

# Sprint 4: Progress Report & Technical Overview

---

In this sprint, we focused on making the job-seeking experience more personal and efficient. We implemented a resume builder and a smart job matching system to help users find the right opportunities faster.

## What we delivered

- **Smart Job Matching:** Users can now see a curated list of jobs that actually fit their skills and interests.
  - **Resume Builder:** We added a way for jobseekers to create and manage their resumes directly in the app.
- 

### 1. Resume Management

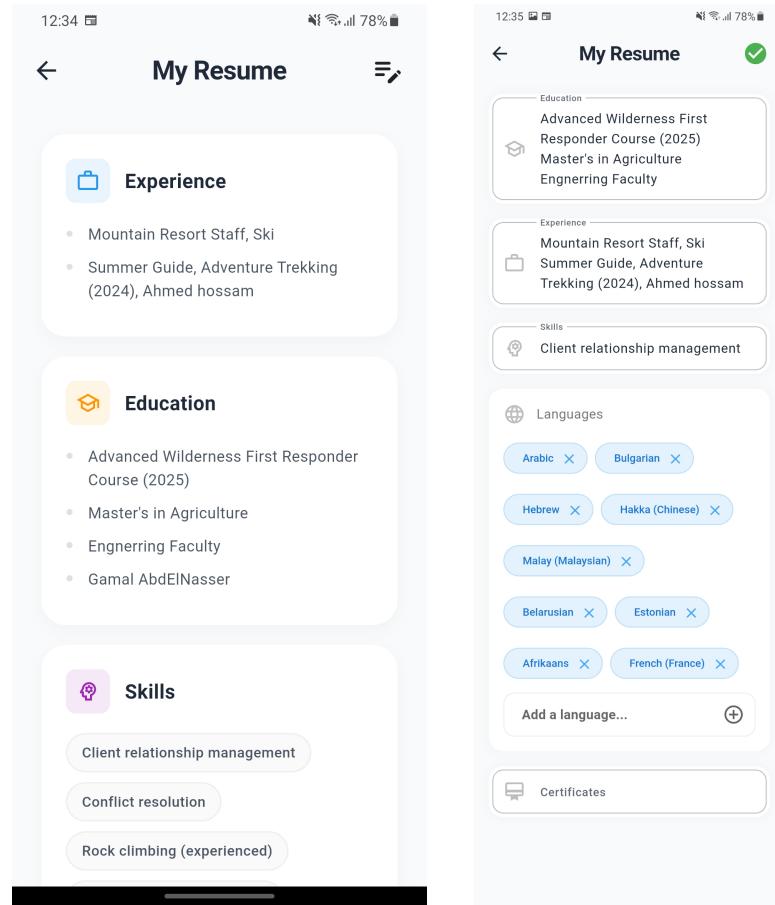
**Test File:** `test/screens/Profile/resume_screen_test.dart`

Feature	Details
Target	ResumeScreen
Tools	Flutter Test, Riverpod, Mock Notifiers

#### What we're verifying:

- **Initial Load:** The screen correctly pulls in the user's existing data (like languages) and displays them as chips.
- **Interactive Editing:** Users can pick new languages from a dropdown and remove them easily. We also made sure the dropdown resets properly after a selection.
- **Smart Validation:** You can't add the same language twice—the app handles duplicates gracefully by disabling them in the list.

#### UI Preview:



## 2. Job Recommendations

**Test File:** `test/providers/recommended_jobs_provider_test.dart`

Feature	Details
<b>Target</b>	<code>RecommendedJobsNotifier</code>
<b>Tools</b>	Riverpod, Mocktail, Unit Testing

### What we're verifying:

- **Loading States:** The app shows a loading indicator immediately while it's fetching the best matches.
- **Data Fetching:** Once we have a valid user, the system calls our matching algorithm and returns the results.
- **Edge Cases:** If a user isn't fully logged in (ID is 0), we skip the API call and just show an empty list instead of crashing.

### UI Preview:

## Recommended For You



 Recommended For You  
Based on your profile

 Andrew Rogers  
📍 Coastal Town, ME 

### Hotel Front Desk Agent (Summer Season)

 FULL\_TIME  Hiring

Starting May 01

\$19 /hourly

[View →](#)

... 1 2 3 4

# Recommended For You

Failed to load recommendations

[Retry](#)

## Technical Decisions

We made a few key architectural choices this sprint to keep the codebase clean and the app fast.

### 1. Reactive State with Riverpod

We're using `AsyncNotifier` for almost everything now. It's great because it handles the "Loading/Error/Data" states automatically. For example, the `RecommendedJobsNotifier` just "watches" the user's profile and automatically re-fetches jobs whenever their interests change.

### 2. Efficient "Diff" Updates

For things like **Fields of Interest (FOI)** and **Resumes**, we stopped sending the whole list back to the server every time. Instead, we just send what was added or removed. This makes the app feel much snappier and saves data.

#### **FOI UI Preview:**

## Fields of Interest



Hospitality X

hwak X

Save Changes

## Add Interest

e.g. Graphic Design

Cancel

Add

### 3. Clean API Layer

We're sticking with **Dio** for our network calls. We've set it up so that we can easily swap in "mock" data during testing, which means we can test the UI without actually needing a working backend.

---

## Key API Endpoints

Endpoint	Action	Purpose
<code>GET /users/FOI/:id</code>	Fetch	Gets the user's current interests.
<code>PATCH /users/:id</code>	Update	Updates profile fields (like FOI) using our new diff pattern.
<code>GET /recommendations</code>	Fetch	Pulls the matched jobs based on the user's profile.

---

## Best Practices We're Following

- 1. Consistent State Handling:** By using `AsyncValue.guard()`, we ensure that every screen in the app handles errors the same way.
- 2. Service Separation:** We keep our API logic (Services) separate from our UI logic (Providers). This makes it way easier to fix bugs without breaking the whole screen.
- 3. Graceful Failures:** If one part of the app fails (like loading favorites), we don't break the whole page. We show what we can and give the user a friendly error message for the rest.
- 4. Test-Driven Reliability:** Every new feature this sprint came with a suite of tests that mock the network, so we know they work even when the server is down.