Address Book Application - Student Guide

Setup Instructions

1. Create a new React application:

```
npx create-react-app address-book
cd address-book
```

2. Add Bootstrap CSS to your `public/index.html` file. Add this line in the `<head>` section:

```
<link rel="stylesheet" href="https://bootswatch.com/5/cosmo/bootstrap.min.css">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.3/font/bootstrap-icons.min.css">
```

Step 1: Create the Contact Context

1. Create a new folder `src/context`
2. Create a new file `src/context/ContactContext.js`
3. Copy this initial code:

```
import React, { createContext, useContext, useState } from 'react';

// Create context
const ContactContext = createContext(undefined);

// Create provider component
export const ContactProvider = ({ children }) => {
  const [contacts, setContacts] = useState([]);

  return (
    <ContactContext.Provider value={{ contacts }}>
      {children}
    </ContactContext.Provider>
  );
};

// Create custom hook
export const useContacts = () => {
  const context = useContext(ContactContext);
  if (context === undefined) {
    throw new Error('useContacts must be used within a ContactProvider');
  }
  return context;
};
```

Step 2: Add Contact Management Functions

1. Update `ContactContext.js` to add contact management functions. Add this inside the ContactProvider before the return statement:

```
  const addContact = (contact) => {
    const newContact = {
      ...contact,
      id: Date.now().toString(),
    };
    setContacts(prevContacts => [...prevContacts, newContact]);
  };

  const deleteContact = (id) => {
    setContacts(prevContacts => prevContacts.filter(contact => contact.id !==
id));
  };
```

2. Update the Context.Provider value to include these functions:

```
  return (
    <ContactContext.Provider value={{ contacts, addContact, deleteContact }}>
      {children}
    </ContactContext.Provider>
  );
```

Step 3: Create the Add Contact Form

1. Create a new folder `src/components`
2. Create a new file `src/components/AddContact.js`
3. Copy this code:

```
import React, { useState } from 'react';
import { useContacts } from '../context/ContactContext';

const AddContact = () => {
  const { addContact } = useContacts();
  const [formData, setFormData] = useState({
    name: '',
    email: '',
    phone: '',
  });

  const handleSubmit = (e) => {
    e.preventDefault();
    addContact(formData);
```

```
    setFormData({ name: '', email: '', phone: '' });
  };

  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData(prev => ({
      ...prev,
      [name]: value
    }));
  };

  return (
    <div className="card mb-4">
      <div className="card-body">
        <h3 className="card-title mb-4">Add New Contact</h3>
        <form onSubmit={handleSubmit}>
          <div className="mb-3">
            <label htmlFor="name" className="form-label">Name</label>
            <input
              type="text"
              className="form-control"
              id="name"
              name="name"
              value={formData.name}
              onChange={handleChange}
              required
            />
          </div>
          <div className="mb-3">
            <label htmlFor="email" className="form-label">Email</label>
            <input
              type="email"
              className="form-control"
              id="email"
              name="email"
              value={formData.email}
              onChange={handleChange}
              required
            />
          </div>
          <div className="mb-3">
            <label htmlFor="phone" className="form-label">Phone</label>
            <input
              type="tel"
              className="form-control"
              id="phone"
              name="phone"
              value={formData.phone}
              onChange={handleChange}
              required
            />
          </div>
          <button type="submit" className="btn btn-primary">Add Contact</button>
        </form>
```

```
      </div>
    </div>
  );
};

export default AddContact;
```

Step 4: Create the Contact Card Component

1. Create a new file `src/components/ContactCard.js`
2. Copy this code:

```
import React from 'react';
import { useContacts } from '../context/ContactContext';

const ContactCard = ({ contact }) => {
  const { deleteContact } = useContacts();

  return (
    <div className="card mb-3">
      <div className="card-body">
        <div className="d-flex justify-content-between align-items-center">
          <h5 className="card-title">{contact.name}</h5>
          <button
            className="btn btn-link text-danger"
            onClick={() => deleteContact(contact.id)}
          >
            <i className="bi bi-trash"></i>
          </button>
        </div>
        <p className="card-text mb-1">
          <strong>Email:</strong> {contact.email}
        </p>
        <p className="card-text">
          <strong>Phone:</strong> {contact.phone}
        </p>
      </div>
    </div>
  );
};

export default ContactCard;
```

Step 5: Create the Contact List Component

1. Create a new file `src/components/ContactList.js`
2. Copy this code:

```
import React from 'react';
import { useContacts } from '../context/ContactContext';
import ContactCard from './ContactCard';

const ContactList = () => {
  const { contacts } = useContacts();

  if (contacts.length === 0) {
    return (
      <div className="alert alert-info" role="alert">
        No contacts found. Add some contacts to get started!
      </div>
    );
  }

  return (
    <div>
      <h3 className="mb-4">Contacts</h3>
      {contacts.map((contact) => (
        <ContactCard
          key={contact.id}
          contact={contact}
        />
      ))}
    </div>
  );
};

export default ContactList;
```

Step 6: Update App.js

1. Replace the contents of `src/App.js` with:

```
import React from 'react';
import { ContactProvider } from './context/ContactContext';
import AddContact from './components/AddContact';
import ContactList from './components/ContactList';

function App() {
  return (
    <ContactProvider>
      <div className="container py-5">
        <h1 className="text-center mb-5">Address Book</h1>
        <div className="row">
          <div className="col-md-4">
            <AddContact />
          </div>
          <div className="col-md-8">
            <ContactList />
```

```
          </div>
        </div>
      </div>
    </ContactProvider>
  );
}

export default App;
```

Optional Step 7: Add Default Contacts

If you want to start with some sample contacts, update the useState in ContactProvider:

```
// Add this after the ContactContext creation:
const defaultContacts = [
  {
    id: '1',
    name: 'John Doe',
    email: 'john.doe@example.com',
    phone: '555-123-4567'
  },
  {
    id: '2',
    name: 'Jane Smith',
    email: 'jane.smith@example.com',
    phone: '555-987-6543'
  }
];

// Then update the useState line in ContactProvider:
const [contacts, setContacts] = useState(defaultContacts);
```

Testing the Application

After completing each step:

1. Save all files
2. Run `npm start` if you haven't already
3. Test the functionality:
    - Add a new contact
    - View the contact in the list
    - Delete a contact
    - Verify the empty state message

Learning Objectives

This project teaches:

1. React Context API for state management

2. React Hooks (useState, useContext)
3. Component composition
4. Form handling in React
5. Bootstrap styling
6. JavaScript array operations
7. Event handling in React