

Имена: Александрина Здравкова Каракехайова **фн:** 81097
Начална година: 2014 **Програма:** бакалавър, (КН) **Курс:** 4
Тема: Автоматизирана тестова система за домашни работи
Дата: 2018-01-24 **Предмет:** wwwTech2017_18_9ed_KN_winter
имейл: alexandrinakarakehayova@gmail.com

преподавател: доц. д-р Милен Петров

ТЕМА: Автоматизирана тестова система за домашни работи

1. Условие

Да се създаде тестова система за домашни работи. Преподавателят трябва да може да качва условия на домашните работи, както и да определя начален и краен срок за всяка задача, като сроковете трябва да могат да бъдат променяни. Преди крайния срок преподавателят трябва да е добавил и файл с тестове, който да се изпълни автоматично върху предадените от студентите задания в 00.00 часа на датата, посочена като краен срок или да се изпълнят ръчно от админ панела чрез натискане на бутон. Студентите трябва да могат да добавят към всяко задание решенията си, като могат да ги променят неограничен брой пъти до настъпване на крайния срок. След като се изпълнят тестовите всеки студент получава автоматично генерирана оценка на база на успешните и неуспешните тестове. Студентът получава две оценки – автоматичната и от преподавателя. Финалната оценка за домашното се формира по формула, която се задава отново в админ панела. Преподавателят трябва да може да качва наведнъж всички домашни, но те да бъдат видими едва след посочената като начална дата.

2. Въведение

При успешен логин в глобалната променлива `$_SESSION` се записва потребителят и се редиректва към началната страница, в зависимост от ролята. При логアウト сесията се унищожавя и се редиректва към логин страницата. /на всяко зареждане на страница се прави проверка дали има записан потребител в сешън променливата/

3. Теория

Архитектурата на приложението е базирана на Model-View-Controller концепцията. Логиката е разделена от графичния интерфейс и данните. Друго използвано ниво на абстракция в проекта е Repository. Това е междинен слой, който осъществява комуникацията между базата данни и контролерите. Там се изпълняват операции върху данните, взети от базата, и се предават в необходимия вид на функциите, които ще ги обработват. Приложението използва psr-4 autoloading за по-лесно зареждане на класове.

4. Използвани технологии

Използваните технологии при направата на проекта са: PHP 7.1 за сървърната част, HTML и CSS за дизайна на потребителския интерфейс, MySQL за запазване на данните, AWS – за деплой на проекта и Codeception, Selenium Webdriver, Chromedriver и Google Chrome за писане и изпълнение на функционални и юнит тестове.

5. Инсталация и настройки

За успешната работа с приложението се нуждаем от 5 таблици с данни /приложен е файл със скрипт, генериращ базата/ - записаните студенти в курса (таблица `users`), настройки на админа (`settings`), регистриралите се студенти (`course_students`) и домашните, качени от

преподавателя (homeworks), решенията на студентите (student_homework). За да се стартира проектът трябва да се изпълни скрипт, в който се инсталират необходимите пакети, например: php7.1-zip, php7.1-curl, php7.1-xml и др. и програми като apache, mysql, php, java, selenium и др. Скриптът прави и конфигурации като създаване на база – извиква sql скрипта, чрез същия скрипт се създава и крон джоба и се конфигурира apache, както и средата за тестване. Файл със скрипта е приложен.

Проектът е деплойнат на ubuntu сървър, хостнат от AWS. Процесът включва следните стъпки: създаваме EC2 (Elastic Compute Cloud) инстанция. Избираме Ubuntu 16.04. Преди launch на инстанцията избираме type t2.micro (тъй като е безплатен). След това генерираме сертификат и го изтегляме. Запазвам го в папка .ssh. Променяме му правата с chmod 400. Влизаме в създадената инстанция, където виждаме нашия Public DNS. За да се свържем изпълняваме командата

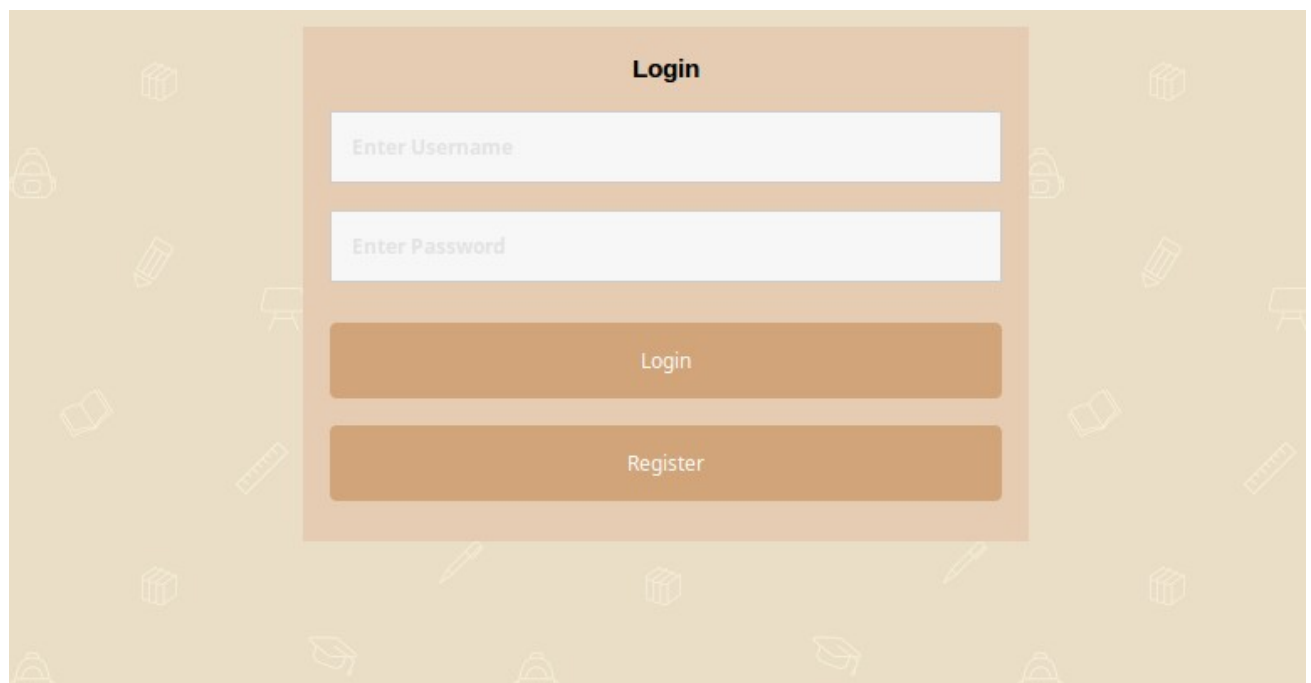
```
ssh -i ~/.ssh/certificate.pem ubuntu@publicDNS
```

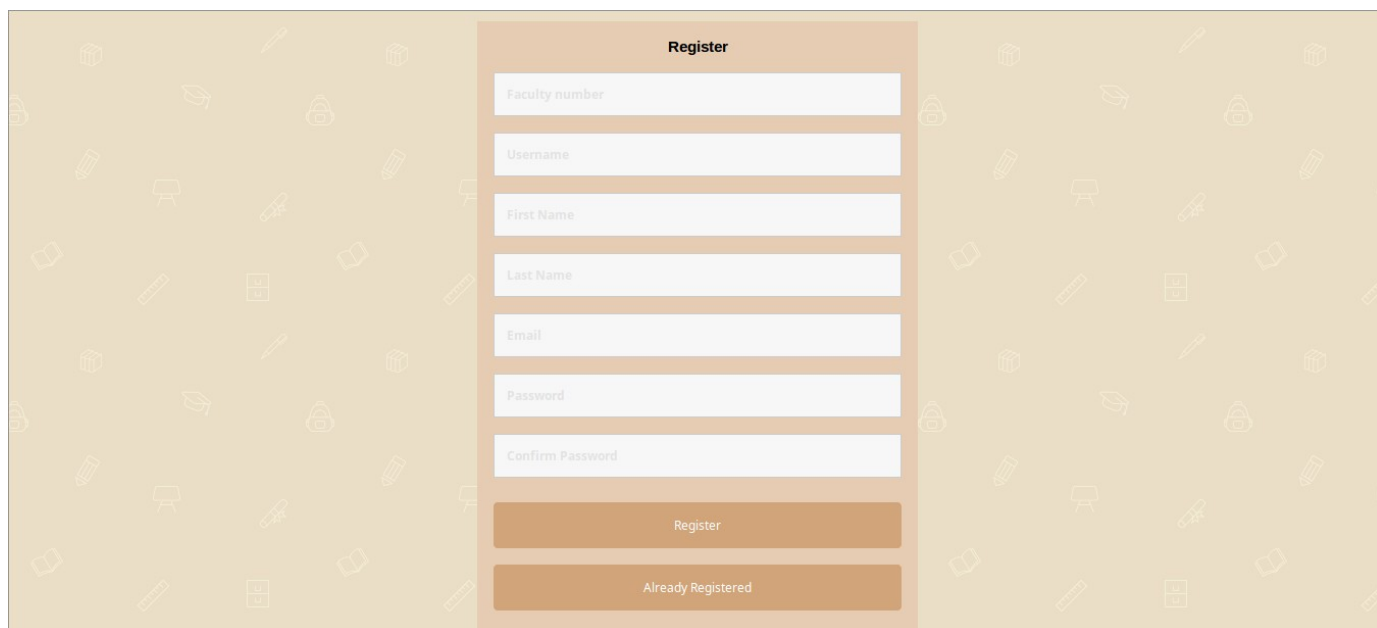
където certificate е името на сертификата, който сме изтеглили, а publicDNS, е генерирания при създаването на инстанцията DNS. Тук изпълняваме приложения скрипт за setup на проекта.

След това в Dashboard на AWS, когато изберем създадената инстанция в таб description, security groups – избираме група и задаваме inbound rule – в случая type - all traffic, source - anywhere.

6. Кратко ръководство на потребителя

На началната страница има навигация, която води до login форма и форма за регистрация. Могат да се регистрират само факултетните номера, зададени от преподавателя. Ако не успеете да си създадете профил, то или сте объркали факултетния си номер, или трябва да се обърнете към преподавателя си да ви добави, като ще получите съответното съобщение.

The image shows a web form titled "Login" in a dark orange box. The form has a light orange background with a pattern of small icons. It contains two input fields: "Enter Username" and "Enter Password", both with placeholder text. Below these fields are two buttons: "Login" and "Register", both in a darker orange color. The "Login" button is positioned above the "Register" button.

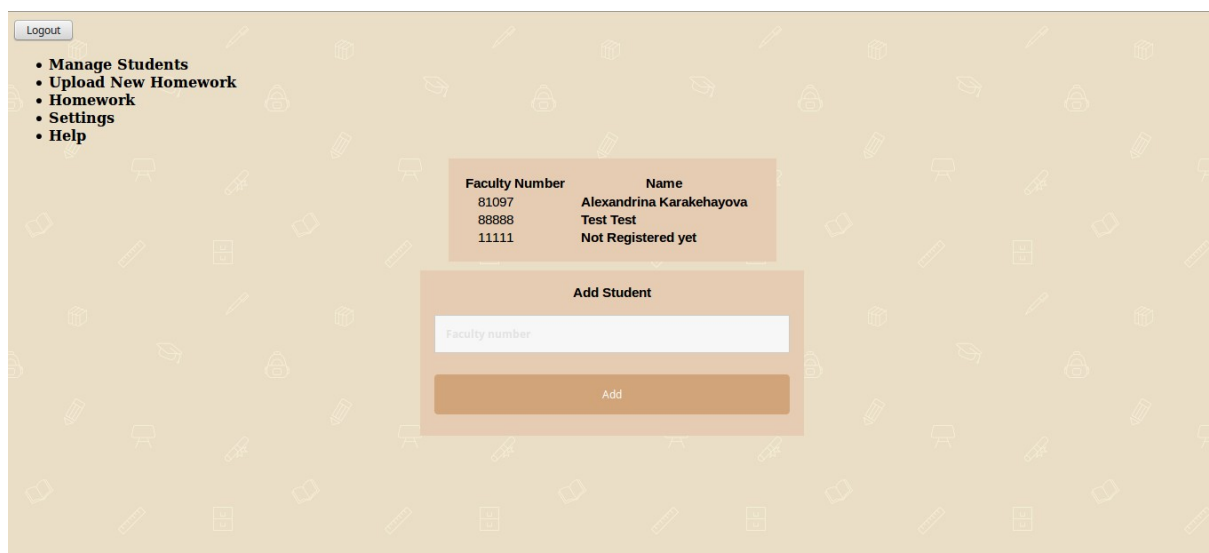
A registration form titled "Register" is centered on a light beige background with a repeating pattern of small icons (books, pencils, graduation caps). The form consists of several input fields stacked vertically: "Faculty number", "Username", "First Name", "Last Name", "Email", "Password", and "Confirm Password". Below these fields are two buttons: "Register" and "Already Registered".

Register	
Faculty number	
Username	
First Name	
Last Name	
Email	
Password	
Confirm Password	
Register	
Already Registered	

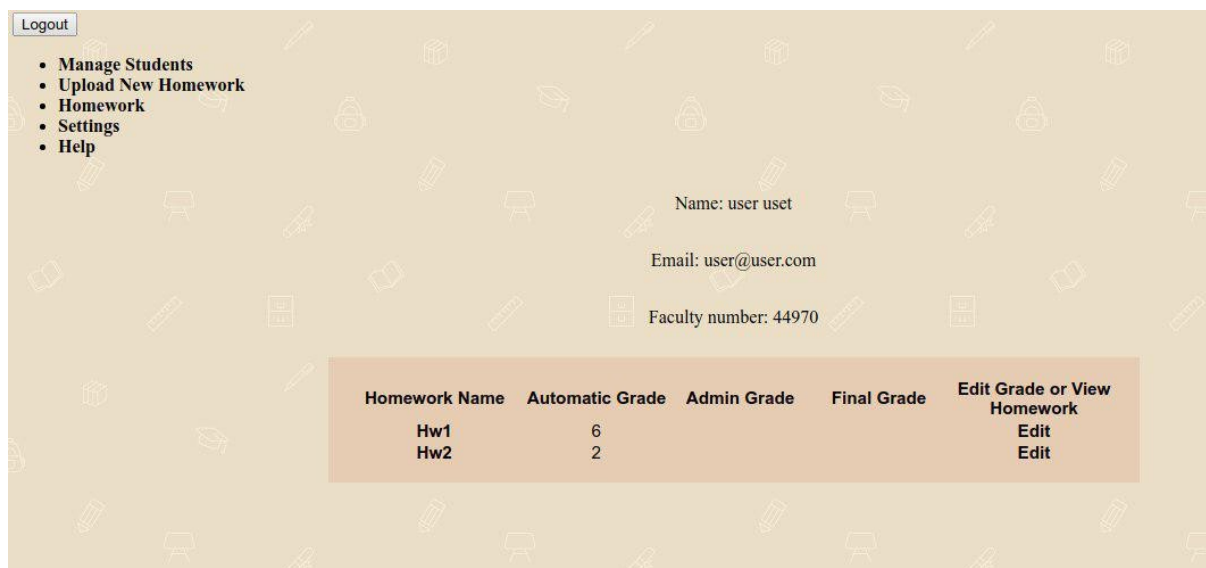
След като успешно влезете в своя профил, там ви очаква меню, в зависимост от ролята – админ или студент.

Админ

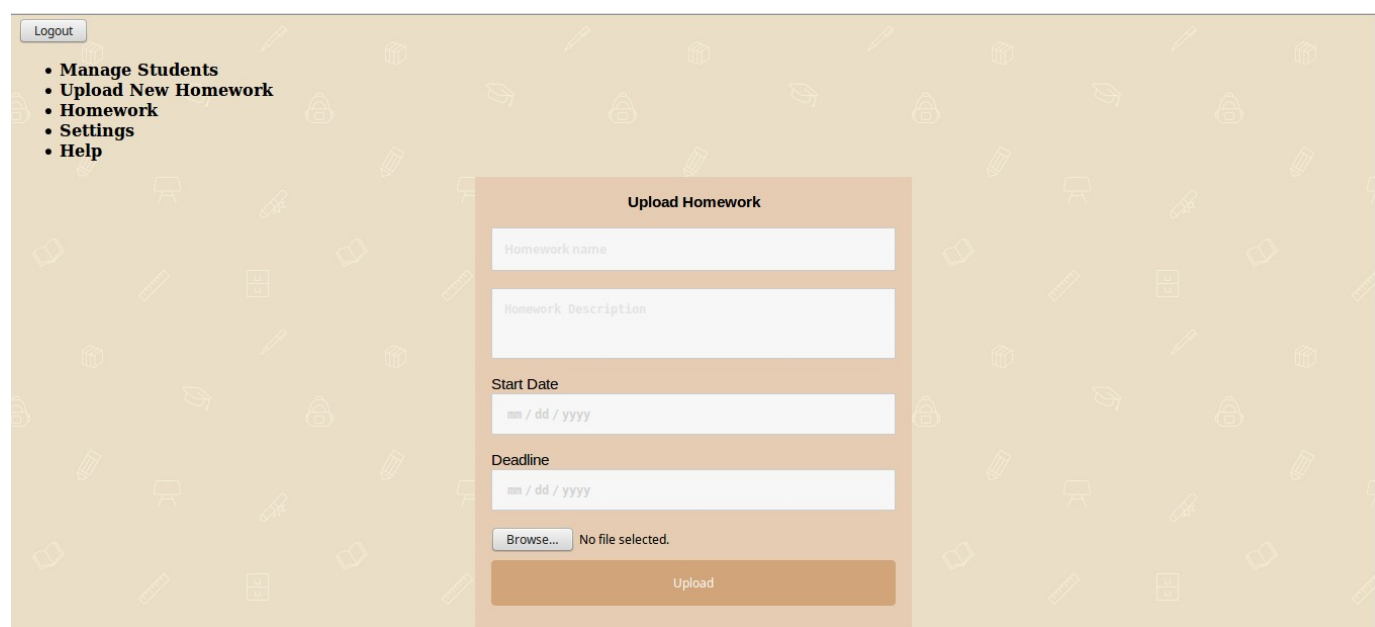
В менюто на админ панела има 5 опции. Първата е Manage Students, където преподавателят може да добави факултетен номер на студент. Тук вижда също дали добавените студенти са регистрирани или все още не. Ако са регистрирани, когато кликне върху името им се препраща към страница с информация за потребителя – имена, имейл, факултетен номер, както и всички предадени домашни от съответния студент.

The admin panel features a "Logout" button in the top left corner. Below it is a menu with five items: "Manage Students", "Upload New Homework", "Homework", "Settings", and "Help". The main content area displays a table of students and an "Add Student" form. The table has two columns: "Faculty Number" and "Name". It lists three students: one with faculty number 81097 and name Alexandrina Karakehayova, one with 88888 and name Test Test, and one with 11111 and name Not Registered yet. Below the table is a form titled "Add Student" with a "Faculty number" input field and an "Add" button.

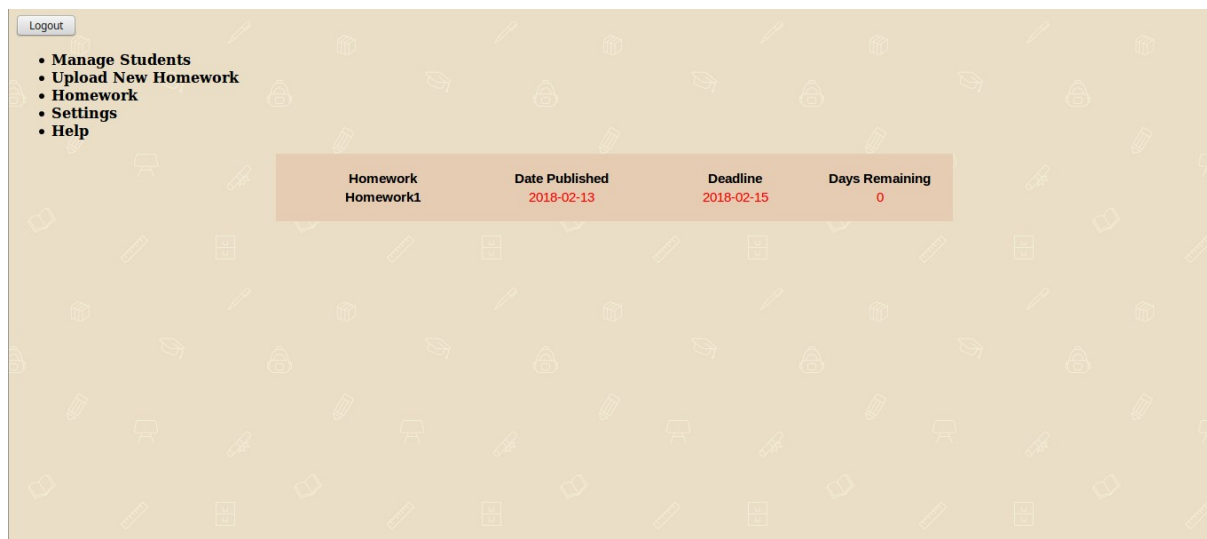
Admin Panel									
Logout									
<ul style="list-style-type: none">• Manage Students• Upload New Homework• Homework• Settings• Help									
<table border="1"><thead><tr><th>Faculty Number</th><th>Name</th></tr></thead><tbody><tr><td>81097</td><td>Alexandrina Karakehayova</td></tr><tr><td>88888</td><td>Test Test</td></tr><tr><td>11111</td><td>Not Registered yet</td></tr></tbody></table>		Faculty Number	Name	81097	Alexandrina Karakehayova	88888	Test Test	11111	Not Registered yet
Faculty Number	Name								
81097	Alexandrina Karakehayova								
88888	Test Test								
11111	Not Registered yet								
<div>Add Student Faculty number Add</div>									



От опцията Upload New Homework преподавателят добавя нова домашна с начален и краен срок, заглавие, условие и качва тестове към нея.



Третата опция е Homework. Тук преподавателят вижда качените вече домашни.



Когато кликне върху името на домашната, вижда нова страница, на която има форма за редактиране на условие, срокове или качване на нови тестове. Малко по-долу има списък с всички студенти, предали съответното домашно, както и техните оценки. Бутонът Test изпълнява тестовете върху всички предадени домашни до момента. От Download Uploaded Tests преподавателят може да види какви тестове е качил и да ги изтегли. При клик на User ID на студент, преподавателят ще бъде препратен към страницата с информация за този студент.

[Logout](#)

- [Manage Students](#)
- [Upload New Homework](#)
- [Homework](#)
- [Settings](#)
- [Help](#)

Edit Homework

Start Date

Deadline

Download Uploaded Tests
Upload New Tests No file chosen

User ID	Name	Email	Faculty Number	Automatic Grade	Admin Grade	Final Grade	Uploaded at
2	User1	user@user.com	44970	2			2018-02-14 22:56:46

При клик върху името на студента ще бъде препратен към страница, където може да види output от тестовете, да види как изглежда домашното, в случай че има html, както и да си изтегли файловете на студента, ако иска да ги разгледа. Тук има също така форма, в която преподавателят да въведе своята оценка. Финалната оценка ще се изчисли по формулата, която се вижда.

Homework Test Output

Link to Homework

Download Code

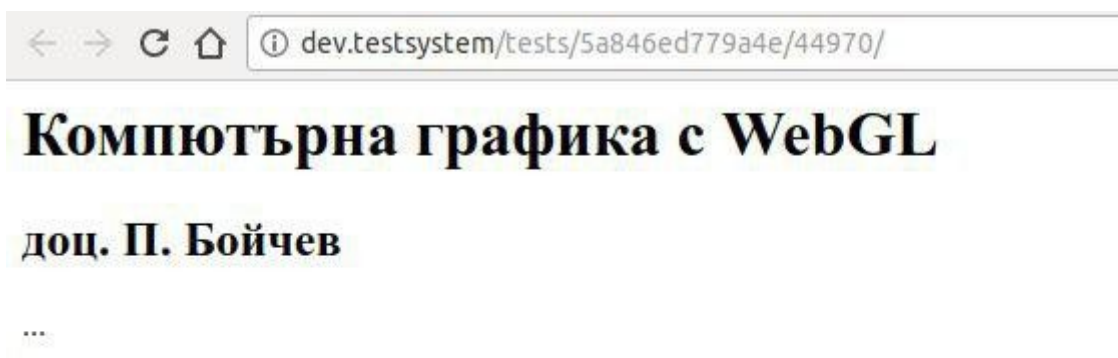
Current Formula: $(x+y)/2$

Automatic Grade

Admin Grade

Final Grade

Следващият скрийшот показва Link to Homework за примерно домашно.



Codeception създава html файлове при фейл на тест. Примерен output от тестове за домашно:

dev.testsystem/tests/5a846ed779a4e/44970/tests/_output/

Index of /tests/5a846ed779a4e/44970/tests/_output

Name	Last modified	Size	Description
 Parent Directory		-	
 Homework9TestCest.tryToTest.fail.html	2018-02-15 00:57	345	
 Homework9TestCest.tryToTest.fail.png	2018-02-15 00:57	20K	
 failed	2018-02-15 00:57	48	

Apache/2.4.18 (Ubuntu) Server at dev.testsystem Port 80

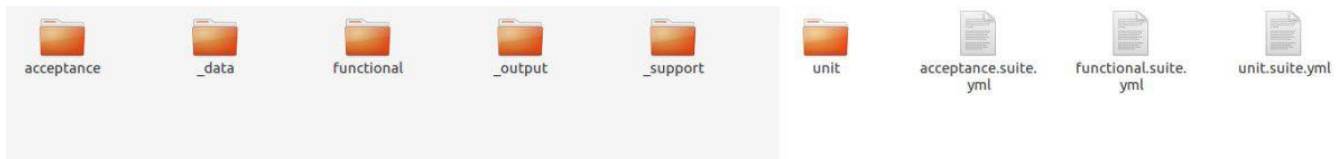
Следващата опция от менюто е Settings. Тук преподавателят задава формулата, по която се изчислява финалната оценка. Работата с тази форма трябва да бъде малко по-внимателна, защото за да се изпълни подадения стринг като математическа формула се използва eval и може да доведе до нежелан ефект при неправилна употреба.

Formula used to calculate the final grade from the automatic grade(x) and the admin grade(Y). Example: 2x + y. Change it carefully as incorrect format might cause errors

(x+y)/2

Save

Как качваме домашни в системата?



Това са нужните директории и файлове, които преподавателят трябва да качи за да се случи автоматичното тестване. Те трябва да бъдат архивирани в zip. Те се генерират автоматично от codeception. Трите yml файлове са конфигурационни. Те трябва да бъдат редактирани спрямо конкретното домашно. Например в acceptance.suite.yml може да се укаже дали за тестването да се използва PHPBrowser или WebDriver. Бихме предпочели използването на WebDriver, когато се изисква да има JavaScript функция в домашното, което искаме да тестваме, тъй като PHPBrowser не го поддържа и подминава JavaScript кода.

При unit тестове трябва да копираме винаги следния ред, видим на следващия скрийншот:

```
Autoload::addNamespace('', 'var/www/html/TestSystem/public/tests/{$hwId}/{$student}');
```

Това е за автоматичното зареждане на класовете, дефинирани от студента.

```
public function testSomeFeature()
{
    Autoload::addNamespace('', 'var/www/html/TestSystem/public/tests/{$hwId}/{$student}');

    $class = new HomeworkOne();

    $data = array(
        'webgl' => array(
            'title' => 'Компютърна графика с WebGL',
            'description' => '...',
            'lecturer' => 'доц. П. Бойчев',
        ),
        'go' => array(
            'title' => 'Програмиране с Go',
            'description' => '...',
            'lecturer' => 'Николай Бачийски',
        )
    );

    $this->assertEquals(
        '<h1>Компютърна графика с WebGL</h1><h2>доц. П. Бойчев</h2><p>...</p>',
        $class->show_page('webgl', $data)
    );
}
```


7. Примерни данни

Примерно попълване на формата за регистрация:

faculty number: 81097

username: Alex

first name: Alexandrina

last name: Karakehayova

email: alexandrinakarakehayova@gmail.com (трябва да бъде валиден имейл)

password и confirm password: двете пароли трябва да съвпадат и да бъдат над 8 символа

Примерно попълване на формата за качване на условие за домашно и файл с тестове от преподавател:

Homework name: Homework1

Homework description: Условие на домашната

Start date и end date се избират от календара. Ако са невалидни(end date е преди start date например) се извежда съобщение след опит за submit.

8. Описание на програмния код

При регистрация на студент запазваме паролата му хеширана, като хеширането се извършва с помощта на функцията `crypt`. Тя приема като първи параметър паролата, а като втори `salt`, който се взима от конфигуриран файл на проекта (`config/config.php`). В `config` файла има освен това необходимите данни за MySQL.

В проекта има `.htaccess`, който пренасочва всяка заявка към `/public/index.php`, като предава всичко след domain name на url в променливата `$_GET['url']`. В `index.php` се зарежда `autoloader`, `config`, както и дефинираните пътища в `routes/routes.php`. В `routes.php` са регистрирани всички `get` и `post` заявки. След това се взима текущия въведен url и се проверява дали е регистриран. Ако е регистриран се взимат контролерът и методът, асоциирани с този път, контролерът се инстанцира и се извиква методът, на който се подават параметри, ако има такива в url-а.

Как се качват файловете от преподавателя и студентите и как се изпълняват тестовете?

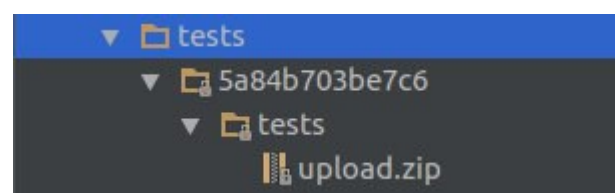
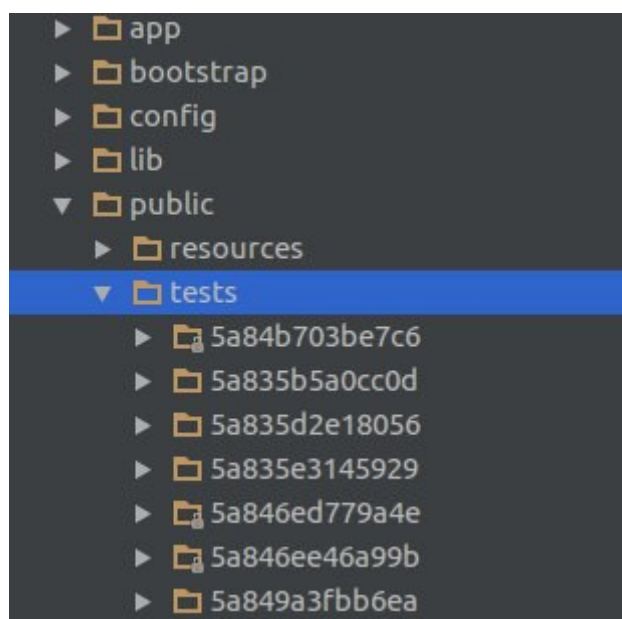
Студентите трябва да качват файловете си, именувани точно както е в условието, за да може автоматичните тестове да ги открият и да могат да минат успешно. Те трябва да ги архивират като `zip`, името на зип-а няма

значение. Трябва да се спазват стриктно условията на домашните, ако е описано функцията да бъде сложена в определен клас, то това трябва да бъде направено за благоприятно изпълнение на тестовете.

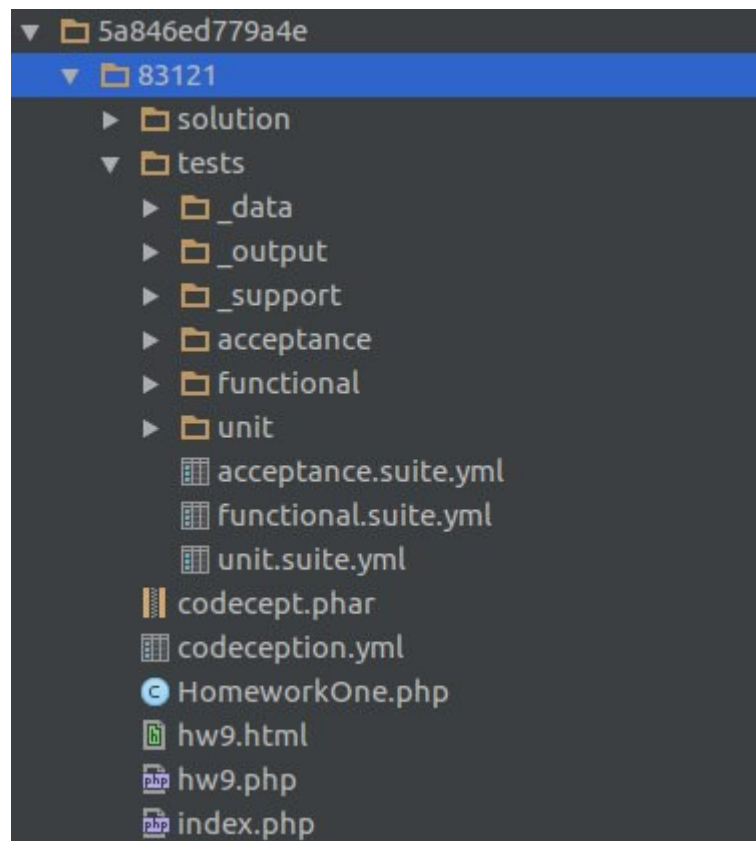
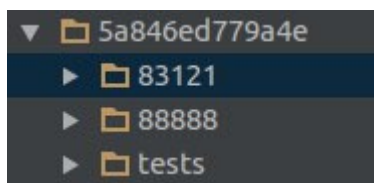
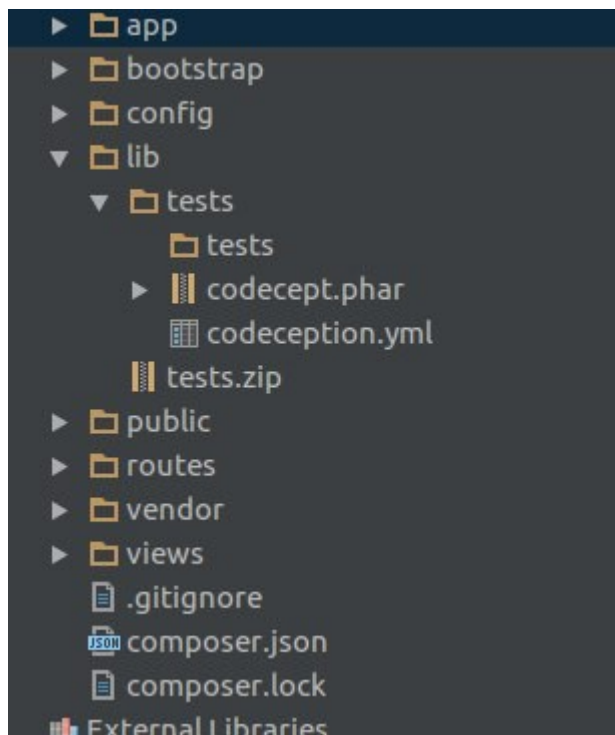


Процес на качване на домашни от преподавателя

Създава се уникално ид на домашната, което се запамятава в базата. След това се създава директория с това ид в public/tests. В нея се създава друга директория tests и в нея се слага каченият архив от преподавателя.



При качване на архив от студента се взима уникалното ид на домашната от базата и в тази директория се създава нова директория с име факултетния номер на студента. В нея се разхвърля съдържанието на lib/tests.zip. Това е source кода на codeception, неговия конфигурационен файл, както и празна директория tests.



В нея се разархивира архива с тестовете към тази домашна, качен от админа. Създава се папка `solution`, в която се записва архива на студента, който може да бъде изтеглен от преподавателя, както по-горе беше описано. Този архив се разархивира на `root` ниво в папката с факултетния

номер на студента и се получава структурата, показана на скрийншота по-горе.

Как се изпълняват тестовете?

Изпълняваме тест върху конкретно домашно – взимаме уникалното ид на домашното и сканираме директорията. Взимаме всички папки и в тях изпълняваме командата 'php codecept.phar run'. Резултатът от тази команда се обработва – взима се броя на assertion-ите, броят на успешните и неуспешните assertion-и и се калкулира оценка спрямо тях. Ако тестът е минал без неуспешни assertion-ите, резултатът от командата е със статус ОК и студентът автоматично получава 6.

Примерен резултат от codeception, 8 assertions, всички от тях са успешни.

```
Codeception PHP Testing Framework v2.3.8
Powered by PHPUnit 6.5.5 by Sebastian Bergmann and contributors.

Acceptance Tests (1) -----
✓ HomeworkCest: Try to test (0.36s)
-----

Functional Tests (0) -----
-----

Unit Tests (0) -----
-----

Time: 484 ms, Memory: 12.00MB

OK (1 test, 8 assertions)
```

Следващият скрийншот показва случай с един неуспешен assertion.

```
Codeception PHP Testing Framework v2.3.8
Powered by PHPUnit 6.5.5 by Sebastian Bergmann and contributors.

Acceptance Tests (1) -----
x HomeworkCest: Try to test
✓ HomeworkCest: Try to test [F] (0.39s)
-----

Functional Tests (0) -----
-----

Unit Tests (0) -----
-----

Time: 522 ms, Memory: 12.00MB

There was 1 failure:

-----
1) HomeworkCest: Try to test
Test tests/acceptance/HomeworkCest.php:tryToTest
Step Can see number of elements "li",5
Fail Number of elements counted differs from expected number
Failed asserting that 3 matches expected 5.

Scenario Steps:

10. $I->click("a") at tests/acceptance/HomeworkCest.php:19
9. $I->canSee("Facebook") at tests/acceptance/HomeworkCest.php:18
8. $I->canSeeNumberOfElements("li",5) at tests/acceptance/HomeworkCest.php:17
7. $I->canSeeElement("ol") at tests/acceptance/HomeworkCest.php:16
6. $I->canSeeElement("p") at tests/acceptance/HomeworkCest.php:15
5. $I->canSeeElement("h1") at tests/acceptance/HomeworkCest.php:14

FAILURES!
Tests: 1, Assertions: 8, Failures: 1.
```

За тестовете, използващи webdriver е необходим инсталиран Selenium server и running chromedriver, който проверяваме със следната команда :

curl -I <http://127.0.0.1:9515/wd/hub>

Тя връща статус код, чрез който разбираме дали е running, ако не е изпълнявам 'chromedriver --url-base=/wd/hub 1>/dev/null &', с която се пуска. След което codeception пуска тестовете, отваряйки chrome headless.

Забележка: Проектът е създаден и тестван на Linux и Chrome и Mozilla Firefox. За да работи на Windows конфигурацията би била различна.

9. Приноси на студента, ограничения и възможности за бъдещо разширение

Може да се добави класация на домашните работи по оценка (както класация за конкретното домашно, така и обща въз основа на всички предадени до момента домашни). При регистрация може да се активира акаунт чрез изпращане на мейл. Би било добре да могат да се добавят чрез user interface-а на приложението още админи.

10. Какво научих

При направата на този проект се научих да деплоявам с Amazon Web Services – нещо, с което не се бях сблъсквала досега. Научих как се прави скрипт, който автоматично да инсталира всички необходими програми и да прави автоматично конфигурацията. Досега не бях писала автоматизирани тестове за php код, научих се да използвам Codeception и PHPUnit (PHPUnit отпадна от финалната версия на проекта, но първоначално юнит тестовете щяха да използват него).

11. Използвани източници

<https://codeception.com/docs/03-AcceptanceTests>

<https://codeception.com/docs/modules/WebDriver.html#Local-Testing>

Предал:

/81097, Александрина Каракехайова, КН, група 1/

Приел:

/доц. Милен Петров/