

My Portfolio

28/1/2020

Overview

In this project a simple webpage showing all the requirements for a portfolio is developed. For this project, a personal portfolio is developed. It will also consist of easy addition and edition of the details. This portfolio is developed based on the previous knowledge and experiences gained during the course of other projects and courses.

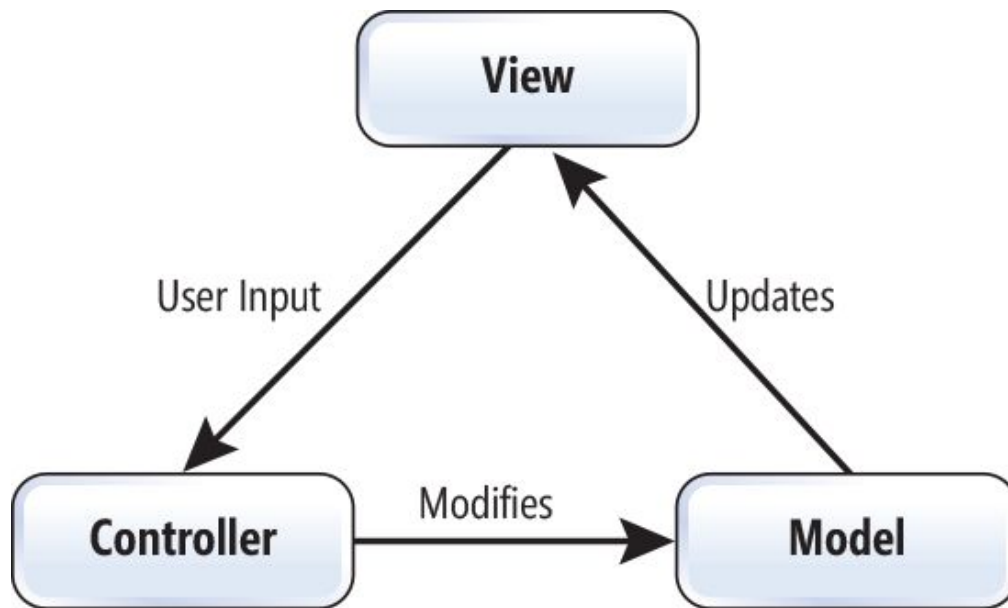
Lessons learned

For this project we use MVC. 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 Program Database Object, It is a way to access databases in an object oriented fashion
- The image below helps to clarify how the MVC model works



Furthermore, NodeJs (a programming runtime environment) and ExpressJs (a wrapper for Javascript development) will be used.

Table of Contents

1. Overview
2. Lessons Learned
3. Project requirements
4. Risk management plan
5. Tasks for the project

6. Chronogram
7. Calendar
8. Git Workflow
9. Technologies used
10. Incidents

Project requirements

Create the necessary mockups to discuss how your user interface will be.

You can be inspired by online portfolios that you consider more appropriate.

At least the interface should have the following sections, but it is important that you contribute what you consider appropriate, use your creativity!

1. About
 - Summary about you
 - At least a degree, a description, and a summary of technologies, soft skills and experience as a CV.
2. Soft skills
 - Soft Skills List
3. Technologies
 - List of technologies with their score according to your experience
4. Experience
 - List / Block showing the main information about each experience
 - It will have at least one title and one main block of text
5. Contact
 - Contact form that when sent enter the contact in the database and send by email. (use a GMAIL configuration to facilitate deployment and not rely on a mail server)

It is important that you review the GMAIL documentation as well as the fact that your account allows you to send emails from an app.

You will also develop a control panel to manage the previous sections that will cover the following needs:

1. Login to identify yourself in the control panel
2. Logout
3. Modify summary information about yourself (About)
4. CRUD of the technologies
5. CRUD of experiences
6. CRUD of soft skills

7. List contact requests that have been made
-

Risk Management

Every project has risks. These risks must be taken into account to improve the workflow of the project. Managing these risks is important for the due completion of the project. Unchecked risks could not only lead to hamper the project deliverance but also a badly executed project. The risks hence associated with the project are documented as follows:

Risk	Risk level
Unfamiliarity with NodeJs	Very High
Getting request for the database	Medium to High
Database design	Low
Passport login system	Very High
Delivering in time	Medium
Unfamiliarity with ExpressJs	High
Completing all the project requirements	Medium
Health issues	High
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 task has 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 a passport login system	5	To create the required controllers.php for connecting the models and views	5	3-4hrs
Create the different views	3	Create different views	4	5-6 hr
Create admin control panel	4	The two views that would be shown to the user	4	3-4 hrs
Installing the required dependencies	5	This important for the project to function fully with interconnecting the controllers, models and views	3	2hrs
Get the data	1	To get the data and show to the user	4	1 hr
Update/insert data	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.

Glt 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
- NodeJs and Express JS
- MySQL
- Lighthouse extension for auditing the website