This code creates a simple web page that allows the user to add, edit, and delete entries in a table. The page also includes a search function that lets the user search for entries in the table.

When the page loads, an empty array called **data** is created. This array will store the entries in the table.

The code includes several functions to help with adding, editing, and deleting entries:

* **addRow**: This function takes an object representing an entry and adds a new row to the table with the entry's data.
* **clearForm**: This function clears the input fields on the page.
* **validateForm**: This function checks that the input fields are not empty and that the name and email combination does not already exist in the table.
* **addEntry**: This function adds a new entry to the table. It first validates the input fields using the **validateForm** function, and then adds the new entry to the **data** array and the table using the **addRow** function. It also calls the **clearForm** function to clear the input fields.
* **editEntry**: This function updates an existing entry in the table. It first validates the input fields using the **validateForm** function, and then updates the corresponding entry in the **data** array. It then clears the table and repopulates it with the updated entries using the **addRow** function. It also calls the **clearForm** function to clear the input fields.
* **deleteEntry**: This function deletes an existing entry from the table. It finds the corresponding entry in the **data** array and removes it. It then clears the table and repopulates it with the remaining entries using the **addRow** function. It also calls the **clearForm** function to clear the input fields.

The code also includes an event listener for the search button. When the user enters a search query and clicks the button, the **search** function is called. This function loops through all the rows in the table and checks if the search query matches any of the cells in the row. If it finds a match, it shows the row. If it doesn't find a match, it hides the row.

Overall, this code provides a simple example of how to create a dynamic web page using jQuery and JavaScript.

This web page that allows users to perform CRUD (Create, Read, Update, and Delete) operations on a table of data. The page consists of an HTML form that users can fill out to add new data, edit existing data, or delete data. There is also a table that displays the data and allows users to select rows.

The table is created using HTML and styled using CSS. It has three columns: ID, Name, and Email. The table data is initially empty but can be populated with data using the JavaScript code in the qwerty.js file. The JavaScript code defines an array called **data** that holds the table data.

The JavaScript code also defines several functions that perform various operations on the table. The **addRow** function takes a single argument, which is an object that contains data for a single row in the table. The function creates a new row in the table and populates it with the data.

The **clearForm** function clears the form by setting the value of each input field to an empty string.

The **validateForm** function checks whether the form data is valid. It checks whether the ID, Name, and Email fields are empty, and whether the combination of Name and Email already exists in the table.

The **addEntry** function adds a new entry to the table. It first checks whether the form data is valid by calling the **validateForm** function. If the data is valid, it creates an object containing the data for the new row, adds it to the **data** array, adds a new row to the table using the **addRow** function, and clears the form using the **clearForm** function.

The **editEntry** function updates an existing row in the table. It first checks whether the form data is valid by calling the **validateForm** function. If the data is valid, it finds the row in the **data** array that matches the ID in the form, updates its data with the new data, and updates the table row using the **addRow** function.

The JavaScript code also defines event listeners for various user actions. When the user clicks on a row in the table, the row is selected, and the data in the row is displayed in the form. When the user clicks the Reset button, the form is cleared and the selected row is deselected. When the user clicks the Search button, a search function is called to search for rows that match the search criteria.