Lab exercise 3 (TDD)

- Write a test suite for testing User class:
  - Test setup: initialize $user as instance of User, with some initial name and email.
  - $user->name() should retrieve name.
  - $user->name('Samy') should change user's name to Samy.
  - $user->email() should retrieve email.
  - $user->email('samy@gmail.com') should change user's email to samy@gmail.com.
  - Test teardown: $user->delete();
- Run the suite and write the code to make it all pass.

```
<phpunit
bootstrap="bootstrap.php">
 <testsuites>
  <testsuite name="Sample">
   <directory>tests</directory>
  </testsuite>
 </testsuites>
</phpunit>
```

phpunit.xml

```
<!--
 * Unit & Automated Testing - Lab 2 Exercise 3
 * Author: Mahmoud Mohamed Kamal
 * Date:   28-03-1998
 * File:   phpunit.xml
 -->

<phpunit bootstrap="bootstrap.php">
 <testsuites>
  <testsuite name="Sample">
   <directory>tests</directory>
  </testsuite>
 </testsuites>
</phpunit>
```

```
<?php
// my app includes
spl_autoload_register(function($class)
{
        @include_once(__DIR__ .
"/classes/{$class}.php");
});
?>
```

bootstrap.php

```
/*
 * Unit & Automated Testing - Lab 2 Exercise 3
 * Author: Mahmoud Mohamed Kamal
 * Date:   28-03-1998
 * File:   bootstrap.php
 */

<?php
// my app includes
spl_autoload_register(function($class) {
        @include_once(__DIR__ . "/classes/{$class}.php");
});
?>
```

```php
<?php
class SampleTest extends PHPUnit\Framework\TestCase {
		private static $user;

	public static function setUpBeforeClass() : void{
		self::$user = new User('Mahmoud','mahmoud@email.com');
	}
		public function test_name(){
			$this->assertTrue(self::$user-
>getName()==='Mahmoud');
		}
		public function test_email(){
		$this->assertTrue(self::$user->getEmail()==='mahmoud@email.com');
	}
		public function test_name_Samy(){
		self::$user->setName('samy');
		$this->assertTrue(self::$user->getName()==='samy');
	}
	public function test_email_Samy(){
		self::$user->setEmail('samy@gmail.com');
		$this->assertTrue(self::$user->getEmail()==='samy@gmail.com');
	}
	public static function tearDownAfterClass() : void{
		self::$user->delete();
	}
}
?>
```

SampleTest.php

```php
/*
 * Unit & Automated Testing - Lab 2 Exercise 3
 * Author: Mahmoud Mohamed Kamal
 * Date:   28-03-1998
 * File:   SampleTest.php
 */

<?php
class SampleTest extends PHPUnit\Framework\TestCase {
        private static $user;

        public static function setUpBeforeClass() : void{
                self::$user = new User('Mahmoud','mahmoud@email.com');
        }
        public function test_name(){
                $this->assertTrue(self::$user->getName()==='Mahmoud');
        }
        public function test_email(){
                $this->assertTrue(self::$user->getEmail()==='mahmoud@email
        }
        public function test_name_Samy(){
                self::$user->setName('samy');
                $this->assertTrue(self::$user->getName()==='samy');
        }
        public function test_email_Samy(){
                self::$user->setEmail('samy@gmail.com');
                $this->assertTrue(self::$user->getEmail()==='samy@gmail.c
        }
        public static function tearDownAfterClass() : void{
                self::$user->delete();
        }
}
?>
```

$ gedit Desktop/lab2/class/User.php &

```php
<?php
class User {
		private $username, $email;

		public function __construct($username, $email) {
			$this->username = $username;
			$this->email    = $email;
		}

		public function getName(){
			return $this->username;
		}

		public function getEmail(){
			return $this->email;
		}

		public function setName($n){
			$this->username = $n;
		}

		public function setEmail($e){
			$this->email = $e;
		}

		public function delete(){
			$this->username = NULL;
			$this->email    = NULL;
		}
}
?>
```

User.php

```php
/*
 * Unit & Automated Testing - Lab 2 Exercise 3
 * Author: Mahmoud Mohamed Kamal
 * Date:   28-03-1998
 * File:   User.php
 */

<?php
class User {
        private $username, $email;

        public function __construct($username, $email) {
                $this->username = $username;
                $this->email    = $email;
        }

        public function getName(){
                return $this->username;
        }

        public function getEmail(){
                return $this->email;
        }

        public function setName($n){
                $this->username = $n;
        }

        public function setEmail($e){
                $this->email = $e;
        }

        public function delete(){
                $this->username = NULL;
                $this->email    = NULL;
        }
}
?>
```

$ ~/phpunit --testdox

```
/*
 * Unit & Automated Testing - Lab 2 Exercise 3
 * Author: Mahmoud Mohamed Kamal
 * Date:   28-03-1998
 * File:   User.php
 */

Sample
 ✔ Name
 ✔ Email
 ✔ Name Samy
 ✔ Email Samy

Time: 00:00.016, Memory: 18.00 MB

OK (4 tests, 4 assertions)
```

Lab exercise 4

- Write a test suite to test a class from your most recent PHP assignment ☺

```
<phpunit bootstrap="bootstrap.php">
 <testsuites>
  <testsuite name="Sample">
   <directory>tests</directory>
  </testsuite>
 </testsuites>
</phpunit>
```

phpunit.xml

```
<!--
 * Unit & Automated Testing - Lab 2 Exercise 4
 * Author: Mahmoud Mohamed Kamal
 * Date:   28-03-1998
 * File:   phpunit.xml
 -->

<phpunit bootstrap="bootstrap.php">
 <testsuites>
  <testsuite name="Sample">
   <directory>tests</directory>
  </testsuite>
 </testsuites>
</phpunit>
```

```
<?php
// my app includes
spl_autoload_register(function($class) {
        @include_once(__DIR__ .
"/classes/{$class}.php");
});
?>
```

bootstrap.php

```
/*
 * Unit & Automated Testing - Lab 2 Exercise 4
 * Author: Mahmoud Mohamed Kamal
 * Date:   28-03-1998
 * File:   bootstrap.php
 */

<?php
// my app includes
spl_autoload_register(function($class) {
        @include_once(__DIR__ . "/classes/{$class}.php");
});
?>
```

```
<?php
class SampleTest extends PHPUnit\Framework\TestCase {
        private static $user;

    public static function setUpBeforeClass() : void{
        self::$user = new User('Mahmoud', 1234567890,
'mahmoud@email.com');
    }
        public function test_name(){
                $this->assertTrue(self::$user-
>getUsername()==='Mahmoud');
        }
        public function test_password(){
        $this->assertTrue(self::$user->getPassword()===1234567890);
    }
        public function test_email(){
        $this->assertTrue(self::$user->getEmail()==='mahmoud@email.com');
    }
        public function test_name_Kamal(){
        self::$user->setUsername('Kamal');
        $this->assertTrue(self::$user->getUsername()==='Kamal');
    }
        public function test_password_Kamal(){
        self::$user->setPassword(1234);
        $this->assertTrue(self::$user->getPassword()===1234);
    }
    public function test_email_Kamal(){
        self::$user->setEmail('kamal@email.com');
        $this->assertTrue(self::$user->getEmail()==='kamal@email.com');
    }
    public static function tearDownAfterClass() : void{
        self::$user->delete();
    }
}
?>
```

SampleTest.php

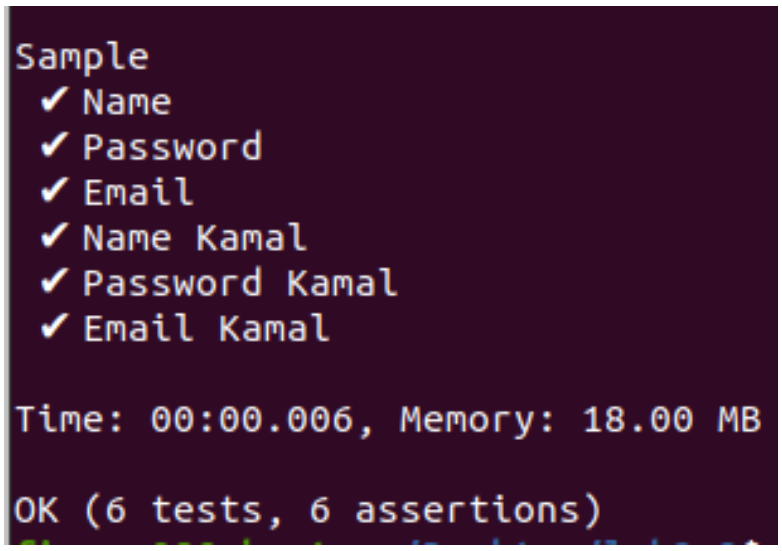
```
/*
 * Unit & Automated Testing - Lab 2 Exercise 4
 * Author: Mahmoud Mohamed Kamal
 * Date:   28-03-1998
 * File:   SampleTest.php
 */

<?php
class SampleTest extends PHPUnit\Framework\TestCase {
        private static $user;

        public static function setUpBeforeClass() : void{
                self::$user = new User('Mahmoud', 1234567890, 'mahmoud@email.com');
        }
        public function test_name(){
                $this->assertTrue(self::$user->getUsername()==='Mahmoud');
        }
        public function test_password(){
                $this->assertTrue(self::$user->getPassword()===1234567890);
        }
        public function test_email(){
                $this->assertTrue(self::$user->getEmail()==='mahmoud@email.com');
        }
        public function test_name_Kamal(){
                self::$user->setUsername('Kamal');
                $this->assertTrue(self::$user->getUsername()==='Kamal');
        }
        public function test_password_Kamal(){
                self::$user->setPassword(1234);
                $this->assertTrue(self::$user->getPassword()===1234);
        }
        public function test_email_Kamal(){
                self::$user->setEmail('kamal@email.com');
                $this->assertTrue(self::$user->getEmail()==='kamal@email.com');
        }
        public static function tearDownAfterClass() : void{
                self::$user->delete();
        }
}
?>
```

```php
<?php
class User
{
   private $username, $password, $email;

   public function __construct($username, $password, $email)
   {
      $this->setEmail($email);
      $this->setUsername($username);
      $this->setPassword($password);
   }

   function getUsername()
   {
      return $this->username;
   }

   function getPassword()
   {
      return $this->password;
   }

   function getEmail()
   {
      return $this->email;
   }

   function setUsername($username)
   {
      $this->username = $username;
   }

   function setPassword($password)
   {
      $this->password = $password;
   }

   function setEmail($email)
   {
      $this->email = $email;
   }

   function delete()
   {
      $this->setEmail(NULL);
      $this->setUsername(NULL);
      $this->setPassword(NULL);
   }
}
```

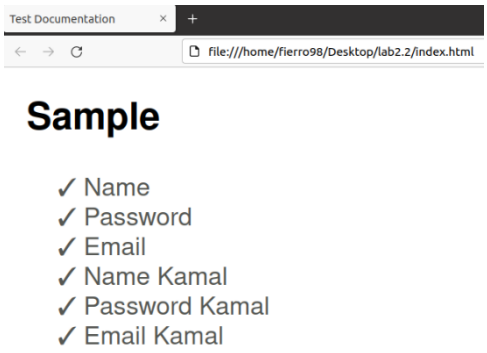
User.php

```php
<!--
    Lab 3
    Author: Mahmoud Mohamed Kamal
    Date:   26-02-2022
    File:   User.php
-->

<?php
class User
{
    private $username, $password, $email;

    public function __construct($username, $password, $email)
    {
        $this->setEmail($email);
        $this->setUsername($username);
        $this->setPassword($password);
    }

    function getUsername()
    {
        return $this->username;
    }

    function getPassword()
    {
        return $this->password;
    }

    function getEmail()
    {
        return $this->email;
    }

    function setUsername($username)
    {
        $this->username = $username;
    }

    function setPassword($password)
    {
        $this->password = $password;
```

$ ~/phpunit --testdox



$ ~/phpunit --testdox-html index.html



$ ~/phpunit --testdox-text index.txt

index.html

```
Sample
 [x] Name
 [x] Password
 [x] Email
 [x] Name Kamal
 [x] Password Kamal
 [x] Email Kamal
```

$ ~/phpunit --testdox-xml index.xml

index.xml | index.html | index.txt

```
<?xml version="1.0" encoding="UTF-8"?>
<tests>
  <test className="SampleTest" methodName="test_name" prettifiedClassName="Sample" prettifiedMethodName="Name" status="0" time="0.000415496" size="-1" groups="default">
    <group name="default"/>
  </test>
  <test className="SampleTest" methodName="test_password" prettifiedClassName="Sample" prettifiedMethodName="Password" status="0" time="0.000180646" size="-1" groups="default">
    <group name="default"/>
  </test>
  <test className="SampleTest" methodName="test_email" prettifiedClassName="Sample" prettifiedMethodName="Email" status="0" time="0.000661703" size="-1" groups="default">
    <group name="default"/>
  </test>
  <test className="SampleTest" methodName="test_name_Kamal" prettifiedClassName="Sample" prettifiedMethodName="Name Kamal" status="0" time="0.000144162" size="-1" groups="default">
    <group name="default"/>
  </test>
  <test className="SampleTest" methodName="test_password_Kamal" prettifiedClassName="Sample" prettifiedMethodName="Password Kamal" status="0" time="0.000204525" size="-1" groups="default">
    <group name="default"/>
  </test>
  <test className="SampleTest" methodName="test_email_Kamal" prettifiedClassName="Sample" prettifiedMethodName="Email Kamal" status="0" time="0.000261289" size="-1" groups="default">
    <group name="default"/>
  </test>
</tests>
```