Portfolio

Overview

Nowadays having a portfolio in web format is a good way to make yourself known and so that everyone can have access to information about you in an easy and dynamic way.

In this project you must create a personal portfolio through NodeJS and ExpressJS.

Table of Contents

Project Requirements

- You must use GIT from the beginning of the project
- You must develop the project with NodeJS and ExpressJS
- You must 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 comments included in the code must be written in English
- Use the camelCase code style
- In the case of using HTML, never use online styles
- In the case of using different programming languages always define the implementation in separate terms
- It is recommended to divide the tasks into several subtasks 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 that the commits and the planned tasks are the same
- Delete unused files
- Try using a PHP Linter to make your code as concise and standard as possible.

Project Specifications

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
    1. Summary about you

      1. At least a degree, a description, and a summary of technologies, soft skills and experience as a CV.

2. Soft skills
   1. Soft Skills List
3. Technologies
   1. List of technologies with their score according to your experience
4. Experience
   1. List / Block showing the main information about each experience
      1. It will have at least one title and one main block of text
5. Contact
   1. 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

You can add new features if you consider it appropriate.

Risk Management Plan

Every project has risks. This risks must be taken into account to improve the workflow of the project. I've listed the risks for the project, along with the impact they might have, and the priority of them.

iD	Risk	Consequence	Prob. (1-5)	imp. (1-5)	Pri. (1-25)	Mitigation approach
1	Breaking my computer	Can't do anything	1	5	5	Keep my repo work to date, look for other computers
2	Getting sick	Wouldn't be as productive	2	3	6	Eat and sleep well
3	Not concentrating enough	Won't com	2	5	10	Focus on the Minimum Viable Product.
4	Unrealistic deadlines	Deadlines wouldn't be met + development shortcuts would have been taken affecting the robustness of the code	2	3	6	Be more organized with the tasks, set new deadlines.
5	Being unfocused	Loss of control over the development flow of the project	4	5	20	Focus on finishing the most important tasks only.

Project Tasks

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:

#	Task	Priority (1-5)	Description	Difficulty (1-5)	Time Estimation
1	Reading the description	4	Reading the description of the project	1	30 min
2	Create Repo	3	Creating git repo for the project	1	2 min

3	Create server	5	Creating server for the developmet of the project	3	30 min
4	MySQL Connection	4	Connecting wth MySQL	4	1 hr
5	Create Views	4	Creating the views of the project, like signup, signin, add, etc	3	1 hr
6	Obtaining data from MySQL	4	Obtaining data from the MySQL database	4	1 hr
7	Deleting data from MySQL	3	Deleting data from MySQL database	3	2 hr
8	Updating data from MySQL	3	Updating data from MySQL database	4	2 hr
9	Connect-flash messages	3	Implementing connect-flash messages	3	1 hr
10	MySQL user registration	4	Being able to register users to MySQL	4	2 hr
11	MySQL User Login	4	Creating login functionality for users	4	2 hr
12	Protecting routes from Server	2	Making information autochthonous to each user	4	1 hr
13	Adapting progress to other sections of the project	5	Adapting the progress for the rest of the sections of the project, such as technologies, experience, etc	3	6 hr
14	Working with nodemailer functionality	3	Working with nodemailer to allow Gmail requirement in the contact section	5	1 hr
15	Adding more styling to the project	2	Adding overall styles to the whole project	1	1 hr

16	Review	2	Review Project	2	30 min

Chronogram

Task	Mon	Tue	Wed	Thur	Fri	Sat	Sun
1	X						
2	X						
3		X					
4		X					
5		X					
6			X				
7			X				
8			X				
9				X			
10				X			
11				X			
12					X		
13					X	X	
14							
15						X	
16							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.

Technologies used

For this project, we will use the following technologies:

- **NodeJS**
- **ExpressJS**
- **Visual Studio Code**
- **Handlebars**
- **MySQL**
- **Google Docs**
- **Git**
- **Nodemailer**
- **Timeago.js**
- **MVC Pattern**

Lessons learned

- NodeJS is a runtime environment that executes JavaScript outside the browser
- NodeJS lets programmers use JavaScript to write command line tools and do server-side programming to produce dynamic web content before the page is sent to the user's web browser.
- ExpressJS is a web application framework for NodeJS, designed for building web applications and APIs.
- ExpressJS is considered the standard server for working with NodeJS.
- The MVC pattern is very useful for organizing your code, and finding bugs in your code.
- Middleware functions are the ones that have access to the request object, the response object, and the next middleware function in the applications request-response cycle.
- The *request object* in Express represents the HTTP request and has properties for the request query string, parameters, body, HTTP readers, and so on. This object is always referred to as "req".

- The *response object* in Express represents the HTTP response that an Express app sends when it gets an HTTP request. This object is always referred to as “res.
- The app object conventionally denotes the Express application. You can create it by calling the top-level express() function exported by the Express module.
- Connection pooling is a technique of creating and managing a pool of connections that are ready for use by any thread that needs them.
- The main benefits of connection pooling are: reduced connection time, simplified programming model and controlled resource usage.
- An async function can contain an await expression that pauses the execution of the async function and waits for the passed Promise's resolution, and then resumes the async function's execution and returns the resolved value. Remember, the await keyword is only valid inside async functions.
- Handlebars.js is an extension to the Mustache templating language created by Chris Wanstrath. Handlebars.js and Mustache are both logicless templating languages that keep the view and the code separated
- Modules are self-contained units of functionality that can be shared and reused across projects. We can use them to augment our applications with functionality that we haven’t had to write ourselves. They also allow us to organize and decouple our code, leading to applications that are easier to understand, debug and maintain.
- The CommonJS (CJS) format is used in Node.js and uses require and module.exports to define dependencies and modules. The npm ecosystem is built upon this format
- Res.redirect redirects to the URL derived from the specified path, with specified status, a positive integer that corresponds to an HTTP status code . If not specified, status defaults to “302 “Found”.
- Nodemailer is a module for Node.js applications to allow easy as cake email sending. The project got started back in 2010 when there was no sane option to send email messages, today it is the solution most Node.js users turn to by default.

Incidents

- First time working with NodeJS and ExpressJS.
- Difficulty understanding basic concepts for the development of the project.

- Difficulty with the Contact section. For example using nodemailer and displaying the content.
- Some difficulty understanding how to create the Controller files for many sections such as About, Contact, among others.
- At first, some difficulty following the tutorial. I had to follow it slowly, step by step.
- Some difficulty implementing the Passport dependency.