This code represents a simple student record management system using HTML, CSS, and JavaScript. Here's a breakdown of what it does:

1. \*\*HTML Structure\*\*:

- The HTML file contains a form to input student details like full name, roll number, class, total marks, grade, and age.

- The form also includes validation error labels that appear if the corresponding fields are left empty.

- There's a table to display the list of student records entered via the form.

2. \*\*Styling (CSS)\*\*:

- The CSS defines the styling for various elements like tables, input fields, buttons, and error messages.

- It provides a structured layout for the form and table display.

3. \*\*JavaScript Functions\*\*:

- `onFormSubmit`: Handles form submission, validates input fields, and either inserts a new record or updates an existing one in the table.

- `readFormData`: Reads the input values from the form fields and returns them as an object.

- `insertNewRecord`: Inserts a new row with student record details into the table.

- `resetForm`: Resets the form fields after submission or editing.

- `onEdit`: Populates form fields with the selected row data for editing.

- `updateRecord`: Updates the data in the selected row with the new values.

- `onDelete`: Deletes a row from the table after confirmation.

- `validate`: Checks for empty input fields and displays validation errors if needed.

4. \*\*Event Handling\*\*:

- Event listeners are used to trigger functions like edit, delete, and form submission based on user actions.

Overall, this code creates an interactive interface where users can input student details, see them displayed in a table, edit existing records, and delete records with confirmation.