

## Main menu

[Skip to content](#)

- [Home](#)
- [ক্যারিয়ার গাইডলাইন](#)
  - [Web Design এবং Web Development কি?](#)
  - [ZCPE কি?](#)
  - [Laravel Framework কি?](#)
  - [Industrial attachment এবং Industrial Training কি?](#)
  - [Computer Networking এবং Server Administration কি?](#)
  - [মোবাইল অ্যাপ ডেভেলপমেন্ট এক উজ্জ্বল ক্যারিয়ারের হাতছানি!](#)
- [ট্রেনিংস](#)
  - [Web Development](#)
    - [Responsive Web Design With WordPress Theme Customization](#)
    - [Professional Web Development with PHP, JAVASCRIPT, MySQL and WordPress Theme Development](#)
    - [Zend Certified PHP Engineering \(ZCPE\) Course](#)
    - [Professional WordPress Plugin Development Beginning to Advance](#)
    - [Laravel Beginning to Advance Course](#)
    - [JavaScript Beginning to Advance Course](#)
  - [Networking And Server Administration](#)
    - [Cisco](#)
      - [CCNA Certification](#)
    - [ISP Setup and Administration using MikroTik](#)
    - [Windows 2016 Server Administration](#)
    - [REDHAT ISP Setup](#)
  - [Mobile Apps Development](#)
    - [Native Android Apps Development with Java](#)
- [About Us](#)
- [Contact Us](#)

### Menu

Posted on [June 23, 2018](#) by [Masud Alam](#)

## বাংলায় PHP LARAVEL FRAMEWORK পর্ব-১৯: Laravel CRUD with Pagination

**CRUD** – Create, Read, Update এবং Delete হচ্ছে যেকোনো Web Application এর প্রয়োজনীয় ফাংশনালিটি গুলোর মধ্যে অন্যতম। লারাবেল ফ্রেমওয়ার্ক এর Resource Controller এবং route Model এর সাহায্যে, এটি সহজেই সম্পন্ন করা যায়। আজকে, আমরা user Module এর জন্য খুব সহজে কিভাবে Laravel CRUD with Pagination অপারেশন করা যায় তা ধাপে ধাপে দেখাবো। চলুন শুরু করা যাক :

### ধাপ-১: প্রস্তুতি

আমরা ধরে নিয়েছি আপনার PC তে Laravel ইতিমধ্যে Install করা আছে। এবং CRUD Operation এর জন্য আমরা Laravel Framework এর default users table টি ব্যবহার করবো। এবং একজন User লগিনের পর যেন CRUD Operation করতে পারে তাই Laravel এর default Login সিস্টেম ব্যবহার করব। আর এর জন্য আপনাকে আমাদের আগের পর্ব গুলোর মধ্যে [পার্ট-১](#) এবং [পার্ট-১৭](#) সম্পর্কে ধারণা থাকতে হবে। আর [পার্ট -৮](#) সম্পর্কে ধারণা থাকলে আপনাকে এই লেখাটি আরো বেশি সুবিধা দিবে।

### ধাপ-২: Resource Controller তৈরী

সব কিছু ঠিক থাকলে এইবার আমরা নিচের artisan command দিয়ে আমাদের resource Controller টি তৈরী করব।

```
1 | php artisan make:controller UserController --resource
```

উপরের command টি আমাদের জন্য একটি UserController তৈরী করে দিবে সাথে index(), create(), store(), show(), edit(), update() এবং destroy() ফাংশনগুলি তৈরী করে দিবে। যেমন নিম্নোক্ত:

```
1 | <?php
2 |
3 | namespace App\Http\Controllers;
4 |
5 | use Illuminate\Http\Request;
6 |
```

```

7  class UserController extends Controller
8  {
9      /**
10     * Display a listing of the resource.
11     *
12     * @return \Illuminate\Http\Response
13     */
14     public function index()
15     {
16         //
17     }
18
19     /**
20     * Show the form for creating a new resource.
21     *
22     * @return \Illuminate\Http\Response
23     */
24     public function create()
25     {
26         //
27     }
28
29     /**
30     * Store a newly created resource in storage.
31     *
32     * @param \Illuminate\Http\Request $request
33     * @return \Illuminate\Http\Response
34     */
35     public function store(Request $request)
36     {
37         //
38     }
39
40     /**
41     * Display the specified resource.
42     *
43     * @param int $id
44     * @return \Illuminate\Http\Response
45     */
46     public function show($id)
47     {
48         //
49     }
50
51     /**
52     * Show the form for editing the specified resource.
53     *
54     * @param int $id
55     * @return \Illuminate\Http\Response
56     */
57     public function edit($id)
58     {
59         //
60     }
61
62     /**
63     * Update the specified resource in storage.
64     *
65     * @param \Illuminate\Http\Request $request
66     * @param int $id
67     * @return \Illuminate\Http\Response
68     */
69     public function update(Request $request, $id)
70     {
71         //
72     }
73
74     /**
75     * Remove the specified resource from storage.
76     *
77     * @param int $id
78     * @return \Illuminate\Http\Response
79     */
80     public function destroy($id)
81     {
82         //
83     }
84 }

```

আপনি যদি কেবলমাত্র controller নাম দেন এবং -resource না দেন, তাহলে আপনি কোনও ফাংশন ছাড়াই একটা Plain Controller ফাইল পাবেন।

**Note:** আপনি যদি resource controller কে একটি ফোল্ডারের ভিতরে তৈরি করতে চান যেমন, admin ফোল্ডারের ভিতরে, আপনি Controller এর নামের আগে ফোল্ডারের নাম উল্লেখ করে কাজটি করতে পারেন। উদাহরণস্বরূপ এইরকম:

```
1 | php artisan make:controller Admin/UserController --resource
```

## ধাপ-৩: Route Setup

এইবার আমরা Route এ আমাদের URL টি ঠিক করে দিবো। যেমন নিম্নোক্ত:

```
1 | Route::resource('users', 'UserController');
```

## ধাপ-৪: Menu সেট করা

নিচের Code দিয়ে আপনি resources/views/layouts/app.blade.php তে Users Menu টি যুক্ত করে দিতে পারেন :

```
1 | <li class="nav-item"><a class="nav-link" href="{{ url('/users') }}">Users</a></li>
```

সম্পূর্ণ কোডটি ঠিক নিচের মতো হবে:

```
1 | <!DOCTYPE html>
2 | <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3 | <head>
4 |     <meta charset="utf-8">
5 |     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 |     <meta name="viewport" content="width=device-width, initial-scale=1">
7 |
8 |     <!-- CSRF Token -->
9 |     <meta name="csrf-token" content="{{ csrf_token() }}">
10 |
11 |     <title>{{ config('app.name', 'Laravel') }}</title>
12 |
13 |     <!-- Scripts -->
14 |     <script src="{{ asset('js/app.js') }}" defer></script>
15 |
16 |     <!-- Fonts -->
17 |     <link rel="dns-prefetch" href="https://fonts.gstatic.com">
18 |     <link href="https://fonts.googleapis.com/css?family=Raleway:300,400,600" rel="stylesheet" type="text/css">
19 |
20 |     <!-- Styles -->
21 |     <link href="{{ asset('css/app.css') }}" rel="stylesheet">
22 | </head>
23 | <body>
24 |     <div id="app">
25 |         <nav class="navbar navbar-expand-md navbar-light navbar-laravel">
26 |             <div class="container">
27 |                 <a class="navbar-brand" href="{{ url('/') }}">
28 |                     {{ config('app.name', 'Laravel') }}
29 |                 </a>
30 |                 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" data-bs-toggle="collapse">
31 |                     <span class="navbar-toggler-icon"></span>
32 |                 </button>
33 |
34 |                 <div class="collapse navbar-collapse" id="navbarSupportedContent">
35 |                     <!-- Left Side Of Navbar -->
36 |                     <ul class="navbar-nav mr-auto">
37 |                         @guest
38 |
39 |                         @else
40 |                             <li class="nav-item"><a class="nav-link" href="{{ url('/users') }}">User</a>
41 |                         @endguest
42 |                     </ul>
43 |
44 |                     <!-- Right Side Of Navbar -->
45 |                     <ul class="navbar-nav ml-auto">
46 |                         <!-- Authentication Links -->
47 |                         @guest
48 |                             <li class="nav-item">
49 |                                 <a class="nav-link" href="{{ route('login') }}">{{ __('Login') }}</a>
50 |                             </li>
51 |                             <li class="nav-item">
52 |                                 <a class="nav-link" href="{{ route('register') }}">{{ __('Register') }}</a>
53 |                             </li>
54 |                         @else
55 |
56 |                         <li class="nav-item dropdown">
57 |                             <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" data-bs-toggle="dropdown">
58 |                                 {{ Auth::user()->name }} <span class="caret"></span>
59 |                             </a>
60 |
61 |                             <div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdown">
62 |                                 <a class="dropdown-item" href="{{ route('logout') }}">
63 |                                     onclick="event.preventDefault();
64 |                                     document.getElementById('logout-form').submit();
65 |                                 </a>
66 |                             </div>
67 |                         </li>
68 |                     </ul>
69 |                 </div>
70 |             </div>
71 |         </nav>
72 |
73 |         <main class="container">
74 |             @yield('content')
75 |         </main>
76 |
77 |         <div class="footer">
78 |             <p>Laravel 5.4</p>
79 |         </div>
80 |     </div>
81 | </body>
82 | </html>
```

```

65         {{ __('Logout') }}
66     </a>
67
68     <form id="logout-form" action="{{ route('logout') }}" method="POST"
69         @csrf
70     </form>
71 </div>
72 </li>
73 @endguest
74 </ul>
75 </div>
76 </div>
77 </nav>
78
79 <main class="py-4">
80     @yield('content')
81 </main>
82 </div>
83 </body>
84 </html>

```

লাইন নম্বর ৩৯ এ লক্ষ্য করুন , আমাদের Users Menu Link টি @else এর মধ্যে দিয়েছি যেন Login এর পর এই menu তে access করতে পারি।

## ধাপ-৫: User List প্রদর্শন

পরবর্তীতে, যখন আমরা localhost:8000/users url টি হিট করব, তখন আপনি একটি ফাঁকা page দেখতে পাবেন। কারণ আমরা এখনও আমাদের controller কনফিগার করি নাই এবং view ফাইল তৈরী করি নাই।

তাহলে চলুন আমাদের users list দেখানোর জন্য প্রথমে resources/views ফোল্ডারের মধ্যে users নামক একটি ফোল্ডার তৈরী করব। এবং সেখানে আমরা all.blade.php file টি তৈরী করব। all.blade.php file টি ঠিক নিচের মতো হবে:

### resources/views/users/all.blade.php

```

1  @extends('layouts.app')
2
3  @section('content')
4  <div class="container">
5      @if(session()->has('status'))
6          <p class="alert alert-info">
7              {{ session()->get('status') }}
8          </p>
9      @endif
10     <div class="panel panel-default">
11         <div class="panel-heading">
12             User Listing
13             <a href="{{ route('users.create') }}" class="btn btn-success btn-xs">Add User</a>
14         </div>
15         <div class="panel-body">
16             @if (count($users))
17                 <div class="table-responsive">
18                     <table class="table table-bordered">
19                         <thead>
20                             <tr>
21                                 <th>Username</th>
22                                 <th>Email</th>
23                                 <th>Created On</th>
24                                 <th>Last Updated</th>
25                                 <th>Action</th>
26                             </tr>
27                         </thead>
28                         <tbody>
29                             @foreach($users as $user)
30                                 <tr>
31                                     <td>{{ $user->name }}</td>
32                                     <td>{{ $user->email }}</td>
33                                     <td>{{ $user->created_at->format('m-d-Y') }}</td>
34                                     <td>{{ $user->updated_at->format('m-d-Y') }}</td>
35                                     <td>
36                                         <a href="{{ route('users.edit', $user->id) }}" class="btn btn-primary btn-xs">Edit</a>
37                                         <a href="{{ route('users.show', $user->id) }}" class="btn btn-info btn-xs">Show</a>
38                                         <form action="{{ route('users.destroy', $user->id) }}" method="POST">
39                                             {{ csrf_field() }}
40                                             {{ method_field('DELETE') }}
41                                             <button class="btn btn-danger btn-xs">
42                                                 <span>DELETE</span>
43                                             </button>
44                                         </form>
45                                     </td>
46                                 </tr>
47                             @endforeach

```

```

48         </tbody>
49     </table>
50 </div>
51 <div class="text-center">
52     {{ $users->links() }}
53 </div>
54 @else
55 <p class="alert alert-info">
56     No Listing Found
57 </p>
58 @endif
59 </div>
60 </div>
61 </div>
62 @endsection

```

user list দেখানোর জন্য আমাদের view রেডি, এখন UserController.php file এ আমাদেরকে তিন জায়গায় পরিবর্তন করতে হবে। প্রথমে file টির শুরু দিকে use App\User; এই namespace টি use করতে হবে , দ্বিতীয়ত একটি constructor function তৈরী করতে হবে , এবং সেখানে আমরা \$this->middleware('auth'); এই middleware টি দিবো , যেন কেও সরাসরি user list access না পায়। সর্বশেষ index() function এ নিচের মতো করে class এবং view file গুলো ব্যবহার করতে হবে।

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\User;
7
8  class UserController extends Controller
9  {
10     public function __construct()
11     {
12         $this->middleware('auth');
13     }
14
15     /**
16      * Display a listing of the resource.
17      *
18      * @return \Illuminate\Http\Response
19      */
20     public function index()
21     {
22         //
23         $users = User::latest()->paginate(5);
24         return view('users.all', compact('users'));
25     }
26
27     /**
28      * Show the form for creating a new resource.
29      *
30      * @return \Illuminate\Http\Response
31      */
32     public function create()
33     {
34         //
35     }
36
37     /**
38      * Store a newly created resource in storage.
39      *
40      * @param \Illuminate\Http\Request $request
41      * @return \Illuminate\Http\Response
42      */
43     public function store(Request $request)
44     {
45         //
46     }
47
48     /**
49      * Display the specified resource.
50      *
51      * @param int $id
52      * @return \Illuminate\Http\Response
53      */
54     public function show($id)
55     {
56         //
57     }
58
59     /**
60      * Show the form for editing the specified resource.
61      *

```

```

62     * @param int $id
63     * @return \Illuminate\Http\Response
64     */
65     public function edit($id)
66     {
67         //
68     }
69
70     /**
71     * Update the specified resource in storage.
72     *
73     * @param \Illuminate\Http\Request $request
74     * @param int $id
75     * @return \Illuminate\Http\Response
76     */
77     public function update(Request $request, $id)
78     {
79         //
80     }
81
82     /**
83     * Remove the specified resource from storage.
84     *
85     * @param int $id
86     * @return \Illuminate\Http\Response
87     */
88     public function destroy($id)
89     {
90         //
91     }
92 }

```

এখানে latest() ফাংশন updated\_at কলাম এর অর্ডার অনুযায়ী user list রিটার্ন করবে। তবে, আপনি একটি কলাম name দিয়ে এটি ignore করতে পারেন, যেমন latest('id') এটি id এর অর্ডার অনুসারে user list return করবে।

সবকিছু ঠিক থাকলে এখন আপনি নিচের মতো ফলাফল দেখতে পারেন।

Laravel Users Masud Alam

User Listing Add User

Username	Email	Created On	Last Updated	Action
Russel Ahmed	russel@mail.com	06-22-2018	06-22-2018	Edit View DELETE
Projon Motors-10	projon@mail.com	06-22-2018	06-22-2018	Edit View DELETE
Omran Jamal	omran@jamal.com	06-22-2018	06-22-2018	Edit View DELETE
Abir Khandakar	abir@khandakar.com	06-22-2018	06-22-2018	Edit View DELETE
Noman Ahmed	Noman@gmail.com	06-22-2018	06-22-2018	Edit View DELETE

< 1 2 >

Laravel Crud with Pagination

## ধাপ-৬: Create

এখন, User Listing Table এর উপরের add user button ক্লিক দিলে আপনি একটি খালি পৃষ্ঠা দেখতে পারেন। কারণ আমরা এখনো প্রয়োজনীয় প্রয়োজনীয় Form তৈরী বা কল করি নাই।

তবে এখানে সমস্যা হচ্ছে আপনি এখানে লারাভেল এর বিল্ট ইন কোনো ফর্ম বিল্ডার পারেন না। আপনাকে manual html form তৈরী করতে হবে। কারণ লারাভেল ৫ থেকে external package গুলোকে তার core থেকে বাদ দিয়েছে, তবে আপনি চাইলে সে গুলোকে [Laravel Collective](#) থেকে install করে আপনার নিজের প্রজেক্ট এ ব্যবহার করতে পারেন। Laravel Collective Install করার জন্য আপনাকে নিচের command টি রান করতে হবে।

```
1 | composer require "laravelcollective/html":"^5.6.0"
```

এবং আপনাকে নিচের মতো একটা রেজাল্ট দেখাবে :

```
Command Prompt
C:\blog>composer require "laravelcollective/html":"^5.6.0"
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
 - Installing laravelcollective/html (v5.6.10): Downloading (100%)
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover
Discovered Package: fideloper/proxy
Discovered Package: laravel/tinker
Discovered Package: nunomaduro/collision
Discovered Package: laravelcollective/html
Package manifest generated successfully.
C:\blog>
```

## Laravel Collective HTML Form Installation

installation complete হয়ে গেলে এবার আপনি resources/views/users ফোল্ডারের মধ্যে create.blade.php ফাইল টি তৈরী করে ফেলতে পারেন ।

```
1 @extends('layouts.app')
2
3 @section('content')
4 <div class="container">
5     @if(session()->has('status'))
6         <p class="alert alert-info">
7             {{ session()->get('status') }}
8         </p>
9     @endif
10    <div class="col-sm-6 col-sm-offset-3">
11        <div class="panel panel-default">
12            <div class="panel-heading">
13                Add User
14            </div>
15            <div class="panel-body">
16                {{ Form::open(['url' => route('users.store'), 'method' => 'POST' ]) }}
17                @include('users._form')
18                {{ Form::close() }}
19            </div>
20        </div>
21    </div>
22 </div>
23 @endsection
```

উপরের কোডে, আমরা \_form.blade.php ফাইল include করেছি। আমি এই কাজ করেছি, যাতে আমরা create এবং update উভয় কাজই একই form ব্যবহার করে করতে পারি। চলুন \_form.blade.php view ফাইল এ এবং নিম্নোক্ত কোডটি লিখি।

```
1 @if (isset($errors) && (count($errors) > 0))
2     <div class="alert alert-danger">
3         <ul>
4             @foreach ($errors->all() as $error)
5                 <li>{{ $error }}</li>
6             @endforeach
7         </ul>
8     </div>
9 @endif
10
11 <div class="form-group">
12     {{ Form::label('name', 'Name') }} <em>*</em>
13     {{ Form::text('name', null, ['class' => 'form-control', 'id' => 'name', 'required' => 'required']) }}
14 </div>
15
16 <div class="form-group">
17     {{ Form::label('email', 'Email') }} <em>*</em>
18     {{ Form::email('email', null, ['class' => 'form-control', 'id' => 'email']) }}
19 </div>
20
21 <div class="form-group">
22     {{ Form::label('password', 'Password') }} <em>*</em>
23     {{ Form::password('password', ['class' => 'form-control', 'id' => 'password']) }}
24 </div>
25
26 <div class="form-group">
27     {{ Form::submit('Submit', ['class' => 'btn btn-success']) }}
28 </div>
```

উপরের \_form.blade.php file এ আমরা ১-৯ নম্বর লাইনে আমাদের form error গুলো দেখানোর ব্যবস্থা করেছি। এবং ১২-২৭ লাইন পর্যন্ত আমরা

laravelcollective এর html package ব্যবহার করে form build করেছি।

এখন আমাদের view file এর কাজ শেষ,এবার UserController.php এ আমাদের create ফাংশনকে নিচের মতো করে Modify করব।

```
1  /**
2   * Show the form for creating a new resource.
3   *
4   * @return \Illuminate\Http\Response
5   */
6  public function create()
7  {
8      //
9      return view('users.create');
10 }
```

এখন আপনি যখন localhost:8000/users/create লিখে hit করবেন অথবা যদি add user button এ click করেন। তাহলে নিচের মতো result দেখতে পাবেন।

Laravel Users

Add User

Name \*

Email \*

Password \*

Submit

## Laravel Create User Form

এখন আপনি যখন form টি পূরণ করে submit করবেন , এটি store() function কে হিট করবে, আমরা store() function দিয়ে ফর্ম এর ডাটা database এ save করার আগে validation করে নিবো। এবং এর জন্য Laravel এর Validator Class এর make() function টি ব্যবহার করব। তবে তার আগে আপনার password কে encrypt বা hash করার জন্য এই line টি Controller Class এর আগে জুড়ে দিতে হবে।

```
1 use Hash;
```

নিচের store() Method টি সহ পুরো UserController Class এর কোড দেওয়া হলো:

```
1  <?php
2
3  namespace App\Http\Controllers;
4  use Illuminate\Support\Facades\Validator;
5  use Hash;
6  use Illuminate\Http\Request;
7  use App\User;
8
9  class UserController extends Controller
10 {
11     public function __construct()
12     {
13         $this->middleware('auth');
14     }
15
16     /**
17      * Display a listing of the resource.
18      *
19      * @return \Illuminate\Http\Response
20      */
21     public function index()
22     {
23         //
24         $users = User::latest()->paginate(5);
25         return view('users.all', compact('users'));
26     }
27
28     /**
```



```

29     * Show the form for creating a new resource.
30     *
31     * @return \Illuminate\Http\Response
32     */
33     public function create()
34     {
35         //
36         return view('users.create');
37     }
38
39     /**
40     * Store a newly created resource in storage.
41     *
42     * @param \Illuminate\Http\Request $request
43     * @return \Illuminate\Http\Response
44     */
45     public function store(Request $request)
46     {
47         //
48         $validatedData = Validator::make($request->all(), [
49             'name' => 'required|max:40',
50             'email' => 'required|email|unique:users',
51             'password' => 'required|min:6'
52         ]);
53         if ($validatedData->fails()) {
54             return redirect('users/create')
55                 ->withErrors($validator)
56                 ->withInput();
57         }
58         $inputs=[
59             'name' => $request['name'],
60             'email' => $request['email'],
61             'password' => Hash::make($request['password']),
62         ];
63         if(User::create($inputs)){
64
65             return redirect('users');
66         }
67     }
68
69     /**
70     * Display the specified resource.
71     *
72     * @param int $id
73     * @return \Illuminate\Http\Response
74     */
75     public function show($id)
76     {
77         //
78     }
79
80     /**
81     * Show the form for editing the specified resource.
82     *
83     * @param int $id
84     * @return \Illuminate\Http\Response
85     */
86     public function edit($id)
87     {
88         //
89     }
90
91     /**
92     * Update the specified resource in storage.
93     *
94     * @param \Illuminate\Http\Request $request
95     * @param int $id
96     * @return \Illuminate\Http\Response
97     */
98     public function update(Request $request, $id)
99     {
100         //
101     }
102
103     /**
104     * Remove the specified resource from storage.
105     *
106     * @param int $id
107     * @return \Illuminate\Http\Response
108     */
109     public function destroy($id)
110     {
111         //

```

```

112     }
113 }
114 }

```

এবার আপনি অনায়াসেই যেকোনো নতুন User তৈরী করতে পারবেন। নিচের ছবিটি লক্ষ্য করুন:

Laravel
Users
Masud Alam

Add User
Name \*

Sohriful Hossain

Email \*

sohriful@hossain.co

Password \*

\*\*\*\*\*

Submit

Create New User in Laravel Crud

## ধাপ-৭: Read

এতক্ষন আমরা কিভাবে Form দিয়ে নতুন User তৈরী করা যায় তা দেখলাম , এখন আমরা দেখবো প্রত্যেকটি User কে আলাদা ভাবে কিভাবে দেখানো যায় সেই ব্যবস্থা। চলুন প্রথমে view.blade.php file এ নিচের কোডটি লিখে ফেলি।

```

1  @extends('layouts.app')
2
3  @section('content')
4  <div class="container">
5      @if(session()->has('status'))
6          <p class="alert alert-info">
7              {{ session()->get('status') }}
8          </p>
9      @endif
10     <div class="col-sm-6 col-sm-offset-3">
11         <div class="panel panel-default">
12             <div class="panel-heading">
13                 User Details
14             </div>
15             <div class="panel-body">
16                 <div class="form-horizontal">
17                     <div class="form-group">
18                         <label class="col-sm-3 control-label" for="name">Name</label>
19                         <div class="col-sm-9">
20                             <p class="form-control">{{ $user->name }}</p>
21                         </div>
22                     </div>
23                     <div class="form-group">
24                         <label class="col-sm-3 control-label" for="name">Email</label>
25                         <div class="col-sm-9">
26                             <p class="form-control">{{ $user->email }}</p>
27                         </div>
28                     </div>
29                     <div class="form-group">
30                         <label class="col-sm-3 control-label" for="name">Password</label>
31                         <div class="col-sm-9">
32                             <p class="form-control">*****</p>
33                         </div>
34                     </div>
35                     <div class="form-group">
36                         <label class="col-sm-3 control-label" for="name">Created On</label>
37                         <div class="col-sm-9">
38                             <p class="form-control">{{ $user->created_at->format('m-d-Y') }}</p>
39                         </div>
40                     </div>
41                     <div class="form-group">
42                         <label class="col-sm-3 control-label" for="name">Last Updated</label>
43                         <div class="col-sm-9">
44                             <p class="form-control">{{ $user->created_at->format('m-d-Y') }}</p>
45                         </div>
46                     </div>
47                     <div class="form-group">
48                         <div class="col-sm-4 col-sm-offset-4">
49                             <a href="{{ route('users.index') }}" class="btn btn-success">Back</a>
50                         </div>
51                     </div>
52                 </div>
53             </div>
54         </div>

```

```

55     </div>
56 </div>
57 @endsection

```

এখন আমাদের view.blade.php file এর কাজ শেষ,এবার UserController.php এ আমাদের show ফাংশনকে নিচের মতো করে Modify করব।

```

1  /**
2   * Display the specified resource.
3   *
4   * @param int $id
5   * @return \Illuminate\Http\Response
6   */
7  public function show($id)
8  {
9      //
10     $user = User::find($id);
11     return view('users.view', compact('user'));
12 }

```

এবার আপনি অনায়াসেই যেকোনো User এর পাশেই view button এ click করলে আপনি সংশ্লিষ্ট User এর বিস্তারিত দেখতে পারবেন। নিচের ছবিটি লক্ষ্য করুন:

Laravel Users

User Details

Name

Abir Khandakar

Email

abir@khandakar.com

Password

\*\*\*\*\*

Created On

06-22-2018

Last Updated

06-22-2018

Back

show user details in Laravel Crud

## ধাপ-৮: UPDATE

READ এর কাজ শেষ , এখন আমরা দেখবো প্রত্যেকটি User কে আলাদা ভাবে কিভাবে edit করা যায় সেই ব্যবস্থা। চলুন প্রথমে update.blade.php file এ নিচের কোডটি লিখে ফেলি।

```

1  @extends('layouts.app')
2
3  @section('content')
4  <div class="container">
5      @if(session()->has('status'))
6          <p class="alert alert-info">
7              {{ session()->get('status') }}
8          </p>
9      @endif
10     <div class="col-sm-6 col-sm-offset-3">
11         <div class="panel panel-default">
12             <div class="panel-heading">
13                 Add User
14             </div>
15             <div class="panel-body">
16
17                 {{ Form::model($user, ['url' => route('users.update',$user->id), 'method' => 'Pl

```

```

18         @include('users._form')
19         {{ Form::close() }}
20     </div>
21 </div>
22 </div>
23 </div>
24 @endsection

```

এবার UserController.php এ আমাদের edit ফাংশনকে নিচের মতো করে Modify করব।

```

1  /**
2   * Show the form for editing the specified resource.
3   *
4   * @param int $id
5   * @return \Illuminate\Http\Response
6   */
7  public function edit($id)
8  {
9      //
10     $user = User::find($id);
11     return view('users.update', compact('user'));
12 }

```

এবার আপনি অনায়াসেই যেকোনো User এর পাশেই edit button এ click করলে আপনি সংশ্লিষ্ট User কে update করার সুযোগ পাবেন। নিচের ছবিটি লক্ষ্য করুন:

Laravel Users

Add User

Name \*

Omran Jamal

Email \*

omran@jama.com

Password \*

Submit

## Laravel User Update Option with Crud

এবার ফর্ম টি submit করলে user যেন আপডেট হয় সেই ব্যবস্থা করার পালা , আর এর জন্য আপনাকে update() Method এ নিচের মতো করে মোডিফাই করতে হবে।

```

1  /**
2   * Update the specified resource in storage.
3   *
4   * @param \Illuminate\Http\Request $request
5   * @param int $id
6   * @return \Illuminate\Http\Response
7   */
8  public function update(Request $request, $id)
9  {
10
11     $validatedData = Validator::make($request->all(), [
12         'name' => 'required|max:40',
13         'email' => 'required|email',
14         'password' => 'required|min:6'
15     ]);
16     if ($validatedData->fails()) {
17         return redirect('users/create')
18             ->withErrors($validatedData)
19             ->withInput();
20     }
21
22     $inputs=[
23         'name' => $request['name'],
24         'email' => $request['email'],
25         'password' => Hash::make($request['password']),

```

```

26     ];
27     User::find($id)->update($inputs);
28
29     return redirect("users");
30
31 }

```

## ধাপ-৯: DELETE

Update এর কাজ শেষ , এখন আমরা দেখাবো কিভাবে User কে delete করা যায় সেই ব্যবস্থা। অবশ্য এর জন্য আপনাকে view তে কোনো কাজ করতে হবে না , শুধু UserController class এর destroy() Method এ নিচের মতো করে Modify করে দিলেই হবে।

```

1  /**
2   * Remove the specified resource from storage.
3   *
4   * @param int $id
5   * @return \Illuminate\Http\Response
6   */
7  public function destroy($id)
8  {
9      //
10     User::find($id)->delete();
11
12     return redirect("users");
13
14 }

```

এবার চলুন UserController class এর সম্পূর্ণ কোড এক সাথে দেখা যাক :

```

1  <?php
2
3  namespace App\Http\Controllers;
4  use Illuminate\Support\Facades\Validator;
5  use Hash;
6  use Illuminate\Http\Request;
7  use App\User;
8
9  class UserController extends Controller
10 {
11     public function __construct()
12     {
13         $this->middleware('auth');
14     }
15
16     /**
17      * Display a listing of the resource.
18      *
19      * @return \Illuminate\Http\Response
20      */
21     public function index()
22     {
23         //
24         $users = User::latest()->paginate(5);
25         return view('users.all', compact('users'));
26     }
27
28     /**
29      * Show the form for creating a new resource.
30      *
31      * @return \Illuminate\Http\Response
32      */
33     public function create()
34     {
35         //
36         return view('users.create');
37     }
38
39     /**
40      * Store a newly created resource in storage.
41      *
42      * @param \Illuminate\Http\Request $request
43      * @return \Illuminate\Http\Response
44      */
45     public function store(Request $request)
46     {
47
48
49         $validatedData = Validator::make($request->all(),[
50             'name' => 'required|max:40',
51             'email' => 'required|email|unique:users',
52             'password' => 'required|min:6'
53         ]);

```

```

54         if ($validatedData->fails()) {
55             return redirect('users/create')
56                 ->withErrors($validatedData)
57                 ->withInput();
58         }
59         $inputs=[
60             'name' => $request['name'],
61             'email' => $request['email'],
62             'password' => Hash::make($request['password']),
63         ];
64         if(User::create($inputs)){
65             //return redirect('users');
66             return redirect("users");
67         }
68     }
69 }
70
71 /**
72  * Display the specified resource.
73  *
74  * @param int $id
75  * @return \Illuminate\Http\Response
76  */
77 public function show($id)
78 {
79     //
80     $user = User::find($id);
81     return view('users.view', compact('user'));
82 }
83
84 /**
85  * Show the form for editing the specified resource.
86  *
87  * @param int $id
88  * @return \Illuminate\Http\Response
89  */
90 public function edit($id)
91 {
92     //
93     $user = User::find($id);
94     return view('users.update', compact('user'));
95 }
96
97 /**
98  * Update the specified resource in storage.
99  *
100  * @param \Illuminate\Http\Request $request
101  * @param int $id
102  * @return \Illuminate\Http\Response
103  */
104 public function update(Request $request, $id)
105 {
106     $validatedData = Validator::make($request->all(),[
107         'name' => 'required|max:40',
108         'email' => 'required|email',
109         'password' => 'required|min:6'
110     ]);
111     if ($validatedData->fails()) {
112         return redirect('users/create')
113             ->withErrors($validatedData)
114             ->withInput();
115     }
116
117     $inputs=[
118         'name' => $request['name'],
119         'email' => $request['email'],
120         'password' => Hash::make($request['password']),
121     ];
122     User::find($id)->update($inputs);
123
124     return redirect("users");
125 }
126
127 /**
128  * Remove the specified resource from storage.
129  *
130  * @param int $id
131  * @return \Illuminate\Http\Response
132  */
133 public function destroy($id)
134 {
135
136

```

```

137 //
138 User::find($id)->delete();
139
140 return redirect("users");
141
142 }
143
144 }

```

**bien hecho !**

[Masud Alam](#)

[Masud Alam](#)



CEO at [TechBeeo Software Company](#)

আমি মাসুদ আলম, বাংলাদেশের ৩৬ তম Zend Certified Engineer । ২০০৯ সালে কম্পিউটার সাইন্স থেকে বেচেলর ডিগ্রী অর্জন করি। দীর্ঘ ৮ বছর আমি Winux Soft, SSL Wireless, IBCS-PRIMAX, Max Group, Canadian International Development Agency (CIDA), Care Bangladesh, World Vision, Hellen Keller সহ বিভিন্ন দেশি বিদেশী কোম্পানিতে ওয়েব ডেভেলপমেন্ট এবং সফটওয়্যার ডেভেলপমেন্ট এর উপর বিভিন্ন লিডিং পজিশন এ চাকরি এবং প্রজেক্ট লিড করি। বিশেষ ভাবে বাংলাদেশের ১০০ জন জেড সার্টিফাইড ইঞ্জিনিয়ার এর মধ্যে ৫২ জন ই আমার হাতে জেড সার্টিফাইড হয়েছে। বর্তমানে TechBeeo Software Company তে সিইও হিসাবে আছি । পাশাপাশি w3programmers ট্রেনিং ইনস্টিটিউট এ PHP এর উপর Professional এবং Advance Zend Certified PHP -7 Engineering কোর্স করাই। আর w3programmers.com সাইট টি আমার।

Tagged [laravel 5 bangla tutorial](#), [laravel 5 project bangla](#), [laravel 5.6 crud example](#), [laravel 5.6 crud step by step](#), [laravel bangla book](#), [laravel bangla documentation](#), [Laravel Bangla Laravel CRUD In Bangla](#), [laravel bangla tutorial pdf](#), [laravel crud generator](#), [laravel crud github](#), [laravel tutorial for beginners step by step bangla](#), [php laravel tutorial bangla](#), [php project tutorial bangla](#), [simple crud application in laravel 5](#)

## Post navigation

[← Previous](#)

[Next →](#)

2 comments to “বাংলায় PHP LARAVEL FRAMEWORK পর্ব-১৯: Laravel CRUD with Pagination”

1.  [Abu Naim Md Asif Khan](#)  
[June 23, 2018 at 4:02 pm](#)

great!

2.  [Md: Afzarul Islam](#)  
[November 18, 2018 at 10:22 am](#)

I faced some problems. When i all code copy and p pest my code editor and run my program. Then i show Route [login] not defined. search <http://localhost:8000/users/all> just first step not login and create.Please help me

## Leave a Reply

You must be [logged in](#) to post a comment.

Search for:

Video Player

# raptor not found.



## Categories

00:00 [Computer Networking](#)

00:00 [CSS](#)

01:04:59 [Git, gitHub, gitLab and Bitbucket](#)

• [HTML](#)  
Use Up/Down Arrow keys to increase or decrease volume.  
• [JavaScript](#)  
• [Laravel E-commerce Project](#)

- [OAuth](#)
- [PHP Arrays](#)
- [PHP Basic](#)
- [PHP Control Structures](#)
- [PHP Data Format & Types](#)
- [PHP Functions](#)
- [PHP Laravel Framework](#)
- [PHP MySQL](#)
- [PHP Object Oriented Programming](#)
- [PHP Operators](#)
- [PHP Socket Programming](#)
- [PHP Strings & Patterns](#)
- [PHP Variable & Constant](#)
- [PHP Web Features](#)
- [Python Basics](#)
- [Python Data Types](#)
- [VueJS](#)
- [WordPress Plugin Development](#)
- [Wordpress Theme Development](#)

## Join Our Group

**W3programmers**  
Facebook Group · 6,155 members

w3programmers is the Official group of the [www.w3programmers.com](http://www.w3programmers.com) website. Here you can discuss everything about Web Development & Technologies such...

## Like Us On Facebook

# raptor not found.



Copyright © 2021 w3programmers.com Bangla