Техническое задание и Архитектура: AI-First IT Resume Builder

1. Обзор системы

Продукт представляет собой SaaS-платформу, где пользователь управляет своей карьерной историей через "Мастер-профиль" и генерирует множество адаптированных (Tailored) версий резюме под конкретные вакансии, используя генеративный ИИ.

2. Пользовательские пути (User Flows)

Flow 1: Онбординг и Создание Мастер-профиля

Цель: Получить максимально полные данные о пользователе с минимальными усилиями.

1. **Вход:** Лендинг -> Кнопка "Start Free" -> Auth (Google/Email).
2. **Выбор метода импорта:**
   - *Вариант А (Парсинг):* Drag & Drop старого PDF/Docx -> Бэкенд парсит текст -> Заполнение полей.
   - *Вариант Б (С нуля):* Пустые формы.
3. **Редактирование (Split-View):**
   - Пользователь видит слева форму, справа — превью.
   - Заполняет блоки: Summary, Experience, Skills, Education.
   - **AI-Trigger:** Пользователь пишет "Разрабатывал API". Нажимает "✨ AI Refine".
   - **AI-Action:** Текст заменяется на "Designed and implemented RESTful APIs using FastAPI, reducing response time by 20%".
4. **Сохранение:** Данные пишутся в `Master_Profile`.

Flow 2: Адаптация под вакансию (Tailoring Loop) — Core Feature

Цель: Создать уникальную версию резюме и Cover Letter под конкретную вакансию.

1. **Дашборд:** Пользователь нажимает "New Job Application".
2. **Ввод данных вакансии:**
   - Вставляет текст вакансии (JD) или URL.
   - Указывает название компании (напр. "Google") и позицию ("Senior Backend Dev").
3. **Анализ (AI Agent):**
   - Система анализирует JD и выделяет Keywords (ключевые навыки).
   - Система сканирует `Master_Profile` пользователя.
4. **Генерация:**
   - Создается объект `Tailored_Resume`.
   - AI переписывает *Summary* (подсвечивая опыт, нужный в JD).
   - AI сортирует *Skills* (вынося релевантные наверх).

- AI генерирует *Cover Letter*.

5. **Ревью:** Пользователь видит результат в Split-View. Может внести ручные правки (они не затрагивают Мастер-профиль).
6. **Экспорт:** Скачивание PDF.

3. Детальные Use Cases (Функциональные требования)

3.1. Управление профилем

- **UC-01 Import Resume:** Система должна уметь извлекать текст из PDF/DOCX и распределять его по секциям (Contact, Exp, Edu).
- **UC-02 PII Masking:** Перед отправкой данных в LLM, система должна автоматически заменять Email, Телефон и Адрес на токены `[PII_EMAIL]`, `[PII_PHONE]` для защиты приватности.

3.2. Редактор (The Editor)

- **UC-03 Live Preview:** Любое изменение в левой панели (Input) должно мгновенно (debounce < 500ms) отражаться на рендере PDF справа.
- **UC-04 AI Bullet Points:** При выделении буллита в опыте работы, система предлагает 3 варианта перефразирования (Action Verbs + Metrics).
- **UC-05 Language Toggle:** Переключатель RU/EN. При смене языка AI переводит контент текущей секции, сохраняя форматирование.

3.3. Адаптация (Tailoring)

- **UC-06 Job Parsing:** Извлечение из текста вакансии: Role, Company Name, Required Hard Skills, Required Soft Skills.
- **UC-07 Gap Analysis:** (Future) Система подсвечивает навыки из вакансии, которых нет в профиле пользователя ("Missing Skills").

3.4. Экспорт

- **UC-08 PDF Generation:** Генерация файла A4. Важно: контроль "висячих строк" и разрывов страниц (Page Breaks) не должны разрывать блоки опыта посередине.

4. Архитектура системы

4.1. High-Level Stack

- **Client (SPA):** React 18 (Next.js App Router), TypeScript, Tailwind CSS.
  - *State:* Zustand (хранение состояния формы редактирования).
  - *PDF:* `@react-pdf/renderer` (рендеринг React-компонентов в PDF на клиенте).
- **API Gateway / Backend:** Python (FastAPI).
  - Выбран Python из-за мощных библиотек для AI (LangChain, OpenAI SDK) и парсинга (PyPDF2).
- **Database:** PostgreSQL (Supabase или Managed Postgres).

- **Storage:** AWS S3 (или аналог) для хранения загруженных исходников и сгенерированных PDF (опционально, можно генерить на лету).
- **AI Orchestration:** Модуль внутри бэкенда, управляющий промптами и контекстом.

4.2. Схема данных (ERD - Conceptual)

1. **Users**
   - `id` (UUID)
   - `email`, `password_hash`
   - `subscription_tier` (Free/Pro)
2. **MasterProfiles** (Связь 1:1 с User)
   - `id`
   - `user_id`
   - `personal_info` (JSON: name, contacts)
   - `experience` (JSONB Array: company, role, dates, bullets[])
   - `education` (JSONB Array)
   - `skills` (JSONB Array)
   - `projects` (JSONB Array)
3. **JobApplications** (Отклики)
   - `id`
   - `user_id`
   - `company_name`
   - `job_description_text` (Text)
   - `status` (Draft, Applied, Interview)
4. **TailoredResumes** (Связь 1:1 с JobApplication)
   - `id`
   - `job_application_id`
   - `base_profile_id` (FK to MasterProfile)
   - `overrides` (JSONB: хранит только те поля, которые отличаются от MasterProfile. Например, переписанное Summary).

4.3. Интеграция с AI (Sequence)

Сценарий: Пользователь просит улучшить опыт работы

1. **Frontend:** Отправляет POST `/api/ai/refine` с телом: `{ text: "Писал код на Java", context: "Senior Developer", locale: "ru" }`.
2. **Backend (FastAPI):**
   - Валидирует токен пользователя.
   - Вызывает `PII_Scrubber` (удаляет личные данные, если они попали в текст).
   - Формирует промпт для LLM (System Prompt + User Input).

- Отправляет запрос в OpenAI (`gpt-4o-mini`).

3. **LLM:** Возвращает: "Разрабатывал масштабируемые энтерпрайз-приложения на Java..."
4. **Backend:** Возвращает JSON клиенту.
5. **Frontend:** Показывает диф (было/стало) и кнопку "Применить".

5. Требования к API (Endpoints Draft)

Auth & User

- `POST /auth/register`
- `GET /users/me`

Profile Management

- `GET /profile/master` — Получить мастер-профиль.
- `PUT /profile/master` — Обновить разделы профиля.
- `POST /profile/parse` — Загрузка PDF для парсинга.

Tailoring & AI

- `POST /jobs/create` — Создать новый контекст (вакансию).
- `POST /ai/analyze-job` — Анализ вакансии (вытащить скиллы).
- `POST /ai/generate-summary` — Генерация Summary на основе Профиля + Вакансии.
- `POST /ai/refine-bullet` — Улучшение конкретного пункта.
- `POST /ai/cover-letter` — Генерация письма.

6. Безопасность и ограничения

1. **Rate Limiting:** Ограничение кол-ва запросов к AI API (например, 50 запросов в минуту на юзера), чтобы избежать скликивания бюджета.
2. **Data Retention:** Пользователь может удалить аккаунт — удаляются все данные из БД (GDPR compliance).
3. **PDF Rendering:** Рендеринг происходит на клиенте (`@react-pdf/renderer`), что снижает нагрузку на сервер и повышает приватность (финальный PDF не обязательно передавать на бэкенд).