**Experiment 11**

**Aim**: Case study on MVC Architecture

* **MVC Architecture**

MVC stands for Model, View and Controller. MVC separates application into three components - Model, View and Controller.

**Model**:

* Model is a data and business logic.
* Model represents shape of the data and business logic.
* It maintains the data of the application. Model objects retrieve and store model state in a database.

**View**:

* View is a User Interface.
* View is a user interface. View display data using model to the user and also enables them to modify the data.

**Controller**:

* Controller is a request handler.
* Controller handles the user request.
* Typically, user interact with View, which in-turn raises appropriate URL request, this request will be handled by a controller.
* The controller renders the appropriate view with the model data as a response.



Figure 1: MVC Architecture

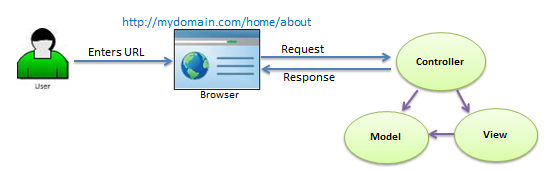
[](https://www.tutorialsteacher.com/Content/images/mvc/request-handling-in-mvc.png)

Figure 2: Flow of User’s Request in ASP.NET MVC

Request/Response in MVC Architecture:

As per the above figure, when the user enters a URL in the browser, it goes to the server and calls appropriate controller. Then, the Controller uses the appropriate View and Model and creates the response and sends it back to the user.

* **Laravel’s Main Features**

Laravel is one of the most popular and widely used opensource frameworks today. Laravel is a PHP MVC framework which offers a standardized and feature packed platform for high performing [PHP web applications development](https://www.techcronus.com/php-web-development-services/).

Here are some of the best features of PHP Laravel Framework which makes it popular among developers and businesses.

**1. Template Engine:**  
Laravel framework is highly acknowledged for its built-in lightweight templates which can be used to create wonderful layouts using dynamic content seeding. In addition to this, it provides multiple widgets incorporating CSS and JS code with robust structures. Laravel templates are innovatively designed to create simple as well as complex layouts with distinctive sections.

**2. MVC Architecture Support:**  
Laravel supports MVC architecture pattern which ensures separates business logic and presentation layers. MVC pattern of Laravel has a lot of built-in functions, improves application performance and increases security as well as scalability.

**3. Eloquent ORM (Object Relational Mapping):**  
Laravel offers Eloquent Object Relational Mapping (ORM) which includes a simple PHP Active Record implementation. This allows web application developers to write database queries with PHP syntax rather than writing SQL code. An ORM is relatively faster than other PHP frameworks.

**4. Security:**  
Laravel framework offers very strong web application security. It uses hashed and salted password mechanism so the password would never be saved as plain text in the database. It also uses “Bcrypt Hashing Algorithm” for generating an encrypted password. Additionally, this PHP web development framework uses prepared SQL statements that prevents SQL injection attacks.

**5. Artisan:**  
Laravel framework offers a built-in command line tool called Artisan which helps automating the majority of tedious repetitive programming tasks. These artisans can also be utilized to create the database structure, a skeleton code, and manage migration so it is pretty easy-to-manage database system. In addition, it can generate basic MVC files through command-line and manage those assets as well as their respective configurations. Artisan even helps developers to create their own commands and use them as required.

**6. Libraries & Modular:**  
Laravel comes with pre-installed Object-Oriented and Modular libraries which are not available in many other PHP frameworks. For example, an Authentication library which is easy-to-implement and has featuressuch as checking active users, Bcrypt hashing, password reset, CSRF (Cross-site Request Forgery) protection, and encryption. Furthermore, this framework is divided into individual modules adopting modern PHP principles facilitating responsive and modular web applications development.

**7. Database Migration System:**  
Laravel migration system helps to expand the web application database structure without re-creating everytime when there is a change in code. Because of this feature, the risk of losing data is very minimal. It not only offers the facility to change the database structure but also helps to do using PHP code instead of SQL. Laravel Schema Builder helps creating database tables and inserts indices or columns promptly.

**8. Unit-Testing:**  
Laravel is a preferred framework for web application developers today due to the way it facilitates unit testing. This framework is capable of running multiple unit tests to ensure that new changes done by the developer do not unexpectedly break the web application. Ideally, Laravel based web applications are capable of stable releases as it is careful of the known failures. It also make it easy for the developers to write unit-tests.

* **Architecture of Laravel**
* Laravel is a software architecture standard that separates the representation of information from users' interaction with it. The architectural standard that it has adopted is not so new; it has been around since the mid-1970s.
* In a very basic MVC workflow, when a user interacts with our application, the steps in the following screenshot are performed. Imagine a simple web application about books, with a search input field. When the user types a book name and presses **Enter**, the following flow cycle will occur:

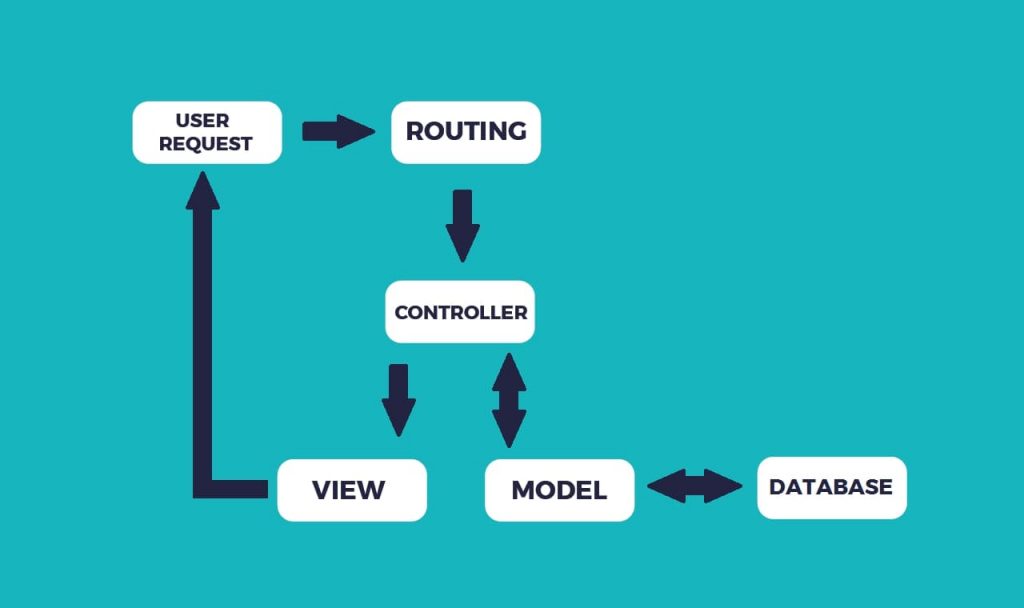


Figure 3: Laravel Architecture

* **Composer**
* Composer is a PHP dependency manager.

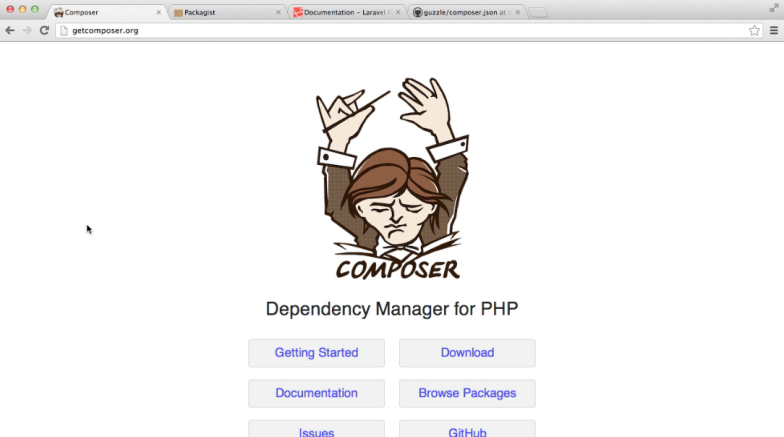


Figure 4: Composer

* It loads automatically once and becomes available system-wide. It follows the DRY principle. This way, you save your precious time and your code stays beautiful. It is very useful if you have more than one developer on the project.
* This dependencies management (or Composer) for a project is not new. It is actually inspired by Bundler in Rails or NPM in Node.js.
* Laravel implements composer for managing dependencies within it. Hence, before use of Laravel, it needs to check whether you have composer setup on your system or not.
* If you don't have composer installed on your computer, first visit this URL to download composer:
* <https://getcomposer.org/download/>
* When you are done installing the composer, cross-check whether it is installed or not, by typing in the command prompt the composer command. You can see the Composer screen in that CMD only.
* It needs to be kept in mind to put the composer's system-wide vendor in bin directory within your $PATH; as a result, your system can locate the executable of laravel.
* SetUp Laravel Using Installer
* First of all, you have to download the installer of Laravel with the help of Composer, like this:
* composer global require "laravel/installer"
* When the installation is done, *new* command of laravel will start a new fresh installation in that directory you provide.
* laravel new nirectory\_name
* **Create Project**
* The next thing you have to do is make a new folder in some specific path within your system, for keeping your Laravel projects. Move to that location where the directory is created. For installing the Laravel, the following command you have to type:
* composer create-project laravel/laravel - prefer -dist
* The command mentioned above will make Laravel installed on that specific directory. Type the next command:
* php artisan serve
* This above code will start Laravel service. A black screen will appear showing the message: Laravel Development server started on http://localhost:8080
* Copy and paste: http://localhost:8080 in your browser, and you can see Laravel home screen appears in your browser.
* **Conclusion:** In this experiment we learned about the MVC architecture, and how it stores the data of the application. MVC stands for Model View and controller. We studied about it and then the main features of Laravel’s architecture.