

## Suplyer Crud

	Php artisan make:model Suplayer -m	cmd
	<pre>&lt;?php namespace App\Models; use Illuminate\Database\Eloquent\Model; class Suplayer extends Model{      protected \$fillable = [         'name',         'email',         'mobile',         'companyName',         'user_id'     ];     public function suplayer(){         return \$this-&gt;belongsTo(Suplayer::class);     } }</pre>	app/Models/ Suplayer.php
	<pre>public function suplayers(){     return \$this-&gt;hasMany(Suplayer::class); }</pre>	app/Models/ User.php
	<pre>Schema::create('suplayers', function (Blueprint \$table) {     \$table-&gt;id();     \$table-&gt;string('name');     \$table-&gt;string('email')-&gt;nullable();     \$table-&gt;string('mobile')-&gt;nullable();     \$table-&gt;string('companyName')-&gt;nullable();     \$table-&gt;unsignedBigInteger('user_id');     \$table-&gt;foreign('user_id')-&gt;references('id')-&gt;on('users')</pre>	database/migrations/2025_07_13_054545_create_suplayers_table.php

	<pre> -&gt;cascadeOnUpdate()-&gt;restrictOnDelete();     \$table-&gt;timestamp('created_at')-&gt;useCurrent();     \$table-&gt;timestamp('updated_at')-&gt;useCurrent()- &gt;useCurrentOnUpdate();     }); </pre>	
Specific migration	<pre> php artisan migrate -- path=database/migrations/2025_07_13_054545_create_suplayers_table.php </pre>	cmd
	Php artisan make:controller SuplayerController	cmd
Suplayer READ		
SuplayerPage Method	<pre> public function SuplayerPage(Request \$request){     \$user_id = \$request-&gt;header('id');     \$suplayers = Suplayer::where('user_id', \$user_id)-&gt;get();     return Inertia::render('SuplayerPage', ['suplayers'=&gt;\$suplayers]); } </pre>	app/Http/Controllers / SuplayerController.php
web	<pre> Route::middleware(SessionAuthenticate::class)-&gt;group(function(){     Route::get('/SuplayerPage', [SuplayerController::class, 'SuplayerPage']-&gt;name('SuplayerPage')); }); </pre>	
SuplayerPage	<pre> &lt;template&gt;     &lt;SideNavLayout&gt;         &lt;SuplayerList/&gt;     &lt;/SideNavLayout&gt; &lt;/template&gt; </pre>	

	<pre> &lt;script setup&gt; import SuplayerList from '../Components/Suplayer/SuplayerList.vue'; import SideNavLayout from '../Layout/SideNavLayout.vue';  &lt;/script&gt; </pre>	
SuplayerList.vue	<pre> &lt;template&gt;   &lt;div class="container-fluid"&gt;     &lt;div class="row"&gt;       &lt;div class="col-12"&gt;&lt;/div&gt;       &lt;div class="col-12"&gt;         &lt;div class="card"&gt;           &lt;div class="card-body"&gt;             &lt;div&gt;               &lt;h3&gt;Suplayer&lt;/h3&gt;             &lt;/div&gt;             &lt;hr/&gt;             &lt;div class="float-end"&gt;               &lt;button @click="submitPost" class="btn btn-success mx-3 btn-sm"&gt;                 Add Suplayer               &lt;/button&gt;             &lt;/div&gt;             &lt;div&gt;               &lt;input placeholder="Search..." class="form-control mb-2 w-auto form-control-sm" type="text"               v-model="searchValue"&gt;             &lt;EasyDataTable </pre>	

```

:headers="Header"
field="searchField"
value="searchValue" show-index>
action="{ id }">
@submit.prevent="editSubmit(id)" style="display: inline;">
name="id" :value="id" />
type="submit" class="btn btn-success mx-3 btn-sm">Edit</button>

class="btn btn-success mx-3 btn-sm"
:href="`/CustomerSavePage?id=${id}`">Edit

btn-danger btn-sm" @click="DeleteClick(id)">Delete</button>
</div>
</div>
</div>
</div>

```

```

buttons-pagination
alternating

```

```

:items="Item"
:rows-per-page="10"
:search-

```

```

:search-

```

```

<template #item-

```

```

<form

```

```

<input type="hidden"

```

```

<button

```

```

</form>

```

```

<!-- <Link

```

```

</Link> -->

```

```

<button class="btn

```

```

</template>

```

```

</EasyDataTable>

```

```
        </div>
</template>
<script setup>

import { usePage, router, Link,useForm } from '@inertiajs/vue3'
import { createToaster } from "@meforma/vue-toaster";
const toaster = createToaster();
import { ref } from "vue";

let page = usePage();
const Header = [

    { text: "Name", value: "name" },
    { text: "Email", value: "email" },
    { text: "Mobile", value: "mobile" },
    { text: "companyName", value: "companyName" },
    { text: "Action", value: "action" },
];

const Item = ref(page.props.suplayers)
const searchValue = ref()

const DeleteClick = (id) => {
    let text = "Do you want to delete";
    if (confirm(text) === true) {
        router.get(`/delete-Suplayer/${id}`)
        toaster.success('Suplayer Deleted Successfully');
    } else {
        text = "You canceled!";
    }
}
```

```
}

const props = defineProps({
  id: {
    type: [String, Number],
    required: true
  }
})

function editSubmit(id) {
  router.post('/SuplayerSaveEdit', { id })
}

const submitPost = () => {
  router.post('/SuplayerSavePage', {
    // any form data here if needed
  })
}

</script>
```

Suplayer Delete

web	<pre>Route::middleware(SessionAuthenticate::class)-&gt;group(function(){     Route::get('/delete-Suplayer/{id}', [SuplayerController::class, 'SuplayerDelete'])-&gt;name('SuplayerDelete'); });</pre>	
	<pre>public function SuplayerDelete(Request \$request,\$id){     \$user_id = \$request-&gt;header('id');     Suplayer::where('user_id', \$user_id)-&gt;where('id', \$id)- &gt;delete();     // return response()-&gt;json([     //     'status' =&gt; 'success',     //     'message' =&gt; 'Customer Deleted successfully'     // ]);     \$data = ['message'=&gt;'Suplayer Deleted successfully','status'=&gt;true,'error'=&gt;''];     return redirect('/SuplayerPage')-&gt;with(\$data); }</pre>	
SuplayerList.vue	<pre>&lt;template&gt; &lt;button class="btn btn-danger btn-sm" @click="DeleteClick(id)"&gt;Delete&lt;/button&gt; &lt;/template&gt;  &lt;script setup&gt; const DeleteClick = (id) =&gt; {     let text = "Do you want to delete";     if (confirm(text) === true) {         router.get(`/delete-Suplayer/\${id}`)         toaster.success('Suplayer Deleted Successfully');</pre>	

	<pre>         } else {             text = "You canceled!";         }     } } &lt;/script&gt; </pre>	
Suplayer Cread		
SuplayerList.vue এর ভিতরে template অংশে	<pre> &lt;button @click="submitPost" class="btn btn-success mx-3 btn-sm"&gt;     Add Suplayer &lt;/button&gt; </pre>	resources/js/Components/ Suplayer/SuplayerList.vue
SuplayerList.vue এর ভিতরে Script অংশে	<pre> const submitPost = () =&gt; {     router.post('/SuplayerSavePage', {         // any form data here if needed     }) } </pre>	
web	<pre> Route::post('/SuplayerSavePage', [SuplayerController::class, 'SuplayerSavePage']-&gt;name('SuplayerSavePage')); </pre>	
	<pre> public function SuplayerSavePage(Request \$request){     \$user_id = \$request-&gt;header('id');     \$suplayers = Suplayer::where('user_id', \$user_id)-&gt;get();     return     Inertia::render('SuplayerSavePage', ['suplayers'=&gt;\$suplayers]); } </pre>	
SuplayerSavePa ge.vue template	<pre> &lt;template&gt; &lt;SideNavLayout&gt;     &lt;SuplayerSaveForm/&gt; </pre>	



	<pre> &lt;/SideNavLayout&gt; &lt;/template&gt; &lt;script setup&gt; import SuplayerSaveForm from '../Components/Suplayer/SuplayerSaveForm.vue'; import SideNavLayout from '../Layout/SideNavLayout.vue';  &lt;/script&gt; </pre>	
SuplayerSaveForm.vue	<pre> &lt;template&gt;   &lt;div class="container-fluid"&gt;     &lt;div class="row d-flex justify-content-center"&gt;       &lt;div class="col-md-6"&gt;         &lt;div class="card"&gt;           &lt;div class="card-body"&gt;             &lt;div class="float-end"&gt;               &lt;a href="/SuplayerPage" class="btn btn- success mx-3 btn-sm"&gt;                 Back               &lt;/a&gt;             &lt;/div&gt;             &lt;form @submit.prevent="submit"&gt;               &lt;div class="card-body"&gt;                 &lt;h4&gt;Save Suplayer&lt;/h4&gt;                 &lt;input id="name" name="name" v- model="form.name" placeholder="Suplayer Name"                   class="form-control" type="text" /&gt;                 &lt;br /&gt; </pre>	



```

import { createToaster } from "@meforma/vue-toaster";
const toaster = createToaster();
import { ref } from "vue";

const form = useForm({ name: '', email: '', mobile: '', companyName: ''})
const page = usePage()

let URL = "/create-suplayer";

function submit() {
  form.post(URL, {
    onSuccess: () => {
      if (page.props.flash.status === true) {
        toaster.success(page.props.flash.message);
        setTimeout(() => {
          router.get("/SuplayerPage")
        }, 500)
      } else {
        toaster.warning(page.props.flash.message)
      }
    }
  })
}

</script>

```

web

```
Route::middleware(SessionAuthenticate::class)->group(function(){
```

	<pre>Route::post('/create-suplayer', [SuplayerController::class, 'SuplayerCreate'])-&gt;name('SuplayerCreate'); });</pre>	
	<pre>public function SuplayerCreate(Request \$request){     // dd-&gt;request-&gt;all();      \$user_id = \$request-&gt;header('id');      \$data = [         'name' =&gt; \$request-&gt;name,         'email' =&gt; \$request-&gt;email,         'mobile' =&gt; \$request-&gt;mobile,         'companyName' =&gt; \$request-&gt;companyName,         'user_id' =&gt; \$user_id     ];      Suplayer::create(\$data);      \$data = ['message'=&gt;'SuplayerPage created successfully', 'status'=&gt;true, 'error'=&gt;''];     return redirect('/SuplayerPage')-&gt;with(\$data); }</pre>	
Suplayer Update		
SuplayerList.vue এর ভিতরে template অংশে	<pre>&lt;template #item-action="{ id }"&gt;     &lt;form @submit.prevent="editSubmit(id)" style="display: inline;"&gt;         &lt;input type="hidden" name="id" :value="id" /&gt;     &lt;button type="submit" class="btn btn-success mx-3 btn-sm"&gt;Edit&lt;/button&gt;</pre>	resources/js/Components/ Suplayer/SuplayerList.vue

	<pre> &lt;/form&gt; &lt;/template&gt; </pre>	
SuplayerList.vue এর ভিতরে Script অংশে	<pre> const props = defineProps({   id: {     type: [String, Number],     required: true   } }) function editSubmit(id) {   router.post('/SuplayerSaveEdit', { id }) } </pre>	
web	<pre> Route::middleware(SessionAuthenticate::class)-&gt;group(function(){   Route::post('/SuplayerSaveEdit', [SuplayerController::class, 'edit'])-&gt;name('SuplayerSaveEdit'); }); </pre>	
	<pre> public function edit(Request \$request){   \$suplayer = Suplayer::findOrFail(\$request-&gt;id);   return   Inertia::render('SuplayerSavePage',['suplayers'=&gt;\$suplayer]); } </pre>	
	<pre> &lt;script setup&gt; import { useForm, usePage, router, Link } from '@inertiajs/vue3' import { createToaster } from "@meforma/vue-toaster"; const toaster = createToaster(); import { ref } from "vue";  const form = useForm({ name: '', email: '', mobile: '', companyName: '',id: ''}) </pre>	

	<pre> const page = usePage() let URL = "/create-suplayer"; let list = page.props.suplayers if ( list['id']!=null) {     URL = "/update-Suplayer";     form.name = list['name']     form.id = list['id']     form.email = list['email']     form.mobile = list['mobile']     form.companyName = list['companyName'] } function submit() {     form.post(URL, {         onSuccess: () =&gt; {             if (page.props.flash.status === true) {                 toaster.success(page.props.flash.message);                 setTimeout(() =&gt; {                     router.get("/SuplayerPage")                 }, 500)             } else {                 toaster.warning(page.props.flash.message)             }         }     }) }  &lt;/script&gt; </pre>	
web	<pre> Route::middleware(SessionAuthenticate::class)-&gt;group(function(){     Route::post('/update-Suplayer', [SuplayerController::class, 'SuplayerUpdate']-&gt;name('SuplayerUpdate')); }); </pre>	

