SEFP ASSIGNMENT-5
CAKEPHP(Web Framework)

CAKEPHP:-

CakePHP is a free,open-source,rapid development framework for php.It is a foundational structure for programmers to create web applications.CakePHP takes the monotony out of web development. It provides you with all the tools you need to get started coding and what you need to get done: the logic specific to your application. Instead of reinventing the wheel every time you begin a new project, check out a copy of CakePHP and get started with the logic of your application.
CakePHP has an active developer team and community, bringing great value to the project. In addition to keeping you from wheel-reinventing, using CakePHP means your application's core is well tested and is being constantly improved.

FEATURES:-

→ Baking: Code Generation.
→ Convention Over Configuration.
→ Active Record.
→ Secure By Default.
→ Access Control List.
→ II 8n.
→ Email.
→ Caching.
→ Authentication.
→ Pagination.

CAKEPHP ARCHITECTURE:-

Cakephp follows the **MVC** designed architecture pattern.

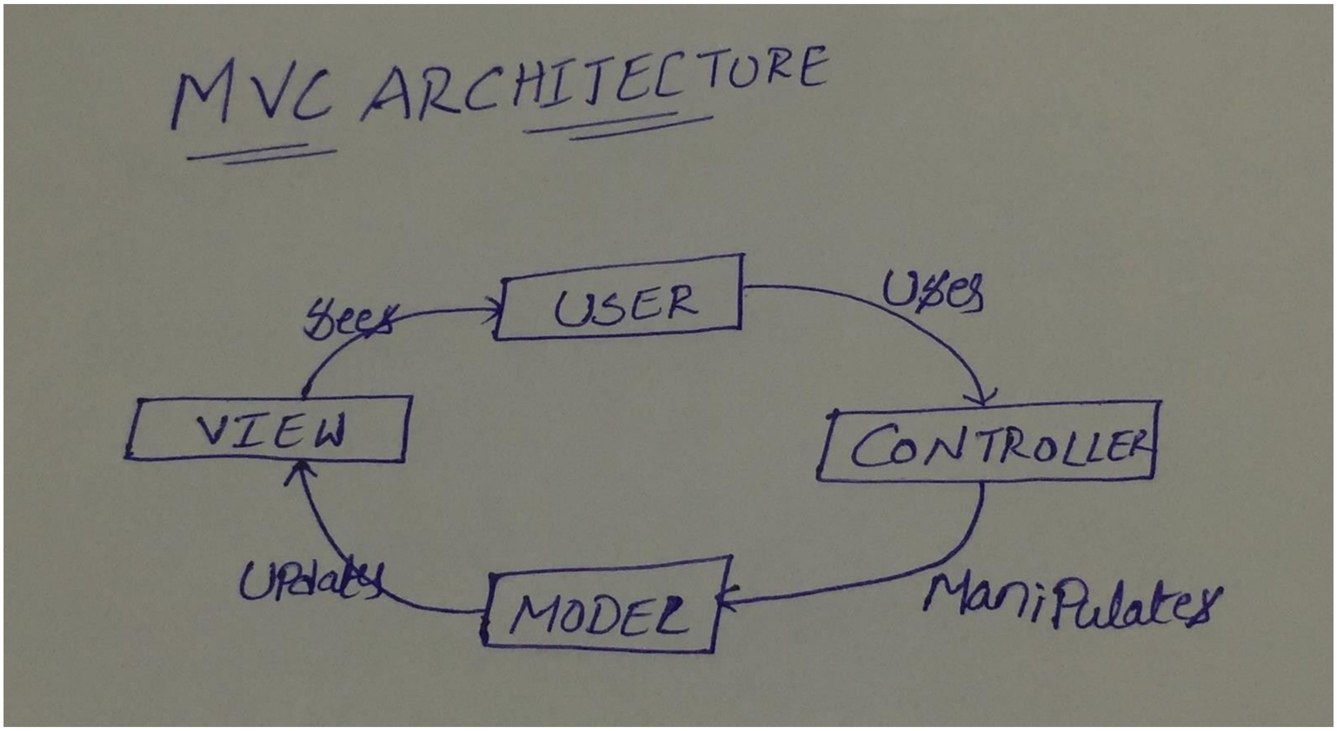
→ MVC ARCHITECTURE:-

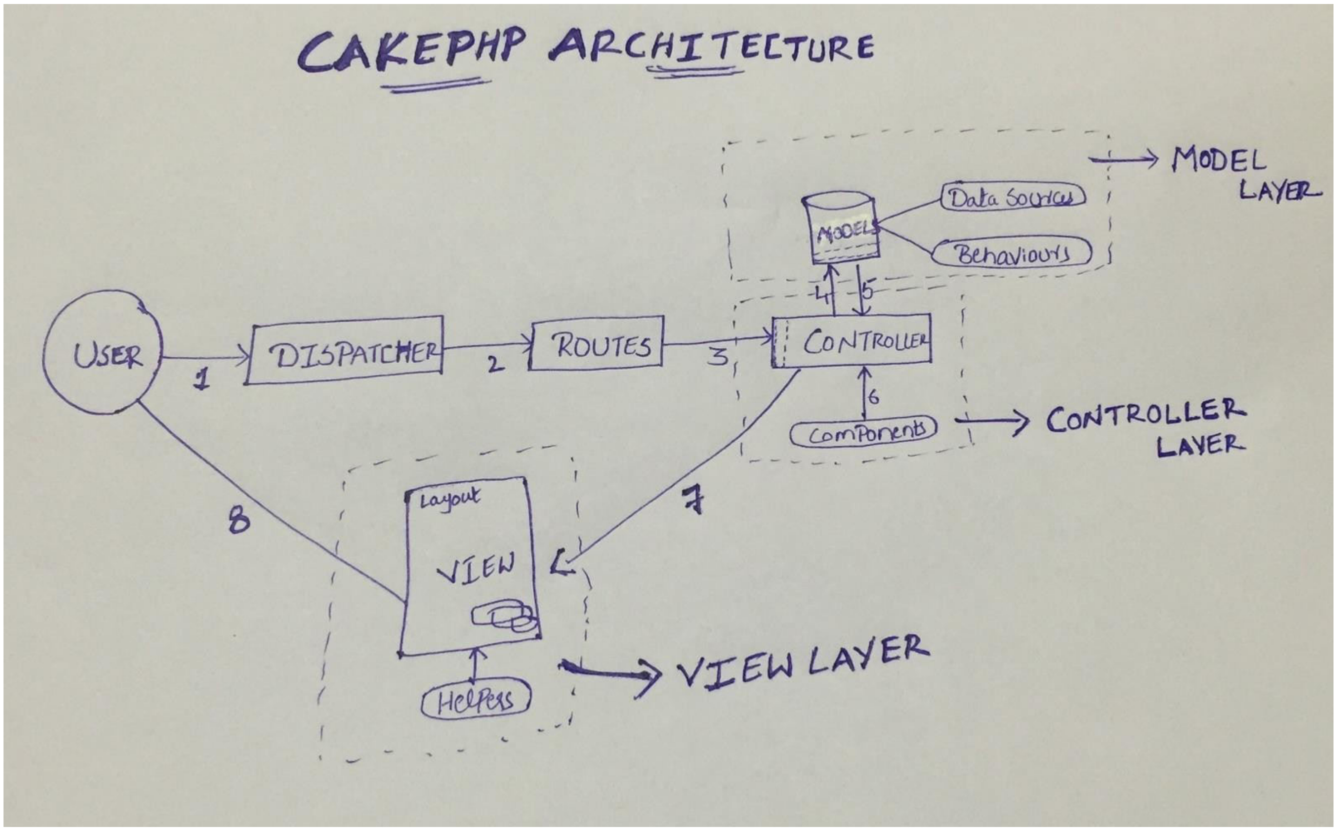
MVC stands for Model View Controller.It's an architecture pattern that describes a way to structure our application and explains the responsibilites & interactions for each part in that structure.

→ **MODEL:-** Wrapping up the data and logics of CakePHP.

→ **VIEW:-** Output to the client.

→ **CONTROLLER:-** Manupulating the data from model and generate the view.


MVC ARCHITECTURE
Sees
USER
Uses
VIEW
CONTROLLER
Updates
MODEL
Manipulates


CAKEPHP ARCHITECTURE
MODEL LAYER
Data Sources
MODELS
Behaviours
4
5
USER
1
DISPATCHER
2
ROUTES
3
CONTROLLER
6
Components
CONTROLLER LAYER
8
7
Layout
VIEW
Helpers
VIEW LAYER

ABOUT ARCHITECTURE:-

The typical CakePHP request cycle starts with a user requesting a page or resource in the application. This request is first processed by a dispatcher which will select the correct controller object to handle it.
Once the request arrives at the controller, it will communicate with the Model layer to process any data-fetching or -saving operation that might be needed. After this communication is over, the controller will proceed to delegate to the correct view object the task of generating output resulting from the data provided by the model.
Finally, when this output is generated, it is immediately rendered to the user.
Almost every request to your application will follow this basic pattern. We'll add some details later on which are specific to CakePHP, so keep this in mind as we proceed.

Programming using **MVC** separates the application into three main parts:-

MODEL LAYER:-

It plays the role of back-end Web Development(Database) for storing and retrieving the data.

The Model layer represents the part of the application that implements the business logic. It is responsible for retrieving data and converting it into meaningful concepts for the application. This includes processing, validating, associating or other tasks related to handling data.

At a first glance, Model objects can be looked at as the first layer of interaction with any database you might be using for the application. But in general they stand for the major concepts around which you implement the application.

In the case of a social network, the Model layer would take care of tasks such as saving the user data, saving friends' associations, storing and retrieving user photos, finding suggestions for new friends, etc. The model objects can be thought as "Friend", "User", "Comment", or "Photo".

VIEW LAYER:-

It plays the role of front-end Web Development(Html,Css and Javascript) for displaying the webpages.

The View renders a presentation of modeled data. Being separated from the Model objects, it is responsible for using the information it has available to produce any presentational interface your application might need.

For example, as the Model layer returns a set of data, the view would use it to render a HTML page containing it, or a XML formatted result for others to consume.

The View layer is not only limited to HTML or text representation of the data. It can be used to deliver a wide variety of formats depending on your needs, such as videos, music, documents and any other format you can think of.

CONTROLLER LAYER:-

In this layer,By using PHP script we will control the Application(Server Side Scripting).

The Controller layer handles requests from users. It is responsible for rendering a response with the aid of both the Model and the View layer.

A controller can be seen as a manager that ensures that all resources needed for completing a task are delegated to the correct workers. It waits for petitions from clients, checks their validity according to authentication or authorization rules, delegates data fetching or processing to the model, selects the type of presentational data that the clients are accepting, and finally delegates the rendering process to the View layer.

BENEFITS:-

→ It's a true software design pattern that turns an application into a maintainable, modular, rapidly developed package.

→ Crafting application tasks into separate models, views, and controllers makes your application very light on its feet.

→ New features are easily added, and new faces on old features are a snap. The modular and separate design also allows developers and designers to work simultaneously, including the ability to rapidly prototype.

→ Separation also allows developers to make changes in one part of the application without affecting the others.

→ If you’ve never built an application this way, it takes some time getting used to, but we’re confident that once you’ve built your first application using CakePHP, you won’t want to do it any other way.

CONCLUSION:-

In my view,here in this framework it consists of three parts..........but I want to propose the design in which this framework by two parts itself,where by combining the models and controllers in one part & views in another part.