

# **MVC ARCHITECTURE AND IT'S APPLICATION**

Major project report submitted in partial fulfillment of the requirement for the  
degree of Bachelor of Technology

In

**Computer Science and Engineering**

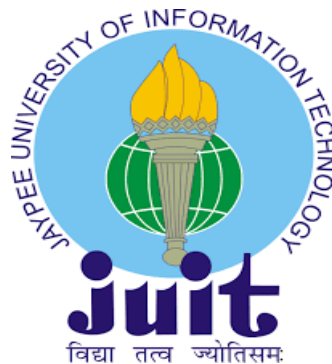
By

**PALAK RUSTAGI (181212)**

**UNDER THE SUPERVISION OF**

Dr Yugal Kumar

Associate Professor, Deptt. Of CSE & IT



Department of Computer Science & Engineering and Information Technology

**Jaypee University of Information Technology, Wahnaghat,  
173234, Himachal Pradesh, INDIA**

## Candidate's Declaration

I hereby declare that the work presented in this report entitled “ **MVC Architecture and it's Applications**” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2015 to December 2015 under the supervision of **Dr. Yugal Kumar**, Associate Professor, Computer science.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student signature)

Student Name:

Palak Rustagi

Roll no. :

181212

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(College Supervisor Signature)

(Company Supervisor Signature)

Supervisor Name:

Dr. [Yugal Kumar](#)

Designation:

Associate Professor

Department name :

Computer science

Dated:

Supervisor Name:

## ACKNOWLEDGEMENTS

Firstly, I express my heartiest thanks and gratefulness to Almighty God for His divine

blessing makes us possible to complete the project work successfully.

I am grateful and wish my profound indebtedness to Supervisor Dr Yugal Kumar, Associate Professor, Department of CSE Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of "Data Science" to carry out this Project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this Project.

I would like to express my heartiest gratitude to Dr Yugal Kumar, Department of CSE, for her kind help to finish my Project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

Palak Rustagi(181212)

## Table of Contents

<b>Chapter 1: INTRODUCTION</b>	<b>1</b>
<b>1.1 Introduction</b>	<b>1</b>
<b>1.2 Problem Statement</b>	<b>7</b>
<b>1.3 Objectives</b>	<b>8</b>
<b>1.4 Organization</b>	<b>10</b>
<b>Chapter 2: LITERATURE SURVEY</b>	<b>13</b>
<b>Chapter-3: SYSTEM DEVELOPMENT</b>	<b>16</b>
<b>Chapter 4: PERFORMANCE ANALYSIS</b>	<b>17</b>
<b>Chapter 5: CONCLUSION</b>	<b>32</b>
<b>5.1 Conclusion</b>	<b>32</b>
<b>5.2 Future Scopes</b>	<b>35</b>
<b>5.3 Application</b>	<b>37</b>
<b>Chapter 6: REFERENCES</b>	<b>41</b>

**LIST OF FIGURES**

<b>Image No.</b>	<b>Title</b>	<b>Page No.</b>
1.	How MVC works	3
2	Basic UI of MVC	3
3	Basic Framework Flow	8
4	About the Organization	10
5	Application of MVC pattern	17
6	Laravel	19
7	Styling the application	22
8	Comparison between MVP, MVC and MVVM	27
9	MVC vs MVP	29
10	MVP vs MVC vs MVVM	30
11	MVC Flow	33
12	Application using asp net architecture	36
13	Applications	37

## ABSTRACT

A request command is made to the Controller by the browser first. In order to deliver and receive data, the Controller interacts with the Model. The Controller communicates with the View to render the data. The View only job is to know about how information is delivered, not how it is presented in the end. It'll be a dynamic HTML file that renders data based on the Controller's input. Finally, the View will process all its final presentation to the Controller, which the Controller will present to the user. It's important to note that the View and the Model never communicate. They can only communicate with one another through the Controller. This refers to the application's logic as well as the user interface. MVC architecture can be used to develop a scalable, transparent, and portable web-based application. MVC architecture teaches us about parallel development since it divides an application's logic into three layers, allowing multiple developers to work on the same web-based application at the same time. Setting up the MVC architecture is usually expensive. Small MVC projects and non-MVC projects often take the same amount of time to process, however large MVC projects are typically handled faster than non-MVC projects

## CHAPTER-1

### INTRODUCTION

#### 1.1 INTRODUCTION

##### **MVC Architecture**

MVC is a software design that is usually used to create user interfaces, data, and control logic. It is focused on the distinction between business logic and software appearance. This "separation of concerns" provides better manpower utilisation and maintenance. Other MVC-based design patterns include MVVM (Model-View-Viewmodel), MVP (Model-View-Presenter), and MVW (Model-View-Web) (Model-View-Whatever).

The MVC software design pattern can be broken down into three parts:

- Data and business logic are handled by the model itself.
- Layout and presentation are handled by View.
- Controls how commands are routed to the model and view elements.

##### **The Model:-**

What information should be provided in the app is specified by the model. The model generally tells the basic look of it (so that the display can be altered as and when required) and the controller when the status of this data changes (if a separate logic to govern the updated view is required).

Returning to our shopping cart app, the model would specify what information should be included in the list items, such as item, price, and so on, as well as which list items are currently present.

## **The View:-**

The view specifies how the data in the app should be displayed. The view in our shopping list app specifies how the user sees the shopping cart and receives data from the model to present...

## **The Controller:-**

In response to user input conditions, the controller implements logic that will define the models view.

Input fields and buttons for adding and removing items, for example, may be present in our shopping list. Because these operations entail modifying the model, the data is transferred to the controller, which in turn modifies the model as needed before sending the modified data to the view.

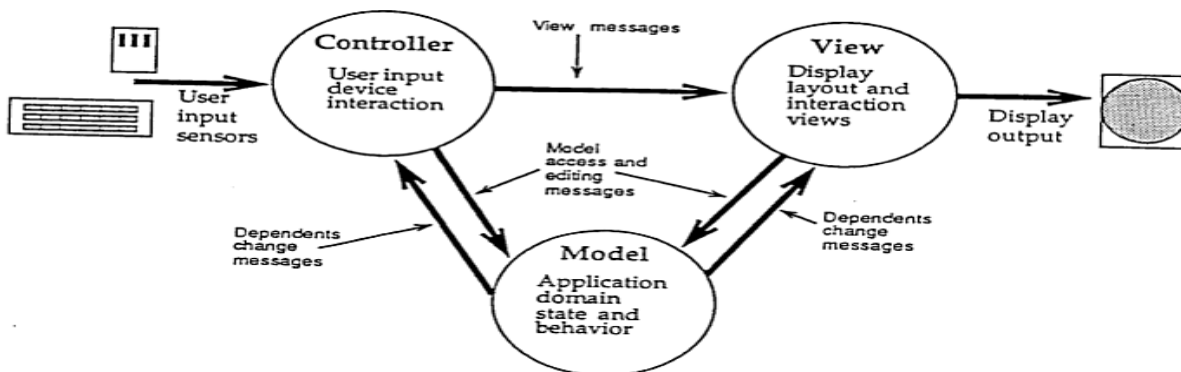
To modify the way the data is displayed, simply edit the view. Change the item order to alphabetical or from lowest to highest price, for example. The controller could handle it in this situation without needing to update the model.

## **MVC , on the web:**

Even if you've never consciously utilised this pattern as a web developer, you're definitely familiar with it. Your data model is almost certainly stored in a table of contents (be it a traditional server-side database like mysql, or a client-side solution such as indexeddb).

The control code for your app is almost probably written in html/js, and the user interface is almost certainly written in html/css whatever else you prefer. This appears to be similar to mvc, but mvc mandates that these components follow a stricter structure.





## How MVC Architecture works

A request command is made to the Controller by the browser first. In order to deliver and receive data, the Controller interacts with the Model.

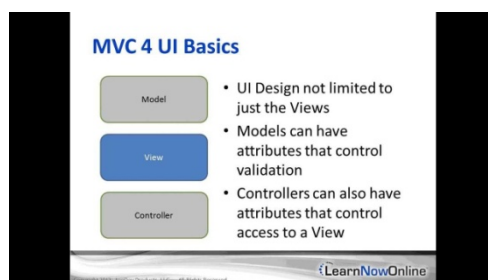
The Controller communicates with the View to render the data. The View only job is to know about how information is delivered, not how it is presented in the end. It'll be a dynamic HTML file that renders data based on the Controller's input.

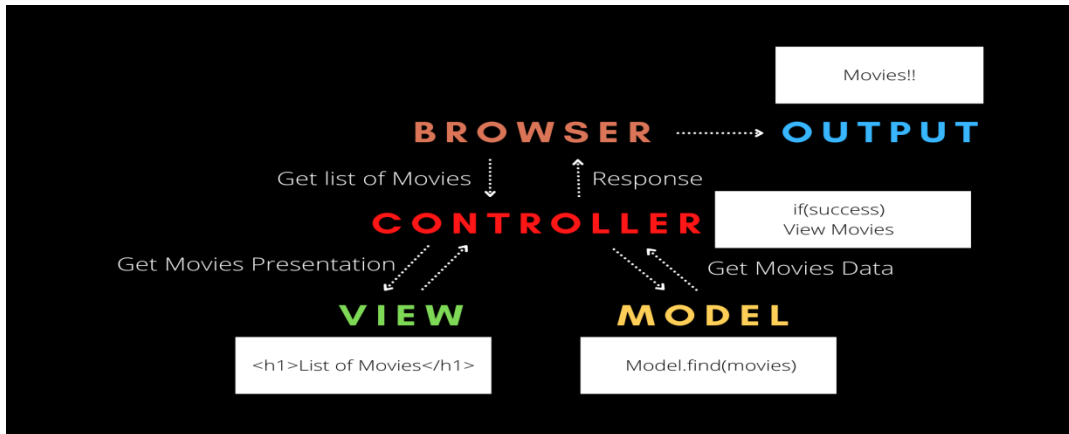
Finally, the View will process all its final presentation to the Controller, which the Controller will present to the user.

It's important to note that the View and the Model never communicate. They can only communicate with one another through the Controller.

This refers to the application's logic as well as the user interface.

## Basic UI of MVC:





Let's take a look at what's going on. First, a user inputs that they want a list of movies through a web browser or a mobile application.

- The browser will have to send a request to the Controller to get the list of movies.
- Next, the Controller will request the Model to find the list of movies from the main database.
- The Model then searches for this database and presents to the Controller a list of movies from the database.
- If the Controller receives a list of movies from the Model as per request then, it will command the View to display the list.
- The View will then receive the request and return to the Controller in HTML the rendered list of movies.
- Finally, the Controller will take that HTML and provide it to the user, resulting in the output of a list of Movies.

### Advantages of MVC:

1. Ease of Debugging: Because the application is separated into components, debugging one component without affecting the others is easier. Debugging the backend, for example, can be done without debugging the frontend.

2. Ease of Use: The user can concentrate on the work at hand rather than worrying about how to use the programme.
3. Ease of Extensibility: Because the programme is broken down into components, it's much easier to add new ones without interfering with the existing ones. If you wish to add a new backend, for example, you can do it without changing the frontend.
4. Fast Development:MVC provides for rapid and concurrent development. One programmer may work on the view while the other works on the controller to construct the web site's business logic if an MVC model is utilised to design a web application. As a result, an application developed using the MVC paradigm can be finished three times faster than one developed using other development methods.
5. The ability to present multiple perspectives:
6. One is able to create several views for a model in the mvc. There is potential growth in the way for demanding for additional ways to access your application today, and MVC development is an excellent answer for this. Moreover, in this method, Code duplication is very
7. Asynchronous Technique Support:The JSarchitecture can be integrated with the MVC design. This implies MVC apps may be configured to function with PDF files, site-specific browsers, and desktop widgets as well. MVC also includes an asynchronous method that allows developers to create applications that load quickly.
8. Data is returned from the MVC model without being formatted: The MVC pattern returns data that hasn't been formatted. As a result, every interface can use and invoke the same components.

### **Popular MVC frameworks:**

- Ruby on Rails
- Django
- CakePHP
- Yii
- CherryPy
- Spring MVC
- Catalyst

- Rails
- Zend Framework
- Laravel

## 1.2 Problem Statement

### Software solves problems:

We aren't just software developers. Code doesn't just appear in our files from nowhere. We break down the requirements into small problems that we are familiar with, and then we solve those small difficulties. Each piece of code you write, have written, or will write solves a specific problem. It doesn't matter if you're trying to save the planet, show kittens on a screen, or make it seem decent in Internet Explorer 8. Don't touch it; it's there for a reason.

Problems can be solved, and the answers to these problems form part of a larger whole. A black box that meets all of the initial criteria. But how do we address these issues? Is my solution the most appropriate? Will other devs (or myself in two months) comprehend what I did.

All problems have a single solution.!

We are compelled to adopt MVC to solve the web-application challenge. Those that don't use it are frequently chastised by the developer community, but they are rarely asked why.

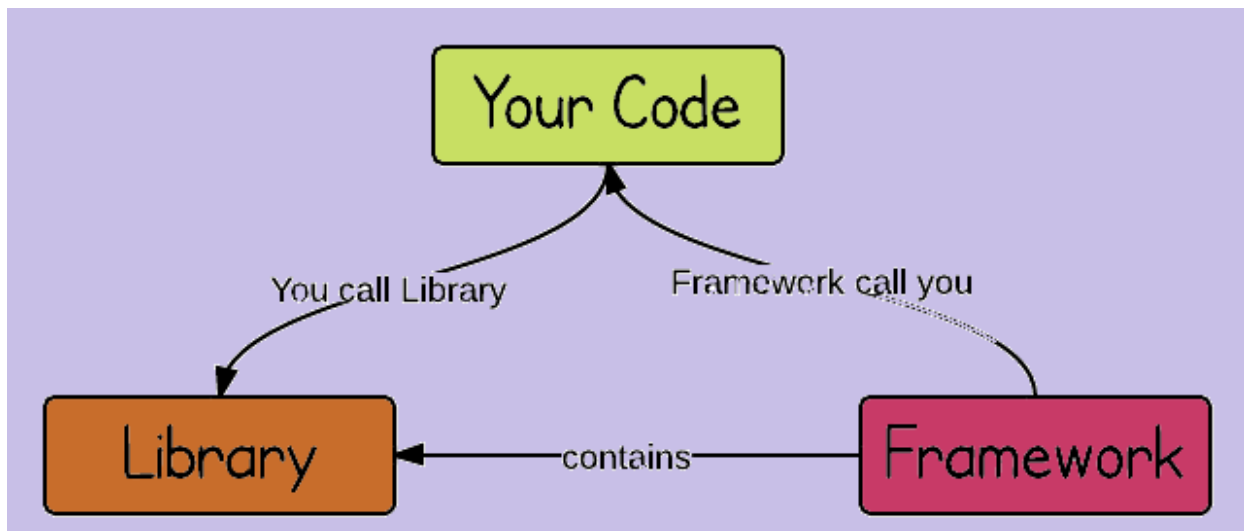
So, why is MVC the most effective? The following are frequently cited reasons:

- Code complexity is reduced.
- Reuse of code
- Greater adaptability
- Separated code

### 1.3 Objectives

Framework is defined by the dictionary as "basic structure supporting a system," which is similar to what a framework is in programming. The framework notion is the same whether the system is a house, a car, a theory, or a mobile app: it offers support and a basic 'guide' for the structure being developed.

In programming terms, a framework is a tool that provides pre-built components or solutions that may be updated to speed up development. A framework can include a library, but it is specified by the inversion of control principle (IoC). Custom code can request into the library to retrieve reusable code in traditional programming. When custom parts of code are required, the framework uses IoC.



In order to develop software designs and create systems, a framework can consist of several programmes, compilers, code libraries, toolsets, and APIs. Open-source frameworks are modified and updated and checked on a regular basis.

### What are the Benefits of Frameworks in Software Development?

The main aim of the framework is to help developers by providing standard, low-level functionality so that they can work on the features of the project to make it better and make it unique.

The inclusion of pre-tested, high-quality functionality improves software stability. It cuts down on development time and makes testing easier. By having an active user base and ongoing code updates, frameworks serve to increase security and give a foundation of support.

Finally, frameworks are used to save time and money.

Choose a framework that provides the functionality required for the project at hand, keeping in mind that each framework has its own set of restrictions and that you should not invest in one that accomplishes far more than your project will ever require.

Consistency — For big or distributed teams, a framework can help with consistency.

Documentation — look for a framework with well-documented code and implementation training.

Choosing a framework that is too new or poorly supported carries the danger of requiring costly retooling if the framework becomes obsolete. Similarly, if the framework includes restrictions that are not clearly understood at the outset, the project may suffer.

## 1.4 Organization

### Real Time Data Services(RTDS):



**About RTDS:** Founded in 2010.

For more than a decade, we have been offering modern IT solutions in the fields of communication and cloud computing to customers all over the world.

Real Time Data Services (RTDS) is a set of organisations that thrive in the field of global information technology by providing Cloud Computing and Communication services to clients. The organisation provides technology solutions to businesses all around the world that help them manage their varied operations.

Our expertise in cloud computing and IT enables us to provide solutions that simplify and improve corporate efficiency. Our clients can expect the following services from us:

Server virtualization and consolidation technologies for establishing private clouds Accounting and business applications in the cloud



Cloud-based communication systems that allow all employees and departments to communicate from any device.

To assist organisations in connecting with the right people, provide a lean, scalable, and cost-effective telephony solution.

We recognise that every customer requirement is unique, which is mostly determined by the sort of sector they are in and the size of their firm, as we have been in the cloud market for some years.

With this in mind, we provide cloud solutions tailored to the sectors our customers serve. You won't have to worry about server specifications or basic mechanics of the IT backbone if you use custom-made cloud solutions.

Cloud is a forward-thinking technology that gives today's telecommunications a new edge. With its extensive feature sets, cloud telephony makes things easier. It is one of the main reasons why most small and large businesses are switching to cloud telephony from their old setup.

When it comes to today's commercial communication difficulties, cloud telephony stands out as a technology that has all the answers.

We give a range of solutions that solve communication needs with some forward-thinking concepts for personal usage or any type of corporate organisation.

### **Success Timeline and visions:**

RTDS began operations in 2010. To provide a smooth hosting experience and 24\*7 support, we've partnered with data centres in Buffalo, Chicago, Dallas, Seattle, New York, and Phoenix.

Services for cloud telephony began in the United States. Salesforce Cloud Telephony Services launched in the United Kingdom, and we became an authorised hosting provider for Intuit.

At AccountexUSA, we received the User Favorite Award for QuickBooks Hosting.

In India, Cloud Telephony Services were launched.

In the United States, 2000 business accounts were subscribed for accounting and business

application hosting.

In India, 1300 small and medium businesses have been onboarded for Cloud Communication Solutions.

In 2019, the RTDS group plans to generate \$100 million in sales and expand into three new nations.

### **Partners:**

- **Ace Cloud Hosting**

Real Time Cloud Services, LLC., whose holding company is RTDS, is the parent business of Ace Cloud Hosting (ACE). RTDS offers ACE a wide range of business support services, including managed and dedicated hosting for tax software, accounting programmes, and other business tools including add-ons, office suites, web tools, and CRMs.

ACE got the User Favorite Award for their services at Accountex USA 2016. ACE is an Intuit Authorized Commercial Host for QuickBooks desktop accounting software and a significant distributor of Intuit products. To ensure the quality of its services, ACE offers specialised 24/7 assistance and a 99.999 percent uptime guarantee.

- **Real PBX**

The Real PBX Pvt. Ltd. owns The Real PBX, a business communication solution. The company, which is situated in London, provides cloud-based solutions to clients all around the world. RTDS provides business support services to the Real PBX (TRP).

TRP provides Hosted PBX, Toll-free numbers, Local Numbers, Dialer Hosting, Dialer Support, and other services. By providing the whole communication configuration over the cloud, TRP reduces the need for local hardware at the client's end. Another feature that distinguishes TRP as a forward-thinking corporate option is that its solutions are compatible with a variety of platforms and operating systems, allowing users to connect from any device – desktop, smartphone, tablet, and so on.

## CHAPTER-2

### LITERATURE SURVEY

Model, view, and controller (MVC) is a well said three-layer development skeleton for developing online applications. The article discusses the MVC layers, their applications, benefits, and best practises for using MVC in web application development. We

Explain each of the three layers and their functions in detail. The study's major goal is to provide a comprehensive picture of each MVC layer as well as the MVC's major features and benefits. We go over the best practises for developing MVC-based web apps in particular

In addition, we discuss the technologies that are utilised in combination with MVC-based web applications.

A well-known 3-layer development pattern for designing online applications is model, view, and controller (mvcc). This article discusses the MVC layers, their applications, benefits, and best practises for using MVC in web application development. We explain each of the three layers and their functions in detail. The study's major goal is to provide a comprehensive picture of each MVC layer as well as the MVC's major features and benefits. We go over the best practises for developing MVC-based web apps in particular.

One developer will work on user input logic (controller logic), another will work a third will work on business logic while the first two work on user interface logic (view). (model).

The MVC pattern architecture will be presented in the second section of our study report. In this section, we'll go through some of the benefits of employing the MVC pattern architecture.

The features of the MVC framework will be described in the third portion of our study report.

#### **When should you utilise the MVC pattern?**

Separation of concerns is introduced by the mvcc pattern design, which allows us to implement it

among its components classes within programmes. Because the interactions between the various components are more obvious and consistent due to the separation of concerns, it is easier to test our programme. MVC helps us apply a test-driven development methodology, in which automated test cases are written before writing code. Before building new code, we use these unit test cases to define and verify new code needs.

If we're developing a client-side application with sufficiently significant stimulation, we shouldn't employ JavaScript alone.

If we're building an application with a lot of server-side lifting and little client-side communication, we shouldn't utilise instead of using the MVC paradigm, we should employ a basic arrangement such as a web-based form model. The features listed below will aid us in deciding whether or not to use MVC design in our application:

- i. On the backend, our application requires asynchronous communication.
- ii. Our application contains a feature that causes it to not work to refresh an entire page, such as when leaving a comment on a blog article using Facebook or scrolling indefinitely.
- iii. The majority of the client (browser) manipulates the data, not the server.

### **Features of MVC framework:**

Because our application's logic is divided into less than four jobs (input, business, and interface logic), evaluating its parameters is simple. It would become quite simple. Testability is quick and adaptable. We can use any mvce-compatible unit test design. It's a pluggable and extendable framework. We can create our application's components in such a way that they are easily interchangeable or modifiable. We can utilise our own view engine, url routing strategy, and other custom features, and serialisation of action method constraints. We utilise the dependency injection (DI) approach to inject objects into classes instead of relying on classes to create objects. Another way for demonstrating object dependency is inversion of control (IOC), which defines which object requires which other object. MVC includes a URL mapping component that allows us to create URLs that are both comprehensible and searchable. MVC supports URL naming patterns instead of file name extensions, which are highly beneficial for search engine optimization (SEO) and representational state transfer (REST) addressing.

Built-in features include form authentication, session management, transactional business logic, online application security, object relational mapping, localization, membership and roles, and URL authorization of several MVC frameworks, such as ASP.NET MVC. The infamous frameworks presently are Backbone.js, ember.js, angular.js, and knockout.js.

I. Backbone.js- The Backbone.js framework comes in handy when our application requires flexibility and we have a variety of requirements. We also want to be able to handle changes as the application develops.

II. Ember.js- If we want our application to interface with a JSON API, we should utilize the ember.js framework.

III. Angular.js- If we want our application to be more reliable and stable, as well as have rigorous testing, we should use the Angular.js framework.

IV. Knockout.js- If we want to create a complicated dynamic application interface, the knockout.js framework will come in handy.

Every design has its own set of benefits or drawbacks. Developers can use any of the frameworks that best suit their web application requirements.

The MVC pattern architecture was addressed in this research report. We've spoken about how it can be used to create web based products. We spoke about the various advantages of the MVC design as well as numerous MVC frameworks. We show how to build an e-commerce application utilizing various tools and technologies as an example of MVC design.

## CHAPTER-3

### SYSTEM DEVELOPMENT

#### **Tools and technologies used:**

With the help of MVC architecture, various tools and technologies may be used to construct online applications.

Developers can construct web apps using any of the tools and technologies available, based on their preferences. Some tools and technologies that can be utilised to construct MVC-based web applications are as follows:

#### **Tools:**

Microsoft Visual Studio: Visual Studio is not only a tool, but also a full dev-environment that allows us to construct a variety of applications. Visual Studio is quite useful for developing applications utilising the ASP.NET MVC framework.

1) MySQL Server is a relational database management server that is used to keep the database up to date.

2) SQL Server is a database engine for maintaining databases, similar to MySQL Server.

The MySQL Workbench is a MySQL database design tool.

4) NetBeans IDE (Integrated Programming Environment) is a full-featured development environment for a wide range of applications.

Glassfish Server is a Java Enterprise Edition application server.

#### **Technologies:**

Design with HTML, CSS, JQUERY, and AJAX.

B. NetBeans Enterprise Java Beans (EJB) technology uses servlets and Java Server Pages (JSP).

JSTL (Java Server Pages Standard Tag Libraries)

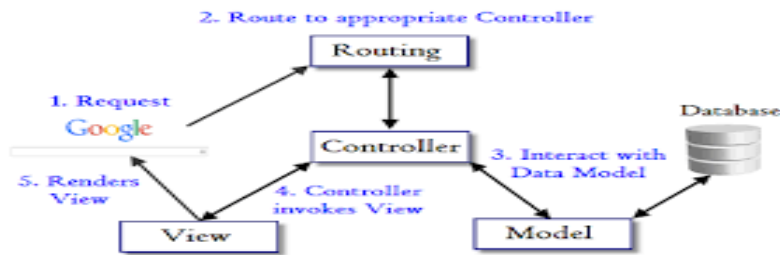
E. JPA (Java persistence API)

F. jdbc (Java database connectivity)

G. Visual Studio with ASP.NET MVC

There are a plethora of different tools and technologies that may However, MVC architecture should be used with caution. We've selected a few that we'll be using in our web development. Application with MVC architecture

### An application using the Mvc pattern as an example:



We sought to build an Ecommerce website as part of our inquiry. In this application, the MVC architecture is employed, It will assist developers in comprehending the features of the model view and controller pattern. The sale and purchase of goods over the internet is referred to as e-commerce. Overstock.com, for example, is a great place to start, selling a wide range of products from a number of producers. Certain products are available for purchase on the internet.

There are various sorts of e-commerce software available, including business to business, consumer to consumer, business to consumer.

We designed a business to consumer application in our software that normally supports a B2C situation and consists of two parts: foremost, a storefront where customers can look at various products data is stored in a database and dynamically displayed to customers on the pages.

The admin console, which is a password-protected area, is the second component. Customers cannot use it; instead, the administration can use it to monitor orders, delivery, and payment

options are examples of transactions processed by customer as well as CRUD access to the database store.

Java is the technology employed in our application. Java is a programming language that includes the basic idea something can be immediately grasped it adheres to an object-oriented approach in which encapsulated objects communicate with one another by passing data. To find mistakes, Java has compile and run time verification. It also has security features for networks. It is portable; the application can be readily ported from one platform to another with minimal or no changes. We can run programmes rapidly because the Java interpreter is independent of the runtime environment. It also has a trash collector that cleans up unused memory.

Java is not only a programming language; it is also a complete platform, which means it is made up of two parts. The JVM (java virtual machine) is the engine that executes the instructions that the java compiler has translated.

It's a JRE instance found in a wide range of products, including operating systems, browsers, and web servers. The platform's second component is the Java application programme interface (API), which provides prewritten software organised into packages. These packages offer classes for designing user interfaces, such as typefaces, buttons, and menus. There are several editions of the Java platform, including Java standard edition, Java micro edition, and Java enterprise edition. Here, we're utilising Java Enterprise Edition.

Java Enterprise Edition includes capabilities for creating and maintaining server-side applications that are robust, consistent, scalable, and secure. There are two sorts of enterprise edition technologies. Web application technologies and enterprise application technologies are two of these categories. We will mostly use web Because we must build a web application, we must use application technologies in our application. Our online application makes use of enterprise technologies such as Servlet, jps, jstls, EJB, and JPA. Java EE technologies are the most commonly utilised technologies on the market, as shown in a figure derived from an independent examination of European market performance in 2007.

The net beans editor, database sql server, and MySQL workbench were the tools we utilised to construct our application. NetBeans is a programming environment that allows you to create professional desktop, mobile, and online applications. MySQL is a database engine that we We

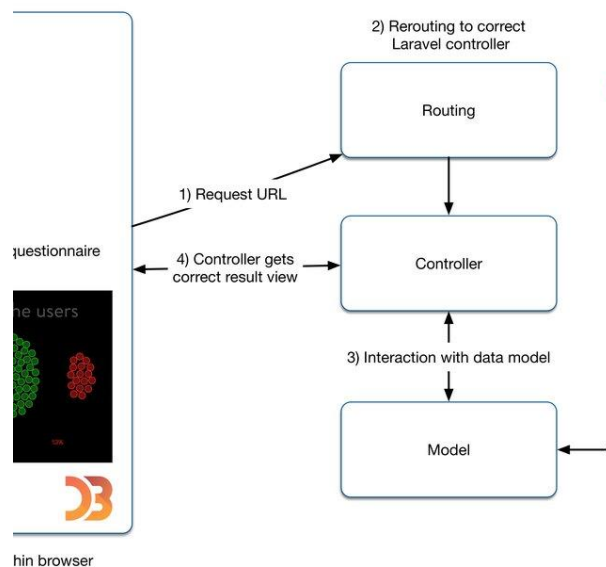


use it to keep our database current. MySQL Workbench is a visual database design tool for ERD.

## One of the most widely used application of MVC is Laravel.

Laravel is a strong and easy-to-understand open-source PHP framework. The design pattern is model-view-controller. Laravel reuses existing components from several frameworks to aid in the development of web applications. The resulting web application is more organised and practical.

Laravel has a large collection of capabilities that combine the best of PHP frameworks like CodeIgniter and Yii, as well as other programming languages like Ruby on Rails. Laravel provides a large number of features that will speed up web development.



## Advantages of Laravel:

- When creating a web application with Laravel, you get the following benefits: The web application becomes more scalable thanks to the Laravel framework.
- Because Laravel reuses components from various frameworks in constructing web applications, significant time is saved in designing the web application.
- It comprises namespaces and interfaces, which aid in resource organisation and

management.

## **Composer:**

Composer is a tool that combines all of the necessary dependencies and libraries into one package. It allows users to create projects that correspond to the stated framework (for example, those used in Laravel installation). Installing third-party libraries is a breeze with Composer.

The `composer.json` file in the source folder contains a list of all dependencies.

Laravel will make your process easier if you are comfortable with Core PHP and Advanced PHP. If you're building a website from the ground up, it'll save you a lot of time. Furthermore, a Laravel-based website is safe.

## **Artisan:**

The Laravel command line interface is known as Artisan. It comes with a set of instructions that will guide you through the process of creating a web application. The Symfony framework has been updated to include these commands, resulting in Laravel 5.1 add-on features (latest version of Laravel).

## **Laravel's Features:**

Laravel has the following essential qualities that make it an excellent choice for web application development:

### **Modularity**

Laravel comes with 20 built-in libraries and modules to help you improve your application. Every module is linked to the Composer dependency management, making updating a breeze.

### **Testability**

Laravel comes with a number of tools and helpers that make testing easier. This feature aids in the maintenance of the code in accordance with the criteria.

## **Management of Configuration**

A web application written in Laravel will run in a variety of contexts, which means its configuration will change frequently. Laravel provides a standard technique to efficiently handle setup.

## **ORM and Query Builder**

Laravel has a query builder that allows you to query databases using simple chain methods. Eloquent is an ORM (Object Relational Mapper) and ActiveRecord implementation.

## **Schema Builder**

Schema Builder is used to maintain database definitions and schema in PHP code. It also keeps track of any database changes done during migrations.

## **Engine for Templates**

The Blade Template engine is used by Laravel, which is a simple template language for constructing hierarchical blocks and layouts with dynamic content.

Laravel offers a mail class that allows web applications to send email with rich content and attachments.

## **Authentication**

In online applications, user authentication is a standard feature. Laravel makes authentication easier by including tools like signup, forget password, and send password reminders.

## **Redis**

Laravel connects to an existing session and uses Redis as a general-purpose cache. Redis communicates directly with the session.

## **Queues**

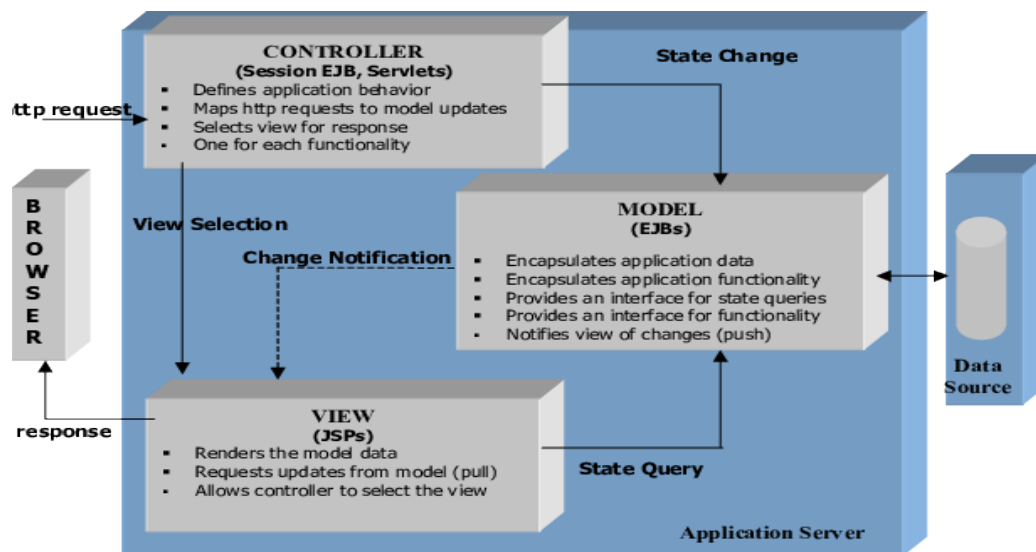
Laravel provides queue services such as emailing a large number of users or running a Cron job.

These queues make it easy to accomplish tasks without having to wait for the prior task to finish.

## The Command and Event Bus

Laravel 5.1 features the Command Bus, which makes it easy to execute commands and dispatch events. The commands in Laravel follow the lifespan of the application.

## Styling the application:



This design's highlights how that product may be used to meet realistic world's needs corporate needs.

We project to acquire customer needs before designing the application. The needs of the customer should be obvious. Our applicationn should meet the following client needs, according to our standards.

- An online depiction of data that needs to be saved in our database. The customer will see the following information about our products: (name, image, and description).
- Place things in a shopping cart that our customer has temporarily chosen.
- The number of the selected products can be changed by the customer.
- The client can remove a product from their shopping basket.

e. The client can see a summary of his shopping cart, including the products he selected and in what quantities.

f. A secure checkout process allows customers to place orders and make payments.

g. Our programme should allow administrators to check the orders and add new products for customers on the server side.

Following the requirements collecting, we should build the application to meet the needs of the customer. Will attempt to create a use case that will describe and encapsulate the behaviour of our application. Because this is about creating an e-com website in our application, our use case will define the stated below actions that will be presented by various actors.

- 1) The client can go through various items in the chosen field.
- 2) The client will place his selected products from various categories in a shopping cart and then purchase them later.
- 3) The client can check the contents of his shopping basket to see what he has chosen.
- 4) The clientt can examine the information in his shopping cart before proceeding to the checkout procedure.
- 5) On the checkout page, the customer will enter his personal information and see his order details.
- 6) The consumer will confirm his order after viewing the checkout process page.
- 7) After successfully logging into our programme, the administrator will be able to see the ordersof various clients.
- 8) The administrator has the ability to add new products to various categories.

We must identify and develop web pages that will be used in our application based on the above-mentioned use-case. Various ways can be used to make mockups. Paper prototyping is a technique we employ to build our application, in which we strive to collaborate with clients by drawing ideas on paper. We created five web pages for our application: a welcome page (where

our main product categories will be displayed), a category page (which will display a list of different products as well as shopping cart and proceed to checkout options), a cart page (which will display a list of all the product items that we have selected for purchase), and a checkout page (which will display a list of all the product items that we have selected for purchase) (which will reflect that his order has been placed successfully).

### **Determining the architecture:**

We should do the following before beginning the application development process: choose the type of architecture we will utilise to design our submission. The functional components should be investigated of our application and identify how they will interact with it. Interact with one another. There are numerous design patterns available for use in the development of an application. We've arrived. The MVC architecture pattern will be used.

Interact with our database. The view is used to convey data from the model to the user. Finally, the controller is used to specify the behaviour of the programme. It interacts with the user's inputs. HTTP GET and HTTP POST are the most common user inputs in web-based applications.

We'll be dealing with jsp technologies here, and our corporate concept will be coded into JSP pages utilising scriptlets. The script allows you to put snippets in `%% >` tags. Servlet pages will be generated from JSP pages.

In 1999, JSP was originally introduced, along with a description of two model designs. The first style involves directly implementing business logic with JSP pages, while the second model employs the MVC pattern design, which we employ in our application.

### **Replanning the data model:**

We should construct a conceptual model of our storage system in this stage by defining the entities in our system and their relationships. Our data model will include both logical and physical design elements that will be used to produce a data definition language (DDL) script that will aid in the creation of a database. We'll utilise MySQL Workbench to construct data models, reverse-engineer SQL into visual presentations, forward-engineer data models into

database schema, and synchronise models with a running MySQL database server, as we mentioned previously.

To create an ERD (entity relationship diagram), we identified four entities in our application: client, category, product, and order.

We used the Eclipse Link feature in the IDE to generate JPA classes from the database schema we created from the ERD.

We adopted the bottom-up strategy, which starts with a pre-existing database schema. It shows how to utilise a forward engineering tool to extract metadata from a schema and produce JPA entity classes in the simplest way possible.

### **Development of application:**

Building JSP pages and arranging them in proper locations, defining headers and footers that will be displayed in every view, and setting up a Servlet that will handle all incoming requests from users are all steps in the application development process. A deployment descriptor (web.xml) was also created, which held all of the application's configuration data.

We allowed our application to communicate with the database by entering sample data into the database and setting up a data source and a connection pool on Glassfish server to test the data source using JSP pages after developing the front-end.

On the index and category pages, we additionally use the JSTL and SQL tag libraries to retrieve and display photos for various categories and goods.

Then, for each table in the database, we generated a java persistent API class with named query annotations, fields (columns), and relationships (foreign keys). Each entity class's access methods were likewise defined using the EJB session façade.

Medel component development in EE is accomplished using the following two technologies:

a) enterprise java beans: This is a server-sside feautre architecture for developing distributed, transactional, and comfortable applications quickly and easily. It generates a pool of component instances that users can share. Thread safety and transaction management are also included. EJB

provides the Java Authentication and Authorization Service (JAAS), which secures our application.

b) java persistent: java persistent stores the data in the database that the java objects contain. It is an object relational technology that allows the programme to handle data between java objects and the database in a way that the developer can see. Data transparency for the developer allows for data caching and speed optimization.

To perform a business function, users call enterprise session beans. Only a limited number of session beans are available. The session bean crashes when the ejb container crashes, and the user must create a new session object to finish the calculation. The EJB provides three different types of session beans: stateful, which is useful for operations that can be performed in a single method call, stateless, which is useful for operations that can be performed in a single method call, and singleton, which is instantiated only once per application and exists until the application is no longer active.

We use a pointless session bean in our application. We used the HTTP protocol, which is a stateless protocol that allows communication over the internet.

Before the information is correctly entered into the database, we must build a session in which user-related information is retained for a period of time. We used HTTP Session to keep track of user sessions (cookies). We additionally secured our application by appropriately configuring the administrative console login process, ensuring that only authorised administrators have access to console services and that unauthorised users are prevented. We've set up a secure data transfer for both the consumer checkout process and the administrator console.

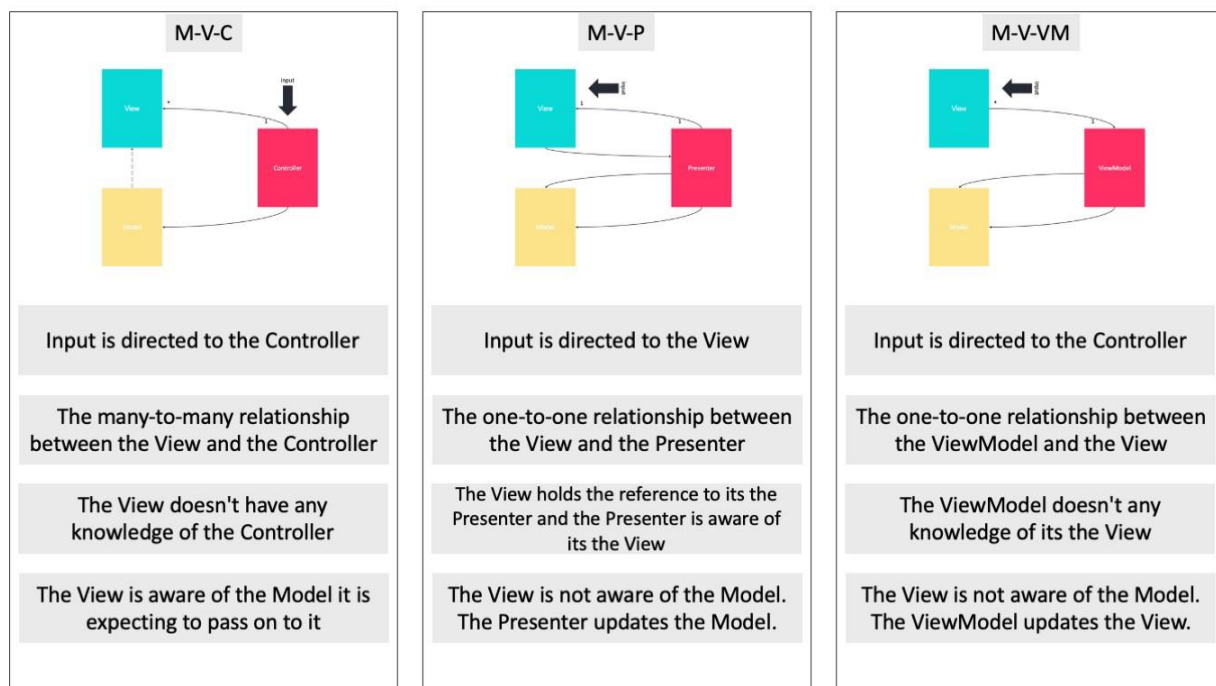
We created users groups that correlate to our application using the Glassfish server administrative dashboard. We also used the JMeter kit plug-in to test our application. It allows us to determine whether or not our application is performing as expected.



## CHAPTER-4

### PERFORMANCE ANALYSIS

Web apps have become such a common occurrence in the modern world. Due to its sophisticated nature and high technicality, Web applications are not widely understood and are incredibly difficult to grasp fixed appliances. We may now express, manage, store, and communicate sensitive consumer data using web applications for immediate and frequent use. Some Web apps display results in a matter of seconds, while others take a long time. Web application performance has been improved with this work. The main goal of this research is to compare Model View Controller and Cloud architecture. This covers developing web applications with CRUD functionality in a variety of architectures.



Web apps have become such a common occurrence in the modern world. Due to its sophisticated nature and high technicality, Web applications are not widely understood and are incredibly difficult to grasp fixed appliances. We may now express, manage, store, and communicate sensitive consumer data using web applications for immediate and frequent use. Some Web apps display results in a matter of seconds, while others take a long time. Web application performance has been improved with this work. The main goal of this research is to compare

Model View Controller and Cloud architecture. This covers developing web applications with CRUD functionality in a variety of architectures.

- Class diagrams for the second application were used to code the applications.
- Using digital data to compare two applications:

As a result of application performance measurement, digital data was obtained. To gather digital data, the following tool was employed.

Performance measurements of applications were taken, and digital data was obtained, in order to compare the performance of two apps. These digital data were obtained using the Jmeter testing tool.

Jmeter is a piece of software that allows you to test and measure performance as well as perform a graphical analysis of it.

When we study the aforementioned statistics, we can conclude that the MVP-written application is easier to test than the MVC-written application, and that the user waits less when doing actions.

Even when we adjusted the number of tests, the number of users or scenarios, or the performance results between two applications, we saw no difference and got results that were similar to the preceding data.

Using Jmeter, the identical action was performed by one, fifty, one hundred and fifty people at a defined usecase, and average time figures were displayed. The following graphical comparison was generated using average time data. The same scenario allowed us to obtain the following data for various numbers of users considering GUI components. It becomes feasible to create a completely tested kind of behaviour. The development of the user interface and the business logic can be done independently and by distinct teams. Components that require communication between Mediator and each other can be developed modularly and separately.

Because unit tests govern functionality, any problems arising from modifications to the user scenario are recognised early. Changes to the user interface are more easier and safer to make. After all, it is well understood that any changes made here will have no influence on the

function.

Model, View, and Controller (MVC) is an architectural pattern that divides an application into three basic logical components. MVVM, on the other hand, allows for the separation of graphical user interface development using mark-up language or GUI code.

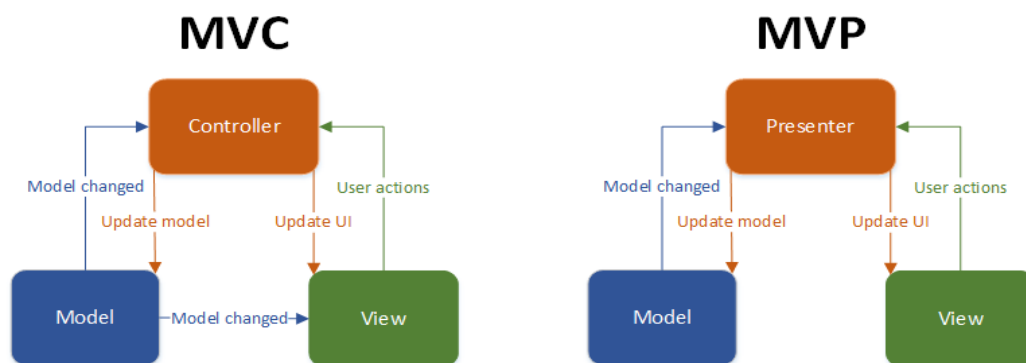
In MVC, the controller is the application's entry point, whereas in MVVM, the view is the application's entry point.

The MVC Model component may be tested independently from the user, whereas MVVM allows for easier unit testing and event-driven coding.

In MVC design, the Controller and View have "one to many" relationships, whereas in MVVM architecture, the View and View Model have "one to many" relationships.

### The main distinctions:

- In MVC, the view and the model are tightly tied, but in MVP, they are loosely coupled.
- Communication: The View-Presenter and Presenter-Model communicate with one other via an interface in MVP. In MVC, however, the controller and view layer are both part of the same activity/fragment.
- User Contribution: User inputs are processed by the Controller in MVC, which instructs the model on how to proceed. In MVP, however, user inputs are handled by the view, which tells the presenter which functions to call.
- Relationship Type: The controller and the view have a many-to-one connection. In MVC, a single Controller can select different views based on the required operations.



MVC has two variants: MVP and MVVM. The main distinction between MVC and its variants is the degree of dependency each layer has on the others, as well as how closely they are linked.

The View is placed on top of our architecture in MVC, and the controller is placed beside it. Models are located beneath the controller, so our Views are aware of our controllers, and controllers are aware of Models. Our Views have direct access to Models in this area. However, depending on the complexity of our application, exposing the entire Model to the View may have security and performance implications. MVVM tries to prevent these problems.

The Presenter takes the position of the controller in MVP. Presenters sit at the same level as views, listening to both the View and the model's events and mediating their activities. There is no way for connecting Views to ViewModels in MVVM, therefore we rely on each View having an interface that allows the Presenter to communicate with it.

As a result of MVVM, we can design View-specific subsets of a Model that can contain state and logic information without having to expose the complete Model to a View. A ViewModel does not need to reference a View, unlike MVP's Presenter.

The View can bind to ViewModel properties, which in turn expose data from Models to the View. As previously stated, the View's abstraction means that less logic is necessary in the code underneath it.

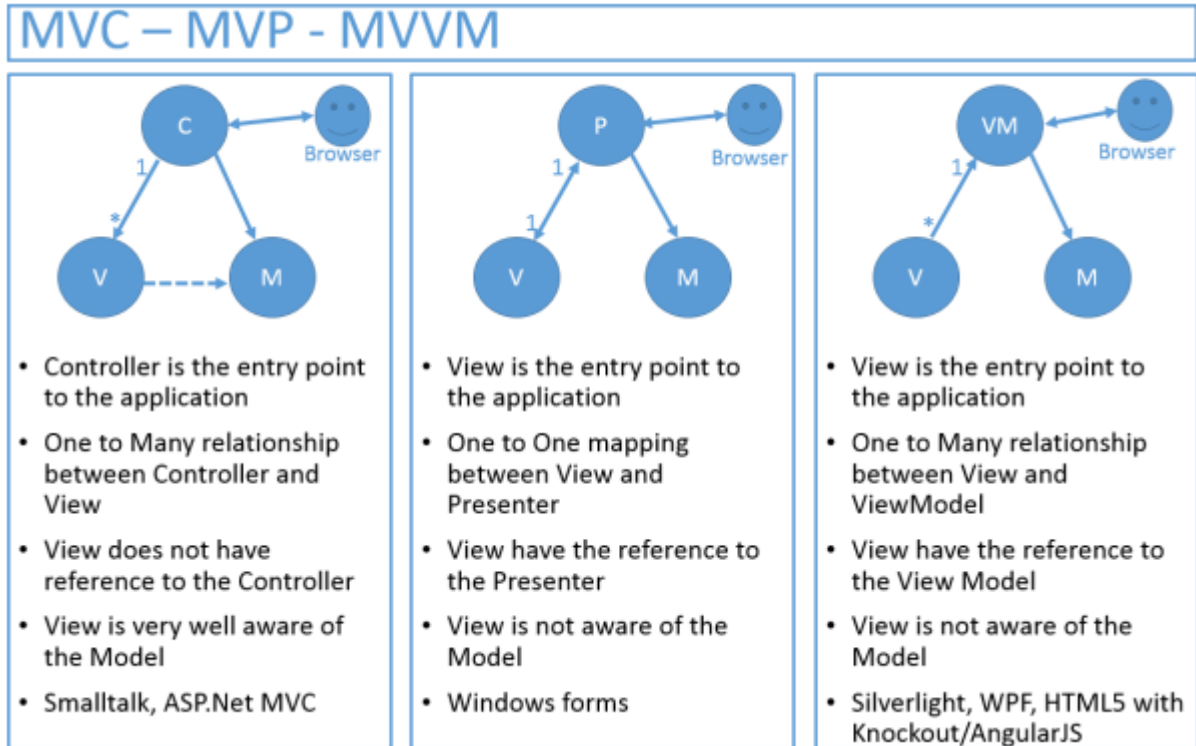
One disadvantage is that a level of interpretation is required between the ViewModel and the View, which can have performance implications. The intricacy of this interpretation varies as well: it can be as basic as copying data or as difficult as modifying it into a form that the View will recognise. This is not a problem with MVC because the entire Model is easily available and such modification can be avoided.

The goal of these architectures is to segregate the tasks of UI applications' displaying, processing, and data management.

**Their objectives are to increase:**

- Modularity
- Flexibility

- Testability
- Maintainability



## CHAPTER-5

### CONCLUSIONS

#### 5.1 Conclusions:

In the area of web-based applications, the MVC design has more clout. It is really beneficial to developers. MVC design allows developers to quickly create web apps.

It simplifies the application's complexity by breaking it down into three primary components: model, view, and controller. It gives developers complete control over the online application. Testing multiple components of an application becomes more easier when using the MVC architecture. MVC design gives their web applications a higher level of abstraction.

#### **Suggestions for Developers:**

MVC architecture is a fantastic alternative for web developers who want to work on web-based apps since it gives more strong features for web applications compared to simple web form-based web applications. The MVC architecture makes developing online applications relatively simple and effective since it divides the concept of web applications into three layers, reducing the complexity of the application.

These three layers can be worked on separately by developers. MVC provides direction and a complete structure for developers who want to create big-scale and complicated web applications. Developers can write java script code that is well-structured and maintainable.

MVC architecture is language neutral, thus developers can utilise it with a variety of high-level languages.

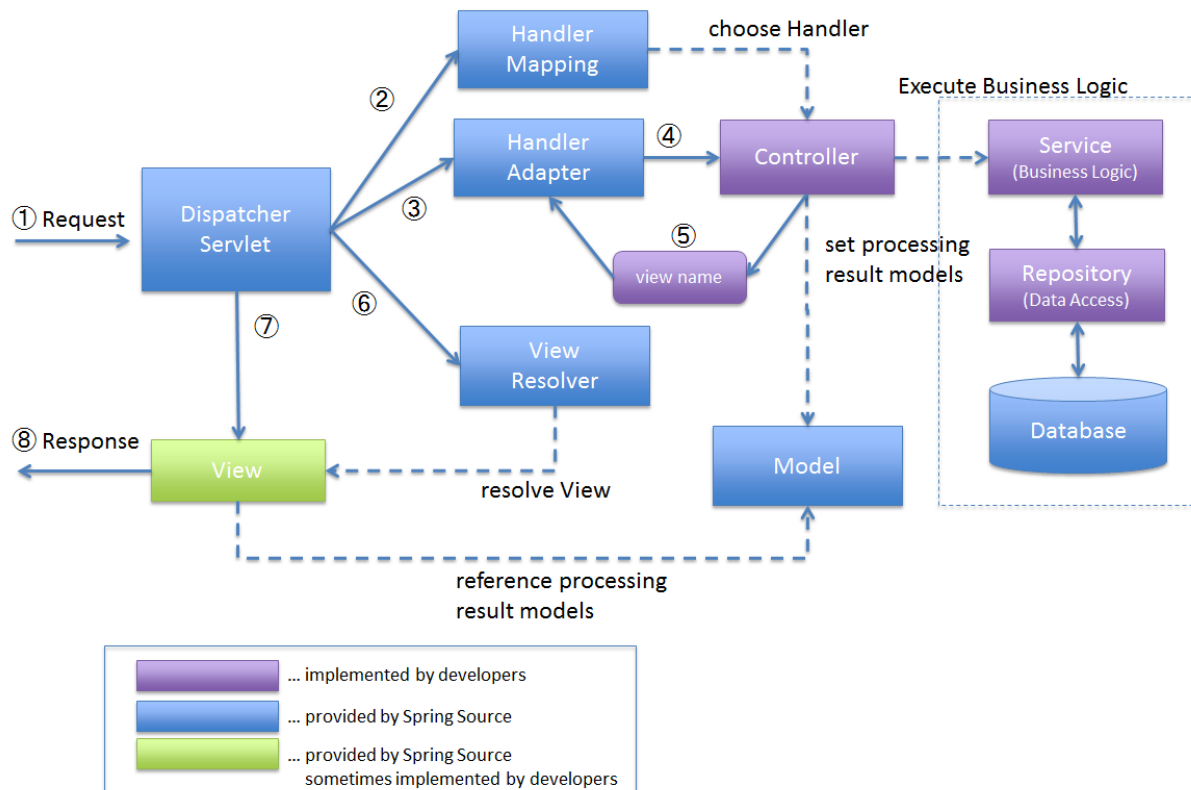
The MVC design allows developers to utilise java, asp.net, and other languages to create online applications. MVC is an excellent alternative for developers who wish to master multiple web-based technologies while also strengthening their skills in Java EE technologies.

We're providing a link for developers interested in working on J2EE web-based applications utilising MVvC architecture and constructing an e-commerce application using various technologies such as JSP technologies by following various stages.

## Tips for Finders/Researchers:

MVC archit is not dependent on programme languages, allowing developers to create web applications in any language of their choice. We have provided a guidance to developers in our study paper for developing web applications using JSPp technology such as JSP, Servlet, EJB, JPA, and others.

**Other researchers can work on different MVC frameworks** and provide developers with guidelines. The ASPp.NnET MVC framework, for example, is the most popular framework today. It has numerous strong tools for developing online applications, including as Ajax, cshtml, and jQuery. Researchers can work with the ASP.net MVC framework to learn about its features and offer assistance to developers.



## To conclude:

The creation of web applications has been eased because to MVC patterns and added

technologies such as JSPp, Servlet, and EJB on the J2EE platform. MVC architecture allows us to create scalable, transparent, and portable web applications. MVC architecture allows for simultaneous development since it divides the concept of the application into 3 layers, allowing multiple developers to work on the same web application at the same time.

The model is the first layer of the MVC design, and it is used to interface with the database. It is our application's logic layer that is utilised to insert, retrieve, and update data in the database.

The 2nd layer is view, which is concerned with the user interface, It is generally used in front end development to describe how people interact with our programme and how it will be shown to them. The controller is the third component of the MVC architecture, and it is used to receive user input. It regulates the state of model objects and the data displayed to consumers in response to user inputs. Our application can respond to user requests via the controller layer.

MVC architecture can be used to develop a scalable, transparent, and portable web-based application. MVC architecture teaches us about parallel development since it divides an application's logic into three layers, allowing multiple developers to work on the same web-based application at the same time. Setting up the MVC architecture is usually expensive. Small MVC projects and non-MVC projects often take the same amount of time to process, however large MVC projects are typically handled faster than non-MVC projects.



## 5.2 FUTURE SCOPES

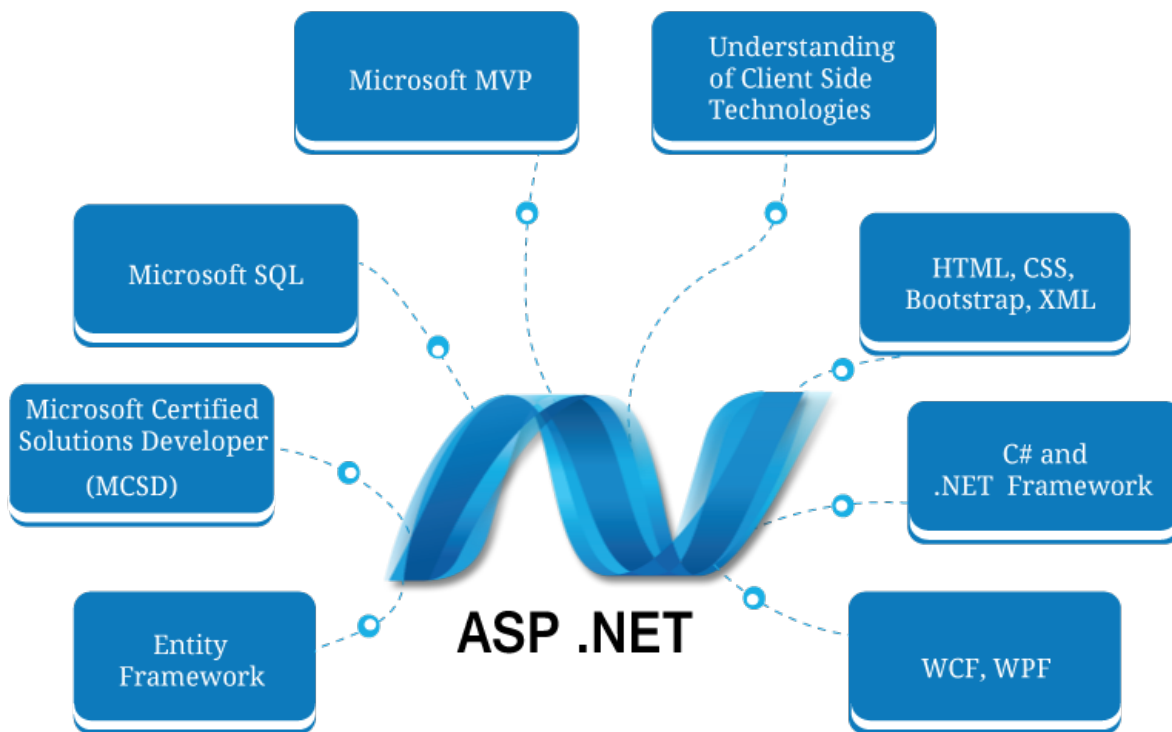
Web applications combine front end and back end technologies, but since the World Wide Web has evolved at such a rapid rate and because developers must use a wide number of technologies to build a single web application, the end result is sometimes difficult to manage and correct. To construct dynamic web pages and applications, developers blend HTML code with server-side programming languages, resulting in extremely entangled and unmaintainable code. Another issue has arisen as web technologies are widely employed to create a wide range of complicated applications. Microsoft uses web technology to entice developers to create apps for Windows 8, its newest operating system.

Additionally, frameworks such as PhoneGap and Appcelerator Titanium exist to assist web developers in creating mobile applications. Furthermore, a mobile operating system is being developed, with devices based entirely on web technology coming later this year (Mozilla Firefox OS). Because of these factors, a web application is typically developed by a team of specialised developers, each using their preferred technology, such as HTML and CSS for the presentation layer, JavaScript for client-side interaction, PHP (or ASP, Java, Python, Pearl, Ruby, and other languages) for server-side logic, and MySQL (or Oracle Database, Microsoft SQL Server, and other databases) for data storage and management. Each of these specialists must collaborate with their peers to ensure that their code parts are integrated into the application's general design.

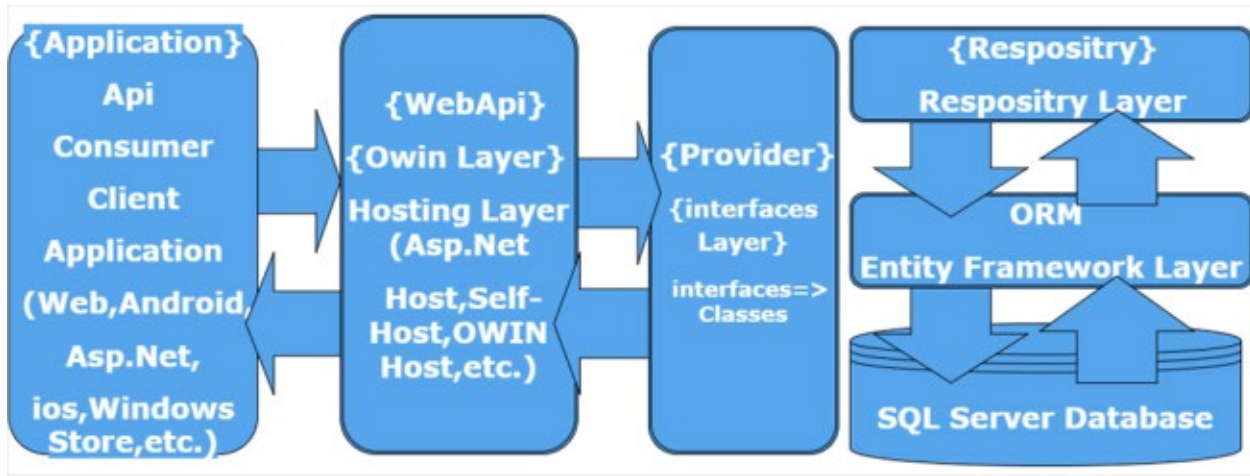
The client-side (data display) developer, for example, must adjust the HTML and CSS code so that the server-side developers' code in the same file does not break. Furthermore, when a database developer modifies an application's schema, the server-side developer may be required to update a large amount of code in order for the programme to function. The most important thing to remember is that display must be maintained separate from functionality and data storage in an application.

There are a number application design paradigms and patterns that can help, but the MVC pattern is the most prevalent right now. We conducted the research using our experience building web apps utilising various systems and frameworks, and we aimed to highlight both the benefits and cons of these systems while presenting our own suggestions for how they could be

improved. The frameworks and systems explored include Symfony, CakePHP, CodeIgniter Zend Framework, Laravel, Fuel PHP, Ruby on Rails, and ASP.NET MVC.



### 5.3 APPLICATIONS



- Model-View-Controller is an acronym for Model-View-Controller, which is a computer language. Model View Controller (MVC) is a lightweight and test-driven open source framework for ASdwP.NET that uhgufigeuig consists of three components: Model, View, and Controller. Model, View, Controller (MVwdC) is a web application desigdwn that is made up of three parts: Model, View, and Controller. The MVC model is divide into three parts. There are no drag-and-drop server controls, server events, or state management capabilities in the MVC approach. ASP.NETt MVC apps are created and developed using Microsoft Visual Studio and Microsoft Visual Web Developer Tools. The following are the three levels, or components, of the ASwdP.NET MVC programming model:Model
  - View
  - Controller

#### Different Versions of ASP.NET MVC:

There are several versions of the MVC programming model in ASP.NET, including MVC1, MVC2, MVC3, MvVC4, MvVC5, and MVC6. Each MVC version has its own set of programming templates. The MVC framework was launched in March 20099, and it works with Visual Studio 2008 and Visual Studio 2008 SP1. The MVC2 was launched in March 2010 and is compatible with both Visual Studio 20088 and Visual Studio 2010. The MVC3 was introduced in January 2011 and uses Visual Studio 20110 to run on the.NETt Framework 4.0. Different

View Engines are supported by the MVC3 pattern. The Razor View Engine is the MVC3 pattern's default View Engine.

MVC4 was published in August 2012, and it works with Visual Studio 2010 SP1 and Visual Studio 2012, and it operates on the .NET Frameworks 4.0 and 4.5. Different View Engines are supported by the MVC4 pattern. The Razor View Engine is the MVC4 pattern's default View Engine. Empty, Basic, Internet Application, Intranet Application, Mobile Application, and Web API are some of the programming templates provided by MVC4 for developing Web applications. MVC5 was launched in October 2013 and works with Visual Studio 2013 and the .NET Framework 4.5 and 4.5.1.

### **Designing Web Applications using ASP.NET MVC 5:**

1. MVC Web applications are available in several flavours, including MVC1, MV2, MV3, MVC4, and MVC5. Each version of the MVC framework has its own set of programming templates. The MVC5 programming model requires the .net framework 4.5 or later version. To construct an ASP.NET MVC5 Web application with Microsoft Visual Studio.net , follow these instructions.
2. Microsoft Visual Studio.NET 2015 Enterprise Edition should now be open.
3. Select New then Project from the File menu in Visual Studio.NET.
4. When we choose Project, the New Project dialogue box appears. Three panes or columns make up the New Project dialogue box. The New Project dialogue box's first pane displays a list of programming languages supported by Microsoft Visual Studio.NET.
5. From the first pane of the New Project dialogue box, we select a programming language for the ASP.NET MVC Web Application.
6. We choose ASP.NET Web Application from the second pane of the New Project dialogue box after selecting a programming language.
7. We select a target Framework for application development from the drop-down list above the second Pane of the New Project dialogue box after selecting ASP.NET Web Application. It should be noted that the MVC Web application template in Microsoft Visual Studio.NET 2015 is only accessible in the .NET Framework 4.5 and later versions; consequently, when creating a new ASP.NET MVC Web application project, always select the .NET Framework 4.5 or later version.

8. Using the Name text field in the New Project dialogue box, we give our new ASP.NET Web Application project a name. Under the first Pane of the New Project dialogue box, the Name text box appears.
9. We now use the Location drop-down list presented beneath the Name text box of the New Project dialogue box to pick a location where to save our Web Application project after giving it a name. We type the location string in the Location text box to manually specify a location; alternatively, we utilise the Browse button of the New Project dialogue box to find a defined location on a computer system for our new MVC Web application project.
10. Using the Solution Name text box, we now give the Web application project's solution a name. Under the Solution drop-down list, the Solution Name text box appears. If we don't give the solution a name, Microsoft Visual Studio.NET will use the project name as the solution name by default.
11. Now, in the New Project dialogue box, press the OK button. When we press the OK button, a new window appears with various project templates, like Empty, Web Forms, MVC, Web API, Single Page Application, and so on. Press the OK button after selecting the MVC template.

### **Folders and Files Structure of MVC 5 Web Application:**

The default directories and files are produced when you establish a new asp.net mvc Web application project. All Web content files or Web pages, such as asp.net and HTML files, Source files, Assemblies files such as.dll and.exe files, Data Source files (.mdb files), References, and other files such as Images and Configuration files, are stored in the default folders and files. The Solution contains the default folders and files that form up the entire structure of the asp.net mvc Web application. An asp.net MVC Web application's basic framework is made up of default directories and files, but we can customise it by adding extra folders and files.

The default folder and file structure makes working with the ASP.NET MVC Web application easier. For the complete asp.net MVC Web application project, Microsoft Visual Studio.NET creates a single project file that includes information about the project, such as the list of files and directories included in the project and any other references that reference other projects. The

project file extension is determined by the programming language used to create the ASP.NET MVC Web application. The project file extension will be .ccsproj if you are using the C# programming language, and .vbproj if you are using the Visual Basic programming language.

By default, the asp.net mvc5 Web application creates and shows the following Folders and Files in Microsoft Visual Studio Solution Explorer window. The mvc5 application's folders and files are listed below:

- Properties
- References
- App\_Data
- App\_Start
- Content
- Controllers
- Fonts
- Models
- Scripts
- Views
- Global.asax
- packages.config
- Web.Config

**REFERENCES:**

1. Abdul Majeed, Ibtisam Rauf. MVC Architecture: A Detailed Insight to the Modern Web Applications Development. Peer Rev J Sol Photoen Sys .1(1). PRSP.000505. 2018.
2. Selfa DM, Carrillo M, Boone MDR (2006) A database and web application based on MVC architecture. 16th International Conference on Electronics, Communications and Computers (CONIELECOMP'06), p. 48.
3. Ahmed M, Uddin MM, Azad MS, Haseeb S (2010) MySQL performance analysis on a limited resource server. SpringSim 10 Proceedings of the 2010 Spring Simulation Multiconference, p. 1.
4. Stratmann E, Ousterhout J, Madan S (2011) Integrating long polling with an MVC framework. Stanford University, USA, p. 10.
5. Shu-qiang H, Huan-ming Z (2008) Research on improved MVC design pattern based on struts and XSL. 2008 International Symposium on Information Science and Engineering, pp. 451-455.

Plag