Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the implementation details for **CloudOps - Technology Resource Manager UI**. The project is a React-based frontend for managing cloud technology resources, featuring dashboards, resource cards, and interactive forms. The design emphasizes a modern, responsive, and futuristic user experience using React, JavaScript, HTML, CSS, and Tailwind.

---

1. System Components

Component	Description	Key Responsibilities
Dashboard	Main overview of resources	Display resource summary, navigation
ResourceCard	Visual card for each resource	Show resource info, actions
ResourceForm	Form for creating/editing resources	Input validation, submit data
ResourceList	Lists all resources	Fetch, filter, and display resources
APIService	Handles API communication	CRUD operations, error handling
ThemeProvider	Manages theme and styling	Apply Tailwind/futuristic styles

---

2. Class/Interface Overview

Component/Class	Key Props/State/Methods	Relationships
`Dashboard`	`resources`, `onAddResource`	Parent of ResourceList, ResourceForm
`ResourceCard`	`resource`, `onEdit`, `onDelete`	Child of ResourceList
`ResourceForm`	`resource`, `onSubmit`, `onCancel`	Used by Dashboard
`ResourceList`	`resources`, `onEdit`, `onDelete`	Child of Dashboard
`APIService`	`getResources()`, `createResource()`, etc.	Used by all components

Key Methods Example:

```
// APIService.js
getResources(): Promise<Resource[]>
createResource(data: Resource): Promise<Resource>
updateResource(id: string, data: Resource): Promise<Resource>
deleteResource(id: string): Promise<void>
```

---

3. Data Structure Overview

Model	Fields
Resource	id: string, name: string, type: string, status: string, metadata: object

Example:

```
type Resource = {
  id: string,
  name: string,
  type: string,
  status: string,
  metadata: object
}
```

---

4. Algorithms / Logic

Resource Fetch & Display Flow:

```
On Dashboard Mount:
  Call APIService.getResources()
  If success:
    Set resources state
  If error:
    Show error notification

On ResourceForm Submit:
  Validate input
  If valid:
    Call APIService.createResource()
    Refresh resource list
  Else:
    Show validation errors
```

---

5. Error Handling

Scenario	Handling Approach
API failure (fetch/create/etc.)	Show user-friendly error message
Form validation error	Highlight invalid fields, show message
Network error	Display retry option
Unauthorized/Forbidden	Redirect to login or show access denied

---

End of Document