

ResumeATS — Complete Execution Blueprint

Executive Summary

ResumeATS is a resume builder SaaS platform targeting job seekers who need ATS-optimized, designer-quality resumes. The market is dominated by players like Enhancv (80+ templates, visual-forward), FlowCV (50+ templates, freemium-dominant), [Resume.io](#) (content-first, guided wizard), Canva (design-heavy but weak ATS), Zety (AI content suggestions), and Novoresume (polished UX). The opportunity lies in combining the best of all worlds: ATS compliance rivaling [Resume.io](#), visual quality matching Enhancv, AI depth exceeding Zety, and a scalable template architecture built on Stitch + Google Antigravity.[1][2][3][4][5]

This blueprint delivers a JSON-based modular template system supporting 50+ unique templates, an ATS optimization framework informed by how parsers like Sovren and RChilli work, a complete technical architecture, and a differentiation strategy anchored in AI-powered resume tailoring and gamified scoring.

Phase 1 — Market & Feature Deep Research

Competitor Analysis

Platform	Template sets	Pricing	Key Strengths	Weaknesses
Enhancv	80+	7-day free trial, then subscription	Visual quality, unique sections (Life Philosophy), ATS+design balance	Paywall after trial
FlowCV	50+	Free (1 resume), Pro \$10/mo	Clean text-based PDFs, granular customization, embedded fonts	Free tier limited to 1 resume
Resume.io	30+	Free plan, 7-day trial \$2.95, Monthly \$24.95	Guided wizard, AI content recommendations, job-specific suggestions	Paywall on PDF downloads
Canvas	1000+	Free + Pro	Massive template library, design flexibility	ATS-unfriendly layouts, tables/graphic s break parsers
Zety	20+	Trial \$2.70/14-day, \$23.70/mo	Pre-written phrases by job title, real-time scoring	Controversial cancellation process, 3.6★ Trustpilot
Novoresume	20+	Free (1 resume), \$19/mo	Polished interface, guided structure	Limited free tier, fewer templates

2][3][6][7][8]

Template Structure Patterns

Top-performing resume templates share consistent structural patterns:

- **Standard section headings:** "Work Experience," "Education," "Skills," "Professional Summary" — these match parser dictionaries used by ATS like Sovren and RChilli[9][10]
- **Reverse-chronological order:** Most recent experience first, which is the format ATS systems parse most reliably[11][12]
- **Contact info in main body:** Not in headers/footers, as ATS struggles to read those regions[10][9]
- **Consistent date formatting:** Clear start/end dates using standard formats[11]
- **Action-verb bullet points:** 1-2 lines each, starting with verbs, including quantified results[13]

Design Systems

Enhancv isolates sections cleanly across columns — Experience on the left, Skills/Education on the right — ensuring reading order stays coherent for both ATS and humans. FlowCV generates clean, text-based PDFs with embedded fonts and simple structure. Canva offers maximum design freedom but requires users to stick to single-column, simple layouts for ATS safety.[14][9][1]

Key design principles across leaders:

- **Color as accent, not distraction:** One primary accent color (navy, charcoal, muted jewel tone) for headings and separators[15]
- **ATS-friendly fonts:** Inter, Roboto, Poppins, Arial, Calibri, Times New Roman, Verdana — all render correctly in parsers[16][15]
- **Font size:** 10-12pt body, margins 0.5-1 inch, clear line spacing[13]
- **Visual hierarchy:** Job titles, companies, and dates must be instantly locatable without searching[11]

UX Flow Patterns

The dominant UX flow across all competitors follows this pattern:

1. **Select template** → visual gallery with category filters
2. **Add content** → guided step-by-step form OR import from LinkedIn/existing PDF
3. **Customize design** → tweak colors, fonts, layout, section order
4. **Optimize** → ATS check, AI suggestions, keyword matching
5. **Export** → PDF (primary), DOCX, TXT, shareable link

[8][17][2]

Monetization Models

The industry standard is **freemium with subscription upsell**, generating 50-60% of revenue from subscriptions:[18]

- **Free tier:** 1 resume, basic templates, watermarked or limited-format export
- **Pro tier:** \$10-25/month — unlimited resumes, all templates, AI features, PDF/DOCX export
- **Lifetime deal:** \$49-99 — everything in Pro, one-time payment
- **Template packs:** \$15-25 for 5-10 templates as impulse buys
- **Industry bundles:** \$20-30 (tech, healthcare, legal)
- **Enterprise/white-label:** \$5,000-50,000+ annually for universities, staffing agencies

Additional revenue streams include resume review services (\$29-99 per review, 50-70% margin), job board affiliate partnerships (15-25% of revenue), and API access (\$0.10-1 per call).[19][18]

Revenue trajectory benchmark: Month 12 at 8,000 users → \$6,000 MRR; Month 24 at 40,000 users → \$30,000 MRR (assuming 5% conversion, \$12 avg subscription, 15% churn).[18]

Phase 2 — Template System Architecture

Core Template Schema (JSON-Based)

The template system should separate **data** (resume content) from **presentation** (template rendering). The JSON Resume open standard is the proven foundation, with extensions for layout metadata.[20] [21]

```
{  
  "$schema": "https://resumeats.com/schema.json",  
  "version": "1.0.0",  
  "basics": {  
    "name": "string",  
    "headline": "string",  
    "email": "string",  
    "phone": "string",  
    "location": { "city": "string", "country": "string" },  
    "url": "string",  
    "profiles": [{ "network": "string", "url": "string" }]  
},  
  "summary": {  
    "title": "string",  
    "content": "string"  
},  

```

```
        }],
    },
    "education": { ... },
    "skills": {
        "title": "Skills",
        "columns": 2,
        "items": [
            {
                "name": "string",
                "level": "string",
                "keywords": ["string"]
            }
        ],
        "projects": { ... },
        "certifications": { ... },
        "languages": { ... },
        "awards": { ... },
        "publications": { ... },
        "volunteer": { ... },
        "references": { ... }
    },
    "customSections": [
        {
            "title": "string",
            "columns": 1,
            "items": [{ "id": "uuid", "content": "string" }]
        }
    ],
    "metadata": {
        "template": "modern-executive",
        "layout": "two-column",
        "theme": {
            "primary": "#2563EB",
            "secondary": "#64748B",
            "background": "#FFFFFF",
            "text": "#1E293B",
            "accent": "#0EA5E9"
        },
        "typography": {
            "heading": { "family": "Inter", "size": 14, "weight": 700 },
            "subheading": { "family": "Inter", "size": 11, "weight": 600 },
            "body": { "family": "Inter", "size": 16, "weight": 400 }
        }
    }
}
```

```
        "body": { "family": "Inter", "size": 10, "weight": 400 },
      },
      "spacing": {
        "sectionGap": 16,
        "itemGap": 8,
        "margins": { "top": 48, "right": 48, "bottom": 48, "left": 48 }
      },
      "sectionOrder": ["summary", "experience", "education", "skills"],
      "sidebarSections": ["skills", "languages", "certifications"],
      "pageSize": "A4"
    }
  }
```

This schema extends JSON Resume's standard with Reactive Resume's metadata approach for layout, theme, and typography, while keeping full ATS compliance through clean semantic structure. [21][20]

Modular Section Components

Each section is an independent, reusable component:

Component	Variants	Description
Header	Centered, Left-aligned, Split (name left / contact right), Banner	Name, title, contact info, photo optional
Summary	Full-width paragraph, Bullet-point, Two-column highlights	Professional summary / objective
Experience	Timeline, Card, Compact list, Detailed with metrics	Work history with highlights
Education	Standard, Compact, With coursework/GPA	Academic background
Skills	Tag cloud, Progress bars (visual only), Grid, Grouped by category	Technical and soft skills
Sidebar	Left sidebar, Right sidebar, None	Contains secondary sections
Timeline	Vertical line, Dot-connected, Minimal	Visual timeline for experience/education
Projects	Card grid, List, Portfolio-style	Side projects, portfolio items
Custom	Free-text, Key-value pairs, Icon list	User-defined sections

Layout Variations (7 Core Types)

- Single Column — Classic:** Full-width, maximum ATS safety, best for traditional industries (finance, law, government)
- Two Column — Professional:** 65/35 or 70/30 split, main content left, sidebar right for skills/education
- Two Column — Reverse:** Sidebar left (darker background), main content right — modern feel
- Hybrid:** Full-width header + summary, then two-column body, then full-width footer sections

5. **Timeline:** Vertical timeline element connecting experience entries, single or two-column
6. **Modern Grid:** CSS Grid-based layout with card-like sections, suitable for creative roles
7. **Minimal:** Maximum whitespace, elegant typography, no borders or heavy decoration

Color System Strategy

Define 6 color palettes, each with 5 shades:

Palette	Primary	Use Case	Tone
Corporate	Navy #1E3A5F	Finance, law, consulting	Formal, trustworthy
Professional	Charcoal #374151	General business, management	Clean, neutral
Modern	Royal Blue #2563EB	Tech, startups, engineering	Dynamic, forward
Creative	Teal #0D9488	Design, marketing, media	Fresh, distinctive
Bold	Deep Purple #7C3AED	Creative directors, strategists	Confident, unique
Soft	Sage Green #6B8F71	Healthcare, education, nonprofit	Warm, approachable

Each palette includes: primary (headings, accents), secondary (subheadings), background, text (near-black), and muted (subtle dividers).[22][15]

Typography Pairing System

Pair Name	Heading Font	Body Font	Mood
Executive	Playfair Display	Source Sans Pro	Sophisticated, editorial
Modern	Inter	Inter	Clean, tech-forward
Classic	Georgia	Arial	Traditional, universally safe
Fresh	Poppins	Open Sans	Friendly, startup-y
Minimal	Roboto Slab	Roboto	Balanced, modern
Creative	Montserrat	Lato	Bold, design-oriented

All fonts are ATS-safe when embedded in PDF.[16][15]

Customization Logic

Users can modify without breaking ATS compliance:

- **Font size:** Body 9-12pt, Headings 12-18pt (bounded range)
- **Spacing:** Section gaps, item gaps, margins — slider controls
- **Color swap:** Select palette or custom hex for primary/secondary/accent
- **Section reorder:** Drag-and-drop with constraints (header always on top)
- **Section visibility:** Toggle sections on/off
- **Layout switch:** Change template while preserving all content data
- **Column count:** Per-section control (skills in 2-col, experience in 1-col)[21]

Phase 3 — ATS Optimization Framework

What Breaks ATS Parsing

ATS Breaker	Why It Fails	Solution
Tables	ATS reads cell-by-cell, scrambling reading order	Use CSS flexbox/grid, never HTML tables
Text in images	OCR is unreliable; most ATS skip images	All text must be selectable HTML text
Headers/Footers	Many ATS ignore header/footer content	Contact info in main document body
Multi-column via text boxes	Reading order becomes unpredictable	Use CSS columns with proper DOM order
Non-standard section titles	Parser dictionaries fail on creative headings	Use "Work Experience" not "My Journey"
Fancy/icon fonts	Characters may render as symbols or blanks	Stick to standard web fonts
Embedded graphics/chars	ATS cannot interpret visual skill bars	Use text labels ("Expert", "Advanced")
PDF from image	No extractable text layer	Always generate from HTML → PDF renderer

12][9][10

HTML/PDF Structure for ATS

The key insight from Enhancv: their column structure isolates entire sections rather than splitting content across columns. The HTML should:[9]

1. **Use semantic HTML:** `<section>`, `<h2>` for section titles, `<h3>` for job titles, `` for bullet points
2. **Maintain linear DOM order:** Even in a 2-column visual layout, the HTML source should read top-to-bottom in logical order (name → summary → experience → education → skills)
3. **Use CSS Grid/Flexbox for layout:** Visual positioning via CSS, not HTML structure manipulation
4. **Embed fonts:** Include fonts directly in PDF to prevent rendering issues
5. **Tag metadata:** Include hidden semantic markers for parsers

```
<article class="resume" role="document">
  <header class="resume-header">
    <h1>John Doe</h1>
    <p class="headline">Senior Software Engineer</p>
    <address>john@email.com | +1-555-0100 | San Francisco, CA</address>
  </header>
  <section class="summary" aria-label="Professional Summary">
    <h2>Professional Summary</h2>
    <p>...</p>
  </section>
  <section class="experience" aria-label="Work Experience">
    <h2>Work Experience</h2>
    <article class="experience-item">
      <h3>Senior Engineer</h3>
      <p class="company">Google</p>
      <time>2020-01 — Present</time>
      <ul>
        <li>Led migration of 50+ microservices...</li>
      </ul>
    </article>
  </section>
</article>
```

Export Logic

PDF Export (Primary):

- Use Puppeteer (headless Chromium) for server-side HTML → PDF rendering[23][24]
- Ensures pixel-perfect reproduction of CSS layouts
- Embedded fonts, selectable text, correct reading order
- A4 and Letter page size support

DOCX Export:

- Use Docxtemplater with PizzZip for client-side or server-side generation[25]
- Map JSON resume data to pre-built .docx templates
- Simpler layouts only (single-column recommended for DOCX)
- Critical for companies that specifically require .docx format

TXT Export:

- Plain text fallback for maximum ATS compatibility
- Auto-generated from resume JSON data
- Stripped of all formatting

AI Keyword Optimization

The AI keyword engine should:

1. **Parse job description:** Extract required skills, qualifications, and key phrases
 2. **Score match rate:** Compare resume content against job requirements (target $\geq 80\%$ match)[26]
 3. **Suggest keyword insertions:** Recommend where to naturally weave missing keywords into experience bullets
 4. **Include acronym expansion:** Both "MBA" and "Master of Business Administration" for maximum reach[10]
 5. **Context-aware rewriting:** Rewrite bullets to mirror the language and priorities of the target employer, not just keyword-stuff[27]
-

Phase 4 — Technical Implementation for ResumeATS

Platform Architecture with Stitch + Google Antigravity

Google Stitch generates production-ready HTML/CSS from prompts and visual inputs, making it ideal for rapidly prototyping and building the 50+ template front-ends. It outputs Tailwind CSS code, supports iterative refinement via chat, and exports to Figma for further polish. [5][28][29]

Google Antigravity is the agentic development platform for building the full-stack application logic — backend APIs, database schema, PDF generation pipeline, AI integration, and deployment. Its autonomous agents can plan, code, test, and verify across editor, terminal, and browser simultaneously.[4][30][31]

Suggested Frontend Structure

```
resumeats/
  └── src/
    ├── app/          # Next.js 14+ App Router
    |   ├── (auth)/    # Auth routes (login, register)
    |   ├── (dashboard)/ # User dashboard
    |   |   ├── resumes/  # Resume list, create new
    |   |   ├── templates/ # Template gallery
    |   |   └── settings/ # Account settings
    |   └── editor/[id]/ # Resume editor (core experience)
    |       └── api/      # API routes
    └── components/
        ├── editor/     # Editor UI components
        |   ├── SectionEditor.tsx
        |   ├── DragDropLayout.tsx
        |   ├── ThemePicker.tsx
        |   └── FontPicker.tsx
        └── preview/     # Live preview components
            ├── ResumePreview.tsx
            └── PageRenderer.tsx
```

```

|   |   └── templates/      # Template renderers
|   |       └── ModernExecutive.tsx
|   |       └── MinimalClean.tsx
|   |       └── ...50+ templates
|   └── ui/                # Shared UI primitives
└── lib/
    ├── schemas/          # Zod validation schemas
    ├── templates/         # Template registry & configs
    ├── ai/                # AI integration (OpenAI/Gemini)
    ├── export/            # PDF/DOCX generation
    └── ats/               # ATS scoring engine
    └── store/             # Zustand or Redux state
    └── styles/            # Global styles, theme tokens
└── templates/           # Template JSON configs
    ├── modern-executive.json
    ├── minimal-clean.json
    └── ...
└── server/
    ├── pdf/              # Puppeteer PDF service
    ├── ai/                # AI API endpoints
    └── db/                # Database models

```

Use **Next.js** (App Router) for the framework, **Tailwind CSS** for styling, **Zustand** for state management, and **React-PDF** for client-side preview rendering.[32]

Data Model Structure

User

```

└── id (UUID)
└── email
└── name
└── plan (free | pro | lifetime)
└── createdAt
└── resumes[] → Resume

```

Resume

```

└── id (UUID)

```

```
|── userId (FK)
|── title
|── templateId (FK)
|── data (JSONB) → Full resume schema
|── metadata (JSONB) → Theme, typography, layout
|── status (draft | published)
|── shareSlug (unique URL)
|── views (int)
|── createdAt
|── updatedAt
└── versions[] → ResumeVersion
```

Template

```
|── id (string slug)
|── name
|── category (professional | creative | minimal | modern | academic | tech
|── layout (single | two-column | hybrid | timeline | grid | minimal | reve
|── thumbnail (URL)
|── isPremium (boolean)
|── industryTags[]
|── defaultTheme (JSONB)
|── defaultTypography (JSONB)
|── componentMap (JSONB) → Maps sections to visual components
```

ResumeVersion

```
|── id (UUID)
|── resumeId (FK)
|── data (JSONB)
|── createdAt
```

Use **PostgreSQL** with JSONB columns for the resume data and metadata, matching Reactive Resume's proven approach.[17]

Template Rendering System

Each template is a React component that receives the standardized resume JSON and renders it:

```
interface TemplateProps {  
    data: ResumeData;      // Content: experience, skills, etc.  
    metadata: ResumeMetadata; // Theme, typography, spacing, layout  
}  
  
// Template Registry  
const templateRegistry: Record<string, React.FC<TemplateProps>> = {  
    'modern-executive': ModernExecutiveTemplate,  
    'minimal-clean': MinimalCleanTemplate,  
    'creative-sidebar': CreativeSidebarTemplate,  
    // ... 50+ entries  
};  
  
// Renderer  
const ResumeRenderer = ({ templateId, data, metadata }) => {  
    const Template = templateRegistry[templateId];  
    return <Template data={data} metadata={metadata} />;  
};
```

For **live preview**: Render the template component in an iframe or React-PDF view that updates in real-time as users edit.[32]

For **PDF export**: Server-side Puppeteer renders the same HTML template to PDF with exact visual fidelity.[23]

PDF Export Approach

```
// Server-side PDF generation with Puppeteer  
const puppeteer = require('puppeteer');  
  
async function generatePDF(resumeHtml, options = {}) {  
    const browser = await puppeteer.launch({ headless: 'new' });  
    const page = await browser.newPage();
```

```

await page.setContent(resumeHtml, { waitUntil: 'networkidle0' });

const pdf = await page.pdf({
  format: options.pageSize || 'A4',
  printBackground: true,
  margin: { top: '0', right: '0', bottom: '0', left: '0' },
  preferCSSPageSize: true,
});

await browser.close();
return pdf;
}

```

For DOCX: Use Docxtemplater on the server to map resume JSON into pre-built Word templates.[25]

AI-Assisted Resume Rewriting Integration

Integrate with OpenAI GPT-4 or Google Gemini API:

1. **Content Generator:** Given job title + industry, generate professional summary and bullet points as starting suggestions[27]
2. **Bullet Rewriter:** Take raw user input ("managed a team") → transform to impact-focused ("Led a cross-functional team of 12 engineers, delivering 3 product launches ahead of schedule, resulting in \$2M incremental revenue")
3. **Job Match Analyzer:** Parse job description, extract requirements, score resume match rate, suggest specific improvements[26]
4. **Keyword Optimizer:** Identify missing keywords from job description and suggest natural insertion points[27]
5. **Tone Adjuster:** Adapt resume language for different industries (corporate formal vs. startup casual)

Template Scalability System

To efficiently maintain 50+ templates:

- **Shared base components:** All templates compose from the same section components (HeaderBlock, ExperienceBlock, SkillsBlock, etc.)
- **Template = Configuration + Layout CSS:** Each template is primarily a CSS file + a component composition map, not entirely separate code
- **Theme tokens:** Colors, fonts, spacing defined as CSS custom properties — one change propagates everywhere
- **Template inheritance:** Create "families" — e.g., "Modern" family has 5 variants (different sidebar widths, color schemes, header styles) sharing 90% of CSS

Performance Considerations

- **Lazy-load templates:** Only load the template renderer for the selected template, code-split all others
- **Debounce editor updates:** Throttle live preview re-renders to avoid jank during fast typing
- **Server-side PDF queue:** Use a job queue (Bull/BullMQ with Redis) for PDF generation to prevent server overload
- **CDN for template thumbnails:** Pre-render template preview images and serve from CDN
- **Database indexing:** Index on userId, templateId, status for fast queries
- **Edge caching:** Cache shared template configs at the edge

Phase 5 — Differentiation Strategy

AI Features That Set ResumeATS Apart

Feature	Description	Competitor Gap
AI Resume Tailoring	One-click rewrite of entire resume for a specific job posting — not just keywords, but context-aware narrative reshaping	Only Reztune does this well; most competitors do surface-level keyword matching[27]
Smart Bullet Generator	Input raw accomplishments → AI generates 3-5 variations with metrics, action verbs, and industry-appropriate language	Zety offers pre-written phrases but they're generic[27]
Resume-to-Job Match Score	Real-time percentage score showing alignment with a pasted job description, with specific fix suggestions	Rezi and Teal have this, but combined with one-click fix is rare[27][26]
AI Interview Prep	Based on the resume + job description, generate likely interview questions with suggested answers	Minimal competition in this space
LinkedIn Profile Import + Optimize	Import LinkedIn data, then AI suggests improvements for both resume and LinkedIn profile	Some competitors import but don't optimize both

Smart Content Suggestions

- **Industry-specific bullet libraries:** Pre-trained on 500+ job titles with role-specific achievement phrases
- **Metric prompts:** When a user writes a bullet without numbers, prompt "Can you quantify this? e.g., 'Increased revenue by X%'"
- **Skill gap analysis:** Compare user's skills against trending job postings in their field, suggest skills to add or learn
- **Summary writer:** Auto-generate professional summary based on experience entries, tailored to target role

Recruiter Scoring Engine

Build a proprietary "ResumeATS Score" (0-100) evaluating:

- **ATS Readiness** (30 points): Format compliance, standard headings, font safety, no tables/images
- **Content Impact** (25 points): Action verbs, quantified achievements, relevant keywords
- **Job Match** (25 points): Skill alignment, keyword density, experience relevance
- **Completeness** (10 points): All standard sections present, appropriate length
- **Readability** (10 points): Sentence length, jargon level, scanning ease

Display as a gamified dashboard with specific improvement suggestions for each dimension.[33][26]

Industry-Specific Templates

Organize templates by industry verticals:

Industry	Template Count	Key Features
Technology	10	GitHub/portfolio links, technical skills grids, project sections
Business/Finance	8	Clean, conservative layouts, metrics-heavy experience sections
Healthcare	5	Certifications prominent, licensure sections, clinical experience
Creative/Design	8	Portfolio integration, color-forward, creative section names
Academic/Research	5	Publications, teaching experience, grants, multi-page support
Legal	3	Bar admissions, case experience, conservative formatting
Entry-Level/Student	6	Education-first, projects/coursework, volunteer work
Executive	5	Board memberships, P&L responsibility, strategic achievements

Gamified Resume Scoring

Transform resume optimization into an engaging experience:

- **Progress bars:** Visual completion indicators for each section
- **Achievement badges:** "ATS Master" (score 90+), "Keyword King" (perfect job match), "Impact Writer" (all bullets have metrics)
- **Streak system:** Daily login streaks unlock premium features or template previews
- **Before/After comparison:** Show score improvement from original upload to optimized version
- **Leaderboard** (opt-in): Anonymous comparison of resume scores within industry/role

Localization Strategy

Resume standards vary significantly by region:[34]

Market	Key Differences
US/Canada	No photo, 1-2 pages, skills-focused
Europe (DACH)	Photo expected, "Lebenslauf" format, personal details included
UK	"CV" terminology, 2 pages standard, references section common
Middle East/Asia	Photo common, personal details expected, longer formats accepted
Latin America	Photo and personal details standard, formal tone

Implementation: AI-powered translation + cultural tone adjustment, region-specific template defaults, localized section labels.[35][34]

50-Template Categorization Framework

Category 1: Professional (12 templates)

1. **Executive Classic** — Single column, serif headings, navy accents
2. **Corporate Edge** — Two-column, charcoal sidebar, clean lines
3. **Business Pro** — Hybrid layout, subtle gray tones
4. **Finance First** — Single column, minimal color, metrics-focused
5. **Consulting Ready** — Two-column, structured sections, bullet-heavy
6. **Legal Brief** — Single column, conservative, serif fonts
7. **Management Suite** — Hybrid, leadership-focused sections
8. **Board Director** — Single column, executive summary prominent
9. **Senior Advisor** — Two-column reverse, professional green
10. **Operations Lead** — Hybrid, process-oriented layout
11. **Sales Champion** — Two-column, metrics dashboard style
12. **HR Professional** — Single column, warm tones, people-first

Category 2: Modern (10 templates)

- 13. **Modern Minimal** — Single column, lots of whitespace, Inter font
- 14. **Tech Forward** — Two-column, blue accents, skills grid
- 15. **Startup Ready** — Hybrid, bold headers, project-focused
- 16. **Digital Native** — Two-column reverse, tech icons, dark mode option
- 17. **Product Builder** — Timeline layout, milestone-focused
- 18. **Data Driven** — Grid layout, metrics cards, visualization-ready
- 19. **Cloud Engineer** — Two-column, technical skills matrix
- 20. **Agile Sprint** — Hybrid, kanban-inspired section cards
- 21. **Full Stack** — Two-column, language/framework tags
- 22. **DevOps Pipeline** — Timeline, tool icons as text labels

Category 3: Creative (8 templates)

- 23. **Design Studio** — Grid layout, portfolio section, teal accents
- 24. **Creative Director** — Two-column reverse, bold purple, large name
- 25. **Brand Maker** — Hybrid, color blocks, marketing-focused
- 26. **Visual Storyteller** — Timeline with accent colors, narrative flow
- 27. **Media Pro** — Two-column, content portfolio, vibrant layout
- 28. **Art Director** — Grid, creative sections, unique typography
- 29. **UX Designer** — Two-column, process-focused, clean pastels
- 30. **Content Creator** — Hybrid, engagement metrics, platform links

Category 4: Minimal (6 templates)

- 31. **Pure Minimal** — Single column, black/white, maximum whitespace
- 32. **Elegant Simple** — Single column, serif accent, subtle lines
- 33. **Clean Slate** — Two-column, hairline dividers, no color
- 34. **Swiss Design** — Grid-based, geometric precision, monochrome
- 35. **Zen Resume** — Single column, generous spacing, sage green
- 36. **Type Focus** — Single column, typography-driven, no icons

Category 5: Academic (5 templates)

- 37. **Research Scholar** — Multi-page, publications section, footnote style
- 38. **PhD Candidate** — Single column, grants & publications prominent
- 39. **Professor** — Two-column, teaching + research split
- 40. **Postdoc** — Hybrid, lab skills, conference presentations
- 41. **Academic CV** — Multi-page, comprehensive, traditional layout

Category 6: Entry-Level (5 templates)

- 42. **Fresh Graduate** — Education-first, projects prominent, blue accents
- 43. **Intern Ready** — Single column, skills-forward, coursework section
- 44. **Career Starter** — Hybrid, volunteer + activities, green tones
- 45. **Student Portfolio** — Two-column, project cards, modern layout
- 46. **First Job** — Single column, simple, guided content prompts

Category 7: Industry-Specific (4 templates)

- 47. **Healthcare Hero** — Single column, certifications prominent, blue/teal
- 48. **Legal Eagle** — Single column, bar admissions, conservative serif
- 49. **Teacher's Edition** — Two-column, classroom achievements, warm colors
- 50. **Federal Resume** — Multi-page, government format, comprehensive

Bonus Templates (2)

- 51. **ATS Maximum** — Plain single column, zero design, maximum parse-ability
 - 52. **Cover Letter Match** — Companion cover letter template that matches any selected resume template
-

Actionable Next Steps Roadmap

Phase 1: Foundation (Weeks 1-4)

- [] Set up Next.js project with Tailwind CSS using Google Antigravity agents[4]
- [] Define and validate JSON resume schema (extend JSON Resume standard)[20]
- [] Set up PostgreSQL database with Prisma ORM
- [] Build user authentication (OAuth with Google/GitHub + email)
- [] Create 5 core template components (Header, Experience, Education, Skills, Summary)
- [] Implement basic resume editor with live preview

Phase 2: Template Engine (Weeks 5-8)

- [] Use Google Stitch to rapidly generate HTML/CSS for first 15 template layouts[5]
- [] Refine templates in Antigravity with responsive CSS, embedded fonts
- [] Build template registry and switching system
- [] Implement color picker, font picker, spacing controls
- [] Add drag-and-drop section reordering
- [] Build template gallery with category filters and previews

Phase 3: Export & ATS (Weeks 9-11)

- [] Set up Puppeteer-based PDF generation service[23]
- [] Implement DOCX export via Docxtemplater[25]
- [] Build ATS scoring engine (format checker + keyword analyzer)
- [] Add TXT export for maximum compatibility
- [] Test exports against major ATS systems (Greenhouse, Lever, Workday, iCIMS)

Phase 4: AI Integration (Weeks 12-14)

- [] Integrate OpenAI/Gemini API for content generation
- [] Build AI bullet rewriter
- [] Implement job description parser + match scoring
- [] Add smart content suggestions and keyword optimization

- [] Build AI-powered professional summary generator

Phase 5: Scale to 50 Templates (Weeks 15-18)

- [] Use Stitch to batch-generate remaining 35+ template layouts[5]
- [] Create industry-specific template packs
- [] Build template families (variants from base templates)
- [] Add LinkedIn import functionality
- [] Implement resume version history

Phase 6: Monetization & Launch (Weeks 19-22)

- [] Implement Stripe payment integration (free, pro, lifetime tiers)
- [] Build paywall logic (free = 1 resume + basic templates, pro = unlimited)
- [] Add shareable resume links with view tracking
- [] Set up analytics dashboard
- [] Beta launch with 50 templates
- [] Gather user feedback, iterate on UX

Phase 7: Growth Features (Weeks 23+)

- [] Add gamified resume scoring dashboard
- [] Build cover letter builder with matching templates
- [] Implement multi-language support and localization[35][34]
- [] Add resume review marketplace
- [] Build job board affiliate integrations
- [] Enterprise/white-label offering for universities and career centers[18]

This blueprint provides a complete, implementation-ready foundation for building ResumeATS into a competitive SaaS product. The combination of Stitch for rapid UI generation and Antigravity for full-stack agentic development creates a uniquely efficient build pipeline that can compress what would typically be 6-12 months of development into 4-5 months.[4][5]