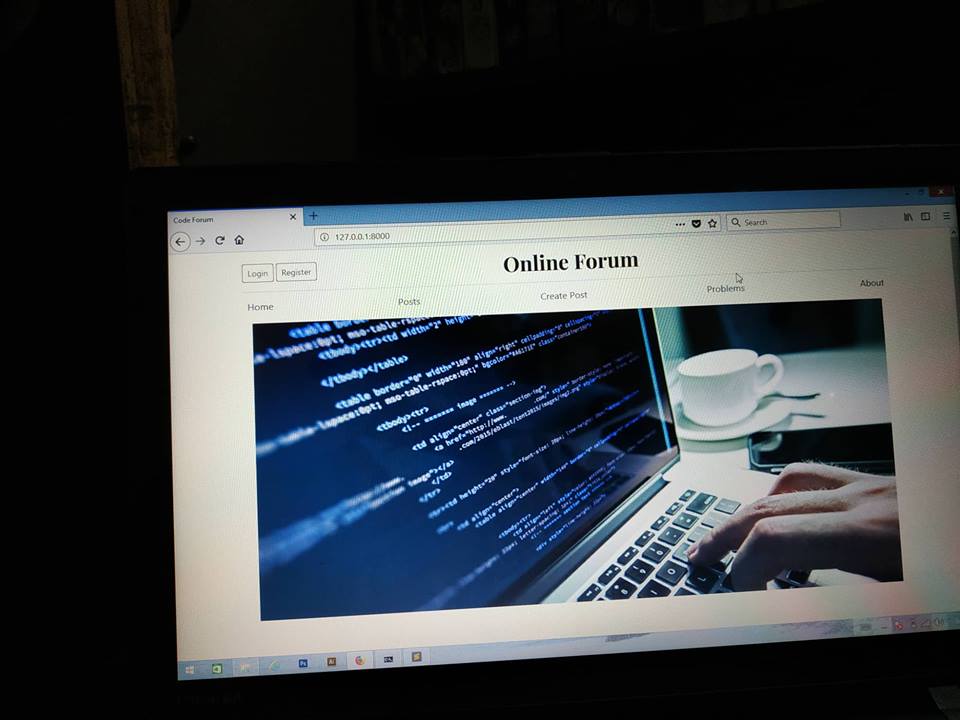
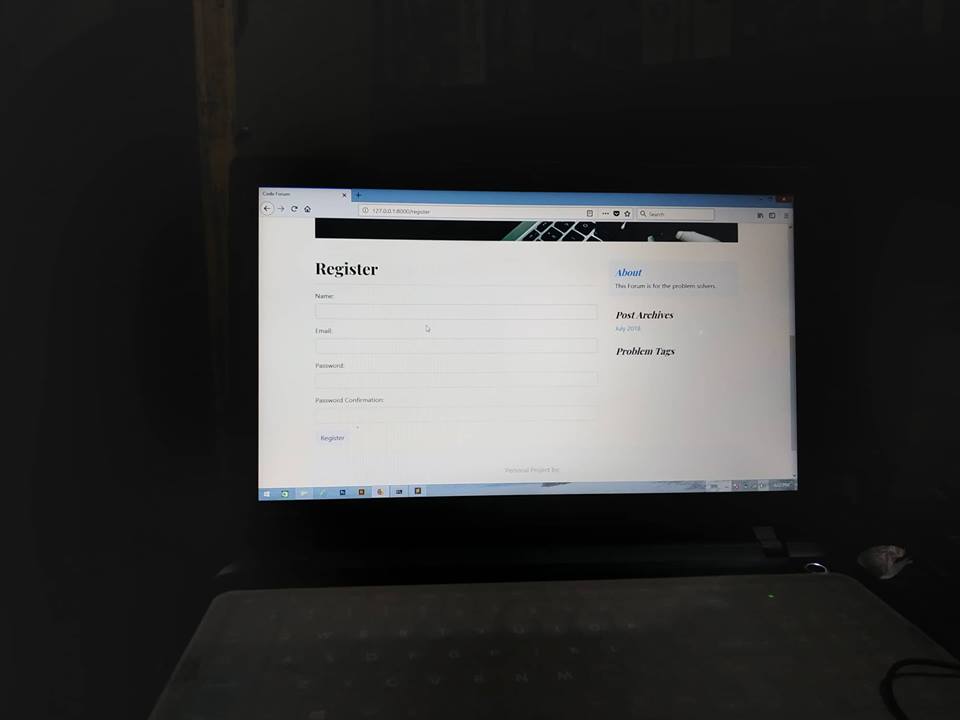
**Project Description:**

The project is about Online Forum. This Forum is for the problem solvers. Anyone who has a knack for programming can join this forum and share their knowledge and views about competitive programming and problem solving. They can post Tutorials or helpful thoughts and also comment on them. Hopefully by sharing their thoughts they will help build the community and make it stronger. We developed the project using Laravel Framework. Version "laravel/framework": "5.6.\*", "php": "^7.1.3", “phpunit/phpunit": "^7.0"



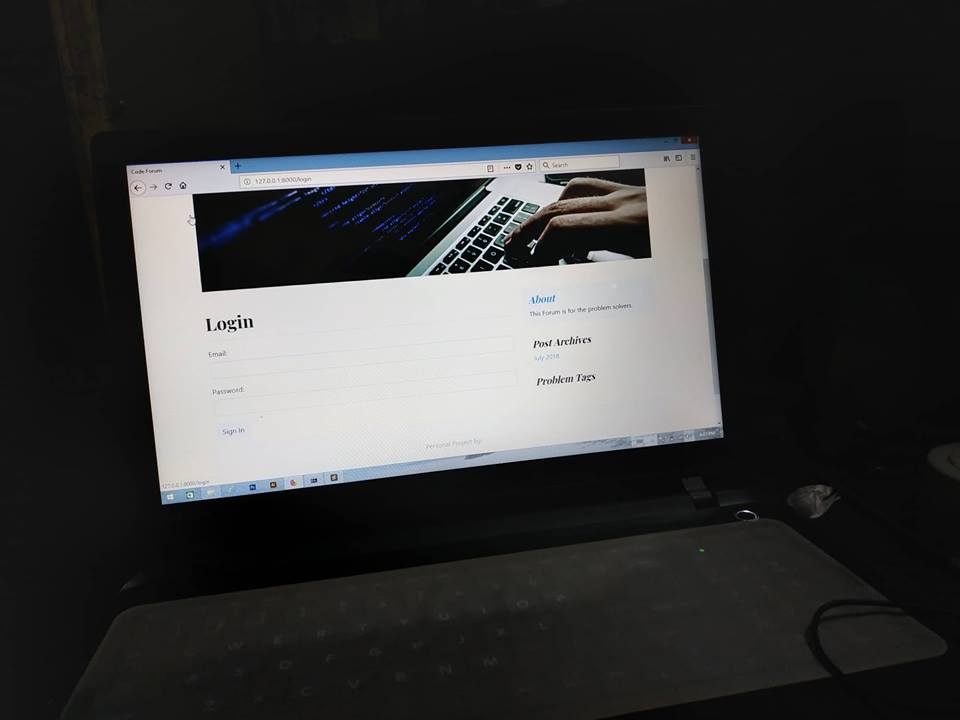
**Fig1: Home page of the project.**

Visitors can able to see the index page of the forum. Navigation Menu contains Home, Post, Create Post, Problems and about section. By clicking on options from menu, it will be redirected to that page.



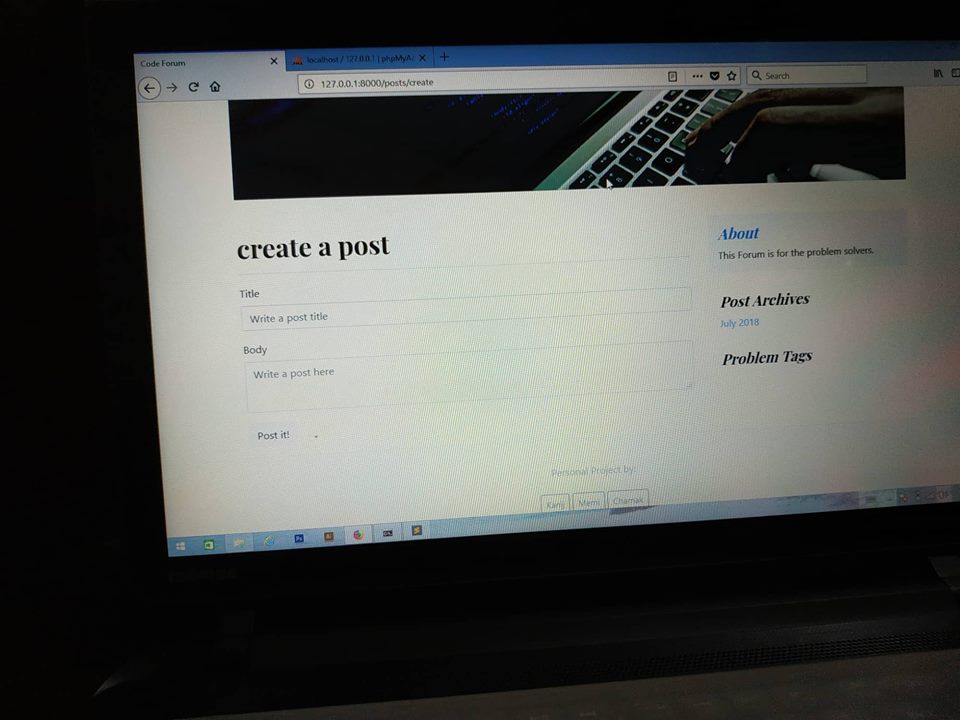
**Fig2: Navigation to registration page of project.**

To be a Member of the forum user have to be registered. Registration requires name, email, password and password confirmation. All fields are required. Otherwise an alert will be shown.

****

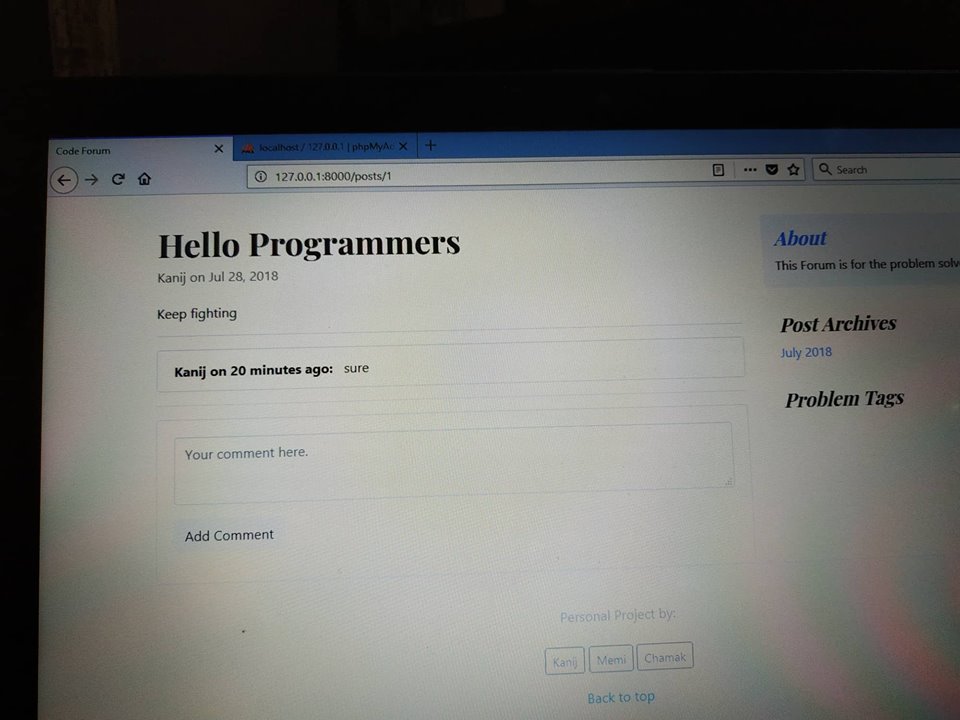
**Fig3: Navigation to Login page of project.**

For Login to the forum, users have to fulfill Email and Password field with valid information. For Invalid or Blank inputs an alert will be shown for entering valid inputs.



**Fig4: Navigation to Create Post page.**

Guest Users can view previous posts but for creating a post user have to be logged in. other Users can also comment on that post.



**Fig5: Navigation to Add Comment on Post.**

**Laravel-Unit-tests**

Laravel is built with testing in mind. In fact, support for testing with PHPUnit is included out of the box and a phpunit.xml file is already set up for the application. The framework also ships with convenient helper methods that allow you to expressively test your applications.

By default, application's  tests directory contains two directories:  Feature and  Unit.[1]

**Environmental Setup:**

When running tests via phpunit, Laravel will automatically set the configuration environment to testing because of the environment variables defined in the phpunit.xml file. Laravel also automatically configures the session and cache to the array driver while testing, meaning no session or cache data will be persisted while testing.

[**Creating & Running Tests**](https://laravel.com/docs/5.6/testing#creating-and-running-tests)**:**

To create a new test case, use the make:test Artisan command:

// Create a test in the Feature directory...

php artisan make:test UserTest

// Create a test in the Unit directory...

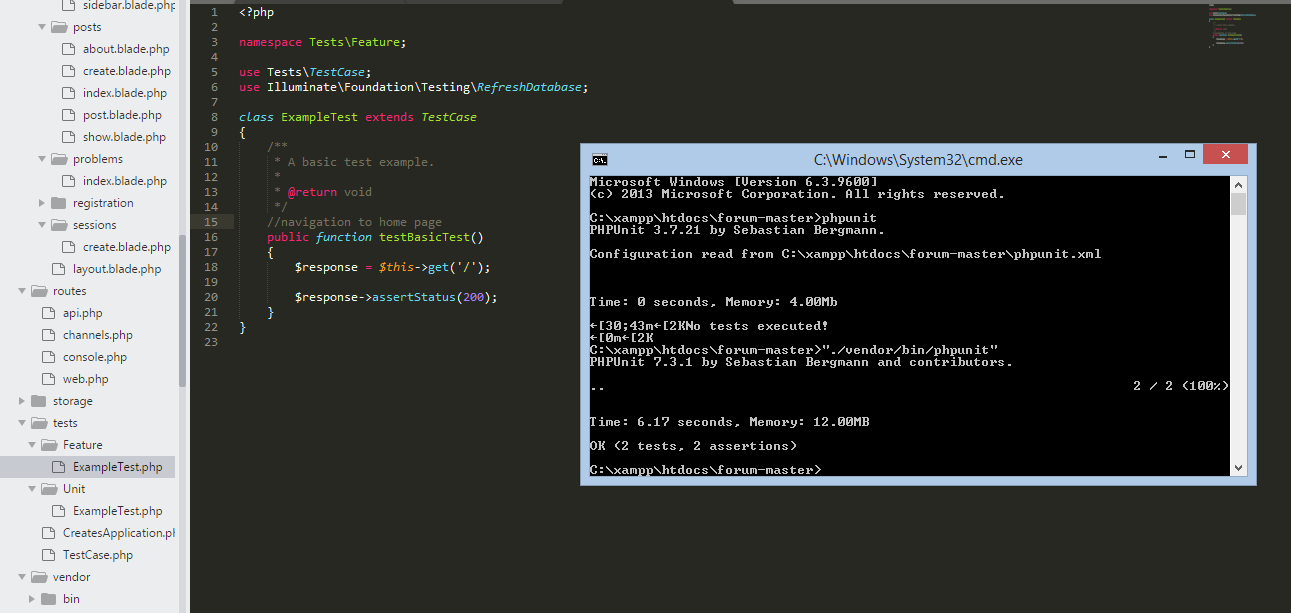
php artisan make:test UserTest --unit

After generating the tests, test methods would be normally used PHPUnit. To run tests, execute the phpunit command from terminal:

For our project we run the tests from root folder terminal and use Sublime Text as code Editor.

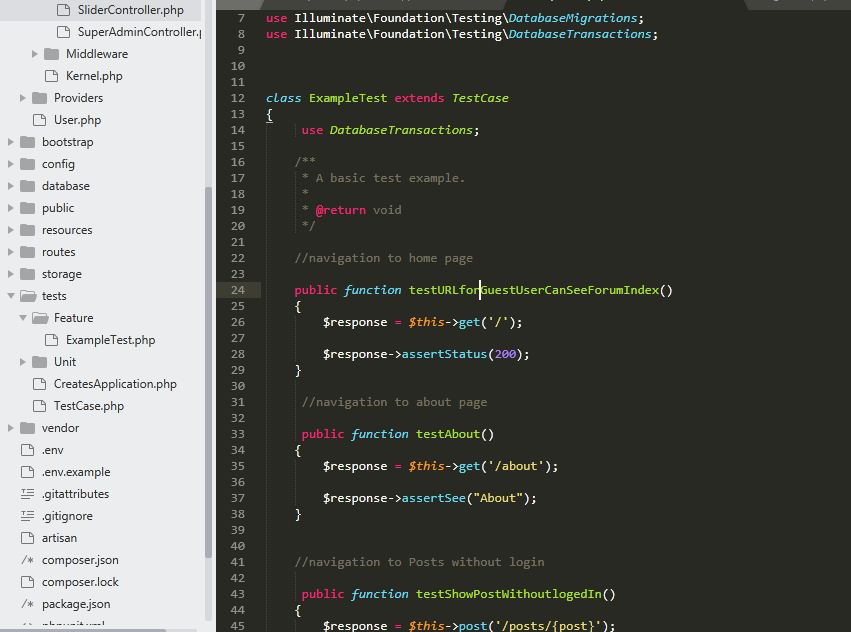
// Running the tests from root folder directory...

“./vendor/bin/phpunit”

****

**Fig6: Executing Laravel’s Default tests.**

We write all URL test for navigation in ExampleTest.php from Feature directory which exist in application’s tests directory.



**Fig7: Executing URL test for navigation.**

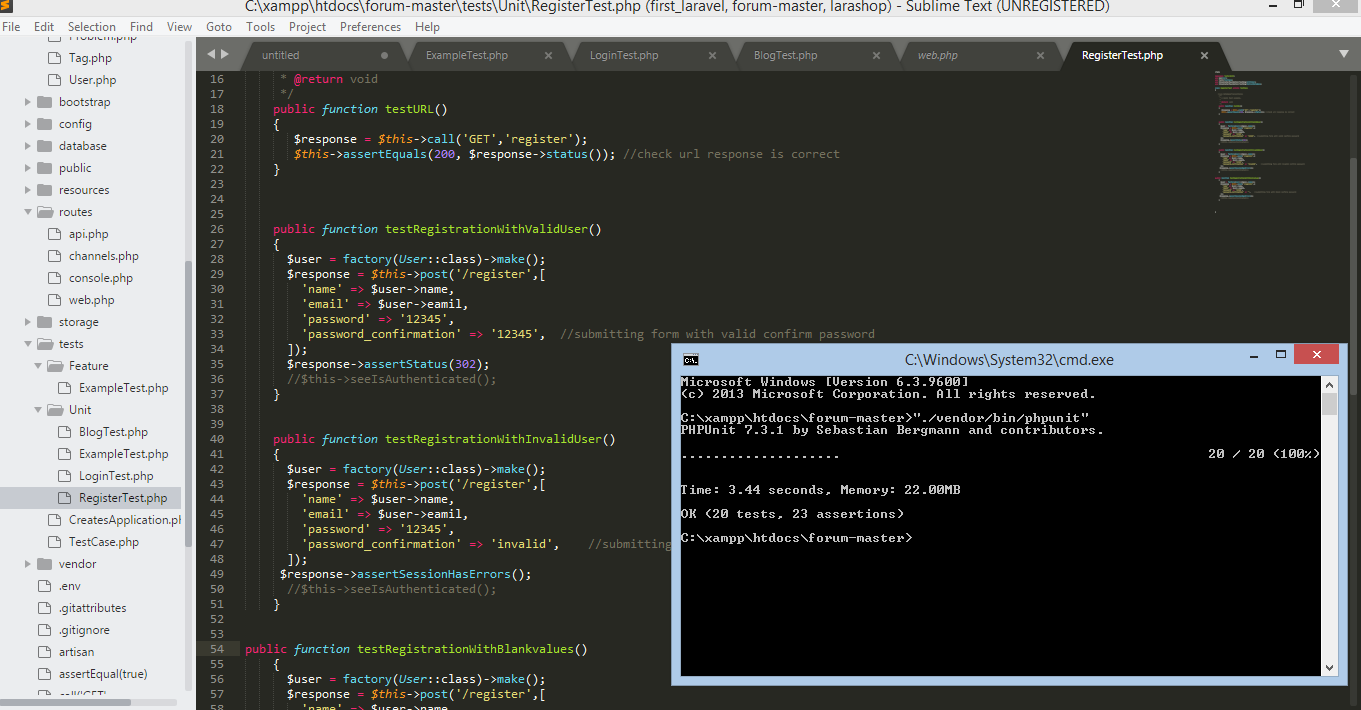
Using artisan Command we create three new unit test file for executing other tests. We create

RegisterTest.php file for registration’s Unit Test

LoginTest.php file for Login’s Unit Test

BlogTest.php file for Post’s and Comment’s Unit Test

**After running all tests related to our project, expected outcomes:**

****

**Fig7: Executing all tests related to the project.**

**References**

**[1]** [**https://laravel.com/docs/5.6/testing**](https://laravel.com/docs/5.6/testing)

**[2]** [**https://github.com/dwightwatson/laravel-auth-tests**](https://github.com/dwightwatson/laravel-auth-tests)

**[3]** [**https://www.youtube.com/watch?v=HqDAGW14gXA&t=447s**](https://www.youtube.com/watch?v=HqDAGW14gXA&t=447s)