|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 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]); |  |