

A **detailed requirements document** for developing a **Resume Builder App as a Proof of Concept (PoC)** in **Mendix**. This will include the **core functionalities, user roles, data model, and integration aspects**, tailored for rapid app development and showcasing Mendix capabilities.

Resume Builder App – PoC Requirements (Mendix)

Objective

Develop a **Resume Builder Application** as a Proof of Concept using Mendix that allows users to create, manage, and download professional resumes. The app should demonstrate **rapid low-code development, reusability, multi-step form handling, and PDF export capabilities**.

User Roles

1. **Guest User**
 - Can register and log in.
 - Can view landing page & features.
2. **Registered User**
 - Can create/edit multiple resumes.
 - Can preview/download resumes.
 - Can manage personal details, education, experience, etc.
3. **Admin**
 - Can manage user data (optional).
 - Can view analytics (optional).

Core Features

1. User Registration and Authentication

- Login/Signup using username, email, and password.
- Basic validation (email format, password strength).
- Forgot password.

2. Dashboard

- View list of created resumes.
- Buttons to **Create New Resume, Edit, Preview, Delete, and Download**.

3. Resume Builder – Multi-Step Form

Split into manageable steps:

Step 1: Personal Information

- Name
- Profile photo (optional)
- Email, Phone
- LinkedIn, GitHub, Website

Step 2: Career Objective or Summary

- Short paragraph about the user.

Step 3: Education

- Degree, Institution, Start/End Date
- GPA/Percentage
- Add multiple entries

Step 4: Work Experience

- Company, Job Title, Duration
- Responsibilities (rich text or bulleted list)
- Add multiple entries

Step 5: Skills

- List of skills (add/remove dynamic list)
- Optionally categorize (e.g., Technical, Soft Skills)

Step 6: Certifications / Awards (Optional)

- Title, Issuer, Date, Description
- Add multiple entries

Step 7: Projects (Optional)

- Title, Description, Technologies Used
- Link to GitHub or Demo (if any)
- Add multiple entries

Step 8: Template Selection & Preview

- Select from 2–3 basic templates (different layouts/styles).
- Live preview functionality.

📄 Output & Download

- Generate a **PDF** version of the selected resume template.
- Use Mendix PDF Document Generation module or integrate with 3rd party service if needed.
- Optionally allow users to email it to themselves.

☛ Data Model (Entities)

User

- Name, Email, Password (use Mendix default account entity)

Resume

- Title
- Owner (reference to User)
- Template (selected template style)

PersonalDetails, Education, Experience, Skills, Certifications, Projects

- Each as separate entities with one-to-many relation to Resume.

☐ Integrations

- **PDF Export Module** – Mendix built-in or third-party.
- **Image Upload** – For profile photo (store as file/document).
- (Optional) **Cloud storage or Email service** for resume delivery.

🌀 UI/UX Considerations

- Responsive UI (mobile/tablet/web)
- Clear navigation between steps
- Use Mendix **Wizard** or **Page Template** for step forms
- Inline validation
- Preview resume as user fills it

✓ MVP Checklist for PoC

Feature	Priority
User Auth	✓
Resume Create/Edit	✓
Multi-Step Form	✓
PDF Download	✓
Template Preview	✓
Resume List Dashboard	✓

✂ Tech & Tools (Mendix Specific)

- **Mendix Studio Pro** (9.x or 10.x)
- **Atlas UI** for styling
- **Nanoflows/Microflows** for logic
- **Custom Widgets** (optional for dynamic preview)
- **PDF Document Generation** module from Marketplace
- **App Security**: Role-based access

PoC Goals

- Showcase Mendix's ability to build modular, form-driven apps.
- Demonstrate dynamic data binding and multi-entity relationships.
- Export high-quality, styled documents (PDF).
- Simple UI with fast development and scalability potential.