**CONTENUE**

1. HTML5: Image, Hyperlinks, List, Tables 03

2. HTML5: Forms 05

3. CSS: Padding, Margin, Float 06

4. JavaScript &jQuery 07

5. Bootstrap 09

6. PHP: Insert, View 10

7. PHP: Delete, Edit 12

8. PHP: MVC 14

9. AJAX 17

10. Laravel: Installation 20

11. Laravel: Routes& Controllers 23

12. Laravel: Model & Migration 25

13. Laravel: View 27

14. Laravel: CRUD Operation 28

15. CMS: WordPress 29

**Lab # 01**

**HTML5**

**Lab 1:**

The objective of this lab task is to learn how to implement a hyper link and how to insert an image on a web page.

* Create New HTML Page
* Add Paragraphs
* Make some content of paragraph as Bold, Italic and Underline
* Insert a hyper link.
* Insert an image.

**Lab 2:**

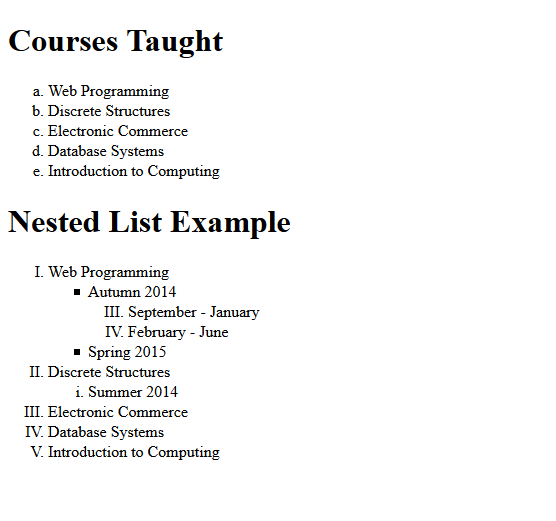
* Apply Heading one, Heading three and Heading 5
* Add 8 Different Meta Tags
* Create new CSS file and link it into HTML file
* Create new Javascript file and link it into HTML file
* Create Basic List with your educational details

**Lab 3:**

The objective of this lab task is to learn how to implement a list and how to insert an table.

* Implement a nested list.
* Implement a complex table.

**Sample Output:**



**Lab # 02**

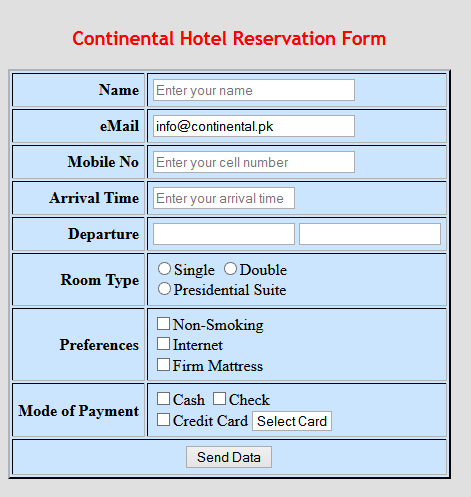
**HTML5**

**Lab :**

The objective of this lab task is to learn how to create a form.

* Create a form.

**Sample Output:**



**Lab # 03**

**Cascade Style Sheet (CSS)**

**Lab :**

The objective of this lab task is to learn how to apply different properties of css to give a web page a professional look. The first task is to organize journal.html by adding ids, classes, spans and divs as seen in lecture. Then, add "boxes" around these sections of the website by adding to style sheet.

* Apply border, margin and padding properties.
* Apply float, font-family and font-weight properties.

**Sample Output:**



**Lab # 04**

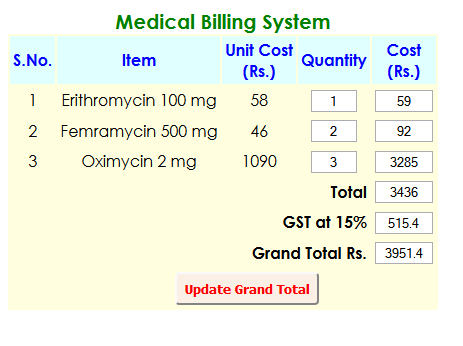
**JavaScript & JQuery**

**Lab :**

Compute Total Cost of Medical Bill.

* JavaScript is used to pre-compute the total cost of an order and display it to the customer before the order is submitted to the server for processing.

**Sample Output:**

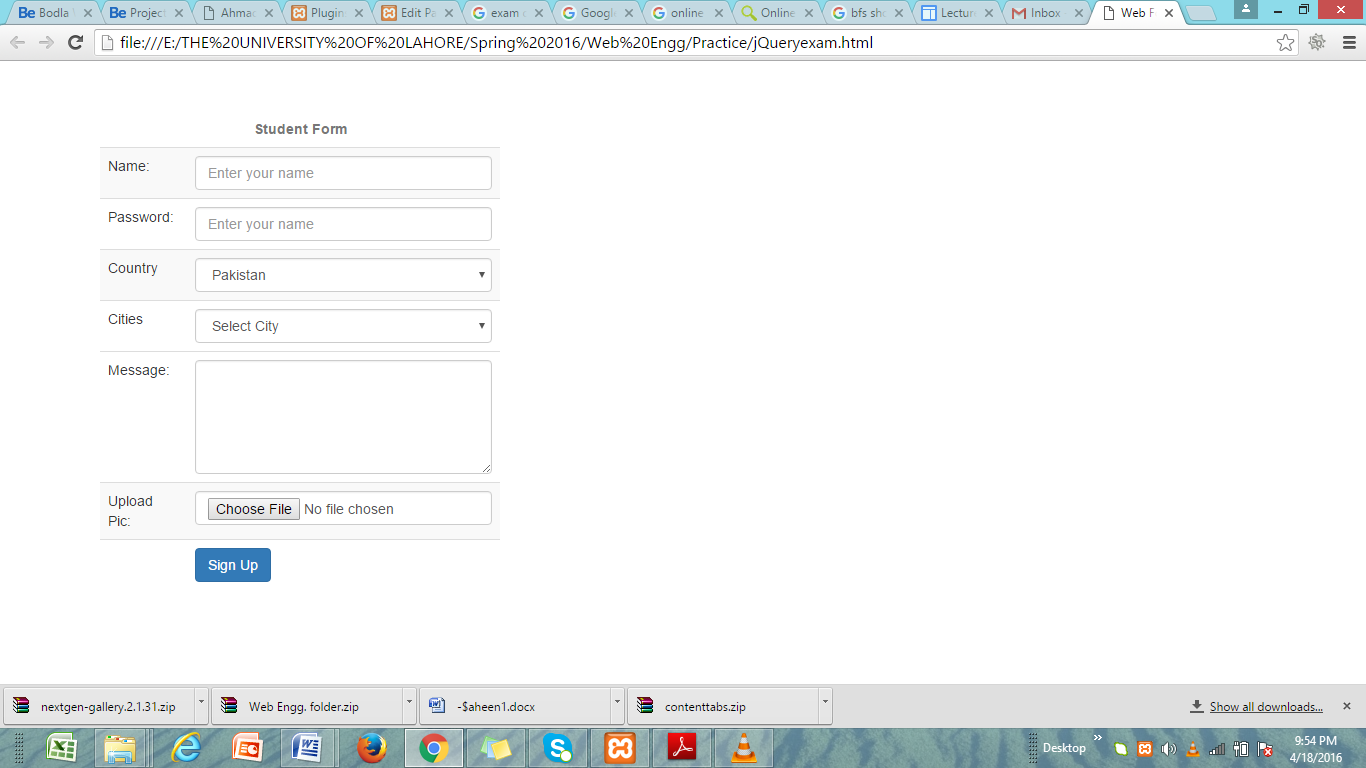
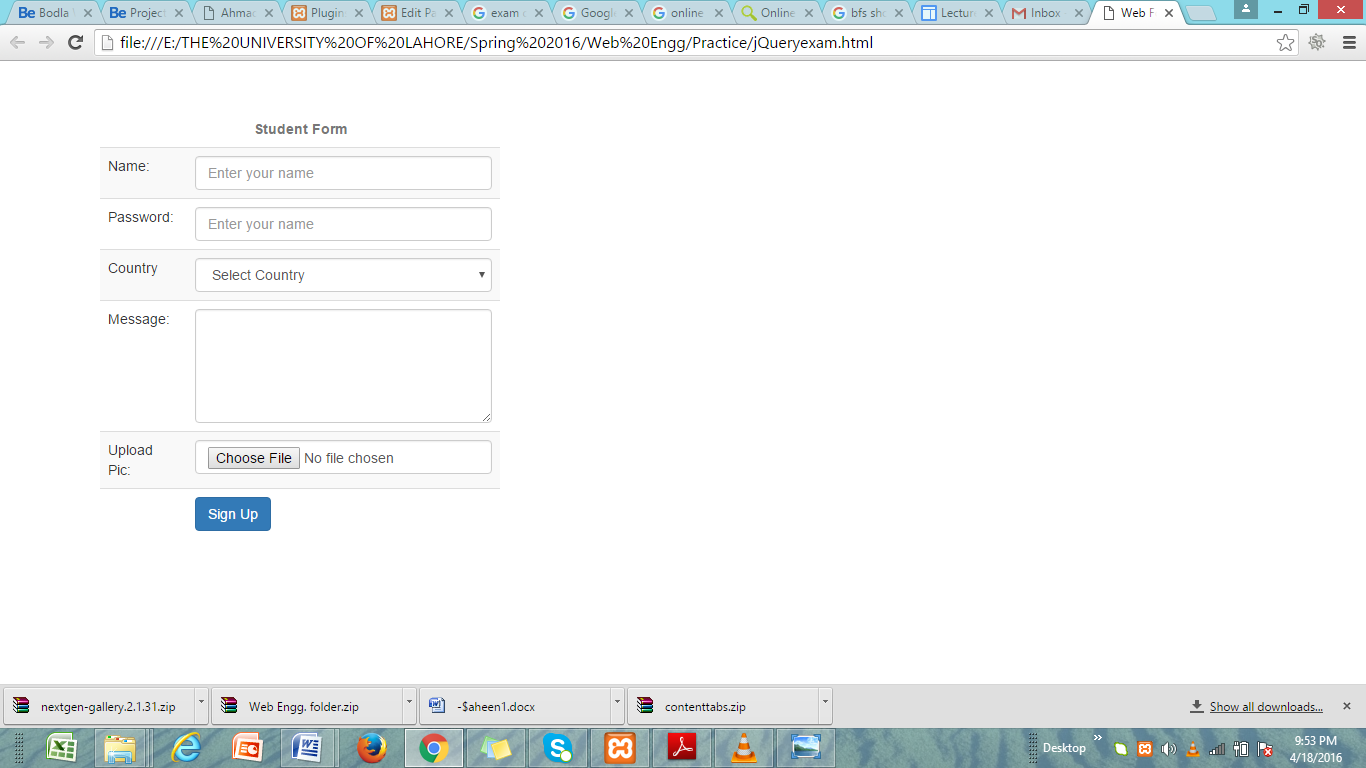


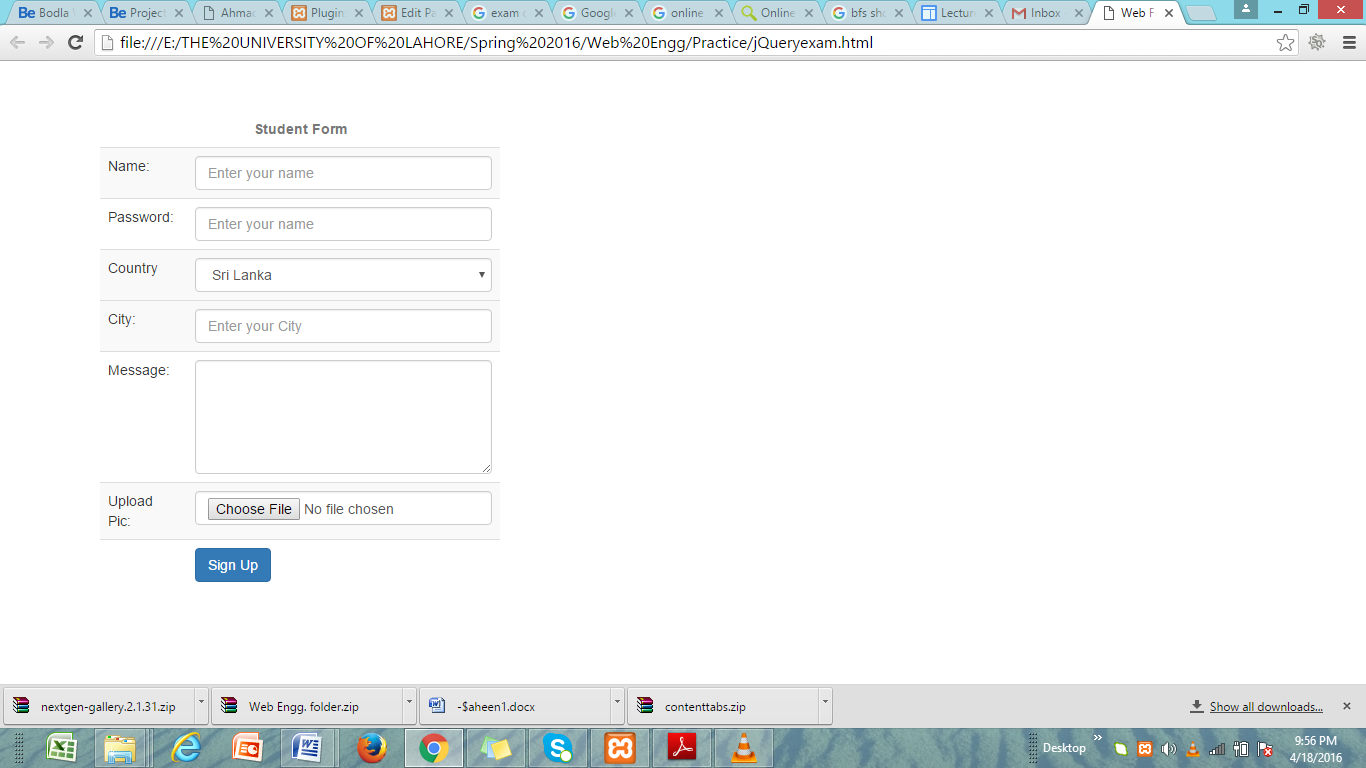
**Lab :**

Apply JQuery on a form. It will perform following functions:

* When page is load, cities will not display.
* When user select country Pakistan, all cities of Pakistan will show only.
* When user select any other country than Pakistan, a field is show to city name manually.

**Sample Output:**





**Lab # 05**

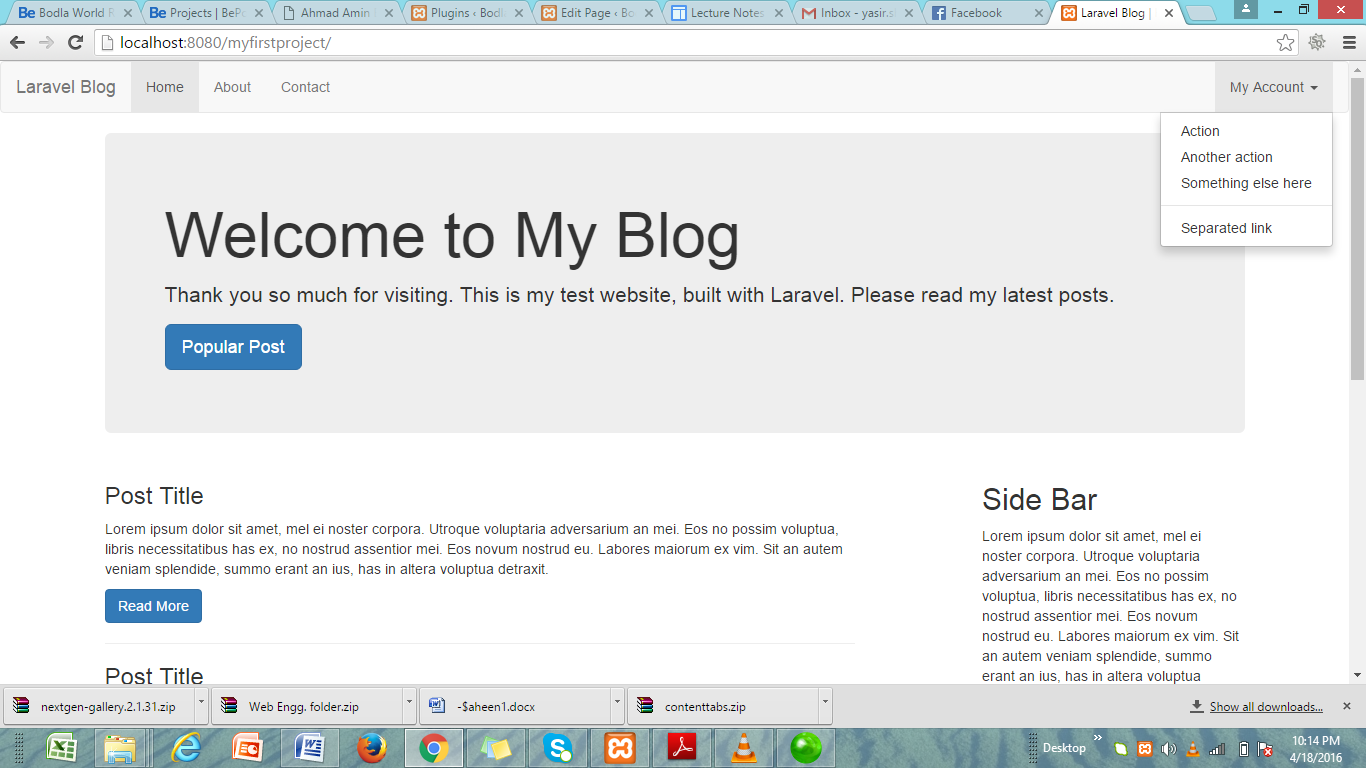
**Bootstrap**

**Lab :**

In this lab, we learn how to implement bootstrap to make pages responsive.

* First set bootstrap environment.
* Apply different classes of bootstrap to meet the output given below.

**Sample Output:**



**Lab # 06**

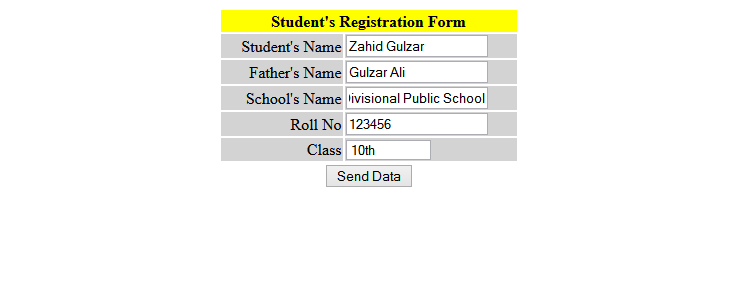
**PHP (Insert& View)**

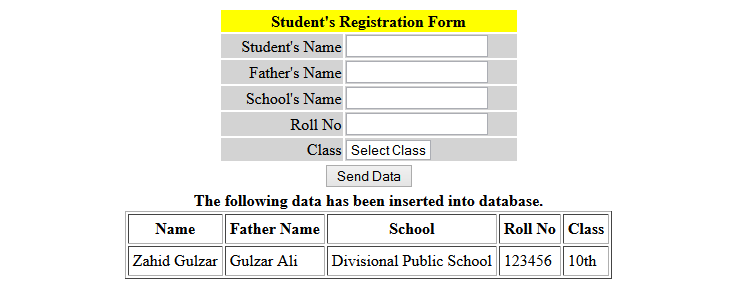
**Lab 1:**

In this lab, we learn how to make a dynamic website.

* Create database connection
* Write PHP code to insert data from a form into MySQL database.

**Sample Output:**



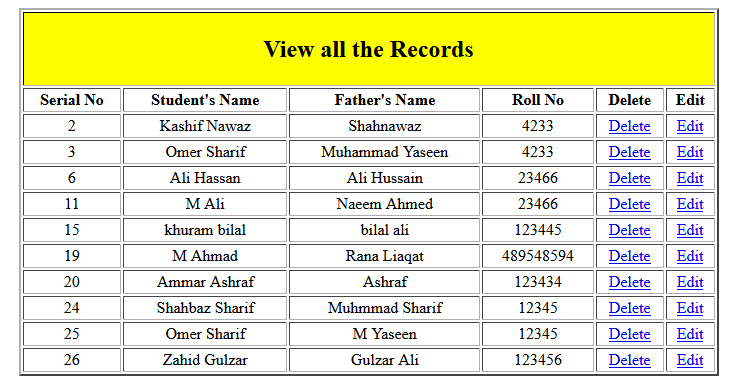


**Lab 2:**

In this lab, we learn how to view data from database on a web page

* In this exercise, retrieve all the records from the database school, show on the web page in table format.
* Add on the table two more columns; one for edit and one for delete the record.

**Sample Output:**



**Lab # 07**

**PHP (Delete & Update)**

**Lab 1:**

In this lab, we learn how to delete a record.

* In this exercise, delete a record from the database using web interface.
* When user clicks on delete it deletes a particular record and display a message that a record has been deleted successfully.

**Sample Output:**

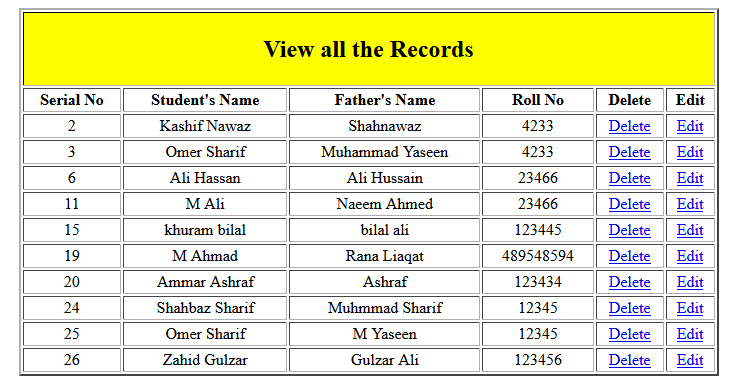


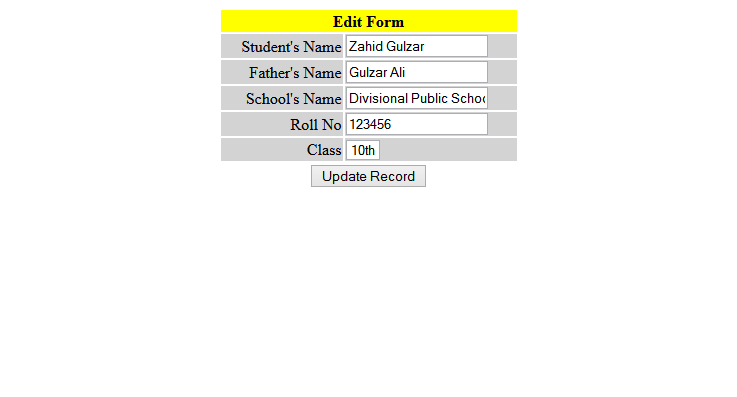
**Lab 2:**

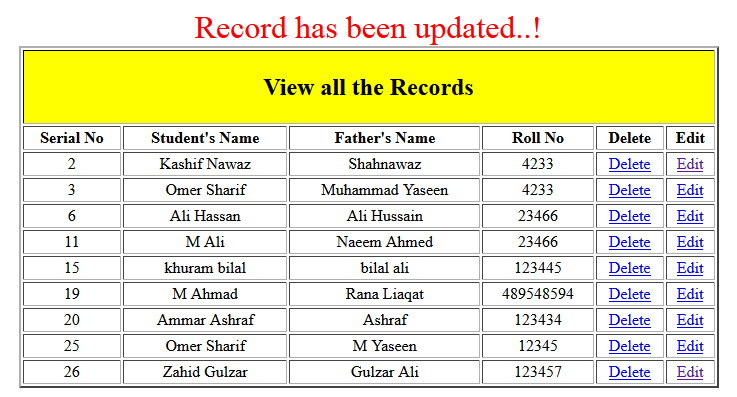
In this lab, we learn how to update/edit a record.

* In this exercise, you have to edit/update a record using a web interface.
* When user clicks on edit, it will able to update his/her record, with a message that record has been updated.

**Sample Output:**







**Lab # 08**

**PHP (MVC)**

**Lab 1:**

In this lab, we learn how to implement MVC design pattern through PHP.

* In this exercise, we show by implementing MVC , how data regarding books can be displayed to user.
* We create model, view and controller.

index.php file

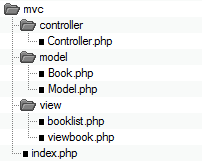
<?php

include\_once("controller/Controller.php");

$controller = new Controller;

$controller🡪invoke();

?>



Model.php file

include\_once("model/Book.php");

class Model {

public function getBookList(){

// here goes some hardcoded values to simulate the database

return array(

"Jungle Book" => new Book("Jungle Book", "R. Kipling", "A classic book."),

"Moonwalker" => new Book("Moonwalker", "J. Walker", ""),

"PHP for Dummies" => new Book("PHP for Dummies", "Some Smart Guy", ""));

}

public function getBook($title){

// we use the previous function to get all the books and then we returnrequested one.

// in a real life scenario this will be done through a db select command

$allBooks = $this->getBookList();

return $allBooks[$title];

}

}

Book.php

class Book {

public $title;

public $author;

public $description;

public function \_\_construct($title, $author, $description)

{

$this->title = $title;

$this->author = $author;

$this->description = $description;

}

}

controller.php file

<?php

include\_once("model/Model.php");

class Controller {

public $model;

public function \_\_construct(){

$this->model = new Model();

}

public function invoke(){

if (!isset($\_GET['book'])){

$books = $this->model->getBookList();

include 'view/booklist.php';

}

else{

$book = $this->model->getBook($\_GET['book']);

include 'view/viewbook.php';

}

}

}

?>

View.php file

<html>

<head></head>

<body>

<?php

echo 'Title:' . $book->title . '<br/>';

echo 'Author:' . $book->author . '<br/>';

echo 'Description:' . $book->description . '<br/>';

?>

</body>

</html>

**Lab # 09**

**AJAX**

**Lab 1:**

In this lab, we learn how to implement AJAX.

* Send a request to PHP page through AJAX.
* Create a folder name ajax in htdocs, and inside it create a page requestpage.php
* Create another page ajaxtest.html
* Create a page name ajax.js
* When ajaxtest.html will load on browser and by clicking on send request ,requestpage.php contents will display on the page without refreshing of the page.

ajaxtest.html file

<!DOCTYPE html>

<html>

<head>

<title>AJAX</title>

</head>

<body>

<a href=”javascript:request(‘requestpage.php’, ‘responseDiv’)”>Send Request</a>

<div id=”responseDiv”></div>

</body>

</html>

requestpage.php file

<?php

echo“Hello World!, AJAX is working on my page”;

?>

ajax.js file

// create a Boolean variable to check for a valid IE instance

varxmlhttp = false;

// check if we are using IE

try {

// If the JavaScript version is greater than 5.

xmlhttp = new ActiveXObject(“Msxml2.XMLHTTP”);

}catch(e){

try {

xmlhttp = new ActiveXObject(“Microsoft.XMLHTTP”);

}catch(E){

xmlhttp = false;

} }

if (!xmlhttp&&typeofXMLHttpRequest != ‘undefined’){

xmlhttp = new XMLHttpRequest();

}

function request(page, objDiv){

varserverPage = page + “?time=” + new Date().getTime();

varobj = document.getElementById(objDiv);

obj.innerHTML = “Loading…”;

xmlhttp.open(“GET”, serverPage);

xmlhttp.onreadystatechange = function(){

if(xmlhttp.readyState == 4 &&xmlhttp.status == 200){

obj.innerHTML = xmlhttp.responseText;

}

}

xmlhttp.send(null);

}

**Sample Output:**



**Lab # 10**

**Laravel(Installation)**

**Lab 1:**

In this lab, we learn how to install Laravel on your computers

* Open [www.laravel.com](http://www.laravel.com) and click on documentation
* Open [www.getcomposer.org](http://www.getcomposer.org) and download and install composer.
* Open terminal CMD and write command

composer global require “laravel/installer”

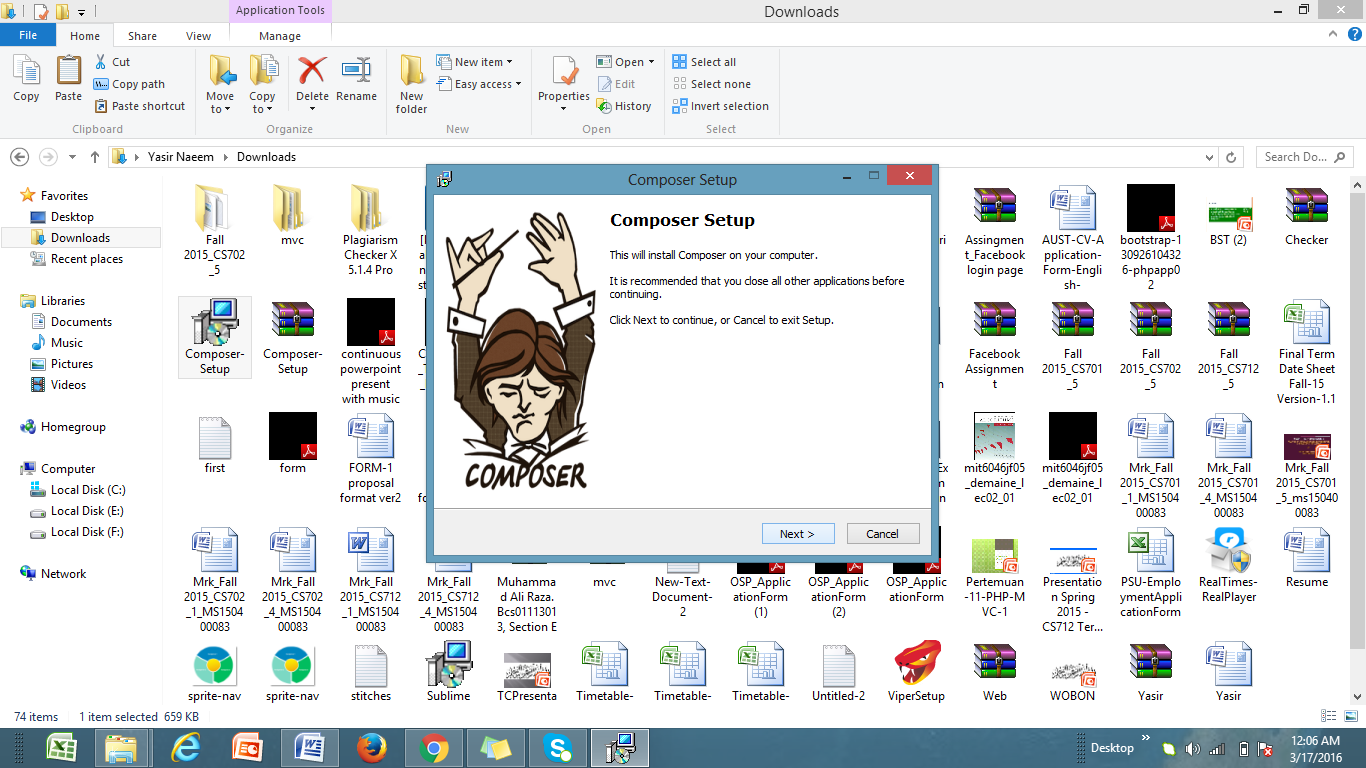
* After that create a project by typing command

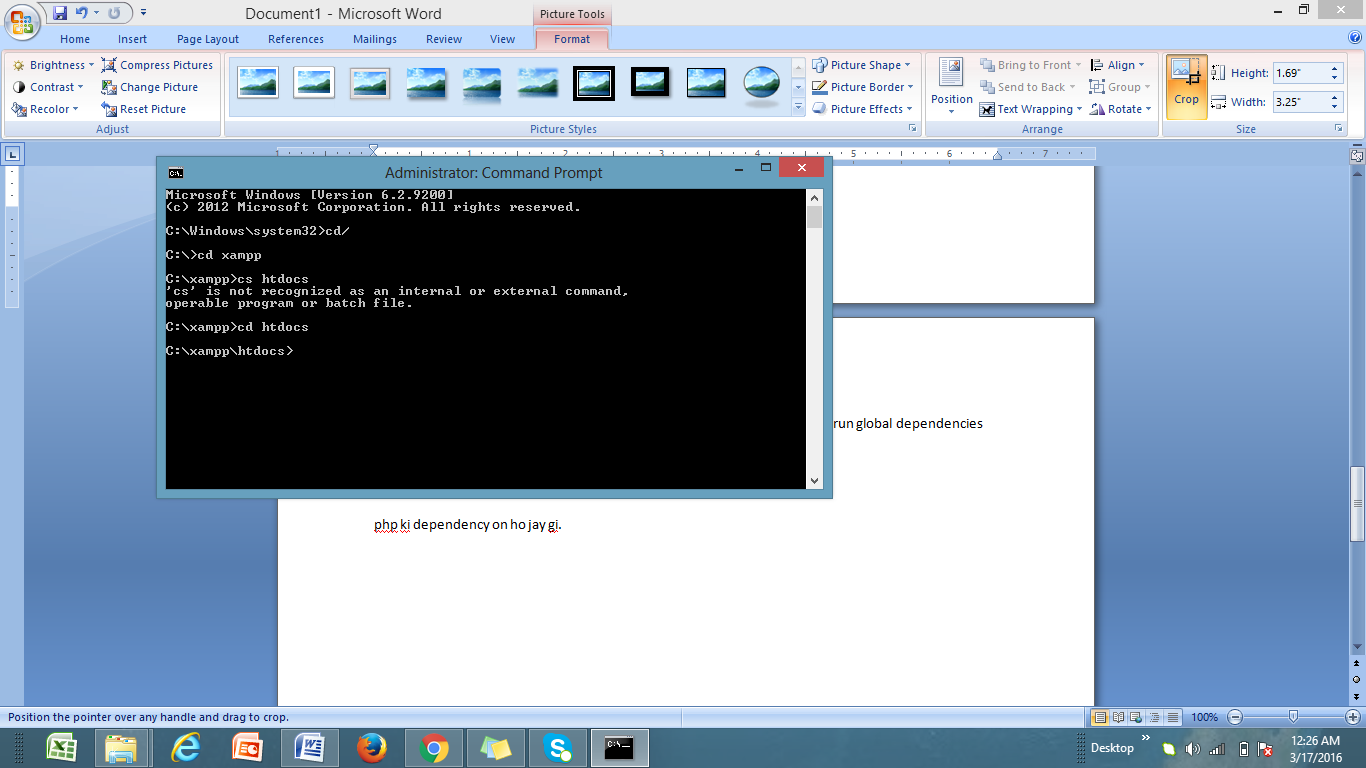
composer create-project –prefer-distlaravel/laravelmyblog

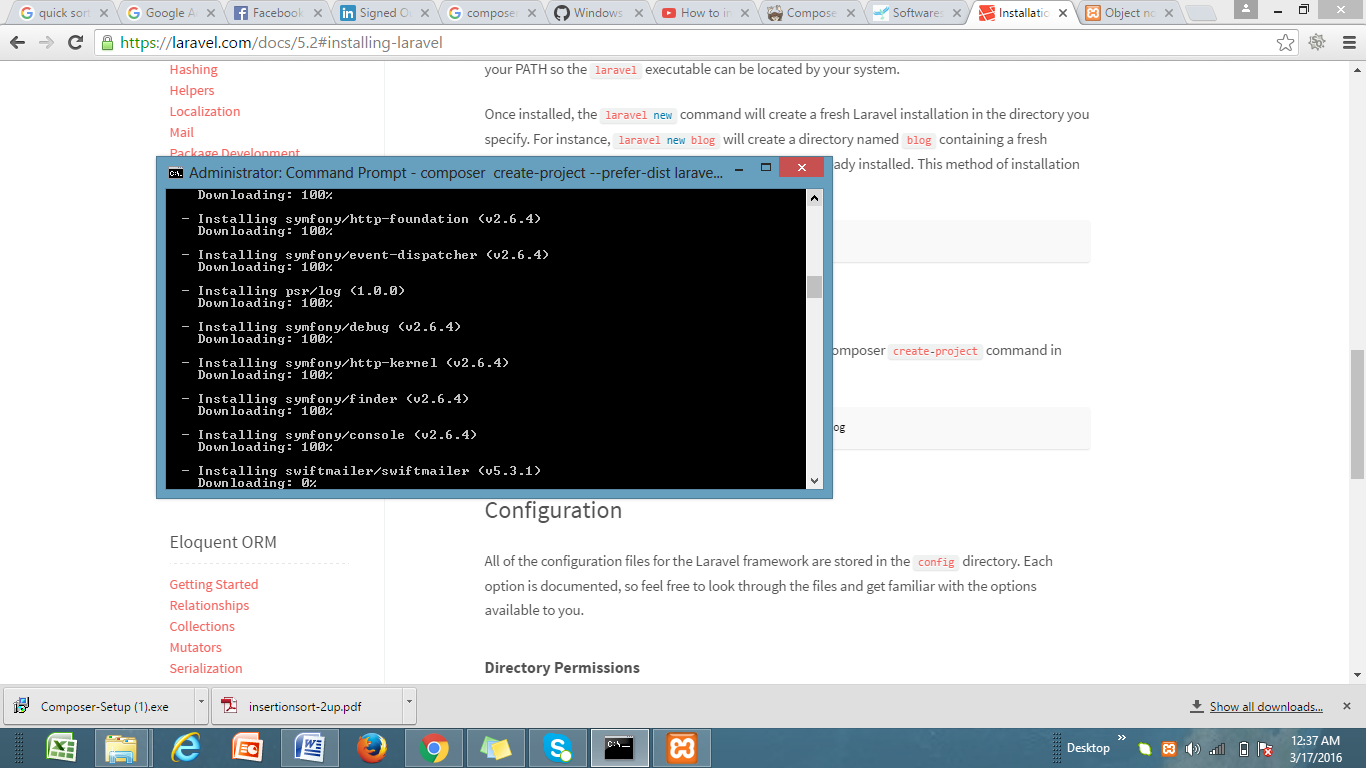
* To check laravel is installed or not type

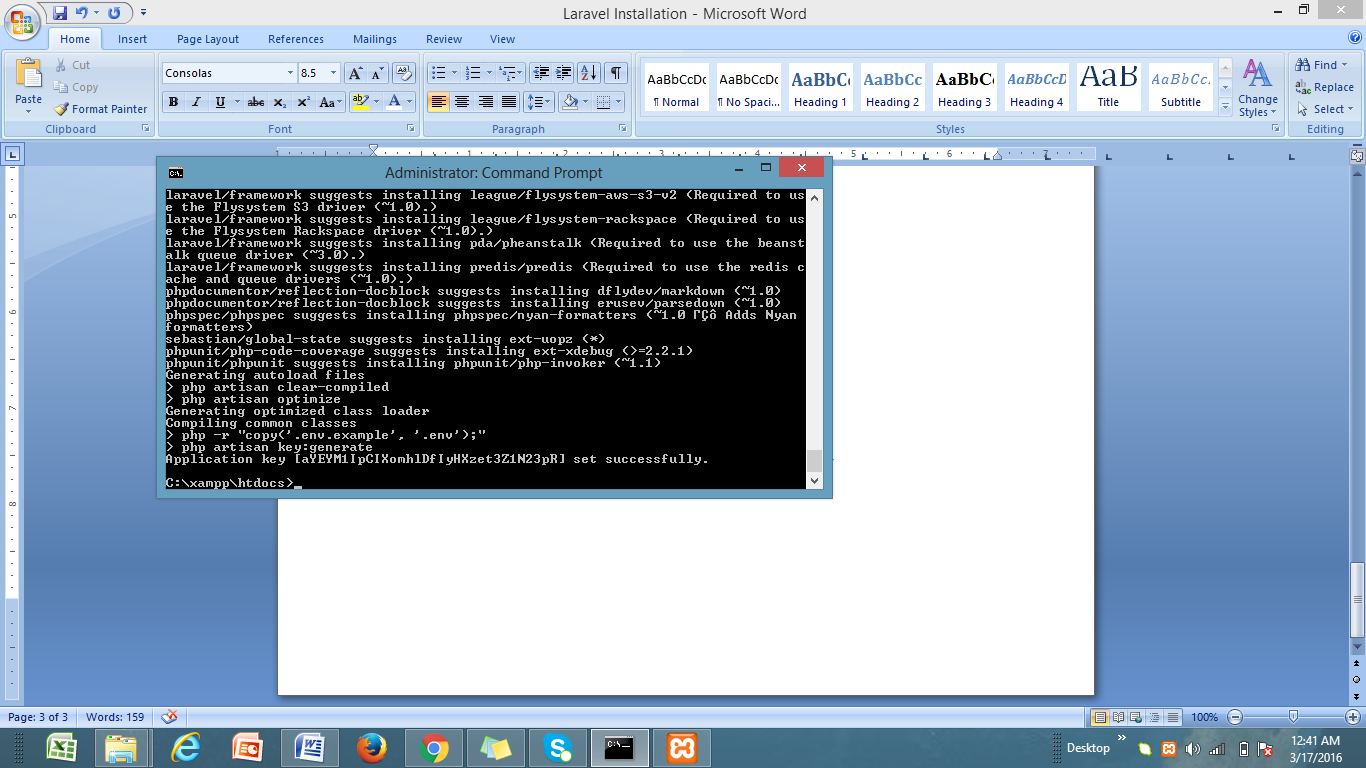
localhost/myblog

**Sample Output:**

****

****

****

****

****

**Lab # 11**

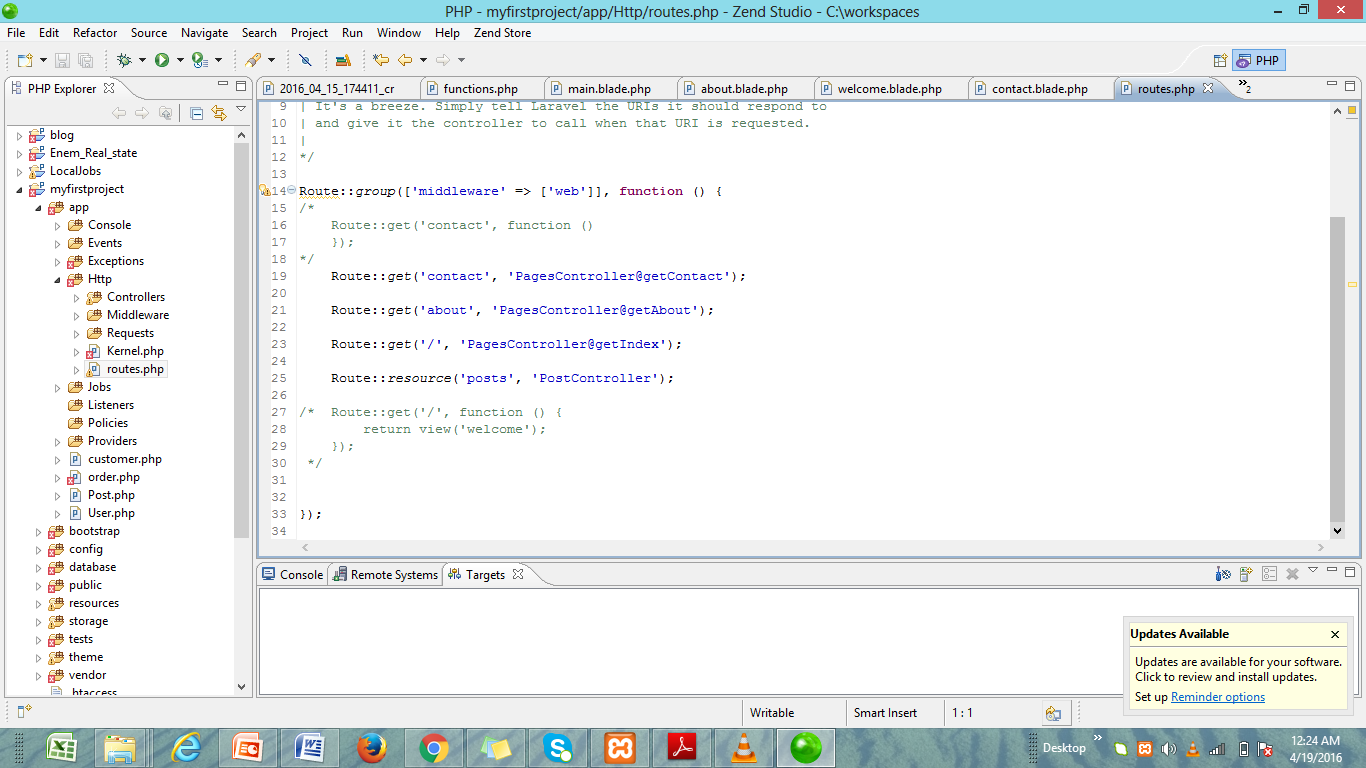
**Laravel(Routes& Controllers)**

**Lab 1:**

In this lab, we learn how to implement routes on laravel.

* Write code to implement different kinds of routes.
* Write code to implement route closure.

**Sample Output:**

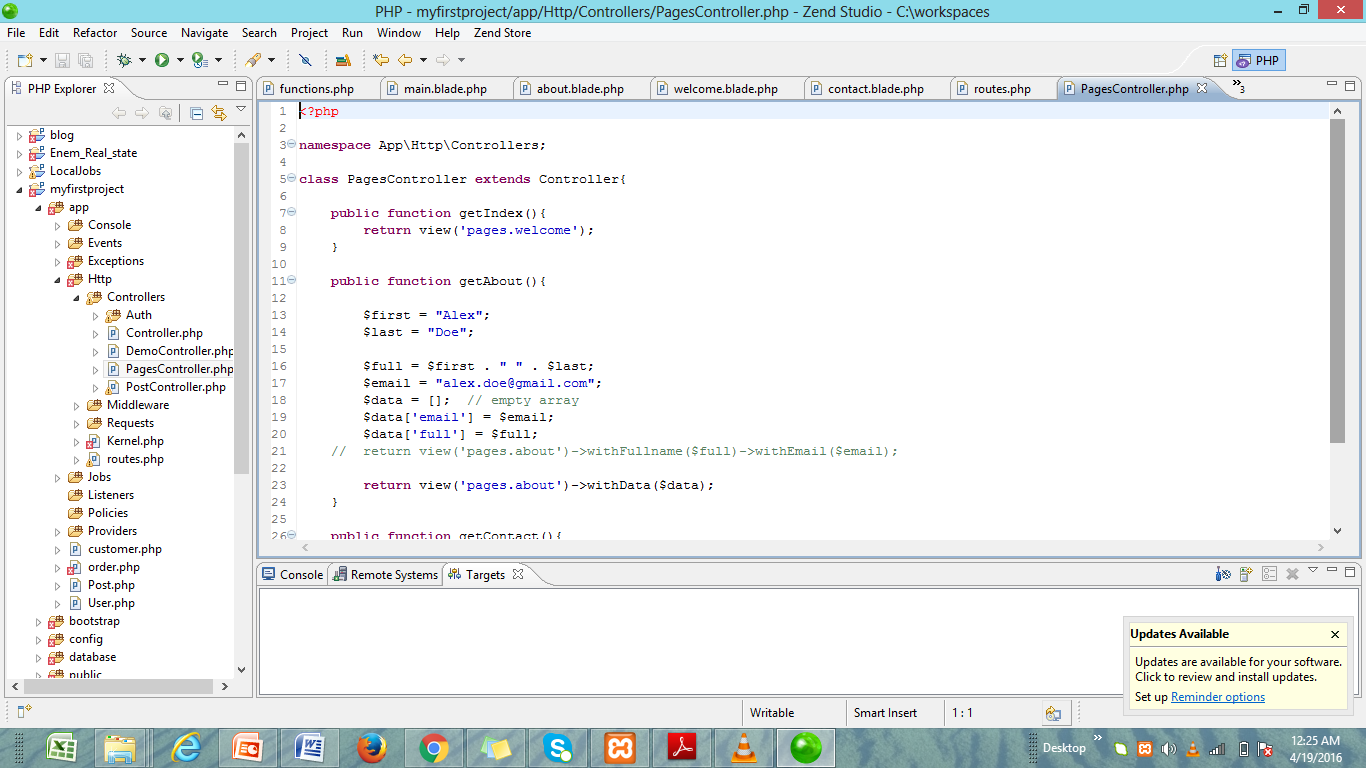
****

**Lab 2:**

In this lab task, we learn how to implement controllers on laravel.

* All controllers are store in app🡪Http folder.
* Create controller through terminal cmd by writing a command.

**Sample Output:**



**Lab # 12**

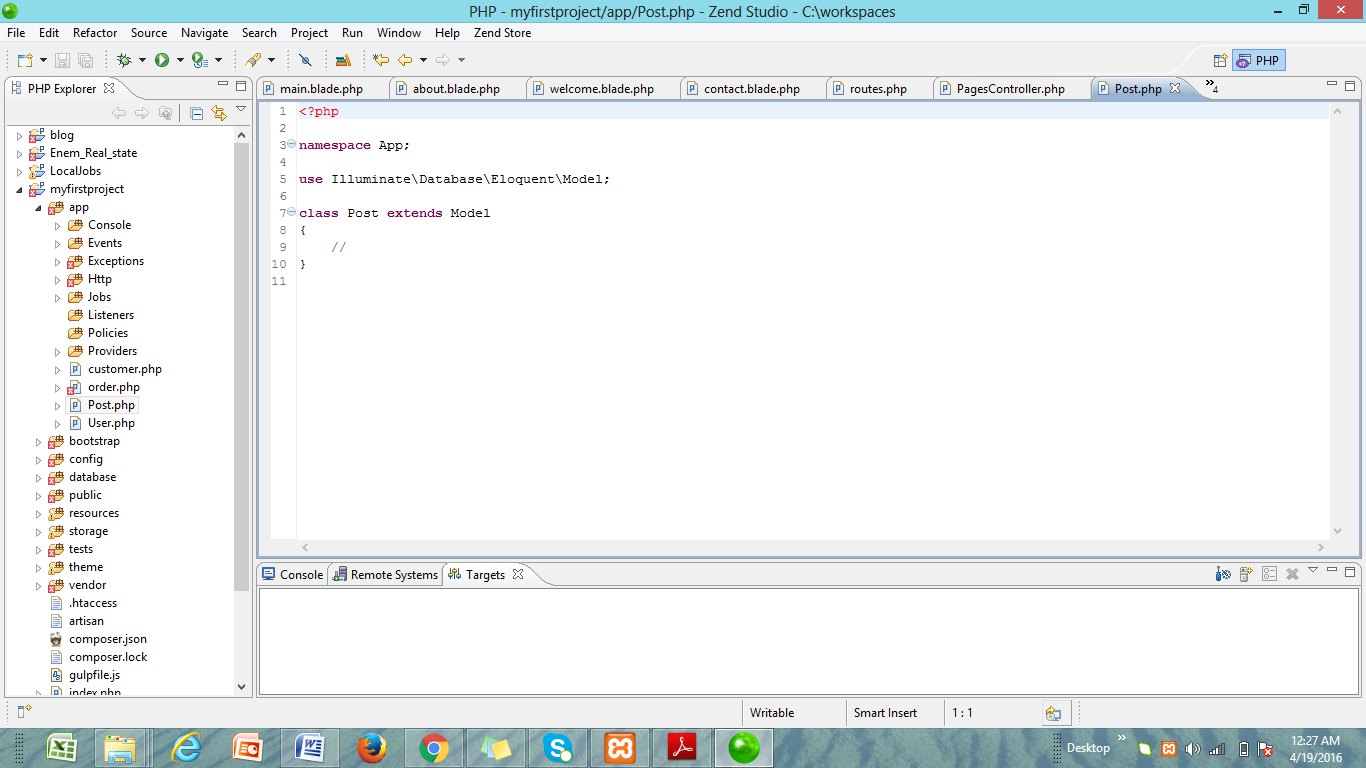
**Laravel(Model & Migrations)**

**Lab 1:**

In this lab, we learn how to create Model in laravel

* All models are store in app folder.
* Write command in terminal cmd to create a model

**Sample Output:**

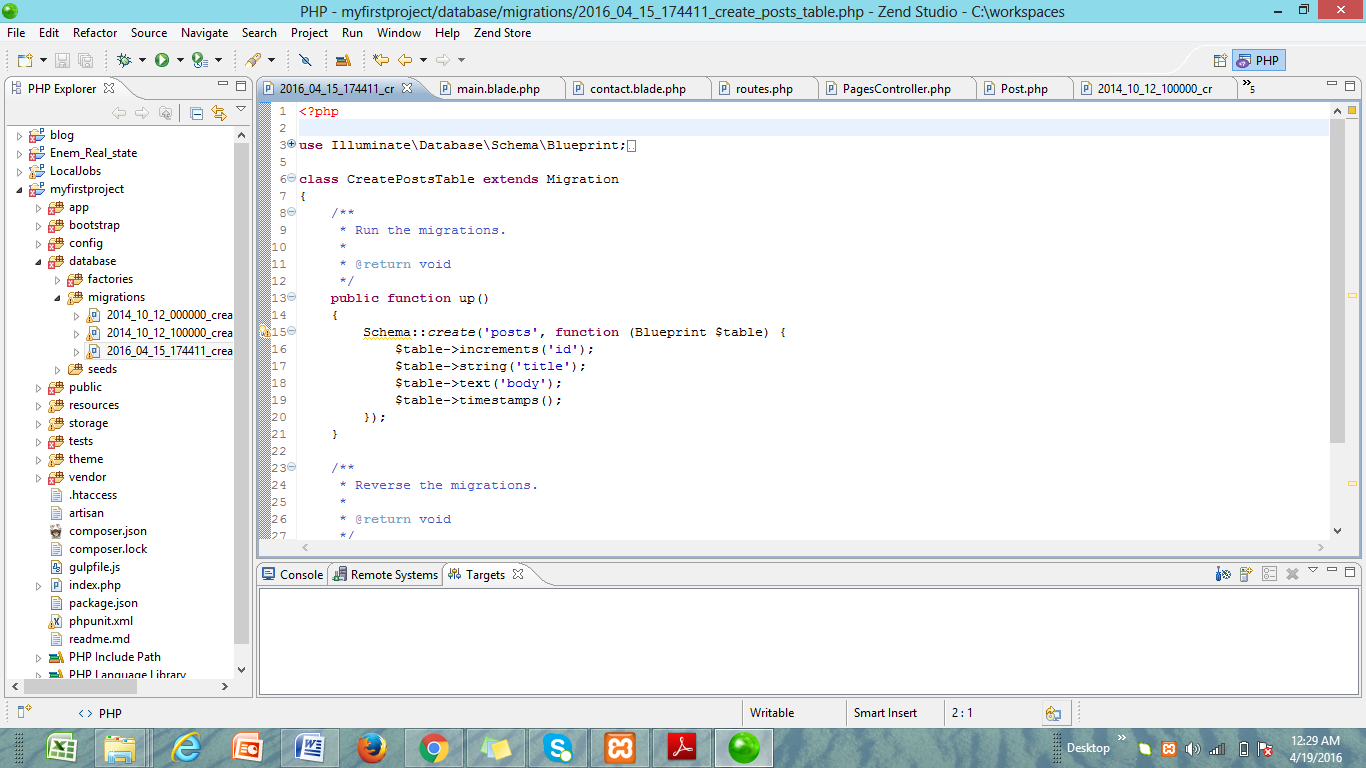


**Lab 2:**

In this lab task, we learn about migrations

* Migration is an easier way to create tables in database.
* Write code of migration in laravel.

**Sample Output:**

****

**Lab # 13**

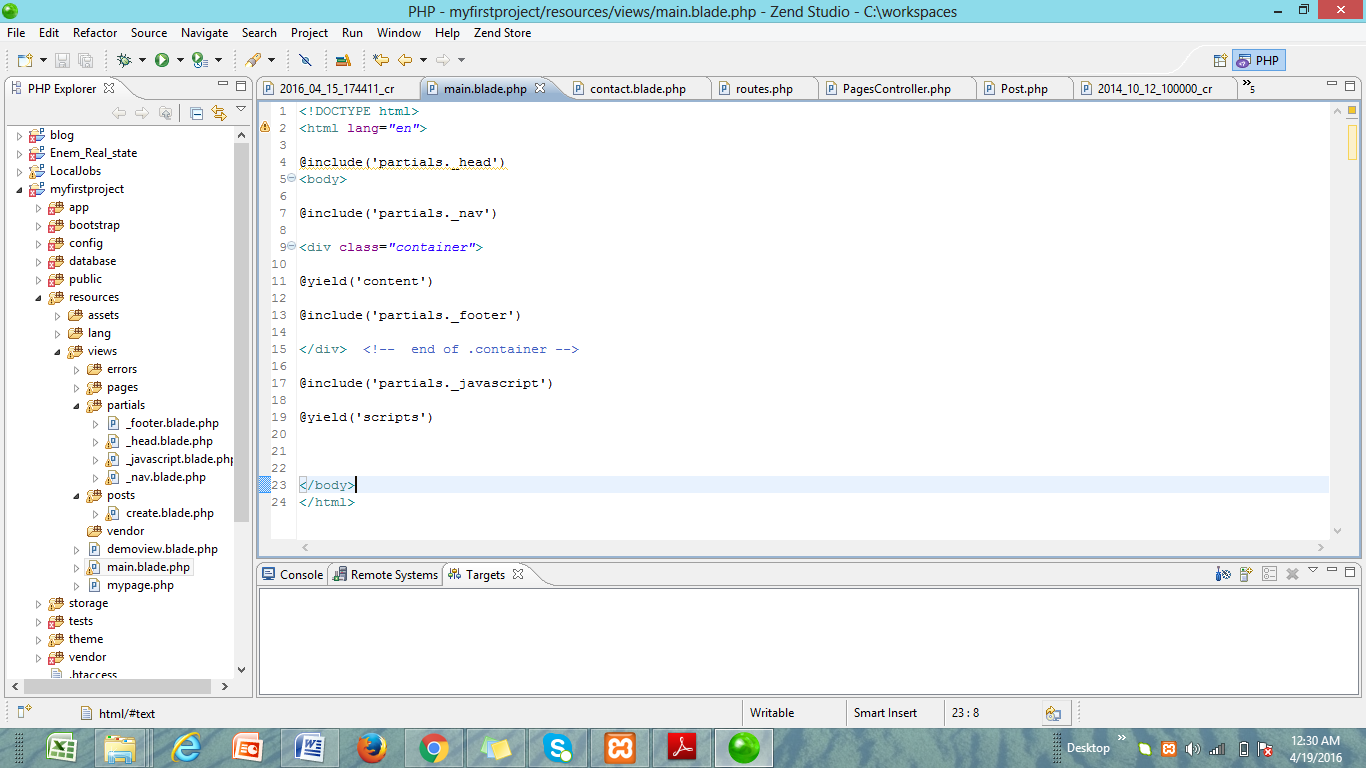
**Laravel(View)**

**Lab 1:**

In this lab, we learn how to create static front-end pages in laravel.

* In laravel, view contains simple HTML + Blade syntax.
* All views are store in resource folder.
* In this lab we create static pages in Laravel.

**Sample Output:**



**Lab # 14**

**Laravel(CRUD Operations)**

**Lab 1:**

In this lab, we learn how to convert static pages into dynamic in laravel.

* In 1st task of today’s lab, we learn how to insert data in database through laravel.

**Lab 2:**

In this lab, we learn how to convert static pages into dynamic in laravel.

* In 2nd task of today’s lab, we learn how to read data from database through laravel.

**Lab 3:**

In this lab, we learn how to convert static pages into dynamic in laravel.

* In 3rd task of today’s lab, we learn how to delete data in database through laravel.

**Lab 4:**

In this lab, we learn how to convert static pages into dynamic in laravel.

* In 4th task of today’s lab, we learn how to update/edit data in database through laravel.

**Lab # 15**

**WordPress**

**Lab 1:**

In this lab, we learn how to use cmswordpress.

* We learn how to install wordpress on our local machines.
* We learn how to customize wordpress themes and plugins.
* We create a simple website on wordpress.

**Sample Output:**

