# Resume Generator Web Application

Service Charter Documentation

## 1. Purpose

To provide users with an intuitive and efficient platform to create professional, visually appealing resumes tailored to various job roles, enhancing their chances of securing employment by simplifying the resume creation process.

## 2.Objective

- Fast and effortless
- Visually professional
- Customizable and AI-enhanced
- Exportable in multiple formats

## 3. Demand / Opportunity

Resume creation remains a time-consuming task for job seekers. This project aims to fill that gap, helping users build resumes that align with current hiring standards.

## 4.Business Requirements

User Login System: Users log in securely to access their saved resumes and preferences.

Resume Form Input: Users fill out personal details, education, experience, skills, etc.

Template Selection: Users choose from multiple layout templates to customize resume appearance.

AI Content Suggestions: Users receive intelligent suggestions for improving grammar, structure, and keywords.

Resume Export: Users export resumes in PDF/DOCX formats.

## 5.Technical Requirements

Login & Session: Implemented via JWT or OAuth with hashed passwords and secure sessions.

Resume Form Processing: Backend APIs handle data storage and processing with validation.

Template Rendering: Templates styled with CSS and selected dynamically through the UI.

AI Integration: NLP-based microservices for suggestions and keyword enhancement.

Export Functionality: PDF/DOCX export via Puppeteer or js PDF, with export logs maintained.

## 6.Technological Requirements

- Frontend: React, Tailwind CSS
- Backend: Node.js + Express
- Database: PostgreSQL
- Auth: JWT / OAuth
- Export Tools: Puppeteer / js PDF
- Hosting: Vercel, Heroku, AWS
- Dev Tools: GitHub, VS Code, Postman

## 7.Team Members

- Vaishnavi Kedar
- Utkarsh Shirbhate
- Sayuj Singh
- Tejashri Darade

## 8.Stakeholders

- Job Seekers / Candidates
- Application Owners / Product Owners
- Developers / Software Engineers

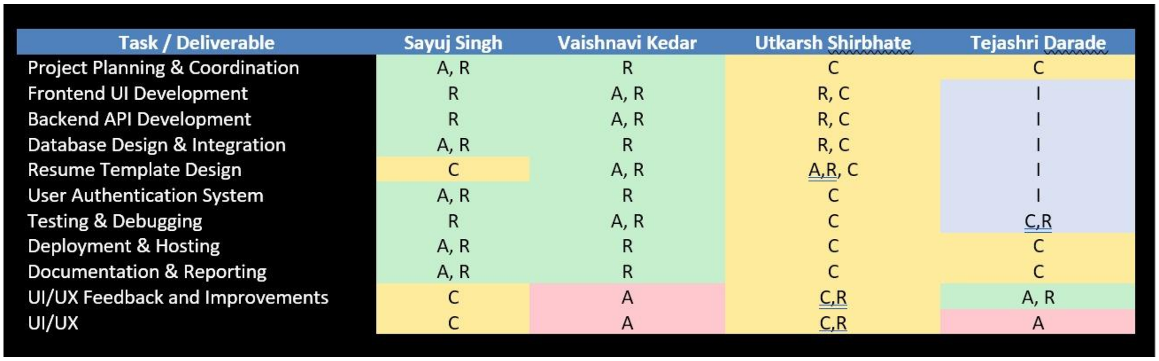
## 9.Resources Needed

- Technical documentation for frameworks and tools
- Cloud deployment access
- Testing utilities (Postman, BrowserStack)
- Human resource availability as per RACI chart

## 10.Risk Analysis (PESTLE-based)

- Political: Cloud service access may be restricted in certain regions
- Economic: Limited budget for premium AI APIs or cloud plans
- Social: Users may not be familiar with AI-assisted features
- Technological: Export failures, integration bugs
- Legal: Data privacy concerns (GDPR compliance)
- Environmental: N/A

## 11.RACI Matrix

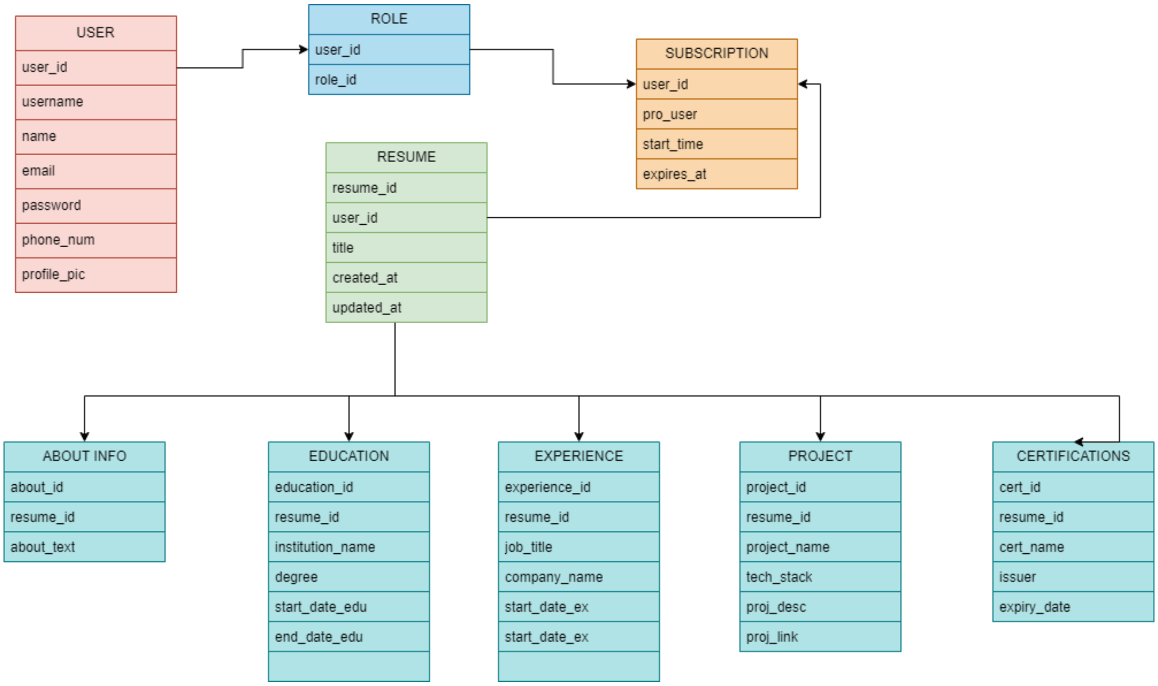


| Task / Deliverable | Sayuj Singh | Vaishnavi Kedar | Utkarsh Shirbhate | Tejashri Darade |
|---|---|---|---|---|
| Project Planning & Coordination | A, R | R | C | C |
| Frontend UI Development | R | A, R | R, C | I |
| Backend API Development | R | A, R | R, C | I |
| Database Design & Integration | A, R | R | R, C | I |
| Resume Template Design | C | A, R | A,R, C | I |
| User Authentication System | A, R | R | C | I |
| Testing & Debugging | R | A, R | C | C,R |
| Deployment & Hosting | A, R | R | C | C |
| Documentation & Reporting | A, R | R | C | C |
| UI/UX Feedback and Improvements | C | A | C,R | A, R |
| UI/UX | C | A | C,R | A |

## 12.Timeline / Milestone

| Milestone | Date |
|---|---|
| Requirement Analysis & Planning | 01 June 2025 |
| Frontend Development | 08 June 2025 |
| Backend Development | 11 June 2025 |
| Testing | 15 June 2025 |
| Deployment | 18 June 2025 |
| Project Completion | 20 June 2025 |

## 13.System Design -ERD



## 14.Tech Stack And Architecture

### ◇ Frontend

- **Framework:** React
- **Styling:** Tailwind CSS
- **Responsiveness:** Mobile-first design with flexible resume layouts

### ◇ Backend

- **Runtime:** Node.js
- **Framework:** Express.js
- **API Services:** RESTful APIs for resume handling, AI suggestions, user management

### ◇ Database

- **Type:** PostgreSQL
- **Entities:**
  - User
  - Resume
  - Template
  - ResumeSection
  - SectionItem
  - Feedback
  - ResumeExportLog
- **Relationships:** Implemented using foreign keys

### ◇ Authentication (Auth)

- **Methods:** JWT / OAuth
- **Security Features:**
  - Password hashing with bcrypt
  - CSRF protection
  - Role-based access control (Admin/User)
  - Secure session/token management

### ◇ Hosting

- Options:
  - **Frontend:** Vercel
  - **Backend:** Heroku or AWS EC2
  - **Database:** MongoDB Atlas or AWS RDS
  - **Resume Export Service:** Puppeteer or js PDF running in isolated containers

### AI Features

- Resume content suggestions using NLP
- Auto-extraction of data from uploaded resumes (e.g., PDFs)
- Grammar and tone improvement
- Keyword optimization for job matching
- AI-based feedback on resume sections

## 15.Testing And QA strategy

### Unit Testing

- API endpoint testing (e.g., resume creation, export, feedback submission)
- Individual modules: AI suggestion engine, template rendering, authentication

### Integration Testing

- Full resume creation flow (user → form → AI suggestions → export)
- Database integration and validation
- AI-feedback loop

### Manual Testing

- End-to-end UI testing for layout responsiveness
- Usability testing (template selection, preview, export)
- Error handling and feedback clarity
- Tested against multiple browsers and devices