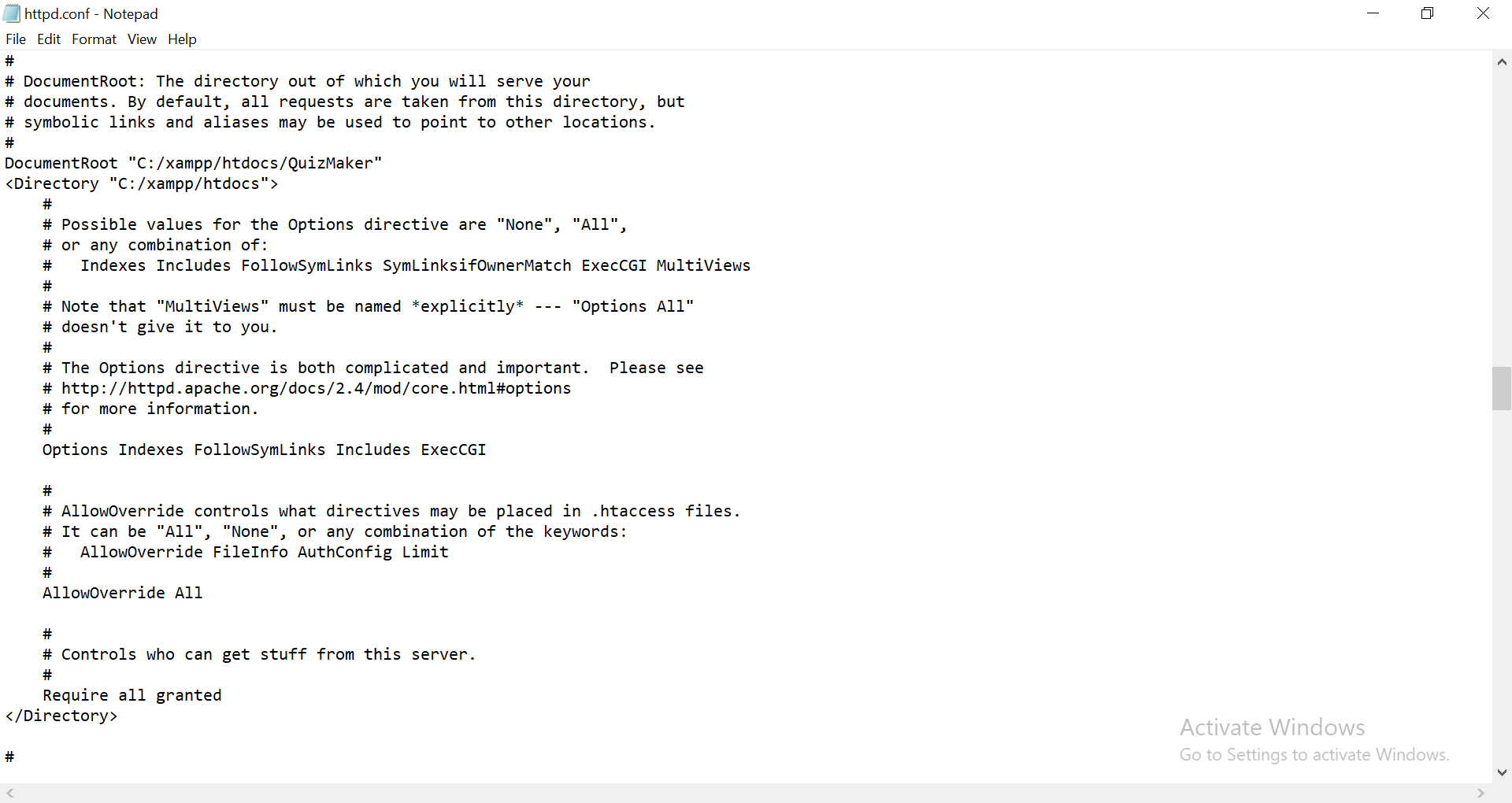


Figure 7: Our httpd.conf config

1. Start Module **Apache** and **MySQL** by press on *Start* button

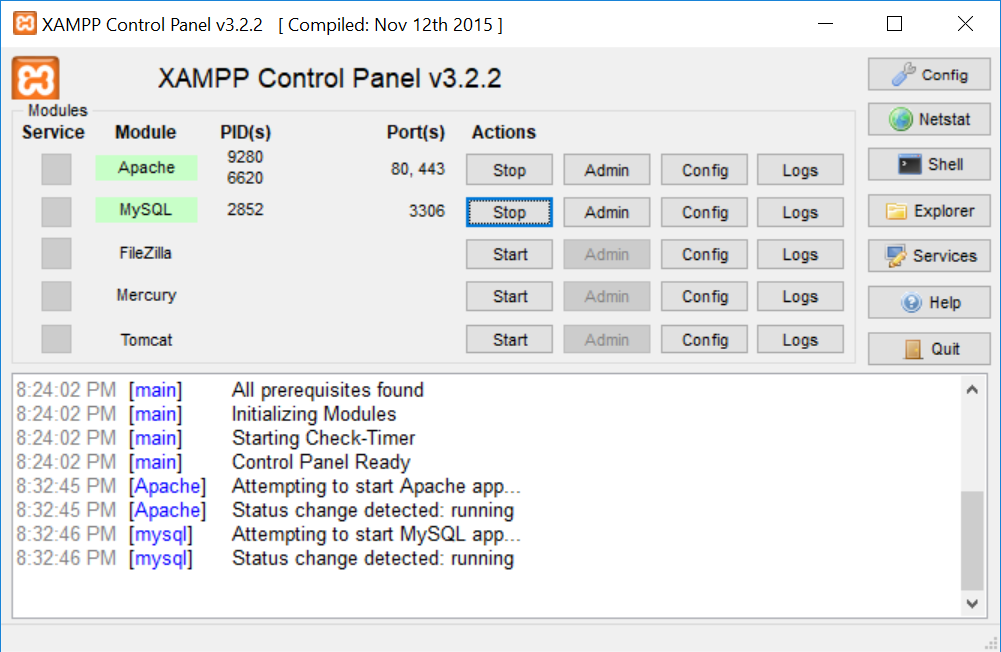


Figure 8: Apache and MySQL modules started

1. By default, QuizMaker are running on localhost, with port 80. Open a web browser, and type localhost to the URL

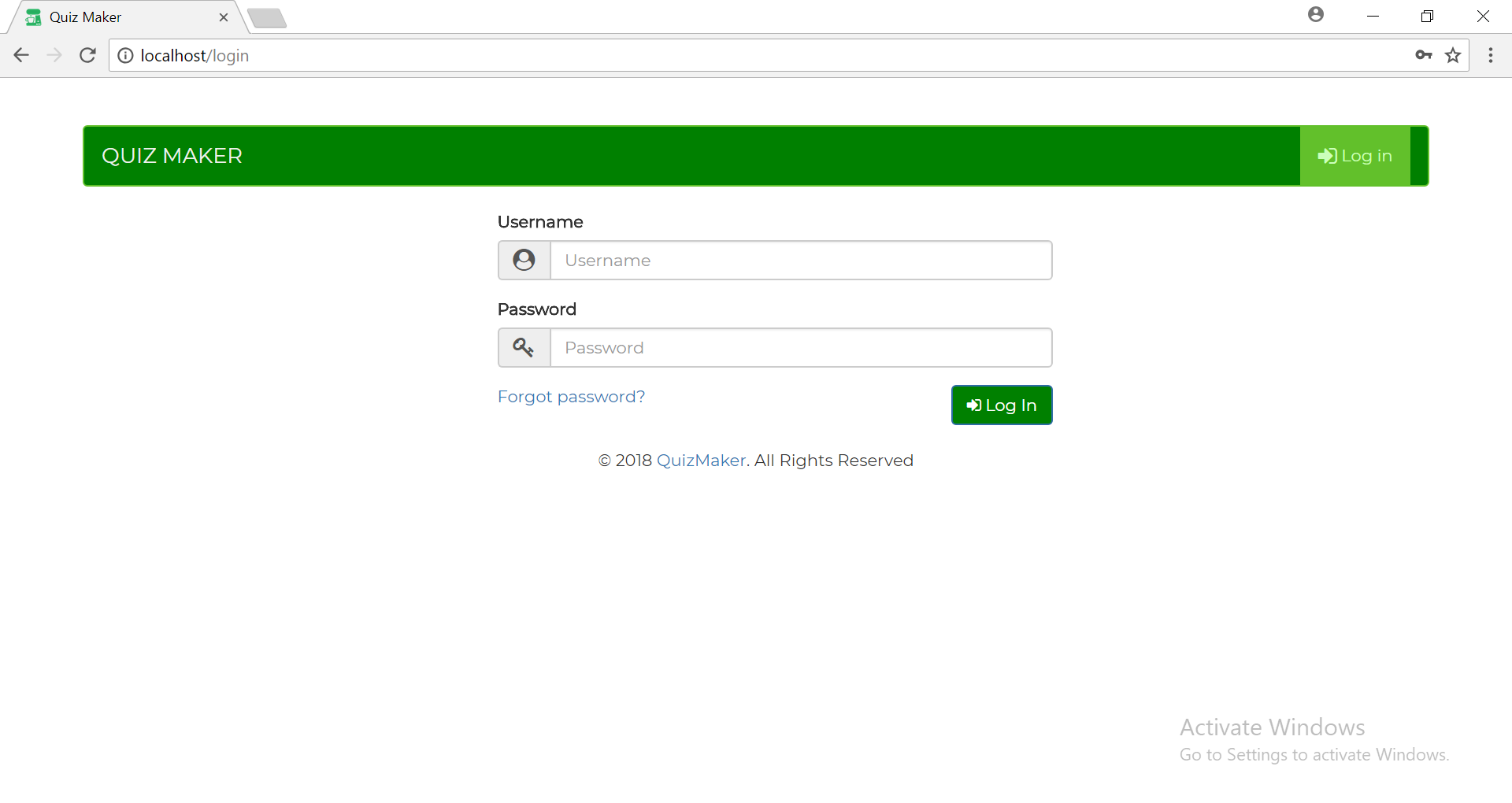


Figure 9: QuizMaker's login page

1. You need to create a database for QuizMaker. For quick start, we provide initDatabase.php to initial the database (which named assignment). To run this script, on URL, type localhost/initDatabase.php and press Enter.

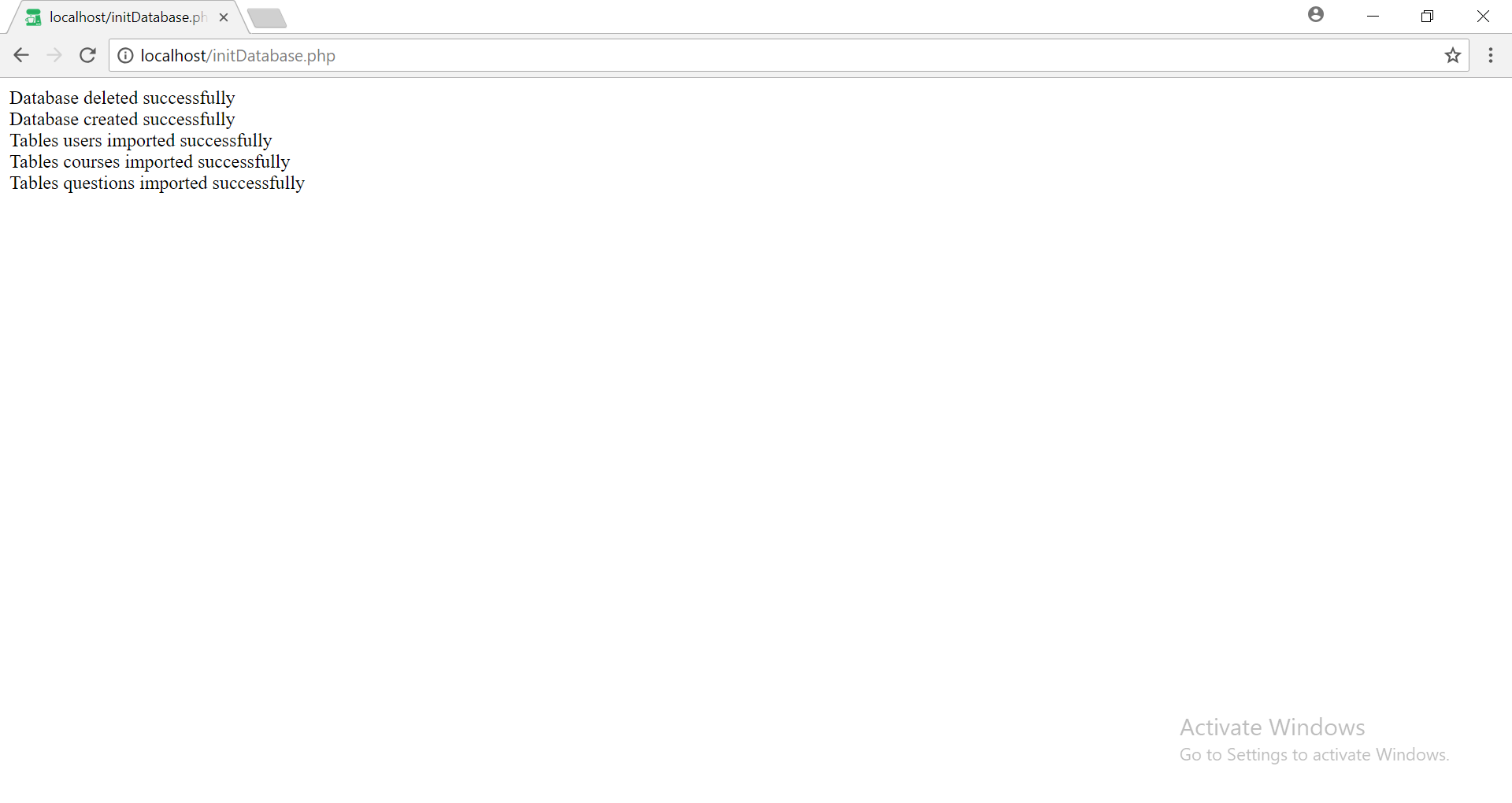


Figure 10: Finished initial database

1. To check the database was successfully initial, on *XAMPP’s* interface, press on *Admin* of **MySQL** module

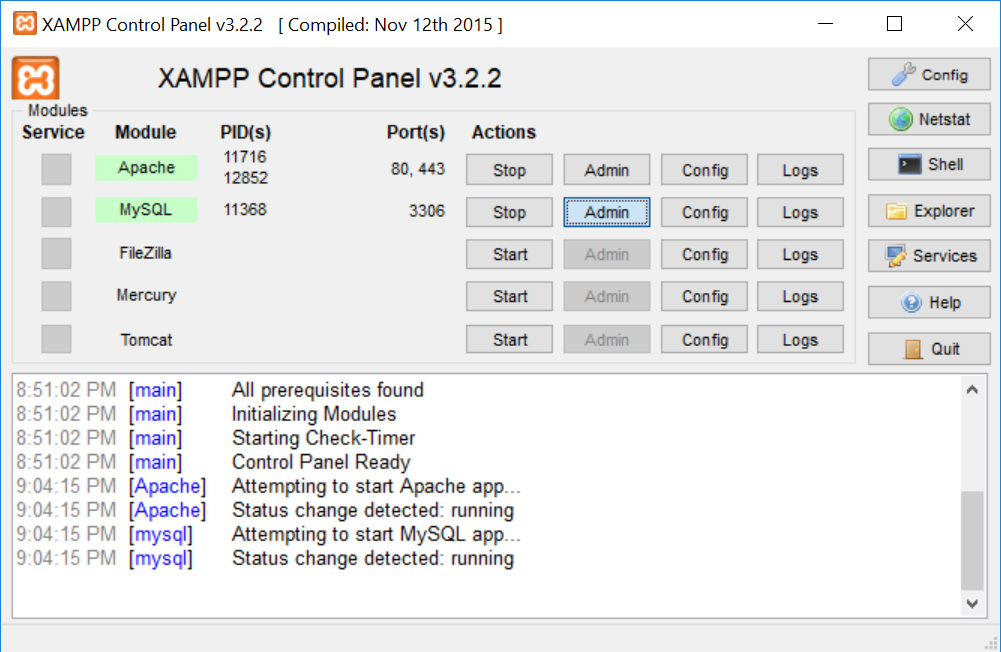


Figure 11: Admin button of MySQL module

1. Database (which named *assignment*) has successfully initialized with 3 tables: *courses*, *questions*, *users* and some stored procedures.

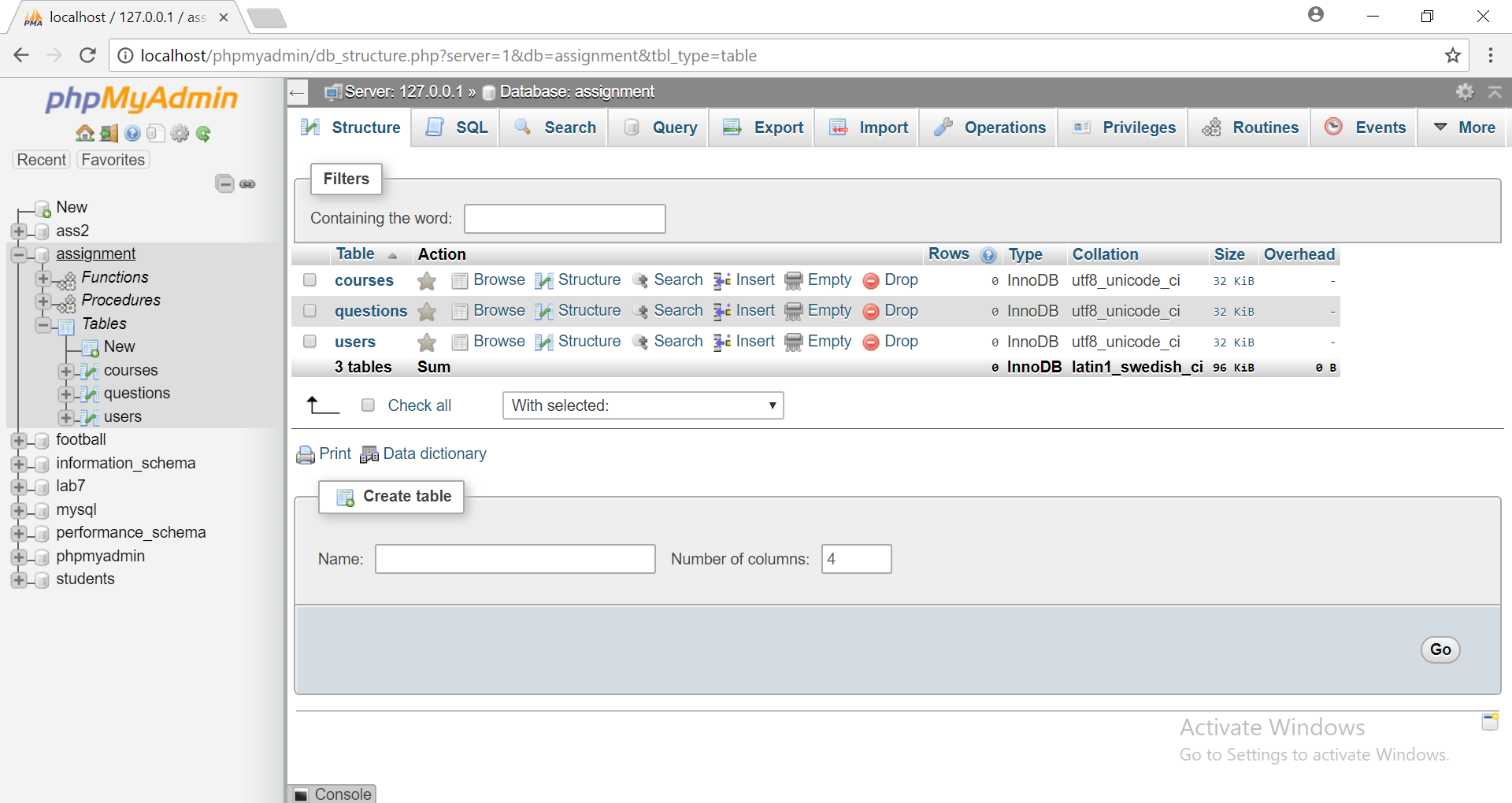


Figure 12: phpMyAdmin's interface

1. Because *QuizMaker* doesn’t allow to sign up, you need to create at least one account, for *QuizMaker’s Admin*. We provide *create-admin.php* to create an *Admin* account with *username*:*password* is *admin*:*admin.* Make sure to change the password after successfully logged in.
2. [Optional] We also provide a script to import some course data. To run it, go to *localhost/import-courses.php.* It will import some courses in our programs.

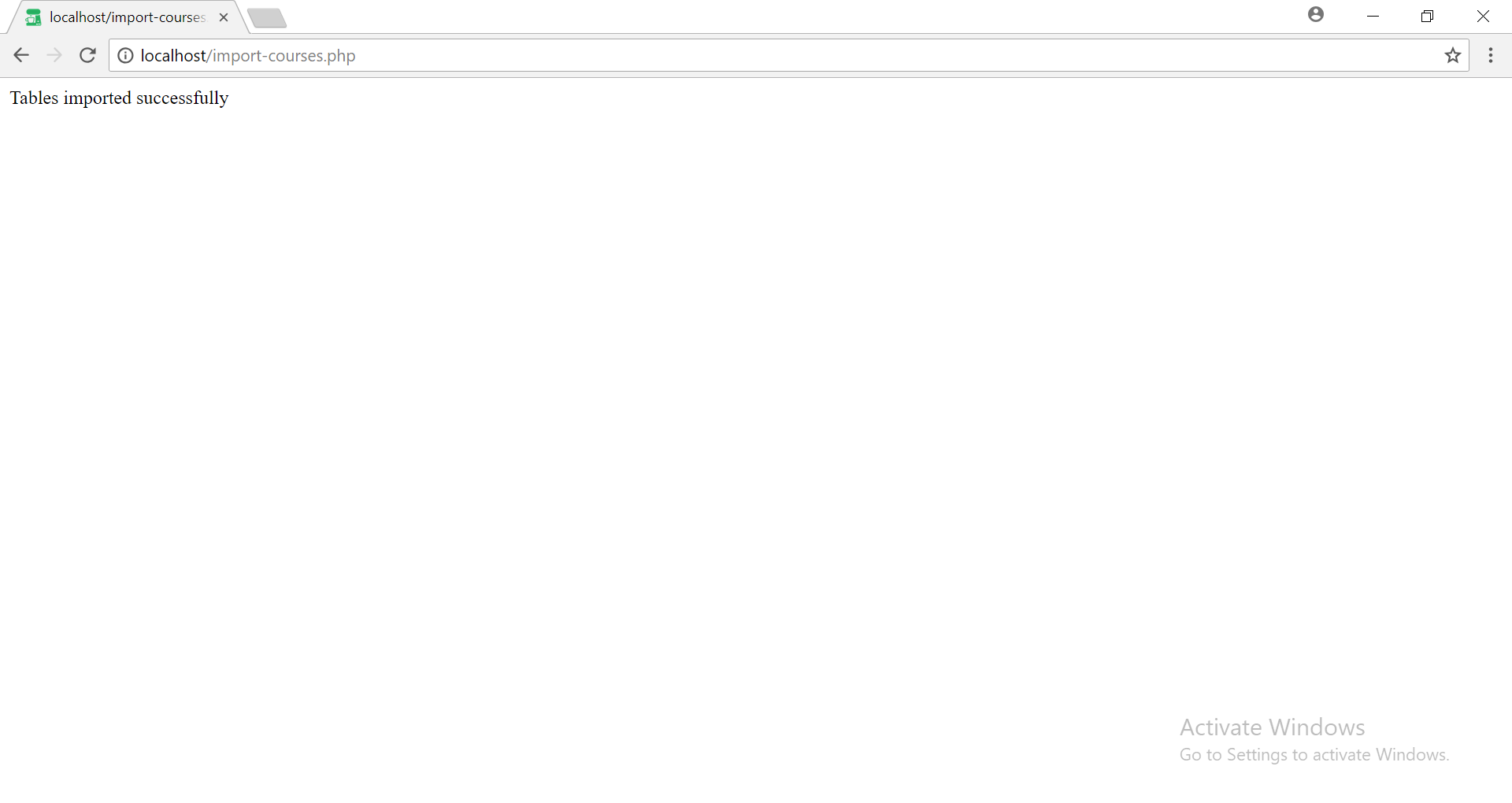


Figure 13: Import course data successfully

1. [Optional] Another way to import data is using Import feature of *phpMyAdmin*. We provide *questions.csv* which store some sample questions. To import it, open *phpMyAdmin*, select your database (*assignment* in our cases), select *questions* table and press on *Import*.

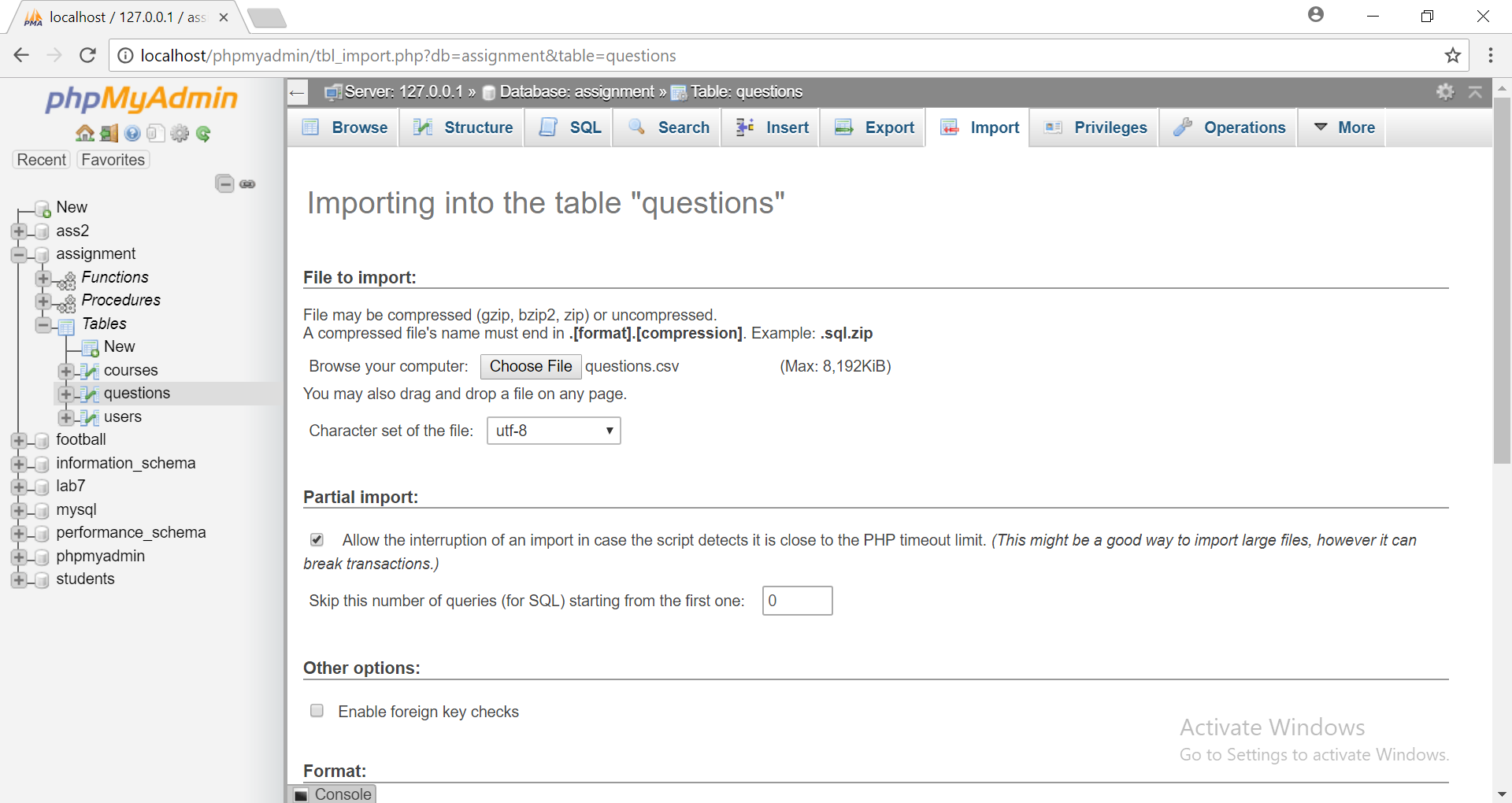


Figure 14: phpMyAdmin's Import interface

1. Click on choose file and select *questions.csv* which is located inside *data* folder. Before press *Go*, uncheck *Enable foreign key checks* option to skip foreign key checks.

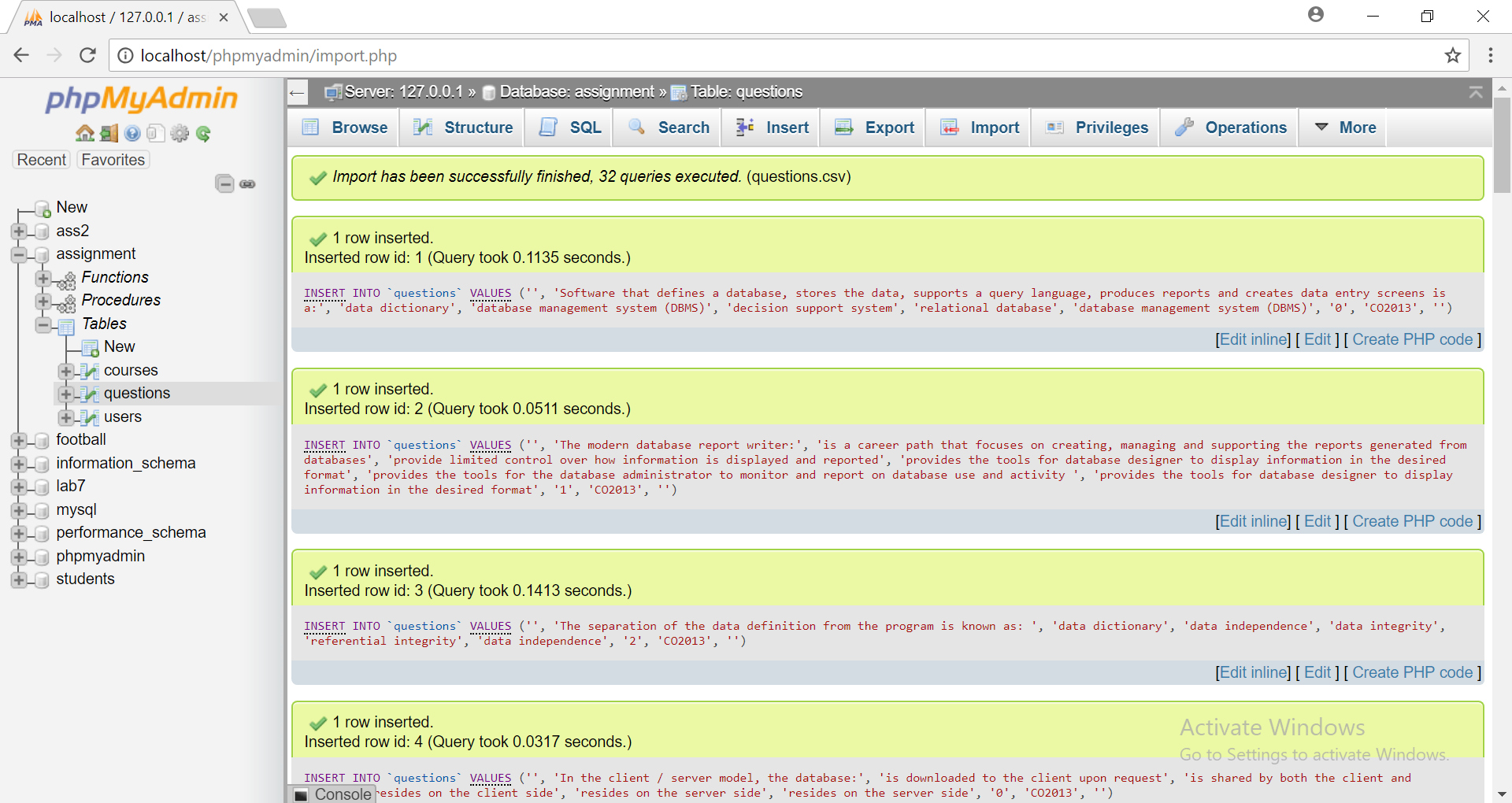


Figure 15: Questions successfully imported

1. **DATA STRUCTURE**

**1. Tables**

|  |  |  |  |
| --- | --- | --- | --- |
| *Users* | | | |
| Name | **Datatype** | **PK** | **Constraint / Note** |
| id | INT(10) | x | Auto increment |
| username | VARCHAR(50) |  | Unique |
| password | VARCHAR(50) |  |  |
| role | ENUM(‘0’, ’1’, ’2’) |  | ‘0’ -> Admin; ‘1’ -> User; ‘2’ -> Staff |
| firstname | VARCHAR(16) |  |  |
| lastname | VARCHAR(16) |  |  |
| middlename | VARCHAR(20) |  |  |
| dateofbirth | DATE |  |  |
| address | VARCHAR(64) |  |  |
| phone | VARCHAR(16) |  |  |
| email | VARCHAR(64) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| *Courses* | | | |
| Name | **Datatype** | **PK** | **Constraint / Note** |
| id | INT(10) | x | Auto increment |
| name | VARCHAR(50) |  |  |
| code | VARCHAR(6) |  | Unique |

|  |  |  |  |
| --- | --- | --- | --- |
| *Questions* | | | |
| Name | **Datatype** | **PK** | **Constraint / Note** |
| id | INT(10) | x | Auto increment |
| question | VARCHAR(1000) |  |  |
| option1 | VARCHAR(1000) |  |  |
| option2 | VARCHAR(1000) |  |  |
| option3 | VARCHAR(1000) |  |  |
| option4 | VARCHAR(1000) |  |  |
| answer | ENUM(‘1’, ‘2’, ‘3’, ‘4’) |  |  |
| difficult | ENUM(‘0’, ‘1’, ‘2’) |  | ‘0’ -> Easy; ‘1’ -> Medium; ‘2’ -> Hard |
| code | VARCHAR(6) |  | Foreign key to Courses.code |
| image | VARCHAR(1000) |  | Image URL |

**2. Stored procedure and Stored function**

***a. sp\_get\_paper\_question***

The procedure uses to get random question based on number of easy, medium, and hard questions

*CREATE PROCEDURE `sp\_get\_paper\_question`(IN `i\_code` VARCHAR(6), IN `i\_easy\_number` INT(32) UNSIGNED, IN `i\_medium\_number` INT(32) UNSIGNED, IN `i\_hard\_number` INT(32) UNSIGNED)*

*BEGIN*

*SET @sql\_text= concat("SELECT \* FROM (" , " (SELECT \* FROM `questions`" , " WHERE code= \'", i\_code, "\' AND difficult= \'0\'" , " ORDER BY RAND() LIMIT ", i\_easy\_number, ")" , " UNION ALL" , " (SELECT \* FROM `questions`" , " WHERE code= \'", i\_code, "\' AND difficult= \'1\'" , " ORDER BY RAND() LIMIT ", i\_medium\_number, ")" , " UNION ALL" , " (SELECT \* FROM `questions`" , " WHERE code= \'", i\_code, "\' AND difficult= \'2\'" , " ORDER BY RAND() LIMIT ", i\_hard\_number, ")) AS T ORDER BY RAND()”);*

*PREPARE stmt FROM @sql\_text;*

*EXECUTE stmt;*

*DEALLOCATE PREPARE stmt;*

*END*

**b. sp\_set\_account\_info**

The procedure uses to insert information of an account when admin create a new account. Then return username (auto generated), and password

*CREATE PROCEDURE `sp\_set\_account\_info`(IN `i\_role` VARCHAR(1), IN `i\_fname` VARCHAR(16), IN `i\_lname` VARCHAR(16), IN `i\_mname` VARCHAR(20), IN `i\_date\_of\_birth` VARCHAR(16), IN `i\_address` VARCHAR(64), IN `i\_phone` VARCHAR(16), IN `i\_email` VARCHAR(64), OUT `o\_username` VARCHAR(20), OUT `o\_password` VARCHAR(8))*

*BEGIN*

*SET @d\_date\_of\_birth := STR\_TO\_DATE(i\_date\_of\_birth, "%d/%m/%Y");*

*SET @username= concat(substr(i\_lname, 1, 1), substr(i\_mname, 1, 1), i\_fname);*

*SET @username= sf\_get\_username(@username);*

*SET @password= UNIX\_TIMESTAMP()% 9000+ 1000;*

*SET @sql\_text= concat("INSERT INTO users (username, password, role, firstname, lastname, middlename, dateofbirth, address, phone, email) VALUES (\'", @username, "\', \'", @password , "\', \'", i\_role, "\', \'", i\_fname , "\', \'", i\_lname, "\', \'", i\_mname , "\', \'", @d\_date\_of\_birth, "\', \'", i\_address , "\', \'", i\_phone, "\', \'", i\_email ,"\');");*

*PREPARE stmt FROM @sql\_text;*

*EXECUTE stmt;*

*DEALLOCATE PREPARE stmt;*

*SET o\_username= @username;*

*SET o\_password= @password;*

*END*

**c. sf\_get\_get\_username**

The function uses to generate username of an account based on first name, lastname, and middle name

*CREATE FUNCTION `sf\_get\_username`(`i\_username` VARCHAR(32)) RETURNS varchar(32)*

*BEGIN*

*SET @username= i\_username;*

*SET @index= 1;*

*getunameloop: LOOP*

*SELECT COUNT(username) FROM `users` WHERE username= @username INTO @check\_exist;*

*IF @check\_exist <> 0*

*THEN*

*SET @username= concat(i\_username, @index);*

*SET @index= @index+ 1;*

*ITERATE getunameloop;*

*END IF;*

*LEAVE getunameloop;*

*END LOOP getunameloop;*

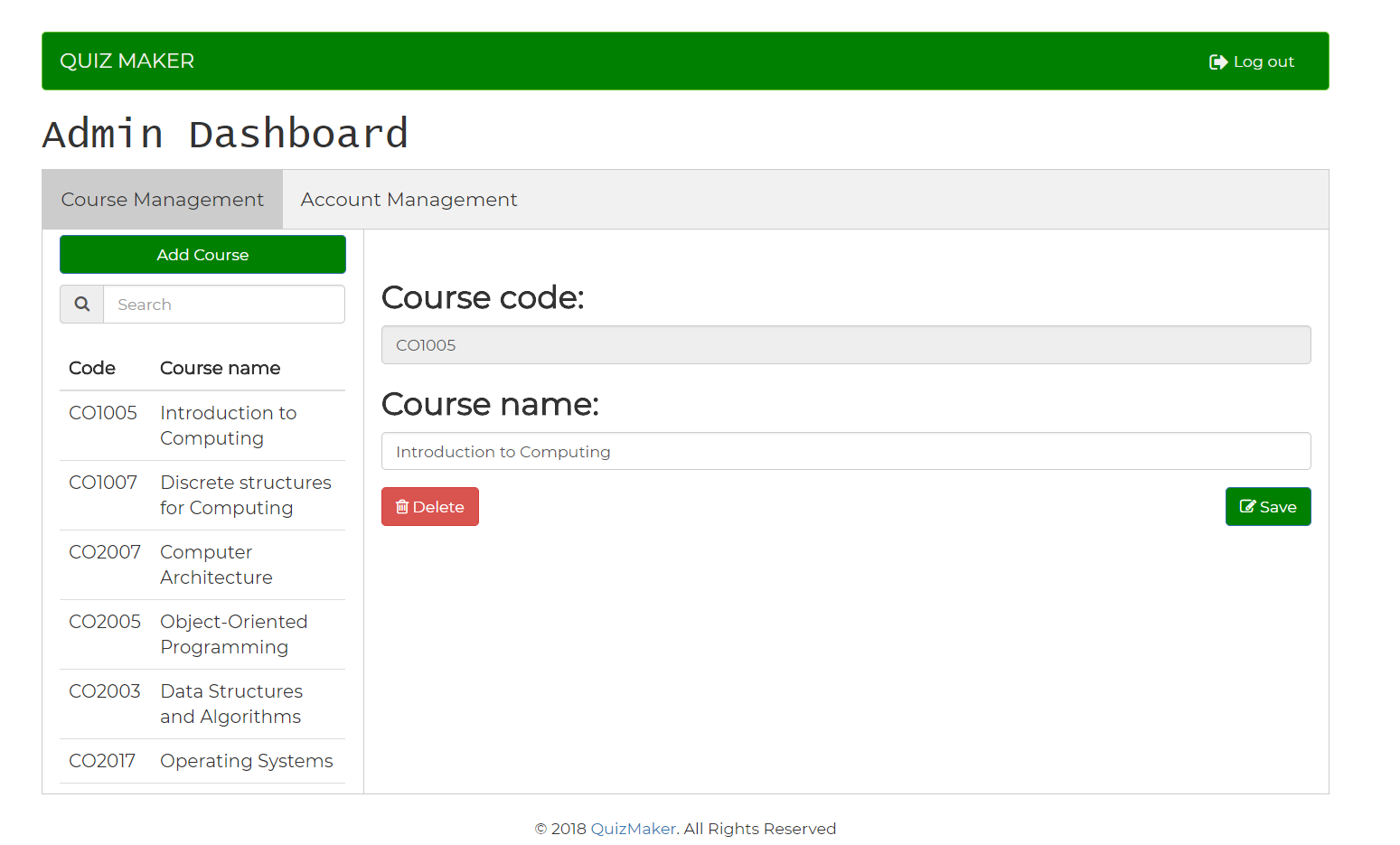
*RETURN @username;*

*END*

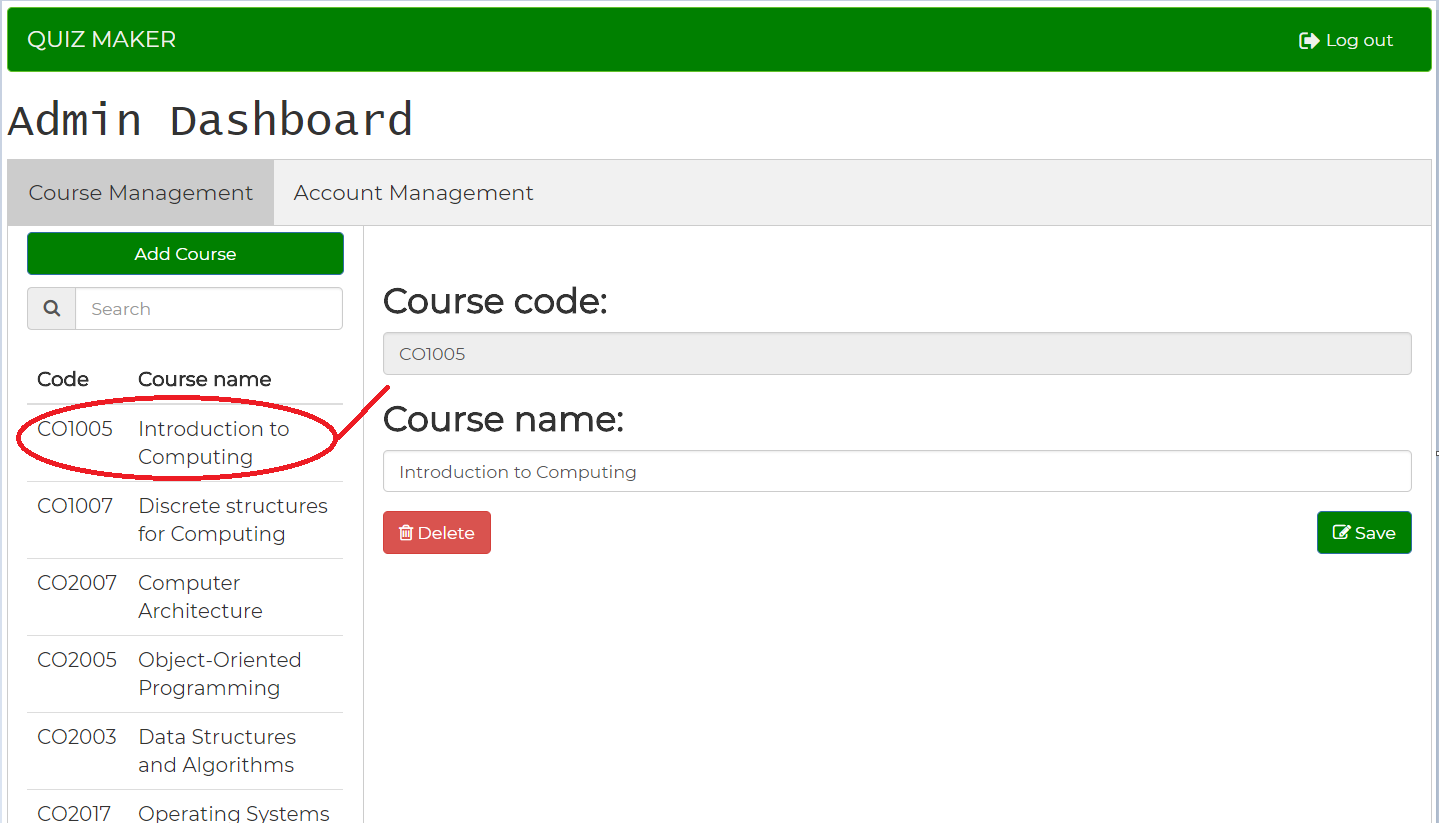
1. **ADMIN DASHBOARD**
2. **Overview**

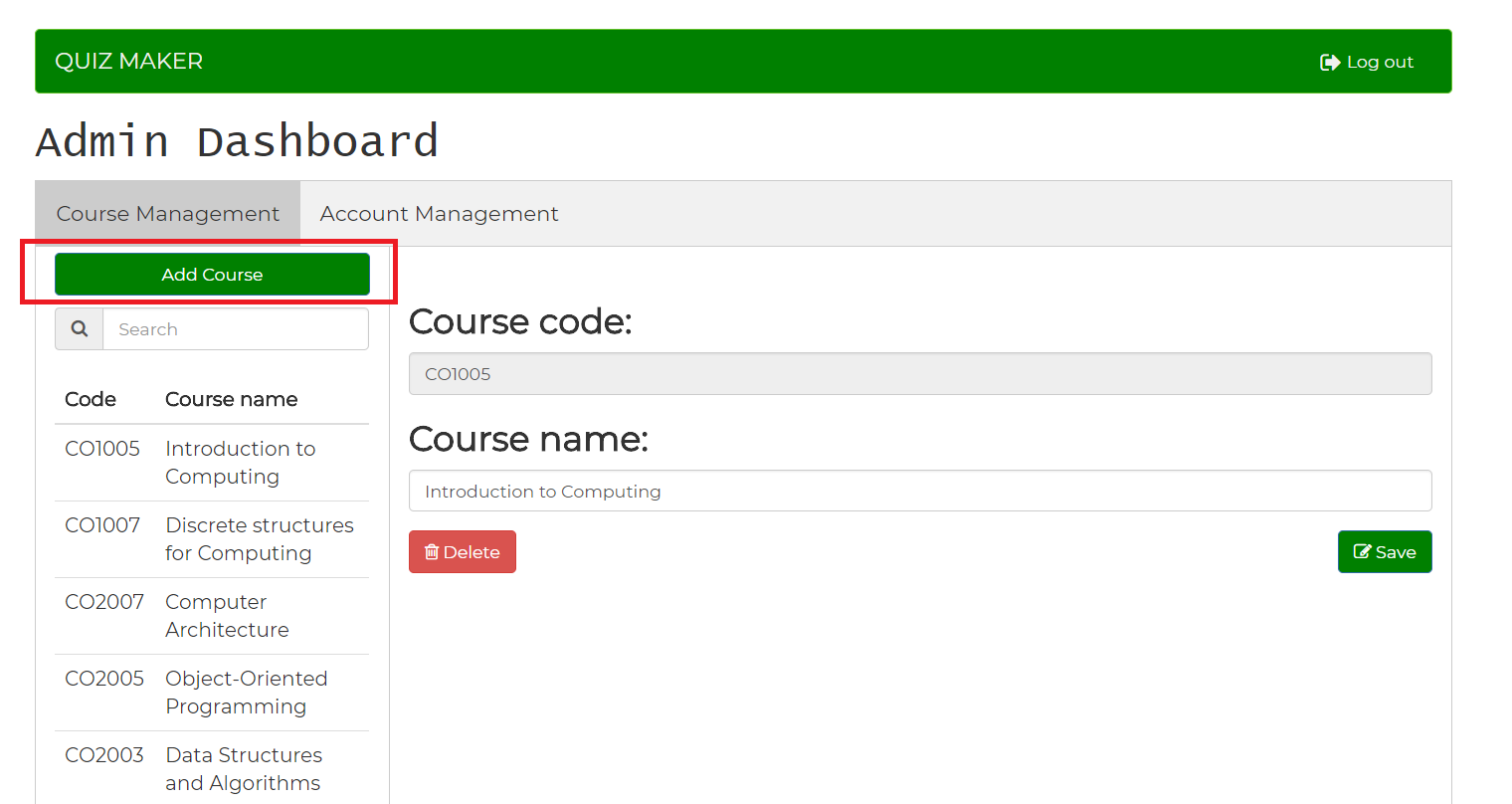
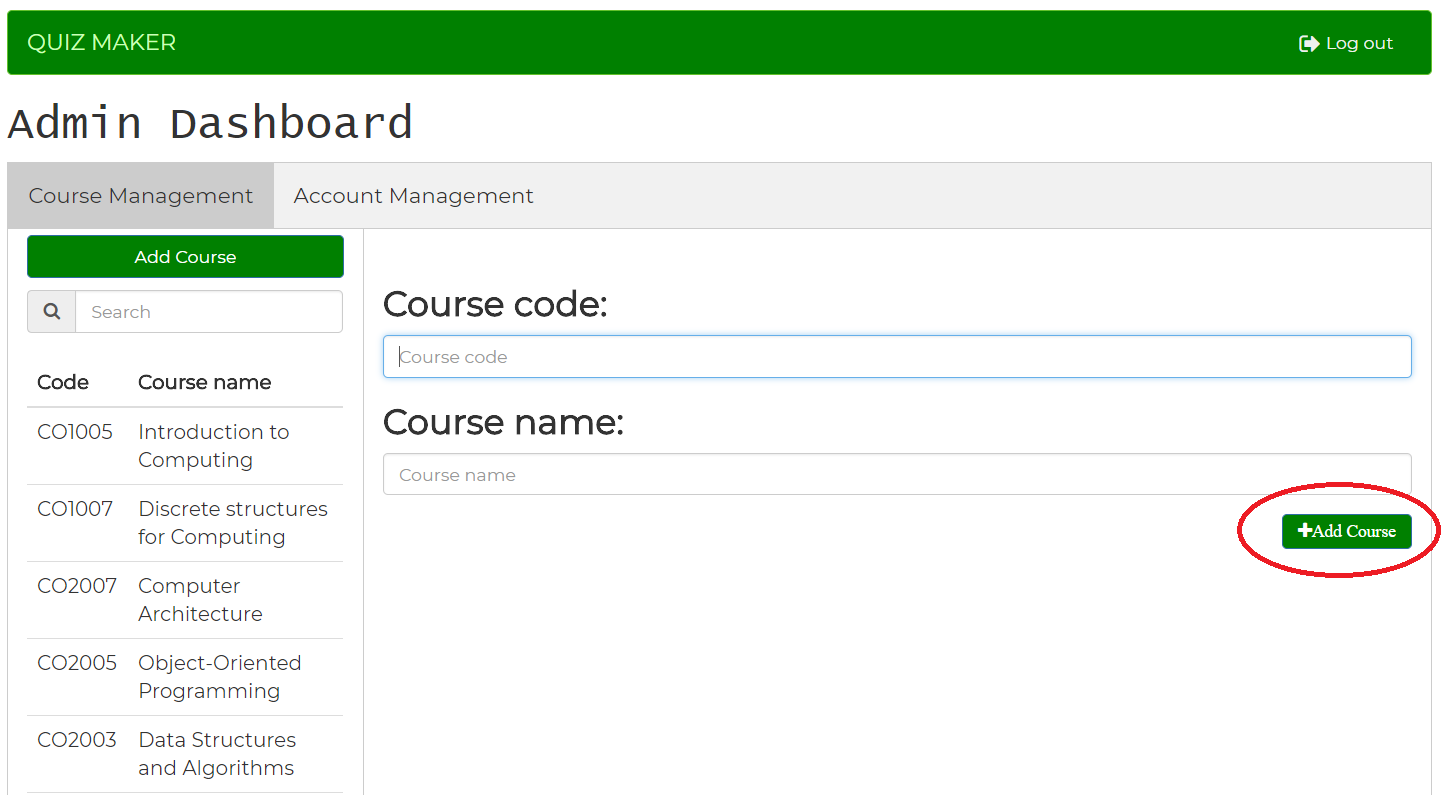
Admin Dashboard supports admin in managing courses and accounts. In detail, admin can create new courses or accounts, modify information of any course or account, as well as delete those.

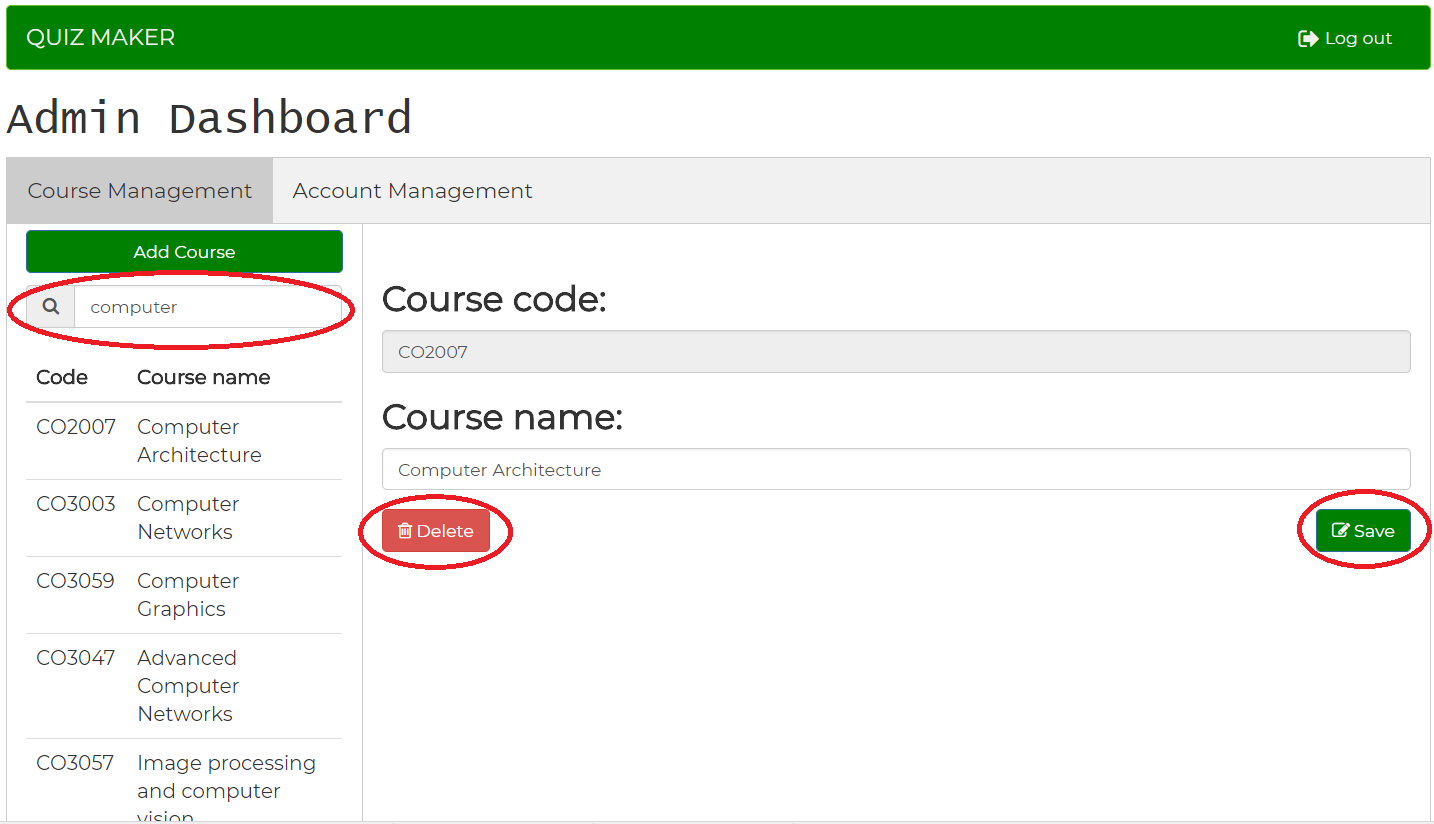
Admin Dashboard is made to simplify admin’s work, help them increase their performance and accuracy.

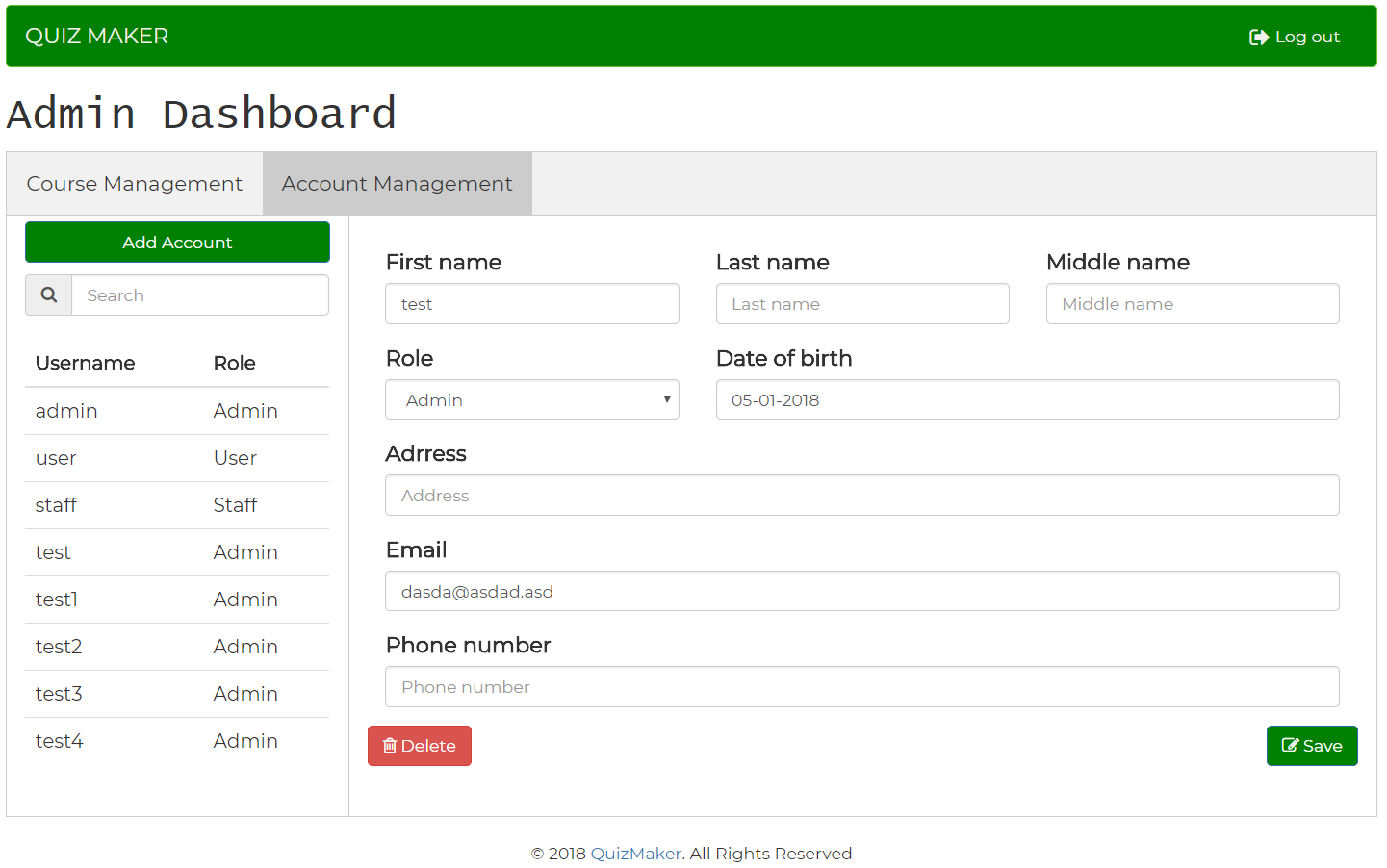
1. **Guide**

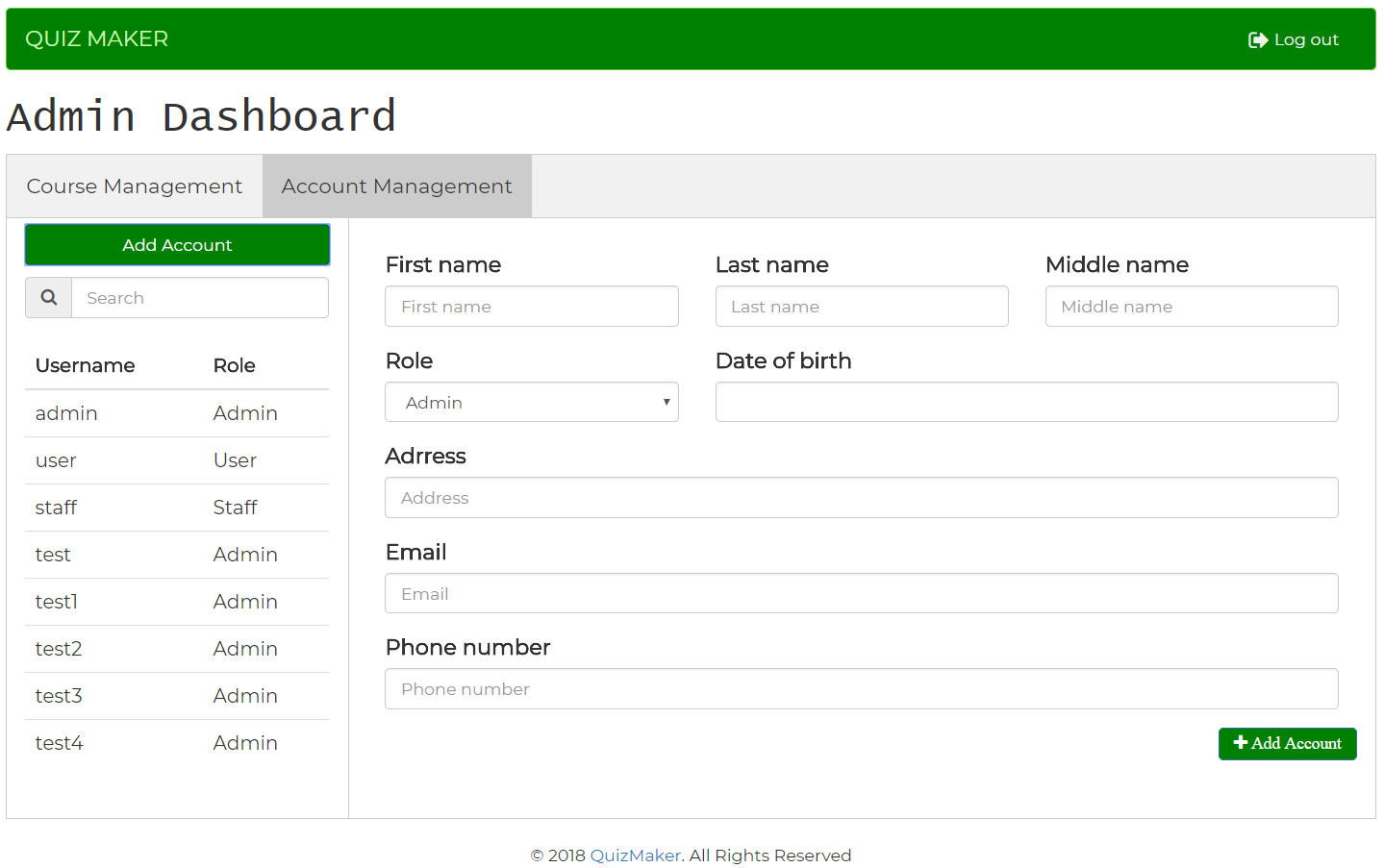
* The picture above is the main window of admin dashboard
* There are two main tabs in the dashboard: **Course Management** tab and **Account Management** tab
* Firstly, in **Course Management** tab:
  + The default window is the information of the first course in the left navigation bar



* + If we need to create a new course, press Add Course button, create window will appear
  + Fill course code and course name in the form, then press Add Course button to create a new course (Course code is in the format of CO[0-9]{4}. Example: CO1234)
  + Next, if we want to modify information of a course, choose the course in the navigation bar. We also can search for a course in the search bar by fill some letters corresponding to the course in order to quick searching
  + Modify course name, then click Save to confirm.
  + We can also delete that course by clicking Delete button



* Secondly, we can do the same things with **Account Management** tab (Create, Search, Modify, Delete)



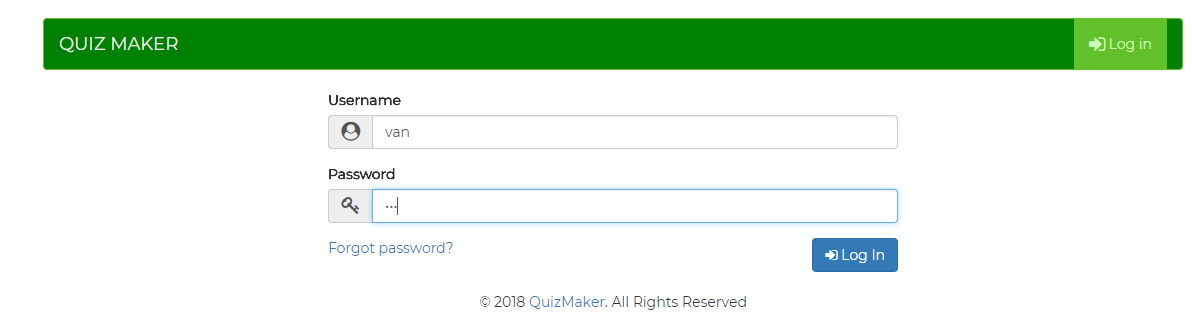
1. **References**

* Google.com

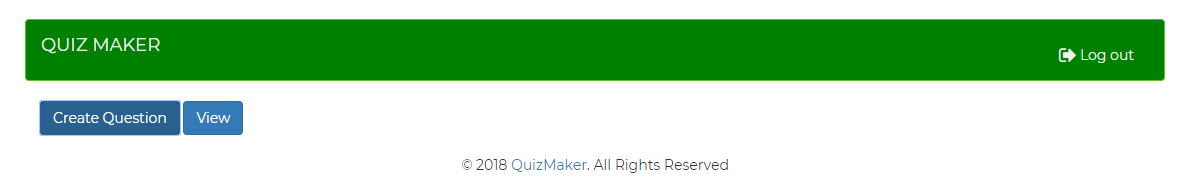
1. **USER**
2. **Overview**

**User accounts** are created for users, who can create questions, options, answers, and difficulty easily. In addition, **user accounts** support users can modify or delete those things.

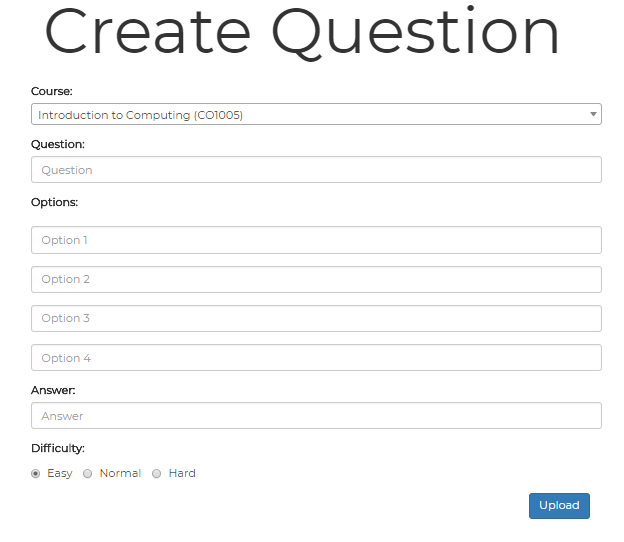
1. **Guide**
2. User sign in with account “van” and password.



1. Now the dashboard will appear.

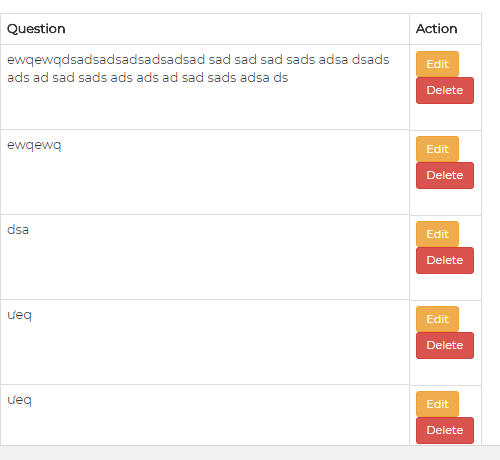


1. Click the tab “Create Question” to create a question.



* Course: the users can choose the correct courses in the list. ( click the arrow, the list will appear )
* Question: the users write down the question that they want to make.
* Options: the users can make 4 different options.
* Answer: the users support the answer of that question.
* Difficulty: have 3 levels to select.
* After finishing, the users click the **Upload** button to store all of those into the database.

1. Click the tab “View” to view all of those.



1. **STAFF**
2. **Overview**

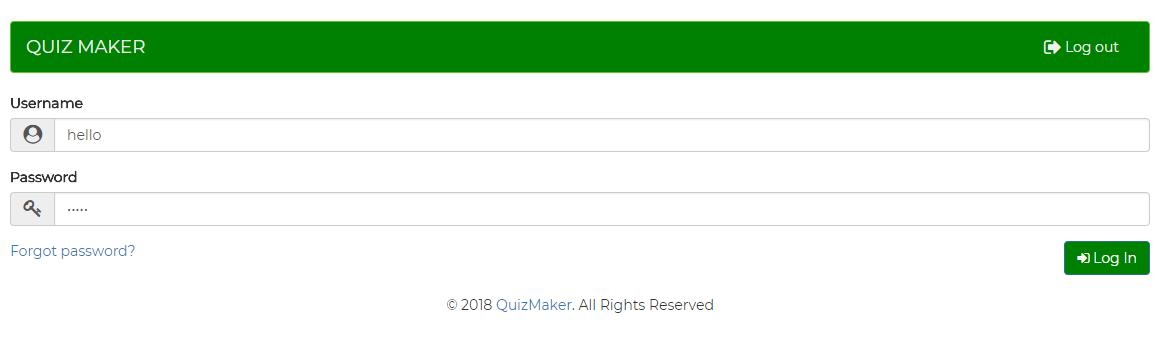
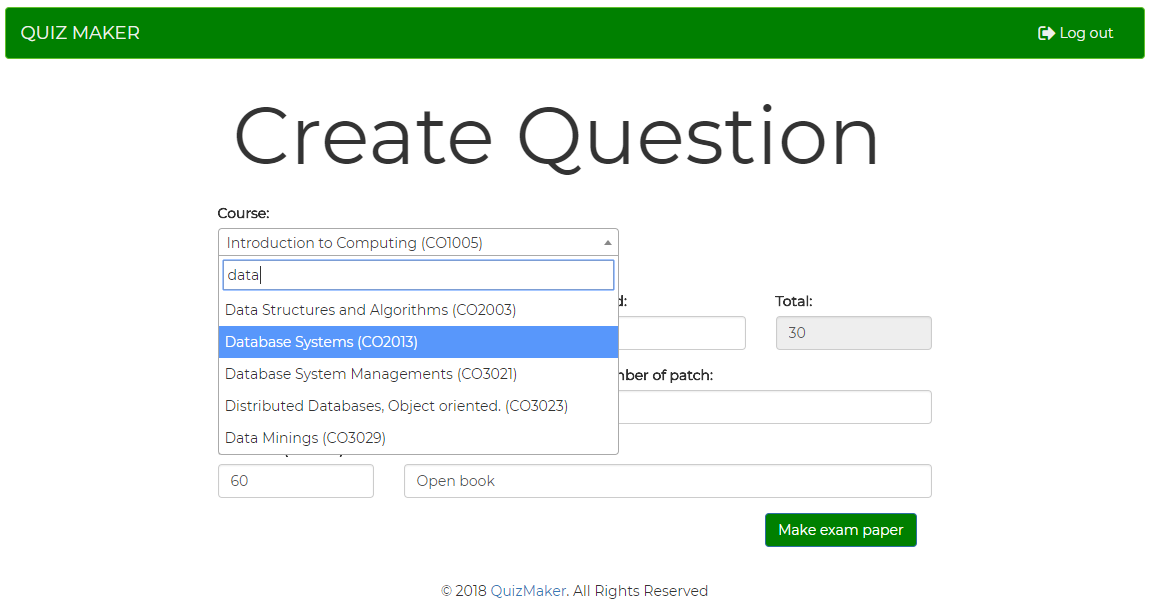
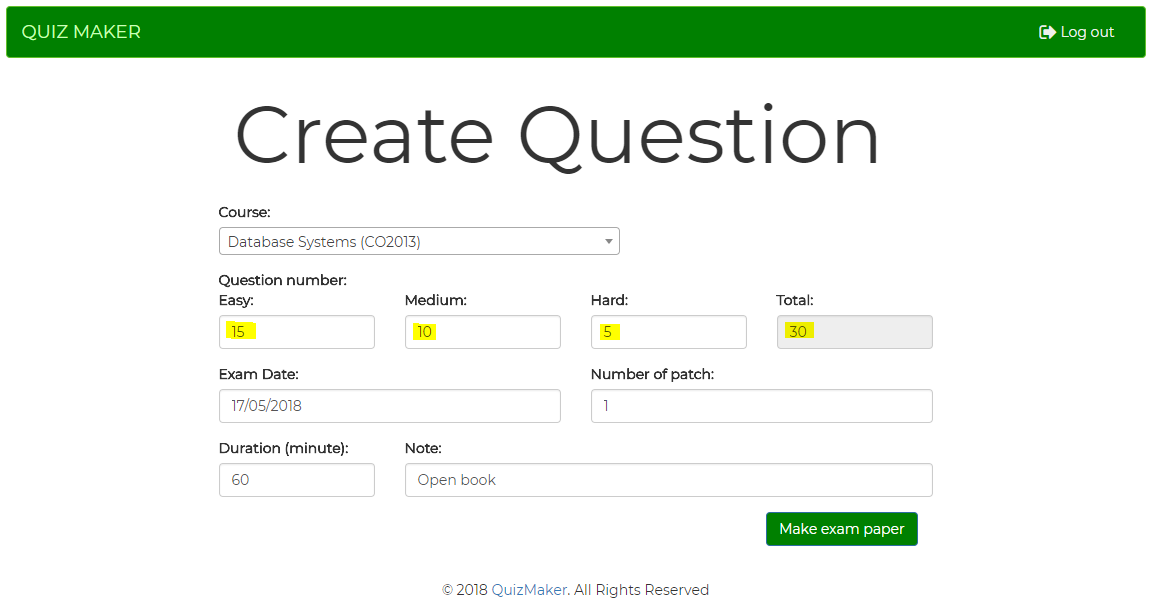
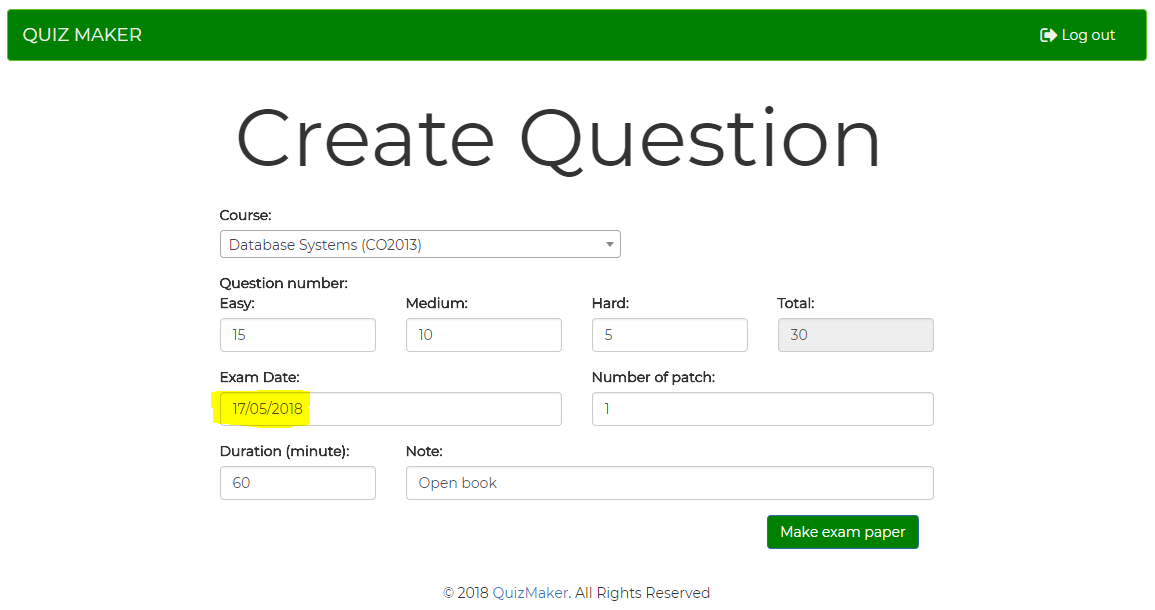
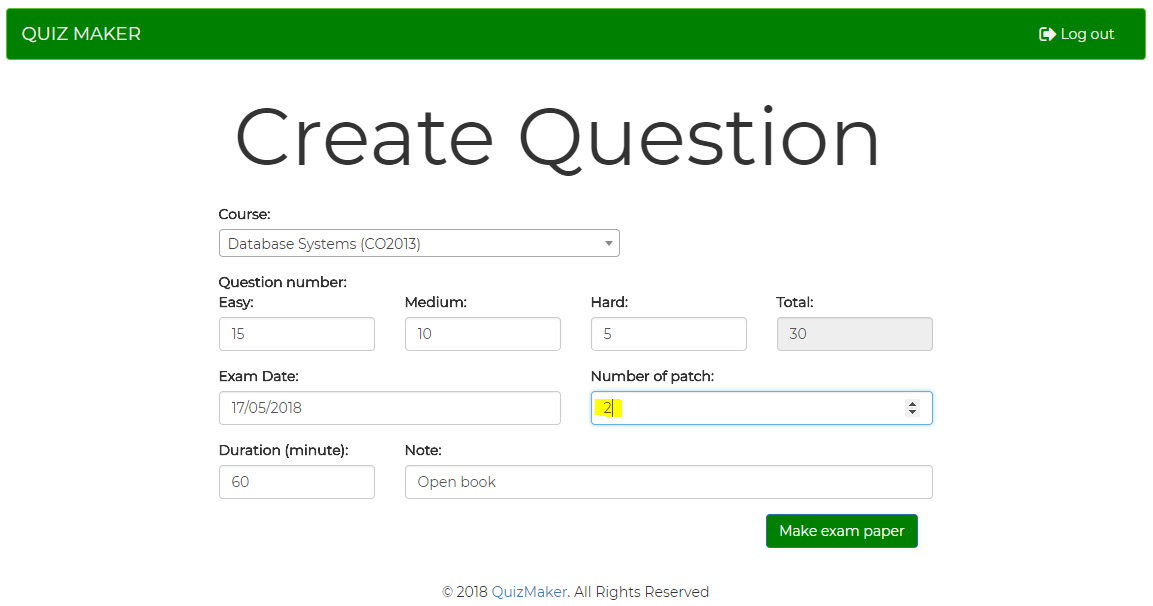
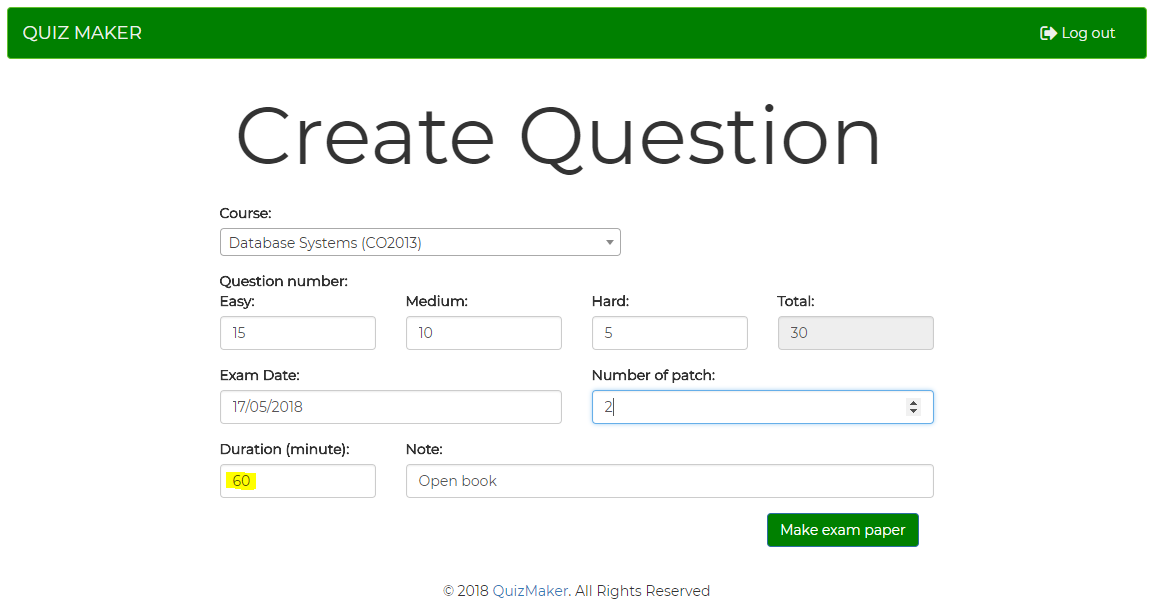
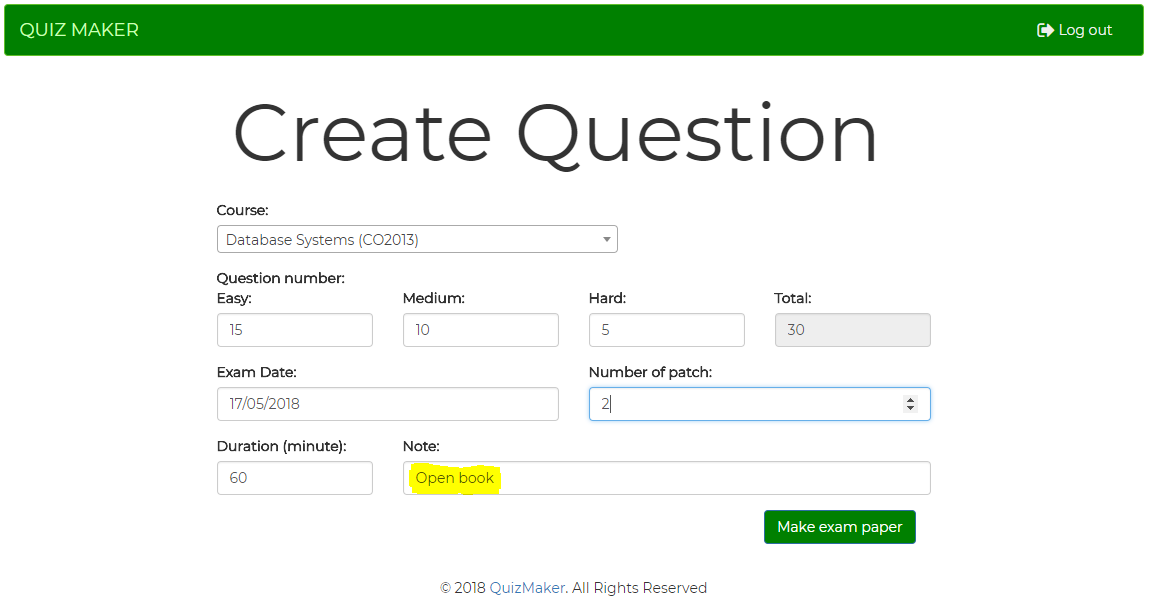
Staff is the creator of the test. When the staff needs to create a new test, enter the following information:

• Total number of multiple choice questions

• How many sentences, how many sentences and how many sentences?

Then the site will randomly ask the number of questions in each difficult level to form a multiple-choice test.

To create a test, the staff needs to follow the steps below:

1. Staff will be entering the system with the account and password.
2. Select a subject to create a test
3. Choose the number of questions at easy, medium and difficult levels
4. Exam day will be the current time
5. Choose the number of patch. Questions will be mixed to create different types of test
6. Select the duration of the test
7. Write notes for the test
8. Click on the “Make exam paper” button.
9. **Guide**
10. Staff will be entering the system with the account is “hello” and password.
11.  Staff want to create a test for Database Systems. He/she can search the course by name (Database Systems) or by code (CO2013).
12. He/she want to create a test with 30 multiple choice questions including 15 easy questions, 10 medium questions and 5 difficult questions .
13. Exam day will be the current time. He/she can customize another day.
14. He/she wants to create two patch.
15. He/she wants the test time to be 60 minutes. This 60 minutes will be printed on the header of the test.
16. He/she wanted to take notes on the exam paper as "open book".
17. After he/she click on the “Make exam paper” button. The test will be created like the image below.

