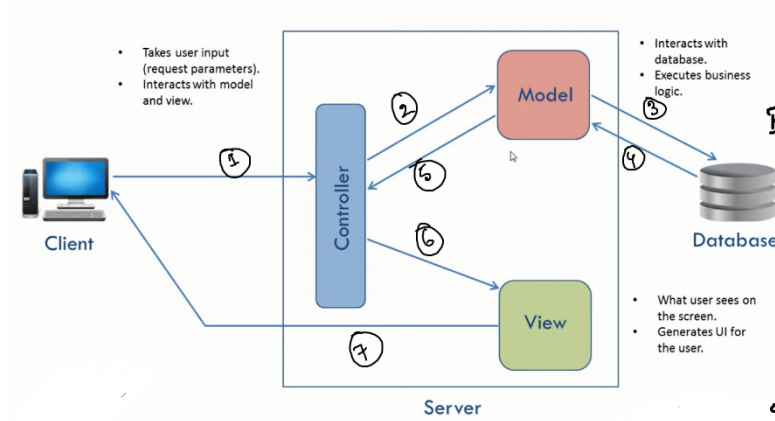


# Model View Controller

এক্ষেত্রে ইউজার এর রিকোয়েস্ট যাবে controller এর কাছে → controller মডেল (যা ডাটাবেসকে represent করে) থেকে ডাটা নিবে → সেই ডাটা view/webpage এ show করবে।



৭ View Raw Data গুলো  
শুদ্ধভাবে নাহি দেয়  
user এর কাছে পাঠায়।

১ User Request আসে server-এ  
Browser দিয়ে আসবে cat এর picture নেওয়া

২ Controller user এর data/  
request receive করে,

৩ Controller request/data কে  
পাঠাবে model এর কাছে;

model: select \* from cat;

query নামিয়ে data retrieve করে database  
থেকে;

এই Raw data গুলো Controller  
এর কাছে পাঠাবে, Raw Data গুলো Human  
Readable না তাই Controller Data পাঠাবে  
View এর কাছে।

## Model

```
SELECT * FROM cats;
```

## View

```
<body>  
  <h1>Cats</h1>  
  ...  
</body>
```

2. Get Cat Data

1. Get Cats



## Controller

```
if (success)  
  View.cats
```

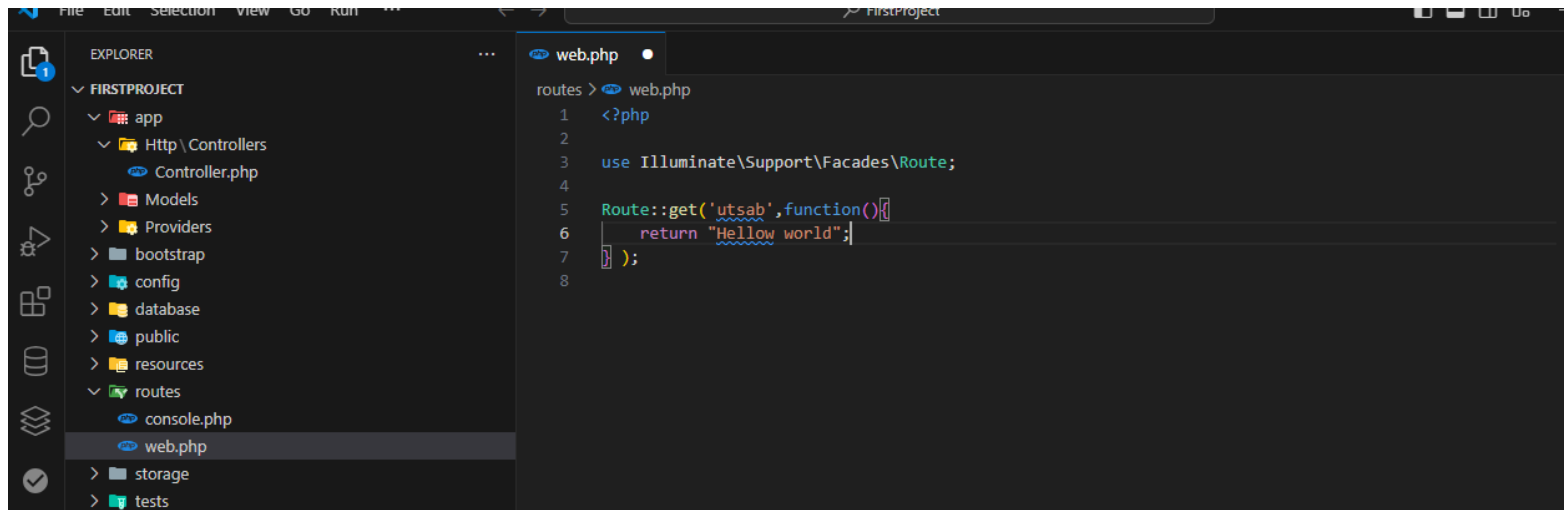
3. Get Cat Presentation

③ Data [?] Human readable  
[?] [?] controller data [?]  
view [?] [?], view data [?]  
[?] [?] represent [?] [?]  
[?] [?]

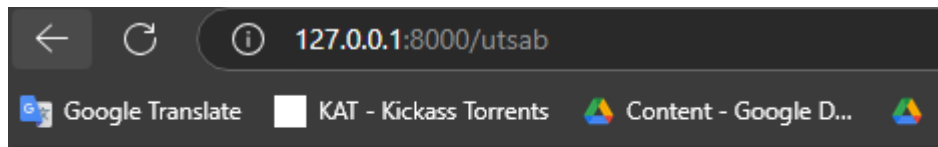
① Request [?]  
'cat' data [?]  
② controller request  
[?] model [?]  
• model data retrieve  
[?] database [?]  
[?] data retrieve success  
[?] [?] controller  
[?] [?]

## 1. Create simple Laravel route and show hellow world

Create route in web.php->and create function for showing data



```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 Route::get('utsab',function() {
6     return "Hellow world";
7 });
8
```



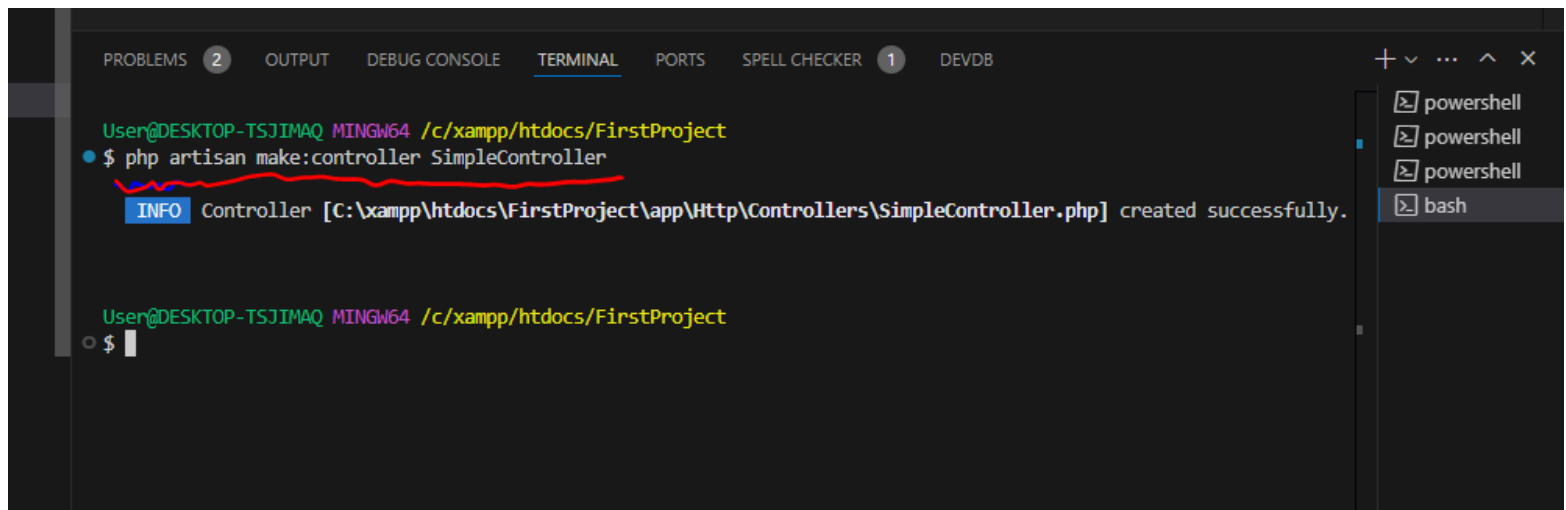
Hellow world

## 2. Create simple route and show message using controller

(i) php artisan make:controller SimpleController

### Step 1: Create a Simple Controller

You can use the `artisan` command to create a controller. For example:



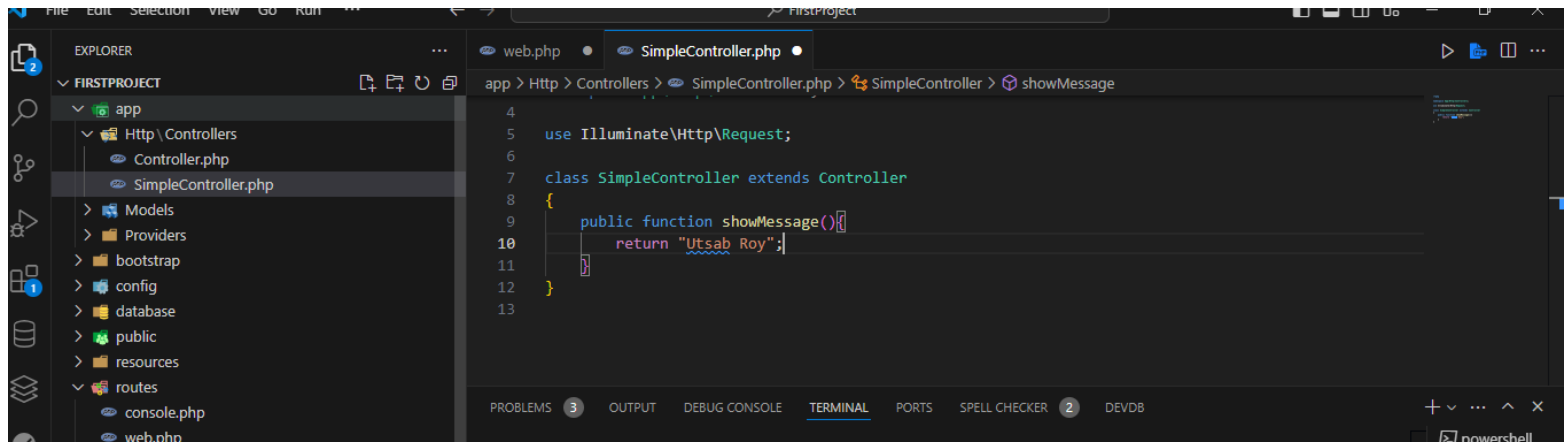
```
User@DESKTOP-TSJ1MAQ MINGW64 /c:/xampp/htdocs/FirstProject
$ php artisan make:controller SimpleController
INFO Controller [C:\xampp\htdocs\FirstProject\app\Http\Controllers\SimpleController.php] created successfully.

User@DESKTOP-TSJ1MAQ MINGW64 /c:/xampp/htdocs/FirstProject
$
```

(ii) method তৈরি করব controller এ

## Step 2: Define the Method in the Controller

Open the `SimpleController.php` file and define a method to return the message. For example:



(iii) route create করব এবং controller এর মেথডকে কল করব।

Best way for call

The **best way** to call a controller method from a route in Laravel is to use the **array syntax**, like this:

```
php  
Route::get('/example', [ExampleController::class, 'method']);
```

Old/Deprecated system

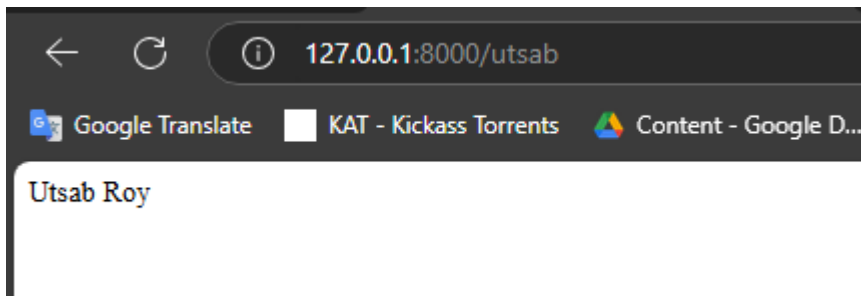
- For example:

```
php  
Route::get('/example', 'ExampleController@method'); // old syntax
```

## Step 3: Define the Route in `web.php`

In the `routes/web.php` file, define a route that points to the `showMessage` method in your `SimpleController`. For example:

```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\SimpleController;
5
6 Route::get('/utsab',[SimpleController::class,'showMessage']);
7
```



### 3.Show message from blade file

(i)create blade file

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>Document</title>
8 </head>
9 <body>
10     Hellow utsab
11 </body>
12 </html>
13
```

(ii)show blade file from controller

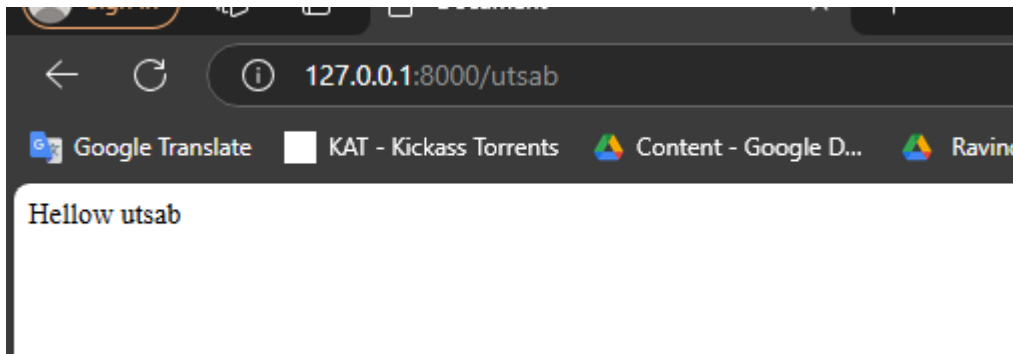


```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class SimpleController extends Controller
8  {
9      public function showMessage(){
10         return view('dashboard.home');
11     }
12 }
13
```

(iii)now call controller method



```
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\SimpleController;
5
6  Route::get('/utsab',[SimpleController::class,'showMessage']);
7
```



## 4. Sending data from controller to blade

### 1. Using Associative Array


```
php Copy code  
  
public function exampleMethod()  
{  
    return view('example', [  
        'name' => 'John Doe',  
        'email' => 'johndoe@example.com'  
    ]);  
}
```

### 2. Using compact()

```
php Copy code  
  
public function exampleMethod()  
{  
    $name = 'John Doe';  
    $email = 'johndoe@example.com';  
  
    return view('example', compact('name', 'email'));  
}
```

### 3. Using `with()` Method


php

 Copy code

```
public function exampleMethod()
{
    return view('example')
        ->with('name', 'John Doe')
        ->with('email', 'johndoe@example.com');
}
```

### 4. Using `with()` for Multiple Variables


php

 Copy code

```
public function exampleMethod()
{
    return view('example')->with([
        'name' => 'John Doe',
        'email' => 'johndoe@example.com'
    ]);
}
```

### 7. Returning Data from a Model

php

 Copy code

```
public function exampleMethod()
{
    $users = User::all(); // Fetching data from a database table
    return view('example', ['users' => $users]);
}
```

(i) create controller and send data using associative array(key value pair) from view method



```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class SimpleController extends Controller
8  {
9      public function showMessage(){
10         $name='utsab';
11         $email='utsab@duet.ac.bd';
12         //send data using associative array
13         return view('dashboard.home',['namex'=>$name,'emaily'=>$email]);
14     }
15 }
16

```

(ii) show data using

### 1. Use `{{ }}` for Outputting Variables

- Blade uses `{{ }}` to escape and display data safely.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Document</title>
8  </head>
9  <body>
10     <p>Name = {{ $namex }}</p>
11     <p>Email = {{ $emaily }}</p>
12 </body>
13 </html>
14

```