# JobQuest Navigator – Week 7 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

## Challenges or Blockers

|  |  |  |
| --- | --- | --- |
| **User Story** | **Task** | **Challenges or Blockers** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Plan for Next Week

Sprint: **Sprint 5**

**Sprint Dates:**15June 2025 to 22 June 2025

|  |  |  |
| --- | --- | --- |
| **Epic** | **Owner** | **Task** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 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.

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