Universidad Autónoma de Madrid

Departamento de Informática

Computer Systems Project
Assignment 2

Roberto Marabini
Alejandro Bellogín

Changelog

Version[1]	**Date**	**Author**	**Description**
1.0	10.10.2022	RM	First version.
1.1	17.11.2022	RM	Small modifications in the description of the assignment.
1.2	5.12.2022	RM	Renumbering assignment 3 -> 2
1.3	12.12.2022	AB	Translation to English
1.4	26.1.2023	RM	Review before upload to *Moodle*

[1] Version control is made using 2 numbers $X.Y$. Changes in Y denote clarifications, more detailed descriptions of some aspect, or translations. Changes in X denote deeper modifications that either change the provided material or the content of the assignment.

Contents

1 Goal

In this assignment you will learn to create an application from scratch using *Vue.js*. We will first see the basic concepts of *Vue.js* together with the architecture of the applications created with this framework.

Then, you will learn to create a CRUD (Create, Read, Update, Delete) application step by step, that will be able to connect to a REST API. When the assignment ends, you will have learned to create **components**, **events**, **methods**, **computed properties**, and **forms**, while at the same time knowing how to manage the **state of the components**, accessing external **APIs**, and managing **routing**.

2 Introduction

Vue.js is an open source code environment developed in JavaScript that is responsible of providing the necessary tools to create views in an MVC (Model, View, Controller) architecture.

For this assignment you will need to have both `Node` and `npm` installed. Note: `npm` is automatically installed when installing `Node`.

Listing 1: How to install `npm` using the CLI. The red arrow shows a line break included to improve the the listing readability.

```
# install node (and npm)
# node is already installed in the lab computers (linux partition)
curl -fsSL https://deb.nodesource.com/setup_14.x | sudo -E bash -
    ↪ sudo apt-get install -y nodejs
```

In this assignment, you will create a management application with *Vue.js*. The application will allow to manage data from different people. We will see **how to install *Vue.js*** and how to configure it. You will see next how is the **structure of a *Vue.js* file**. Then, you will learn to create components, properties, methods, events, and different sentence types of *Vue.js*.

To create the application interface, we will use tables and forms, adding fields to the latter together with proper validations. Finally, you will also learn to deploy an application created with *Vue.js* in *Render.com* and how to connect it to an API.

In `https://tutorialvue.onrender.com/` you may see the result of the application you will create.

3 Assignment organisation

The assignment is divided in three sections that are separately described in three PDF files. In the first, you will create an application with all the required functionality except the ability to persist data. In the second part, the application will access a `REST` API created in *Django* that will allow data persistence. Finally, in the third part, we will discuss about how you may navigate between different URLs without needing to connect with the server.

4 Time assignment for the different tasks

Three weeks (about 9 hours of class) have been allocated to finish this assignment. It is expected that, approximately, half of the effort is devoted to the first part and a quarter of it to each of the remaining two parts.

5 Material to submit when the assignment ends

1. You must clearly indicate, when submitting the assignment in *Moodle*, the URLs where the *Vue.js* and *Django* projects are deployed in *Render.com*. This information must correctly appear configured in the `settings.py` file, submitted with the *Django* project (variables `CORS_ORIGIN_WHITELIST` (*Vue.js*) and `ALLOWED_HOSTS` (*Django*)). For *Render.com*, the base used must be created in `https://neon.tech` and the admin interface *Django* must be accessible through a user and password (`alumnodb`).

2. In this assignment three projects will be created, two in *Vue.js* and another in *Django* called `tutorial-vue`, `twopages` and `persona` respectively. For each of these projects, submit to *Moodle* the file obtained after executing the command `zip -r ../xxxxx.zip .git` from the root folder of the project. Remember that you have to add (*git add*) and send (*git commit*) the files to *git* before running that command. Check the content of the zip file is correct by executing the order: `cd /tmp; rm -rf tmpDir; unzip xxxxx.zip; git clone .  tmpDir; ls tmpDir`. The folder `tmpDir` should include the project.

   Since three projects are being created, two in *Vue.js* and another in *Django*, you will have to upload to *Moodle* three ZIP files called `tutorial-vue.zip`, `twopages.zip`, and `persona.zip` respectively.

3. Do not upload nor add to the repository the virtual Python environment (*Django*), nor the folder `node_modules` (*Vue.js*) since those are only valid for the computer where they have been created.

6 Evaluation criteria

The grade of this assignment will be *Pass* or *No Pass*. To pass the assignment it is necessary to satisfy ALL the following criteria:

- The code created for this assignment should be compatible with the versions of the different programs that are described in the report. In particular, Python 3.9, *Django* 3.2, and *Vue.js* 3.2
- All the required functionalities in the assignment guide must be satisfied.
- Application data must be persisted in a PostgreSQL database stored in `https://neon.tech`.
- The application devoted to people management (projects `tutorial-vue` and `persona`) must be deployed in *Render.com*, and must be accesible through the provided URL. In *Render.com*, the admin interface (*Django*) must be able to visualize and change data with the appropriate privileges, that is, by using the created superuser (username: `alumnodb`, password: `alumnodb`).
- It is not required to deploy the application `twopages` in *Render.com*.
- The code created in this assignment must be stored in a private repository through `git`. **There must be at least one git submission for each class day by each member of the group**. And these submissions must be registered in the `.git` directory uploaded to *Moodle*.
- The necessary code to run the application must be submitted to *Moodle*, and it must be possible to run it locally.

NOTE: The code used to evaluate this assignment will be the one uploaded to *Moodle*. In no case the existing code in the *git* repository or in *Render.com* will be used.