

**Project Report**

Course Management System

Submitted By

**Gautam Sood**

**1911985065**

# **ACKNOWLEDGEMENT**

We take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. We extend our sincere and heartfelt thanks to our esteemed guides, Mr. Mukesh Sir and Mr. Mananjay Sir, for providing us with the right guidance and advice at the crucial junctures and for showing me the right way. I am overwhelmed in all humbleness and gratefulness to acknowledge my depth to all those who have helped me to put these ideas, well above the level of simplicity and into something concrete.

**PURPOSE**

This course management system is an online management software application designed for educational institutions. The primary goal of the project is to facilitate seamless interaction between students and instructors in schools, colleges, and universities concerning the submission of projects, assignments, thesis, and receiving feedback from instructors. LMS is designed to help an individual to develop, manage and provide online courses and programs to learn. It provides a platform for the students and instructors to learn and highlight their skills wherever and whenever they want as per their convenience. Asking Doubts and giving feedbacks becomes very convenient.

**Module Description:**

**Number of Modules**

The system after careful analysis has been identified to be presented with the following modules:

**The modules involved are:**

* ADMINISTRATOR MODULE
* STUDENTS MODULE
* INSTRUCTOR MODULE

**Module Description:**

It has mainly divided into three modules

• ADMINISTRATOR MODULE:

This module is designed exclusively for managing administrative functions like creating accounts for students and instructors, creating the curriculum, coding the subjects, managing the employees, payroll, and so on. Basically, this module lays the groundwork for the other two modules.

• STUDENTS MODULE:

This module is designed for the usage of students. They can log in to their accounts to view their coursework, submit their projects, get feedback from instructors, etc.

• INSTRUCTOR MODULE:

This module is for the instructors who can log in to their accounts and check the projects submitted by the students, communicate with the students, and offer guidance to them.

This project aims to promote the sharing of information between qualified instructors and students via the Internet.

# **Technologies Used:**

Various web development technologies are used to create Student Portal. These are:

**Front-end technologies:** HTML, CSS, JS, React JS, Axios, Formic, React Router, Tailwind, React-Query, JS-cookie.

* HTML: (Hypertext Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. HTML is the standard markup language for Web pages. With HTML we can create our own website.
* CSS: Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML. CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.
* JS: It is a lightweight, interpreted programming language. It is designed for creating network-centric applications. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform. JavaScript can be used for client-side developments as well as server-side developments. It is used to make webpages interactive (e.g., having complex animations, clickable buttons, popup menus, etc.
* React JS: It is a JavaScript library that combines the speed of JavaScript and uses a new way of rendering web pages, making them highly dynamic and responsive to user input. aces. React is not a framework – it's not even exclusive to the web. It's used with other libraries to render to certain environments. Reacts primary goal is to minimize the bugs that occur when developers are building UIs. It does this through the use of components — self-contained, logical pieces of code that describe a portion of the user interface. These components can be composed

together to create a full UI, and React abstracts away much of the rendering work, leaving you to concentrate on the UI design.

* Axios: It is a JavaScript used to make HTTP requests to the server to perform certain actions. These requests are promise based and use the concept of asynchronous JavaScript.
* Formic: It is a JS library used to perform JS object validation.
* React Router: It is a standard library for routing in React. It enables the navigation among views of various components in a React Application, allows changing the browser URL, and keeps the UI in sync with the URL.
* Tailwind: Tailwind CSS is basically a utility-first CSS framework for rapidly building custom user interfaces. It is a highly customizable, low-level CSS framework that gives you all of the building blocks you need to build bespoke designs without any annoying opinionated styles you have to fight to override.
* React-Query: React Query is often described as the missing data-fetching library for React, but in more technical terms, it makes fetching, caching, synchronizing and updating server state in your React applications a breeze.
* JS-cookie: A simple, lightweight JavaScript API for handling cookie
* Works in all browsers
* Accepts any character
* Heavily tested
* No dependency
* Supports ES modules
* Supports AMD/CommonJS
* RFC 6265 compliant
* Useful Wiki
* Enable custom encoding/decoding

**Backend technologies:** Express JS, JWT, NodeJS, MongoDB, Cookie-Parser, BCrypt JS, CORS, Multer **.**

**•** Express JS:Express.js, or simply Express, is a back-end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js. Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

• JWT: To authenticate a user, a client application must send a JSON Web Token (JWT) in the authorization header of the HTTP request to your backend API. API Gateway validates the token on behalf of your API, so you don't have to add any code in your API to process the authentication. JSON Web Token (JWT) is a standard RFC 7519 for exchanging cryptographically signed JSON data. It is probably the most popular current standard of authorization on the web, especially when it comes to microservices and distributed architecture.

• NodeJS: Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project. Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant. A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception rather than the norm.

• MongoDB: It is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server-Side Public License (SSPL). MongoDB, Inc. is an American software company that develops and provides commercial support for the source-available database MongoDB, a NoSQL database that stores data in JSON-like documents with flexible schemas. MongoDB Atlas makes it easy to control access to your database. Your database instances are deployed in a unique Virtual Private Cloud (VPC) to ensure network isolation. Other security features include IP whitelisting or VPC Peering, always-on authentication, encryption at rest and encryption in transit, sophisticated role-based access management, and more.

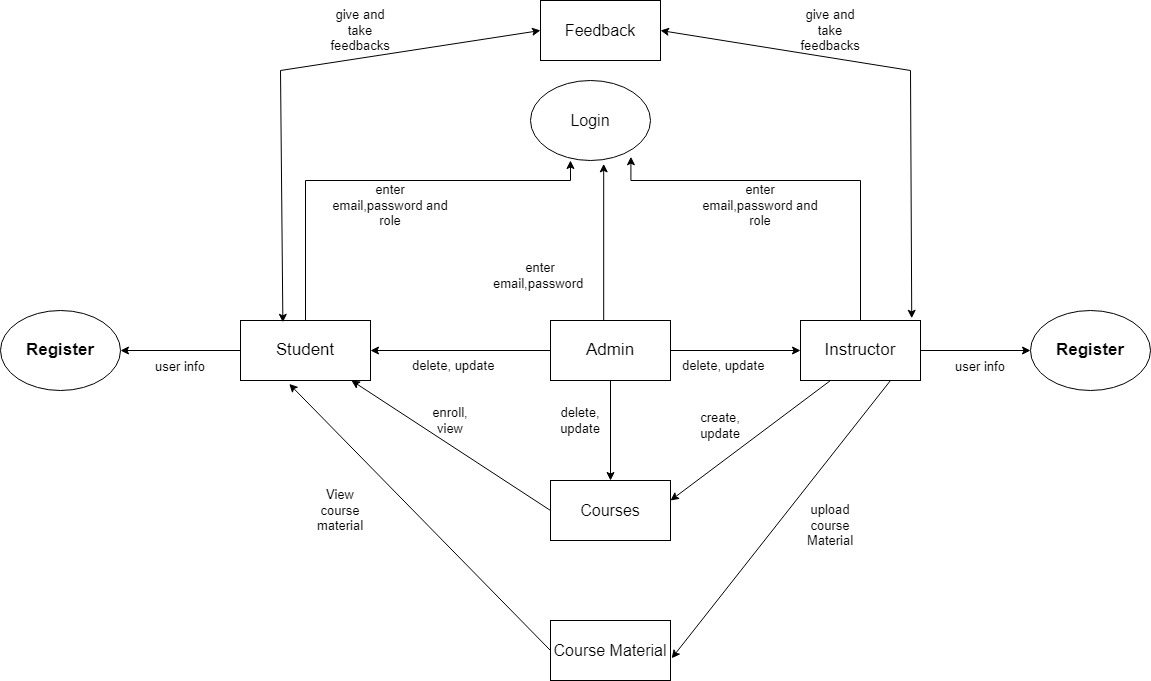
•Cookie-Parser: Parse Cookie header and populate req.cookies with an object keyed by the cookie names. Optionally you may enable signed cookie support by passing a secret string, which assigns req.secret so it may be used by other middleware.

• BCrypt JS: bcrypt is a password-hashing function designed by Niels Provos and David Mazières, based on the Blowfish cipher and presented at USENIX in 1999. Besides incorporating a salt to protect against rainbow table attacks, bcrypt is an adaptive function: over time, the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computation power. Blowfish is notable among block ciphers for its expensive key setup phase. It starts with subkeys in a standard state, then uses this state to perform block encryption using part of the key, and uses the result of that encryption (which is more accurate at hashing) to replace some of the subkeys.

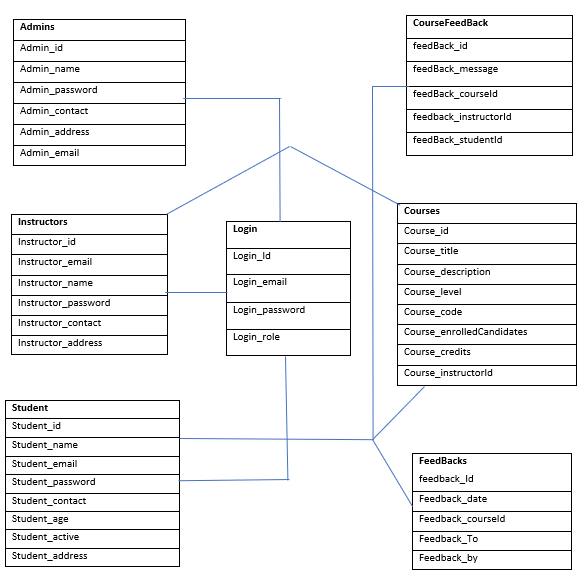
• CORS: Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served. A web page may freely embed cross-origin images, stylesheets, scripts, iframes, and videos. Certain "cross-domain" requests, notably Ajax requests, are forbidden by default by the same-origin security policy. CORS defines a way in which a browser and server can interact to determine whether it is safe to allow the cross-origin request. It allows for more freedom and functionality than purely same-origin requests but is more secure than simply allowing all cross-origin requests.

* Multer: Multer is a node js middleware for handling multipart/form-data , which is primarily used for uploading files. It is written on top of busboy for maximum efficiency. NOTE: Multer will not process any form which is not multipart ( multipart/form-data ).
* Cloudinary: Cloudinary's Node.js SDK simplifies the generation of transformation URLs for easy embedding of assets in your Node.js application. After you or your users have uploaded image assets to Cloudinary, you can deliver them via dynamic URLs. You can include instructions in your dynamic URLs that tell Cloudinary to transform your assets using a set of transformation parameters. All transformations are performed automatically in the cloud and your transformed assets are automatically optimized before they are routed through a fast CDN to the end user for optimal user experience.

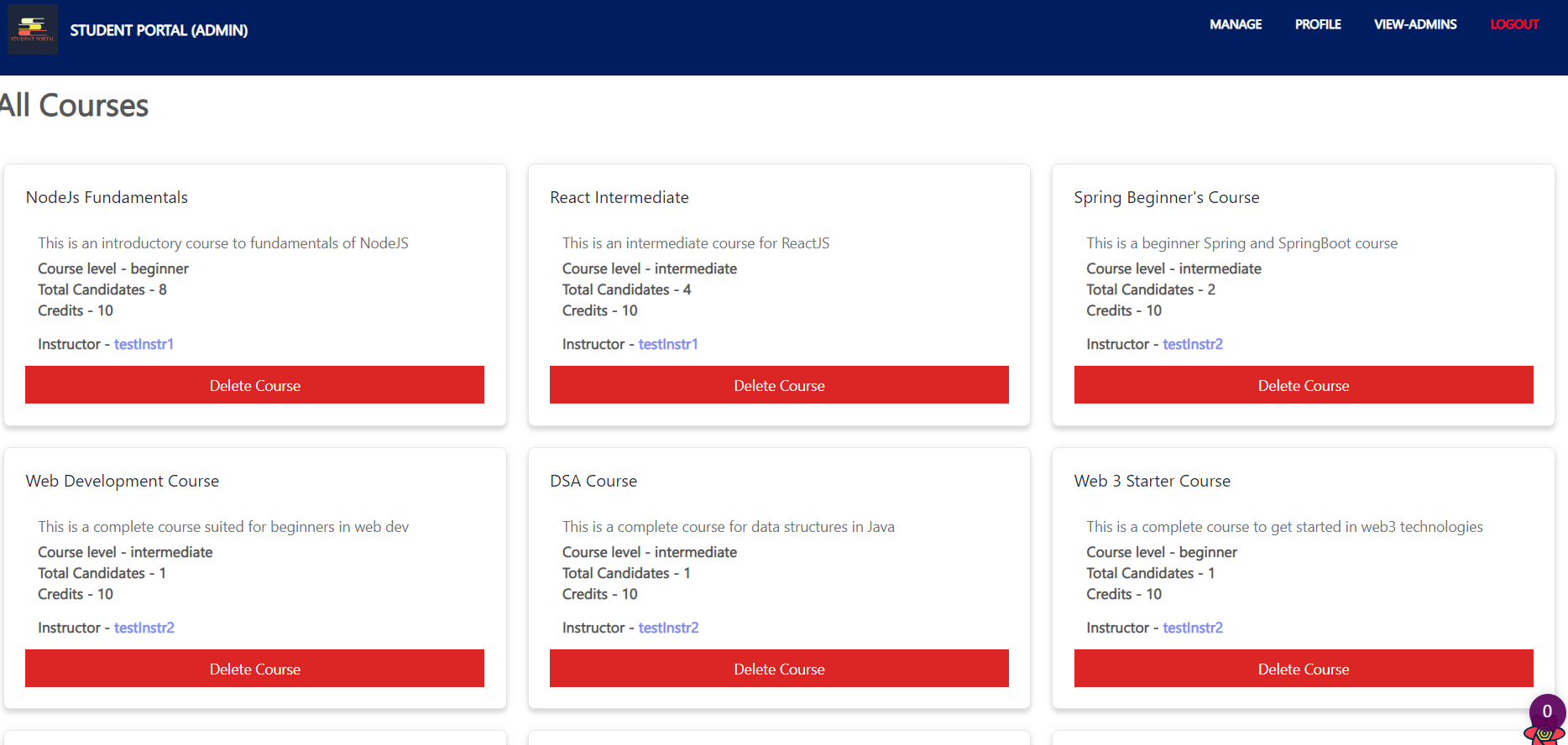
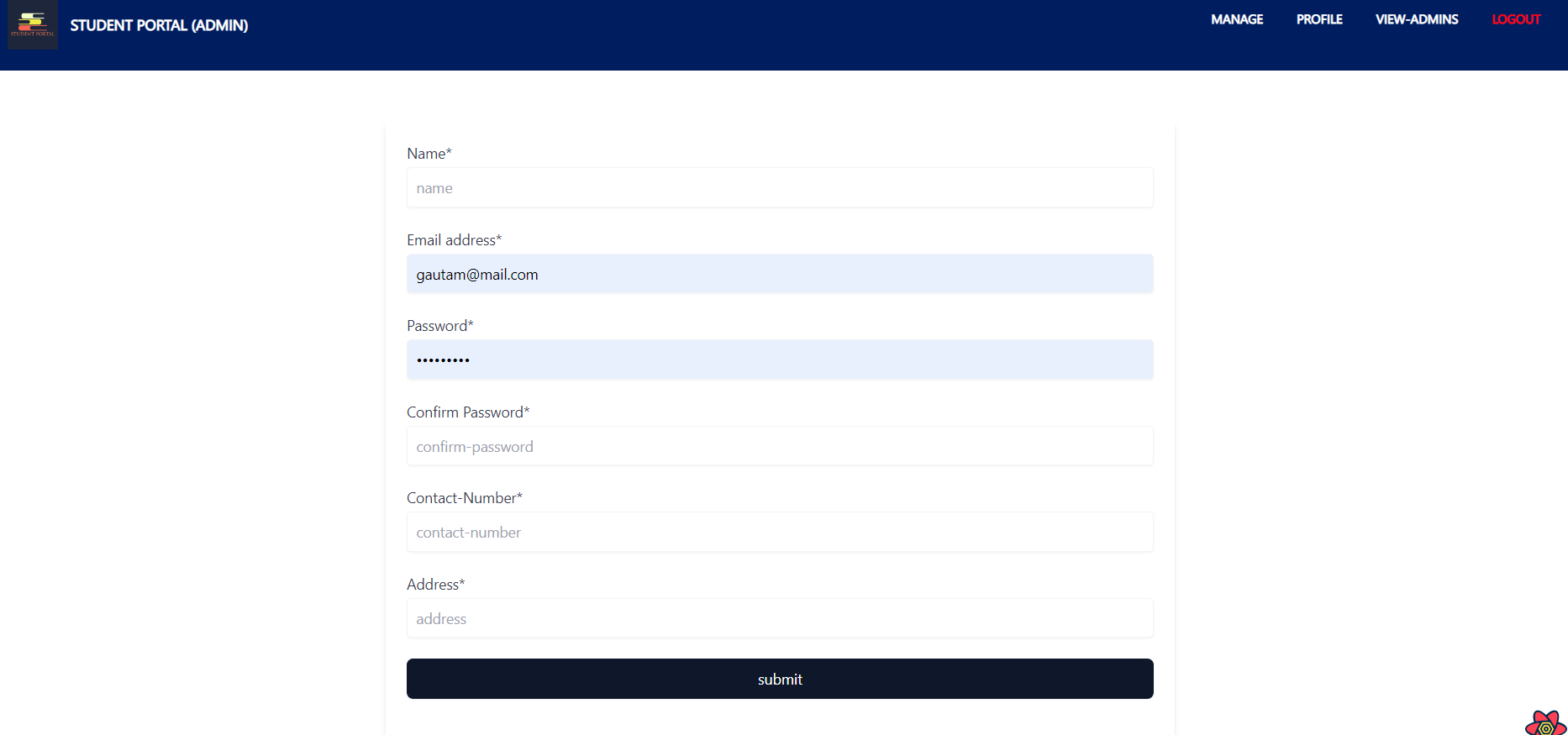
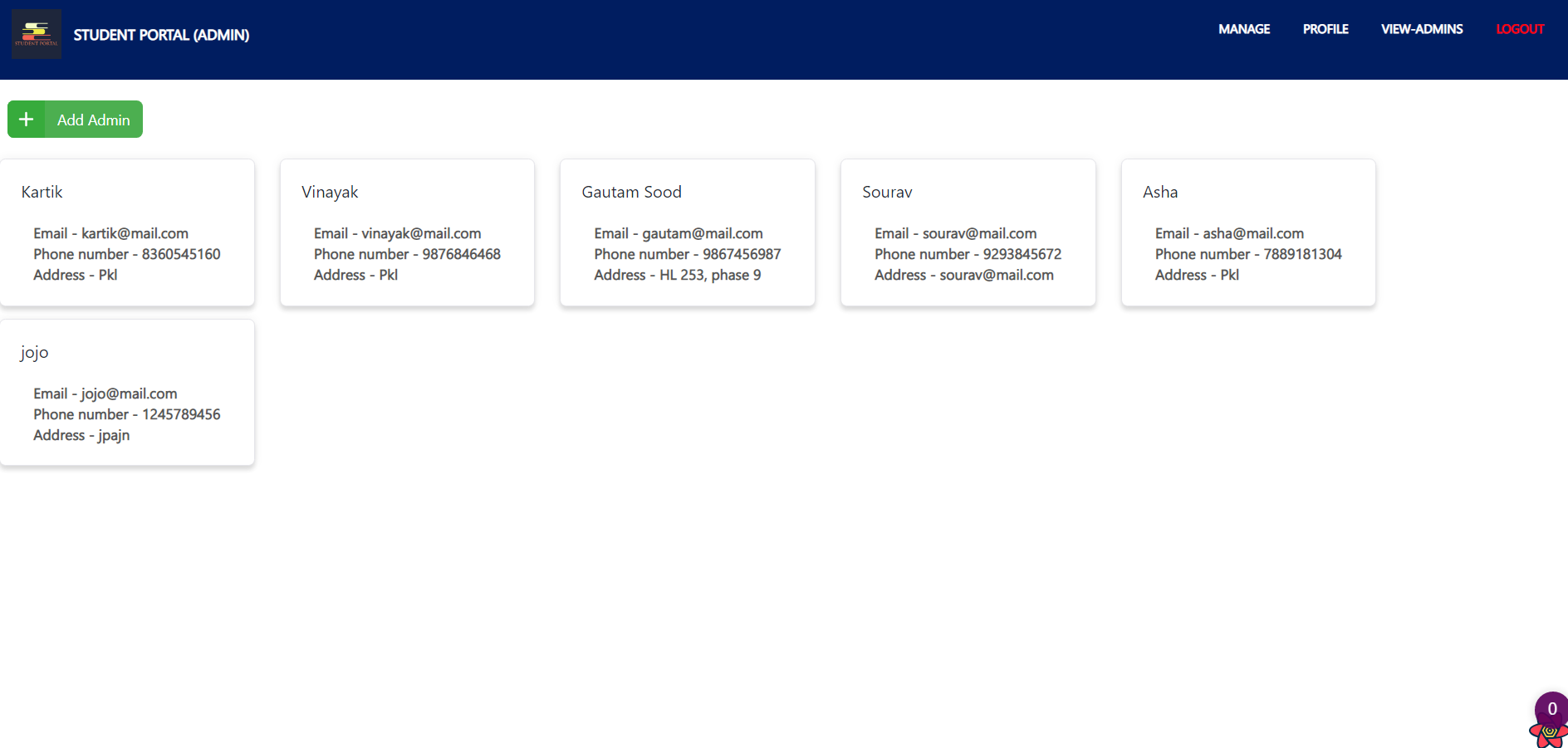
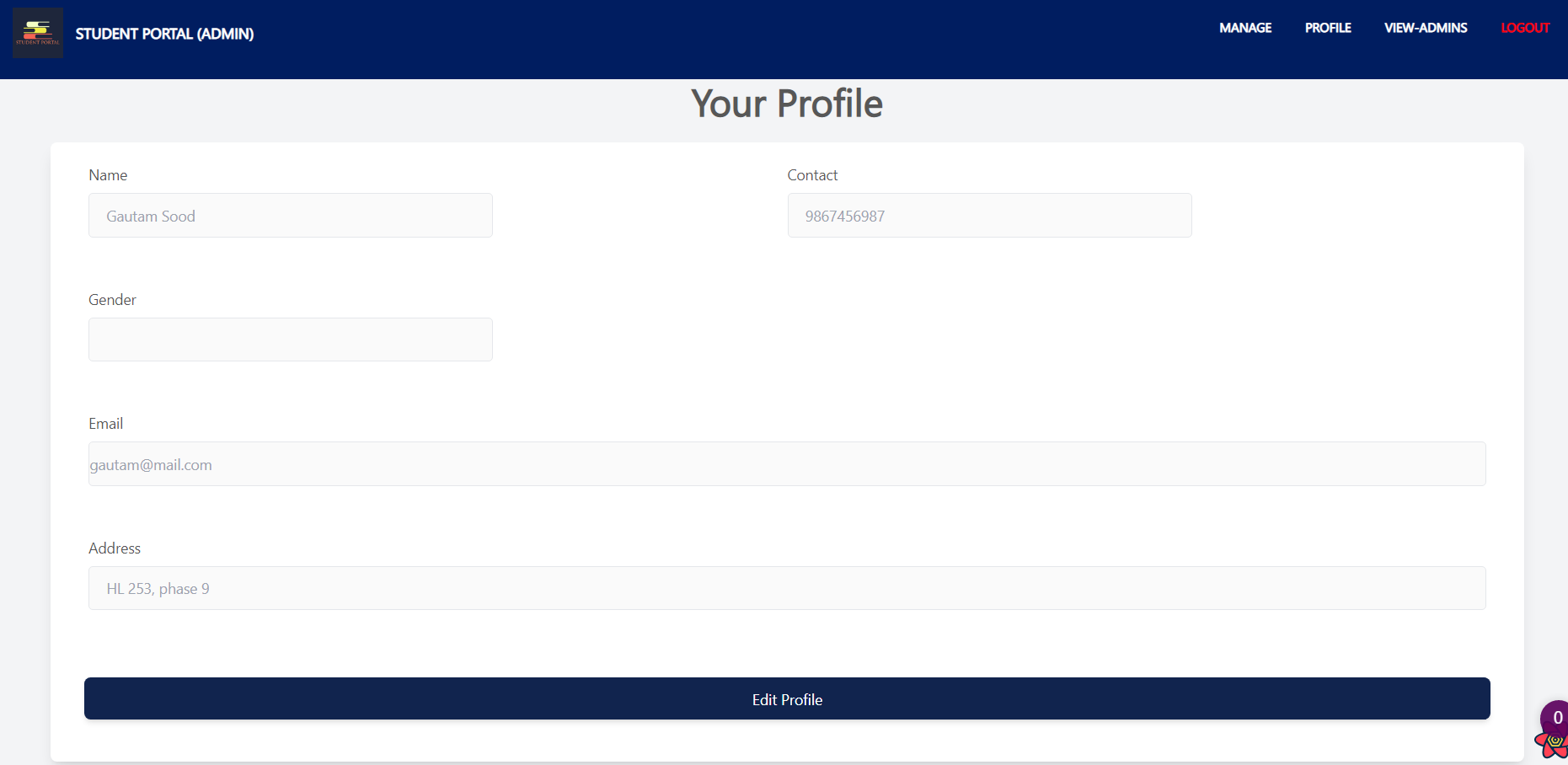
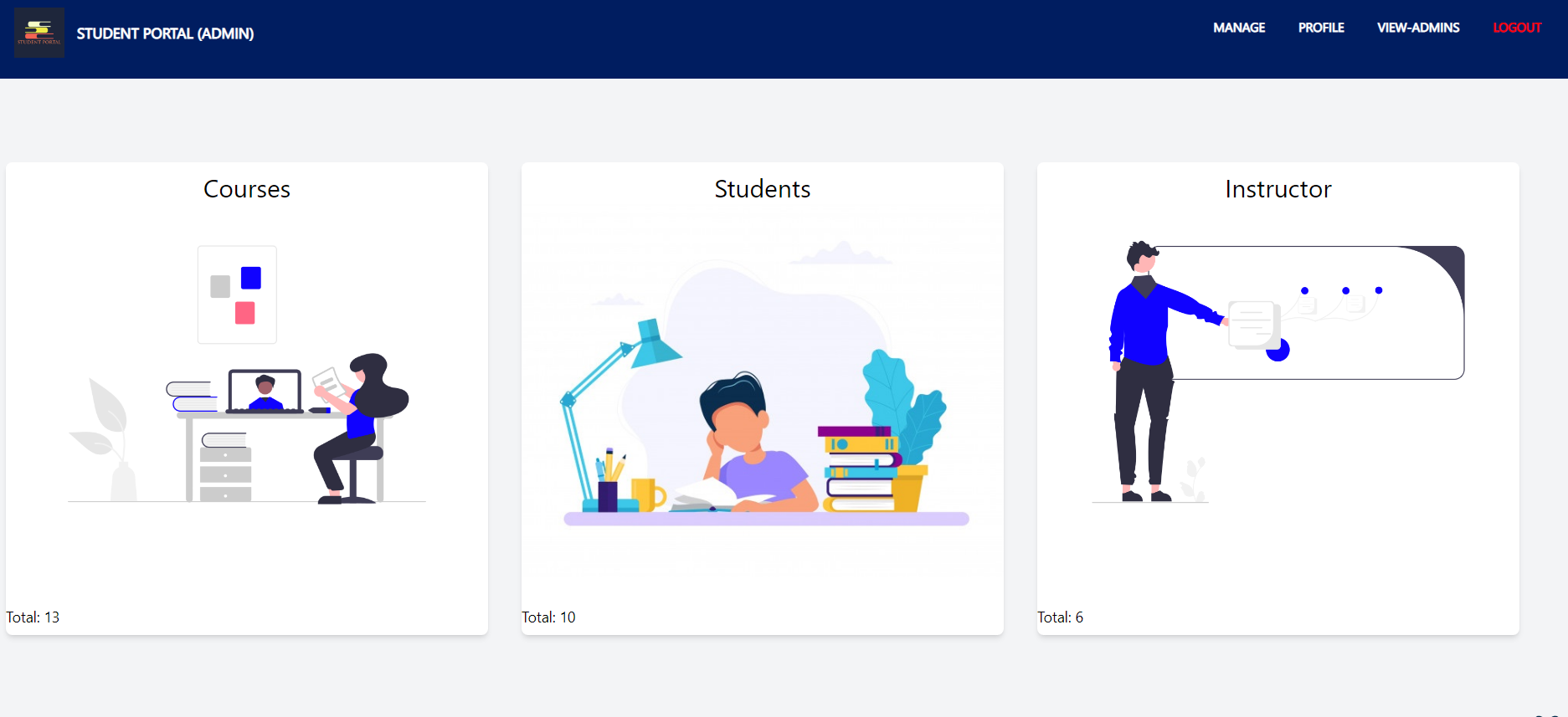
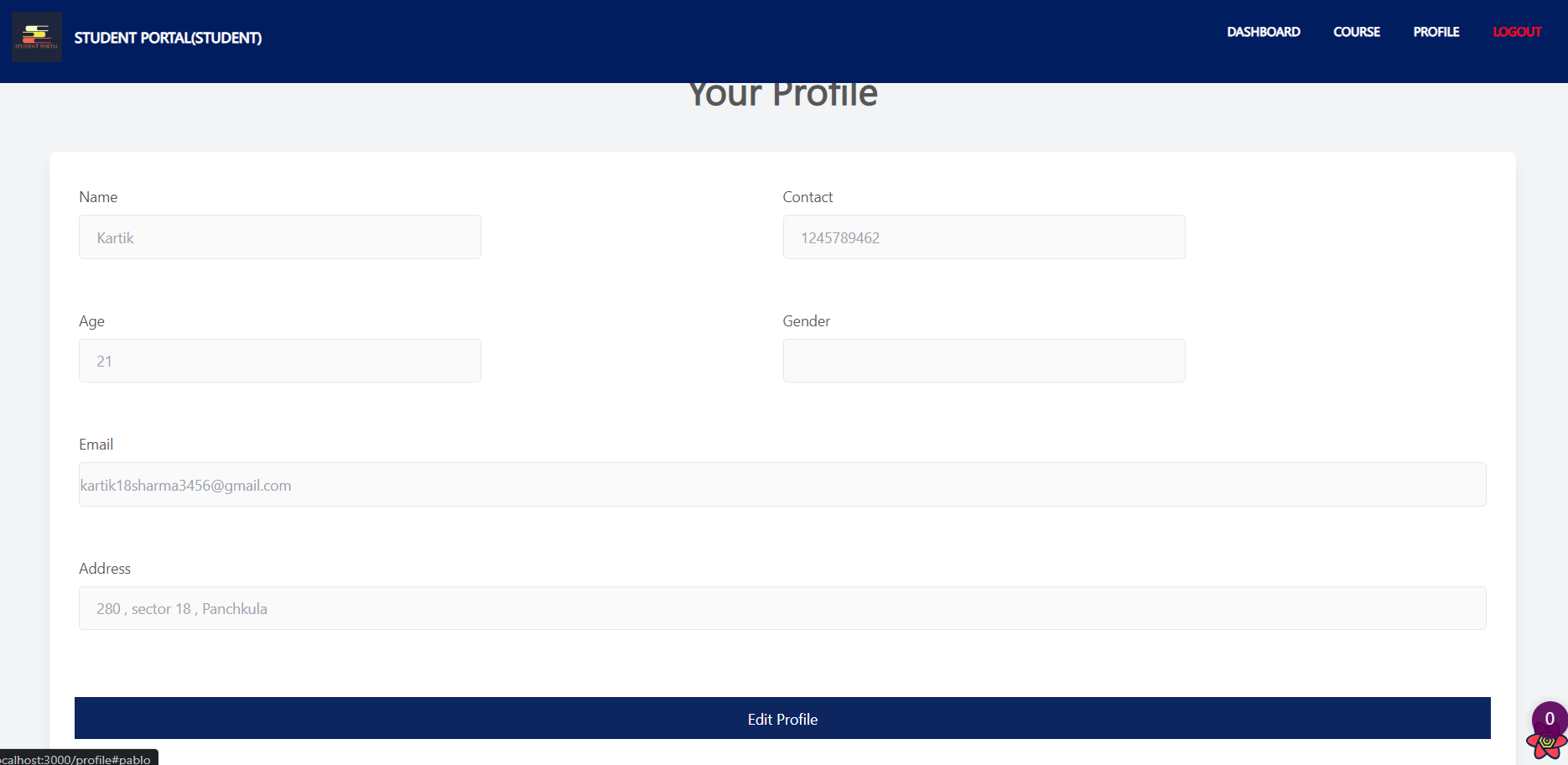
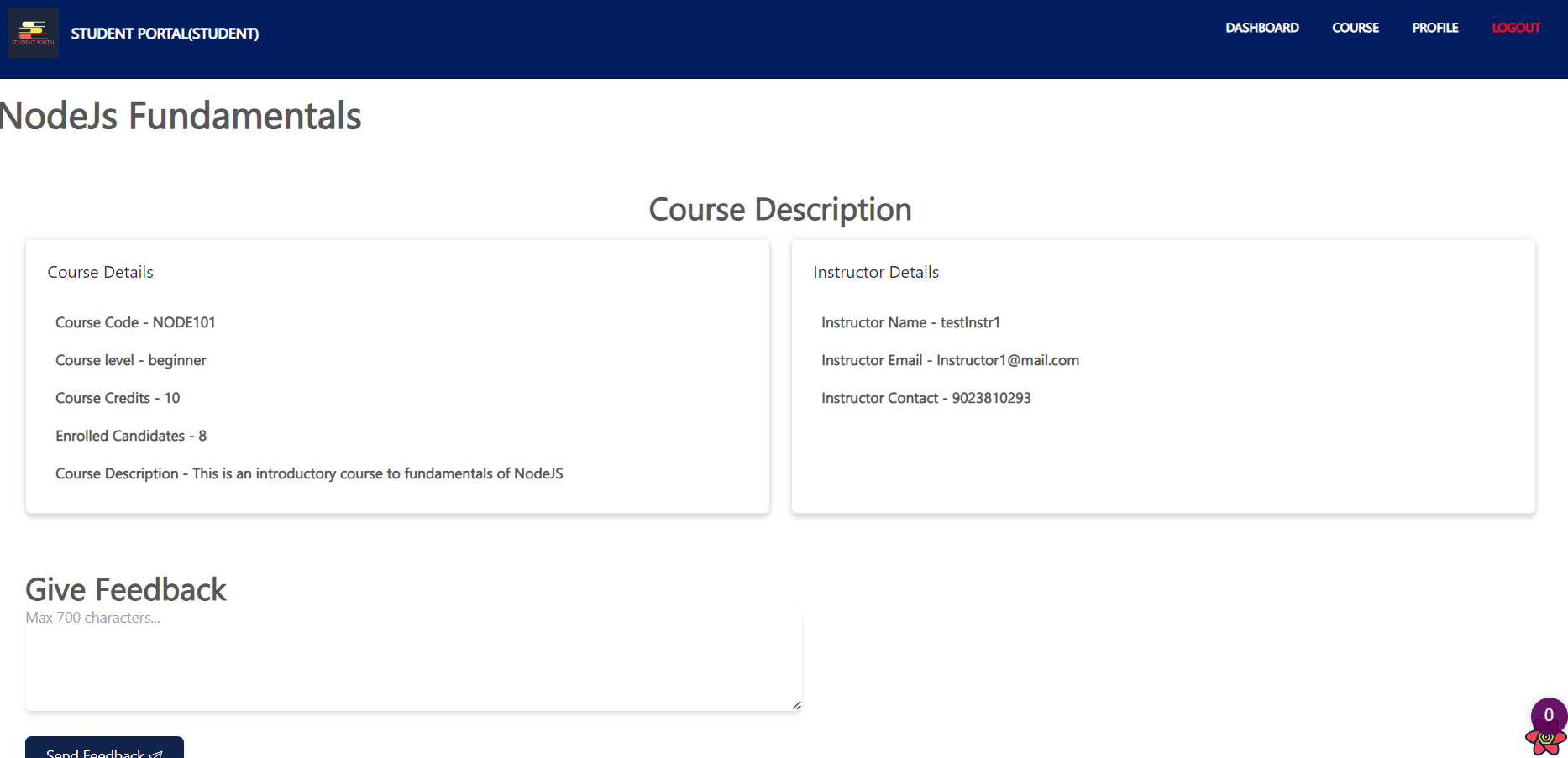
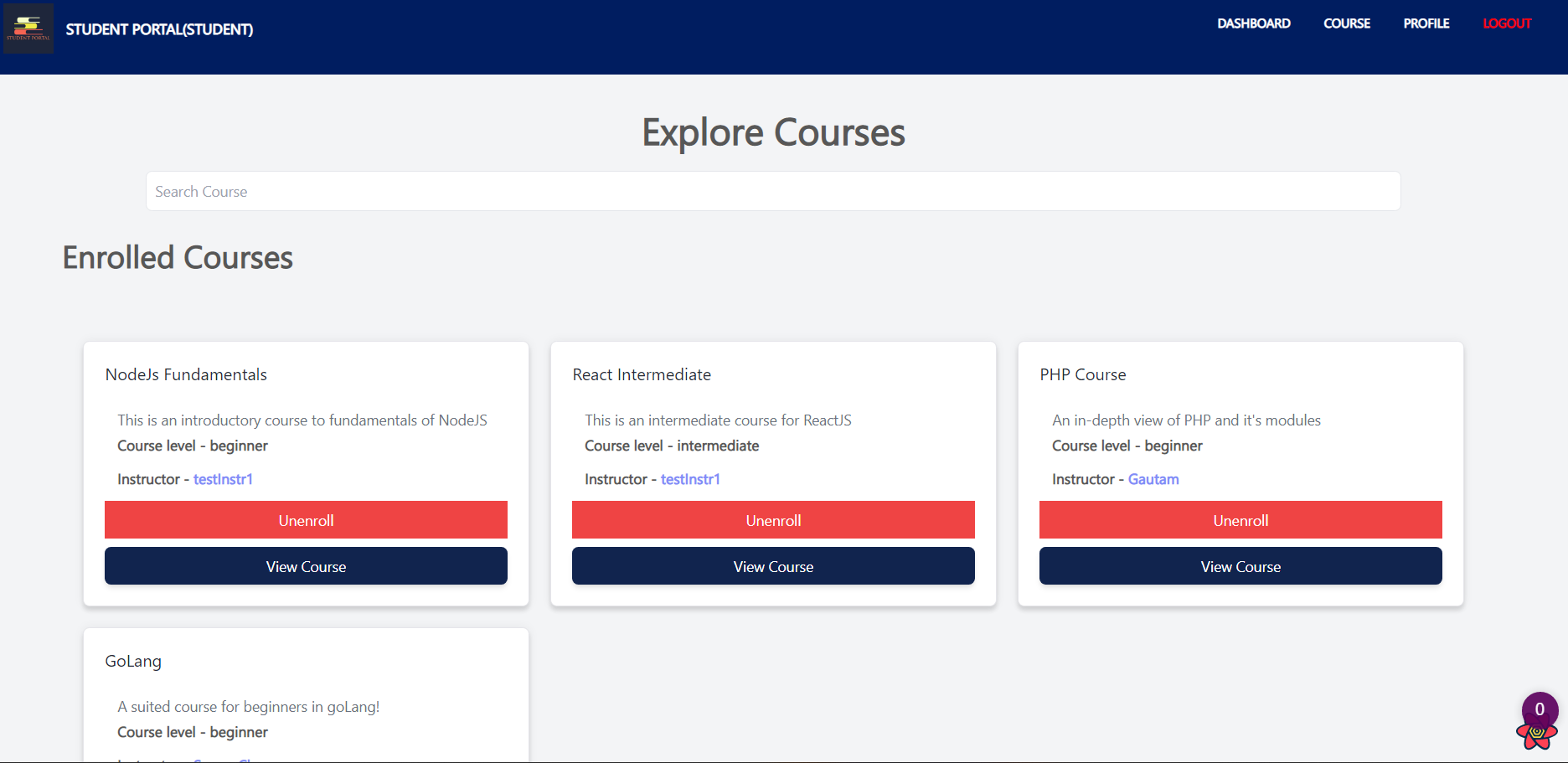
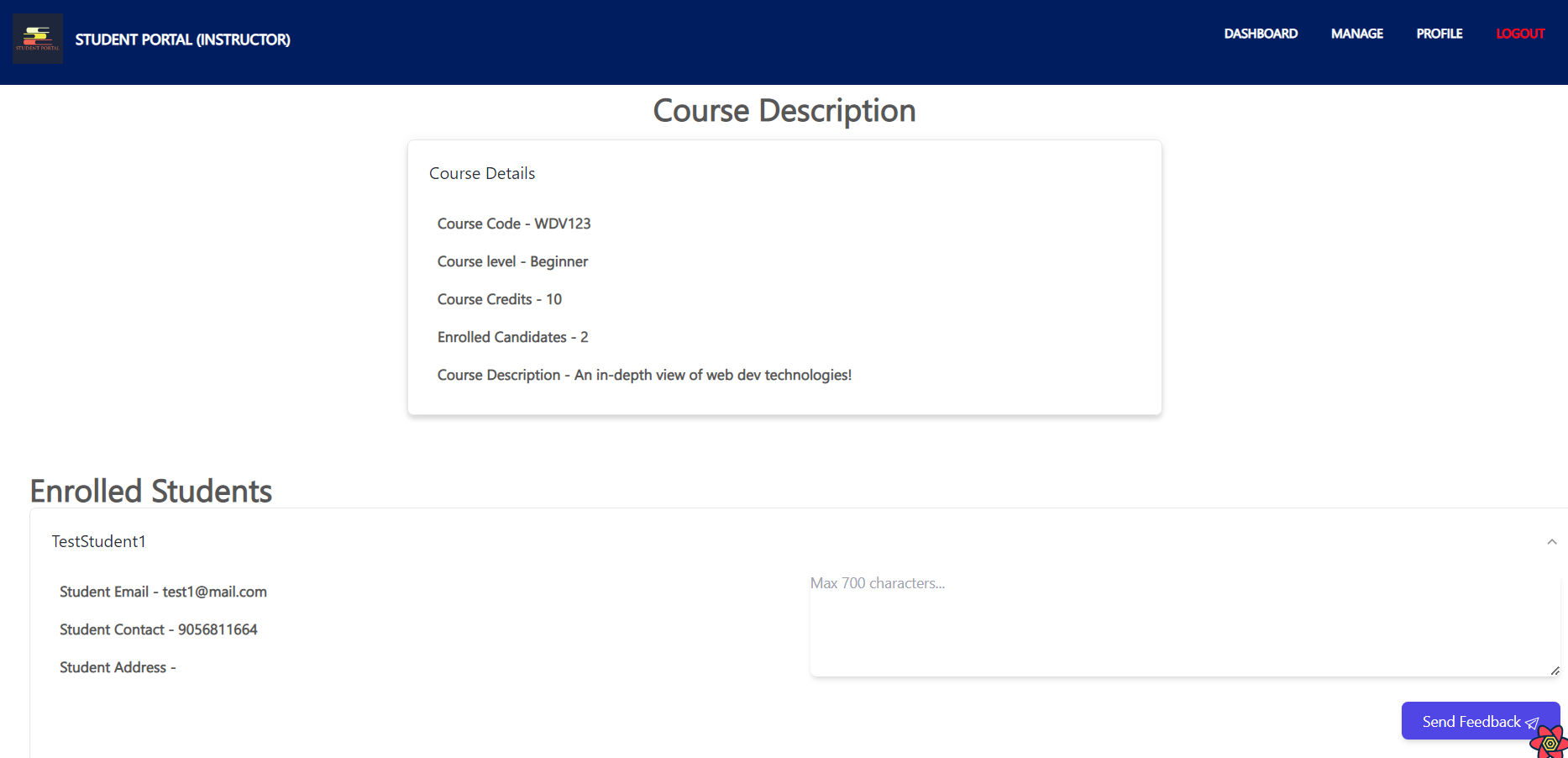
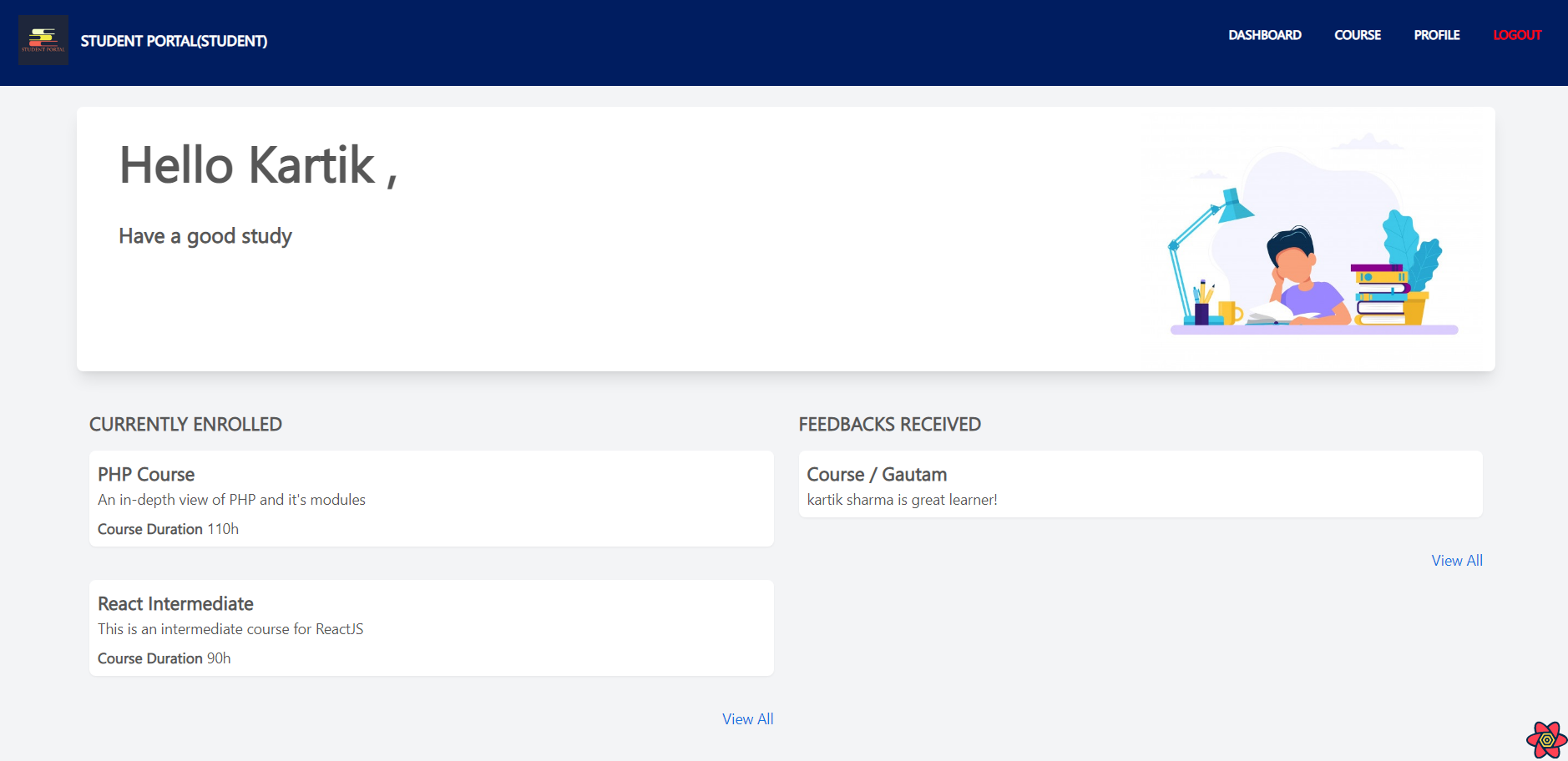
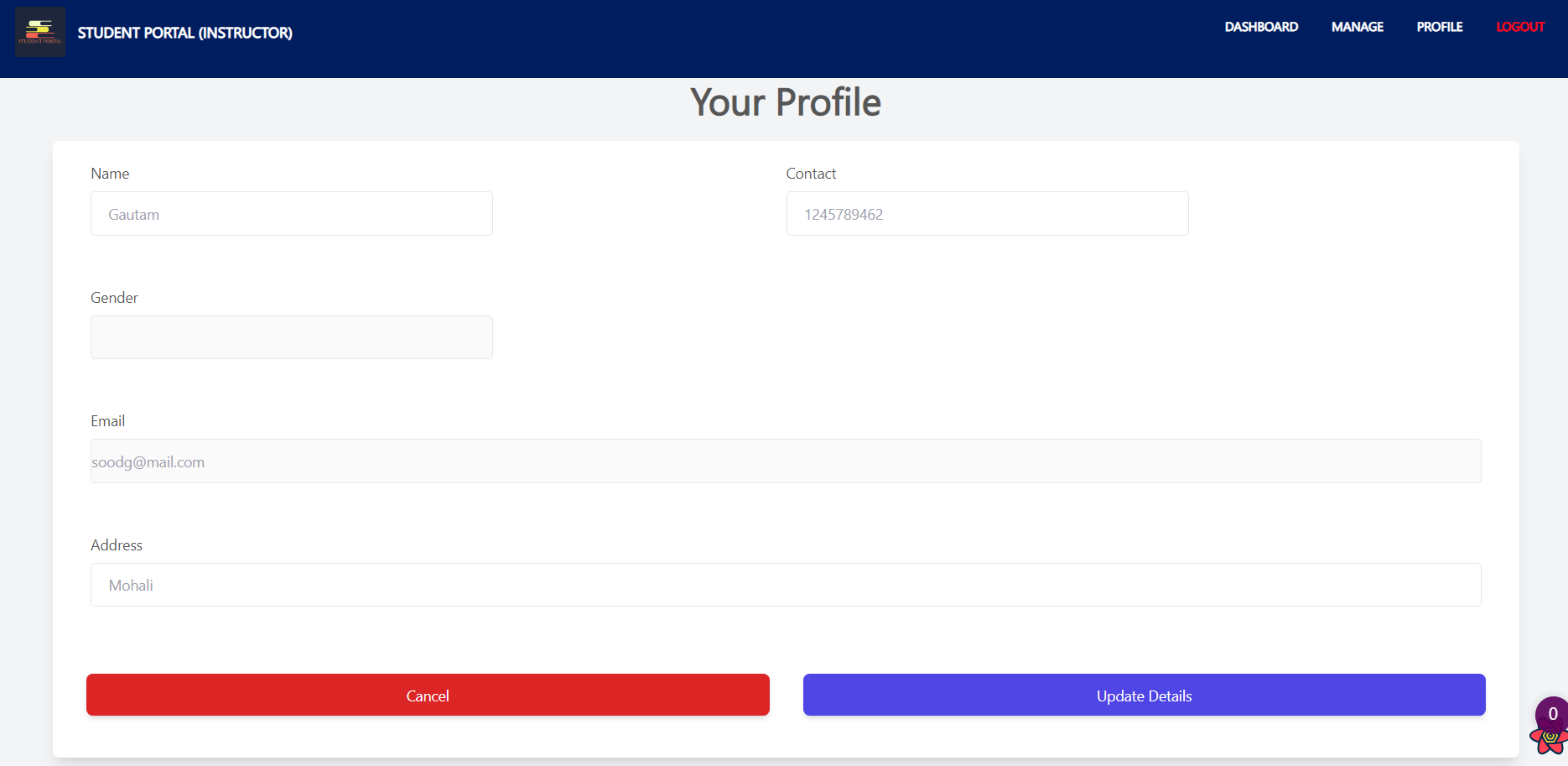
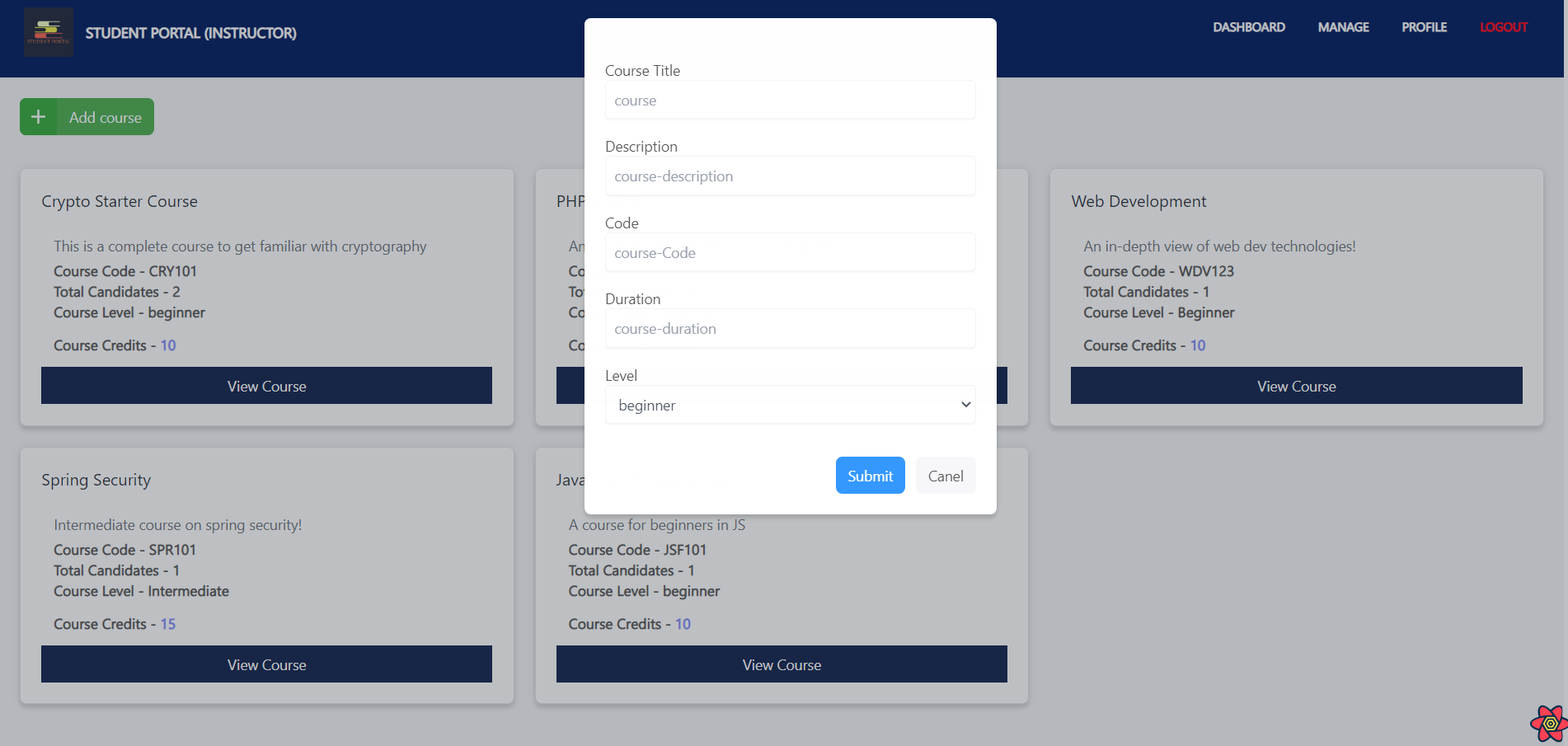
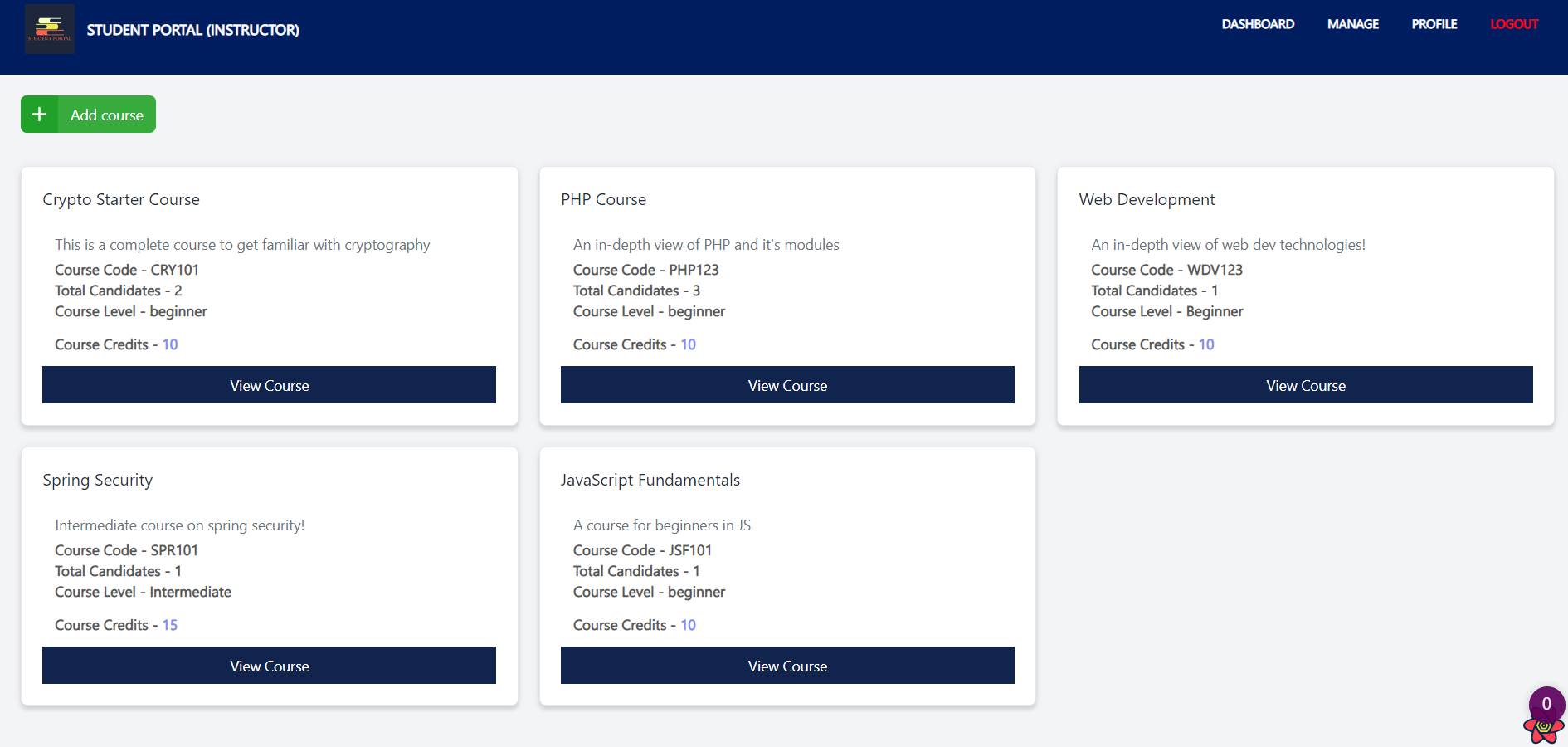
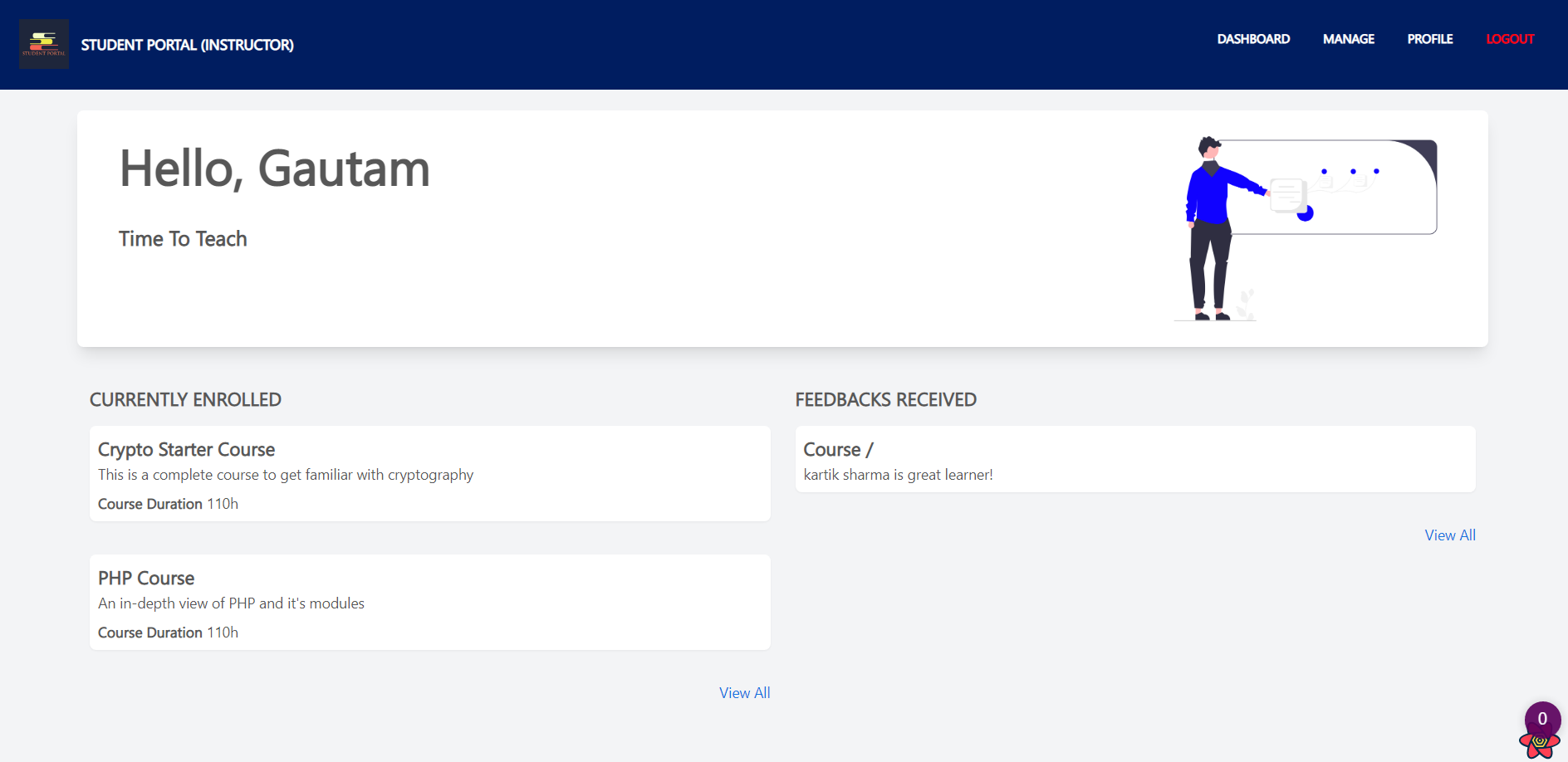
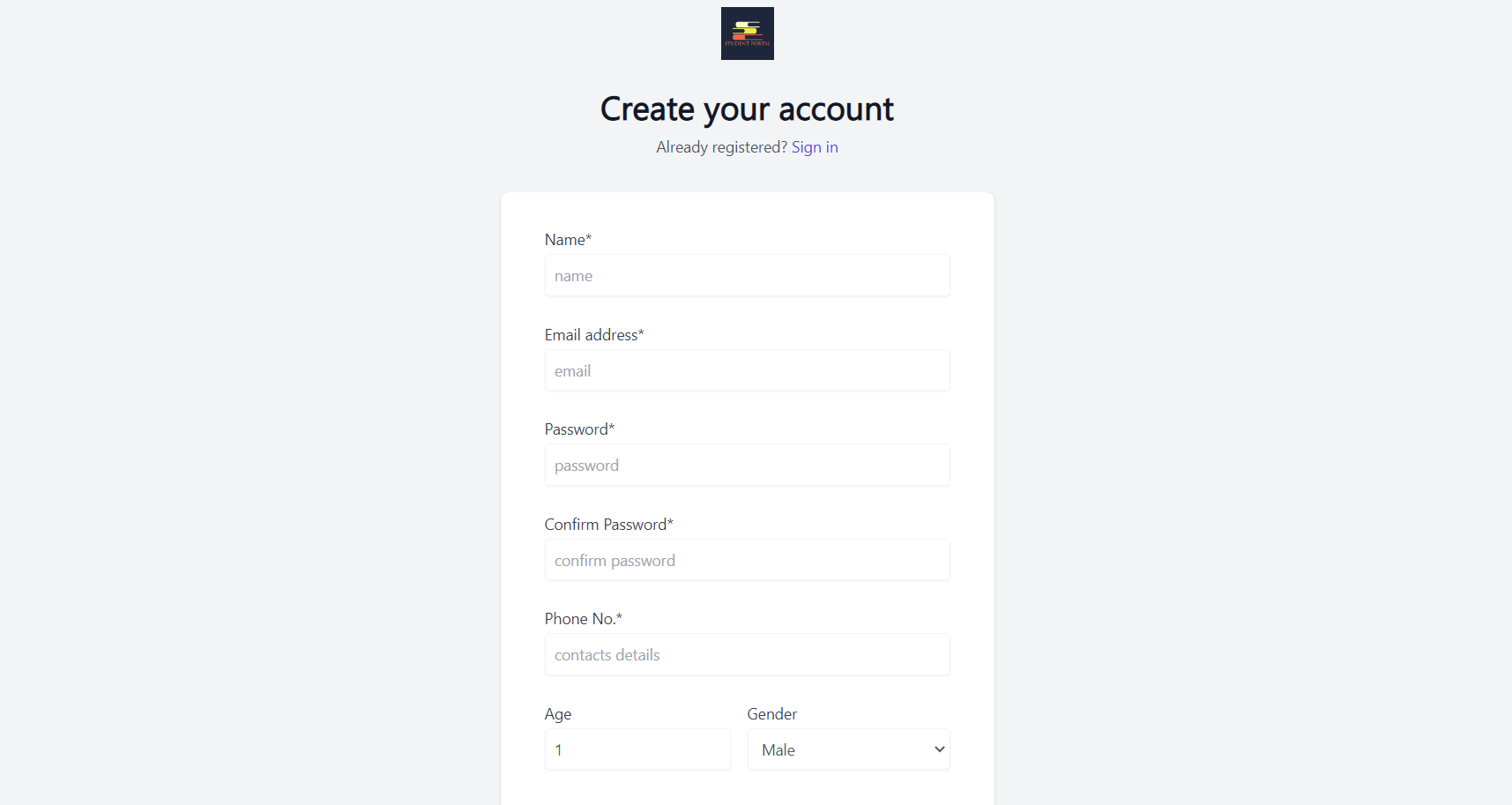
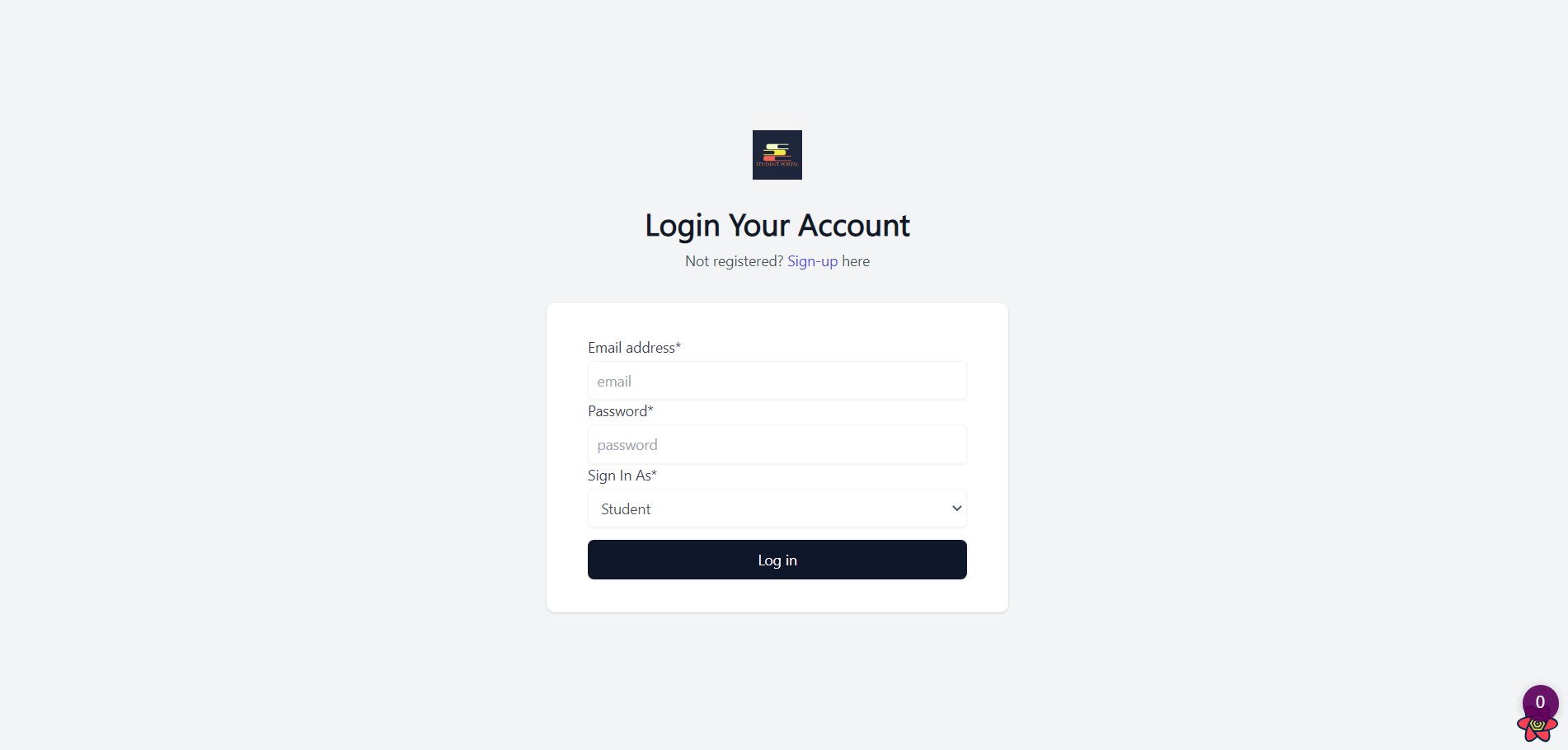
# **Data Flow Diagram:**



**ER diagram:**



**Screen shots**



**Future Scope:**

1. Student Assignments:

It provides opportunity for students to learn, practice and demonstrate they have achieved the learning goals. It provides the evidence for the teacher that the students have achieved the goals. Instructor can give students assignments and student can add their assignments

2. Minor Bug Fix is needed

3. One to One chat application.

4. Forget Password