

# Executive Summary & Product Vision

---

## 1. Executive Summary

The Resume Screening & Generation Platform is an AI-powered tool that helps both **job seekers** and **recruiters** assess the fit of a resume for a given role in the **software product and services industry**.

Users can upload a resume (PDF/DOCX) or authorize LinkedIn profile import. The system uses **Gemini 2.0** to parse, evaluate, score, and suggest improvements. It then optionally generates a polished, ATS-friendly resume using one of five modern templates — exported as a **watermarked PDF** (“Generated using [nayaneshgupte.com](https://nayaneshgupte.com)”).

This product aims to bring **transparency, consistency, and speed** to resume evaluation, while giving job seekers high-quality, personalized feedback that actually helps them land interviews.

---

## 2. Problem Statement

Today’s resume screening process suffers from three major issues:

### (1) Job seekers don’t know how well their resume matches a role

They apply blindly and make changes based on guesswork. Most cannot benchmark their skills, seniority, or experience strength reliably.

### (2) Recruiters drown in manual screening

Recruiters in software companies deal with hundreds of resumes per role. Manual scanning for relevant skills, experience depth, and keywords consumes 60–70% of hiring time.

### (3) Resume-building tools are generic and lack role alignment

Existing resume builders provide templates but do not:

- Assess resume vs job role

- Score skills or keyword alignment
- Provide actionable insights powered by real LLM reasoning

There's a clear gap for a tool that combines **screening + scoring + improvements + tailored resume generation** — all in one workflow.

---

## 3. Product Vision

**“Create a simple, intelligent assistant that evaluates a resume like a top recruiter and rewrites it like a world-class career coach.”**

The vision is to build the **most accurate, AI-first resume screening and generation platform** tailored to the software product/services world.

This platform should:

- Understand job roles deeply (Product Manager, SDE, QA, DevOps, Data Scientist, etc.)
- Benchmark skills and experience realistically
- Provide transparent scoring (100-point scale with category weights)
- Suggest actionable improvements, not generic tips
- Generate ATS-proof resumes instantly
- Work with both uploaded resumes AND LinkedIn data
- Never store user data permanently (privacy first)

Ultimately, the user should feel:

**“This tool not only tells me my score — it tells me exactly what I need to do to improve.”**

---

## 4. Strategic Rationale

Why this product matters:

**(a) AI-driven hiring is becoming the norm**

Companies increasingly rely on automated tools to screen resumes. This product puts the same power in the hands of job seekers and recruiters.

**(b) Gemini 2.0's deep reasoning is perfect for nuanced career evaluation**

Unlike older models, Gemini 2.0 understands:

- Seniority indicators
- Product impact
- Skill depth
- Industry-specific experiences
- Role-specific expectations

**(c) Huge audience value**

- Job seekers want higher interview conversion.
- Recruiters want time saved and better candidate matching.
- Career coaches want a tool for guided consultation.

**(d) Resume generation is a natural upsell path**

Once users see their score + improvement tips, they are highly motivated to create a better resume.

---

## 5. Success Metrics (KPIs)

### Primary KPIs

1. **Resume Screening Completion Rate**  
% of users who finish the screening process.
  2. **Improvement Recommendation Adoption Rate**  
% who implement suggestions or generate a new resume.
  3. **Resume Generation Conversion Rate**  
Users who choose a template and download a new PDF.
  4. **Accuracy Satisfaction Score (User Rated)**  
“How accurate was the evaluation?” (1–5)
  5. **Average Processing Time**  
Target: < 10 seconds from upload → results.
- 

## Secondary KPIs

6. **Repeat User Rate** (multiple screenings)
  7. **LinkedIn OAuth Success Rate**
  8. **Template Preference Analytics**
  9. **Session Abandonment Rate** at key steps
  10. **Time Spent on Insights Page**
- 

## Long-term North Star

### “Resume Improvement Impact Score”

Measures how much a user’s score improves after using the generated resume.

A 20–40 point improvement indicates strong product value.

---

# Market Research & Competitive Analysis

This includes:

- Industry overview
  - Competitor 1: Rezi
  - Competitor 2: KickResume
  - Competitor 3: LinkedIn Resume Builder
  - Competitor 4: AI resume tools (Teal, Enhancv, etc.)
  - What gaps exist
  - Strategic differentiators
- 

# Market Research & Competitive Analysis

## 6. Industry Overview

The resume optimization and screening ecosystem has exploded in the last 3 years, driven by:

### **(1) Massive shift toward AI-assisted hiring**

ATS systems now account for **70–80%** of initial resume screening in mid-to-large tech companies.

### **(2) Job seekers increasingly rely on AI tools**

Generative AI adoption (ChatGPT, Gemini, Claude) has normalized using AI for:

- Resume writing
- LinkedIn profile enhancement

- Role-specific tailoring

### (3) The market is fragmented

Dozens of AI resume tools exist, but most fail on:

- True role-to-resume matching
- Deep experience evaluation
- Comprehensive scoring frameworks
- Integration with LinkedIn

### (4) AI resume builders lack “recruiter-like reasoning”

Most tools rewrite resumes—but do *not* evaluate or benchmark quality realistically.

### (5) Recruiters need faster screening workflows

Technical hiring is flooded with similar resumes; tools that surface strong matches early save huge time.

#### Opportunity:

There is no single product that **accurately evaluates, scores, recommends, and generates** resumes in one flow, powered by deep LLM reasoning.

---

## 7. Competitive Benchmarking

Below is a competitive landscape evaluation across 12 parameters.

### Competitor 1: Rezi

#### Strengths:

- Good ATS-ready templates
- Simple UI

- Decent keyword matching

#### **Weaknesses:**

- Weak scoring logic
- No deep “role alignment reasoning”
- No LinkedIn data import
- Feedback is generic and template-based
- Does not evaluate seniority fit or experience quality

#### **Gap we fill:**

Real recruiter-style scoring + comprehensive improvement suggestions.

---

## **Competitor 2: KickResume**

#### **Strengths:**

- Strong template variety
- Good text auto-generation

#### **Weaknesses:**

- Role-to-resume match scoring is minimal
- No skill depth analysis
- Does not provide improvement insights beyond rewriting

#### **Gap we fill:**

Our LLM engine provides diagnostic scoring + detailed tips.

---

## **Competitor 3: LinkedIn Resume Builder**

### **Strengths:**

- Uses user's own profile data
- Good basic formatting

### **Weaknesses:**

- No scoring
- No role match evaluation
- Limited creative control
- No tips for improvement
- No ATS optimization

### **Gap we fill:**

Full LinkedIn import + AI evaluation + resume rewrite.

---

## **Competitor 4: Teal HQ**

### **Strengths:**

- Great for job tracking
- Keyword highlighting

### **Weaknesses:**

- No scoring model
- No resume rewriting
- No deep parsing

**Gap we fill:**

Full-stack screening + generation system.

---

## Competitor 5: ChatGPT-based “AI resume generators”

**Strengths:**

- Flexible
- Good rewriting capability

**Weaknesses:**

- Not structured
- No standardized scoring
- No downloadable templates with consistent formatting
- No persistent weight logic
- No LinkedIn import flow

**Gap we fill:**

A real, productized system — not a conversation.

---

## 8. Product Differentiators

This product stands out with:

**✓ Recruiter-grade scoring engine**

Most competitors match keywords.

We evaluate:

- Skill depth

- Experience seniority indicators
- Impact statements
- Project relevance
- Role-specific competencies

### ✓ Real-time weighted scoring with stable weights

Each screening session maintains consistent scoring weights so users can track progress scientifically.

### ✓ LinkedIn full import — fully parsed

This is rare among consumer resume tools.

### ✓ Actionable improvement engine

Not just “add more keywords.”

We produce:

- Skill gaps
- Experience alignment issues
- Formatting fixes
- Keyword recommendations
- Achievement rewriting suggestions

### ✓ Resume generation using professional ATS templates

Clean, structured, recruiter-approved templates.

### ✓ Privacy-first temporary processing

Unlike Rezi or KickResume, we **don't store resumes**.

### ✓ Built fully on Google AI Studio with Gemini 2.0

Better reasoning  
Better parsing  
Better consistency

---

## 9. Target Market Segments

### Primary Market

#### Job seekers in the software product & services industry

- SDEs
- Product Managers
- QA Engineers
- DevOps + Cloud
- Data Engineers / Scientists
- UI/UX designers
- Technical PMs
- Engineering Managers

### Secondary Market

#### Recruiters in tech companies

- Tech recruiters
- Hiring managers
- HR OPS teams

### Tertiary Market

#### Career coaches

- Universities
- Bootcamps
- Private consultants

## Long-term Market (Phase 2 or 3)

- Non-tech job markets
  - Freelancers (Upwork/Fiverr integration)
  - Staffing agencies
- 

# User Personas & Detailed User Journeys

This includes:

- Primary Personas (Job Seeker, Recruiter)
  - Secondary Personas (Career Coach)
  - Motivations & frustrations
  - Detailed task flows
  - End-to-end journeys
  - Screens and expected behavior
-

# User Personas & Detailed User Journeys

This section defines the core user types, their motivations, frustrations, and exactly how they will move through the product.

---

## 10. Primary User Personas

### Persona 1: The Job Seeker (Tech Professional)

**Name:** "Rohan Singh"

**Age:** 26–40

**Roles:** SDE, PM, QA, Data Engineer, DevOps, UX Designer, EM

**Tech-savviness:** High

**Motivation:** Get more interview calls, stand out from competition

**Pain Points:**

- Not sure if resume matches the target role
- Unsure how to present achievements in strong formats
- Struggles with ATS-friendly formatting
- Confusion around keywords
- Wants specific improvement pointers
- LinkedIn profile and resume often inconsistent

**What they want from the product:**

- A **resume score** that feels like a recruiter's evaluation
- Concrete, actionable **improvement suggestions**
- Ability to **regenerate a better version** instantly
- A professional **ATS-friendly template**

---

## Persona 2: The Recruiter

**Name:** "Aditi Sharma"

**Age:** 30–45

**Industry:** Software product/service companies

**Motivation:** Speed + Accuracy

**Pain Points:**

- Too many resumes to screen manually
- Hard to detect true skill depth
- Keyword stuffing leads to false positives
- Wants faster, consistent screening
- Needs a clear fit score and explanation

**What they want from the product:**

- Upload resume stacks (in future phase)
- Get quick scoring + reasoning
- Improve shortlisting accuracy
- Share insights with hiring managers

**Note:** Recruiter features will deepen in later phases — this PRD focuses mainly on resume-level screening per your scope.

---

## 11. Secondary Personas

### Persona 3: Career Coach / Counsellor

**Motivation:**

Use the tool as a coaching assistant to guide clients better.

**Desired outputs:**

- Resume analysis
  - Gap identification
  - Rewrite suggestions
  - Role-fit scoring
- 

## 12. Motivations & Pain Points (Across Personas)

 ***Motivations***

1. Better interview conversion
2. Save time in tailoring resumes
3. Understand seniority expectations
4. Improve LinkedIn + resume alignment
5. Generate cleaner, professional resumes
6. Remove guesswork in job applications

 ***Common Pain Points***

1. Unsure if resume is good enough
2. Hard to quantify role-fit

3. Existing tools give generic advice
  4. LinkedIn vs Resume mismatch
  5. Formatting issues lead to rejection
  6. Difficulty in highlighting impact
- 

## 13. Task Flows

The system supports three main flows:

---

### Flow A — Resume Screening Flow

**Goal:** Evaluate resume vs job role

**Steps:**

1. User lands on homepage
  2. Chooses: upload resume OR import LinkedIn
  3. Provides target role
  4. System parses resume/profile
  5. LLM generates structured data
  6. LLM applies scoring model
  7. Weighted categories computed
  8. Score dashboard displayed
  9. Tips & improvement suggestions shown
-

## **Flow B — Resume Improvement Flow**

**Goal:** Improve based on tips

**Steps:**

1. User reviews improvement areas
  2. Chooses to auto-update content
  3. LLM rewrites appropriate sections
  4. User approves changes
  5. User can rescore for comparison
  6. Score weight consistency is maintained
- 

## **Flow C — Resume Generation Flow**

**Goal:** Generate PDF resume

**Steps:**

1. User selects 1 of 5 templates
  2. User allows use of resume or LinkedIn data
  3. LLM generates structured resume content
  4. Formatting engine applies template
  5. PDF is generated with watermark
  6. User downloads resume
- 

## **14. End-to-End User Journeys**

Below are detailed user journeys for the two core personas:

---



## Journey 1 — Job Seeker (Most Common Path)

**Trigger:** User wants to apply for a role like “Senior Product Manager”.

### Step-by-step:

#### 1. Homepage → Upload Resume

- Upload PDF/DOCX.
- Option: “Import from LinkedIn (Recommended)”.

#### 2. Input Target Role

- User types: “Senior Product Manager”.
- Auto-suggestion: “Is this correct?”

#### 3. System Parses Resume

Gemini 2.0 extracts:

- Skills
- Experience
- Achievements
- Metrics
- Project relevance

#### 4. LLM Evaluation + Scoring

Weighted categories:

- Skills Match
- Experience Depth
- Keywords

- Seniority Alignment
- Formatting

## 5. Result Dashboard

- Score: 62/100
- Subscores visible
- “Top 3 strengths”
- “Top 5 improvement areas”

## 6. Tips Section

Examples:

- “Add more metrics to achievements.”
- “PM role expects ownership of product lifecycle — missing in description.”

## 7. Generate Resume

- User picks template #3
- PDF generated with watermark

## 8. Download

User gets an improved resume.



## Journey 2 — Recruiter Screening a Candidate

1. Upload candidate resume
2. Enter role (e.g., “Backend Engineer - Golang”)
3. Get score + reasoning
4. Share or export insights

5. Make shortlist decision

Recruiter mostly uses the **score dashboard + reasoning**.

---

## Journey 3 — Career Coach

1. Imports user's resume
  2. Reviews insights
  3. Advises candidate using the system-generated tips
  4. Helps regenerate resume
  5. Provides candidate with a final PDF
- 

## Functional Requirements (FRD)

This will include:

- Feature-by-feature requirements
  - Detailed BRDs
  - UX behavior for each feature
  - Acceptance criteria
  - Edge cases
  - Error handling flows
-

# Functional Requirements (FRD)

*(Very detailed, enterprise-level, feature-by-feature breakdown)*

This section defines exactly what the system must do — including BRDs, acceptance criteria, error states, and edge cases.

---

## 15. Feature-Level Requirements Overview

The system includes the following major functional modules:

1. Resume Upload Module
2. LinkedIn OAuth Import Module
3. Resume Parsing & Structuring Engine
4. Role Input & Interpretation Module
5. Screening & Scoring Engine
6. Improvement Suggestions Engine
7. Resume Generation Module
8. Template Rendering Engine (PDF + watermark)
9. Download & Export Module
10. Privacy & Auto-Deletion System
11. Onboarding & Consent Screens

Each module is detailed below.

---

# 16. Business Requirement Definitions (BRDs)

## BRD 1 — Resume Upload (PDF/DOCX)

### Description:

Users can upload their existing resume in PDF or Word (DOCX) format.

### Requirements:

- Accept .pdf and .docx files
- Max file size: 10 MB
- Virus/Malware scan (Cloud Functions / Cloud Run)
- Convert DOCX → text
- Extract text from PDF
- Handle multi-column layouts

### Acceptance Criteria:

- User can successfully upload a file
- System displays filename + size
- System rejects unsupported formats
- Parsing succeeds 95% of the time

### Edge Cases:

- Scanned PDF (image-only) → Show message: “Unable to parse scanned documents. Upload a text-based PDF.”
- Empty file → Error

- Encrypted PDFs → Error
- 

## BRD 2 — LinkedIn OAuth Integration

### Description:

User can import their profile using LinkedIn login.

### Requirements:

- OAuth 2.0 with full profile read access
- Retrieve:
  - Experience
  - Education
  - Certifications
  - Skills
  - About/Summary
- Normalize data into our structured schema

### Acceptance Criteria:

- User sees LinkedIn login flow
- On success, profile data appears in preview
- If denied access → Resume upload fallback

### Edge Cases:

- LinkedIn API limits
- Missing sections on LinkedIn

- Name mismatch between resume and profile
- 

## BRD 3 — Role Input Module

### Description:

User specifies the role for which they want evaluation.

### Requirements:

- Free text input
- Autosuggestions for common roles
- Validate role (LLM classification)
- Store role for scoring model

### Acceptance Criteria:

- Role is accepted and validated
- User can edit the role anytime

### Edge Cases:

- Too generic role (“Software Developer”) → Ask for level
  - Multiple roles → Ask user to pick one
- 

## BRD 4 — Resume Parsing Engine

### Description:

Gemini 2.0 converts resume/LinkedIn content to structured JSON.

### Required Extracted Fields:

- Name
- Contact
- Skills (grouped by category)
- Experience entries
- Seniority signals
- Metrics & achievements
- Projects
- Education
- Certifications
- Tools/technologies

#### **Acceptance Criteria:**

- Parsing accuracy ~90%+ on structured resumes
- Skill extraction must categorize skills correctly
- Achievements must be detected as bullet points

#### **Edge Cases:**

- Non-English resumes
- Poorly formatted resumes
- Very long resumes (>3 pages)

---

## **BRD 5 — Scoring Engine (Out of 100)**

**Description:**

LLM evaluates the candidate's fit for the target role using weighted metrics.

**Scoring Categories (fixed totals, weights stable per user session):**

1. Skills Match – 30
2. Experience Depth – 30
3. Role-Specific Keywords – 20
4. Seniority Alignment – 10
5. Formatting & Clarity – 10

**Requirements:**

- Auto-adjust scoring weights (LLM) but freeze them per session
- Provide subscore breakdown
- Provide textual explanation for each subscore

**Acceptance Criteria:**

- Final score always = 100
- Subscores add up to 100
- Score reasoning is human-readable

**Edge Cases:**

- Resume has no skills section → Assign zero + tips
- Resume is extremely short → Lower experience weight

---

## BRD 6 — Improvement Suggestions Engine

**Description:**

Generate actionable improvements based on weaknesses.

**Must include:**

- Missing skills
- Missing keywords
- Weak experience descriptions
- Achievements lacking metrics
- Formatting issues
- Seniority mismatch
- Soft skills missing (if relevant)

**Acceptance Criteria:**

- At least 5 meaningful, role-based suggestions
- Suggestions should match subscores

**Edge Cases:**

- Resume already very strong → Show “minor improvements”
  - Resume poorly structured → Provide “rebuild recommended”
- 

## BRD 7 — Resume Generation Module

**Description:**

Generate a rewritten resume using:

- Parsed structured data
- LLM-enhanced text

- User's preferred template

### **Output:**

- Watermarked PDF: "Generated using nayaneshgupte.com"
- ATS-friendly formatting

### **Acceptance Criteria:**

- No LLM hallucinations
- No invented companies/education
- Bullet points should follow STAR/impact style
- Headings consistent across templates

### **Edge Cases:**

- Missing data (skills/education) → Generate placeholder section
  - No achievements → Suggest metric-driven rewrites
- 

## **BRD 8 — Template Library (5 Templates)**

Templates must be:

- ATS-friendly
- Modern + professional
- Clean typographic hierarchy
- Minimal use of symbols
- Consistent margins and spacing

## **Template Types:**

1. Classic ATS
  2. Modern Minimalist
  3. Product/PM Style
  4. Engineering-Focused
  5. Leadership/EM Template
- 

## **BRD 9 — PDF Rendering & Export**

### **Requirements:**

- Render structured resume content into selected template
- Add footer watermark
- Output only PDF (no DOCX to avoid formatting issues)

### **Acceptance Criteria:**

- PDF renders within <4 seconds
  - Consistent formatting
  - No broken text
- 

## **BRD 10 — Privacy & Auto-Deletion System**

### **Requirements:**

- Do not store resumes or profile data

- Auto-delete all processed data within 5 minutes
- Display privacy consent prior to upload

#### **Acceptance Criteria:**

- Deletion logs visible in backend
- Immediate purge option available in UI

#### **Edge Cases:**

- User uploads twice → Both purge events logged
  - LinkedIn session expires → Data deletion triggered
- 

## **17. Acceptance Criteria Summary (Condensed Table)**

<b>Feature</b>	<b>Acceptance Criteria</b>
Resume Upload	Accepts PDF/DOCX, parses text accurately
LinkedIn Import	Profile loaded with experience/skills
Role Input	Role validated by LLM
Parsing Engine	Extracts fields into structured JSON
Scoring Engine	Delivers 100-point scoring + breakdown
Suggestions Engine	At least 5 actionable suggestions
Resume Generator	Creates ATS-formatted, error-free PDF
Templates	5 selectable templates, consistent layout

---

Auto-Deletion	Removes all data in <5 min
---------------	----------------------------

---

## 18. Edge Cases & Error Handling

### User Errors

- Uploads wrong file → Show “Unsupported file type”
- File too large → Show size error
- Scanned PDF → Suggest re-upload

### Data Errors

- Missing sections in resume → Show warnings
- LinkedIn API returns partial data → Merge intelligently

### System Errors

- LLM fails → Retry with fallback prompt
- Template render fails → Suggest another template

### Timeouts

- If screening exceeds 10 seconds → Show progress bar + retry
  - If LinkedIn tokens expire → Re-authenticate
- 

## AI/LLM Design (Core of the System)

This includes:

- LLM use cases
- Prompt architectures
- Scoring model logic
- Parsing strategies
- Consistent scoring weight retention
- Tips engine patterns
- Resume generation LLM flows

This is the **heart of the product**, the most technical section.

---

## AI / LLM Design (Core of the System)

*(This is the heart of the product — extremely detailed, enterprise-level, technical + practical + PM-friendly)*

This section defines EXACTLY how Gemini 2.0 will parse, score, reason, generate suggestions, and rewrite resumes.

Every component below becomes a building block in Google AI Studio.

---

## 19. LLM Use Cases (High-Level)

Gemini 2.0 will be used across these core functions:

1. **Resume Text Extraction → Structured JSON**
2. **LinkedIn Profile Structuring**
3. **Role Understanding & Requirements Extraction**

4. **Resume vs Role Scoring**
  5. **Weighted Scoring Breakdown Calculation**
  6. **Improvement Suggestions Generation**
  7. **Rewrite/Enhance Resume Content**
  8. **Template-Ready Resume Formatting Data**
  9. **Final Consistency Checker** (optional LLM pass)
- 

## 20. Prompt Engineering Framework

To ensure predictable, stable results, we use a **three-layer prompt architecture**:

---

### Layer 1 — System Prompt (Global Rules)

Attached to **every** request for consistent behavior:

You are a senior technical recruiter with 12+ years experience hiring for software product and services companies.  
You analyze resumes with deep reasoning, evaluate skill depth, detect seniority signals, and benchmark candidates against target roles.  
Follow ATS principles, avoid hallucination, and never invent experience or data.  
Output structured JSON following the exact schema provided.

---

### Layer 2 — Task Prompts (Per Feature)

Examples:

- **Parsing Prompt**
- **Scoring Prompt**
- **Keyword Extraction Prompt**
- **Improvement Suggestions Prompt**
- **Resume Generation Prompt**

Each prompt has:

- Objective
  - Input structure
  - Output schema
  - Safety guardrails
- 

## **Layer 3 — Output Enforcement Prompt (Schema Validator)**

Ensures correct structured output:

If output does not match the schema, correct it automatically.  
Do not add extra fields. Do not omit required fields.  
Return only valid JSON.

---

## **21. LLM Scoring Model (Weighted Scoring Logic)**

This is one of the most critical parts.

## Fixed Category Weights (Total = 100)

Category	Weight
Skills Match	30
Experience Match	30
Role Keywords	20
Seniority Alignment	10
Formatting	10

**Weights remain fixed for the entire session**, even if user updates resume.

## How Auto-Adjustment Works

Gemini adjusts *internal logic* per role — not the weights.

Example:

A PM role may emphasize communication & ownership, while an SDE role emphasizes coding depth.

Weights stay the same; **scoring rationale changes dynamically**.

---

## Scoring Prompt (Core Prompt)

**Input:**

- Structured resume JSON
- Structured role expectation JSON
- LLM weight object
- User session ID

**Output:**

```
{  
  "total_score": 00,
```

```
"scores": {
    "skills_match": {
        "score": 00,
        "justification": "..."
    },
    "experience_match": {...},
    "keywords": {...},
    "seniority_alignment": {...},
    "formatting": {...}
},
"final_summary": "Short recruiter-style summary"
}
```

---

## 22. Resume & LinkedIn Parsing Strategy

### Parsing Approach

1. Extract raw text
2. Use Gemini to interpret sections using category classification
3. Convert into structured JSON:

```
{
  "name": "",
  "contact": "",
  "skills": { "technical": [], "soft": [], "tools": [] },
  "experience": [
    {
      "company": "",
      "role": "",
      "duration": "",
      "achievements": [],
      "tools_used": [],
      "impact_metrics": []
    }
  ]
}
```

```
    },
],
"education": [],
"certifications": [],
"projects": []
}
```

## LinkedIn Data Merge Logic

- Where sections overlap, LinkedIn takes priority
- Resume-only sections retained
- Missing fields filled from LinkedIn

Example:

LinkedIn has 10 skills, resume has 3 → merge both.

---

## 23. Role Understanding & Requirement Extraction

Gemini takes a role input like:

“Senior Product Manager”

and generates:

- Required hard skills
- Required soft skills
- Expected seniority signals
- Expected responsibilities
- Relevant keywords

- Minimum years of experience
- Tooling preference (Jira, SQL, Figma etc.)

#### **Structured Output Example:**

```
{
  "role_title": "Senior Product Manager",
  "expected_skills": ["roadmapping", "product strategy", "A/B testing"],
  "seniority_signals": ["end-to-end ownership", "cross-functional leadership"],
  "keywords": ["KPIs", "OKRs", "user research"],
  "expected_experience_level": "6-10 years"
}
```

---

## **24. Scoring Consistency Logic**

*(Critical for user trust)*

#### **Rule:**

Weights freeze once the scoring engine runs the **first time** in a session.

#### **Stored in session:**

```
{
  "session_id": "",
  "weights": {
    "skills": 30,
    "experience": 30,
    "keywords": 20,
    "seniority": 10,
    "formatting": 10
  }
}
```

**Even if the user:**

- Uploads a new resume
- Imports LinkedIn
- Changes template

**Weights do NOT change.**

This ensures:

- Before/after comparisons are valid
  - User improvements feel measurable
- 

## **25. Improvement Tips Engine**

**Input:**

- Subscores
- Reasoning
- Parsed resume
- Parsed role
- Keyword mismatch list
- Skill gap list

**Output Categories:**

1. **Skills Missing**
2. **Experience Weaknesses**

3. **Keyword Gaps**
4. **Seniority Indicators Missing**
5. **Formatting Issues**
6. **Impact/Metric Improvements**

### **Example Output:**

```
{  
  "tips": [  
    {  
      "title": "Add metric-driven achievements",  
      "description": "Your Software Engineer role descriptions lack  
      quantifiable  
          metrics such as performance gains or efficiency  
      improvements."  
    },  
    {  
      "title": "Missing role keywords",  
      "description": "The target role requires experience with  
      Kubernetes, which  
          is not mentioned in your resume."  
    }  
  ]  
}
```

---

## **26. Resume Generation LLM Flow**

Uses:

- Parsed resume
- LinkedIn data
- Role expectations

- Tips engine insights

### **Steps:**

1. LLM rewrites experience using STAR-style
2. Ensures role relevance
3. Adds metric-driven language
4. Creates crisp headers
5. Balances technical & soft skills

### **Output JSON:**

```
{  
  "header": {...},  
  "sections": {  
    "summary": "...",  
    "skills": [...],  
    "experience": [...],  
    "education": [...],  
    "certifications": [...]  
  }  
}
```

### **Passed to Template Engine**

This JSON maps to placeholders in the template.

---

## **27. Consistency Checker (Optional 2nd LLM Pass)**

Ensures:

- No hallucinated companies
- No fictitious dates
- No skill mismatches

If detected → LLM repairs the output.

---

## Technical Architecture & System Design

This includes:

- High-level architecture
- Component architecture
- Sequence diagrams
- Data flow diagrams
- LinkedIn API integration
- Security & scalability strategies
- Infra plan (Cloud Run + Firebase + GCS + AI Studio)

This next section is **very detailed & technical**, typically used for engineering & architecture teams.

---

## Technical Architecture & System Design

*(Enterprise-level, extremely detailed. This is the “Engineering Blueprint” for building the platform.)*

---

# 28. High-Level Architecture Overview

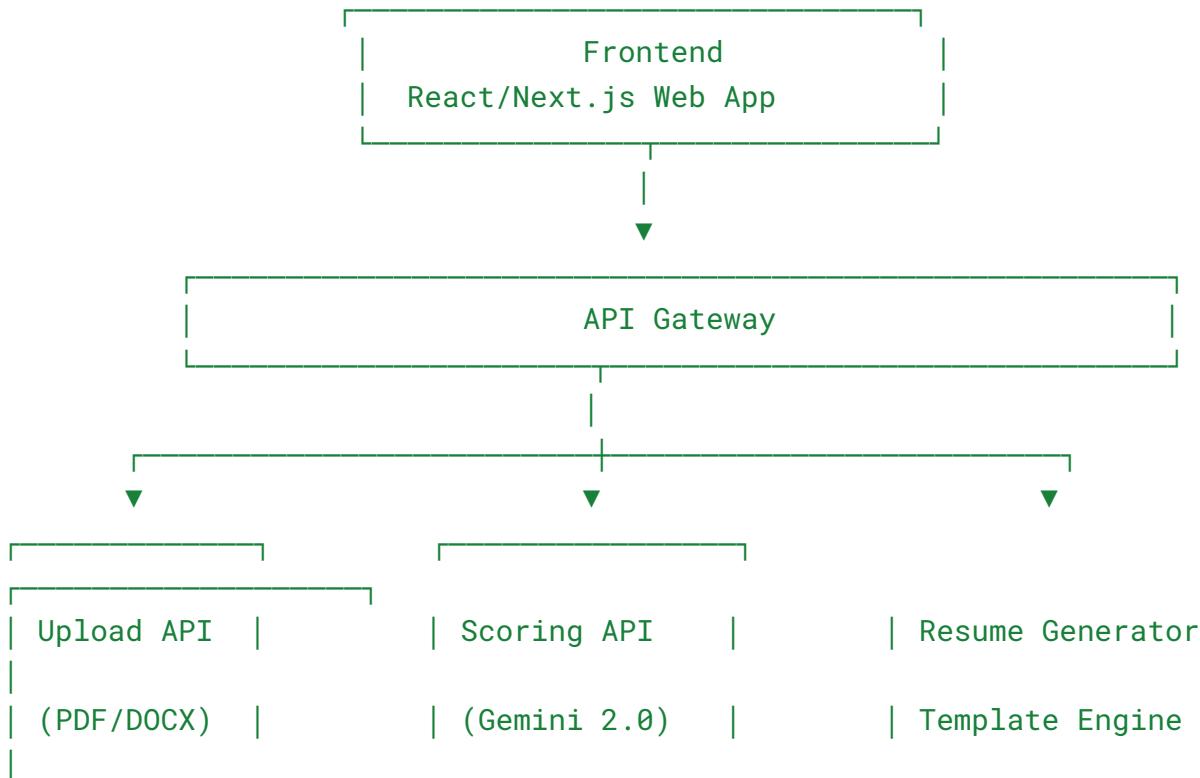
The system leverages Google Cloud + Google AI Studio + Gemini 2.0.

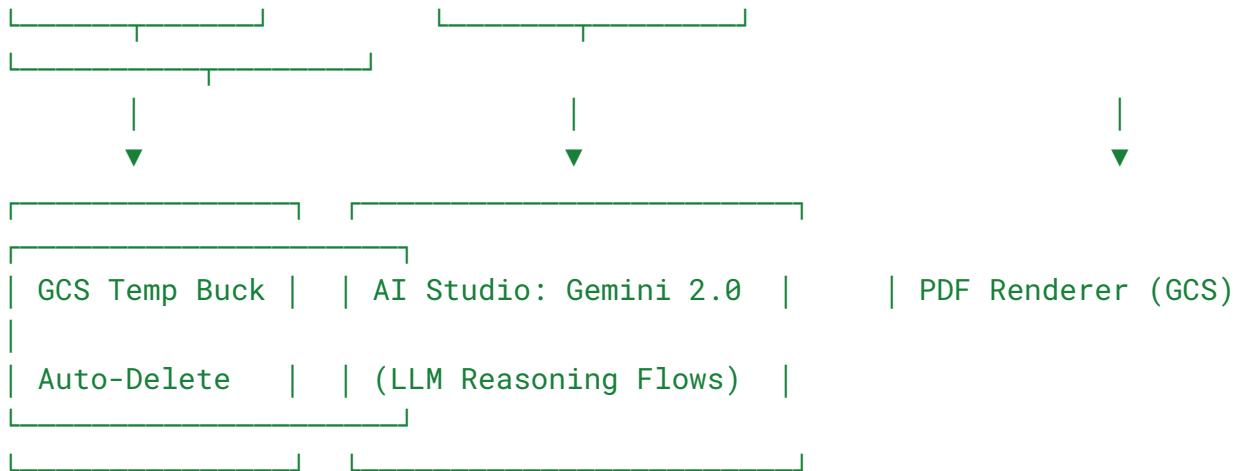
## Core Architectural Principles

- Serverless, scalable, low-maintenance
- Modular service components
- Stateless backend (resume data deleted after processing)
- Strict data isolation per session
- LLM as orchestrator + analyzer, not a monolith
- Clean API boundaries

---

## High-Level Architecture Diagram (Textual Representation)





## 29. Component-Level Architecture

Below is a breakdown of each service and its responsibility.

---

### 1. Frontend (React / Next.js)

- User onboarding
- Resume upload UI
- LinkedIn OAuth screen
- Role input
- Scoring dashboard
- Template selection
- Resume preview
- Download PDF

**Key Integrations:**

- Firebase Authentication
  - LinkedIn OAuth
  - Backend REST APIs
- 

## 2. Backend (Cloud Run Microservices)

### Service A — Resume Upload Service

- Receives PDF/DOCX
- Converts to raw text
- Sends extracted text → Parsing LLM

### Service B — LinkedIn Importer

- Manages OAuth flow
- Fetches profile
- Normalizes data
- Merges with resume data

### Service C — Parsing Service

- Calls Gemini 2.0
- Produces structured resume JSON

### Service D — Scoring Service

- Calls Gemini 2.0 with:
  - resume JSON

- role JSON
- frozen weights
- Returns score & breakdown

## **Service E — Improvement Suggestion Service**

- Uses LLM to produce tips

## **Service F — Resume Generator Service**

- Uses LLM to rewrite content
- Produces template-ready JSON

## **Service G — Template Renderer (PDF Service)**

- Renders PDF with watermark
  - Retrieves assets from GCS
- 

## **3. Data Storage (Temporary Only)**

### **GCS Bucket (with lifecycle rules)**

- Stores uploaded resume files
- Stores rendered PDF
- Auto-delete after 5 minutes

### **Firestore**

- Only stores session metadata:

- session\_id
  - frozen scoring weights
  - timestamps
- Does **not** store resume content
- 

## 4. Google AI Studio + Gemini 2.0

The core intelligence:

- Text extraction reasoning
  - Resume parsing
  - Role understanding
  - Scoring
  - Suggestions
  - Resume rewriting
  - Consistency checks
- 

## 30. API Endpoints & Contracts (High-Level)

### POST /upload

**Input:** PDF/DOCX  
**Output:** file\_id

### POST /parse

**Input:** file\_id or linkedin\_profile\_id  
**Output:** structured\_resume\_json

### **POST /role**

**Input:** role string  
**Output:** structured\_role\_json

### **POST /score**

**Input:** resume\_json + role\_json  
**Output:** score breakdown

### **POST /suggestions**

**Input:** score breakdown  
**Output:** tips array

### **POST /generate**

**Input:** resume\_json + template\_id  
**Output:** template\_json

### **POST /render**

**Input:** template\_json  
**Output:** PDF file URL

---

## **31. Sequence Diagrams**

---

### **Sequence 1 — Resume Screening**

User → Frontend: Upload resume  
Frontend → Upload API: File  
Upload API → GCS: Store Temp  
Upload API → Parsing API: Extract text  
Parsing API → Gemini: Parse resume  
Gemini → Parsing API: structured JSON

Parsing API → Scoring API: send resume JSON + role JSON  
Scoring API → Gemini: scoring prompt  
Gemini → Scoring API: results  
Scoring API → Frontend: score breakdown

---

## Sequence 2 — Resume Generation

User → Frontend: select template  
Frontend → Generator API: request rewrite  
Generator API → Gemini: rewrite resume sections  
Gemini → Generator API: template-ready JSON  
Generator API → PDF API: render template  
PDF API → GCS: store temp PDF  
PDF API → Frontend: PDF URL  
User downloads PDF

---

## 32. Data Flow Diagram (Text Form)



## 33. LinkedIn API Integration Architecture

### OAuth Flow:

1. User clicks “Import LinkedIn”
2. Redirect to LinkedIn OAuth screen
3. User grants permissions
4. LinkedIn sends auth code → Backend
5. Backend exchanges code ↔ access token
6. Backend fetches profile using `/v2/me`, `/v2/experience`, `/v2/skills`

### Data Normalization:

- Convert LinkedIn fields → internal schema
- Extract roles, skills, achievements
- Remove emojis, proprietary formatting

### Merge Rules:

- Experience: union + dedupe by company-role
- Skills: union
- Summary: prefer resume if stronger
- Education: union

---

## 34. Security Architecture

### Transport Security:

- HTTPS everywhere
- TLS 1.2+
- Strict CORS

### Access Controls:

- Firebase Auth
- OAuth tokens stored only in memory

### Data Privacy:

- No resume data stored
- Temp GCS bucket auto-deletes
- No logging of file contents

### LLM Safety:

- No hallucination allowed
  - No generation of inaccurate personal data
  - Strict output schema validations
- 

## 35. Scalability Design

## **Stateless Compute:**

- Cloud Run auto-scales instances
- Each LLM request independent

## **Load Handling:**

- CI/CD for rollout
- Rate limiting per user
- Queue retries for PDF rendering

## **Anticipated Loads:**

- 10–100 uploads/min
  - 300 scoring operations/min
  - 50 resume generations/min
- 

# **36. Latency Targets**

<b>Operation</b>	<b>Target Latency</b>
Resume Upload	< 2 sec
Parsing	< 4 sec
Scoring	< 5 sec
Suggestions	< 4 sec
Resume Generation Rewrite	< 6 sec
PDF Render	< 4 sec

Total End-to-End

< 15 sec

---

## 37. Infra Summary

- **Frontend:** Next.js on Vercel or Cloud Run
  - **Backend Services:** Cloud Run
  - **Storage:** GCS (auto-delete)
  - **Auth:** Firebase + LinkedIn OAuth 2.0
  - **LLM:** Gemini 2.0 via Google AI Studio
  - **Monitoring:** Cloud Logs + Error Reporting
- 

## Data, Privacy & Compliance

Includes:

- Data schema
  - Temporary storage rules
  - Auto-deletion workflows
  - Consent handling
  - GDPR-lite compliance
  - Token handling and expiry
-

# Data, Privacy & Compliance

(This part ensures the product is safe, compliant, privacy-first, and follows global standards.)

Your app's strongest brand differentiator is **privacy-first operation** — no user resumes or LinkedIn data are stored permanently.

This section defines **exact data schemas, retention policies, deletion workflows, compliance safeguards, and legal messaging**.

---

## 38. Data Schema (Internal Use Only)

The system does *not* store resume or LinkedIn content, but it still needs **temporary structured data** to perform operations.

Below is the minimal and temporary schema.

---

### Session Metadata (Firestore)

Stored for max **10 minutes**.

```
{  
  "session_id": string,  
  "created_at": timestamp,  
  "weights": {  
    "skills": number,  
    "experience": number,  
    "keywords": number,  
    "seniority": number,  
    "formatting": number  
  },  
  "status": "active" | "completed" | "expired"  
}
```

---

## Temporary Resume File (GCS)

Stored for < 5 minutes.

Metadata:

```
{  
  "file_id": string,  
  "session_id": string,  
  "uploaded_at": timestamp,  
  "deletes_at": timestamp  
}
```

**No resume text is stored in Firestore.**

Only the raw file in a temporary bucket before deletion.

---

## Temporary JSON Structures (In memory only)

Used for:

- resume\_json
- role\_json
- scoring\_json
- generation\_json
- suggestions\_json
- template\_json

These are passed between microservices but **never persisted**.

---

## 39. Temporary Data Processing Model

Here is how the system ensures privacy:

## Retention Model

Data Type	Where Stored	Duration	Auto-Delete ?
Uploaded PDF/DOCX	GCS Bucket	< 5 min	Yes
Generated Resume PDF	GCS Bucket	< 5 min	Yes
Session Metadata	Firestore	< 10 min	Yes
LLM Inputs/Outputs	Memory	Process lifetime	Auto-cleared

---

## Why memory-only?

- Ensures no long-term user data persistence
  - Complies with global data minimization standards
  - Reduces risk surface
- 

# 40. Auto-Deletion Workflow

*(This is critical for privacy compliance.)*

## Trigger Points:

- File upload
- Resume parsed
- Resume scored
- Resume generated
- 5-minute timer starts

---

## Auto-Deletion Sequence

Upload → Store → Process → Delete (within 5 minutes)

### Workflow Detail:

1. User uploads PDF
  2. GCS writes file with metadata:  
`deletes_at = now() + 5 minutes`
  3. Cloud Scheduler runs every minute
  4. Cloud Function checks GCS metadata
  5. Files past `deletes_at` are deleted
  6. Firestore session metadata cleaned if expired
- 

## 41. OAuth Permissions & Token Handling

### LinkedIn Permissions Requested

- r\_liteprofile
- r\_emailaddress
- r\_fullprofile
- r\_organization\_social (optional future)

### Token Handling Rules

- Access tokens stored only in RAM (backend instance memory)

- Never stored in Firestore
- Expire automatically
- Revoked after processing

## **LinkedIn Data Handling**

- Retrieved → normalized → used → discarded
  - No permanent storage
- 

# **42. Privacy Disclosures**

A dedicated screen must appear before any resume upload:

### **Privacy Notice:**

We do NOT store your resume, LinkedIn data, or personal information. All files and data are processed temporarily and auto-deleted after completion.

Your data is never used for model training or shared with third parties.

### **Key Requirements**

- Checkbox: “I understand and agree”
- Cannot proceed unless accepted

### **Additional Disclosures:**

- “This tool processes your resume using Google’s Gemini 2.0 models.”
- “No data is used to train any AI model.”

---

## 43. GDPR-Lite & Compliance Checklist

Even if not formally GDPR-certified, your product will follow **GDPR-like principles**:

---

### Data Minimization

Only essential data processed.

### Purpose Limitation

Data used strictly for resume evaluation & generation.

### Storage Limitation

Auto deletion within minutes.

### Right to Erasure

User can click “Delete Now” anytime → triggers immediate purge.

### Transparency

Clarity on what the LLM does and doesn't do.

### No Model Training

User data never enters training pipelines.

### Security Standards

- TLS 1.2+
- HTTPS enforced

- Encrypted storage for temp GCS
  - Strict IAM roles
- 

## 44. Token Expiry & Revocation Rules

- LinkedIn tokens expire in ~60 minutes
  - You explicitly revoke them after processing
  - Token revocation API call required
  - Ensures no unintended authorization remains
- 

## 45. Legal Considerations

- Terms of Use must state that resume evaluation is a recommendation tool
  - Must clarify that recruitment decisions remain user's responsibility
  - LinkedIn integration subject to LinkedIn API terms
  - PDF watermark ensures brand awareness and compliance
- 

## Resume Generation Module Specs

This includes:

- Template library design

- Template architecture
- JSON → Template mapping
- PDF generation rules
- Watermark rules
- ATS compliance standards
- Edge cases for poor resumes

## Resume Generation Module Specs

*(Deep-dive into templates, generation logic, ATS rules, rendering, watermarking, and edge cases.)*

This section defines the **full blueprint** for how resumes are generated, structured, and exported as PDFs.

---

## 46. Resume Generation Overview

The Resume Generation Module transforms:

- Parsed resume JSON
- LinkedIn data (optional)
- Improvement suggestions
- Role expectations
- LLM-enhanced rewritten content

into a **professionally formatted, ATS-friendly résumé** using 1 of 5 templates, exported as a PDF with the watermark:

“Generated using nayaneshgupte.com”

This module has three sub-systems:

**1 LLM Content Generator**

**2 Template Engine (JSON-to-layout mapper)**

**3 PDF Renderer + Watermark Layer**

---

## 47. Content Generation Specification (LLM)

This is where Gemini 2.0 rewrites the resume content using advanced, recruiter-style reasoning.

### Content Generation Goals

- Improve clarity
  - Add measurable impact
  - Use strong action verbs
  - Remove fluff
  - Ensure role alignment
  - Maintain authenticity (no hallucinations)
- 

### LLM Rewrite Rules

#### ✓ Rule 1 — STAR/Impact Format

Every experience bullet must follow:

- **Situation** (implicit)
- **Task** (optional)
- **Action** (strong verb)
- **Result** (metric)

Example rewrite:

**Weak:**

“Worked on backend APIs.”

**Strong:**

“Developed and optimized backend APIs using Node.js, improving response time by 27% for high-traffic modules.”

---

## ✓ Rule 2 — Keyword Reinforcement

Inject **role-specific terminology** without keyword stuffing.

## ✓ Rule 3 — Seniority Signal Enhancement

For senior roles (PM, EM, Lead SDE), emphasize:

- Ownership
- Leadership
- Strategy
- Cross-functional collaboration
- Metrics-driven outcomes

---

## ✓ Rule 4 — LinkedIn Alignment

Where available, ensure the resume does not contradict LinkedIn’s timeline or roles.

---

## LLM Output Structure (Resume JSON)

Final resume JSON:

```
{  
  "header": {  
    "name": "",  
    "email": "",  
    "phone": "",  
    "location": "",  
    "links": []  
},  
  "summary": "",  
  "skills": {  
    "technical": [],  
    "tools": [],  

```

---

This JSON maps directly into different templates.

# 48. Template Library Specifications (5 Templates)

Each template follows:

- ATS compliance
  - Clean section hierarchy
  - Font size 10–11
  - Minimal icons
  - No tables
  - No text inside images
  - Simple two-column layout (optional templates only)
- 

## Template 1: Classic ATS (Default)

- Single column
- Strict ATS formatting
- Bold section headers
- Ideal for tech roles

## Template 2: Modern Minimalist

- Clean sans-serif font
- Subtle separators
- Slight two-column layout

## **Template 3: Product Manager Style**

- Emphasis on *Summary* and *Impact*
- Highlight Product Skills (Leadership, Strategy, UX)

## **Template 4: Engineering-Focused Template**

- Highlights technical stack
- Focus on Skills + Experience
- Ideal for SDE/DevOps/Data roles

## **Template 5: Leadership/EM Template**

- Strong header
  - Emphasis on ownership, outcomes
  - Short summary + concise achievements
- 

# **49. Template Engine Architecture**

The Template Engine converts **resume JSON → PDF layout**.

### **Components:**

- JSON-to-HTML mapper (or JSON-to-Canvas)
  - Styling layer (CSS-like definition)
  - Renderer to PDF
-

## Template Mapping Rules

Example mapping:

```
header.name → <h1>
summary → <p class="summary">
skills.technical[] → bullet list under Technical Skills
experience[].achievements[] → bullet list under each role
```

Each template will have a **mapping config file**:

```
{
  "header_section": [ "header.name", "header.email", ... ],
  "sections": [
    {
      "title": "Experience",
      "fields": [ "experience[ ].company", "experience[ ].achievements" ]
    }
  ]
}
```

This makes template engine **fully modular**.

---

## 50. PDF Rendering Rules

The PDF renderer should:

**✓ Use a vector-based PDF generator**

(Google Cloud's built-in rendering or a Node.js PDF engine)

**✓ Ensure consistent spacing**

- 1.15 line height
- 6–8px section spacing

- 10–12px margin

✓ **Use safe fonts**

- Roboto
- Lato
- Open Sans  
(all ATS-safe)

✓ **No text should break across columns or sections**

---

## 51. Watermark System

Every generated resume must include:

**Footer text: “Generated using nayaneshgupte.com”**

**Watermark Rules:**

- Appears in **footer of every page**, right-aligned
- Font: 8px gray
- Transparent (60% opacity)

Ensures branding + anti-plagiarism + compliance.

---

## 52. ATS Compliance Standards

The resume **MUST** pass all ATS checks.

**Required:**

- No tables
- No images
- No icons
- No vertical text
- No floating elements
- No color-coded sections
- Only standard section headers:

Summary

Skills

Experience

Projects

Education

Certifications

### Ideal Fonts:

- Sans-serif only
- No decorative fonts

### Ideal File Type:

- PDF, text-readable
- No embedded images or layers

---

## 53. Edge Cases & Handling

### Edge Case 1 — Candidate has very little experience

Solution:

- Expand projects + skills
  - Add soft skills
  - Add “Academic Projects”
  - Provide stronger Summary
- 

## **Edge Case 2 — Candidate has no measurable achievements**

Solution:

- Rewrite bullets to include **role-specific impact**
  - Suggest metrics (even if estimated)
- 

## **Edge Case 3 — Candidate has mixed roles**

Solution:

- Emphasize experience matching the target role
  - Push mismatched experience to lower sections
- 

## **Edge Case 4 — Resume has inconsistent dates**

Solution:

- LLM corrects format
  - Leaves dates but ensures consistency
-

## Edge Case 5 — Resume too long

Solution:

- Summarize older roles
  - Focus on last 3–4 experiences
- 

## Detailed UX Flows + Screens

Will include:

- Screen-by-screen specifications
- UX flow diagrams
- Error/empty state designs
- Interaction rules
- Microcopy (button text, helper text)

## Detailed UX Flows + Screens

*(Highly detailed, product-manager + UX-designer style. Includes screens, flow diagrams, microcopy, and error/empty states.)*

This section serves as the **design blueprint** for the frontend engineers and UX/UI designers.

---

## 54. UX Principles

Before diving into screens:

## **1. Simple & Guided**

The user should always know:  
“**What’s the next step?**”

## **2. Trust & Transparency**

Especially during scoring & data processing:

- Clear privacy statements
- Clear scoring rationale

## **3. Instant Feedback Loops**

- Inline file validation
- Real-time progress indicators
- Retry options

## **4. Mobile-Friendly but Desktop-Optimized**

Most resume editing users are desktop users, but mobile should still be clean.

---

# **55. Key User Flows**

We'll document four major flows:

- 1. Onboarding + Consent Flow**
  - 2. Resume Screening Flow**
  - 3. Score Dashboard Flow**
  - 4. Resume Generation Flow**
-

# 56. User Flow #1 — Onboarding + Consent

## Screens Included

- Landing Screen
  - Data Privacy Consent Screen
  - File Upload Screen
  - OR LinkedIn Import Screen
- 

## Screen 1: Landing Page

**Purpose:** Introduce product, convert user into flow.

### Sections:

- Header
- Tagline
- CTA buttons

### UI Copy:

Title: Your AI-powered Resume Screening & Builder

Subtitle: Evaluate your resume like a top recruiter. Improve it with AI. Get a professional PDF instantly.

Primary CTA: Start Screening

Secondary CTA: Try with LinkedIn

### Elements:

- Illustration (optional)

- No login required
  - Footer with basic links
- 

## Screen 2: Privacy Consent Page

Mandatory before uploading or importing.

Copy:

We do NOT store your resume, LinkedIn data, or personal information. All data is processed temporarily and auto-deleted within minutes.

- ✓ Your data is not used for training any AI model.
- ✓ Your resume is never shared with anyone.
- ✓ You can delete all data anytime.

Checkbox:

I understand and agree to the Privacy Notice

Button:

[Continue](#)

---

## Screen 3: Resume Upload / LinkedIn Import

Layout:

Two large cards:

### Option A: Upload Resume

- Drag & Drop Zone
- “Upload PDF or Word (DOCX)”
- Supported formats displayed

## **Option B: Import from LinkedIn**

Button: “Import via LinkedIn →”

Uses OAuth redirect flow.

---

# **57. User Flow #2 — Resume Screening Flow**

## **Screens**

1. Role Input
  2. Parsing Loader
  3. Scoring Loader
  4. Score Dashboard
- 

## **Screen 1: Role Input**

**Copy:**

What role do you want to apply for?

Input field with auto-suggestions:

- Software Engineer
- Product Manager
- Data Analyst
- DevOps Engineer

- QA Automation Engineer
- Engineering Manager
- UI/UX Designer

Button: **Continue**

Error States:

- Role not recognized → “Please enter a valid job role.”
  - Too broad → “Please specify level — e.g., Senior Software Engineer.”
- 

## Screen 2: Parsing Loader

**UI:**

Progress indicator with steps:

- ✓ Upload
- ▶ Parsing Resume
- Scoring
- Tips
- Generate Resume

Message:

Analyzing your resume with Gemini 2.0...

Extracting skills, experience, achievements, and seniority signals.

Time: 2–4 seconds.

---

## Screen 3: Scoring Loader

Message:

Comparing your profile with the role: Senior Product Manager

Evaluating skills, experience depth, keywords, and seniority...

Time: 3–5 seconds.

---

## Screen 4: Score Dashboard

This is the hero screen.

### Header

Your Resume Score: 68 / 100

### Breakdown (visual bars)

- Skills Match — 22/30
- Experience Match — 18/30
- Keywords — 12/20
- Seniority Alignment — 8/10
- Formatting — 8/10

### Strengths Section:

Example:

- Strong experience in product execution
- Clear technical background
- Good cross-functional collaboration indicators

### Areas of Improvement:

Example:

- Missing key PM-specific metrics
- Lack of roadmap ownership signals
- Insufficient A/B testing experience

CTA:

[View Improvement Suggestions →](#)

---

## 58. User Flow #3 — Improvement Suggestions Flow

### Screen: Tips Breakdown

Organized by category:

1. **Skills Missing**
2. **Experience Gaps**
3. **Keyword Gaps**
4. **Seniority Indicators**
5. **Formatting Issues**

Each tip is expandable with examples.

Example Tip:

**Title:** Add measurable product KPIs

**Description:**

Your resume mentions product launches but does not include measurable outcomes.

Hiring managers expect KPIs such as conversion %, retention %, or DAU impact.

CTA:

[Generate Improved Resume](#)

---

## 59. User Flow #4 — Resume Generation Flow

### Screens

1. Template Selection
  2. Preview
  3. PDF Ready
- 

### Screen 1: Choose a Template

Grid layout with 5 templates:

- Classic ATS
- Modern Minimal
- Product Manager Focused
- Engineering Focused
- Leadership / EM

Button under each:

[Use This Template](#)

---

### Screen 2: Preview Screen

Left panel: PDF preview

Right panel: Editable fields (optional)

- Name
- Summary
- Skills
- Experience (editable bullet points)

Button:

[Generate PDF](#)

---

## Screen 3: Success Screen

Message:

Your resume is ready!

Downloaded PDF includes watermark: Generated using nayaneshgupte.com

Buttons:

- [Download PDF](#)
  - [Start Over](#)
- 

## 60. Error States

### Upload Errors

- “Unsupported file type. Please upload a PDF or DOCX.”
- “This file appears to be scanned. Please upload a text-based file.”

## **Parsing Errors**

- “We couldn’t parse your resume. Please try with a cleaner format.”

## **LinkedIn Errors**

- “Your LinkedIn session expired. Please reconnect.”

## **LLM Failures**

- “We encountered an issue while processing your resume. Please try again.”
- 

# **61. Empty States**

## **No Skills Found**

Display card:

“Your resume does not list any skills. We recommend adding a Skills section.”

## **No Achievements Found**

Suggest rewriting experience entries.

## **No Keywords Found**

Show required keywords for role.

---

# **62. Microcopy (Buttons & Helper Text)**

- Start Screening
- Upload Resume

- **Import from LinkedIn**
- **Continue**
- **View Suggestions**
- **Generate Resume**
- **Download PDF**
- **Delete My Data Now**

Helper text examples:

- “We don’t store your data.”
  - “Drag and drop or click to upload.”
  - “Your data is processed securely with Gemini 2.0.”
- 

## API Layer

Includes:

- Detailed API endpoint definitions
- Input/output schemas
- Errors and status codes
- Security rules
- LLM request/response structure

Reply “**continue**” to proceed to **PART 10**.

You said:

continue

ChatGPT said:

# API Layer (Backend Contract + LLM Interface Design)

*(Enterprise-grade, engineering-ready, with exact request/response schemas, status codes, and error handling.)*

This section defines the **complete backend API contract** used by frontend, backend microservices, and Google AI Studio LLM flows.

---

## 63. API Architecture Principles

All APIs follow:

- ✓ RESTful design
  - ✓ Stateless services
  - ✓ JSON request/response
  - ✓ OAuth-secured endpoints (for LinkedIn)
  - ✓ Strict validation
  - ✓ No storage of personal data
  - ✓ Temporary file handling only
- 

## 64. API Overview (Master List)

API	Method	Purpose
/upload	POST	Upload resume (PDF/DOCX)
/linkedin/auth	GET	Redirect to LinkedIn OAuth
/linkedin/callback	GET	Token exchange + profile fetch
/resume/parse	POST	Convert raw resume → structured JSON
/role/parse	POST	Convert job role → requirements JSON
/score	POST	Evaluate resume vs role
/suggestions	POST	Generate improvement suggestions
/resume/generate	POST	Generate rewritten resume JSON
/resume/render	POST	Render template → PDF
/session/delete	POST	Delete all user data immediately

---

## 65. API-by-API Specification (Deep Detail)

---

### ◆ 1. POST /upload

#### Description

Uploads PDF or DOCX resume and stores temporarily in GCS.

#### Request (multipart/form-data)

```
file: <binary>
session_id: string
```

#### Response (200)

```
{
  "file_id": "rsm_837472",
```

```
        "message": "File uploaded successfully."
    }
```

## Errors

Code	Meaning
400	invalid_file_format
413	file_too_large
500	upload_failed

---

## ◆ 2. GET /linkedin/auth

### Description

Redirect user to LinkedIn OAuth.

### Params

```
redirect_uri=<url>
state=<session_id>
```

### Behavior

- Generates OAuth URL
- Redirects user to LinkedIn

---

## ◆ 3. GET /linkedin/callback

### Description

LinkedIn sends authorization code to this endpoint.

## Request Query

```
?code=<auth_code>&state=<session_id>
```

## Response (200)

```
{
  "profile_id": "lnk_293847",
  "message": "LinkedIn import successful."
}
```

## Errors

- invalid\_oauth\_token
  - expired\_token
  - permission\_denied
- 

## ◆ 4. POST /resume/parse

### Description

Uses Gemini 2.0 to structure resume.

### Request

```
{
  "file_id": "rsm_837472",
  "session_id": "sess_12345"
}
```

## Response (200)

```
{
  "resume_json": {
    "name": "",
```

```
        "skills": {...},
        "experience": [...],
        "education": [...],
        "projects": []
    }
}
```

## Errors

- parsing\_failed
  - empty\_resume
  - malformed\_content
- 

## ◆ 5. POST /role/parse

### Description

Convert user-entered role into structured expectations.

### Request

```
{
  "role": "Senior Product Manager",
  "session_id": "sess_12345"
}
```

### Response (200)

```
{
  "role_json": {
    "role_title": "Senior Product Manager",
    "expected_skills": [...],
    "keywords": [...],
    "expected_seniority": "6-10 years"
  }
}
```

```
    }
}
```

---

## ◆ 6. POST /score

### Description

LLM scores resume vs role.

### Request

```
{
  "resume_json": {},
  "role_json": {},
  "weights": {
    "skills": 30,
    "experience": 30,
    "keywords": 20,
    "seniority": 10,
    "formatting": 10
  },
  "session_id": "sess_12345"
}
```

### Response (200)

```
{
  "total_score": 68,
  "scores": {
    "skills_match": { "score": 22, "justification": "..." },
    "experience_match": { "score": 18, "justification": "..." },
    "keywords": { "score": 12, "justification": "..." },
    "seniority_alignment": { "score": 8, "justification": "..." },
    "formatting": { "score": 8, "justification": "..." }
  },
  "final_summary": "..."
}
```

---

## ◆ 7. POST /suggestions

### Description

Generate actionable improvement tips based on score.

### Request

```
{  
  "resume_json": {},  
  "role_json": {},  
  "scores": {}  
}
```

### Response (200)

```
{  
  "tips": [  
    {  
      "title": "Add product KPIs",  
      "description": "You mention product delivery but no measurable  
      impact."  
    }  
  ]  
}
```

---

## ◆ 8. POST /resume/generate

### Description

Generate improved resume content.

### Request

```
{  
    "resume_json": {},  
    "role_json": {},  
    "tips": [],  
    "template_id": "template_3"  
}
```

## Response (200)

```
{  
    "generated_resume_json": {  
        "header": {...},  
        "summary": "...",  
        "skills": {...},  
        "experience": [...],  
        "education": [...],  
        "projects": []  
    }  
}
```

---

## ◆ 9. POST /resume/render

### Description

Convert resume JSON + template → final PDF.

### Request

```
{  
    "resume_json": {},  
    "template_id": "template_3",  
    "session_id": "sess_12345"  
}
```

## Response (200)

```
{
```

```
"pdf_url": "https://gcs/tmp/resume_12345.pdf",
"expires_in": 300
}
```

---

## ◆ 10. POST /session/delete

### Description

Immediately delete all temp data.

### Request

```
{
  "session_id": "sess_12345"
}
```

### Response (200)

```
{
  "message": "All resume and profile data has been deleted
successfully."
}
```

---

## 66. Authentication & Security Rules

- ✓ Firebase Auth (optional for saved sessions)
- ✓ LinkedIn OAuth
- ✓ CSRF protection for all POST endpoints
- ✓ Rate Limit
  - 10 uploads/hour

- 20 scoring operations/hour

✓ API Keys stored in Secret Manager

✓ 100% encrypted transport via HTTPS

---

## 67. LLM Request/Response Structure (Gemini 2.0)

Each LLM call must follow:

### Request Structure:

```
{  
  "system_prompt": "...",  
  "task_prompt": "...",  
  "input_json": {...},  
  "output_schema": {...}  
}
```

### Response:

```
{  
  "output_json": {...},  
  "confidence_score": 0.98  
}
```

---

## 68. Error Handling (Global)

All endpoints return:

```
{  
  "error": {  
    "code": "string",  
    "message": "string"  
  }  
}
```

```
        "message": "string"  
    }  
}
```

Standard error codes:

- INVALID\_INPUT
  - MISSING\_FIELD
  - PARSING\_FAILED
  - LLM\_ERROR
  - PDF\_RENDER\_ERROR
  - LINKEDIN\_AUTH\_EXPIRED
  - INTERNAL\_SERVER\_ERROR
- 

## QA, Monitoring & Analytics

Includes:

- QA strategy
- Test cases and categories
- Performance benchmarks
- Monitoring (logs, traces, error reporting)
- Analytics dashboards
- A/B testing strategy

# QA, Monitoring & Analytics

(This section ensures product reliability, performance, correctness, and long-term observability. Includes QA strategy, test plans, monitoring rules, and analytics dashboards.)

---

## 69. QA Strategy Overview

The product combines:

- File uploads
- LLM reasoning
- PDF generation
- Third-party OAuth
- Temporary data deletion

Therefore, QA covers **functional testing + LLM validation + performance testing + security testing + UX testing**.

---

## 70. Test Categories

### 1. Functional Testing

Validate that each API and UI flow works as expected.

Includes tests for:

- Upload
- LinkedIn import

- Resume parsing
  - Role parsing
  - Scoring
  - Suggestions
  - Resume generation
  - PDF rendering
  - Auto-deletion
- 

## 2. LLM Output Validation Testing

Unique to AI systems.

### Test aspects:

- Schema compliance
  - Safety and hallucination checks
  - Consistency of scoring
  - Accuracy of parsing
  - Relevance of improvement suggestions
  - Correctness of rewritten resume content
- 

## 3. Integration Testing

Between:

- Cloud Run ↔ GCS

- Cloud Run ↔ Firebase
  - Cloud Run ↔ LinkedIn API
  - Cloud Run ↔ PDF Renderer
  - Frontend ↔ Backend APIs
  - Backend ↔ AI Studio
- 

## 4. Performance Testing

Targets:

- Upload < 2 seconds
- Parsing < 4 seconds
- Scoring < 5 seconds
- Resume generation < 6 seconds
- PDF rendering < 4 seconds

Load tests for:

- 500 concurrent users
  - 100 scoring requests/minute
- 

## 5. Security Testing

- OAuth token leakage testing
- GCS bucket public access checks

- Resume file deletion tests
  - API rate limit testing
  - JWT authentication testing
- 

## 6. UX Testing

- A/B tests for templates
  - Test clarity of score explanations
  - Test microcopy language
  - User journey friction assessment
- 

# 71. Detailed Test Cases (Examples)

---

## 1. Upload Module Test Cases

Test Case	Expected Result
Upload valid PDF	File stored, parsing begins
Upload DOCX	Converted successfully
Upload large file > 10 MB	Error message
Upload scanned PDF	Show warning + request new file

---

## 2. LinkedIn Import Test Cases

Test Case	Expected Result
-----------	-----------------

---

Valid OAuth flow	Profile imported
Expired token	Re-auth prompt
Partial profile	Missing fields fallback
Permission denied	Show error & retry options

---

### 3. Resume Parsing Tests

Test	Result
Extract skills correctly	Skills array filled properly
Identify achievements	Bullet points parsed
Seniority signals extracted	Correct JSON fields

---

### 4. Scoring Engine Tests

Test	Expected Outcome
Resume with missing skills	Lower Skills Match score
Resume with senior-level experience	Higher Seniority score
Resume with formatting issues	Formatting = low score
Role mismatch	Score significantly reduced

---

### 5. Suggestions Engine Tests

Test	Expected Outcome
Skill gaps	Clear tips list
Missing metrics	Tips with metric examples

Wrong keywords   Suggest role-specific keywords

---

## 6. Resume Generator Tests

Test	Expected Outcome
Generate with Template 1	ATS layout consistent
Generate with Template 2	Modern layout consistent
Missing data	Template shows placeholder or hides section

---

## 7. Auto-Deletion Tests

Test	Expected Outcome
Wait 5 min	File deleted
Manual delete	Immediate purge
Retry retrieval after deletion	404 file not found

---

## 72. Performance Benchmarks

### Upload Latency Target

- < 2 seconds
- DOCX → text < 1 second

### LLM Calls

- Parsing: < 4 seconds
- Scoring: < 5 seconds
- Suggestions: < 4 seconds
- Resume rewrite: < 6 seconds

## PDF Rendering

- < 4 seconds
- 

# 73. Monitoring Architecture

Using:

- Cloud Logging
  - Cloud Error Reporting
  - Cloud Trace
  - Cloud Profiler
  - GA4 or Mixpanel (for front-end actions)
- 

## Logs Generated

### Backend Logs

- Resume upload success/failure
- Parsing success/failure

- Scoring request logs
- LLM request durations
- Template rendering logs
- GCS delete logs

## Frontend Logs

- CTA clicks
  - Page exits
  - Errors displayed
- 

# 74. Metrics & Dashboards

Dashboards will track:

### ✓ Resume Screening Metrics

- Screening success rate
- Average time to score
- Score distribution histogram

### ✓ Resume Generator Metrics

- Template usage heatmap
- PDF generation success rate
- Retry count

### ✓ LinkedIn Integration Metrics

- OAuth success rate
- Import success rate
- Average profile completeness

## ✓ Quality Metrics

- Average “Accuracy Satisfaction Score”
  - % of users who rescore after making changes
  - Improvement delta (before vs after score)
- 

# 75. A/B Testing Strategy

## Goals:

- Improve conversion to “Generate Resume”
- Improve clarity of dashboard
- Compare templates for user preference

## Experiments:

1. **Score Display Format A vs B**
  2. **Template Grid Layout vs Carousel Layout**
  3. **Summary rewrite vs no-summary rewrite**
  4. **Tips grouping styles**
-

# 76. Error Monitoring & Alerting

## Alerts Triggered When:

- Scoring engine error > 5%
- PDF rendering fails > 3%
- LinkedIn OAuth errors > 20%
- GCS cleanup failures
- LLM response time > 10 seconds

## Alert Channels:

- Slack
  - Email
  - PagerDuty (optional)
- 

# Release Plan & Roadmap

## Includes:

- MVP scope
- Phase 2, 3, 4 feature expansions
- Growth strategy
- Future integrations (Indeed, GitHub, Naukri, etc.)
- Timeline breakdown

# Release Plan & Roadmap

(Final section of the PRD. Enterprise-grade release plan, phased roadmap, timelines, and growth strategy.)

This outlines what goes into the MVP, what comes next, and how the product can evolve into a full ecosystem for resume screening, career guidance, and job-matching.

---

## 77. MVP Scope (What ships first)

The MVP focuses on delivering the **core user value**:

- ✓ **Resume Upload (PDF/DOCX)**
- ✓ **LinkedIn OAuth + Data Import**
- ✓ **Resume Parsing (Gemini 2.0)**
- ✓ **Role Parsing (Gemini 2.0)**
- ✓ **Resume Scoring (100-point weighted model)**
- ✓ **Improvement Suggestions**
- ✓ **Resume Generation (5 Templates)**
- ✓ **PDF Rendering + Watermark**
- ✓ **Temporary data processing + Auto deletion**
- ✓ **Privacy Consent**
- ✓ **Dashboard UI**

Everything included is necessary to provide:

**“Upload → Score → Improve → Generate → Download”.**

---

## 78. Non-MVP Items (Later Release)

These features require more engineering, product validation, or UX design:

- ✗ Batch Resume Upload for Recruiters
- ✗ Saving user profile history
- ✗ Resume version history
- ✗ Multi-language support
- ✗ Job description import from LinkedIn/Indeed pages
- ✗ Chrome extension
- ✗ Interview preparation module
- ✗ AI-powered cover letter generator
- ✗ Jobs-matching engine

These will be evaluated post-launch based on adoption.

---

## 79. Phase-Wise Feature Roadmap

---

### Phase 0 — Foundational Setup (1–2 weeks)

- Setup Cloud Run services
- Configure Firestore & GCS with lifecycle rules
- Setup Google AI Studio environment

- Create project structure
  - Setup LinkedIn OAuth credentials
  - Setup CI/CD pipelines
  - UX wireframes finalized
- 

## Phase 1 — MVP Build (4–6 weeks)

### Backend

- Upload → Parsing → Scoring pipeline
- Suggestions engine
- Resume generator
- Template engine
- PDF renderer
- Auto-deletion system

### Frontend

- Onboarding
- Privacy consent screen
- Upload screen
- Role input
- Dashboard
- Suggestions display

- Template selection
- Preview & download screen

## QA

- Functional testing
  - LLM schema validation
  - OAuth testing
  - Rendering tests
  - E2E flows
- 

# Phase 2 — Enhanced Personalization (3–4 weeks)

## Features

- Editable resume sections in UI
- Multi-role scoring (e.g., PM + SDE)
- More advanced template customization
- Tips with sample rewrite suggestions
- “Re-score After Improvements” button

## LLM Improvements

- Better impact quantification
- Less hallucination via schema-hardening

- Domain-specific scoring improvements
- 

## Phase 3 — Recruiter Tools (4–6 weeks)

### Features

- Bulk upload of 50+ resumes
- Candidate comparison view
- Fit ranking
- "Share Scorecard" PDF export
- Recruiter login & dashboards

### LLM Enhancements

- Team-level consistency scoring
  - Experience de-duplication
  - Score justification optimization
- 

## Phase 4 — Career Ecosystem Expansion (6–8 weeks)

### New Modules

- Cover letter generator
- LinkedIn “About” section generator

- Interview question generator
- JD-to-Resume tailoring assistant

## Integrations

- Indeed job link import
  - Naukri job link import
  - GitHub profile parsing
  - Portfolio website parsing
- 

# Phase 5 — AI-powered Job Matching Platform (8–12 weeks)

## Features

- AI-based role recommendations
- Skill-gap → learning path generation
- Candidate marketplace (opt-in)
- Recruiter search engine

This transforms the system from a **resume tool** → **full career intelligence platform**.

---

# 80. Release Strategy

---

## **Soft Launch (Private Beta)**

Invite-only for:

- 50–100 users
- Mix of job seekers & recruiters

### **Success Criteria**

- Scoring accuracy > 80% satisfaction
  - Resume generator used by > 40% users
  - LLM errors < 2%
- 

## **Public Launch (Phase 1 Complete)**

Marketing channels:

- LinkedIn content
  - Personal branding ([nayaneshgupte.com](http://nayaneshgupte.com))
  - PM/SDE groups
  - YouTube demo video
- 

## **Post-Launch Monitoring**

- Heatmaps on score dashboard
- Template usage analytics
- Resume generation conversion

---

## 81. Long-Term Vision (North Star)

The end goal is for the product to become:

**“The intelligence layer of your career — from resume to interview to job matching.”**

Where users rely on you for:

- Resume optimization
- Role matching
- Career gap analysis
- Interview coaching
- Salary research
- Job application tracking
- Recruiter ranking

This becomes the **platform moat**.

---