Alfonso De La Guardia Prattya Datta

Blog Project

Due 1/14/2019

Overview

In this project you must create a simple blog through custom programming. This project is a test of the knowledge gained from previous projects and implement them for this new project. It is required that the project is completed and delivered in time encompassing all the requirements.
This is a simple blog demanded by the client to publish contents in a simple way without the use of CMS.

To carry out the project, it is necessary that there is a smooth connection between the front-end and the back-end. The above two main roles are briefly described as:

- The backend system with PHP that is responsible for obtaining the information from the database
- The frontend system that will be responsible for consuming the data provided by your own backend system.

It is important to keep in mind that the project has two FRONTS, which are necessary for the blog posts to work well and according to examples available on the web. This documentation is developed into the following categories

Table of Contents

1. Project requirements
2. Project specifications
3. Risk management plan
4. Tasks for the project
5. Chronogram
6. Calendar
7. Git Workflow
8. Technologies used
9. Lessons learned
10. Incidents

Project Requirements

- You must use GIT from the beginning of the project
- For the PHP development part you should not use any framework
- You will need to use MySql to manage the database
- All code must be properly documented
- You must include a small user guide to understand how to interact with the tool
- The interface must be fully responsive
- All code included comments need to be write in English
- Use a code style like camelCase
- In the case of using HTML never use inline styles
- In the case of using different programming language always define the implementation in separate terms
- It is recommended to divide the tasks into several subtasks so that you can associate each particular step of the construction with a specific commitment.
- For the project documentation a PDF version is required within the repository
- You should try as much as possible that the commits and planned tasks are the same
- Delete unused files

Project Specifications

For the project to be completed according to the requirements mentioned in the document, it is necessary that it contains the following specifications. The interface with the two FRONTS should be able to:

- Login
- Logout
- List posts
- Be able to comment on a post anonymously
- Be able to share a post
- Be able to search for posts by their content
- Be able to assign keywords to each post
- Be able to organize posts by categories
- Be able to leave a post in a “draft” state (will not be visible to the user)
- Be able to indicate the publication date of a post (it will not be visible if the date is before the publication date)

And, the second control panel should consist of the following:

- Login

- Logout
- List posts
- Edit a post
- Delete a post
- Create a post
- Delete a comment
- Create Categories
- Edit Categories
- Remove Categories
- List Categories

MVC Architecture Pattern

The files of the Project were organized in the following way:

```
database/
src/
    assets/css
    components/
         backend/
              config/
                 config.php
             controllers/
                 aboutUs.php
                 categories.php
                 crteBlogController.php
                 errorShown.php
                 home.php
                 main.php
                 myBlogPtController.php
                 newModelController.php
                 updteDelController.php
          libs/
                 app.php
                 controller.php
                 createDb.php
                 model.php
                 view.php
          models/
                 blogData.php
                 crteBlog.php
                 myBlogPt.php
                 newModel.php
                 updteDel.php
```

```
 views/
  aboutUs/
            index.php
  categories/
            index.php
  errors/
            index.php
            loginError.php
            passwordError.php
   home/
            index.php
   main/
            index.php
   myBlog/
            create.php
            edit.php
            index.php
            myPage.php
    new/
            index.php
Footer.php
Header.php
frontend/
            login.js
            passwordCheck.js
 .gitignore
 .htaccess
 Index.php
 README.md
```

Risk Management Plan

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. The risks hence associated with the project are documented as follows:

Risk	Risk level
Unfamiliarity with MVC	Low
Getting request for the database	High
Database design	Medium to High

Using SQL for the database	Medium
Unfamiliarity with Webpack	Medium
Delivering in time	High
Adding user functionality in Frontend	Medium
Low experience with SQL	Low
Completing all the project requirements	High
Unfamiliarity with MVC	High
Internet access and workplace environment	Medium

Tasks for the project

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	1 hr
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	1	2 mins
Create user login and logout functionality	4	This involves creation of a php system which can perform smooth user and login functionality. It also checks password errors and login errors	5	3-4 hrs
Create functionality to list posts	5	This is related to the creation of a php system functionality that allows users to list posts as they go	4	4 hrs

Create functionality for user interaction with posts	6	This functionality will allow users to interact with posts, for example for editing and deleting	5	8 hrs
Create functionality to draft posts	5	This functionality will enable users to draft posts, so they can work on them before publishing	5	8-10 hrs
Design control panel	5	The control panel will give the user more functionalities to interact with the blog	6	8-10 hrs
Create the visual design for the website	5	The visual aspect of the website will make the website have a more appealing design for the user.	5	6 hrs

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

#	Tue	Wed	Thur	Fri	Sat	Sun	Mon	Tue
1	X							
2	X							
3	X							
4		X						
5		X						

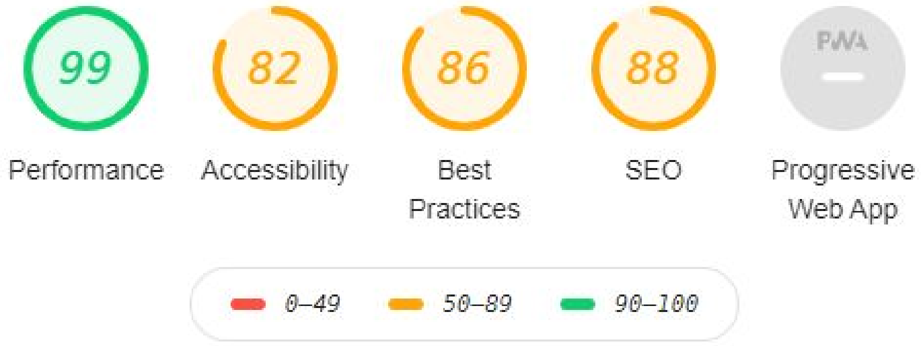
6			X					
7				X	X			
8					X	X		
9						X	X	x

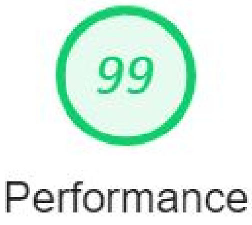
Git Workflow

For this project, all commits we'll be pushed directly to the Master branch. All commits will use a descriptive message, so that myself or other users can easily go to the Git version that they need to. This is very important for working in teams as it increases communication and efficiency between all members.

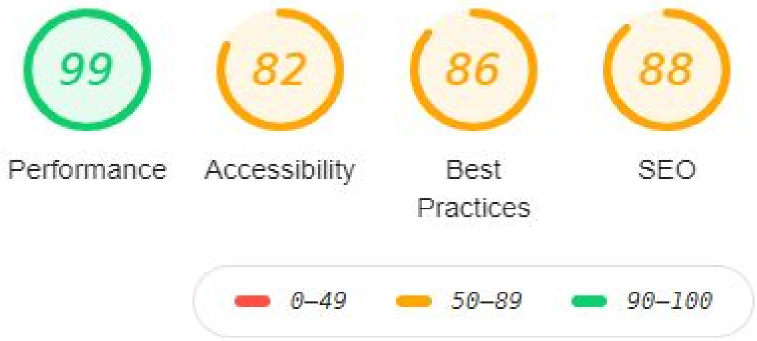
Quality Metrics

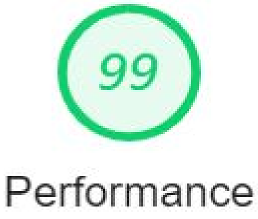
For this project we are using the Lighthouse Extension for Chrome. When analyzing the **Desktop Version** of the Webpage, it displays the following data:





For the **Mobile Version** of our Website, we got the following results, when using the Lighthouse Extension:





Technologies used

For this project, we will use the following technologies:

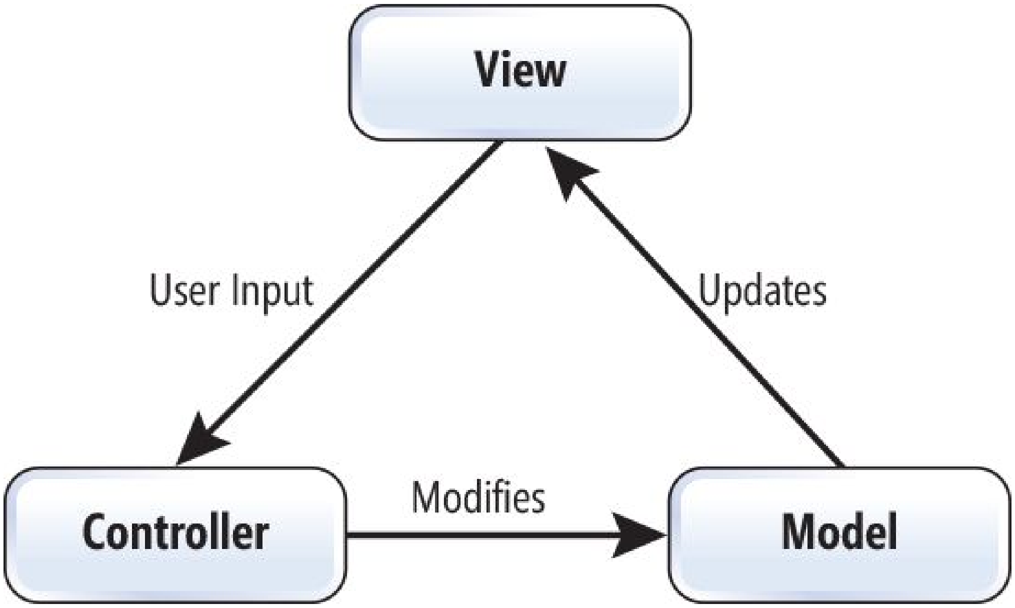
- HTML
- CSS & Bootstrap
- Php
- MySQL
- Lighthouse extension for auditing the website

Lessons learned

- 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:



Incidents

- Difficulty figuring out the best way to design the website.
- Difficulty conceptualizing the project and figuring out the best way to implement it.
- Some difficulty in interacting between PHP and MYSQL.
- Some difficulty of understanding how to combine Object-Oriented Programming with MVC.

Questions

Alfonso:

- ¿Mi compañer@ me a apoyado en el proyecto?

  Sí
- ¿Mi compañer@ a liderado el proyecto / ha dejado que yo lo lidere o hemos conseguido un equilibrio en el liderazgo en el cual a cumplido con su parte?

  -
- ¿Me ha ayudado cuando le he pedido ayuda?

  Sí
- ¿A sabido escuchar mis opiniones?

  Sí
- ¿Está abiert@ a escuchar otros puntos de vista?

  Sí
- Cede o siempre se tiene que hacer lo que el o ella diga aunque no lo tengamos claro por todas las partes?

  A veces cede
- Me siento apoyad@

  Sí
- Técnicamente tiene los conocimientos y en caso de no tenerlos es capaz de buscarlos y aprenderlos

  Sí

```
DOCUMENTATION": 3,
"ORGANIZATION": 1,
"HTML": 3,
"CSS": 2,
"PHP": 0,
"MYSQL": 0,
"MVC": 0,
"GIT": 3,
```

Prat:

- ¿Mi compañer@ me a apoyado en el proyecto?

  Sí

- ¿Mi compañer@ a liderado el proyecto / ha dejado que yo lo lidere o hemos conseguido un equilibrio en el liderazgo en el cual a cumplido con su parte?

  Ha liderado el proyecto
- ¿Me ha ayudado cuando le he pedido ayuda?

  Sí
- ¿A sabido escuchar mis opiniones?

  Sí
- ¿Está abiert@ a escuchar otros puntos de vista?

  Sí
- Cede o siempre se tiene que hacer lo que el o ella diga aunque no lo tengamos claro por todas las partes?

  A veces cede
- Me siento apoyad@

  Sí
- Técnicamente tiene los conocimientos y en caso de no tenerlos es capaz de buscarlos y aprenderlos

  Sí

DOCUMENTATION": 2,
"ORGANIZATION": 0,
"HTML": 1,
"CSS": 1,
"PHP": 12,
"MYSQL": 8,
"MVC": 8,
"GIT": 3,