**////////////////////////////////Start/////////////////////////////\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Before laravel**

* Download and Install vc9, vc10, vc11, vc13, vc15.
* Download and install wamp( wamp2.5,php5.5.12)
* Then go to wamp>www>index.php change line **$****suppress\_localhost=true** to **$suppress\_localhost=false**
* Next install composer
* Next install laravel

**Composer install**

* <https://getcomposer.org>
* download
* <https://getcomposer.org/download>
* windows installer>composer Setup.exe
* composer Setup.exe >next>next(do not download shell)>next(c:/wamp/php/php.exe)>next->install->next->finish
* (note check before install composer wamp>php>php.ini e ssl search uncomment extension=php\_openssl.dll)

**Resolve composer is not recognized as an internal or external command after Composer install,when write composer install in wamp/www/laravelproject**

* <http://leedavis81.github.io/global-installation-of-composer-on-windows/>
* **Open a terminal and navigate to the directory your php binary is**  
  (or any other directory that exists under your system path)
* **Run the following:** wamp/bin/php/php5.5.12 then shift+right click

php -r "eval('?>'.file\_get\_contents('https://getcomposer.org/installer'));"

* **Create a composer.bat by running the following command:**

echo "@ECHO OFF" "php '%~dp0composer.phar' %\*" > composer.bat

* And your done (you may have to restart your terminal). Just type in "composer"
* and you should see the usage options.
* Tip: If you're using windows powershell you might get the following warning;
* '■@' is not recognized as an internal or external command,
* If that's the case simply delete the composer.bat file, create a new file, copy in the following contents and save as composer.bat:

@ECHO OFF

php "%~dp0composer.phar" %\*

den restart the pc.

**Laravel install**

* download zip file from github <https://github.com/laravel/laravel> and unzip and put in on wamp>www>folderproject>shift+right click>open command window here>composer install

**create project (u can project ,in 2 ways one is described in 1 and other is described in 2)**

1. go to wamp>www>hold shift and click right mouse ->open command window here->paste composer create-project laravel/laravel in command prompt
   1. then go to laravel project then shift+ right click ,open command window here>write **php artisan serve**
2. download zip file from github <https://github.com/laravel/laravel> and unzip and put in on wamp>www>folderproject>shift+right click>open command window here>**composer install**

**Config Laravel**

* after install laravel then goes to wamp>www>laravelproject
* copy from here .env.example ,and paste it here then .env - Copy.example will create,
* then open sublime ,go to .env - Copy.example file right click and rename to .env and enter
* then goes to wamp>www>laravelproject ,shift+right click ,write php artisan key:generate

**To move Laravelproject folder one pc to another pc**

* to move laravelproject one pc to another ,first check wamp,composer are intalled ,
* then put laravelproject on **wamp/www/** directory ,then in project ,click shift+right click->open window here-> write **composer install**
* copy env.example ,and paste it here then .env - Copy.example will create,
* coz after write composer install .env will be removed
* set .env like before
* write **php artisan key:generate**

**laravel public folder Remove**

* <https://www.youtube.com/watch?v=zQsyxumCZqQ>
* Go to mainproject/public>>

1. .htacess
2. favicon.ico
3. index.php
4. robots.txt
5. web.config

1.      cut these 5 files from public folder,and then paste on the main project folder that’s means out side of public folder… mainproject/files

2.      Next after paste ,open index.php ,modify

        require \_\_DIR\_\_.'/…/bootstrap/autoload.php'; to

         require \_\_DIR\_\_.'/bootstrap/autoload.php';

 3. modify

$app = require\_once \_\_DIR\_\_.'/../bootstrap/app.php'; to

$app = require\_once \_\_DIR\_\_.'/bootstrap/app.php';

**import tables in db by migration command**

1. Php artisan make:model userdefined\_modelname

**Laravel class ‘input’ not found solution**

Add 'Input' => Illuminate\Support\Facades\Input::class, in config/app.php alias

**Laravel class ‘form’ not found solution**

<https://laravelcollective.com/docs/5.2/html>

1. Begin by installing this package through Composer. Edit your project's composer.json file to require laravelcollective/html.

"require": {

"laravelcollective/html": "5.2.\*"

}

1. Next, update Composer from the Terminal:

composer update

**[1 and 2 instead e composer require laravelcollective/html composer a paste korleo hobe]**

1. Next, add your new provider to the providers array of config/app.php:

'providers' => [

// ...

Collective\Html\HtmlServiceProvider::class,

// ...

],

1. Finally, add two class aliases to the aliases array of config/app.php:

'aliases' => [

// ...

'Form' => Collective\Html\FormFacade::class,

'Html' => Collective\Html\HtmlFacade::class,

// ...

],

* *Another way if package not install from composer*

<http://stackoverflow.com/questions/32666164/laravel-5-installing-form-and-html-classes-without-composer>

<https://github.com/LaravelCollective/html>

|  |  |
| --- | --- |
|  | **Installing laracollective/html:**  **1)** Download zip package from [extension's git repository](https://github.com/LaravelCollective/html)  **2)** Unpack contents and create this directory structure: laravelcollective/html/{contents of html-5.1 directory}  **3)** Copy this structure to the vendor folder in your Laravel installation.  **4)** In vendor/composer/autoload\_classmap.php add these lines  'Collective\\Html\\FormBuilder' => $vendorDir . '/laravelcollective/html/src/FormBuilder.php',  'Collective\\Html\\FormFacade' => $vendorDir . '/laravelcollective/html/src/FormFacade.php',  'Collective\\Html\\HtmlBuilder' => $vendorDir . '/laravelcollective/html/src/HtmlBuilder.php',  'Collective\\Html\\HtmlFacade' => $vendorDir . '/laravelcollective/html/src/HtmlFacade.php',  'Collective\\Html\\HtmlServiceProvider' => $vendorDir . '/laravelcollective/html/src/HtmlServiceProvider.php',  after  'ClassPreloader\\Parser\\NodeTraverser' => $vendorDir . '/classpreloader/classpreloader/src/Parser/NodeTraverser.php',  so that Laravel knows where to look for these classes when requested.  **5)** Add this line as a last element to return array(...) in vendor/composer/autoload\_files.php  'f18cc91337d49233e5754e93f3ed9ec3' => $vendorDir . '/laravelcollective/html/src/helpers.php',  **6)** Add this line to return array(...) in vendor/composer/autoload\_psr4.php  // 'Doctrine\\Instantiator\\' => array($vendorDir . '/doctrine/instantiator/src/Doctrine/Instantiator'),  'Collective\\Html\\' => array($vendorDir . '/laravelcollective/html/src'),  // 'ClassPreloader\\' => array($vendorDir . '/classpreloader/classpreloader/src'),  **7)** Add providers to providers array of config/app.php:  'providers' => [  // ...  Collective\Html\HtmlServiceProvider::class,  // ...  ],  **8)** Add two class aliases to the aliases array of config/app.php:  'aliases' => [  // ...  'Form' => Collective\Html\FormFacade::class,  'Html' => Collective\Html\HtmlFacade::class,  // ...  ],  **Laravel Controller**   * How to create controller   Php artisan make:controller userdefined\_controller\_name   * How to access view ,suppose In AdminController.php   Public function login(){  Return view(‘admin.login’);  }  Then Route.php  Route::get(‘/login’,array(‘as’=>’login’,’uses’=>’AdminController@login’);  Route::get(‘/url,array(‘as’=>’login’,’uses’=>’controllername@functionname);   * To **retrieve all data from database** use modelname::all() and use App\Customer; in controller upper part   public function index()  {  $customers=Customer::all();  return view('customer',['customers'=>$customers]);  }  -**various ways of retrieve data’s**  1. $customers=Customer::all();  2. $product\_lists=Product::lists('productname','id');  3. $data=Product::select('price')->where('id',$request->id)->first();  4. $students=DB::table('tblstudent')->get();  5. $data = Project::all(array("id", "project\_title"));  6. $data\_qry= DB::select("SELECT ex.\*,  proj.project\_title,  a.first\_name as 'addedBy',  p.first\_name as 'paidTo'  FROM expenses ex, projects proj, users p, users a  WHERE proj.id= ex.project\_id AND  ex.paid\_to= p.id AND  ex.added\_by= a.id AND  ex.expenses\_date BETWEEN ? AND ?  AND project\_id=?",[$var\_start\_date,$var\_end\_date,$var\_project\_name\_invoice]);  7. $data=DB::table('ch\_info')->select('name', 'ch\_id as ch','addr','id')->take(100)->get();  8.$datapr=DB::select("SELECT pr.product\_id as prid,  pr.p\_code as pcode,  pr.brand\_name as brand,  pr.pack\_s as ppack,  pr.p\_group as pgroup,  pr\_price.T\_P as price  FROM product\_info pr, product\_info\_price pr\_price WHERE pr.p\_code=pr\_price.p\_code");   * To redirect same page use return back(); in controller   Or  return redirect()->back()->with('success','Expenses Created Successfully');   * To insert data model\_name::insert($data);   or  $customers=new Incustomer;  $customers->firstname=$request->fn;  $customers->lastname=$request->ln;  $customers->save();  Or  $response=Salary::create($data);   * To return view with data one is compact another []   $product\_lists=Product::lists('productname','id');  return view('invoice.info',compact('product\_lists'));  or  $customers=Customer::all();  return view('customer',['customers'=>$customers]);  or  $data = Project::all(array("id", "project\_title"));  $users = EmployeeModel::all(array("id", "first\_name"));  return view('addsalary')->withProjects($data)->withUsers($users);  or  return view('projectBudget' )->with('success',"Budget Added")->with('project\_title', $response->project\_title)->with('project\_id', $response->id)->with('budget',2000);  or  $usersdata=DB::table('users')->get();  $projectdata =DB::table('projects')->get();  return view('view-invoice',compact('usersdata','projectdata'));   * To update data in database table   DB::update('update users set votes = 100 where name = ?', ['John']);  Or  DB::table('users')->where('id', 1)->update(['price' => 1]);  Composer laravel cmd:   * Php artisan make:auth (Authentication) * Php artisan migrate // Run migration * Php artisan make:controller userdefined\_controller\_name * Php artisan key:generate * Composer install // For install composer packages * php artisan make:request **StoreBlogPostRequest** * Php artisan make:model userdefined\_model\_name   **Prob :Cannot access empty property**  Solution:  foreach ($object->someArray as $key->$val) {  // do something  }  Of course, I should have had:  foreach ($object->someArray as $key=>$val) {  // do something  } TokenMismatchException in VerifyCsrfToken.php line 67 in Laravel 5.2 Add <input type="hidden" name="\_token" value="{{ csrf\_token() }}"> before every input field like  <input type="hidden" name="\_token" value="{{ csrf\_token() }}">  <input type="text" name="last\_name" id="last\_name" placeholder="Last Name" class="form-control">  **How to use Datatable In Laravel 5.1**   * Database: datatable * table-tblstudent   -id  -first\_name  -last\_name  -pob  -phone   * composer require yajra/laravel-datatables-oracle * In provider (config/app.php)   Yajra\Datatables\DatatablesServiceProvider::class,   * In aliases(config/app.php)   'Datatables'=>Yajra\Datatables\Datatables::class,   * php artisan vendor:publish   -------------------------downloading coding below---------------------------------  -cdn.datatables.net/1.10.10/css/jquery.dataTables.css ---and put this file inside of assets/css/  -cdn.datatables.net/1.10.10/js/jquery.dataTables.js ---and put this file inside of assets/js/  -https://jquery.com/download/ ---and put this file inside of assets/js/  -https://github.com/DataTables/DataTables/tree/master/media/images ---and put this file inside of assets/ images/  favicon.ico sort\_asc.png sort\_asc\_disabled.png sort\_both.png sort\_desc.png sort\_desc\_disabled.png   * add following in before of body end tag   {!!Html::script('../resources/assets/js/jquery-1.12.1.js')!!}  {!!Html::script('../resources/assets/js/jquery.dataTables.js')!!}  <script type="text/javascript">  $('.student').dataTable({    });  </script>  </body>  </html>  And add following codes inside of header part  {!!Html::style('../resources/assets/css/jquery.dataTables.css')!!}  {!!Html::style('../resources/assets/css/bootstrap.css')!!}   * php artisan make:controller StudentController * use use DB; in controller's first * add this function in controller   public function index(){  $students=DB::table('tblstudent')->get();  return view('student',compact('student'));  }   * create student.blade.php in resources/views * in route.php ,write Route::get('/datTable',array('as'=>'viewStudent','uses'=>'StudentController@index'));  Setting Sublime Text up For Laravel Blade <https://www.youtube.com/watch?v=FeuZtBdS20c>   * Go to below link   <https://packagecontrol.io/installation#st3>   * copy following lines for sublime text3   import urllib.request,os,hashlib; h = '2915d1851351e5ee549c20394736b442' + '8bc59f460fa1548d1514676163dafc88'; pf = 'Package Control.sublime-package'; ipp = sublime.installed\_packages\_path(); urllib.request.install\_opener( urllib.request.build\_opener( urllib.request.ProxyHandler()) ); by = urllib.request.urlopen( 'http://packagecontrol.io/' + pf.replace(' ', '%20')).read(); dh = hashlib.sha256(by).hexdigest(); print('Error validating download (got %s instead of %s), please try manual install' % (dh, h)) if dh != h else open(os.path.join( ipp, pf), 'wb' ).write(by)   * Open sublime ,View > Show Console menu and paste upper lines into here * Then close sublime * The again open sublime and go to Tools>common palette (ctlr+shift+p) * Write package install ,click it,write laravel hightight and click it  Class 'App\Http\Controllers\Validator' not foundGenerating migration php artisan make:migration create\_users\_table  php artisan migrate |
|  |  |