



Associate Software Developer Assignment - upliance.ai

About Us:

upliance.ai has built India's first AI cooking assistant. Designed for beginners, the upliance simplifies and automates all parts of cooking for yourself. We're on a mission to push the boundaries of what home appliances can do for young India. As a disruptive startup, we take pride in our unique approach, leveraging the power of AI to create hardware products that blend innovation with everyday utility. Directly collaborating closely with industry leaders like OpenAI and building fast.

Please find details of the assignment here:

React + Redux Form Builder Assignment

1. Objective

Build a dynamic form builder using **React, Typescript, MUI, and localStorage**. Users should be able to:

1. Create dynamic forms with customizable fields and validations
2. Preview how a form behaves for the end user
3. View a list of saved forms
4. Persist all form configurations in **localStorage** (No backend required)

x____x

2. Routes

Route	Description
-------	-------------

/create	Build a new form by dynamically adding and configuring fields
----------------	---

/preview	Interact with the currently built form like an end user
-----------------	---

/myforms	View all previously saved forms from localStorage
-----------------	---

x____x



3. Form Builder (/create)

Users should be able to:

1. Add new fields of the following types:

- a. Text
- b. Number
- c. Textarea
- d. Select
- e. Radio
- f. Checkbox
- g. Date

2. Configure each field with:

- a. Label
- b. Required toggle
- c. Default value
- d. Validation rules including:
 - i. Not empty
 - ii. Minimum/maximum length
 - iii. Email format
 - iv. Custom password rule (e.g., minimum 8 characters, must contain a number)

3. Mark fields as **Derived Fields**:

- a. Derived field computes its value based on one or more parent fields
- b. Users must:
 - i. Select parent field(s)
 - ii. Define a formula or logic for computation
- c. *Example*: Age derived from Date of Birth

4. Reorder or delete fields

5. Save the form:

- a. Prompt user for a form name
- b. Store the complete form schema in localStorage
- c. ⚠️ Only save schema/configuration — storage of user input values is not required.



4. Preview (/preview)

1. Render the entire form as it would appear to an end user
2. All fields must:
 - a. Support user input
 - b. Validate input based on configured rules
 - c. Show validation error messages appropriately
3. Derived fields should auto-update as their parent fields change

x____x

5. My Forms (/myforms)

1. Display a list of all saved forms from localStorage
2. Each form entry should show:
 - a. Form name
 - b. Date of creation
3. Clicking a form should open its /preview page

x____x

6. Evaluation Criteria

1. Code should be clean, modular, and easy to extend
2. predictable and organized state management
3. Handle edge cases and validation errors gracefully
4. TypeScript should be used effectively to enforce strong type safety and catch potential issues during development.
5. Preview must reflect validations and derived fields accurately
6. UI should be intuitive to use

x____x

7. Submission

- A live deployed link (e.g., via Vercel, Netlify, etc.)
- A public GitHub repository
- Note: No authentication is required

x____x



Deadline: You have **2 days/48 hours** to submit the assignment.

Submission Format: Please submit using the provided **google form link**.

Feel free to connect if you have any queries regarding the assignment.