

Fontend configuratin			
1step	App.js configar	<pre>import 'bootstrap/dist/css/bootstrap.css'; import './bootstrap'; import './Assets/css/main.css'</pre>	resources/js/app.js
	Assets	All file such as css and image	
2 <sup>nd</sup> step	<b>NavLayout.vue</b>	<pre>&lt;template&gt;   &lt;slot&gt;    &lt;/slot&gt; &lt;/template&gt;</pre>	Layout/NavLayout.vue
	<b>SideNavLayout.vue</b>	<pre>&lt;template&gt;   &lt;div&gt;     &lt;nav id="topNav" class="navbar fixed-top bg-success top-navbar"&gt;       &lt;div class="container-fluid"&gt;         &lt;a id="MenuBar" @click="NavOpenClose" class="icon-nav mx-0 my-1 h5"&gt;           &lt;i class="fa text-white fa-bars"&gt;&lt;/i&gt;         &lt;/a&gt;       &lt;/div&gt;     &lt;/nav&gt;     &lt;div id="sideNav" class="side-nav-open"&gt;       &lt;div class="side-nav-top text-center"&gt;         &lt;img alt="" class="side-nav-logo" src="../../Assets/img/logo.svg" /&gt;       &lt;/div&gt;</pre>	Layout/SideNavLayout.vue

```

        <Link href="/ProfilePage" class="side-bar-
item">
            <span class="side-bar-item-icon"><i
class="fa fa-user-circle text-green" /></span>
            <span class="side-bar-item-
caption">Profile</span>
        </Link>

        <Link href="/user-logout" class="side-bar-
item">
            <span class="side-bar-item-icon"><i
class="fa fa-sign-out-alt text-green" /></span>
            <span class="side-bar-item-
caption">Logout</span>
        </Link>

    </div>
    <div id="content" class="content">
        <div class="container-fluid">
            <main>
                <slot></slot>
            </main>
        </div>
    </div>
</div>

</template>
<script setup>
import { Link } from '@inertiajs/vue3'
</script>

```

	Test1.vue	<pre> &lt;script setup&gt;  import SideNavLayout from '../Layout/SideNavLayout.vue';  &lt;/script&gt; &lt;template&gt;   &lt;SideNavLayout&gt; &lt;h2&gt;Test1&lt;/h2&gt;   &lt;/SideNavLayout&gt;  &lt;/template&gt; &lt;script&gt; &lt;/script&gt; </pre>	resources/js/Pages/Test1.vue
Home page	web	<pre> Route::get('/', function () {     return Inertia::render('Home'); }); </pre>	Home page
	Home template	<pre> &lt;script setup&gt;   import { Link } from '@inertiajs/vue3';   import NavLayout from '../Layout/NavLayout.vue'; &lt;/script&gt;  &lt;template&gt;   &lt;NavLayout&gt;     &lt;div class="container"&gt;       &lt;div class="row justify-content-center"&gt;         &lt;div class="col-md-7 col-lg-6 center-screen" &gt;           &lt;div class="card animated fadeIn w-90 p-4"&gt; </pre>	Home.vue



		<pre>import LoginForm from '../Components/User/LoginForm.vue'; &lt;/script&gt;</pre>	
	LoginForm.vue	<pre>&lt;template&gt;   &lt;div class="container"&gt;     &lt;div class="row justify-content-center"&gt;       &lt;div class="col-md-7 animated fadeIn col-lg-6 center-screen"&gt;         &lt;div class="card w-90 p-4"&gt;           &lt;form @submit.prevent="submit"&gt;             &lt;div class="card-body"&gt;               &lt;h4&gt;SIGN IN&lt;/h4&gt;               &lt;br/&gt;               &lt;input id="email" placeholder="User Email" class="form-control" type="email"/&gt;               &lt;br/&gt;               &lt;input id="password" placeholder="User Password" class="form- control" type="password"/&gt;               &lt;br/&gt;               &lt;button type="submit" class="btn w-100 btn-success"&gt;Login&lt;/button&gt;               &lt;hr/&gt;               &lt;div class="float-end mt-3"&gt;                 &lt;span&gt;                   &lt;Link class="text-center ms-3 h6" href="/registration"&gt;Sign Up &lt;/Link&gt;                   &lt;span class="ms-1"&gt; &lt;/span&gt;</pre>	js/Components/User /LoginForm.vue



	Login method modify purpose inertia	<pre> function userLogin(Request \$request){     // dd(\$request-&gt;all());     \$count = User::where('email', \$request- &gt;input('email'))-&gt;where('password', \$request- &gt;input('password'))-&gt;select('id')-&gt;first();     if(\$count !== null){          // User login -&gt; JWT token issue         // \$token = JWTToken::CreateToken(\$request-&gt;input('email'),\$count-&gt;id);          // return response()-&gt;json([         //     'status' =&gt; 'success',         //     'message' =&gt; 'User login successfully',         //     'token' =&gt; \$token         // ],200)-&gt;cookie('token', \$token, 60 * 24 * 30);          \$email = \$request-&gt;input('email');         \$user_id = \$count-&gt;id;          \$request-&gt;session()-&gt;put('email', \$email);         \$request-&gt;session()-&gt;put('user_id', \$user_id);          \$data = ['message'=&gt;'User login successfully','status'=&gt;true,'error'=&gt;''];         return redirect('/DashboardPage')- &gt;with(\$data); </pre>	Http/Controllers/UserContr oller.php
--	---	---	---

		<pre>         }else{              // return response()-&gt;json([             //     'status' =&gt; 'failed',             //     'message' =&gt; 'unauthorized'             // ],200);              \$data = ['message'=&gt;'Login faild','status'=&gt;false,'error'=&gt;''];             return redirect('/login')-&gt;with(\$data);         }     } </pre>	
		Php artisan make:middleware SessionAuthenticate	cmd
	SessionAuthenticate.php	<pre> public function handle(Request \$request, Closure \$next): Response{     \$email = \$request-&gt;session()- &gt;get('email','default');     \$user_id = \$request-&gt;session()- &gt;get('user_id','default');     if(\$email == 'default'){         return redirect('/login');     }else{         \$request-&gt;headers-&gt;set('email',\$email);         \$request-&gt;headers-&gt;set('id',\$user_id);         return \$next(\$request);     } } </pre>	app/Http/Middleware /SessionAuthenticate.php
	Web route change	<pre> Route::middleware(SessionAuthenticate::class)- &gt;group(function(){ </pre>	এখন TokenVerificationMiddlewar e এর পরিবর্তে



		<pre>Route::get('/invoice-delete/{id}', [InvoiceController::class, 'InvoiceDelete'])- &gt;name('InvoiceDelete'); });</pre>	SessionAuthenticate হবে কারণ এখন সেশন দিয়ে চেক করতে হবে
	Toster install	npm i @meforma/vue-toaster	cmd
	<b>LoginForm.vue</b>  <b>update</b>	<pre>&lt;template&gt;   &lt;div class="container"&gt;     &lt;div class="row justify-content-center"&gt;       &lt;div class="col-md-7 animated fadeIn col-lg-6 center-screen"&gt;         &lt;div class="card w-90 p-4"&gt;           &lt;form @submit.prevent="submit"&gt;             &lt;div class="card-body"&gt;               &lt;h4&gt;SIGN IN&lt;/h4&gt;               &lt;br/&gt;               &lt;input id="email" v-model="form.email" placeholder="User Email" class="form-control" type="email"/&gt;               &lt;br/&gt;               &lt;input id="password" v-model="form.password" placeholder="User Password" class="form-control" type="password"/&gt;               &lt;br/&gt;               &lt;button type="submit" class="btn w-100 btn-success"&gt;Login&lt;/button&gt;               &lt;hr/&gt;               &lt;div class="float-end mt-3"&gt;                 &lt;span&gt;                   &lt;Link class="text-center ms-3 h6" href="/registration"&gt;Sign Up &lt;/Link&gt;</pre>	resources/js/Components/User/LoginForm.vue

```
<span class="ms-1">|</span>  
<Link class="text-center  
ms-3 h6" href="send-otp">Forget Password</Link>  
    </span>  
  </div>  
</div>  
</form>  
</div>  
</div>  
</div>  
</div>  
</template>  
  
<script setup>  
  
// useForm == data send  
// usePage == response  
// router == route  
  
import { Link, useForm, usePage, router } from  
"@inertiajs/vue3";  
  import { createToaster } from "@meforma/vue-toaster";  
  const toaster = createToaster({ position: "top-right"  
});  
  
  const form = useForm({ email: "", password: "" });  
  const page = usePage();  
  
  function submit(){
```

		<pre>         if(form.email.length === 0){             toaster.warning("Email is required");         }else if(form.password.length === 0){             toaster.warning("Password is required");         }else{             form.post("/user-login", {                 onSuccess: () =&gt; {                     if(page.props.flash.status === true){                         router.get('/DashboardPage');                         toaster.success("Login successful");                     }else{                         toaster.error(page.props.flash.mess age);                     }                 }             });         }     } &lt;/script&gt; </pre>	
	DashboardPage method update	<pre> public function DashboardPage(Request \$request){     // \$user = \$request-&gt;header('email');     // return response()-&gt;json([     //     'status'=&gt;'success',     //     'message'=&gt;'user Login successfully',     //     'user'=&gt;\$user, </pre>	

		<pre>                 // ],200);                  return Inertia::render('DashboardPage');              } </pre>	
	DashBordPage template	<pre> &lt;template&gt;      &lt;SideNavLayout/&gt;  &lt;/template&gt; &lt;script setup&gt;     import SideNavLayout from     '../Layout/SideNavLayout.vue'; &lt;/script&gt; </pre>	DashBordPage.vue
Logout method    update			
	Logout method	<pre> public function UserLogout(Request \$request){     // return response()-&gt;json([     //     'status' =&gt; 'success',     //     'message' =&gt; 'User logout successfully',     // ],200)-&gt;cookie('token', '', -1);      \$request-&gt;session()-&gt;forget('email');     \$request-&gt;session()-&gt;forget('user_id');      \$data = ['message'=&gt;'User logout successfully','status'=&gt;true,'error'=&gt;'']; </pre>	app/Http/Controllers UserController.php

		<pre>return redirect('/login')-&gt;with(\$data);  }</pre>	
Registration update			
	Web route	<pre>Route::get('/registration',[UserController::class, 'RegistrationPage'])-&gt;name('registration.page');</pre>	
	RegistrationPage method	<pre>public function RegistrationPage(Request \$request){     return Inertia::render('RegistrationPage'); }</pre>	UserController.php
	RegistrationPage.vue	<pre>&lt;template&gt;   &lt;NavLayout&gt;     &lt;RegistrationForm/&gt;   &lt;/NavLayout&gt;  &lt;/template&gt; &lt;script setup&gt; import RegistrationForm from '../Components/User/RegistrationForm.vue'; import NavLayout from '../Layout/NavLayout.vue';  &lt;/script&gt;</pre>	resources/js/Pages /RegistrationPage.vue
	RegistrationForm.vue	<pre>&lt;template&gt;   &lt;div class="container d-flex justify-content-center align-items-center min-vh-100"&gt;     &lt;!-- Full viewport height and center alignment --&gt;</pre>	js/Components/User /RegistrationForm.vue

```

<div class="row justify-content-center w-100">
  <div class="col-md-10 col-lg-6">
    <div class="card animated fadeIn w-100 p-3">
      <form @submit.prevent="submit">
        <div class="card-body">
          <h4>Sign Up</h4>
          <hr />
          <div class="container-fluid m-0 p-0">
            <div class="row m-0 p-0">
              <div class="col-md-12 p-2">
                <label>Name</label>
                <input id="name" v-
model="form.name" placeholder="First Name" class="form-
control" type="text" />
              </div>
              <div class="col-md-12 p-2">
                <label>Email Address</label>
                <input id="email" v-model="form.email"
placeholder="User Email" class="form-control" type="email"
/>
              </div>
              <div class="col-md-12 p-2">
                <label>Mobile Number</label>
                <input id="mobile" v-
model="form.mobile" placeholder="Mobile" class="form-
control" type="mobile" />
              </div>
              <div class="col-md-12 p-2">
                <label>Password</label>
                <input id="password" v-
model="form.password" placeholder="User Password"
class="form-control" type="password" />
            </div>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>

```

```

    </div>
    <div class="col-md-12 p-2">
      <label>Confirm Password</label>
      <input id="cpassword" placeholder="User
Confirm Password" class="form-control" type="password" />
    </div>
  </div>
  <div class="row m-0 p-0">
    <div class="col-md-12 p-2">
      <button type="submit" class="btn mt-3
w-100 btn-success">Complete</button>
    </div>
  </div>
</div>
</div>
</form>
</div>
</div>
</div>
</div>
</template>
<script setup>
import { Link, useForm, usePage, router } from
"@inertiajs/vue3";
import { createToaster } from "@meforma/vue-toaster";
const toaster = createToaster();

const form = useForm({ name: "", email: "", mobile: "",
password: "" });
const page = usePage();

function submit(){

```

		<pre> if(form.name.length === 0){     toaster.warning("Name is required"); }else if(form.email.length === 0){     toaster.warning("Email is required"); }else if(form.mobile.length === 0){     toaster.warning("Mobile is required"); }else if(form.password.length === 0){     toaster.warning("Password is required"); }else{     form.post("/user-registration", {         onSuccess: () =&gt; {             if(page.props.flash.status === true){                 router.get('/DashboardPage');                 toaster.success("Login successful");             }else{                 toaster.error(page.props.flash.mess age);             }         }     }); } } &lt;/script&gt; </pre>	
	UserRegistration method	<pre> function UserRegistration(Request \$request){      // dd(\$request-&gt;all());     try {         \$request-&gt;validate([ </pre>	app/Http/Controllers /UserController.php



```
        'name' => 'required',
        'email' =>
'required|email|unique:users,email',
        'password' => 'required',
        'mobile' => 'required',
    ]);

$email=$request->input('email');
$name=$request->input('name');
$mobile=$request->input('mobile');
$password=$request->input('password');

$user = User::create([
    'name'=>$name,
    'email'=>$email,
    'mobile'=>$mobile,
    'password'=>$password
]);

// return response()->json([
//     'status' => 'success',
//     'message' => 'User created
successfully',
//     'data' => $user
// ],200);

$data = ['message'=>'User created
successfully','status'=>true,'error'=>''];
return redirect('/DashboardPage')->with($data);

}
```

		<pre>         catch (Exception \$e) {             // return response()-&gt;json([                 //     'status' =&gt; 'failed',                 //     'message' =&gt; \$e-&gt;getMessage()             // ],200);              \$data = ['message'=&gt;'Something went wrong','status'=&gt;false,'error'=&gt;''];             return redirect('/registration')-&gt;with(\$data);         }     } </pre>	
			1.42 hour
Forget Password			
	web	<pre> Route::get('/send-otp',[UserController::class, 'SendOTPPage']-&gt;name('sendotp.page')); </pre>	
	SendOTPPage method	<pre> public function SendOTPPage(Request \$request){     return Inertia::render('SendOTPPage'); } </pre>	app/Http/Controllers /UserController.php
	SendOTPPage.vue	<pre> &lt;script setup&gt; import NavLayout from '../Layout/NavLayout.vue'; import SendOTPForm from '../Components/User/SendOTPForm.vue'; &lt;/script&gt;  &lt;template&gt;   &lt;NavLayout&gt;     &lt;SendOTPForm/&gt;   &lt;/template&gt; </pre>	resources/js/Pages /SendOTPPage.vue

		<pre>         &lt;/NavLayout&gt; &lt;/template&gt;  &lt;style scoped&gt;  &lt;/style&gt; </pre>	
	SendOTPForm.vue	<pre> &lt;template&gt;   &lt;div class="container"&gt;     &lt;div class="row justify-content-center"&gt;       &lt;div class="col-md-7 col-lg-6 center-screen"&gt;         &lt;div class="card animated fadeIn w-90 p-4"&gt;           &lt;form @submit.prevent="submit"&gt;             &lt;div class="card-body"&gt;               &lt;h4&gt;EMAIL ADDRESS&lt;/h4&gt;               &lt;br/&gt;               &lt;label&gt;Your email address&lt;/label&gt;               &lt;input id="email" v- model="form.email" placeholder="User Email" class="form- control" type="email"/&gt;               &lt;br/&gt;               &lt;button type="submit" class="btn mt-3 w-100 btn-success"&gt;Next&lt;/button&gt;             &lt;/div&gt;           &lt;/form&gt;         &lt;/div&gt;       &lt;/div&gt;     &lt;/div&gt;   &lt;/div&gt; &lt;/template&gt; </pre>	resources/js/Components/User/SendOTPForm.vue

```

</template>

<script setup>
  import { Link, useForm, usePage, router } from
"@inertiajs/vue3";
  import { createToaster } from "@meforma/vue-toaster";
  const toaster = createToaster({ position: "top-right"
});

  const form = useForm({ email: "", password: "" });
  const page = usePage();

  function submit(){
    if(form.email.length === 0){
      toaster.warning("Email is required");
    }else{
      form.post("/send-otp", {
        onSuccess: () => {
          if(page.props.flash.status === true){
            router.get('/verify-otp');
            toaster.success(page.props.flash.me
ssage);
          }else{
            toaster.error(page.props.flash.mess
age);
          }
        }
      });
    }
  }
</script>

```

	SendOTPCode Method modify	<pre> public function SendOTPCode(Request \$request){     \$email = \$request-&gt;input('email');     \$otp = rand(1000,9999);      \$count = User::where('email',\$email)-&gt;count();      if(\$count == 1){         // Mail::to(\$email)-&gt;send(new OTPMail(\$otp));         User::where('email', \$email)-&gt;update(['otp' =&gt; \$otp]);         \$request-&gt;session()-&gt;put('email', \$email);         // return response()-&gt;json([         //     'status' =&gt; 'success',         //     'message' =&gt; "4 Digit {\$otp} OTP send successfully",         // ],200);          \$data = ["message"=&gt;"4 Digit {\$otp} OTP send successfully","status"=&gt;true,"error"=&gt;''];         return redirect('/verify-otp')-&gt;with(\$data);     }else{         // return response()-&gt;json([         //     'status' =&gt; 'fail',         //     'message' =&gt; 'unauthorized'         // ]);          \$data = ['message'=&gt;'unauthorized','status'=&gt;false,'error'=&gt;''];         return redirect('/registration')-&gt;with(\$data);     } } </pre>	

	Web	<pre>Route::get('/verify-otp',[UserController::class, 'VerifyOTPPage']-&gt;name('VerifyOTPPage'));</pre>	
		<pre>public function VerifyOTPPage(Request \$request){     return Inertia::render('VerifyOTPPage'); }</pre>	
		<pre>&lt;script setup&gt; import NavLayout from '../Layout/NavLayout.vue'; import VerifyOTPForm from '../Components/User/VerifyOTPForm.vue'; &lt;/script&gt;  &lt;template&gt;   &lt;NavLayout&gt;     &lt;VerifyOTPForm/&gt;   &lt;/NavLayout&gt; &lt;/template&gt;  &lt;style scoped&gt;  &lt;/style&gt;</pre>	
	<b>VerifyOTPForm.vue</b>	<pre>&lt;template&gt;   &lt;div class="container"&gt;     &lt;div class="row justify-content-center"&gt;       &lt;div class="col-md-7 col-lg-6 center-screen"&gt;         &lt;div class="card animated fadeIn w-90 p-4"&gt;            &lt;form @submit.prevent="submit"&gt;             &lt;div class="card-body"&gt;</pre>	resources/js/Components/User/VerifyOTPForm.vue

```

        <h4>ENTER OTP CODE</h4>
        <br/>
        <label>4 Digit Code
Here</label>

        <input id="otp" v-
model="form.otp" placeholder="Code" class="form-control"
type="text"/>

        <br/>
        <button type="submit"
class="btn w-100 btn-success">Next</button>
    </div>
</form>
</div>
</div>
</div>
</div>
</template>

<script setup>
import { Link, useForm, usePage, router } from
"@inertiajs/vue3";
import { createToaster } from "@meforma/vue-toaster";
const toaster = createToaster({ position: "top-right" });

const form = useForm({ otp: ""});
const page = usePage();

function submit(){
    if(form.otp.length === 0){
        toaster.warning("OTP is required");
    }else{
        form.post("/verify-otp", {

```

		<pre> onSuccess: () =&gt; {     if(page.props.flash.status === true){         router.get('/reset-password');         toaster.success(page.props.flash.me ssage);     }else{         toaster.error(page.props.flash.mess age);     } } }); } } &lt;/script&gt; </pre>	
	verifyOTP method update	<pre> public function VerifyOTP(Request \$request){     // \$email = \$request-&gt;input('email');     \$email = \$request-&gt;session()-&gt;get('email');     \$otp = \$request-&gt;input('otp');      \$count = User::where('email', \$email)-&gt;where('otp', \$otp)-&gt;count();      if(\$count == 1){         User::where('email', \$email)-&gt;update(['otp' =&gt; 0]);          // \$token = JWTToken::CreateTokenForSetPassword(\$request- &gt;input('email')); </pre>	



		<pre>                 \$request-&gt;session()-&gt;put('otp_verify','yes');                  // return response()-&gt;json([                 //     'status' =&gt; 'success',                 //     'message' =&gt; 'OTP verification successfully',                 // ],200)-&gt;cookie('token', \$token, 60 * 24 * 30);                  \$data = ["message"=&gt;"OTP verification successfully","status"=&gt;true,"error"=&gt;''];                 return redirect('/reset-password')-&gt;with(\$data);             }else{                 // return response()-&gt;json([                 //     'status' =&gt; 'fail',                 //     'message' =&gt; 'unauthorized'                 // ]);                 \$data = ['message'=&gt; 'unauthorized','status'=&gt;false, 'error'=&gt;''];                 return redirect('/login')-&gt;with(\$data);             }         }     } </pre>	
	web	<pre> Route::middleware(SessionAuthenticate::class)-&gt;group(function(){     Route::get('/reset-password',[UserController::class, 'ResetPasswordPage']); }); </pre>	app/Http/Controllers /UserController.php

		<pre> public function ResetPasswordPage(Request \$request){     return Inertia::render('ResetPasswordPage'); } </pre>	
	ResetPasswordPage.vue	<pre> &lt;script setup&gt; import NavLayout from '../Layout/NavLayout.vue'; import ResetPasswordForm from '../Components/User/ResetPasswordForm.vue'; &lt;/script&gt;  &lt;template&gt;   &lt;NavLayout&gt;     &lt;ResetPasswordForm/&gt;   &lt;/NavLayout&gt; &lt;/template&gt;  &lt;style scoped&gt;  &lt;/style&gt; </pre>	resources/js/Pages/ResetPasswordPage.vue
	ResetPasswordForm.vue	<pre> &lt;template&gt;   &lt;div class="container"&gt;     &lt;div class="row justify-content-center"&gt;       &lt;div class="col-md-7 col-lg-6 center-screen"&gt;         &lt;div class="card animated fadeIn w-90 p-4"&gt;           &lt;form @submit.prevent="submit"&gt;             &lt;div class="card-body"&gt;               &lt;h4&gt;SET NEW PASSWORD&lt;/h4&gt;               &lt;br/&gt;               &lt;label&gt;New Password&lt;/label&gt; </pre>	resources/js/Components/User/ResetPasswordForm.vue

```

        <input id="password" v-
model="form.password" placeholder="New Password"
class="form-control" type="password"/>
        <br/>
        <label>Confirm Password</label>
        <input id="cpassword" v-
model="form.cpassword" placeholder="Confirm Password"
class="form-control" type="password"/>
        <br/>
        <button type="submit"
class="btn w-100 btn-success">Next</button>
    </div>
</form>
</div>
</div>
</div>
</div>
</template>

<script setup>
import { Link, useForm, usePage, router } from
"@inertiajs/vue3";
import { createToaster } from "@meforma/vue-toaster";
const toaster = createToaster({ position: "top-right" });

const form = useForm({ password: "", cpassword: ""});
const page = usePage();

function submit(){
    if(form.password.length === 0){
        toaster.warning("Password is required");
    }else if(form.cpassword.length === 0){

```

		<pre>         toaster.warning("Confirm Password is required");     }else if(form.password !== form.cpassword){         toaster.warning("Password and Confirm Password does not match");     }else{         form.post("/reset-password", {             onSuccess: () =&gt; {                 if(page.props.flash.status === true){                     router.get('/login');                     toaster.success(page.props.flash.messag e);                 }else{                     toaster.error(page.props.flash.message) ;                 }             }         });     } } &lt;/script&gt; </pre>	
	ResetPassword method modify	<pre> public function ResetPassword(Request \$request){     try{         // \$email = \$request-&gt;header('email');         \$email = \$request-&gt;session()- &gt;get('email','default');         \$password = \$request-&gt;input('password');          \$otp_verify = \$request-&gt;session()- &gt;get('otp_verify','default');         if(\$otp_verify === 'yes'){ </pre>	

```

        User::where('email', $email)-
>update(['password' => $password]);
        $request->session()->flush();

        $data = ['message'=> 'Password reset
successfully','status'=>true, 'error'=>' '];
        return redirect('/login')->with($data);
    }else{
        $data = ['message'=> 'Request
fail','status'=>false, 'error'=>' '];
        return redirect('/reset-password')-
>with($data);
    }
    // return response()->json([
    //     'status' => 'success',
    //     'message' => 'Password reset
successfully',
    // ],200);

    }catch(Exception $e){
        // return response()->json([
        //     'status' => 'fail',
        //     'message' => 'something went wrong'
        // ]);
        $data = ['message'=> $e-
>getMessage(),'status'=>false, 'error'=>' '];
        return redirect('/reset-password')-
>with($data);
    }
}

```

2.31 hour

Category Page			
	web	<pre>Route::middleware(SessionAuthenticate::class)- &gt;group(function(){     Route::get('/CategoryPage', [CategoryController::class, 'CategoryPage'])- &gt;name('CategoryPage'); });</pre>	
	CategoryPage Method	<pre>public function CategoryPage(Request \$request){     \$user_id = \$request-&gt;header('id');     \$categories = Category::where('user_id', \$user_id)- &gt;get();     return Inertia::render('CategoryPage',[ 'categories'=&gt;\$categories]) ; }</pre>	app/Http/Controllers /CategoryController.php
	CategoryPage.vue	<pre>&lt;template&gt;     &lt;SideNavLayout&gt;         &lt;CategoryList/&gt;     &lt;/SideNavLayout&gt; &lt;/template&gt; &lt;script setup&gt;     import SideNavLayout from '../Layout/SideNavLayout.vue';     import CategoryList from '../Components/Category/CategoryList.vue'; &lt;/script&gt;</pre>	resources/js/Pages /CategoryPage.vue
	CategoryList.vue	<pre>&lt;template&gt;     &lt;div class="container-fluid"&gt;         &lt;div class="row"&gt;</pre>	resources/js/Components/C ategory/CategoryList.vue

```

<div class="col-12"></div>
<div class="col-12">
  <div class="card">
    <div class="card-body">
      <div>
        <h3>Category</h3>
      </div>
      <hr />
      <div class="float-end">
        <Link
href="/CategorySavePage?id=0" class="btn btn-success mx-3
btn-sm">
          Add Category
        </Link>
      </div>

      <!-- Modal -->

      <div>
        <input placeholder="Search..."
class="form-control mb-2 w-auto form-control-sm"
type="text"

        v-model="searchValue">
        <EasyDataTable
          buttons-pagination
          alternating
          :headers="Header"
          :items="Item"
          :rows-per-page="10"
          :search-field="searchField"
          :search-value="searchValue"
          show-index

```

```
>
<template #item-number="{
id, name }">

    <a class="btn btn-
success mx-3 btn-sm"
:href="/CategorySavePage?id=${id}">Edit</a>
    <button class="btn btn-
danger btn-sm" @click="DeleteClick(id)">Delete</button>
    </template>
</EasyDataTable>
</div>
</div>
</div>
</div>
</div>
</template>

<script setup>
import { Link, useForm, usePage, router } from
"@inertiajs/vue3";
import { createToaster } from "@meforma/vue-toaster";
const toaster = createToaster({ position: "top-right"
});

import { ref } from "vue";

let page = usePage();
const Header = [
    { text: "Name", value: "name" },
    { text: "Action", value: "number" },
]
```



		<pre> const Item = ref(page.props.categories);  const searchValue = ref(); const DeleteClick = (id) =&gt; {   let text = "Do you want to delete"   if(confirm(text)===true){     router.get(`/delete-category/\${id}`);     toaster.success("Category Deleted successfully");   }else{     text = "you canceled!";   } } &lt;/script&gt; </pre>	
	CategoryDelete method update	<pre> public function CategoryDelete(Request \$request,\$id){   \$user_id = \$request-&gt;header('id');   Category::where('user_id', \$user_id)-&gt;where('id', \$id)-&gt;delete();   // return response()-&gt;json([   //   'status' =&gt; 'success',   //   'message' =&gt; 'Category Deleted successfully'   // ]);    \$data = ['message'=&gt;'Category Deleted successfully','status'=&gt;true,'error'=&gt;''];   return redirect('/CategoryPage')-&gt;with(\$data);  } </pre>	app/Http/Controllers /CategoryController.php

	Web	<pre>Route::get('/CategorySavePage', [CategoryController::class, 'CategorySavePage'])-&gt;name('CategorySavePage');</pre>	
	CategorySavePage Method	<pre>public function CategorySavePage(Request \$request){     \$category_id = \$request-&gt;query('id');     \$user_id = \$request-&gt;header('id');     \$category = Category::where('id', \$category_id)-&gt;where('user_id', \$user_id)-&gt;first();     return     Inertia::render('CategorySavePage', ['category'=&gt;\$category])     ; }</pre>	app/Http/Controllers/CategoryController.php
	CategorySavePage.vue	<pre>&lt;template&gt;   &lt;SideNavLayout&gt;     &lt;CategorySaveForm/&gt;   &lt;/SideNavLayout&gt; &lt;/template&gt; &lt;script setup&gt;   import SideNavLayout from   '../Layout/SideNavLayout.vue';   import CategorySaveForm from   '../Components/Category/CategorySaveForm.vue'; &lt;/script&gt;</pre>	resources/js/Pages/CategorySavePage.vue
	CategorySaveForm.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;</pre>	resources/js/Components/Category/CategorySaveForm.vue

```

<div class="card-body">
  <div class="float-end">
    <Link href="/CategoryPage"
class="btn btn-success mx-3 btn-sm">
      Back
    </Link>
  </div>
  <form @submit.prevent="submit">
    <div class="card-body">
      <h4>Save Category</h4>
      <input id="name" v-
model="form.name" name="name" placeholder="Category Name"
class="form-control"
type="text" />
      <br />
      <button type="submit"
class="btn w-100 btn-success">Save Change</button>
    </div>
  </form>
</div>
</div>
</div>
</div>
</div>
</template>

<script setup>
import { Link, useForm, usePage, router } from
"@inertiajs/vue3";
import { createToaster } from "@meforma/vue-toaster";
const toaster = createToaster({ position: "top-right" });
import { ref } from "vue";

```

```
const urlParams = new
URLSearchParams(window.location.search);
let id = ref(parseInt(urlParams.get("id")));

const form = useForm({ name: "", id: id });
const page = usePage();

let URL = "/create-category";
let category = page.props.category;

if (id.value !== 0 && category !== null) {
  URL = "/update-category";
  form.name = category.name;
}

function submit() {
  if (form.name.length === 0) {
    toaster.warning("Name is required");
  } else {
    form.post(URL, {
      onSuccess: () => {
        if (page.props.flash.status === true) {
          router.get('/CategoryPage');
          toaster.success(page.props.flash.message);
        } else {
          toaster.error(page.props.flash.message);
        }
      }
    });
  }
}
```

		<pre>     } }  &lt;/script&gt; </pre>	
	CreateCategory method modify	<pre> public function CreateCategory(Request \$request){     \$user_id = \$request-&gt;header('id');     Category::create([         'name' =&gt; \$request-&gt;name,         'user_id' =&gt; \$user_id     ]);     // return response()-&gt;json([     //     'status' =&gt; 'success',     //     'message' =&gt; 'Category created successfully'     // ]);      \$data = ['message'=&gt;'Category created successfully','status'=&gt;true,'error'=&gt;''];     return redirect('/CategoryPage')-&gt;with(\$data);  } </pre>	app/Http/Controllers /CategoryController.php
	CreateUpdate modify	<pre> public function CategoryUpdate(Request \$request){     \$user_id = \$request-&gt;header('id');     \$id = \$request-&gt;input('id');     Category::where('id', \$id)-&gt;where('user_id', \$user_id)-&gt;update([         'name' =&gt; \$request-&gt;input('name')     ]);     // return response()-&gt;json([     //     'status' =&gt; 'success', </pre>	app/Http/Controllers /CategoryController.php



[illegible]