Sure. CRUD stands for Create, Read, Update, and Delete. These are the four basic operations that are performed on data in a database. Laravel provides a number of features that make it easy to create CRUD applications.

To create a CRUD application in Laravel, you will need to:

1. Create a database table to store your data.
2. Create a model class that represents the database table.
3. Create a controller class that handles the CRUD operations.
4. Create views that display the data and allow users to interact with it.

Once you have created these files, you can use Laravel's built-in routing and templating system to create a working CRUD application.

Here is an example of how to create a CRUD application for a database table called products:

1. Create the database table:

Code snippet

php artisan make:migration create\_products\_table

1. Migrate the database:

Code snippet

php artisan migrate

1. Create the model class:

Code snippet

php artisan make:model Product

1. In the the model class:

Code snippet

protected $table = 'products';

protected $primaryKey = 'product\_id';

protected $fillable = ['product\_name','brand\_name','product\_quantity'];

/// always the column

//name in database table and in the name of the inputs

1. Create the controller class:

Code snippet

php artisan make:controller ProductController

1. Create the views:

Code snippet

resources/views/products/index.blade.php //form submition

 <div class="container" style="margin-top:20vh">

        <h1 class="display-4">Enter Product</h1>

        <form action="" method="post">

            @csrf

            <div class="mb-3">

                <label  class="form-label">Product Name</label>

                <input type="text" name="product\_name" class="form-control">

            </div>

            <div class="mb-3">

                <label class="form-label">Brand Name</label>

                <input type="text" name="brand\_name" class="form-control">

            </div>

            <div class="mb-3">

                <label class="form-label">Product Quantity</label>

                <input type="number" name="product\_quantity" class="form-control">

            </div>

            <button type="submit" class="btn btn-primary">Submit</button>

        </form>

    </div>

resources/views/products/show.blade.php //datashow in table

    <div class="container">

    <table class="table">

        <thead>

          <tr>

            <th scope="col">#Id</th>

            <th scope="col">Product Name</th>

            <th scope="col">Brand Name</th>

            <th scope="col">Quantity</th>

            <th scope="col">Actions</th>

          </tr>

        </thead>

        <tbody>

            @foreach ($product as $product )

          <tr>

            <th scope="row">{{$product->product\_id}}</th>

            <td>{{$product->product\_name}}</td>

            <td>{{$product->brand\_name}}</td>

            <td>{{$product->product\_quantity}}</td>

            <td>

                <a type="button" class="btn btn-danger" href="{{url('/show')}}/{{$product->product\_id}}">

                    Delete

                </a>

                <a type="button" class="btn btn-primary" href="{{url('/show/edit')}}/{{$product->product\_id}}">

                    edit

                </a>

            </td>

          </tr>

          @endforeach

        </tbody>

      </table>

    </div>

resources/views/products/create.blade.php resources/views/products/edit.blade.php //edit the data

    <div class="container" style="margin-top:20vh">

        <h1 class="display-4">Edit Product</h1>

        <form action="{{url('/show/update')}}/{{$product->product\_id}}" method="post">

            @csrf

            <fieldset>

            <div class="mb-3">

                <label  class="form-label">Product Name</label>

                <input type="text" name="product\_name" class="form-control" value="{{$product->product\_name}}">

            </div>

            <div class="mb-3">

                <label class="form-label">Brand Name</label>

                <input type="text" name="brand\_name" class="form-control" value="{{$product->brand\_name}}">

            </div>

            <div class="mb-3">

                <label class="form-label">Product Quantity</label>

                <input type="number" name="product\_quantity" class="form-control" value="{{$product->product\_quantity}}">

            </div>

            <button type="submit" class="btn btn-success">Update</button>

            </fieldset>

        </form>

    </div>

1. Edit the routes/web.php file to add the following routes:

Code snippet

Route::get('/products', 'ProductController@index');

Route::get('/products/create', 'ProductController@create');

Route::post('/products', 'ProductController@store');

Route::get('/show', 'ProductController@show');

Route::get('/show/edit/{id}', 'ProductController@edit);

Route::get('/show/{product}', 'ProductController@destroy');

Route::post('/show/update/{id}', 'ProductController@update')->name(‘product.update’);

1. Edit the ProductController.php file to add the following methods:

Code snippet

public function index() {

$products = Product::all();

return view('products.index', compact('products'));

}

public function create()

{

return view('products.create');

}

public function store(Request $request) {

$product = new Product;

$product->fill($request->all());

$product->save();

return redirect('/show);

}

public function show()

{

$product = Product::all();

return view('products.show', compact('product'));

}

public function update(Request $request,$id) {

$product = Product::findOrFail($id);

$product->fill($request->all());

$product->save();

return redirect('/show);

}

public function edit($id){

$product = Product::findOrFail($id);

return view('products.edit',compact('product'));

}

public function destroy(Product $product) {

$product->delete();

return redirect('/show);

}

8. Visit the /products route in your browser to see the CRUD application in action.

You can now create, read, update, and delete products in your database.

**Sources**

1. <https://asdosit.com/konten/cara-membuat-login-multi-user-dengan-laravel-7>
2. <https://github.com/Fgeorgiou/IMS_backend_laravel> subject to license (MIT)
3. <https://github.com/09liweis/what-I-ate>
4. <https://github.com/Buendy/laravel-shop>
5. <https://github.com/DesarrolloWeb2/laravel>
6. <https://github.com/AMarchmarti/restapi>
7. <https://github.com/AlexeiBrunner/catalog-api>
8. <https://github.com/gustibagusbayu/onlineshop_laravel_flutter>
9. <https://github.com/Miguel-Leite/simple-crud-with-laravel>
10. <https://github.com/Hnooz/E-com>
11. <https://github.com/simtj-tdi/laravel_basic_clone>