Model–View–Controller (MVC) Laravel API

Lab 08 - Solutions

1- Create a Laravel app named Lab 8
2- Change the .env file

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=lab8
DB_USERNAME=root
DB_PASSWORD=
```

Create an api for file upload:

3- **php artisan make:request UploadRequest**
4- Change the Rules function in UploadRequest.php file in app/Http/Requests

```php
public function rules()
    {
        return [
            "file" => "required",
        ];
    }
```

5- **php artisan make:controller UploadController**
6- Add this function to the class:

```php
public function __invoke(Request $request)
    {
        $result=$request->file('file')->store('uploads');
        return ["Result:"=>$result];
    }
```

7- Add this route to api.php

```php
Route::post("upload", \App\Http\Controllers\UploadController::class);
```

To create a database for menu items and apis for CRUD operations:

8- ***php artisan make:model Menu --migration***

9- Go to app\Models\Menu.php and specify the columns for the table:
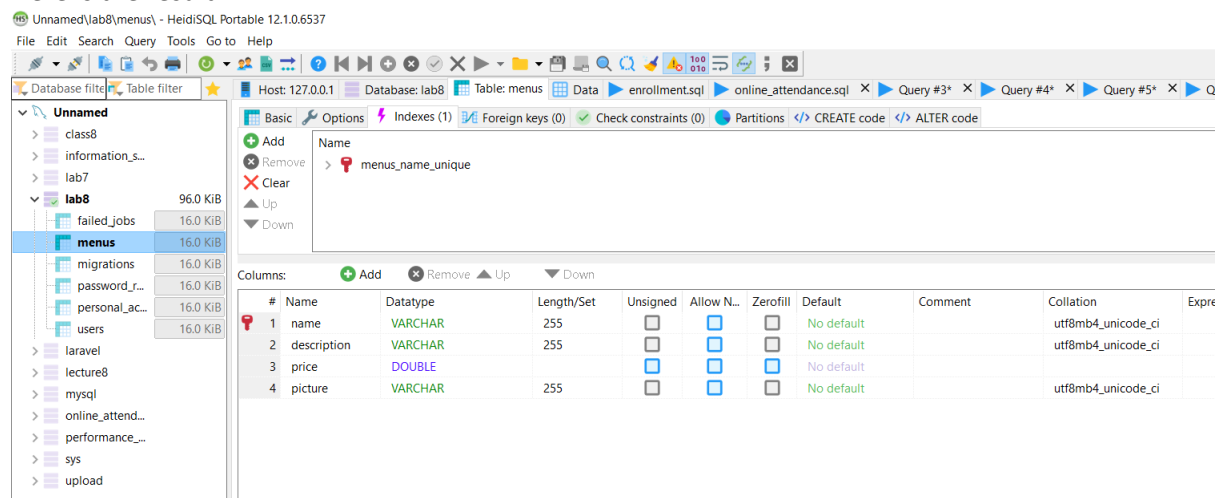
```
protected $fillable = ['name','description', 'price', 'picture'];
```

10- Then, go to database\migration and change the up function with this:

```php
public function up(): void
    {
        Schema::create('menus', function (Blueprint $table) {
            $table->string('name');
            $table->string('description');
            $table->double('price');
            $table->string('picture');
            $table->unique('name');

        });
    }
```

11- ***php artisan migrate***

12- Here is the result:

13- ***php artisan make:controller MenuController --api***

14- Change the Menu Controller:

```php
namespace App\Http\Controllers;


use Illuminate\Http\Request;
use App\Models\Menu;
use Illuminate\Support\Facades\DB;


class MenuController extends Controller
{
    /**
     * Display a listing of the resource.
```

```php
     */
    public function index()
    {
        return Menu::all();
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        return Menu::create($request->all());
    }

    /**
     * Display the specified resource.
     */
    public function find(Request $request)
    {

        $name = $request->query('name');
        $result = DB::table('menus')->where('name', $name)->get();

        return $result;

    }


    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request)
    {
        $name = $request->input('name');
        $description = $request->input('description');
        $price = $request->input('price');
        $picture = $request->input('picture');
        $item = DB::update('update menus set name =
?,description=?,price=?,picture=? where name = ?',
            [$name,$description,$price,$picture,$name]);
        return $item;
    }

    /**
     * Remove the specified resource from storage.
     */
    public function delete(Request $request)
    {
```

```
        $name = $request->query('name');
        $item = DB::delete('delete from menus where name = ?',[$name]);
        return $item;
    }
}
```

15- Change the api.php in routes:

```php
<?php

<?php

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\MenuController;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "api" middleware group. Make something great!
|
*/


Route::post("upload", \App\Http\Controllers\UploadController::class);
Route::group(['prefix' => 'menu'], function () {
    Route::get('/', [MenuController::class, 'index']);
    Route::get('find', [MenuController::class, 'find']);
    Route::post('/', [MenuController::class, 'store']);
    Route::put('/', [MenuController::class, 'update']);
    Route::delete('/', [MenuController::class, 'delete']);

});
```

Test APIs:
File Upload:
16- Run Postman and test the file upload:
In the form-data, you need to choose a file type and name the key as file. The pick the file and send. You must be able to see the result:
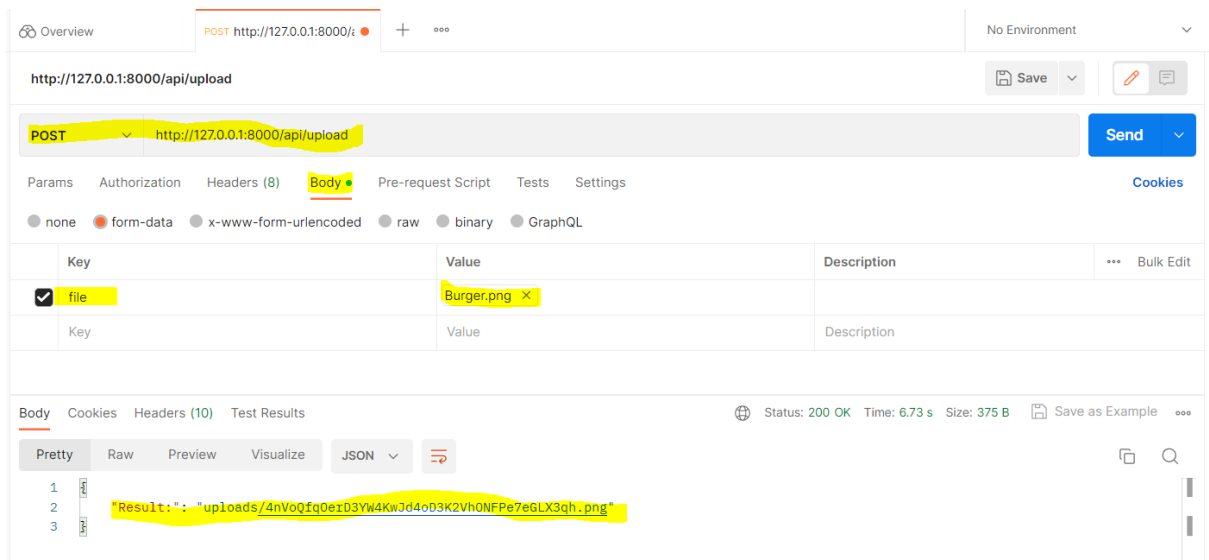
17- The file is uploaded in the uploads folder under storage\app
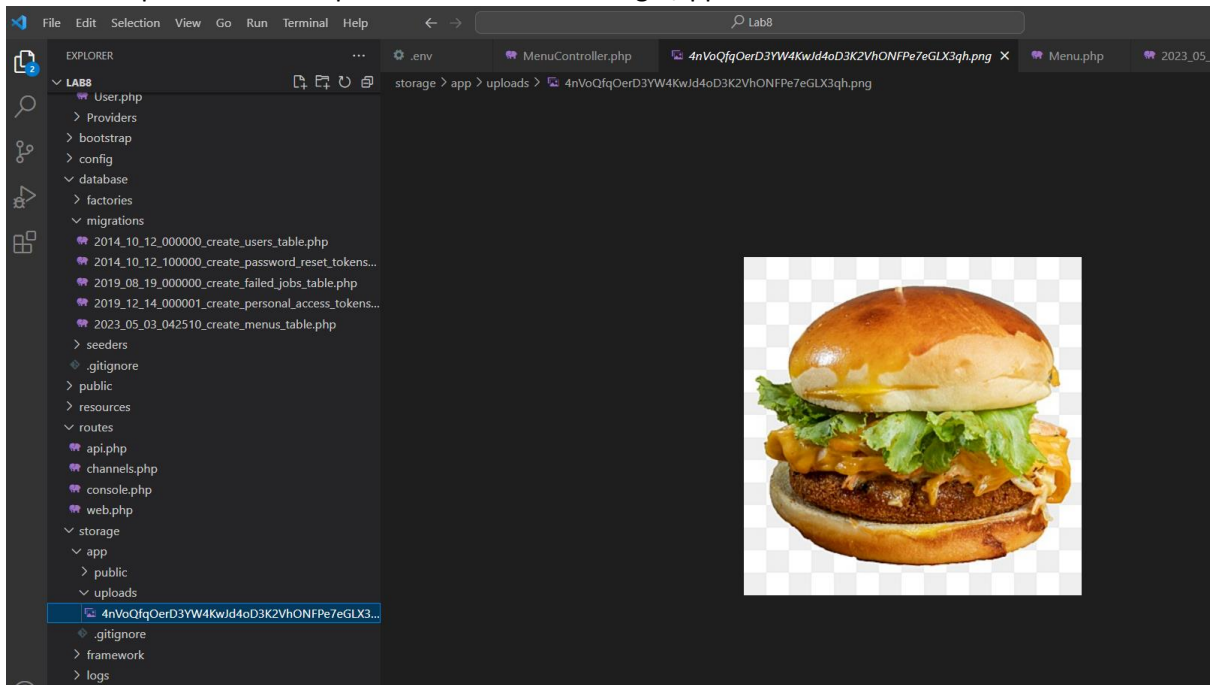
18- Copy the file path and use it for creating a new menu item:

19- Check the new record in the table:

20- Test index method:

21- Test find method:

22- Test update method:

23- Test delete method:

http://127.0.0.1:8000/api/menu/?name=Burger   💾 Save   ⌄   ✏️ 💬

DELETE ⌄	http://127.0.0.1:8000/api/menu/?name=Burger	**Send** ⌄

Params ●   Authorization   Headers (8)   Body ●   Pre-request Script   Tests   Settings   Cookies

Query Params

	Key	Value	Description	○○○ Bulk Edit
☑	name	Burger		
☐	price	8.99		
☐	description	Beef, Tomato, Sauce		
☐	picture	uploads/4nVoQfqOerD3YW4KwJd4oD3K2VhONFPe7eGL...		
	Key	Value	Description	

Body   Cookies   Headers (10)   Test Results   🌐 Status: **200 OK**   Time: **4.95 s**   Size: **317 B**   💾 Save as Example ○○○

Pretty   Raw   Preview   Visualize   📋 🔍

1