

Frontend R&D Intern Assignment

Dynamic Interface Compiler

Problem Statement

You are building a **no-code-like interface builder** for internal tools at a startup. The goal is to allow users to **design UI blocks** (like forms, tables, text sections) dynamically — and **compile** those definitions into actual working React components — **without refreshing the page**.

The assignment evaluates:

- Component rendering based on schema
- Dynamic code interpretation (using JSON or sandboxed logic)
- State orchestration across dynamic layouts
- Research ability (e.g., `eval`, dynamic imports, component factories)

Requirements

Create a mini app that allows:

◆ 1. UI Block Definition Panel

- JSON editor or form-based schema builder
- Let users define components like:

```
json
CopyEdit
{
  "type": "form",
  "fields": [
    { "label": "Email", "type": "email", "required": true },
    { "label": "Age", "type": "number", "min": 18 }
```

```
],  
  "submitText": "Register"  
}
```

- Store definitions in local state or localStorage

◆ 2. Live Renderer

- Parse the above JSON into actual React components
- Render the UI as defined (forms, headings, buttons, etc.)
- Maintain internal state, validation, form submission, etc.
- Form submission should log the values in console (or render a success message)

◆ 3. Composable Component Library

- Support these 3 block types:
 - `form`
 - `text`
 - `image`

Each of these should be defined through schema and interpreted dynamically.

◆ 4. Logic Injection (Advanced)

- Allow users to attach simple logic blocks like:

```
json  
CopyEdit  
"onSubmit": "if (values.age < 21) return 'Too young';"
```

- Execute that logic on form submit using sandboxed `Function` (not raw `eval`)
- Show error messages dynamically from user-defined logic

Bonus Points

- Live drag-and-drop layout (like `dnd-kit`)
 - Visual JSON editor (Monaco or CodeMirror)
 - Export schema as shareable JSON
-

Tech Stack (Flexible)

- React or Next.js
 - TailwindCSS or any styling framework
 - JSON-based schema design
 - React Hook Form or custom validator
 - Optional: `monaco-editor` , `dnd-kit` , `zustand`
-

Deliverables

- GitHub repo with README
 - Hosted demo
 - List challenges faced and what you would optimize
-

Estimated Time

16–24 hours.

Evaluation Criteria

- Problem-solving and research ability
- Architecture and reusability of components
- Dynamic rendering and state handling
- Clean code, component boundaries
- UX thoughtfulness