

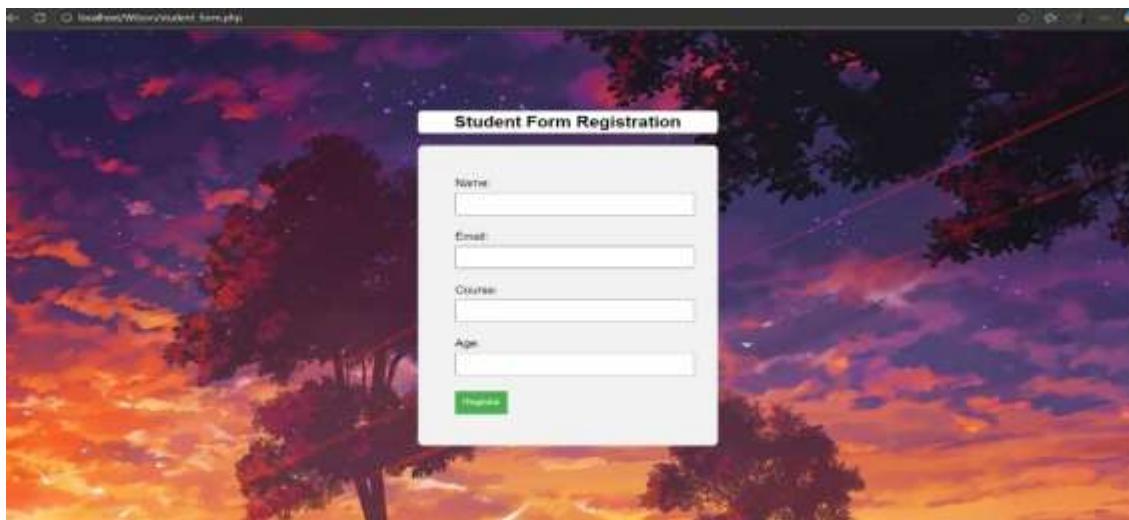
Document Title: Student Registration System Documentation

Overview:

The images illustrate a **Student Registration System** that enables users to register and manage student information, including their name, email, course selection, and age. The system consists of two main components:

1. Student Registration Form:

- The form shown in the first image allows users to input their information (name, email, course, and age) and submit it to the system for registration.
- After submitting, the data is stored in a database and can be managed or edited later.
- The design of the form is simple, user-friendly, and visually appealing, with a serene background of animated skies that provide a calm atmosphere for users.



2. Registered Students List:

- The second image shows a table of registered students.
- The table includes key details such as student ID, name, email, course, and age, which are populated from the form submissions.
- Each row in the table has action buttons ("Edit" and "Delete") to allow administrators to manage the records. These functions enable the modification of student information or the removal of student records from the system.

Registered Students List						
#	ID	Name	Email	Course	Age	Actions
29	asdas	asdasd@yahoo.com	cs	10	Edit Delete	
30	dfqdfg	dfqds@yahoo.com	asd	12	Edit Delete	
31	adadf	dataf@yahoo.com	cs	3	Edit Delete	
32	id	asda@gmail.vom	css	3	Edit Delete	

3. Database View (phpMyAdmin):

- The third image illustrates the actual **MySQL database** in **phpMyAdmin**, where the student data is stored.
 - The records are stored in a table named "students," and each row corresponds to a student's registration information.
 - The interface provides options for administrators to edit, delete, or export records directly from the database.

Showing 1 result(s) | Columns | Copy | Captain SQL | Generate PDF | Go to top | Help

	<input type="checkbox"/> Show all	Number of rows:	25	<input type="button" value="▼"/>	Filter rows:	Search this table:	<input type="button" value="Sort by key"/>	<input type="button" value="None"/>	<input type="button" value="▼"/>
Extra options									
	<input type="button" value="New"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	id	name	email	course	age
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	29	asdas	asdasd@yahoo.com	cs	10	
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	30	dgfdg	dfgds@yahoo.com	asd	12	
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	31	adadf	daaf@ yahoo.com	cs	3	
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	32	id	asda@gmail.com	css	3	
<input type="button" value="Check all"/> <input type="button" value="With selected"/> <input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/> <input type="button" value="Export"/>									

Purpose:

1. Student Registration:

- The main purpose of the registration form is to allow students to provide their basic details such as name, email, course, and age to sign up for a course. This data is then submitted to the database, facilitating the management of student information.

2. Data Management and Access:

- The system's interface allows administrators to view and manage the list of registered students. Through the registered students' list, the admin can track enrollment, modify details, or remove records when necessary.

3. Database Integration:

- The system uses a **MySQL database** to store student records, which ensures that data is persistent across sessions.
- The use of phpMyAdmin provides a powerful interface for admins to access and manage the data. The ability to edit and delete records directly from the database makes the process of managing registrations more efficient.

4. User-Friendly Interface:

- The system's clean, straightforward design makes it easy for users to interact with. Students can quickly input their information, and administrators can easily manage the data.

5. Real-Time Updates and Actions:

- After data is submitted through the registration form, the records are instantly reflected in the database, allowing the admin to see new entries in real time.
 - Actions like editing and deleting allow for flexibility in managing the students' information, ensuring data accuracy and integrity.
-

Conclusion:

This Student Registration System combines a user-friendly interface for students with an efficient backend management system for administrators. The integration of the registration form, student list, and database interface allows for smooth data entry, management, and storage. The system provides flexibility and simplicity for managing student data while maintaining a clean, visually appealing design.