Pathwise - Split and Structured Requirements

Must Have

1. User Profile Storage

1. The Pathwise backend SHALL define a database schema for user profiles including fields for skills, experience, goals, and education, ensuring that the schema can store at least 1,000 user records without performance degradation.
2. The Pathwise backend SHALL implement Django model classes matching the user profile schema, allowing ORM-based create, read, update, and delete (CRUD) operations.
3. The Pathwise backend SHALL provide REST API endpoints for creating, updating, deleting, and retrieving user profiles, returning responses within 500 ms for 95% of requests.

2. User Survey Interface

1. The Pathwise frontend SHALL display a multi-step survey form built with Bootstrap, showing no more than 5 fields per step for better usability.
2. The Pathwise frontend SHALL prevent survey submission unless all mandatory fields are completed, displaying an error message within 0.5 seconds of clicking submit.

3. Job Platform API Integration

1. The Pathwise backend SHALL connect to the chosen job platform API (e.g., LinkedIn, Magneto) using valid API credentials, rejecting all requests with missing or expired tokens.
2. The Pathwise backend SHALL send job search requests to the job platform API based on stored profile data, retrieving at least 20 matching job listings per query.
3. The Pathwise backend SHALL parse retrieved job listings into a normalized format and store them in the database within 2 seconds of API response.

4. Career Roadmap Engine

1. The Pathwise backend SHALL send user profile data to the AI API and receive a career roadmap, completing the request in less than 5 seconds for 90% of cases.
2. The Pathwise backend SHALL structure the AI roadmap output into at least three actionable steps, each containing a description, required skills, and estimated timeline.
3. The Pathwise backend SHALL store the generated roadmap in the database, linking it to the user profile by a unique user ID.

Should Have

5. Job Matching Filter

1. The Pathwise backend SHALL filter retrieved job listings by required skills, location, and target job title, discarding results that match less than 70% of the profile criteria.
2. The Pathwise backend SHALL integrate the filtering function into the job retrieval pipeline, ensuring all displayed jobs are filtered before being sent to the frontend.

6. UI for Job Suggestions & Career Roadmap

1. The Pathwise frontend SHALL display job suggestions in a responsive Bootstrap template, showing at least 5 job listings per page.
2. The Pathwise frontend SHALL display the career roadmap in a visually distinct section, highlighting upcoming steps in a different color.
3. The Pathwise frontend SHALL adapt job suggestions and roadmap layouts to fit devices with a minimum screen width of 320px without requiring horizontal scrolling.

7. Profile Update & Dynamic Roadmap

1. The Pathwise backend SHALL provide an API endpoint to update user profiles, saving changes in under 1 second for 95% of requests.
2. The Pathwise backend SHALL trigger roadmap regeneration automatically upon profile update, generating a new roadmap within 5 seconds.
3. The Pathwise backend SHALL re-run job matching immediately after roadmap regeneration, replacing outdated job results in the user's dashboard.

Could Have

8. Notification System

1. The Pathwise backend SHALL trigger a notification when a new job listing matches at least 80% of the stored profile criteria.
2. The Pathwise system SHALL deliver notifications via email or in-app messages within 1 minute of detection.

9. In-Platform Job Applications

1. The Pathwise backend SHALL integrate with the job platform's application API to allow direct job applications from the platform.
2. The Pathwise system SHALL pre-fill application forms with stored profile data, allowing the user to review before submission.

Won't Yet

10. AI-Based Resume Review

1. The Pathwise development team SHALL design a future workflow for uploading resumes and sending them to an AI module for analysis, but SHALL NOT implement it in the current release.