EECS 581 Requirement Stack

Team 34

Prof Hossain Saiedian

Project 3: Initial Requirements Stack

This document contains the Agile Reference Stories and Requirements Stack for the EECS 581 Project 34 team application: AI Job Filter Search. The application matches users' skills to relevant jobs scraped from multiple online platforms. The following sections detail the reference stories used for estimating effort and the initial list of project requirements.

Agile Reference Stories Spreadsheet

The table below lists 18 reference stories across 6 story point pools (1, 2, 3, 5, 8, 13). Each story includes its source and a 25–50 word summary describing the scope and complexity.

Reference Story No.	Pool	Story Source	Story Summary
1	1	EECS 168 FA24	Developed a Python program to calculate averages and standard deviations for a list of inputs.
2	1	EECS 268 FA24	Created a basic linked list implementation to manage a list of student records.
3	1	EECS 348 FA24	Wrote a script to perform input validation and simple error handling.
4	2	Internship Project	Designed a small API endpoint using Flask to return mock data for testing front-end integration.
5	2	EECS 268 FA22	Implemented a sorting algorithm to organize data in a linked list.
6	2	EECS 581 Project 1	Built a RESTful API that returns data from a local JSON

			database.
7	3	EECS 581 Project 2	Implemented a login and authentication system using JWT and Flask.
8	3	Internship	Created an SQL database with tables for users, jobs, and applications.
9	3	Personal Project	Develop an API that sends continuous updates using a synchronous web socket.
10	5	Personal Project	Integrated a web scraper to collect code samples from multiple websites and presenting it to user.
11	5	Internship	Deployed a Flask app on AWS EC2 instance using Gunicorn and Nginx for production.
12	5	Internship	Used Flask blueprints to simplify backend code across multiple endpoints.
13	8	Internship Project	Display real time transit data in a React APP
14	8	Internship	Implemented a background worker to asynchronously fetch and process data for performance optimization.
15	8	Hackathon Project	Developed a chatbot using NLP to answer

			spoken user queries about programming questions.
16	13	Personal Project	Create a web app to allow user to rate foods using Elo algorithm.
17	13	Personal Project	Built a full-stack web app integrating Flask backend and React frontend with user authentication.
18	13	Internship	Implemented a cloud-based logging and monitoring system for production servers.

Requirements Stack Spreadsheet

The following table lists the initial project requirements for the AI Job Filter Search application. Each requirement is assigned a story point value and priority, following Agile estimation practices.

Requirement ID	Description	Story Points	Priority	Sprint No.
1	Design and implement SQLite schema for storing job listings.	3	1	
2	Develop web scraper for job listings.	8	2	
3	Create additional web scraper for additional job listings.	5	4	
4	Parse job descriptions to extract relevant skill keywords.	3	5	
5	Develop Python script to classify and normalize skill names.	5	6	
6	Implement Flask API for querying jobs by skills.	8	7	
7	Design frontend input form for users to enter skills.	3	8	
8	Implement dynamic frontend display of job search results.	5	3	
9	Integrate machine learning model to enhance skill-	13	9	

	job matching.		
10	Add authentication for users saving searches.	5	10
11	Deploy backend and database on cloud platform (AWS).	8	11
12	Implement data refresh scheduler to update job database weekly.	8	12