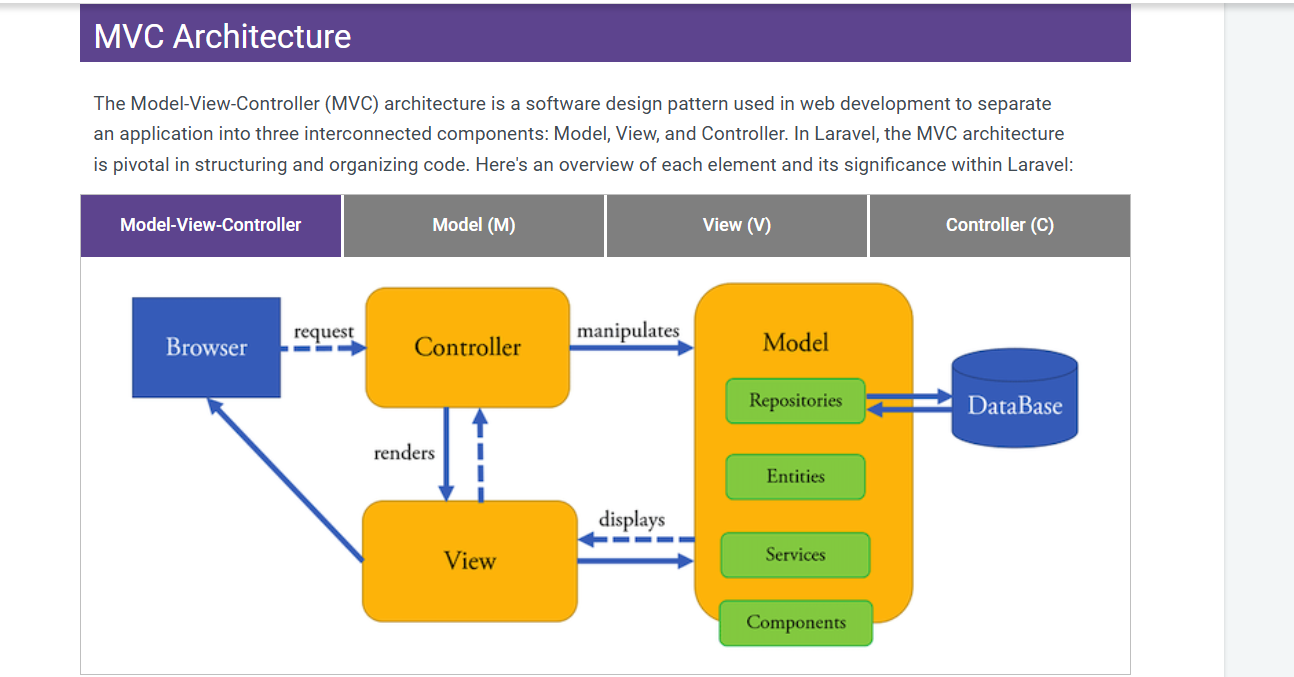
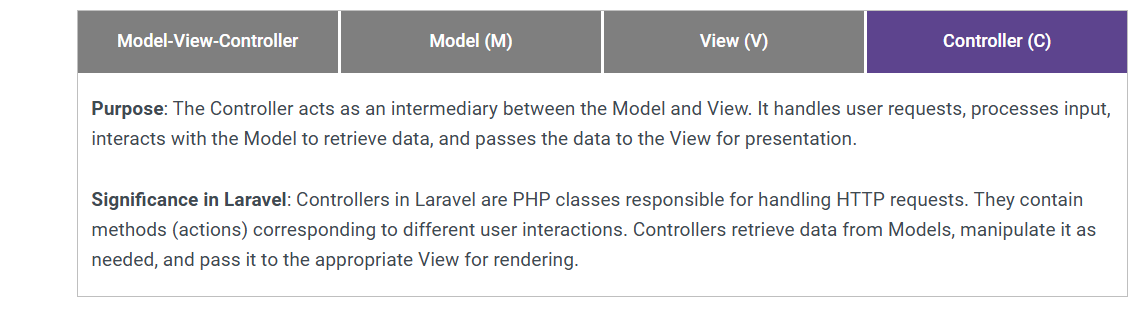
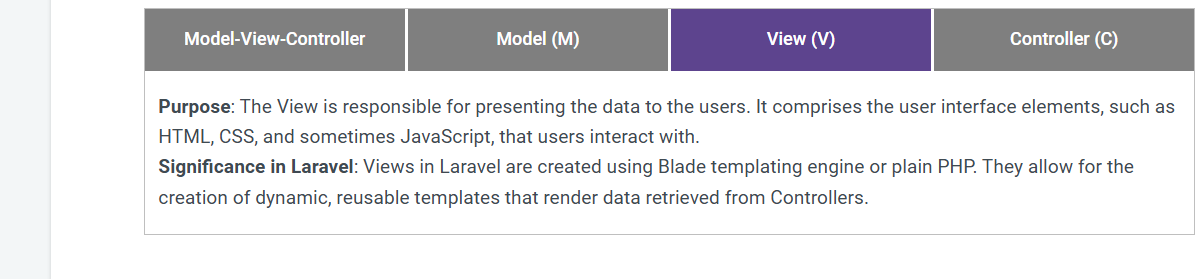
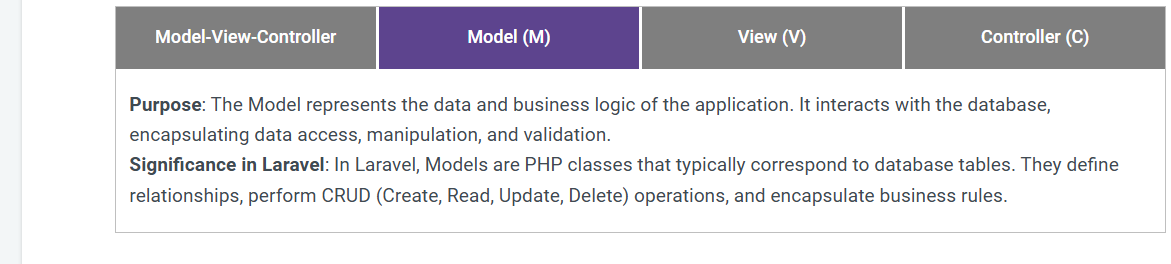
IMVC (Model-View-Controller) is a **design pattern** used in software development to organize code neatly. It divides a program into **three main parts**:

1. **Model (Data & Logic) 📊**
   * Think of this as the **brain** of the application.
   * It handles **data, business logic, and database interactions**.
   * Example: A database table storing student records.
2. **View (User Interface) 🖥️**
   * This is **what the user sees** (UI).
   * It displays data from the Model and takes input from users.
   * Example: A webpage showing student details.
3. **Controller (Handles User Actions) 🎮**
   * Acts as a **bridge** between Model and View.
   * It processes **user inputs**, updates the Model, and refreshes the View.
   * Example: Clicking "Save" on a form triggers the Controller to store data in the database.





**Model** is represented by Eloquent ORM entity:  
  
  
**<?php**  
**namespace App;**  
**use Illuminate\Database\Eloquent\Model;**  
  
  
**class User extends Model**  
**{**  
**public function posts() {**  
**return $this->hasMany(Post::class);**  
**}**  
**}**

**Eloquent ORM**: A feature in Laravel that helps interact with the database using objects instead of raw SQL.

**View** is represented by Blade Templates:  
  
**<!-- resources/views/users/show.blade.php -->**  
**<h1>User Profile</h1>**  
**<p>Name: {{ $user->name }}</p>**  
**<p>Email: {{ $user->email }}</p>**

**Controller** is represented by the corresponding classes:  
  
**<?php**  
**namespace App\Http\Controllers;**  
**use App\User; use Illuminate\Http\Request;**  
  
  
**class UserController extends Controller**  
**{**  
**public function show($id) {**  
**$user = User::find($id);**  
**return view('users.show', ['user' => $user]);**  
**}**  
**}**