Lab report no: 06

Name of the report: Eloquent ORM

Objectives:
Understand the relationship among object and how to mapping them. Also understand the different types of relation. Learn the basics of tinker.

Theory:
One of the most important feature of the laravel framework is that, it comes with built in ORM (Object Relation Mapping) called Eloquent ORM. Eloquent ORM refer to an advanced implementation of the PHP Active Record Pattern, which makes it very easy to interact with application database. Eloquent ORM is the very powerful yet very expressive ORM, which allow us to work with the database objects and relationships using much eloquent and expressive syntax. In Laravel, each database table is mapped into corresponding eloquent model and each of the eloquent model object include various methods for retrieving and updating the database.

Eloquent relationships are defined as methods on your Eloquent model classes. Since, like Eloquent models themselves, relationships also serve as powerful query builders defining relationships as methods provides powerful method chaining and querying capabilities. ORM stands for Object relation mapping.

different types of relationships:

- ➢ One To One
- ➢ One To Many
- ➢ Many To Many
- ➢ Has One Through
- ➢ Has Many Through
- ➢ One To One (Polymorphic)
- ➢ One To Many (Polymorphic)
- ➢ Many To Many (Polymorphic)

Creating database:

Here I create a database name mbstu_db.
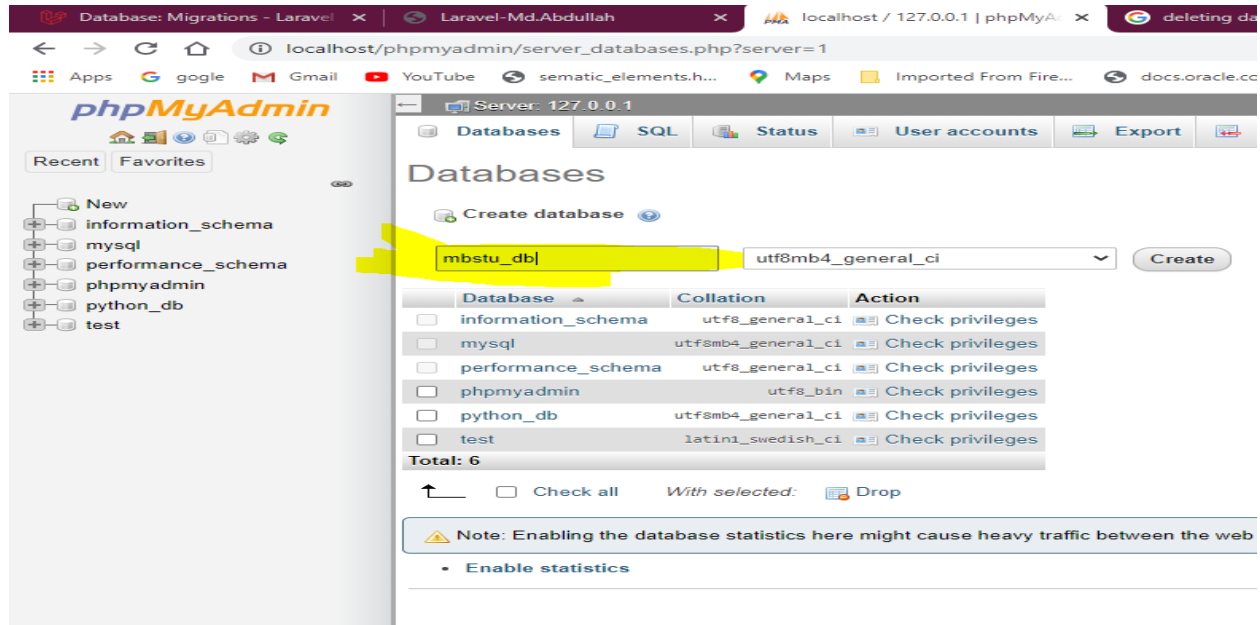
Setingup the .env file: to connect with database that I created above I need to setup the .env file with databases name and many things:

APP_NAME=MBSTU

APP_ENV=local

APP_KEY=base64:zAlNrq3yLqjyVWpBPWyyTZthjrvOe3zu90lP/uHusAE=

APP_DEBUG=true

APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=mbstu_db

DB_USERNAME=root

DB_PASSWORD=

Define table properties:

I want to create a three table name teacher, courses and student. So the properties of the table are given below:

Teacher table:

```php
database > migrations > 🐘 2020_07_14_112321_create_teachers_table.php > ✨ CreateTeachersTable > 🔷 up
 7    class CreateTeachersTable extends Migration
 8    {
 9        /**
10         * Run the migrations.
11         *
12         * @return void
13         */
14        public function up()
15        {
16            Schema::create('teachers', function (Blueprint $table) {
17                $table->id();
18                $table->string('name');
19                $table->string('email') ->unique();
20                $table->string('Designation');
21                $table->timestamps();
22            });
23        }
24
25        /**
26         * Reverse the migrations.
27         *
28         * @return void
29         */
30        public function down()
31        {
32            Schema::dropIfExists('teachers');
33        }
34    }
```

Courses table:

```php
base > migrations > 🐘 2020_08_14_111056_create_courses_table.php > ✨ CreateCoursesTable > 🔷 up
        /**
         * Run the migrations.
         *
         * @return void
         */
        public function up()
        {
            Schema::create('courses', function (Blueprint $table) {
                $table->id();
                $table->string('code') ->unique();
                $table->string('title') ->unique();
                $table->float('credit') ->nullable();
                $table->unsignedBigInteger('teacher_id');
                $table->foreign('teacher_id')->references('id')->on('teachers');
                $table->timestamps();
            });
        }

        /**
         * Reverse the migrations.
         *
         * @return void
         */
        public function down()
        {
            Schema::dropIfExists('courses');
        }
    }
```

Student table:

```
2020_08_14_101851_create_students_table.php ×

atabase > migrations > 🐘 2020_08_14_101851_create_students_table.php > ⌁ CreateStudentsTable > ⊕ up
 9          /**
10           * Run the migrations.
11           *
12           * @return void
13           */
14          public function up()
15          {
16              Schema::create('students', function (Blueprint $table) {
17                  $table->id();
18                  $table->string('name');
19                  $table->float('age') ->nullable();
20                  $table->string('contact') ->nullable();
21                  $table->timestamps();
22              });
23          }
24
25          /**
26           * Reverse the migrations.
27           *
28           * @return void
29           */
30          public function down()
31          {
32              Schema::dropIfExists('students');
33          }
34      }
35
```

Adding fillables:

Course table:

```
🐘 Course.php ×

app > 🐘 Course.php > ...
 1      <?php
 2
 3      namespace App;
 4
 5      use Illuminate\Database\Eloquent\Model;
 6
 7      class Course extends Model
 8      {
 9          //
10          protected $fillable=[
11              'code','title','credit','teacher_id'
12          ];
13      }
14
```

Teacher table:

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Teacher extends Model
{
    //
    protected $fillable=[
        'name','email','Designation'
    ];
}
```

Student table:

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Student extends Model
{
    //
    protected $fillable=[
        'name','age','contact'
    ];
}
```

Now the following commands need to execute inorder to create tables:

php artisan migrate

After the execution of the above command my mbstu_db database has all of the above tables look like the following figures:
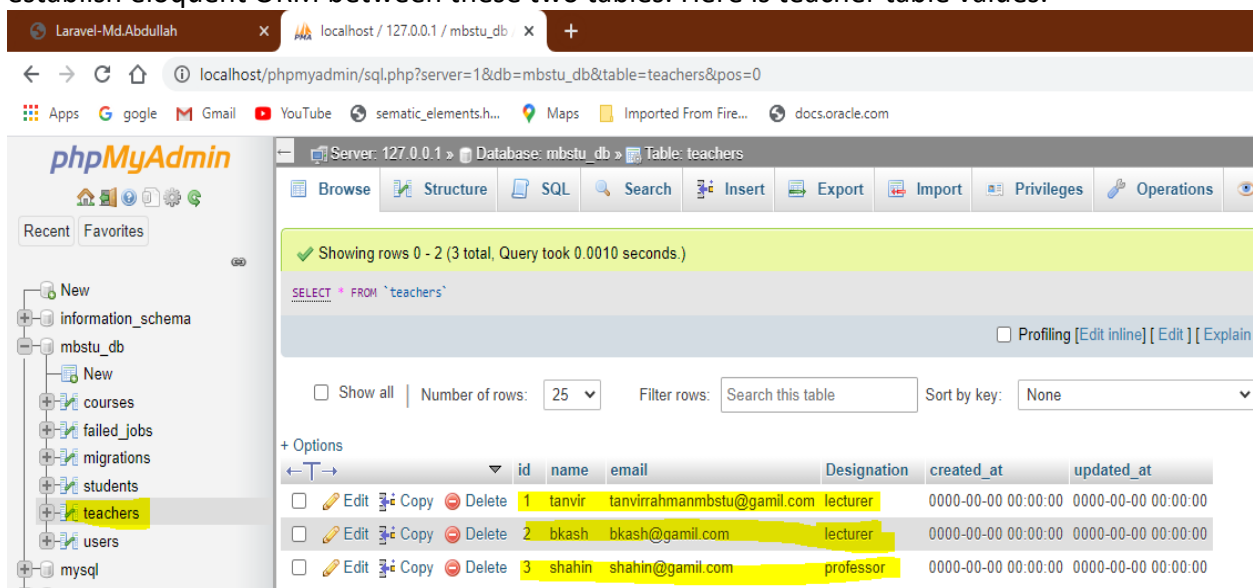
The tables are also created in the migratinos file:
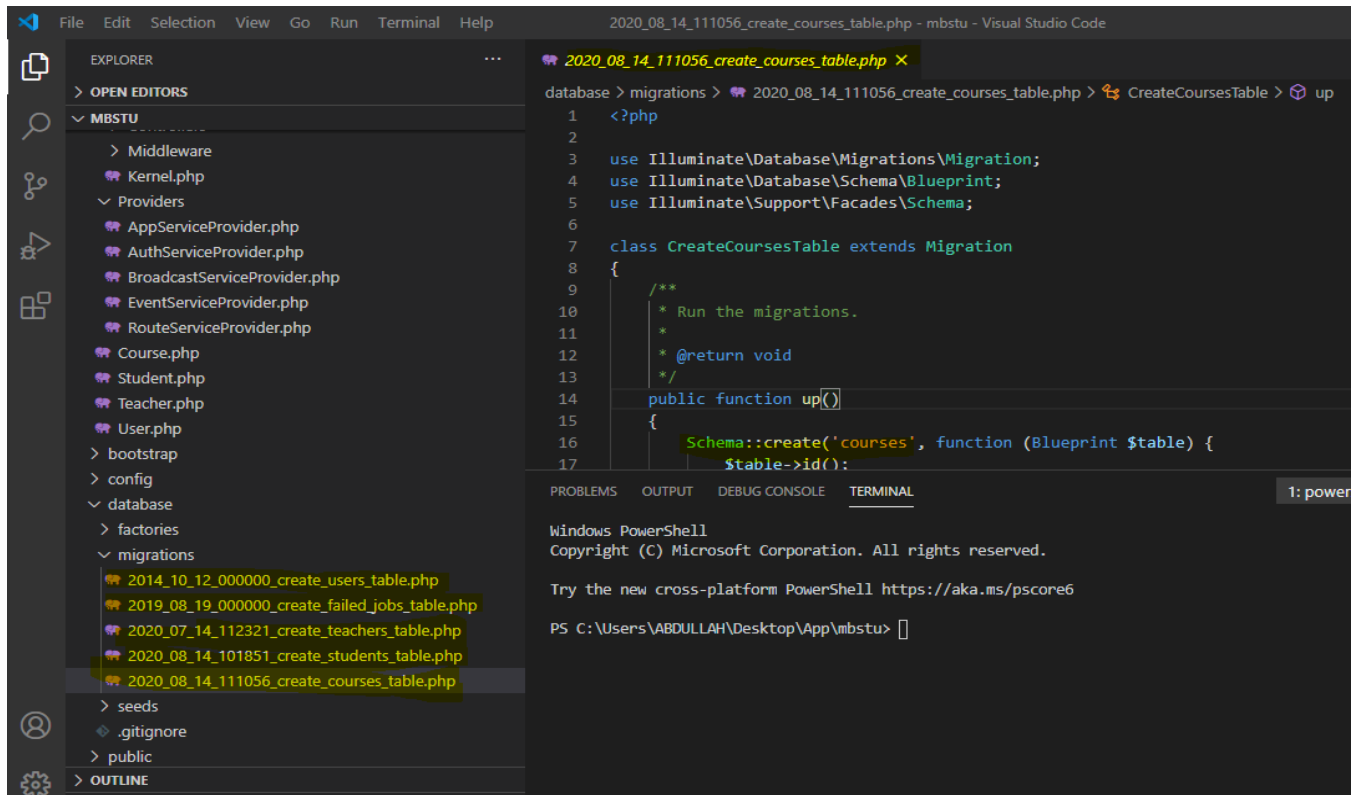
Inserting a value:

I manually insert the following values in courses table and teacher table because I want to establish eloquent ORM between these two tables. Here is teacher table values.
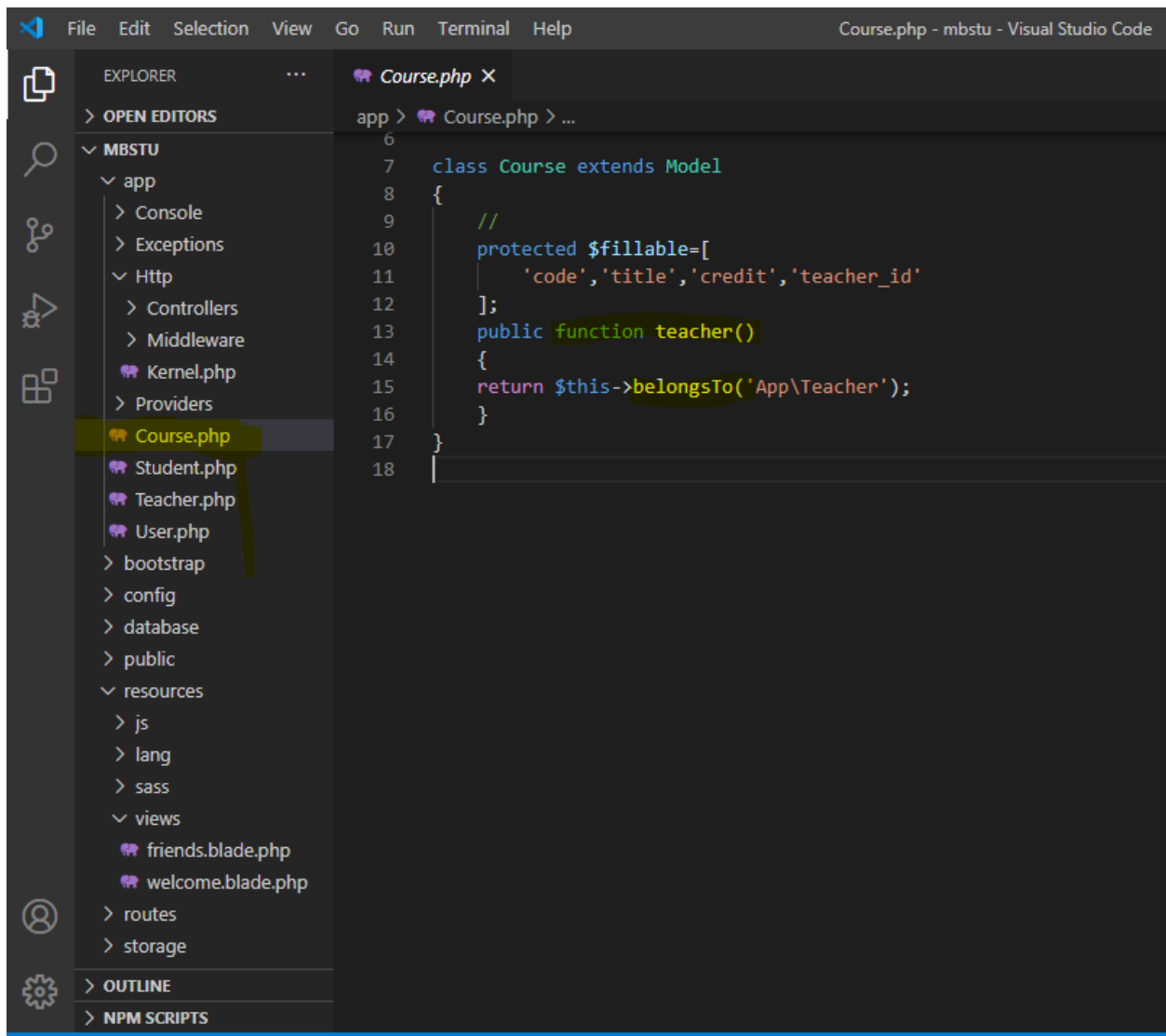
Course table values:

Establishing relationship:

We create a teacher table and course table. We want to build a relationship with teacher able and course table. A teacher can take multiple courses so we can say that a teacher has one to many relationships with course but on the other hand a course can teach by only one teacher, one course one teacher, so we can say that a course has one to one relationship with teacher.

So the relationships are:

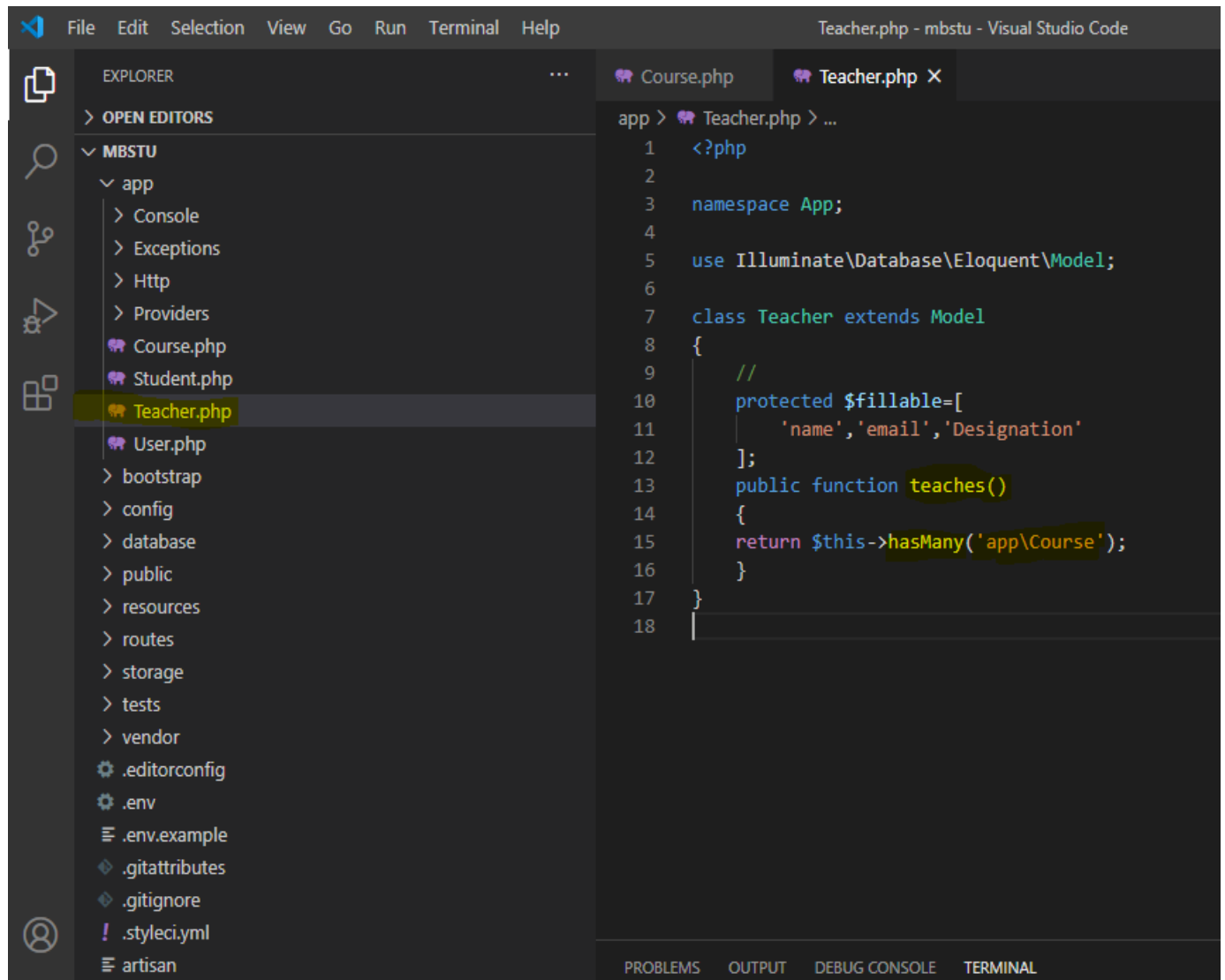❖ One to many – teacher with course
❖ One to one – course with teacher

Eloquent ORM:

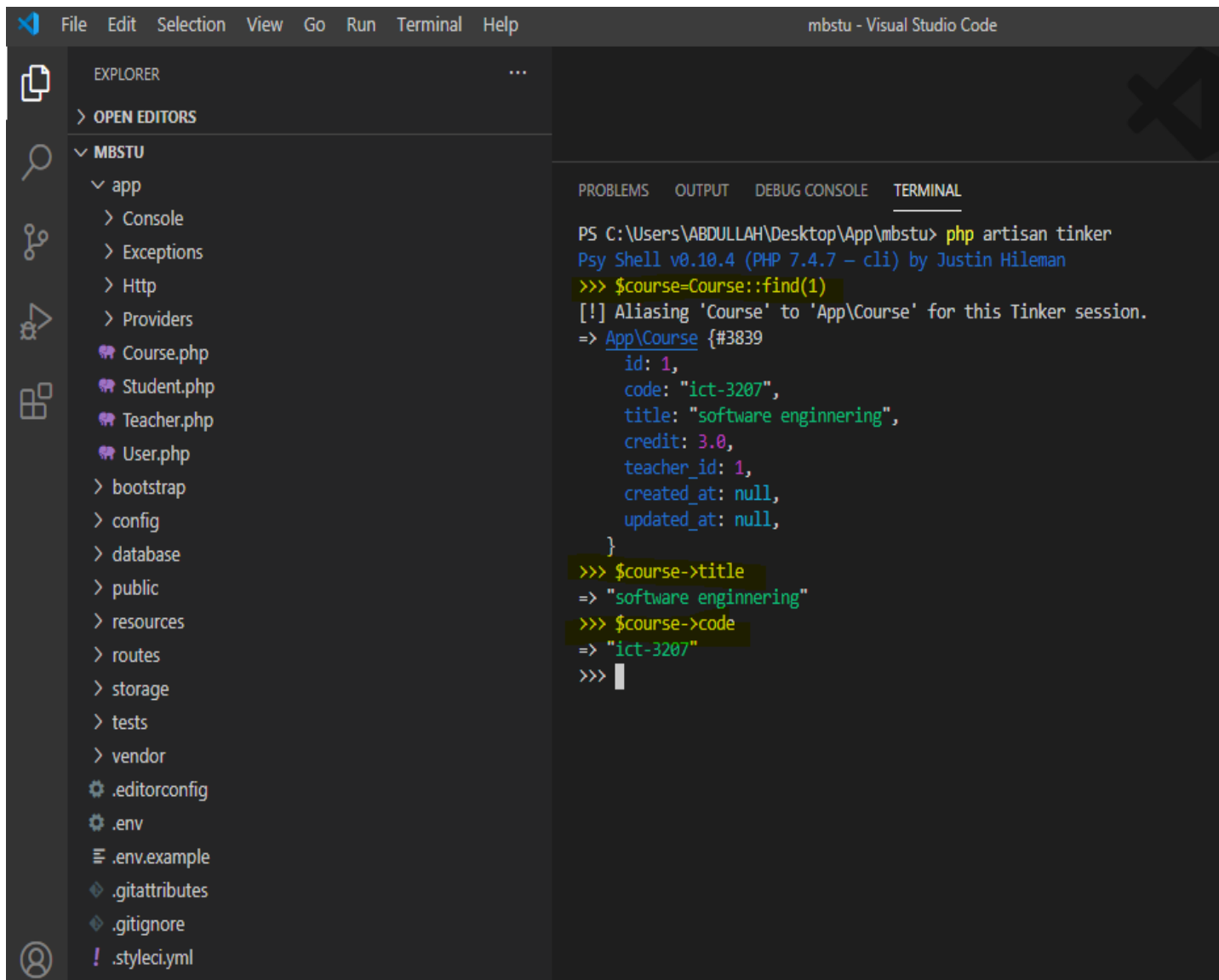**One to one:** course with teacher


```php
class Course extends Model
{
    //
    protected $fillable=[
        'code','title','credit','teacher_id'
    ];
    public function teacher()
    {
        return $this->belongsTo('App\Teacher');
    }
}
```

One to one: teacher with courses

File   Edit   Selection   View   Go   Run   Terminal   Help   Teacher.php - mbstu - Visual Studio Code

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Teacher extends Model
{
    //
    protected $fillable=[
        'name','email','Designation'
    ];
    public function teaches()
    {
    return $this->hasMany('app\Course');
    }
}
```

Finding the value:

Searching from course with specific id.

Has one relationship:


```
File  Edit  Selection  View  Go  Run  Terminal  Help                          mbstu - Visual Studio Code

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ABDULLAH\Desktop\App\mbstu> php artisan tinker
Psy Shell v0.10.4 (PHP 7.4.7 — cli) by Justin Hileman
>>> $course=Course::find(1)
[!] Aliasing 'Course' to 'App\Course' for this Tinker session.
=> App\Course {#3839
     id: 1,
     code: "ict-3207",
     title: "software enginnering",
     credit: 3.0,
     teacher_id: 1,
     created_at: null,
     updated_at: null,
   }
>>> $course->teacher
=> App\Teacher {#3773
     id: 1,
     name: "tanvir",
     email: "tanvirrahmanmbstu@gmail.com",
     Designation: "lecturer",
     created_at: "0000-00-00 00:00:00",
     updated_at: "0000-00-00 00:00:00",
   }
>>>
```

Searching from teacher:


```
File  Edit  Selection  View  Go  Run  Terminal  Help                    mbstu - Visual Studio Code

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ABDULLAH\Desktop\App\mbstu> php artisan tinker
Psy Shell v0.10.4 (PHP 7.4.7 — cli) by Justin Hileman
>>> $teacher=Teacher::find(1)
[!] Aliasing 'Teacher' to 'App\Teacher' for this Tinker session.
=> App\Teacher {#3839
     id: 1,
     name: "tanvir",
     email: "tanvirrahmanmbstu@gmail.com",
     Designation: "lecturer",
     created_at: "0000-00-00 00:00:00",
     updated_at: "0000-00-00 00:00:00",
   }
>>> $teacher->teaches
=> Illuminate\Database\Eloquent\Collection {#3773
     all: [
       App\Course {#3928
         id: 1,
         code: "ict-3207",
         title: "software enginnering",
         credit: 3.0,
         teacher_id: 1,
         created_at: null,
         updated_at: null,
       },
       App\Course {#3983
         id: 2,
         code: "ict-3208",
         title: "software enginnering lab",
         credit: 1.0,
         teacher_id: 1,
         created_at: null,
         updated_at: null,
       },
       App\Course {#3984
         id: 4,
```

File   Edit   Selection   View   Go   Run   Terminal   Help

EXPLORER   ...

> OPEN EDITORS
∨ MBSTU
  ∨ app
    > Console
    > Exceptions
    > Http
    > Providers
    🐘 Course.php
    🐘 Student.php
    🐘 Teacher.php
    🐘 User.php
  > bootstrap
  > config
  > database
  > public
  > resources
  > routes
  > storage
  > tests
  > vendor
  ⚙ .editorconfig
  ⚙ .env
  ☰ .env.example
  ◈ .gitattributes
  ◈ .gitignore
  ! .styleci.yml
  ☰ artisan
> OUTLINE
> NPM SCRIPTS

PROBLEMS   OUTPUT   DEBUG CONSOLE   **TERMINAL**

```
    Designation: "lecturer",
    created_at: "0000-00-00 00:00:00",
    updated_at: "0000-00-00 00:00:00",
  }
>>> $teacher->teaches
=> Illuminate\Database\Eloquent\Collection {#3773
     all: [
       App\Course {#3928
         id: 1,
         code: "ict-3207",
         title: "software enginnering",
         credit: 3.0,
         teacher_id: 1,
         created_at: null,
         updated_at: null,
       },
       App\Course {#3983
         id: 2,
         code: "ict-3208",
         title: "software enginnering lab",
         credit: 1.0,
         teacher_id: 1,
         created_at: null,
         updated_at: null,
       },
       App\Course {#3984
         id: 4,
         code: "ict-3206",
         title: "computer networks lab",
         credit: 1.0,
         teacher_id: 1,
         created_at: null,
         updated_at: null,
       },
     ],
   }
>>>
```

Find all the courses that teacher tanvir take.

```
File    Edit    Selection    View    Go    Run    Terminal    Help

EXPLORER              ...        PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

> OPEN EDITORS
                                     Designation: "lecturer",
∨ MBSTU                               created_at: "0000-00-00 00:00:00",
  ∨ app                               updated_at: "0000-00-00 00:00:00",
    > Console                        }
    > Exceptions             >>> $tanvir->teaches
    > Http                   => Illuminate\Database\Eloquent\Collection {#3997
    > Providers                  all: [
    🐘 Course.php                  App\Course {#4000
    🐘 Student.php                     id: 1,
    🐘 Teacher.php                     code: "ict-3207",
    🐘 User.php                        title: "software enginnering",
    > bootstrap                        credit: 3.0,
    > config                           teacher_id: 1,
    > database                         created_at: null,
    > public                           updated_at: null,
    > resources                      },
    > routes                         App\Course {#4001
    > storage                          id: 2,
    > tests                            code: "ict-3208",
    > vendor                           title: "software enginnering lab",
    ⚙ .editorconfig                    credit: 1.0,
    ⚙ .env                             teacher_id: 1,
    ☰ .env.example                     created_at: null,
    ◆ .gitattributes                   updated_at: null,
    ◆ .gitignore                     },
    ! .styleci.yml                   App\Course {#4002
    ☰ artisan                          id: 4,
                                       code: "ict-3206",
> OUTLINE                             title: "computer networks lab",
> NPM SCRIPTS                          credit: 1.0,
                                       teacher_id: 1,
                                       created_at: null,
                                       updated_at: null,
                                     },
                                   ],
                                 }
                             >>> ▌
```
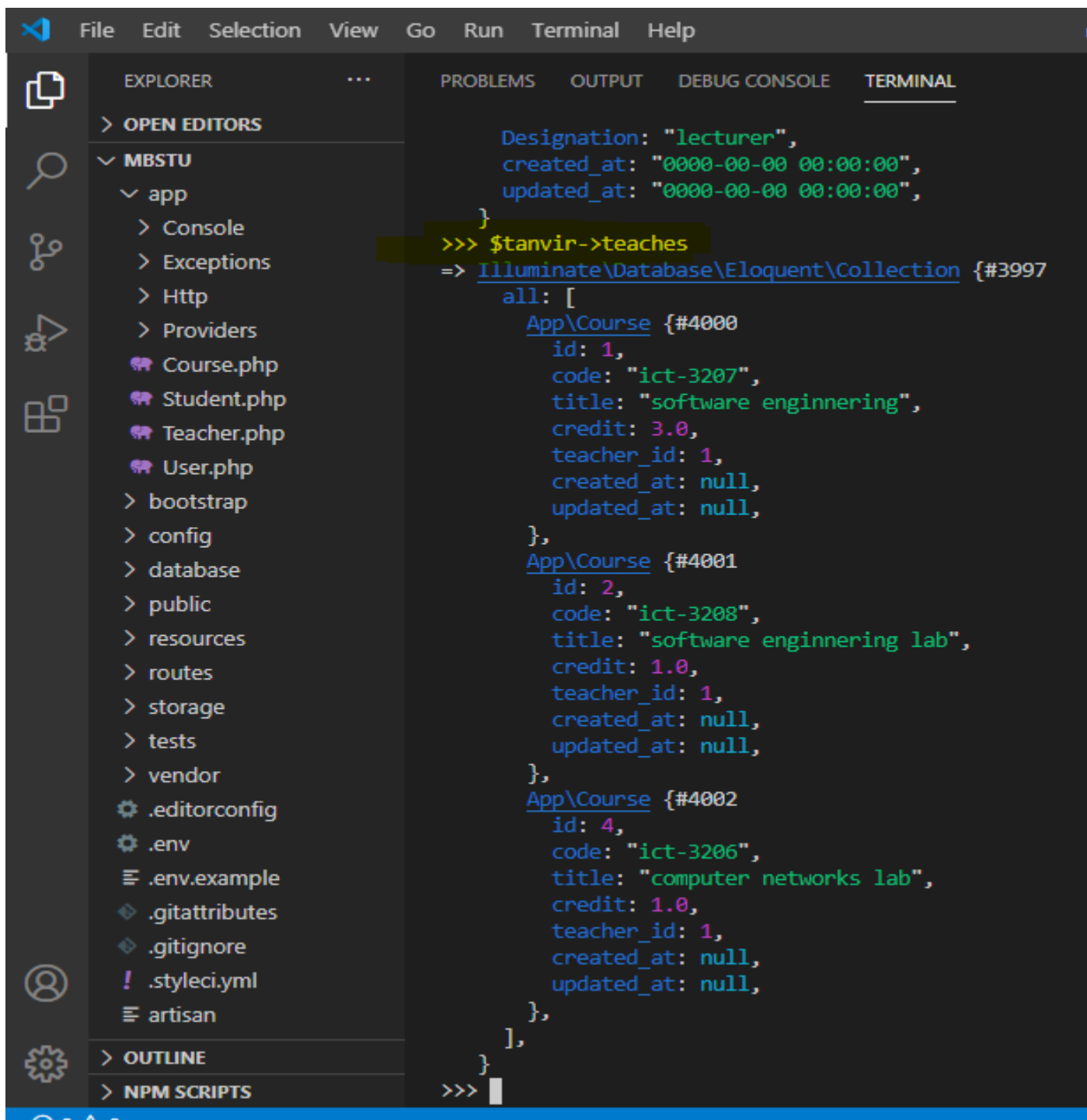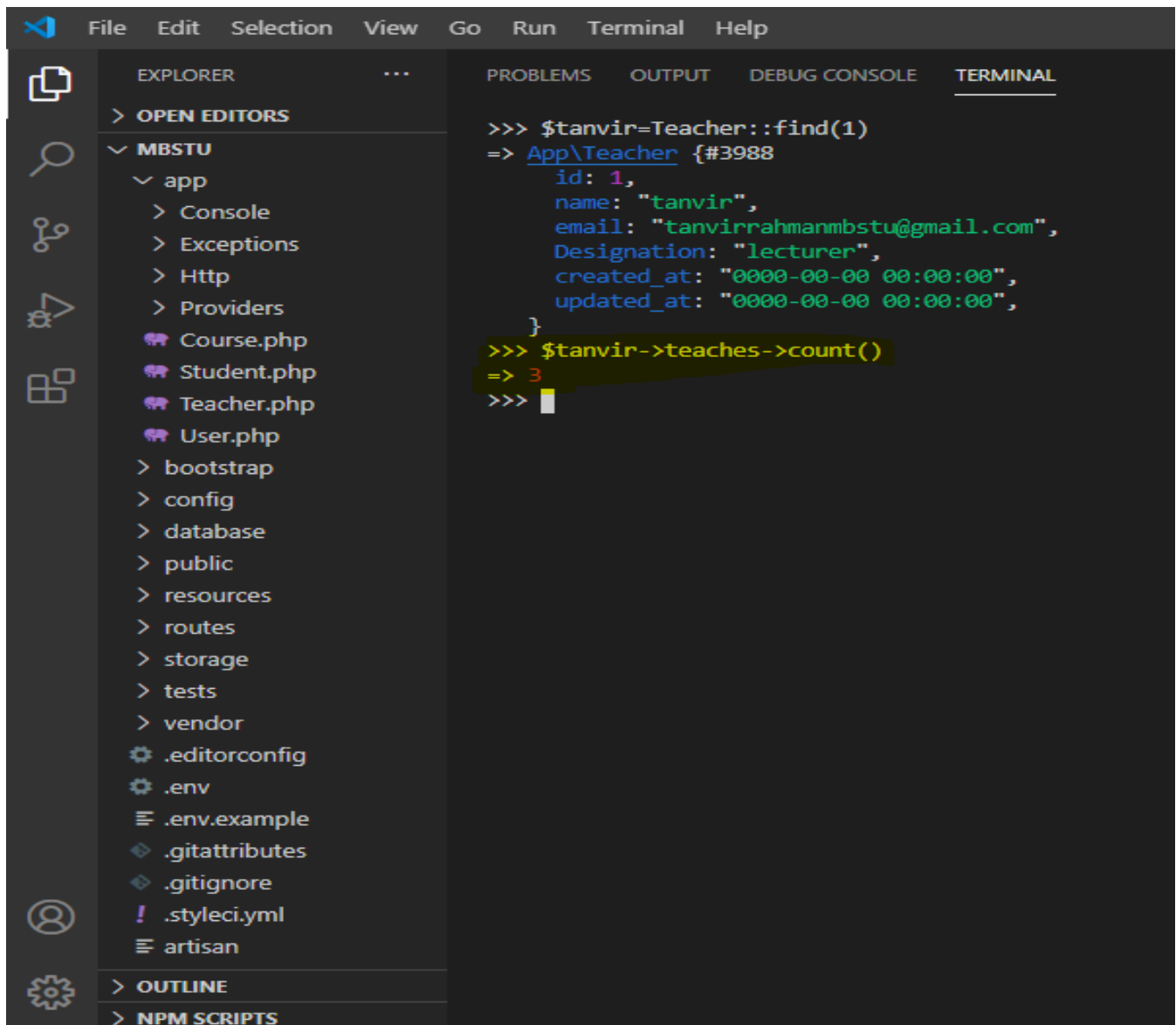
Counting the number of courses that a teacher take:

Here I count the number of courses that tanvir take.

Discussion:

I understand the basics of eloquent ORM or object mapping relationships. I also learn the basic of different types of relationship such as hasOne, one to many, many to many, one to one etc. This lab also helps me to understand the basic of tinker, how to access data using tinker