# Student Management System - Documentation

## Overview

This Python-based Student Management System allows users to perform CRUD (Create, Read, Update, Delete) operations on student records. It stores data in a JSON file and provides functionality to export student records into a CSV file.

## Requirements

• Python 3  
• Pandas library  
• JSON and OS modules (built-in)

## Features

• Add Student

• View All Students

• Search Student

• Update Student Record

• Delete Student Record

• Export Data to CSV

## Implementation Details

### Class: Student

The `Student` class handles all operations related to student records.

#### \_\_init\_\_(self, std\_name, std\_age)

• Initializes a student with a name and age.  
• Stores them as instance variables.

#### add\_student(self)

• Prompts the user to input subject names and marks. Stores data in a dictionary and saves it to Student\_data.json.

#### del\_student()

• Deletes a student record from the JSON file.

#### update\_student()

• Allows updating a student’s name, age, or marks.

#### search\_student()

• Searches for a student by name and displays their details.

#### view\_students()

• Displays all students stored in Student\_data.json.

#### export\_to\_csv()

• Converts JSON data to a CSV file using Pandas.

## User Interface

The script provides a command-line interface for users to interact with:

1. Add Student 2. View All Students 3. Search Student  
4. Update Record 5. Delete Record 6. Export to Excel

## Example Usage

### Adding a Student

Enter Student Name: John Doe  
Enter Student Age: 20  
Enter Number of Subjects: 2  
Enter Subject Name: Math  
Enter Marks: 85

### Searching a Student

Enter the name of the student to search: John Doe  
  
Student Found!  
Name: JOHN DOE  
Age: 20  
Marks:  
 - Math: 85  
 - Science: 90

## File Storage

All student data is stored in `Student\_data.json`. The JSON structure:

{ "JOHN DOE": { "Student Name": "JOHN DOE", "Age": 20, "Marks": { "Math": 85, "Science": 90 } } }

## Error Handling

• Checks for duplicate names when adding/updating a student.  
• Handles invalid inputs for age and marks.  
• Prevents division by zero during calculations.

## Conclusion

This Student Management System efficiently manages student records, allows updates and deletions, and supports exporting data to CSV. It is useful for educational institutions or personal record-keeping.