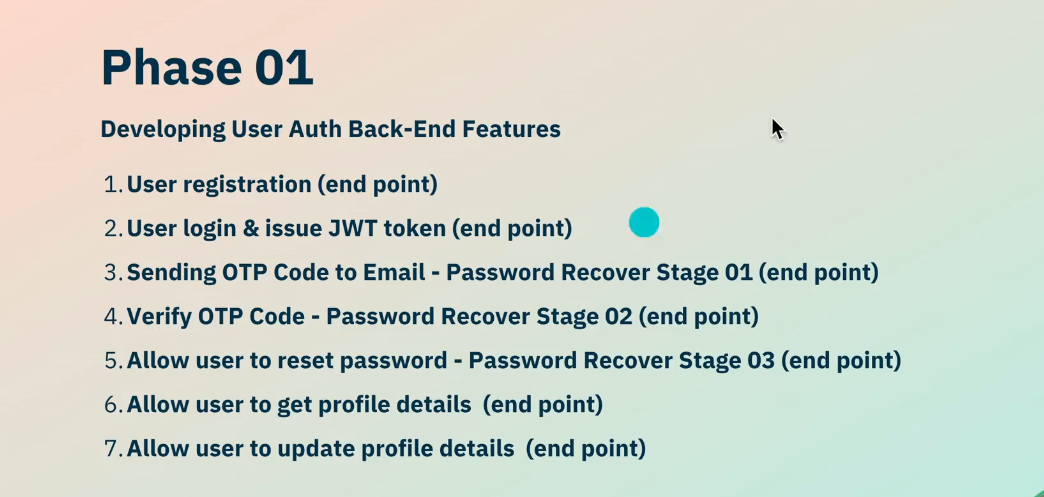
Prerecorded Video : Module 20

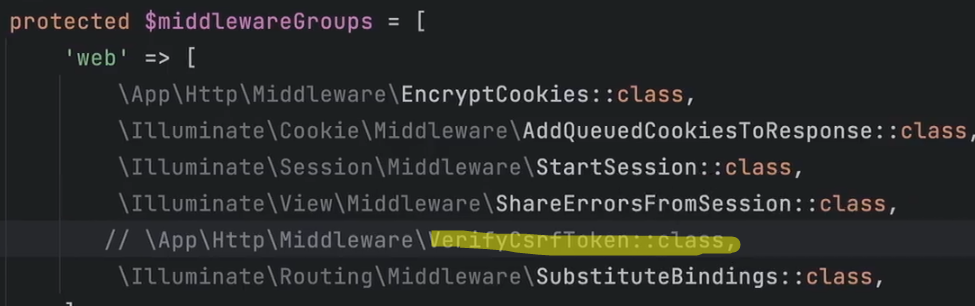
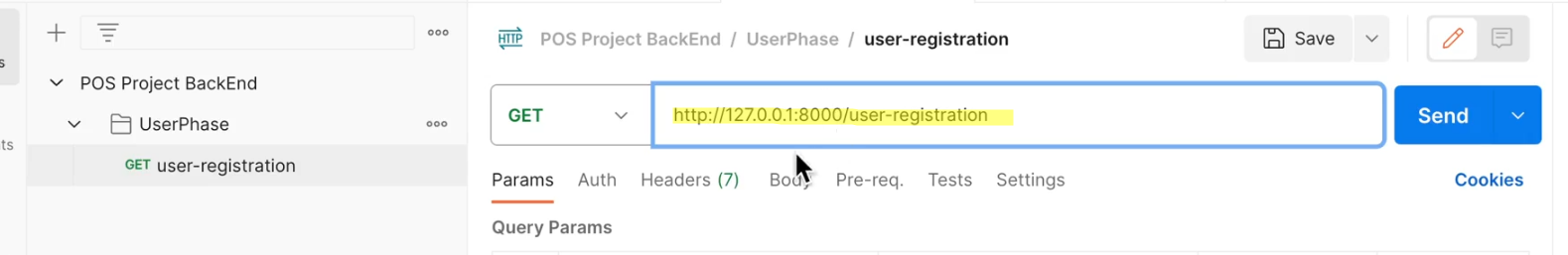
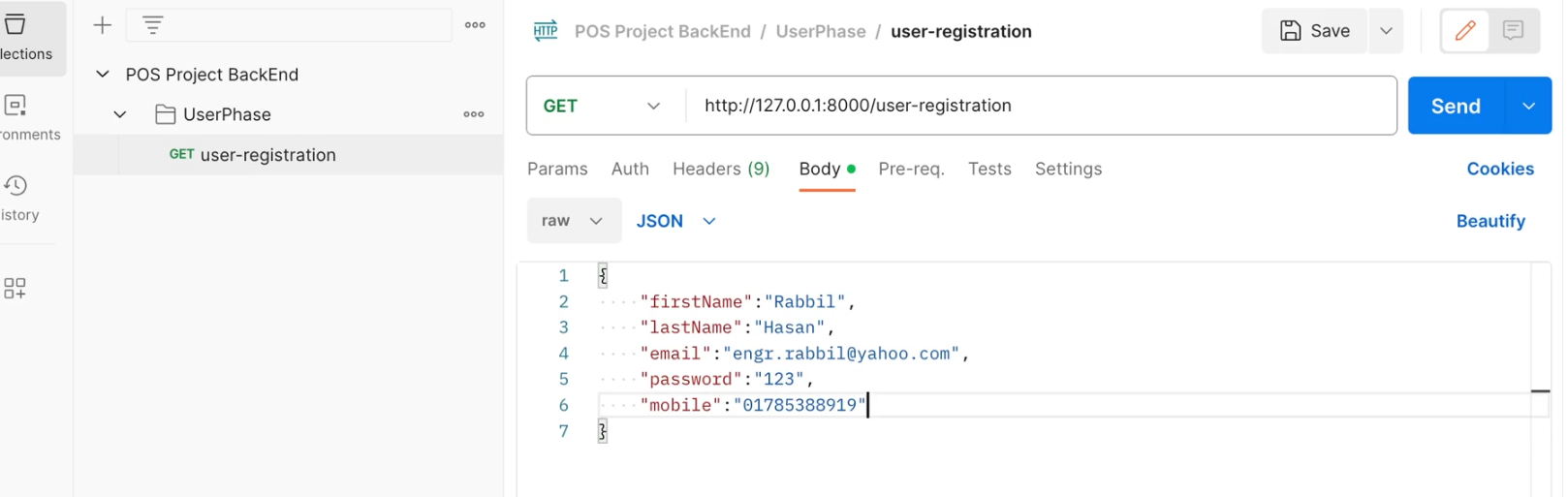
Smart Shop



User Module : 2[POS] Registration

* Create a Project smart-shop
* - composer create-project laravel/laravel smart-shop
* Edit .env File
* - change database name
* - change DB\_PORT if necessary
* Edit user Table
* - php artisan migrate
* Create UserController
* - php artisan make:controller UserController
* - write code for User Create
* Create UserModel
* - php artisan make:model User

Create User Registration API

* Route->web.php
* Route::post('/user-registration',[UserController::class,'UserRegistration']);
* Disable ->Http->kernel.php->CSRF Security (App\Http\Middleware\VerifyCsrfToken::class
* ) for testing purpose
* 
* Test API Endpoint with Postman
* Goto Postman -> Create a Collection POS Project Backend->then create a Folder User Module
* Create a New Request : user-registration
* 
* Pass json data through Body
* 
* php artisan serve – to run the server for postman testing
* Test with postman

3[POS] Registration :

4 [POS] JWT For Authentication :

* Install JWT Package
* - composer require firebase/php-jwt
* create a folder app->helper
* create a file app->Helper->JWTToken.php
* create function CreateToken(){}
* create function VerifyToken(){}
* edit .env file & write JWT\_KEY = 123456 in the end of the file

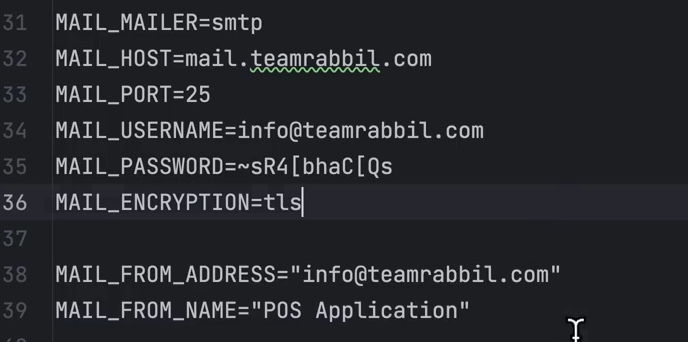
5 [POS] User Login And Issue JWT Token :

* UserController.php
* create a function UserLogin
* edit web.php & create a Route for User Login
* test with Postman
* to transfer a project from one pc to another
* - following problem may happen in .env file
* - php artisan key:generate – if no key found in .env file
* - write JWT\_KEY = 664644 in .env file
* - install composer to update vendor file

6 [POS] Working With OTP Email :

* to send an OTP we need to create a email class
* - php artisan make:mail OTPMail(mail name)
* - a mail class has been created in folder app-> mail
* create a folder naming email in resource->views folder
* - create a file name OTPmail.blade.php

7 [POS] Working With OTP Email :

* search OTP email template in google
* copy from google & paste it to OTPmail.blade.php file
* configure SMTP mail in .env file
* 
* goto UserController -> create a function for OTP send mail

8 [POS] Working With OTP Email :

* create API Endpoint for OTP Send send-otp
* - test with postman

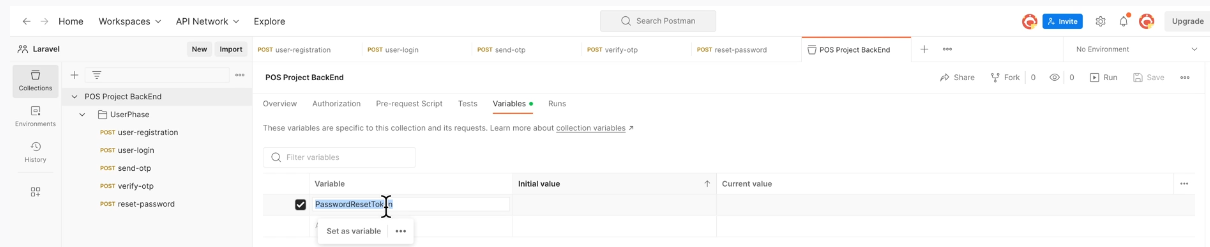
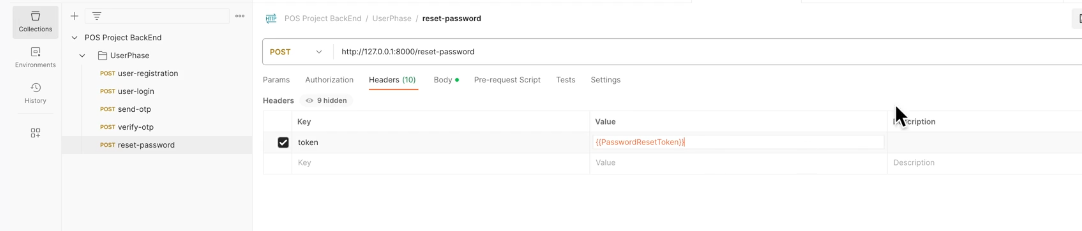
9 [POS] Verify OTP And Issue JWT Password Reset Token :

* in UserController create a function for VerifyOTP
* in JWTToken.php create a function CreateTokenForSetPassword for reset password token
* create API Endpoint for OTP Verification Route::post('/verify-otp',[UserController::class,'VerifyOTP']);
* Back End Test with Postman : verify-otp

10 [POST] Working With Token Verification Middleware And Password Reset :

* create a middleware for Token Verification
* - php artisan make:middleware TokenVerificationMiddleware : it creates a file in app-> http ->middleware->TokenVerificationMiddleware.php
* We will receive the token through header instead of body
* open TokenVerificationMiddleware.php
* goto userController.php : create function ResetPassword for password reset
* add Route API Endpoint for reset password

11 [POS] Completing Password Reset :

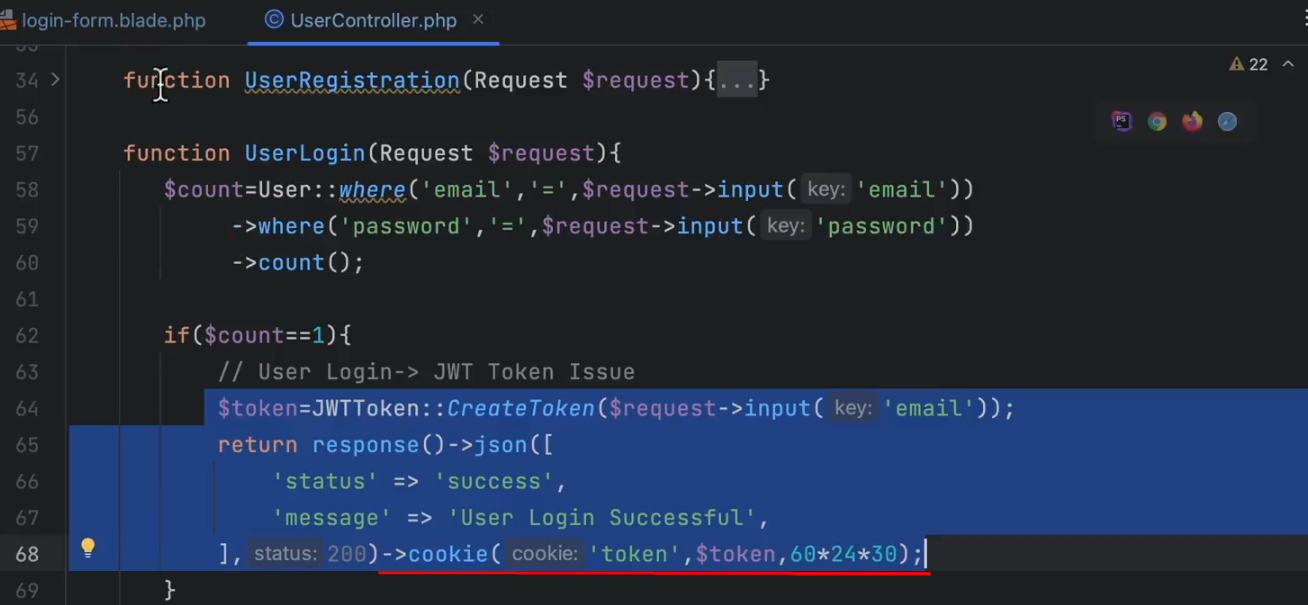
* Back End Test with Postman : reset-password
* goto postman documentation variable section
* set a variable PasswordResetToken
* 
* 

12 [POS] Front End Explain Front End Assets :

* Front End Development
* create page for UserLogin, UserRegistration, SendOTP, VerifyOTP, ResetPassword, Dashboard etc
* Add page routes in routes->web.php
* create resources->views->layout->app.blade.php and sidenav.blade.php
* HTML, CSS, Javascript strategy etc.

13 [POS] Front End User Login :

* login-form.blade.php
* create function for SubmitLogin
* goto User Controller -> User Login
* store token in browser : cache, cookies, index db, local storage & session storage, but recommended is COOKIE.
* to set Token in the Cookie we should open userController.php & do the following :



14 [POS] Front End User Registration :