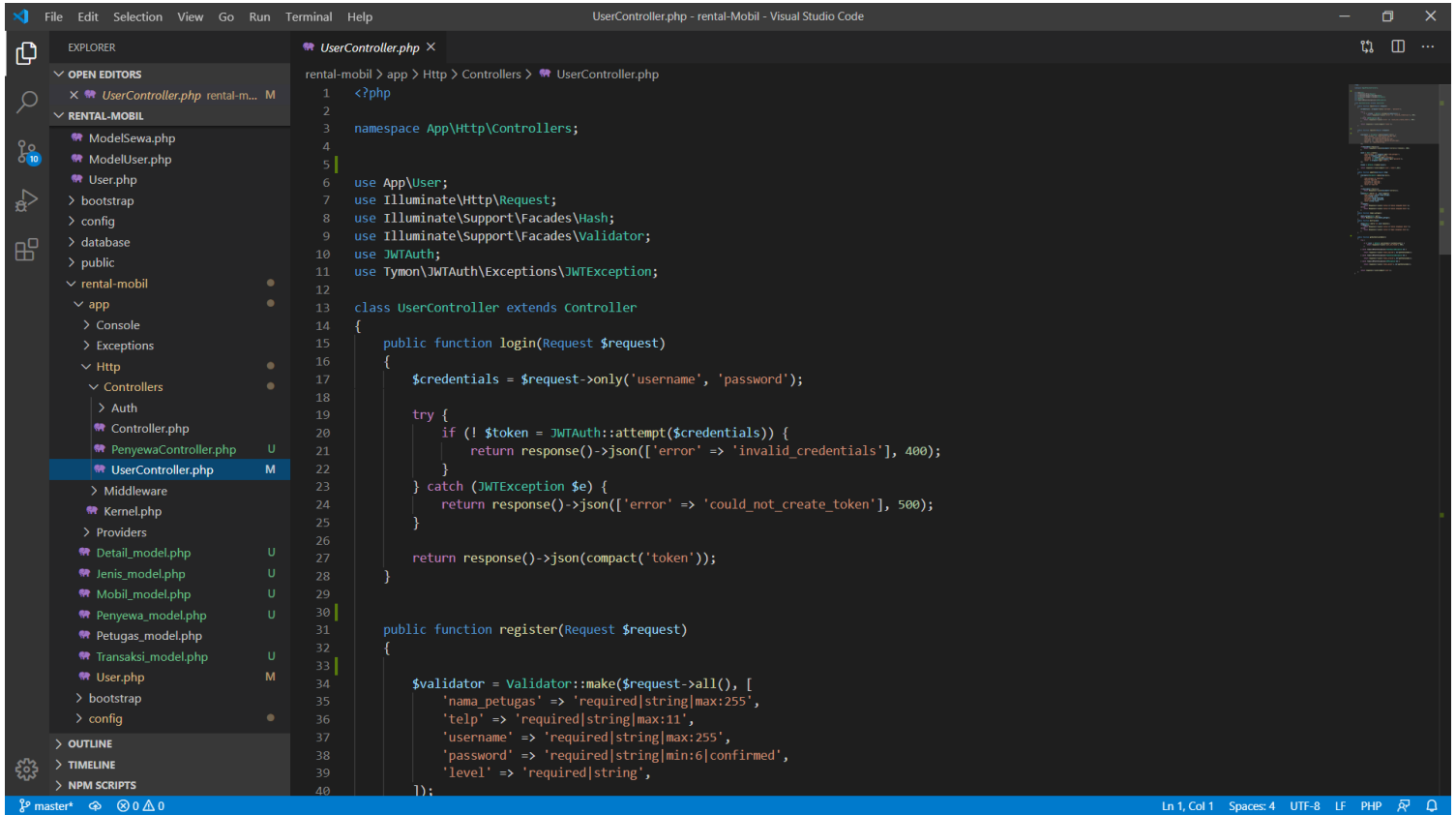
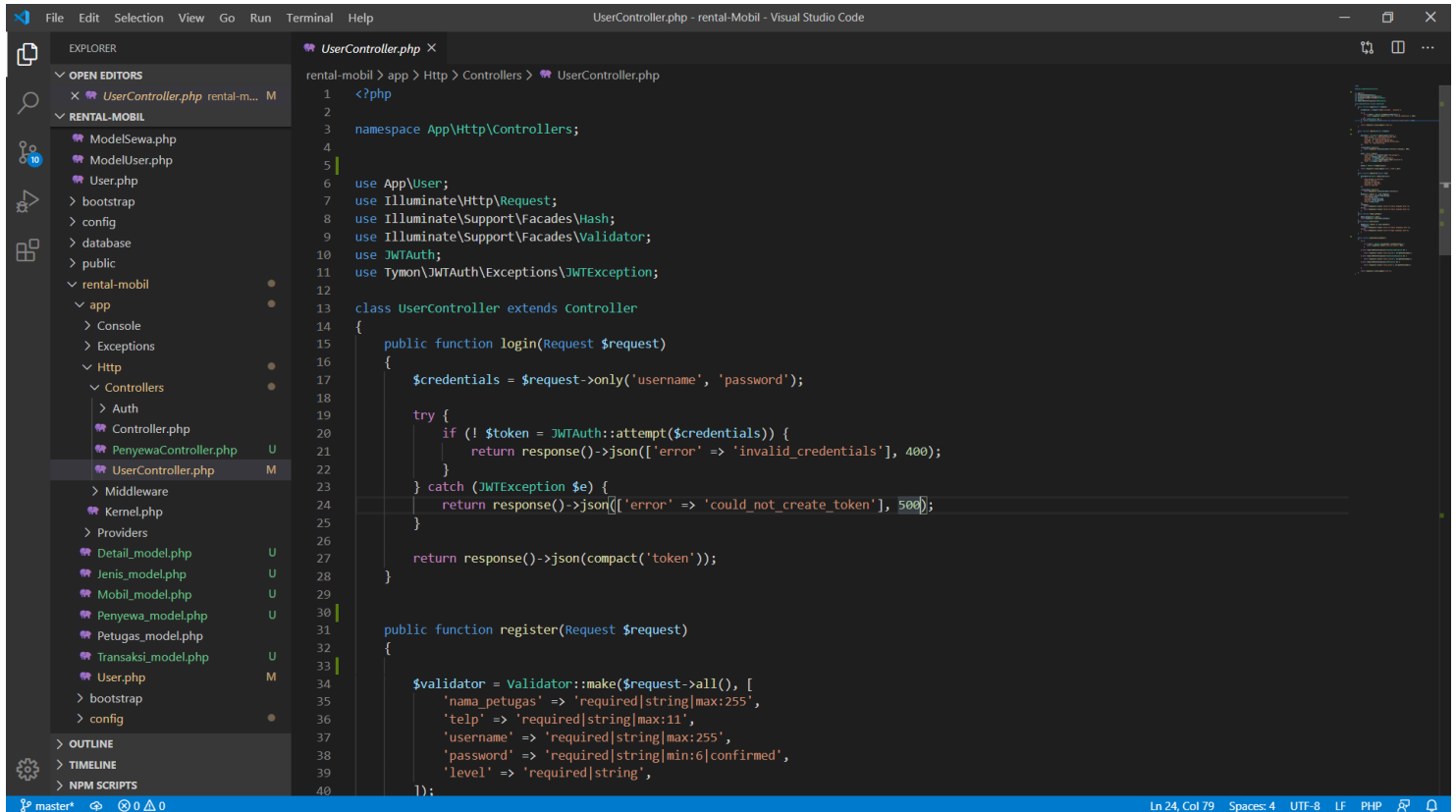


Script UserController.php



```
UserController.php X
rental-mobil > app > Http > Controllers > UserController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5
6  use App\User;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Hash;
9  use Illuminate\Support\Facades\Validator;
10 use JWTAuth;
11 use Tymon\JWTAuth\Exceptions\JWTException;
12
13 class UserController extends Controller
14 {
15     public function login(Request $request)
16     {
17         $credentials = $request->only('username', 'password');
18
19         try {
20             if (! $token = JWTAuth::attempt($credentials)) {
21                 return response()->json(['error' => 'invalid_credentials'], 400);
22             }
23         } catch (JWTException $e) {
24             return response()->json(['error' => 'could_not_create_token'], 500);
25         }
26
27         return response()->json(compact('token'));
28     }
29
30     public function register(Request $request)
31     {
32
33         $validator = Validator::make($request->all(), [
34             'nama_petugas' => 'required|string|max:255',
35             'telp' => 'required|string|max:11',
36             'username' => 'required|string|max:255',
37             'password' => 'required|string|min:6|confirmed',
38             'level' => 'required|string',
39         ]);
40     }
41 }
```



```
UserController.php X
rental-mobil > app > Http > Controllers > UserController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5
6  use App\User;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Hash;
9  use Illuminate\Support\Facades\Validator;
10 use JWTAuth;
11 use Tymon\JWTAuth\Exceptions\JWTException;
12
13 class UserController extends Controller
14 {
15     public function login(Request $request)
16     {
17         $credentials = $request->only('username', 'password');
18
19         try {
20             if (! $token = JWTAuth::attempt($credentials)) {
21                 return response()->json(['error' => 'invalid_credentials'], 400);
22             }
23         } catch (JWTException $e) {
24             return response()->json(['error' => 'could_not_create_token'], 500);
25         }
26
27         return response()->json(compact('token'));
28     }
29
30     public function register(Request $request)
31     {
32
33         $validator = Validator::make($request->all(), [
34             'nama_petugas' => 'required|string|max:255',
35             'telp' => 'required|string|max:11',
36             'username' => 'required|string|max:255',
37             'password' => 'required|string|min:6|confirmed',
38             'level' => 'required|string',
39         ]);
40     }
41 }
```

```
UserController.php X
rental-mobil > app > Http > Controllers > UserController.php
37 'username' => 'required|string|max:255',
38 'password' => 'required|string|min:6|confirmed',
39 'level' => 'required|string',
40 ];
41
42 if($validator->fails()){
43     return response()->json($validator->errors()->toJson(), 400);
44 }
45
46 $user = User::create([
47     'nama_petugas' => $request->get('nama_petugas'),
48     'telp' => $request->get('telp'),
49     'username' => $request->get('username'),
50     'password' => Hash::make($request->get('password')),
51     'level' => $request->get('level'),
52 ]);
53
54 $token = JWTAuth::fromUser($user);
55
56 return response()->json(compact('user', 'token'), 201);
57 }
58
59 public function update($id, Request $req)
60 {
61     $validator=Validator::make($req->all(),
62     [
63         'nama_petugas'=>'required',
64         'telp'=>'required',
65         'username'=>'required',
66         'password'=>'required',
67         'level'=>'required'
68     ]);
69
70     if($validator->fails()){
71         return Response()->json($validator->errors());
72     }
73
74     $ubah=User::where('id', $id)->update([
75         'nama_petugas'=>$req->nama_petugas,
76         'telp'=>$req->telp,
77         'username'=>$req->username,
```

```
UserController.php X
rental-mobil > app > Http > Controllers > UserController.php
72 }
73 $ubah=User::where('id', $id)->update([
74     'nama_petugas'=>$req->nama_petugas,
75     'telp'=>$req->telp,
76     'username'=>$req->username,
77     'password'=>$req->password,
78     'level'=>$req->level
79 ]);
80
81 if($ubah){
82     return Response()->json(['status'=>'sukses mengubah data!']);
83 }else{
84     return Response()->json(['status'=>'sukses mengubah data!']);
85 }
86
87 public function tampil_petugas()
88 {
89     $data_petugas=User::get();
90     return Response()->json($data_petugas);
91 }
92
93 public function destroy($id)
94 {
95     $hapus=User::where('id',$id)->delete();
96     if($hapus){
97         return Response()->json(['status'=>'sukses menghapus data!']);
98     }else{
99         return Response()->json(['status'=>'gagal menghapus data!']);
100     }
101 }
102
103 public function getAuthenticatedUser()
104 {
105     try {
106         if (! $user = JWTAuth::parseToken()->authenticate()) {
107             return response()->json(['user_not_found'], 404);
108         }
109     } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {
110     }
```

```
UserController.php X
rental-mobil > app > Http > Controllers > UserController.php
92 {
93     $shapus=User::where('id',$id)->delete();
94     if($shapus){
95         return Response()->json(['status'=>'Sukses menghapus data!']);
96     }else{
97         return Response()->json(['status'=>'Gagal menghapus data!']);
98     }
99 }
100
101 public function getAuthenticatedUser()
102 {
103     try {
104         if (! $user = JWTAuth::parseToken()->authenticate()) {
105             return response()->json(['user_not_found'], 404);
106         }
107     } catch (Tymon\JWTAuth\Exceptions\TokenExpiredException $e) {
108         return response()->json(['token_expired'], $e->getStatusCode());
109     } catch (Tymon\JWTAuth\Exceptions\TokenInvalidException $e) {
110         return response()->json(['token_invalid'], $e->getStatusCode());
111     } catch (Tymon\JWTAuth\Exceptions\JWTException $e) {
112         return response()->json(['token_absent'], $e->getStatusCode());
113     }
114     return response()->json(compact('user'));
115 }
116
117
118
119
120
121
122
123
124
125
126
127
```

Script Api.php

```
api.php X
rental-mobil > routes > api.php
1 <?php
2
3 use Illuminate\Http\Request;
4
5 /*
6 |-----
7 | API Routes
8 |-----
9 |
10 | Here is where you can register API routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | is assigned the "api" middleware group. Enjoy building your API!
13 |
14 */
15
16 Route::middleware('auth:api')->get('/user', function (Request $request) {
17     return $request->user();
18 });
19 // Petugas
20 Route::post('register', 'UserController@register');
21 Route::post('login', 'UserController@login');
22 Route::put('/ubah_petugas/{id}', 'UserController@update')->middleware('jwt.verify');
23 Route::delete('/hapus_petugas/{id}', 'UserController@destroy')->middleware('jwt.verify');
24 Route::get('/tampil_petugas', 'UserController@tampil_petugas')->middleware('jwt.verify');
25
26 // Penyewa
27 Route::post('/simpan_penyewa', 'PenyewaController@store')->middleware('jwt.verify');
28 Route::put('/ubah_penyewa/{id}', 'PenyewaController@update')->middleware('jwt.verify');
29 Route::delete('/hapus_penyewa/{id}', 'PenyewaController@destroy')->middleware('jwt.verify');
30 Route::get('/tampil_penyewa', 'PenyewaController@tampil_penyewa')->middleware('jwt.verify');
```

Script User.php

File Edit Selection View Go Run Help

User.php - rental-Mobil - Visual Studio Code

EXPLORER

- api.php
- User.php

OPEN EDITORS

- api.php rental-mobil/routes M
- User.php rental-mobil/app M

RENTAL-MOBIL

- database
 - public
- rental-mobil
 - app
 - Console
 - Exceptions
 - Http
 - Controllers
 - Auth
 - Controller.php
 - PenyewaController.php U
 - UserController.php M
 - Middleware
 - Kernel.php
 - Providers
 - Detail_model.php U
 - Jenis_model.php U
 - Mobil_model.php U
 - Penyewa_model.php U
 - Petugas_model.php U
 - Transaksi_model.php U
 - User.php M
 - bootstrap
 - config
 - database
 - public
 - resources
 - routes
 - api.php M
 - OUTLINE
 - TIMELINE
 - NPM SCRIPTS

rental-mobil > app > User.php

```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Notifications\Notifiable;
6 use Illuminate\Contracts\Auth\MustVerifyEmail;
7 use Illuminate\Foundation\Auth\User as Authenticatable;
8 use Tymon\JWTAuth\Contracts\JWTSubject;
9
10 class User extends Authenticatable implements JWTSubject
11 {
12     protected $table="petugas";
13     protected $primaryKey="id_petugas";
14     public $timestamps=false;
15     use Notifiable;
16
17     /**
18      * The attributes that are mass assignable.
19      *
20      * @var array
21      */
22     protected $fillable = [
23         'nama_petugas', 'telp', 'username', 'password', 'level'
24     ];
25
26     /**
27      * The attributes that should be hidden for arrays.
28      *
29      * @var array
30      */
31     protected $hidden = [
32         'password', 'remember_token',
33     ];
34
35     public function getJWTIdentifier()
36     {
37         return $this->getKey();
38     }
39     public function getJWTCustomClaims()
40     {
```

Hasil register dan login

