Day 1:
9am-10am:

- ✓ Setup xampp
- ✓ Setup composer
- ✓ Use composer to create a project
- ✓ View default home page
- ✓ Use artisan to check Laravel version
- ✓ Explaining the Model-View-Controller architecture
- ✓ Other PHP MVC Frameworks

10am-11am:

- ✓ Checking the folders for the project:
    - /resources/views
    - /app/Http/Controllers
    - /routes/web.php
- ✓ Creating a new home page
- ✓ Route the home page
- ✓ Create 3 more pages and practice routing
- ✓ Calling views using a controller class
    - Create controller using artisan
        - php artisan make:controller controller1
    - import other namespaces

```
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Input;
use View;
```

11am-12nn:

- ✓ Create a blade template in /resources/views/template1.blade.php (see sample page 8-9)
- ✓ Explain blade position markers (yield,section,extends)
- ✓ Add navigation to blade template, hyperlinks can be as follows:
    - Relative
        - <a href="./page1">page1</a>
    - Absolute
        - <a href="localhost/project1/public/page1">page 1</a>

12nn-1pm: LUNCH

1pm-2pm:

- ✓ Passing data and submitting forms
    - Get and post methods
    - Checking security middlewares
        - App/Http/Middleware/VerifyCsrfToken.php
    - Adding a {{ csrf_field() }}
    - Displaying data:
        - Escaped (prevents xss attacks) {{ }}
        - Unescaped (raw) {!! !!}
- ✓ View→route(processing)→View
- ✓ View→controller(processing)→View

2pm-3pm:

- ✓ Using the request object
    - Add namespace

```
use Illuminate\Http\Request;
```

- Use as follows:

```
Route::post('/page2', function(Request $request) {
   $data['text1']=$request->input('text1');
   $data['text2']=$request->input('text2');

   return view('pages/page2')->with('data',$data);
});
```

- ✓ Using route parameters

```
Route::put('user/{id}', 'UserController@update');
```

Sample usage in a controller function

```
public function update(Request $request, $id)
   {
      //
   }
```

- ✓ Other request object functions

```
$a = $request->path()               //sample output: "pages/page1"
If($request->is('admin/*'){…}       //checks if path matches a pattern
$url = $request->url()              // Without Query String
$url = $request->fullUrl();         // With Query String...

$input = $request->only(['username', 'password']);
$input = $request->only('username', 'password');
$input = $request->except(['credit_card']);
$input = $request->except('credit_card');

if ($request->has('name')) { … }
```

3pm-4pm:

- ✓ Automatic input trimming and normalization
    - By default, Laravel includes the TrimStrings and ConvertEmptyStringsToNull middleware in your application's global middleware stack.
- ✓ Enabling and disabling middleware
    - /app/http/kernel.php
- ✓ File management (/storage/app/public)
    - Create, write and append if existing then display the contents and the file size

```
Route::post('/createfile',function(Request $request){

        $filename=$request->input('text1');
        $content=$request->input('area1');
```

```
    if(Storage::disk('local')->exists($filename)){
        Storage::disk('local')->append($filename,$content);
        $entries=Storage::disk('local')->get($filename);
        $size=Storage::disk('local')->size($filename);
        echo "success!<br />
            contents:<br />
            <textarea>$entries</textarea><br />
            file size:$size bytes<br />
            <a href='./page3'>back</a>";
    }else{
        Storage::disk('local')->put($filename,$content);
        $entries=Storage::disk('local')->get($filename);
        echo "success!<br />
            contents:<br />
            <textarea>$entries</textarea><br />
            file size:$size bytes<br />
            <a href='./page3'>back</a>";
    }
} );
```

end of day 1