Synopsis

On

PHP MVC Framework

Submitted in Partial Fulfilment of the Requirements

For the Award

Degree of B.Tech.

In Computer Science Engineering

To

Guru Gobind Singh Indraprastha University, Delhi

Under the guidance of

Mr. Gourav Sharma

Submitted By:

ASHISH KUMAR SINGH {06425602714}

DEEPAK CHOPRA {06325602714}

Affiliated to GGSIP University, New Delhi
Approved by AICTE & Council of Architecture

Computer Science Engineering

DELHI TECHNICAL CAMPUS

KNOWLEDGE PARK-III, GREATER NOIDA,

MAY- 2018

TABLE OF CONTENT

INTRODUCTION

The project title "**PHP MVC Framework**" is actually a software framework project. A software framework is a universal, reusable software platform used to develop applications, products and solutions. The MVC framework is a well-established design pattern that is widely used for applications with a user interface component .Then this project also includes this feature. This type of framework avoids the unauthorized access and it authenticates the authorized users or hosts.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows. Although originally developed for desktop computing, MVC has been widely adopted as an architecture for World Wide Web applications in major programming languages. Several web frameworks have been created that enforce the pattern. These software frameworks vary in their interpretations, mainly in the way that the MVC responsibilities are divided between the client and server.

The fundamental purpose of designing this project is to provide a structure and some inbuilt functions that are required by every web application in order to function optimally. Web applications share some common data amongst themselves through API that can be added in this framework. The main reason to develop this framework was to provide just the basic functionalities not any extra classes that might slow down the overall process cycle of request-response. The idea for choosing the mvc pattern was that it is easily upgradable and the user can add its own functionalities and its own libraries and usually can even improve the pre-built core files. The Framework has got a built in functionality to load all the classes automatically and the user does not need to add any include files on its own.

OBJECTIVES

The main objective of this project is to create a web development framework that provides all the essential functionalities and libraries to kick start a project easily.

Customers Intended

Any web application or website project that are intended for a large scale users and can be scaled to even more users.

Features

Following are some important features that come along with this project these are:-

- The application built on framework is secured.
- It is easy to understand the whole structure.
- Easy to troubleshoot errors because of its easy understanding.
- We can extend the modules.
- Separate backend logic.
- Separate frontend logic.

Assumption

There are some assumptions that are needed to be taken before studying the project that are:

- MVC architecture pattern and its working knowledge.
- Most of the web applications need to follow some structure in order to easily deploy.
- Basic knowledge of web applications html, css and php is required for this framework.

TOOLS AND SOFTWARE USED

Functional Requirements for this project are as follows:

Hardware Requirement:

- Processor: Intel i3 and above
- Processor Speed: 1.4 GHz and up
- System Memory: 512MB recommended
- RAM: 512MB and up
- Hard Disk: 100GB
- Monitor: HP 22es
- Keyboard: 104 US keys
- Mouse: Laser
- Higher configurations are also supported

Software Configuration:

- Front end tool: Angular or Pure JavaScript or HTML + CSS combination
- Backed: PHP and htaccess files for apache
- Operating System: Windows 7 and up.

MODULARIZATION OF PROJECT

This project consist of three main modules basically model, controller and views part and each one of them is important to the application. The purpose of modules and frameworks is not so much to achieve independence of functions as it is to achieve reusability and flexibility. In most other occupations, this kind of reuse of function and flexibility is central to achieving success.

Think about how difficult it would be to work on your car if you had to buy a whole new tool set for each car, because the tools were not interchangeable. Or worse, you have to build the tools from scratch, because there wasn't a factory to create them. Both of these concepts (universal plug ability and object factories) are present in well-written modules and frameworks.

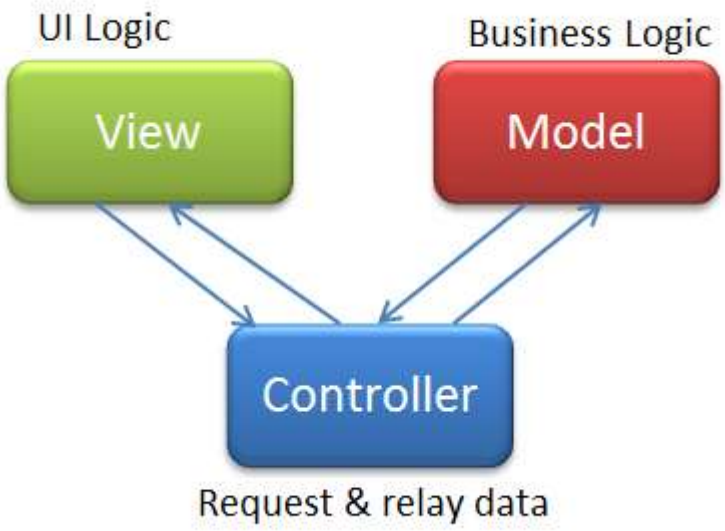


FIGURE 1: MVC CYCLE

MVC is a pattern for the architecture of a software application. It separates an application into the following components:

- **Models** for handling data and business logic.
- **Controllers** for handling the user interface and application.
- **Views** for handling graphical user interface objects and presentation.

- **FULL REQUEST RESPONSE CYCLE**

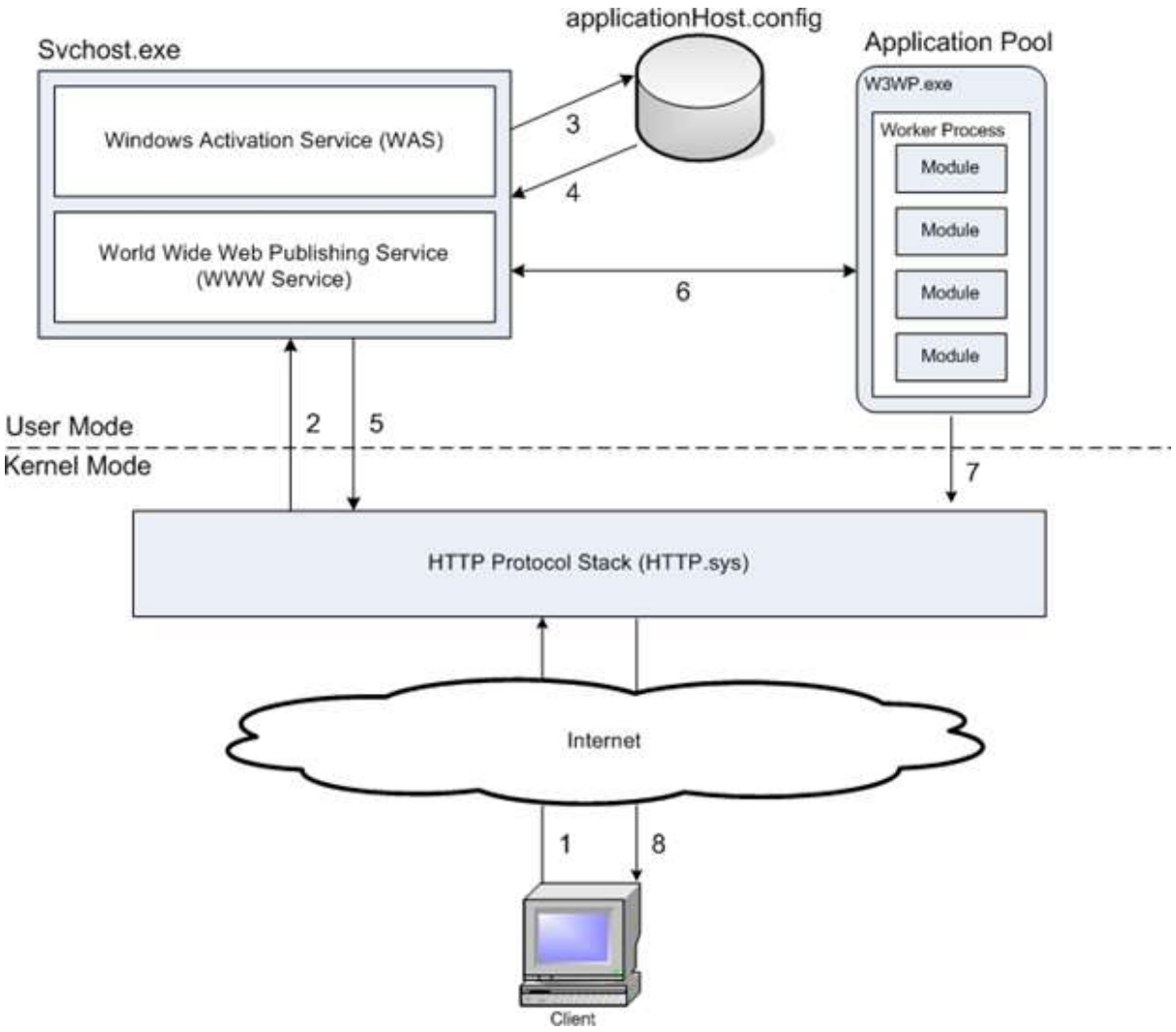


FIGURE 2: REQUEST – RESPONSE CYCLE

To Understand the MVC request execution Process, we have to start from the very beginning which is IIS. Let's look at it from an IIS perspective now. As MVC is sitting on top of framework, so from the user request intercept by HTTP.sys to worker process processes the request stays the same as web application. We will concentrate the part where the HTTP request enters the MVC Framework.

- **SEQUENCE**

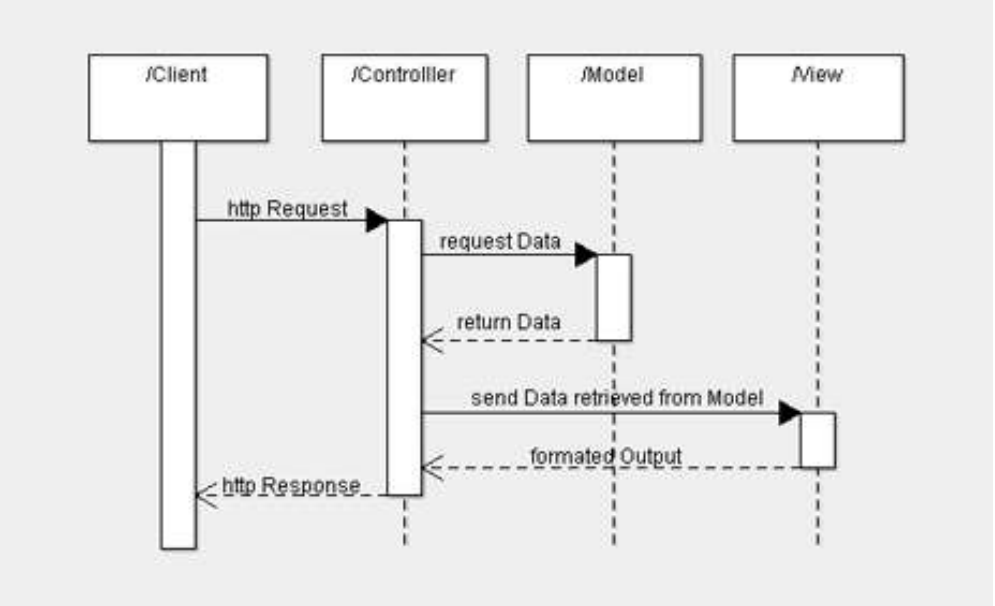


FIGURE 3: HTTP SEQUENCE

- ## Model and Entity Classes

The Model represents the data and the logic of an application, what many calls business logic. Usually, it's responsible for:

- Storing, deleting, and updating the application data. Generally it includes the database operations, but implementing the same operations invoking external web services or APIs is not an unusual at all.

- Encapsulating the application logic. This is the layer that should implement all the logic of the application. The most common mistakes are to implement application logic operations inside the controller or the view (presentation) layer.

PROJECT PLANNING OR SCHEDULING

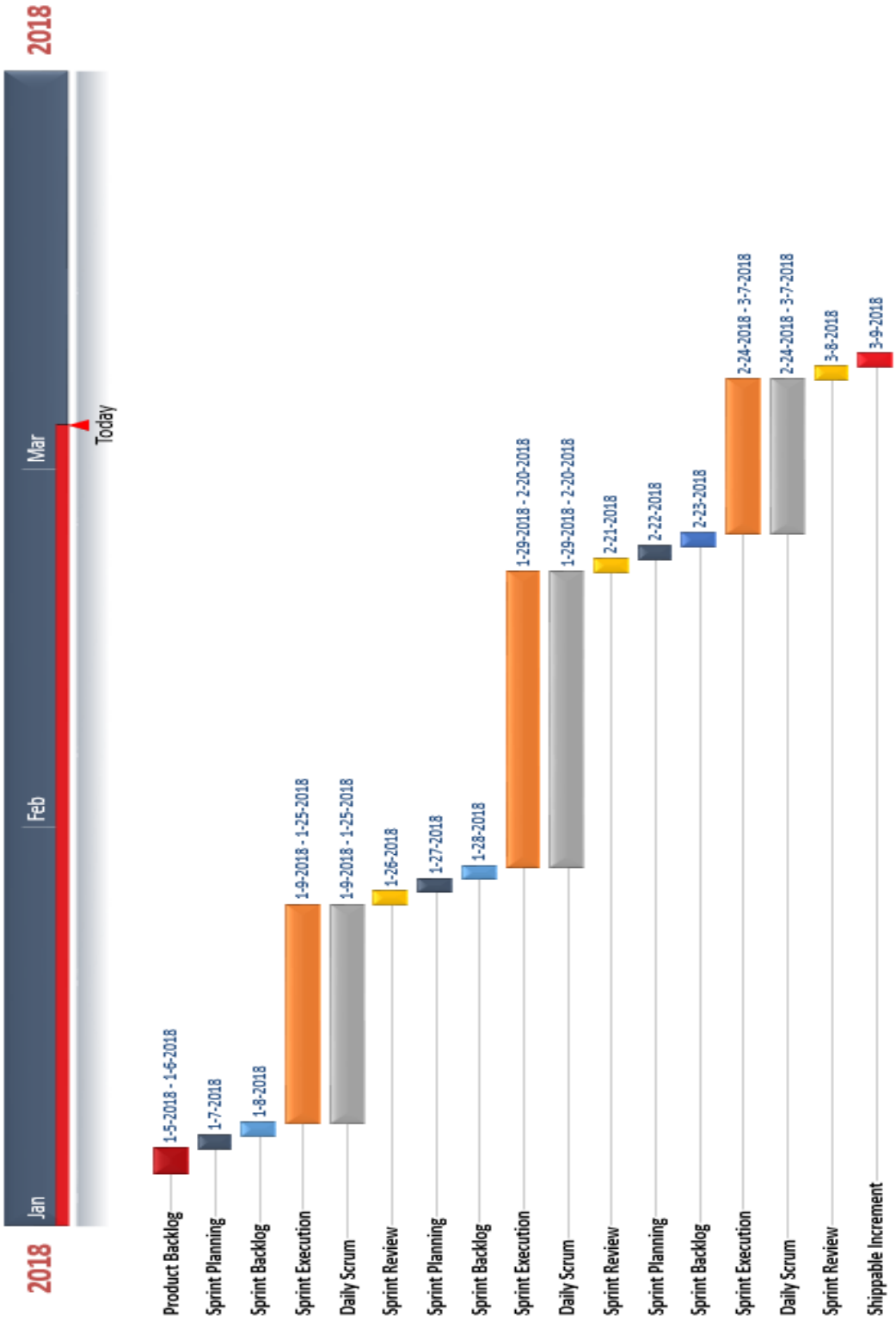


FIGURE 4 : GANTT CHART

In Figure 4, Gantt chart uses a Software Development methodology among **Agile** Methodology, knows as **Scrum.**

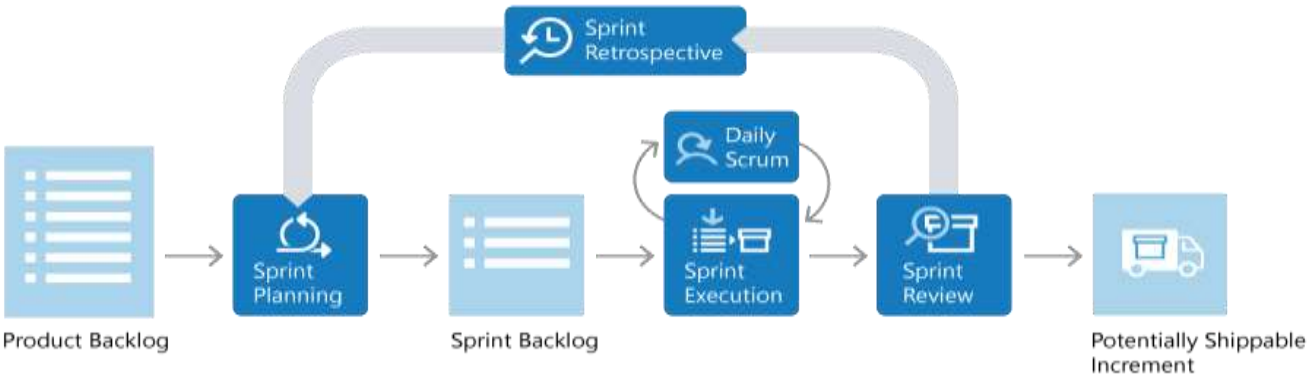


FIGURE 5 : FLOW DEPICTION OF SCRUM

Product Backlog

The Product Backlog is a prioritized list of value the team can deliver. The Product Owner owns the backlog and adds, changes, and reprioritizes as needed. The items at the top of the backlog should always be ready for the team to execute on.

Sprint Planning and Sprint Backlog

In Sprint Planning, the team chooses the backlog items they will work on in the upcoming sprint. The team chooses backlog items based on priority and what they believe they can complete in the sprint. The Sprint Backlog is the list of items the team plans to deliver in the sprint. Often, each item on the Sprint Backlog is broken down into tasks. Once all members agree the Sprint Backlog is achievable, the Sprint starts.

Sprint Execution and Daily Scrum

Once the Sprint starts, the team executes on the Sprint Backlog. Scrum does not specify how the team should execute. That is left for the team to decide. Scrum defines a practice called a Daily Scrum, often called the Daily Standup. The Daily Scrum is daily meeting limited to 15 minutes. Team members often stand during the meeting, to ensure it stays brief. Each team member briefly reports their progress since yesterday, the plans for today, and anything impeding their progress.

Sprint Review

The team demonstrates what they’ve accomplished to stakeholders. They demo the software and show its value.

Sprint Retrospective

The team takes time to reflect on what went well and which areas need improvement. The outcomes of the retrospective are actions for next sprint.

Shippable Increment

It should meet all the quality criteria set by the team and Product Owner so that it is of shippable quality and ready to dispatch to the client/customer.

ER DIAGRAM

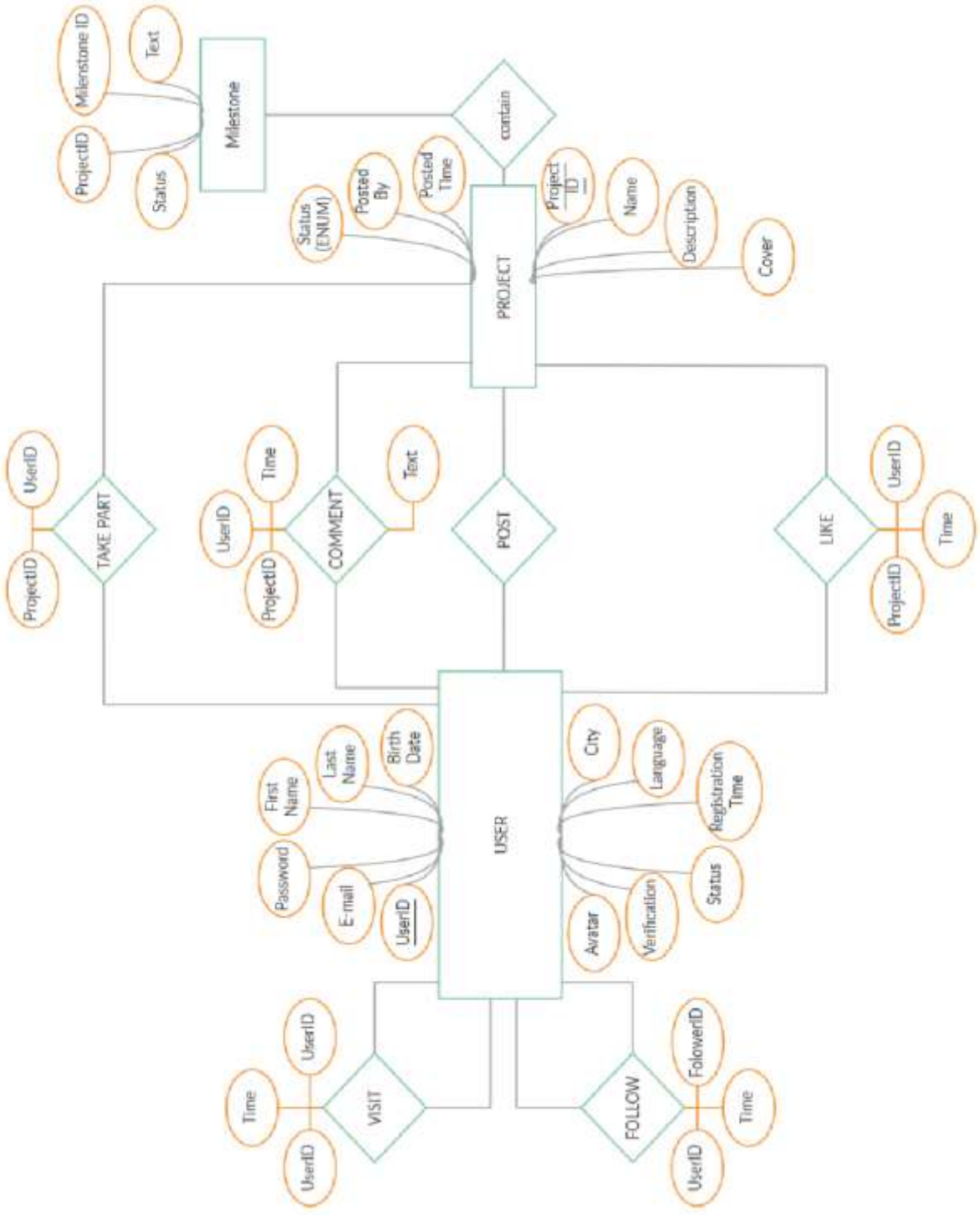


FIGURE 6 : ER DIAGRAM

Entities with their corresponding attributes:

1. **User**
   Contains user details

- User id, Email, Password, First name, Last name, Avatar, City, Language
- Follow (further connected to User entity) has user id, time, and follower id.
- Follow (further connected to User entity) has user id, time, visit id

2. **Project**
   It contains the project details.

- Status, posted by, time, project id, name, description.
- Milestone (further connected to project entity)

3. **Comment**
   It contains details of comment on projects.

- Project id , user id, time, text
- Comment is connected to project and user both

4. **Like**
   Whether a user likes a project or not.

- User id, project id, time
- Connected to user and project

5. **Visit, Take Part, follow**
   Are entities that contain the information on whether a user visits a project or likes or is a part of that project

- Contains user id , project id and time
- Is usually connected to other entities by the primary key of user entity and project entity.

FUTURE SCOPE

The scope of this project is to have a secure php framework that can help in easier and faster development of the web applications that can be easily deployed to the web servers and can be scaled even further for the use of many users.

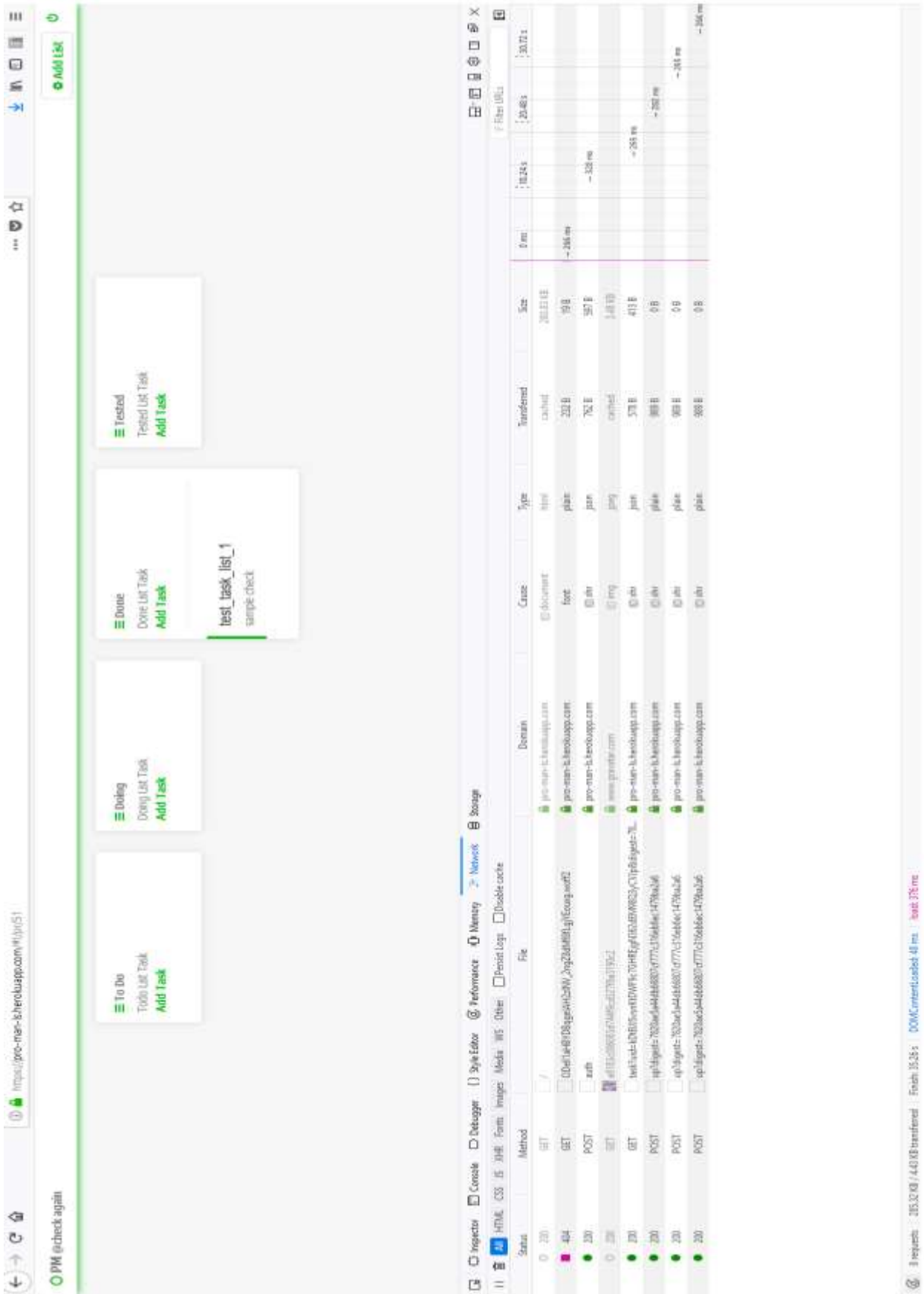
PHP is one of the most common server-side scripting language generally known for its simplicity, allows to developers to create dynamic web applications easily. PHP has their many frameworks with pre-build modules that gives to developers a platform to create robust, reusable components, and by enabling faster development cycles.

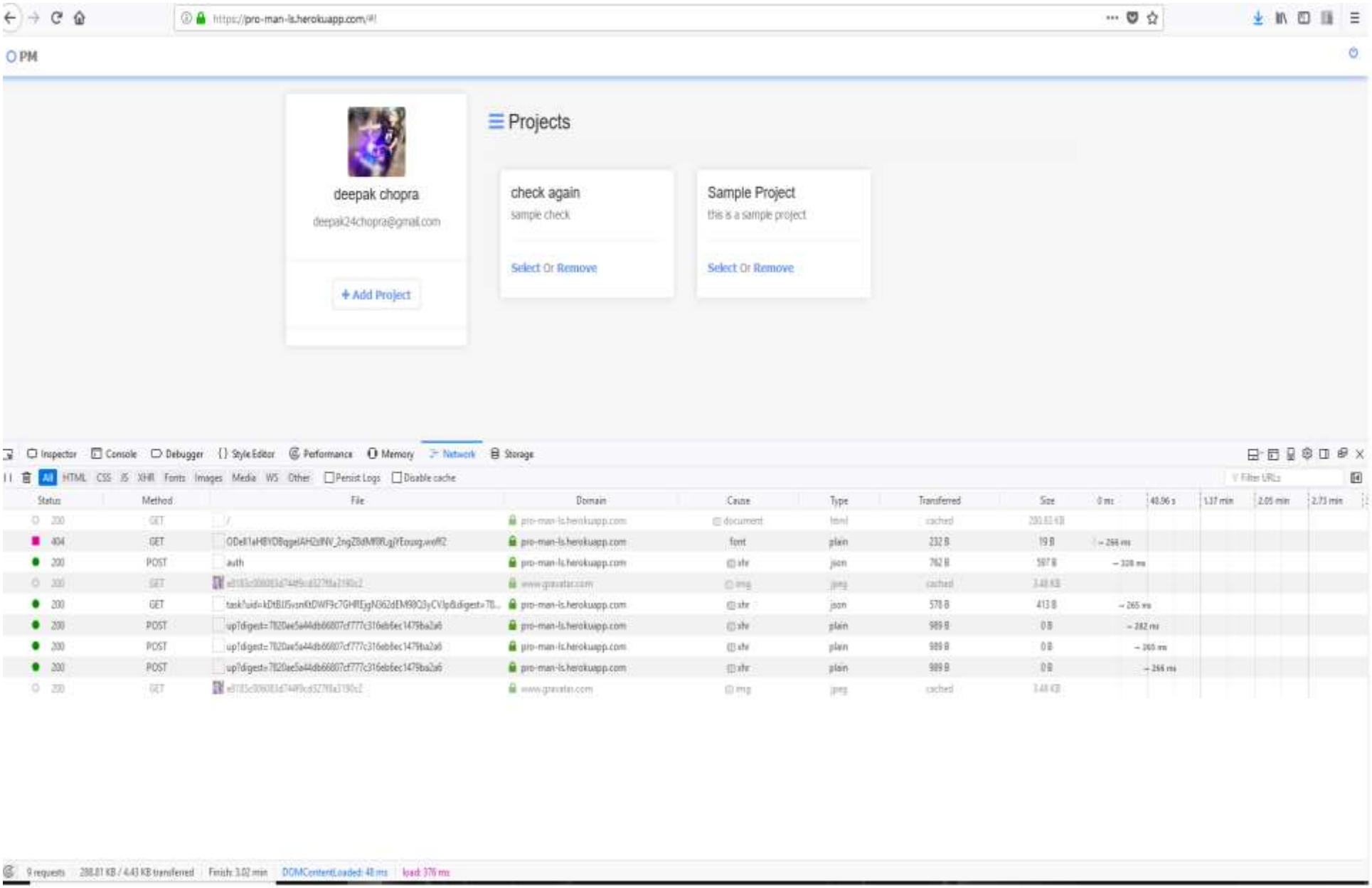
These frameworks use coding standards and development guidelines so help in standardizing the process and stabilizing the product. All PHP frameworks use Model View Controller (MVC) architecture, where the development of the business logic is independent and view and models are separately interlinked with each other.

Advantages of using frameworks:-

1.) Excellent community support.
2.) Easy to upgrade and maintain the developed applications.
3.) Default Security features such as CSRF Protection and Output Encoding.
4.) Rapid development using pre-built libraries and frameworks.
5.) Very strong encryption packages.
6.) Totally developer friendly doesn't need any special dependencies or supports
7.) Outperforms most other frameworks (non-MVC)
8.) Good Documentation and easy to use

PROJECT SCREENSHOTS




PM
deepak chopra
deepak24chopra@gmail.com
+ Add Project
Projects
check again
sample check
Select Or Remove
Sample Project
this is a sample project
Select Or Remove
Inspector
Console
Debugger
Style Editor
Performance
Memory
Network
Storage
Status
Method
File
Domain
Cause
Type
Transfered
Size
9 requests
288.81 KB / 4.43 KB transfered
Finish: 3.02 min
DOMContentLoaded: 48 ms
load: 376 ms

BIBLIOGRAPHY

Books:

- PHP 7 packt publishing
- Internetworking Technologies - An Engineering Perspective (2002)
- Apache Cookbook

Web Sites:

- http://www.lynda.com (For PHP tutorials)
- https://www.wamp.com (Windows apache mysql php)