**Summary of "Student Records Manager (Mini Project)"**

**Project Overview:** The "Student Records Manager" is a web-based mini project designed to manage and display student records, including their names and scores. The project consists of three HTML files (index.html, home.html, and all\_students.html), a CSS file, a JavaScript file, and a JSON file for data storage. The application allows users to add new student records, calculate and display average scores, and retrieve stored records from a JSON file.

**Purpose of Each Functionality:**

**index.html:** This file serves as the primary interface for adding new students and displaying statistics. Users input student names and scores (comma-separated), which are then added to the student list.

**home.html:** Provides a simple navigation interface that allows users to choose between adding new students and fetching data from the JSON file.

**all\_students.html:** Displays a table of all student data fetched from a JSON file, calculating the average score for each student.

**CSS file:** Defines the layout, styling, and responsive design for the HTML elements, ensuring a clean and user-friendly interface.

**JavaScript file (Script.js):**

**addStudent():** Adds a new student to the list and updates the display.

**calculateAverageScore():** Computes the average score for a student.

**findHighestLowestScores():** Determines the highest and lowest scores across all students.

**displayStudents():** Updates the DOM to reflect the current list of students and their respective average scores.

**saveToJson():** Simulates saving student data to a JSON file (this function does not actually save data but logs it).

**getData():** Fetches and displays student records from the JSON file.

**JSON file (students.json):** Stores the student data in a structured format, providing a persistent data source for fetching and displaying student records.

**Challenges Faced:** One of the main challenges encountered during development was handling the process of putting data into the JSON file while using a live server. Managing JSON files dynamically from the client side requires server-side handling or backend support, and during testing, it was difficult to update or modify JSON files directly from the live server setup.