

Sprint 4 Documentation

Project: Seasonal Job Matching Platform

Sprint Identifier: SJMP Sprint 4

Total Estimated Effort: 21 Story Points

1. Feature: Personalized Job Matching (SJMP-207)

Status: Done

Goal: As a jobseeker, I want to see the top jobs that match my interests, skills, and preferences so that I can quickly discover the most relevant opportunities without manually searching.

🛠️ Departmental Breakdown & Living Documentation

Department	Task / Sub-task	Living Documentation (Executable Test)
AI	Build the matching engine and tune accuracy (SJMP-215, 218)	<code>test_match_ranking</code> : Verifies jobs are sorted by match score with highest first
API	Create recommended jobs API and integrate with engine (SJMP-211, 212)	<code>test_api_latency</code> : Asserts API returns results in < 1 second
Mobile / Web	Display top jobs and handle loading/empty states (SJMP-208, 209, 210)	<code>test_ui_empty_state</code> : Verifies "No matching jobs found yet" displays correctly
DevOps	Populate database with seeded job data (SJMP-216)	<code>test_db_seed_count</code> : Confirms sufficient data for the matching engine

☑ Acceptance Criteria

- Users see top matching jobs immediately upon opening "For You"
- Results are personalized based on skills, interests, and experience
- Backend API returns jobs sorted by match score

📷 Implementation Screenshots

The image contains three screenshots related to the Recommendations View:

- Loading Screen:** Shows a large, semi-transparent gray overlay with the text "Recommended For You" at the top left.
- Recommendations View:** Shows a list of recommended jobs. One job is highlighted: "Andrew Rogers" from "Coastal Town, ME" as a "Hotel Front Desk Agent (Summer Season)" working "FULL_TIME" and "Hiring". It shows a starting date of "Starting May 01" and a rate of "\$19 /hourly". A "View →" button is at the bottom right.
- Error State:** Shows a message "Failed to load recommendations" with a "Retry" button below it.

Caption: Recommendations View

2. Feature: Resume Management (SJMP-219)

Status: Done

Goal: As a jobseeker, I want to create and edit my resume; as an employer, I want to view applicant resumes to assess suitability.

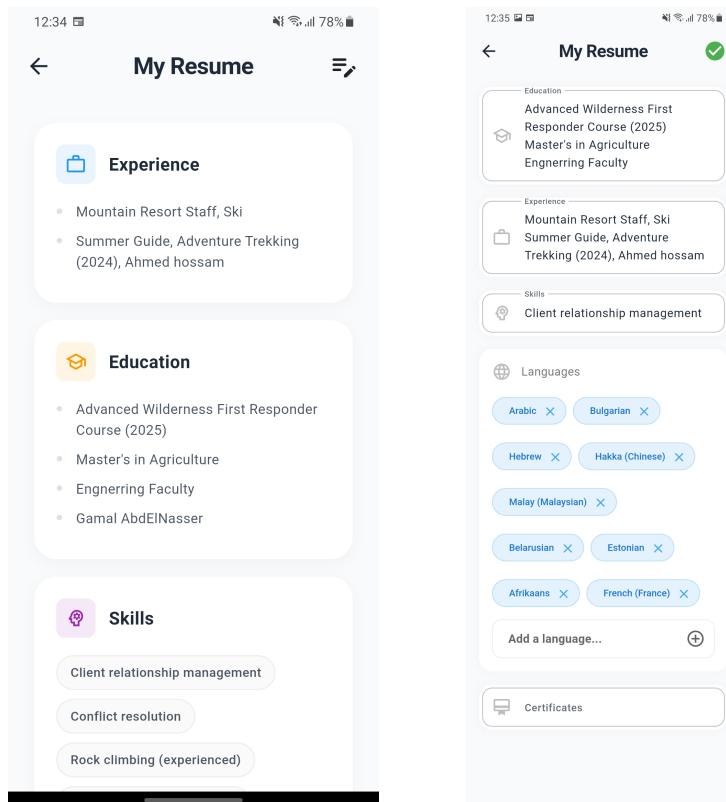
🛠️ Departmental Breakdown & Living Documentation

Department	Task / Sub-task	Living Documentation (Executable Test)
API	Create APIs for resume CRUD operations (SJMP-220)	<code>test_resume_persistence</code> : Verifies that edits are saved and reflected immediately
Web / Mobile	Build resume editor and employer-facing display (SJMP-221, 222)	<code>test_resume_formatting</code> : Asserts consistent layout for employer views

☑ Acceptance Criteria

- Jobseekers can create/edit resumes with fields: personal info, education, experience, and skills
- Changes are saved and reflected in the system immediately
- Employers can view resumes only for applicants who have applied to their jobs

📷 Implementation Screenshots



🔗 Cross-Departmental Contract (Handshake)

- **AI ↔ API:** AI model output must be exposed in a format usable by the backend (SJMP-217)
 - **API ↔ Frontend:** The backend must handle sorting logic and ensure rapid response times (<1s)
 - **Security:** API must strictly enforce that resumes are only "Readable" by authorized employers
-

Behavioral Specification (Gherkin Format)

Feature: AI Match & Resume Updates

Scenario 1: Accurate Matching

```
Given a user has "Python" skills and the DB has "Python Developer" jobs
When the user views recommendations
Then the matching engine should rank those jobs at the top
```

Scenario 2: Real-time Resume Edit

```
Given a user is editing their resume
When they save a new skill
Then the API must update the record immediately
```