

Step-by-Step Guide for Building an Address Book Application

1. Define Your Goal

Understand the core functionalities of the application:

- Add new contacts.
 - Edit existing contact details.
 - Delete contacts.
 - Search for contacts by name or email.
 - Sort contacts alphabetically by name.
 - Dynamically render the contact list.
-

2. Plan the Structure

Break the application into smaller components:

1. **Data Storage:** Use an array to store contact details.
 2. **Methods:** Define functions for adding, editing, deleting, searching, sorting, and rendering contacts.
 3. **Event Handling:** Handle interactions like form submissions, search inputs, and button clicks.
-

3. Start Writing Code

Begin with the basics and build incrementally:

A. Set Up an Address Book Object

Create a JavaScript object (`addressBook`) to manage contacts and their associated actions:

- Properties:
 - `contacts`: Array to store contact objects.
 - `isEditing`: Boolean to track editing state.
 - `editingEmail`: String to identify the contact being edited.
- Methods:
 - `handleContactForm()`
 - `addContact()`
 - `updateContact()`
 - `deleteContact()`

- `searchContacts()`
- `sortContacts()`
- `renderContacts()`
- `resetForm()`

B. Implement Core Methods

Write the methods in a modular way:

- **`handleContactForm`**: Determine whether to add or update a contact based on the editing state.
- **`addContact`**: Add a new contact with details like name, email, and phone.
- **`updateContact`**: Update the details of an existing contact.
- **`deleteContact`**: Remove a contact from the list based on their email.
- **`searchContacts`**: Filter contacts based on a search query (name or email).
- **`sortContacts`**: Sort the contact list alphabetically by name.
- **`renderContacts`**: Dynamically update the DOM to display the contact list.
- **`resetForm`**: Clear the form and reset editing state.

C. Attach Event Listeners

Handle the interactions:

- **Form Submission**: Use `addEventListener` on the form to trigger `handleContactForm` for adding or updating contacts.
- **Search Input**: Use `addEventListener` on the search input to trigger `searchContacts` and dynamically render the results.
- **Edit and Delete Buttons**: Dynamically generate buttons and attach handlers for editing and deleting contacts.

D. Test and Debug

- Test each method individually in the browser console.
- Validate inputs for edge cases (e.g., empty fields, duplicate emails).

4. Order of Implementation

Follow this sequence:

Initialize the Address Book Object:

```
const addressBook = { contacts: [], isEditing: false, editingEmail: null };
```

1.

2. **Add Core Methods to the Object:**

- `handleContactForm()`
- `addContact()`
- `updateContact()`
- `deleteContact()`
- `searchContacts()`
- `sortContacts()`
- `renderContacts()`
- `resetForm()`

3. **Create Event Handlers:**

- Write the logic for form submission and attach it to the "Submit" button.
- Write the logic for search input and attach it to the search bar.
- Write the logic for edit and delete actions and attach them to dynamically generated buttons.

4. **DOM Manipulation:**

- Use `document.createElement` and `appendChild` to render contacts.
- Dynamically update the DOM for contact list and actions.

5. **Test the Flow:**

- Add contacts.
- Edit and update contacts.
- Delete contacts.
- Search and sort contacts.
- Ensure the DOM updates correctly.

5. Add Features Incrementally

Once the basics work, enhance the application:

- **Advanced Search:** Allow searching by phone number or partial matches.
- **Validation:** Ensure fields are filled correctly and emails are unique before submission.
- **Styling:** Apply CSS classes for better UI and user experience.

6. Checklist for Completion

- Contacts are added correctly with unique IDs.

- Contacts can be edited and updated.
- Contacts can be deleted.
- The contact list can be searched and sorted.
- The contact list renders dynamically and updates after actions.
- Form validation ensures proper input.

By following this roadmap, you will systematically build the Address Book Application with clarity and focus.