Grader Assignment System

by

Team 11

Madison Hokstad

Do Kyung Lee

Alexandra Ontiveros

Gaby Salazar Mejia

Anh Tran

Prof. Alagar

CS 4485.0W1

University of Texas at Dallas

February 10, 2025

Project Overview

When a new semester begins, UTD always comes across a problem with assigning graders to all the courses needed as they have to manually assess the applicants based on the criteria of the classes. This process takes a long time and affects the grading of the individual courses, that's where this project comes in. GAS - the grader assignment system - aims to solve that problem by automating the assignment process of graders to classes based on criteria that graders put into the application. This will help relieve the work of faculty and advisors at the start of each school semester.

Project Scope

1. File Handling & Data Processing

- The system must support reading and generating CSV and Excel files.
- The system must be able to process resumes exported(CSV/Excel format).
- The system must read and extract relevant data from PDF files, including:
  - Grader resumes
  - Course descriptions
  - Job descriptions for grader roles

2. Matching Algorithm & Candidate Selection

- Implement a matching algorithm that assigns graders to courses based on:
  - Course requirements (e.g., programming languages, specific coursework)
  - Grader qualifications (e.g., past experience, academic background)
  - Professor preferences
- The system must provide detailed reasoning for each assignment to allow hiring managers to verify the reliability of the match.

3. Automation & Manual Review

- Hiring managers should be able to:

    - Automate the grader assignment process using the matching algorithm.
    - Review and modify assignments.
    - Reassign graders manually in special cases.
- Professors should be able to log in and specify preferred graders for their courses.

4. Dynamic & Incremental Assignment Updates

- The grader assignment process must be incremental:
    - New courses and graders can be added at any time.
    - If an assignment is canceled, the system will automatically trigger a reassignment process based on available candidates.
    - If a new course is added, the system will integrate it into the matching process dynamically.

5. User Authentication & Role-Based Access

- The system will integrate **UTD authentication** (NetID login) for secure access.
- User roles and permissions:
    - **Professors**: log in, specify grader preferences, and review assignments.
    - **Hiring Managers**: oversee, modify, and approve assignments.

Project Objectives

- Must automate the matching process for graders
    - Via web app and backend algorithms
- Must find the optimal assignment of candidates to professors and courses
    - Via backend algorithms
    - Must test algorithms against edge cases to ensure optimal results are produced
- Must be able to run matching algorithm for single grader assignment at Professors will
- Must provide different views and flexibility for the hiring manager
    - Via web app
    - Must provide multiple views and functionality within the web app
- Must provide ease of use for all users

    - Via web app
    - Must test web app to ensure it functions as intended and use is straight-forward

Specifications

User Interface (UI) Design

- The project will be accessible through any platform as it is a web application
- The web application will have the following pages:
    - Home
    - Grader Assignment
    - Status
    - Manage
- User Interaction elements that may be present in the web application are the following:
    - Buttons are used to generate the assignment of graders to fit courses.
    - Menus are used to navigate the web application's pages or for choosing other functionalities within a page.
    - Forms are used to gather inputs for managing and modifying the assignment after reviewing

Backend & APIs

- Database:
    - Users Table(id, password, name, email, user_type, created_at)
    - Courses Table(id, name, code, professor_id)
    - Files Table(id, file_name, file_path, file_type, created_at)
    - Grader Assignments Table(id, grader_id, course_id, status, created_at)
- API: will use requests and responses to retrieve information from the database and provide it to the web page
- There will be several security considerations for whether the person logging in is a Professor or the Admin, such as:
    - Multi-factor authentication
    - Implementing account lockout policies
    - Limiting login attempts.
    - Strong password policies

    - Single credentials only for the admin user to access the database (avoiding SQL injection)

Tech Stack

- **Frontend:** React
- **Backend:** Node.js, Express.js
- **Database:** MySQL
- **Cloud & Hosting:** Vercel, Docker
- **Version Control & Project Management**: Github
- **UI/UX Design:** Figma

Hardware Requirements

- Personal computers capable of web app development

Software Requirements

- Web browser
- Node.js
- Express.js
- MySQL
- Docker
- Git
- React
- IDE capable of web app development

Project Timeline

Required to be either in phases or week-by-week schedule.

Tasks are to be segregated according to front end, back end and general categories. You can add further categories if needed.

Phase	**Duration**	**Tasks**		
		Front end	**Back end**	**General**

Phase 1	February 12 - February 24	Define scope, research, and setup (General)
Phase 2	February 25 -March 11	UI/UX design (Front end), data collection (General), API setup (Back end)
Phase 3	March 12 - April 2	Development and initial implementation (General)
Phase 4	April 3 - April 17	Testing and integration (General)
Phase 5	April 18 - May 2	Final testing, deployment, and presentation (General)

Project Team

Role	Team Member	Responsibilities
Frontend Developer	Dokyung Lee	UI development
Frontend Developer	Gaby Salazar	API Integration
Backend Developer	Alexandra Ontiveros	Database Design
Backend Developer	Anh Tran	Web Application Framework
Backend Developer	Madison Hokstad	Testing & validation

Team Leader Rotation

Date	Team Leader	Date	Team Leader
2/14	Alexandra Ontiveros	3/28	Dokyung Lee
2/21	Alexandra Ontiveros	4/04	Gaby Salazar
2/28	Anh Tran	4/11	Gaby Salazar
3/07	Anh Tran	4/18	Madison Hokstad
3/14	Dokyung Lee	4/25	Madison Hokstad
3/21	Spring break	5/02	Alexandra Ontiveros

Links

- **GitHub Repository:** https://github.com/catally3/grader-assignment-system.git
- **Project Management:** https://github.com/users/catally3/projects/1/views/2
- **Design Document:** https://www.figma.com/design/WSsbU6rMs9asbQ9c95GT7S/Untitled?node-id=0-1&t=Q6vonct8qV7bDIK6-1