	Invoice	php artisan make:model Invoice -m	cmd

	migrations	public function up(): void{
		        Schema::create('invoices', function (Blueprint $table) {
		            $table->id();
		            $table->unsignedBigInteger('user_id');
		            $table->unsignedBigInteger('customer_id');
		
		            $table->foreign('user_id')->references('id')->on('users')
		            ->cascadeOnUpdate()->restrictOnDelete();
		            $table->foreign('customer_id')->references('id')->on('customers')
		            ->cascadeOnUpdate()->restrictOnDelete();
		
		            $table->string('total',100);
		            $table->string('discount',50)->nullable();
		            $table->string('vat')->nullable();
		            $table->string('Payable')->nullable();
		            $table->timestamp('created_at')->useCurrent();
		            $table->timestamp('updated_at')->useCurrent()->useCurrentOnUpdate();
		        });
		    }
			2025_07_06_054918_create_invoices_table.php

	Invoice model	protected $fillable = [
		        'user_id',
		        'customer_id',
		        'total',
		        'discount',
		        'vat',
			Model/invoice.php

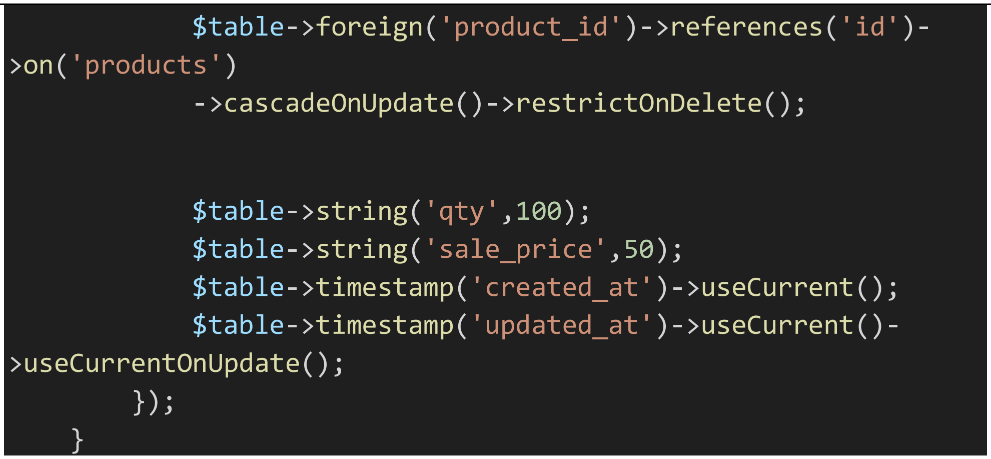
		`        'Payable',`
		`    ];`
		
		`    public function user(){`
		`        return $this->belongsTo(User::class);`
		`    }`
		`    public function customer(){`
		`        return $this->belongsTo(Customer::class);`
		`    }`	

	Add this method	`public function invoices(){`
		`        return $this->hasMany(Invoice::class);`
		`    }`
			পাসের মেথড টি (invoices) User & Customer Model যুক্ত করতে হবে

		php artisan migrate --path=database/migrations/2025_07_06_054918_create_invoices_table.php	
	Invoice	php artisan make:model Invoice_product -m	cmd

		`public function up(): void{`
		`        Schema::create('invoice_products', function (Blueprint $table) {`
		`           $table->unsignedBigInteger('user_id');`
		`            $table->unsignedBigInteger('invoice_id');`
		`            $table->unsignedBigInteger('product_id');`
		
		`            $table->foreign('user_id')->references('id')->on('users')`
		`            ->cascadeOnUpdate()->restrictOnDelete();`
		`            $table->foreign('invoice_id')->references('id')->on('invoices')`
		`            ->cascadeOnUpdate()->restrictOnDelete();`	

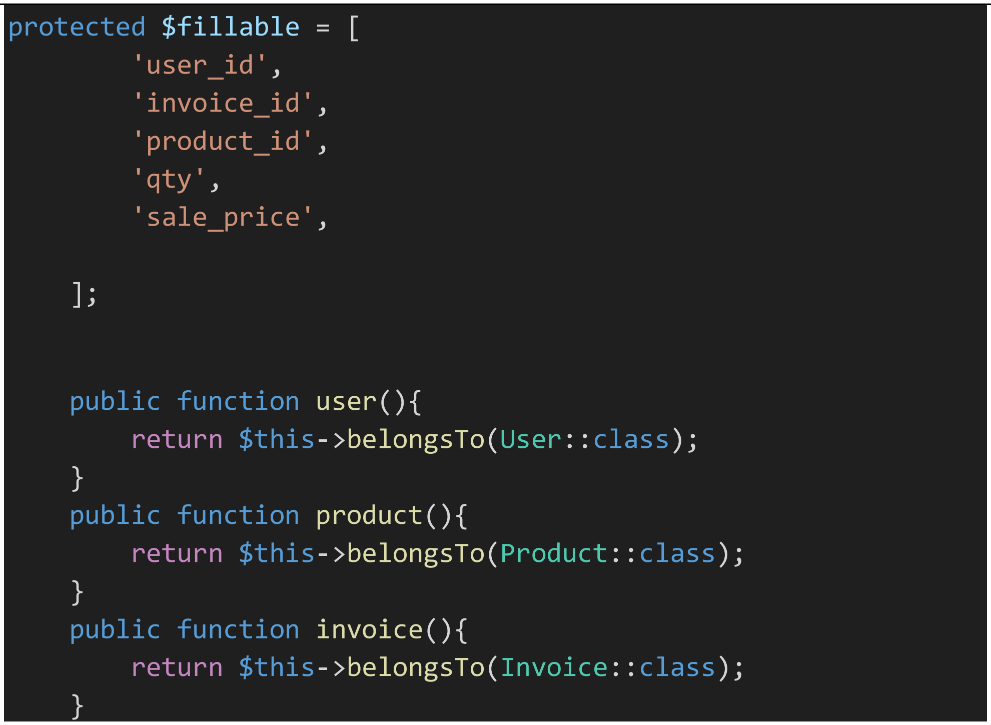
<table>
<tr>
<td></td>
<td></td>
<td>



```
            $table->foreign('product_id')->references('id')-
>on('products')
            ->cascadeOnUpdate()->restrictOnDelete();


            $table->string('qty',100);
            $table->string('sale_price',50);
            $table->timestamp('created_at')->useCurrent();
            $table->timestamp('updated_at')->useCurrent()-
>useCurrentOnUpdate();
        });
    }
```

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



```
protected $fillable = [
        'user_id',
        'invoice_id',
        'product_id',
        'qty',
        'sale_price',

    ];


    public function user(){
        return $this->belongsTo(User::class);
    }
    public function product(){
        return $this->belongsTo(Product::class);
    }
    public function invoice(){
        return $this->belongsTo(Invoice::class);
    }
```

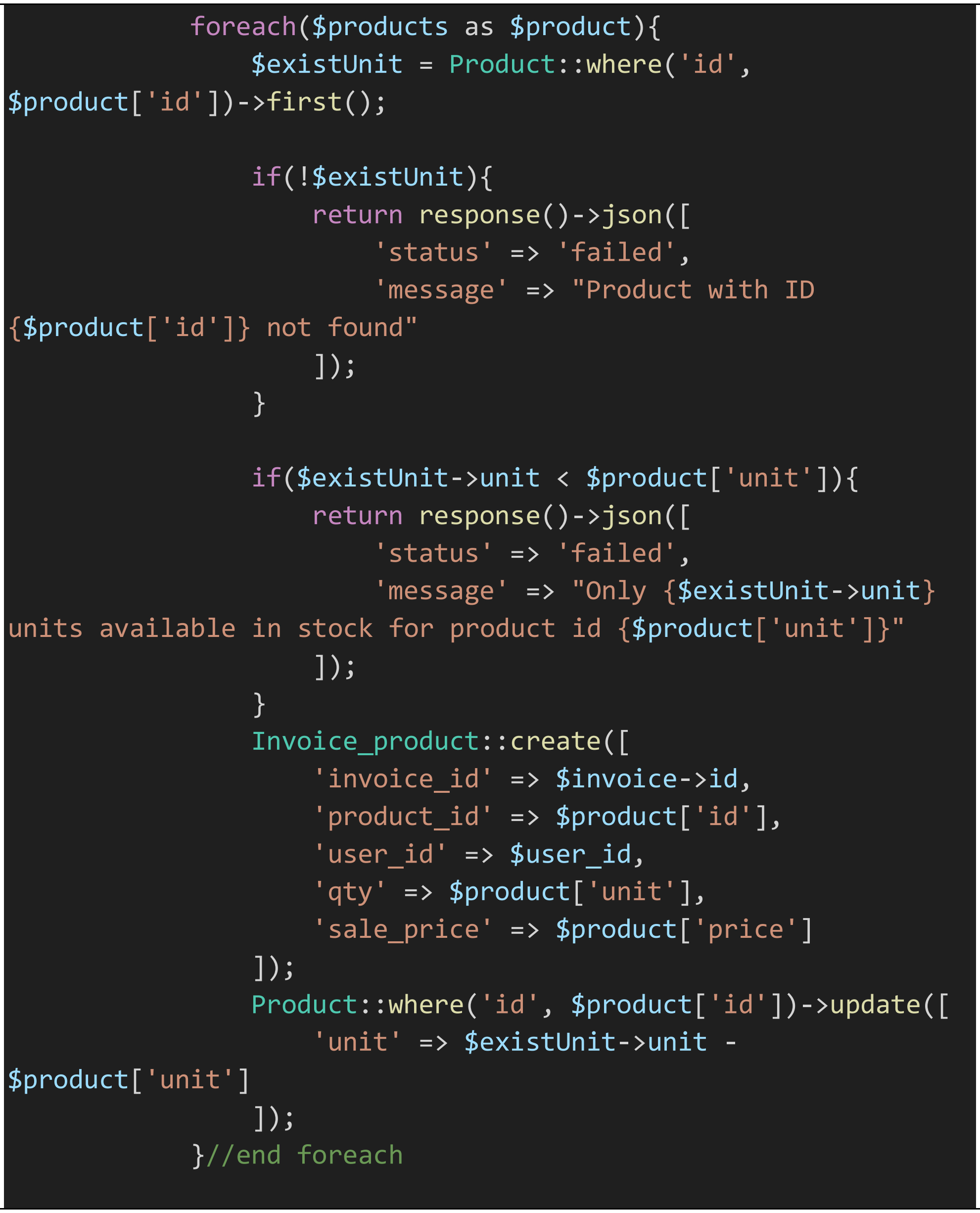
</td>
<td>Invoice_product model</td>
</tr>
</table>

<table>
<tr><td></td><td>Add this method</td><td><pre><code>public function Invoice_product(){
        return $this->hasMany(Invoice_product::class);
    }</code></pre></td><td>পাসের মেথড টি (Invoice_product) User & Invoice,product Model যুক্ত করতে হবে</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>php artisan migrate --path=database/migrations/2025_07_06_061745_create_invoice_products_table.php</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>Php artisan make:controller InvoiceController</td><td></td></tr>
<tr><td></td><td>InvoceCreate method</td><td><pre><code>public function InvoiceCreate(Request $request){
        // dd($request->all());
        DB::beginTransaction();
        try {
            $user_id = $request->header('id');

            $data = [
                    'user_id' => $user_id,
                    'customer_id' => $request->customer_id,
                    'total' => $request->total,
                    'vat' => $request->vat,
                    'Payable' => $request->payable,
                    'discount' => $request->discount
            ];

            $invoice = Invoice::create($data);

            $products = $request->input('products');</code></pre></td><td>Model/invoiceCreate.php</td></tr>
</table>



```
            foreach($products as $product){
                $existUnit = Product::where('id',
$product['id'])->first();

                if(!$existUnit){
                    return response()->json([
                        'status' => 'failed',
                        'message' => "Product with ID
{$product['id']} not found"
                    ]);
                }

                if($existUnit->unit < $product['unit']){
                    return response()->json([
                        'status' => 'failed',
                        'message' => "Only {$existUnit->unit}
units available in stock for product id {$product['unit']}"
                    ]);
                }
                Invoice_product::create([
                    'invoice_id' => $invoice->id,
                    'product_id' => $product['id'],
                    'user_id' => $user_id,
                    'qty' => $product['unit'],
                    'sale_price' => $product['price']
                ]);
                Product::where('id', $product['id'])->update([
                    'unit' => $existUnit->unit -
$product['unit']
                ]);
            }//end foreach
```

		DB::commit();
		            return response()->json([
		                'status' => 'success',
		                'message' => 'Invoice created successfully'
		            ]);
		
		        }catch(Exception $e){
		            DB::rollBack();
		            return response()->json([
		                'status' => 'failed',
		                'message' => "Something went wrong"
		            ]);
		
		        }
		    }	

	web	Route::middleware(TokenVerificationMiddleware::class)->group(function(){
		      Route::post('/invoice-create', [InvoiceController::class, 'InvoiceCreate'])->name('InvoiceCreate');
		});
			postman

	InvoiceList method	public function InvoiceList(){
		        $user_id = request()->header('id');
		        $invoices = Invoice::with('customer')->where('user_id', $user_id)->get();
		        return $invoices;
		    }	

	web	`Route::middleware(TokenVerificationMiddleware::class)->group(function(){`
		`Route::get('/invoice-list', [InvoiceController::class, 'InvoiceList'])->name('InvoiceList');`
		`});`
			postman

	InvoiceDetails Method	`public function InvoiceDetails(Request $request){`
		`$user_id = request()->header('id');`
		`$customerDetails = Customer::where('user_id', $user_id)->where('id', $request->customer_id)->first();`
		`$invoiceDetails = Invoice::where('user_id', $user_id)->where('id', $request->invoice_id)->first();`
		`$invoiceProduct = Invoice_product::where('invoice_id', $request->invoice_id)`
		`->where('user_id',$user_id)->with('product')`
		`->get();`
		`return [`
		`'customer' => $customerDetails,`
		`'invoice' => $invoiceDetails,`
		`'product' => $invoiceProduct`
		`];`
		`}`
			InviceController.php

	Web	`Route::middleware(TokenVerificationMiddleware::class)->group(function(){`
		`Route::post('/invoice-details', [InvoiceController::class, 'InvoiceDetails'])->name('InvoiceDetails');`
		`});`
			postma

<table>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>InvoiceDelete</td><td><pre>
public function InvoiceDelete(Request $request, $id){
        DB::beginTransaction();
        try {
            $user_id = request()->header('id');
            Invoice_product::where('invoice_id', $id)
                ->where('user_id', $user_id)
                ->delete();

            Invoice::where('id', $id)
                ->where('user_id', $user_id)
                ->delete();

            DB::commit();
            return response()->json([
                'status' => 'success',
                'message' => 'Invoice deleted successfully'
            ]);

        }catch(Exception $e){
            DB::rollBack();
            return response()->json([
                'status' => 'failed',
                'message' => "Something went wrong"
            ]);

        }
    }
</pre></td><td></td></tr>
<tr><td></td><td>web</td><td><pre>
Route::middleware(TokenVerificationMiddleware::class)-
>group(function(){
</pre></td><td>Check postman</td></tr>
</table>

		`Route::get('/invoice-delete/{id}', [InvoiceController::class, 'InvoiceDelete'])->name('InvoiceDelete');`
		`});`	