Grader Assignment System Proposal

Team-Member

Name: Havish Chaganti
E-mail: hxc200026@utdallas.edu
Phone: 469-400-4737

Name: Yosep Ha
E-mail: YXH180059@utdallas.edu
Phone: 714-337-5596

Name: Haeun Kim
E-mail: HXK200042@utdallas.edu
Phone: 972-836-5975

Name: Sanved Paladhi
E-mail: SXP210288@utdallas.edu
Phone: 210-355-3075

Project Overview

The **Grader Assignment System (GAS)** is a web-based application aimed at automating the grader selection process for university courses. The current process is manual, time-consuming, and lacks optimization, leading to inefficiencies in candidate matching. The proposed system will streamline the workflow by automating candidate filtering, optimizing grader-course assignments, and allowing hiring managers to make final adjustments. By reducing processing time and providing transparency in selection criteria, the system improves fairness and efficiency in grader hiring.

Project Scope

Core Functionalities:

- Automate candidate filtering based on job description criteria (e.g., GPA, major, etc.).
- Implement a matching algorithm to optimize grader-course assignments.
- Provide reasoning/explanations for each match.
- Support input and output of CSV, Excel, and PDF files.
- Offer a web interface for managing and reviewing assignments.

Additional Features:

- Add professor preferences for specific graders.
- Handle returning graders from previous semesters.

Project Objectives

1. Automate the grader selection process to reduce manual effort and improve efficiency.
2. Optimize grader-course matching using a rule-based algorithm
3. Highlight expected outcomes and deliverables.
4. Enhance transparency by providing clear reasoning for each match.

Specifications

User Interface (UI) Design

· Website based platform

Login for Professor

Upload resume

Upload candidate list

upload course professor /requirements

- **Screen that allows user to input cv data**

    - Buttons: Submit, Upload CV, Cancel
    - Upload Button (Upload Excel file)
    - Buttons: Start Matching, Cancel

- **Screen that displays the matches and their reasons why**

    - List of assigned graders per course
    - Matching Score Breakdown (e.g., GPA +3, Major Match +5)
    - Buttons: View Details, Approve, Reassign

- **Screen for verification and edits to selection**

    - Finalized list of graders & their assigned courses
    - Downloadable CSV/Excel file
    - Buttons: Export Excel, Download Report
    - Make manual edit available to selections

Backend

Reading in the file and processing it

- Get CV data, compile candidates into csv excel file
- Gather the data from pdf file and store it into database
    - Python to read the pdf data files and store it in a database

Database programming

- Extract data of eligible candidates who follow the category we require from all list of candidates using MySQL

Matching process, implement an algorithm

- Compare the excel having the data of eligible candidates with the excel having the data of the professors who require graders and implement an excel sheet having data of the candidates who are eligible to be graders for a particular professor using an algorithm

Manual edits

- Allows the professors to make changes to the file.

Output data as a csv file

- Output the data as a csv file.

Tech Stack

- **Frontend:** React.js
- **UI/UX:** Figma
- **Backend:** Node.js, Express
- **Database:** mongodb or mySQL

Hardware Requirements

- Standard development laptops/PCs.

Software Requirements

- Specify required tools, SDKs, and platforms.
- Version control: GitHub

Project Timeline (Weekly)

Phase	Duration	Tasks		
		Front end	**Back end**	**General**
Phase 1 Team Leader: Havish Chaganti	[2/17/25] - [2/23/25]	• Sketch rough layouts before converting it to digital wireframe ○ Identify key elements • Use Figma to refine the sketches • Design a hello world page in local machine/ learn components on react • Research how to display data on web page	• play with databases (setup mongodb and mysql. CRUD operations) • play with strings, compare them, find relations • Setup python environment	• Define scope, research, and setup • Work on group proposal • Learn what fullstack is
Phase 1 Team Leader: Havish Chaganti	[2/24/25] - [3/2/25]	• Continue on Figma ○ Website design • download from files backend to frontend	• Import data from database table into excel file • Write python that loads pdf files, and code the separates each candidate	• UI/UX design • Research on tech stack to use • Work on group proposal

Team Leader: Haeun Kim	[3/3/25] - [3/9/25]	• Finish figma design • Start to code the structure of the web pages	• Work on setting up server • Import data from excel file into database table • Write python that loads pdf files, and code the separates each candidate	Development and initial implementation
Team Leader: Haeun Kim	[3/10/25] - [3/16/25]	• Work on web pages to match figma • Finish login page		
Team Leader: Haeun Kim	[3/17/25] - [3/23/25]	• Finish upload page • Login authentication (tbd) •		
Team Leader: Yosep Ha	[3/24/25] - [3/30/25]	• Manual upload page		
Team Leader: Yosep Ha	[3/31/25] - [4/6/25]			
Team Leader: Yosep Ha	[3/31/25] - [4/13/25]			Testing and integration
Team Leader: Sanved Paladhi	[4/13/25] - [4/20/25]			
Team Leader: Sanved Paladhi	[4/20/25] - [4/27/25]			Final testing, deployment, and presentation slides

Team Leader: Sanved Paladhi	[4/27/25] - [5/2/25]			Finish everything

Team Leader

- Monitor team progress

[2/10/25] - [3/2/25]: Havish
[3/3/25] - [3/23/25]: Haeun
[3/24/25] - [4/13/25]: Yosep
[4/14/25] - [5/4/25]: Sanved

Project Team

Role	Team Member	Responsibilities
Frontend Developer	Haeun Kim	UI development
Frontend Developer	Yosep Ha	Front-end logic & API
Backend Developer	Sanved Paladhi	Database design
Backend Developer	Havish Chaganti	Server-side logic & API

Links

- **GitHub Repository:** GitHub Link
- **Agile Board:** [Insert Link]
- **Design Document:** [Insert Link] (If you are using tools like canva, figma, etc...)