# JobQuest Navigator – Week 4 Update Report

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

Team 9

The Zombies of CAA

Seneca Polytechnic

Course Code: CAA900

David Chan

## Project Summary

**Weekly Meeting Minutes and Materials**

Access the documents from our weekly meetings

[00 Documents/0099 Final Documents/Reports/Meetings](https://github.com/MariaVSoto/JobQuest_Navigator_CAA/tree/main/00%20Documents/0099%20Final%20Documents/Reports/Meetings)

## Accomplishments

## Completed Work

* **User Story 1.1: Integrate Google for Jobs API**
  + **All tasks completed during Sprint 1.**
    - Created data dictionary.
    - Researched API and authentication.
    - Implemented API integration and data fetching.
    - Developed backend service for job storage and updates.
    - Wrote unit tests for API integration.
* **UI/UX Design**
  + Completed Figma UI design for the full application workflow.
* **Frontend Implementation**
  + Built 5 key pages using React.js, CSS, and JavaScript:
    - Login & Signup
    - Dashboard
    - Job Search
    - Job Map Interface
    - Navigation Header/Footer
  + **Map Integration Features:**
    - Google Maps implementation
    - Job pin rendering using mock job data
    - User location marker
    - Job popup modal on pin click
  + Followed modular component structure and consistent visual styling.
* **Epic 2:**
  + - Completed the logic design of the resume management module
    - Completed the development of the main program, T
    - Completed the design of database-related tables
    - Completed the writing of the resume backend management interface

## Challenges or Blockers

|  |  |  |
| --- | --- | --- |
| **User Story** | **Task** | **Challenges or Blockers** |
| 1.2 | SM-33 | Learning cost of using the Django framework |
| 1.2 | SM-34 | Need to integrate Map API. Facing an issue with the job search API and Map API integration. |
| 1.4 | SM-46 | Filtering and prioritization are in progress |

## Plan for Next Week

Sprint: **Sprint 3**

**Sprint Dates:**07June 2025 to 14 June 2025

|  |  |  |
| --- | --- | --- |
| **Epic** | **Owner** | **Task** |
| 0 | Shruti | IU Design |
| 1 | Kevin | User Story 1.2 (A): Design and implement a visual map interface. |
| 2 | Kevin | User Story 2.2: Integrate Jobscan’s resume library |
| 3 | Maria Soto | User Story 3.1 (B): Implement AI suggestions for resume alterations. |
| 4 | Ishan | User Story 4.1: Integrate Careerflow’s skill gap analysis. |

## Sprint Status and Timeline

* **Progress Tracking:**
  + Use Jira for task management and sprint tracking.
    - [List - The Zombies of CAA - Jira](https://myseneca-team-pi6s3gm8.atlassian.net/jira/software/projects/SM/list?sortBy=customfield_10015&direction=ASC)
  + Update GitHub weekly for documentation and progress updates.
    - <https://github.com/MariaVSoto/JobQuest_Navigator_CAA/tree/dev>
* **Next Steps:**
  + Work on assigned tasks per the above plan.
  + Address blockers and share progress in weekly meetings.
  + Ensure all documentation and code are up-to-date in the project repository.

A screenshot of a computer

AI-generated content may be incorrect.

## Tools to develop the project

Based on project requirements and your preferences, the following technology stack is selected:

* **Frontend:** React.js
* **Backend:** Django (Python, full-featured, powerful ORM, suitable for rapid development and robust microservices)
* **Backend runtime:** Lambda (serverless deployment for backend services; enables automatic scaling, pay-per-use billing, and simplified infrastructure management. Each microservice or API endpoint can be deployed as an independent Lambda function, integrated with API Gateway for HTTP access. This approach reduces operational overhead and is well-suited for event-driven or microservice architectures.)
* **Database:** MySQL (for all structured data, including users, jobs, applications, certifications, interviews, etc.)
* **File/Object Storage:** AWS S3 (for all resume files and version history, only metadata stored in MySQL)
* **Map/Geo Service:** Google Maps API (for geolocation display and geocoding)
* **AI Service:** OpenAI API (for AI-powered suggestions, interview preparation, etc.)
* **Job Data API:** Adzuna (the only job data aggregation source for now, extensible in the future)
* **CI/CD:** GitHub Actions (for automated testing, building, and deployment)
* **CD Environment & Infrastructure:** Provisioned and managed using Terraform
* **API Debugging & Testing:** Performed using Postman

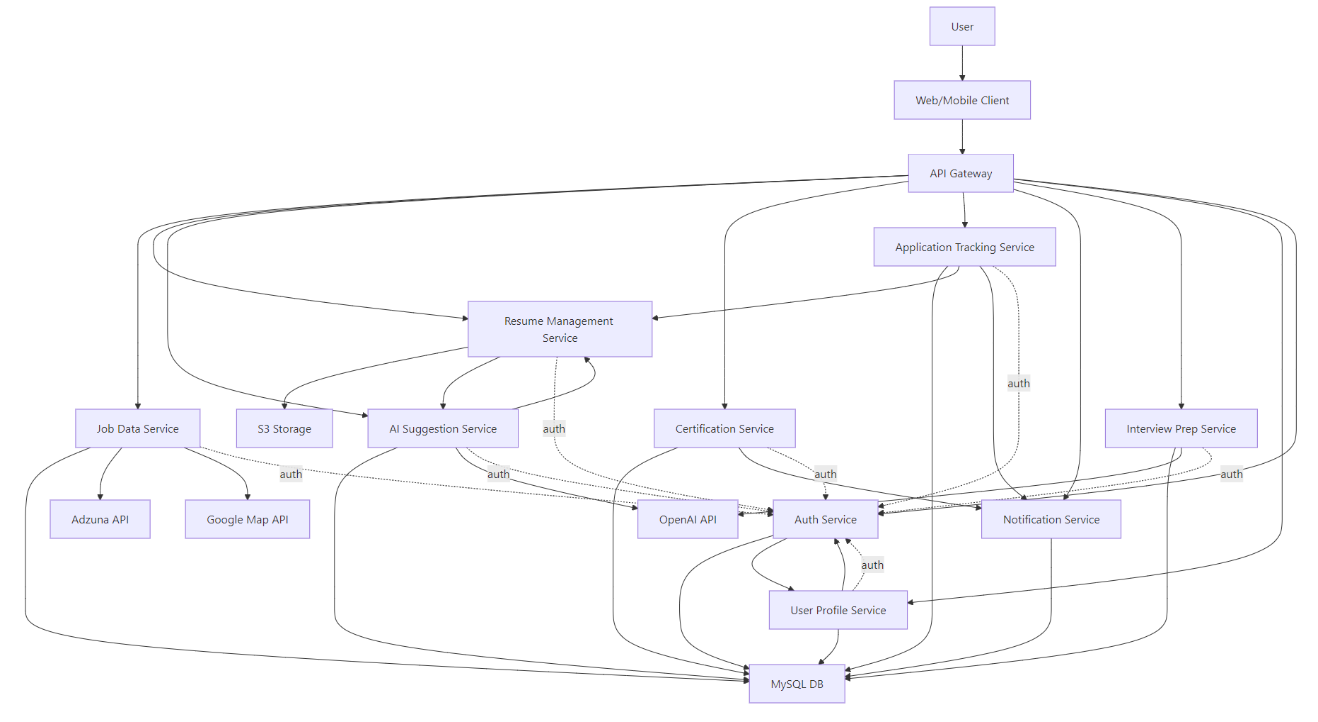
## Architecture Diagram

* **Location**

[**JobQuest\_Navigator\_CAA/00 Documents/0099 Final Documents/Architecture Decision at dev · MariaVSoto/JobQuest\_Navigator\_CAA**](https://github.com/MariaVSoto/JobQuest_Navigator_CAA/tree/dev/00%20Documents/0099%20Final%20Documents/Architecture%20Decision)

* **Purpose:**

Visual representation of the system architecture for team and stakeholder review.

****

**8. Additional Notes**

* **Version Control & Documentation:**
  + GitHub is used for project documentation and tracking weekly progress.
* **Collaboration:**
  + Jira is used for task management, sprint planning, and tracking blockers.
* **Quality Assurance:**
  + Automated testing and code reviews are integrated via GitHub Actions and team collaboration.