

Lab report no: 08

Name of the report: Php mysql CRUD applications.

Objectives: After completing this lab we will learn about:

- How to create,
- How to read,
- How to update
- How to delete in mysql databases with php laravel framework.

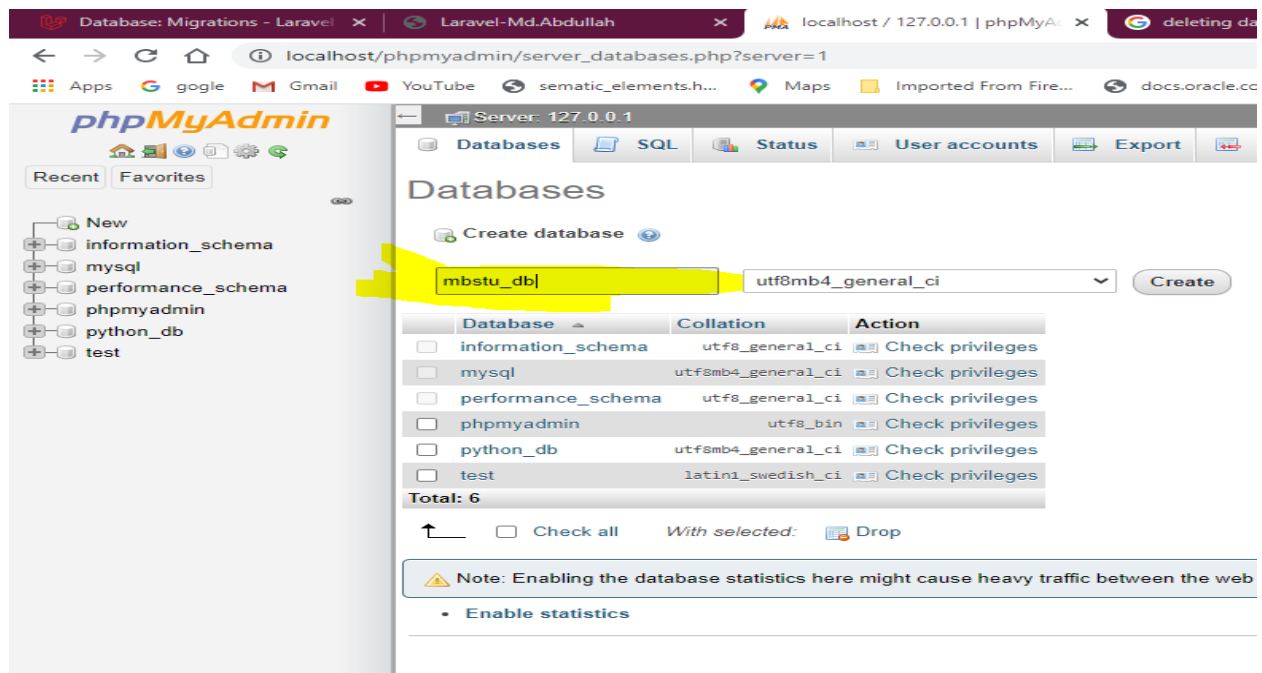
Theory:

What is CRUD

CRUD is an acronym for Create, Read, Update, and Delete. CRUD operations are basic data manipulation for database. When we are building APIs, we want our models to provide four basic types of functionality. The model must be able to Create, Read, Update, and Delete resources. Computer scientists often refer to these functions by the acronym CRUD. A model should have the ability to perform at most these four functions in order to be complete. If an action cannot be described by one of these four operations, then it should potentially be a model of its own.

The CRUD paradigm is common in constructing web applications, because it provides a memorable framework for reminding developers of how to construct full, usable models.

Creating database and tables:

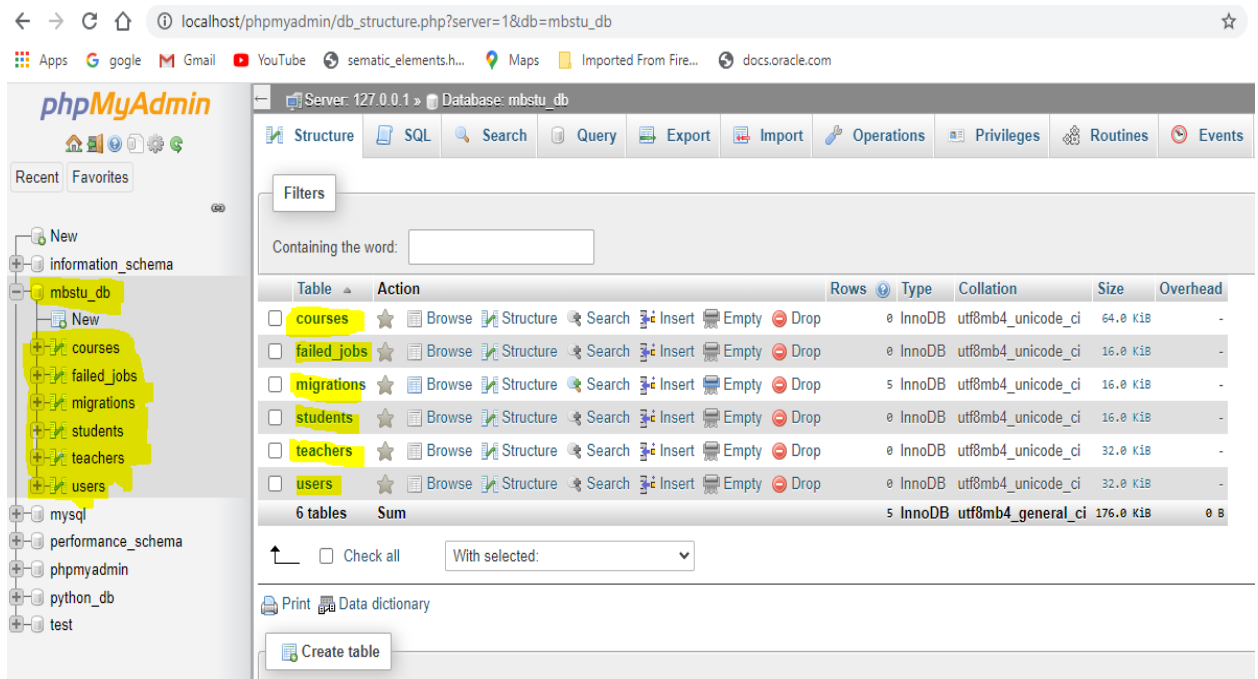
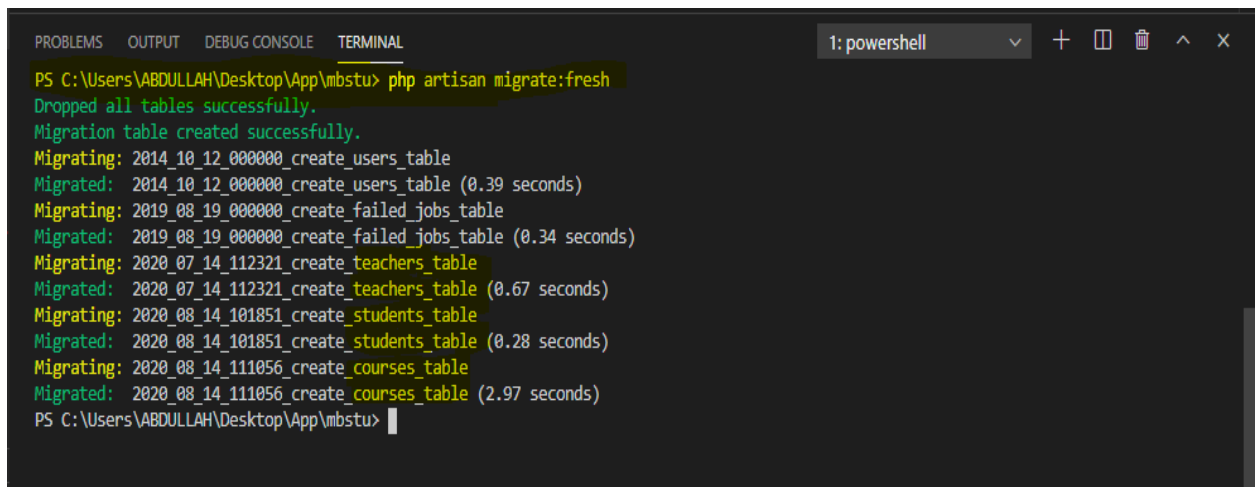
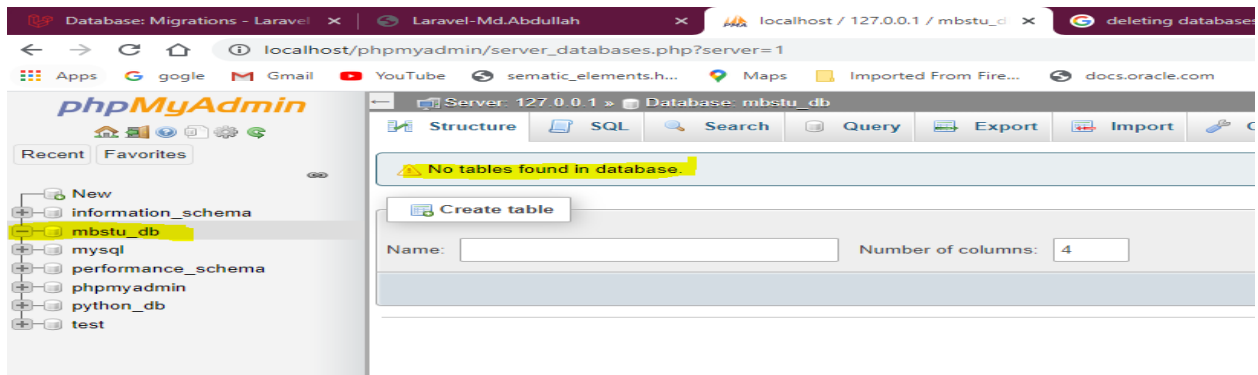


The screenshot shows the phpMyAdmin web interface. The 'Databases' tab is selected. In the 'Create database' section, the database name 'mbstu_db' is entered and highlighted in yellow. The collation is set to 'utf8mb4_general_ci'. Below this, a table lists existing databases:

Database	Collation	Action
<input type="checkbox"/> information_schema	utf8_general_ci	Check privileges
<input type="checkbox"/> mysql	utf8mb4_general_ci	Check privileges
<input type="checkbox"/> performance_schema	utf8_general_ci	Check privileges
<input type="checkbox"/> phpmyadmin	utf8_bin	Check privileges
<input type="checkbox"/> python_db	utf8mb4_general_ci	Check privileges
<input type="checkbox"/> test	latin1_swedish_ci	Check privileges

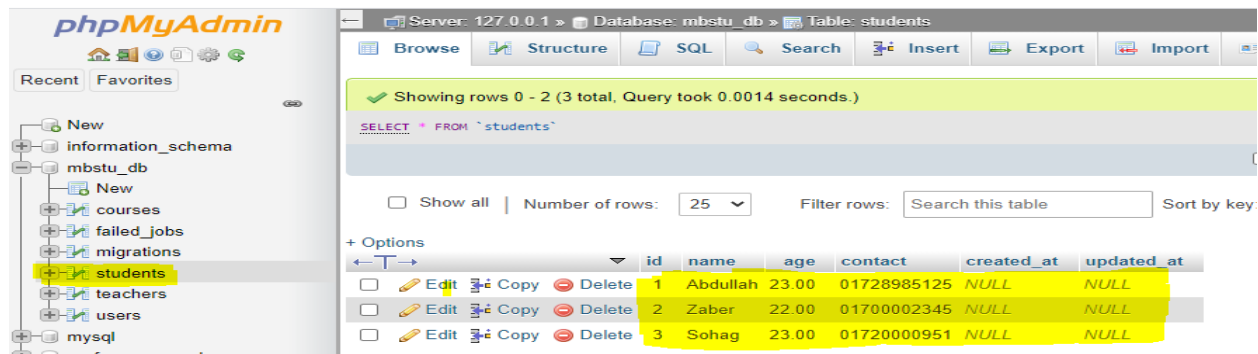
Total: 6

Below the table, there are options to 'Check all' and 'Drop' selected databases. A note at the bottom states: 'Note: Enabling the database statistics here might cause heavy traffic between the web'. There is a link to 'Enable statistics'.



I want to work with student table. For this first I need to insert data into table. So I insert the data into the table using the following command:

```
INSERT INTO `students`(`name`,`age`,`contact`) VALUES ('Abdullah',23,'01728985125')
```



The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible, with 'students' highlighted under 'mbstu_db'. The main panel shows the 'students' table with 3 rows of data. The table has columns: id, name, age, contact, created_at, and updated_at. The data rows are highlighted in yellow.

id	name	age	contact	created_at	updated_at
1	Abdullah	23.00	01728985125	NULL	NULL
2	Zaber	22.00	01700002345	NULL	NULL
3	Sohag	23.00	01720000951	NULL	NULL

Now create a route for accessing the students table data:

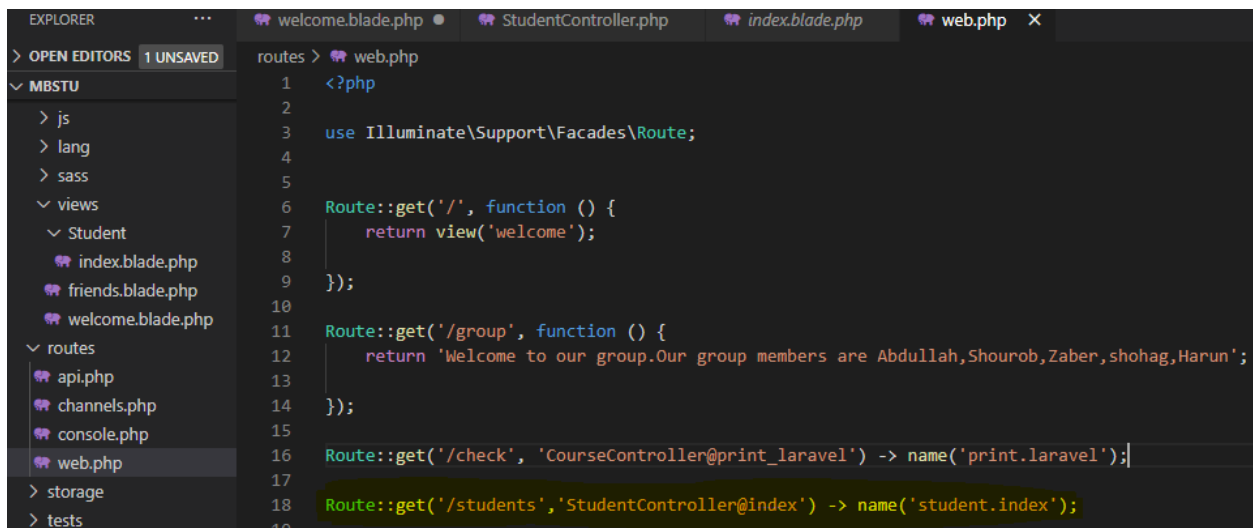
First I create a menu name STUDENTS in the home menu.

```
87 <div class="links">
88     <a href="https://nova.laravel.com">University</a>
89     <a href="https://forge.laravel.com">Department</a>
90     <a href="https://laracasts.com">Year</a>
91     <a href="https://laravel-news.com">semester</a>
92     <a href="https://github.com/laravel/laravel">About</a>
93     <a href="{{route('student.index')}}">Students</a>
94 </div>
95 </div>
96 </div>
97 </body>
98 </html>
```

MBSTU

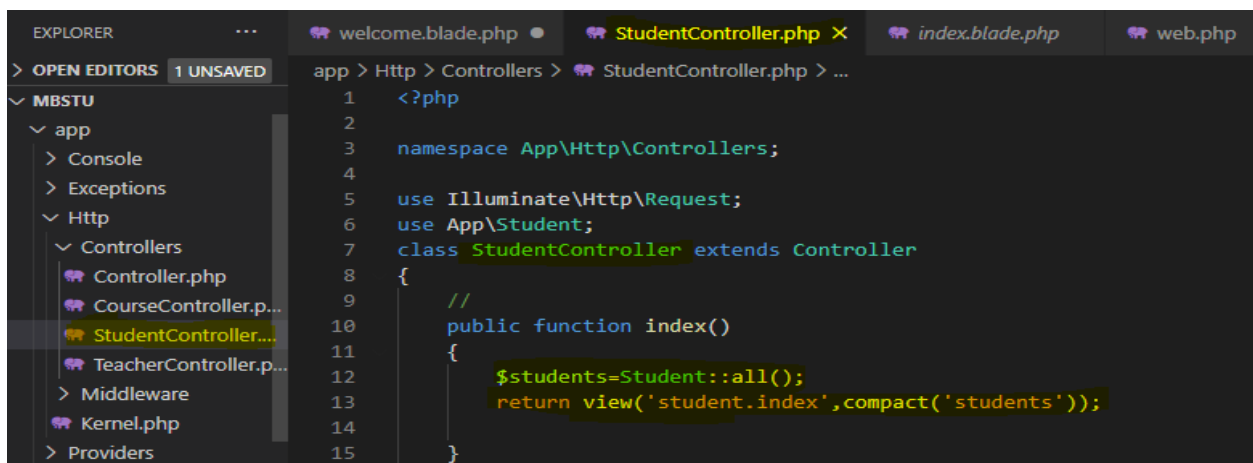
[UNIVERSITY](#)[DEPARTMENT](#)[YEAR](#)[SEMESTER](#)[ABOUT](#)[PRINT_LARAVEL](#)[STUDENTS](#)

Now I create a route of student in web.php:



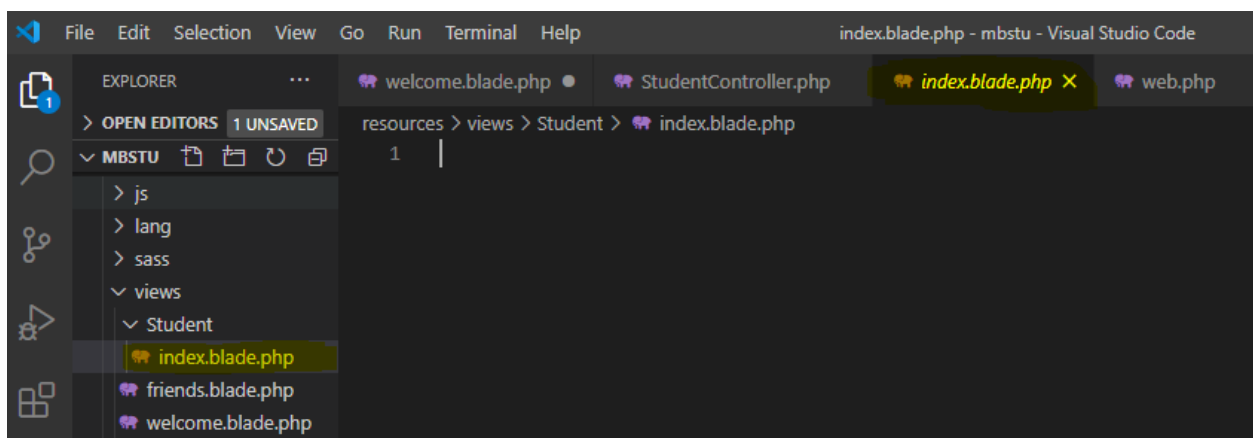
```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5
6 Route::get('/', function () {
7     return view('welcome');
8 });
9
10
11 Route::get('/group', function () {
12     return 'Welcome to our group.Our group members are Abdullah,Shourob,Zaber,shohag,Harun';
13 });
14
15
16 Route::get('/check', 'CourseController@print_laravel') -> name('print.laravel');
17
18 Route::get('/students','StudentController@index') -> name('student.index');
```

Setting up route functions in controller named StudentController:



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Student;
7 class StudentController extends Controller
8 {
9     //
10     public function index()
11     {
12         $students=Student::all();
13         return view('student.index',compact('students'));
14     }
15 }
```

I also create a index.blade.php file in student folder under resources:



```
1 |
```

Create and Read:

To create resources in a REST environment, we most commonly use the HTTP POST method. POST creates a new resource of the specified resource type. To read resources in a REST environment, we use the GET method. Reading a resource should never change any information - it should only retrieve it. To read a database, table and access the table entries from web need to create a route, controller function and blade.php file.

Route:

We need to create a route/url that will take our data, store it and will show into the webpage. Route are generally created at web.php file. Here the web.php file,

```
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome');
});

Route::get('/group', function () {
    return 'Welcome to our group.Our group members are Abdullah,Shouroob,Zaber,shohag,Harun';
});

Route::get('/check', 'CourseController@print_laravel') -> name('print.laravel');
Route::get('/students', 'StudentController@index') -> name('student.index');
Route::post('/new_students', 'StudentController@store') -> name('student.store');
```

Controller function:

Now need to define the store function in Controller. Because we work with students table we need to define the function in StudentController. Here the code in Student controller to store the data.

```
<?php
```

```

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Student;
class StudentController extends Controller
{
    //
    public function index()
    {
        $students=Student::all();
        return view('student.index',compact('students'));
    }
    public function store(Request $request)
    {
        Student::create([
            'name'=>$request-> name,
            'age'=> $request-> age,
            'contact'=>$request-> contact,

        ]);
        return redirect() -> route('student.index');
    }
}

```

Index.blade.php file:

We need to create a **index.blade.php** file and need to write a html file on it to show our data on the webpage. Student table has three fields name, age and contact. First we need to create a form by which data will be taken and will be shown on the table. The index.blade.php file is given below:

```

<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Laravel-Md.Abdullah</title>

    <!-- Fonts -->

```

```
<link href="https://fonts.googleapis.com/css?family=Nunito:200,600" rel="stylesheet">
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">

<!-- Styles -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKAizap3H66lZH81PoY1FhbGU+6BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-B4gt1jrGC7Jh4AgTPSdU0BvffO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV" crossorigin="anonymous"></script>

</head>

<body>
<br>
<br>
<br>
<div class='container-fluid'>
  <div class='row'>
    <div class="col-md-4">
      <div class='card'>
        <div class='card-header'>
          Add new student
        </div>
      </div>
      <div class='card-body'>
        <form action="{{route('student.store')}}" method=" POST">
          @csrf
          <div class="form-group">
            <label for="name">Name</label>
            <input type="text" class="form-control" id="name" name="name" aria-describedby="nameHelp" placeholder="Enter name">
          </div>
          <div class="form-group">

```

```

        <label for="age">Age</label>
        <input type="number" class="form-
control" id="age" placeholder="age">
    </div>
    <div class="form-group">
        <label for="contact">Contact</label>
        <input type="text" class="form-
control" id="contact" name="contact" aria-
describedby="contactHelp" placeholder="Enter contact">
    </div>
    <button type="submit" class="btn btn-
primary">Submit</button>
</form>
</div>
</div>
<div class="col-md-8">
    <div class='card'>
        <div class='card-header'>
            Existing student
        </div>
        </div>
        <div class='card-body'>
            <table class="table">
                <thead>
                    <tr>
                        <th scope="col">#</th>
                        <th scope="col">Name</th>
                        <th scope="col">Age</th>
                        <th scope="col">Contact</th>
                    </tr>
                </thead>
                <tbody>
                    @foreach($students as $student)
                        <tr>
                            <td>{{ $student ->id }}</td>
                            <td>{{ $student ->name }}</td>
                            <td>{{ $student ->age }}</td>
                            <td>{{ $student ->contact }}</td>
                        </tr>
                    @endforeach
                </tbody>
            </table>
        </div>
    </div>
</div>

```



```
</div>
</body>

</html>
```

Read data from webpage:

What is CRUD? | Codecademy

localhost / 127.0.0.1 / mbstu_db

Laravel-Md.Abdullah

+

localhost:8000/students

Apps google Gmail YouTube semantic_elements.h... Maps Imported From Fire... docs.oracle.com

Add new student

Existing student

Name

Enter name

Age

age

Contact

Enter contact

Submit

#	Name	Age	Contact
1	Abdullah	23	01728985125
2	Zaber	22	01700002345
3	Sohag	23	01720000951

Read data from web page and inserting data from webpage:

What is CRUD? | Codecademy

localhost / 127.0.0.1 / mbstu_db

Laravel-Md.Abdullah

+

localhost:8000/students

Apps google Gmail YouTube semantic_elements.h... Maps Imported From Fire... docs.oracle.com

Add new student

Existing student

Name

shourob

Age

24

Contact

01700221133

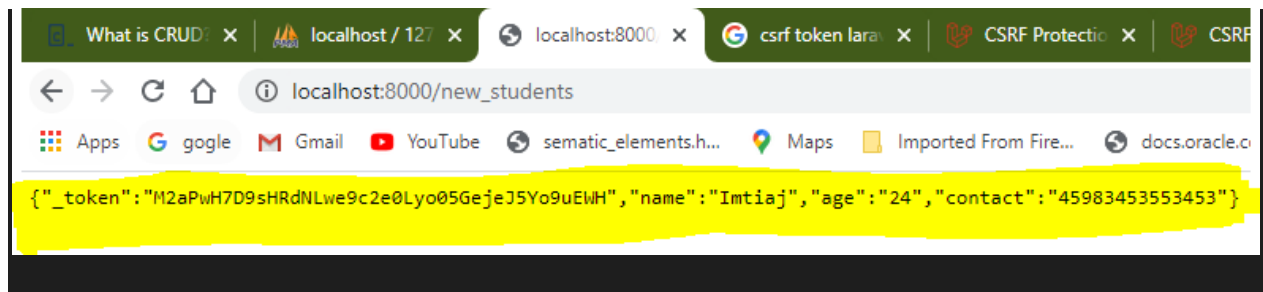
Submit

#	Name	Age	Contact
1	Abdullah	23	01728985125
2	zaber	23	01700002345
3	sohag	23	01720000951
4	shourob	24	01700221133

CSRF:

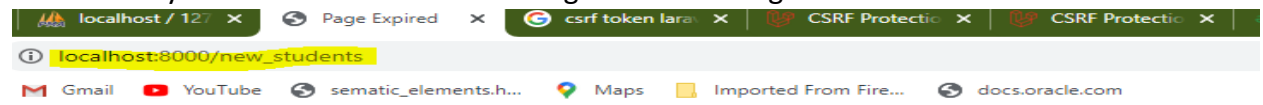
We need to provide the `@csrf` token at the every form that we create like following. It is a laravel token it is not a html function or class. Laravel makes it easy to protect application from cross-site request forgery (CSRF) attacks. Cross-site request forgeries are a type of malicious exploit whereby unauthorized commands are performed on behalf of an authenticated user. Laravel automatically generates a CSRF "token" for each active user session managed by the application. This token is used to verify that the authenticated user is the one actually making the requests to the application. Every computer generate different CSRF token.

```
<form action="{{route('student.store')}}" method="post">
    @csrf
    <div class="form-group">
        <label for="name">Name</label>
        <input type="text" class="form-
control" id="name" name="name" aria-
describedby="nameHelp" placeholder="Enter name">
    </div>
    <div class="form-group">
        <label for="age">Age</label>
        <input type="number" name='age' class="form-
control" id="age" placeholder="age">
    </div>
    <div class="form-group">
        <label for="contact">Contact</label>
        <input type="text" class="form-
control" id="contact" name="contact" aria-
describedby="contactHelp" placeholder="Enter contact">
    </div>
    <button type="submit" class="btn btn-
primary">Submit</button>
</form>
```



Error without CSRF:

When we try to insert a new value we will get the following error If we don't use the CSRF :



419 | Page Expired

Update:

Update means updating the existing elements of the database such as rename, change the current value etc. For updating the database we need to create a route, controller function and blade.php file. So we first create a route.

Route:

For updating the existing database we need to create a route for accessing the databases. Here I create a route name edit and update with update function.

```
StudentController.php  index.blade.php  web.php  edit.blade.php  Student.php
routes > web.php > ...
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5
6  Route::get('/', function () {
7      return view('welcome');
8  });
9
10
11 Route::get('/group', function () {
12     return 'Welcome to our group.Our group members are Abdullah,Shourob,Zaber,shohag,Harun';
13 });
14
15
16 Route::get('/check', 'CourseController@print_laravel') -> name('print.laravel');
17 Route::get('/students','StudentController@index') -> name('student.index');
18 Route::post('/new_students','StudentController@store') -> name('student.store');
19 Route::get('/edit/student/{id}','StudentController@edit') -> name('student.edit');
20 Route::post('/update/student','StudentController@update') -> name('student.update');
```

Controller function:

Here I create a controller function edit and update to updating the table entries.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Student;
class StudentController extends Controller
{
    //
    public function index()
    {
        $students=Student::all();
        return view('student.index',compact('students'));
    }
}
```

```

public function store(Request $request)
{
    Student::create([
        'name'=>$request-> name,
        'age'=> $request-> age,
        'contact'=>$request-> contact,

    ]);
    return redirect() -> route('student.index');
}
public function edit($id)
{
    $student=Student::findOrFail($id);
    return view('student.edit',compact('student'));

}
public function update(Request $request)
{
    $student = Student :: findOrFail($request->student_id);
    $student ->name =$request -> name;
    $student ->age =$request -> age;
    $student ->contact =$request -> contact;
    $student -> update();
    return redirect() -> route('student.index');

}

```

Blade.php file:

Here I create a blade.php file in student folder under the views named edit.blade.php to edit the entries.

```

<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Edit Student</title>

    <!-- Fonts -->
    <link href="https://fonts.googleapis.com/css?family=Nunito:200,600" rel="stylesheet">

```

```

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5
.2/css/bootstrap.min.css" integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="an
onymous">

<!-- Styles -->
<script src="https://code.jquery.com/jquery-
3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpmoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="an
onymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.mi
n.js" integrity="sha384-
9/reFTGAw83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN" crossorigin="an
onymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.
min.js" integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUt0Bvf08shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV" crossorigin="an
onymous"></script>
</head>
<body>
  <br>
  <br>
  <br>
  <div class='container-fluid'>
    <div class='row'>
      <div class="col-md-4">
        <div class='card'>
          <div class='card-header'>
            Edit student
          </div>
        </div>
        <div class='card-body'>
          <form action="{{route("student.update")}}" method="post">
            @csrf
            <input type="hidden" name='student_id' value="{{ $student
->id }}">
            <div class="form-group">
              <label for="name">Name</label>
              <input type="text" class="form-
control" id="name" name="name" aria-describedby="nameHelp" value="{{ $student -
>name }}">
            </div>
            <div class="form-group">
              <label for="age">Age</label>

```

```

        <input type="number" name='age' class="form-
control" id="age" value="{{ $student ->age }}">
    </div>
    <div class="form-group">
        <label for="contact">Contact</label>
        <input type="text" class="form-
control" id="contact" name="contact" aria-
describedby="contactHelp" value="{{ $student ->contact }}">
    </div>
    <button type="submit" class="btn btn-
primary">Update</button>
    </form>

</div>
</div>
</div>
</div>
</body>

</html>

```

Update output:

localhost / 127.0.0.1 / mbstu_db

New Student

localhost:8000/students

Apps

google

Gmail

YouTube

semantic_elements.h...

Maps

Imported From Fire...

docs.oracle.com

New student

Name

Enter name

Age

age

Contact

Enter contact

Submit

Existing student

#	Name	Age	Contact	Action
1	Abdullah	23	01728985125	<div>Edit</div> <div>Delete</div>
2	zaber	23	01700002345	<div>Edit</div> <div>Delete</div>
3	sohag	23	0172000951	<div>Edit</div> <div>Delete</div>
4	shourobo	21	00000000000	<div>Edit</div> <div>Delete</div>

Before update the value of id=4 is:

#	Name	Age	Contact
1	Abdullah	23	01728985125
2	zaber	23	01700002345
3	sohag	23	0172000951
4	shourob	24	01700221133

Delete:

The CRUD operation Delete corresponds to the HTTP method DELETE. It is used to remove a resource from the system. To delete a table entries first need to create a route. So the route for deleting is:

Route:

```
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome');
});

Route::get('/group', function () {
    return 'Welcome to our group.Our group members are Abdullah,Shourob,Zaber,sho
hag,Harun';
});

Route::get('/check', 'CourseController@print_laravel') -> name('print.laravel');
```

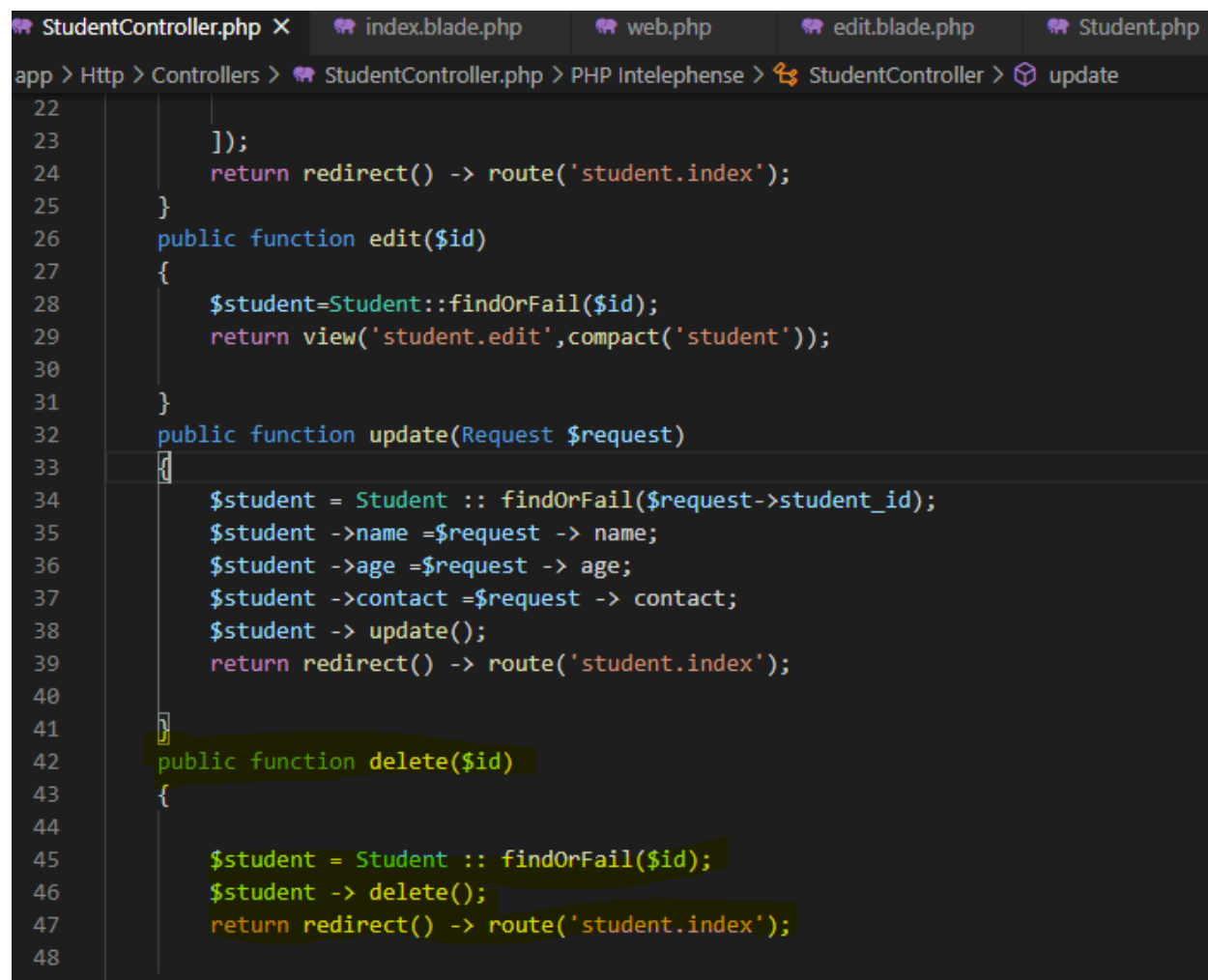


```

Route::get('/students','StudentController@index') -> name('student.index');
Route::post('/new_students','StudentController@store') -> name('student.store');
Route::get('/edit/student/{id}','StudentController@edit') -
> name('student.edit');
Route::post('/update/student','StudentController@update') -
> name('student.update');
Route::get('/delete/student/{id}','StudentController@delete') -
>name('student.delete');

```

Controller function:



```

StudentController.php X index.blade.php web.php edit.blade.php Student.php
app > Http > Controllers > StudentController.php > PHP Intelephense > StudentController > update
22
23     });
24     return redirect() -> route('student.index');
25 }
26 public function edit($id)
27 {
28     $student=Student::findOrFail($id);
29     return view('student.edit',compact('student'));
30 }
31 }
32 public function update(Request $request)
33 {
34     $student = Student :: findOrFail($request->student_id);
35     $student ->name =$request -> name;
36     $student ->age =$request -> age;
37     $student ->contact =$request -> contact;
38     $student -> update();
39     return redirect() -> route('student.index');
40 }
41 }
42 public function delete($id)
43 {
44
45     $student = Student :: findOrFail($id);
46     $student -> delete();
47     return redirect() -> route('student.index');
48 }

```

The index.blade.php file that was created before now need to modified in the deleted filed.

```
<td>

        <a class="btn btn-
info" href="{{route('student.edit',$student->id)}}">Edit</a>
        <a class="btn btn-
danger" href="{{route('student.delete',$student->id)}}">Delete</a>
    </td>
```

Deleting the Element:

We have four elements as follows:

The screenshot shows a web browser at localhost:8000/students. On the left, there is a form titled "Add new student" with fields for Name (shourob), Age (24), and Contact (01700221133), and a Submit button. On the right, there is a table titled "Existing student" with columns #, Name, Age, and Contact. The table contains four rows, with the last row (ID 4, Name shourob, Age 24, Contact 01700221133) highlighted in yellow.

#	Name	Age	Contact
1	Abdullah	23	01728985125
2	zaber	23	01700002345
3	sohag	23	0172000951
4	shourob	24	01700221133

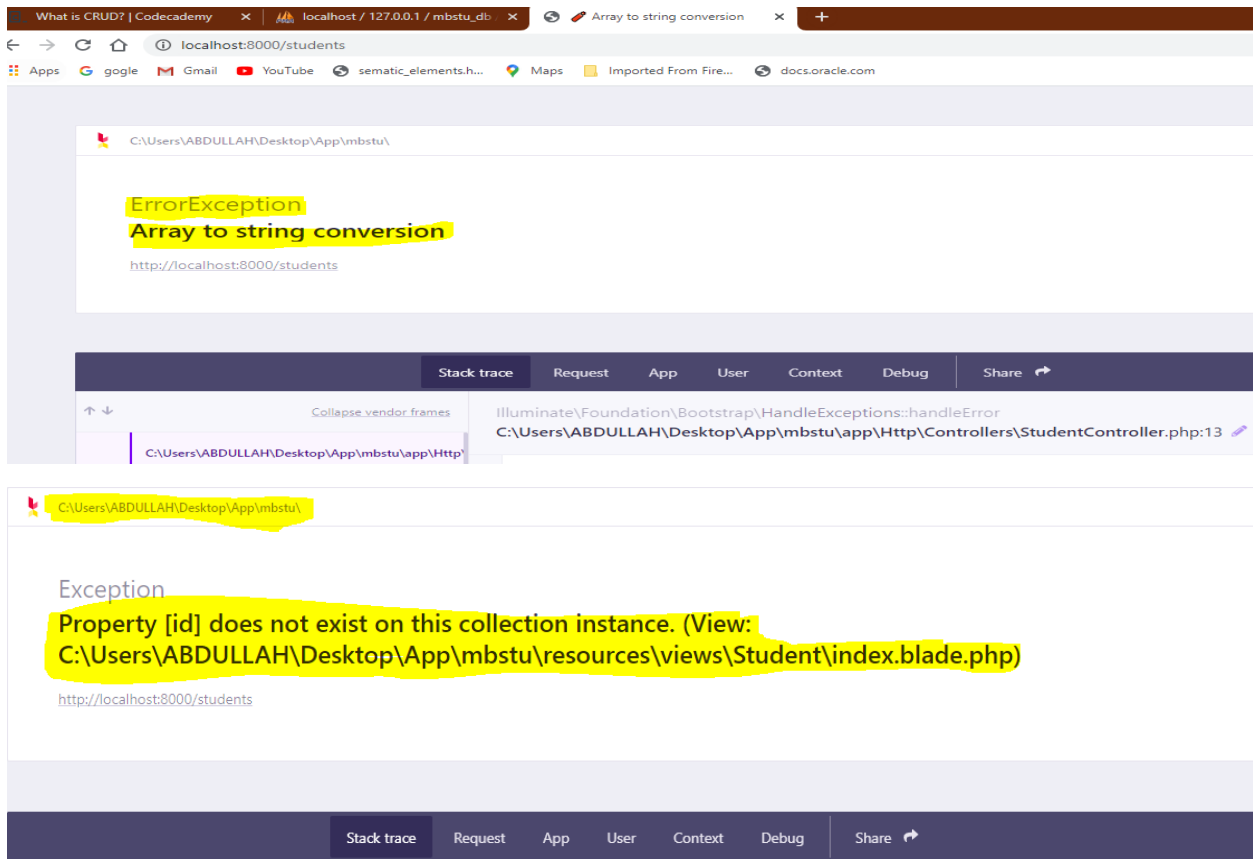
So we want to delete the element shohag, zaber. Now the output is,

The screenshot shows the same web application after deleting two students. The "Add new student" form is now titled "New student" and has empty fields for Name, Age, and Contact, with a Submit button. The "Existing student" table now has two rows: ID 1 (Abdullah, Age 23, Contact 01728985125) and ID 4 (shourob, Age 21, Contact 0000000000). Each row has "Edit" and "Delete" buttons in the "Action" column.

#	Name	Age	Contact	Action
1	Abdullah	23	01728985125	Edit Delete
4	shourob	21	0000000000	Edit Delete

Discussion:

This was an interesting lab. I learn many things from this lab such as how to create, read, update and deleting a table entry from the web page. I also face many problems given as below but I can be able to overcome those problems.



404 | Not Found