

**Professor Portal**

**Algarve University**

Faculty of Sciences and Technology

In Partial Fulfillment

of the Requirements for the subject of

Web Application Development during my Erasmus grant

**By:**

Yousef Hisham Shaban

**Presented To:**

Noélia Susana Costa Correia

**Faro, Portugal**

**Fall 202**

Table of Contents

[CHAPTER 1: Introduction 3](#_Toc123940744)

[1.1 Overview 3](#_Toc123940745)

[1.2 Project Motivation 3](#_Toc123940746)

[CHAPTER 2: Project Layout 4](#_Toc123940747)

[2.1 Students Table 4](#_Toc123940748)

[2.2 Add Student Form 4](#_Toc123940749)

[2.3 Edit Student Form 5](#_Toc123940750)

[CHAPTER 3: Functionalities and Requirements 5](#_Toc123940751)

[3.1 Simple Mail Transfer Protocol 5](#_Toc123940752)

[3.2 Server Side 6](#_Toc123940753)

[3.3 Local Database 7](#_Toc123940754)

[CHAPTER 4: Project Setup 8](#_Toc123940755)

[4.1 installing the development environment 8](#_Toc123940756)

[4.2 Getting familiar with directories 8](#_Toc123940757)

[4.3 Downloading libraries 8](#_Toc123940758)

[4.4 Starting the environment 8](#_Toc123940759)

[4.5 Enjoy the experience! 8](#_Toc123940760)

[4.6 Common Errors 8](#_Toc123940761)

# CHAPTER 1: Introduction

## 1.1 Overview

University cores are without a doubt a significant component of modern societies and education; consequently, there must be systems that oversee such units, such as the subjects registration system, in which teachers input the students' information in general into the main system. Because the method may be frustrating and unfair to certain students, an alternative to Excel must be found with more communication with the student.

## 1.2 Project Motivation

Most professors in multiple universities have the issue in which they cannot access certain functionalities such as “calculating the average, viewing passed/failed students, calculating the total number of students, rating passed students, sending emails regarding that, etc." unless they submit their grades on the portal, which causes another problem in that they cannot edit their grades without approval from certain sides, and this is where the idea of PPO(Professor Portal) comes from.

# CHAPTER 2: Project Layout

Graphical user interface, application

Description automatically generated

Chapter 2 figure shows the main project look, which contains the following components:

### 2.1 Students Table

This table will contain all students stored in the database, also it will contain an action for each record of any student.

Under the table, there will be multiple buttons to get certain statistics based on the above records.

Be aware that each student mark background will be changing based on failing or passing the subject.

## 2.2 Add Student Form

This form will be the client-side method to add any students into the students table (on the right) and to the database.

Please notice that each student that is being added will receive an email regarding that.

## 2.3 Edit Student Form

Same as adding a student, this form will be the client-side method to edit any students in the students table (on the right) and to the database.

Please notice that each student that who got edited will receive an email regarding that.

# CHAPTER 3: Functionalities and Requirements

## 3.1 Simple Mail Transfer Protocol

A screenshot of a computer

Description automatically generated with medium confidence

Figure 3.1 shows the code block which is responsible to be sending a promise regarding if the student successfully received an email or not, knowing that this feature will me essential for adding students or editing them.

## 3.2 Server Side

Text

Description automatically generated

Figure 3.2.1 and Figure 3.2.2 will be showing the server-side validations for any input accruing, which will be very similar to client-side validations, which each will include all case scenarios.

Text

Description automatically generated

## 3.3 Local Database

Graphical user interface, text

Description automatically generated

Figure 3.3 shows the local database which contains the record of the students that was displayed in chapter2.

# CHAPTER 4: Project Setup

## 4.1 installing the development environment

- First, please make sure of [which Operating system](https://www.wikihow.com/Check-a-PC-Operating-System#:~:text=Press%20the%20Windows%20key.,the%20Apple%20logo%20is%20there.) you are using currently

- After so, go ahead and press here to go ahead and download [visual studio code](https://code.visualstudio.com/download).

- Finally, make sure to download [Node.js](https://nodejs.org/en/download/).

## 4.2 Getting familiar with directories

- Open Visual Studio Code, and Open Folder from File menu.

- Open the folder of the project from the destination you extracted it.

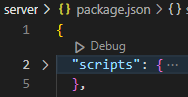
- scroll through and start memorizing the files inside.

## 4.3 Downloading libraries

- Make sure to check each file required libraries by simply checking errors they give in the Problems sections down the screen, and after so go ahead and google they proper syntax for the installation in the terminal, some of the most important imports will be:

path, nedb, nodemailer, fs, express, react, axios, etc.

## 4.4 Starting the environment



- Go ahead and check for the “package.json” located in both client and server folders, you will see a grey “Debug” button above the scripts, please go ahead and debug the server then the client side.

## 4.5 Enjoy the experience!

Congratulations, you just launched the portal and now it’s ready to be used!

## 4.6 Common Errors

-Make sure to always start the debug of the serve-side before the client-side.

-Make sure that the ports used for server and client sides is not reserved by other processes.