**Resource Management Application – Development Document**

**📌 1. Overview**

The **Resource Management Application (RMA)** is a web-based system designed to manage the entire lifecycle of project resource allocation using SoWs (Statements of Work). It supports transparent project planning, resource deployment, fulfillment tracking, governance, and bench management.

**🔍 Key Goals**

* Enable role-based access for controlled management.
* Track resource demand vs. availability in real-time.
* Ensure alignment with project goals, budgets, and SoWs.
* Improve decision-making through dashboards and reports.
* Minimize revenue leakage via proactive bench and risk management.

**🧱 2. Technologies Used**

**🖥️ Frontend**

* **React JS**: For building dynamic UI with routing and component-based architecture.
* **React Router**: For SPA (Single Page Application) navigation.
* **Axios**: To communicate with the backend APIs.
* **TailwindCSS / Bootstrap**: For quick and responsive UI styling.
* **Formik / React Hook Form**: For managing form data and validations.

**⚙️ Backend**

* **Java (Spring Boot)**: For building RESTful APIs with dependency injection and modular architecture.
* **Spring Data JPA / Hibernate**: For ORM and interaction with PostgreSQL.
* **Spring Security**: For implementing RBAC (Role-Based Access Control).

**🗃️ Database**

* **PostgreSQL**: Relational database for storing entities like resources, projects, SoWs, and master data.

**🖼️ 3. UI Pages and Their Functionalities**

Each page is implemented as a separate React route/component. Components interact with the backend using Axios and are styled using Tailwind or Bootstrap.

**🛠️ A. Master Data Management Pages**

These pages are primarily used by **Super Admin** or **Resource Management Team**.

| **Page** | **Route** | **Functionalities** |
| --- | --- | --- |
| **Location Master** | /masters/locations | Add/Edit/Delete/View office locations |
| **Title Master** | /masters/titles | Manage job titles/roles (e.g., Developer, Analyst) |
| **Skillset Master** | /masters/skills | Maintain skills library mapped to resources |
| **Position Type Master** | /masters/positions | Define position types (Contract, Full-Time, etc.) |
| **Shift Timings Master** | /masters/shifts | Define work shift names and times |
| **Allocation % Master** | /masters/allocation-percentages | Set values like 25%, 50%, 100% |
| **Interview Panel Master** | /masters/interview-panel | Maintain L1, L2, L3 panel members |
| **Practice/IT Verticals Master** | /masters/practices | Maintain verticals like Data, Infra, Dev |
| **Project Status Master** | /masters/status | Define project status options |
| **Resource Master** | /masters/resources | Bulk/individual resource entry, assign skills, title, etc. |

**🔁 B. Transaction Pages**

These represent core workflows and are primarily used by RMT, Delivery Managers, and Practice Heads.

| **Page** | **Route** | **Functionalities** |
| --- | --- | --- |
| **SoW Upload & Project Creation** | /sow-upload | Upload SoW document, define project, required roles |
| **Resource Allocation** | /resource-allocation | Allocate bench resources, define % and timelines |
| **Fulfillment Tracking** | /fulfillment | Track open demands, add notes, expected closure |
| **Resource Release Request** | /release-request | Submit and approve release requests with reasons |
| **Additional Resource Request** | /additional-request | Request new resources for in-flight projects |
| **Project Extension Request** | /extension-request | Update SoW, extend project timeline |
| **Project Status Update** | /status-update | Log milestone, progress %, risks/issues |
| **Lessons Learned** | /lessons-learned | Capture post-project insights, searchable by keyword/project |

**📊 C. Reporting and Dashboards**

Used by all stakeholders for tracking, governance, and analytics.

| **Page** | **Route** | **Functionalities** |
| --- | --- | --- |
| **In-Flight Projects Report** | /reports/inflight | View all ongoing projects, status summary |
| **Proposed Projects Report** | /reports/proposed | View projects in planning stage |
| **Spend Tracking Report** | /reports/spend | Budgeted vs. actual spend by project |
| **Project Status Report** | /reports/project-status | View milestones, deliverables, delays |
| **Risks & Issues Report** | /reports/risks | Open issues, mitigation steps, status |
| **Resource Allocation Report** | /reports/resource-allocation | Filter by skill, practice, utilization |
| **Bench Tracking Report** | /reports/bench | View unallocated or underutilized resources |
| **Forecasting Report** | /reports/forecast | Future demand vs. resource availability |
| **Financial Metrics Report** | /reports/financials | Budget burn, revenue recognition |
| **Governance Report** | /reports/governance | Approvals, SoW compliance, audit logs |
| **Portfolio Dashboard** | /dashboard | Portfolio health by region, practice |
| **Notifications & Alerts** | /notifications | Alerts for tasks, deadlines, budgets |
| **Change Requests Report** | /reports/change-requests | Track change requests and approvals |
| **Lessons Learned Repository** | /reports/lessons | Searchable post-project insights |

**🧑‍💼 4. User Roles and Permissions (RBAC)**

| **Role** | **Access Level** |
| --- | --- |
| **Super Admin** | Full access, user management, master setup |
| **Resource Management Team (RMT)** | Create/Update SoWs, allocate resources |
| **Project Managers/Delivery Managers** | Request/add resources, update status |
| **Finance Controllers** | View-only access to financial data |
| **Practice Heads/IT Leaders** | Approve requests, view dashboards |

RBAC is enforced both:

* On frontend (UI rendering, route access)
* On backend (Spring Security + role-based authorization)

**✅ 5. Key Features Across Pages**

* **Audit Logging** (on all forms):  
  Created By, Created Date, Updated By, Updated Date
* **Form Validations** (frontend + backend):  
  Mandatory fields like project name, skill, SoW, dates, etc.
* **Search & Filter Options**:  
  Most listing pages include filters (date range, skill, title, region)
* **Approval Workflows**:  
  Integrated into requests like release, extension, fulfillment