**Project Report**

On

**Student Management System**

**Acknowledgement**

I am grateful to all my faculty who have contributed in inspiring and clarifying my thought over the years; (Sonal mam) and many other faculty of SUNBEAM CDAC. Special gratitude I give to my respected head of the division Mr.Nitin sir, for allowing me to use the facilities available and also help me to coordinate my project. I am thankful also for the tireless generosity to my sponsors, well-wishers and my family for spiritual, moral and even financial support. Thanks also to the entire staff of my College, Furthermore, I would also like to acknowledge with much appreciation the crucial role of faculty members on this occasion. Last but not least, I would like to thank friends who help me to assemble the parts and gave a suggestion about the project. Finally, Honor and glory to the highest God for having enabled me successfully accomplishes.

.

**----------------------**

### Abstract

The main objective of this project is to build a student database system that will store records of students. It is purposed to reduce time spent on administrative tasks. The system is intended to accept process, generate students. The system is also intended to provide better services to users, provide meaningful, consistent, and timely data and information and finally promotes efficiency by converting paper processes to electronic form. The system was developed using basic technologies such as MySQL database and React JS. The system is free of errors and very efficient and less time consuming due to the care taken to develop it. All the phases of software development cycle are employed and it is worthwhile to state that the system is user friendly and strong. Provision is made for future development in the system.

**Introduction**

Student Management System is software which is helpful for students as well as the school authorities. In the current system all the activities are done manually. It is very time consuming and costly. Our Student Management System deals with the various activities related to the students.

The two main users involved in this system are

1. User(i.e., Students)
2. Admin

In the Software we can register a student by the authentication code i.e., student id and password with help of student id and password student can login his/her account and view all the notices which is announced by admin.

**Purpose**

The objective of **Student Management System** is to allow the administrator of any organization to edit and find out the personal details of a student and allows the student to keep up to date his profile .It’ll also facilitate keeping all the records of students, such as their id, name, mailing address, phone number, DOB etc. So, all the information about a student will be available in a few seconds.

Overall, it’ll make Student Information Management an easier job for the administrator and the student of any organization.

**Advantages:**

* It helps the educational administrator to handle and manage students’ records.
* It helps educational administrator to generate report.
* It brings transparency and efficiency in the working of educational system.

**Disadvantages:**

* The system can only handle single educational organization.
* The system does not include bank payment, dd, cheque status.

**Applications:**

The website **Student Management System** is aimed towards recording a considerable number of student records and needs online assistance for managing records of students. Website should be user-friendly, ‘quick to learn’ and reliable website for the above purpose. **Student Management System** is intended to be a stand-alone product andshould not depend on the availability of other website. The system will also have an administrator who has full-fledged rights with regards to performing all actions related to control and management of the website.

**Feasibility study**

Whenever we design a new system, normally the management will ask for a feasibility report of the new system. The management wants to know the technicalities and cost involved in creation of new system.

- Technical feasibility

- Economic feasibility

- Physical feasibility

**Technical feasibility:**

Technical feasibility involves study to establish the technical capability of the system being created to accomplish all requirements to the user. The system should be capable of handling the proposed volume of data and provide users and operating environment to increase their efficiency.

For example, system should be capable of handling the proposed volume of data and provide users.

**Economic feasibility:**

Economic feasibility involves study to establish the cost benefit analysis. Money spent on the system must be recorded in the form of benefit from the system. The benefits are of two types:

**Tangible benefits:**

* + Saving man labor to do tedious tasks saves time.

**Intangible benefits:**

* + Improves the quality of organization.

**Physical feasibility:**

It involves study to establish the time responses of the new system being created. For e.g., if the new system takes more than one day to prepare crucial finance statement for the management, wherever it was required in an hour, the system fails to provide the same.

It should be clearly established that the new system requirements in the form of time responses would be completely met with. It may call for increase in cost. If the required cost is sacrificed then the purpose of the new system may not be achieved even if it was found to be technically feasible.

**Scope of the Project**

The proposed system will affect or interface with the user (student) and administrator. The system works and fulfills all the functionalities as per the proposed system. It will provide reduced response time against the queries made by different users. This project is based on React & Node JS with MYSQL database which manage the details of the student because it is a tedious job for any organization. Student Information system will store all the details of the students including their background information.

All possible features such as verification, validation, security, user friendliness etc. have been considered.

The different types of modules present in this project are

1. Admin
2. User

**Admin:**

1. **Dashboard**: In this section, admin can see all detail in brief like Total Classes, Total Students, Total Class Notices and Total Public Notices.
2. **Class**: In this section, admin can manage class (Add/Update/Delete).
3. **Students**: In this section, admin can manage the students (Add/Update/Delete).
4. **Notices:** In this section, the admin can manage notices (Add/Update/Delete).
5. **Public Notices:** In this section, the admin can manage public notices.
6. **Pages:** In this section admin, can manage about us and contact us page of administration
7. **Search:** In this section admin, can search students by their student id.
8. **Reports:** In this section admin, can view how much students has been register in particular period.
9. Admin can also update his profile, change the password and recover the password.

**User (Students):**

1. **Dashboard**: It is welcome page for students.
2. **View Notices**: In this section, user can view notices which are announced by administrator.
3. Student can also view his profile, change the password and recover the password.

**User (Non-Register):**

1. **Home**: It is welcome page for user.
2. **About**: User can view about us page.
3. **Contact:** User can view contact us page.

**Software & Hardware requirements**

**Software Requirements:**

* NodeJS
* VSCode Editor
* Xampp(For MySQL)
* Operating System: Microsoft Windows/Linux

**Hardware Requirements:**

* Processor: Pentium-III (or) Higher
* Ram: 64MB (or) Higher
* Hard disk: 80GB (or) Higher

Here are the definitions of the technologies used in this project:

1. **React**: React is a JavaScript library used for building user interfaces. It allows developers to create reusable UI components and efficiently update and render them in response to data changes.
2. **Node.js**: Node.js is a server-side JavaScript runtime environment. It allows developers to build scalable and high-performance web applications by executing JavaScript code on the server.
3. **Express**: Express is a web application framework for Node.js. It provides a set of robust features and middleware that simplifies the process of building web applications and handling HTTP requests.
4. **MySQL**: MySQL is an open-source relational database management system. It is widely used for storing and managing structured data. In this project, MySQL is utilized for storing and retrieving data related to doctors, patients, appointments, and other system information.

**System Design**

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data

**Unified Modelling Language Diagrams (UML):**

* + The unified modelling language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.
  + A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

**User Model View**

* + 1. This view represents the system from the user’s perspective.
    2. The analysis representation describes a usage scenario from the end-user’s perspective**.**

**Structural model view**

In this model the data and functionality are arrived from inside the system.

 This model view models the static structures.

**Behavioural Model View**

 It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

**Implementation Model View**

* + In this the structural and behavioural as parts of the system are represented as they are to be built.

**Environmental Model View**

In these the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are

* + UML Analysis modelling, which focuses on the user model and structural model views of the system?
  + UML design modelling, which focuses on the behavioural modelling, implementation modelling and environmental model views**.**

### Use Case Diagrams User/Student

### Use Case Diagrams Admin

#### ENTITY-RELATIONSHIP Diagrams

E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in the table.

## The symbols used in E-R diagrams are:

SYMBOL PURPOSE

Represents Entity sets.

Represent attributes.

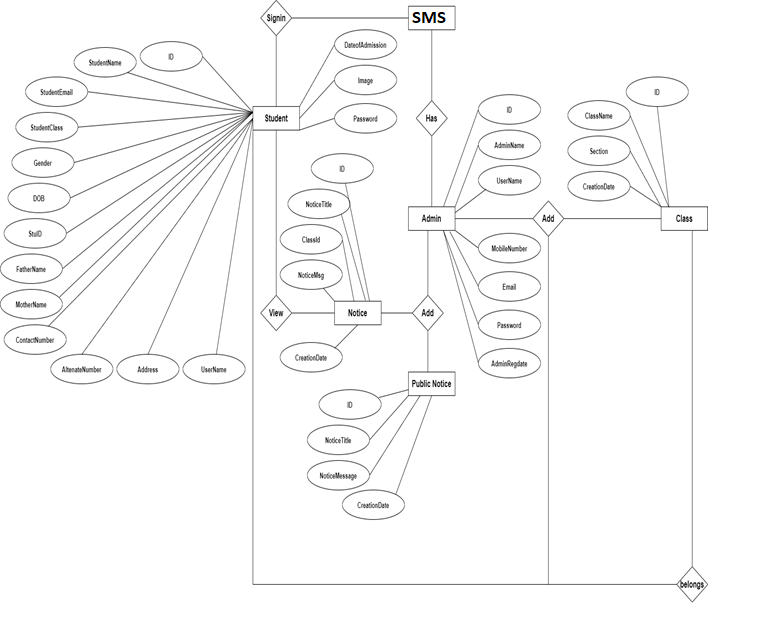
Represent Relationship Sets.

Line represents flow

Structured analysis is a set of tools and techniques that the analyst.

To develop a new kind of a system:

The traditional approach focuses on the cost benefit and feasibility analysis, Project management, and hardware and software selection a personal consideration.



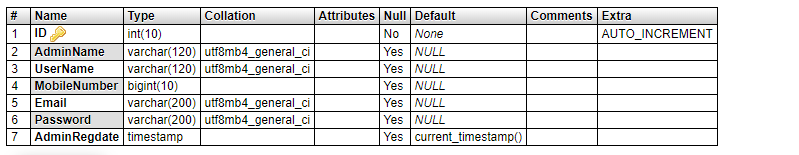
**Database Design**

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

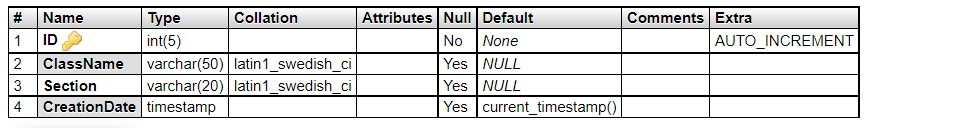
A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MySQL database has been chosen for developing the relevant databases.

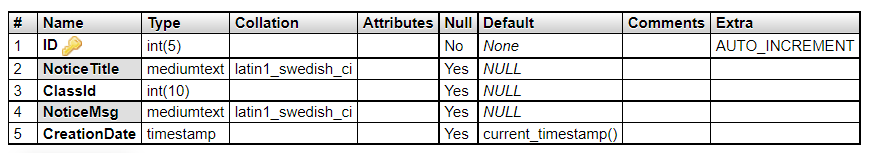
**Student Management System (SMS) contains 6 MySQL tables :**

**tbladmin table Structure :** This table store the admin login and personal Details.

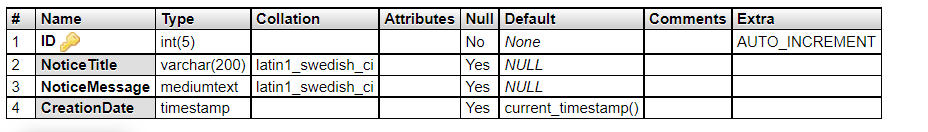


**tblclass table Structure :** This table store the class and section.

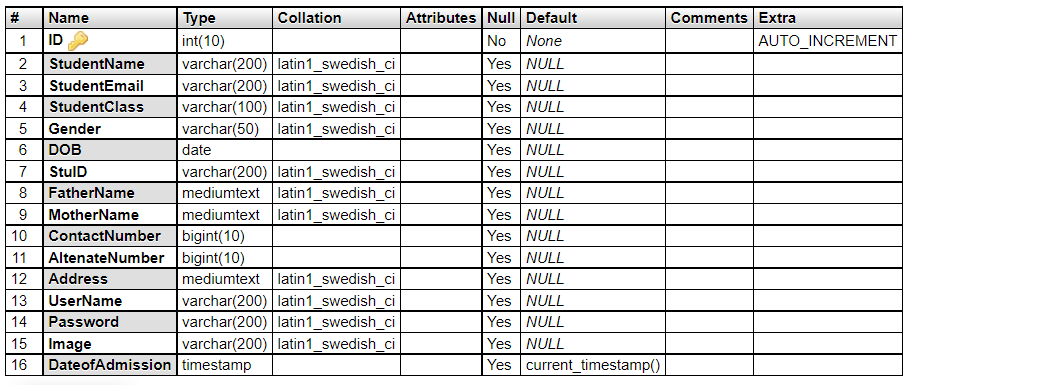


**tblnotice table Structure :** This table store the notices detail which is announced by admin.

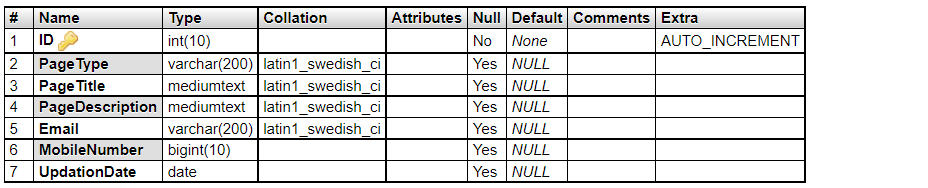
**tblpublicnotice table Structure :** This table store the public notices detail which is announced by admin.



**tblstudent table Structure :** This table store the students details of educational organization.

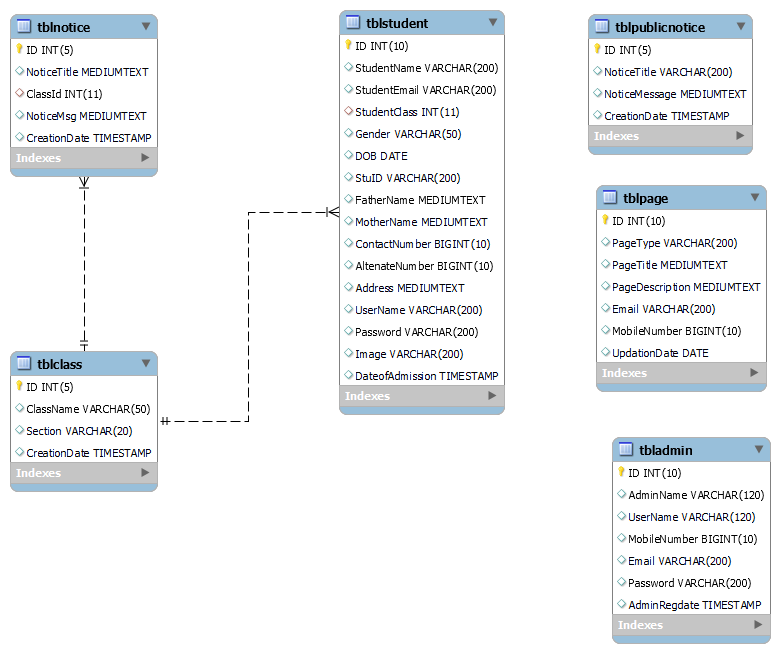


**tblpage table Structure :** This table store the details of about us and contact us pages.



**Class Diagram:**

The class diagram shows a set of classes, interfaces, collaborations and their relationships.



**System Testing**

**SOFTWARE TESTING TECHNIQUES:**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, designing and coding.

**TESTING OBJECTIVES:**

1. Testing is process of executing a program with the intent of finding an error.
2. A good test case design is one that has a probability of finding an as yet undiscovered error.
3. A successful test is one that uncovers an as yet undiscovered error.

These above objectives imply a dramatic change in view port.

Testing cannot show the absence of defects, it can only show that software errors are present.

There are three types of testing strategies

1. Unit test
2. Integration test

3. Performance test

**Unit Testing:**

Unit testing focuses verification efforts on the smallest unit of software design module. The unit test is always white box oriented. The tests that occur as part of unit testing are testing the module interface, examining the local data structures, testing the boundary conditions, execution all the independent paths and testing error-handling paths.

**Integration Testing:**

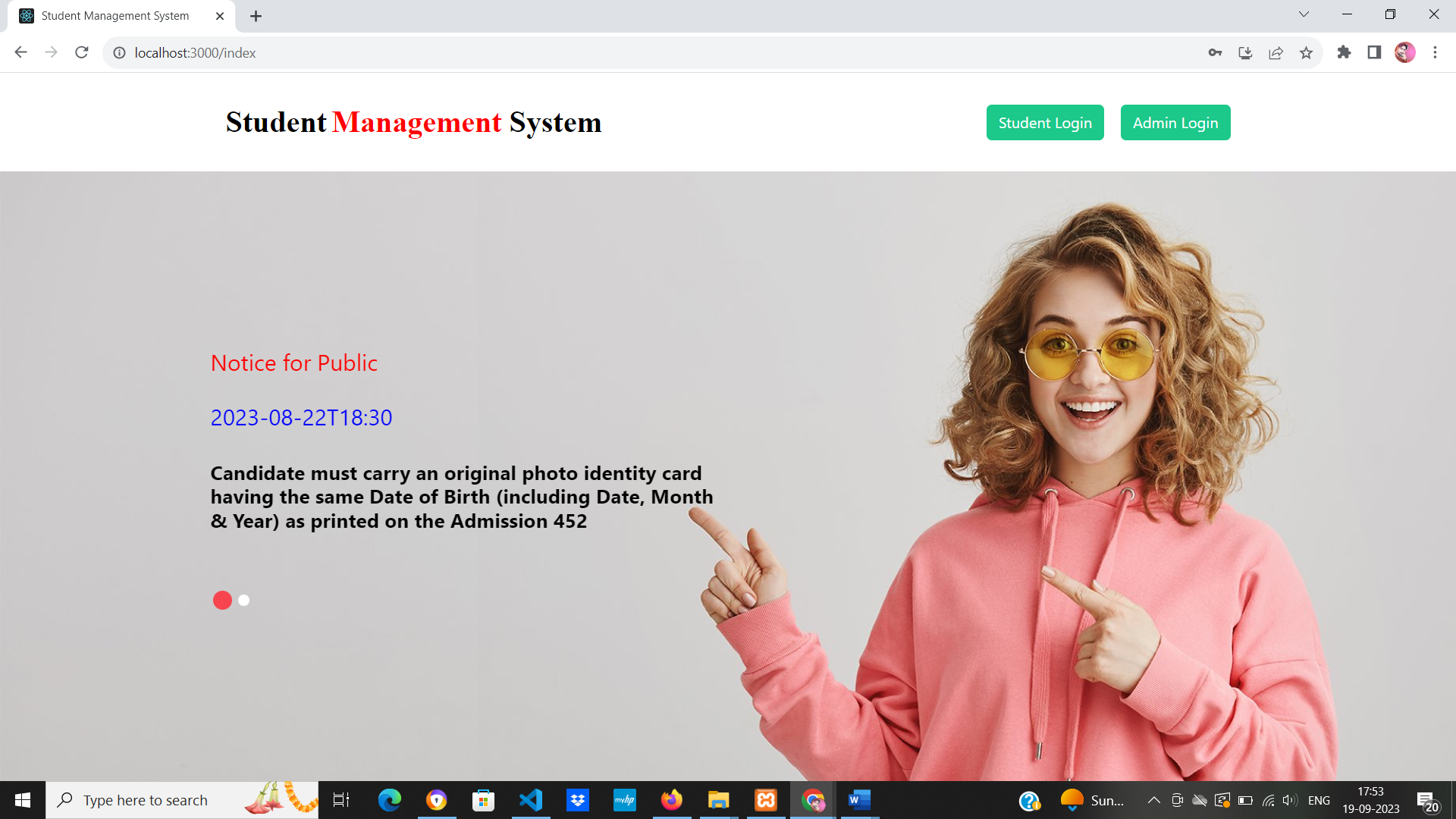
Integration testing is a systematic technique or construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. Scope of testing summarizes the specific functional, performance, and internal design characteristics that are to be tested. It employs top-down testing and bottom-up testing methods for this case.

**Performance Testing:**

Timing for both read and update transactions should be gathered to determine whether system functions are being performed in an acceptable timeframe.

**Output Screen of Project**

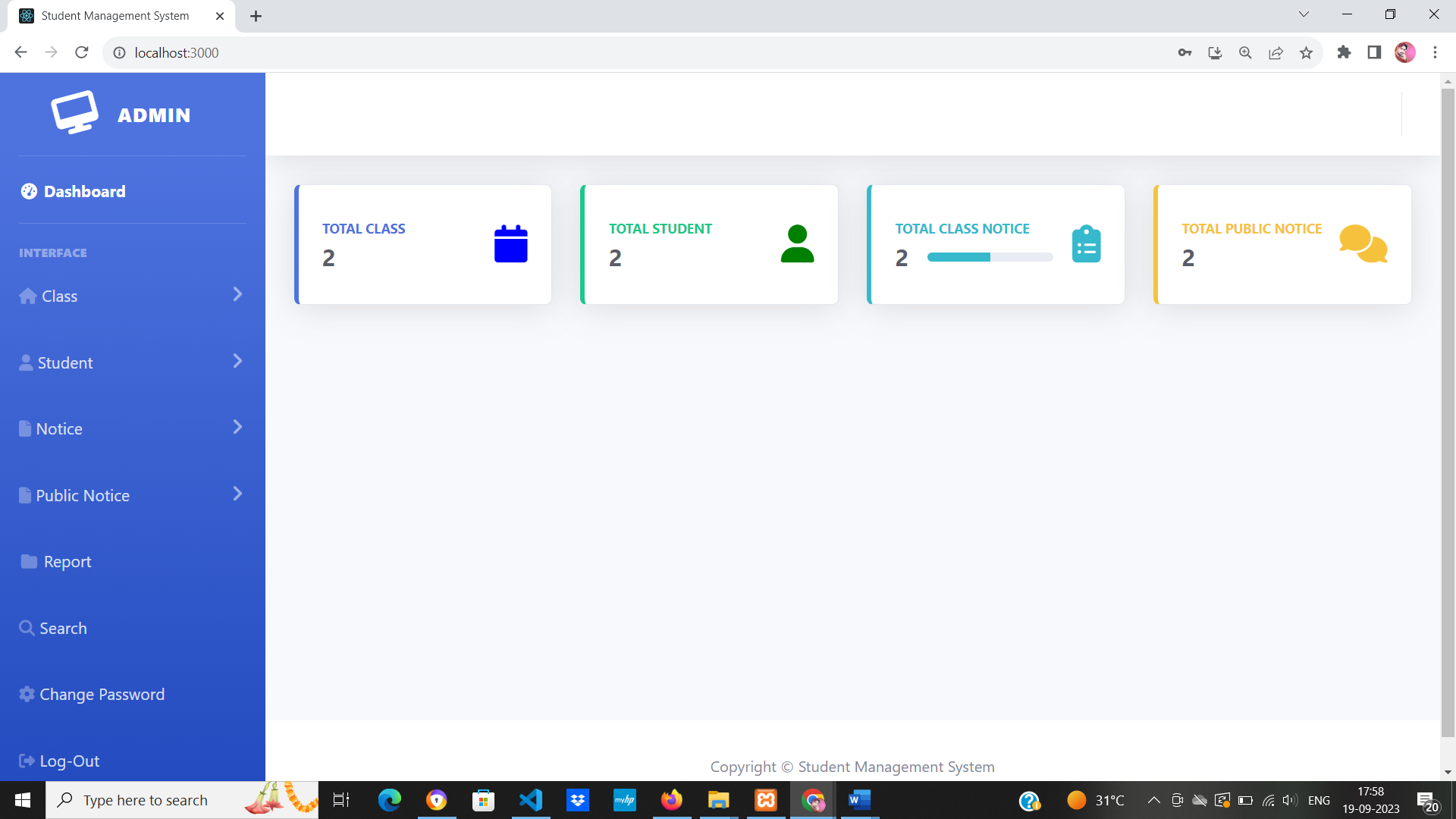
**Home Page**



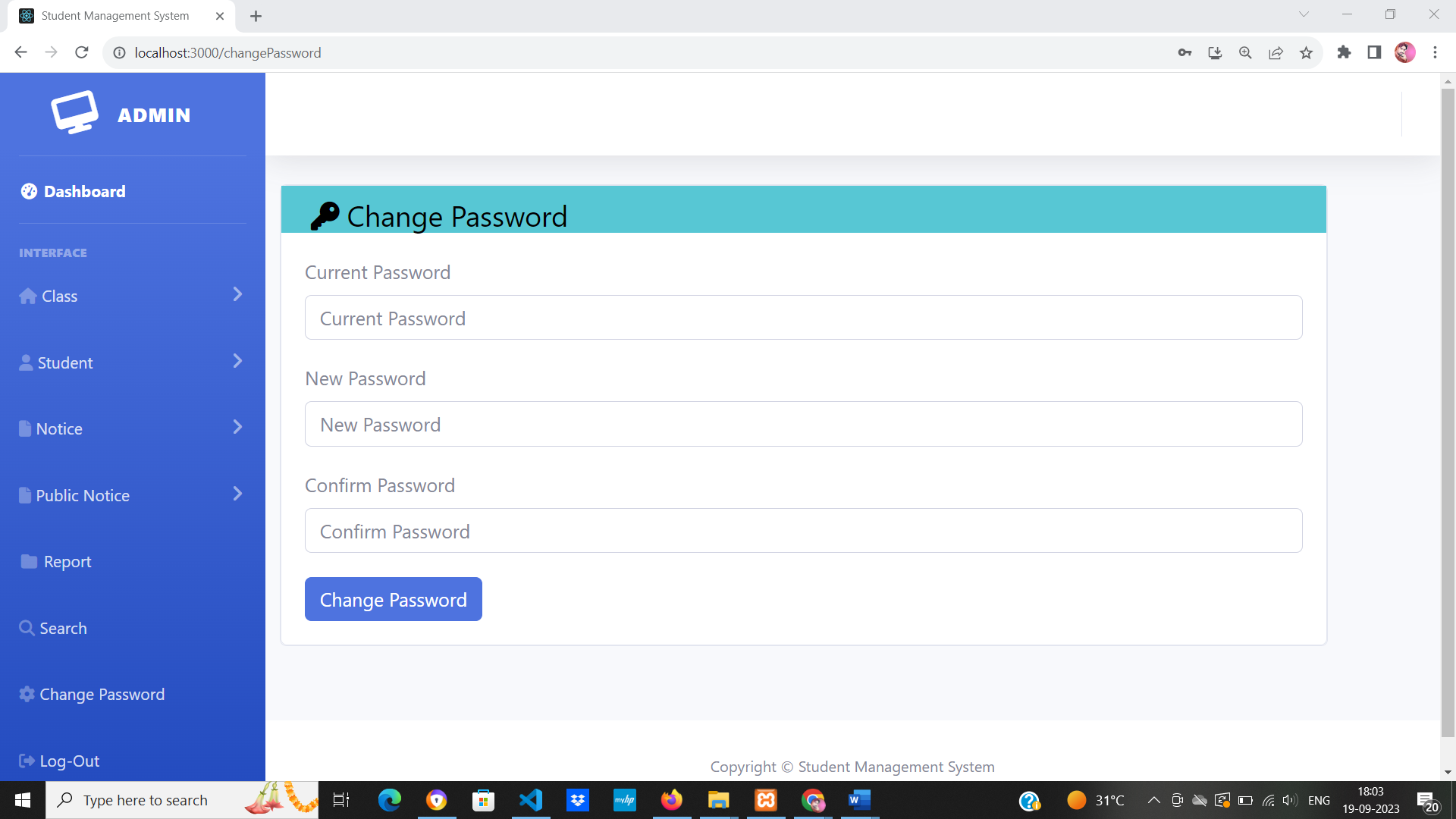
**Admin Login Page**



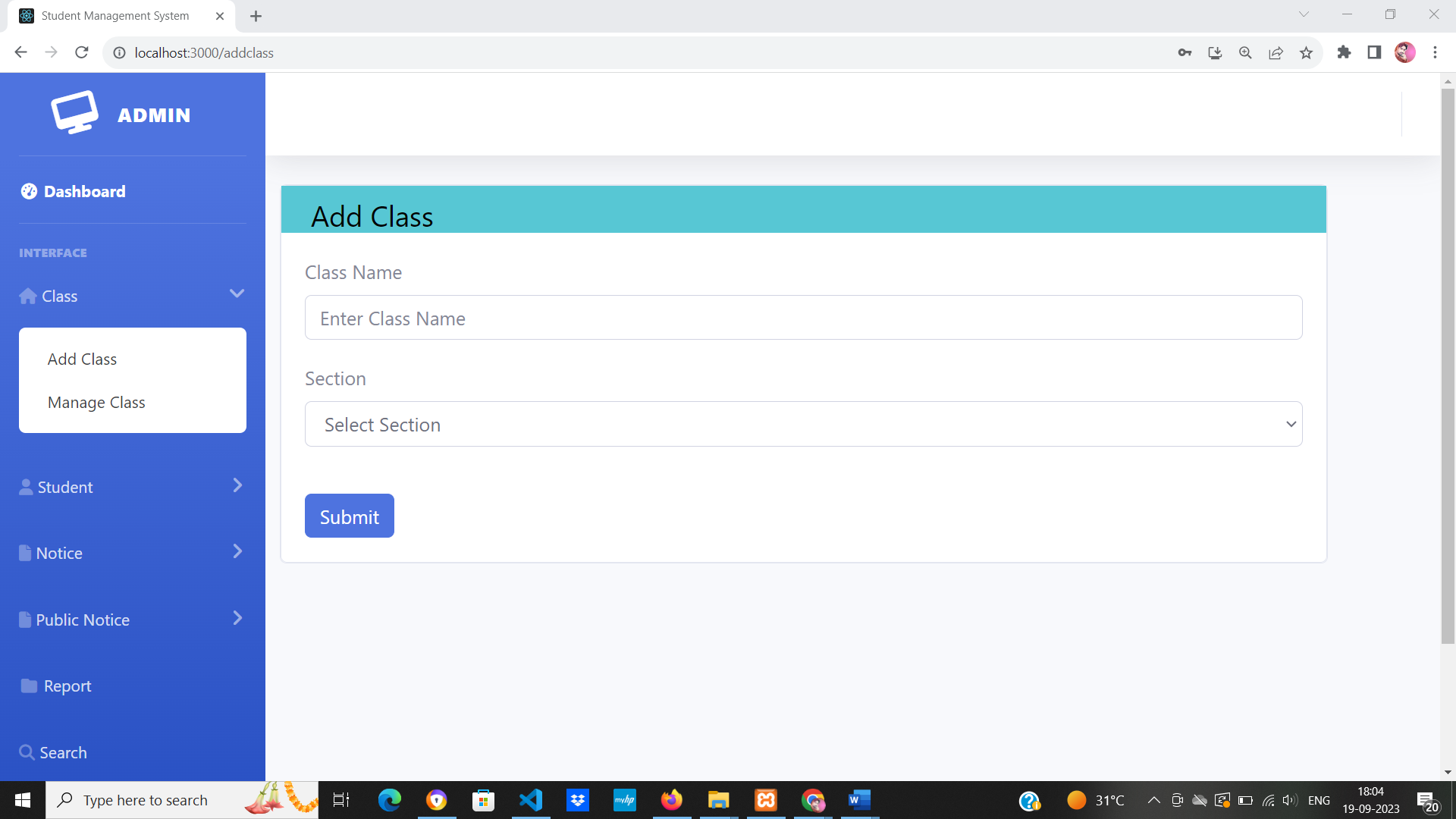
**Dashboard**



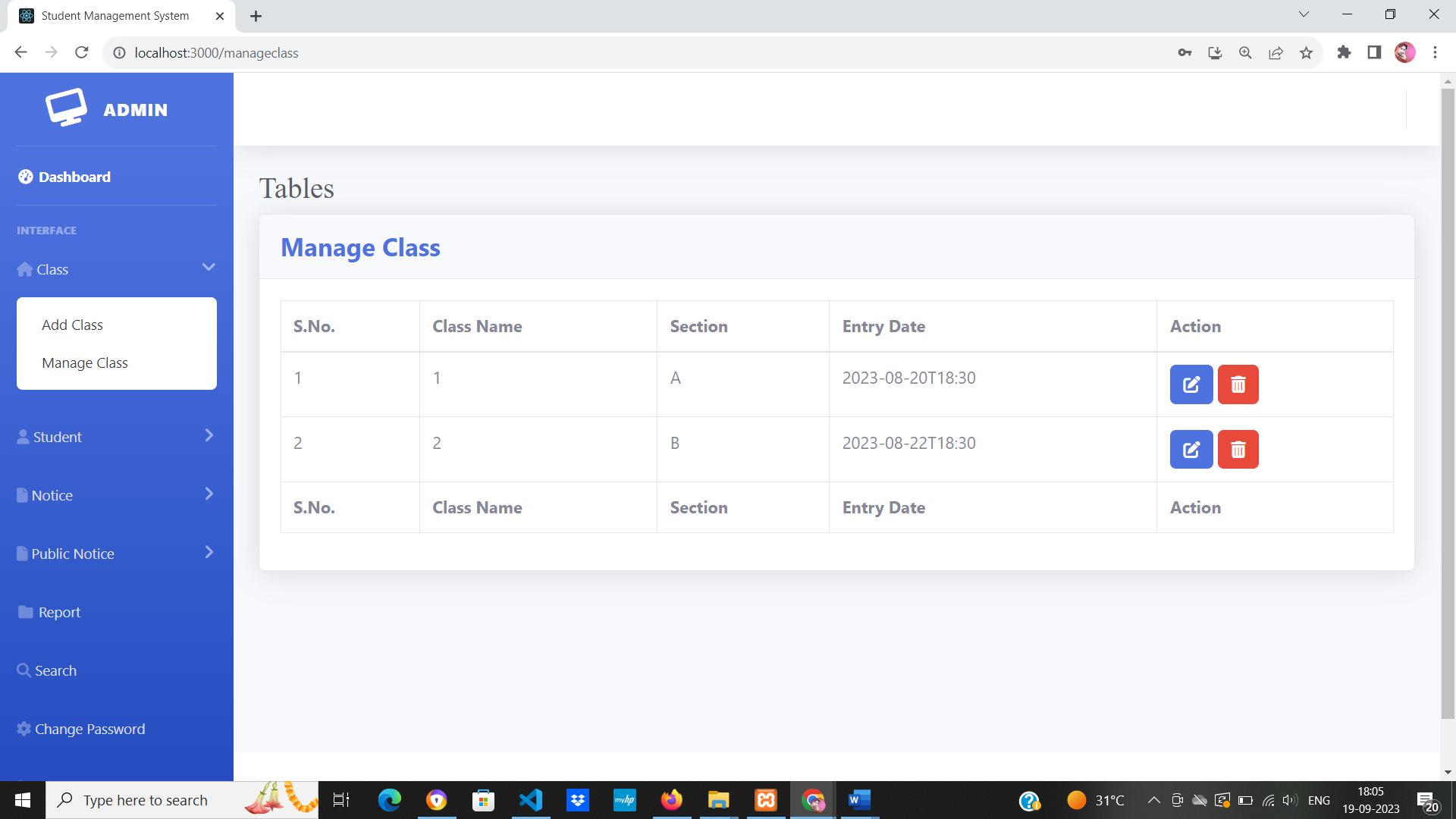
**Change Password**



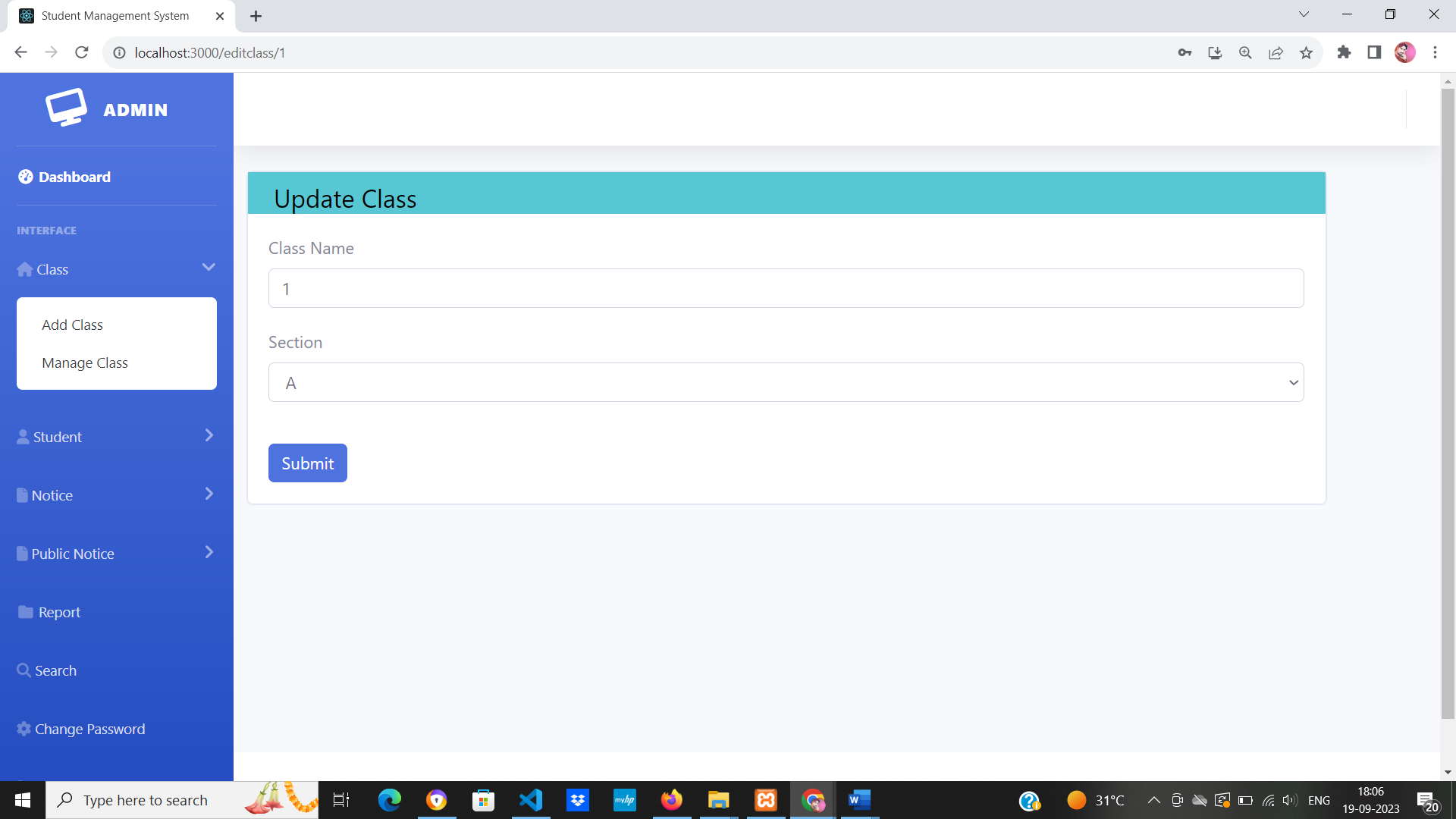
**Add Class**



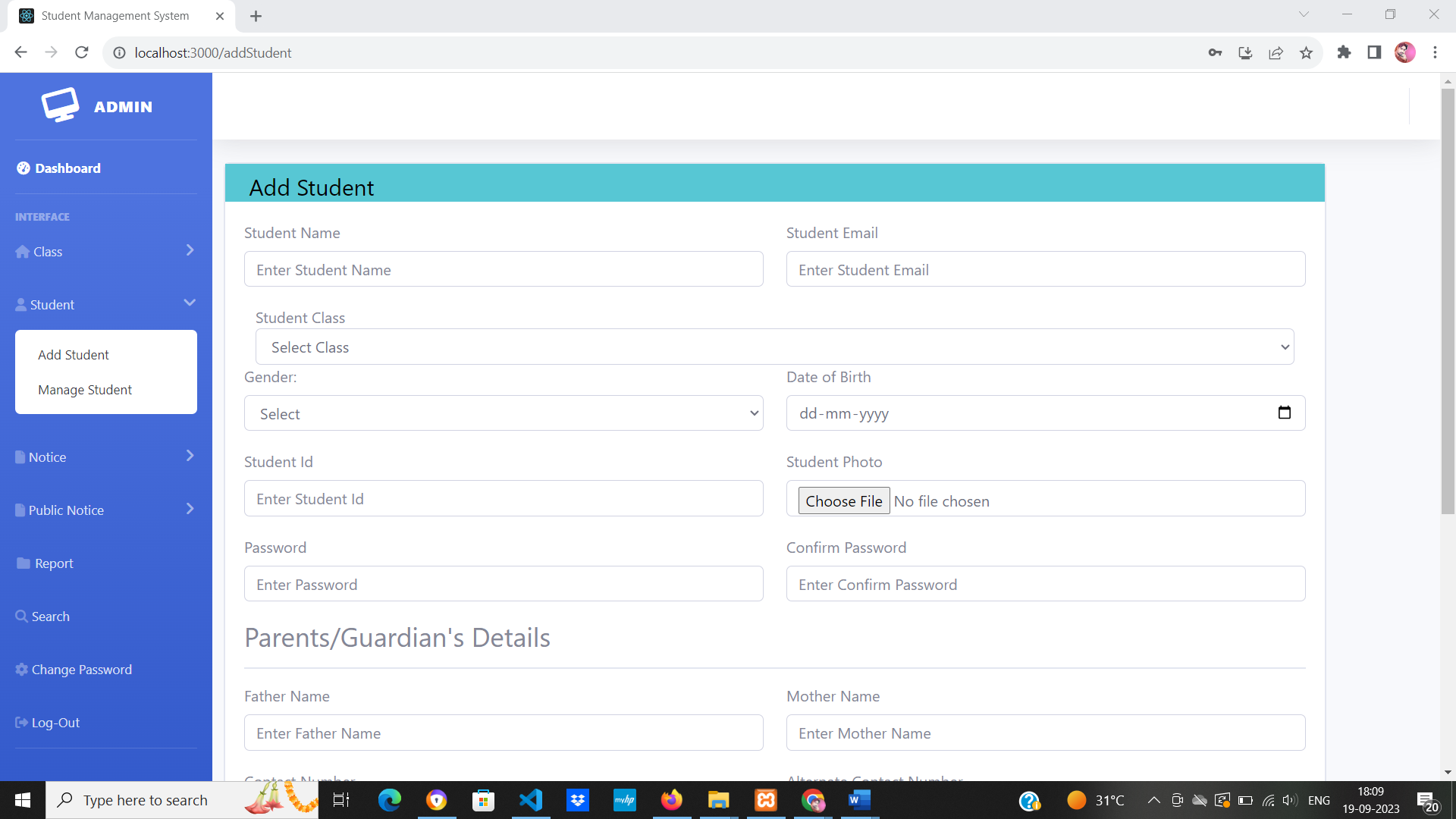
**Manage Class**



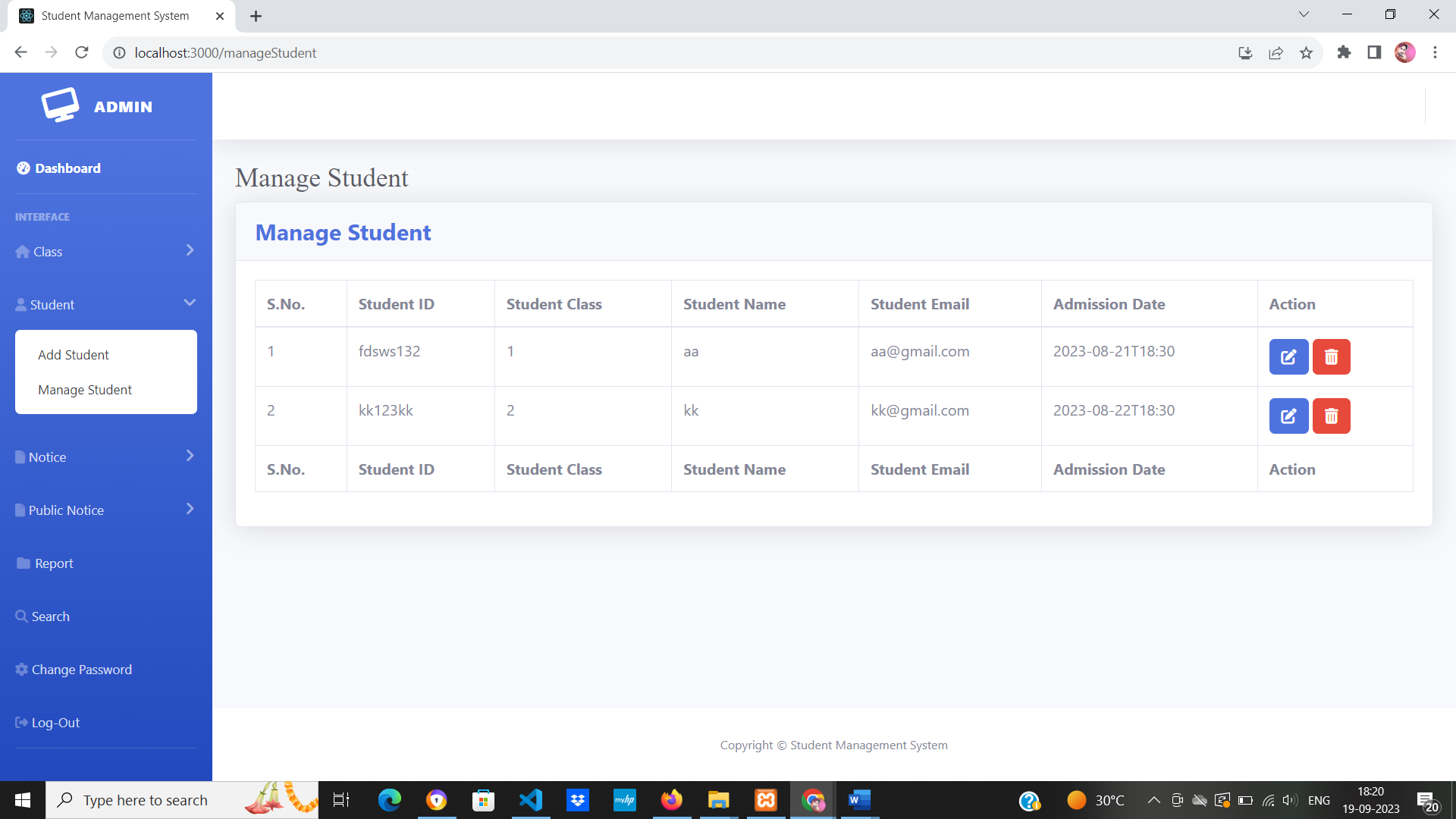
**Update Class**



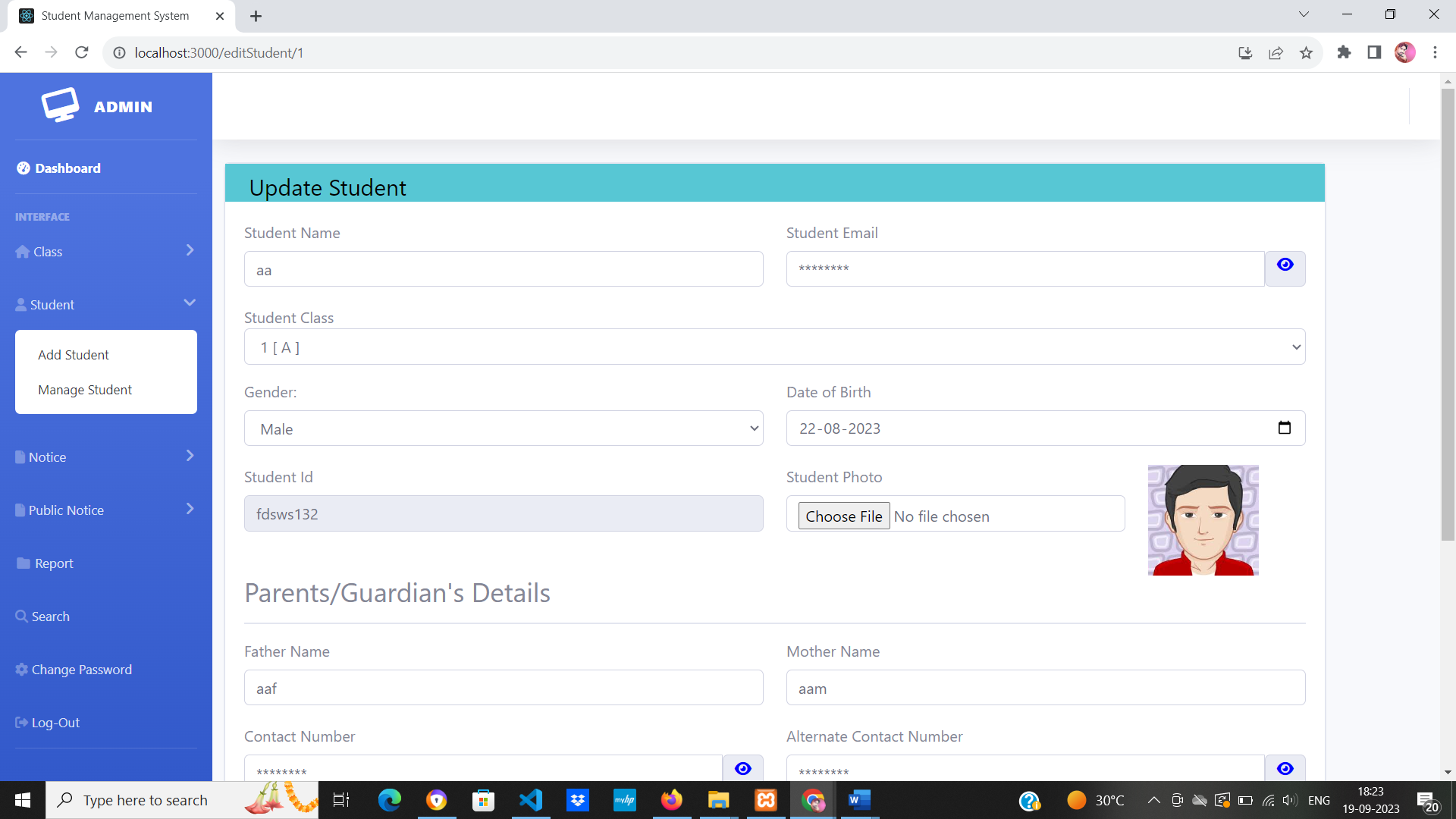
**Add Students**



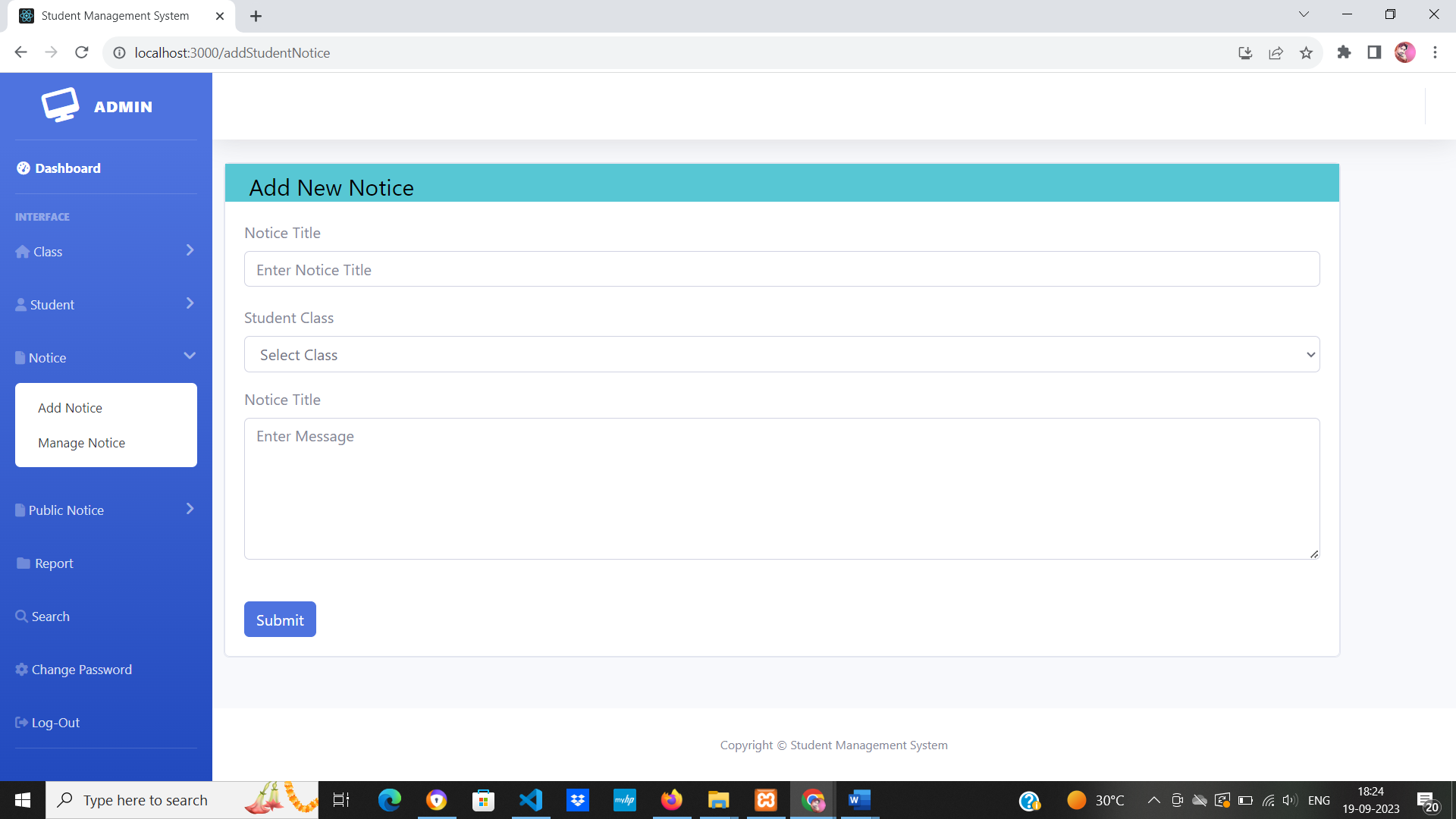
**Manage Students**



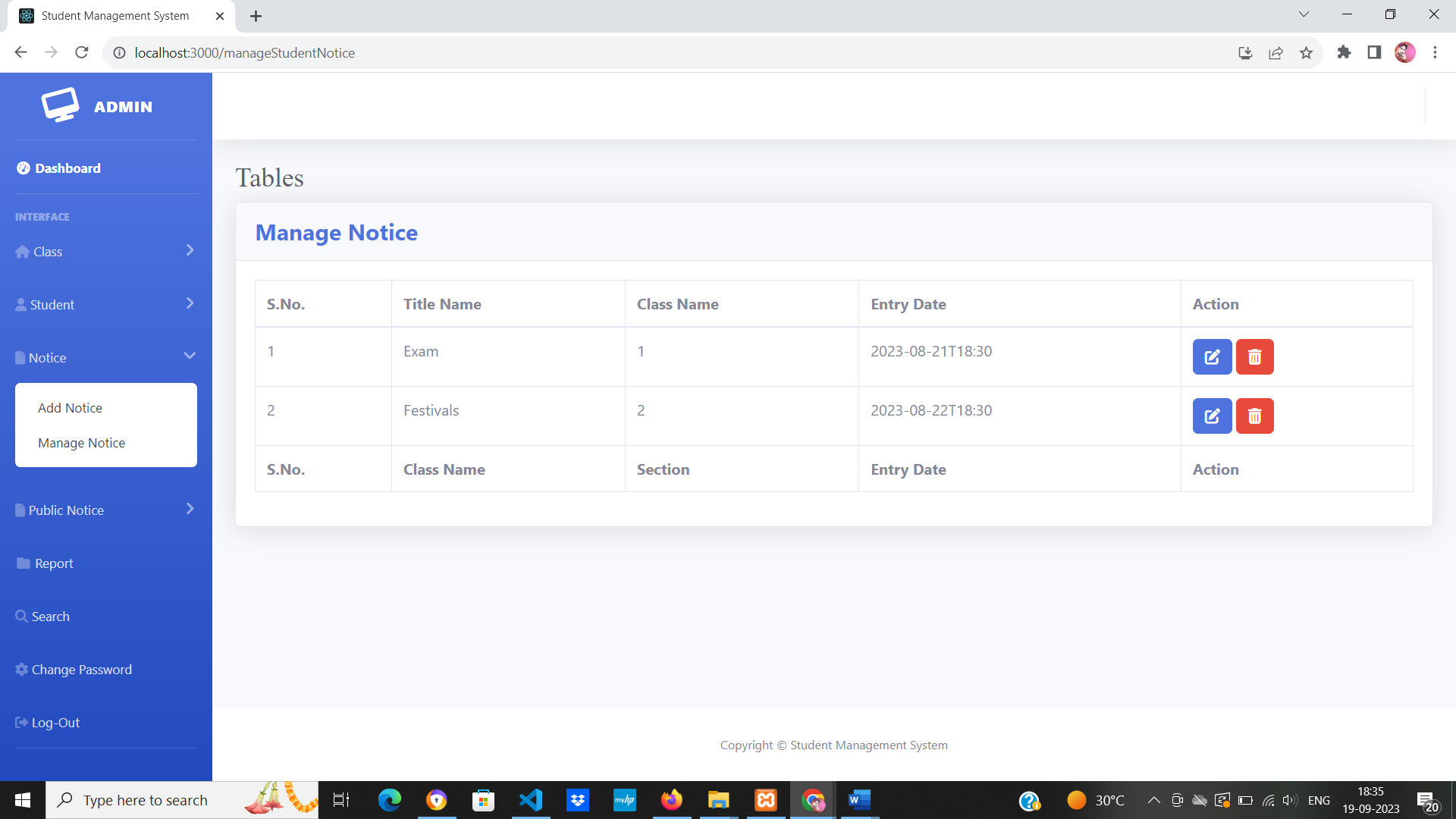
**Update Students**



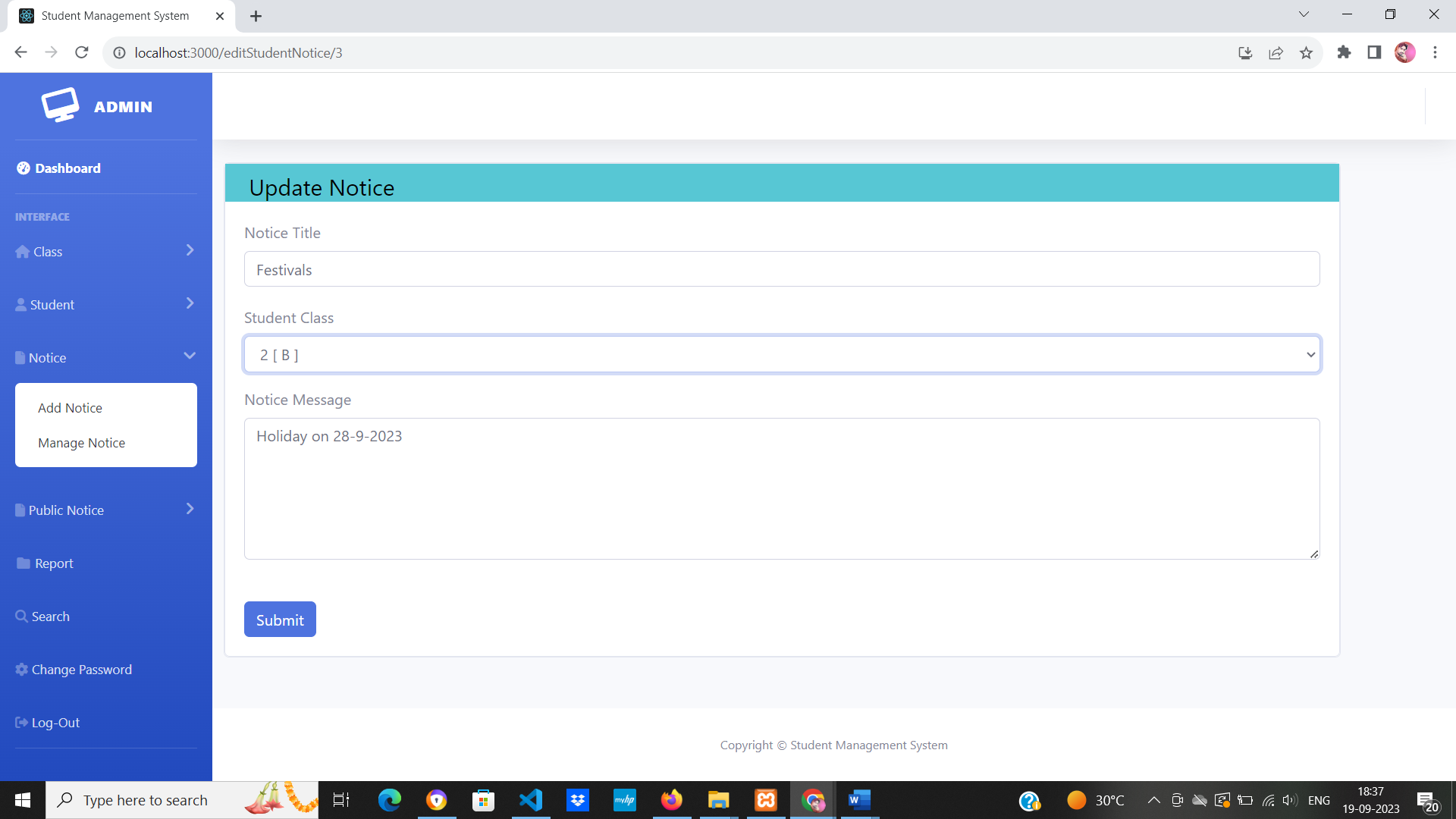
**Add Notice**



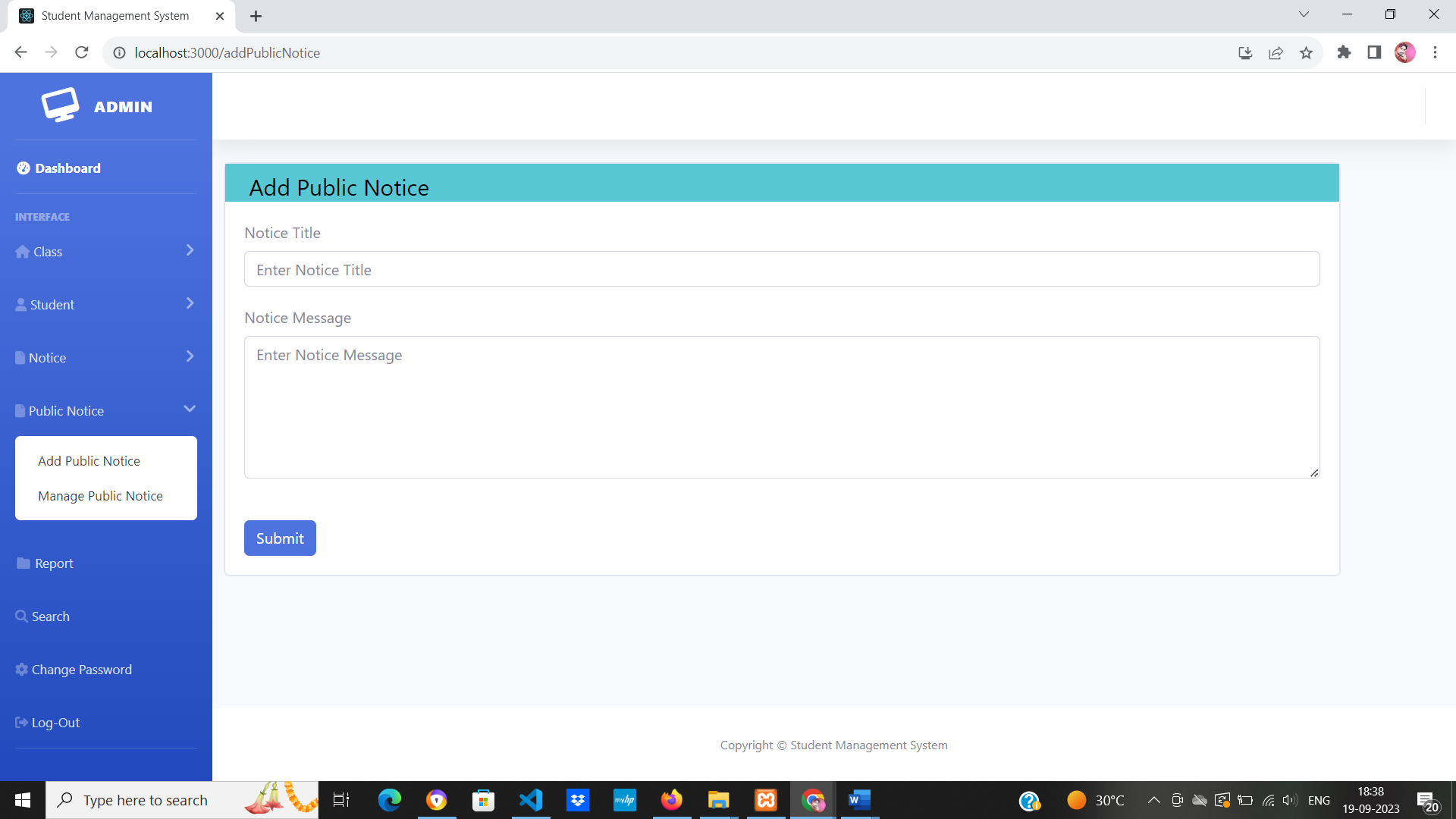
**Manage Notice**



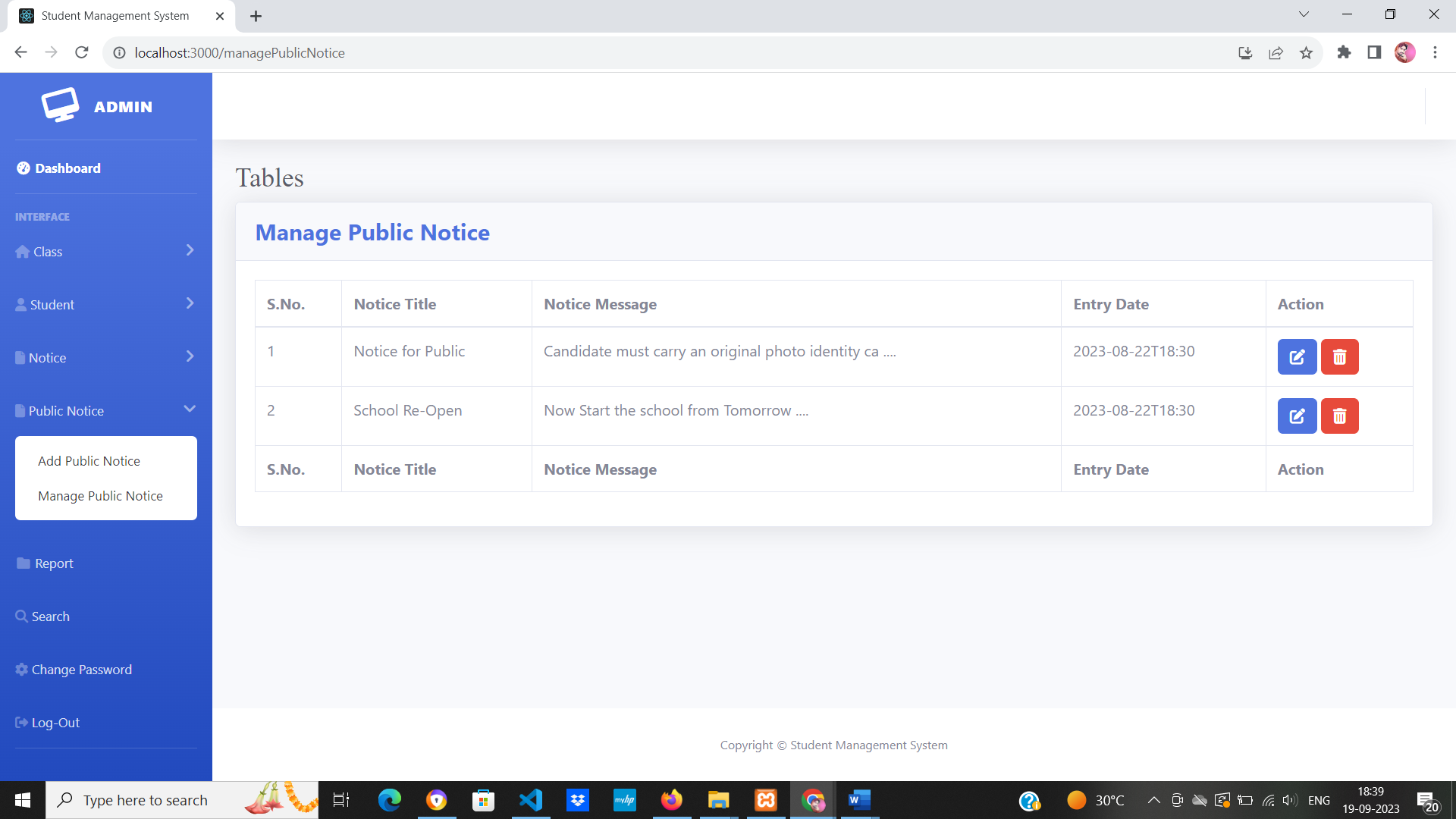
**Update Notice**



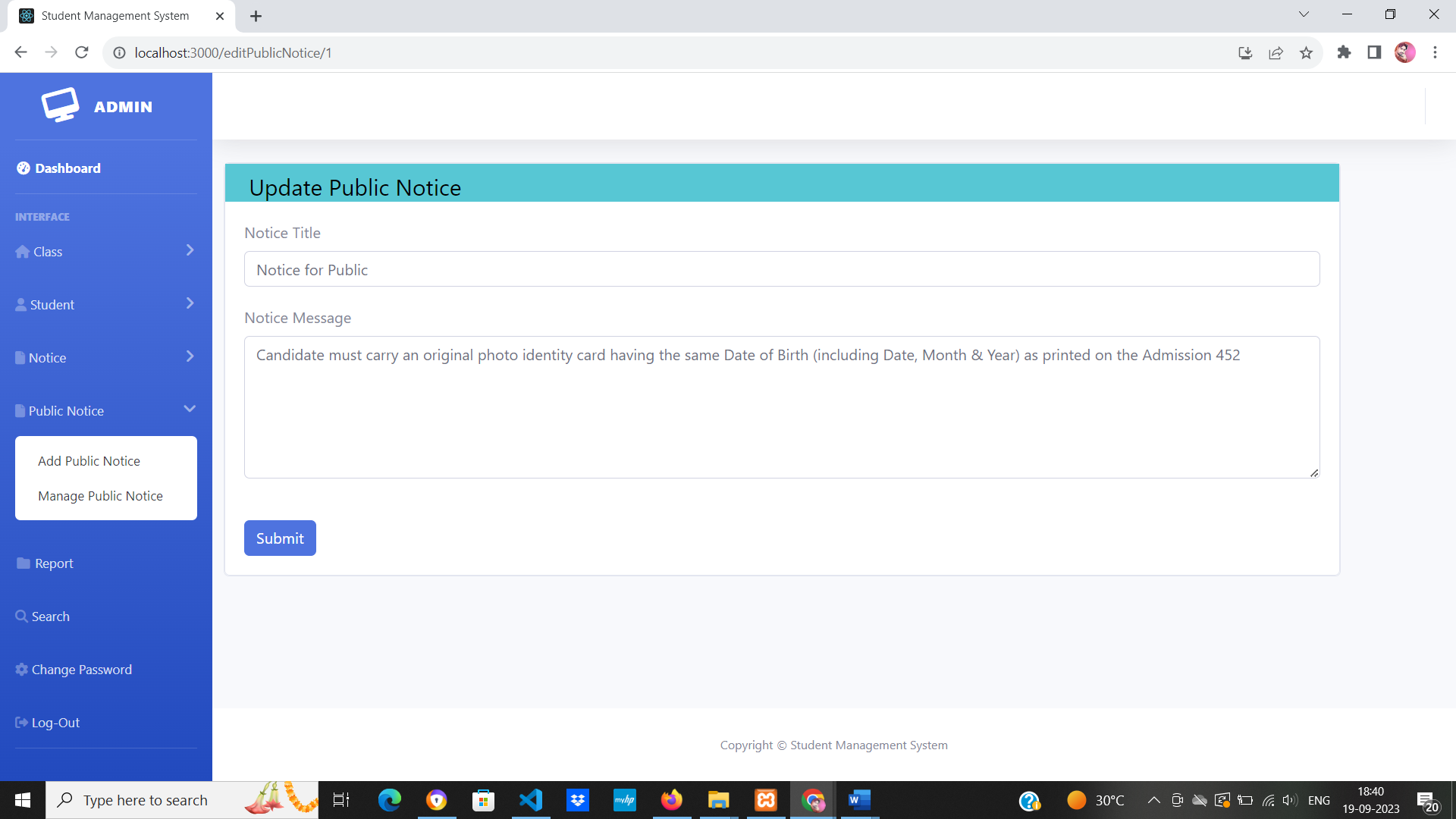
**Add Public Notice**



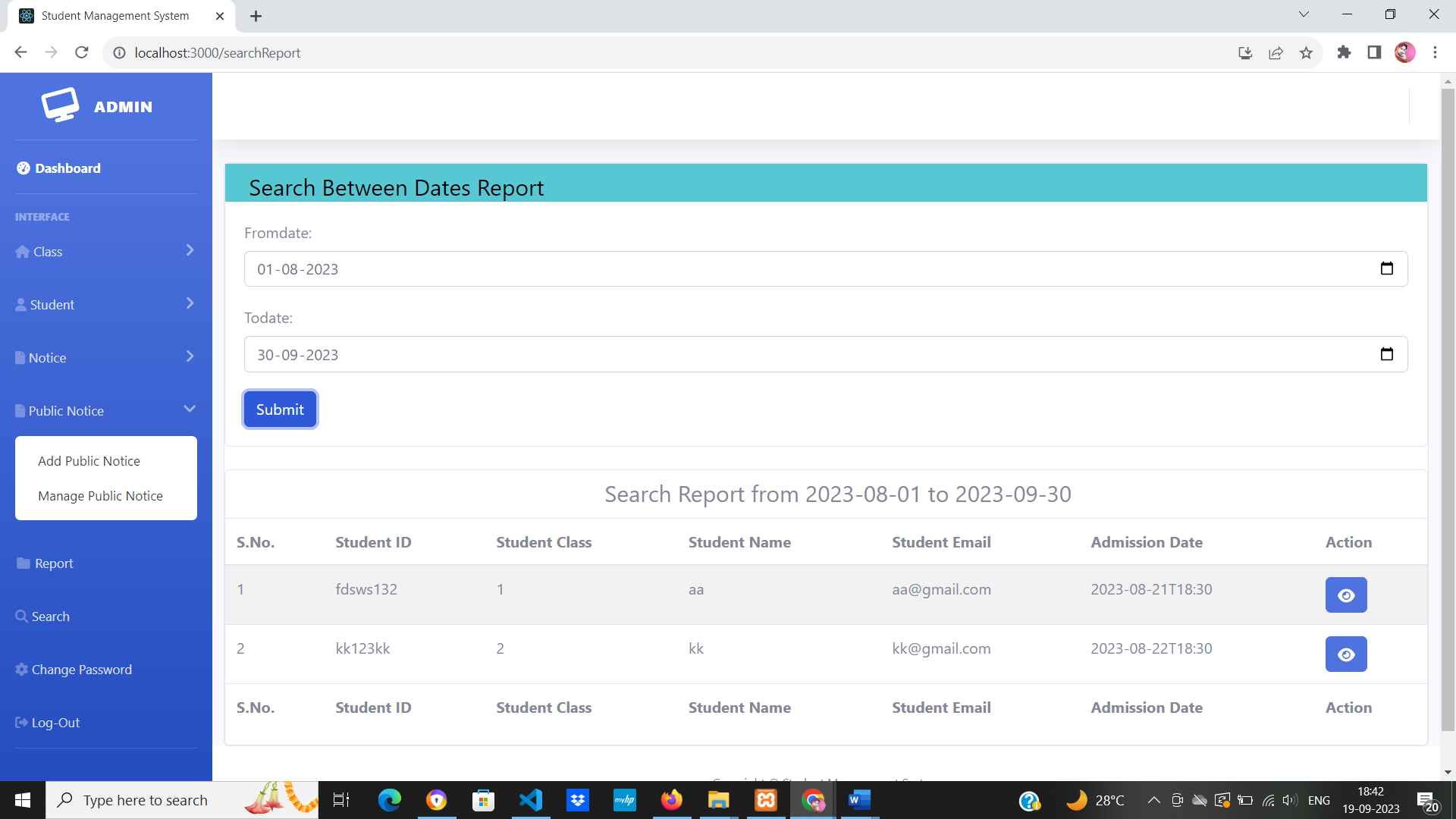
**Manage Public Notice**



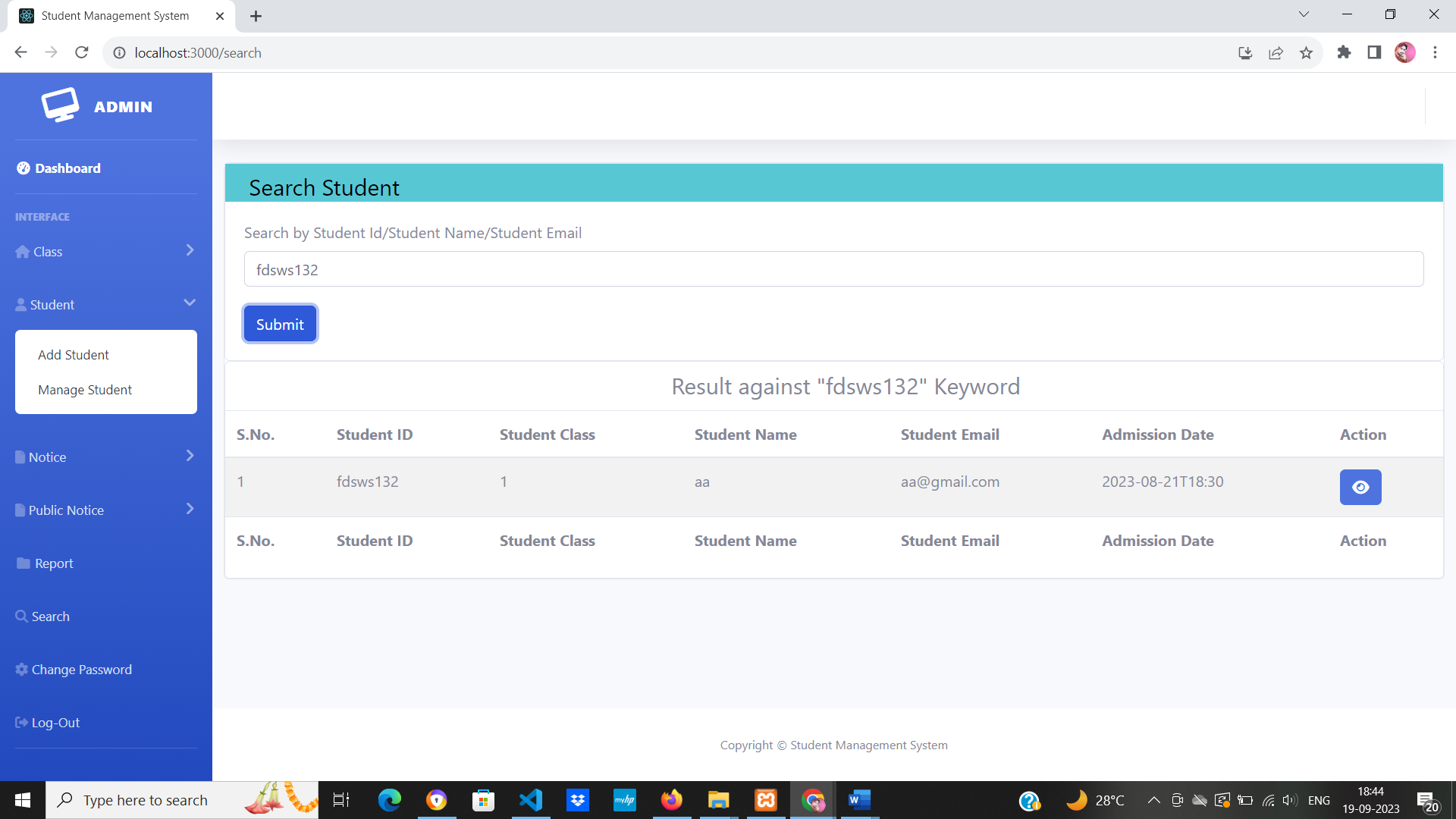
**Update Public Notice**



**Between dates report**



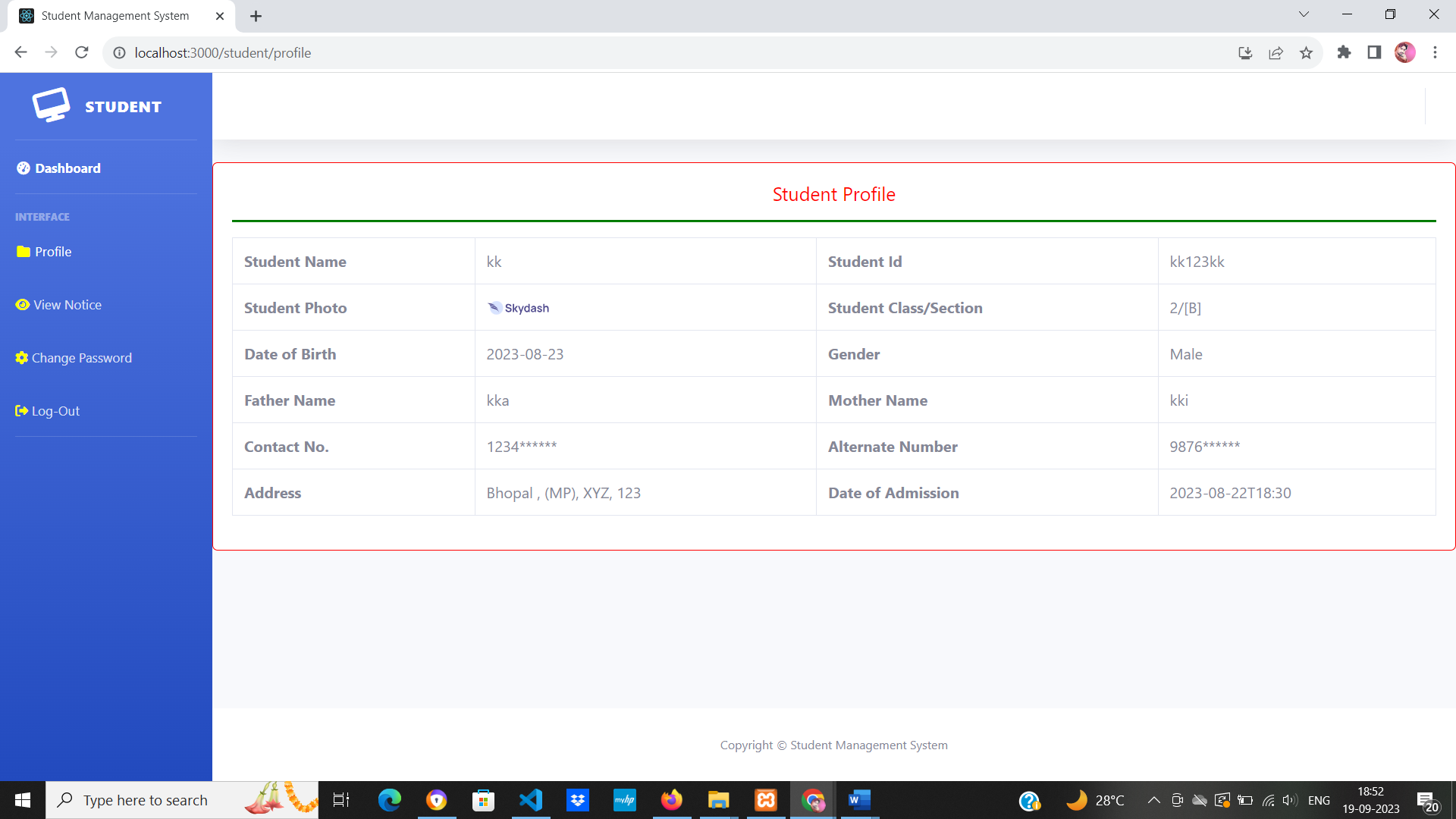
**Search Students**



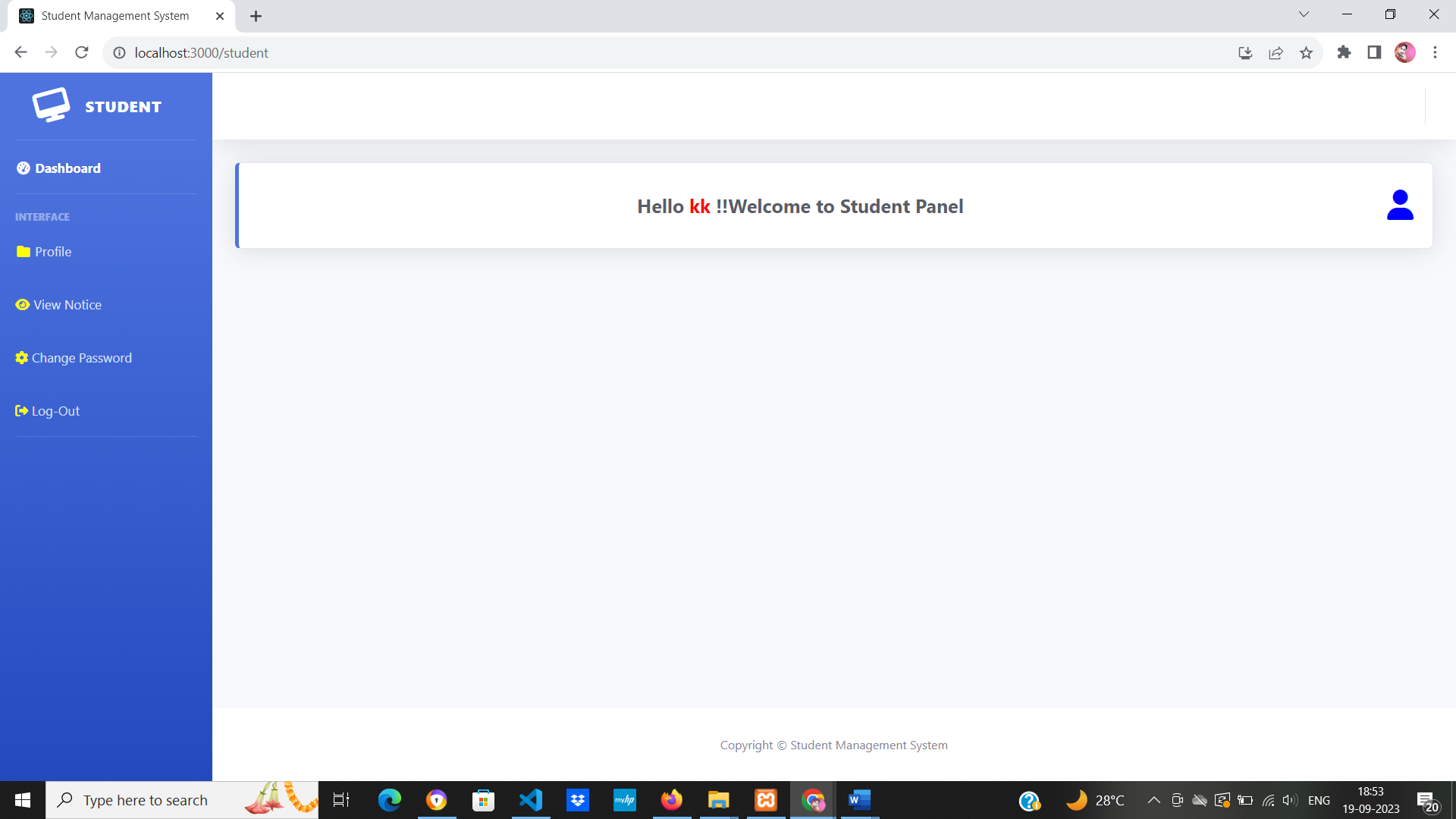
**Student Login Page**



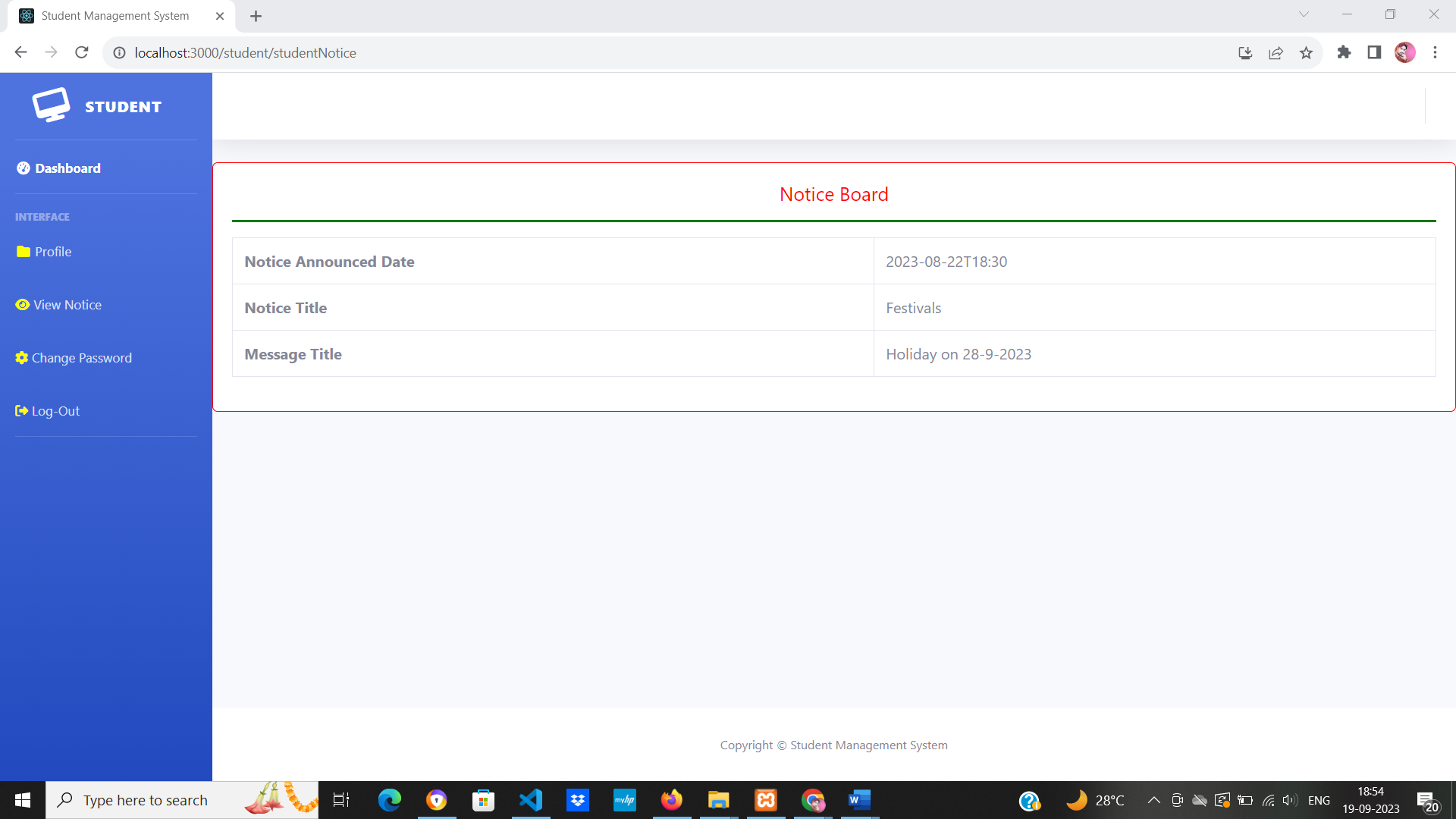
**Students Profile**



**Dashboard**



**View Notice**



**Conclusion**

The project titled as **Student Management System** was deeply studied and analyzed to design the code and implement. It was done under the guidance of the experienced project guide. All the current requirements and possibilities have been taken care during the project time.

**Student Management System** can be used by education institutes to maintain the records of students easily. Achieving this objective is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information may be very time consuming. All these problems are solved using this project.

**Bibliography**

* **React.js:**
* Official React Documentation: <https://reactjs.org/>
* React Tutorial: <https://reactjs.org/tutorial/>
* React Router Documentation: <https://reactrouter.com/>
* React Bootstrap Documentation: <https://react-bootstrap.github.io/>
* **Node.js:**
* Official Node.js Documentation: <https://nodejs.org/>
* Node.js API Documentation: <https://nodejs.org/api/>
* Express.js Documentation: <https://expressjs.com/>
* NPM (Node Package Manager) Registry: <https://www.npmjs.com/>
* **MySQL:**
* Official MySQL Documentation: <https://dev.mysql.com/doc/>
* MySQL Tutorial: <https://www.mysqltutorial.org/>
* MySQL Workbench Documentation: <https://dev.mysql.com/doc/workbench/en/>
* MySQL Developer Zone: <https://dev.mysql.com/>
* phpMyAdmin Official Website: <https://www.phpmyadmin.net/>
* **REFERENCE BOOKS**
* "Learning React: Functional Web Development with React and Redux" by Alex Banks and Eve Porcello
* "Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node" by Vasan Subramanian
* "Node.js, MongoDB, and Angular Web Development: The definitive guide to using the MEAN stack" by Brad Dayley
* "MySQL Cookbook: Solutions for Database Developers and Administrators" by Paul DuBois
* "Express in Action: Writing, building, and testing Node.js applications" by Evan Hahn
* "React Router: Core Concepts and Best Practices" by Todd Sammons
* "Web Development with Node and Express: Leveraging the JavaScript Stack" by Ethan Brown
* "Node.js Design Patterns: Design and implement production-grade Node.js applications using proven patterns and techniques" by Mario Casciaro and Luciano Mammino