# Setup

* To work with laravel you should have composer and Wamp/Xampp installed on your device
* To install laravel run the following command:  
  composer create-project laravel/laravel app
* To navigate to the created laravel project run following command:  
  cd app
* To install Jetstream, run following command:  
  composer require laravel/Jetstream
* To install livewire run following command:  
  php artisan jetstream:install livewire
* To install dependencies run following command:  
  npm install  
  npm run dev
* To create the database navigate to .env file and change the database name: (line 14)  
  DB\_DATABASE=practice  
  then run the following command:  
  php artisan migrate
* Then start the laravel server using following command:  
  php artisan serve

# Git

* To use the laravel project after cloning from git do the following
* To install vendor folder run:  
  composer install
* To install node\_modules folder run:  
  npm install  
  npm run dev
* Create .env file and copy entire .env.example into .env and run:  
  php artisan key:generate
* Change database name and then run:  
  php artisan migrate
* Now start laravel server:  
  php artisan serve

# Route - Basic

* All routes are defined in web.php file
* Route::get('/greeting', function () {  
   return 'Hello World';  
  });  
  this route gets the url and then execute the function
* Route::get('/user', [UserController::class, 'index']);  
  this route gets the url and then redirect to the controller in which the function will be executed
* Following are the routers methods:  
  Route::get($uri, $callback);  
  Route::post($uri, $callback);  
  Route::put($uri, $callback);  
  Route::patch($uri, $callback);  
  Route::delete($uri, $callback);  
  Route::options($uri, $callback);
* When handling multiple http requests, you can use match:  
  Route::match(['get', 'post'], '/', function () {  
   // ...  
  });
* You can also use any:  
  Route::any('/', function () {  
   // ...  
  });
* You can add dependency injection in your route’s callback section:  
  Route::get('/users', function (Request $request) {  
   // ...  
  });
* Any form pointing to POST, PUT, PATCH, or DELETE routes should have @csrf token  
  <form method="POST" action="/profile">  
   @csrf  
   ...  
  </form>
* When using route that redirects to another URI, use redirect routes:  
  Route::redirect('/here', '/there');
* When using route that redirects to view, use view route:  
  Route::view('/welcome', 'welcome');
* You can also pass a data with view:  
  Route::view('/welcome', 'welcome', ['name' => 'Taylor']);

# Route - Parameters

* You can pass parameters with route:  
  Route::get('/user/{id}', function (string $id) {  
   return 'User '.$id;  
  });
* You can also make the parameters optional:  
  Route::get('/user/{name?}', function (?string $name = null) {  
   return $name;  
  });
* You can pass regular expression in route to define how parameter should be constrained:  
  Route::get('/user/{name}', function (string $name) {  
   // ...  
  })->where('name', '[A-Za-z]+');
* You can also define global constraint by adding the following code inside routeserviceproviders boot function:  
  public function boot(): void  
  {  
   Route::pattern('id', '[0-9]+');  
  }

# Route - Named