Custom Auth Login and Registration

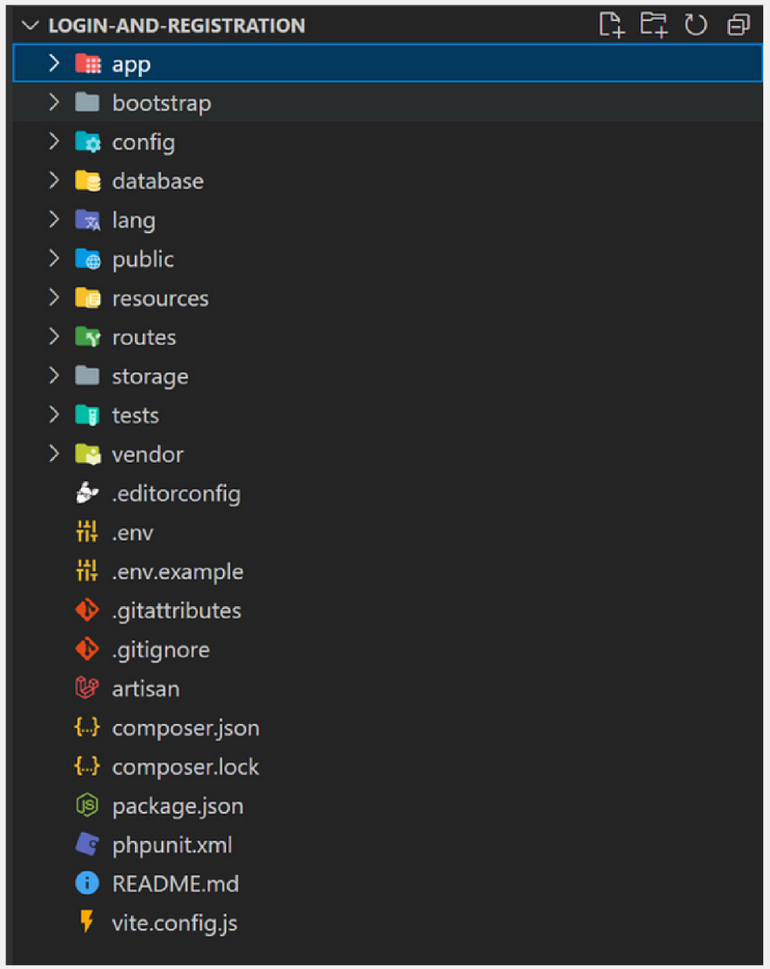
CUSTOM AUTH LOGIN AND REGISTRATION

- CREATE LARAVEL APP
- CONNECT TO DATABASE
- SET UP AUTH CONTROLLER
- CREATE AUTH ROUTES
- CREATE AUTH BLADE VIEW FILES
- RUN LARAVEL DEVELOPMENT SERVER

Step 1:

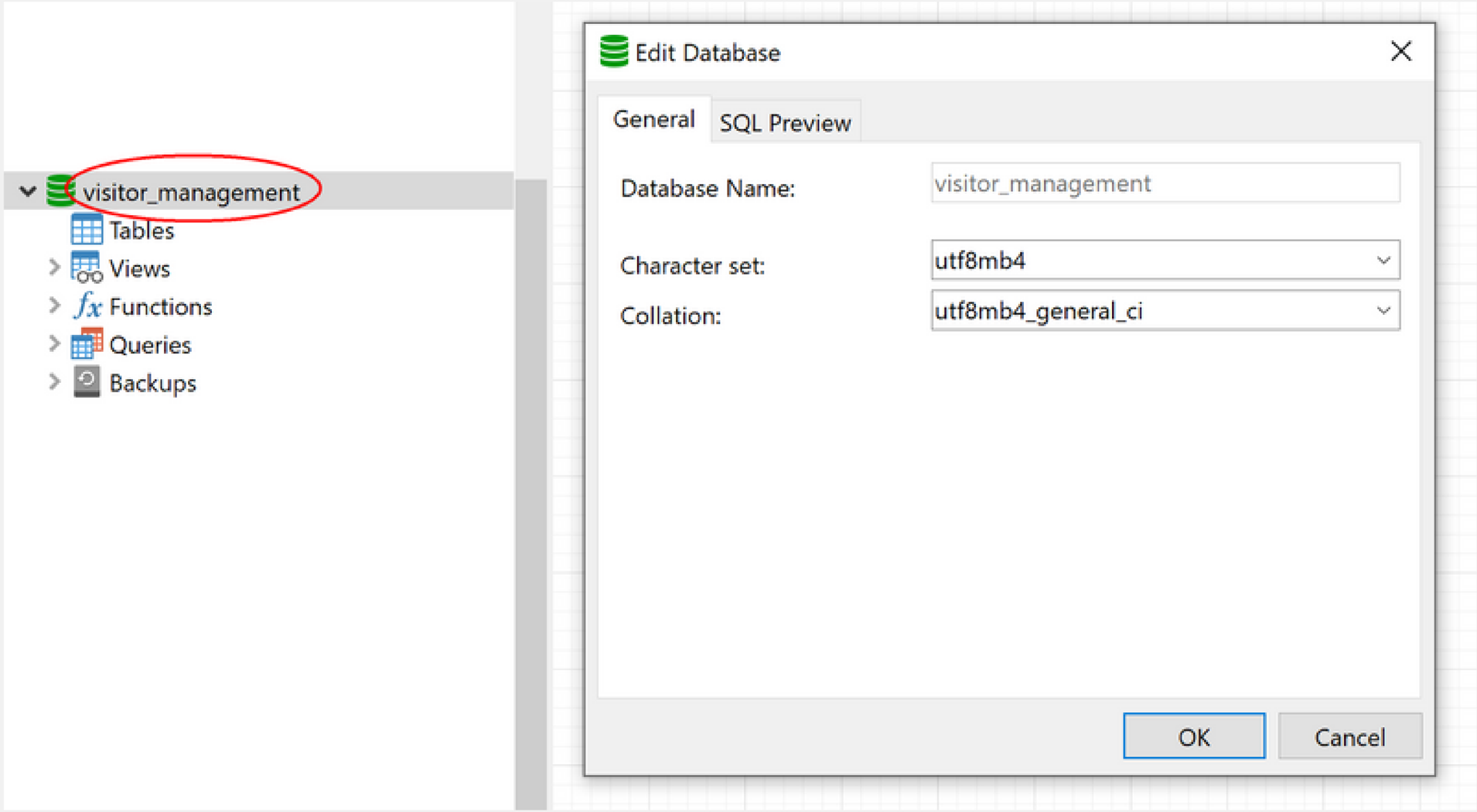
CREATE LARAVEL APP

composer create-project --prefer-dist laravel/laravel *login-and-registration*

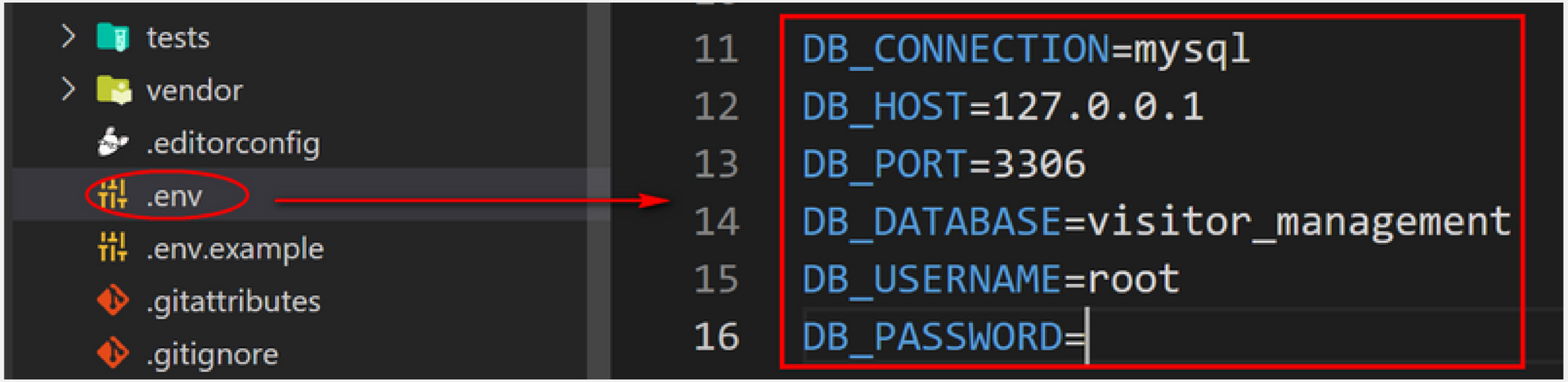


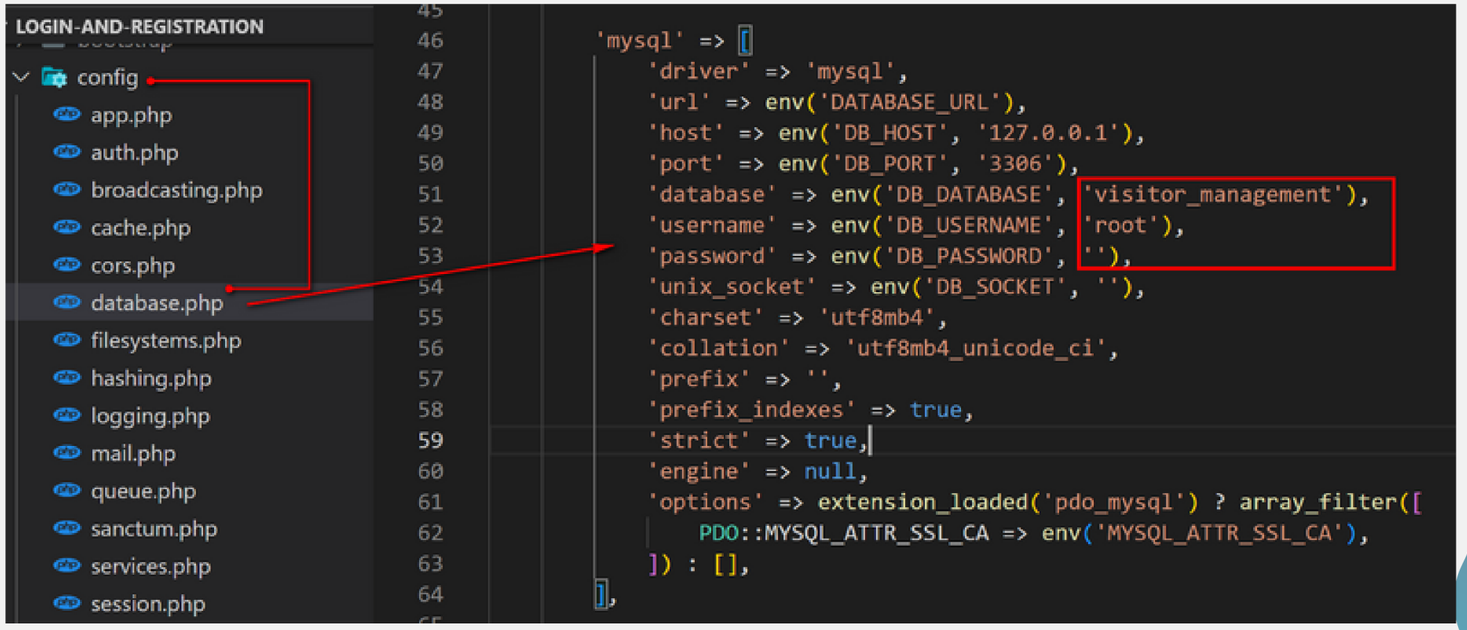
Step 2:

CONNECT TO DATABASE

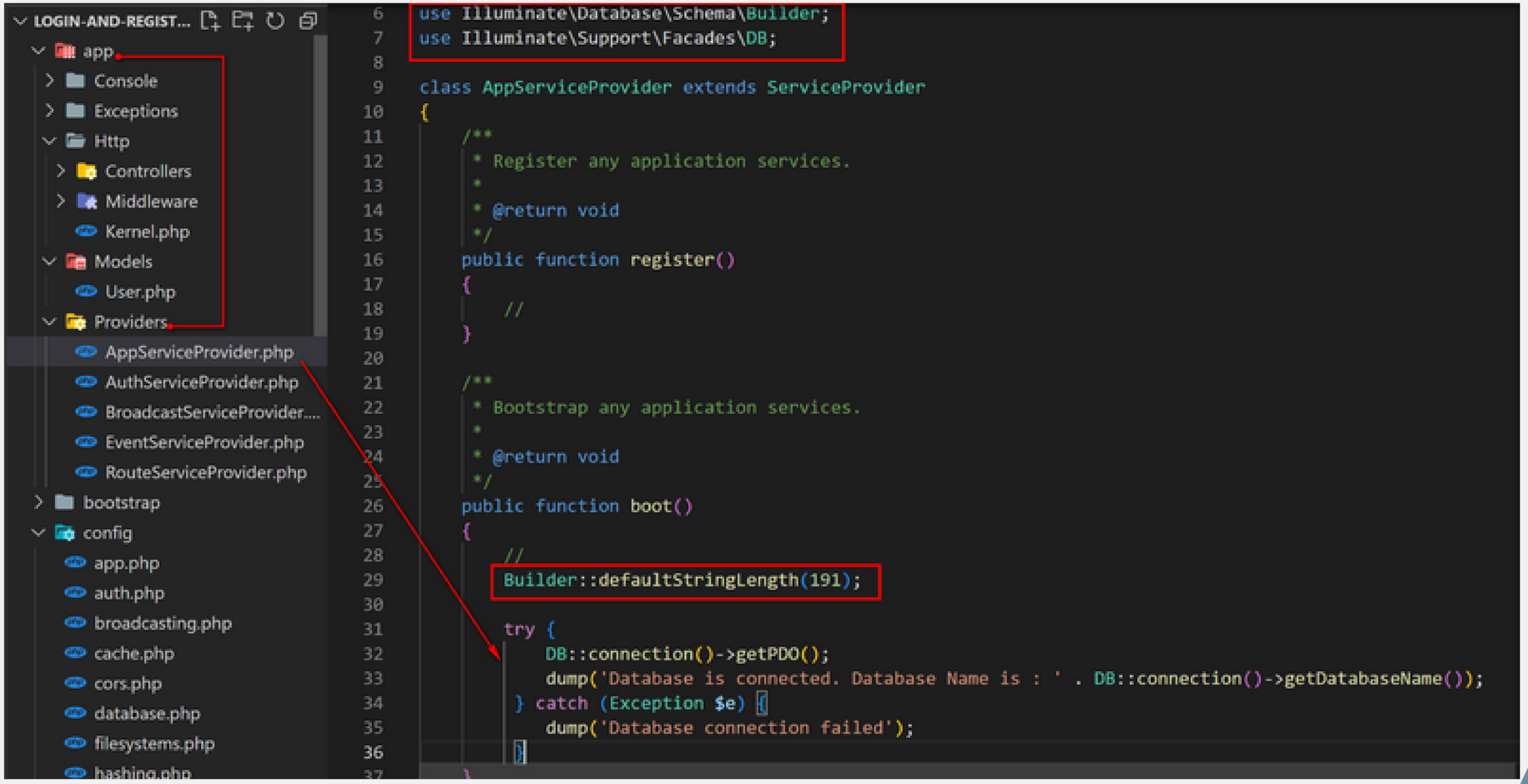


Now, you have to add database **name**, **username**, and **password** into the .env configuration file to connect the laravel app to the database:



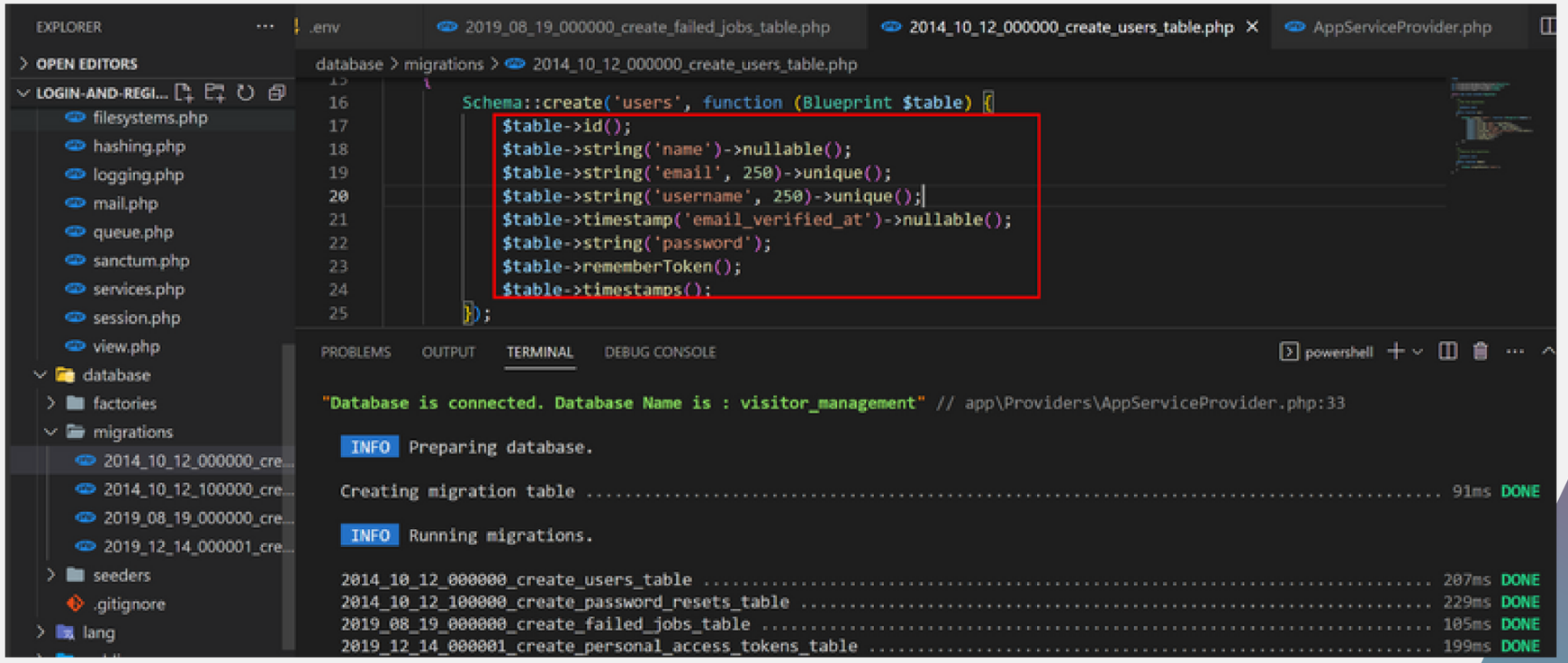


Check your connection database

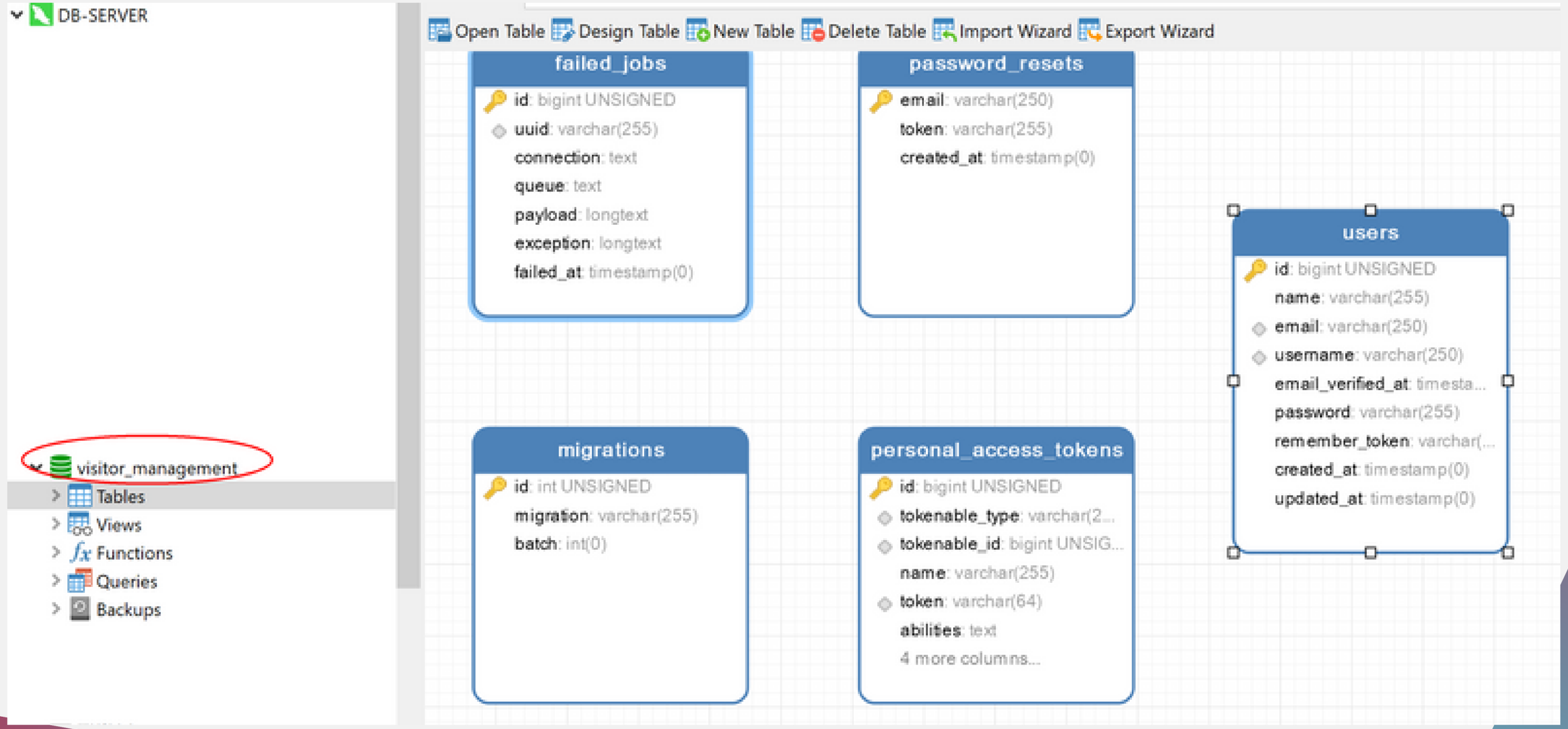


The Laravel default comes with a User model and migration files. But before running the migrate command we need to update your user table migration which can be found here > {project_folder}\database\migrations\2014_10_12_000000_create_users_table.php and add the username field then update the **name** field to nullable so that in registration we require the **email**, **username**, and **password**. See the updated migration code below.

php artisan migrate

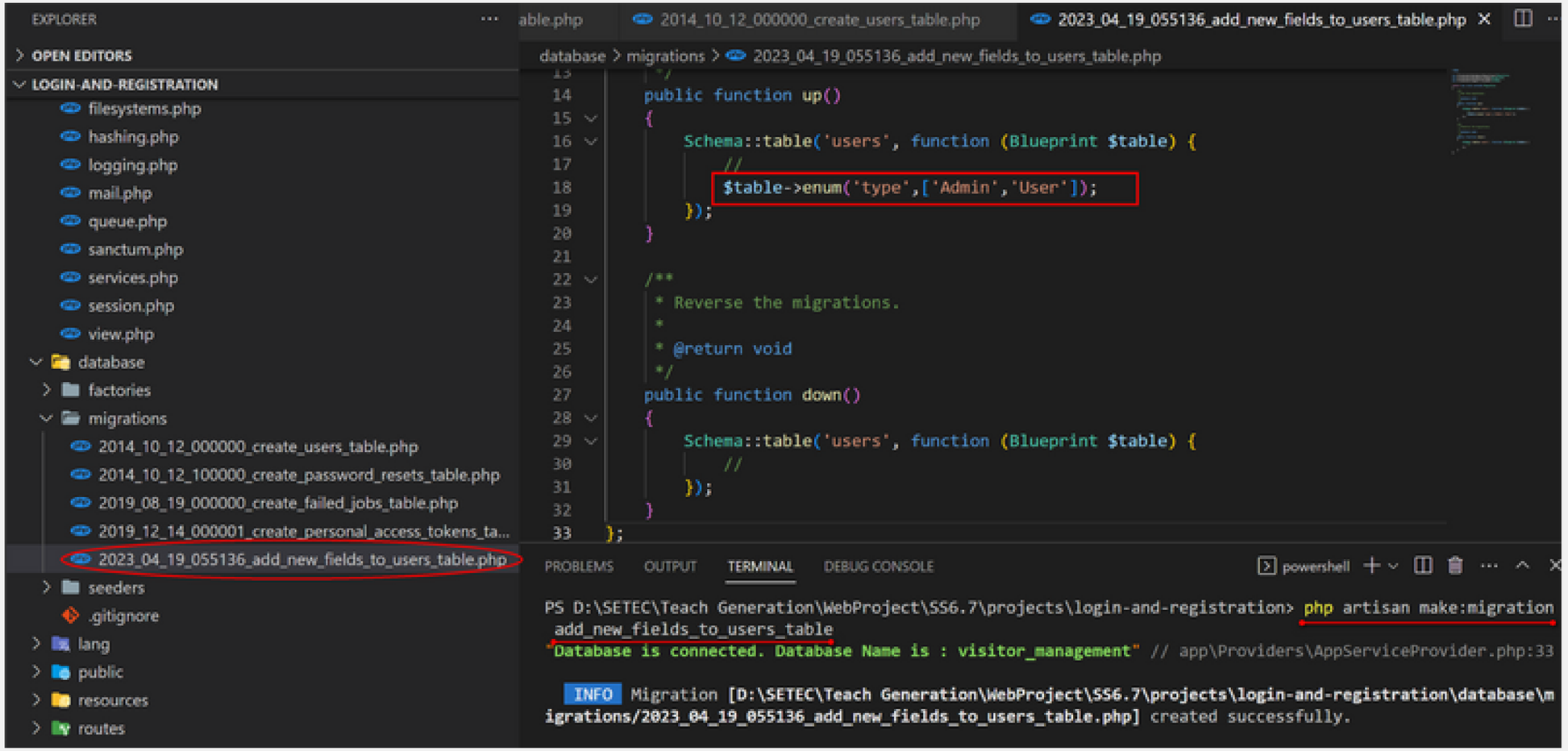


Go to check your tables

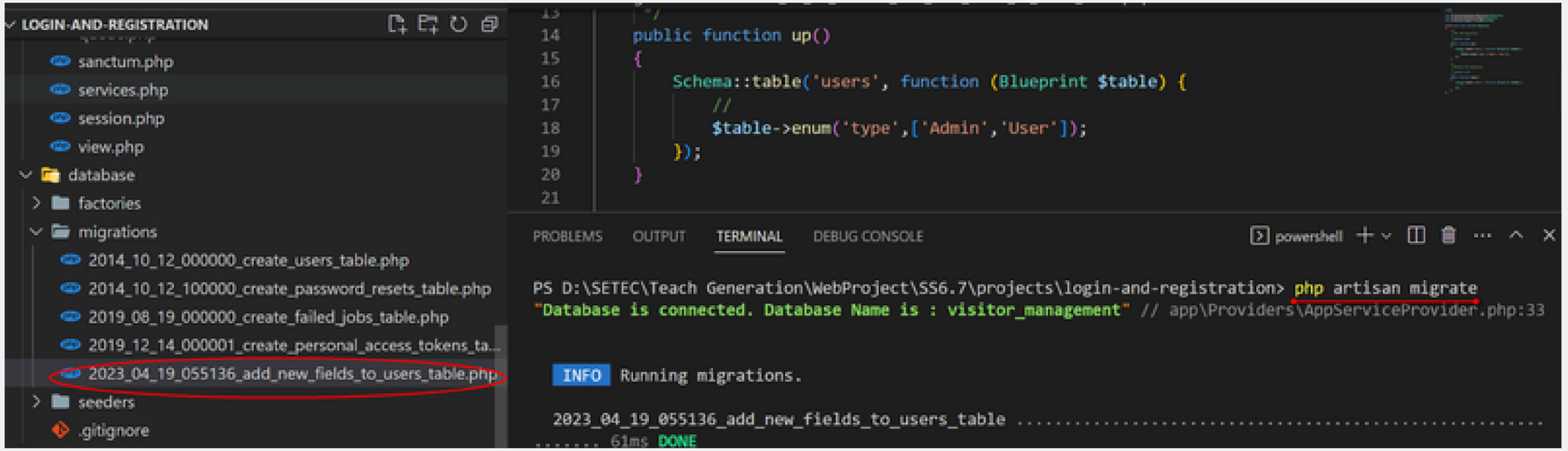


if you want add more fields to **users** table

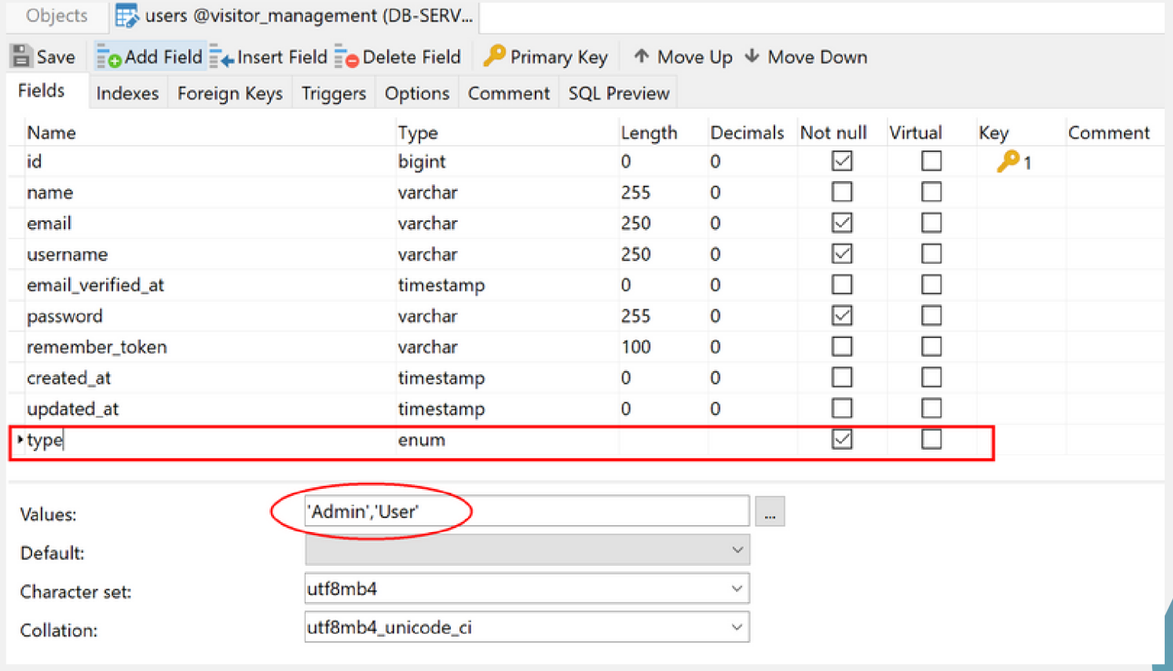
php artisan make:migration add_new_fields_to_users_table



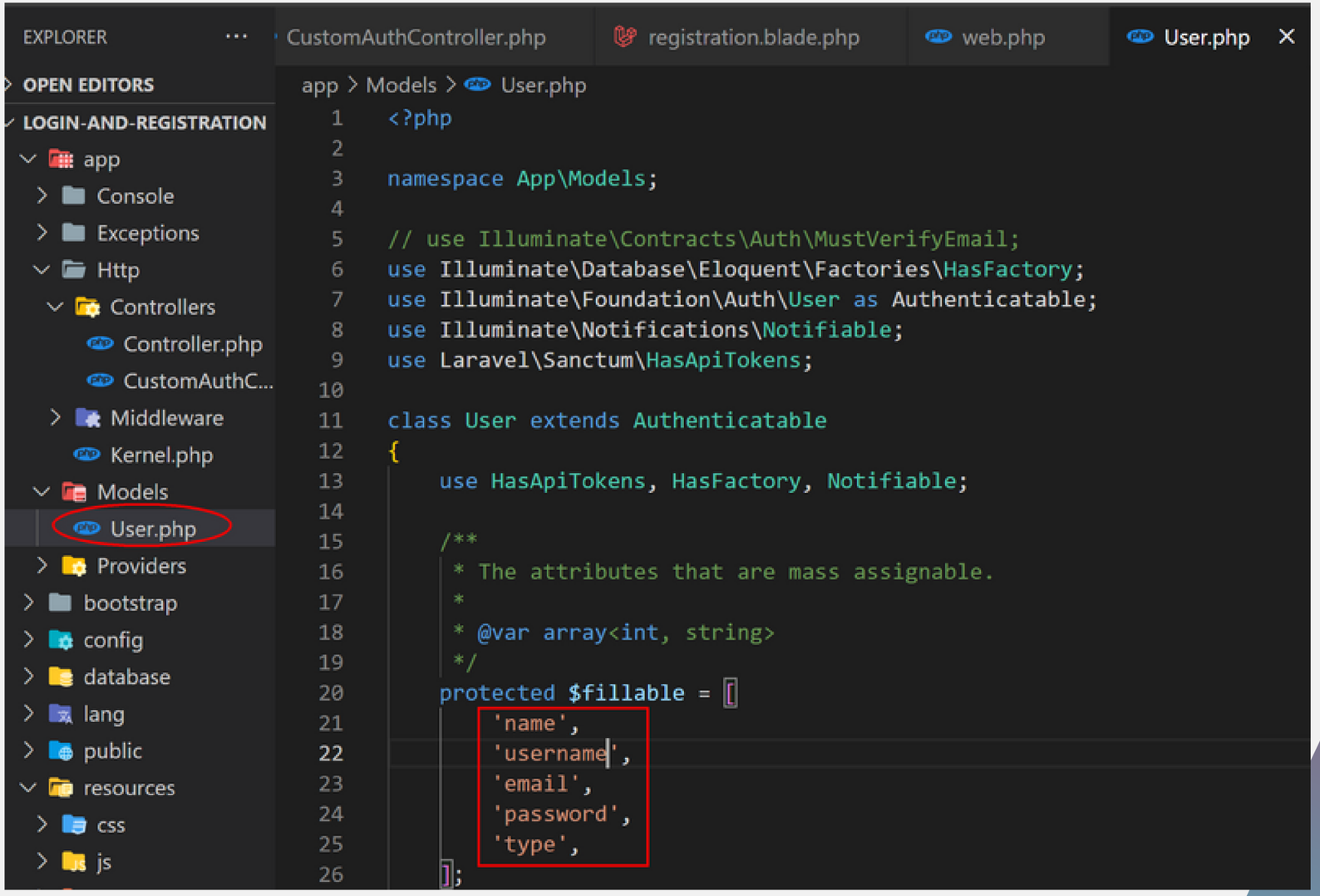
php artisan migrate



Go to check **users** table



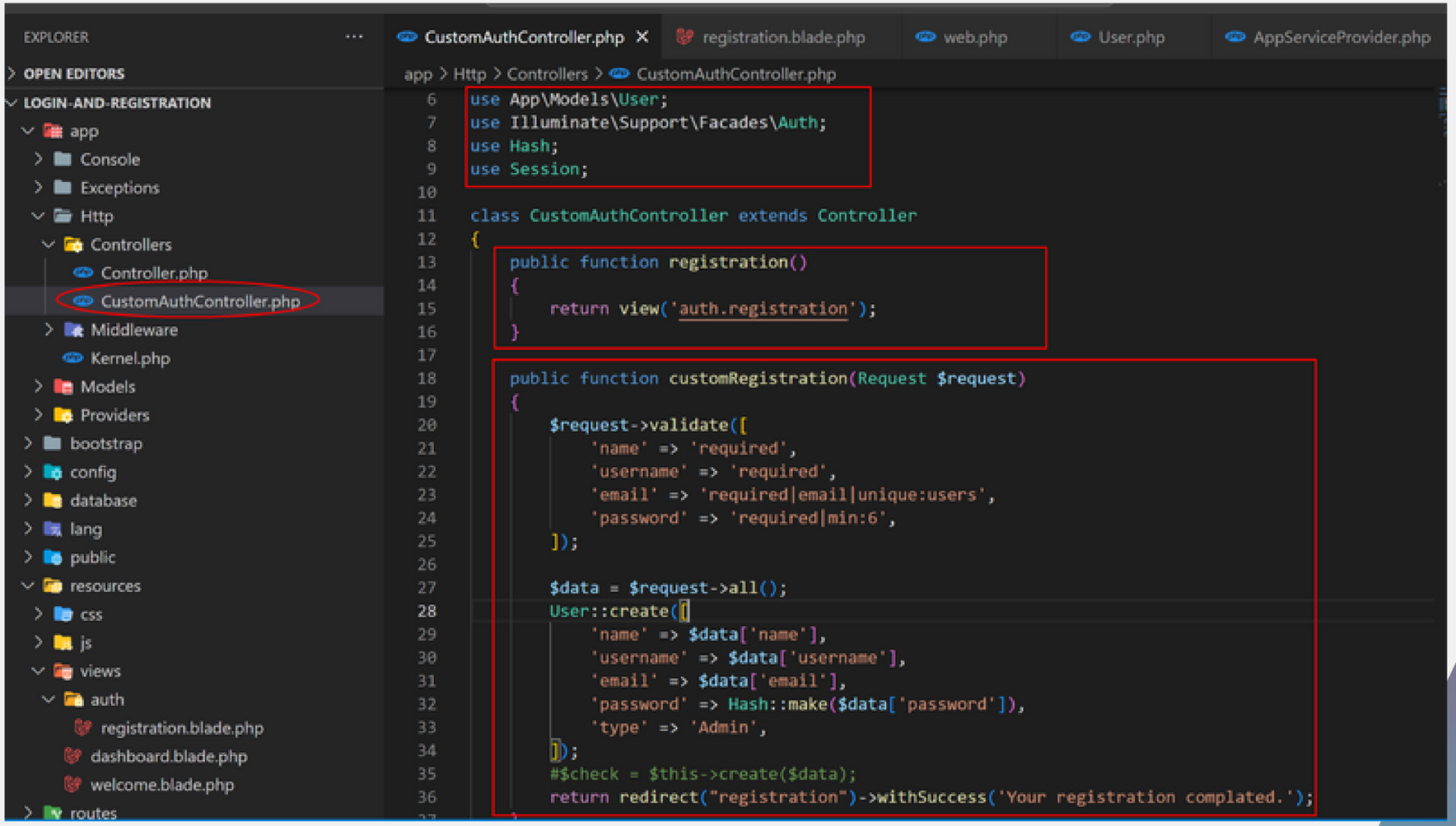
Go to update User model



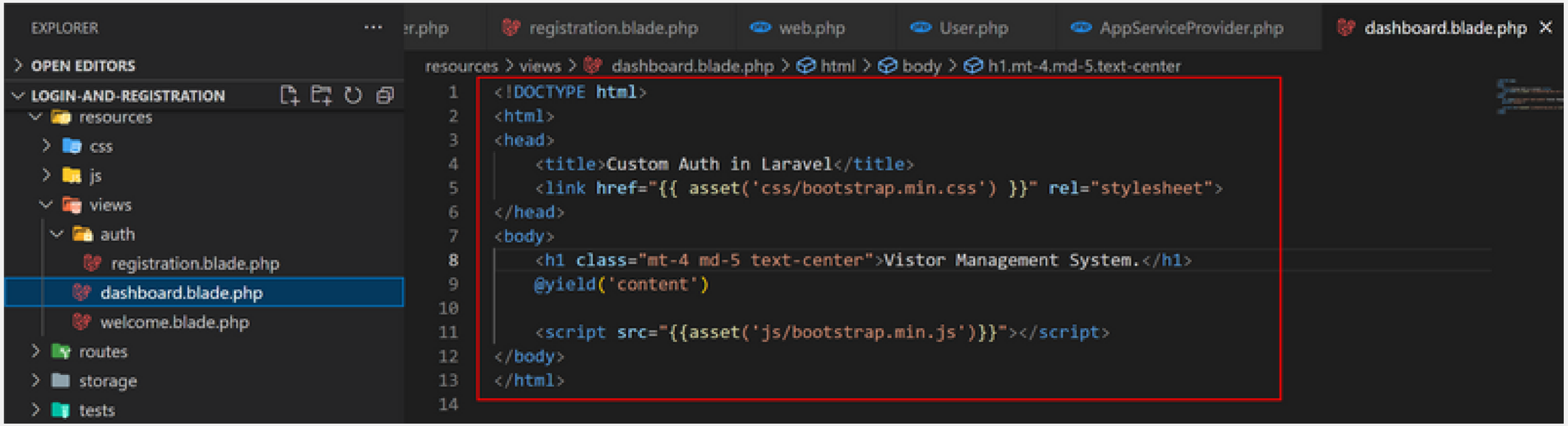
Step 3:

SET UP AUTH CONTROLLER

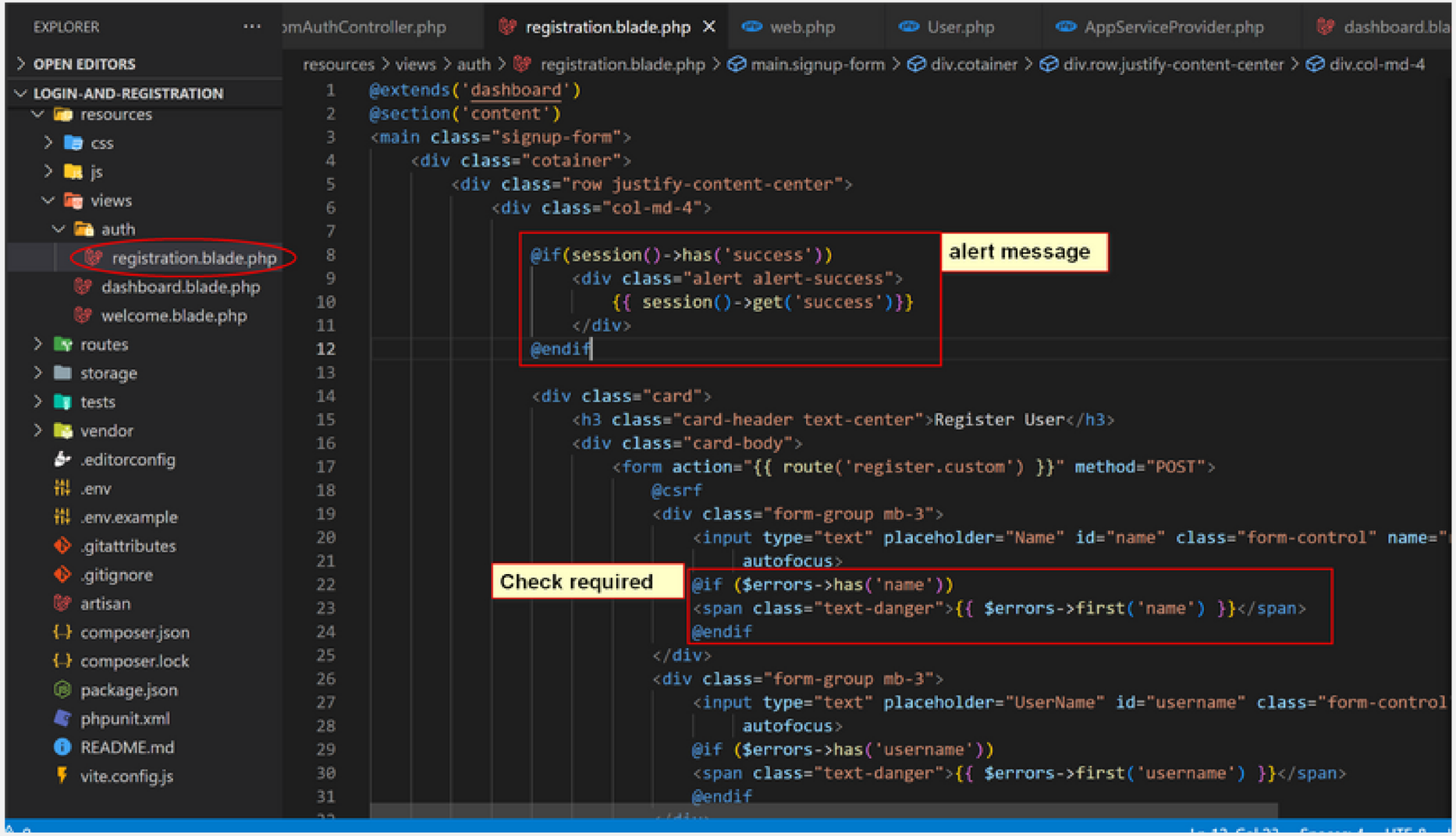
php artisan make:controller CustomAuthController



Create *dashboard.blade.php* in view file



Create *registration.blade.php* in auth file



Create routes

php artisan serve

Fill data all fields

Vistor Management System.

Your registration complated.

Register User

Meas Dara

Dara

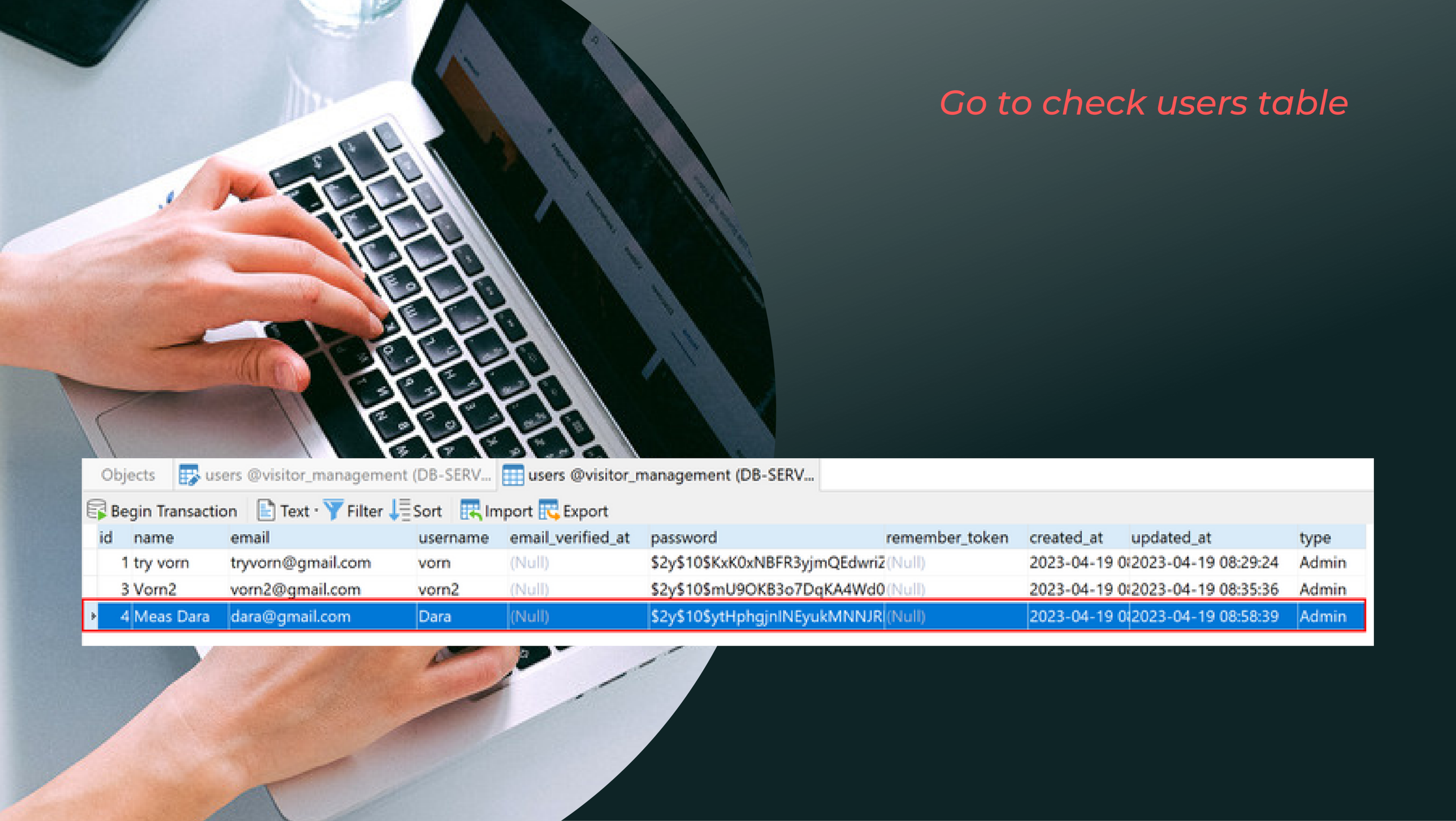
dara@gmail.com

••••••••••••••

☐ Remember Me

Sign up

Go to check users table



Objects | users @visitor_management (DB-SERV... | users @visitor_management (DB-SERV...

Begin Transaction | Text | Filter | Sort | Import | Export

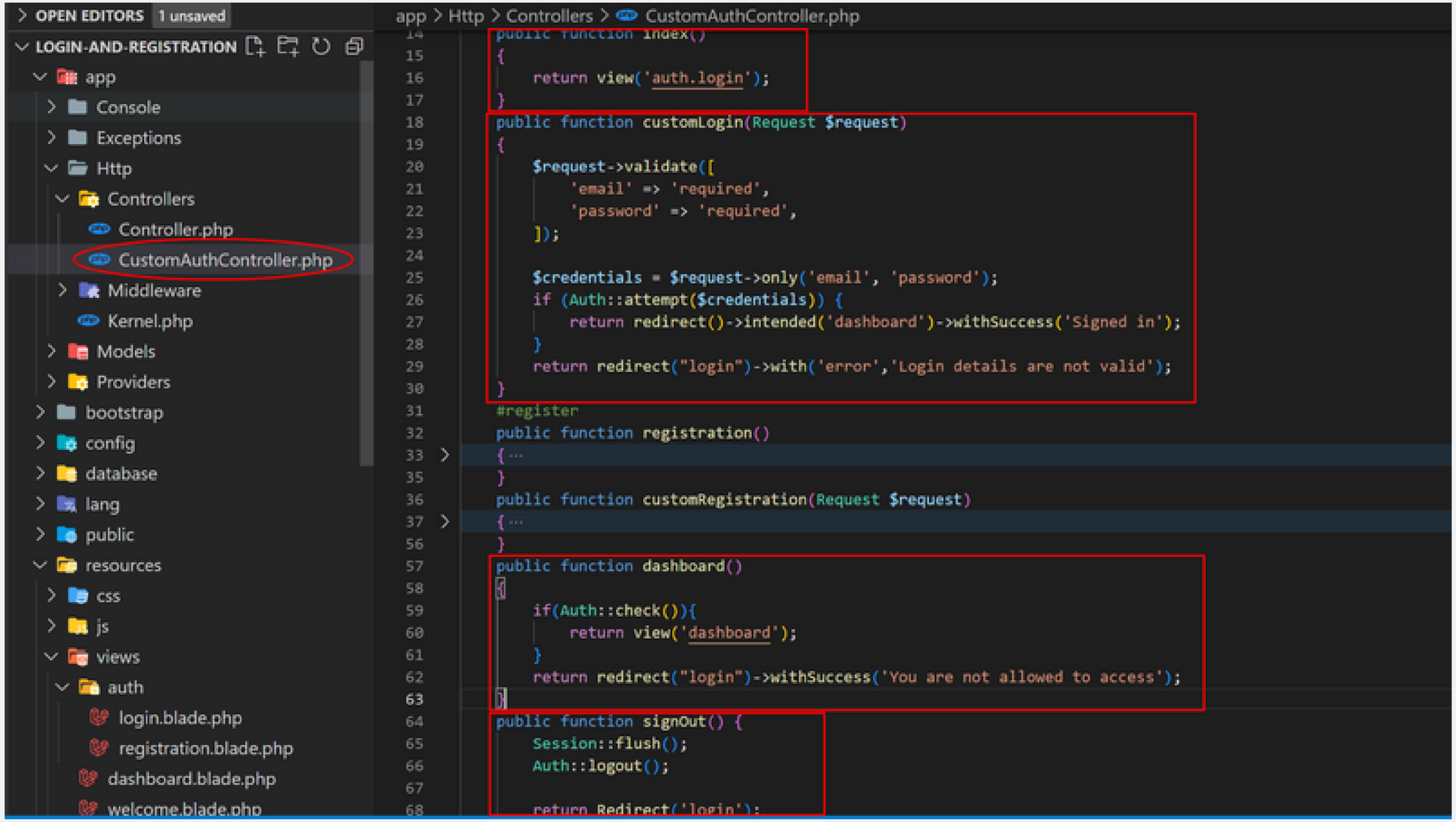
id	name	email	username	email_verified_at	password	remember_token	created_at	updated_at	type
1	try vorn	tryvorn@gmail.com	vorn	(Null)	$2y$10$KxK0xNBFR3yjmQEdwri2	(Null)	2023-04-19 0	2023-04-19 08:29:24	Admin
3	Vorn2	vorn2@gmail.com	vorn2	(Null)	$2y$10$mU9OKB3o7DqKA4Wd0	(Null)	2023-04-19 0	2023-04-19 08:35:36	Admin
4	Meas Dara	dara@gmail.com	Dara	(Null)	$2y$10$ytHphgjnINEyukMNNJR	(Null)	2023-04-19 0	2023-04-19 08:58:39	Admin

CUSTOM LOGIN

- MAKE LOGIN PAGE
- CREATE METHOD FOR LOGIN PAGE
- CREATE LOGIN PAGE WITH DISPLAY VALIDATE ERROR
- CREATE METHOD FOR VALIDATE LOGIN CREDENTIALS
- DISPLAY INVALID LOGIN DETAILS ERROR MESSAGE
- DISPLAY DIFFERENT CONTENT TO GUST AND LOGIN USER
- SET ROUTE FOR CONTROLLER METHOD
- CHECK OUTPUT IN YOUR BROWSER

MAKE LOGIN PAGE

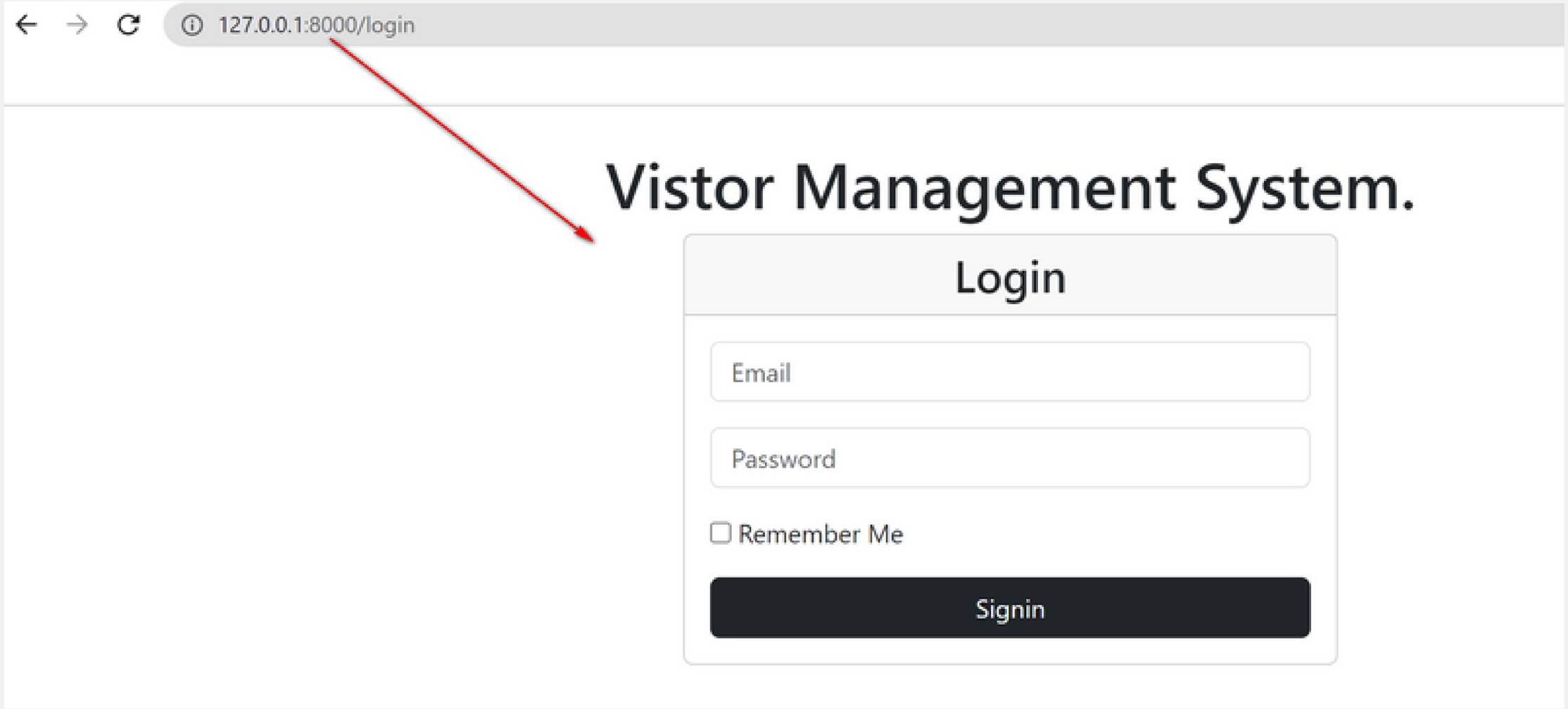
CREATE METHOD FOR LOGIN PAGE



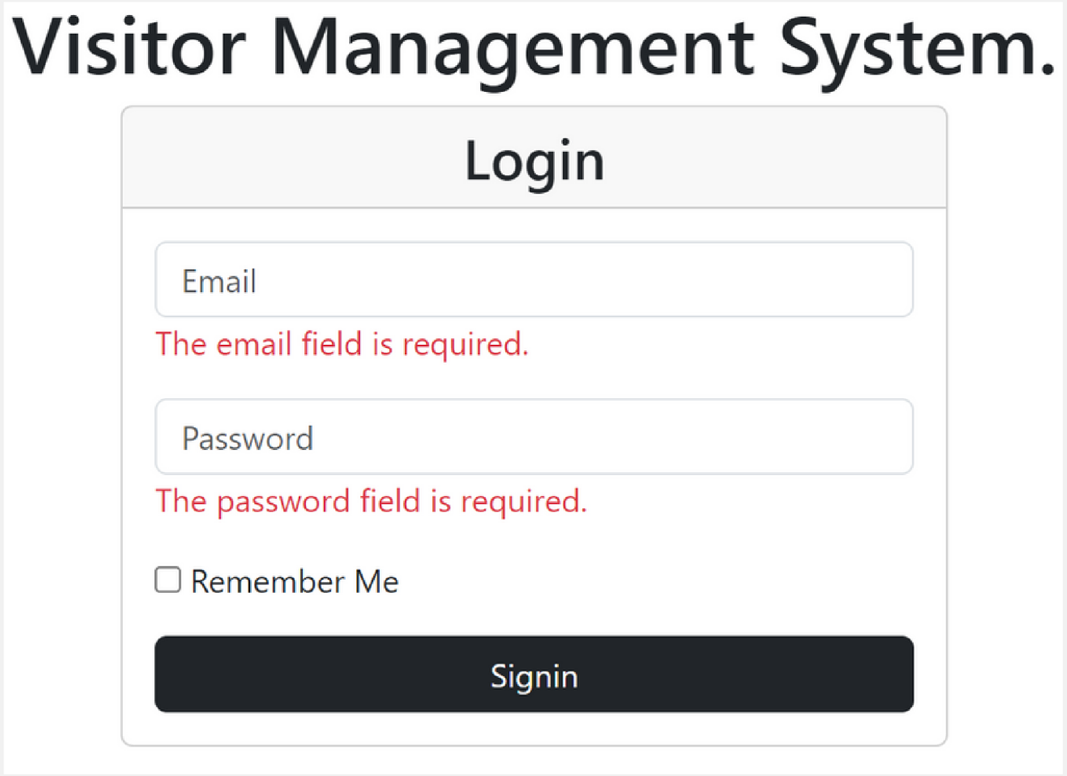
CUSTOM DASHBOARD PAGE

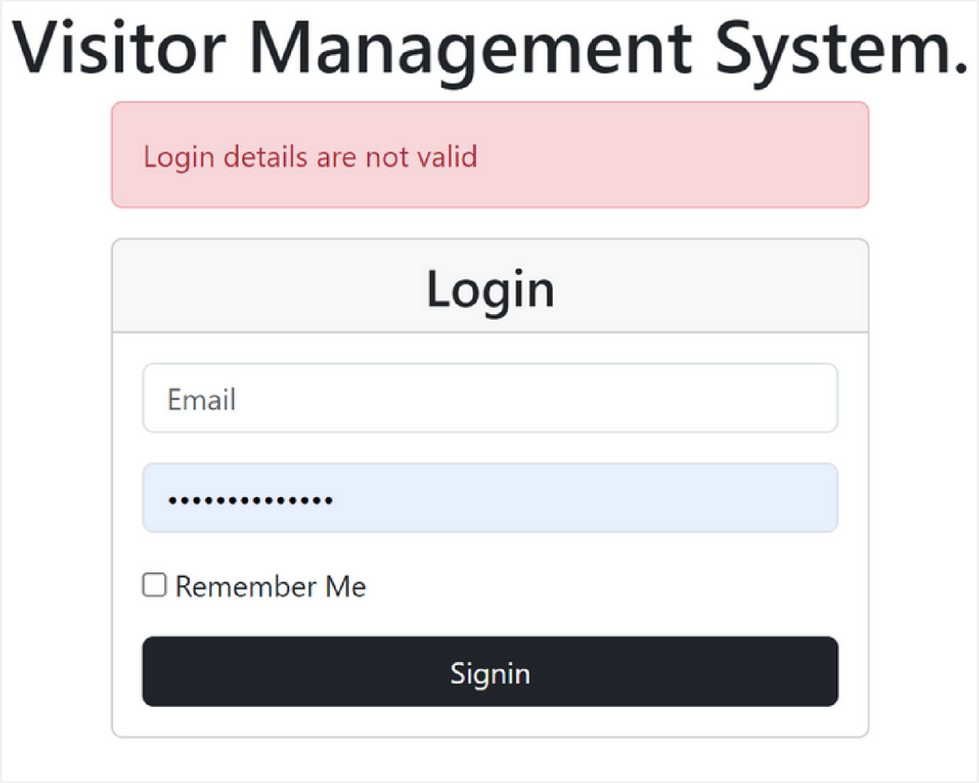
SET ROUTE FOR CONTROLLER METHOD

CHECK OUTPUT IN YOUR BROWSER

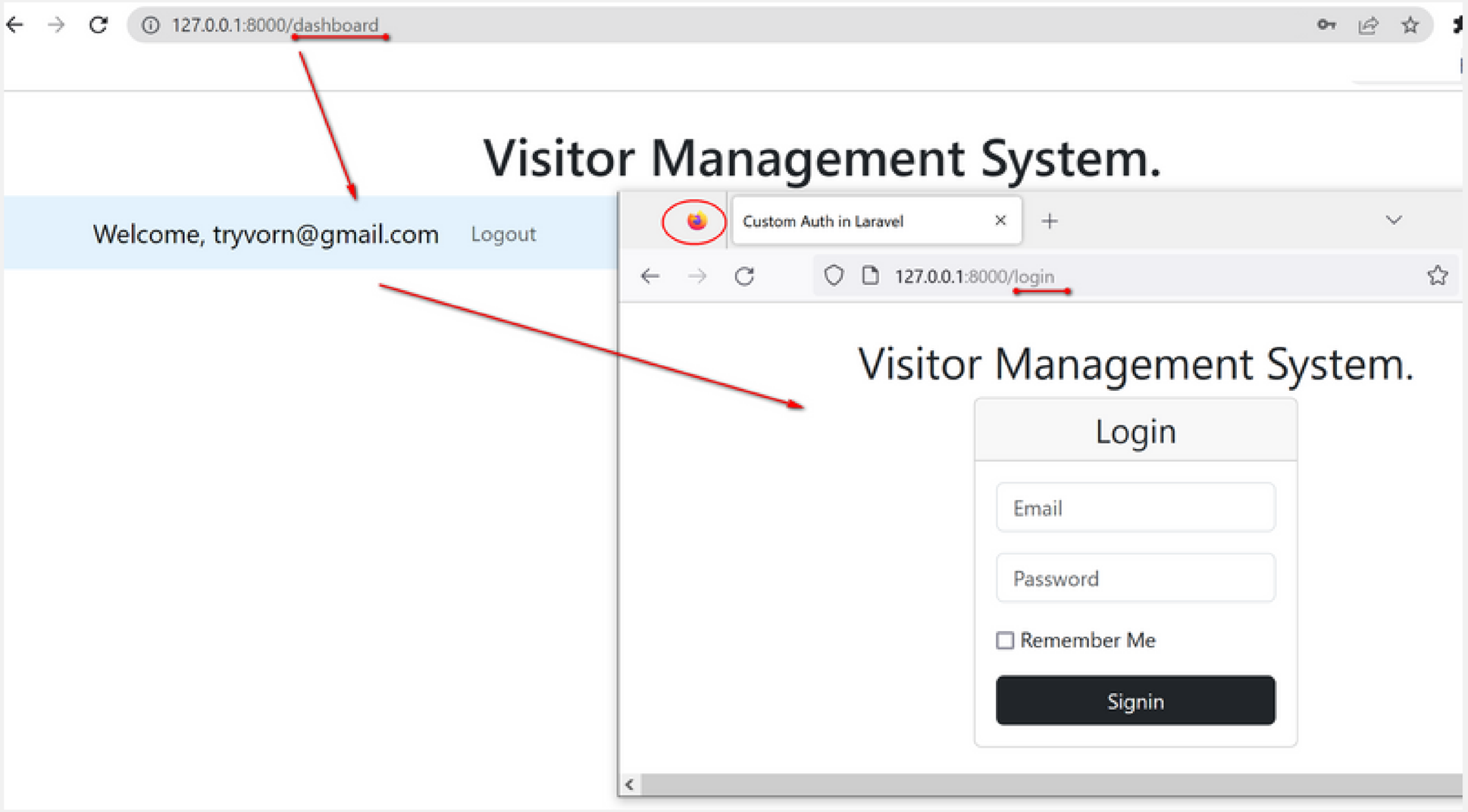


DISPLAY INVALID LOGIN DETAILS ERROR MESSAGE





DISPLAY DIFFERENT CONTENT TO GUST AND LOGIN USER



FETCH SINGLE USER DATA & CREATE PROFILE PAG

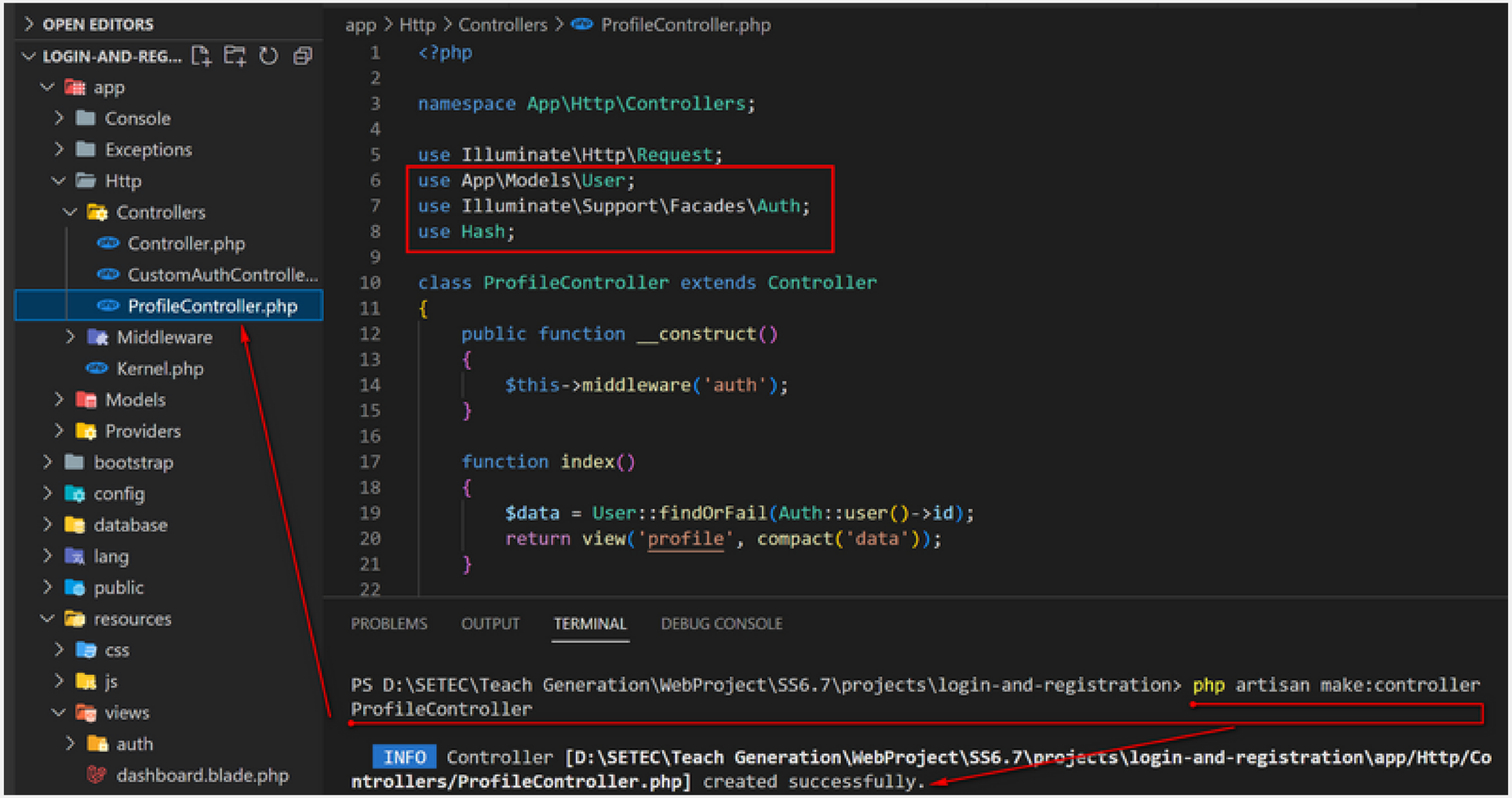
- CREATE PROFILE CONTROLLER METHOD
- PREVENT GUEST USER TO ACCESS PROFILE PAGE
- CREATE INDEX() METHOD FOR LOAD PROFILE PAGE
- CREATE PROFILE FORM WITH FILL DATA
- DISPLAY SUCCESS MESSAGE ON THE WEB PAGE
- SET ROUTE FOR CONTROLLER METHOD
- CHECK OUTPUT IN YOUR BROWSER

CREATE PROFILE CONTROLLER METHOD

Update dashboard.blade.php file

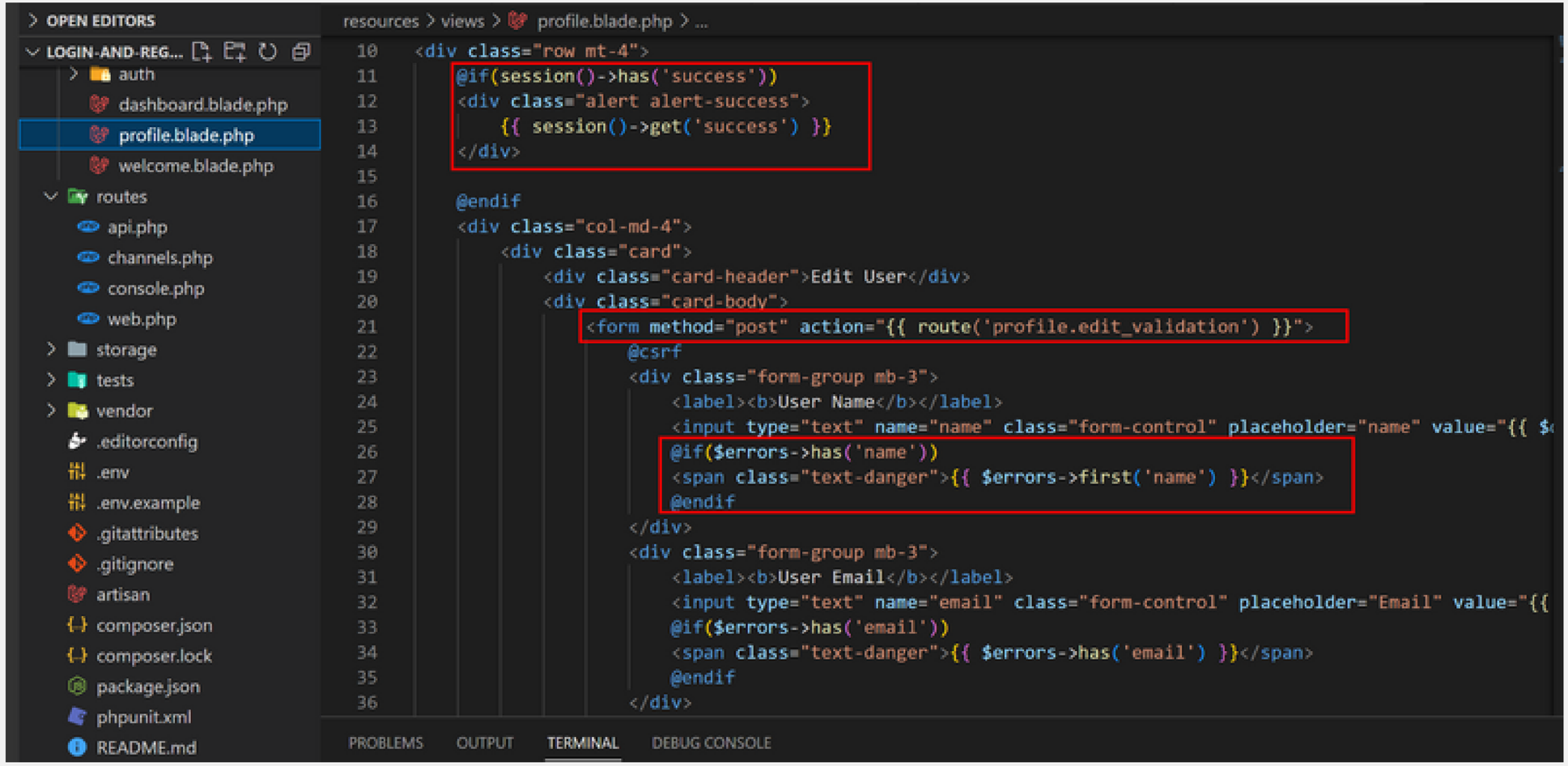
CREATE PROFILE CONTROLLER METHOD (COUNT...)

using command: *php artisan make:controller ProfileController*



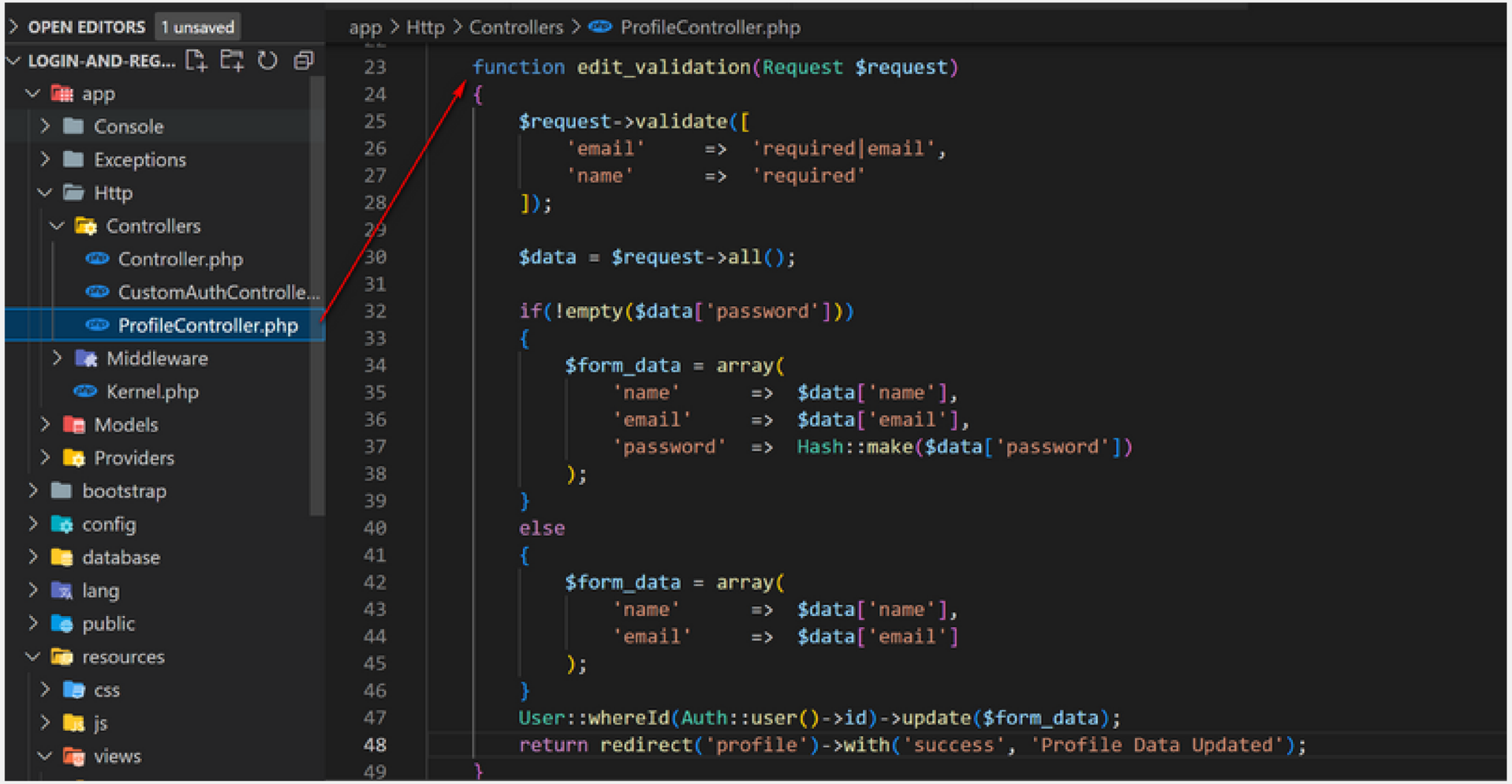
CREATE PROFILE CONTROLLER METHOD (COUNT...)

- Create *profile.blade.php*
- Set validation & message



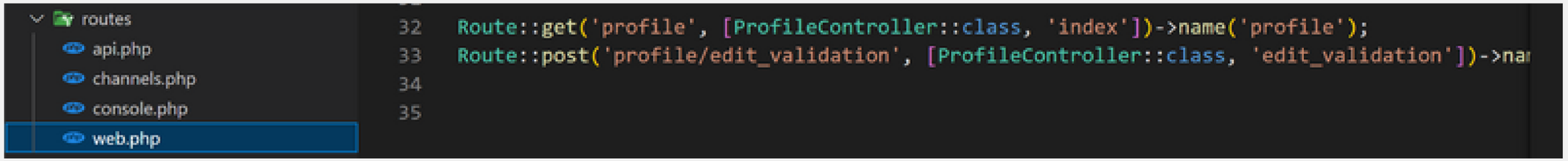
CREATE PROFILE CONTROLLER METHOD (COUNT...)

Back to *ProfileController* to set validation



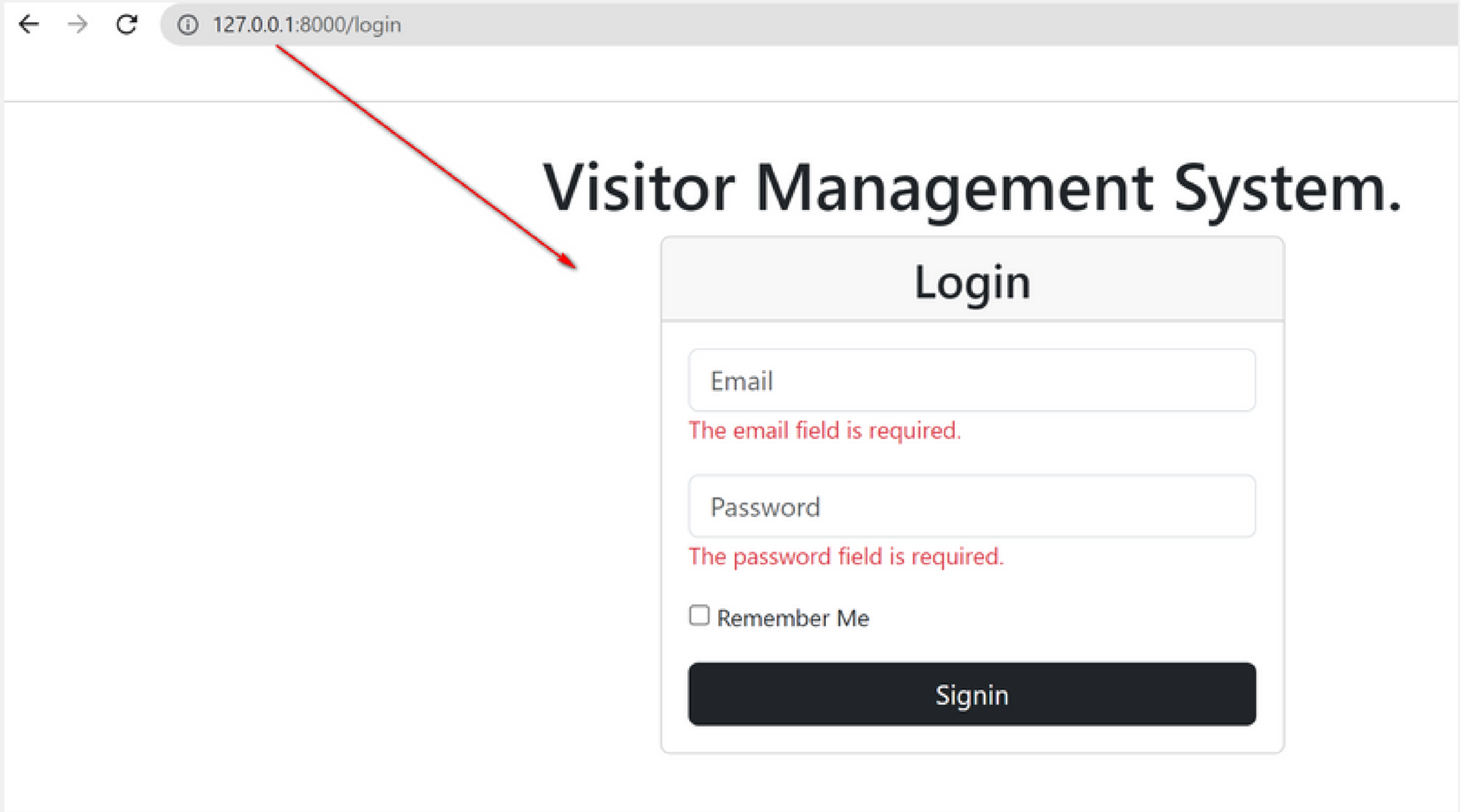
CREATE PROFILE CONTROLLER METHOD (COUNT...)

set route for controller method

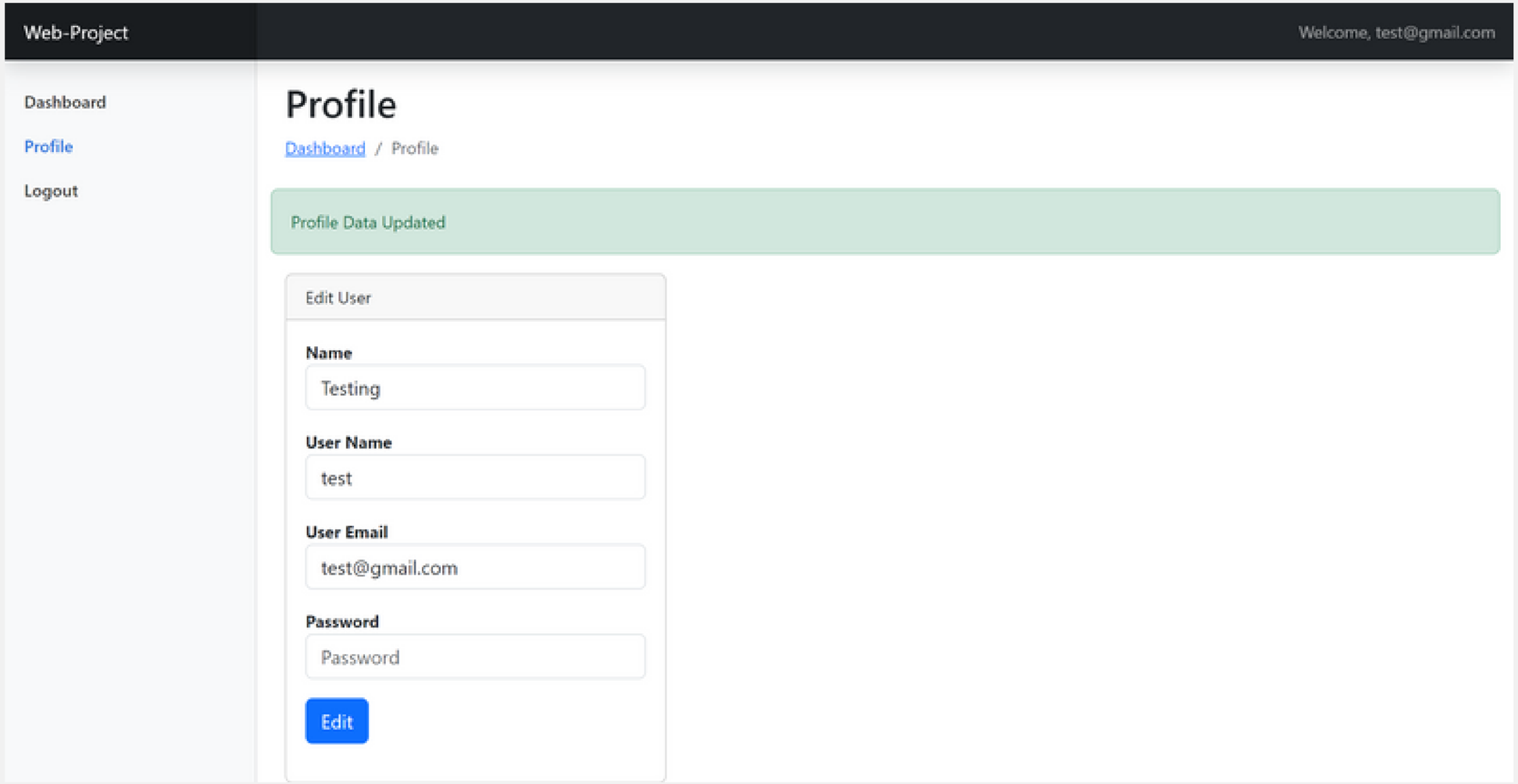


Run command: *php artisan serve*

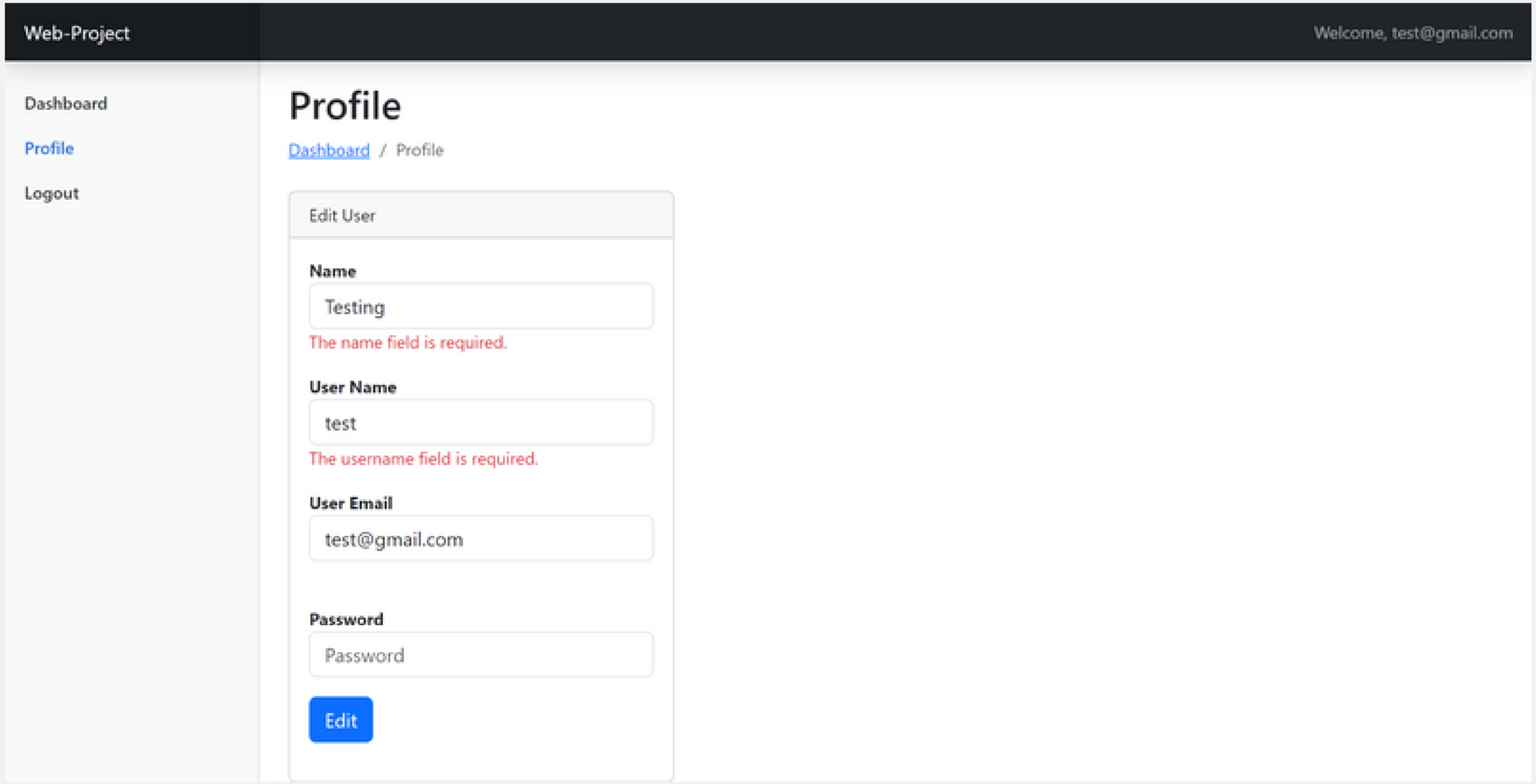
CHECK OUTPUT IN YOUR BROWSER



CHECK OUTPUT IN YOUR BROWSER (COUNT...)



CHECK OUTPUT IN YOUR BROWSER (COUNT...)

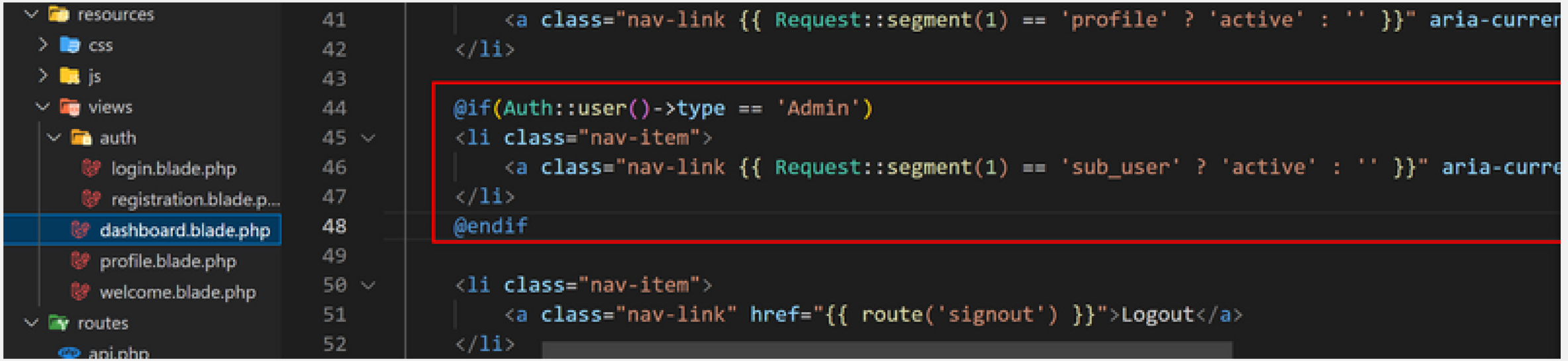


LOAD USER DATA IN YAJRA DATATABLES - 7

- CREATE SUB USER CONTROLLER METHOD
- DOWNLOAD & INSTALL YAJRA DATATABLE PACKAGE
- PREVENT ACCESS TO GUEST USER
- CREATE INDEX() METHOD
- CREATE VIEW BLADE FILE WITH JQUERY CODE FOR DATATABLE
- WRITE PHP SCRIPT FOR FETCH DATA FROM USER TABLE
- SET ROUTE FOR CONTROLLER METHOD
- CHECK OUTPUT IN YOUR BROWSER

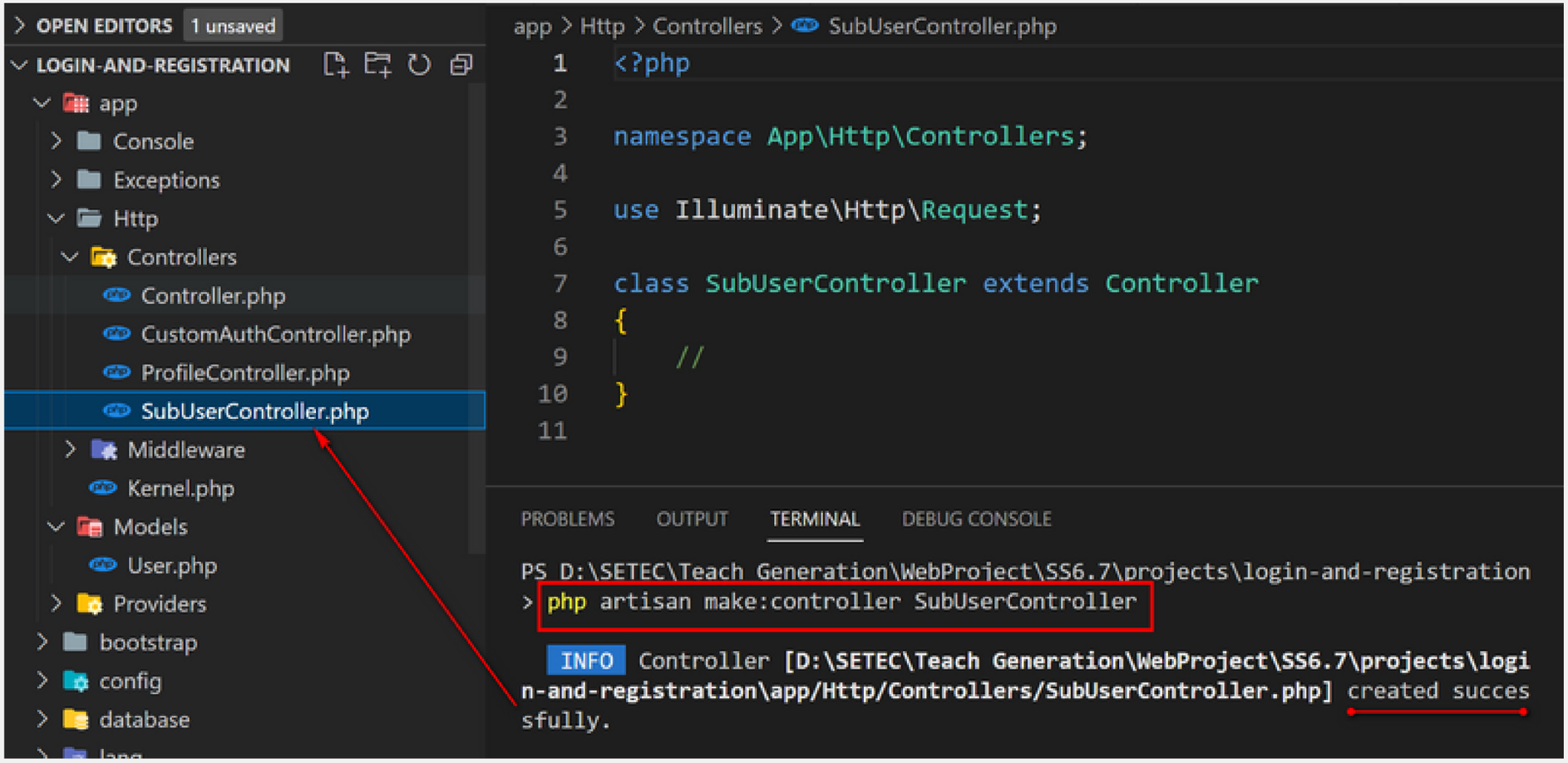
ADD SUB_USER LINK

Go to add in *dashboard.blade.php* file



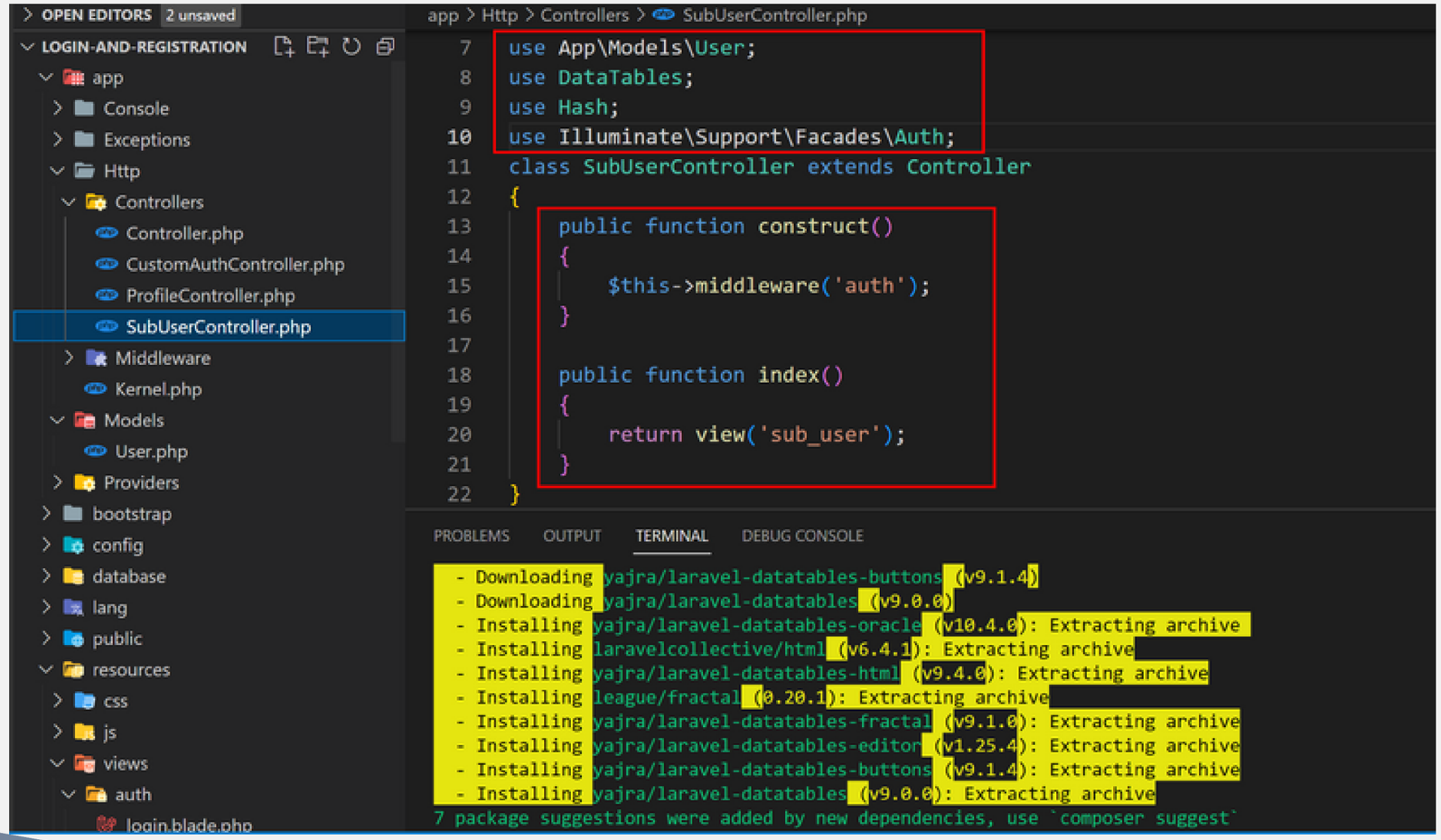
CREATE SUB USER CONTROLLER CLASS

using command: *php artisan make:controller SubUserController*



DOWNLOAD & INSTALL YAJRA DATATABLES PACKAGE

- using command: *composer require yajra/laravel-datatables:"^9.0"*
- After downloaded success go to *SubUserController* and Add Required Class at the header of Controller Class

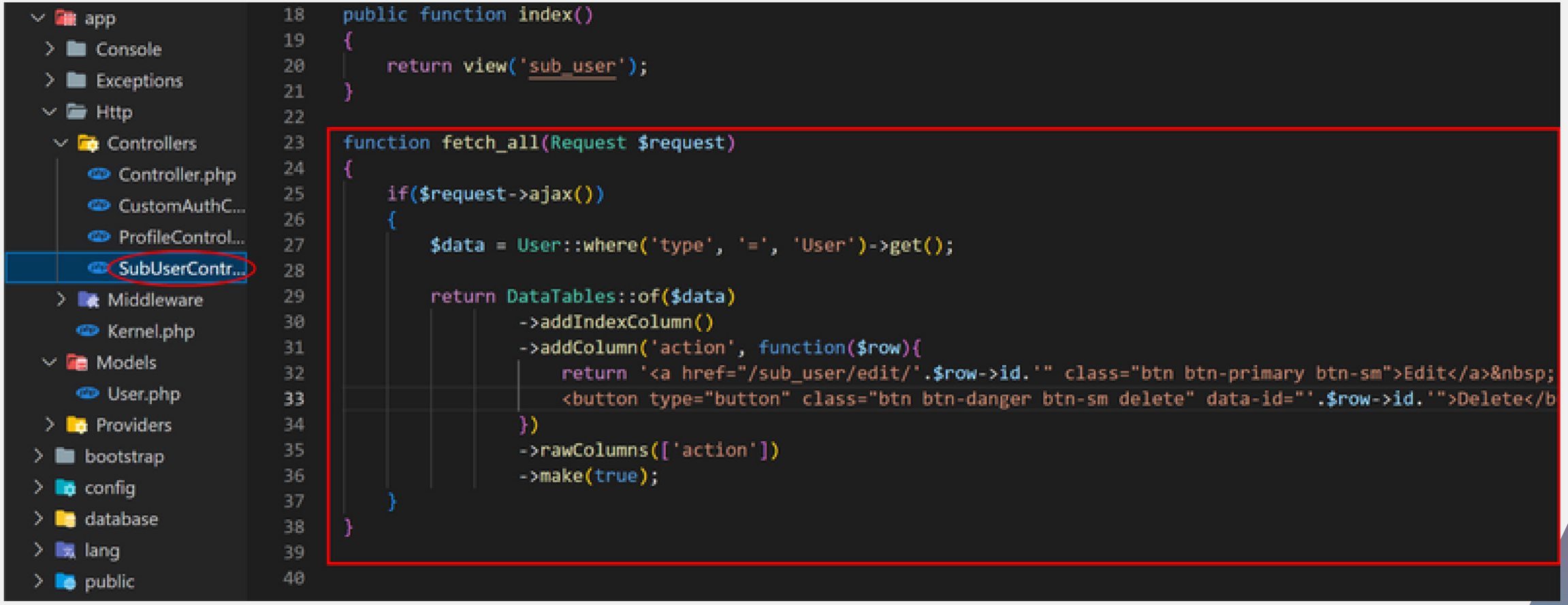


CREATE SUB_USER FILE

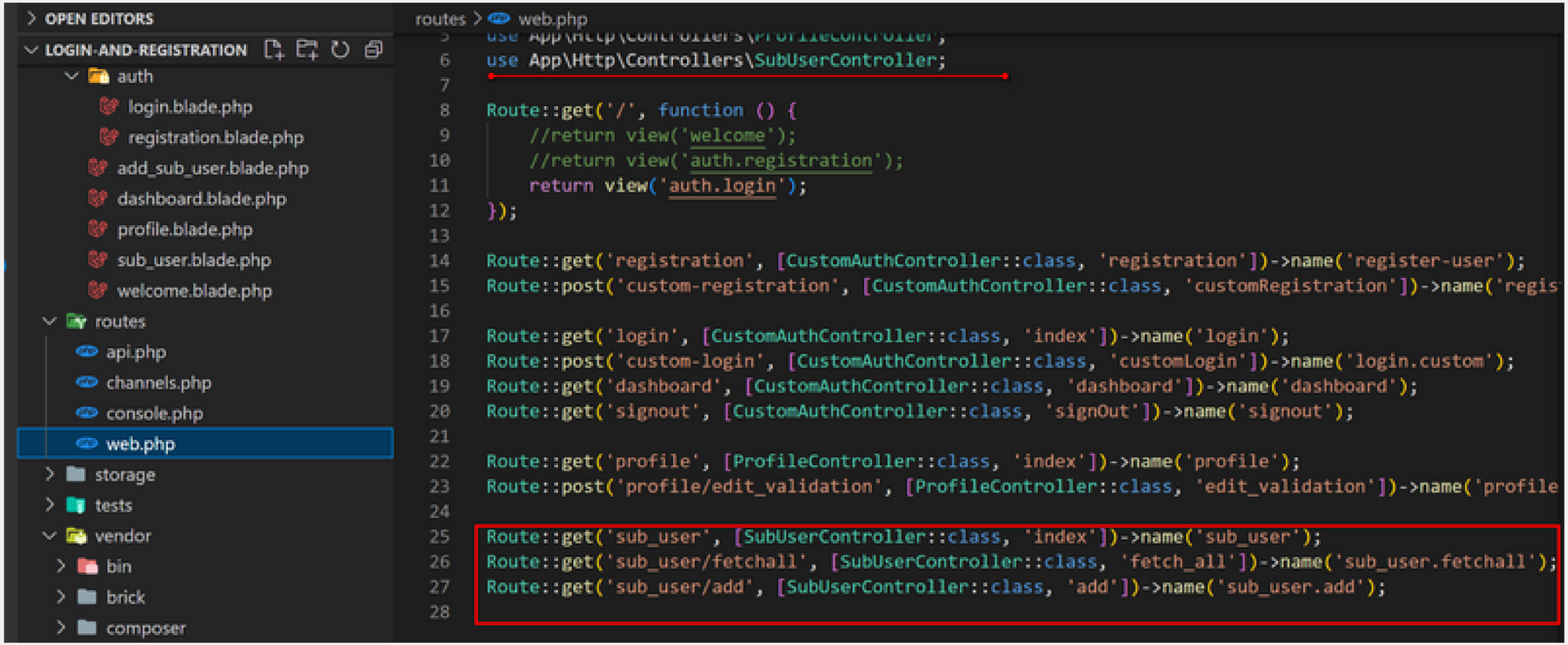
- Create *sub_user.blade.php* file

CREATE FUNCTION TO GET DATA

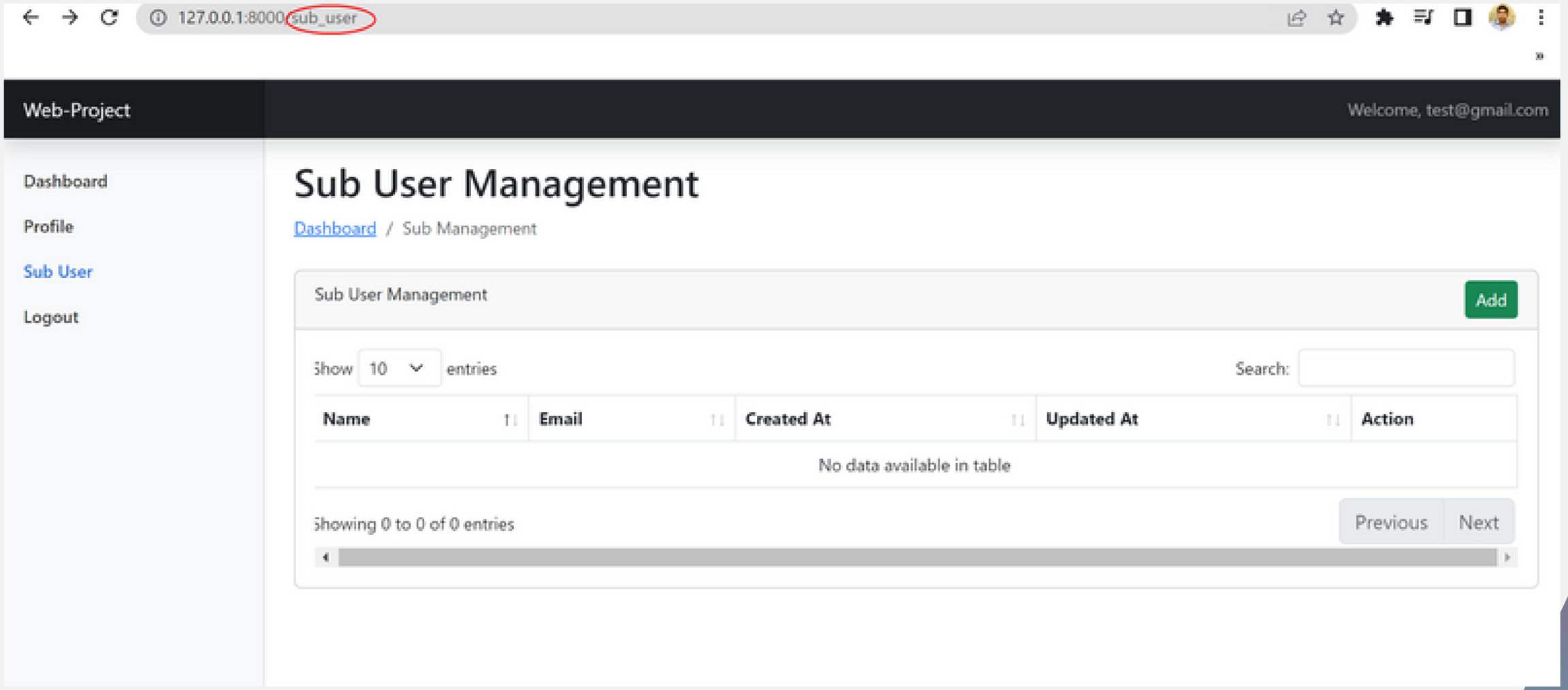
- Add function to get data and action (Edit & Delete)



SET ROUTE FOR CONTROLLER METHOD



CHECK OUTPUT IN BROWSER

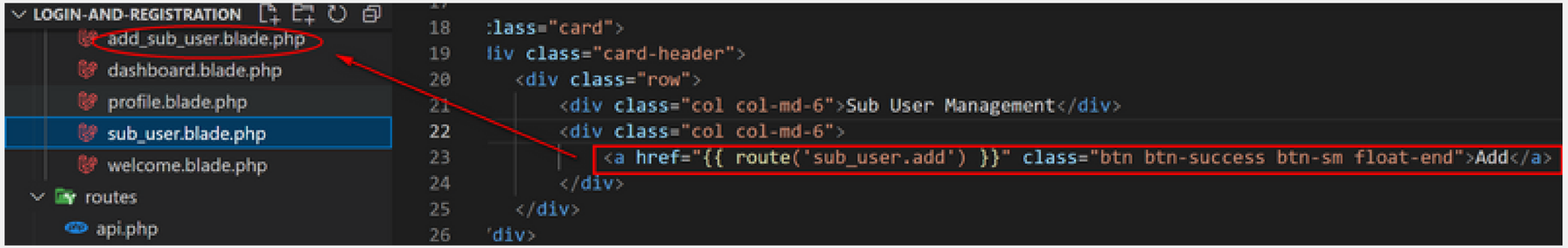


INSERT NEW USER DATA IN MYSQL TABLE

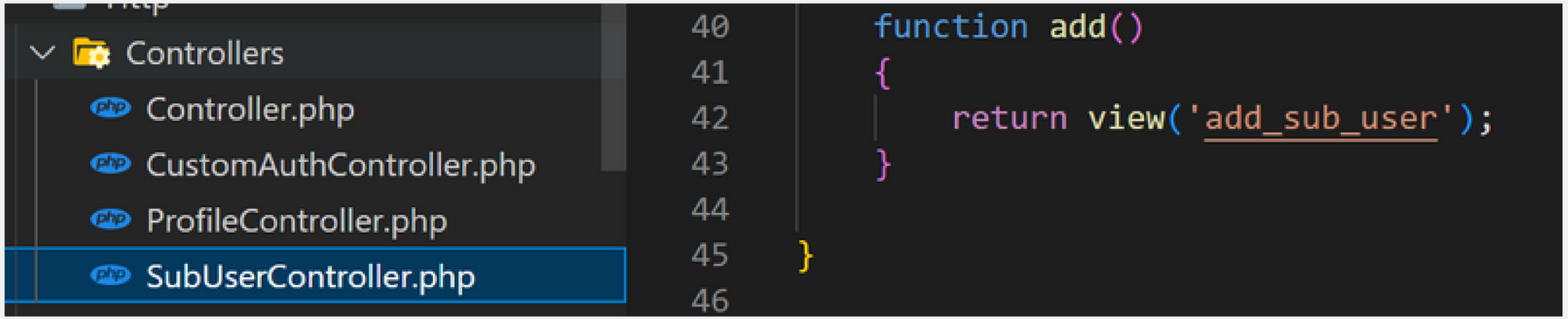
- CREATE LINK FOR GO TO ADD SUB USER FORM
- CREATE ADD() METHOD FOR LOAD SUB USER FORM
- MAKE HTML FORM FOR ADD NEW SUB USER WITH DISPLAY VALIDATION ERROR
- CREATE ADD_VALIDATION() METHOD FOR HANDLE ADD USER FORM DATA
- VALIDATE FORM DATA
- INSERT OR ADD DATA INTO MYSQL TABLE
- DISPLAY SUCCESS MESSAGE ON THE WEB PAGE
- SET ROUTE FOR ADD() & ADD_VALIDATION() METHOD
- CHECK OUTPUT IN BROWSER

CREATE ADD_SUB_USER

- Create *add_sub_user.blade.php*



- Add function add

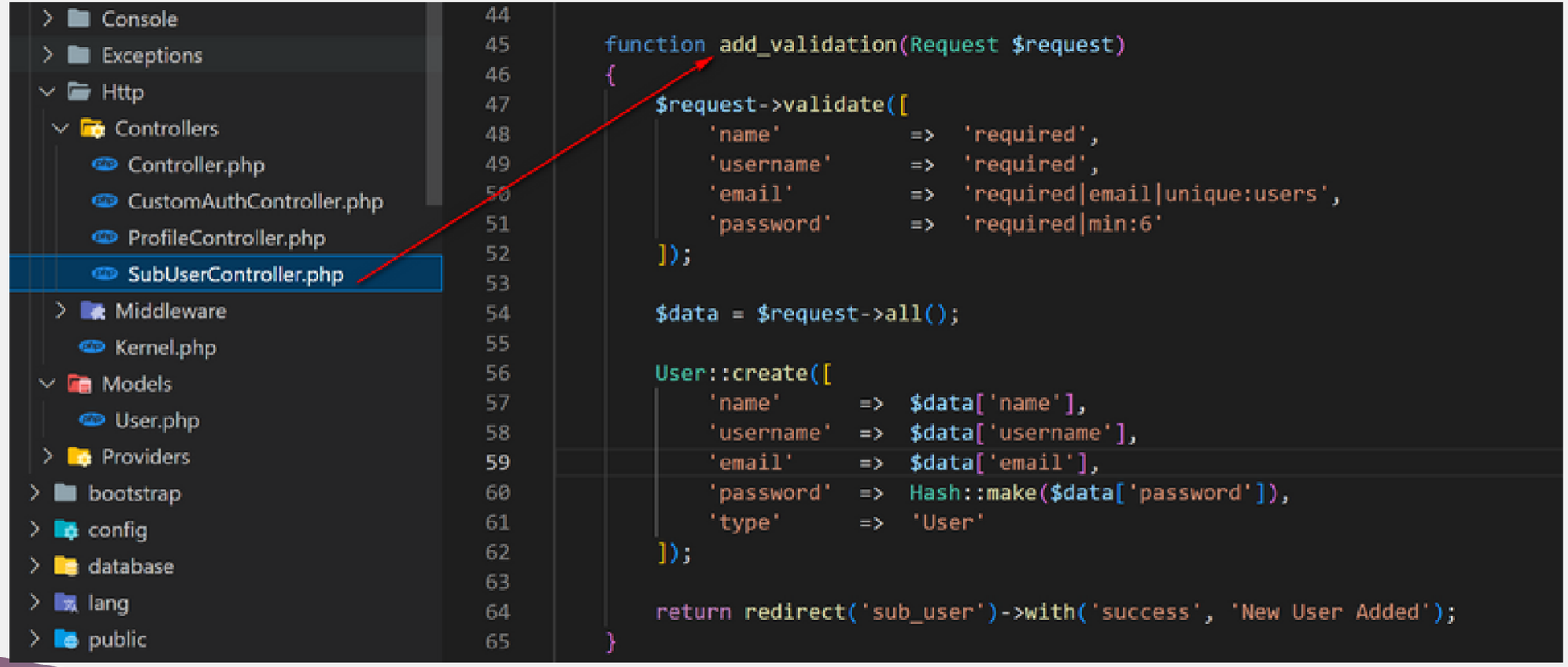


CREATE ADD_SUB_USER (COUNT...)

- Edit *add_sub_user.blade.php*
- Go to add function *add_validation*
- Make HTML Form for Add new Sub User with display validation error

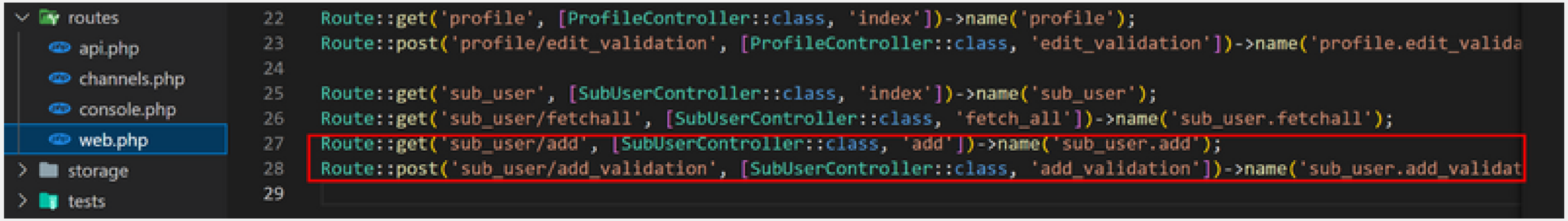
CREATE ADD_SUB_USER (COUNT...)

- Create *add_validation()* method for handle add user form data
- Validate Form data
- Insert or Add data into MySQL table
- Display Success message on the Web page



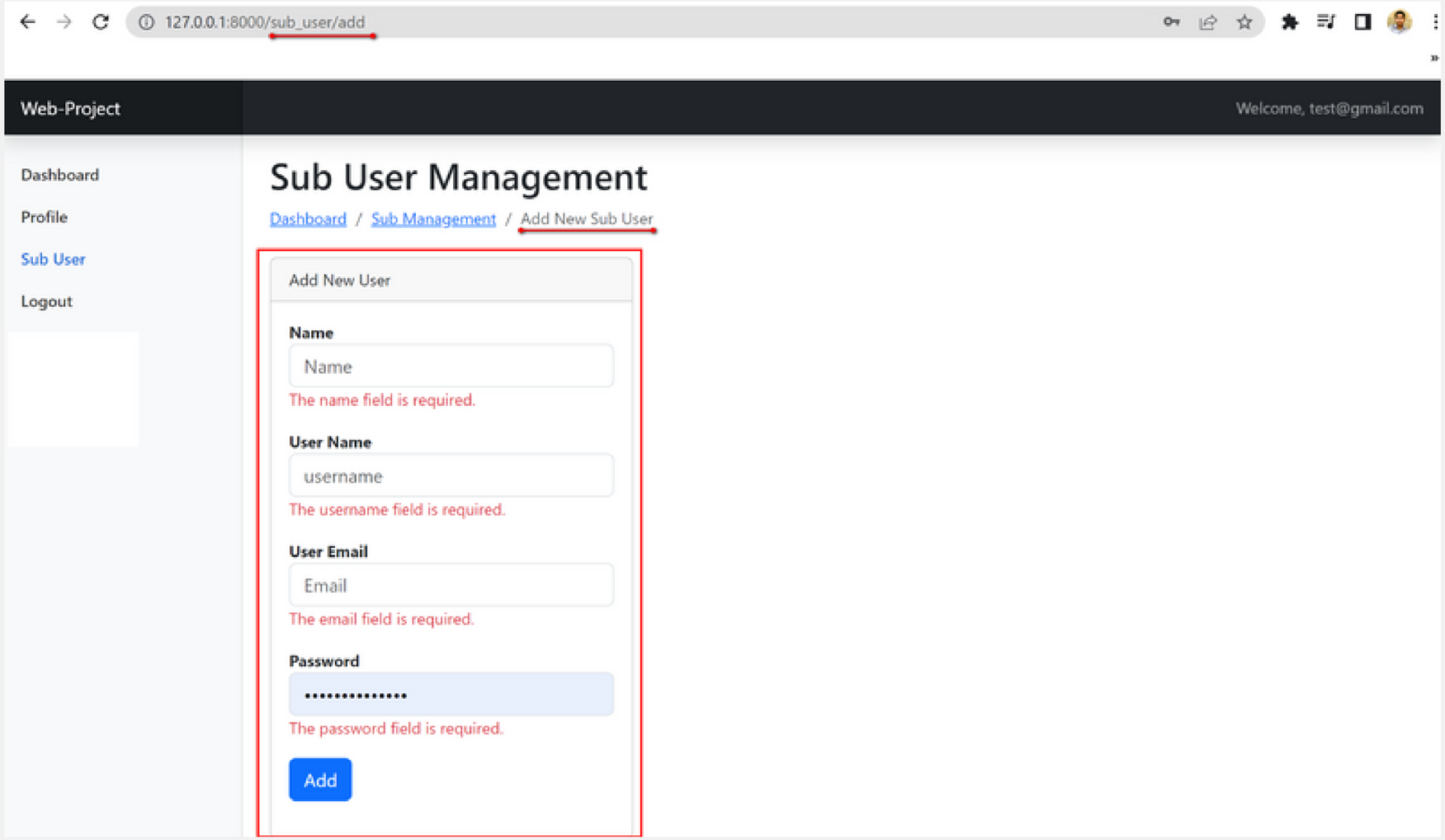
CREATE ADD_SUB_USER (COUNT...)

Set Route for *add()* & *add_validation()* method

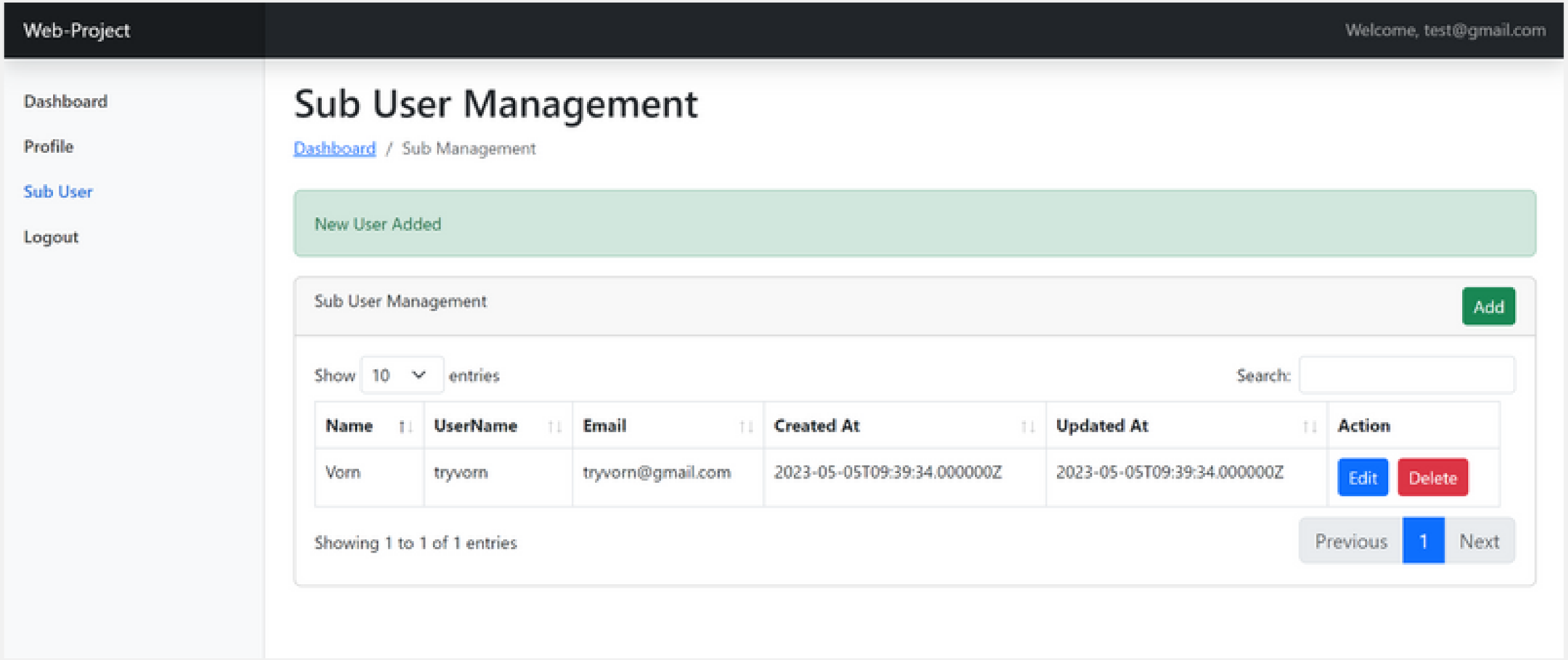


CHECK OUTPUT IN BROWSER

Checking validation message

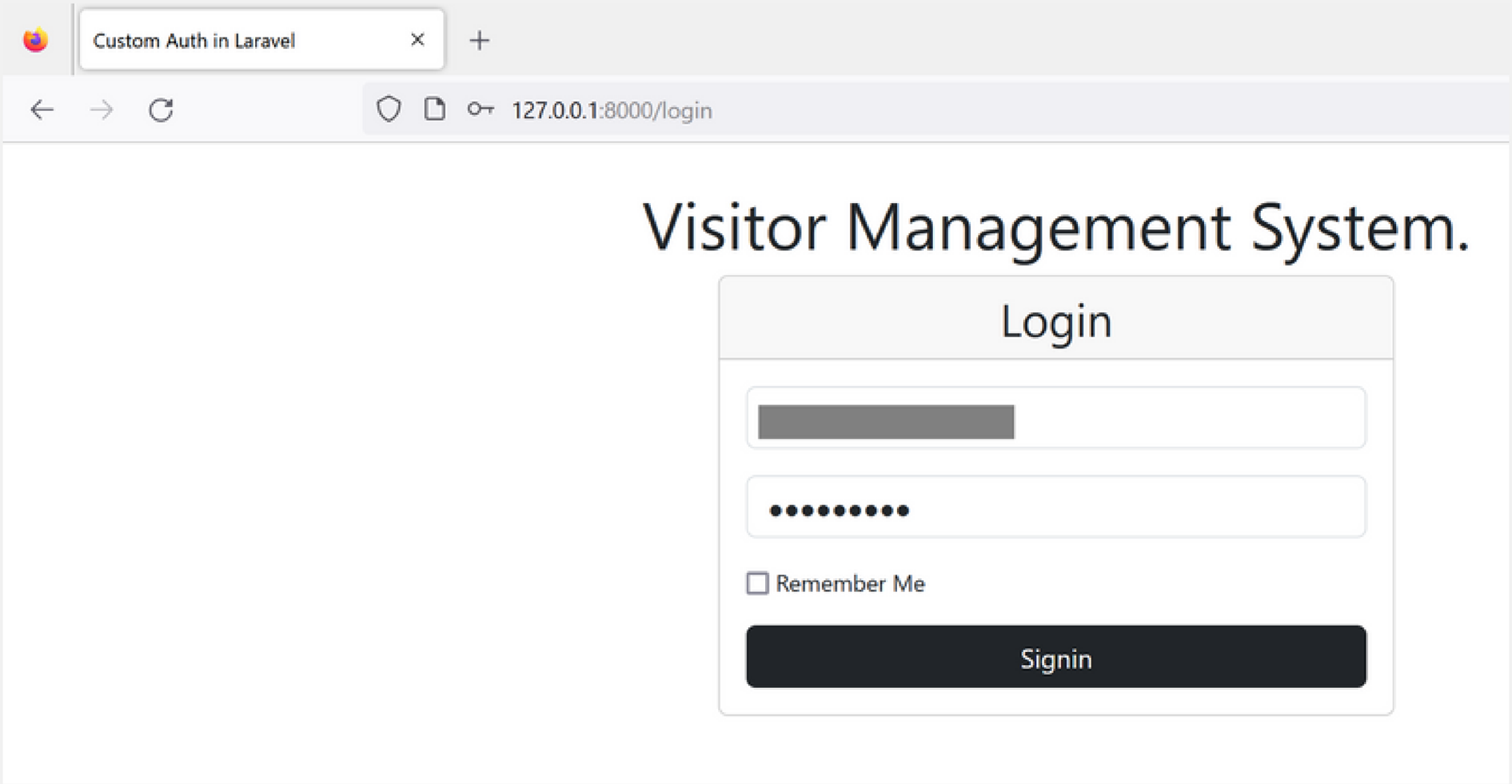


CHECK OUTPUT IN BROWSER (COUNT...)



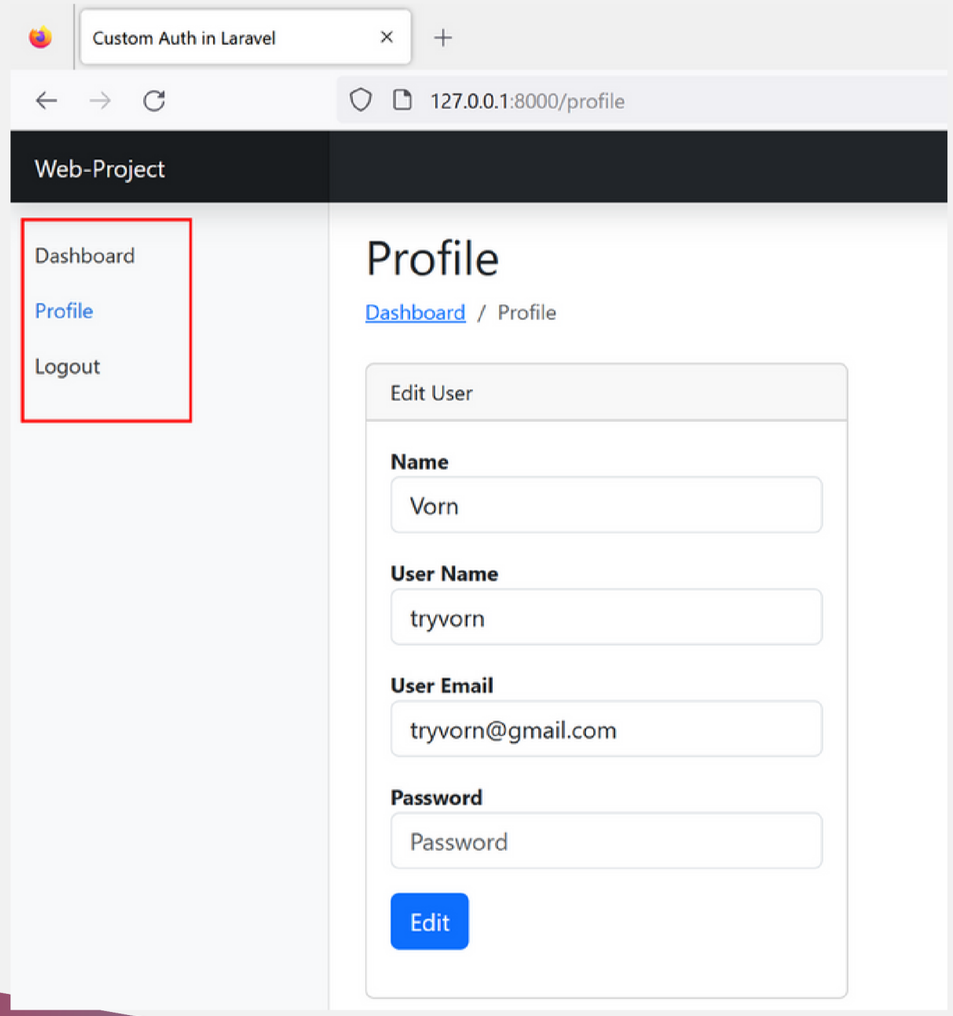
CHECK OUTPUT IN BROWSER (COUNT...)

- To open others browser (Testing with Firefox)



CHECK OUTPUT IN BROWSER (COUNT...)

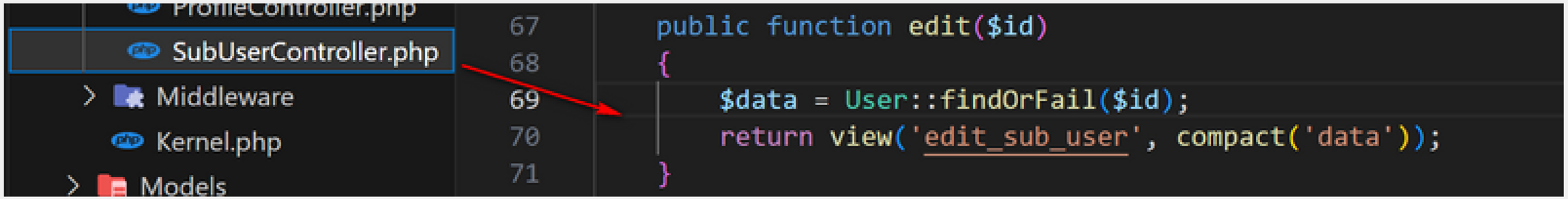
- With Firefox



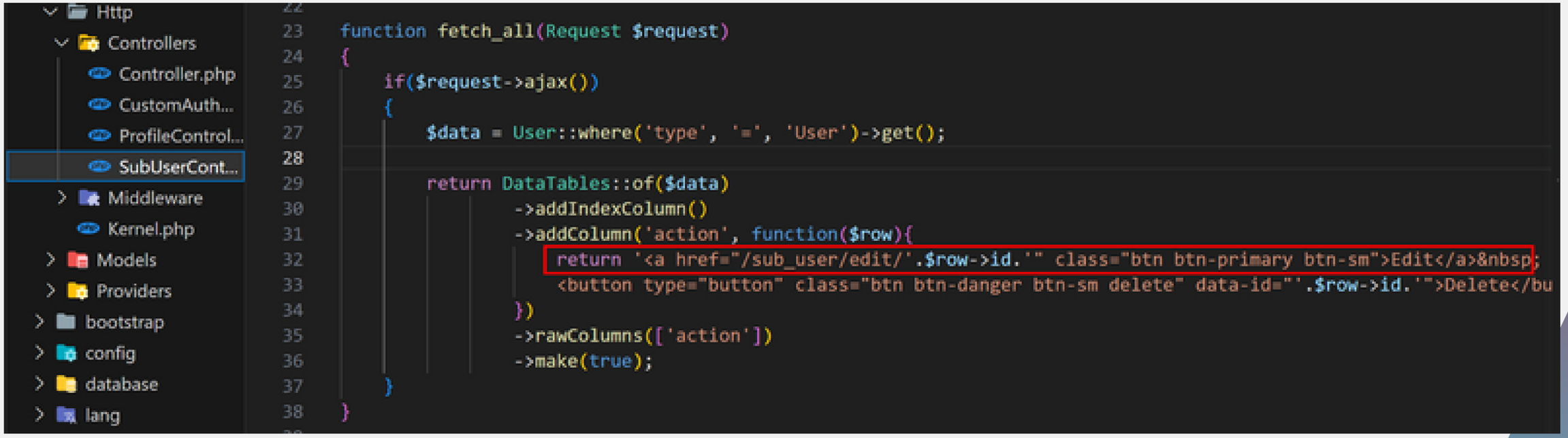
UPDATE USER DATA

- CREATED EDIT() METHOD FOR LOAD EDIT FORM IN THE BROWSER
- CREATE HTML FORM FOR EDIT USER DATA
- FILL EDIT FORM FIELD WITH DATA
- CREATE EDIT_VALIDATION() METHOD FOR HANDLE EDIT USER FORM DATA
- SET ROUTE FOR CONTROLLER METHOD
- CHECK OUTPUT IN BROWSER

CREATED EDIT() METHOD FOR LOAD EDIT FORM IN THE BROWSER



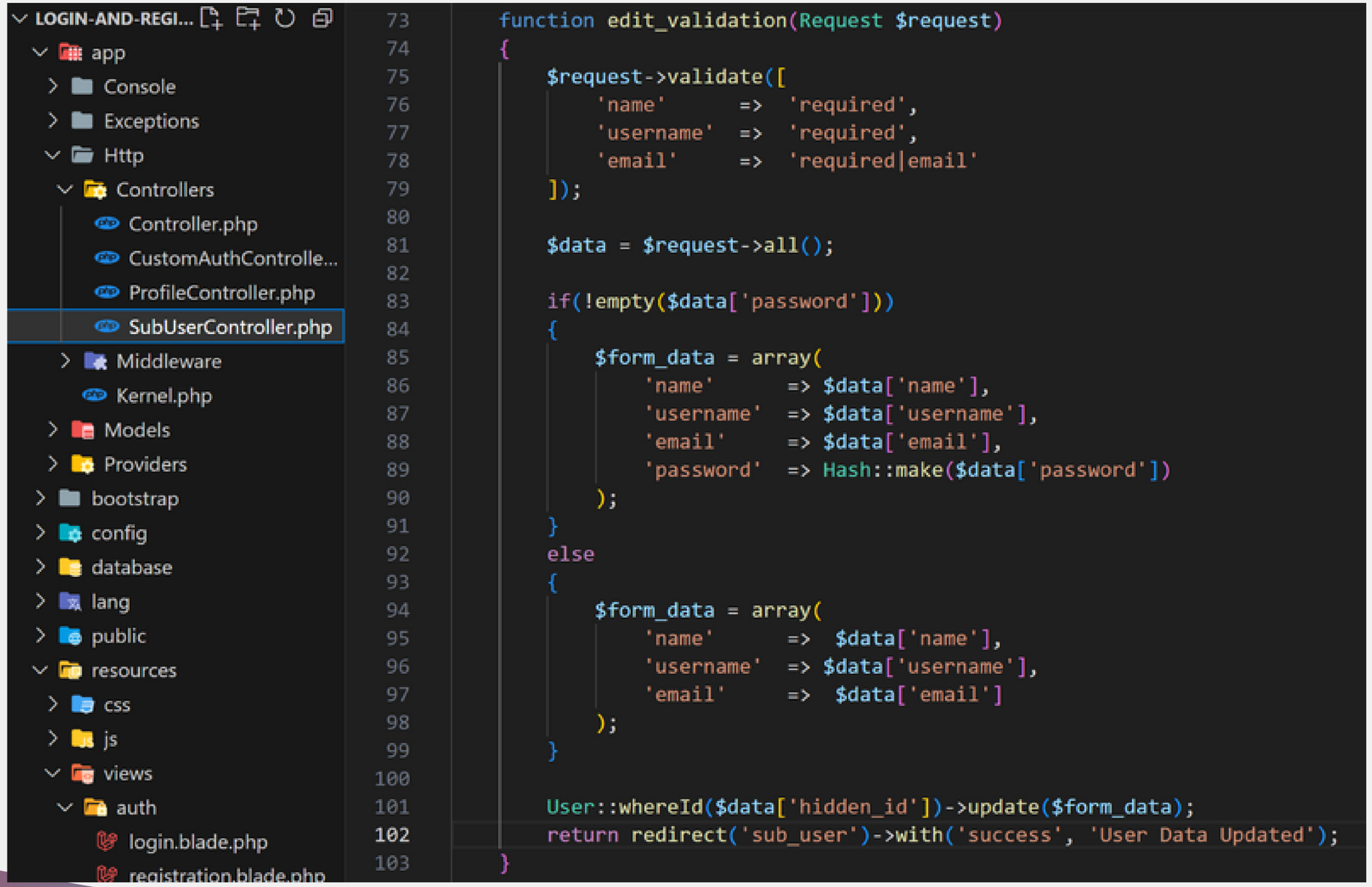
- Set Edit action



CREATED EDIT() METHOD FOR LOAD EDIT FORM IN THE BROWSER (COUNT...)

- Create HTML form for Edit user Data (*edit_sub_user.blade.php*)
- Fill Edit form field with Data

CREATE EDIT_VALIDATION() METHOD FOR HANDLE EDIT USER FORM DATA



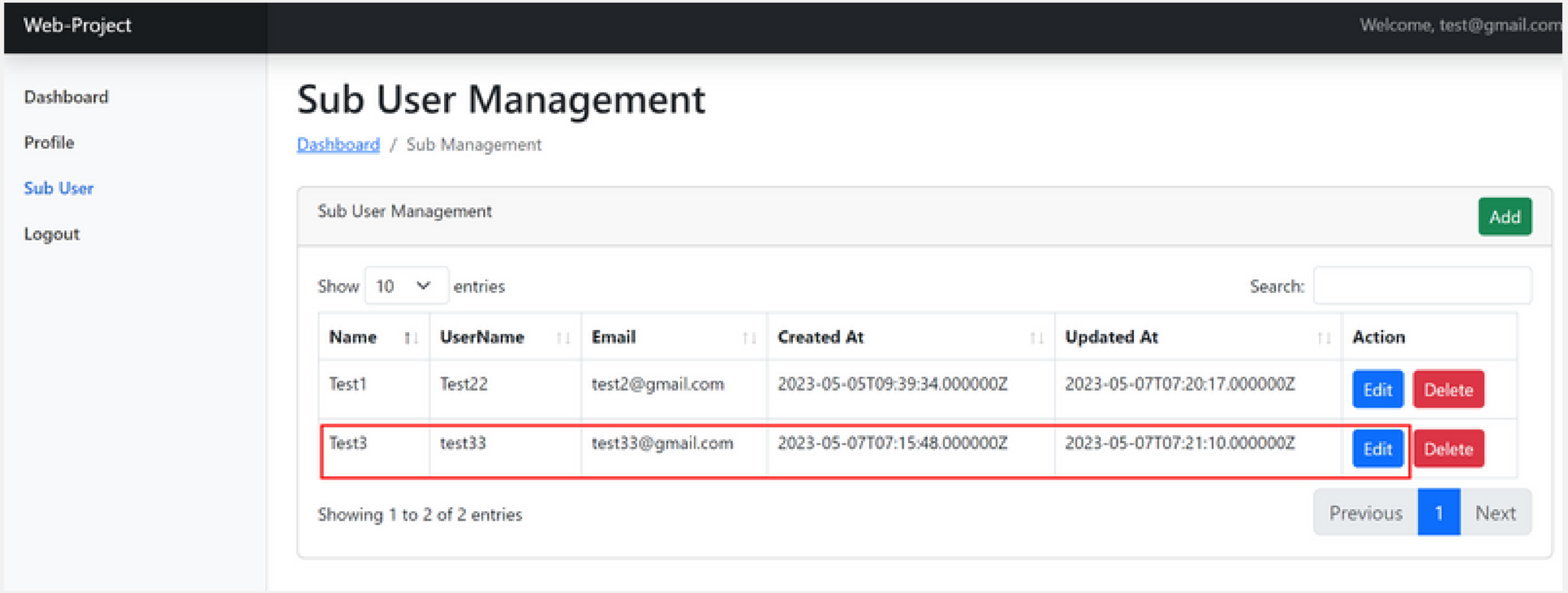

```php
function edit_validation(Request $request)
{
    $request->validate([
        'name'      =>  'required',
        'username'  =>  'required',
        'email'     =>  'required|email'
    ]);

    $data = $request->all();

    if(!empty($data['password']))
    {
        $form_data = array(
            'name'      => $data['name'],
            'username'  => $data['username'],
            'email'     => $data['email'],
            'password'  => Hash::make($data['password'])
        );
    }
    else
    {
        $form_data = array(
            'name'      =>  $data['name'],
            'username'  => $data['username'],
            'email'     =>  $data['email']
        );
    }

    User::whereId($data['hidden_id'])->update($form_data);
    return redirect('sub_user')->with('success', 'User Data Updated');
}
```

SET ROUTE FOR CONTROLLER METHOD

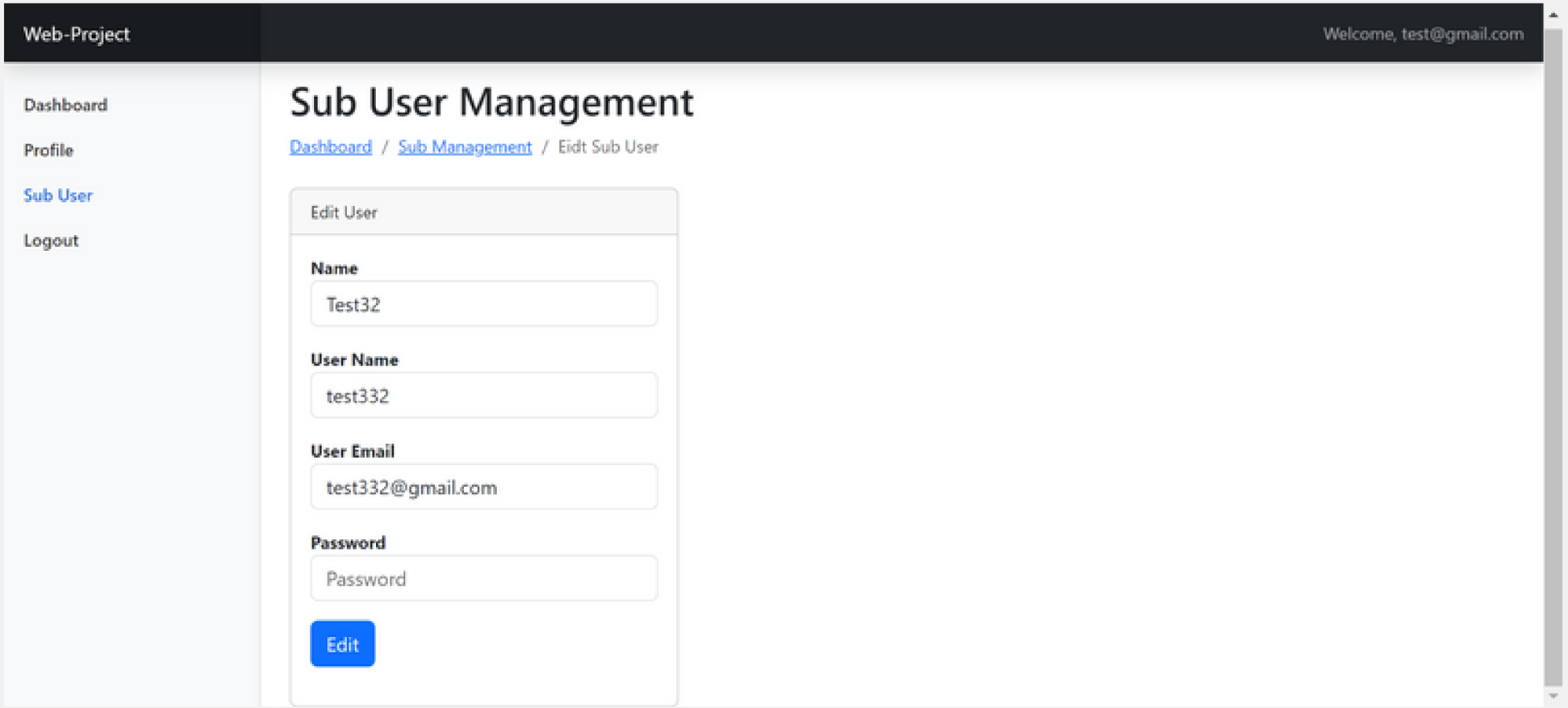
web.php

```
30  Route::get('sub_user/edit/{id}', [SubUserController::class, 'edit'])->name('edit');
31  Route::post('sub_user/edit_validation', [SubUserController::class, 'edit_validation'])->name('sub_user.edit_val
```

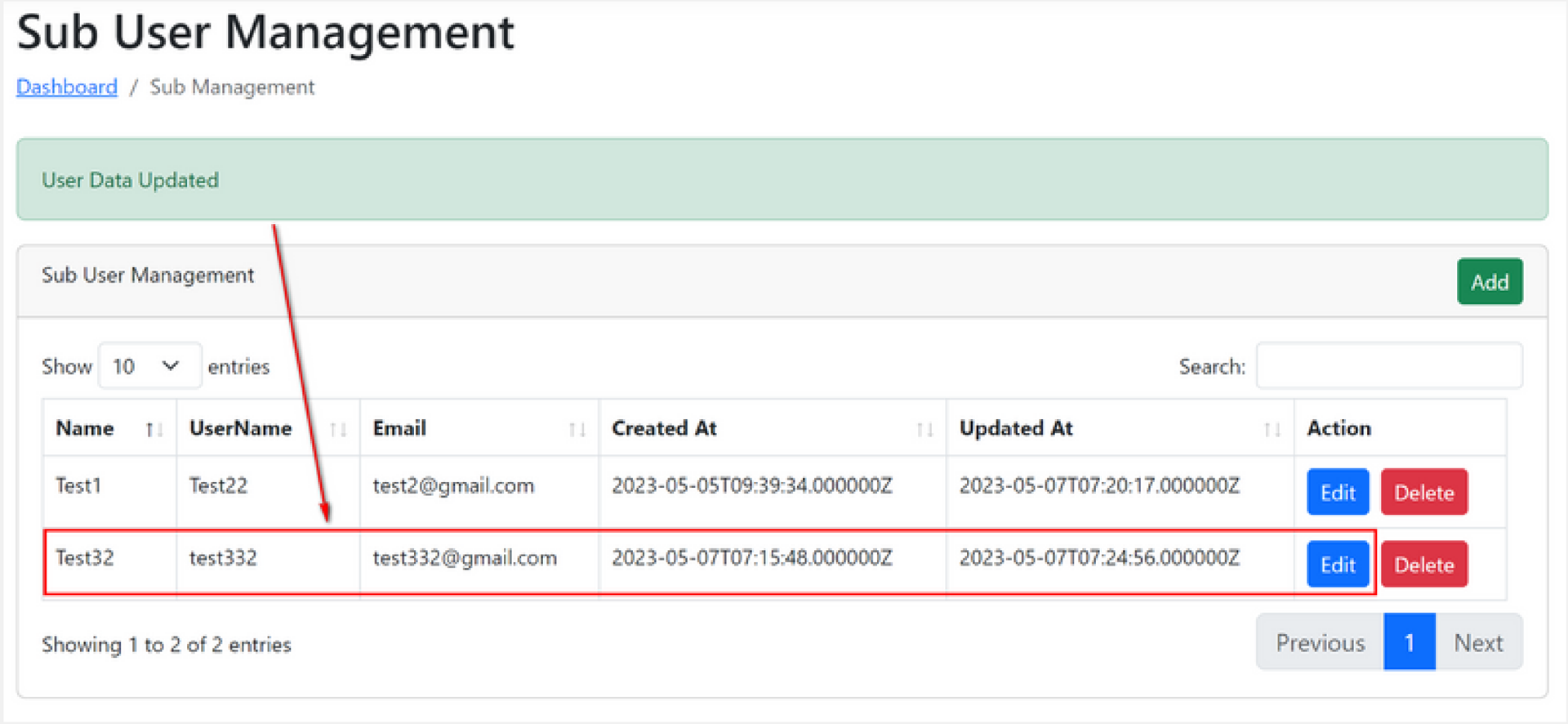
Check Output in browser



Check Output in browser (Count...)



Check Output in browser (Count...)



DELETE USER DATA

- WRITE JAVASCRIPT FOR CONFIRMATION BEFORE DELETE DATA
- CREATE CONTROLLER METHOD FOR DELETE DATA
- SET ROUTE FOR CONTROLLER METHOD
- CHECK OUTPUT IN BROWSER

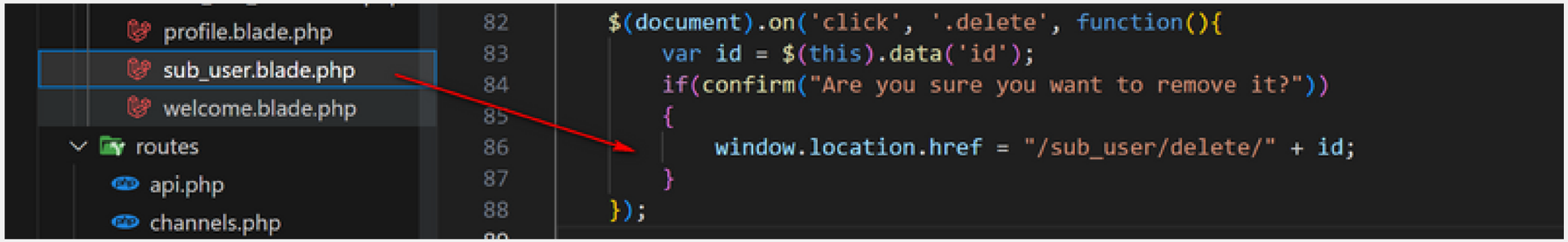
WRITE JAVASCRIPT FOR CONFIRMATION BEFORE DELETE DATA

Add Delete button action

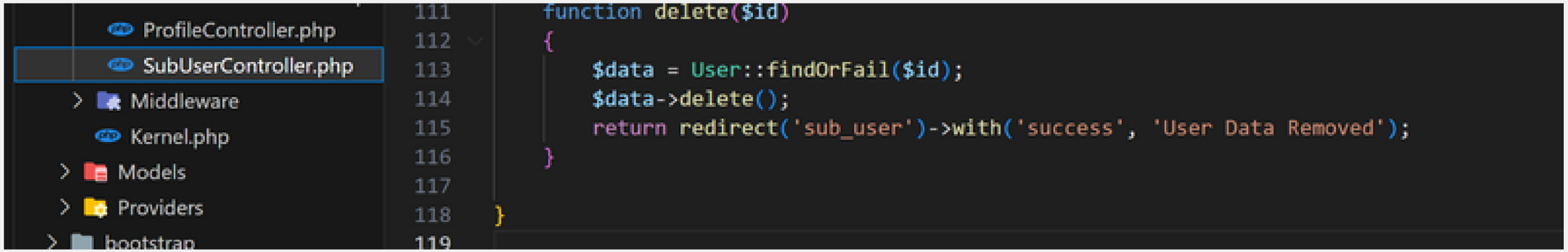
WRITE JAVASCRIPT FOR CONFIRMATION BEFORE DELETE DATA

Add Delete button action

Confirmation before delete data



CREATE CONTROLLER METHOD FOR DELETE DATA



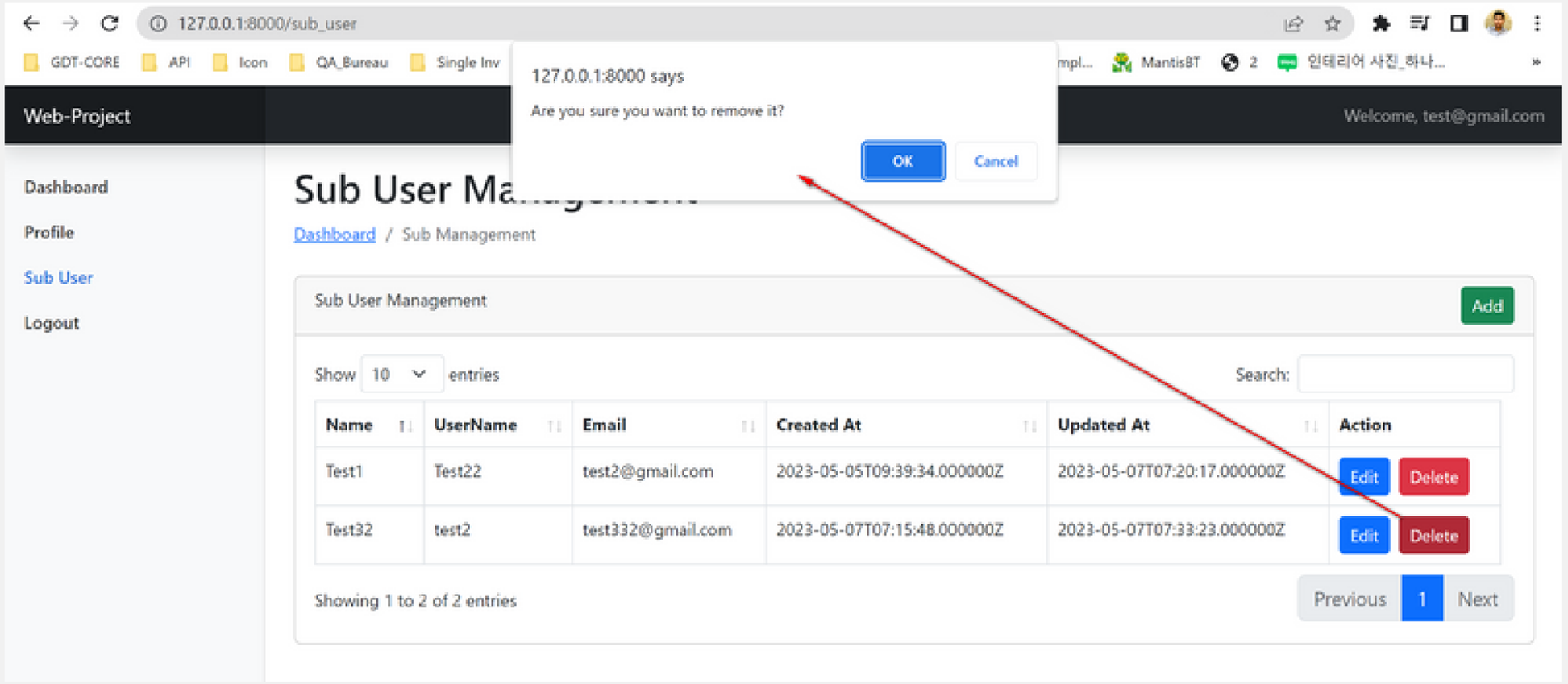
After deleted success show message

SET ROUTE FOR CONTROLLER METHOD

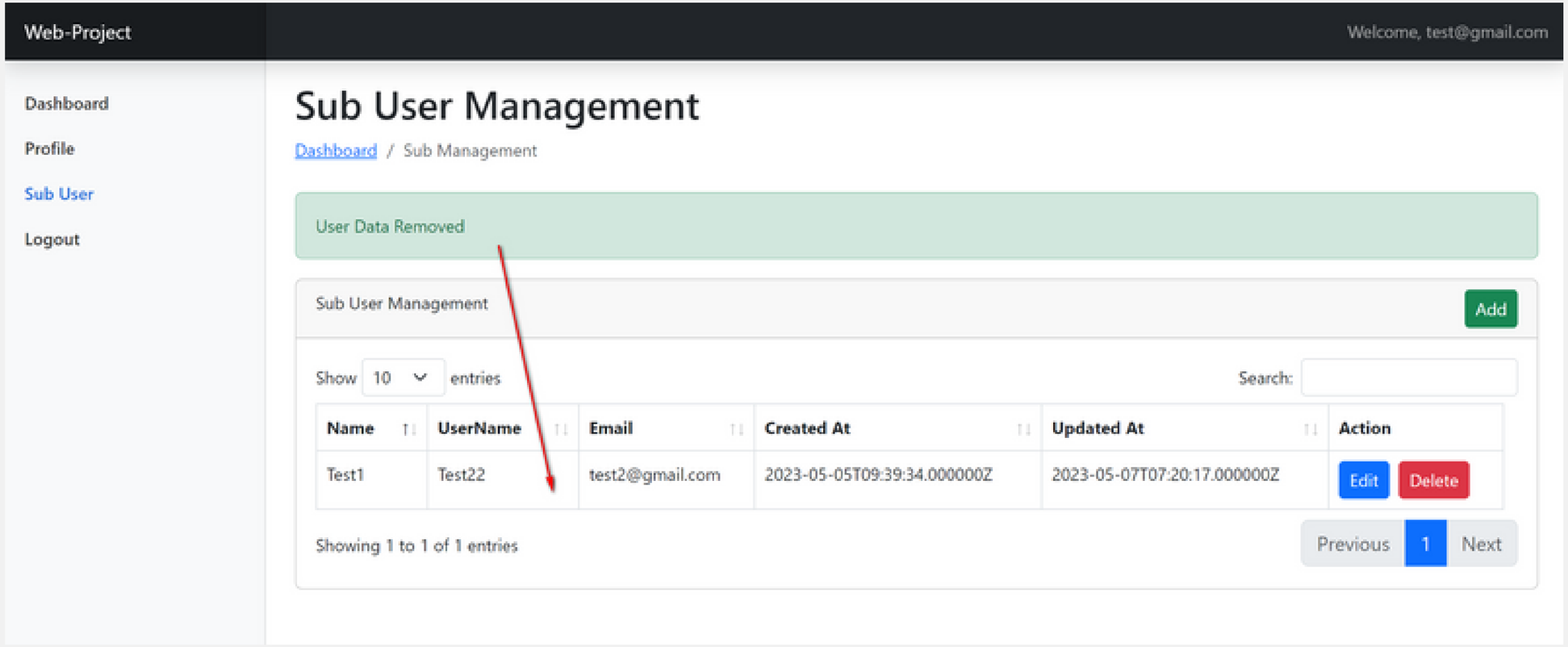
routes
- api.php
- channels.php
- console.php
- web.php

```
29
30  Route::get('sub_user/edit/{id}', [SubUserController::class, 'edit'])->name('edit');
31  Route::post('sub_user/edit_validation', [SubUserController::class, 'edit_validation'])->name('sub_user.ed
32
33  Route::get('sub_user/delete/{id}', [SubUserController::class, 'delete'])->name('delete');
34
```

Check Output in browser



Check Output in browser (Count...)



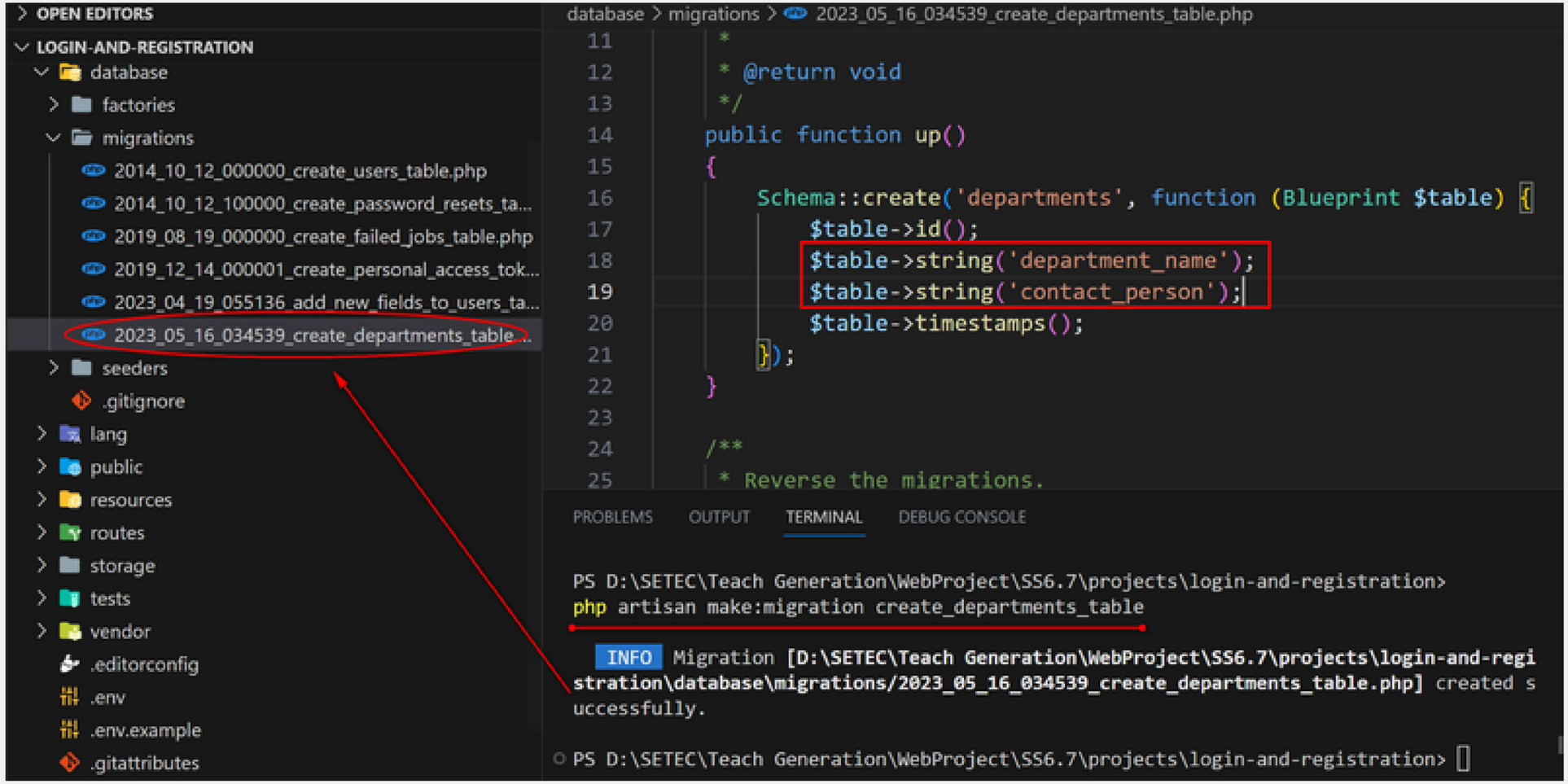
DATATABLES SERVER SIDE PROCESSING OF DEPARTMENT DATA

- CREATE DEPARTMENTS TABLE IN MYSQL FROM LARAVEL APPLICATION
- MAKE DEPARTMENT MODEL CLASS FILE FOR DATABASE OPERATION
- CREATE DEPARTMENT CONTROLLERS CLASS FILE FOR HANDLE HTTP REQUEST
- PREVENT ACCESS TO CONTROLLER METHOD FROM GUEST USER
- CREATE INDEX() METHOD IN DEPARTMENT CONTROLLER FILE
- CREATE VIEW FILE WITH JQUERY CODE FOR DATATABLES
- WRITE PHP SCRIPT FOR LOAD DATA IN YAJRA DATATABLE
- ADD DEPARTMENT LINK AT MAIN DASHBOARD FILE
- SET ROUTE FOR DEPARTMENT CONTROLLER METHOD & CHECK OUTPUT IN THE BROWSER

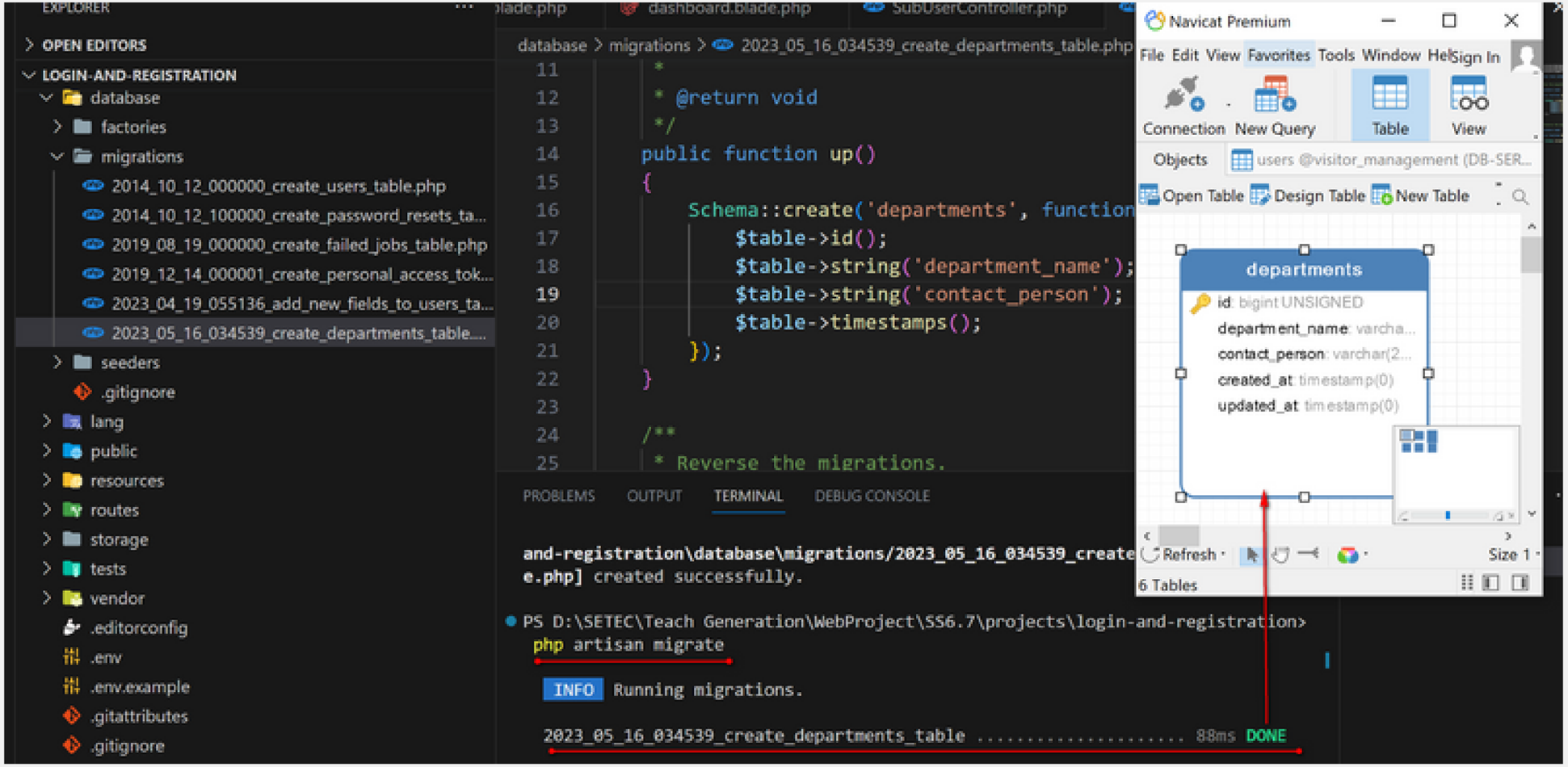
CREATE DEPARTMENTS TABLE IN MYSQL FROM LARAVEL APPLICATION

- Create **department** table migration file:

  php artisan make:migration create_departments_table

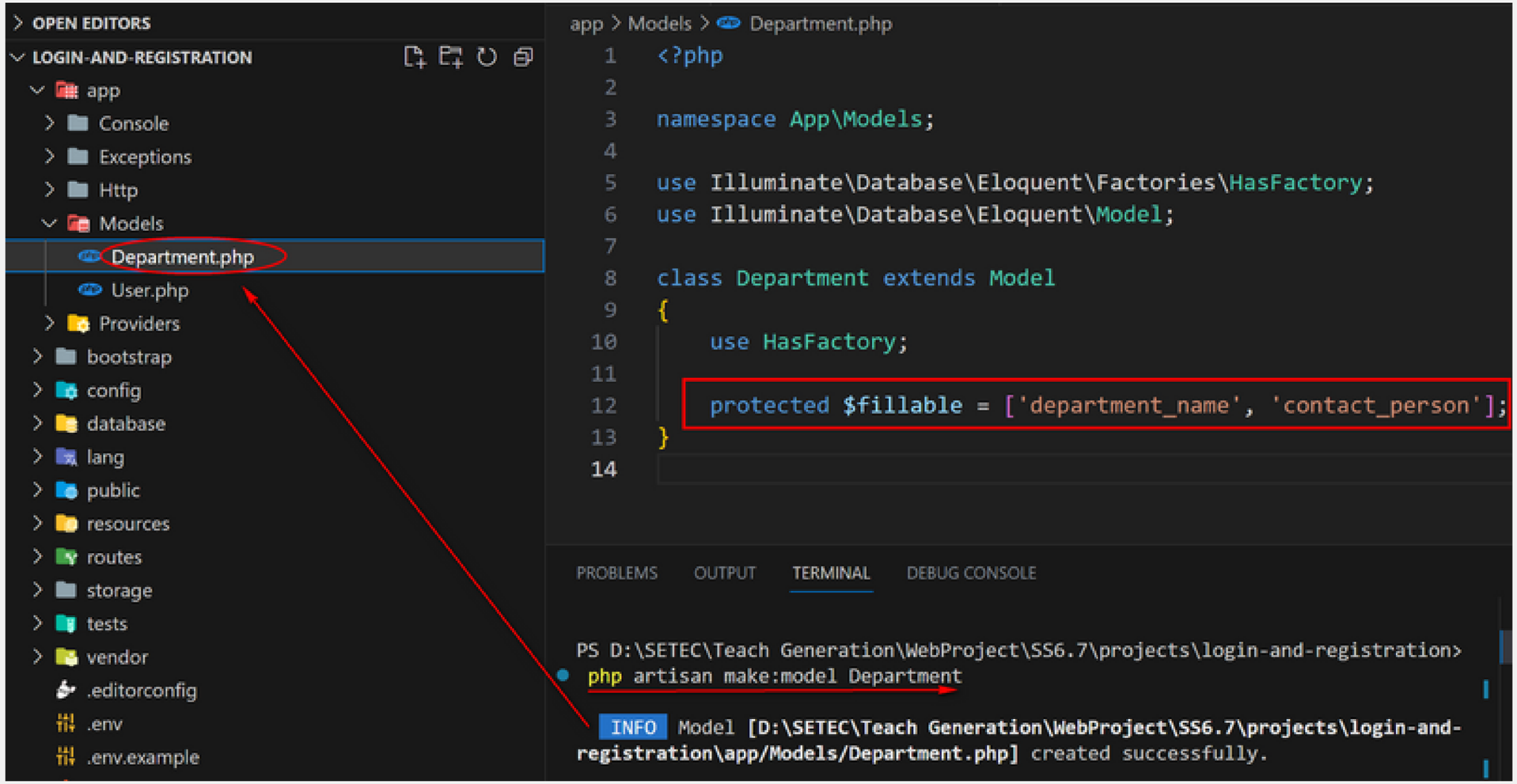


- using command: *php artisan migrate*



MAKE DEPARTMENT MODEL CLASS FILE FOR DATABASE OPERATION

- Create **Department** model: *php artisan make:model Department*

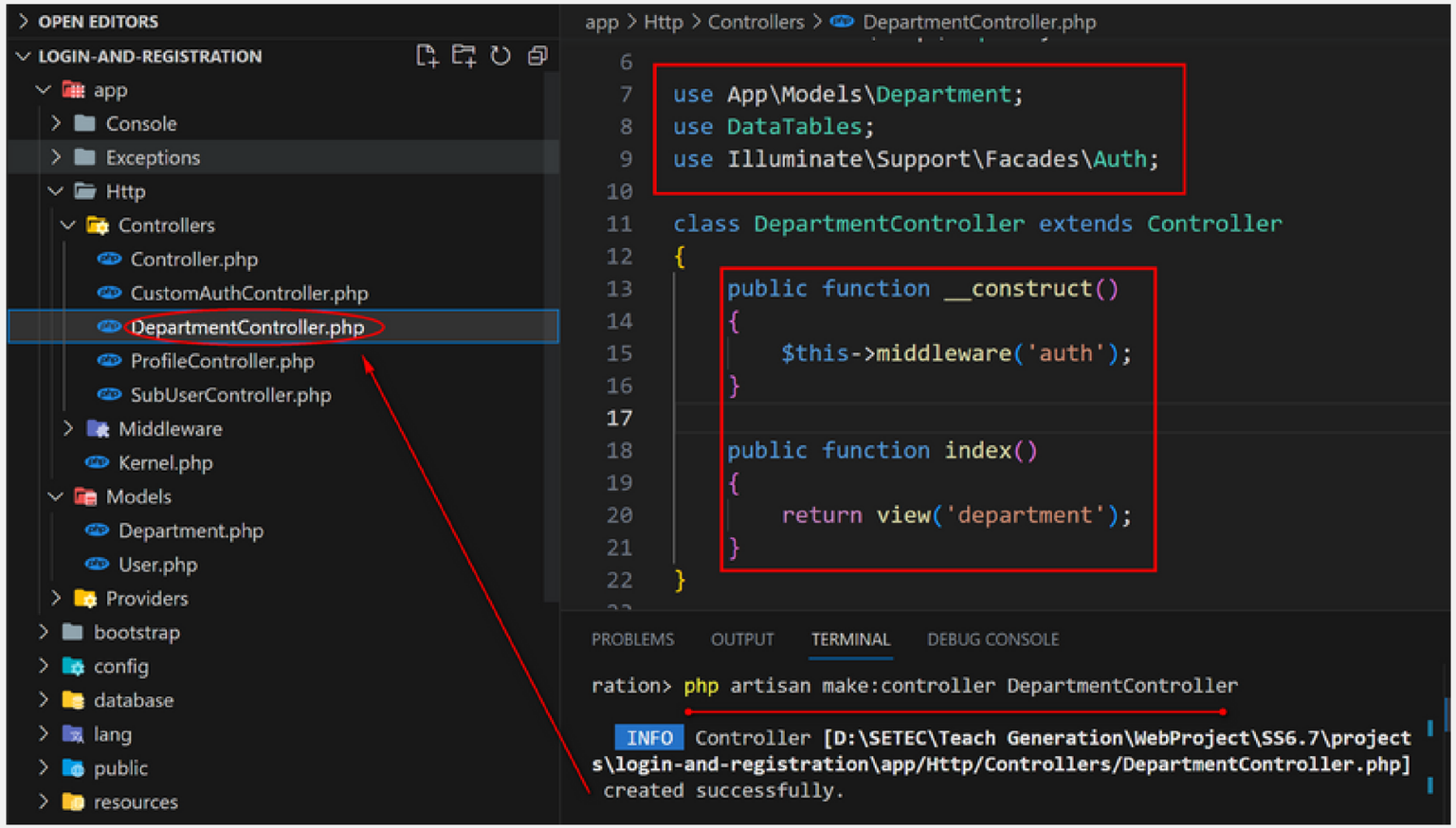


CREATE DEPARTMENT CONTROLLERS CLASS FILE FOR HANDLE HTTP REQUEST

- Create **Department** controller file:

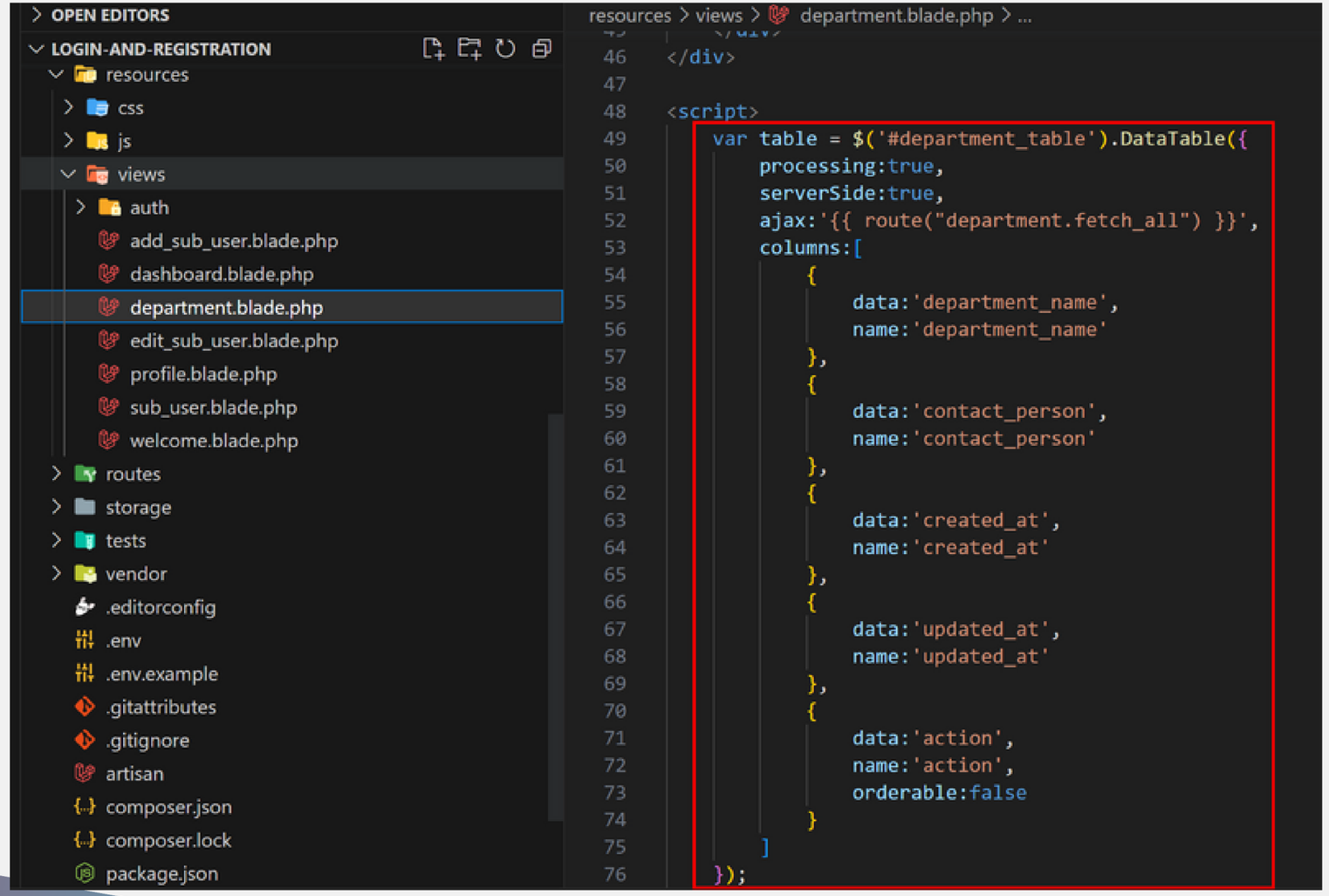
  php artisan make:controller DepartmentController

- Prevent Access to Controller method from Guest User



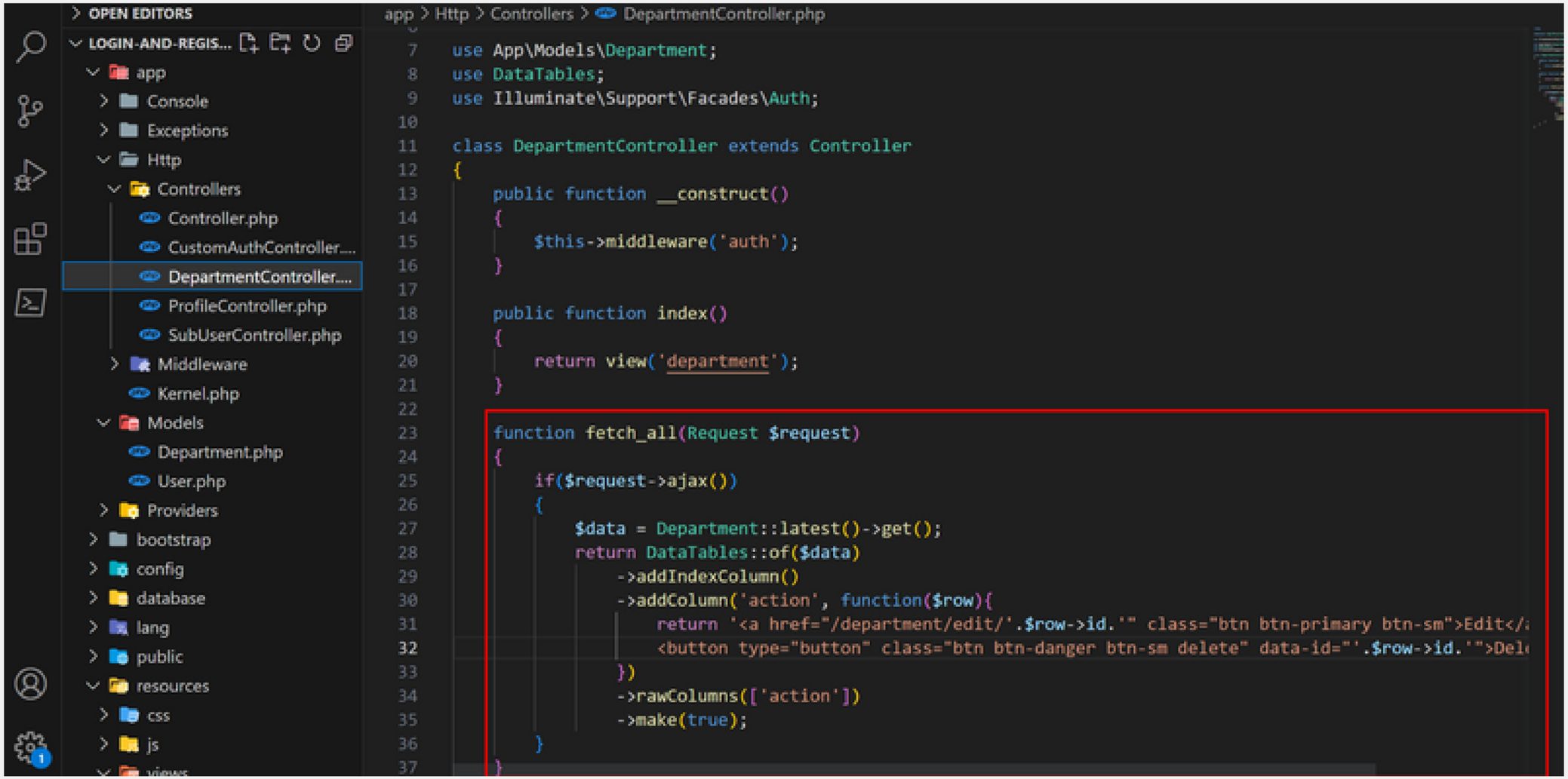
CREATE INDEX() METHOD IN DEPARTMENT CONTROLLER FILE

- Create **department.blade.php** file.
- Create View file with jQuery Code for DataTables



CREATE INDEX() METHOD IN DEPARTMENT CONTROLLER FILE (COUNT...)

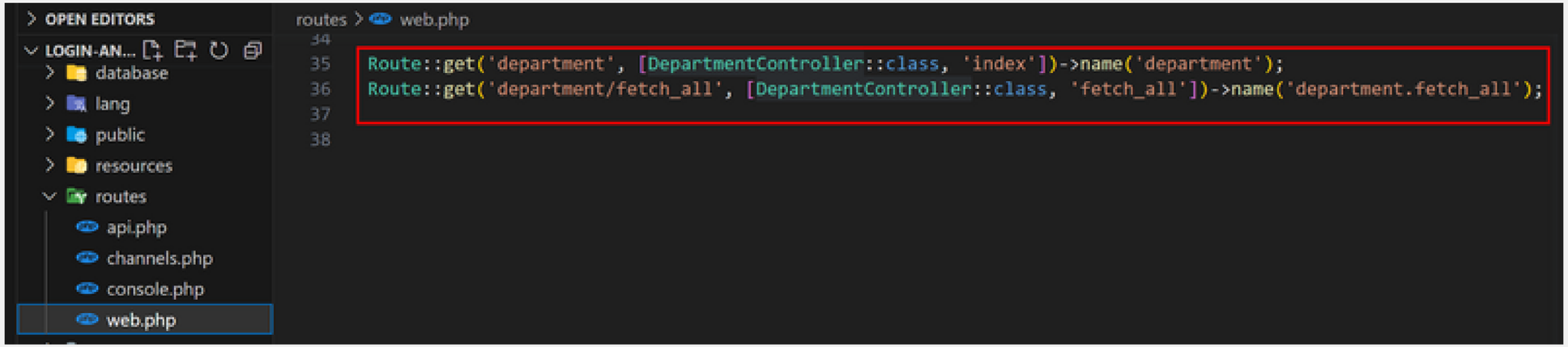
- Write PHP Script for Load Data in DataTable



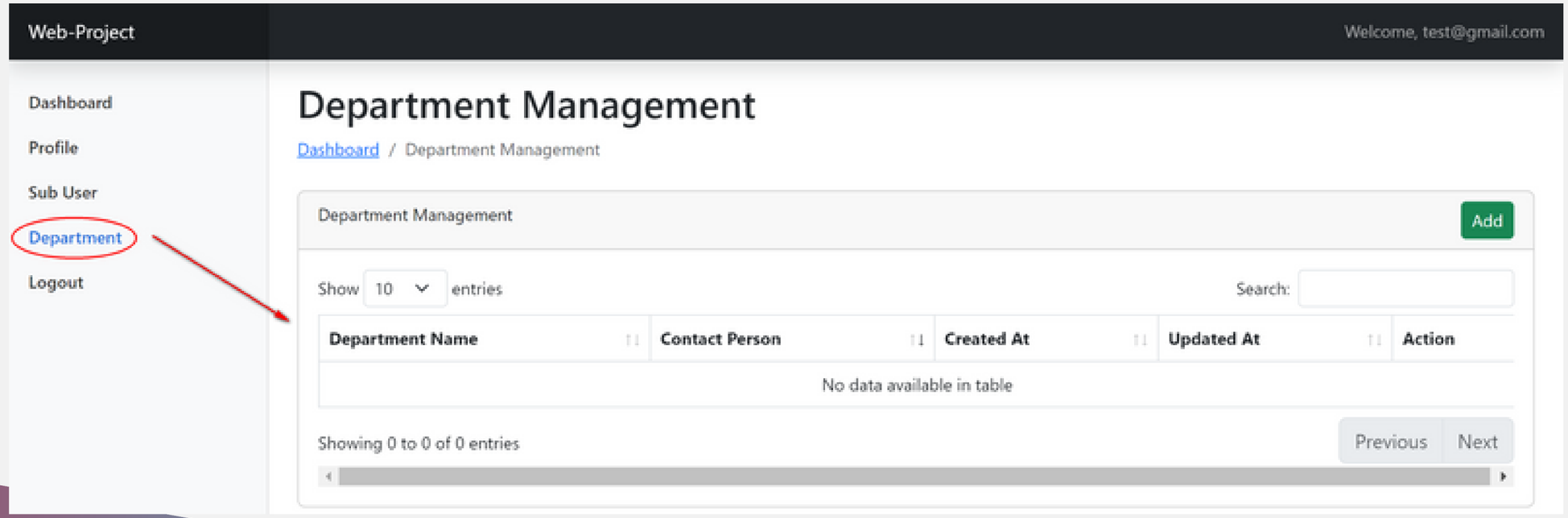
ADD DEPARTMENT LINK AT MAIN DASHBOARD FILE

- Go to dashboard.blade.php

SET ROUTE FOR DEPARTMENT CONTROLLER METHOD



CHECK OUTPUT IN THE BROWSER

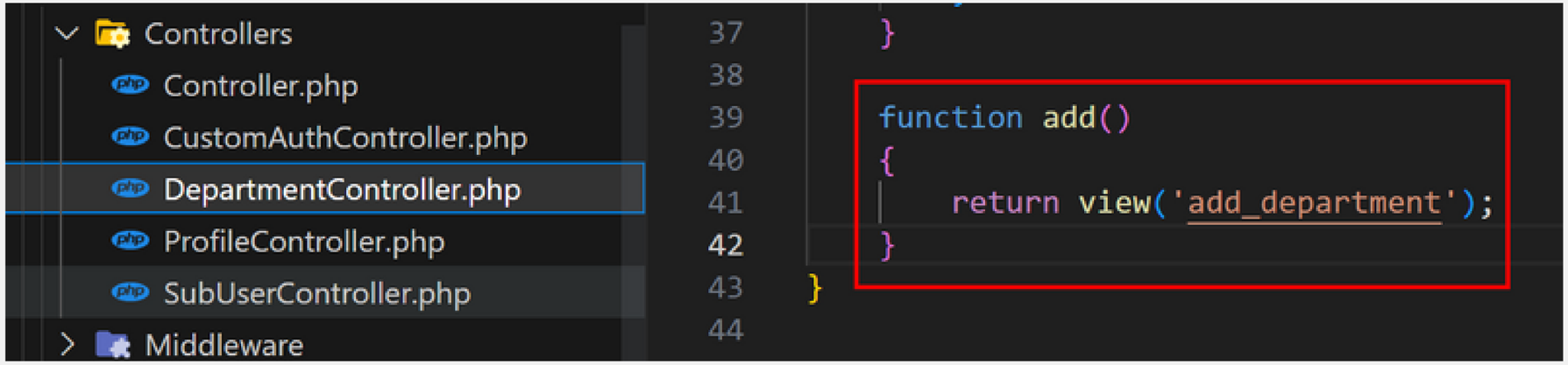


ADD DEPARTMENT DATA IN MYSQL TABLE

- CREATE LINK FOR GO TO ADD DEPARTMENT FORM
- CREATE ADD() METHOD FOR LOAD ADD DEPARTMENT FORM IN THE BROWSER
- CREATE HTML FORM FOR ADD DEPARTMENT DATA WITH DISPLAY VALIDATION ERROR
- WRITE JQUERY CODE FOR ADD OR REMOVE INPUT FIELD DYNAMICALLY
- CREATE ADD_VALIDATION() METHOD FOR HANDLE INSERT ADD DEPARTMENT FORM DATA
- INSERT DATA INTO MYSQL TABLE USING LARAVEL ELOQUENT
- DISPLAY SUCCESS MESSAGE ON THE WEB PAGE AFTER INSERTING OF DATA
- CHECK OUTPUT IN THE BROWSER

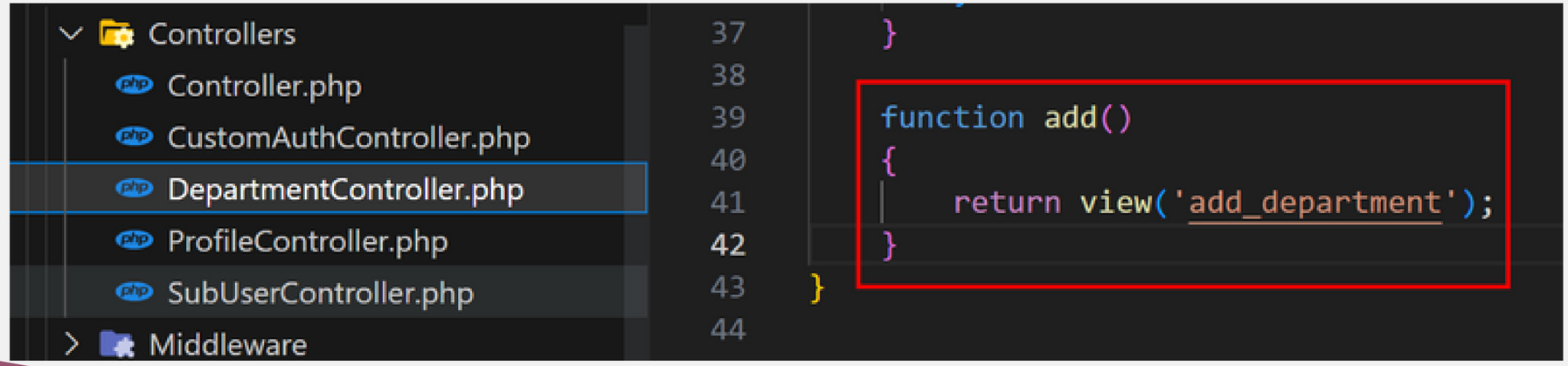
CREATE LINK FOR GO TO ADD DEPARTMENT FORM

- Go to department controller class & create **add** method



CREATE LINK FOR GO TO ADD DEPARTMENT FORM

- Go to department controller class
- Create **add()** method for load add department form in the browser



CREATE HTML FORM FOR ADD DEPARTMENT DATA WITH DISPLAY VALIDATION ERROR

- Display validation error

WRITE JQUERY CODE FOR ADD OR REMOVE INPUT FIELD DYNAMICALLY

CREATE ADD_VALIDATION() METHOD FOR HANDLE INSERT ADD DEPARTMENT FORM DATA

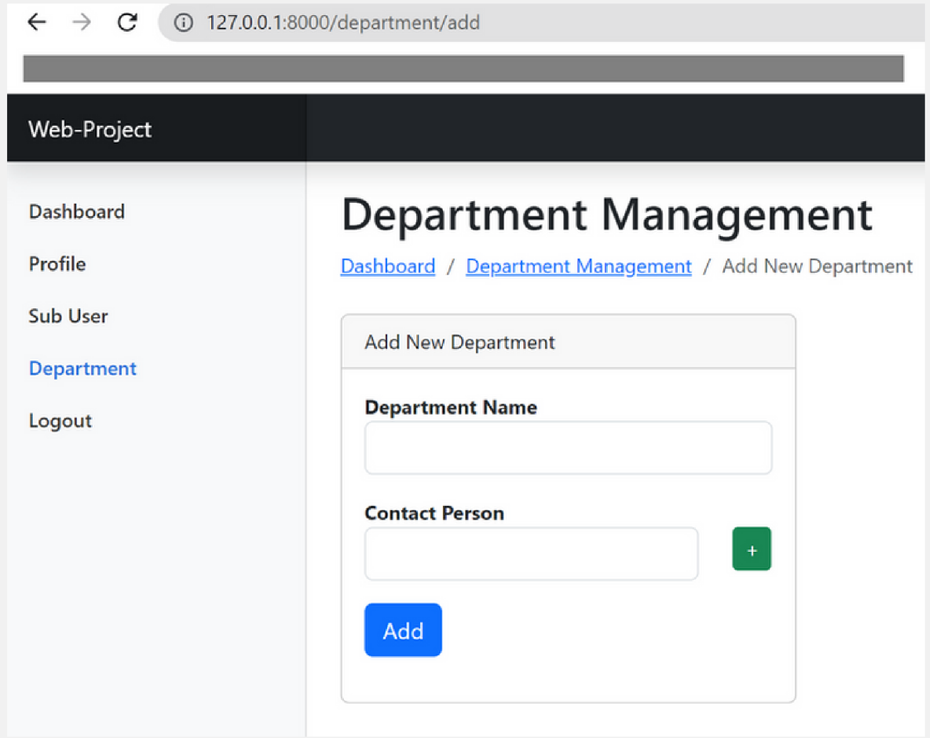
CREATE ADD_VALIDATION() METHOD FOR HANDLE INSERT ADD DEPARTMENT FORM DATA (COUNT...)

- Set Route for Department Controller method

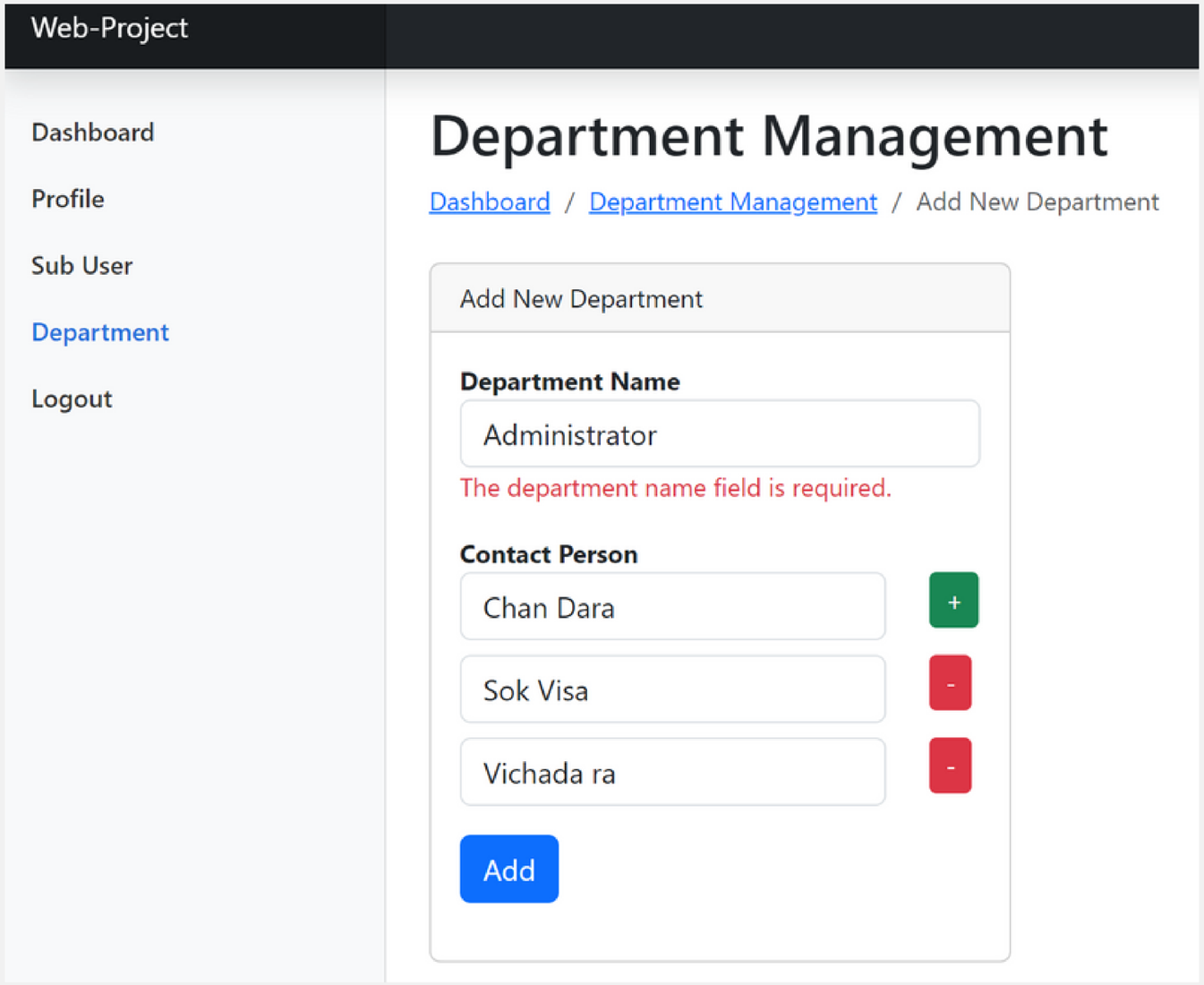
routes
- api.php
- channels.php
- console.php
- web.php

```
34
35  Route::get('department', [DepartmentController::class, 'index'])->name('department');
36  Route::get('department/fetch_all', [DepartmentController::class, 'fetch_all'])->name('department.fetch
37  Route::get('department/add', [DepartmentController::class, 'add'])->name('add');
38  Route::post('department/add_validation', [DepartmentController::class, 'add_validation'])->name('depar
39
```

- Check Output in the Browser

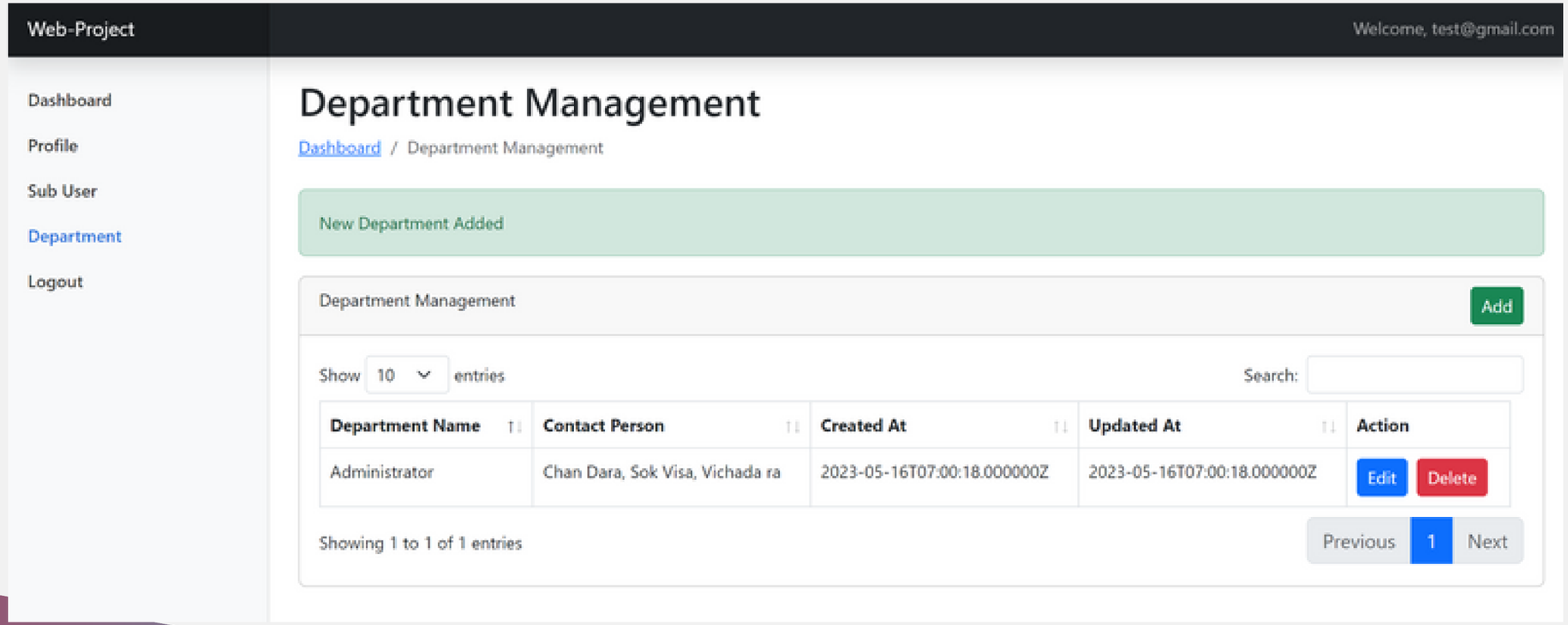


INSERT DATA INTO MYSQL TABLE USING LARAVEL ELOQUENT



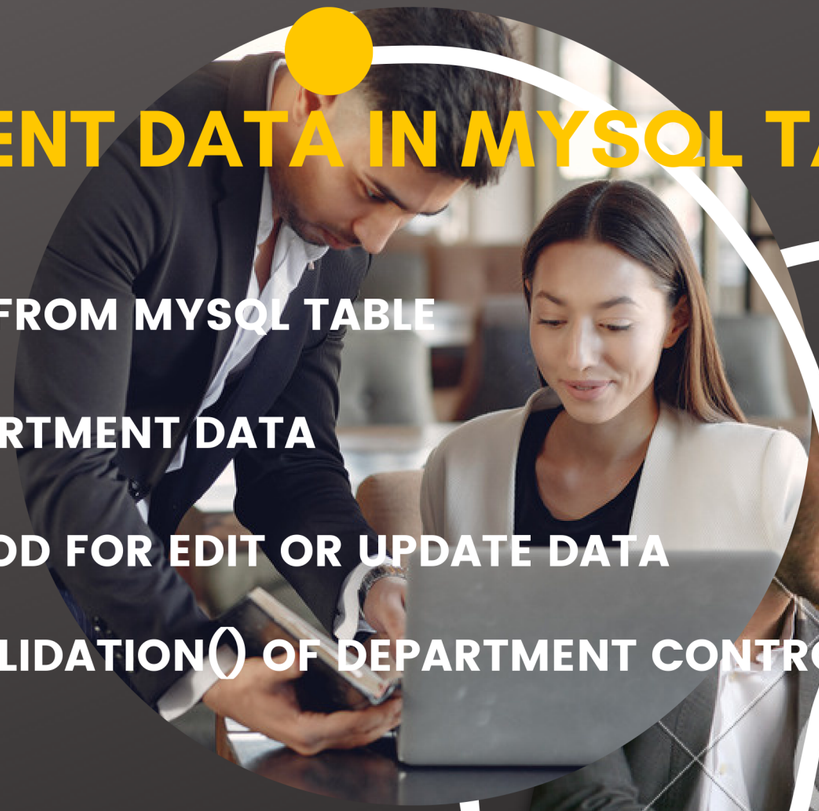
DISPLAY SUCCESS MESSAGE ON THE WEB PAGE AFTER INSERTING OF DATA

- After added success

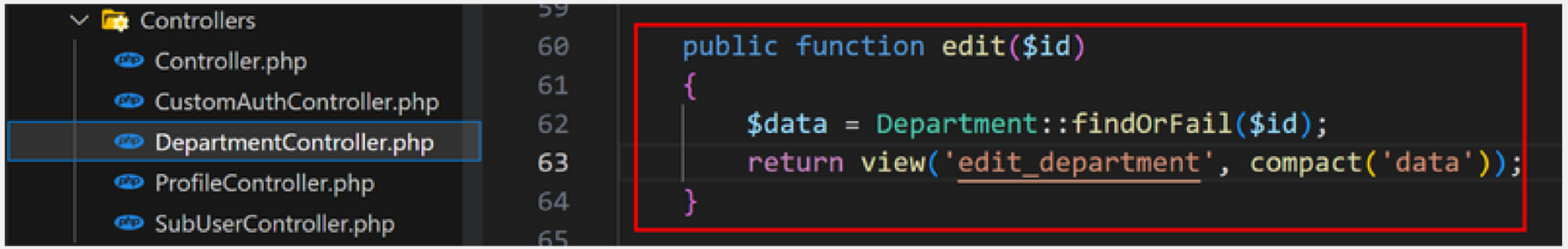


FETCH & EDIT DEPARTMENT DATA IN MYSQL TABLE

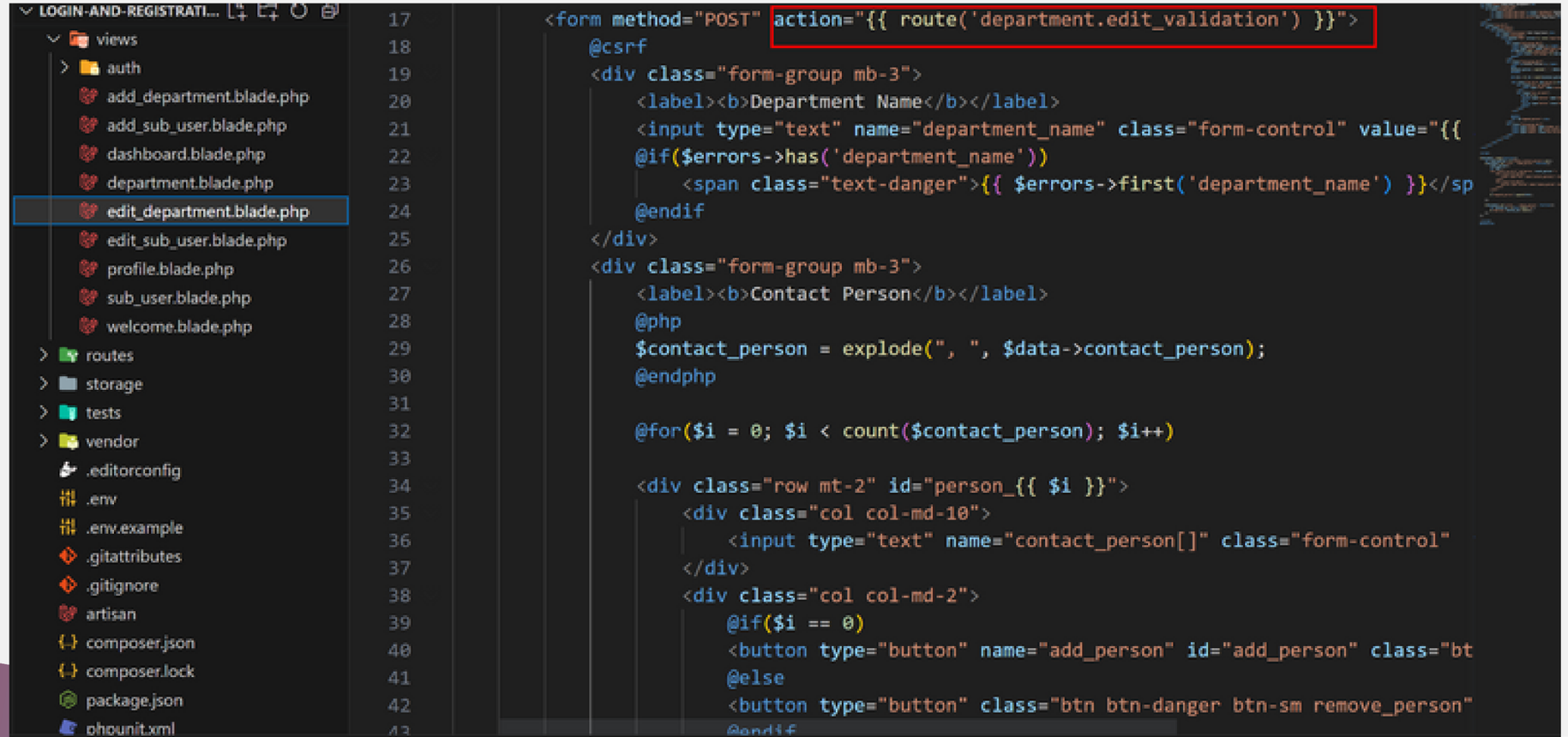
- FETCH SINGLE DEPARTMENT DATA FROM MYSQL TABLE
- FILL EDIT FORM WITH SINGLE DEPARTMENT DATA
- CREATE EDIT_VALIDATION() METHOD FOR EDIT OR UPDATE DATA
- SET ROUTE FOR EDIT() OR EDIT_VALIDATION() OF DEPARTMENT CONTROLLER METHOD
- CHECK OUTPUT IN THE BROWSER



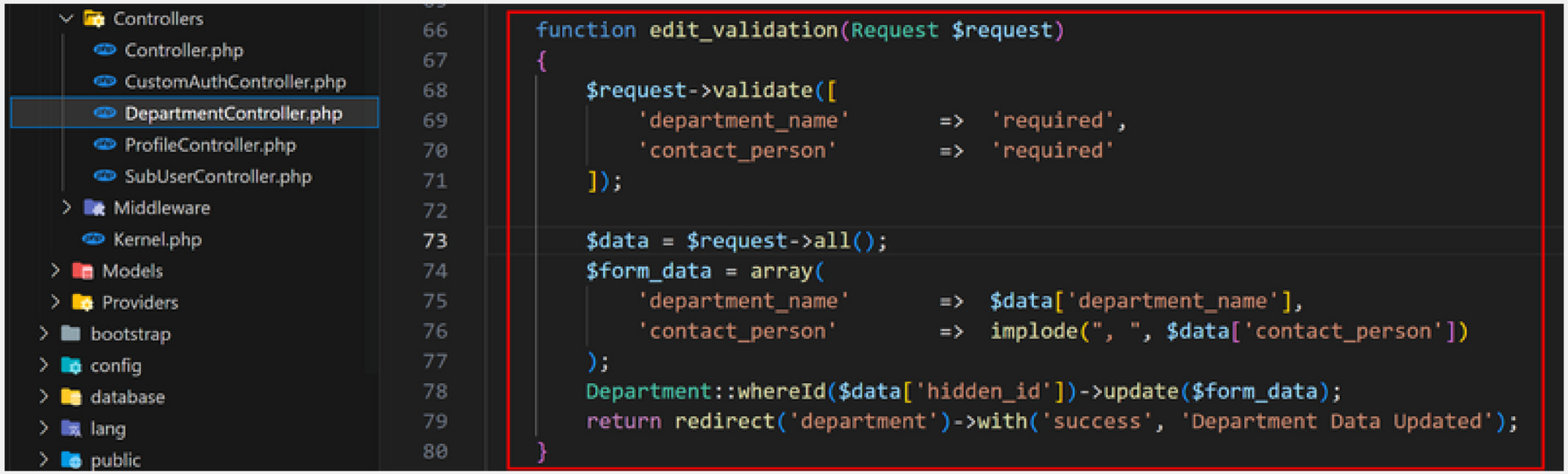
FETCH SINGLE DEPARTMENT DATA FROM MYSQL TABLE



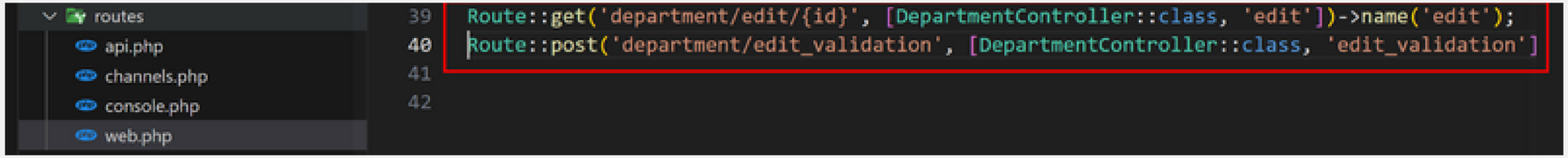
- Fill Edit form with Single Department Data



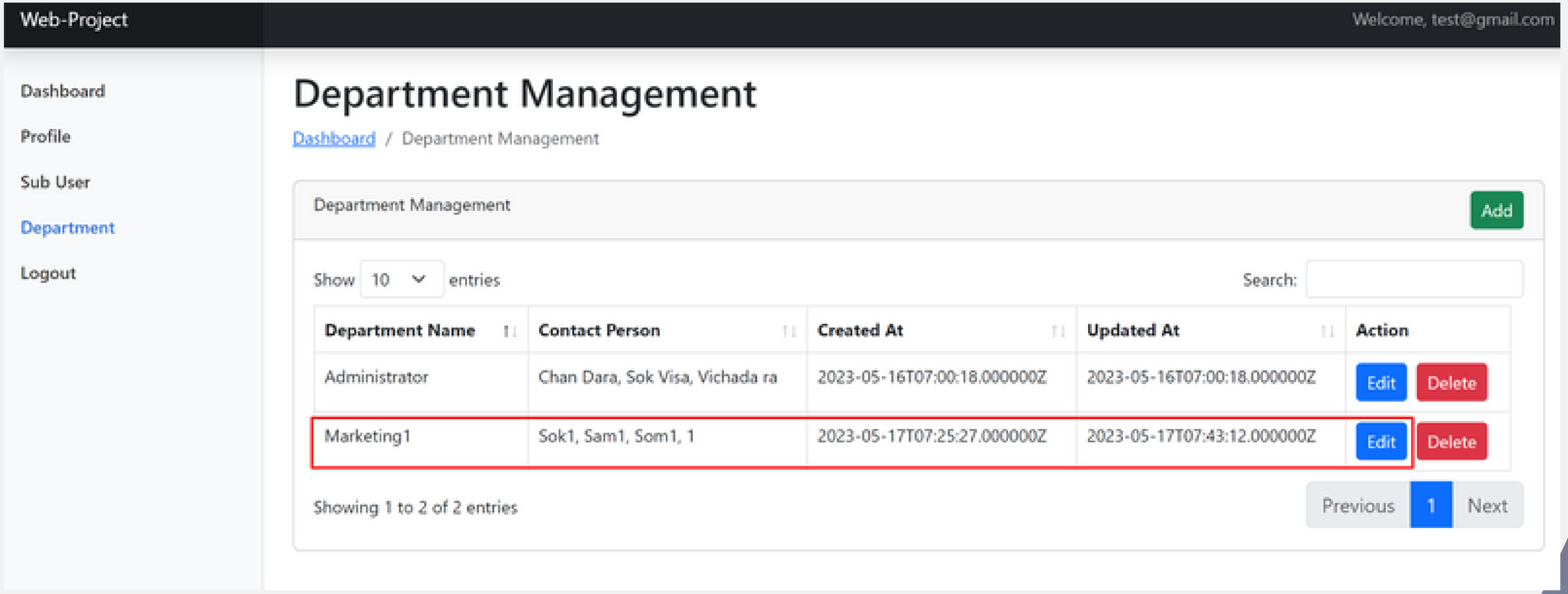
CREATE EDIT_VALIDATION() METHOD FOR EDIT OR UPDATE DATA



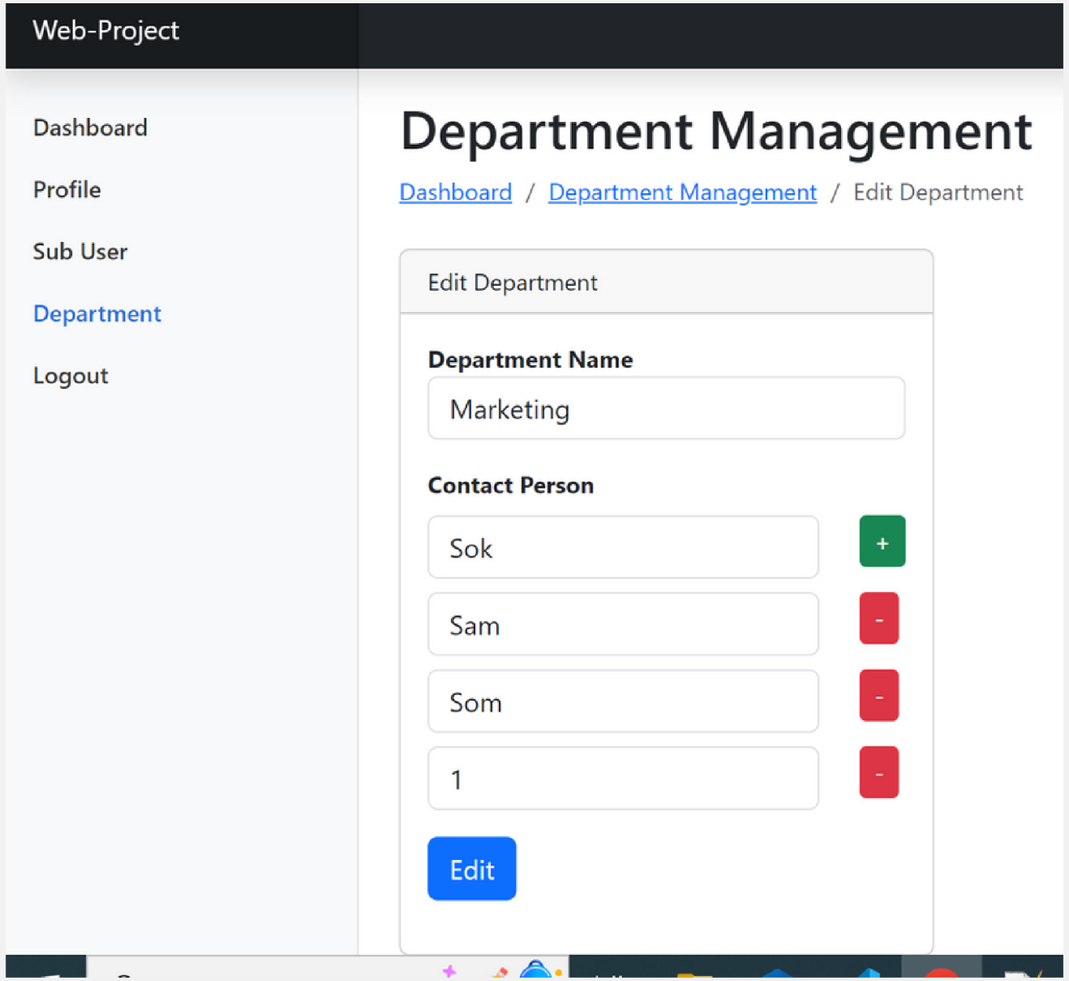
- Set Route for edit() or edit_validation() of Department Controller method

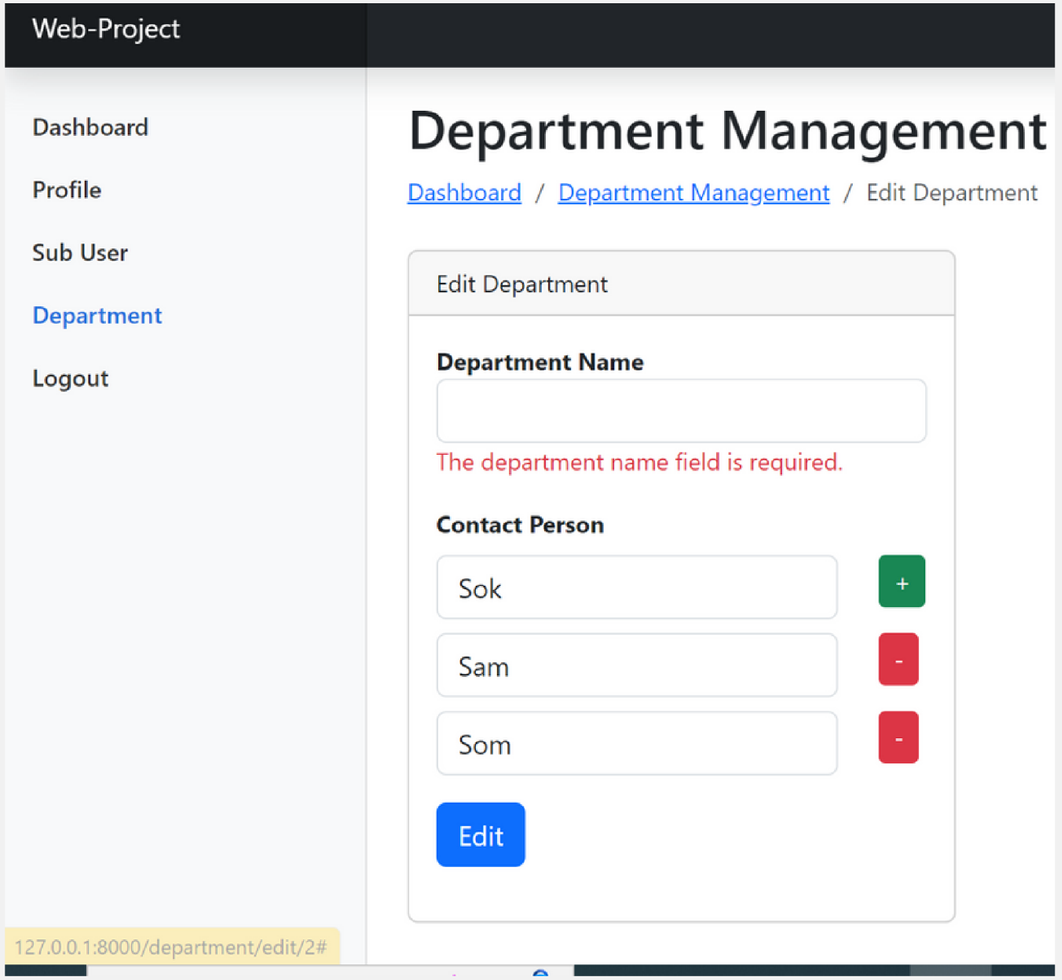


CHECK OUTPUT IN THE BROWSER

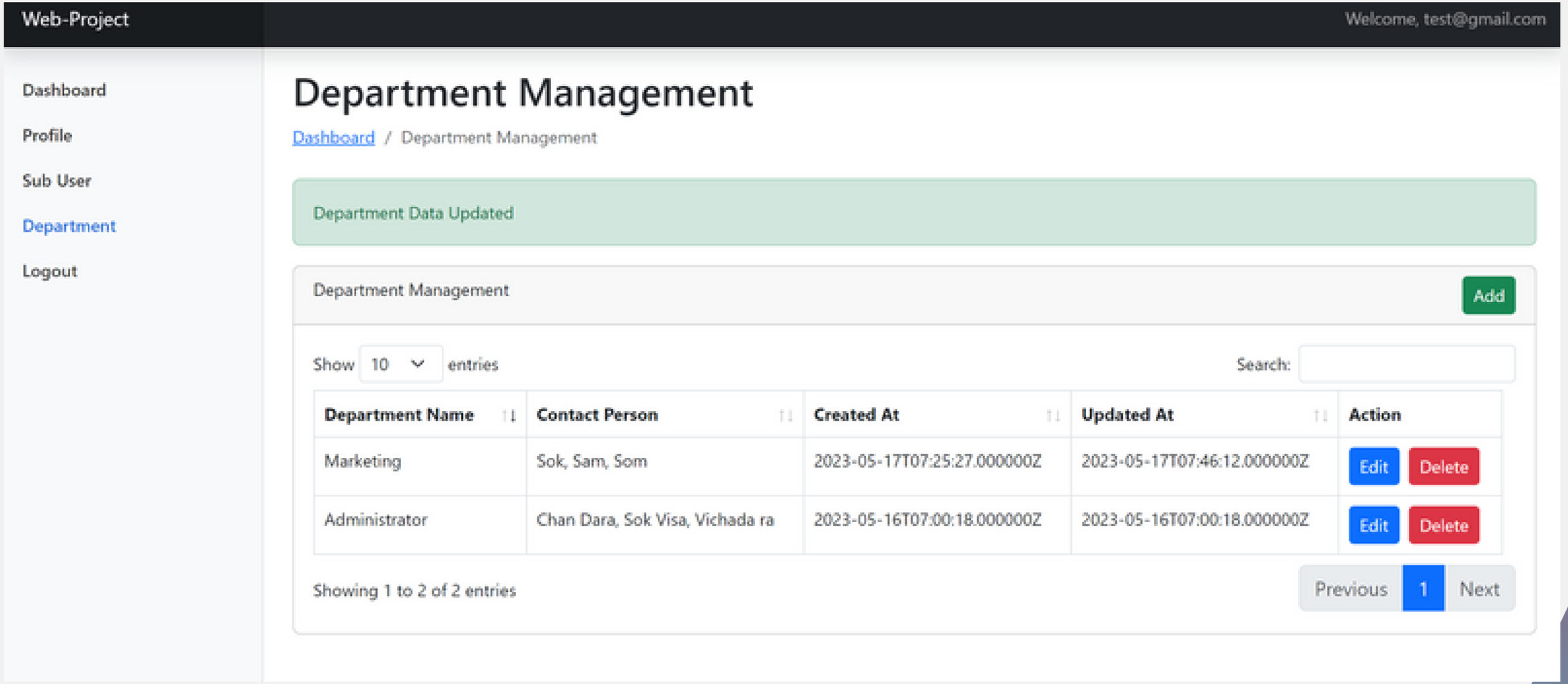


CHECK OUTPUT IN THE BROWSER (COUNT...)



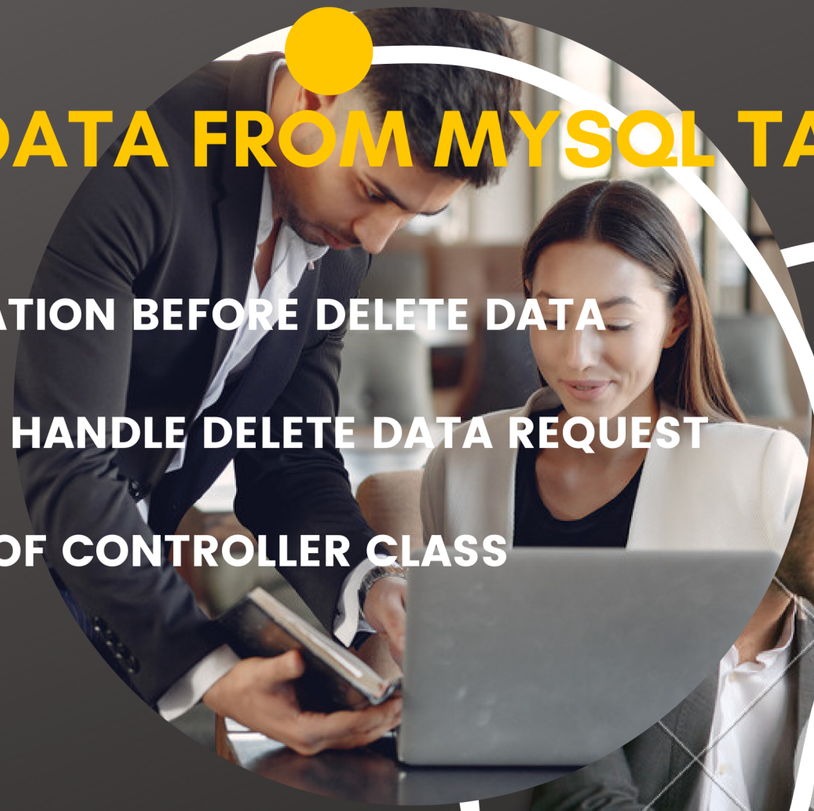


CHECK OUTPUT IN THE BROWSER (COUNT...)



REMOVE DEPARTMENT DATA FROM MYSQL TABLE

- WRITE JAVASCRIPT FOR CONFIRMATION BEFORE DELETE DATA
- CREATE CONTROLLER METHOD FOR HANDLE DELETE DATA REQUEST
- SET ROUTE FOR DELETE() METHOD OF CONTROLLER CLASS
- CHECK OUTPUT IN THE BROWSER

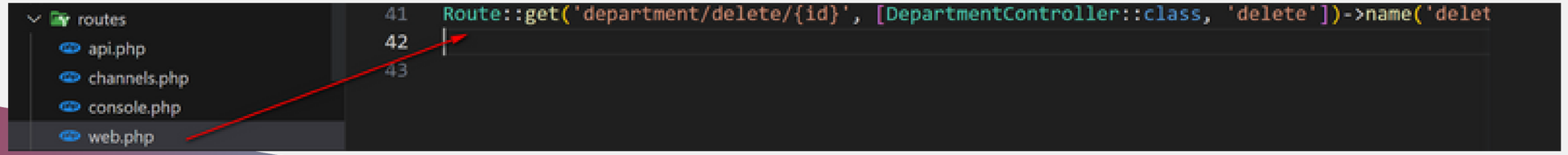


WRITE JAVASCRIPT FOR CONFIRMATION BEFORE DELETE DATA

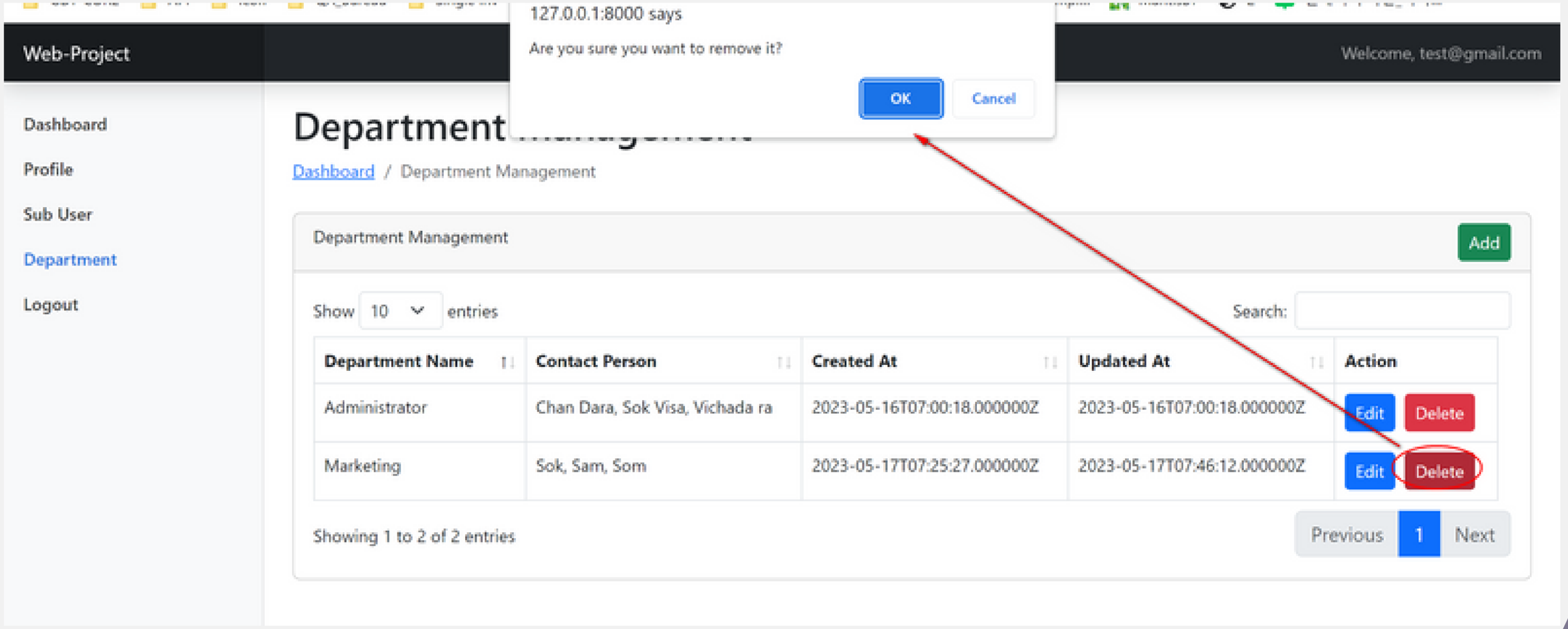
- Go to *department.blade.php*

- Create Controller method for handle delete data request

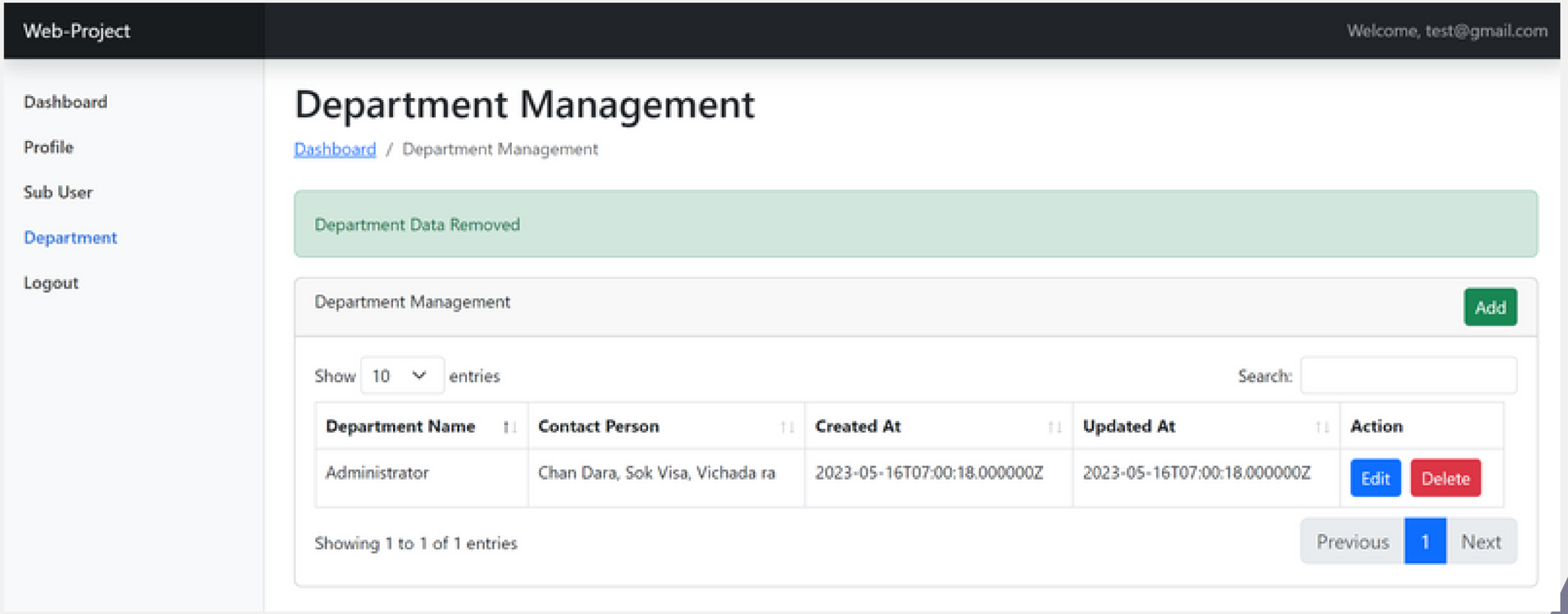
- Set Route for delete() method of Controller class



CHECK OUTPUT IN THE BROWSER

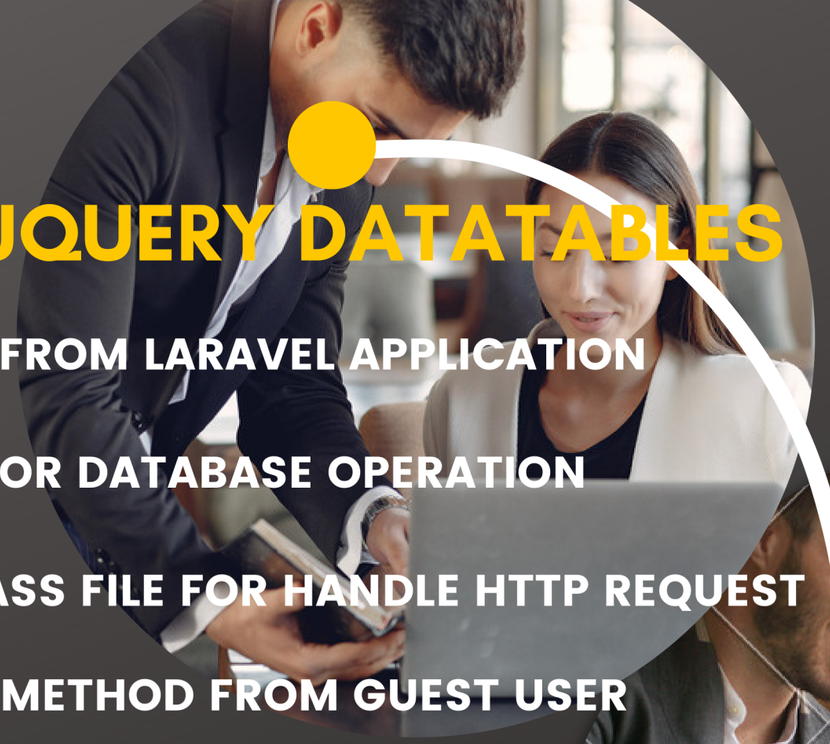


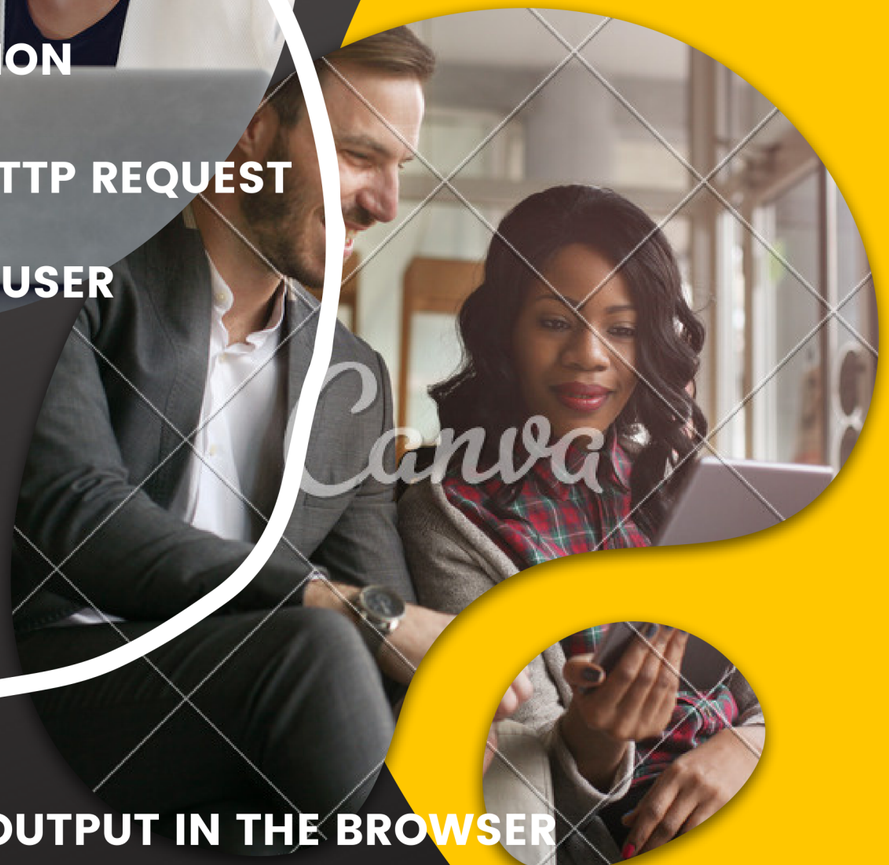
CHECK OUTPUT IN THE BROWSER (COUNT...)



LOAD VISITOR DATA IN JQUERY DATATABLES

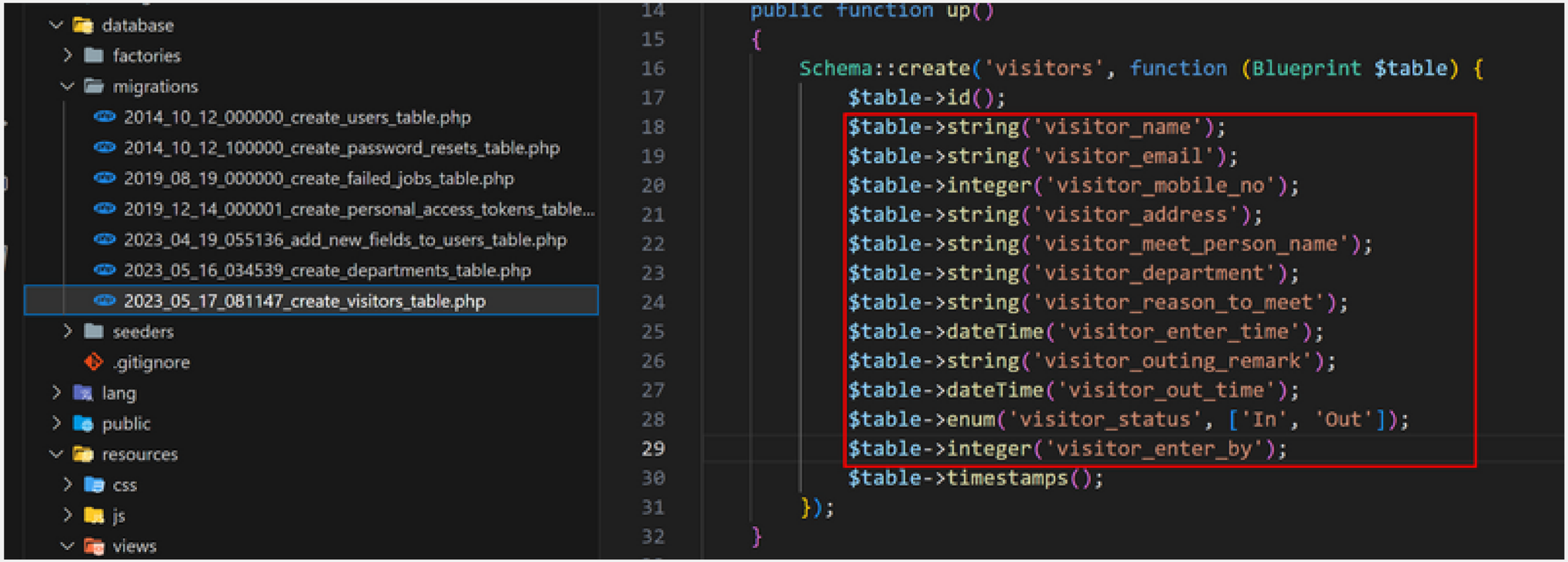
- CREATE VISITORS TABLE IN MYSQL FROM LARAVEL APPLICATION
- MAKE VISITOR MODEL CLASS FILE FOR DATABASE OPERATION
- CREATE VISITOR CONTROLLERS CLASS FILE FOR HANDLE HTTP REQUEST
- PREVENT ACCESS TO CONTROLLER METHOD FROM GUEST USER
- CREATE INDEX() METHOD IN VISITOR CONTROLLER FILE
- CREATE VIEW FILE WITH JQUERY CODE FOR DATATABLES
- WRITE PHP SCRIPT FOR LOAD DATA IN YAJRA DATATABLE
- ADD VISITOR WEB PAGE LINK AT MAIN DASHBOARD FILE
- SET ROUTE FOR VISITOR CONTROLLER METHOD & CHECK OUTPUT IN THE BROWSER





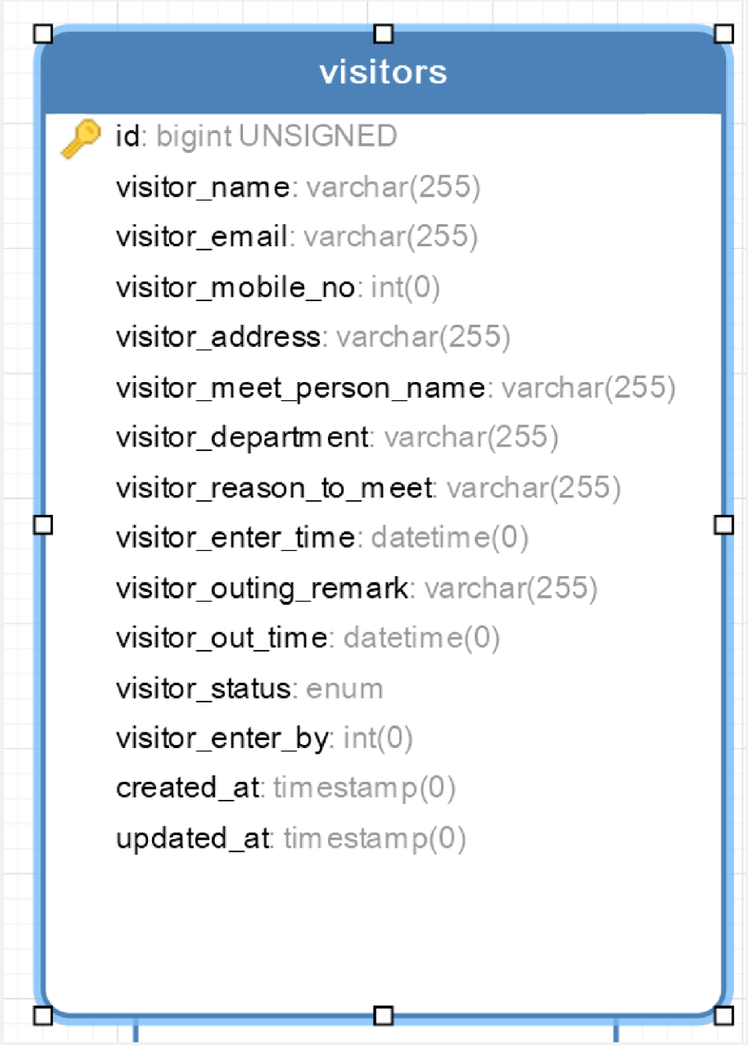
CREATE VISITORS TABLE IN MYSQL FROM LARAVEL APPLICATION

php artisan make:migration create_visitors_table



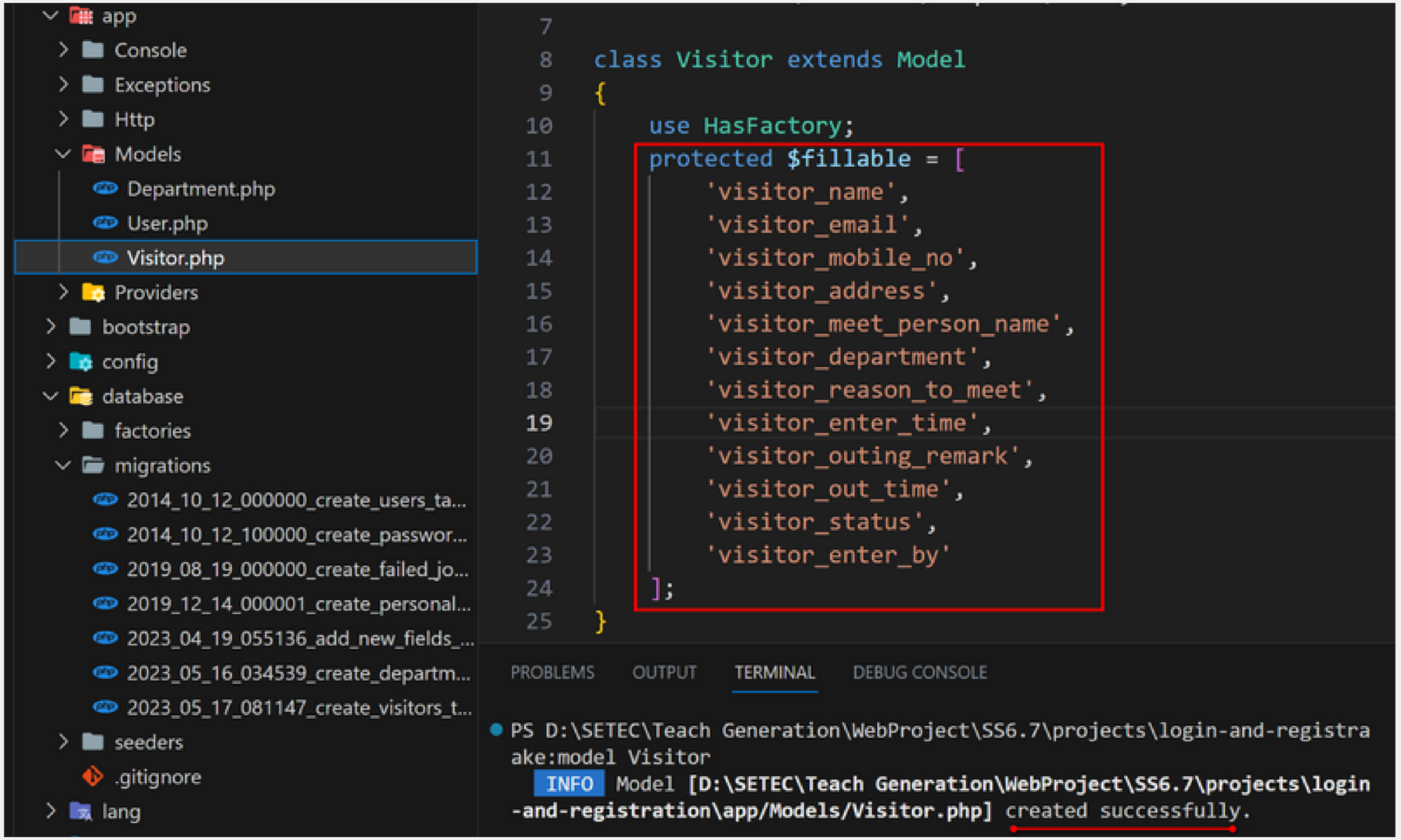
CREATE VISITORS TABLE IN MYSQL FROM LARAVEL APPLICATION (COUNT...)

php artisan migrate



MAKE VISITOR MODEL CLASS FILE FOR DATABASE OPERATION

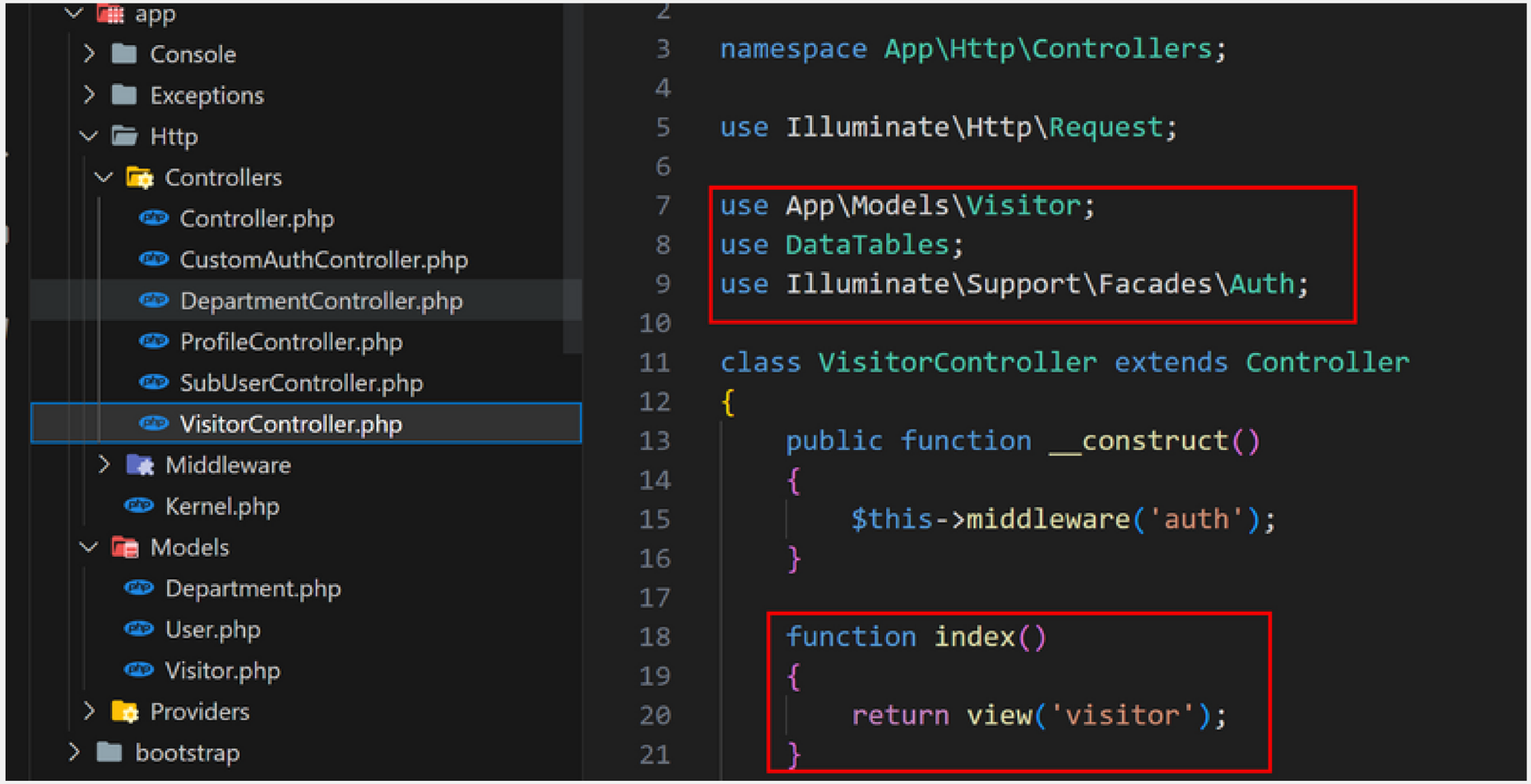
php artisan make:model Visitor



CREATE VISITOR CONTROLLERS CLASS FILE FOR HANDLE HTTP REQUEST

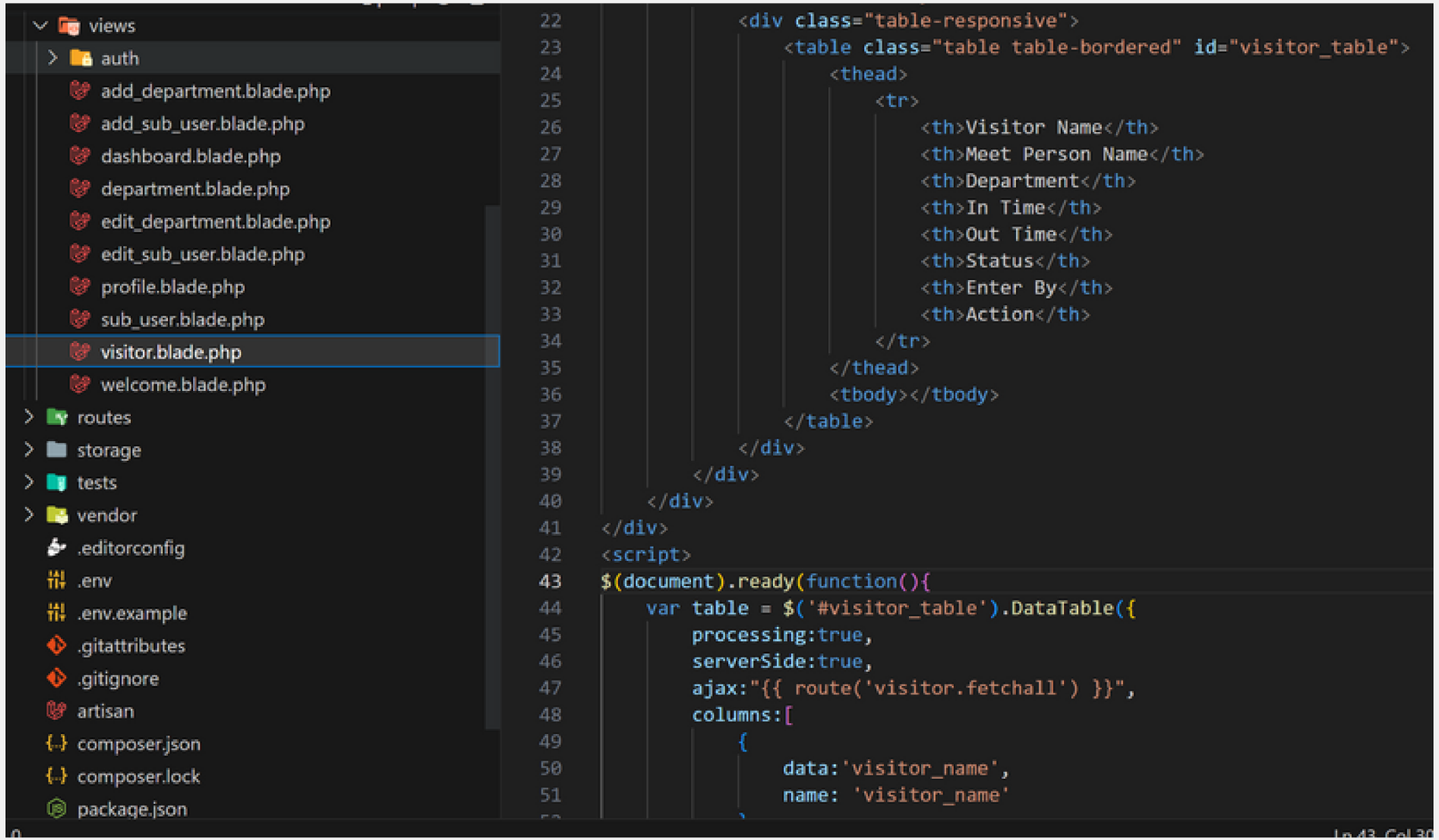
php artisan make:controller VisitorController

- Prevent Access to Controller method from Guest User

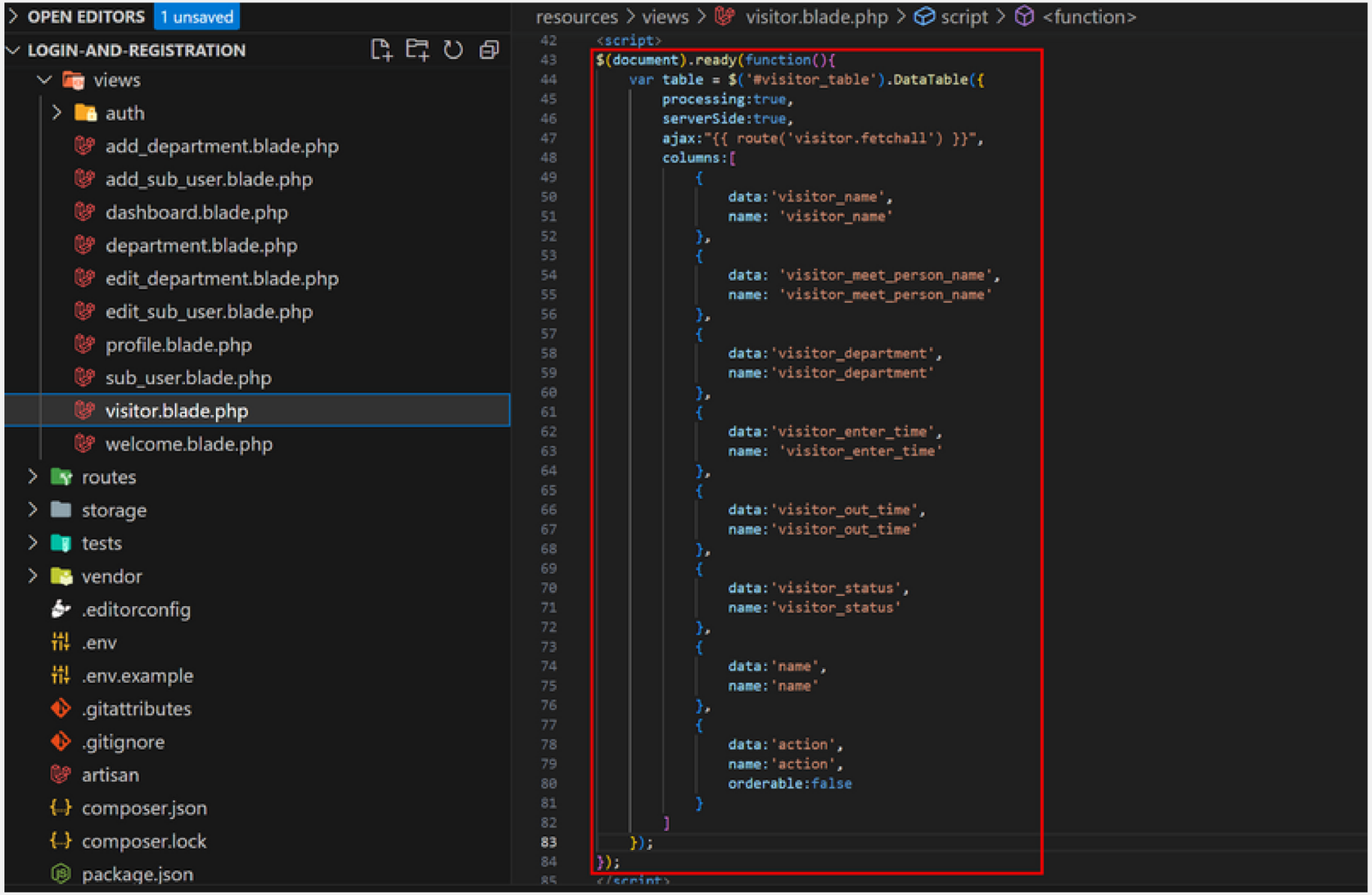


CREATE INDEX() METHOD IN VISITOR CONTROLLER FILE

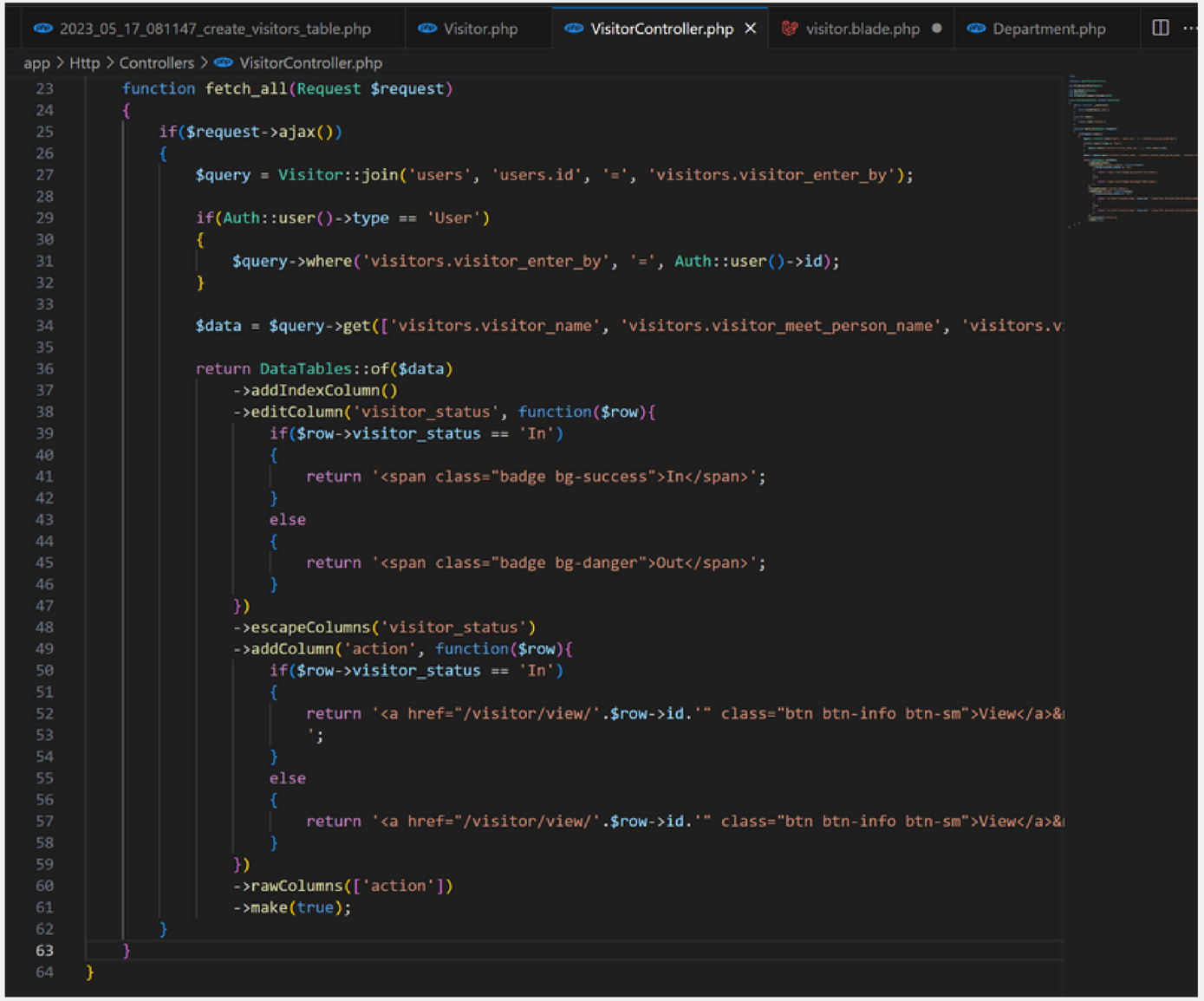
create *visitor.blade.php*



CREATE VIEW FILE WITH JQUERY CODE FOR DATATABLES



WRITE PHP SCRIPT FOR LOAD DATA IN YAJRA DATATABLE



ADD VISITOR WEB PAGE LINK AT MAIN DASHBOARD FILE

Set Route for Visitor Controller method

CHECK OUTPUT IN THE BROWSER