****

**DEBRE BIRHAN UNIVERSITY**

**College of Computing**

**Department of Computer Science**

**Selected topics in computer science**

**Individual assignment**

**Name Genet Gebreyes**

**Id\_No DBUE/785/11**

1..Explain MVC of laravel

* **[Laravel](https://www.geeksforgeeks.org/laravel-installation-and-configuration/)** is a free and open-source web PHP framework, which is based on **MVC (Model-View-Controller) architecture.**
* AFramework provides structure and starting point for creating your application. It helps to provide an amazing developer experience while providing powerful features through dependency.

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.

Laravel is a Full Stack Framework, Which helps a developer create **Full Stack Applications** with the help of **Laravel**.

**How to create your first Laravel project:**

composer create-project laravel/laravel app-name

**Features of Laravel:**

**1. Authentication:**It is a very important part of any web application, to authenticate the user in their system, we need to write so much of the code which consumes a lot of time. But here, Laravel makes it much easier for a developer. **The authentication feature**was introduced after **Laravel 5**, where we only need to configure the Model, View, and Controller to use Authentication in our Web Application.

**2. Dynamic Template:**Laravel provides an innovative template engine called “**Blade template**“**,** which allows developers to create a dynamic web page. The blade is a **powerful templating engine** in a Laravel framework that helps to make **Dynamic Template** in web Applications.

**3. Database Migrations:**It is a very useful feature provided by Laravel Framework, Where we can **easily share database schema without any extra effort**. Migrations are files that contain codeto create a database table or make changes in any tables, this files can be used by other team members to make changes in the database by just executing that migration file. It **doesn’t have SQL code** it just **contains PHP code**which makes it much easier for any PHP developer to make changes in the Database.

**4. MVC Architecture:**It divides any application code into 3 parts, which **makes code more maintainable**and makes modification easier which makes the development process much faster.

**5. Unit Testing:**Testing is the main part of any application development cycle to check all cases. Laravel provides the feature of **Unit Testing,**it checks and makes sure that new updates in the code won’t affect another part of the code. It runs several test cases to check that the changes in the code won’t affect another part of the code. A Laravel developer can also write their own test cases.

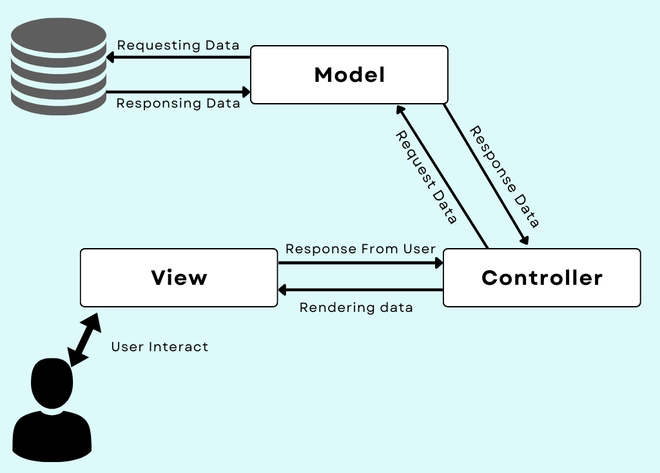
**6. Security:**Web Application Security is also considered to be important while the development cycle. But Laravel provides some feature that makes the application more secure. Laravel provides the feature of the **“Bcrypt Hashing Algorithm”**which generated hashed passwords and stores them in a Database which makes it almost impossible to resolve passwords.

**7. Artisan:**It is a built-in feature for command-line **Artisan**. This command-line feature can be very useful to developers. Artisan tool can be used to create skeleton code, database schema, and migration files, and also used to execute those migration files, so it makes it easy to manage a database schema of the system.

To understand Laravel more, we need to learn about **MVC (Model-View-Controller)** structure.

**Overview of MVC:**MVCbased 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. It is also known as the **“Heart of the application in MVC”.**



In the above image, we can understand how the MVC framework interacts with the **User**and the **Database.**

**Model:**This component of the MVC framework handles data used in your application. It helps to retrieve the data from the **database** and then perform some operation that your application is supposed to perform then it stores that data back in the database.

In simple words, we can say that Model is responsible for managing data that is passed between the database and the User Interface (**View**).

**View:**This component is User Interface, which defines the template which is sent as a response to the browser. This View components contain the part of code which helps to display data to the  ***User Interface***on the user’s browser. **For example**, we can say the buttons, textbox, dropdown menu, and many more such widgets on the browser screen are the part of View Component.

**Controller:**This controller component helps to interact with the model component to fetch data from the database and then pass that data to the view component to get the desired output on the user’s browser screen. Same way, when the user enters some data the controller fetches that data and then performs some operation or just inserts that data into the database with the use of the model components.

**Why MVC:**It is mainly used to separate Application code into **user interfaces, data, and controlling logic.** It can benefit the developer to easily maintain the code which can help to make a development process much smoother.

**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.

2. Explain Routing

Routing

* A Router is a process of selecting path along which the data can be transferred from source to the destination. Routing is performed by a special device known as a router.
* A Router works at the network layer in the OSI model and internet layer in TCP/IP model
* A router is a networking device that forwards the packet based on the information available in the packet header and forwarding table.
* The routing algorithms are used for routing the packets. The routing algorithm is nothing but a software responsible for deciding the optimal path through which packet can be transmitted.
* The routing protocols use the metric to determine the best path for the packet delivery. The metric is the standard of measurement such as hop count, bandwidth, delay, current load on the path, etc. used by the routing algorithm to determine the optimal path to the destination.
* The routing algorithm initializes and maintains the routing table for the process of path determination.

Routing Metrics and Costs

Routing metrics and costs are used for determining the best route to the destination. The factors used by the protocols to determine the shortest path, these factors are known as a metric.

Metrics are the network variables used to determine the best route to the destination. For some protocols use the static metrics means that their value cannot be changed and for some other routing protocols use the dynamic metrics means that their value can be assigned by the system administrator.

**The most common metric values are given below:**

* **Hop count:** Hop count is defined as a metric that specifies the number of passes through internetworking devices such as a router, a packet must travel in a route to move from source to the destination. If the routing protocol considers the hop as a primary metric value, then the path with the least hop count will be considered as the best path to move from source to the destination.
* **Delay:** It is a time taken by the router to process, queue and transmit a datagram to an interface. The protocols use this metric to determine the delay values for all the links along the path end-to-end. The path having the lowest delay value will be considered as the best path.
* **Bandwidth:** The capacity of the link is known as a bandwidth of the link. The bandwidth is measured in terms of bits per second. The link that has a higher transfer rate like gigabit is preferred over the link that has the lower capacity like 56 kb. The protocol will determine the bandwidth capacity for all the links along the path, and the overall higher bandwidth will be considered as the best route.
* **Load:** Load refers to the degree to which the network resource such as a router or network link is busy. A Load can be calculated in a variety of ways such as CPU utilization, packets processed per second. If the traffic increases, then the load value will also be increased. The load value changes with respect to the change in the traffic.
* **Reliability:** Reliability is a metric factor may be composed of a fixed value. It depends on the network links, and its value is measured dynamically. Some networks go down more often than others. After network failure, some network links repaired more easily than other network links. Any reliability factor can be considered for the assignment of reliability ratings, which are generally numeric values assigned by the system administrator.

Types of Routing

Routing can be classified into three categories:

* Static Routing
* Default Routing
* Dynamic Routing

Static Routing

* Static Routing is also known as Nonadaptive Routing.
* It is a technique in which the administrator manually adds the routes in a routing table.
* A Router can send the packets for the destination along the route defined by the administrator.
* In this technique, routing decisions are not made based on the condition or topology of the networks

Advantages Of Static Routing

Following are the advantages of Static Routing:

* **No Overhead:** It has ho overhead on the CPU usage of the router. Therefore, the cheaper router can be used to obtain static routing.
* **Bandwidth:** It has not bandwidth usage between the routers.
* **Security:** It provides security as the system administrator is allowed only to have control over the routing to a particular network.

Disadvantages of Static Routing:

Following are the disadvantages of Static Routing:

* For a large network, it becomes a very difficult task to add each route manually to the routing table.
* The system administrator should have a good knowledge of a topology as he has to add each route manually.

Default Routing

* Default Routing is a technique in which a router is configured to send all the packets to the same hop device, and it doesn't matter whether it belongs to a particular network or not. A Packet is transmitted to the device for which it is configured in default routing.
* Default Routing is used when networks deal with the single exit point.
* It is also useful when the bulk of transmission networks have to transmit the data to the same hp device.
* When a specific route is mentioned in the routing table, the router will choose the specific route rather than the default route. The default route is chosen only when a specific route is not mentioned in the routing table.

Dynamic Routing

* It is also known as Adaptive Routing.
* It is a technique in which a router adds a new route in the routing table for each packet in response to the changes in the condition or topology of the network.
* Dynamic protocols are used to discover the new routes to reach the destination.
* In Dynamic Routing, RIP and OSPF are the protocols used to discover the new routes.
* If any route goes down, then the automatic adjustment will be made to reach the destination.

Advantages of Dynamic Routing:

* It is easier to configure.
* It is more effective in selecting the best route in response to the changes in the condition or topology.

Disadvantages of Dynamic Routing:

* It is more expensive in terms of CPU and bandwidth usage.
* It is less secure as compared to default and static routing

3.Explain migration and relationships

* Migration can be defined as the movement of people from one region to another region of a state or a country by settling temporarily or permanently, for better living conditions. Most of the migrations are due to economic conditions, business opportunities, education, and employment.
* In the twentieth century semi-skilled Indians, for example, craftsmen, dealers, and assembly line laborers relocated to adjoining nations like Thailand, Malaysia, Singapore, Indonesia, Brunei, and African nations. In 2005, 191 million individuals were considered living external of the nation of their country. On the off chance that the migrants resided in a similar spot, the international migrants would frame the fifth most populous country on the planet. Education became the most important in the world; many students are migrating for higher education. Getting a degree from the World’s top Universities has the highest recognition and value, and the chance of getting the best, highest-paid jobs. A large percentage of numbers increased, migrating for foreign studies. In today’s world, highly educated professionals like doctors, engineers, software engineers, management consultants, financial experts, media persons, etc., are migrating to other countries like the US, Canada, UK, Australia, New Zealand, Germany, etc., for better opportunities, and they are the highest paid.

**Causes of Migration**

* In search of better economic conditions, employment, business opportunities, and education.
* Migration to the towns in order to secure better living conditions, especially better schooling for their children.
* To escape poor climate conditions such as drought, and natural disasters.
* As punishment for crimes committed, the criminals and banned from a country due to their work against government laws.
* Migration as a result of enslavement.

**Types of Migration**

**In the view of the origin and destination of migrants**

**Internal migration**

The movement of individuals within the geographical boundaries of a country or a state. Internal migration alludes to a change of residence within a state, region, city, or municipality.

**Internal migration is additionally classified into four types.**

**Rural to Urban Migration** The movement of population from rural areas to the nearby towns and cities mainly in search of better livelihood and standard of living, i.e., employment, education, and recreation facilities.

**Rural to Rural Migration** Mostly agricultural workers, because of marriages, and sometimes looking for land for cultivation.

**Urban to Urban Migration**The relocation from one metropolitan community to the next looking for more significant compensation and another market for business potential opens doors.

**Urban to Rural Migration** The movement from urban areas or cities to rural areas to get freed off the urban problems like air pollution, overcrowding, noise pollution, and returning to their native place after retirement from jobs.

**International migration**

This alludes to a change of residence over national boundaries. A global migrant is somebody who moves to an alternate country.

**International migrants are additionally classified as**

1. Legal immigrants are the individuals who moved with the lawful authority of the beneficiary country.
2. Unlawful immigrants are individuals who moved without lawful authorization.
3. Refugees are those who crossed a worldwide limit to get away from abuse.

**In view of the time of stay of migrants in the region**

* **Short-term migration:** The migrants stay outside just for a brief length prior to getting back to their place. Examples: Tourists, business trips.
* **Long-term migration:** The migrants stay outside, essentially, for a couple of years. Examples: Companies send their employees for projects and students for education.
* **Seasonal migration:** Usually, a group of people moves from their local spots during a specific season and return after the finish of that season. Agriculture-based labor is an example of seasonal migration.

**In view of the readiness of the individuals for migration**

**Voluntary migration:** If the movement happens on migrant choice, drive and want to reside in a good place. Example: Business company CEOs.

**Involuntary migration:** If the relocation happens against the desire of migrants, it is called involuntary migration. Example: Refuges.

**Positive side of migration**

* Migrants send money to their homes, helping in the growth of the economy of the region.
* International Indian migrants are one of the major sources of foreign exchange in India.
* States like Punjab, Kerala, and Tamil Nadu receive a huge amount of money from their international migrants.
* Green revolution in Punjab, and Haryana result because of people’s migration from Uttar Pradesh, and Bihar.

**Negative side consequences**

* Overcrowding is one of the major negative impacts on the metropolitan cities of India due to migration. Resulting in many negative side effects.
* The development of slums in industrially developed states is another major negative consequence.
* Shortage of skilled labor because most migrate to urban areas.
* Under-development of rural regions due to lack of skilled people and economy of the region.
* Imbalances in sex ratio due to large male migration.
* Women in the rural areas will have more vulnerability, leaving with extra pressure when men are away from home.

4.Explain Blade template engine

* Laravel is one of the best PHP framework which is highly acknowledged for its inbuilt lightweight templates that help you create amazing layouts using dynamic content seeding.
* Pre-requisites: I assume you are familiar with installing and running Laravel application and are familiar with creating Routes.The templates of the Laravel framework are innovatively designed to create a simple layout with distinctive sections.

When compared to other templating engines, Blade is unique in the following ways

We can use plain PHP code in views. The blade views thus designed, are compiled and cached until they are modified.

Remember: you can clear the compiled and cached view files by running this command in the terminal.

php artisan view:clear

Blade view files use the .blade.php file extension and are typically stored in the resources/views directory.

# Inheritance

Two of the primary benefits of using Blade are template inheritance and sections. We can define a blade page as a combination of layout and sections.

* Since most of the general web applications will have the same layout across the web pages. In this example, we will examine the “master layout” with sections like “Top nav”, “Sidenav” and the “body content”.

I'm using MaterializeCss in this example as it has some useful prebuilt components like sidenav.

Lets start:

Blade view files use the .blade.php file extension and are typically stored in the resources/views directory.

Let us create the following blade files in your project’s resource/views directory.

master.blade.php

topnav.blade.php

sidenav.blade.php

home.blade.php

In master.blade.php, let us initiate the html template with MaterializeCss CDN’s.

<!DOCTYPE html>

<html>

<head>

<!--Import materialize.css-->

<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">

<!--style sheets section-->

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

</head>

<body>

@include('topnav') <!-- include topnav.blade.php-->

@include('sidenav') <!-- include sidenav.blade.php-->

<div class="container">

@yield('body-content')

</div>

<!--JavaScript of body for optimized loading-->

<!-- Jquery cdn-->

<script src="https://code.jquery.com/jquery-3.4.1.min.js"

integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo="

crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>

</script>

<!-- also get script from the child view -->

<script>

@yield('page-script') <!-- to get script from page -->

</script>

</body>

</html>

When defining a child view, use the Blade @extends directive to specify which layout the child view should "inherit". Views which extend a Blade layout may inject content into the layout's sections using @section directives. Remember, as seen in the example above, the contents of these sections will be displayed in the layout using @yield:

@include is the directive used to include one blade file into another. Here we are including both the topnav.blade.php and sidenav.php inside the master.blade.php with the below lines of code

5.Directive (programming)

* This article needs additional citations for verification. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed.

Find sources: "Directive" programming – news · newspapers · books · scholar · JSTOR (December 2013) (Learn how and when to remove this template message)

In computer programming, a directive or pragma (from "pragmatic") is a language construct that specifies how a compiler (or other translator) should process its input. Directives are not[disputed – discuss] part of the grammar of a programming language, and may vary from compiler to compiler. They can be processed by a preprocessor to specify compiler behavior, or function as a form of in-band parameterization.

In some cases directives specify global behavior, while in other cases they only affect a local section, such as a block of programming code. In some cases, such as some C programs, directives are optional compiler hints, and may be ignored, but normally they are prescriptive, and must be followed. However, a directive does not perform any action in the language itself, but rather only a change in the behavior of the compiler.

This term could be used to refer to proprietary third party tags and commands (or markup) embedded in code that result in additional executable processing that extend the existing compiler, assembler and language constructs present in the development environment. The term "directive" is also applied in a variety of ways that are similar to the term command."cc" ,c "ccc", "cc"

Directive (programming)

PrintPrintReading time 8:0

In computer programming, a directive or pragma (from "pragmatic") is a language construct that specifies how a compiler (or other translator) should process its input. Directives are not part of the grammar of a programming language, and may vary from compiler to compiler. They can be processed by a preprocessor to specify compiler behavior, or function as a form of in-band parameterization.

In some cases directives specify global behavior, while in other cases they only affect a local section, such as a block of programming code. In some cases, such as some C programs, directives are optional compiler hints, and may be ignored, but normally they are prescriptive, and must be followed. However, a directive does not perform any action in the language itself, but rather only a change in the behavior of the compiler.

This term could be used to refer to proprietary third party tags and commands (or markup) embedded in code that result in additional executable processing that extend the existing compiler, assembler and language constructs present in the development environment. The term "directive" is also applied in a variety of ways that are similar to the term command