|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Basic setup | | | |
|  |  |  |  |
|  |  | composer global require laravel/installer |  |
|  |  | laravel new inventory-shop-11 | None kit |
|  |  |  |  |
|  |  | Php artisan install:api |  |
|  | Bootstrap/app.php | ->withMiddleware(function (Middleware $middleware) {  $middleware->validateCsrfTokens(except: [  '\*'  ]);  }) | CSRF Disable check purpose Postman |
| **User method** | | | |
| Registration | | | |
|  |  | php artisan make:controller UserController | cmd |
|  | User migration modify | [User migration](database/migrations/0001_01_01_000000_create_users_table.php) |  |
|  | User model modify | [User model](app/Models/User.php) |  |
|  |  | [UserRegistration](app/Http/Controllers/UserController.php) method |  |
|  | Web/ api | Route::post('/user-registration', [UserController::class, 'userRegistration'])->name('userRegistration'); |  |
|  |  |  |  |
| Login | | | |
|  |  |  |  |
|  |  | composer require firebase/php-jwt | cmd |
|  |  | App/Helper |  |
|  | Secreate toeken | JWT\_KEY=123XYSPOHBN7864Wlkp | .env |
|  |  | [CreateToken](app/Helper/JWTToken.php) method | JWTToken.php |
|  |  | [UserLogin](app/Http/Controllers/UserController.php) method | UserController.php |
|  | Web & API | Route::post('/user-login', [UserController::class, 'userLogin'])->name('userLogin'); | Check postman |
|  | | | |
|  |  | [UserLogout](app/Http/Controllers/UserController.php) method | UserController.php |
|  | Web | Route::get('/logout', [UserController::class, 'userLogout'])->name('userLogout'); | Check postman |
| send-otp | | | |
|  |  | Php artisan make:mail OTPMail | cmd |
|  |  | Mail[/ OTPMail](app/Mail/OTPMail.php).php |  |
|  | View | email/ [OTPMail](resources/views/email/OTPMail.blade.php).blade.php |  |
|  | .env | MAIL\_MAILER=smtp  MAIL\_HOST=sandbox.smtp.mailtrap.io  MAIL\_PORT=2525  MAIL\_USERNAME=53ecad6e321186  MAIL\_PASSWORD=a86f9e136620f6  MAIL\_FROM\_NAME="Sales Inventory Practice API" | configuration |
|  |  | [SendOTPCode](app/Http/Controllers/UserController.php) method | UserController.php |
|  | Web & API | Route::post('/send-otp', [UserController::class, 'SendOTPCode'])->name('SendOTPCode'); |  |
|  |  |  |  |
| **VerifyOTP** | | | |
|  |  | [CreateTokenForSetPassword](app/Helper/JWTToken.php) method | App/Helper.php |
|  |  | [VerifyOTP](app/Http/Controllers/UserController.php) method | UserController.php |
|  | Web & API | Route::post('/verify-otp', [UserController::class, 'VerifyOTP'])->name('VerifyOTP'); |  |
|  |  |  |  |
| reset-password | | | |
|  |  | [VerifyToken](app/Helper/JWTToken.php) method | App/Helper.php |
|  |  | Php artisan make:middleware TokenVerificationMiddleware | cmd |
|  |  | Php artisan make:middleware TokenVerificationAPIMiddleware |
|  |  | [TokenVerificationMiddleware](app/Http/Middleware/TokenVerificationMiddleware.php) & [TokenVerificationAPIMiddleware](app/Http/Middleware/TokenVerificationAPIMiddleware.php) | class |
|  |  | [ResetPassword](app/Http/Controllers/UserController.php) method | UserController.php |
|  | Web | Route::post('/reset-password',[UserController::class,'ResetPassword'])->middleware([TokenVerificationMiddleware::class]); |  |
|  | API | Route::post('/reset-password',[UserController::class,'ResetPassword'])->middleware([TokenVerificationAPIMiddleware::class]); |  |
|  |  |  |  |
| UserProfile | | | |
|  |  | [UserProfile](app/Http/Controllers/UserController.php) method | UserController.php |
|  | Web | Route::get('/user-profile',[UserController::class,'UserProfile'])->middleware([TokenVerificationMiddleware::class]); |  |
|  | API | Route::get('/user-profile',[UserController::class,'UserProfile'])->middleware([TokenVerificationAPIMiddleware::class]); |  |
|  |  |  |  |
| UpdateProfile | | | |
|  |  | [UpdateProfile](app/Http/Controllers/UserController.php) method | UserController.php |
|  | Web | Route::post('/user-update',[UserController::class,'UpdateProfile'])->middleware([TokenVerificationMiddleware::class]); |  |
|  | API | Route::get('/user-profile',[UserController::class,'UserProfile'])->middleware([TokenVerificationAPIMiddleware::class]); |  |
|  |  |  |  |
| Category | | | |
|  |  | php artisan make:migration categories | cmd |
|  |  | php artisan make:model Category |
|  |  | php artisan make:controller CategoryController |
|  |  | [2025\_01\_19\_131909\_categories](database/migrations/2025_01_19_131909_categories.php).php |  |
|  |  | [Category](app/Models/Category.php) Model |  |
|  |  | Php artisan migrate | cmd |
|  |  |  |  |
|  |  | [CategoryCreate](app/Http/Controllers/CategoryController.php) Method | CategoryController |
|  | Web | Route::post("/create-category",[CategoryController::class,'CategoryCreate'])->middleware([TokenVerificationMiddleware::class]); |  |
|  | API | Route::post("/create-category",[CategoryController::class,'CategoryCreate'])->middleware([TokenVerificationAPIMiddleware::class]); |  |
|  |  |  |  |
|  |  | [CategoryList](app/Http/Controllers/CategoryController.php) Method | CategoryController |
|  | Web | Route::get("/list-category",[CategoryController::class,'CategoryList'])->middleware([TokenVerificationMiddleware::class]); |  |
|  | API | Route::get("/list-category",[CategoryController::class,'CategoryList'])->middleware([TokenVerificationAPIMiddleware::class]); |  |
|  |  |  |  |
|  |  | [CategoryByID](app/Http/Controllers/CategoryController.php) Method | CategoryController |
|  | Web | Route::post("/category-by-id",[CategoryController::class,'CategoryByID'])->middleware([TokenVerificationMiddleware::class]); |  |
|  | API | Route::post("/category-by-id",[CategoryController::class,'CategoryByID'])->middleware([TokenVerificationAPIMiddleware::class]); |  |
|  |  |  |  |
|  |  |  |  |
|  |  | [CategoryUpdate](app/Http/Controllers/CategoryController.php) Method | CategoryController |
|  | Web | Route::post("/update-category",[CategoryController::class,'CategoryUpdate'])->middleware([TokenVerificationMiddleware::class]); |  |
|  | API | Route::post("/update-category",[CategoryController::class,'CategoryUpdate'])->middleware([TokenVerificationAPIMiddleware::class]); |  |
|  |  |  |  |
|  |  |  |  |
|  |  | CategoryDelete Method | CategoryController |
|  | Web | Route::post("/delete-category",[CategoryController::class,'CategoryDelete'])->middleware([TokenVerificationMiddleware::class]); |  |
|  | API | Route::post("/delete-category",[CategoryController::class,'CategoryDelete'])->middleware([TokenVerificationAPIMiddleware::class]); |  |
|  |  |  |  |
|  |  |  |  |
| **Customer** | | | |
|  |  | php artisan make:model Customer | cmd |
|  |  | php artisan make:migration customers |
|  |  | php artisan make:controller CustomerController |
|  |  |  |  |
|  |  | [2025\_01\_21\_151603\_customers](database/migrations/2025_01_21_151603_customers.php).php | modify |
|  | Model | [Customer](app/Models/Customer.php).php | modify |
|  |  | Php artisan migrate | cmd |
|  |  | [CustomerCreate](app/Http/Controllers/CustomerController.php) method | CustomerController |
|  | Web | Route::post("/create-customer",[CustomerController::class,'CustomerCreate'])->middleware([TokenVerificationMiddleware::class]); |  |
|  | API | Route::post("/create-customer",[CustomerController::class,'CustomerCreate'])->middleware([TokenVerificationAPIMiddleware::class]); | Not working |
|  |  |  |  |
| **Product** | | | |
| **Create** | | | |
|  |  | php artisan make:model Product | cmd |
|  |  | php artisan make:migration products |
|  |  | php artisan make:controller ProductController |
|  |  | Uploads | Make folder inside of public folder |
|  |  | [2025\_01\_21\_151603\_customers](database/migrations/2025_01_21_151603_customers.php).php | modify |
|  | Model | [Product](app/Models/Product.php).php | modify |
|  |  | Php artisan migrate | cmd |
|  |  | [CreateProduct](app/Http/Controllers/ProductController.php) method | ProductController |
|  | Web | Route::post("/create-product",[ProductController::class,'CreateProduct'])->middleware([TokenVerificationMiddleware::class]); |  |
|  |  |  |  |
|  |  |  |  |
|  |  | **ProductList** |  |
|  |  | [ProductList](app/Http/Controllers/ProductController.php) Method | ProductController |
|  | Web | Route::get("/list-product",[ProductController::class,'ProductList'])->middleware([TokenVerificationMiddleware::class]); |  |
|  | API | Route::get("/list-product",[ProductController::class,'ProductList'])->middleware([TokenVerificationAPIMiddleware::class]); |  |
|  |  |  |  |
|  |  |  |  |
|  |  | Single product |  |
|  |  | [ProductByID](app/Http/Controllers/ProductController.php) Method | ProductList Method |
|  | Web | Route::post("/product-by-id",[ProductController::class,'ProductByID'])->middleware([TokenVerificationMiddleware::class]); |  |
|  | API | Route::post("/product-by-id",[ProductController::class,'ProductByID'])->middleware([TokenVerificationAPIMiddleware::class]); |  |
|  |  |  |  |
|  |  |  |  |
|  |  | **Delete** |  |
|  |  | [DeleteProduct](app/Http/Controllers/ProductController.php) Method | ProductList Method |
|  | Web | Route::post("/delete-product",[ProductController::class,'DeleteProduct'])->middleware([TokenVerificationMiddleware::class]); |  |
|  | API | Route::post("/delete-product",[ProductController::class,'DeleteProduct'])->middleware([TokenVerificationAPIMiddleware::class]); |  |
|  |  |  |  |
|  |  |  |  |
|  |  | **UpdateProduct** |  |
|  |  | [UpdateProduct](app/Http/Controllers/ProductController.php)  Method | ProductList Method |
|  | Web | Route::post("/update-product",[ProductController::class,'UpdateProduct'])->middleware([TokenVerificationMiddleware::class]); |  |
|  | API | Route::post("/update-product",[ProductController::class,'UpdateProduct'])->middleware([TokenVerificationAPIMiddleware::class]); |  |
|  |  |  |  |
|  |  | **Invoice** |  |
|  |  |  |  |
|  |  | php artisan make:migration invoices | laravel 10 migration |
|  |  | php artisan make:migration invoice\_products |
|  |  |  |  |
|  |  | php artisan make:migration create\_flights\_table | laravel 11 migration |
|  |  | php artisan make:model InvoiceProduct -m | laravel 11 model + migration |
|  |  |  |  |
|  |  | [2025\_01\_28\_184939\_invoices](database/migrations/2025_01_28_184939_invoices.php).php & [2025\_01\_28\_185018\_invoice\_products](database/migrations/2025_01_28_185018_invoice_products.php).php |  |
|  |  |  |  |
|  |  | php artisan make:model Invoice | model |
|  |  | php artisan make:model InvoicePorduct |
|  |  | [Invoice](app/Models/Invoice.php).php & [InvoicePorduct](app/Models/InvoicePorduct.php).php |  |
|  |  | php artisan migrate | cmd |
|  |  |  |  |
|  |  | php artisan make:controller InvoiceController | controller |
|  |  |  |  |
|  |  | [invoiceCreate](app/Http/Controllers/InvoiceController.php) Method | controller |
|  |  |  |  |
|  | Web | Route::post("/invoice-create",[InvoiceController::class,'invoiceCreate'])->middleware([TokenVerificationMiddleware::class]); |  |
|  |  |  |  |
|  |  | [invoiceSelect](app/Http/Controllers/InvoiceController.php) Method | InvoiceController |
|  |  |  |  |
|  | Web | Route::get("/invoice-select",[InvoiceController::class,'invoiceSelect'])->middleware([TokenVerificationMiddleware::class]); |  |
|  |  |  |  |
|  |  | [invoice-details](app/Http/Controllers/InvoiceController.php)  Method | InvoiceController |
|  |  |  |  |
|  | Web | Route::post("/invoice-details",[InvoiceController::class,'InvoiceDetails'])->middleware([TokenVerificationMiddleware::class])->name('invoiceDetails'); |  |
|  |  |  |  |
|  |  |  |  |
|  |  | [invoiceDelete](app/Http/Controllers/InvoiceController.php) Method | InvoiceController |
|  |  |  |  |
|  | Web | Route::post("/invoice-delete",[InvoiceController::class,'invoiceDelete'])->middleware([TokenVerificationMiddleware::class])->name('invoiceDelete'); |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| dashbord | | | |
|  |  | php artisan make:controller DashboardController |  |
|  |  | [DashboardPage](app/Http/Controllers/DashBoardController.php) Method | DashboardController |
|  |  | [pages.dashboard.dashboard-page.blade](resources/views/pages/dashboard/dashboard-page.blade.php).php | view |
|  | Web | Route::get('/dashboard',[DashboardController::class,'DashboardPage'])->middleware([TokenVerificationMiddleware::class]); |  |
|  |  |  |  |
| Auth page | | | |
|  |  | [LoginPage](app/Http/Controllers/UserController.php) Method | UserController |
| [Tutorial prepared 12 backend](file:///D:\project%20guidline%20sales%20inventory\1%20ইনভেন্টরি%20প্রোজেক্ট%20ব্যাক-ইন্ড%20ডেভেলপমেন্ট\Ostad%20-%2012%20%5bPOS%5d%20Front%20End%20Explain%20Front%20End%20Assets.ts) |  | [RegistrationPage](app/Http/Controllers/UserController.php) Method |
|  | [SendOtpPage](app/Http/Controllers/UserController.php) Method |
|  | [VerifyOTPPage](app/Http/Controllers/UserController.php) Method |
|  | [ProfilePage](app/Http/Controllers/UserController.php) Method |
|  | [ResetPasswordPage](app/Http/Controllers/UserController.php) Method |
|  | [DashboardPage](app/Http/Controllers/DashBoardController.php) Method | DashBoardController |
|  |  |  |
|  |  |  |
|  | [login-page](resources/views/pages/auth/login-page.blade.php) | View |
|  | [registration-page](resources/views/pages/auth/registration-page.blade.php) |
|  | [send-otp-page](resources/views/pages/auth/send-otp-page.blade.php) |
|  | [verify-otp-page](resources/views/pages/auth/verify-otp-page.blade.php) |
|  | [profile-page](resources/views/pages/dashboard/profile-page.blade.php) |
|  | [reset-pass-page](resources/views/pages/auth/reset-pass-page.blade.php) |
|  | [dashboard-page](resources/views/pages/dashboard/dashboard-page.blade.php) |
|  |  |  |
|  |  |  |
| Web | Route::get('/',[UserController::class,'LoginPage']); |  |
| Web | Route::get('/userRegistration',[UserController::class,'RegistrationPage']); |  |
| Web | Route::get('/sendOtp',[UserController::class,'SendOtpPage']); |  |
| Web | Route::get('/verifyOtp',[UserController::class,'VerifyOTPPage']); |  |
| Web | Route::get('/resetPassword',[UserController::class,'ResetPasswordPage'])->middleware([TokenVerificationMiddleware::class]); |  |
| Web | Route::get('/dashboard',[DashBoardController::class,'DashboardPage'])->middleware([TokenVerificationMiddleware::class]); |  |
| Web | Route::get('/userProfile',[UserController::class,'ProfilePage'])->middleware([TokenVerificationMiddleware::class]); |  |
|  |  |  |  |
|  |  | <script>  async function VerifyEmail() {  let email = document.getElementById('email').value;  if(email.length === 0){  errorToast('Please enter your email address')  }  else{  showLoader();  let res = await axios.post('/send-otp', {email: email});  hideLoader();  if(res.status===200 && res.data['status']==='success'){  successToast(res.data['message'])  sessionStorage.setItem('email', email);  setTimeout(function (){  window.location.href = '/verifyOtp';  }, 1000)  }  else{  errorToast(res.data['message'])  }  }  }  </script> | [send-otp-page.blade.php](resources/views/components/auth/send-otp-form.blade.php)  using javascript |
|  |  |  |  |
|  |  | <script>  async function VerifyOtp() {  let otp = document.getElementById('otp').value;  if(otp.length !==4){  errorToast('Invalid OTP')  }  else{  showLoader();  let res=await axios.post('/verify-otp', {  otp: otp,  email:sessionStorage.getItem('email')  })  hideLoader();  if(res.status===200 && res.data['status']==='success'){  successToast(res.data['message'])  sessionStorage.clear();  setTimeout(() => {  window.location.href='/resetPassword'  }, 1000);  }  else{  errorToast(res.data['message'])  }  }  }  </script> | [verify-otp-form.blade.php](resources/views/components/auth/verify-otp-form.blade.php)  using javascript |
|  |  |  |  |
|  |  | <script>  async function ResetPass() {  let password = document.getElementById('password').value;  let cpassword = document.getElementById('cpassword').value;  if(password.length===0){  errorToast('Password is required')  }  else if(cpassword.length===0){  errorToast('Confirm Password is required')  }  else if(password!==cpassword){  errorToast('Password and Confirm Password must be same')  }  else{  showLoader()  let res=await axios.post("/reset-password",{password:password});  hideLoader();  if(res.status===200 && res.data['status']==='success'){  successToast(res.data['message']);  setTimeout(function () {  window.location.href="/userLogin";  },1000);  }  else{  errorToast(res.data['message'])  }  }  }  </script> | [reset-pass-form.blade.php](resources/views/components/auth/reset-pass-form.blade.php)  using javascript |
|  |  |  |  |
|  |  | [TokenVerificationMiddleware](app/Http/Middleware/TokenVerificationMiddleware.php) & [TokenVerificationAPIMiddleware](app/Http/Middleware/TokenVerificationAPIMiddleware.php) modify |  |
| [Tutorial 19](file:///D:\project%20guidline%20sales%20inventory\1%20ইনভেন্টরি%20প্রোজেক্ট%20ব্যাক-ইন্ড%20ডেভেলপমেন্ট\Ostad%20-%2019%20%5bPOS%5d%20Auth%20Restriction%20In%20Page%20And%20End%20Point.ts) |  | public function handle(Request $request, Closure $next): Response{  $token=$request->header('token');  $result=JWTToken::VerifyToken($token);  if($result=="unauthorized"){  return redirect('/'); //userLogin;  // return response()->json(['message' => 'unauthorized'], 401); // checking postman  }  else{  $request->headers->set('email',$result->userEmail);  $request->headers->set('id',$result->userID);  return $next($request);  }  } |  |
|  |  |  |  |
|  | Data view | <script>  getProfile();  async function getProfile(){  showLoader();  let res=await axios.get("/user-profile")  hideLoader();  if(res.status===200 && res.data['status']==='success'){  let data=res.data['data'];  document.getElementById('email').value=data['email'];  document.getElementById('name').value=data['name'];  document.getElementById('mobile').value=data['mobile'];  document.getElementById('password').value=data['password'];  }  else{  errorToast(res.data['message'])  }  }  </script> | [profile-form.blade.php](resources/views/components/dashboard/profile-form.blade.php) |
|  |  |  |  |
|  | Data Update | async function onUpdate() {    let name = document.getElementById('name').value;  let mobile = document.getElementById('mobile').value;  let password = document.getElementById('password').value;  if(name.length===0){  errorToast('name')  }    else if(mobile.length===0){  errorToast('Mobile is required')  }  else if(password.length===0){  errorToast('Password is required')  }  else{  showLoader();  let res=await axios.post("/user-update",{  name:name,  mobile:mobile,  password:password  })  hideLoader();  if(res.status===200 && res.data['status']==='success'){  successToast(res.data['message']);  await getProfile();  }  else{  errorToast(res.data['message'])  }  }  } | profile-form.blade.php |
|  |  |  |  |
| **Category Page** | | | |
|  | view | <h2>category page</h2> | [category-page.blade.php](resources/views/pages/dashboard/category-page.blade.php) |
|  | Method | function CategoryPage(){  return view('pages.dashboard.category-page');  } | [CategoryController.php](app/Http/Controllers/CategoryController.php) |
|  | Route | Route::get('/categoryPage',[CategoryController::class,'CategoryPage'])->middleware([TokenVerificationMiddleware::class]); | web |
|  | Nav-bar | <a href="{{url("/categoryPage")}}" class="side-bar-item">  <i class="bi bi-list-nested"></i>  <span class="side-bar-item-caption">Category</span>  </a> | [sidenav-layout.blade.php](resources/views/layout/sidenav-layout.blade.php) |
|  |  |  |  |
|  | Modify | @extends('layout.sidenav-layout')  @section('content')  @include('[components.category.category-](resources/views/components/category/category-list.blade.php)list')    @endsection | [category-page.blade.php](file:///C:\xampp\htdocs\inventory-shop-11\resources\views\pages\dashboard\category-page.blade.php) |
|  |  |  |  |
|  |  | <div class="container-fluid">  <div class="row">  <div class="col-md-12 col-sm-12 col-lg-12">  <div class="card px-5 py-5">  <div class="row justify-content-between ">  <div class="align-items-center col">  <h4>Category</h4>  </div>  <div class="align-items-center col">  <button data-bs-toggle="modal" data-bs-target="#create-modal" class="float-end btn m-0 bg-gradient-primary">Create</button>  </div>  </div>  <hr class="bg-secondary"/>  <div class="table-responsive">  <table class="table" id="tableData">  <thead>  <tr class="bg-light">  <th>No</th>  <th>Category</th>  <th>Action</th>  </tr>  </thead>  <tbody id="tableList">  </tbody>  </table>  </div>  </div>  </div>  </div>  </div> | [category-list.blade.php](resources/views/components/category/category-list.blade.php) |
| [Tutorial 26](file:///D:\project%20guidline%20sales%20inventory\2%20ইনভেন্টরি%20প্রোজেক্ট%20ফ্রন্ট-ইন্ড\Ostad%20-%2025%20%5bPOS%5d%20Set%20Up%20JavaScript%20Data%20table%20For%20Category.ts) | Category List load | <script>  getList();  async function getList() {  showLoader();  let res=await axios.get("/list-category");  hideLoader();    let tableList=$("#tableList");  let tableData=$("#tableData");  tableData.DataTable().destroy();  tableList.empty();  res.data.forEach(function (item,index) {  let row=`<tr>  <td>${index+1}</td>  <td>${item['name']}</td>  <td>  <button data-id="${item['id']}" class="btn editBtn btn-sm btn-outline-success">Edit</button>  <button data-id="${item['id']}" class="btn deleteBtn btn-sm btn-outline-danger">Delete</button>  </td>  </tr>`  tableList.append(row)  })  new DataTable('#tableData',{  order:[[0,'desc']],  lengthMenu:[5,10,15,20,30]  });  }  </script> | [category-list.blade.php](file:///C:\xampp\htdocs\inventory-shop-11\resources\views\components\category\category-list.blade.php)  note: category no auto search |
|  |  |  |  |
| **Create category view** | | | |
|  | Modify | @extends('layout.sidenav-layout')  @section('content')  @include('components.category.category-list')  @include(['components.category.category-create'](resources/views/components/category/category-create.blade.php))    @endsection | [category-page.blade.php](file:///C:\xampp\htdocs\inventory-shop-11\resources\views\pages\dashboard\category-page.blade.php) |
|  |  |  |  |
| [Tutorial 27](file:///D:\project%20guidline%20sales%20inventory\2%20ইনভেন্টরি%20প্রোজেক্ট%20ফ্রন্ট-ইন্ড\Ostad%20-%2027%20%5bPOS%5d%20Working%20With%20Category%20Create%20And%20Refresh%20List.ts) |  | <script>  async function Save() {  let categoryName = document.getElementById('categoryName').value;  if (categoryName.length === 0) {  errorToast("Category Required !")  }  else {  document.getElementById('modal-close').click();  showLoader();  let res = await axios.post("/create-category",{name:categoryName})  hideLoader();  if(res.status===201){  successToast('Request completed');  document.getElementById("save-form").reset();  await getList();  }  else{  errorToast("Request fail !")  }  }  }  </script> |  |
|  |  |  |  |
| **category delete** | | | |
| [Tutorial 29](file:///D:\project%20guidline%20sales%20inventory\2%20ইনভেন্টরি%20প্রোজেক্ট%20ফ্রন্ট-ইন্ড\Ostad%20-%2029%20%5bPOS%5d%20Working%20With%20Delete%20In%20Smart%20Way.ts) | Modify | @extends('layout.sidenav-layout')  @section('content')  @include('components.category.category-list')  @include(['components.category.category-create'](resources/views/components/category/category-create.blade.php))  @include(['components.category.category-delete'](resources/views/components/category/category-delete.blade.php))  @endsection | [category-page.blade.php](file:///C:\xampp\htdocs\inventory-shop-11\resources\views\pages\dashboard\category-page.blade.php) |
|  |  |  |  |
| **Category Edit View** | | | |
|  | Modify | @extends('layout.sidenav-layout')  @section('content')  @include('components.category.category-list')  @include(['components.category.category-create'](resources/views/components/category/category-create.blade.php))  @include(['components.category.category-delete'](resources/views/components/category/category-delete.blade.php))  @include('[components.category.category-update'](resources/views/components/category/category-update.blade.php))  @endsection | [category-page.blade.php](file:///C:\xampp\htdocs\inventory-shop-11\resources\views\pages\dashboard\category-page.blade.php) |
|  | Add this script option | $('.editBtn').on('click', async function () {  let id= $(this).data('id');  $("#update-modal").modal('show');  }) | components.category.category-list |
| **Cateagory data show** | | | |
|  | Add this line | <script>  async function FillUpUpdateForm(id){  document.getElementById('updateID').value=id;  showLoader();  let res=await axios.post("/category-by-id",{id:id})  hideLoader();  document.getElementById('categoryNameUpdate').value=res.data['name'];  }  </script> | [components.category.category-update'](file:///C:\xampp\htdocs\inventory-shop-11\resources\views\components\category\category-update.blade.php) |
|  | Add this script option | $('.editBtn').on('click', async function () {  let id= $(this).data('id');  await FillUpUpdateForm(id);  $("#update-modal").modal('show');  }) | components.category.category-list |
|  | Add this script option | async function Update() {          let categoryName = document.getElementById('categoryNameUpdate').value;          let updateID = document.getElementById('updateID').value;          if (categoryName.length === 0) {              errorToast("Category Required !")          }          else{              document.getElementById('update-modal-close').click();              showLoader();              let res = await axios.post("/update-category",{name:categoryName,id:updateID})              hideLoader();              if(res.status===200 && res.data===1){                  document.getElementById("update-form").reset();                  successToast("Request success !")                  await getList();              }              else{                  errorToast("Request fail !")              }          }        } | components.category.category-update |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |