MVC

16/1/2020

Overview

In this project a simple php webpage is deployed to understand the framework of MVC in depth. The idea of this pill is to ensure complete if not all understanding of MVC and make its use in future projects seamlessly.

Lessons learned

MVC (Model-View-Controller) is a software architecture pattern, which separates the data and business logic of an application from its representation and the module responsible for managing events and communications.
Some web development frameworks such as AngularJS, Spring, Symfony or Laravel make use of this pattern to organize the resources of the application. This is because it is a very popular and easy to implement pattern. Thanks to this pattern we manage to obtain an architecture that allows us to organize our project resources in a scalable way.

- MVC is a pattern or way to organize your files, which is commonly used for developing user interfaces.
- MVC allows for code reuse and parallel development. It also makes easier to find errors or programs in the project.
- MVC stands for Model-View-Controller. Under this pattern, projects are organized in these three major components. The Model is the central component of the pattern. It manages the data and logic of the application. The View component is the visual aspect of the webpage, for example tables or diagrams. The Controller like the name implies, allows the user to control the application. It accepts inputs.

- .htaccess is a configuration file for Apache which allows you to make configuration changes on a per-directory basis.
- Php allows you to create and interact with Databases.
- MySQL is the most used database with PHP.
- The data in MySQL is stored in tables. A table is a collection of related data, and consists of columns and rows.
- PDO stands for Php Database Object, It is a way to access databases in PHP.
- The image below helps to clarify how the MVC model works

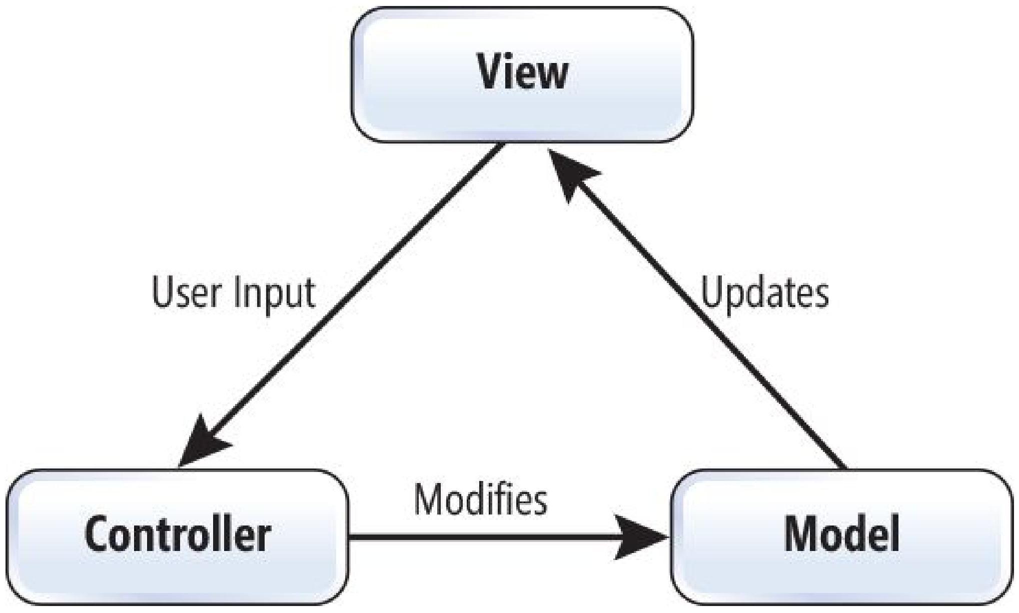


Table of Contents

9. Technologies used
10. Incidents

Project requirements

- Implement the MVC pattern using PHP without any library
- You must have a clear directory structure to be able to implement the pattern
- You must have a minimum of:
    - 2 controllers
    - 2 data models that obtain information from a database (you can use MySQL or local file)
        - You must attach a copy of the database in the repository of your project with the necessary information to carry out the tests.
    - 2 view
    - 2 URL's that demonstrate that the pattern is implemented correctly and are capable of displaying the data obtained from the source in a view.
        - URL's can be controlled by:
            - URL parameters
            - mod_rewrite (you have to take into account that this option requires the use of the ".htaccess" file

- Create a clear and orderly directory structure
- The main page should show the URL's links with which you can navigate to the other controllers
- You must create a main controller that is responsible for receiving the request from the main page
- Both the code and the comments must be written in English
- Use the camelCase code style to define variables and functions
- In the case of using HTML, never use online styles
- Remember that it is important to divide the tasks into several sub-tasks so that in this way you can associate each particular step of the construction with a specific commit
- For the project documentation a PDF version is required within the repository
- You should try as much as possible, each commit is associated with a task
- Delete files that are not used or are not necessary to evaluate the project
- You must create a correctly documented README file in the root directory of the project (see guidelines in Resources)

Risk Management

Every project has risks. This risks must be taken into account to improve the workflow of the project. Managing these risks is important for due completion of project. Unchecked risks could not only lead to hamper of project deliverance but also a badly executed project. ***However, the risks have been greatly reduced after performing the last project of blog.*** The risks hence associated with the project are documented as follows:

Risk	Risk level
Unfamiliarity with MVC	Low
Getting request for the database	Medium to Low
Database design	Low
Using SQL for the database	Low
Delivering in time	Medium
Low experience with SQL	Low
Completing all the project requirements	Medium
Unfamiliarity with MVC	Low
Internet access and workplace environment	Medium

Task Management

The following are the tasks which needed to be accomplished for the completion of the project. The tasks have been divided according to the project specifics. Each tasks have been clearly defined and their priority as well as their difficulty and time needed. Difficulty level is explained on a scale of 1 to 5 (5 being the most difficult). Priority is explained on the level of 1 to 5 (again 5 the highest parameter being the most prioritized work).

Task	Priority	Description	Difficulty	Time

Read the description of the project	5	This task involves complete understanding of requirements for the project	1	30 min
Create git repo	5	Creating of git repository for project execution and delivery	1	2 mins
Create files for the project	4	Creating the necessary files, folders and subfolders for this project. This is important especially for this project of MVC	1	5 mins
Create header and footer view	3	Creating a header.php and footer.php for this project which is used in all the views	2	10-20 mins
Create SQL database and tables	4	A database in Xampp and corresponding tables are created for use in this project	3	30 mins
Create 2 controllers	2	To create the required controllers.php for connecting the models and views	4	1 hr
Create 2 models	2	Create the models which will interact with the data	4	1-1:30 hr
Create 2 views	2	The two views that would be shown to the user	4	30mins
Create the libs folders and files	3	This important for the project to function fully with interconnecting the controllers, models and views	3	2hrs
Get the data	5	To get the data and show to the user	4	1 hr
Update/insert data of salary as a 2nd mode	1	To change the data once the user clicks a button	5	1 hr

Defining this part is crucial to the development of the project. It is important to make a good analysis of the situation to organize the project in a good way.

Chronogram

This shows the time-line it took for the project to finish and also the tasks that were achieved during those timeline.

The tasks

Tasks	Thur 1000 hrs	Thur 1100 hrs	Thur 1200 hrs	Thur 1300 hrs	Thur 1400 hrs	Thur 1500 hrs	Thur 1600 hrs	Thur 1700 hrs
Read project descriptio-n	X							
Git Repo	X							
Create files	X							
Header and footer		X						
SQL dB and tables		X						
controllers			X					
Models				X	X	X		
views						X	X	
Get data							X	X

Update data								**X**

Git Workflow

For this project, all commits we'll be pushed directly to the Master branch. All commits will use a descriptive message, so that the admin and other internet user can follow through the processes.

Incidents

- None

Technologies used

For this project, we will use the following technologies:

- HTML
- CSS & Bootstrap
- Php
- MySQL
- Lighthouse extension for auditing the website

Project specific questions

1. What is an architecture pattern?

An **architectural pattern** is a general reusable solution to a commonly occurring problem in software architecture within a given context.] Architectural patterns are similar to software design patterns but have a broader scope. The architectural patterns address various issues in software engineering, such as computer hardware performance limitations, high availability and minimization of business risk. Some architectural patterns have been implemented within software frameworks. The following are the types of architecture patterns in software developement:

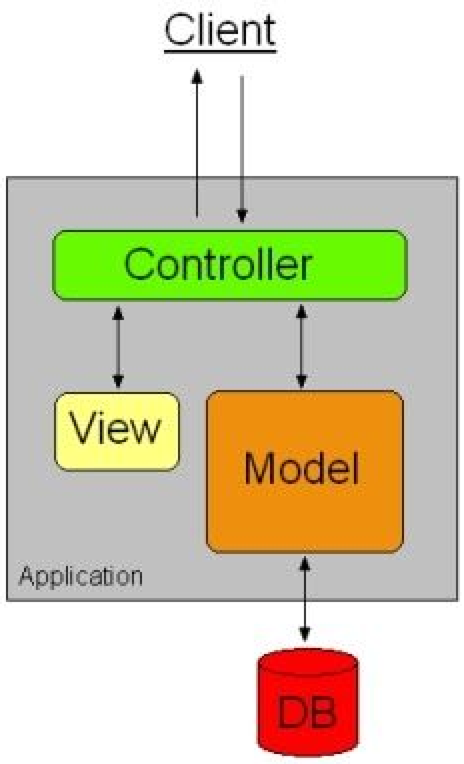
- The layered architectural pattern
- The microkernel architectural pattern
- The CQRS Architectural Pattern
- The Event Sourcing Architectural Pattern
- The Microservices Architectural Pattern

2. What is MVC pattern?

MVC Pattern stands for Model-View-Controller Pattern. This pattern is used to separate application's concerns.

a. Model - Model represents an object or JAVA POJO carrying data. It can also have logic to update controller if its data changes.
b. View - View represents the visualization of the data that model contains.
c. Controller - Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate

3. Draw MVC pattern scheme?



4. Explain where MVC pattern is used?

   At the heart of MVC is what I call Separated Presentation. The idea behind Separated Presentation is to make a clear division between domain objects that model our perception of the real world, and presentation objects that are the GUI elements we see on the screen. Domain objects should be completely self contained and work without reference to the presentation, they should also be able to support multiple presentations, possibly simultaneously.

5. What advantages does this pattern have?

   1. Faster development process:

   MVC supports rapid and parallel development. If an MVC model is used to develop any particular web application then it is possible that one programmer can work on the view while the another can work on the controller to create business logic of the web application. Hence this way, the application developed using MVC model can be completed three times faster than applications that are developed using other development patterns.

2. Ability to provide multiple views:

In the MVC Model, you can create multiple views for a model. Today, there is an increasing demand for new ways to access your application and for that MVC development is certainly a great solution. Moreover, in this method, Code duplication is very limited because it separates data and business logic from the display.

3. Support for asynchronous technique:

The MVC architecture can also integrate with the JavaScript Framework. This means that MVC applications can be made to work even with PDF files, site-specific browsers, and also with desktop widgets. MVC also supports asynchronous technique, which helps developers to develop an application that loads very fast.

4. The modification does not affect the entire model:

For any web application, the user interface tends to change more frequently than even the business rules of the .net development company. It is obvious that you make frequent changes in your web application like changing colors, fonts, screen layouts, and adding new device support for mobile phones or tablets. Moreover, Adding new type of views are very easy in the MVC pattern because the Model part does not depend on the views part. Therefore, any changes in the Model will not affect the entire architecture.

5. MVC model returns the data without formatting:

MVC pattern returns data without applying any formatting. Hence, the same components can be used and called for use with any interface. For example, any kind of data can be formatted with HTML, but it could also be formatted with Macromedia Flash or Dream viewer.

6. SEO friendly Development platform:

MVC platform supports the development of SEO friendly web pages or web applications. Using this platform, it is very easy to develop SEO-friendly URLs to generate more visits from a specific application. This development architecture is commonly used in the Test-Driven Development applications. Moreover, Scripting languages like JavaScript and jQuery can be integrated with MVC to develop feature-rich web applications.