INSTITUTE OF TECHNOLOGY,

NIRMA UNIVERSITY

2CSDE86 Application Development Framework

PRACTICAL 2

Name : Jaimik S. Chauhan

Roll No. : 20BCE503

Semester : 7th

Teacher's Signature :

Aim :

i) Comparative Study of various Web-Development Frameworks by considering various parameters for comparison.

a) **Frontend Frameworks** : Also called as Client Side framework. This is used for dealing with user interfaces. It’s the part where user interacts with. A website’s user experience is of paramount importance to any business. Although complex operations and functions may occur in the background, what users experience and see must be seamless. In order to do this, frontend frameworks are used to facilitate the development of interactive, user-centric websites.

b) **Backend Frameworks** : Backend Frameworks are nothing more than libraries of modules and tools that assist developers in creating website structures . It consists of the server, the database and the code that interacts with them. A server side framework ensures that the website functions smoothly in the background.

Following Parameters are included in Each framework : Model they follow (MVC OR MVT), Language Used, Frontend / Backend

1) Django

- Django is a **python**-based web framework that allows you to quickly create efficient web applications.
- Django Provides built-in features for everything including Django admin interface, default database – SQLlite3, etc.
- Django provides ready-made components to use and that too for rapid development.
- Django takes security seriously and helps developers to avoid many security mistakes.
- Django is Exceedingly Scalable.
- Django is **Backend + frontend** Framework.
- Follows **MVT** (Model-View-Template) model

2) Vue.js

- Vue.js is an open source MVC **Frontend JavaScript** framework.
- It optimizes re-rendering.
- The virtual DOM improves the performance of the application and the efficiency of DOM updates. Vue uses it to determine what parts of the DOM need to be re-rendered and which ones should be left intact.
- Supports two-way data binding.

- It’s Lightweight, fast to launch which results in seamless user experience.
- Follows **MVVM** (Model-View-ViewModel) Pattern

3) Laravel
    - Open Source **backend** framework written in **PHP**.
    - Follows **MVC** architecture, which makes developing and deploying web applications easy and quick.
    - Provides Lightweight templates in Laravel that can be used to create layouts and produce content.

4) Angular
    - Google Introduced AngularJS open-source **Frontend JavaScript** framework.
    - Angular reduces the build time by allowing developers to reuse components and even the architecture to simplify the development process.
    - Encourage reusability and improves application scalability.
    - Simplifies testing process
    - Angular CLI is regarded as one of the best CLI for building, scaffolding, and maintaining web applications.
    - **MVC** based framework

5) CakePHP
    - Open Source web framework.
    - Follows Model-view-controller (**MVC**) approach and written in **PHP**.
    - **Backend** Framework

6) Ember JS
    - Similar To Angular, It Provides two way data binding. It aims to satisfy the growing demand for contemporary technologies flawlessly.
    - **Front End** Framework
    - Follows **MVVM** Paradigm

7) ASP
    - Follows **MVC**
    - Compatible with **Javascript** based front-end frameworks.
    - Its an **Backend** Framework

8) NextJS
    - Its **React Framework** for Production.

- It gives you the best developer experience with all the features like hybrid static & server rendering, Typescript support, smart bundling, route pre-fetching, and more.
- Supports pages with dynamic routes.
- By default, Next.JS pre-renders every page.
- **Frontend Javascript** Framework

9) Flutter

- Open source framework by google for building multi platform applications from a single codebase.
- Fast, Multi Platform and take control of your codebase with automated testing, developer tooling and everything else you need to build production quality apps.
- **Frontend** framework based on **Dart**
- **MVC** Based Framework

10) ExpressJS

- Open-source, lightweight **back-end** framework for **Node.js** and is designed to build web applications, mobile applications and APIs.
- Its Caching potential, it can dramatically reduce a site's preserving a webpage's state using URLs.
- Provides a middleware system and several HTTP methods that can be used to build Node.JS apps and APIs quickly.
- **MVC** based Framework

11) Rails

- It's Open Source **Backend** Web development Framework based on **MVC**(Model-View-controller).
- It has powerful and robust library called the active record, which simplifies designing database queries.
- It equips developers with all essential tools to build a high-quality database access library, product AJAX library, and common tasks library.

12) Spring

- Open Source Framework.
- Its an application framework and inversion of control container for the **Java** Platform.
- Follows an **MVC** framework.
- **Backend** Framework

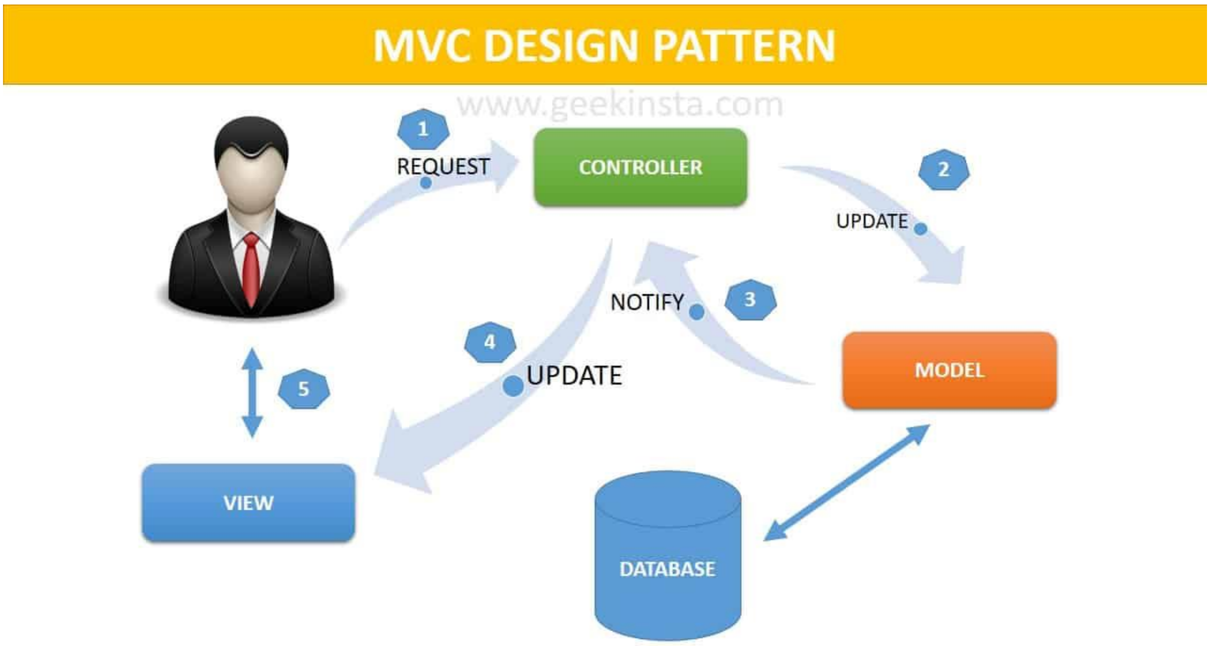
ii) Introduction to MVC and MVT architecture. Comparative study on MVC and MVT by considering various parameters for comparison

MVC (Model View Controller)

- It’s software design pattern that is used to implement user interfaces and gives emphasis on separating data representation from the components which interacts and process the data.

It has 3 components and each has specific approach :

- Model is the central component of this architecture and manages the data, logic as well as other constraints of the application.
- View deals with how the data will be displayed to the user and provides various data representation components.
- Controller manipulates the Model and renders the view by acting as a bridge between both of them.



Advantages:

- Makes it easy to develop large applications
- Easy for multiple developers to collaborate and work together.

Disadvantages:

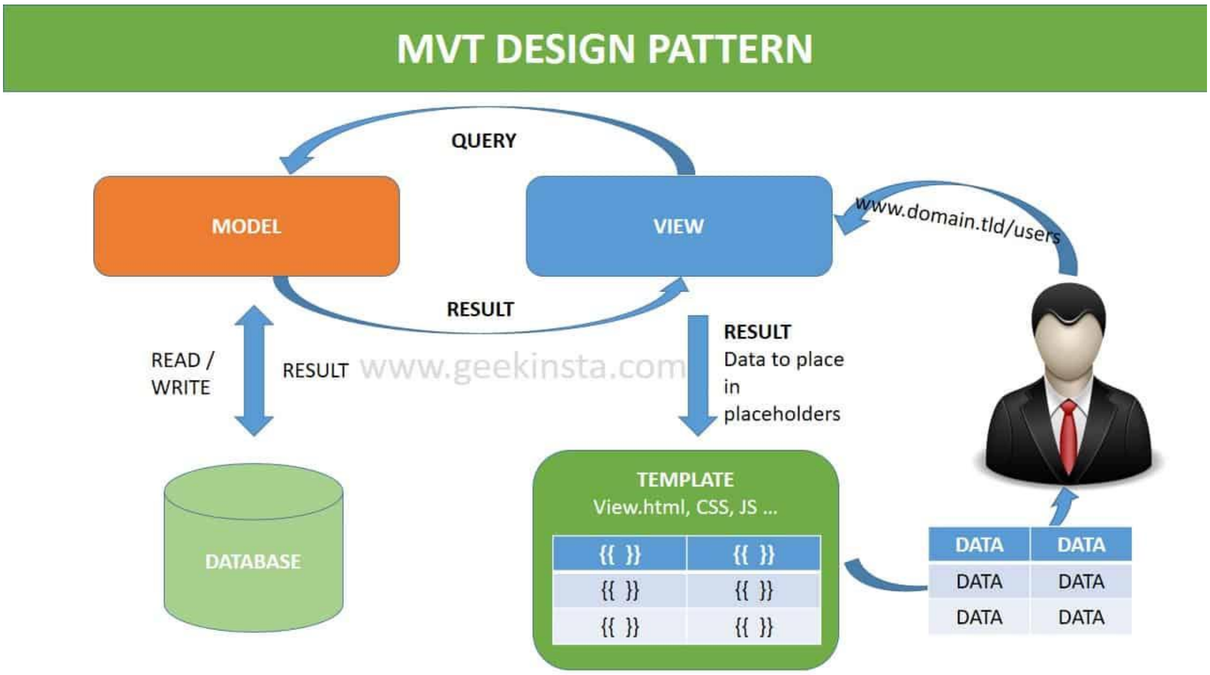
- View is controlled by Model and Controller
- Not suitable for small applications

MVT (Model View Template)

- Similar to MVC, but in contrast to MVC the controller part is taken care for us by the framework itself.

It has 3 components :

- Model similar to MVC acts as an interface for your data and is basically the logical structure behind the entire web application which is represented by a database such as MySql, PostgreSQL.
- View Executes the business logic and interacts with the model and renders the template. It accepts HTTP request and then return HTTP responses.
- Templates is the component which makes MVT different from MVC. Templates acts as the presentation layer and are basically HTML code that renders the data. Content in these files can be either static on dynamic.



Advantages:

- Less coupled.
- Suitable for small to large-scale applications.
- Easy to Modify.

Disadvantages:

- Sometimes, understanding the flow can be confusing
- Modification of models/views should be done carefully without affecting templates.

MVC V/S MVT

Model View Controller (MVC)	Model View Template (MVT)
MVC has controller that drives both model and view	MVT has views for receiving HTTP request and returning HTTP response
View tells how the user data will be presented	Templates are used in MVT for that purpose
Highly coupled	Loosely coupled
Modifications are difficult	Modifications are easy
Suitable for development of large applications but not for small applications	Suitable for both small and large applications
Flow is clearly defined thus easy to understand	Flow is sometimes harder to understand as compared to MVC
It doesn’t involve mapping of URLs	URL pattern mapping takes place
Ex: Spring MVC	Ex: Django