



Product Requirements & Specification Document

Project Name

Formify - Dynamic Form Builder

Description

Formify is a web application enabling users to create, preview, and manage dynamic forms. Built with React, JavaScript, and Tailwind CSS, it demonstrates modern form handling and HTML elements. Target users include startups and educational institutions seeking rapid form prototyping.

1. Goals & Objectives

Goal	Description
Dynamic Form Creation	Allow users to build forms with various field types
Real-time Preview	Instantly preview forms as they are built
Form Management	Save, edit, and delete created forms
Modern UI/UX	Clean, responsive interface using Tailwind CSS
Demonstrate Modern HTML Elements	Support latest HTML5 form controls

2. Core Features

Feature	Description
Form Builder	Drag-and-drop or add/remove fields; reorder fields
Field Types	Text, Email, Number, Date, Checkbox, Radio, Select, Textarea
Field Properties	Label, Placeholder, Required, Default Value, Options (where applicable)
Live Preview	Real-time rendering of the form as fields are added/edited
Form Management	Save, load, edit, and delete forms (local storage)
Responsive Design	Mobile-friendly layout using Tailwind CSS
Validation	Basic client-side validation for required fields and types

3. User Stories

As a...	I want to...	So that...
User	Add/remove/reorder form fields	I can customize my form
User	Preview the form in real-time	I can see changes instantly



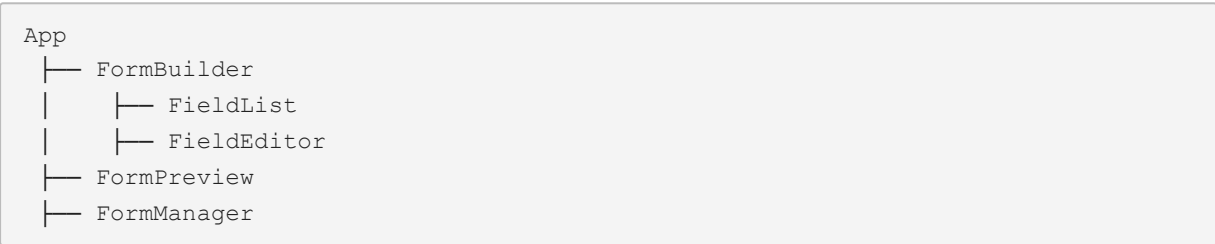
User	Save and manage multiple forms	I can reuse or edit forms later
User	Use modern form elements	My forms are up-to-date and accessible

4. Technical Specifications

4.1 Architecture

- **Frontend:** React (functional components, hooks)
- **Styling:** Tailwind CSS
- **State Management:** React useState/useReducer
- **Persistence:** Browser localStorage

4.2 Component Structure



4.3 Data Model (Pseudocode)

```
Form = {
  id: string,
  name: string,
  fields: [
    {
      id: string,
      type: 'text' | 'email' | 'number' | 'date' | 'checkbox' | 'radio' | 'select',
      label: string,
      required: boolean,
      placeholder?: string,
      defaultValue?: string,
      options?: [string], // for select, radio
    },
    ...
  ]
}
```

5. UI/UX Requirements

Requirement	Specification
Responsive Layout	Mobile, tablet, and desktop support
Intuitive Controls	Clear buttons/icons for add, edit, delete, reorder
Theming	Light, modern, startup/education-friendly palette



Accessibility	Keyboard navigation, ARIA labels
---------------	----------------------------------

6. Non-Functional Requirements

Requirement	Specification
Performance	Instant feedback, minimal load times
Security	No external data storage; local only
Browser Support	Latest Chrome, Firefox, Edge, Safari
Code Quality	ESLint, Prettier, clear documentation

7. Out of Scope

- No backend or user authentication
- No advanced conditional logic (e.g., field dependencies)
- No third-party form integrations

8. Acceptance Criteria

ID	Criteria
AC1	User can add, edit, remove, and reorder fields in a form
AC2	User sees real-time preview of the form
AC3	User can save, load, edit, and delete forms from local storage
AC4	All supported field types and properties are available
AC5	UI is responsive and accessible

9. Milestones

Milestone	Description	Target Date
UI Skeleton	Basic layout and navigation	Week 1
Form Builder	Add/edit/remove/reorder fields	Week 2
Live Preview	Real-time form rendering	Week 2
Form Management	Save/load/delete forms	Week 3
Styling & Accessibility	Tailwind, responsive, a11y	Week 3
Testing & QA	Manual and automated tests	Week 4

10. Risks & Mitigations



Risk	Mitigation
Browser compatibility	Test on all major browsers
Data loss (localStorage)	Warn users before clearing forms
UI complexity	Keep interface minimal and focused

11. References

- [React Documentation](#)
- [Tailwind CSS Documentation](#)
- [MDN HTML Forms Guide](#)

End of Document