Technothon 2025, Resume Review Tool – Initial Project Plan

Product Identity

Vision: AI-powered resume screening tool that empowers busy hiring managers while providing HR with actionable insights into the hiring process.

Core Problem: HR posts job requirements but overwhelmed project/product managers (acting as hiring managers) must manually review hundreds of resumes on top of their regular responsibilities. The current process creates bottlenecks, inconsistent evaluation, and frustrated stakeholders.

Problems We Face

Current State Pain Points

- **Overload**: Project and product managers must manually review resumes while managing deadlines. Work is divided among others with the same issues.
- **Inconsistent Evaluation**: Different people apply different criteria and standards
- **HR Disconnect**: Job Posts with restricted time and/or applicant volume limits could result in quality candidates being missed.
    - This could easily happen with automated applications with partial matching criteria across job boards and AI.
- **Bias and Blind Spots**: Manual review introduces unconscious bias that may also eliminate qualified candidates
- **Time Bottlenecks**: Days/weeks to get through candidate reviews, slowing hiring velocity, and increasing costs.
- **No Learning Loop**: HR can't improve job postings based on actual hiring outcomes

Success Metrics for Our Solution

- Reduce manager review time from hours to minutes per batch
- Increase consistency in candidate evaluation across teams
- Provide HR visibility into hiring manager preferences and bottlenecks
- Enable data-driven job posting optimization

Technical Architecture

Initial Tech Stack

- **Backend:** Java with Spring Boot + Spring AI
- **Frontend:** Angular (mobile-responsive for managers)
- **Database:** PostgreSQL
- **AI Platform:** AWS Bedrock (Claude Sonnet 4 + other models)
- **File Processing:** Apache Tika for multi-format resume parsing
- **Others?**

Core System Components

Spring Boot + Spring AI AWS Bedrock
 Resume Processing Service (AI-powered parsing)
 Job Template Service (AI-assisted creation)
 Intelligent Matching Engine (semantic analysis)
 Human Review Interface (score overrides + intangibles)
 Bias Detection Service (AI monitoring)
 Analytics Engine (HR insights dashboard)

AI Integration Points

1. **Resume Parsing**: Claude extracts structured data from unstructured resume PDFs
2. **Semantic Matching**: AI understands "React development" relates to "frontend frameworks"
3. **Explainable Scoring**: Claude provides reasoning for each candidate assessment
4. **Bias Detection**: AI flags potentially problematic language in job descriptions
5. **Natural Language Queries**: "Show me candidates with strong React skills but junior experience"

AI Choices and Usage Strategy

AI as Development Tool

- **Code Generation**: Claude generates Spring Boot boilerplate, Angular components, test cases
- **Documentation**: Auto-generate API specs and user guides

- **Test Data**: AI creates realistic resume samples for testing
- **Prompt Engineering**: Iterative AI-assisted prompt optimization

AI in Final Product

- **Primary Scoring Engine**: Claude Sonnet 4 performs initial candidate assessment (75% weight)
- **Dynamic Job Templates**: Natural language job creation - “Create template for senior Java developer”
- **Contextual Understanding**: AI grasps nuanced skill relationships and experience levels
- **Continuous Learning**: AI adapts scoring based on hiring manager feedback patterns

Hybrid Scoring Model

Final Score = AI Assessment (75%) + Human Intangibles (25%)

- **AI Component**: Technical skills, experience match, qualification alignment
- **Human Component**: Cultural fit, leadership potential, communication style, growth mindset, etc.

48-Hour Sprint Plan

MUST BUILD (Core Demo - 36 hours)

1. **Spring Boot + Spring AI + Bedrock setup** (4 hours)
2. **Resume parsing with Apache Tika** (6 hours)
3. **Core Claude prompts for resume analysis** (4 hours)
4. **Basic Angular dashboard for candidate review** (8 hours)
5. **Score override/intangibles interface** (6 hours)
6. **Batch upload and processing** (4 hours)
7. **Test data generation (20+ realistic resumes)** (4 hours)

MOCK/SIMULATE (Quick Wins - 8 hours)

1. **Job template creation** - Hardcode 2-3 job types (Java Dev, Product Manager, Designer)
2. **User authentication** - Single hardcoded “HR Manager” user
3. **File storage** - Local filesystem instead of S3
4. **Real-time notifications** - Console logging + basic progress bars

DOCUMENT AS FUTURE (No Build Time)

1. **Advanced bias detection algorithms**
2. **Complex user role management**
3. **Email notification system**
4. **Advanced analytics and reporting**
5. **Production-grade security and audit logging**
6. **Integration with existing HR systems**

DEMO PREPARATION (4 hours)

1. **Demo script with 3 scenarios**: Batch processing, manual override, bias detection
2. **Presentation slides** emphasizing AI culture change
3. **Record coding demo** showing AI development assistance

Division of Responsibilities (48-Hour Focus)

5-Person Team Roles - all can assume and share tasks

Role 1: AI Integration

- **Core Build**: Spring AI + Bedrock setup, Claude prompt engineering, scoring engine

- **Time Allocation**: 16 hours core AI work, 8 hours team coordination/architecture decisions
- **Key Deliverable**: Working AI resume analysis with explainable scores
- **AI Development**: Live demo of AI-generated Spring services during presentation

Role 2: Backend Developer

- **Core Build**: Spring Boot REST API, PostgreSQL setup, batch processing endpoints
- **Mock/Skip**: Authentication (hardcode user), file storage (local), complex data models
- **Time Allocation**: 20 hours backend core, 4 hours integration testing
- **AI Development**: Claude-generated controllers, test cases, API documentation

Role 3: Frontend Developer

- **Core Build**: Angular candidate dashboard, score override interface, batch upload UI
- **Mock/Skip**: Advanced responsive design, complex animations, user management
- **Time Allocation**: 18 hours core UI, 6 hours polish and mobile basics
- **AI Development**: AI-generated Angular components, TypeScript interfaces

Role 4: Product & Demo Lead

- **Core Build**: Test data creation (20+ resumes), demo scenarios, presentation materials
- **Mock/Document**: Advanced job templates, complex bias scenarios, enterprise features
- **Time Allocation**: 12 hours demo prep, 8 hours product validation, 4 hours presentation
- **AI Product**: Design natural language query features, explainability requirements

Role 5: Integration & Testing Lead

- **Core Build**: End-to-end testing, performance validation with batch processing, bug fixes
- **Mock/Document**: Production security, audit logging, advanced error handling
- **Time Allocation**: 16 hours testing/integration, 8 hours documentation and future roadmap
- **AI Usage**: AI-generated test scenarios, edge case identification, validation scripts

AI Work Distribution Strategy

AI Development Tools (How we build faster):

- **Everyone**: Use Claude for boilerplate generation, debugging assistance, documentation
- **Backend/Frontend**: AI-generated code scaffolding, component templates
- **Product Lead**: AI-created test data, realistic resume generation
- **Testing Lead**: AI-generated test cases and validation scenarios

AI Product Features (What customers see):

- **AI Integration Lead**: Owns all Claude interactions, prompt optimization, model evaluation
- **Product Lead**: Designs AI user experience, natural language features, scoring explanations
- **Frontend Lead**: Implements AI response displays, loading states, explanation interfaces

Daily Coordination

- **Hour 0**: Architecture setup and AI integration foundation
- **Hour 12**: First integration checkpoint - AI parsing working
- **Hour 24**: Full pipeline demo - upload to scored results
- **Hour 36**: Human override features complete, demo rehearsal
- **Hour 48**: Final presentation with live AI development demonstration

AI Work Division Strategy

AI Development Usage (How our team uses AI to build)

- **Backend**: Claude generates Spring services, integration patterns, error handling
- **Frontend**: Claude creates Angular components, responsive layouts, TypeScript models
- **All**: Real-time code assistance, debugging support, documentation generation

AI Product Features (AI in the final solution)

- **AI Integration**: Each team member must get familiar with Claude/Bedrock calls, prompt optimization, model evaluation. Euan and Lucas have significant Claude prompting and response analysis experience
- **Product Development**: Each team member designs AI user interactions, natural language features, explainability requirements
- **Testing**: Validate AI outputs, verify bias detection accuracy, and edge case handling

Shared AI Responsibilities

- **Prompt Engineering**: ALL. Euan and Lucas can lead initially
- **AI Ethics/Bias**: All team members contribute insights, testing, and validate
- **Performance Optimization**: As possible, take advantage of AI capabilities to maximize but do not linger on ‘poor’ performance, POC are throwaways and data set can be optimized.

Deliverable Focus

Primary Demo Goal: Show a busy hiring manager processing 50 resumes in 10 minutes instead of 5 hours, with explainable AI reasoning and human override capabilities.

Secondary Goal: Demonstrate how AI tools accelerated your development process and can transform team productivity