**Onboarding & HireAI - Complete Technical Documentation**

**Table of Contents**

1. System Overview
2. Architecture
3. Frontend Application
4. Backend API
5. AI Features
6. Database Schema
7. Deployment
8. API Endpoints
9. Workflow
10. Security & Privacy

**1. System Overview**

**Onboarding & HireAI** is a comprehensive AI-powered recruitment platform that automates the entire hiring process from application screening to technical and HR interviews. The system leverages multiple AI technologies to evaluate candidates objectively and efficiently.

**Live URLs**

* **Frontend Application**: <https://hr-genie-z5zb.vercel.app/>
* **Backend API**: <https://hrgenie-production.up.railway.app/>

**Technology Stack**

| Component | Technology |
| --- | --- |
| **Frontend** | React, TypeScript, Tailwind CSS, Vercel |
| **Backend** | Node.js, Express, TypeScript, Railway |
| **Database** | PostgreSQL with Supabase |
| **AI Services** | Google Gemini AI, GitHub API |
| **File Processing** | PDF.js, Cheerio |
| **Authentication** | Supabase Auth |

**2. Architecture**

**System Architecture Diagram**

text

┌─────────────────┐ ┌──────────────────┐ ┌─────────────────┐

│ Frontend │ │ Backend API │ │ AI Services │

│ (Vercel) │────│ (Railway) │────│ (Gemini AI) │

│ │ │ │ │ │

│ React App │ │ Express Server │ │ Google Gemini │

│ TypeScript │ │ TypeScript │ │ GitHub API │

└─────────────────┘ └──────────────────┘ └─────────────────┘

│ │ │

│ │ │

└────────────────────────┼────────────────────────┘

│

┌────────┴────────┐

│ Database │

│ (Supabase) │

│ PostgreSQL │

└─────────────────┘

**Data Flow**

1. **Candidate Application** → Frontend Form Submission
2. **Resume Processing** → Backend API → AI Analysis
3. **HR Interview** → Real-time AI Conversation
4. **Technical Assessment** → AI-generated Tests
5. **Final Evaluation** → Combined AI Scoring

**3. Frontend Application**

**Key Pages & Components**

**3.1 ApplyPage (/apply/:id)**

**Purpose**: Multi-step candidate application form

**Features**:

* 3-step progressive form
* Resume upload with validation (PDF/DOCX, max 5MB)
* Professional information collection
* Education and work experience tracking
* Real-time form validation

**Steps**:

1. **Basic Information**: Phone, LinkedIn, GitHub
2. **Education & Experience**: College, GPA, degree, previous roles
3. **Resume Upload**: File validation and submission

**3.2 HRInterviewPage (/interview/:id)**

**Purpose**: AI-powered HR interview interface

**Features**:

* Real-time speech recognition
* Webcam proctoring with face detection
* Full-screen mode enforcement
* 15-minute timer with warnings
* Automatic question generation (5 questions max)
* Proctoring event tracking

**Technical Implementation**:

* **Speech Recognition**: Web Speech API
* **Face Detection**: face-api.js
* **State Management**: React hooks with ref synchronization
* **Proctoring**: Tab switch detection, face presence monitoring

**3.3 Dashboard**

**Purpose**: Application management and tracking

**Features**:

* Application status monitoring
* Interview results and scores
* Candidate profile management
* Progress tracking

**Frontend Technologies**

* **Framework**: React 18 with TypeScript
* **Styling**: Tailwind CSS
* **Routing**: React Router DOM
* **State Management**: React Hooks (useState, useEffect, useCallback)
* **HTTP Client**: Fetch API
* **UI Components**: Custom components with Lucide icons

**4. Backend API**

**Server Configuration**

typescript

// Server Setup

const app = express();

app.use(cors({

origin: allowedOrigins,

credentials: true

}));

app.use(express.json({ limit: "10mb" }));

**Core Dependencies**

| Package | Purpose |
| --- | --- |
| express | Web server framework |
| @google/generative-ai | Gemini AI integration |
| pdfjs-dist | PDF text extraction |
| @octokit/rest | GitHub API client |
| cheerio | HTML parsing for link validation |
| axios | HTTP requests for external APIs |
| cors | Cross-origin resource sharing |
| dotenv | Environment variable management |

**Environment Variables**

env

GOOGLE\_API\_KEY=your\_gemini\_api\_key

GITHUB\_ACCESS\_TOKEN=your\_github\_token

NODE\_ENV=production

PORT=3001

**5. AI Features**

**5.1 Resume Analysis & Scoring**

**Endpoint**: POST /api/analyze-profile

**AI Technology**: Google Gemini 2.5 Flash Lite

**Process Flow**:

1. **PDF Text Extraction**: Uses PDF.js to parse resume content
2. **Link Discovery**: Identifies and extracts hyperlinks from resume
3. **Link Validation**: Scrapes linked content using Cheerio
4. **AI Analysis**: Gemini evaluates resume against job requirements
5. **Scoring**: Generates 0-100 score with detailed analysis

**Output Structure**:

json

{

"resume\_score": 85,

"resume\_analysis": {

"strengths": ["Strong technical skills", "Relevant project experience"],

"weaknesses": ["Limited leadership experience", "Gaps in employment"],

"red\_flags": ["Unverifiable project claims", "Inconsistent dates"]

}

}

**5.2 GitHub Profile Analysis**

**Endpoint**: POST /api/analyze-profile

**AI Technology**: Google Gemini 2.5 Flash Lite + GitHub API

**Process Flow**:

1. **Profile Fetching**: Retrieves user data from GitHub API
2. **Repository Analysis**: Examines last 5 pushed repositories
3. **Activity Assessment**: Evaluates commit history and project maintenance
4. **Quality Evaluation**: Assesses code quality through project descriptions
5. **Scoring**: Generates 0-100 GitHub proficiency score

**Analysis Criteria**:

* Project originality vs tutorial forks
* Documentation quality
* Code activity and maintenance
* Relevance to job position
* Technical stack alignment

**5.3 AI HR Interview Question Generation**

**Endpoint**: POST /api/generate-response

**AI Technology**: Google Gemini 2.5 Flash Lite

**Features**:

* **Contextual Awareness**: Maintains conversation flow using transcript history
* **Adaptive Questioning**: Adjusts questions based on response depth
* **Competency Coverage**: Ensures comprehensive skill assessment
* **Progressive Disclosure**: Builds upon previous answers
* **Anti-Repetition**: Avoids duplicate questions

**Questioning Strategy**:

* Detailed answers → New competency areas
* Short/vague answers → Clarifying follow-ups
* Behavioral focus → Situation-based questions
* Technical alignment → Role-specific scenarios

**5.4 HR Interview Final Evaluation**

**Endpoint**: POST /api/generate-response (with isFinalAnalysis: true)

**AI Technology**: Google Gemini 2.5 Flash Lite

**Evaluation Factors**:

* Communication skills assessment
* Response quality and depth
* Professionalism and articulation
* Proctoring behavior analysis
* Overall interview performance

**Proctoring Integration**:

* Tab switching detection and counting
* Face presence monitoring
* Behavioral penalty application
* Integrity verification

**5.5 Technical Assessment Generation**

**Endpoint**: POST /api/generate-technical-assessment

**AI Technology**: Google Gemini 2.5 Flash Lite

**Assessment Components**:

**Aptitude Questions (10 items)**:

* Multiple-choice format
* Logical reasoning tests
* Problem-solving scenarios
* Technical knowledge verification

**Coding Challenges (2 items)**:

* Language-specific starter code (JavaScript, Python, Java)
* Problem description and requirements
* Difficulty progression (easy to hard)
* Real-world scenario simulation

**5.6 Code Analysis & Evaluation**

**Endpoint**: POST /api/analyze-code

**AI Technology**: Google Gemini 2.5 Flash Lite

**Analysis Dimensions**:

| Dimension | Evaluation Criteria |
| --- | --- |
| **Quality** | Readability, structure, naming conventions, best practices |
| **Correctness** | Logical accuracy, edge case handling, algorithm validity |
| **Complexity** | Time efficiency (Big O), space optimization, scalability |
| **Suggestions** | Specific improvements, optimizations, better approaches |

**Output**:

json

{

"quality": "Well-structured with good variable naming",

"correctness": "Handles edge cases appropriately",

"complexity": "O(n) time complexity, efficient solution",

"suggestions": "Consider using hash map for better performance"

}

**5.7 Technical Interview Scoring**

**Endpoint**: POST /api/finish-technical-interview

**AI Technology**: Google Gemini 2.5 Flash Lite + Custom Algorithm

**Scoring Algorithm**:

typescript

// Aptitude Questions (1 point each)

const aptitudeWeight = 1;

// Coding Challenges (5 points each)

const codingWeight = 5;

// Scoring Calculation

let rawScore = correctAptitudeCount \* aptitudeWeight;

codingSubmissions.forEach(sub => {

if (sub.code?.trim()) rawScore += codingWeight \* 0.6; // Attempt

if (sub.passedAllTests) rawScore += codingWeight \* 1.0; // Success

});

// Proctoring Penalties

if (tabSwitches > 2) technicalScore -= 10;

if (copyEvents > 0 || pasteEvents > 0) technicalScore -= 10;

**5.8 Intelligent Transcript Analysis**

**Process**: Embedded in HR and technical evaluations

**AI Technology**: Google Gemini 2.5 Flash Lite

**Analysis Capabilities**:

* Natural language processing of responses
* Key competency extraction and mapping
* Communication pattern recognition
* Skill and experience identification
* Cultural fit assessment through language analysis

**5.9 Adaptive Questioning System**

**Process**: Integrated throughout HR interview flow

**AI Technology**: Google Gemini 2.5 Flash Lite

**Adaptive Features**:

* **Response Depth Detection**: Identifies detailed vs superficial answers
* **Competency Gap Analysis**: Targets unexplored skill areas
* **Progressive Complexity**: Increases question difficulty gradually
* **Contextual Follow-ups**: Generates relevant clarifying questions
* **Conversation Flow Maintenance**: Ensures natural interview progression

**5.10 Multi-dimensional Candidate Profiling**

**Process**: Combines all AI assessment data

**AI Technology**: Google Gemini 2.5 Flash Lite + Custom Scoring

**Profile Dimensions**:

| Dimension | Data Sources | Weight |
| --- | --- | --- |
| **Technical Skills** | Resume analysis, coding tests | 40% |
| **Problem-Solving** | Aptitude tests, code analysis | 30% |
| **Communication** | HR interview evaluation | 20% |
| **Professionalism** | Proctoring data, response quality | 10% |

**6. Database Schema**

**6.1 Applications Table**

sql

CREATE TABLE onboarding.applications (

id uuid PRIMARY KEY DEFAULT gen\_random\_uuid(),

profile\_id uuid REFERENCES onboarding.profiles(id),

job\_id uuid REFERENCES onboarding.jobs(id),

-- Application Status

status text DEFAULT 'submitted',

-- AI Analysis Fields

resume\_url text,

resume\_score integer DEFAULT 0,

resume\_analysis jsonb DEFAULT '{}',

github\_score integer DEFAULT 0,

github\_analysis jsonb DEFAULT '{}',

-- Interview Data

technical\_interview\_transcript jsonb DEFAULT '[]',

technical\_interview\_score integer DEFAULT 0,

hr\_interview\_transcript jsonb DEFAULT '[]',

hr\_interview\_summary text,

hr\_interview\_score integer DEFAULT 0,

-- Candidate Information

education jsonb DEFAULT '{}',

experience jsonb DEFAULT '{}',

full\_name text,

phone text,

linkedin\_url text,

github\_url text,

-- Timestamps

created\_at timestamptz DEFAULT now(),

updated\_at timestamptz DEFAULT now()

);

**6.2 Profiles Table**

sql

CREATE TABLE onboarding.profiles (

id uuid PRIMARY KEY REFERENCES auth.users(id),

email text UNIQUE NOT NULL,

full\_name text NOT NULL,

phone text,

linkedin\_url text,

github\_url text,

created\_at timestamptz DEFAULT now(),

updated\_at timestamptz DEFAULT now()

);

**6.3 Jobs Table**

sql

CREATE TABLE onboarding.jobs (

id uuid PRIMARY KEY DEFAULT gen\_random\_uuid(),

title text NOT NULL,

description text NOT NULL,

requirements jsonb DEFAULT '{}',

department text NOT NULL,

location text NOT NULL,

employment\_type text DEFAULT 'Full-time',

salary\_range text,

is\_active boolean DEFAULT true,

created\_at timestamptz DEFAULT now(),

updated\_at timestamptz DEFAULT now()

);

**6.4 Coding Challenges Table**

sql

CREATE TABLE onboarding.coding\_challenges (

id uuid PRIMARY KEY DEFAULT gen\_random\_uuid(),

title text NOT NULL,

description text NOT NULL,

difficulty text CHECK (difficulty IN ('easy', 'medium', 'hard')),

topic text NOT NULL,

test\_cases jsonb DEFAULT '[]',

starter\_code jsonb DEFAULT '{}',

solution text,

created\_at timestamptz DEFAULT now()

);

**7. Deployment**

**7.1 Frontend Deployment (Vercel)**

**Platform**: Vercel  
**Configuration**:

* Automatic GitHub integration
* Zero-configuration deployment
* Global CDN distribution
* Automatic SSL certification
* Environment variable management

**Build Process**:

* Automatic React detection
* Optimized production builds
* Asset compression and caching

**7.2 Backend Deployment (Railway)**

**Platform**: Railway  
**Configuration**:

json

{

"buildCommand": "npm install --production=false && tsc",

"startCommand": "node dist/index.js",

"source": "packages/onboarding/server"

}

**Environment**:

* Node.js 22.16.0 runtime
* Automatic dependency installation
* Health check monitoring
* Log aggregation and monitoring

**7.3 Database (Supabase)**

**Platform**: Supabase  
**Features**:

* PostgreSQL database
* Row-level security
* Real-time subscriptions
* Automated backups
* Performance monitoring

**8. API Endpoints**

**8.1 Core AI Endpoints**

**Profile Analysis**

http

POST /api/analyze-profile

Content-Type: application/json

{

"resumeUrl": "https://example.com/resume.pdf",

"githubUrl": "https://github.com/username",

"jobTitle": "Software Engineer",

"jobDescription": "Full job description here..."

}

**HR Interview Management**

http

POST /api/generate-response

Content-Type: application/json

{

"jobTitle": "Product Manager",

"jobDescription": "Job description text...",

"previousTranscript": [

{"role": "assistant", "content": "Question 1", "timestamp": "..."},

{"role": "user", "content": "Answer 1", "timestamp": "..."}

],

"proctoringEvents": [

{"type": "tab\_switch\_away", "timestamp": "..."}

],

"isFinalAnalysis": false

}

**Technical Assessment**

http

POST /api/generate-technical-assessment

Content-Type: application/json

{

"jobTitle": "Frontend Developer",

"jobDescription": "Job description..."

}

**Code Analysis**

http

POST /api/analyze-code

Content-Type: application/json

{

"code": "function solution(arr) { return arr.sort(); }",

"language": "javascript",

"challenge": {

"title": "Array Sorting",

"description": "Sort the array in ascending order"

}

}

**Technical Interview Completion**

http

POST /api/finish-technical-interview

Content-Type: application/json

{

"jobTitle": "Backend Engineer",

"transcript": [...interview\_data...],

"proctoringEvents": [...proctoring\_events...]

}

**8.2 Response Formats**

**Success Response**:

json

{

"resume\_score": 85,

"resume\_analysis": {...},

"github\_score": 78,

"github\_analysis": {...}

}

**Error Response**:

json

{

"message": "Failed to analyze profile",

"detail": "Specific error details"

}

**9. Workflow**

**9.1 Complete Candidate Journey**

**Phase 1: Application & Screening**

1. **Candidate Application**
   * Multi-step form completion
   * Resume upload and validation
   * Basic information collection
2. **AI Resume Screening**
   * Automated resume parsing and analysis
   * GitHub profile evaluation (if provided)
   * Initial scoring and assessment
3. **Application Status Update**
   * Database record creation
   * Status: submitted → profile\_analyzed

**Phase 2: HR Interview**

1. **AI-Powered Interview**
   * Real-time speech recognition
   * Adaptive question generation
   * Behavioral assessment
2. **Proctoring & Monitoring**
   * Face detection and tracking
   * Tab switching detection
   * Session integrity verification
3. **Evaluation & Scoring**
   * Communication skills assessment
   * Response quality analysis
   * Professionalism evaluation
   * Status: hr\_screening

**Phase 3: Technical Assessment**

1. **Customized Testing**
   * Role-specific aptitude questions
   * Coding challenges with multiple languages
   * Real-world problem solving
2. **Code Analysis**
   * Automated code quality assessment
   * Performance and efficiency evaluation
   * Best practices verification
3. **Comprehensive Scoring**
   * Technical proficiency measurement
   * Problem-solving ability assessment
   * Status: technical\_review

**Phase 4: Final Evaluation**

1. **Combined Assessment**
   * Integration of all evaluation data
   * Overall candidate scoring
   * Strengths and weaknesses analysis
2. **Hiring Recommendation**
   * Data-driven decision support
   * Comparative analysis
   * Status: completed

**9.2 Status Flow**

text

submitted → profile\_analyzed → hr\_screening → technical\_review → completed

**10. Security & Privacy**

**10.1 Data Protection**

* **API Security**: Environment variable protection
* **CORS Configuration**: Restricted origin access
* **File Validation**: Type and size verification
* **Input Sanitization**: Request validation and cleaning

**10.2 AI Ethics & Compliance**

* **Bias Mitigation**: Standardized evaluation criteria
* **Transparency**: Clear scoring methodology disclosure
* **Candidate Consent**: Explicit permission for AI analysis
* **Data Retention**: Controlled storage periods
* **Privacy Protection**: Secure data handling procedures

**10.3 Monitoring & Auditing**

* **API Logging**: Request and response tracking
* **Error Monitoring**: System failure detection
* **Performance Metrics**: Response time monitoring
* **Security Audits**: Regular vulnerability assessments

**Conclusion**

The **Onboarding & HireAI** platform represents a cutting-edge approach to automated recruitment, leveraging multiple AI technologies to provide objective, efficient, and comprehensive candidate evaluation. The system's modular architecture, comprehensive AI features, and robust security measures make it a scalable solution for modern hiring processes.

**Key Advantages**:

* **Objective Assessment**: AI-driven evaluation reduces human bias
* **Efficiency**: Automated processes speed up hiring timelines
* **Comprehensive**: Multi-dimensional candidate profiling
* **Scalable**: Cloud-native architecture supports growth
* **Secure**: Enterprise-grade security and privacy protections

This documentation provides complete technical specifications for development, deployment, and maintenance of the Onboarding & HireAI platform.