# JobQuest Navigator – Product Requirements Document (PRD)

Maria Soto, Shruti Amit Vasanwala, Zhihuai Wang, Ishan Aakash Patel

Team 9

The Zombies of CAA

Seneca Polytechnic

Course Code: CAA900

David Chan

# 1. Project Background, Goals & Pain Points

## Background:

JobQuest Navigator is designed to address the challenges faced by modern job seekers in efficiently finding, applying for, and preparing for jobs. The current job search landscape is fragmented, with users needing to navigate multiple platforms, tailor resumes manually, and keep track of applications and interview preparation separately.

## Pain Points:

- Difficulty in discovering relevant job opportunities nearby

- Manual and repetitive resume tailoring for each application

- Lack of clear guidance on upskilling and certifications

- Poor visibility into application status and history

- Insufficient support for company research and interview preparation

## Project Goal:

Deliver an all-in-one intelligent job search platform that enables users to discover jobs, optimize resumes, enhance skills, track applications, and prepare for interviews efficiently, with a seamless and user-friendly experience.

## Overall Acceptance Criteria:

- All six major epics are implemented with core features available to users

- The platform supports job discovery, resume management, AI-driven suggestions, certification planning, application tracking, and interview preparation

- User flows are intuitive and tested with real users

- Alpha demo is delivered by 6.28, covering at least Epics 1/2/3 core features

# 2. User Personas & Typical Scenarios

Persona 1: Emily, Recent Graduate

- Needs to find her first job in a new city

- Wants to quickly discover nearby opportunities and tailor her resume for each application

- Lacks experience in interview preparation

## Scenario:

Emily uses JobQuest Navigator to view jobs on a map, gets AI suggestions to improve her resume for each job, tracks her applications, and uses the interview prep module to practice questions for her target companies.

Persona 2: Michael, Mid-career Professional

- Looking to switch industries and upskill

- Needs guidance on which certifications are in demand

- Wants to keep track of multiple job applications and resume versions

## Scenario:

Michael leverages the certification roadmap to identify skill gaps, receives personalized certification recommendations, and manages multiple tailored resumes for different job applications.

# 3. MVP Scope & Feature Prioritization

## MVP Must-Have Features:

- Geolocation-based job mapping (Epic 1)

- Automated resume versioning and AI suggestion engine (Epic 2)

- Resume alteration suggestions with feedback loops (Epic 3)

## MVP Nice-to-Have Features:

- Job application tracking with resume used (Epic 5)

- AI-driven company research and interview prep module (Epic 6)

## Post-MVP/Stretch Features:

- Dynamic certification roadmap with market demand alerts (Epic 4)

# 4. Non-Functional Requirements

- Performance: Job search, resume management, and application tracking should respond within 2 seconds for 95% of user actions

- Security: All user data (resumes, applications, personal info) must be encrypted in transit and at rest

- Privacy: User data will not be shared with third parties without explicit consent

- Availability: The platform should have 99% uptime during the pilot/alpha phase

- Compatibility: Support for latest versions of Chrome, Firefox, Safari, and Edge

- Accessibility: Key flows should meet WCAG 2.1 AA accessibility standards

# 5. Milestones & Deliverables

- Week 1: Project kickoff, team formation, topic selection, initial direction documented

- Week 2: PRD draft completed and reviewed, requirements clarified

- Week 3: Tech stack selected, UI/UX prototypes for core features, task breakdown for all epics

- Week 4: Initial versions of map UI, resume management, and AI resume suggestions implemented; backend integration for job data and resume versioning functional

- Week 5: Map UI, resume recommendation, and AI suggestions refined; frontend and backend integrated for core flows

- Week 6: Certification roadmap, application tracking, and interview prep modules implemented; all modules integrated and tested for basic functionality

- Week 7: All features integrated and tested; Alpha Demo ready and presentation materials prepared

- By 6.28: Alpha version demo delivered, covering at least Epics 1/2/3 core features

# 6. Risks & Assumptions

## Risks:

- Third-party API changes or deprecation (Google for Jobs, Jobscan, LinkedIn, etc.)

- Delays in AI model development or integration

- Data privacy or security incidents

- Underestimation of UI/UX complexity or integration effort

- Team member availability or resource constraints

## Assumptions:

- All required third-party APIs will be accessible and stable during the project

- Team members have the necessary skills or can quickly ramp up

- Users will provide feedback during alpha testing

# 7. Detailed Functional Requirements (by Epic)

## Epic 1: Geolocation-Based Job Mapping

Goal: Enable users to discover and prioritize nearby job opportunities using an interactive map interface.

### User Story 1.1: Integrate Google for Jobs API to fetch job listings

- Description: As a user, I want the system to fetch job listings from Google for Jobs so that I can see available jobs nearby.

- Goal: Provide up-to-date, location-based job data.

### Acceptance Criteria:

- The system fetches job listings from Google for Jobs API.

- Job data includes job title, company, location, and posting date.

- Data refreshes at least once every 24 hours.

### Task Breakdown:

- Research Google for Jobs API and authentication (1 day)

- Implement API integration and data fetching (2 days)

- Create backend service to store and update job listings (1 day)

- Write unit tests for API integration (1 day)

### User Story 1.2: Design and implement a visual map interface

- Description: As a user, I want to see job listings on a map so that I can easily identify nearby opportunities.

- Goal: Provide an intuitive, interactive map for job discovery.

### Acceptance Criteria:

- Map displays pins for each job location.

- Pins are clickable and show job details.

- Map supports zoom and pan.

### Task Breakdown:

- Design UI mockups for map interface (1 day)

- Implement map component with pins (3 days)

- Integrate job data with map pins (2 days)

- Write UI tests (2 days)

### User Story 1.3: Sync job listings with map pins based on geolocation

- Description: As a user, I want the map to prioritize and highlight jobs near my current location.

- Goal: Make job search more relevant by proximity.

### Acceptance Criteria:

- Map centers on user's current geolocation.

- Jobs within a 50-mile radius are highlighted.

- User can filter jobs by distance.

### Task Breakdown:

- Implement geolocation detection (1 day)

- Add filtering and highlighting logic (2 days)

- UI updates for filters (1 day)

- Testing and bug fixes (1 day)

### User Story 1.4: Implement filtering and prioritization of nearby jobs

- Description: As a user, I want to filter jobs by distance and prioritize the closest ones.

- Goal: Allow users to focus on the most relevant job opportunities.

### Acceptance Criteria:

- User can set distance filter (e.g., 10, 25, 50 miles).

- Jobs are sorted by proximity.

- Filter settings persist between sessions.

### Task Breakdown:

- Add filter UI controls (1 day)

- Implement sorting logic (1 day)

- Persist filter settings (1 day)

## Epic 2: Automated Resume Versioning and Suggestion Engine

Goal: Help users manage multiple resume versions and receive AI-driven recommendations for each job application.

### User Story 2.1: Build AI-driven resume version control system

- Description: As a user, I want the system to manage multiple versions of my resume automatically so I can apply with the best fit.

- Goal: Enable efficient resume management and selection.

### Acceptance Criteria:

- System tracks different resume versions.

- Users can view, edit, and select resume versions.

- Version history is maintained.

### Task Breakdown:

- Design version control data model (1 day)

- Implement backend versioning logic (3 days)

- Build UI for version management (2 days)

- Write unit and integration tests (2 days)

### User Story 2.2: Integrate Jobscan's resume library

- Description: As a user, I want access to Jobscan's resume templates and library to improve my resume quality.

- Goal: Provide high-quality resume templates and import/export features.

### Acceptance Criteria:

- Resume templates from Jobscan are accessible.

- Users can import and export resumes.

- Integration is secure and reliable.

### Task Breakdown:

- Research Jobscan API or data access (1 day)

- Implement integration and data sync (2 days)

- UI integration for template selection (1 day)

- Testing and validation (1 day)

### User Story 2.3: Develop recommendation engine for best resume per job

- Description: As a user, I want AI to recommend the best resume version for each job application.

- Goal: Maximize job application success with tailored resume suggestions.

### Acceptance Criteria:

- AI analyzes job description and matches resume versions.

- Recommendations are displayed with confidence scores.

- Users can override recommendations.

### Task Breakdown:

- Design AI recommendation algorithm (2 days)

- Implement backend AI service (3 days)

- Integrate recommendations into UI (2 days)

- Testing and tuning AI model (1 day)

### User Story 2.4: UI for users to view and select recommended resumes

- Description: As a user, I want a clear interface to review and select AI-recommended resumes.

- Goal: Make it easy for users to compare and choose resumes.

### Acceptance Criteria:

- Recommendations are easy to view and compare.

- Users can accept or reject suggestions.

- Changes are saved and reflected in applications.

### Task Breakdown:

- Design UI mockups (1 day)

- Implement recommendation UI (2 days)

- Connect UI with backend services (1 day)

- Write UI tests (1 day)

## Epic 3: Resume Alteration Suggestion with Feedback Loops

Goal: Continuously improve resume quality through AI suggestions and user feedback.

### User Story 3.1: Implement AI suggestions for resume alterations

- Description: As a user, I want AI to suggest changes to my resume to improve job fit.

- Goal: Help users optimize resumes for each job.

### Acceptance Criteria:

- AI provides specific, actionable suggestions.

- Suggestions cover keywords, formatting, and content.

- Suggestions update dynamically based on job description.

### Task Breakdown:

- Develop AI suggestion engine (3 days)

- Integrate with resume editor (2 days)

- Testing and validation (2 days)

- Documentation (1 day)

### User Story 3.2: Build UI for users to accept or reject changes

- Description: As a user, I want to review AI suggestions and decide which to apply.

- Goal: Empower users to control resume changes.

### Acceptance Criteria:

- Suggestions are presented clearly.

- Users can accept, reject, or modify suggestions.

- Accepted changes update the resume immediately.

### Task Breakdown:

- Design UI for suggestion review (1 day)

- Implement accept/reject functionality (2 days)

- Sync changes with resume versioning (1 day)

- Testing (1 day)

### User Story 3.3: Create feedback loop to improve suggestions over time

- Description: As a system, I want to learn from user feedback to improve AI suggestions.

- Goal: Make AI smarter and more relevant over time.

### Acceptance Criteria:

- User actions (accept/reject) are logged.

- AI model updates based on feedback.

- Feedback improves suggestion accuracy.

### Task Breakdown:

- Implement feedback data collection (2 days)

- Develop model retraining pipeline (3 days)

- Integrate updated model into suggestion engine (2 days)

- Testing and monitoring (1 day)

## Epic 4: Dynamic Certification Roadmap with Market Demand Alerts

Goal: Guide users to upskill with certifications that match market demand and personal skill gaps.

### User Story 4.1: Integrate Careerflow's skill gap analysis

- Description: As a user, I want to identify skill gaps based on my profile and job market data.

- Goal: Help users understand and address their skill gaps.

### Acceptance Criteria:

- Skill gap analysis runs on user data.

- Results highlight missing or weak skills.

- Analysis updates regularly.

### Task Breakdown:

- Research Careerflow API (1 day)

- Implement integration and data sync (2 days)

- UI to display skill gaps (1 day)

- Testing (1 day)

### User Story 4.2: Fetch real-time certification recommendations from Indeed

- Description: As a user, I want certification suggestions based on current market demand.

- Goal: Keep users competitive with up-to-date certification advice.

### Acceptance Criteria:

- System fetches certification data from Indeed.

- Recommendations are personalized based on skill gaps.

- Data refreshes daily.

### Task Breakdown:

- Research Indeed data sources (2 days)

- Implement data fetching and processing (3 days)

- Integrate with skill gap results (2 days)

- Testing (1 day)

### User Story 4.3: Design certification roadmap UI

- Description: As a user, I want a clear roadmap showing certifications to pursue.

- Goal: Visualize and track certification progress.

### Acceptance Criteria:

- Roadmap visualizes certifications and timelines.

- Users can mark certifications as completed.

- Roadmap updates dynamically.

### Task Breakdown:

- Design UI mockups (1 day)

- Implement roadmap visualization (3 days)

- Add user interaction features (1 day)

- Testing (1 day)

### User Story 4.4: Implement alert system for market demand changes

- Description: As a user, I want to receive alerts when certification demand changes.

- Goal: Keep users informed of important market shifts.

### Acceptance Criteria:

- Alerts trigger on significant market changes.

- Users can customize alert preferences.

- Alerts delivered via email or app notifications.

### Task Breakdown:

- Design alert logic and thresholds (1 day)

- Implement notification system (2 days)

- UI for alert preferences (1 day)

- Testing (1 day)

## Epic 5: Job Application Tracking with Resume Used

Goal: Enable users to track all job applications and associated resume versions, with timely notifications.

### User Story 5.1: Track job applications and associated resume versions

- Description: As a user, I want to track each job application and the resume version used.

- Goal: Provide a complete record of job search activity.

## - Acceptance Criteria:

- Users can log job applications.

- Each application links to a specific resume version.

- Application status is tracked (applied, interview, offer, etc.).

## - Task Breakdown:

- Design data model for applications and resume links (1 day)

- Implement backend tracking logic (2 days)

- UI for application entry and status updates (1 day)

- Testing (1 day)

### User Story 5.2: UI to view application status and resume used

- Description: As a user, I want a dashboard to view all applications and resumes used.

- Goal: Make it easy to monitor and update application progress.

## - Acceptance Criteria:

- Dashboard lists applications with status and resume info.

- Users can filter and sort applications.

- Status updates are easy to make.

## - Task Breakdown:

- Design dashboard UI (1 day)

- Implement filtering and sorting (2 days)

- Connect UI with backend data (1 day)

- Testing (1 day)

### User Story 5.3: Notifications for application updates

- Description: As a user, I want to receive notifications when application statuses change.

- Goal: Keep users informed of important updates.

## - Acceptance Criteria:

- Notifications sent via email or app.

- Users can customize notification preferences.

- Notifications are timely and accurate.

## - Task Breakdown:

- Implement notification triggers (1 day)

- Build notification delivery system (1 day)

- UI for notification settings (1 day)

## Epic 6: AI-Driven Company Research and Interview Prep Module

Goal: Equip users with company insights and AI-powered interview preparation tools.

### User Story 6.1: Integrate LinkedIn Jobs data for company insights

- Description: As a user, I want company insights from LinkedIn Jobs to prepare for interviews.

- Goal: Provide rich, up-to-date company information.

### Acceptance Criteria:

- System fetches company profiles and job data.

- Data includes company size, culture, recent news.

- Data refreshes regularly.

### Task Breakdown:

- Research LinkedIn Jobs API (2 days)

- Implement data fetching and storage (3 days)

- UI integration for company insights (2 days)

- Testing (1 day)

### User Story 6.2: Develop predictive interview question generator

- Description: As a user, I want AI-generated interview questions tailored to the company and role.

- Goal: Help users prepare for interviews with relevant questions.

### Acceptance Criteria:

- AI generates relevant questions based on job description.

- Questions cover technical and behavioral topics.

- Users can save and practice questions.

### Task Breakdown:

- Design AI question generation model (3 days)

- Implement backend AI service (3 days)

- UI for question review and practice (2 days)

### User Story 6.3: Build company dossier UI

- Description: As a user, I want a consolidated view of company info and interview prep materials.

- Goal: Centralize all relevant information for interview preparation.

### Acceptance Criteria:

- Dossier includes company insights, job details, and questions.

- UI is easy to navigate and visually appealing.

- Users can add notes.

### Task Breakdown:

- Design dossier UI (1 day)

- Implement dossier components (3 days)

- Add note-taking feature (1 day)

- Testing (1 day)

### User Story 6.4: Combine insights into interview prep module

- Description: As a user, I want a single module that integrates all interview prep features.

- Goal: Streamline the interview preparation process.

### Acceptance Criteria:

- Module includes company research, questions, and notes.

- Users can track prep progress.

- Module is accessible from job application pages.

### Task Breakdown:

- Design module architecture (1 day)

- Integrate components (3 days)

- Implement progress tracking (1 day)

- Testing (1 day)

# 8. Sprint-by-Sprint Goals & Acceptance Criteria

Week 1 (5.10-5.16)

Sprint Goal: Team is formed, project topic is selected, and initial direction is agreed upon by all members.

## Sprint Acceptance Criteria:

- All team members participate in topic research and team formation.

- Project direction is documented and agreed upon.

Week 2 (5.17-5.23)

Sprint Goal: Complete PRD draft and requirements analysis.

## Sprint Acceptance Criteria:

- PRD is drafted and reviewed by all members.

- Key requirements and user flows are clarified.

Week 3 (5.24-5.30)

Sprint Goal: Finalize tech stack, produce initial prototypes, and break down tasks for development.

## Sprint Acceptance Criteria:

- Tech stack is selected and documented.

- UI/UX prototypes are created for core features.

- Task breakdown is completed for all epics.

Week 4 (5.31-6.6)

Sprint Goal: Begin core development of Epics 1/2/3.

## Sprint Acceptance Criteria:

- Initial versions of map UI, resume management, and AI resume suggestions are implemented.

- Backend integration for job data and resume versioning is functional.

Week 5 (6.7-6.13)

Sprint Goal: Iterate and integrate Epics 1/2/3, address feedback, and improve core features.

## Sprint Acceptance Criteria:

- Map UI, resume recommendation, and AI suggestions are refined based on feedback.

- Frontend and backend are successfully integrated for core flows.

Week 6 (6.14-6.20)

Sprint Goal: Develop and integrate Epics 4/5/6, expand platform functionality.

## Sprint Acceptance Criteria:

- Certification roadmap, application tracking, and interview prep modules are implemented.

- All modules are integrated and tested for basic functionality.

Week 7 (6.21-6.27)

Sprint Goal: Integrate, test, and optimize for Alpha Demo.

## Sprint Acceptance Criteria:

- All features are integrated and tested.

- Alpha Demo is ready and presentation materials are prepared.

## By 6.28.25:

- Alpha version demo is delivered, covering at least Epics 1/2/3 core features.

# 9. Acceptance & QA Process

- Each feature and epic will be reviewed against its acceptance criteria

- User testing will be conducted for all major flows

- Bugs and issues will be tracked and resolved before release

- Alpha demo will be validated by the team and test users

# 10. Gantt Chart

Note: For the task schedule and timeline line please check the Jira project:

https://myseneca-team-pi6s3gm8.atlassian.net/jira/software/projects/SM/boards/1?atlOrigin=eyJpIjoiNjU1NDVjMzUyN2UwNGRlZmIzYjk0OTY4ZWVmYWMxZmQiLCJwIjoiaiJ9

A screenshot of a computer

AI-generated content may be incorrect.