Customer

	Customer	php artisan make:model Customer –m	cmd

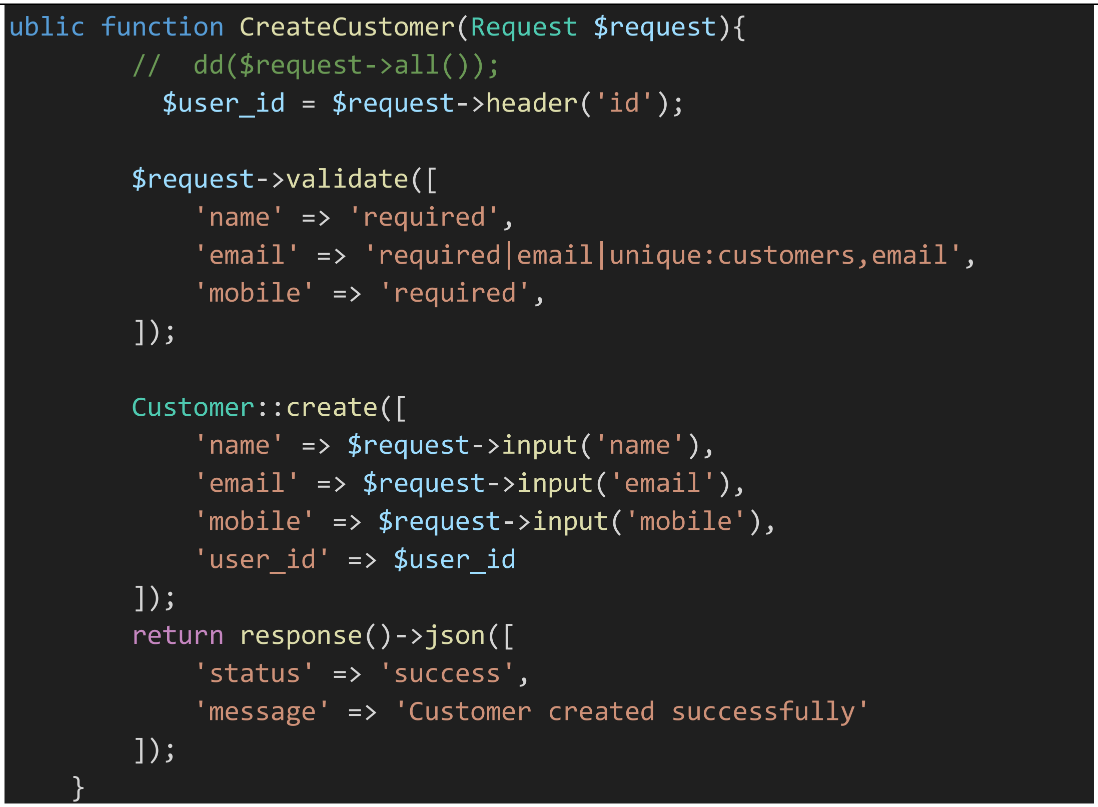
		`protected $fillable = [`
		`'name',`
		`'email',`
		`'mobile',`
		`'user_id'`
		`];`
			Model/Customer.php

	Migrations file	`public function up(): void{`
		`Schema::create('customers', function (Blueprint $table) {`
		`$table->id();`
		`$table->string('name');`
		`$table->string('email');`
		`$table->string('mobile');`
		`$table->unsignedBigInteger('user_id');`
		`$table->foreign('user_id')->references('id')->on('users')`
		`->cascadeOnUpdate()->restrictOnDelete();`
		`$table->timestamp('created_at')->useCurrent();`
		`$table->timestamp('updated_at')->useCurrent()->useCurrentOnUpdate();`
		`});`
		`}`
			2025_07_05_194415_create_customers_table.php

Add this method	Customers method	`public function customers(){`
		`return $this->hasMany(Customer::class);`
		`}`
			Model/users.php

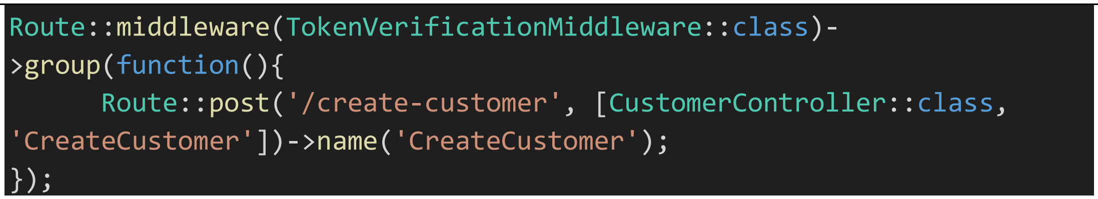
	Specifiq migration	php artisan migrate --path=database/migrations/2025_07_05_194415_create_customers_table.php	cmd
	Customer controller	php artisan make:controller CustomerController	cmd

<table>
<tr>
<td></td>
<td>CreateCustomer method</td>
<td>



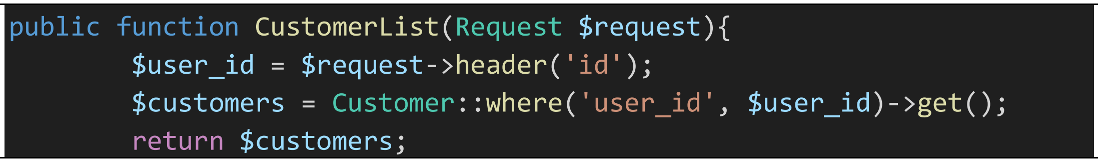
```
ublic function CreateCustomer(Request $request){
        //  dd($request->all());
          $user_id = $request->header('id');

        $request->validate([
            'name' => 'required',
            'email' => 'required|email|unique:customers,email',
            'mobile' => 'required',
        ]);

        Customer::create([
            'name' => $request->input('name'),
            'email' => $request->input('email'),
            'mobile' => $request->input('mobile'),
            'user_id' => $user_id
        ]);
        return response()->json([
            'status' => 'success',
            'message' => 'Customer created successfully'
        ]);
    }
```

</td>
<td>CustomerController.php</td>
</tr>
<tr>
<td></td>
<td>web</td>
<td>



```
Route::middleware(TokenVerificationMiddleware::class)-
>group(function(){
      Route::post('/create-customer', [CustomerController::class,
'CreateCustomer'])->name('CreateCustomer');
});
```

</td>
<td>checkpostman</td>
</tr>
<tr>
<td></td>
<td>Customer list</td>
<td>



```
public function CustomerList(Request $request){
        $user_id = $request->header('id');
        $customers = Customer::where('user_id', $user_id)->get();
        return $customers;
```

</td>
<td></td>
</tr>
</table>

		}	

	web	Route::middleware(TokenVerificationMiddleware::class)->group(function(){
		      Route::get('/list-customer', [CustomerController::class, 'CustomerList'])->name('CustomerList');
		});
			checkpostman

	customerByID method	public function CustomerById(Request $request){
		        $user_id = $request->header('id');
		        $customer =Customer::where('id', $request->id)->where('user_id', $user_id)->first();
		        return $customer;
		    }	

	web	Route::middleware(TokenVerificationMiddleware::class)->group(function(){
		      Route::post('/customer-by-id', [CustomerController::class, 'CustomerById'])->name('CustomerById');
		});
			checkpostman

	CustomerUpdate mothod	public function CustomerUpdate(Request $request){
		        $user_id = $request->header('id');
		        $id = $request->input('id');
		        Customer::where('id', $id)->where('user_id', $user_id)->update([
		            'name' => $request->input('name'),
		            'email' => $request->input('email'),
		            'mobile' => $request->input('mobile'),
		        ]);
		        return response()->json([
		            'status' => 'success',	

<table>
<tr><td></td><td></td><td colspan="2"><pre>
            'message' => 'Customer Updaetd successfully'
        ]);

    }
</pre></td><td></td></tr>
<tr><td></td><td>web</td><td colspan="2"><pre>
Route::middleware(TokenVerificationMiddleware::class)->group(function(){
      Route::post('/update-customer', [CustomerController::class, 'CustomerUpdate'])->name('CustomerUpdate');
});
</pre></td><td>checkpostman</td></tr>
<tr><td></td><td>customerDelete mothod</td><td colspan="2"><pre>
public function CustomerDelete(Request $request,$id){
        $user_id = $request->header('id');
        Customer::where('user_id', $user_id)->where('id', $id)->delete();
        return response()->json([
            'status' => 'success',
            'message' => 'Customer Deleted successfully'
        ]);
</pre></td><td></td></tr>
<tr><td></td><td>web</td><td colspan="2"><pre>
Route::middleware(TokenVerificationMiddleware::class)->group(function(){
     Route::get('/delete-customer/{id}', [CustomerController::class, 'CustomerDelete'])->name('CustomerDelete');
});
</pre></td><td>checkpostman</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table>
