

# Vue js

---

## Project Outline

This project covers the idea of understanding and implementing Vue framework. The method to achieve this is by designing a SPA capable of rendering images from a public API. To completely understand the basics of VUE framework and how it renders is the main objective of the project

## Objectives

- Improve understanding of VUE
- Design a SPA
- Use AXIOS to get http requests
- Develop an infinite scroll image gallery
- Use scroll position to load images

## Project requirements

At the level of design and functionality, you must create a simple image gallery (you can choose any type of design) that will load groups of 5 images as the user scrolls vertically, so you will have to take into account:

- Scroll position of the user who is viewing the page
- How to consume the API in a paginated way (obtain blocks of images in each request)
- How to control the number of requests made to the API
- Understand that it is v-model in VueJS and use it as long as necessary
- Understand what are the methods of a component in VueJS and implement them

## Scope of the Project

The scope of the project is to understand use cases of VUE framework and when it is used. To develop the interest of the developer for the vue framework

## Tools

The following tools were used for the execution of the project:

- GIT workflow, for smooth control and development of the project
- NPM, to install various modules required for the project
- WEBPACK dependency manager, to manage and work as a one stop gap for all the dependencies required for the project
- AXIOS, library to get http requests

## Points to keep in mind

The following aspects are necessary to keep in mind while executing the project

- Implementing good practices with GIT, it is important that you pay attention to the number of commits as well as their content.
- Make use of the AXIOS library to make requests to the Image API
- You must use a dependency manager
- You must extract the images from the public API <https://jsonplaceholder.typicode.com/photos>
- Use Sass for the CSS part
- Create a clear and orderly directory structure
- 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
- In the case of using different programming languages always define the implementation in separate term
- 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
- You should try as much as possible that the commits and the planned tasks are the same
- Delete files that are not used or are not necessary to evaluate the project

## Project Calendar

This section describes the project calendar or the estimated delivery times needed for the project;

Deliverable	Estimated Time	Date
NPM install	05 mins	09/12/2019
Webpack dependencies	2-5 mins for each dependency	09-12/12/2019
AXIOS get images paginated	5-8hrs	10-12/12/2019

VUE show images	5-8 hrs	10-12/12/2019
Sass	1-2 hrs	11-12/12/2019
Documentation	30mins-1hr	12/12/2019

## Risk Management

Following are the risks associated with the execution of the project:

Risk	Risk level
Implementation of VUE	Very High
AXIOS get paginate images	High
Show images paginated way	Medium to High
Image styling using Sass	Medium
Putting together the whole project	Low

After the risks have been pointed out, it is important to show the following calendar which will guide the project all through out. The following shows the list of the tasks to be performed with the details of the tasks to be performed including difficulty and priority level. 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 ).

Tasks	Priority	Difficulty
Get images from HTTP request	5	2
Install necessary dependencies	4	2
Use Vue to render data	4	5
Get images in paginated way	2	4
Show images based on scroll	3	4