**User Story: Manage Records Web Application**

**Title: Web Application for Managing Records**

**As a user, I want to be able to manage records (create, read, update, delete) so that I can maintain and modify information efficiently.**

**Main Features**

1. **Create Record**: The ability to add new records.
2. **Read Records**: The ability to view a list of all records.
3. **Update Record**: The ability to modify existing records.
4. **Delete Record**: The ability to remove records.

**Detailed User Stories**

**User Story 1: Create Record**

* **Title**: Add New Record
* **Description**: As a user, I want to be able to add new records to the database so that I can keep track of new information.
* **Acceptance Criteria**:
  + The user can see a form with input fields for "Name" and "Description".
  + The form has a button labeled "Add Record".
  + When the user fills in the fields and clicks "Add Record", the record is added to the database.
  + The new record appears in the list of records without the need to refresh the page.
* **Notes**:
  + Input fields should validate that the "Name" field is not empty.
  + The user should receive feedback if the record was successfully added or if an error occurred.

**User Story 2: Read Records**

* **Title**: View All Records
* **Description**: As a user, I want to view a list of all records so that I can see the existing data.
* **Acceptance Criteria**:
  + The user can see a list of all records in the database.
  + Each record displays the "Name" and "Description" fields.
  + The records are displayed in a user-friendly format.
* **Notes**:
  + The list should be automatically updated when a new record is added or an existing record is modified or deleted.

**User Story 3: Update Record**

* **Title**: Modify Existing Record
* **Description**: As a user, I want to update existing records so that I can correct or change the information.
* **Acceptance Criteria**:
  + Each record in the list has an "Edit" button.
  + When the user clicks the "Edit" button, an update form with pre-filled "Name" and "Description" fields appears.
  + The form has a button labeled "Update Record".
  + When the user modifies the fields and clicks "Update Record", the record is updated in the database.
  + The updated record appears in the list without the need to refresh the page.
* **Notes**:
  + Input fields should validate that the "Name" field is not empty.
  + The user should receive feedback if the record was successfully updated or if an error occurred.

**User Story 4: Delete Record**

* **Title**: Remove Existing Record
* **Description**: As a user, I want to delete records so that I can remove obsolete or incorrect information.
* **Acceptance Criteria**:
  + Each record in the list has a "Delete" button.
  + When the user clicks the "Delete" button, the record is removed from the database.
  + The record is removed from the list without the need to refresh the page.
  + The user is prompted to confirm the deletion before the record is removed.
* **Notes**:
  + The user should receive feedback if the record was successfully deleted or if an error occurred.

**User Interface Requirements**

* **Forms**:
  + Should be clearly labeled and easy to understand.
  + Should validate input and provide meaningful error messages.
  + Should use a modern, clean, and minimalistic design.
* **List of Records**:
  + Should be easy to read and navigate.
  + Should update dynamically to reflect changes without the need to refresh the page.
  + Should provide clear options for editing and deleting records.
* **Feedback**:
  + Should provide clear and immediate feedback for user actions, such as adding, updating, or deleting records.

**Technical Requirements**

* **Backend**:
  + Use Node.js and Express for the server.
  + Use MySQL for the database.
  + Implement RESTful API endpoints for CRUD operations.
* **Frontend**:
  + Use HTML, CSS, and JavaScript for the user interface.
  + Ensure the application is responsive and works well on different devices and screen sizes.
  + Use Fetch API or a similar method to communicate with the backend without reloading the page.
* **Deployment**:
  + Host the application on AWS.
  + Use GitHub for source control.
  + Implement continuous deployment to keep the application up-to-date with the latest changes.

**User Testing**

* **Test Scenarios**:
  + Test adding a new record with valid and invalid inputs.
  + Test viewing the list of records.
  + Test updating an existing record with valid and invalid inputs.
  + Test deleting a record and confirming the deletion.
  + Test responsiveness and usability on different devices and screen sizes.
* **Feedback**:
  + Collect feedback from users on the ease of use and any issues encountered.
  + Use the feedback to make further improvements to the application.