

DEBRE BERHAN UNIVERSITY



COLLEGE OF COMPUTING

DEPARTEMENT OF COMPUTER SCINCE

Selected Topics in Computer Science Individual Assignment

NAME

EMEBET BESHAWRED

ID No:

DBUE /777/11

Submitted to: Girmachew

Submission date:20/05/15E.C

1. Explain MVC of Laravel

- ✓ Laravel is a free and open-source web PHP framework, which is based on **MVC (Model-View-Controller) architecture**.
- ✓ Laravel Framework is used to develop complex Web Applications. Laravel can help a developer to develop a secure web application. It is a Scalable framework and it also has a wide Community across the world.
- ✓ Basically, laravel is a structure which provides the capability to the user to create a program. It is a PHP framework, help to figure out the shape of your software and connect it with different applications program interface (APIs).

How laravel use MVC?

To understand how laravel use MVC first we have to understand the MVC. So basically PHP MVC is a design pattern that separates the data and business logic from the presentation. MVC stands for the Model View & Controller.

- ❖ **Model:** - Model is basically the collection of application data and business logic. It can be used to perform many operations like data validations, process the data and store data to the database. Data can be any kind like and static file, database, XML data or any other kind of valid resources.
- ❖ **View:** -The view is just for presenting the data to the user. This is the bridge for the interaction of users to our server. This is basically written in the HTML form.
- ❖ **Controller:** - The controller is the part which deals with the user's request, it accepts the user request and response as per the user requested. For example, if the user requested through URL like /index.php?Products=list than the controller will load the list of product data from the model and output to the user.

Now come to the point i.e how laravel using MVC. In laravel MVC used the following manners

1. Create a project in laravel.

- ✓ `laravel new product-store`

2. Create a model by using the following command.

- ✓ `php artisan make:model Product`

3. Connect the database using .env file then create a migration for storing the list of products.

- ✓ `php artisan make:migration create_products_table`

4. Now create a controller for handling the request.

- ✓ `php artisan make:controller ProductController -r`

5. Now create a route to get the request and send it to the controller

6. Now create a blade template in view by which the user would request

2. Explain Routing

- **Routing** is the process of selecting a path for traffic in a network or between or across multiple networks. Broadly, routing is performed in many types of networks, including circuit-switched networks, such as the public switched telephone network (PSTN), and computer networks, such as the Internet.
- In packet switching networks, routing is the higher-level decision making that directs network packets from their source toward their destination through intermediate network nodes by specific packet forwarding mechanisms. Packet forwarding is the transit of network packets from one network interface to another. Intermediate nodes are typically network hardware devices such as routers, gateways, firewalls, or switches. General-purpose computers also forward packets and perform routing, although they have no specially optimized hardware for the task.
- The routing process usually directs forwarding on the basis of routing tables. Routing tables maintain a record of the routes to various network destinations. Routing tables may be specified by an administrator, learned by observing network traffic or built with the assistance of routing protocols.
- Routing, in a narrower sense of the term, often refers to IP routing and is contrasted with bridging. IP routing assumes that network addresses are structured and that similar addresses imply proximity within the network. Structured addresses allow a single routing table entry to represent the route to a group of devices. In large networks, structured addressing (routing, in the narrow sense) outperforms unstructured addressing (bridging). Routing has become the dominant form of addressing on the Internet. Bridging is still widely used within local area networks.

3. Explain Migration and Relationships

- ✚ Migrations are like version control for database, allowing a team to define and share the application's database schema definition. If we have ever had to tell a teammate to manually add a column to their local database schema after pulling in our changes from source control, we've faced the problem that database migrations solve.
- ✚ The `Laravel Schema facade` provides database agnostic support for creating and manipulating tables across all of Laravel's supported database systems. Typically, **migrations** will use this facade to create and modify database tables and columns.
- ✚ The migration will be placed in our `app/database/migrations` folder, and will contain a timestamp which allows the framework to determine the order of the migrations.
- ✚ We may also specify a `--path` option when creating the migration. The path should be relative to the root directory of our installation:
- ✚ Some migration operations are destructive, meaning they may cause you to lose data. In order to protect you from running these commands against your production database, you will be prompted for confirmation before these commands are executed. To force the commands to run without a prompt, use the `--force` flag:
- A relationship, in the context of databases, is **a situation that exists between two relational database tables when one table has a foreign key that references the primary key of the other table**. Relationships allow relational databases to split and store data in different tables, while linking disparate data items.

- For example, in a bank database a CUSTOMER_MASTER table stores customer data with a primary key column named CUSTOMER_ID; it also stores customer data in an ACCOUNTS_MASTER table, which holds information about various bank accounts and associated customers. To link these two tables and determine customer and bank account information, a corresponding CUSTOMER_ID column must be inserted in the ACCOUNTS_MASTER table, referencing existing customer IDs from the CUSTOMER_MASTER table. In this case, the ACCOUNTS_MASTER table's CUSTOMER_ID column is a foreign key that references a column with the same name in the CUSTOMER_MASTER table. This is an example of a relationship between the two tables.
- Relationships are meaningful associations between tables that contain related information they're what make databases useful. Without some connection between tables in a database, you may as well be working with disparate spreadsheet files rather than a database system.

There are 3 different types of relations in the database:

- **One-to-one.** Relationship means that each record in Table A relates to one, and only one, record in Table B, and each record in Table B relates to one, and only one, record in Table A.
- **One-to-many.** Relationship means a record in Table A can relate to zero, one, or many records in Table B. Many records in Table B can relate to one record in Table A. The *potential* relationship is what's important; for a single record in Table A, there might be no related records in Table B, or there might be only one related record, but there *could* be many.
- **Many-to-many.** A **many-to-many** relationship indicates that multiple records in a table are linked to multiple records in another table

4. Explain 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.
- **Laravel** is an MVC framework while Blade is what enhances or handles the default View. **Blade** is a capable templating engine that's included with Laravel.
- In comparison to other PHP templating engines, Blade does not prevent you from utilizing plain PHP code in your code. In reality, all Blade layouts are compiled into plain PHP code and cached until they are altered. This means that Blade includes essentially zero lag to your application.
- Blade layout files utilize the `.blade.php` file extension and can be found within the resources/views folder.
- Data can be passed from the routes or controller depending on the nature of your logic or application. The controller and routes can return the view i.e., the `.blade.php`
- From the route we would return the greeting.blade.php like this:

5. Explain Directives

- PHP directives are the configuration settings used to control various behaviors 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.
- For sites hosted on LCN Web hosting packages, changes to PHP directives can be applied by uploading a file named .user.ini to the hosting web directory. Changes specified in this file will only apply to the individual site that the file is uploaded to.
- A number of configuration directives are supported in the php.ini file, which can be used to alter the behavior when PHP scripts run.
- Directives always begin with an @ symbol, followed by a unique name. They may be used at specified parts of the GraphQL schema.
- For example directive @upper Case may be used on field definitions to UPPERCASE the result.

Reference

- ✓ www.geeksforgeeks.org
- ✓ www.oodlestechnologies.com
- ✓ www.Laravel.com
- ✓ www.cloudways.com
- ✓ www.educative.io
- ✓ www.techopedia.com
- ✓ Wikipedia