

TRƯỜNG ĐẠI HỌC THỦY LỢI  
KHOA CNTT

**BÀI TẬP THỰC HÀNH SỐ 03 – K65**  
**PHẦN 1 - Xây dựng CSDL**

**PRAC 01:**

**Tạo cơ sở dữ liệu và định nghĩa các Migrations để tạo bảng trong dự án, sau đó định nghĩa các bộ sinh Seeder để sinh dữ liệu minh họa.**

**medicines:**

medicine\_id (INT PRIMARY KEY): Mã thuốc  
name (VARCHAR(255)): Tên thuốc  
brand (VARCHAR(100)): Thương hiệu (tùy chọn)  
dosage (VARCHAR(50)): Thông tin liều lượng (ví dụ., 10mg tablets).  
form (VARCHAR(50)): Dạng thuốc (viên nén, viên nang, xi-rô, v.v.).  
price (DECIMAL(10,2)): Giá trên một đơn vị thuốc.  
stock (INT): Số lượng tồn trong kho.

**sales:**

sale\_id (INT PRIMARY KEY): Mã giao dịch bán hàng.  
medicine\_id (INT): Khóa ngoại tham chiếu drug\_id trong bảng Thuốc.  
quantity (INT): Số lượng thuốc bán ra.  
sale\_date (DATETIME): Ngày giờ bán hàng  
customer\_phone (VARCHAR(10)): Số điện thoại người mua (tùy chọn).

## **PRAC 02:**

**Tạo cơ sở dữ liệu và định nghĩa các Migrations để tạo bảng trong dự án, sau đó định nghĩa các bộ sinh Seeder để sinh dữ liệu minh họa.**

### **students**

id (INT, Primary Key): Mã học sinh.  
first\_name (VARCHAR(50)): Tên học sinh.  
last\_name (VARCHAR(50)): Họ đệm.  
date\_of\_birth (DATE): Ngày sinh  
parent\_phone (VARCHAR(20)): Số điện thoại phụ huynh.  
class\_id (INT, Khóa ngoại tham chiếu classes.id)  
**classes (Optional - Depending on your needs)**

id (INT, Primary Key): Mã lớp.  
grade\_level (ENUM('Pre-K', 'Kindergarten')): Cấp độ lớp.  
room\_number (VARCHAR(10)): Phòng học (Ví dụ: VH7, VH8).

## **PRAC 03:**

**Tạo cơ sở dữ liệu và định nghĩa các Migrations để tạo bảng trong dự án, sau đó định nghĩa các bộ sinh Seeder để sinh dữ liệu minh họa.**

**computers**

id (INT, Primary Key): Mã máy tính.  
computer\_name (VARCHAR(50)): Tên máy tính (ví dụ., "Lab1-PC05").  
model (VARCHAR(100)): Tên phiên bản (ví dụ., "Dell Optiplex 7090").  
operating\_system (VARCHAR(50)): Hệ điều hành (ví dụ "Windows 10 Pro").  
processor (VARCHAR(50)): Bộ vi xử lý (e.g., "Intel Core i5-11400").  
memory (INT): Bộ nhớ RAM (GB).  
available (BOOLEAN): Trạng thái hoạt động

**issues**

id (INT, Primary Key): Mã vấn đề báo cáo.  
computer\_id (INT, khóa ngoại tham chiếu computers.id)  
reported\_by (VARCHAR(50)): Người báo cáo sự cố (tùy chọn).  
reported\_date (DATETIME): Thời gian báo cáo.  
description (TEXT): Mô tả chi tiết vấn đề.  
urgency (ENUM('Low', 'Medium', 'High')): Mức độ sự cố.  
status (ENUM('Open', 'In Progress', 'Resolved')): Trạng thái hiện tại  
của sự cố (Mở, đang xử lý hoặc đã giải quyết).

## **PHẦN 2 - Xây dựng dự án Task List với Laravel Framework**

Tham khảo Video tương tự hướng dẫn Laravel:

<https://www.youtube.com/watch?v=SQ98BEIhZRo>

<https://www.youtube.com/watch?v=D5hLWZn71q8>

**Mục tiêu:**

- Sinh viên nắm vững kiến thức về CSDL quan hệ, thiết kế CSDL cho ứng dụng web sử dụng Laravel Migration.
- Sinh viên có khả năng xây dựng ứng dụng Task List với Laravel Framework

**Yêu cầu:**

- Sử dụng PHP và MySQL để xây dựng website.
- Áp dụng mô hình MVC trong tổ chức mã nguồn.
- Website có hai vai trò: người dùng khách và quản trị viên.
- Người dùng khách có thể xem tin tức, tìm kiếm tin tức, xem chi tiết tin tức.
- Quản trị viên có thể đăng nhập, đăng xuất, thêm, sửa, xóa tin tức.

### **HƯỚNG DẪN**

**Chuẩn bị:** SV cần cài đặt môi trường làm việc theo HD trên lớp

- SV thực hiện lại Bài tập theo cá nhân và hiểu Lý thuyết thông qua việc thực hành từng bước.

#### **1. Thiết kế cơ sở dữ liệu**

Thiết kế cơ sở dữ liệu cho một ứng dụng Task List đơn giản. Ứng dụng này cho phép người dùng tạo và quản lý các task (nhiệm vụ) cá nhân. Mỗi task bao gồm các thông tin sau:

- Tiêu đề (title)

- Mô tả ngắn gọn (description)
- Mô tả chi tiết (long\_description, không bắt buộc)
- Trạng thái hoàn thành (completed)

### Mô tả dạng lược đồ:

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	INT	PRIMARY KEY, AUTO_INCREMENT	Khóa chính, tự động tăng
title	VARCHAR(255)	NOT NULL	Tiêu đề của task
description	TEXT	NOT NULL	Mô tả ngắn gọn của task
long_description	TEXT	NULL	Mô tả chi tiết của task (không bắt buộc)
completed	BOOLEAN	NOT NULL, DEFAULT FALSE	Trạng thái hoàn thành của task (mặc định là false)
created_at	TIMESTAMP	NOT NULL	Thời gian tạo task
updated_at	TIMESTAMP	NOT NULL	Thời gian cập nhật task

### Mô tả dạng câu lệnh SQL

```
CREATE TABLE tasks (
    id INT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(255) NOT NULL,
    description TEXT NOT NULL,
    long_description TEXT,
    completed BOOLEAN NOT NULL DEFAULT FALSE,
    created_at TIMESTAMP NOT NULL,
    updated_at TIMESTAMP NOT NULL
```

);

### Mô tả dạng code định nghĩa trong Migration của Laravel:

```
Schema::create('tasks', function (Blueprint $table) {  
    $table->id();  
  
    $table->string('title');  
    $table->text('description');  
    $table->text('long_description')->nullable();  
    $table->boolean('completed')->default(false);  
  
    $table->timestamps();  
});
```

## 2. Tạo dự án Laravel

Xác định thư mục lưu trữ dự án, tại đó mở cửa sổ lệnh thực hiện

### Cách 1: Tạo trực tiếp

```
composer create-project laravel/laravel task-list-app
```

### Cách 2: Tạo dự án với laravel installer

Nếu chưa cài laravel installer thì thực hiện trước:

```
composer global require laravel/installer
```

Nếu đã cài laravel installer thì thực hiện

```
laravel new task-list-app
```

Và thực hiện lựa chọn các tùy chọn theo gợi ý

### 3. Tạo Cơ sở dữ liệu (Tạo các tệp migration)

**Bước 1:** Di chuyển vào thư mục dự án

```
cd task-list-app
```

**Bước 2:** Tạo migration cho bảng tasks

```
php artisan make:migration create_tasks_table --create=tasks
```

**Bước 3:** Mở file migration vừa tạo trong thư mục **database/migrations** và thay thế nội dung bằng đoạn mã sau:

```
public function up(): void
```

```
{
```

```
    Schema::create('tasks', function (Blueprint $table)
{
```

```
        $table->id();
```

```
        $table->string('title');
```

```
        $table->text('description');
```

```
        $table->text('long_description')->nullable();
```

```
        $table->boolean('completed')->default(false);
```

```
        $table->timestamps();
```

```
});
```

```
}
```

**Bước 4:** Cấu hình kết nối CSDL MySQL

Tạo CSDL có tên: **taskapp**

Sửa đổi tệp tin **.env** và khai báo các tham số kết nối CSDL của ứng dụng theo cấu hình trên máy bạn

```
DB_CONNECTION=mysql
```

```
DB_HOST=127.0.0.1
```

```
DB_PORT=3306
```

```
DB_DATABASE=taskapp  
DB_USERNAME=root  
DB_PASSWORD=
```

### Bước 5: Chạy migration

Chạy lệnh sau để tạo bảng **tasks** trong cơ sở dữ liệu:

```
php artisan migrate
```

### Bước 6: Tạo Model

Tạo model Task bằng lệnh sau:

```
php artisan make:model Task
```

### Cách 1: Sử dụng Factory

#### Bước 7: Tạo Factory

Tạo factory cho model Task để sinh dữ liệu giả:

```
php artisan make:factory TaskFactory
```

Mở file factory vừa tạo trong thư mục database/factories và thay thế nội dung bằng đoạn mã sau:

```
public function definition(): array
{
    return [
        'title' => fake()->sentence(),
        'description' => fake()->paragraph(),
        'long_description' => fake()->text(),
        'completed' => fake()->boolean(),
    ];
}
```

#### Bước 8: Sinh dữ liệu giả

Mở file **database/seeders/DatabaseSeeder.php** và thêm dòng sau vào hàm run():

```
\App\Models\Task::factory(10)->create();
```

Chạy lệnh sau để sinh 10 task giả:

```
php artisan db:seed
```

### Cách 2: Sử dụng seeder

## Bước 7: Tạo Seeder

Tạo factory cho model Task để sinh dữ liệu giả:

```
php artisan make:seeder TaskSeeder
```

Mở file seeder vừa tạo trong thư mục database/seeders và thay thế nội dung bằng đoạn mã sau:

```
public function run(): void
{
    $tasks = [
        [
            'title' => 'Đi chợ mua rau',
            'description' => 'Mua rau muống, rau cải, cà chua, hành lá',
            'long_description' => 'Nhớ mua rau sạch, rau hữu cơ ở cửa hàng uy tín. Chú ý chọn rau tươi, không bị dập nát.',
            'completed' => false,
        ],
        [
            'title' => 'Hoàn thành báo cáo',
            'description' => 'Hoàn thành báo cáo cuối tháng',
            'long_description' => 'Báo cáo cần bao gồm các số liệu về doanh thu, chi phí, lợi nhuận. Phân tích các yếu tố ảnh hưởng đến kết quả kinh doanh.',
            'completed' => true,
        ],
        [
            'title' => 'Học tiếng Anh',
            'description' => 'Học 30 từ vựng mới',
            'long_description' => null,
            'completed' => false,
        ],
        // Thêm các task khác vào đây...
    ];

    foreach ($tasks as $task) {
        Task::create($task);
    }
}
```

hoặc dùng Faker

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\\Seeder;
use App\Models\Task;
use Faker\Factory as Faker;

class TaskSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $faker = Faker::create();

        foreach (range(1, 10) as $index) {
            Task::create([
                'title' => $faker->sentence(),
                'description' => $faker->paragraph(),
                'long_description' => $faker->optional()->text(),
                'completed' => $faker->boolean(),
            ]);
        }
    }
}
```

#### Bước 8: Sinh dữ liệu giả

Mở file **database/seeders/DatabaseSeeder.php** và thêm dòng sau vào hàm run():

```
$this->call([
    TaskSeeder::class,
]);
```

Chạy lệnh sau để sinh 10 task giả:

```
php artisan db:seed
```

hoặc chỉ định trực tiếp

```
php artisan db:seed --class=TaskSeeder
```

### Bước 9: (Tùy chọn) Tạo Controller và Route

Bạn có thể tạo controller và route để hiển thị danh sách task, tạo task mới, v.v.

Ví dụ:

```
php artisan make:controller TaskController -resource
```

Thêm route vào file routes/web.php:

```
Route::resource('tasks', TaskController::class);
```

```
<?php

namespace App\Http\Controllers;

use App\Models\Task;
use Illuminate\Http\Request;

class TaskController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $tasks = Task::all();
        return view('tasks.index', compact('tasks'));
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
```

```
        return view('tasks.create');
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        $request->validate([
            'title' => 'required',
            'description' => 'required',
        ]);

        Task::create($request->all());

        return redirect()->route('tasks.index')->with('success', 'Task created successfully.');
    }

    /**
     * Display the specified resource.
     */
    public function show(Task $task)
    {
        return view('tasks.show', compact('task'));
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit(Task $task)
    {
        return view('tasks.edit', compact('task'));
    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, Task $task)
```

```

{
    $request->validate([
        'title' => 'required',
        'description' => 'required',
    ]);

    $task->update($request->all());

    return redirect()->route('tasks.index')-
>with('success', 'Task updated successfully.');
}

/**
 * Remove the specified resource from storage.
 */
public function destroy(Task $task)
{
    $task->delete();

    return redirect()->route('tasks.index')-
>with('success', 'Task deleted successfully.');
}
}

```

#### Bước 10: Giao diện (views/tasks):

```

App.blade.php

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <title>Task List App</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css
/bootstrap.min.css" rel="stylesheet" integrity="sha384-

```

```
9ndCyUaIbzAi2FUVXJi0CjmCapSm07SnpJef0486qhLnuZ2cdeRh002iuK6
FUUVM" crossorigin="anonymous">
</head>
<body>

    <nav class="navbar navbar-expand-lg navbar-light bg-
light">
        <div class="container">
            <a class="navbar-brand" href="{{
route('tasks.index') }}>Task List App</a>
            <button class="navbar-toggler" type="button"
data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse"
id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link active" aria-
current="page" href="{{ route('tasks.index') }}>Danh sách
Task</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{
route('tasks.create') }}>Thêm mới</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container mt-4">
        @yield('content')
    </div>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/b
```

```
ootstrap.bundle.min.js" integrity="sha384-  
geWF76RCwLtnZ8qwWowPQNgul3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3T  
e4Bkz" crossorigin="anonymous">></script>  
</body>  
</html>
```

### index.blade.php

```
@extends('layouts.app')  
  
@section('content')  
    <div class="container">  
        <h1>Danh sách Task</h1>  
        <a href="{{ route('tasks.create') }}" class="btn  
btn-primary">Thêm mới</a>  
        <table class="table">  
            <thead>  
                <tr>  
                    <th>ID</th>  
                    <th>Tiêu đề</th>  
                    <th>Mô tả</th>  
                    <th>Trạng thái</th>  
                    <th>Hành động</th>  
                </tr>  
            </thead>  
            <tbody>  
                @foreach ($tasks as $task)  
                    <tr>  
                        <td>{{ $task->id }}</td>  
                        <td>{{ $task->title }}</td>  
                        <td>{{ $task->description }}</td>  
                        <td>{{ $task->completed ? 'Hoàn  
thành' : 'Chưa hoàn thành' }}</td>  
                        <td>  
                            <a href="{{ route('tasks.show',  
$task->id) }}" class="btn btn-info">Xem</a>
```

```

                <a href="{{ route('tasks.edit', $task->id) }}" class="btn btn-warning">Sửa</a>
                <form action="{{ route('tasks.destroy', $task->id) }}" method="POST"
style="display: inline-block;">
                @csrf
                @method('DELETE')
                <button type="submit"
class="btn btn-danger" onclick="return confirm('Bạn có chắc
chắn muốn xóa?')">Xóa</button>
            </form>
        </td>
    </tr>
@endforeach
</tbody>
</table>
</div>
@endsection

```

### **create.blade.php**

```

@extends('layouts.app')

@section('content')


<h1>Thêm mới Task</h1>
    <form action="{{ route('tasks.store') }}"
method="POST">
        @csrf
        <div class="form-group">
            <label for="title">Tiêu đề:</label>
            <input type="text" class="form-control"
id="title" name="title" required>
        </div>
        <div class="form-group">
            <label for="description">Mô tả:</label>
            <textarea class="form-control"
id="description" name="description" required></textarea>


```

```

        </div>
        <div class="form-group">
            <label for="long_description">Mô tả chi
tiết:</label>
            <textarea class="form-control"
id="long_description" name="long_description"></textarea>
        </div>
        <div class="form-group form-check">
            <input type="checkbox" class="form-check-
input" id="completed" name="completed">
            <label class="form-check-label"
for="completed">Hoàn thành</label>
        </div>
        <button type="submit" class="btn btn-
primary">Lưu</button>
    </form>
</div>
@endsection

```

```

show.blade.php
@extends('layouts.app')

@section('content')
    <div class="container">
        <h1>Chi tiết Task</h1>
        <p><strong>Tiêu đề:</strong> {{ $task->title }}</p>
        <p><strong>Mô tả:</strong> {{ $task->description
}}</p>
        <p><strong>Mô tả chi tiết:</strong> {{ $task-
>long_description }}</p>
        <p><strong>Trạng thái:</strong> {{ $task->completed
? 'Hoàn thành' : 'Chưa hoàn thành' }}</p>
        <a href="{{ route('tasks.index') }}" class="btn
btn-secondary">Quay lại</a>
    </div>
@endsection

```

**edit.blade.php**

```

@extends('layouts.app')

@section('content')
    <div class="container">
        <h1>Sửa Task</h1>
        <form action="{{ route('tasks.update', $task->id) }}"
} method="POST">
            @csrf
            @method('PUT')
            <div class="form-group">
                <label for="title">Tiêu đề:</label>
                <input type="text" class="form-control"
id="title" name="title" value="{{ $task->title }}"
required>
            </div>
            <div class="form-group">
                <label for="description">Mô tả:</label>
                <textarea class="form-control"
id="description" name="description" required>{{ $task-
>description }}</textarea>
            </div>
            <div class="form-group">
                <label for="long_description">Mô tả chi
tiết:</label>
                <textarea class="form-control"
id="long_description" name="long_description">{{ $task-
>long_description }}</textarea>
            </div>
            <div class="form-group form-check">
                <input type="checkbox" class="form-check-
input" id="completed" name="completed" {{ $task->completed
? 'checked' : '' }}>
                <label class="form-check-label"
for="completed">Hoàn thành</label>
            </div>
            <button type="submit" class="btn btn-
primary">Lưu</button>
        </form>

```

```
</div>  
@endsection
```

#### 4. Một số giao diện gợi ý

##### The list of tasks

[Add Task!](#)

~~Unde architecto aut enim ducimus dolor laboriosam.~~

~~Consequatur aut quisquam magnam et.~~

Maxime minus qui inventore ducimus recusandae et.

~~Deleniti amet quibusdam sint suscipit consequatur aliquid eveniet.~~

Nulla sit quas labore ex quia eum qui.

A minima dignissimos similique assumenda voluptas nesciunt.

~~Minima adipisci illo aut enim.~~

Unde inventore fuga eaque.

~~Asperiores dolor consequuntur ipsam natus quia nemo.~~

Totam excepturi molestiae doloremque cumque necessitatibus odit ipsa.

Showing 1 to 10 of 20 results

## Add Task

TITLE

DESCRIPTION

LONG DESCRIPTION

[Add Task](#)

[Cancel](#)

# Unde architecto aut enim ducimus dolor laboriosam.

[← Go back to the task list!](#)

Aut neque consequatur vel est libero architecto ducimus. Itaque magni ullam sint ipsam asperiores repudiandae hic. Ut sed consequuntur consequatur inventore nobis est rerum.

Alias sed veniam aspernatur voluptatem mollitia consequatur. Nulla aperiam quisquam sit laborum dolor magnam. Sed et exercitationem quia et quia accusamus voluptas. Deleniti quod odio enim quia eum. Inventore laboriosam sed necessitatibus aut eveniet quis pariatur. Totam quas quia libero qui esse et voluptates.

Created 2 minutes ago • Updated 2 minutes ago

Completed

Edit

Mark as not completed

Delete

## Edit Task

### TITLE

Unde architecto aut enim ducimus dolor laboriosam.

### DESCRIPTION

Aut neque consequatur vel est libero architecto ducimus. Itaque magni ullam sint ipsam asperiores repudiandae hic. Ut sed consequuntur consequatur inventore nobis est rerum.

### LONG DESCRIPTION

Alias sed veniam aspernatur voluptatem mollitia consequatur. Nulla aperiam quisquam sit laborum dolor magnam. Sed et exercitationem quia et quia accusamus voluptas. Deleniti quod odio enim quia eum. Inventore laboriosam sed necessitatibus aut eveniet quis pariatur. Totam quas quia libero qui esse et voluptates.

[Update Task](#) [Cancel](#)

## The list of tasks

**Success!**

X

Task deleted successfully!

Add Task!

~~Consequatur aut quisquam magnam et.~~

Maxime minus qui inventore ducimus recusandae et.

~~Deleniti amet quibusdam sint suscipit consequatur aliquid eveniet.~~

Nulla sit quas labore ex quia eum qui.

A minima dignissimos similique assumenda voluptas nesciunt.

~~Minima adipisci illo aut enim.~~

Unde inventore fuga eaque.

~~Asperiores dolor consequuntur ipsam natus quia nemo.~~

Totam excepturi molestiae doloremque cumque necessitatibus odit ipsa.

Quos et rem sed quisquam et et quisquam.

Showing 1 to 10 of 19 results

Maxime minus qui inventore ducimus  
recusandae et.

**Success!**

X

Task updated successfully!

[← Go back to the task list!](#)

Ducimus hic dolores quibusdam quasi. Qui est qui deserunt rerum alias.  
Impedit vero earum facilis. Eum ut molestiae odit atque.

At suscipit fugiat ab quo quisquam quia ut eveniet. Tempora magni eum  
exercitationem et quis odio sunt. Molestiae ut reiciendis autem illo  
consectetur. Ut dolor doloribus laboriosam sunt ratione ut odit et. Ut qui  
tempore nisi odio tempora nemo et. Modi soluta aliquam tempore qui  
voluptatibus voluptatem at. Beatae sunt aut est totam eum distinctio.  
Earum at iusto totam assumenda necessitatibus maxime quasi et.  
Blanditiis ratione consequuntur assumenda deserunt. Odit ut aut dolor  
asperiores.

Created 3 minutes ago • Updated 0 seconds ago

**Completed**

Edit

Mark as not completed

Delete

HẾT