

Product Lead (Role Card)

Read README.md aloud and ensure the team aligns on acceptance criteria:

- Name ≥ 2 , Email valid, Phone 10–13 digits (numeric), Email unique
- getContacts() sorted by name ASC
- searchContacts() case-insensitive, search name/email
- removeContact(email) returns true/false
- Persist in localStorage under key `contacts_v1`

Keep conversations on-track and time-box decisions.

QA (Role Card)

Open tests.js and explain **exact** expectations:

- Function names: addContact, getContacts, searchContacts, removeContact
- Storage key: contacts_v1
- Duplicates rejected with a clear error message
- Case-insensitive search across name and email

Do NOT modify tests.js during the challenge.

API Developer (Role Card)

Open `api.js`. Implement validations and persistence:

- Normalize inputs (trim, lower-case email)
- Enforce unique email
- Phone: digits only, length 10–13
- Sort in `getContacts` by name ASC
- Use `localStorage` key ``contacts_v1``

Return `{ok:true}` or `{ok:false, error:'...'}` from `addContact`.

UI Developer (Role Card)

Open index.html and app.js:

- Wire the form to call `api.addContact({name,email,phone})`
- Render list using `api.getContacts()`
- Remove via `api.removeContact(email)`
- Live search uses `api.searchContacts(query)`
- Keep UI minimal but usable; avoid inline alerts if you prefer inline errors.