

1. First Get Data From DataBase

Php artisan make:model user

1. Model: `User.php`

Location: `app/Models/User.php`

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
```

```
class User extends Authenticatable
{
```

```
    use HasFactory;
```

```
    protected $table = 'college'; // 'college' table name
    public $timestamps = false;
```

```
    protected $fillable = [
        'name',
        'email',
        'password',
        'mob', // Include 'mob' if it's in your database
    ];
}
```

2. Routes: `web.php`

Location: `routes/web.php`

```
<?php
```

```
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\MController;
```

```
Route::get('/', function () {
```

```
    return view('welcome');
});

// Route to fetch and display college data
Route::get('/users', [MController::class, 'ani']);
```

3. Controller: MController.php

Location: app/Http/Controllers/MController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\User;

class MController extends Controller
{
    public function ani()
    {
```

```
        $stu = \App\Models\User::all();

        return $stu;
    }
}
```

4. Blade Template: display.blade.php

Location: resources/views/display.blade.php

```
html
Copy code
<!DOCTYPE html>
<html>
<head>
    <title>College Data</title>
</head>
<body>
    <h1 align="center">College Data</h1>
    <table border="1" cellpadding="5"
align="center">
        <tr>
```

```

        <th>ID</th>
        <th>Name</th>
        <th>Email</th>
        <th>Mobile</th>
    </tr>
    @foreach($data as $ke)
    <tr>
        <td>{{ $ke->id }}</td>
        <td>{{ $ke->name }}</td>
        <td>{{ $ke->email }}</td>
        <td>{{ $ke->mob }}</td>
    </tr>
    @endforeach
</table>
</body>
</html>

```

2.Second Crud operation with database

Same as upper phle model create karenge or table ka nam dennge
(Route)Web.php or (View)display.blade.php same honge

```

<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class MController extends Controller
{
    public function ani()
    {
        Read Operation
        $stu = DB::table('college')->get();

        Insert Operation
        // $res = DB::table('college')->insert([
        //     'name' => 'Ani',
        //     'email' => 'ani@123',
        //     'mob' => '1234567890',
        // ]);
    }
}

```

```
// if ($res) {
//     return "Data Inserted";
// } else {
//     return "Data Not Inserted";
// }
```

Update Operation

```
// $res = DB::table('college')->where('name', 'aniket')->update(['name' => 'Aniket']);
// if ($res) {
//     return "Data Updated";
// } else {
//     return "Data Not Updated";
// }
```

Delete Operation

```
// $res = DB::table('college')->where('id', '3')->delete();
// if ($res) {
//     return "Data Deleted";
// } else {
//     return "Data Not Deleted";
// }
```

```
return $stu;
```

```
}
```

```
}
```

3.Session in Laravel

```
session(['user' => 'Aniket']);
```

```
<h3>Session Value: {{ session('user') }}</h3>
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```

use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Redirect;

class MyController extends Controller
{
    public function ani(Request $req)
    {
        // Session Banane ke liye Command
        $req->session()->put('name', $req->name);

        // Session ki value dekhane ke liye
        // return $req->session()->get('name');

        // Session se kisi particular name ki value delete karne ke liye
        // $user = $req->session()->pull('user');

        // Session ka all Data Remove Karta hai
        // $req->session()->flush();

        // Session ko destroy karta hai
        // $req->session()->invalidate();

        return view('second');
    }
}

```

```

$req->session()->flash('message', 'User Has been
Successfully Added');

return redirect('form');

```

```

}
}

?>

```

<!-- Blade Template: resources/views/second.blade.php -->

```

<div>
    <form action="/act" method="POST">
        @csrf
        <h3>Login Page</h3>
        <input type="text" placeholder="Enter Your Name" name="name"/><br><br>
        <input type="text" placeholder="Enter Your City" name="city"/><br><br>
    </form>
</div>

```

```
<button type="submit">Submit</button>
</form>
```

```
<h3>Session Value: {{ session('name') }}</h3>
</div>
```

Route file Web.php

```
<?php
```

```
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\MyController;
```

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

```
Route::view('/form', 'second');
```

```
Route::post('act', [MyController::class, 'ani']);
```

```
?>
```

3.UnCompleate Localization

Php artisan lang:publish run this cammand than lang folder and in this lang folder auto create en folder

En folder create this file en/welcome.php

```
<?php

return [

    "heading1"=>"Hello how are you",

    "heading2"=>"Welcome Page",
```

```
    "Dynamic Heading"=>"Hello My Name is :name Sonawat"  
]  
?>
```

Then create next folder hindi/welcome.php

```
<?php  
  
return [  
    "heading1"=>"आप कैसे हो",  
    "heading2"=>"स्वागत",  
    "Dynamic Heading"=>"नमस्ते मेरा नाम है :name सोनावत",  
]  
?>
```

Welcome.blade.php(view)

```
<h2>{{__( 'welcome.heading1' )}}</h2>  
  
<a href="/about">About</a>  
  
<a href="/about">contact</a>  
  
<a href="/about">login</a>
```

```
<h2>{{__('welcome.Dynamic
Heading', ['name'=>"Aniket"])}}</h2>

<a href="/setlang/en">English</a><br><br>

<a href="/setlang/hindi">Hindi</a>
```

Web.php

```
<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\MController;

Route::get('/', function () {

    return view('welcome');

});

Route::view('/about', 'second');

?>
```

.env File change line no 8

```
APP_LOCALE=hindi
```

Then run code

```
APP_LOCALE=en
```

Then run code

Final MySQL Crud Operation

1. Model

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Ani extends Model
{
    protected $table = 'college';
    public $timestamps = false;
}
```

2. Controller

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\model\Ani;
```

```
class MyController extends Controller
{
    function select()
    {
        $data = Ani::all();

        return view('components/data', ['data'=>$data]);
    }

    function ani(Request $req)
    {
        $student = new Ani();

        $student->name = $req->name;

        $student->email = $req->email;

        $student->mob = $req->mob;

        $student->save();

        $data = Ani::all();

        return view('components/data', ['data'=>$data]);
    }
}
```

```
}
```

```
function del($id)
```

```
{
```

```
    $res = Ani::destroy($id);
```

```
    if($res = 1)
```

```
    {
```

```
        $data = Ani::all();
```

```
        return
```

```
view('components/data', ['data'=>$data, 'res'=>$res]);
```

```
    }
```

```
}
```

```
function up($id)
```

```
{
```

```
    // $data = Ani::where('name', 'Anik')-
```

```
>where('id', 2)->get();
```

```
        // $data = Ani::where('name', 'Anik')-  
>orWhere('id',1)->get();  
  
        $data = Ani::find($id);  
  
        return view('components/Update', ['data'=>$data]);  
  
    }  
  
    function upstu($id, REQUEST $req)  
  
    {  
  
        $student = Ani::find($id);  
  
        $student->name = $req->name;  
  
        $student->email = $req->email;  
  
        $student->mob = $req->mob;  
  
        $student->save();  
  
        $data = Ani::all();  
  
        return view('components/data', ['data'=>$data]);  
  
    }  
  
    function search(Request $req)  
  
    {
```

```

        $data = Ani::where('name','like',"%$req->search%")->get();

        return view('components/data',['data'=>$data]);

    }
}

```

3. Routes

<?php

```

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\MyController;

```

```

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

```

```

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

```

```

Route::post('/act', [MyController::class, 'ani']);
Route::get('/delete/{id}', [MyController::class, 'del']);
Route::get('/data', [MyController::class, 'select']);
Route::get('/update/{id}', [MyController::class, 'up']);
Route::put('/update-student/{id}', [MyController::class, 'upstu']);
Route::get('/search', [MyController::class, 'search']);

```

4. Views

Ensure the folder structure in **resources/views/components** contains:

- `data.blade.php` for listing records
- `update.blade.php` for the update form

Example `data.blade.php`:

```
<div>
  <h2>Student List</h2>
  <form action="/search">
    <input type="text" placeholder="Search by name" name="search">
    <button type="submit">Search</button>
  </form>
  <table border="1">
    <tr>
      <th>ID</th>
      <th>Name</th>
      <th>Email</th>
      <th>Mobile</th>
      <th>Delete</th>
      <th>Update</th>
    </tr>
    @foreach($data as $val)
      <tr>
        <td>{{ $val->id }}</td>
        <td>{{ $val->name }}</td>
        <td>{{ $val->email }}</td>
        <td>{{ $val->mob }}</td>
        <td><a href="/delete/{{ $val->id }}">Delete</a></td>
        <td><a href="/update/{{ $val->id }}">Update</a></td>
      </tr>
    @endforeach
  </table>
  <a href="/user">Add New Student</a>
</div>
```

Example `update.blade.php`:

```
<div>
```

```

<h2>Update Student</h2>
<form action="/update-student/{{ $data->id }}" method="POST">
    @csrf
    @method('PUT')
    <label>Name:</label>
    <input type="text" name="name" value="{{ $data->name }}">
    <label>Email:</label>
    <input type="text" name="email" value="{{ $data->email }}">
    <label>Mobile:</label>
    <input type="text" name="mob" value="{{ $data->mob }}">
    <button type="submit">Update</button>
</form>
</div>

```

Welcome.blade.php

```

<h1>Welcome Page</h1>

<a href="/user">Go to form</a>

<a href="/data">Go to data</a>

```

File Upload Page

Step 1: Set Up the Database Table

Agar Table ka nam Users ho to

Php artisan make:model user

Agar Table ka nam Images ho to

Php artisan make model image

```

<?php

namespace App\Models;

```

```
use Illuminate\Database\Eloquent\Model;

class Image extends Model
{
    public $timestamps = false;
}
```

Step 2 Web.php

```
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Mycontroller;

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

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

Route::post('/act', [Mycontroller::class, 'ani']); // File upload
Route::get('/show', [Mycontroller::class, 'aniket']); // Display images
```

Step 3 Mycontroller

```
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Image;
```



```

class Mycontroller extends Controller
{
    public function ani(Request $req)
    {
        $path = $req->file('file')->store('images', 'public'); // Store file in public
storage
        $imgName = explode('/', $path)[1]; // Get only the filename
        $img = new Image();
        $img->img = $imgName;
        $img->save();
        return redirect('/show');
    }

    public function aniket()
    {
        $data = Image::all(); // Fetch all images
        return view('Show', ['data' => $data]); // Pass data to the view
    }
}

```

Run cammand

Php Artisan storage:link

And Storage / App ke andar private folder ho to delete kre

Step 4 : Upload View

```

<div>
    <form action="/act" method="post" enctype="multipart/form-data">
        @csrf
        <input type="file" name="file" />
        <button type="submit">Upload</button>
    </form>
</div>

```

Step 5 :Show View

```
<div>
    @foreach($data as $val)
        <h2>{{ $val->img }}</h2>
        <div>
            
        </div>
    @endforeach
</div>
```

Step 6: Welcome Page

```
<h2>Jay Mata Di</h2>
<a href="/form">Upload Page</a>
<a href="/show">Show Page</a>
```

pagination in laravel

Step 1 : Controller

```
namespace App\Http\Controllers;
```

```
use App\Models\Post;
use Illuminate\Http\Request;
```

```
class PostController extends Controller
{
    public function index()
    {
        $posts = Post::paginate(10);

        return view('posts.index', ['posts' => $posts]);
    }
}
```

Step2. Blade Template

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Posts</title>
</head>
<body>
    <h1>Posts</h1>
    <table border="1" cellspacing="0" cellpadding="10">
        <tr>
            <th>ID</th>
            <th>Title</th>
            <th>Content</th>
        </tr>
        @foreach($posts as $post)
        <tr>
            <td>{{ $post->id }}</td>
            <td>{{ $post->title }}</td>
            <td>{{ $post->content }}</td>
        </tr>
        @endforeach
    </table>

    <!-- Pagination Links -->
    <div>
        {{ $posts->links() }}
    </div>
</body>
</html>

```

Component [php artisan make:component layout]

Blade Layout File: resources/views/layout.blade.php

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>{{ $title }}</title>
</head>
<body>
  <div class="header">
    <ul>
      <li><a href="{{ url('/') }}">Home</a></li>
      <li><a href="{{ url('/about') }}">About</a></li>
      <li><a href="{{ url('/login') }}">Login</a></li>
      <li><a href="{{ url('/contact') }}">Contact</a></li>
    </ul>
  </div>

  <div>
    {{ $main }}
  </div>

  <div class="footer">
    <p>Footer Page</p>
  </div>
</body>
</html>

```

About Page: `resources/views/about.blade.php`

```

<x-layout>
  <x-slot name="title">About Page</x-slot>
  <x-slot name="main">
    <h2>About Page Heading</h2>
    <h2>About Page Sub Heading</h2>
    <p>Dummy Paragraph for About Page</p>
  </x-slot>
</x-layout>

```

Home Page: `resources/views/home.blade.php`

```

<x-layout>
  <x-slot name="title">Home Page</x-slot>
  <x-slot name="main">
    <h2>Home Page Heading</h2>
    <h2>Home Page Sub Heading</h2>
    <p>Dummy Paragraph for Home Page</p>
  </x-slot>

```

</x-layout>

Web Routes: `routes/web.php`

<?php

```
use Illuminate\Support\Facades\Route;
```

```
// Route for the welcome page
Route::get('/', function () {
    return view('welcome');
});
```

```
// Route for the About page
Route::view('/about', 'about');
```

```
// Route for the Home page
Route::view('/home', 'home');
```

Migration In Laravel

1.php `artisan make:migration create_test_table`

- **Purpose:** Creates a new migration file for a table named `test`.
- **What happens:**
 - Laravel generates a migration file in the `database/migrations` directory with a timestamp in its filename.
 - The migration file includes a `up` method (for creating the table or making changes) and a `down` method (for reverting the changes).

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
```

```
use Illuminate\Database\Schema\Blueprint;
```

```
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration
```

```
{  
  
    public function up(): void  
    {  
  
        Schema::create('test', function (Blueprint $table) {  
  
            $table->id();  
  
            $table->string('name');  
  
            $table->string('email');  
  
            $table->timestamps();  
  
        });  
  
    }  
}
```

2.php artisan migrate

- **Purpose:** Runs all pending migrations.
- **What happens:**
 - Executes the up method of all migration files that haven't been run yet.
 - Creates or modifies database tables and schemas according to the migration files.

3.php artisan migrate:reset

- **Purpose:** Rolls back all migrations.
 - **What happens:**
 - Executes the down method of every migration file in reverse order.
 - Completely resets the database schema to its state before migrations.
-

4. `php artisan migrate:rollback --step=3`

- **Purpose:** Rolls back the last three batches of migrations.
 - **What happens:**
 - Executes the `down` method of the last three migration batches in reverse order.
 - Allows you to selectively roll back a specific number of batches without affecting earlier migrations.
-

5. `php artisan migrate --path=/database/migrations/2025_01_16_014924_create_test_table.php`

- **Purpose:** Runs a specific migration file.
 - **What happens:**
 - Executes the `up` method of the specified migration file.
 - Useful when you want to migrate only a single migration file instead of all pending migrations.
-

6. `php artisan migrate:refresh`

- **Purpose:** Resets and re-runs all migrations.
 - **What happens:**
 - Executes the `migrate:reset` command, which rolls back all migrations.
 - Executes the `migrate` command to run all migrations again.
 - Useful when you want to completely rebuild the database schema without losing the migration files.
-

Summary Table

| Command | Effect |
|--|--|
| <code>make:migration</code> | Creates a new migration file. |
| <code>migrate</code> | Runs all pending migrations. |
| <code>migrate:reset</code> | Rolls back all migrations. |
| <code>migrate:rollback --step=N</code> | Rolls back the last N batches of migrations. |
| <code>migrate --path=<path></code> | Runs a specific migration file. |
| <code>migrate:refresh</code> | Rolls back all migrations and re-applies them. |

-
1. **php artisan down**: Puts the Laravel app in maintenance mode, showing a maintenance page to users.
 2. **php artisan up**: Brings the app back online, making it accessible to users.
 3. **php artisan down --secret="your-secret"**: Enables maintenance mode but provides a secret URL (e.g., `http://localhost/your-secret`) to bypass it for testing.
-

User Se Condition Related Value get krna like First name Capital

1. Routes:

Defined in `web.php`:

```
php
CopyEdit
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Mycontroller;

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

Route::get('list', [Mycontroller::class, 'ani']);
```

2. Routes:

Defined in `web.php`:

```
php
CopyEdit
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Mycontroller;

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

Route::get('list', [Mycontroller::class, 'ani']);
```


- **GET /:** Loads the default welcome view.
 - **GET /list:** Calls the `ani` method in `Mycontroller` to fetch all user records.
-

3. Controller: `Mycontroller`

```
namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;

class Mycontroller extends Controller
{
    public function ani()
    {
        return User::all(); // Fetch and return all users
    }
}
```

Model: `user.php`

```
<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    use HasFactory, Notifiable;

    public function getNameAttribute($val)
    {
        return strtoupper($val);
    }

    /**
     * Accessor to append "@gmail.com" to the email attribute
     if not already present.
     */
    public function getEmailAttribute($val)
    {

```

```

        return str_ends_with($val, '@gmail.com') ? $val : $val
        . '@gmail.com';
    }

}
}

```

DataBase Me Condition Like First Letter Capital Save Karne ke liye Upper Vala Get ke Liye

Route

```
<?php
```

```

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Mycontroller;

```

```

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

```

```

// Route to fetch all users
Route::get('list', [Mycontroller::class, 'ani']);

```

```

// Route to save a new user
Route::get('save', [Mycontroller::class, 'save']);

```

Mycontroller

```
<?php
```

```
namespace App\Http\Controllers;
```

```

use App\Models\User;
use Illuminate\Http\Request;

```

```

class Mycontroller extends Controller
{
    public function ani()
    {

```

```

        return User::all();
    }

    function save()
    {
        $student = new User();
        $student->name = 'Pitar';
        $student->email = 'pitar';
        $student->password = 123;

        if ($student->save()) {
            echo "Data inserted successfully.";
        } else {
            echo "Failed to insert data.";
        }
    }
}

```

Model User.php

```
<?php
```

```

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    use HasFactory, Notifiable;

    function getNameAttribute($val)
    {
        return strtoupper($val);
    }
    function getEmailAttribute($val)
    {
        return $val."@gmail.com";
    }
    function setNameAttribute($val)
    {
        $this->attributes['name'] = ucfirst($val);
    }
}

```

```
}
```

Api in Larvel

Run Cammand : **php artisan install:api**

After this Cammnad auto make routes folder in api.php file in this file code

```
<?php
```

```
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Mycontroller;
```

```
Route::get('/user', function (Request $request) {
    return $request->user();
})->middleware('auth:sanctum');
```

```
// Route::get('/test',function(){
//     return ['name'=>'aniket','Email'=>'ani@123'];
// });
```

```
Route::get('/data',[Mycontroller::class,'ani']);
Route::post('/add',[Mycontroller::class,'add']);
Route::put('/update',[Mycontroller::class,'update']);
Route::delete('/delete/{id}',[Mycontroller::class,'delete']);
Route::get('/data/{name}',[Mycontroller::class,'search']);
```

Controller : Mycontroller

```
<?php
```

```
namespace App\Http\Controllers;
use App\Models\User;
```

```
use Illuminate\Http\Request;
```

```
class Mycontroller extends Controller
{
    function ani()
    {
        return User::all();
    }
}
```

```

}
function add(Request $req)
{

    $student = new User();
    $student->name = $req->name;
    $student->email = $req->email;
    $student->password = $req->password;
    if($student->save())
    {
        return "Student Added";
    }else{
        return "not Add";
    }
}

function update(Request $req)
{
    $student = User::find($req->id);
    $student->name = $req->name;
    $student->email = $req->email;
    $student->password = $req->password;
    if($student->save())
    {
        return "Student Added";
    }else{
        return "not Add";
    }
}

function delete($id)
{
    $student = User::destroy($id);
    if ($student) {
        return "Student Delete";
    }else{
        return "Student Not Delete";
    }
}

function search($name)
{
    $student = User::where('name',$name)->get();

```

```
        return $student;
    }
}
```

Routes : Web.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Mycontroller;

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

Laravel And React Combine Project

Run Cammand : **php artisan install:api**
Routes/api.php

```
<?php

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

Route::get('/user', function (Request $request) {
    return $request->user();
})->middleware('auth:sanctum');

Route::get('/user',[Mycontroller::class,'sonawat']);
Route::post('/add',[Mycontroller::class,'sonawa']);
```

MyController.php

```
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
```

```

use App\Models\myapp;

class MyController extends Controller
{
    function sonawat()
    {
        return myapp::all();
    }
    function sonawa(Request $req)
    {
        $path = $req->file('img')->store('images', 'public');
        $student = new myapp();
        $student->name = $req->name;
        $student->email = $req->email;
        $student->mobile = (int)$req->mobile;
        $student->img = $path;
        // $student->img = "";
        if($student->save())
        {
            return "Jay Mata Di";
        }
    }
}

```

App.jsx

```

import React, { useState, useEffect } from "react";
import axios from "axios";

const App = () => {
    const [data, setData] = useState([]);
    const [name, setName] = useState("");
    const [email, setEmail] = useState("");
    const [mobile, setMobile] = useState("");
    const [img, setImg] = useState(null);

    useEffect(() => {
        const fetchData = async () => {
            const res = await axios.get("http://localhost:8000/api/user");
            setData(res.data);
        };
        fetchData();
    }, []);

```

```

const add = async (e) => {
  e.preventDefault();
  console.log('act');

  const formData = new FormData();
  formData.append("name", name);
  formData.append("email", email);
  formData.append("mobile", mobile);
  formData.append("img", img);

  try {
    const res = await axios.post("http://localhost:8000/api/add", formData, {
      headers: {
        "Content-Type": "multipart/form-data",
      },
    });
    console.log(res.data);
  } catch (error) {
    console.error(error);
  }
};

return (
  <div>
    <h1>Upload User Data</h1>
    <form onSubmit={add} encType="multipart/form-data">
      <input type="text" placeholder="Name" value={name} onChange={(e) =>
setName(e.target.value)} />
      <input type="email" placeholder="Email" value={email} onChange={(e) =>
setEmail(e.target.value)} />
      <input type="number" placeholder="Mobile" value={mobile} onChange={(e) =>
setMobile(e.target.value)} />
      <input type="file" onChange={(e) => setImg(e.target.files[0])} />
      <button type="submit">Add</button>
    </form>

    <table border="1" cellPadding="5">
      <thead>
        <tr>
          <th>ID</th>
          <th>Name</th>
          <th>Email</th>

```



```

        <th>Mobile</th>
        <th>Image</th>
    </tr>
</thead>
<tbody>
    { data.map((item, index) => (
        <tr key={index}>
            <td>{index + 1}</td>
            <td>{item.name}</td>
            <td>{item.email}</td>
            <td>{item.mobile}</td>
            <td>
                <img
                    src={`http://localhost:8000/storage/${item.img}`}
                    alt={item.name}
                    width="50"
                    height="50"
                />
            </td>
        </tr>
    ))}
</tbody>
</table>
</div>
);
};
export default App;

```

Second Combine React Laravel Crud Operation

Api.php

```

<?php

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

Route::get('/user', function (Request $request) {

```

```
return $request->user();
})->middleware('auth:sanctum');
```

```
Route::get('/fetch',[Mycontroller::class,'fetch']);
Route::get('/fetch/{id}',[Mycontroller::class,'fetchone']);
Route::post('/add',[Mycontroller::class,'add']);
Route::delete('/del/{id}',[Mycontroller::class,'del']);
Route::post('/update/{id}',[Mycontroller::class,'update']);
```

Mycontroller

```
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\myapp;

class MyController extends Controller
{
    function fetch()
    {
        return myapp::all();
    }
    function add(Request $req)
    {
        $path = $req->file('img')->store('images','public');

        $student = new myapp();
        $student->name = $req->name;
        $student->email = $req->email;
        $student->mobile = (int)$req->mobile;
        $student->img = $path;
        if($student->save())
        {
            return "Jay Mata Di";
        }else{
            return "dsdkjs";
        }
    }
    function del($id)
    {
```

```

        if(myapp::destroy($id))
        {
            return "User Successfully Deleted";
        }else{
            return "User Not Deleted";
        }
    }

function fetchone($id)
{
    return myapp::find($id);
}
function update(Request $req,$id)
{
    $path = $req->file('img')->store('images','public');
    $student = myapp::find($id);
    $student->name = $req->name;
    $student->email = $req->email;
    $student->mobile = (int)$req->mobile;
    $student->img = $path;

    if($student->save())
    {
        return "Success";
    }else{
        return "not Success";
    }
}
}
}

```

App.jsx

```

import React from 'react'
import Home from './Home'
import {BrowserRouter, Routes,Route} from 'react-router-dom'
import Update from './Update'

const App = () => {

    return (
        <div>
            <BrowserRouter>
            <Routes>

```

```

    <Route path="/" element={ <Home/> }/>
    <Route path="/update/:id" element={ <Update/> }/>

  </Routes>
</BrowserRouter>
</div>
)
}

export default App

```

Home.jsx

```

import React, { useRef, useEffect, useState } from 'react';
import axios from 'axios';
import './App.css';
import { NavLink } from 'react-router-dom';

const Home = () => {
  const fi = useRef();
  const [data, setData] = useState([]);
  const [name, setName] = useState("");
  const [email, setEmail] = useState("");
  const [mobile, setMobile] = useState("");
  const [img, setImage] = useState("");

  const fetch = async () => {
    const res = await axios.get('http://localhost:8000/api/fetch');
    setData(res.data);
  };

  useEffect(() => {
    fetch();
  }, []);

  const add = async (e) => {
    e.preventDefault();
    const formdata = new FormData();
    formdata.append('name', name);
    formdata.append('email', email);
    formdata.append('mobile', mobile);
    formdata.append('img', img);
  };

```

```

const a = await axios.post('http://localhost:8000/api/add', formdata, {
  headers: {
    'Content-Type': 'multipart/form-data',
  },
});
console.log(a.data);
fetch();
setName("");
setEmail("");
setMobile("");
setImage("");
fi.current.value = "";
};

```

```

const del = async (p) => {
  const b = await axios.delete(`http://localhost:8000/api/del/${p}`);
  console.log(b.data);
  fetch();
};

```

```

return (
  <div className="container">
    <div className="form-container">

      <h1 className="app-heading">User Management Dashboard</h1>
      <form onSubmit={add} encType="multipart/form-data">
        <input
          placeholder="Enter Your name"
          type="text"
          value={name}
          onChange={(e) => setName(e.target.value)}
        />
        <input
          placeholder="Enter Your email"
          type="text"
          value={email}
          onChange={(e) => setEmail(e.target.value)}
        />
        <input
          placeholder="Enter Your mobile"
          type="text"
          value={mobile}

```

```

    onChange={ (e) => setMobile(e.target.value)}
  />
  <input
    ref={fi}
    type="file"
    onChange={ (e) => setImage(e.target.files[0])}
  />
  <button type="submit">Submit</button>
</form>
</div>

```

```

{ /* Right: Table */}
<div className="table-container">
  <table>
    <thead>
      <tr>
        <th>Id</th>
        <th>Img</th>
        <th>Name</th>
        <th>Email</th>
        <th>Mobile</th>
        <th>Delete</th>
        <th>Update</th>
      </tr>
    </thead>
    <tbody>
      {data.map((item, i) => (
        <tr key={i}>
          <td>{item.id}</td>
          <td>
            <img
              style={{ width: '70px', height: '40px' }}
              src={`http://localhost:8000/storage/${item.img}`}
              alt="Uploaded"
            />
          </td>
          <td>{item.name}</td>
          <td>{item.email}</td>
          <td>{item.mobile}</td>
          <td>
            <button onClick={() => del(item.id)}>Delete</button>
          </td>
          <td>

```

```

        <NavLink to={` /update/${item.id}`}><button
style={{ backgroundColor:'lime'}}>Update</button></NavLink>
      </td>
    </tr>
  )}
</tbody>
</table>
</div>
</div>
);
};

```

```
export default Home;
```

Update.jsx

```

import React, { useRef } from 'react'
import { useNavigate, useParams } from 'react-router-dom'
import { useState } from 'react'
import axios from 'axios'
import { useEffect } from 'react'
const Update = () => {
  const navigate = useNavigate();
  const fi = useRef();
  const {id} = useParams();
  const [data,setData] = useState([]);
  const [name, setName] = useState("");
  const [email, setEmail] = useState("");
  const [mobile, setMobile] = useState("");
  const [img, setImage] = useState("");

  const fetch = async ()=>{
    const response = await axios.get(`http://localhost:8000/api/fetch/${id}`);
    setData(response.data);
  }
  useEffect(()=>{
    fetch();
    setName(data.name);
    setEmail(data.email);
    setMobile(data.mobile);
  },[]);

  const up = async(e)=>{

```

```

e.preventDefault();
const formData = new FormData();
formData.append("name", name);
formData.append("email", email);
formData.append("mobile", mobile);
formData.append("img", img);

try {
const res = await axios.post(`http://localhost:8000/api/update/${id}`, formData, {
headers: {
"Content-Type": "multipart/form-data",
},
});
console.log(res.data);
} catch (error) {
console.error(error);
}

navigate('/');
}

```

```

return (
<div>

```

```

<h2>Update Page {id}</h2>

```

```

<form onSubmit={up} encType="multipart/form-data">

```

```

  <input
    placeholder={data.name}
    type="text"
    value={name}
    onChange={(e) => setName(e.target.value)}
  />

```

```

  <input
    placeholder={data.email}
    type="text"
    value={email}
    onChange={(e) => setEmail(e.target.value)}
  />

```

```

  <input
    placeholder={data.mobile}
    type="text"
    value={mobile}

```



```
      onChange={ (e) => setMobile(e.target.value)}
    />
    <input type='file' ref={fi} onChange={ (e)=>setImage(e.target.files[0])}/>

    <button type="submit">Submit</button>
  </form>
</div>
)
}
export default Update
```