**Student Management System**

Manar Merhi

Table of Contents

[**Introduction** 3](#_Toc199073508)

[**Problem Statement** 4](#_Toc199073509)

[**Design Process** 5](#_Toc199073510)

[**Sitemap:** 5](#_Toc199073511)

[**UI Design:** 6](#_Toc199073512)

[**Color Schemes:** 6](#_Toc199073513)

[**Typography:** 6](#_Toc199073514)

[**Navigation:** 6](#_Toc199073515)

[**Views:** 7](#_Toc199073516)

[**Testing and Debugging:** 16](#_Toc199073517)

[**Conclusion** 17](#_Toc199073518)

# **Introduction**

This Student Management System is a Laravel-based web application built to help educational staff manage students, courses, and grades with ease. Designed with the intent of intranet use in academic institutions, the system provides role-based access for teachers and admins, enabling secure and efficient handling of student-related data. Teachers can add, edit, view, or delete records, assign grades, and generate PDF reports. Admins on the other hand can also manage non-admin user access. With a clean interface and essential tools, the platform simplifies academic administration and supports smooth day-to-day operations.

# **Problem Statement**

Managing student data, courses, and grades can be time-consuming and error-prone when handled manually or through fragmented tools. Many institutions lack a centralized, user-friendly system that enables staff to efficiently oversee academic records and student performance. This project addresses that need by developing a Laravel-based Student Management System tailored for intranet use in educational institutions. It simplifies tasks like enrollment, grade tracking, and user management through an intuitive interface, helping administrators and teachers streamline their daily operations with accuracy and ease.

# **Design Process**

## **Sitemap:**

At the beginning of the project, I created a basic sitemap of this project so that I can more easily manage my files and set my mini-goals one by one. Simple sketches of each page's design and layout to serve as a clear guide during implementation. Obviously there existed a room for improvement in each page, but the insignificance of said room made it so that I could do said improvements on the go during the development phase.

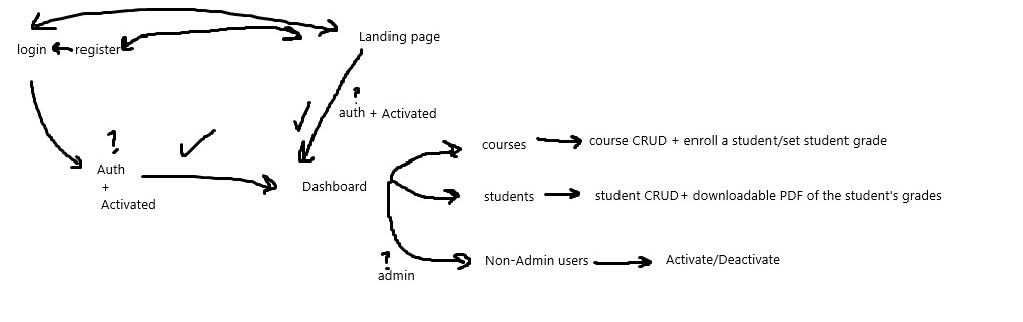
****

Figure 1. Sitemap

# **UI Design:**

## **Color Schemes:**

* Bootstrap was used for styling, the color scheme is in accordance with that; Blue, yellow, and red colors were used for button distinction, light blue and white were used as the main theme of the website.

## **Typography:**

* **Fonts:**
* Instrument Sans
* **Styling:**
* Larger, bold fonts for headings (h1, h2) to draw attention.
* Medium, clean fonts for subheadings and navigational links.
* Smaller fonts for secondary content and error messages, maintaining a balance of focus.
* Readability and aesthetics are prioritized to ensure that users can easily navigate and engage with the website.

## **Navigation:**

* **Header Navigation:**
* Contains the username/website name (based on login status).
* Contains links to major sections: Home, Dashboard, and logout. Or Home, login, and register (based on login status).



Figure 2. Header (logged in)



Figure 3. Header (guest)

* **Footer Design:**
* Simple, includes credits to the Dev, with a dynamic current year being displayed.



Figure 4. Footer

## **Views:**

* **Home Page:**

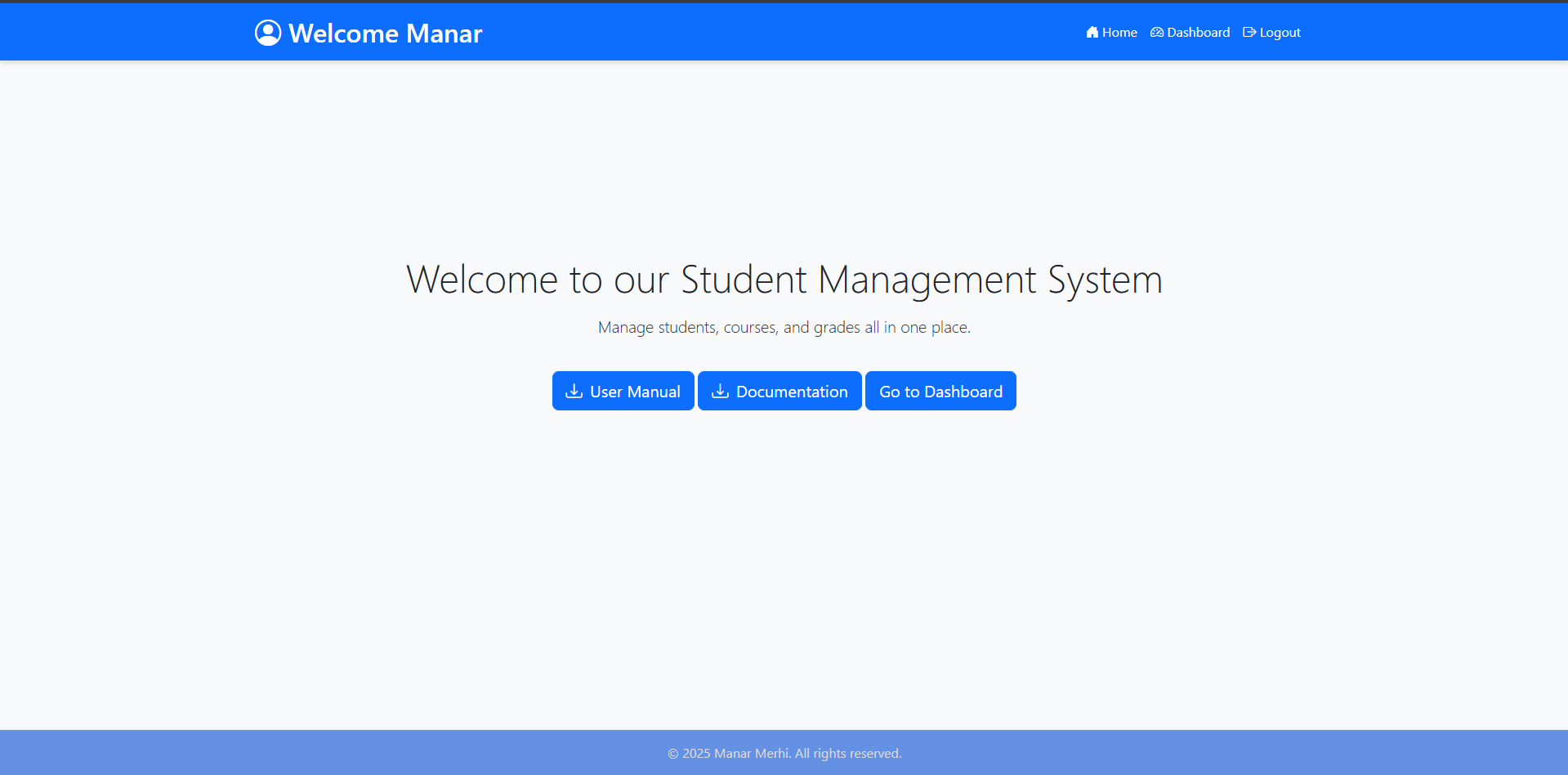


Figure 5. Homepage (user)

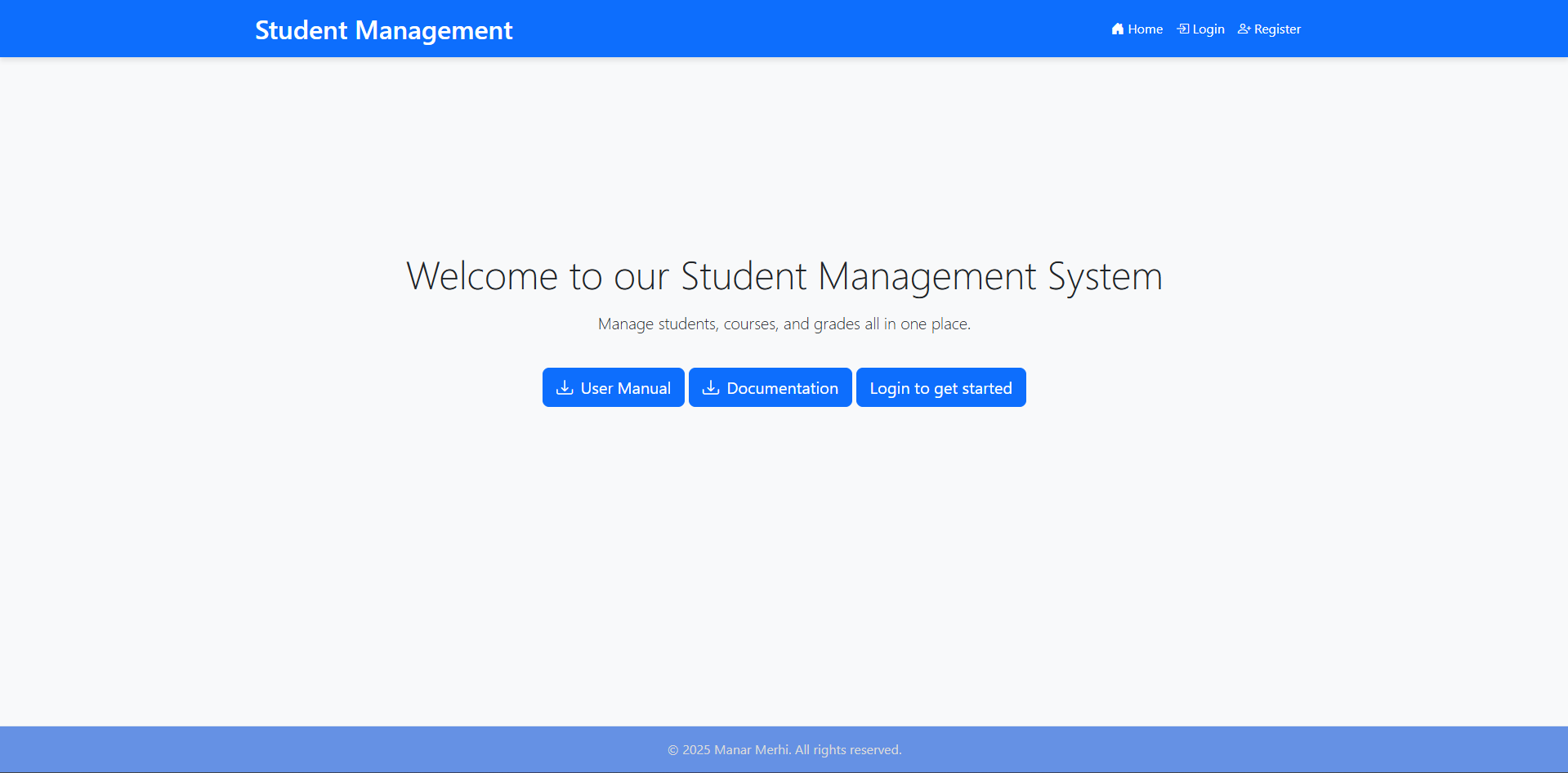


Figure 6. Homepage (guest)

This is what the user first sees, a brief description of what the website is all about, with options to download this very documentation or the user manual.

Additionally, in case the user is logged in there is a link to the dashboard, otherwise there is a link to the login page.

* **Dashboard:**

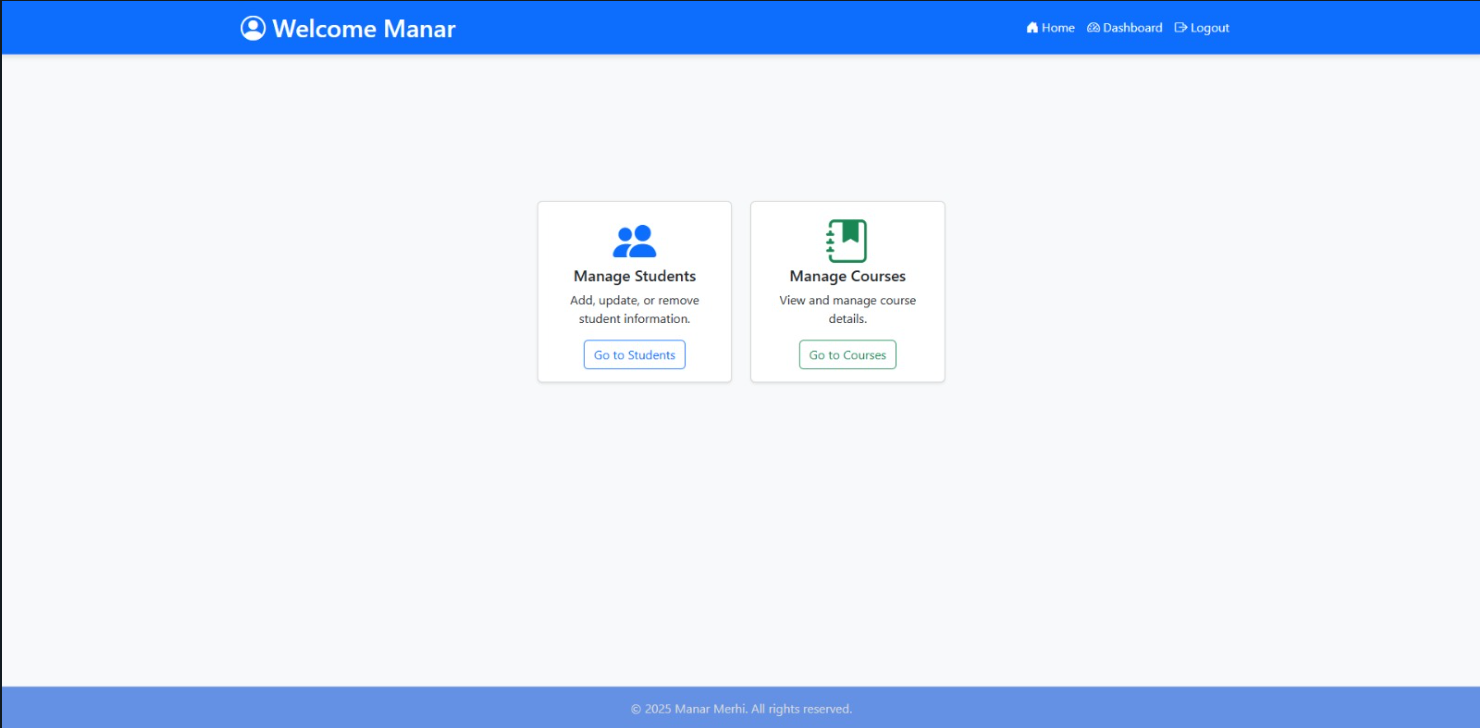


Figure 7. Dashboard (teacher)

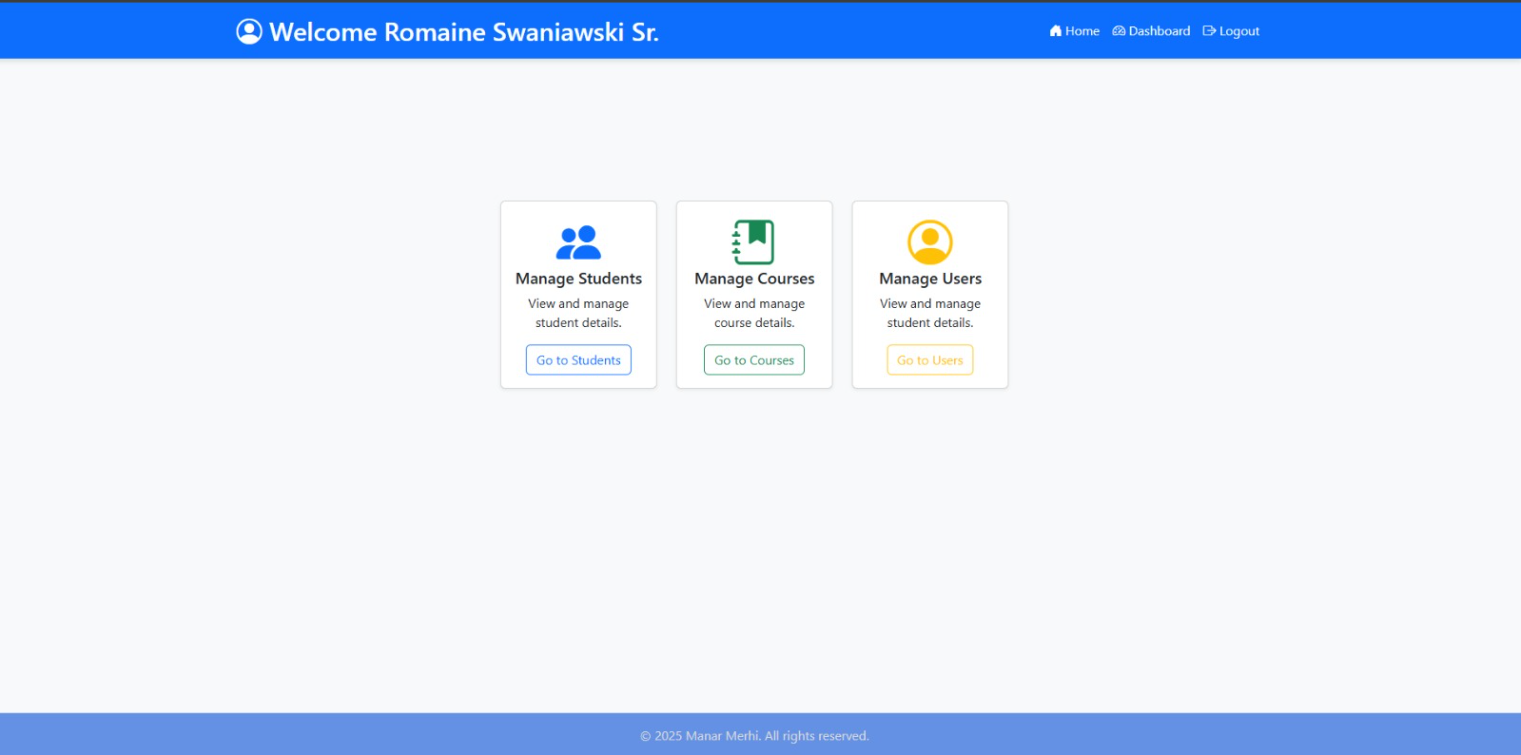


Figure 8. Dashboard (admin)

This is what the user sees, the navbar, the footer, and a section containing the 2 (or 3) management options, the user can choose to manage students/courses or users (in case it’s an admin).

* **Students Page:**

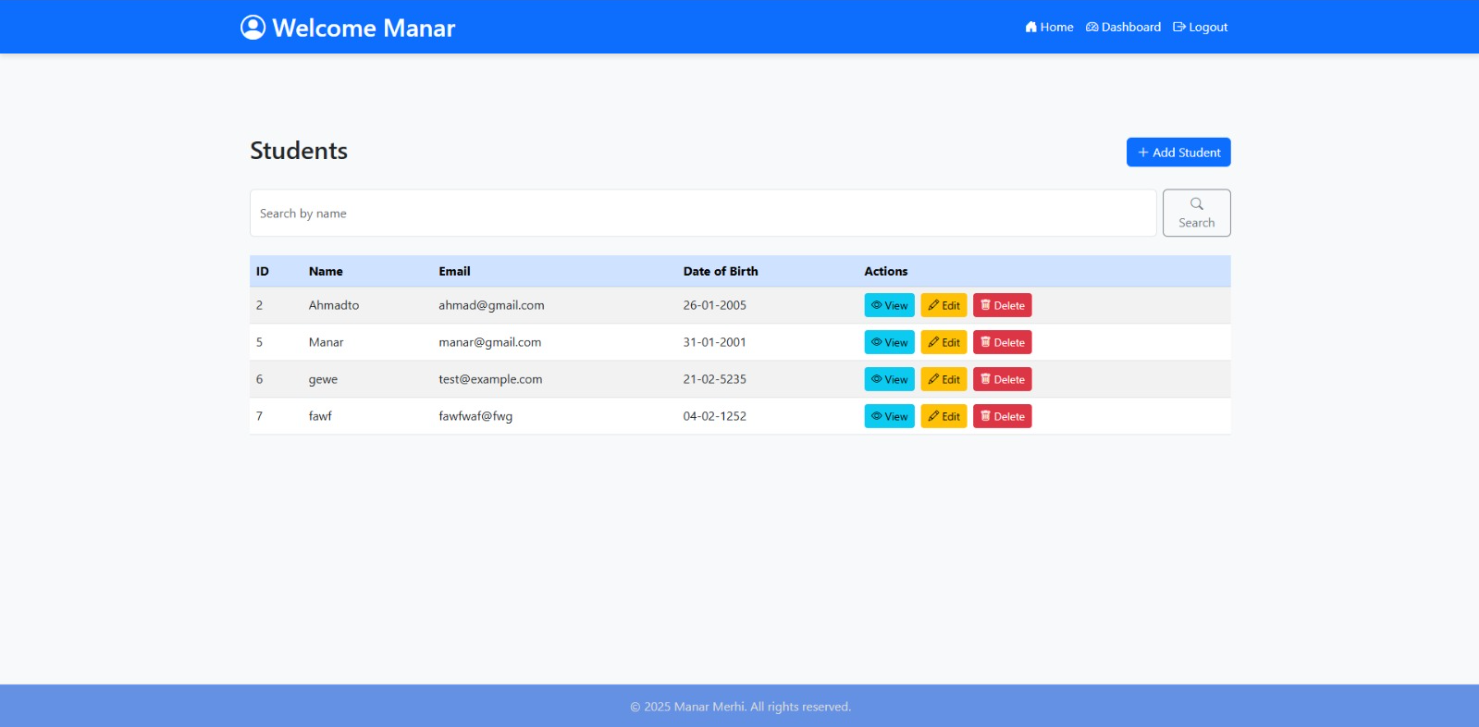


Figure 9. Students page

This is what the user sees when “Go to students” in the dashboard is clicked. A student page showing all the students with the option to search for a student by the name. CRUD functionality is accessible from here, clicking “edit” takes us to the student editing page, “delete” deletes the student after a confirmation, “Add student” takes us to the student adding page, and “view” shows us the profile of this student.

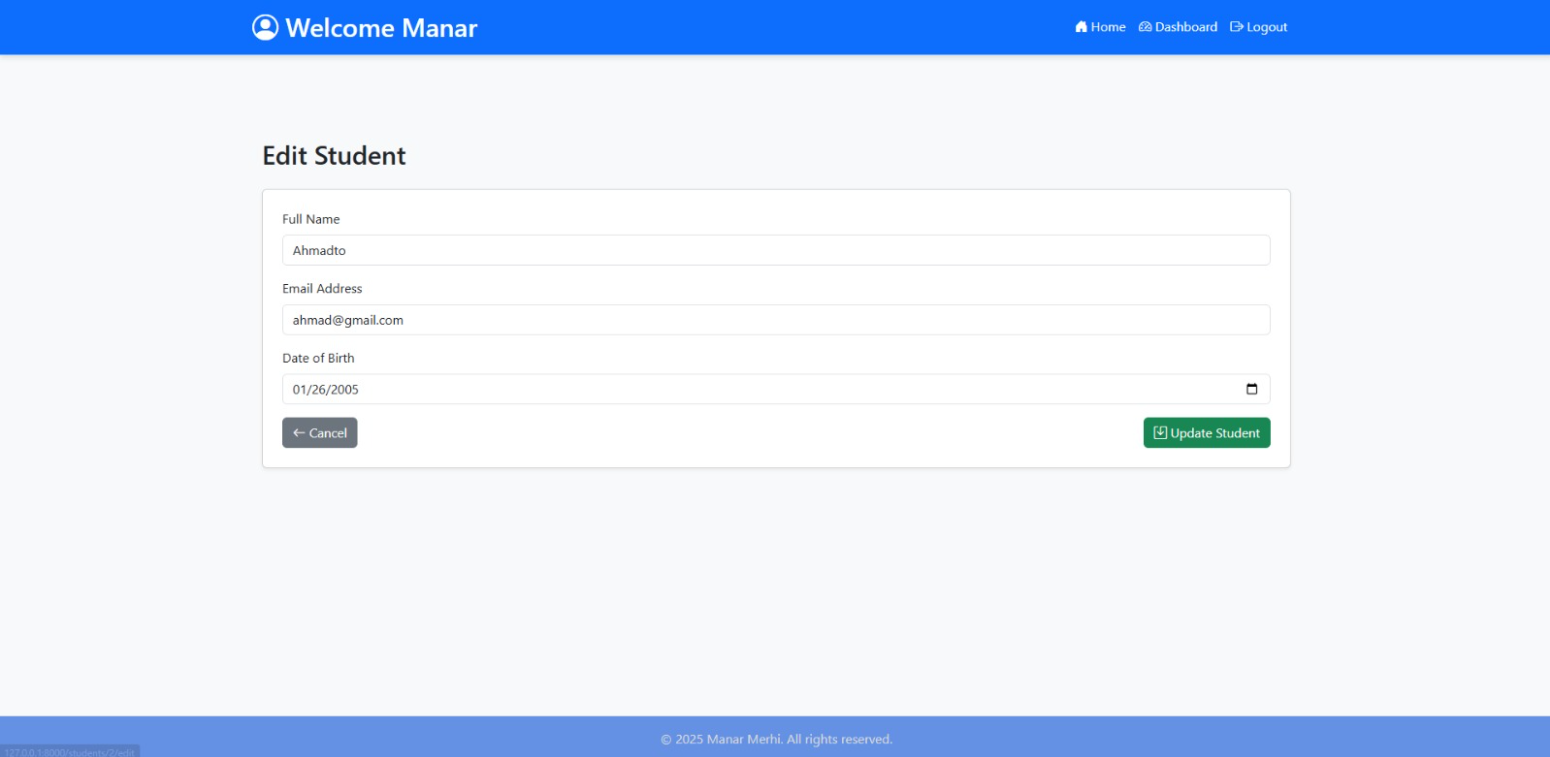


Figure 10. Student Editing page

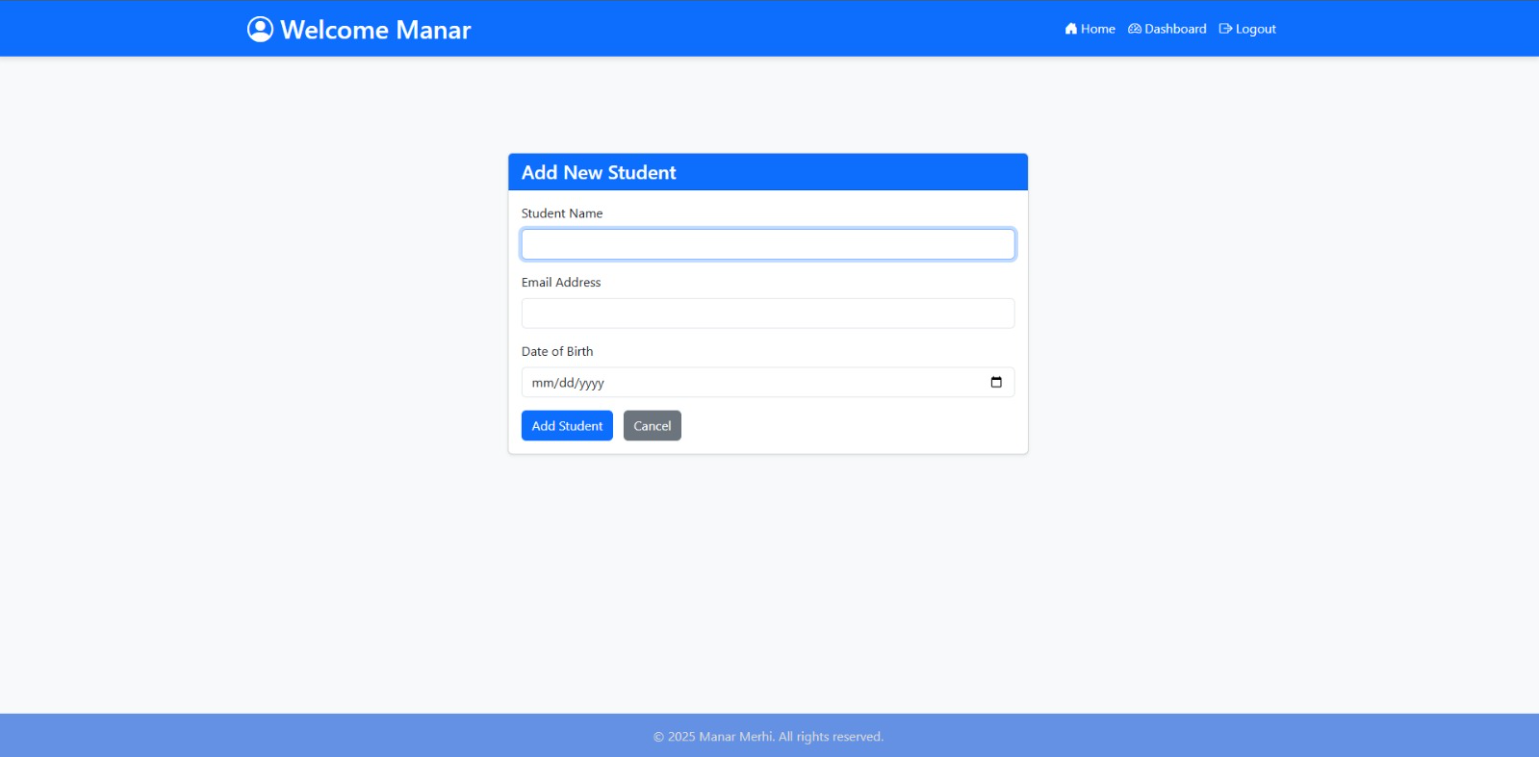


Figure 11. Student Adding page

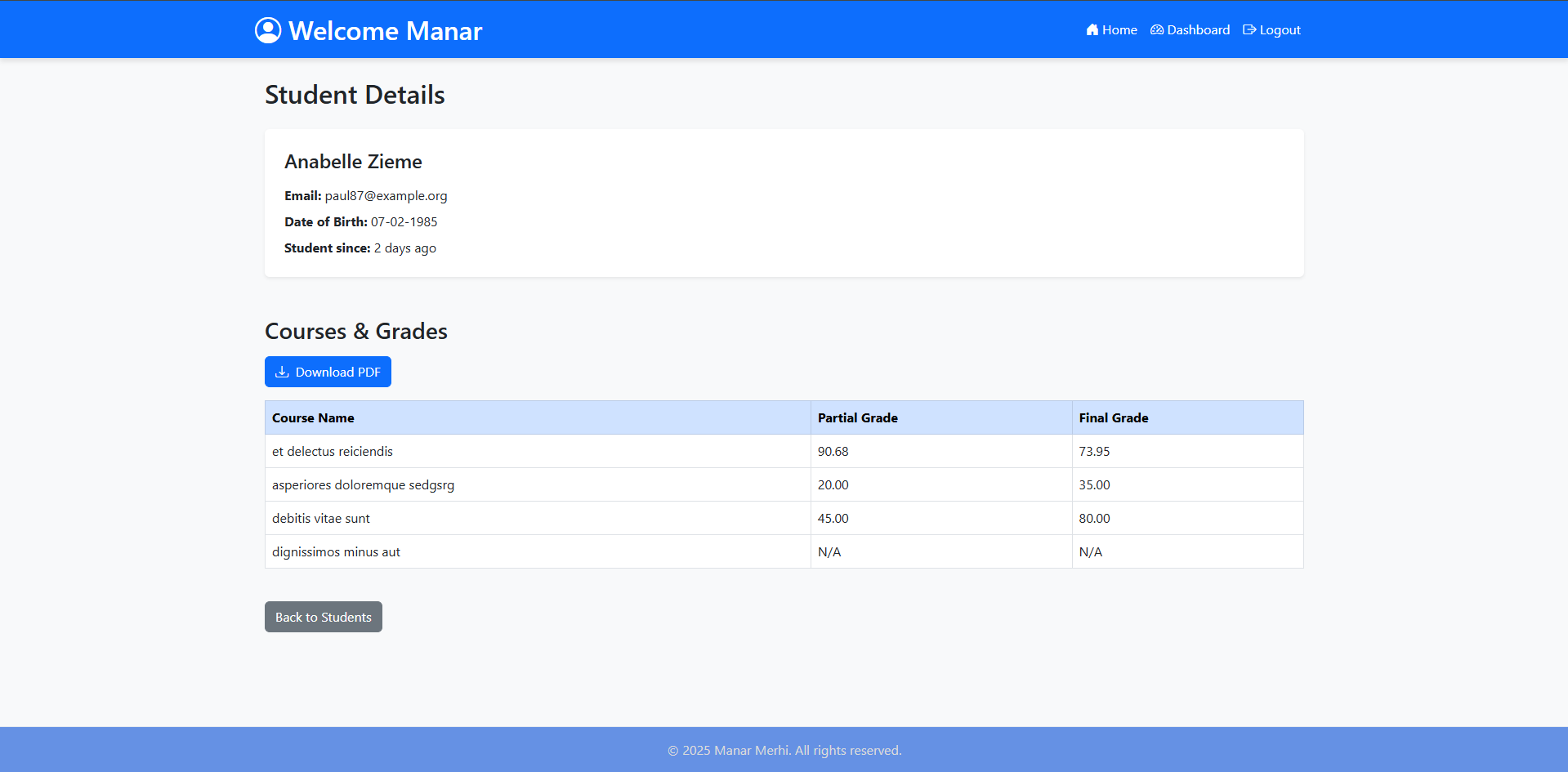


Figure 12. Student Details page

This page showcases the student’s information, as well as a table showcasing all of the courses the student is enrolled in, alongside their partial and final grades for this student. There is an option to download this info as a PDF report.

* **Login/Register:**

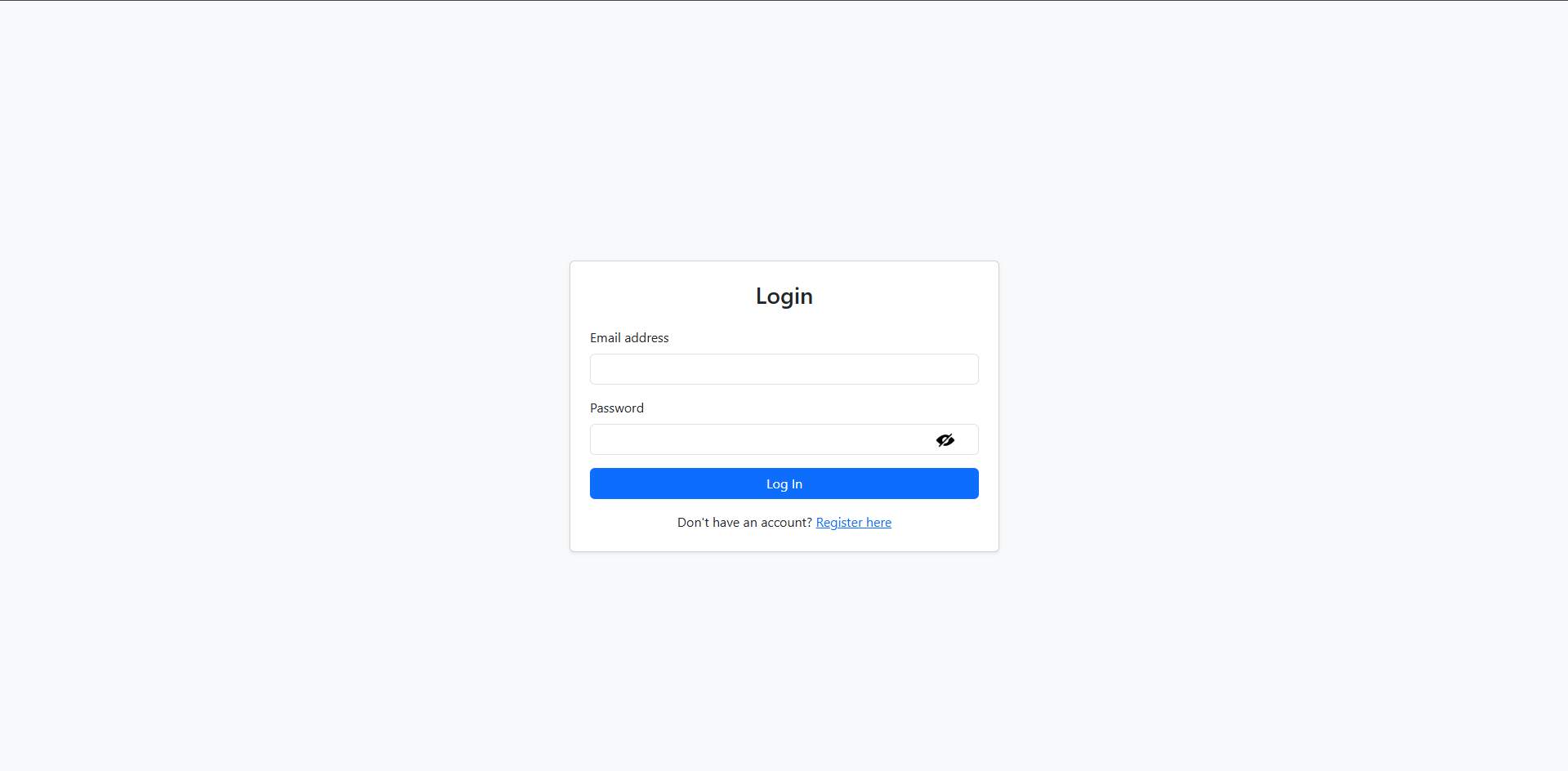


Figure 13. Login page

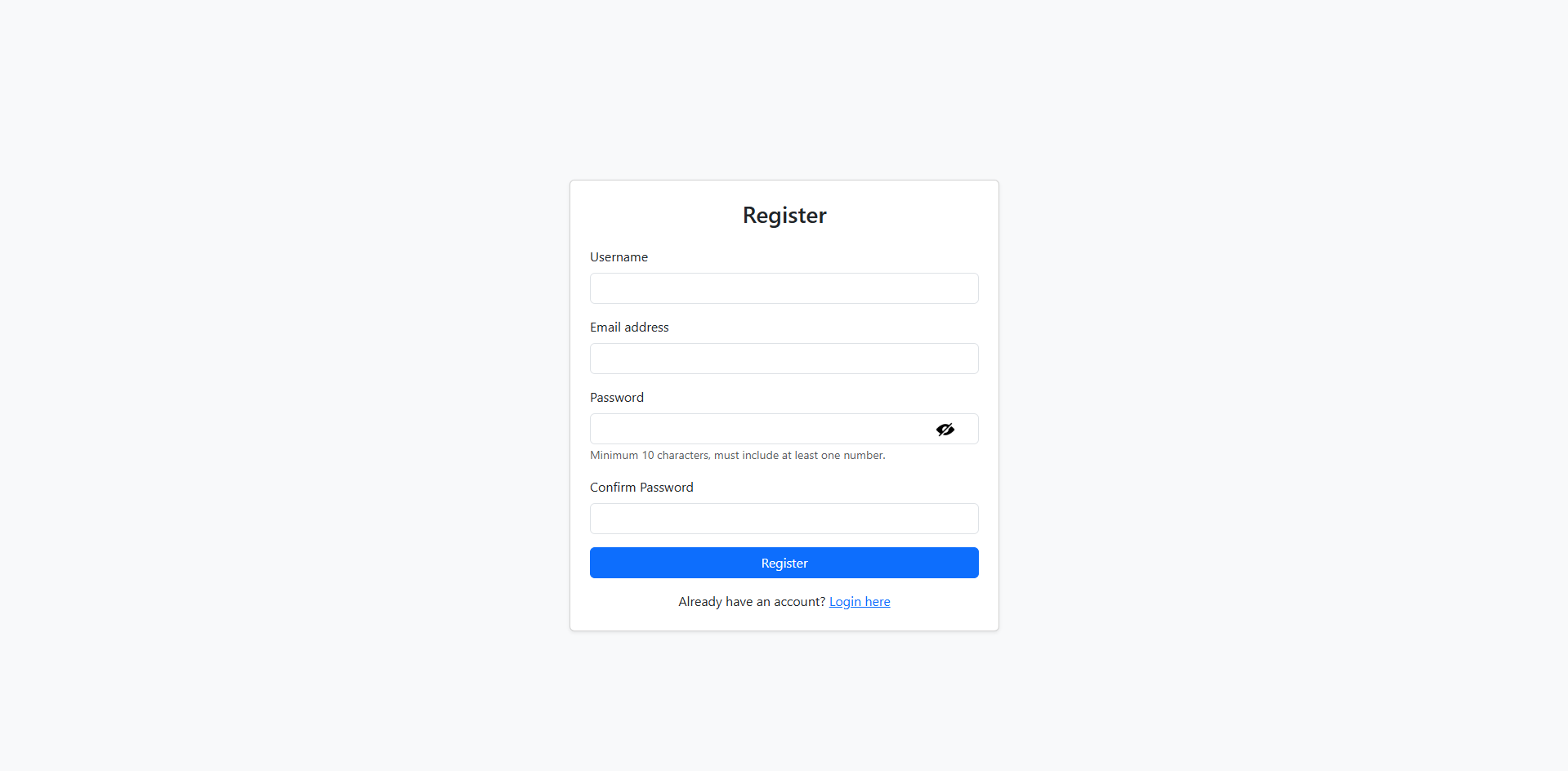


Figure 14. Register page

As you can see in these pages there is the ability to show/hide the password using the eye icon.

* **Users:**

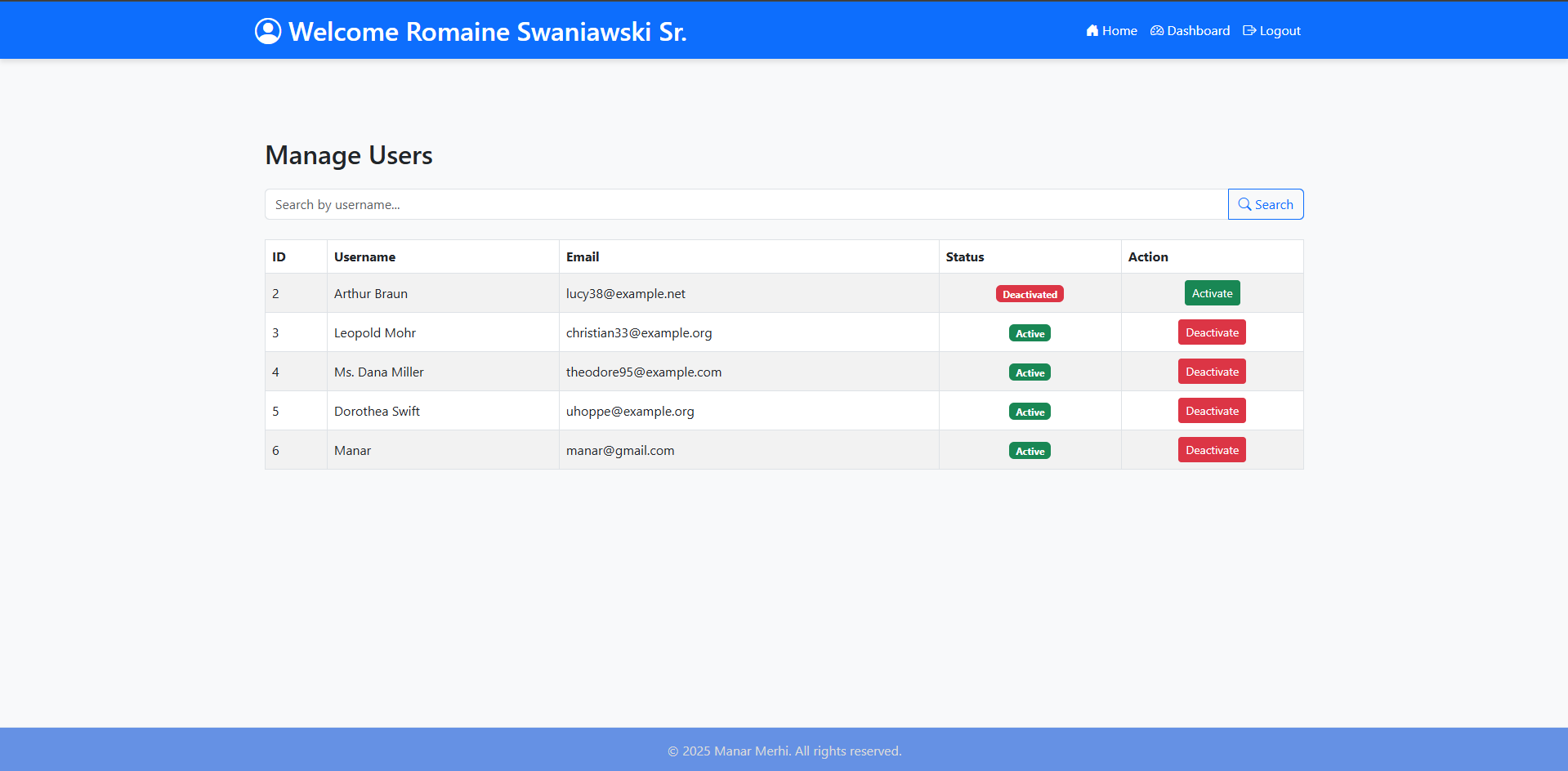


Figure 15. User Management

This is the only page that only admins can access, it contains an action to activate/deactivate any user based on their current status as well as showcasing the user’s information, only non-admin users are shown here obviously.

* **Course page:**

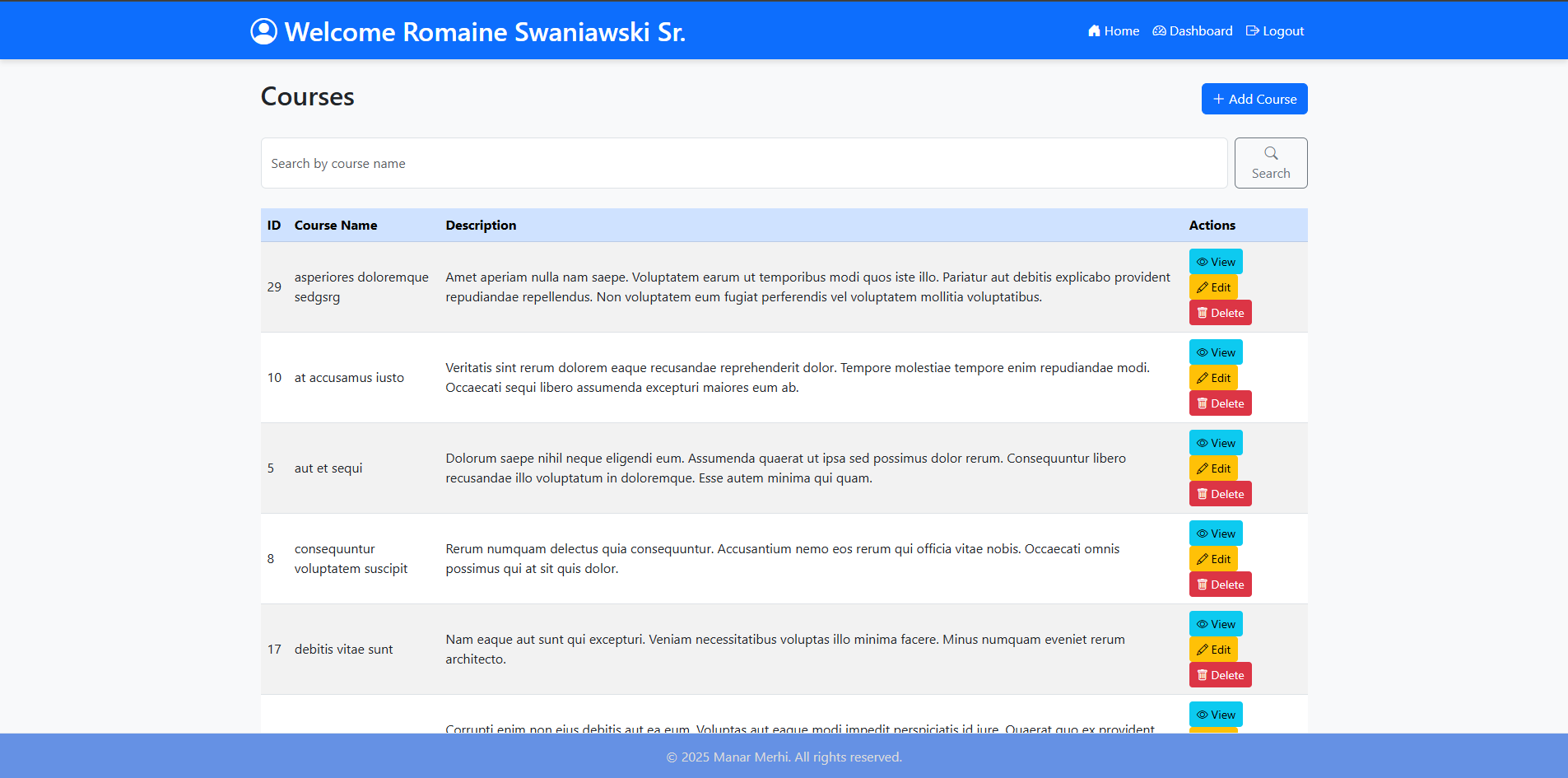
****

Figure 16. Courses page

This is what the user sees when “Go to courses” in the dashboard is clicked. A course page showing all the courses with the option to search for a course by the name. CRUD functionality is accessible from here, clicking “edit” takes us to the course editing page, “delete” deletes the course after a confirmation, “Add Course” takes us to the course adding page, and “view” shows us the information of that course.

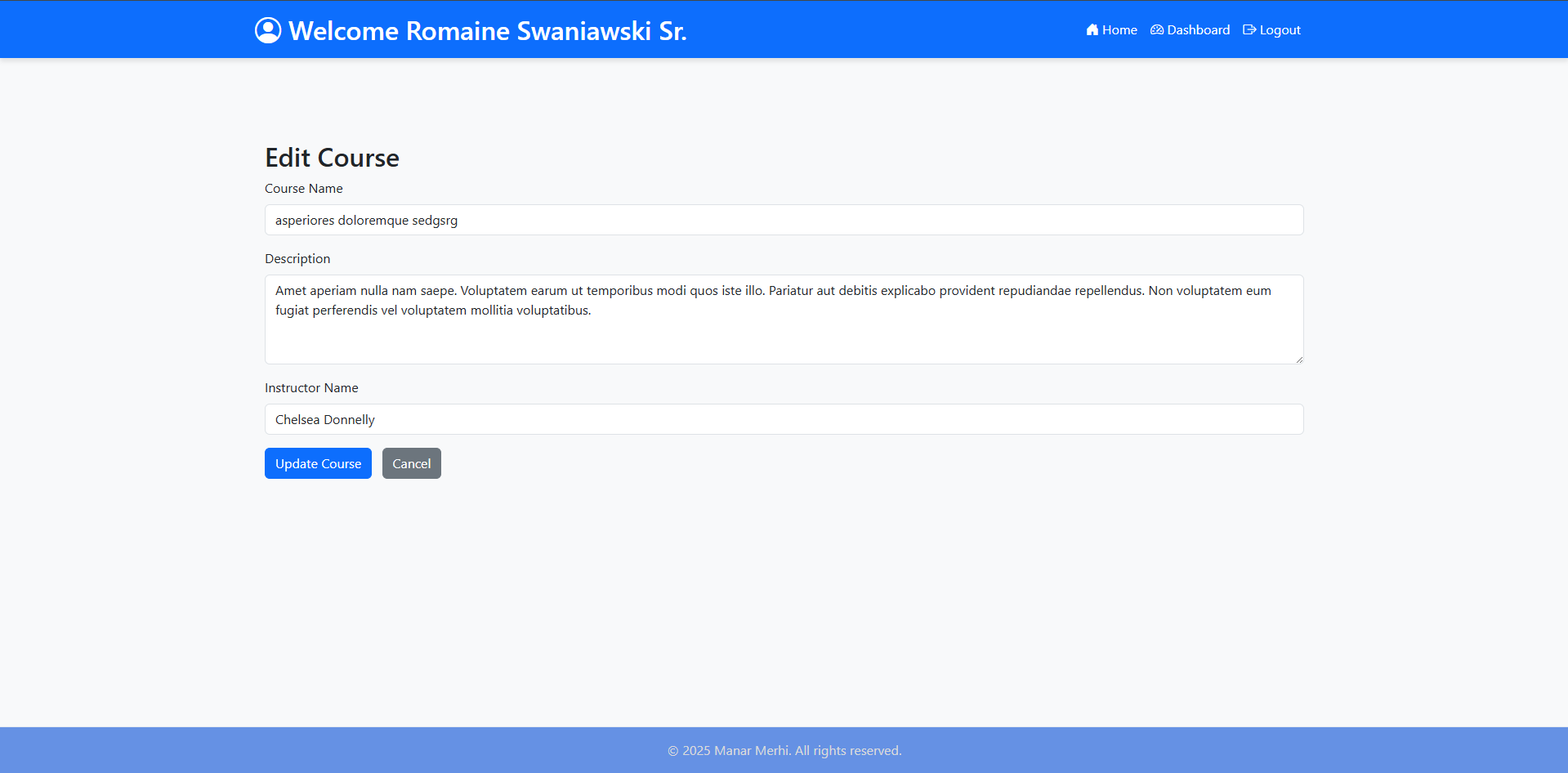


Figure 17. Course Editing page

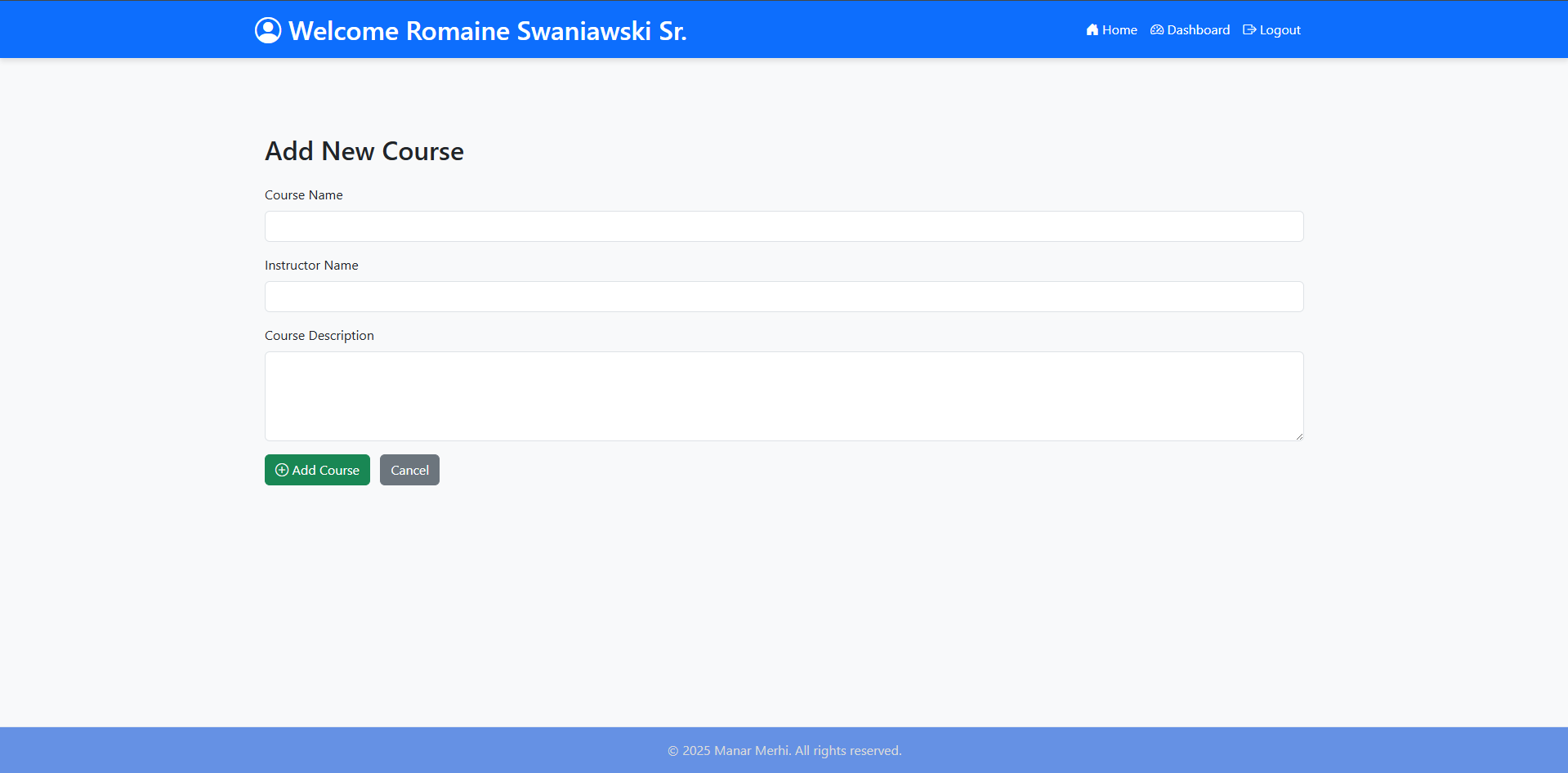


Figure 18. Course Adding page

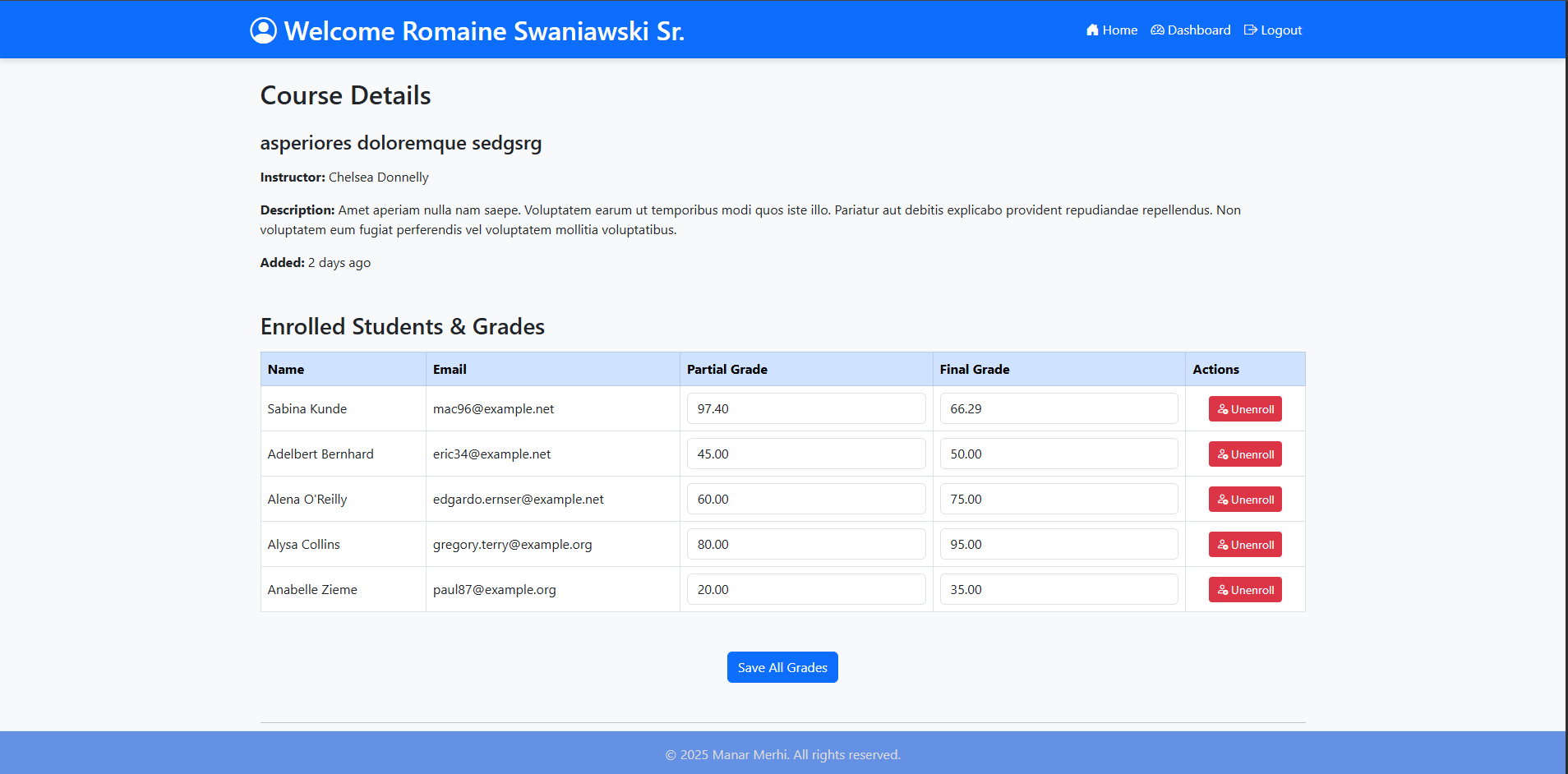


Figure 19. Course viewing 1

This is the landing view of the course details page, it showcases all of the course’s details, as well as all of the enrolled students with the option to unenroll any of them, it also showcases the partial and final grades of those enrolled students which can be edited on the spot, clicking “Save All Grades” saves all of the grades in the table simultaneously.

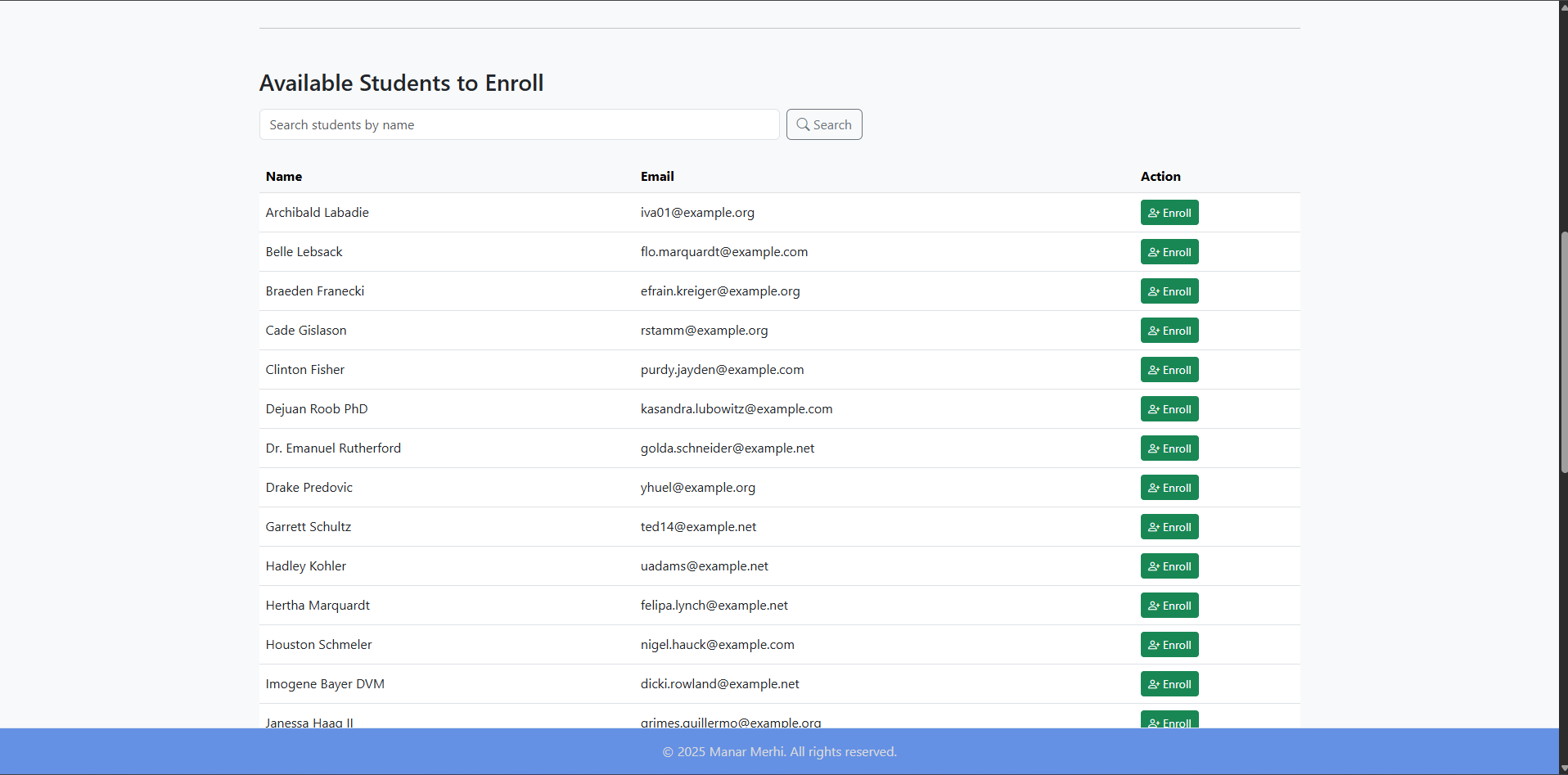


Figure 20. Course viewing 2

Scrolling down we find this, a table showcasing unenrolled students which we can enroll with the click of a button to this course, we can also search for an unenrolled student by the name.

# **Testing and Debugging:**

* Unit and feature testing were implemented using Laravel’s built in testing. I created test files that test for various use cases and “told” Laravel what to expect in every single use-case scenario, then to test at various development stages “php artisan test” was run in the terminal, producing detailed testing results which massively helped me in debugging and ensuring the application was done to perfection with negligible security risks, intuitive, and user-friendly design/functionality.

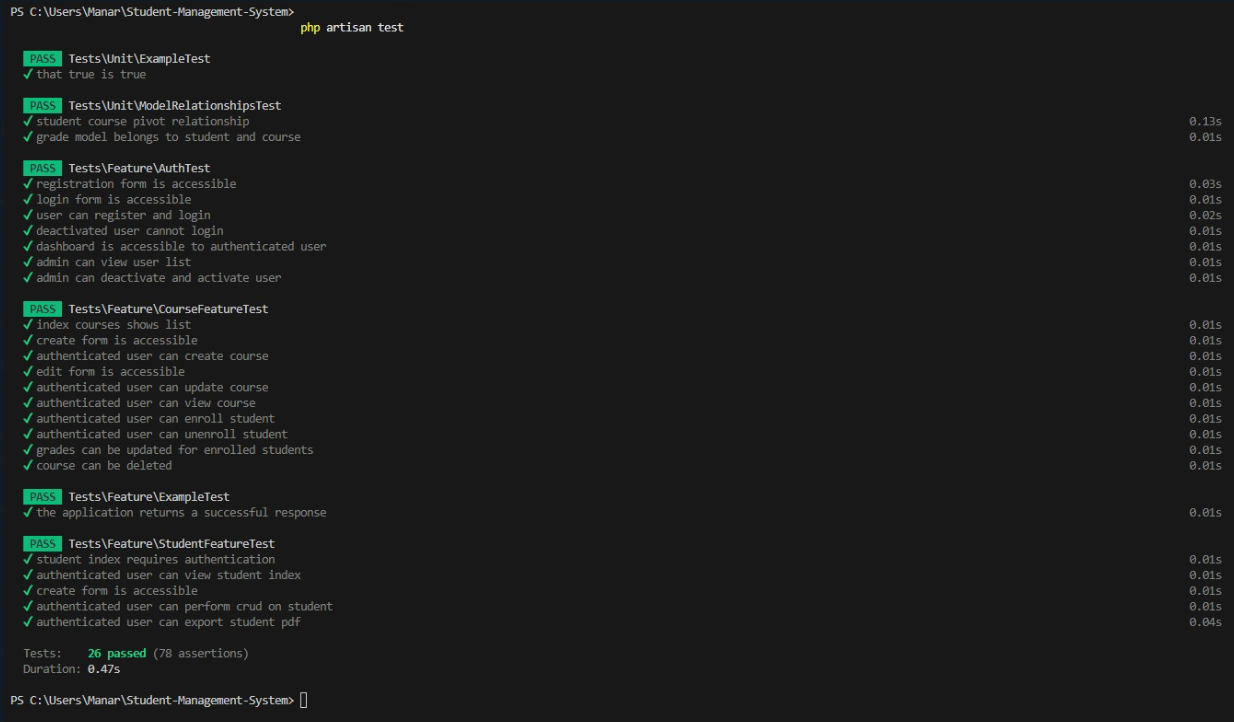


Figure 21. My tests all passing

# **Conclusion**

My Student Management System offers a clean, intuitive interface designed to streamline academic administration. Built with Laravel and modern web technologies, it ensures fast performance, cross-browser compatibility, and easy navigation for staff members managing students, courses, and grades. The platform emphasizes accessibility and usability—allowing users to quickly perform key actions like enrolling students, updating grades, or exporting reports without unnecessary complexity. With role-based access and a focus on user experience, the system is an efficient, reliable solution for educational institutions operating in an intranet setting.