

Backend Development

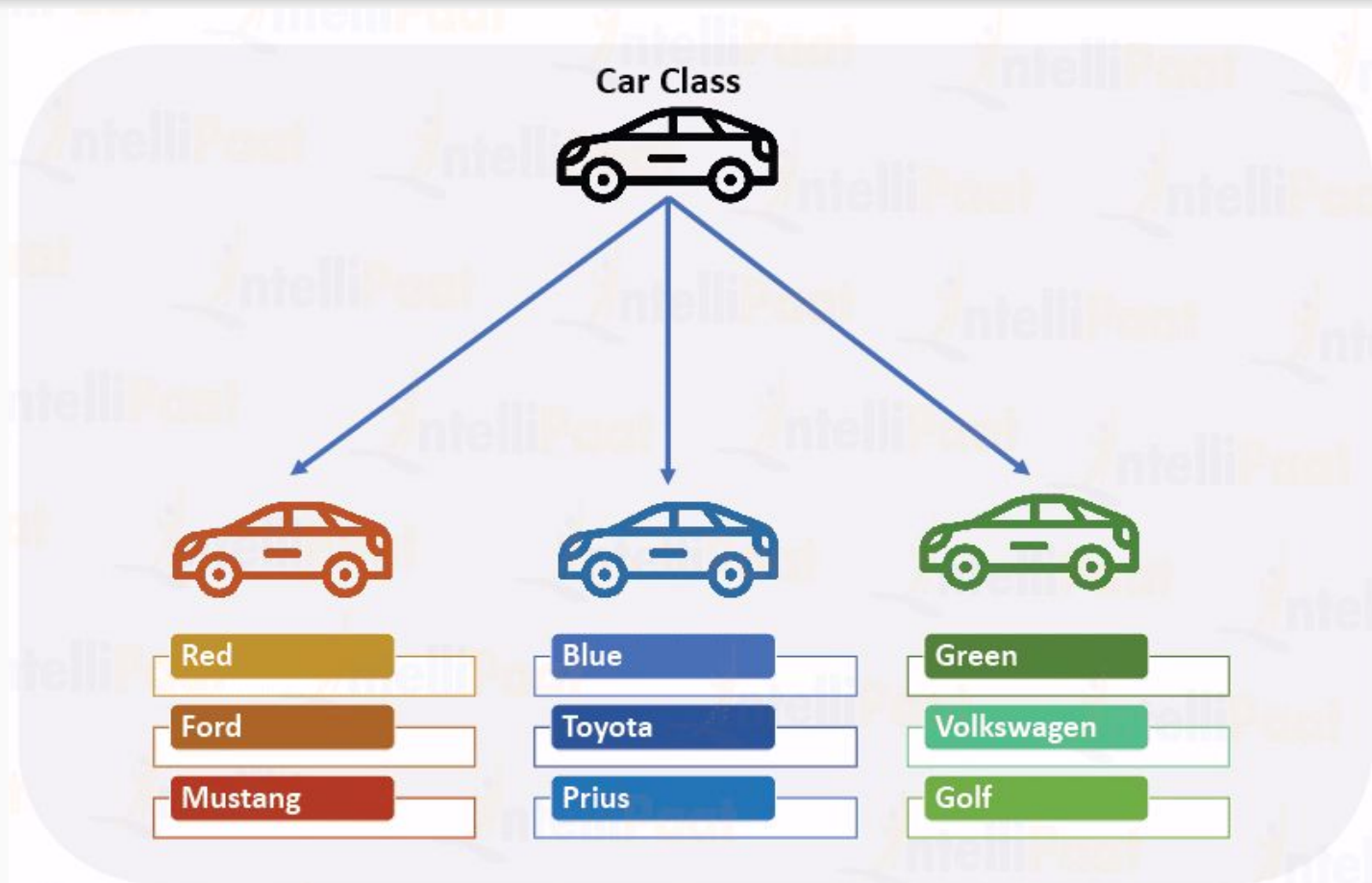
Laravel

Learning Objectives

- OOP in PHP
- MVC
- Laravel
 - Installation & Run
 - Discovery
 - Routing
 - View
 - Controllers
 - Models

Object Oriented Programming

- Why can't a child answer $2+3$ but can answer 2 apples + 3 apples ?



OOP - Access Modifiers

```
class Student {  
    public $name;  
    protected $gender;  
    private $age;  
}  
  
$std = new Student();  
$std->name = "Ahmed";  
$std->gender = "male"; // error;  
$std->age = 20; // error;
```

OOP - Properties & Methods

```
class Student {  
    public $name;  
  
    public function speak() {  
        echo "Hi, I'm $this->name";  
    }  
}  
  
$std = new Student();  
$std->name = "Ahmed";  
$std->speak(); // Hi, I'm Ahmed
```

OOP - Constructor

```
class Student {  
    public $name, $age;  
    public function __construct($n, $a) {  
        $this->name = $n;  
        $this->age = $a;  
    }  
    public function speak() {  
        echo "Hi, I'm $this->name, $this->age years old!";  
    }  
}  
  
$std = new Student("Ahmed", 21);  
$std->speak(); // Hi, I'm Ahmed, 21 years old!
```


OOP - Inheritance

```
class Human {  
    public $name;  
    public function __construct($n) {  
        $this->name = $n;  
    }  
    public function speak() {  
        echo "Hi, I'm $this->name";  
    }  
}  
  
class Student extends Human {}  
  
$std = new Student("Ahmed", 21);  
$std->speak(); // Hi, I'm Ahmed, 21 years old!
```


OOP - Inheritance

```
class Human {  
    public function speak() {  
        echo "I'm Human";  
    }  
}  
  
class Student extends Human {  
    public function speak() {  
        echo "I'm Student";  
    }  
}  
  
$std = new Student();  
$std->speak(); // I'm Student
```

```
class User {  
    protected string $table = "students";  
  
    public int $id;  
    public string $username, $email;  
    private string $password;  
    public $created_at, $updated_at;  
}
```

	#	Name	Type
<input type="checkbox"/>	1	id 	int(10)
<input type="checkbox"/>	2	username	varchar(121)
<input type="checkbox"/>	3	email	varchar(121)
<input type="checkbox"/>	4	password	varchar(255)
<input type="checkbox"/>	5	created_at	timestamp
<input type="checkbox"/>	6	updated_at	timestamp

```
class User {
    protected string $table = "students";
    public static function find($id) {
        $query = "SELECT * FROM $this->table";
        $query .= " WHERE id = $id";
        $result = $conn->query($query);
        $row = $result->fetch_assoc();
        $this->id = $row["id"];
        $this->username = $row["username"];
        $this->email = $row["email"];
        $this->password = $row["password"];
    }
}

$user = User::find($id);
```

Ignore using \$this within a static function, we should use self::\$table as a static property or create a new instance

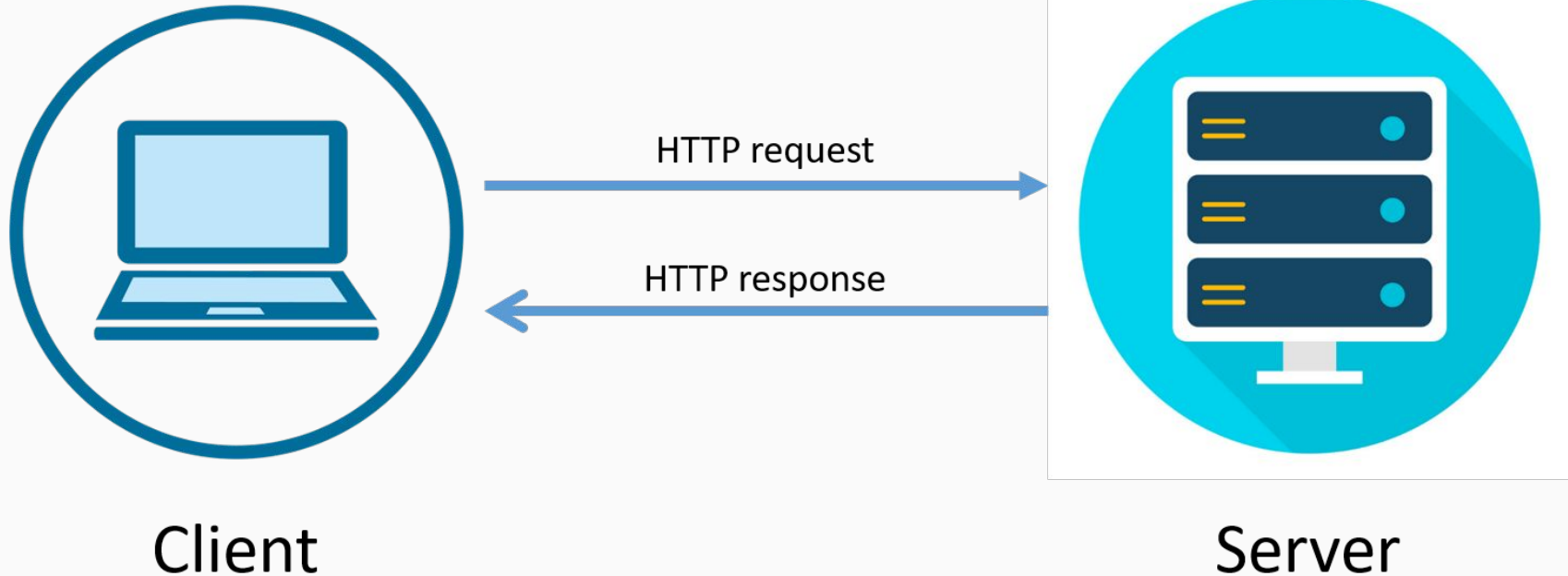
id	username	email	password
----	----------	-------	----------

```
class Model {
    protected string $table;
    public static function find($id) {
        $query = "SELECT * FROM $this->table";
        $query .= " WHERE id = $id";
        $result = $conn->query($query);
        $row = $result->fetch_assoc();
        $this->id = $row["id"];
        // ...
    }
}
```

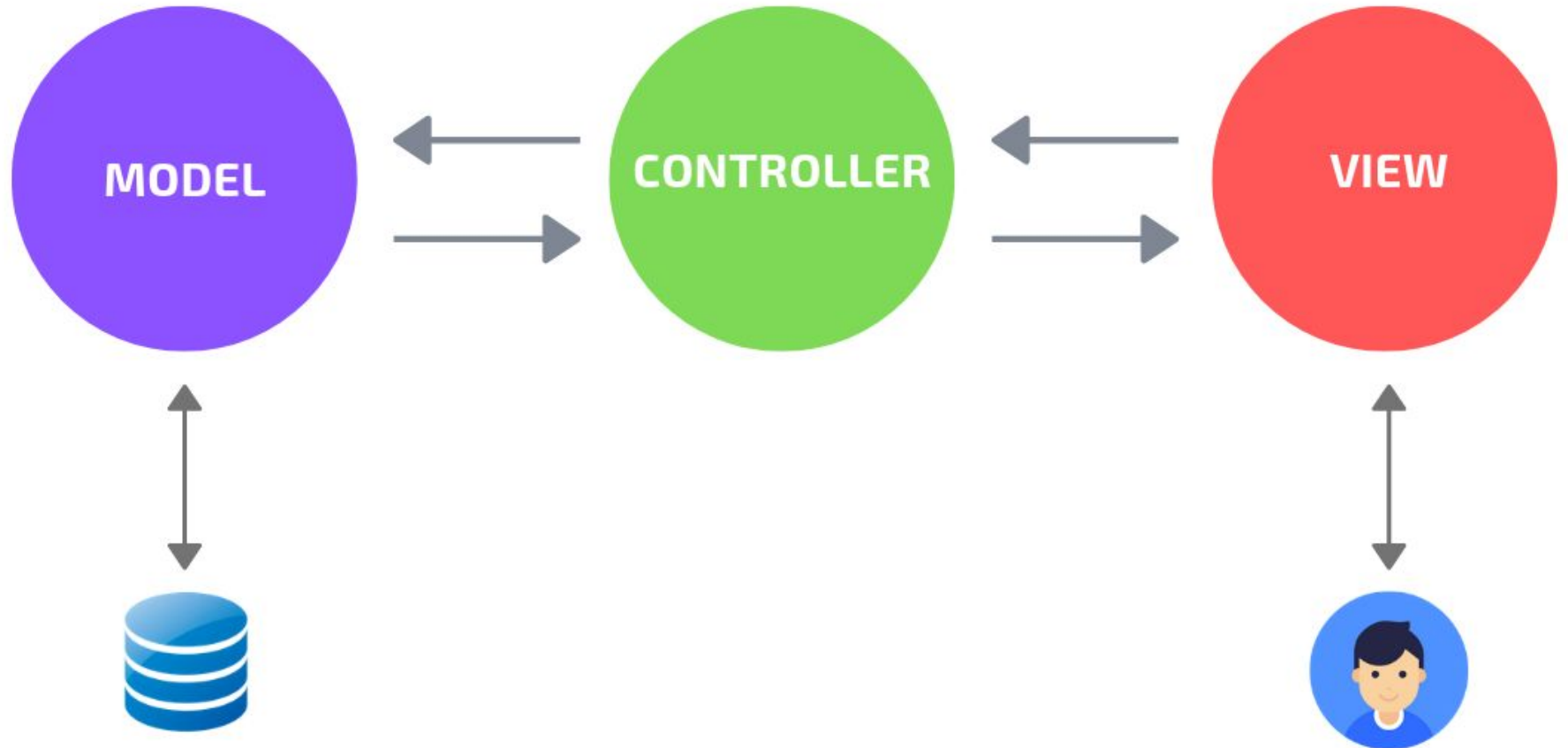
```
class User extends Model {
    protected string $table = "users";
}
```

id	username	email	password
1	ahmed	ahmed@gmail.com	\$2y\$10\$PNY.

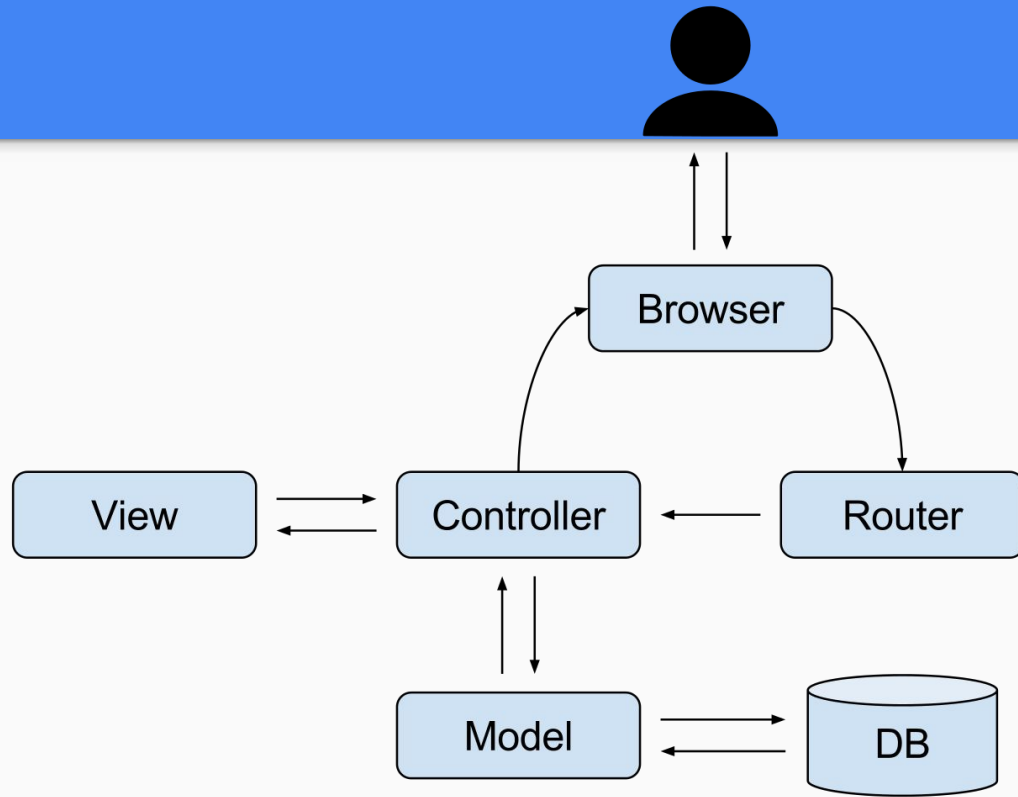
Request & Response



Request & Response



Request & Response

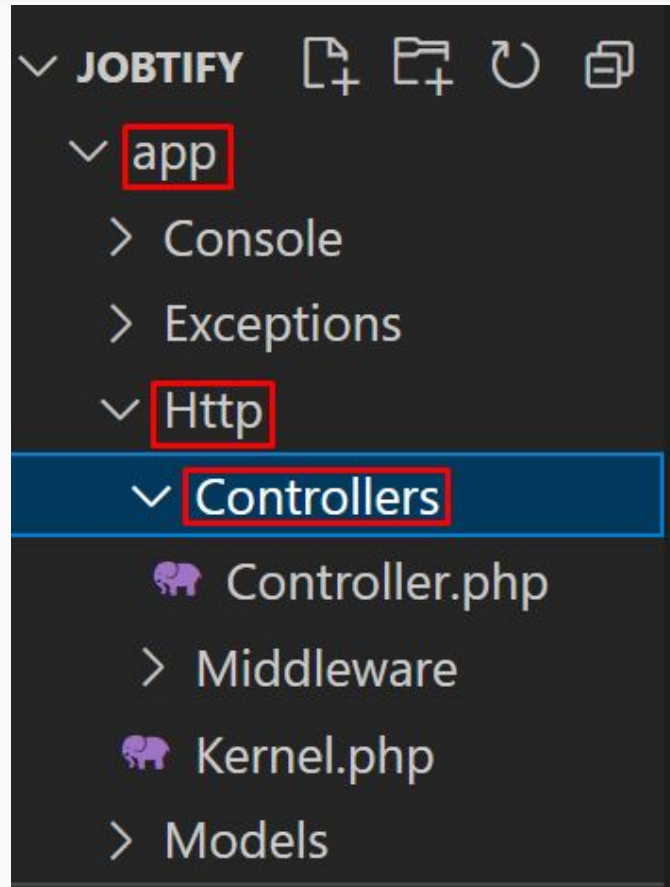


Install Laravel

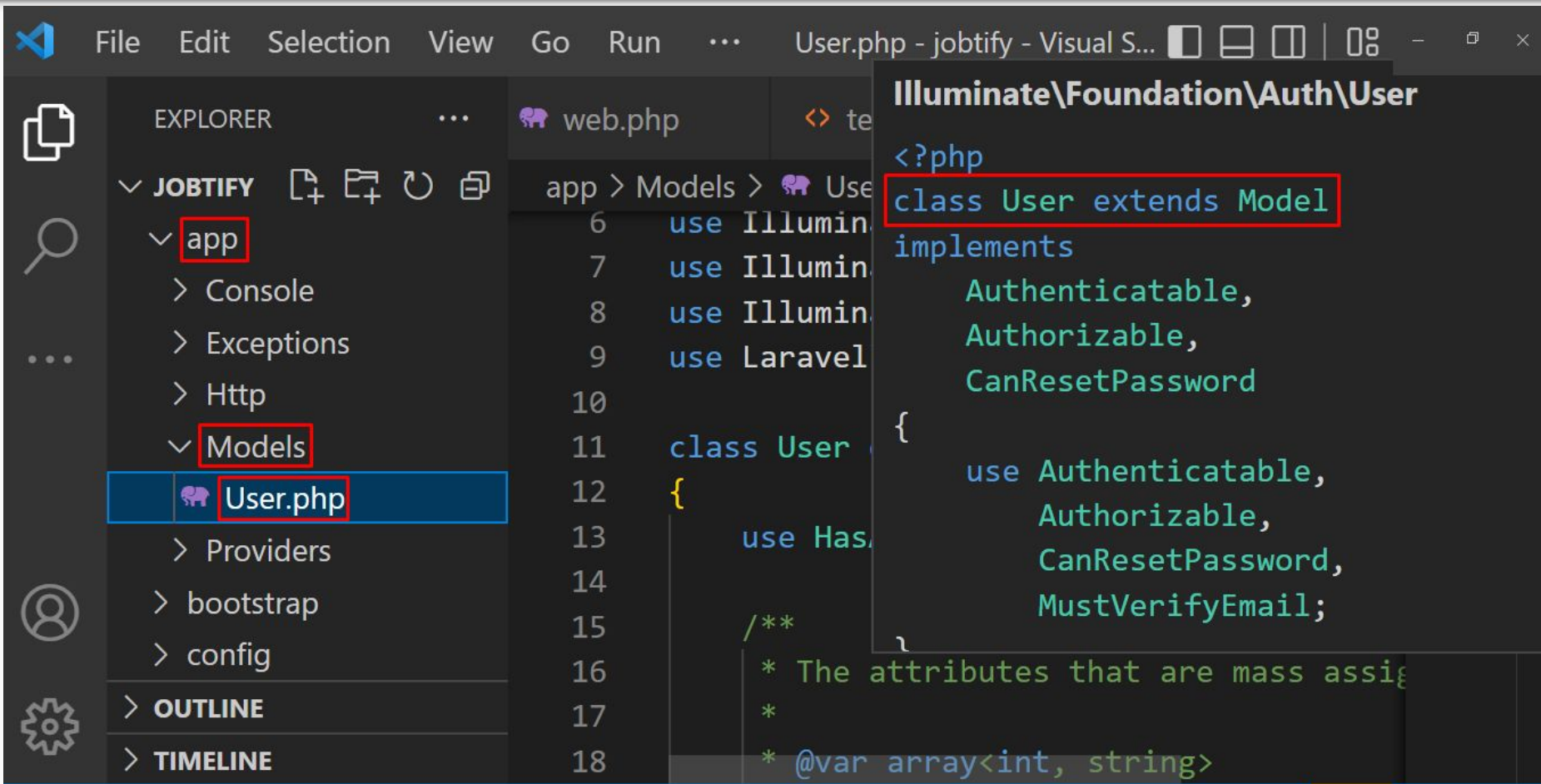
- Install [Composer](#)
- Open CMD or any terminal

```
C:\Users\zeyad>composer global require laravel/installer
Changed current directory to C:/Users/zeyad/AppData/Roaming/Composer
Using version ^4.2 for laravel/installer
./composer.json has been updated
Running composer update laravel/installer
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating autoload files
11 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
No security vulnerability advisories found

C:\Users\zeyad>laravel new myproject_
```

Laravel - Discover - Models



The screenshot shows the Visual Studio Code interface with the Laravel file explorer on the left and the User.php model file open in the editor. The file explorer shows the following structure:

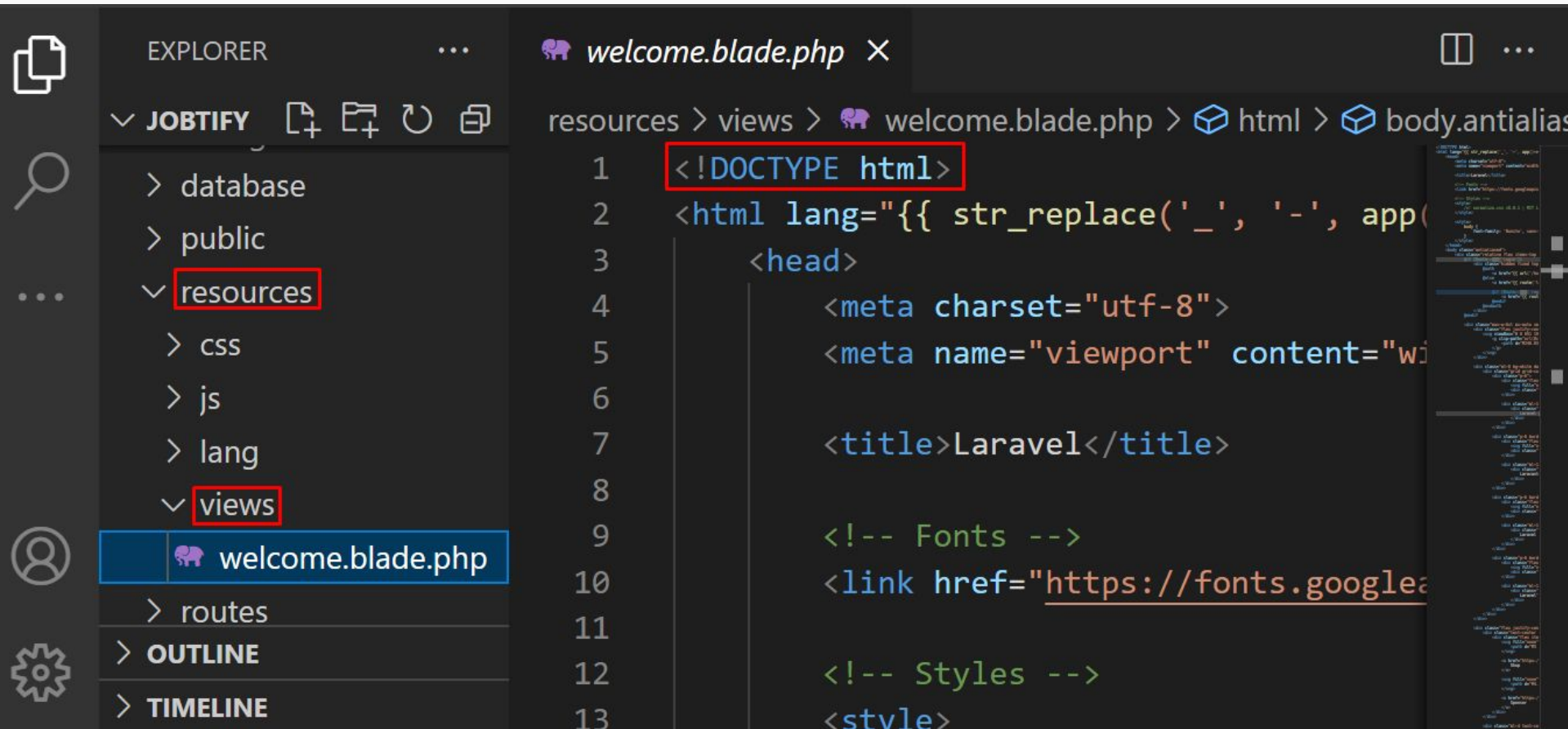
- JOBTIFY
 - app
 - Console
 - Exceptions
 - Http
 - Models
 - User.php
 - Providers
 - bootstrap
 - config
 - OUTLINE
 - TIMELINE

The User.php file is highlighted in the file explorer. The editor shows the following code:

```
web.php
app > Models > User.php
6 use Illuminate\Foundation\Auth\User as Authenticatable;
7 use Illuminate\Notifications\Notifiable;
8 use Illuminate\Contracts\Auth\CanResetPassword;
9 use Laravel\Passport\HasApiTokens;
10
11 class User extends Model
12 {
13     use HasApiTokens;
14
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var array<int, string>
```

The code in the editor is a PHP class definition for the User model, which extends the Illuminate\Foundation\Auth\User class. The class implements the Authenticatable, Authorizable, and CanResetPassword interfaces. It also uses the HasApiTokens trait. The class has a protected \$fillable array and a protected \$password attribute.

Laravel - Discover - Views



The image shows a code editor interface with a dark theme. On the left is the 'EXPLORER' sidebar. It contains a tree view of the project files. The 'resources' folder is expanded, and the 'views' subfolder is selected. The file 'welcome.blade.php' is highlighted. On the right, the code for 'welcome.blade.php' is displayed. The code is a Blade template for a Laravel application. It starts with a DOCTYPE declaration, followed by an HTML lang attribute that uses a Laravel helper function. The template includes a head section with meta tags for charset and viewport, a title tag, and comments for fonts and styles. The 'DOCTYPE html' tag on line 1 is highlighted with a red box. The breadcrumb navigation at the top of the editor shows the path: resources > views > welcome.blade.php > html > body.antialias.

EXPLORER

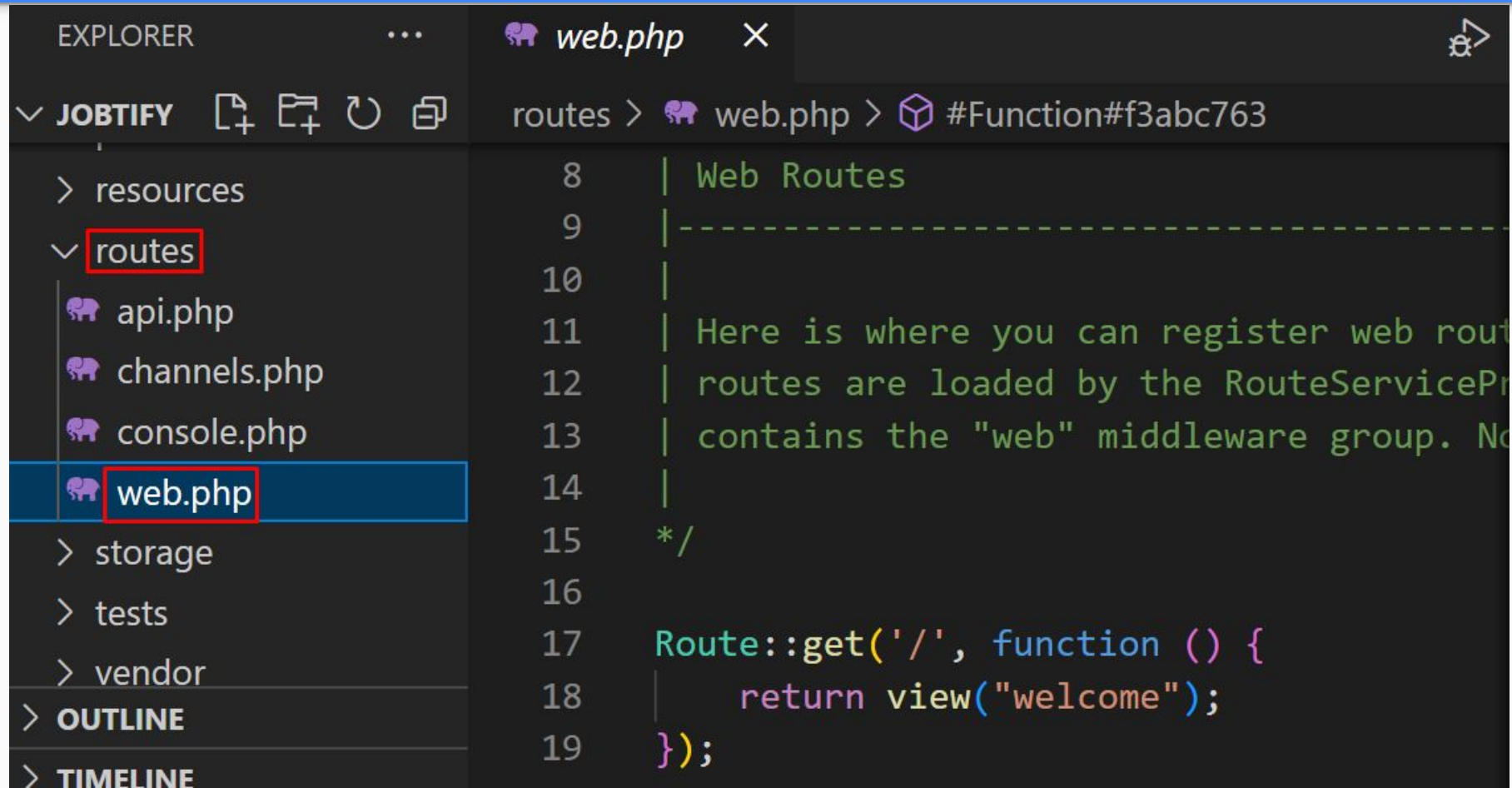
JOBTIFY

- database
- public
- resources
 - css
 - js
 - lang
 - views
 - welcome.blade.php
- routes
- OUTLINE
- TIMELINE

resources > views > welcome.blade.php > html > body.antialias

```
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}" >
3     <head>
4         <meta charset="utf-8">
5         <meta name="viewport" content="width=device-width, initial-scale=1">
6
7         <title>Laravel</title>
8
9         <!-- Fonts -->
10        <link href="https://fonts.googleapis.com/css?family=Roboto:400,700" rel="stylesheet">
11
12        <!-- Styles -->
13        <style>
```

Laravel - Discover - Router



The screenshot shows an IDE interface with a file explorer on the left and a code editor on the right. The file explorer, titled 'EXPLORER', shows a project named 'JOBIFY'. Under the 'resources' directory, the 'routes' folder is expanded, and 'web.php' is selected. The code editor shows the content of 'web.php', which includes a comment about web routes and a route definition for the root path.

EXPLORER

resources

routes

api.php

channels.php

console.php

web.php

storage

tests

vendor

OUTLINE

TIMELINE

web.php

routes > web.php > #Function#f3abc763

```
8 | Web Routes
9 | -----
10 |
11 | Here is where you can register web routes.
12 | routes are loaded by the RouteServiceProvider
13 | contains the "web" middleware group. No
14 |
15 | */
16 |
17 | Route::get('/', function () {
18 |     return view("welcome");
19 | });
```

Practical