Step-by-Step Guide for Building a Social Media Post Management System

1. Define Your Goal

Understand the core functionalities of the application:

- Add new posts.
- Edit existing posts.
- Like posts.
- Add comments to posts.
- Clone posts.
- Render all posts dynamically.

---

2. Plan the Structure

Break the application into smaller components:

1. **Data Storage**: Maintain a structure to store posts (e.g., an array of post objects).
2. **Methods**: Define functions for adding, editing, liking, commenting, cloning, and rendering posts.
3. **Event Handling**: Handle user interactions like form submissions, button clicks, and prompts.

---

3. Start Writing Code

Begin with the basics and build incrementally:

A. Set Up a Social Media Object

Create a JavaScript object (`socialMediaApp`) to manage posts and their associated actions:

- Properties: `posts` to store all posts.
- Methods:
    - `addPost()`
    - `editPost()`
    - `likePost()`
    - `addComment()`
    - `clonePost()`
    - `renderPosts()`

B. Implement Core Methods

Write the methods in a modular way:

- `addPost`: Add a new post with content, likes, and comments.
- `editPost`: Update the content of a specific post by its ID.
- `likePost`: Increment the likes for a specific post.
- `addComment`: Add a comment to a specific post.
- `clonePost`: Create a new post with the same content as an existing one but with a new ID and timestamp.
- `renderPosts`: Dynamically update the DOM to display all posts.

C. Attach Event Listeners

Handle the interactions:

- **Form Submission**: Use `addEventListener` on the form to call `addPost` and re-render the posts.
- **Like, Edit, and Clone Buttons**: Dynamically generate buttons and attach event handlers for these actions.

D. Test and Debug

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

---

4. Order of Implementation

Follow this sequence:

Initialize the Social Media Object:

```
const socialMediaApp = { posts: [] };
```

1.
2. **Add Core Methods to the Object**:
    - `addPost()`
    - `editPost()`
    - `likePost()`
    - `addComment()`
    - `clonePost()`
    - `renderPosts()`

3. **Create Event Handlers**:
    - Write `addPost` and attach it to the form's `submit` event.
    - Write `editPost`, `likePost`, and `clonePost` functionalities, and attach them to buttons dynamically generated for each post.
4. **DOM Manipulation**:
    - Use `document.createElement` and `appendChild` to render posts.
    - Update the content dynamically for actions like editing, liking, and cloning posts.
5. **Test the Flow**:
    - Add, edit, like, and clone posts.
    - Ensure the DOM updates correctly.

---

5. Add Features Incrementally

Once the basics work, enhance the application:

- **Comments**: Add a comment input field and display comments dynamically for each post.
- **Validation**: Ensure post content is not empty before adding or editing.
- **Styling**: Apply CSS classes for better UI and user experience.

---

6. Checklist for Completion

- Posts are added correctly with unique IDs.
- Posts can be edited.
- Likes increment correctly.
- Posts can be cloned with new IDs and timestamps.
- The application handles empty inputs gracefully.
- Posts render dynamically, including updates for all actions.

By following this roadmap, you will systematically build the Social Media Post Management System with clarity and focus.