

DEBRE BERHAN UNIVERSITY



COLLEGE OF COMPUTING

Selected Topics in Computer Science

DEPARTEMENT OF COMPUTER SCINCE

Individual Assignment

NAME

TSEDEY SHEWAKENA

ID No:

DBUE /809/11

Submitted to: Girmachew

Submission date:20/05/15E.C

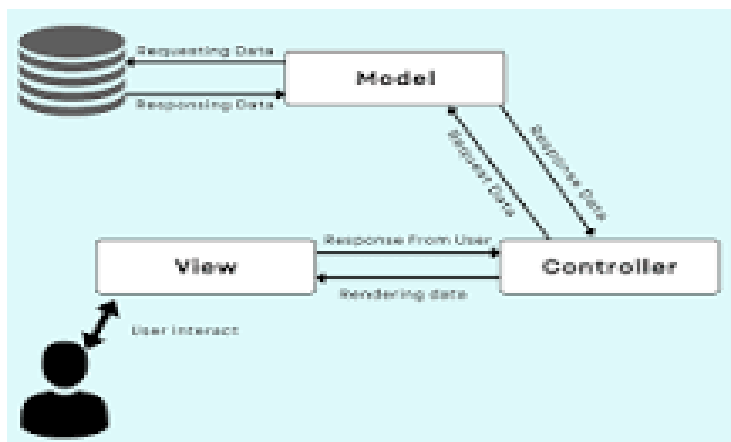
❖ Explain MVC of laravel

MVC stands for Model, View & Controller. It represents architecture developers adopt when building applications.

Laravel is a free and open-source web PHP framework, which is based on MVC (Model-View-Controller) architecture. A Framework provides structure and starting point for creating your application. It helps to provide an amazing developer experience while providing powerful features through dependency.

MVC based framework mainly divides the whole application into three components:

- **Model:** It interacts with the database.
- **View:** User Interface. It contains everything which a user can see on the screen.
- **Controller:** It helps to connect Model and View and contains all the business logic.



MVC module

The MVC module type implements the model-view-controller pattern, which separates an application into three main components: Models implement the domain logic and often store and retrieve data from the database. Views render the module's user interface (UI).

Advantages of Using MVC framework:

- Organizing large-scale web application projects.
- Easier to perform Modification.
- Modification in any part won't affect any other part of the code.
- Helps in a faster development process.
- Helps for Asynchronous Method Invocation.

❖ Explain routing

Routing is **the process of path selection in any network**. A computer network is made of many machines, called nodes, and paths or links that connect those nodes. Communication between two nodes in an interconnected network can take place through many different paths.

Routing is a process that is performed by layer 3 (or network layer) devices in order to deliver the packet by choosing an optimal path from one network to another.

types of routing

1. Static routing –

Static routing is a process in which we have to manually add routes to the routing table.

Advantages –

- No routing overhead for router CPU which means a cheaper router can be used to do routing.
- It adds security because an only administrator can allow routing to particular networks only.
- No bandwidth usage between routers.

Disadvantage –

- For a large network, it is a hectic task for administrators to manually add each route for the network in the routing table on each router.

- The administrator should have good knowledge of the topology. If a new administrator comes, then he has to manually add each route so he should have very good knowledge of the routes of the topology.

2. Default Routing –

This is the method where the router is configured to send all packets towards a single router (next hop). It doesn't matter to which network the packet belongs, it is forwarded out to the router which is configured for default routing. It is generally used with stub routers. A stub router is a router that has only one route to reach all other networks.

3. Dynamic Routing –

Dynamic routing makes automatic adjustments of the routes according to the current state of the route in the routing table. Dynamic routing uses protocols to discover network destinations and the routes to reach them. [RIP](#) and [OSPF](#) are the best examples of dynamic routing protocols. Automatic adjustments will be made to reach the network destination if one route goes down.

features:

1. The routers should have the same dynamic protocol running in order to exchange routes.
2. When a router finds a change in the topology then the router advertises it to all other routers.

Advantages –

- Easy to configure.
- More effective at selecting the best route to a destination remote network and also for discovering remote network.

Disadvantage –

- Consumes more bandwidth for communicating with other neighbors.
- Less secure than static routing.

Static routing vs dynamic routing

Static routing and dynamic routing are two methods used to determine how to send a packet toward its destination.

Static routes are configured in advance of any network communication. Dynamic routing, on the other hand, requires routers to exchange information with other routers to learn about paths through the network. Static and dynamic routing are used where appropriate, and some networks use both.

❖ **Explain migration and relationship**

➤ **migration in Laravel**

Migrations are like version control for your database, allowing a team to easily modify and share the application's database schema. Migrations are typically paired with Laravel's schema builder to easily build your application's database schema.

Laravel Migration is an essential feature in Laravel that allows you to create a table in your database. It allows you to modify and share the application's database schema. You can modify the table by adding a new column or deleting an existing column.

types of migration

Internal migration: moving within a state, country, or continent.

External migration: moving to a different state, country, or continent.

Emigration: leaving one place to move to another.

Immigration: moving into a new place.

Return migration: moving back to where you came from.

➤ **Relationship in Laravel**

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. For example, we may chain additional constraints on this posts relationship:

Types of relationship

One To One

A one-to-one relationship is a very basic relation. For example, a User model might be associated with one Phone. To define this relationship, we place a phone method on the User model. The phone method should call the has One method and return its result:

The first argument passed to the has One method is the name of the related model. Once the relationship is defined, we may retrieve the related record using Eloquent's dynamic properties. Dynamic properties allow you to access relationship methods as if they were properties defined on the model.

One To Many

A "one-to-many" relationship is used to define relationships where a single model owns any amount of other models. For example, a blog post may have an infinite number of comments. Like all other Eloquent relationships, one-to-many relationships are defined by placing a function on your Eloquent model.

One To Many (Inverse)

Now that we can access all of a post's comments, let's define a relationship to allow a comment to access its parent post. To define the inverse of a has Many relationship, define a relationship function on the child model which calls the belongs To method.

Many To Many

Many-to-many relations are slightly more complicated than has One and has Many relationships. An example of such a relationship is a user with many roles, where the roles are also shared by other users. For example, many users may have the role of "Admin". To define this relationship, three database tables are needed: users, roles, and role_user. The role_user table is derived from the alphabetical order of the related model names, and contains the user_id and role_id columns.

Many-to-many relationships are defined by writing a method that returns the result of the belongsToMany method. For example, let's define the roles method on our User model.

Has Many Through

The "has-many-through" relationship provides a convenient shortcut for accessing distant relations via an intermediate relation. For example, a Country model might have many Post models through an intermediate User model. In this example, you could easily gather all blog posts for a given country. Let's look at the tables required to define this relationship.

Polymorphic Relations

Table Structure

Polymorphic relations allow a model to belong to more than one other model on a single association. For example, imagine users of your application can "comment" both posts and videos. Using polymorphic relationships, you can use a single comments table for both of these scenarios. First, let's examine the table structure required to build this relationship.

Two important columns to note are the commentable_id and commentable_type columns on the comments table. The commentable_id column will contain the ID value of the post or video, while the commentable_type column will contain the class name of the owning model. The commentable_type column is how the ORM determines which "type" of owning model to return when accessing the commentable relation.

Many To Many Polymorphic Relations

Table Structure

In addition to traditional polymorphic relations, you may also define "many-to-many" polymorphic relations. For example, a blog Post and Video model could share a polymorphic relation to a Tag model. Using a many-to-many polymorphic relation allows you to have a single list of unique tags that are shared across blog posts and videos. First, let's examine the table structure.

❖ Explain blade template engine

Blade is the simple, yet powerful templating engine that is included with Laravel. Unlike some PHP templating engines, Blade does not restrict you from using plain PHP code in your templates.

Laravel Blade template engine enables the developer to produce HTML based sleek designs and themes. All views in Laravel are usually built in the blade template. Blade engine is fast in rendering views because it caches the view until they are modified.

purpose of Laravel Blade template engine

Laravel Blade template engine enables the developer to produce HTML based sleek designs and themes. All views in Laravel are usually built in the blade template. Blade engine is fast in rendering views because it caches the view until they are modified. All the files in resources/views have the extension.

❖ Directives

PHP directives are the configuration settings used to control various behaviours of PHP functions on your site. This can include for example; the memory limit assigned for PHP, the maximum amount of time before a PHP process will time out and the maximum file-size that could be uploaded via a PHP script.

Blade directives are shortcut codes for the implementation of basic PHP structure control, such as loop and conditional statements. It makes your code snippets clean and easy to understand. Note: These directives are basically only used in the blade template, so don't try to use them on your controllers.

How to Create Custom Blade Directive in Laravel?

- 1. Step 1: Create Custom Blade Directive. in app service provide file you have to declare custom blade directive. ...**
- 2. Step 2: Create Route. now we will create one route for example how to use custom directives in laravel blade. ...**
- 3. Step 3: Create Blade File**

Reference

- ✓ <http://www.educative.io>
- ✓ [http://www.itsoiution](http://www.itsoiutionstuff.com)stuff.com
- ✓ <http://www.laravel.com>
- ✓ wikipedia