MOHAMMED V University of Rabat

Faculty of Sciences

Université Mohammed V
Faculté des Sciences
Rabat

IT Department

Computer science bachelor
(SMI)

GRADUATION PROJECT

titled :

Grades Management System for Semesters, Years and Diplomas

Presented by:

FADEL LAMIAA
JBILI AYOUB

Thesis defended on 27 October 2020 before the Jury:

RIADSOLH Anouar	Professor, FSR	*President*
ZITI Soumia	Professor, FSR	*Supervisor*
KHARMOUM Nassim	Professor, ESTM	*Supervisor*
EL BOUCHTI Karim	Engineer, CNESTEN	*Examiner*
LAMJID Asmaa	Phd student, FSR	*Examiner*

Academic Year 2019-2020

Acknowledgment

We want to thank Mrs. ZITI Soumia and Mr.KHARMOUM Nassim. They were a continual source of learning during all our discussions and meetings, and without their support, the project could not have attained its goal. We are grateful to all the educational staff of the computer science department of Mohammad V University of Rabat.

Summary :

In this report, we will present the result of our work within the FSR, which is part of our graduation project. Given the scope of commitments of FSR officials, it was essential to have an application providing a rade management system. indeed, the establishment of such an application will generate several advantages to the organization, namely the simplicity of the procedures of checking transcriptions and managing the subjects and staff The purpose of this report is to explain the approach we have taken throughout this achievement as well as the various technical choices made .

Contents

List of Figures

Chapter 1

Context

1.1 Preface

In this chapter, we will have a small presentation about the host organization, the problematic and objectives of our project, and lastly, in the end, we will explain the used methodology and the adopted formalism.

1.2 Host organization

1.2.1 Faculty of Science

The Faculty of Sciences-Rabat (FSR) was created in 1952. It is part of all the faculties of the Mohammed V University of Rabat. It is the faculty where baccalaureate students pursue their higher studies in the scientific field (mathematics, computer science, physics, the science of life and the universe, chemistry). The headquarters of the current Faculty of Science was put into operation in 1952 as the "Center for Higher Scientific Studies". Became "Faculty of Sciences" in 1957-1958, name formalized by the Dahir in 1959 after the creation of the Mohammed V University in 1957. This establishment was to experience a significant extension of its premises with the doubling in 1966 of the area covered, then its quadrupling towards the end of the years 1960 - 1970. It was necessary to face the rapid increase in the number of students and to meet the increased needs of research laboratories.

1.2.2 Organisational structure

The FSR adopts a governance model that revolves around several organs and entities that work to fulfill specific functions, both at the academic and scientific level and at the level of the proper functioning of the faculty. It has an administrative organization whose main role is to ensure the quality of the services offered to its students and researchers as well as to professionals within the framework of continuing education.

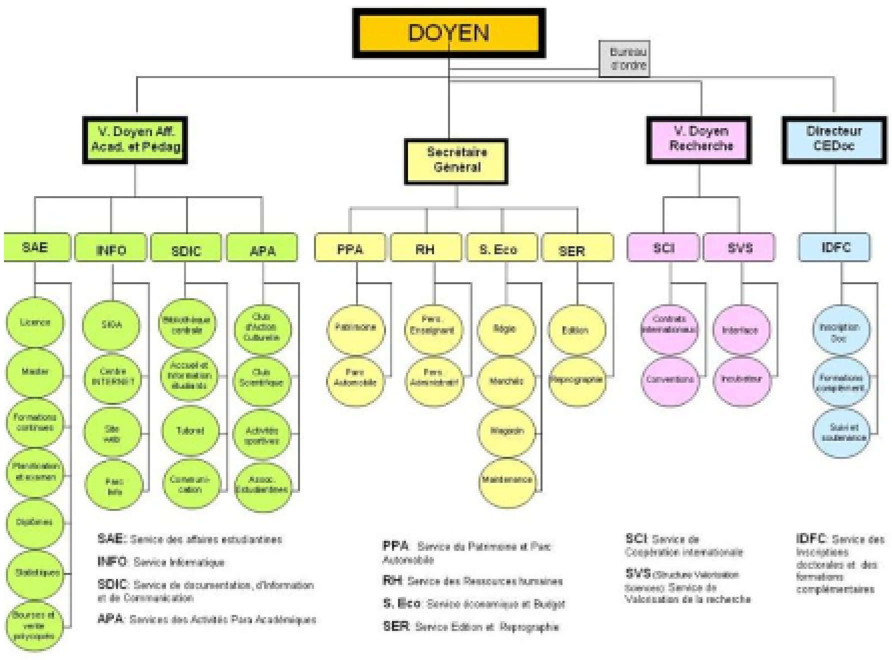

DOYEN
Bureau d'ordre
V. Doyen Aff. Acad. et Pedag
Secrétaire Général
V. Doyen Recherche
Directeur CEDoc
SAE
INFO
SDIC
APA
PPA
RH
S. Eco
SER
SCI
SVS
IDFC
Licence
Master
Formations continues
Planification et examen
Diplômes
Statistiques
Bourses et vente polycopiés
SIGA
Centre INTERNET
Site web
Parc Info
Bibliothèque centrale
Accueil et information étudiants
Tutorat
Communication
Club d'Action Culturelle
Club Scientifique
Activités sportives
Assoc. Estudiantines
Patrimoine
Parc Automobile
Pers. Enseignant
Pers. Administratif
Régie
Marchés
Magasin
Maintenance
Edition
Reprographie
Contrats internationaux
Conventions
Interface
Incubateur
Inscription Doc
Formations complément.
Suivi et soutenance
SAE: Service des affaires estudiantines
INFO: Service Informatique
SDIC: Service de documentation, d'Information et de Communication
APA: Services des Activités Para Académiques
PPA: Service du Patrimoine et Parc Automobile
RH: Service des Ressources humaines
S. Eco: Service économique et Budget
SER: Service Edition et Reprographie
SCI: Service de Coopération internationale
SVS (Structure Valorisation Sciences): Service de Valorisation de la recherche
IDFC: Service des Inscriptions doctorales et des formations complémentaires

Figure 1.2.1: Organisational structure

1.3 Problematic

The problem that was given to us is to create an application that will be able to provide the professors to know the students who passed the semester after the deliberations by simply allowing the system to display the results after providing it with data needed.

1.4 Required work

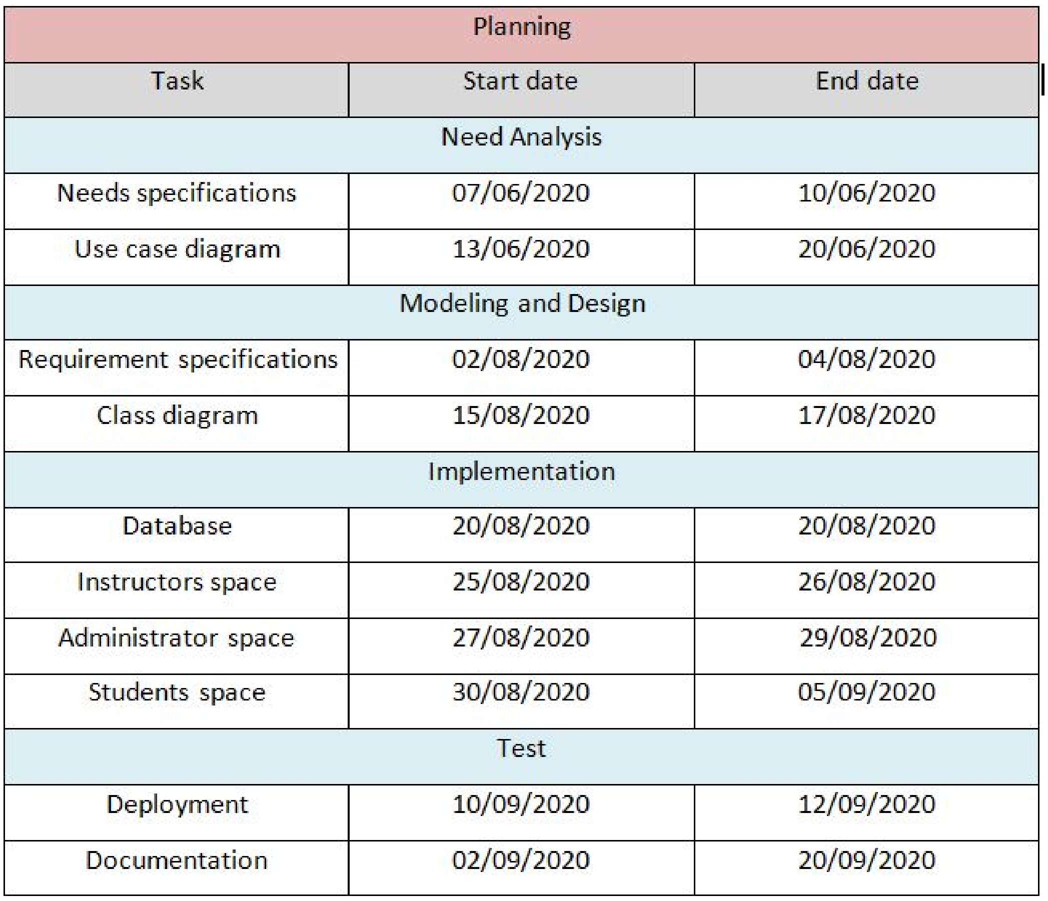
The objective of this application is to create a simple platform for the educational staff to use by implementing a database that holds all the marks of the students to be able to differentiate between the students who passed a certain semester and passed to the other or obtain a certain diplomat and those who did not.

1.5 Methodology

We will be working with UML(Unified Modeling Language) to create a class diagram and sequence diagram that explain the general concept of the project, we will be using Laraval as the back end framework and as the front-end framework, we will be using Bootstrap, providing the user a simple yet efficient design.

1.6 Planning

The progress of a project cannot be completed without the implementation of a plan with two dimensions: time and the tasks to be accomplished before the deadlines. Project planning is the process you go through to establish the steps required to define your project objectives, clarify the scope of what needs to be done, and develop the task list to do it. After many talks with our supervisor to figure out the objectives and goals of our project, we've finally agreed on a plan to follow. The research and implementation of this project are carried out according to the project management method: A list of tasks, with each to be performed given a specified amount of time.



Planning		
Task	Start date	End date
Need Analysis		
Needs specifications	07/06/2020	10/06/2020
Use case diagram	13/06/2020	20/06/2020
Modeling and Design		
Requirement specifications	02/08/2020	04/08/2020
Class diagram	15/08/2020	17/08/2020
Implementation		
Database	20/08/2020	20/08/2020
Instructors space	25/08/2020	26/08/2020
Administrator space	27/08/2020	29/08/2020
Students space	30/08/2020	05/09/2020
Test		
Deployment	10/09/2020	12/09/2020
Documentation	02/09/2020	20/09/2020

1.7 Software Development Life Cycle (SDLC)

Software Development Life Cycle is a well-defined and systematic approach, practiced for the development of a reliable high-quality software system. There are tons of SDLC models available. The process of building computer software and information systems has been always imposed by different development methodologies. A software development methodology refers to the framework that is used to plan, manage, and control the process of developing an information system. These classic software life cycle models usually include some version or subset of the following activities:

- Planning
- Requirement Analysis.
- Software Modeling and Design.
- Coding.
- Documentation.
- Testing.
- Deployment and Maintenance.

The sequence and/or presence of these activities depend on the model chosen for the project development.

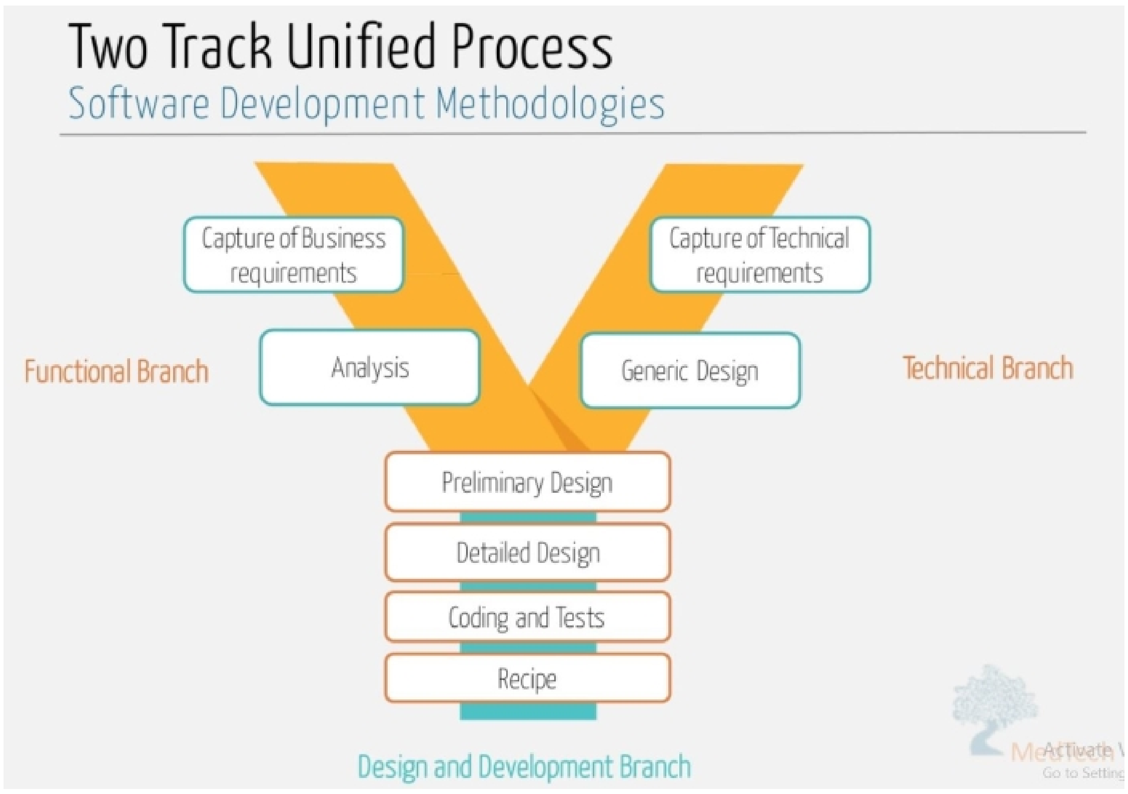

Two Track Unified Process
Software Development Methodologies
Capture of Business requirements
Capture of Technical requirements
Functional Branch
Analysis
Generic Design
Technical Branch
Preliminary Design
Detailed Design
Coding and Tests
Recipe
Design and Development Branch

Figure 1.7.1: Two Track Unified Process (2TUP) or Y model.

2TUP is a unified process (i.e. software development process) built on the UML modeling language. Every process answers the following characteristics:

- It is an incremental process, allowing a better technical and functional risk management.
- It is component-oriented, offering flexibility to the model and supporting the re-use.
- It is user-oriented, built from their expectations.

1.8 Conclusion

As seen before, this chapter establishes an introductory section where we discussed some big points related to our work.

Chapter 2

Preliminary study

A preliminary study describing the fundamental concepts relating to a given problematic is, with no doubt, a vital step to every project implementation. In this chapter, we present the preliminary study we crafted, to accomplish the conception and development phases of our project. We will begin our study by identifying the basic notions related to the grades and deliberation management application, and the different technologies and methods adopted.

2.1 Grades management

The used grading system is as follows:
if a student obtains a mark above ten -which is considered the average- passes that subject, if not then he fails but has another chance in the catch-up session to re-pass the exam if he then obtains a mark equal to five or more but bellow 10 he have a right of compensation if he has a certain subject that can fill to reach the average then it's competing if not the right of compensation is revoked. it's not possible to subscribe for more than six subjects and not less than that, you can only unlock a semester if u pass more than four subjects and a semester is not considered passed only if you pass all six subjects.

2.2 Deliberations

The deliberations come after all exams and catch up session have passed and it's a long and careful consideration or discussion that aims to replace the mark for a better one by a marginal change .it's held by the educational staff and takes into consideration the presence, the participation: and behavior of the student.

2.3 Conclusion

We have presented the principle concept of the project and based on these facts, we've got a certain image of the output of the applications.

Chapter 3

Requirements specification

This stage depends on understanding the system's context. It's all about determining the functionalities, the relevant actors and, to point out the critical risks, and to identify the initial use cases.

3.1 System actors

An actor specifies a role played by a user or any other system that interacts with the subject, but which is external to the subject.
Our application manages the students' grades and deliberations, in which it will interact with three different actors:

Administrator: He has total visibility over the databases and users. In other words, he is concerned with everything that has to do with the application.

Instructor: manages the students' marks and final grades (deliberations).

Student: have easy access to check their grades.

3.2 Functional requirements

Functional needs represent the actions that the system must perform, it only becomes operational if it satisfies them all. this application must mainly cover the following functional needs:

Adding Grades: the system must allow instructors easy access to add students grades.

Displaying results: the system must allow the students to view their marks.

Content management: the system must allow administrators to manage content such as courses, branches and diplomas.

Project management: the system must allow administrators to add, modify or delete students and instructors .

3.3 Non-functional requirements

They are requirements that do not specifically concern the behavior of the system but rather identify the internal and external constraints of the system. The main non-functional requirements of our application are summarized in the following points:

Requirements for performance: describe the system's run-time performance generally in terms of response time.

Ergonomics: the application offers a user-friendly and easy-to-use interface.

Integrity: guarantee the integrity and consistency of the data for each update and each insertion.

Readability: the code must be clear to allow future improvements.

In what follows, we will deepen the description of the needs previously specified by the construction of the corresponding class diagram for better visibility of the designed application.

Chapter 4

Project conception

Designing a creative process is of major importance in the development cycle of a project. So this chapter will be devoted to the presentation of the different stages of the design of our application and to present this part as well as possible we will give a global view describing the general architecture of our system to extract the different modules that make it up.

4.1 Global solution architecture

The architecture of a system is its high-level design. Computer architecture defines the structuring of a computer system, hardware, and software, in terms of components and organization of its functions. The structural views of software architecture are:

The logical view: which defines the main components of architecture without worrying about the physical details (machines, equipment ...).

The physical view: which is a description of the integration and distribution of the software part on the hardware part.

4.2 Logical view

A subsystem is a coherent definition which deals with part of the problem defined in terms of the services it forms. It is a set of classes, association, operation, event of constraint, having well defined and restricted interfaces with other subsystems. The division of the system into independent sub-systems.

4.3 Application framework

We decided to work with Laravel which is a web application framework with expressive, elegant syntax that aims to make the development process a pleasing one for the developer without sacrificing application functionality. It combines the very best of what has been seen in other web frameworks, including frameworks implemented in other languages,

such as Ruby on Rails, ASP.NET MVC, and Sinatra.
Laravel is accessible, yet powerful, providing tools needed for large, robust applications. A superb inversion of control container, expressive migration system, and tightly integrated unit testing support give you the tools you need to build any application with which you are tasked. Laravel is a free, open-source PHP web framework which means that it uses PHP as its main source language and MYSQL as its (DBMS) database management system.

PHP: (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML, but can also work like any programming language. PHP is an object-oriented imperative language it helps in the creation of not only applications but also dynamic websites.

MySQL: MySQL is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for MySQL however, is for a web database, which is where we will be using it.

4.4 The Model View Controller(MVC) design pattern

The Model View Controller (MVC) design pattern specifies that an application consists of a data model, presentation information(view), and control information. The pattern requires that each of these is separated into different objects.

Model: contains only the pure application data, it contains no logic describing how to present the data to a user.

View: presents the model's data to the user. The view knows how to access the model's data, but it does not know what this data means or what the user can do to manipulate it.

Controller: exists between the view and the model. It listens to events triggered by the view (or another external source) and executes the appropriate reaction to these events. In most cases, the reaction is to call a method on the model. Since the view and the model are connected through a notification mechanism, the result of this action is then automatically reflected in the view.

4.5 Conclusion

In this chapter, we have presented the design pattern used to design our application. Then we moved on to the detailed design to explain the classes of our application. In the next chapter, we will move on to a description of the state.

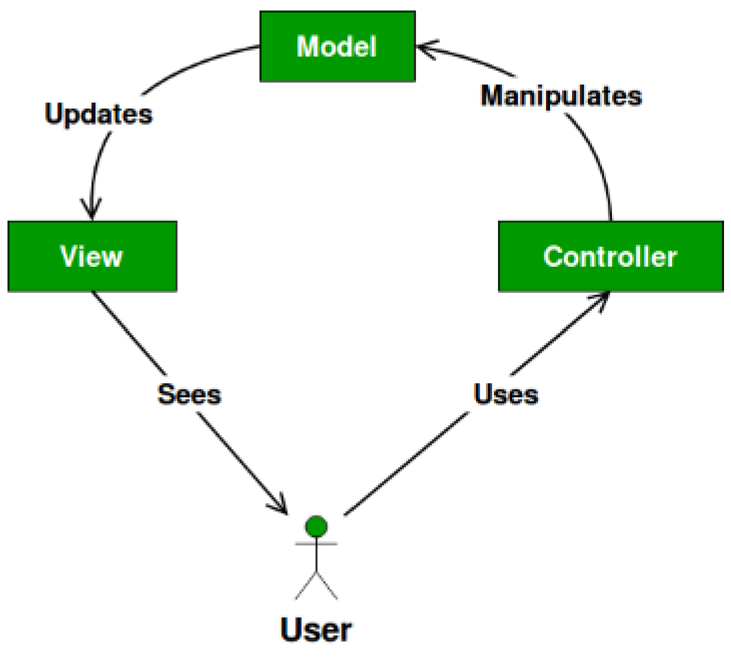


Figure 4.4.1: Model View Controller(MVC) architecture.

Chapter 5

Implementation

This chapter constitutes the last part of this report, its objective is to expose the completed work. To do this, we start with a description of the work tools. Then, we close this chapter by the presentation of the result obtained with some screenshots application interface screen.

5.1 Software environment

We used the following software environment:

Development tools: Bootstrap 4 and CSS.

Application server: Laravel.

SGBD: TablePlus.

Development language: PHP.

Source-code editor: Visual Studio Code.

UML design: Modelio.

5.2 Tools of work

5.2.1 Bootstrap 4

Bootstrap is a powerful front-end framework for faster and easier web development. It includes HTML and CSS based design templates for creating common user interface components like forms, buttons, navigations, dropdowns, alerts, modals, tabs, accordions, carousels, tooltips, and so on.

Bootstrap gives you ability to create flexible and responsive web layouts with much less efforts.

Bootstrap was originally created by a designer and a developer at Twitter in mid-2010. Before being an open-sourced framework, Bootstrap was known as Twitter Blueprint.

5.2.2 CSS

The term CSS is the acronym for Cascading Style Sheets, CSS is a computer language used on the internet to format HTML or XML files. So the style sheets also called the CSS files, include code that allows you to manage the design of a page in HTML. Although HTML can be formatted using tags provided for this purpose, nowadays it makes more sense to use CSS and only use XHTML for content. The advantage of using a CSS file for formatting a site lies in the possibility to modify all the titles of the site in one go by modifying a single part of the CSS file. Without this CSS file, it would be necessary to modify each title of each site page (difficult to imagine for huge sites of several thousand pages). Other strengths are noticeable. It is for example possible to create a style sheet specific for printing documents, which allows you to remove all styling effects and all unnecessary parts when printing. Likewise, a stylesheet can be used for mobile phone users, which helps to better manage fitness especially for the small screens of these devices.

5.2.3 PHP5

PHP5 is a free programming language, mainly used to produce dynamic web pages via an HTTP server, but can also operate like any language interpreted locally. PHP is an imperative object-oriented language. Allowed to create a large number of websites famous, like Facebook, Wikipedia, etc. It is considered one of the bases of the creation of so-called dynamic websites but also web applications.

5.2.4 MySQL

MySQL is a DBMS (Database managing system), It allows us to store data in bulk and be able to use it at will by sorting it as needed. Wich advantages are numerous:

Performance: MySQL ranks among the fastest RDBMS by combining it with asynchronous Ajax calls if the number of gigantic records.

Scalability and reliability: possibility of distributing the load on several servers, and benefit from the advantages of multiprocessor systems.

Speed of implementation: development, deployment, and administration applications for the web are accelerated thanks to the many features dedicated.

Security: It offers fine management of security and data encryption.

5.3 Class diagram:

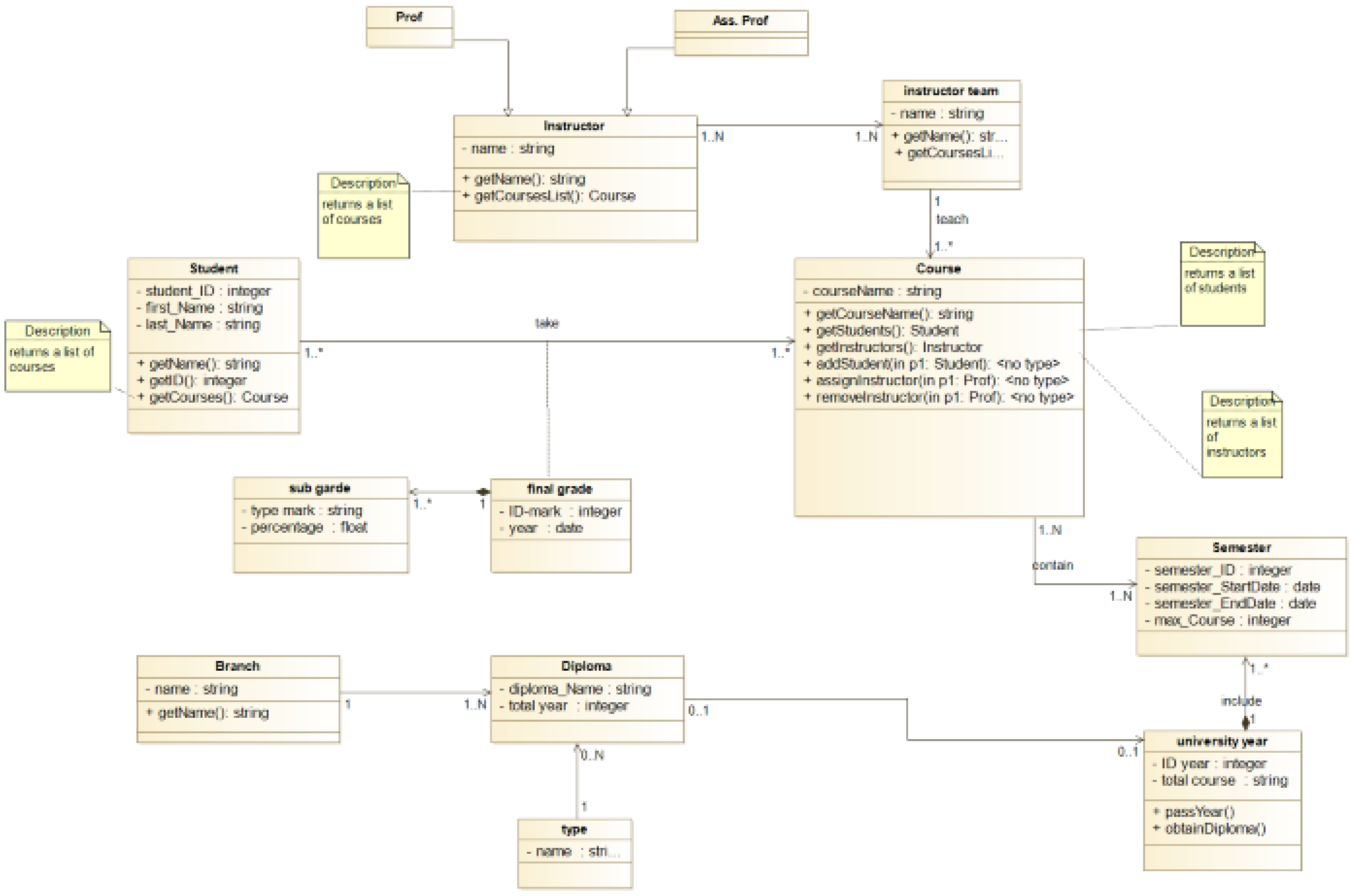


Figure 5.3.1: Class diagram

Chapter 6

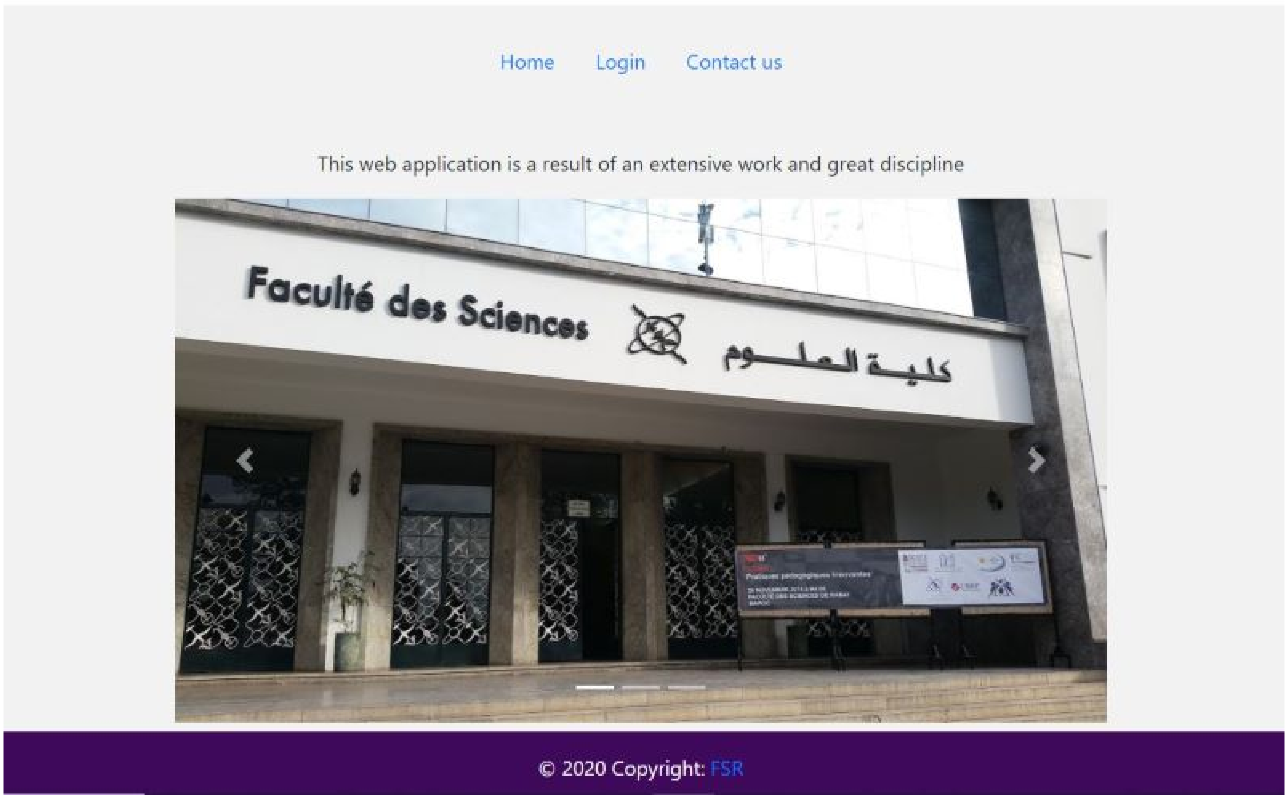
Interface

We will start by presenting the interface of our application by adding a few parts of the code with a brief explanation.

6.1 Interface presentation

6.1.1 Home page

On this page we the logo of the application that is named GRADER it has a" Home" button, "Login" button, and a "Contact us" button that we are going to explore in the following part.

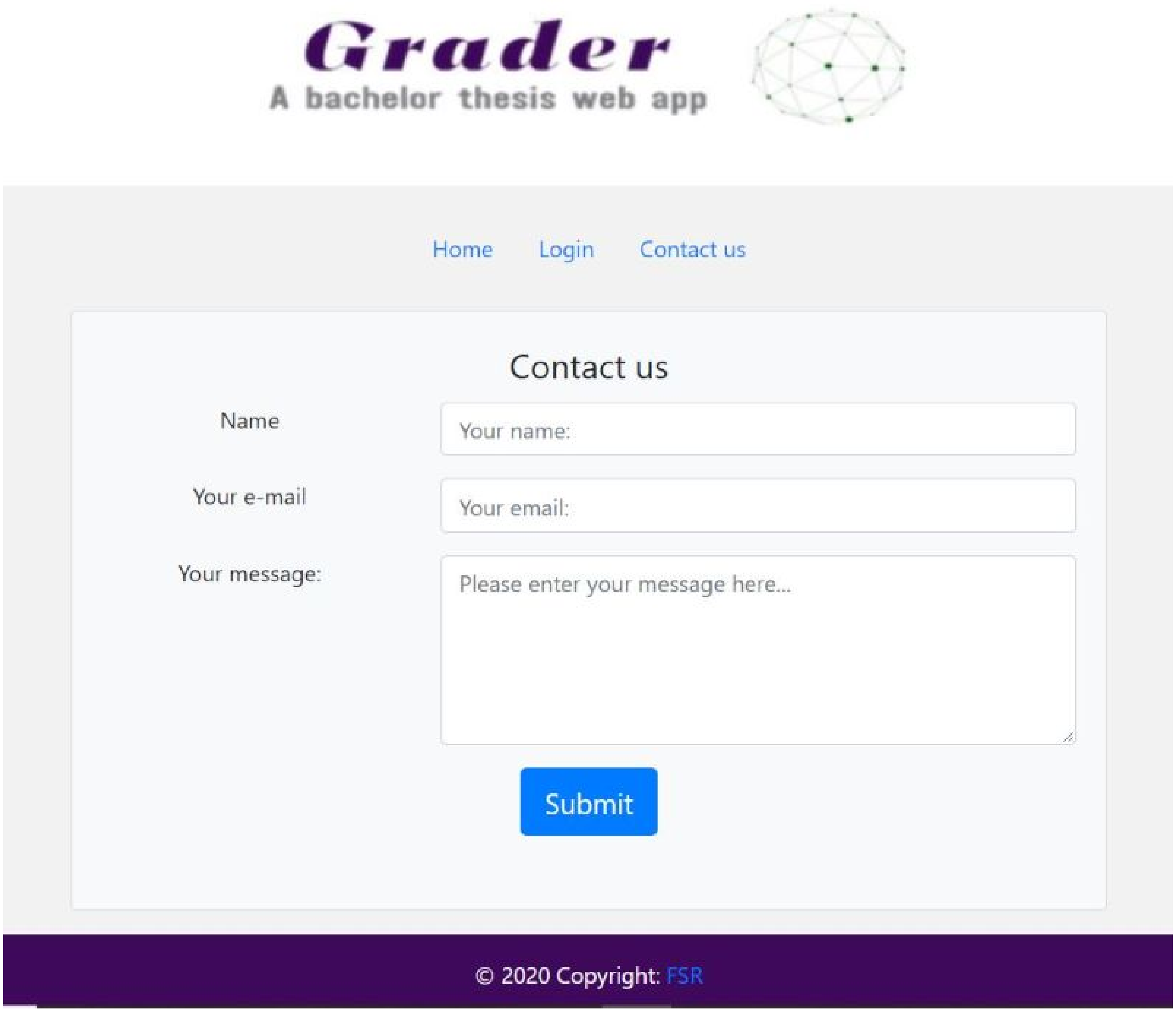


This web application is a result of an extensive work and great discipline

© 2020 Copyright: FSR

6.1.2 Contact us page

It's a simple interface where the user will be able to have contact with the administrators.


Grader

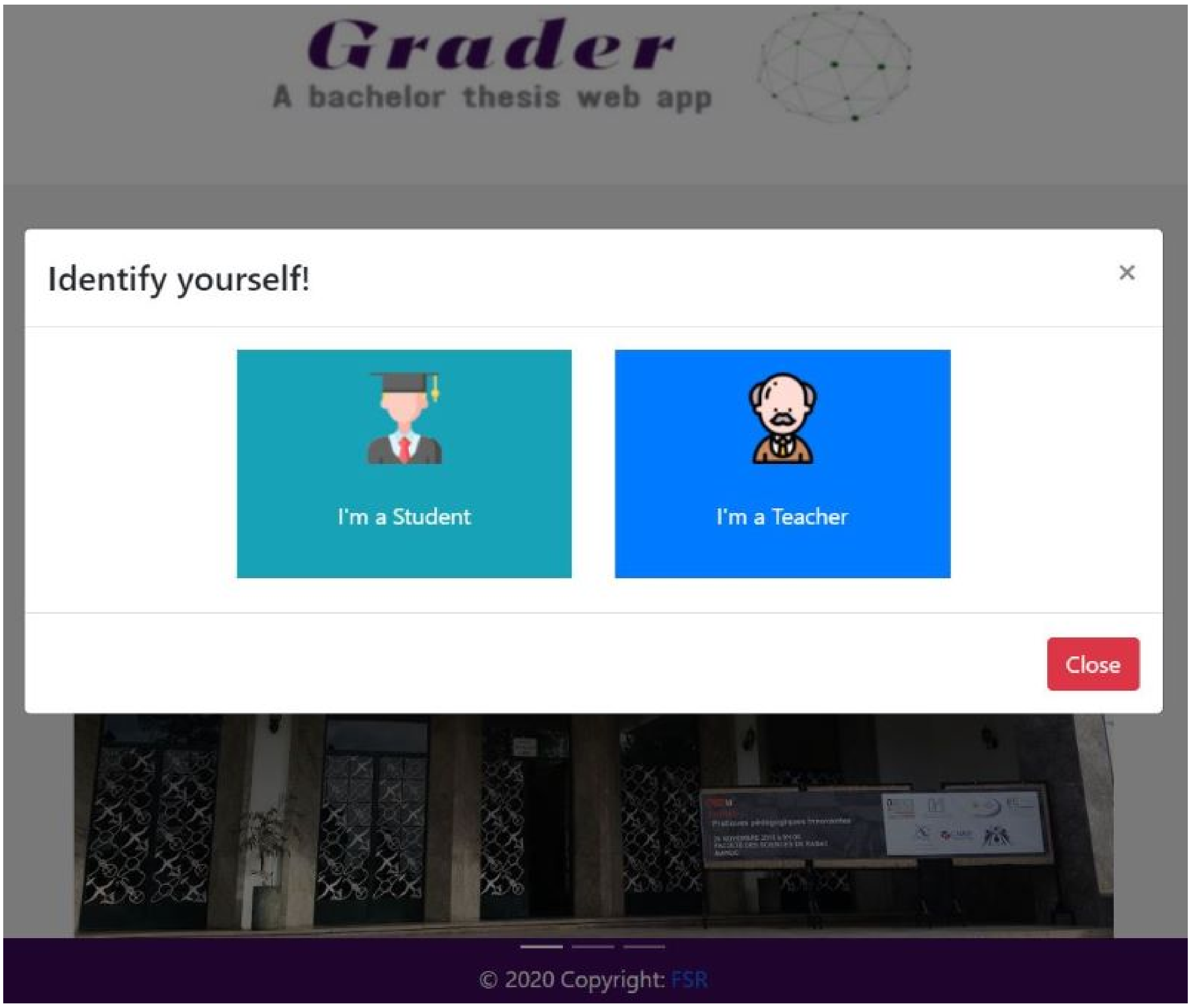
A bachelor thesis web app

Contact us

Name	Your name:
Your e-mail	Your email:
Your message:	Please enter your message here...

© 2020 Copyright: FSR

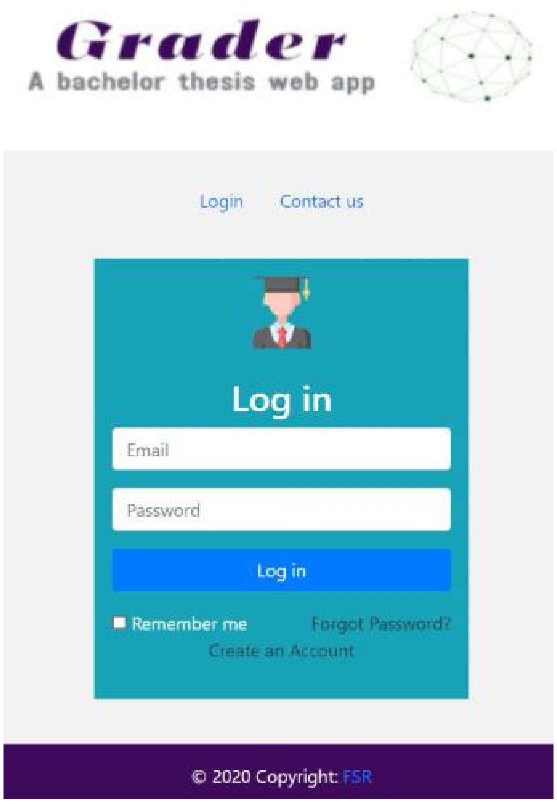
6.1.3 Login



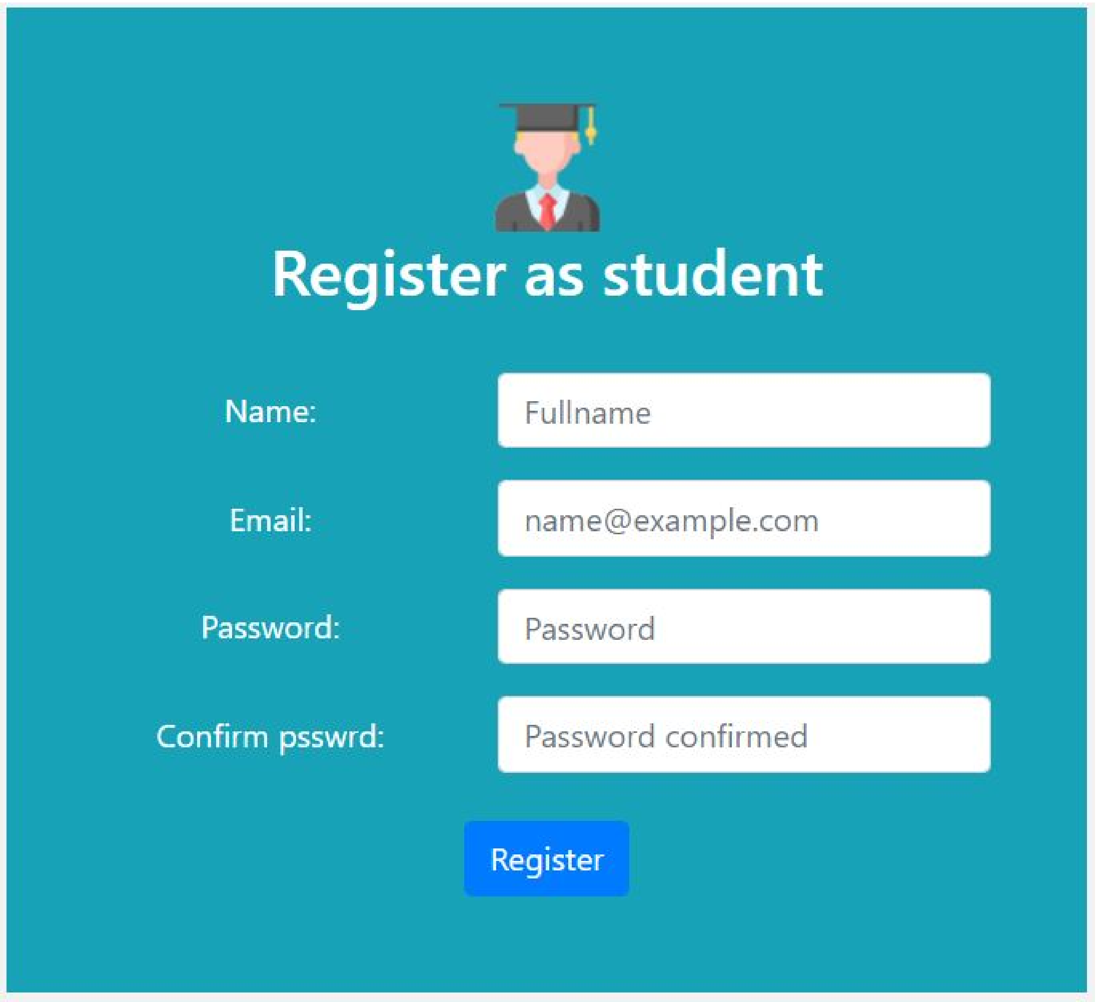
When you try to access the login page it presents you with to choices you click on the one that applies to you.

6.1.4 Student interface

The student can log in by writing his email and password he can click "Log in".

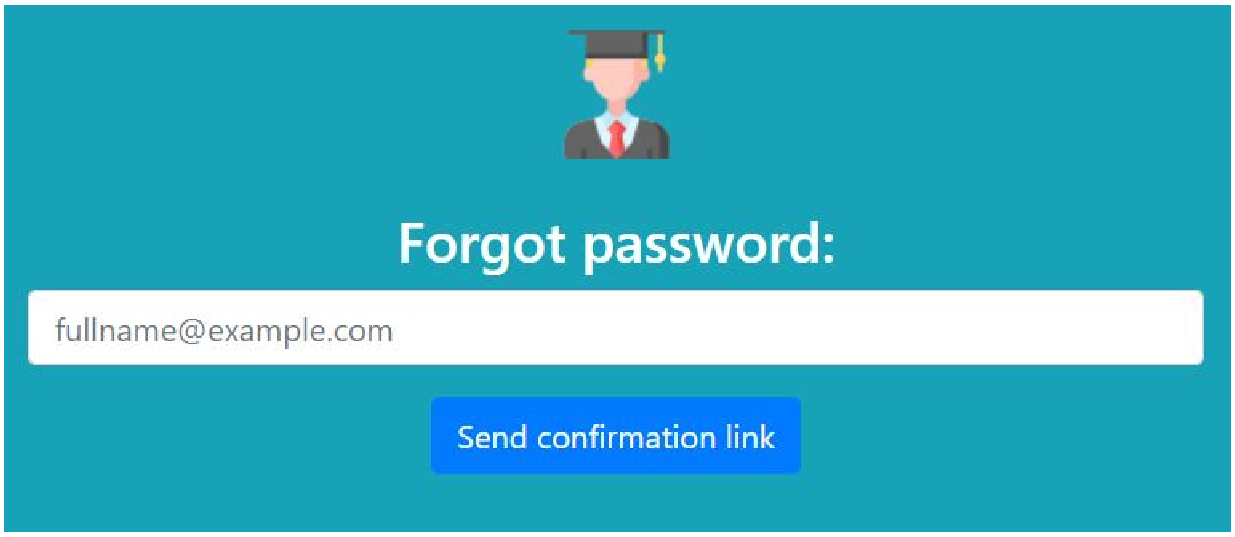


In case the student does not already have an account he will have to register.


Register as student

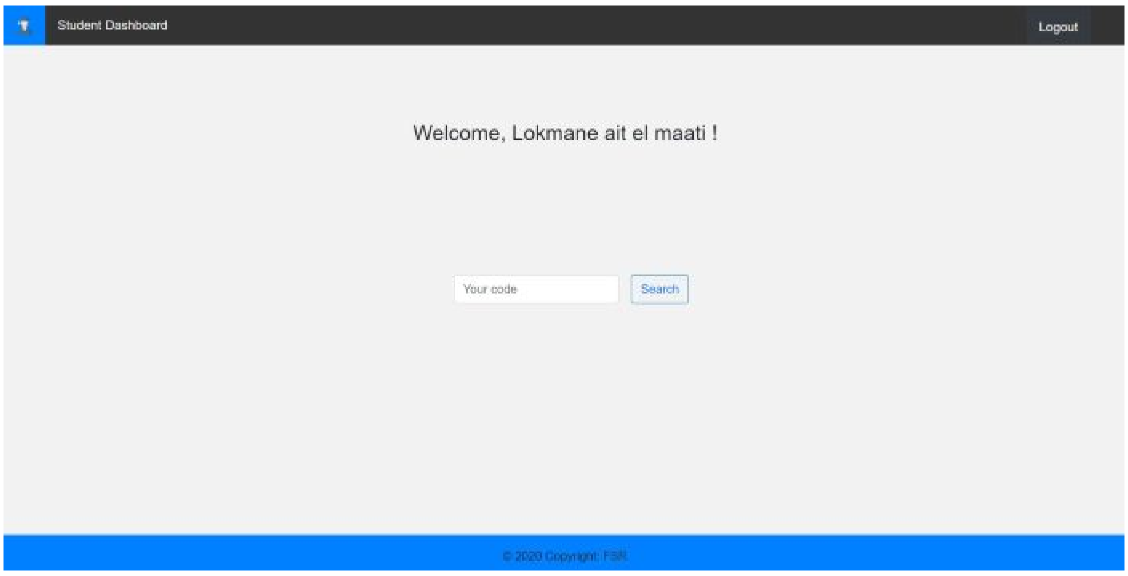
Name:	Fullname
Email:	name@example.com
Password:	Password
Confirm psswrd:	Password confirmed

And if the student forgot his password, he can always request a "reset password link".


Forgot password:

fullname@example.com

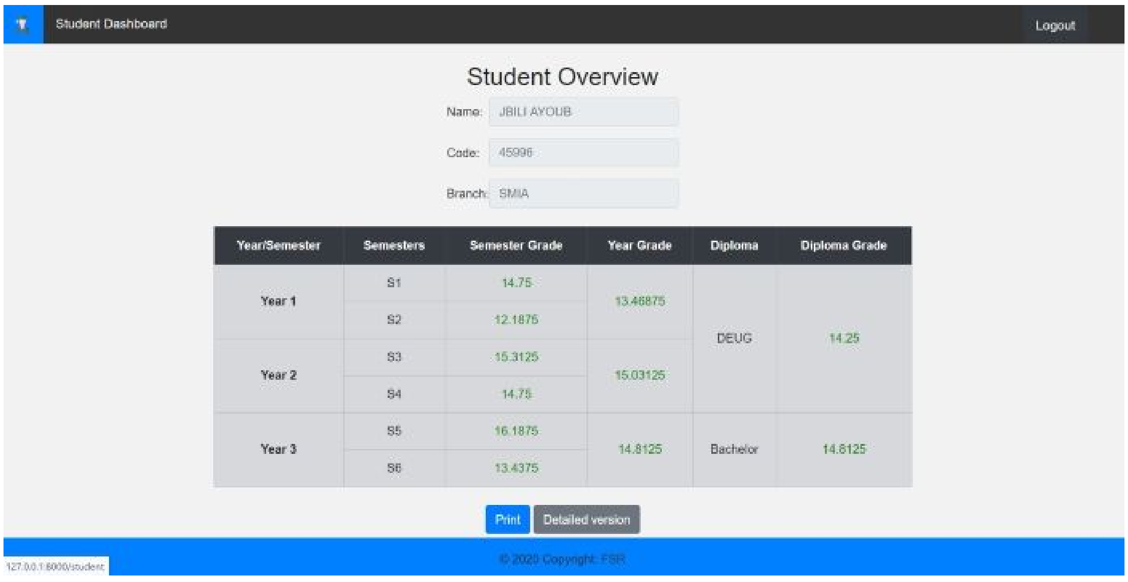
As soon as the student fills in the required cases he will be broth back to the student login page and will be able to log in.



Welcome, Lokmane ait el maati !

Your code

Here comes the part where once you put in your code you should be able to have access to your transcript, where the system accesses the database that holds student's marks with two added options the first one allows us to print the transcripts and the other gives us more detailed version of marks.

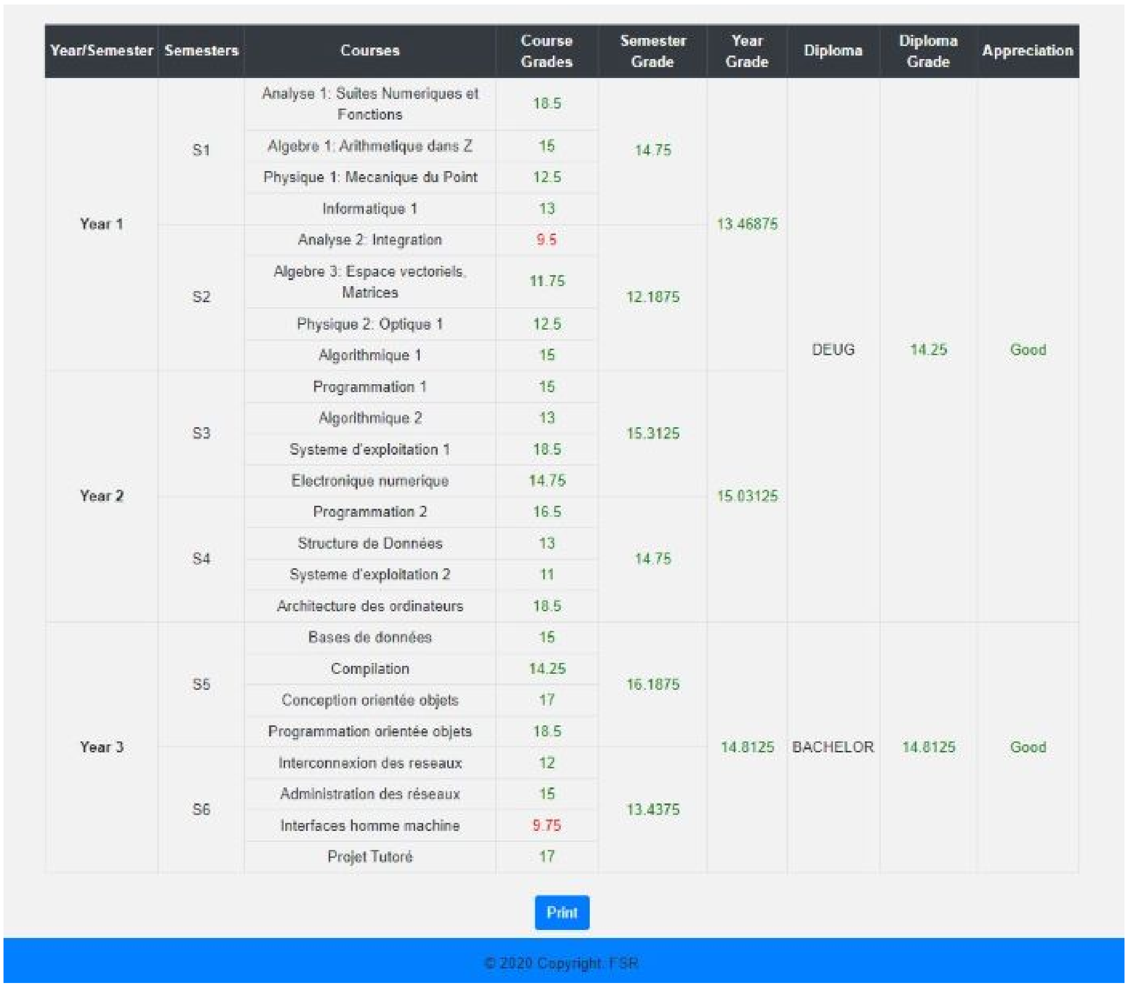


Student Overview

Name:	JBILI AYOUB
Code:	45996
Branch:	SMIA

Year/Semester	Semesters	Semester Grade	Year Grade	Diploma	Diploma Grade
Year 1	S1	14.75	13.46875	DEUG	14.25
	S2	12.1875			
Year 2	S3	15.3125	15.03125		
	S4	14.75			
Year 3	S5	16.1875	14.8125	Bachelor	14.8125
	S6	13.4375			

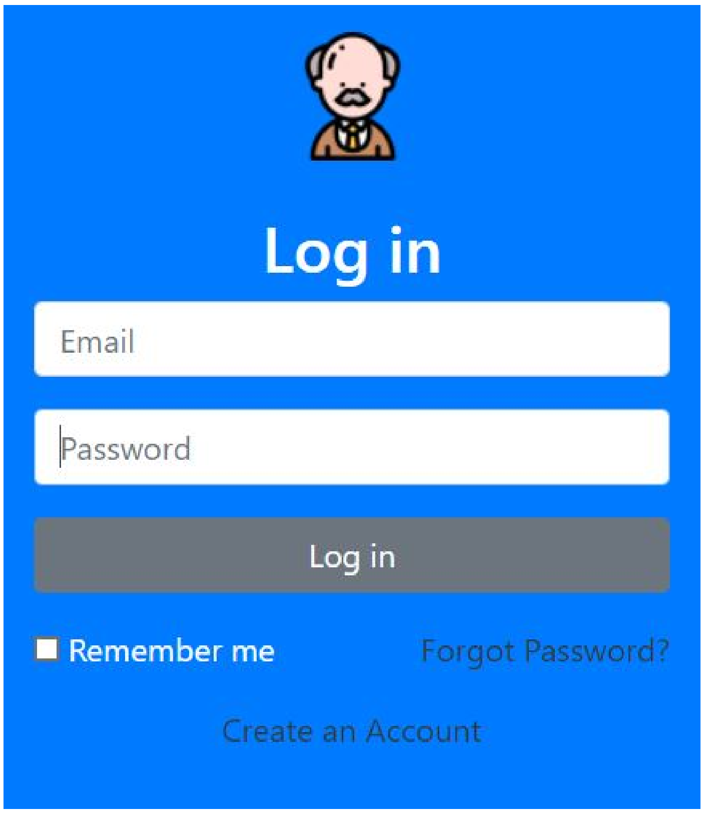
The student can also check the detailed version of his bulletin.



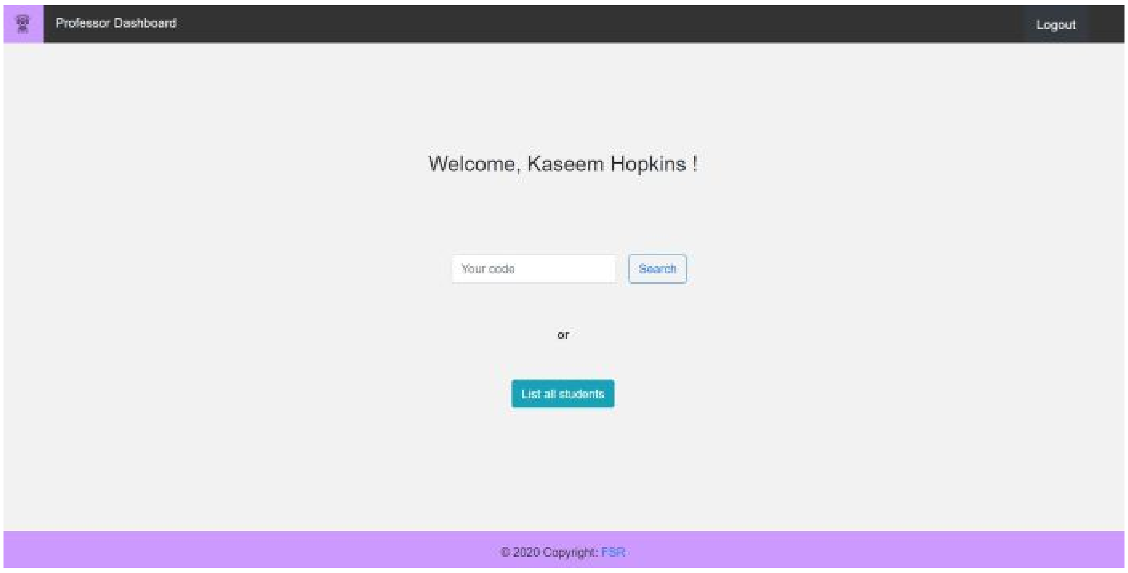
Year/Semester	Semesters	Courses	Course Grades	Semester Grade	Year Grade	Diploma	Diploma Grade	Appreciation
Year 1	S1	Analyse 1: Suites Numeriques et Fonctions	18.5	14.75	13.46875	DEUG	14.25	Good
		Algebre 1: Arithmetique dans Z	15					
		Physique 1: Mecanique du Point	12.5					
		Informatique 1	13					
	S2	Analyse 2: Integration	9.5	12.1875				
		Algebre 3: Espace vectoriels, Matrices	11.75					
		Physique 2: Optique 1	12.5					
		Algorithmique 1	15					
Year 2	S3	Programmation 1	15	15.3125	15.03125			
		Algorithmique 2	13					
		Systeme d'exploitation 1	18.5					
		Electronique numerique	14.75					
	S4	Programmation 2	16.5	14.75				
		Structure de Données	13					
		Systeme d'exploitation 2	11					
		Architecture des ordinateurs	18.5					
Year 3	S5	Bases de données	15	16.1875	14.8125	BACHELOR	14.8125	Good
		Compilation	14.25					
		Conception orientée objets	17					
		Programmation orientée objets	18.5					
	S6	Interconnexion des reseaux	12	13.4375				
		Administration des réseaux	15					
		Interfaces homme machine	9.75					
		Projet Tutoré	17					

6.1.5 Teacher interface

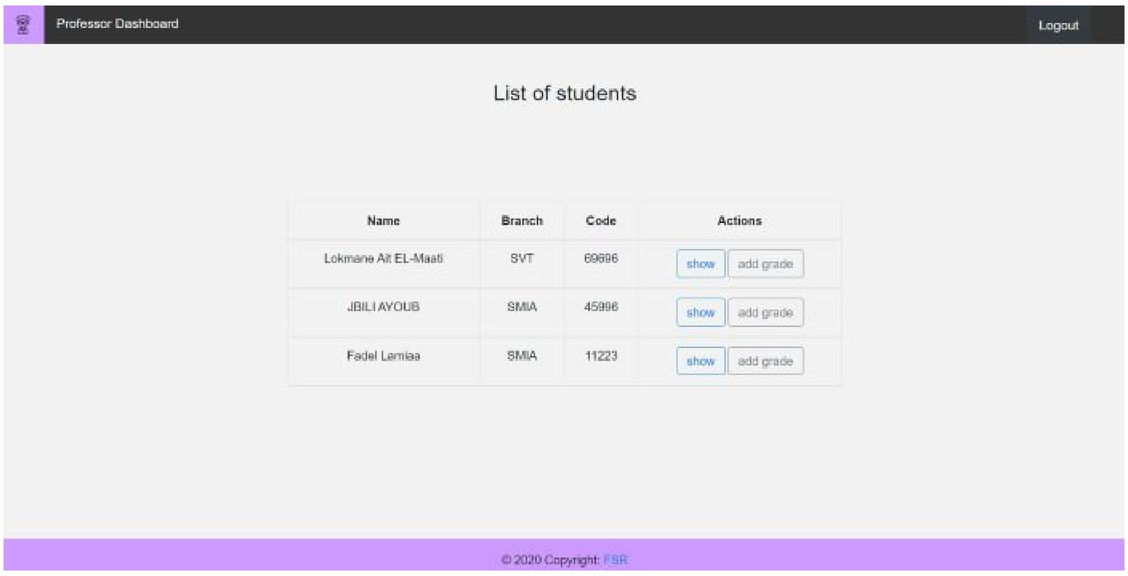
The teacher can log in by writing his email and password he can click on "remember me " so that the next time he logs in will be easier.



After the login, we will go to this page where we have two options either provide a code or to list of all students.



we can search for a specific student and we can also list the students registered in the database.

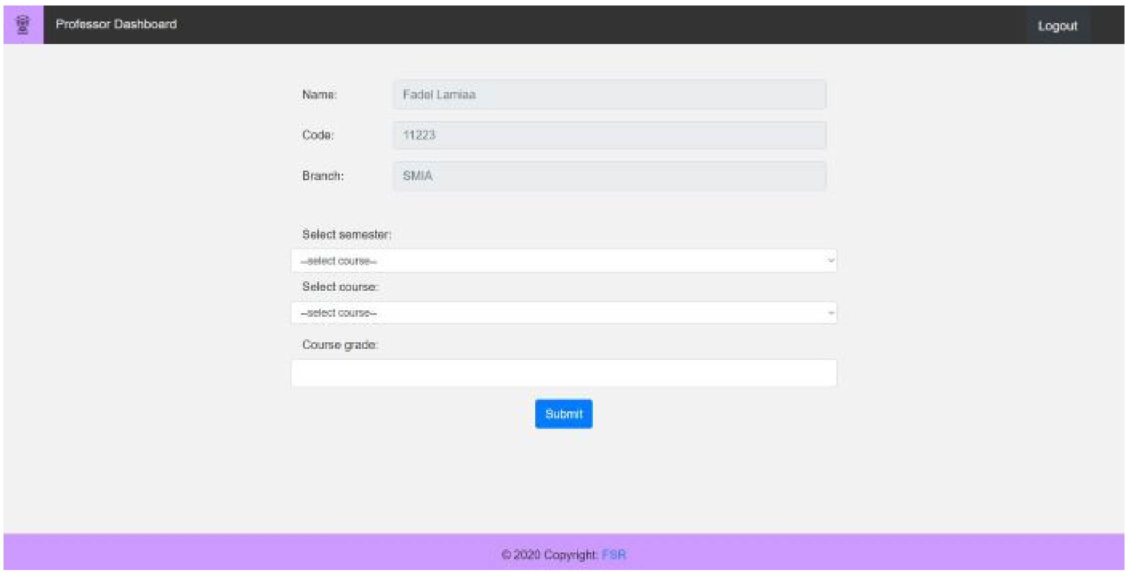


List of students

Name	Branch	Code	Actions
Lokmane Ait EL-Maati	SVT	69696	show add grade
JBILI AYOUB	SMIA	45996	show add grade
Fadel Lamiaa	SMIA	11223	show add grade

© 2020 Copyright: FSR

Here we can also obtain a transcript either a simple one or the detailed version plus the ability to add marks allowed to the instructors after we click on add grade we get the following interface :



Name:	Fadel Lamiaa
Code:	11223
Branch:	SMIA

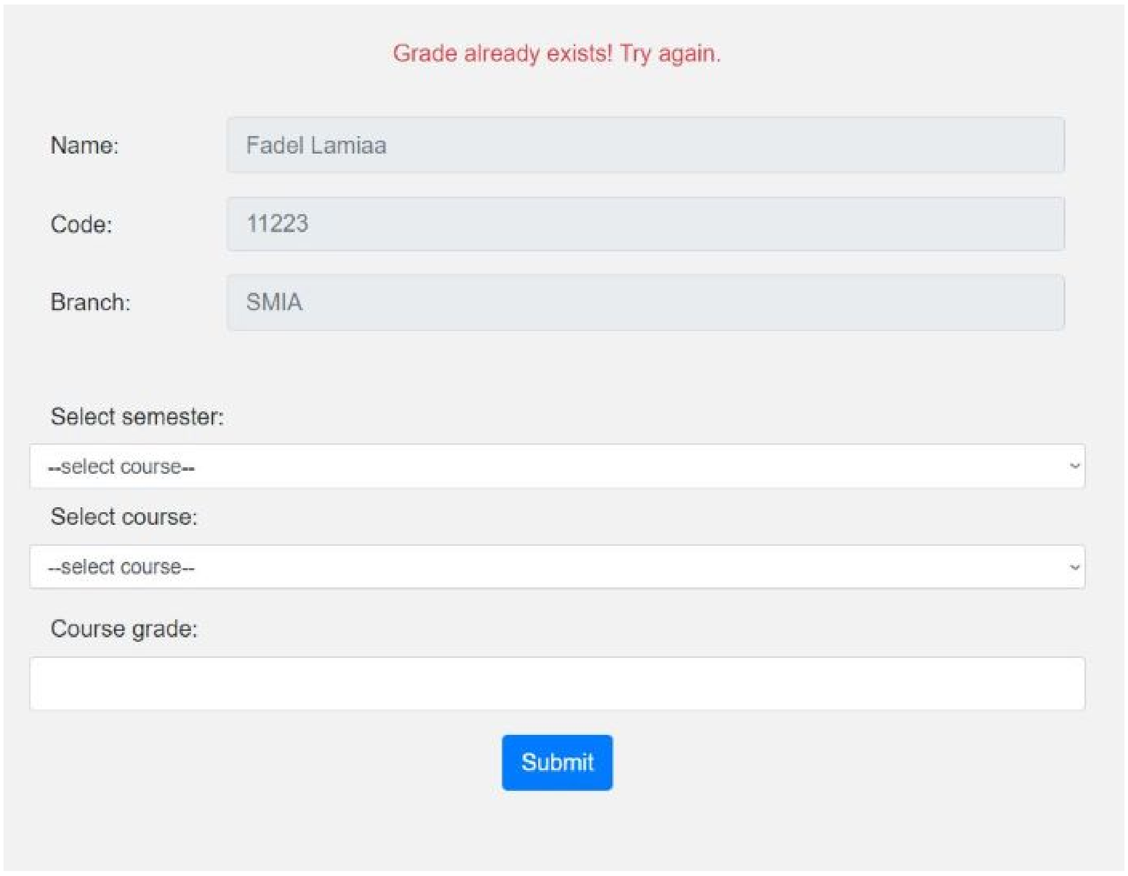
Select semester: --select course--

Select course: --select course--

Course grade:

© 2020 Copyright: FSR

If a given a grade already exists in the system, the student will get the following error message:


Grade already exists! Try again.

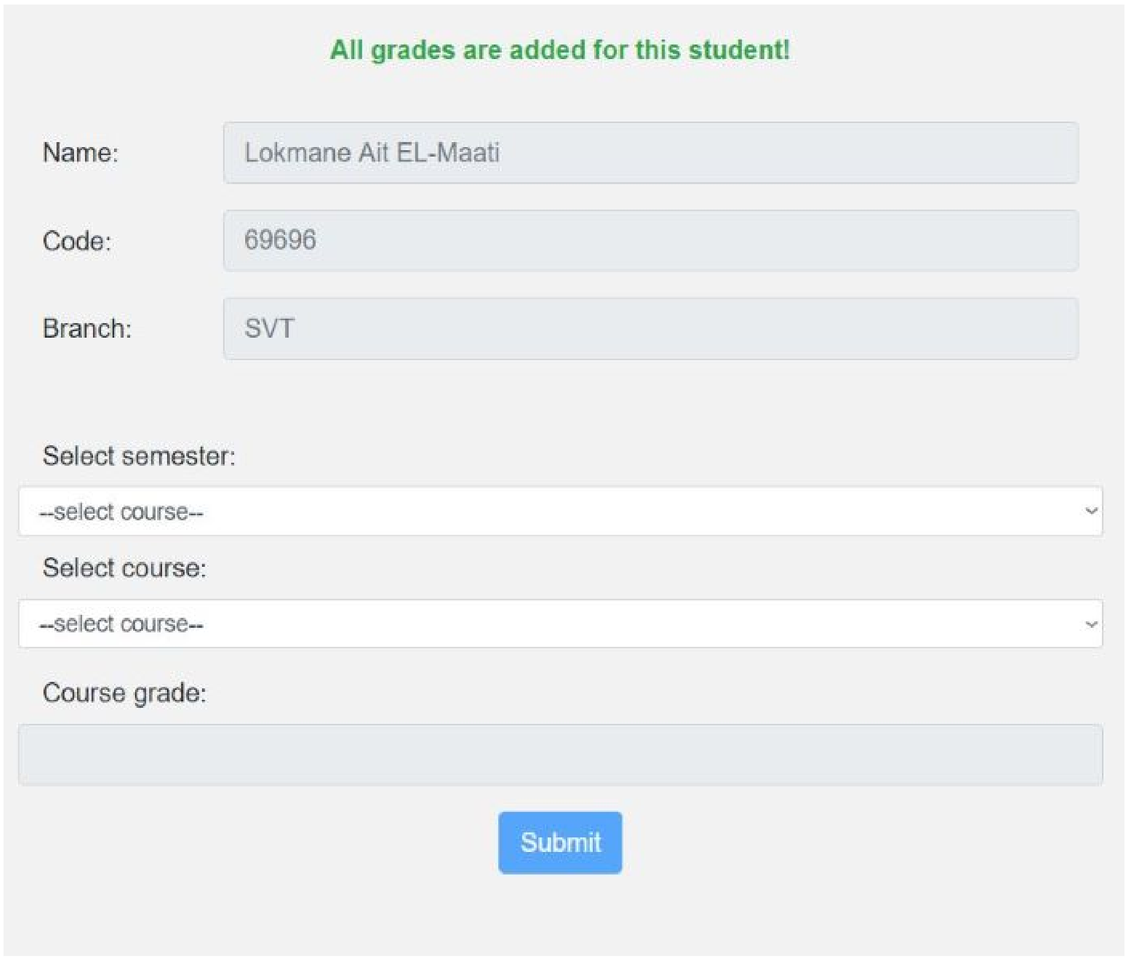
Name:	Fadel Lamiaa
Code:	11223
Branch:	SMIA

Select semester: --select course--

Select course: --select course--

Course grade:

When a teacher filled in all the grades, a success message will show up:


All grades are added for this student!

Name: Lokmane Ait EL-Maati

Code: 69696

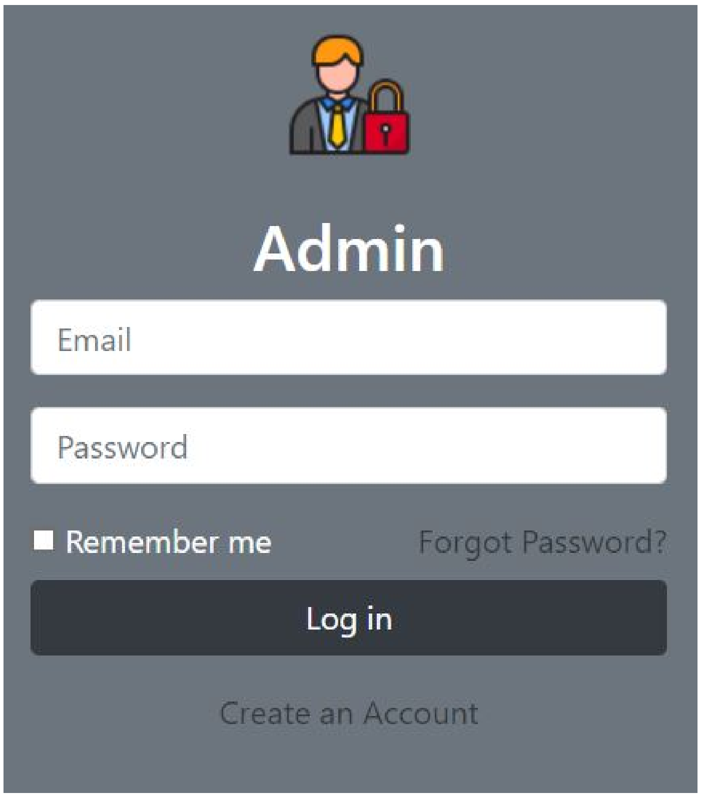
Branch: SVT

Select semester: --select course--

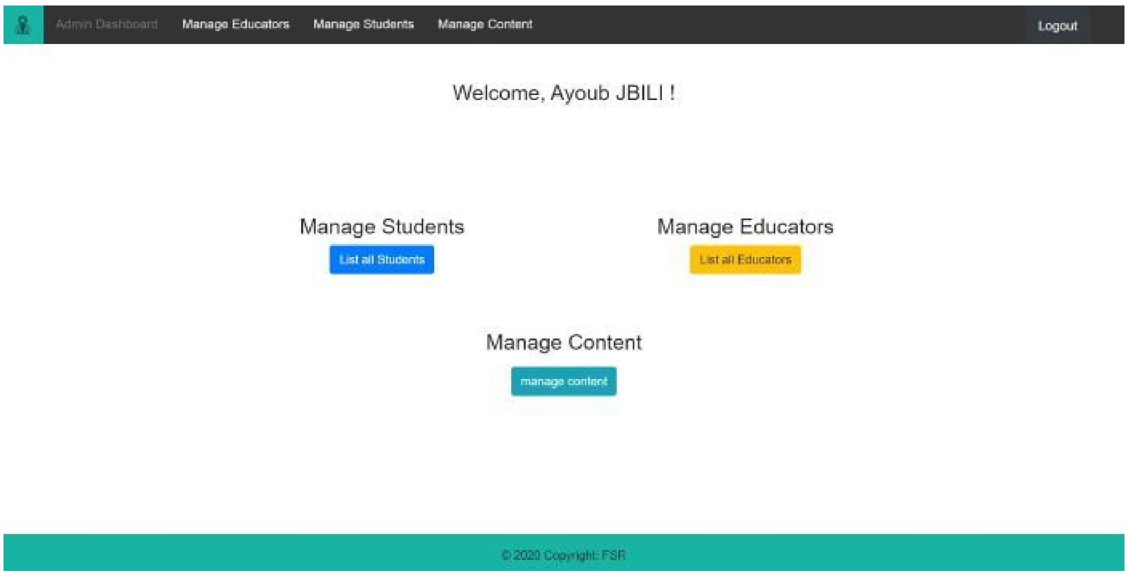
Select course: --select course--

Course grade:

6.1.6 Administrator interface



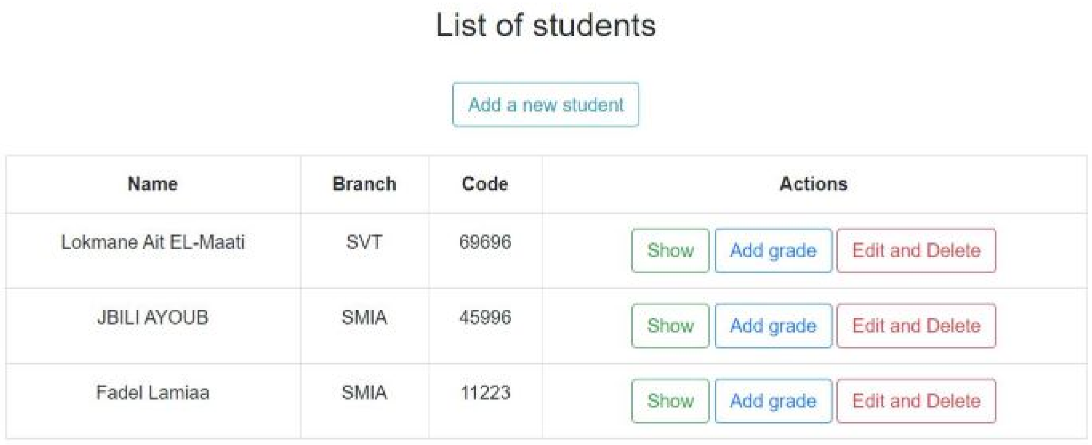
after successful login, he will be redirected to the this page:



Welcome, Ayoub JBILI !

We are presented with three buttons « list all students », « list all educators » and « manage content ».

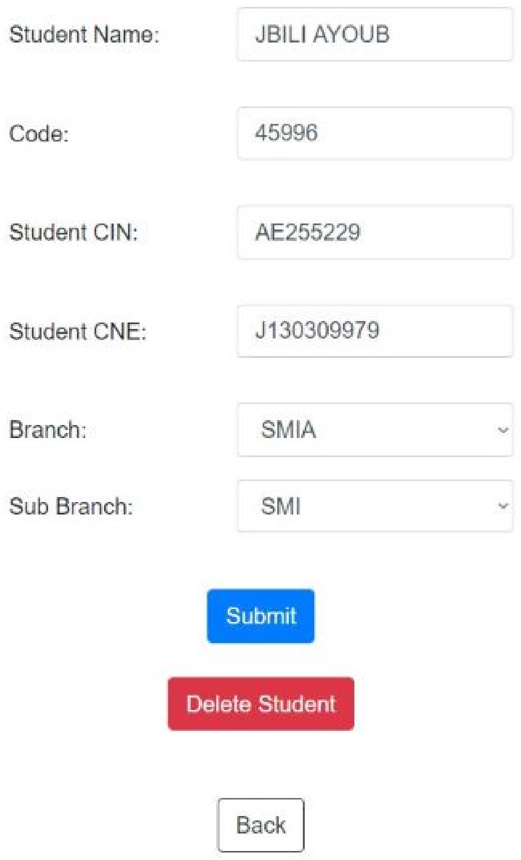
Manage students:


List of students

Add a new student

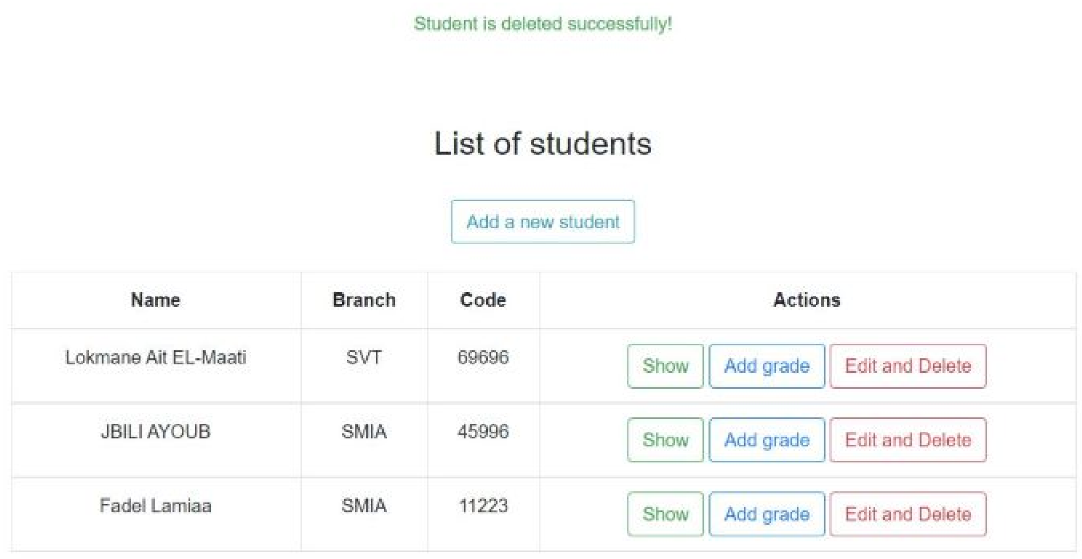
Name	Branch	Code	Actions
Lokmane Ait EL-Maati	SVT	69696	Show / Add grade / Edit and Delete
JBILI AYOUB	SMIA	45996	Show / Add grade / Edit and Delete
Fadel Lamiaa	SMIA	11223	Show / Add grade / Edit and Delete

This page also presents us with a list of the students but with three options, two of them are no different than the list of student's accessed by the professor but the third option is the one where lies the difference that is the admin is allowed to edit and delete anything in the database.



Student Name:	JBILI AYOUB
Code:	45996
Student CIN:	AE255229
Student CNE:	J130309979
Branch:	SMIA
Sub Branch:	SMI

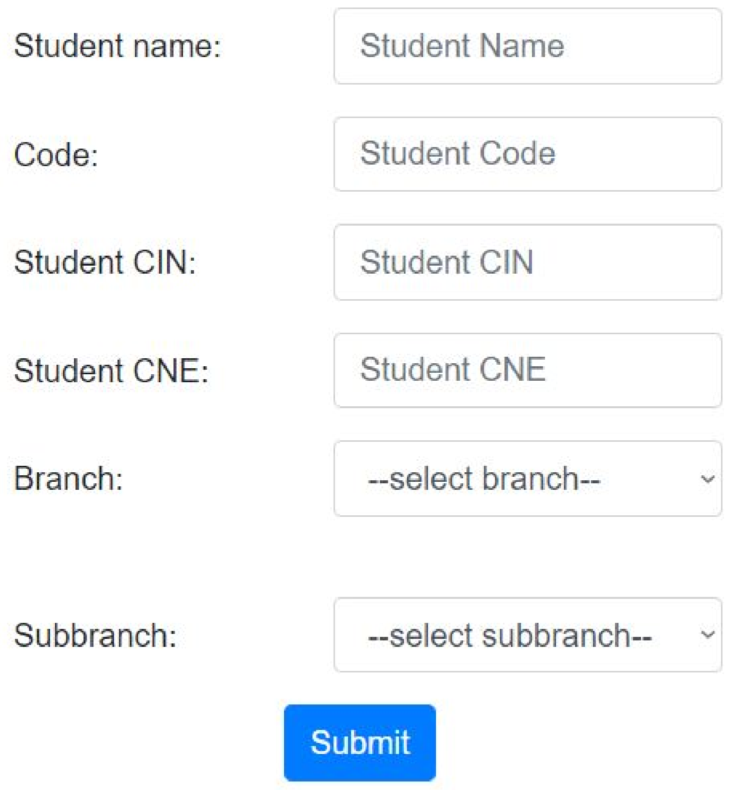
Here the admin can change the student's name, branch, sub-branch, CNI, CNE, and submit the change or can simply delete the student.
After a student is successfully deleted we get this message :


Student is deleted successfully!

List of students

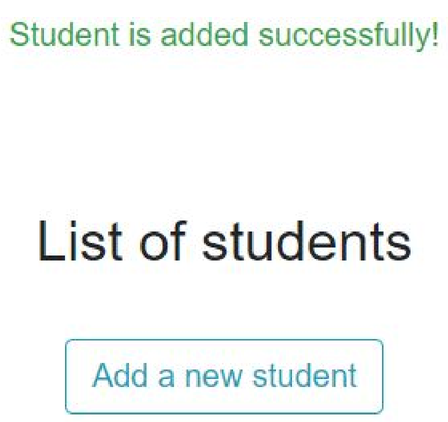
Name	Branch	Code	Actions
Lokmane Ait EL-Maati	SVT	69696	Show · Add grade · Edit and Delete
JBILI AYOUB	SMIA	45996	Show · Add grade · Edit and Delete
Fadel Lamiaa	SMIA	11223	Show · Add grade · Edit and Delete

We can also add a student by clicking on « Add a new student »:



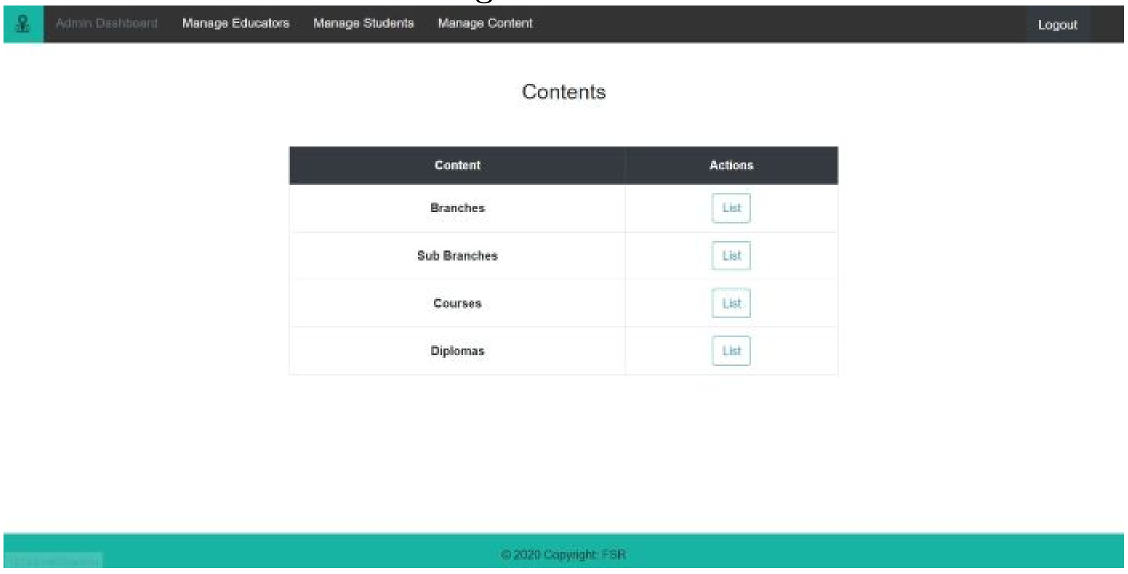
Student name:	Student Name
Code:	Student Code
Student CIN:	Student CIN
Student CNE:	Student CNE
Branch:	--select branch--
Subbranch:	--select subbranch--

After filling in all the cases we submit it then we get this message :


Student is added successfully!

List of students

Manage content:

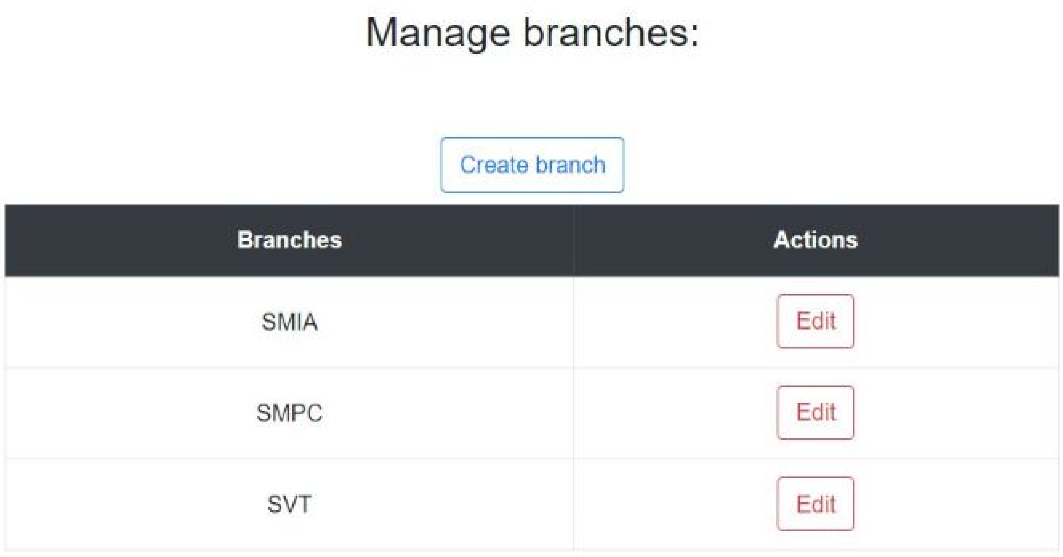


Contents

Content	Actions
Branches	List
Sub Branches	List
Courses	List
Diplomas	List

© 2020 Copyright: FSR

This interface allows us to manage the content like branches, sub-branches, courses and diplomas.

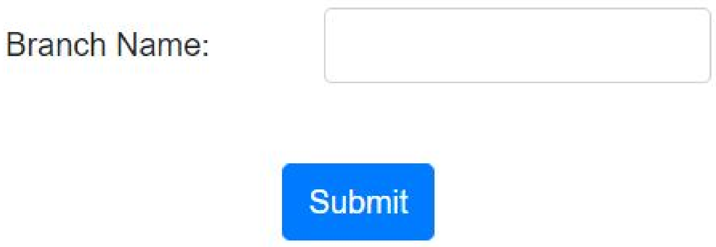


Manage branches:

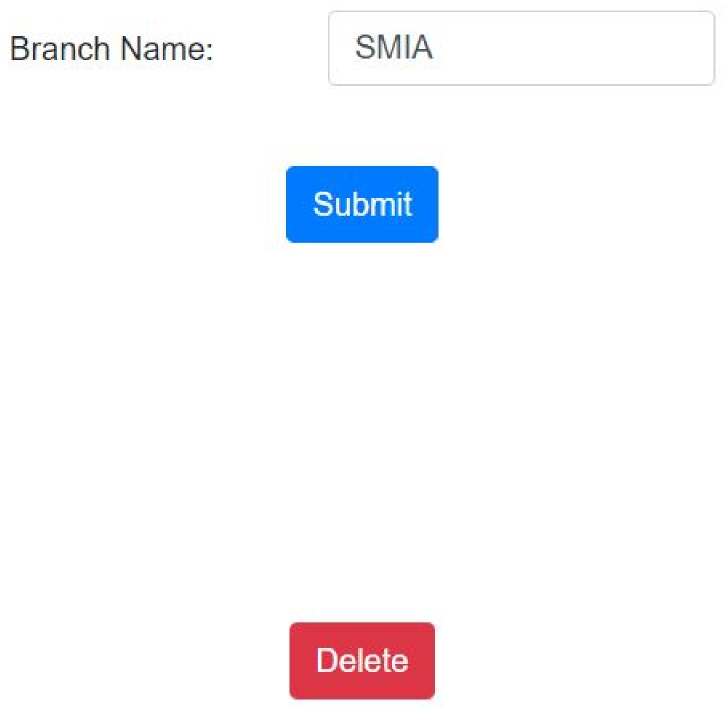
Create branch

Branches	Actions
SMIA	Edit
SMPC	Edit
SVT	Edit

we can create a new branch.

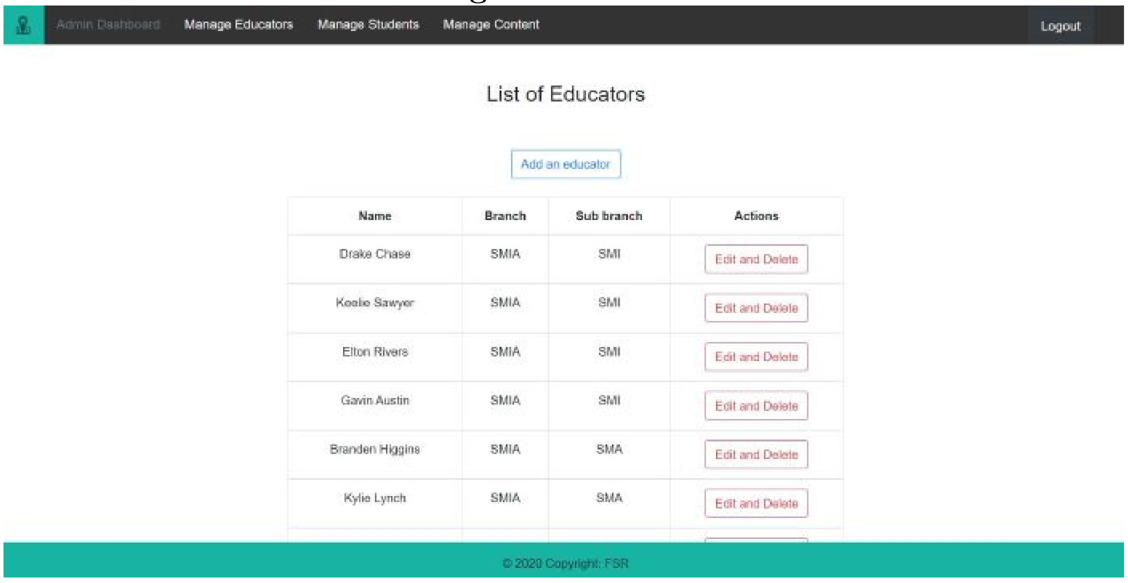


We can also edit an existing branch and we can also delete the branch if it doesn't have any sub-branches assigned to it.



Branch Name:	SMIA

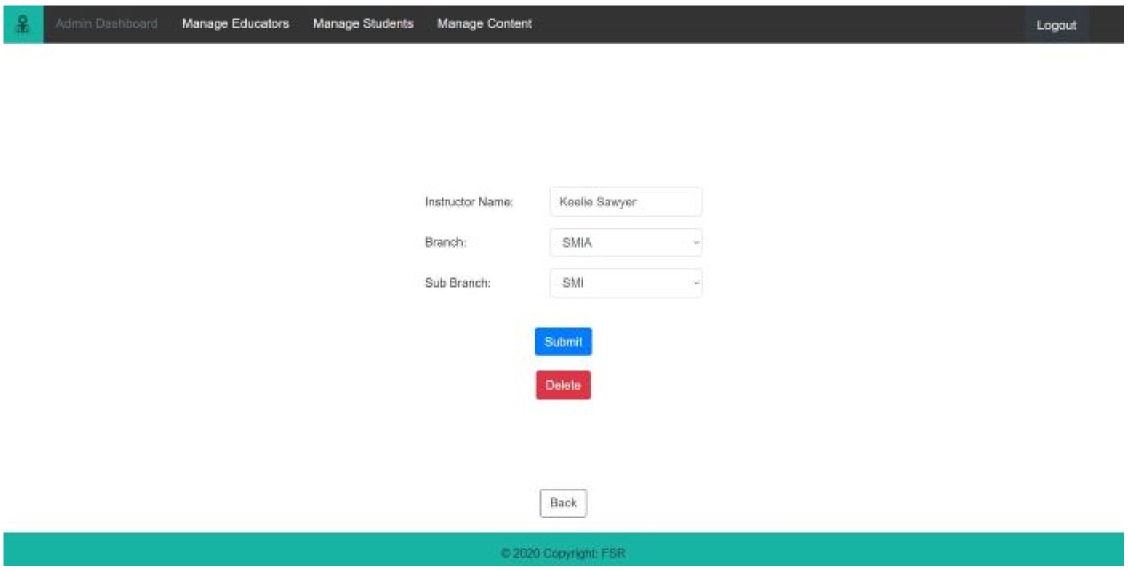
Manage teachers:



List of Educators

Name	Branch	Sub branch	Actions
Drake Chase	SMIA	SMI	Edit and Delete
Keelie Sawyer	SMIA	SMI	Edit and Delete
Elton Rivers	SMIA	SMI	Edit and Delete
Gavin Austin	SMIA	SMI	Edit and Delete
Branden Higgins	SMIA	SMA	Edit and Delete
Kylie Lynch	SMIA	SMA	Edit and Delete

We have the full list of teachers with the options of adding or editing and adding if we select "edit/delete" we get :

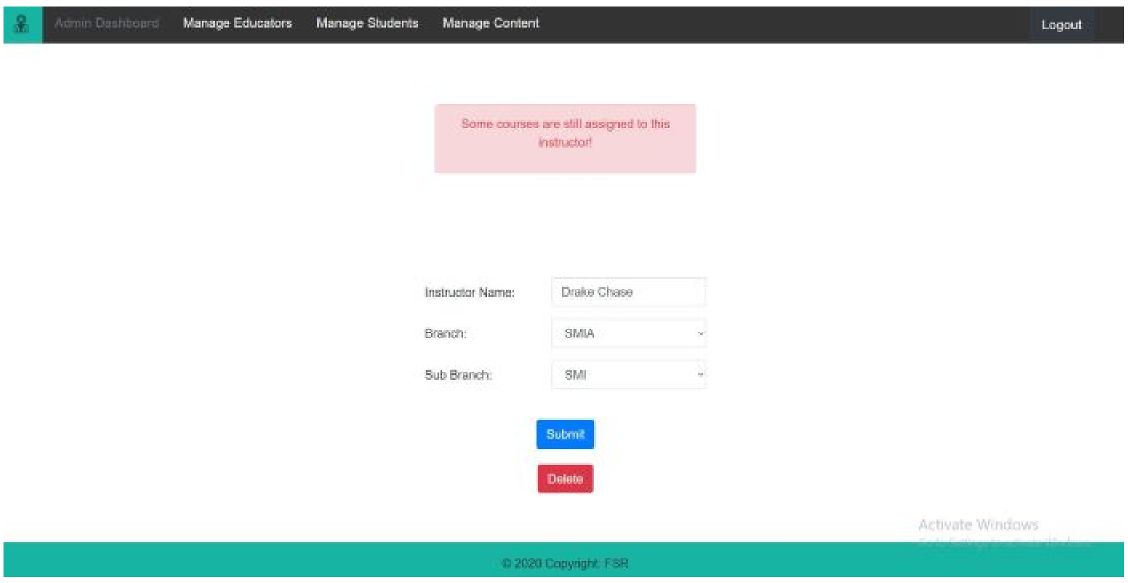


Instructor Name: Keelie Sawyer

Branch: SMIA

Sub Branch: SMI

We will get this alert if the teacher we are trying to delete has some branches assigned to him if not the teacher will be deleted successfully.



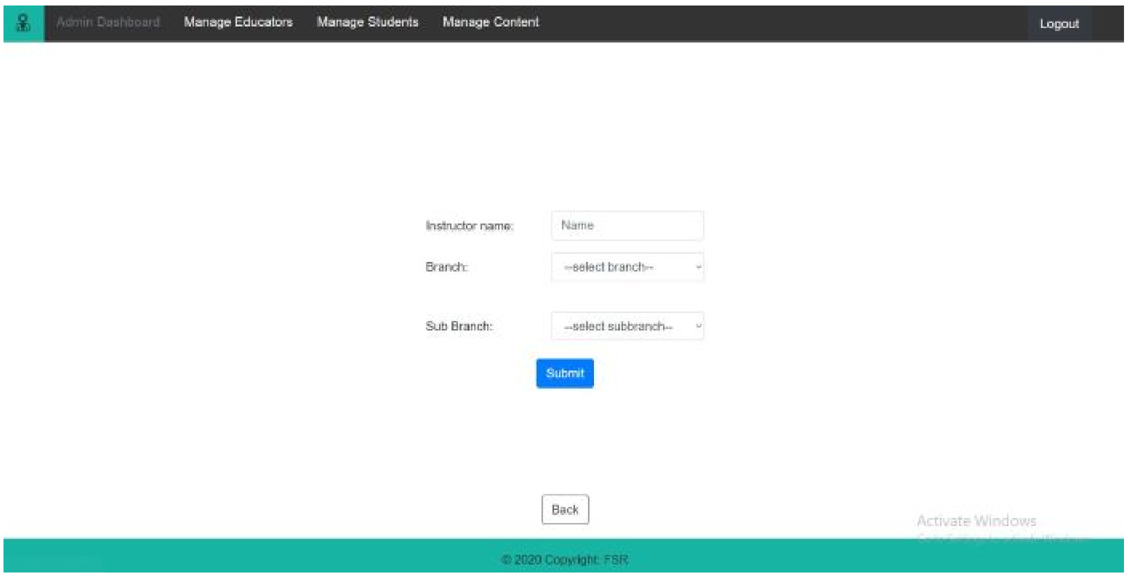
Some courses are still assigned to this instructor!

Instructor Name: Drake Chase

Branch: SMIA

Sub Branch: SMI

We can create a new teacher and assign him to a branch:



Instructor name: Name

Branch: --select branch--

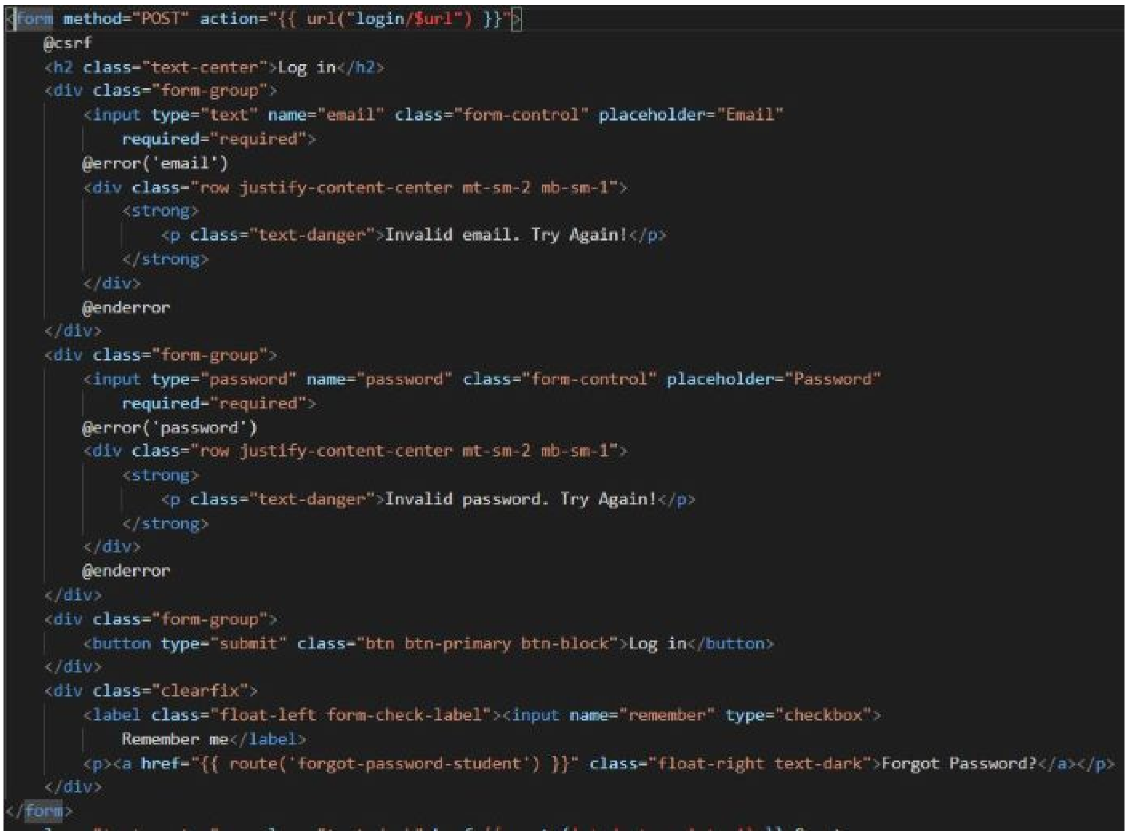
Sub Branch: --select subbranch--

6.1.7 Logout

We can logout different users by clicking on the button in the top right of each page.

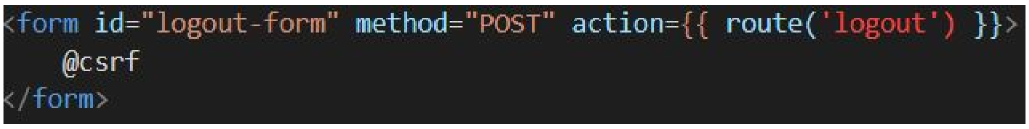
6.2 Code

6.2.1 Login code:



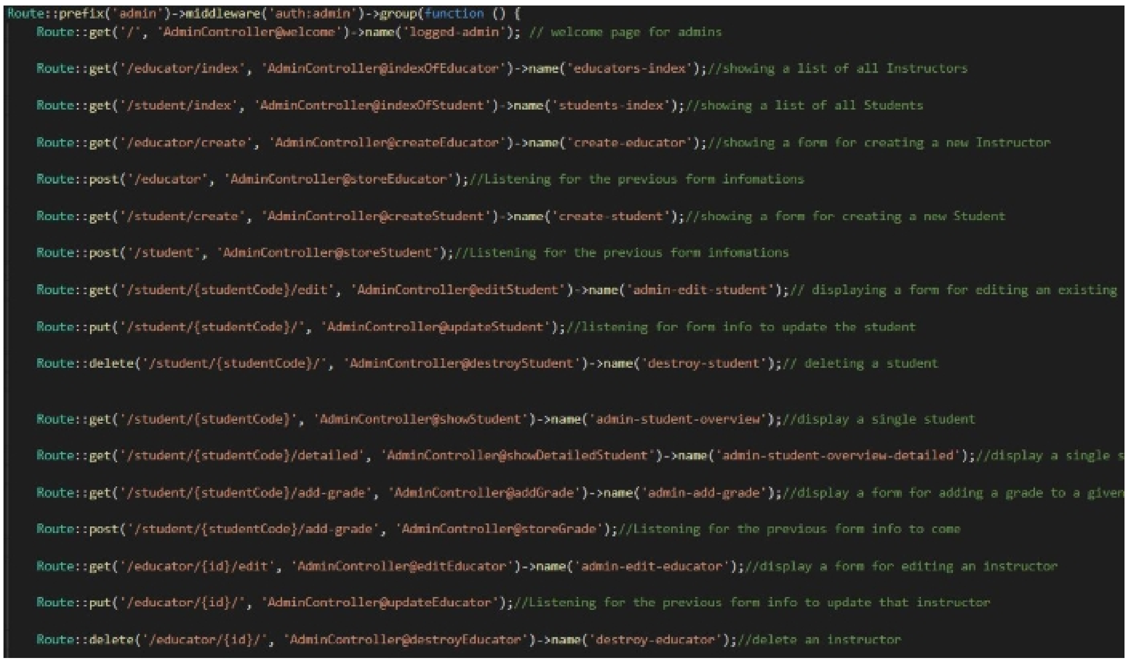
```
<form method="POST" action="{{ url("login/$url") }}">
    @csrf
    <h2 class="text-center">Log in</h2>
    <div class="form-group">
        <input type="text" name="email" class="form-control" placeholder="Email"
            required="required">
        @error('email')
        <div class="row justify-content-center mt-sm-2 mb-sm-1">
            <strong>
                <p class="text-danger">Invalid email. Try Again!</p>
            </strong>
        </div>
        @enderror
    </div>
    <div class="form-group">
        <input type="password" name="password" class="form-control" placeholder="Password"
            required="required">
        @error('password')
        <div class="row justify-content-center mt-sm-2 mb-sm-1">
            <strong>
                <p class="text-danger">Invalid password. Try Again!</p>
            </strong>
        </div>
        @enderror
    </div>
    <div class="form-group">
        <button type="submit" class="btn btn-primary btn-block">Log in</button>
    </div>
    <div class="clearfix">
        <label class="float-left form-check-label"><input name="remember" type="checkbox">
            Remember me</label>
        <p><a href="{{ route('forgot-password-student') }}" class="float-right text-dark">Forgot Password?</a></p>
    </div>
</form>
```

6.2.2 Logout code:



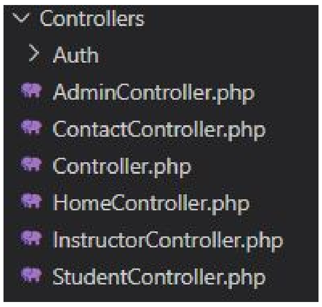
```
<form id="logout-form" method="POST" action={{ route('logout') }}>
    @csrf
</form>
```

6.2.3 Routes:



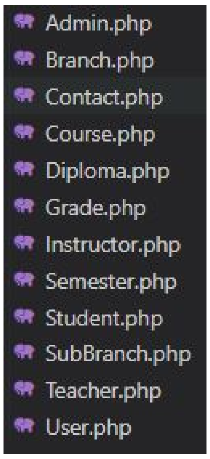
```php
Route::prefix('admin')->middleware('auth:admin')->group(function () {
    Route::get('/', 'AdminController@welcome')->name('logged-admin'); // welcome page for admins

    Route::get('/educator/index', 'AdminController@indexOfEducator')->name('educators-index');//showing a list of all Instructors

    Route::get('/student/index', 'AdminController@indexOfStudent')->name('students-index');//showing a list of all Students

    Route::get('/educator/create', 'AdminController@createEducator')->name('create-educator');//showing a form for creating a new Instructor

    Route::post('/educator', 'AdminController@storeEducator');//Listening for the previous form infomations

    Route::get('/student/create', 'AdminController@createStudent')->name('create-student');//showing a form for creating a new Student

    Route::post('/student', 'AdminController@storeStudent');//Listening for the previous form infomations

    Route::get('/student/{studentCode}/edit', 'AdminController@editStudent')->name('admin-edit-student');// displaying a form for editing an existing

    Route::put('/student/{studentCode}/', 'AdminController@updateStudent');//listening for form info to update the student

    Route::delete('/student/{studentCode}/', 'AdminController@destroyStudent')->name('destroy-student');// deleting a student


    Route::get('/student/{studentCode}', 'AdminController@showStudent')->name('admin-student-overview');//display a single student

    Route::get('/student/{studentCode}/detailed', 'AdminController@showDetailedStudent')->name('admin-student-overview-detailed');//display a single s

    Route::get('/student/{studentCode}/add-grade', 'AdminController@addGrade')->name('admin-add-grade');//display a form for adding a grade to a given

    Route::post('/student/{studentCode}/add-grade', 'AdminController@storeGrade');//Listening for the previous form info to come

    Route::get('/educator/{id}/edit', 'AdminController@editEducator')->name('admin-edit-educator');//display a form for editing an instructor

    Route::put('/educator/{id}/', 'AdminController@updateEducator');//Listening for the previous form info to update that instructor

    Route::delete('/educator/{id}/', 'AdminController@destroyEducator')->name('destroy-educator');//delete an instructor
```

6.2.4 Controllers:



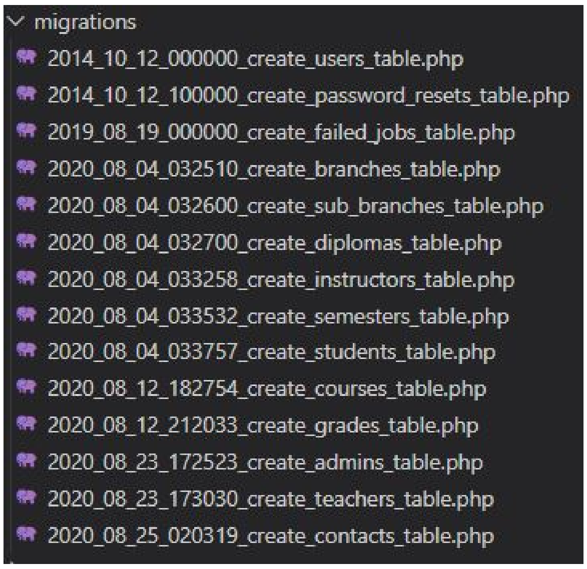
```
Controllers
  Auth
  AdminController.php
  ContactController.php
  Controller.php
  HomeController.php
  InstructorController.php
  StudentController.php
```

6.2.5 Models:



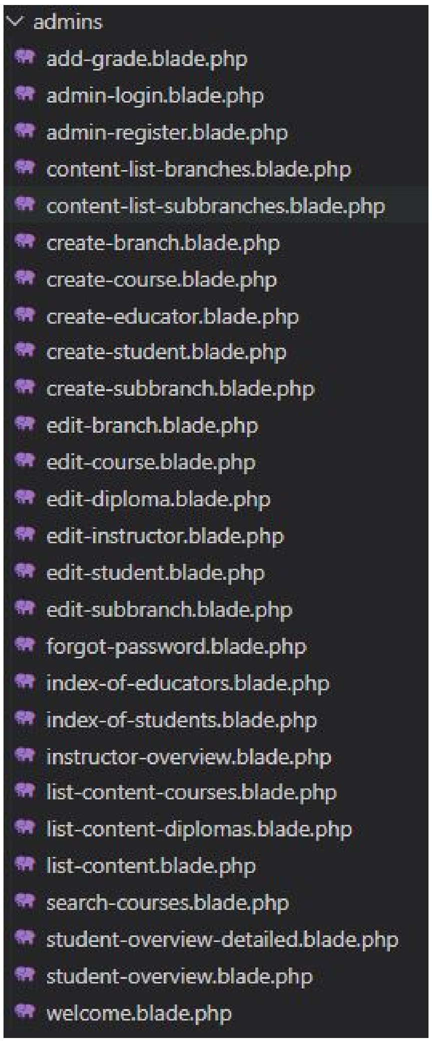
```
Admin.php
Branch.php
Contact.php
Course.php
Diploma.php
Grade.php
Instructor.php
Semester.php
Student.php
SubBranch.php
Teacher.php
User.php
```

6.2.6 Migrations:



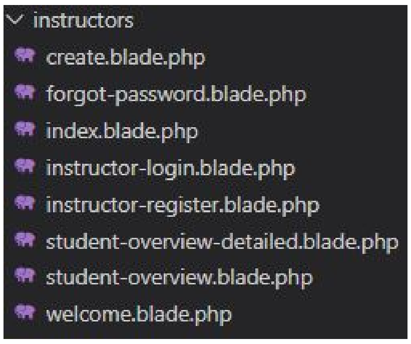
```
migrations
2014_10_12_000000_create_users_table.php
2014_10_12_100000_create_password_resets_table.php
2019_08_19_000000_create_failed_jobs_table.php
2020_08_04_032510_create_branches_table.php
2020_08_04_032600_create_sub_branches_table.php
2020_08_04_032700_create_diplomas_table.php
2020_08_04_033258_create_instructors_table.php
2020_08_04_033532_create_semesters_table.php
2020_08_04_033757_create_students_table.php
2020_08_12_182754_create_courses_table.php
2020_08_12_212033_create_grades_table.php
2020_08_23_172523_create_admins_table.php
2020_08_23_173030_create_teachers_table.php
2020_08_25_020319_create_contacts_table.php
```

6.2.7 Admin views:



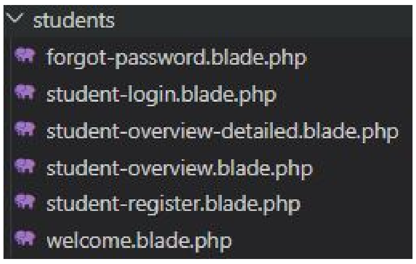
```
admins
add-grade.blade.php
admin-login.blade.php
admin-register.blade.php
content-list-branches.blade.php
content-list-subbranches.blade.php
create-branch.blade.php
create-course.blade.php
create-educator.blade.php
create-student.blade.php
create-subbranch.blade.php
edit-branch.blade.php
edit-course.blade.php
edit-diploma.blade.php
edit-instructor.blade.php
edit-student.blade.php
edit-subbranch.blade.php
forgot-password.blade.php
index-of-educators.blade.php
index-of-students.blade.php
instructor-overview.blade.php
list-content-courses.blade.php
list-content-diplomas.blade.php
list-content.blade.php
search-courses.blade.php
student-overview-detailed.blade.php
student-overview.blade.php
welcome.blade.php
```

6.2.8 Teacher views:



```
instructors
  create.blade.php
  forgot-password.blade.php
  index.blade.php
  instructor-login.blade.php
  instructor-register.blade.php
  student-overview-detailed.blade.php
  student-overview.blade.php
  welcome.blade.php
```

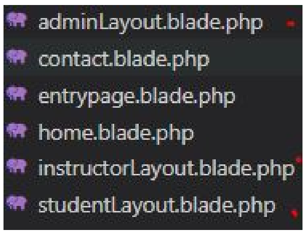
6.2.9 Student views:



```
students
  forgot-password.blade.php
  student-login.blade.php
  student-overview-detailed.blade.php
  student-overview.blade.php
  student-register.blade.php
  welcome.blade.php
```

6.2.10 Layouts:

We use layouts to easily create a standard page template.



```
adminLayout.blade.php
contact.blade.php
entrypage.blade.php
home.blade.php
instructorLayout.blade.php
studentLayout.blade.php
```

General conclusion

The first professional experience is a major step in a career since it allows you to compare your ideas with the reality of the profession. This first experience was materialized by the creation of this application which is made as a grade management system, this project was the occasion of total immersion in the real conditions of the working world. Thus, after a phase of skills development during which we were able to benefit from technical training and behavioral advice, we discovered the profession of developer computer science.
In this context, we had to understand the needs of project leaders and find good suitability of the proposed solution to finally deliver a functional product This profession has a relational component probably as important as the quality of the computer code produced.

Bibliography

[1] https://www.laravel.com/docs/7.x/

[2] Matt Stauffer - Laravel: Up Running: A Framework for Building Modern PHP Apps

[3] https://www.laracasts.com

[4] Martin Bean - Laravel 5 Essentials

[5] Matt Lambert - Learning Bootstrap 4

[6] https://www.w3schools.com/bootstrap4/

[7] https://www.getbootstrap.com