MENTORING REPORT

IU DSIP FALL'23

12-08-2023

Prepared by:
Sravya Vujjini
Anas Omary
Julia Nguyen
Abhinav Krishna Narvaneni
Thomas Koutsidis

INTRODUCTION

The IU Foundation Mentoring program helps employees grow and learn together. The current process of matching the mentors and mentees involves a manual process. Our project aims to create an automated system that collects data and matches mentors and mentees automatically.

EXECUTIVE SUMMARY

This section presents our key recommendations. Further elaboration on the rationale and detailed findings is provided in the subsequent sections below.

Processes automation:

After evaluating various tools, we have determined that the combination of PowerApps and PowerAutomate by Microsoft is the most suitable solution for addressing most problems identified, for the following reasons:

- Reduce Manual Work and Human Errors: Automate the extraction and processing of inputs, minimizing manual efforts and mitigating potential human errors. Then output a spreadsheet containing all pertinent information for the matching process.
- Workflow Setup for Applicant Notifications: Establish a workflow where applicants automatically receive a copy of their questionnaire response upon submission.
- Auto-Notification Capabilities: Enable the setup of ongoing auto-notifications for distributing mailing notices to mentors and mentees based on their submission dates.
- Auto-Storage Capabilities: Automatically save copies of responses by month to designated cloud folders via SharePoint.

Data collection:

- Incorporate the estimated survey completion time to incentivize sign-ups and discourage procrastination. For example:
    - Complete this **5-min** survey to enroll in the program, or
    - This survey will only take 5 minutes.
- Include an availability query in the survey to aid in the pairing process and

demonstrate consideration for participants' schedules. For example:
    - How many hours are you able to mentor:
      a. 1-2 hours/week, b. 3-4 hours/week, c. 5-7 hours/week
- Add a question for the mentor: *How many mentees would you like to have?*

- Remove the additional skills free-text field.

SDS report:

- Option 1: Execute the PDF scraper file in the Commander environment where all SDS reports are stored. This option allows you to generate a report at any time without relying on others.
- Option 2: Reach out to SDS admin (sds@parinc.com) – POC: Mary Hanson, to request a copy of the report, as they currently lack the capability to generate and distribute a monthly report.

Improve Awareness and Engagement:

- Consider implementing a communication plan from Leadership, proven to be most effective, especially in departments where employees possess most in-demand skills.
- Highlight the incentives for mentors and articulate the benefits for mentees.

Other:

- To address the mentor shortage in the interim, offer internal training on the top 3 most sought-after skills by mentees including: Fundraising Principles, Conflict Management, Building Relationship.

PROBLEM STATEMENTS

After several discussions, we've identified the following problems:

1. **Data collection**
    1.1. The manual data input process into an Excel Sheet is inefficient, time consuming and susceptible to human error.
    1.2. Extracting results from newly submitted forms involves a multi-step, time consuming process.
    1.3. The Excel matching sheet does not incorporate the 3-letter personalized summary code from the SDS survey, which is provided by a 3rd-party

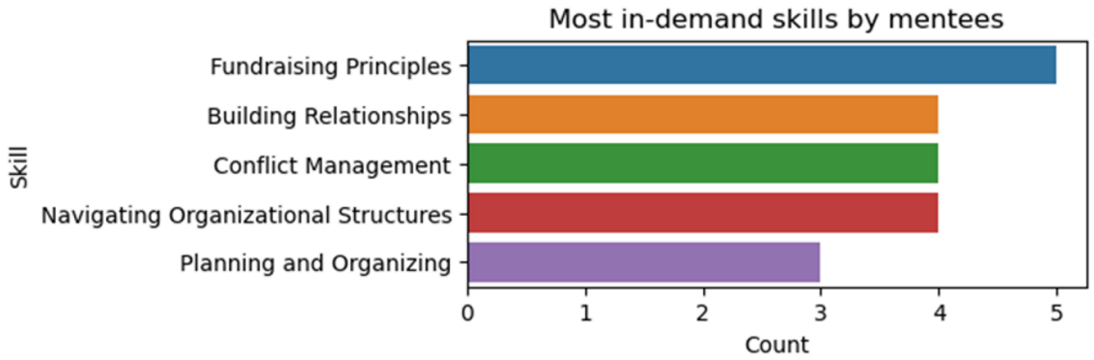
supplier, limiting comprehensive consideration.

2. **Matching**
   - 2.1. The manual process of matching mentors with mentees is time consuming, non-scalable and grows exponentially with the number of participants
   - 2.2. Matching SDS results with mentors/mentees is labor-intensive and requires manual logging
   - 2.3. Matching additional skills with existing skills is a subjective, manual process, leading to unnecessary skill list inflation or duplicate entries.
3. **Maintenance**
   - 3.1. Manual categorization of responses based on the month is time consuming, error prone and difficult to maintain.
   - 3.2. There is no system to inform mentors/mentees if a match has not occurred.
4. **Non-technical**
   - 4.1. Lower-than-expected rate of participation in the mentoring program.

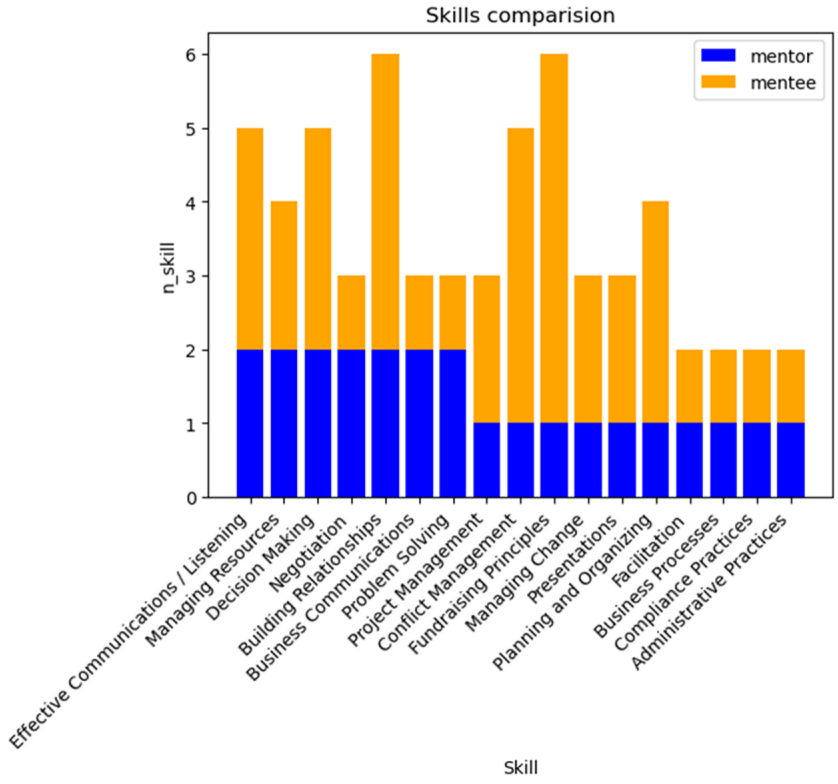
DESCRIPTIVE ANALYSIS

While the dataset is relatively small (n=17), we've conducted analyses revealing some valuable insights.

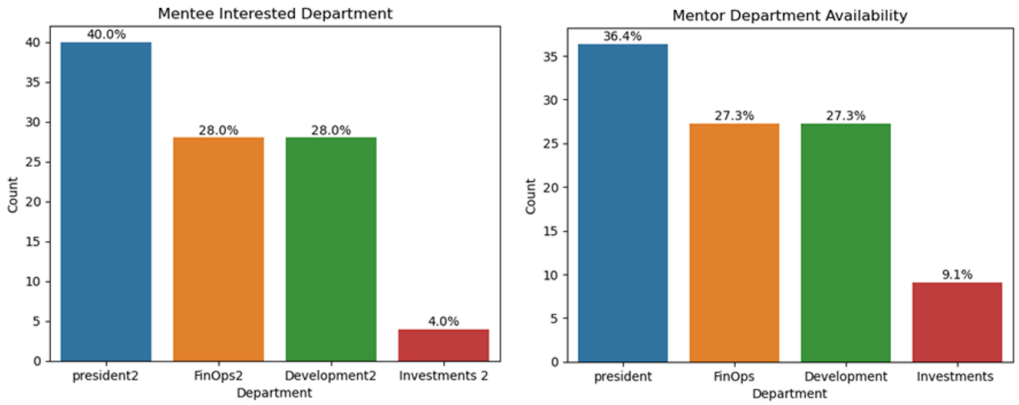
The top 5 most in-demand skills identified by mentees are as follows:



The chart below illustrates the misalignment between the skills mentees are interested in and the skills possessed by current mentors. Notably, gaps are prominent in the following areas: Fundraising Principles, Conflict Management, and Building Relationships.



An overview by departments:



It is encouraging to note the presence of subsequent mentors in most departments where mentees express interest. In the investment department, there appears to be an excess of available mentors. A detailed examination of the skills prevalent in the investment department includes:

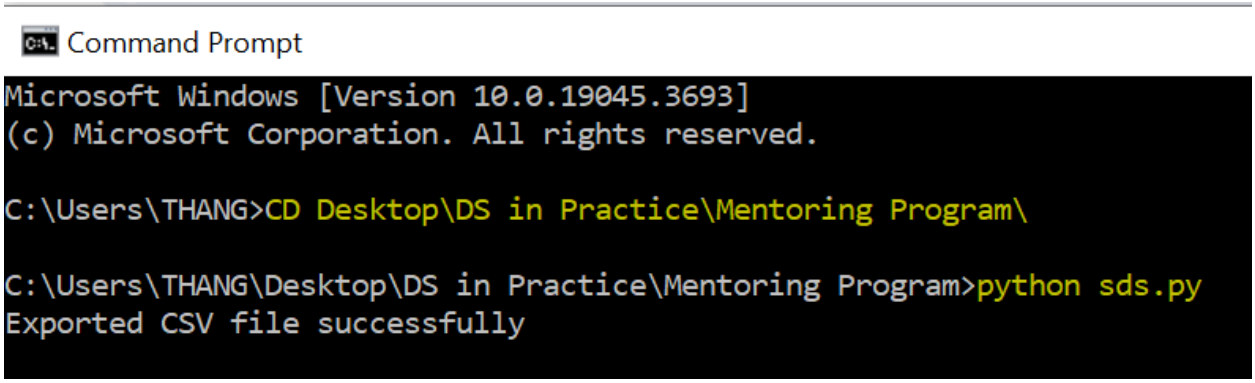
Managing Change	Conflict Management	fundraising principles;
Building Relationships	legal	navigating cultural
Branding and Marketing	planned giving;	differences;

building relationships; navigating organizational structures Compliance Practices Project Management	Building Relationships Effective Communications presentations	Managing Resources Planning and Organizing Administrative Practice

2. Extracting SDS Codes:

Here are the instructions for running the SDS code extraction script:

- Save the Python script in the folder where you store all PDF SDS reports.
- Open Command Prompt.
- Navigate to the folder using the command: cd file_path (For example: cd Desktop\Mentoring\).
- Execute the script by entering: python sds.py



A spreadsheet containing all the extracted codes will be generated in the same folder.

3. MS FORM, POWER APP & POWER AUTOMATE

As stated, existing process included many manual steps leaving it susceptible to human error, there were many challenges in data that could be summarized in:

1. **Manual Data collection:** data is being collected through Microsoft forms, which is ok if the options presented to the user do not change, but in the case of mentor\mentee registration form, the skill list can change, the departments can change, which meant the user interface (i.e. the MS form) had to change and that by it self poses a challenge for users, adding to all of that, after the user chooses

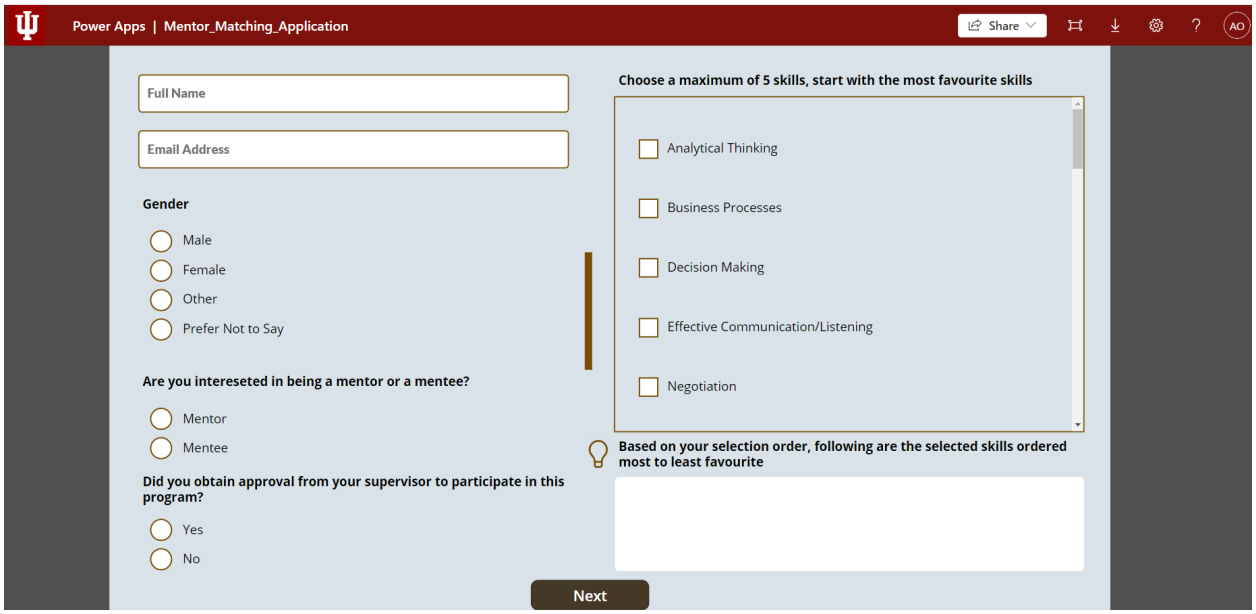
their desired skills from the skill list, the user has to, again, manually enter their desired skills in the form of free text to order them from most to least favorite, many of the work that is done by the sponsor's team depends on these manually entered skills being ordered correctly and above all, entered correctly without typos.

2. **Existing MS Forms:** In the initial phases of our project, MS Forms were utilized to collect responses from the users. While MS forms successfully gathered responses, we encountered a blocker in the prioritization of the skills due to the lack of dynamic listing functionality, making it challenging to create priority lists of the skills based on the user's preferences.
3. **Manual Data consolidation and transformation:** the output of MS forms was being inserted in an excel file, users who filled their form and submitted it were not receiving an automated confirmation that their application was received, instead the sponsor's team were sending those confirmation emails manually. Also, after the data has been collected, someone has to take the input and manually re-enter applicants information and order their chosen skills.
4. **Manual data analysis:** after the data has been collected, re-entered, and ordered someone has to look at the ordered skills and manually match mentees with mentors based on the order of skills.

In summary, the entire process was carried out in a manual fashion, even the Ms forms were not dynamic enough and not future proof.

Queue Power Apps & Power Automate

We basically want to automate the entire matching process and create a user interface that is dynamic and future proof, for data collection interface we chose to combine two of the most powerful microsoft automation tools; Power Apps and Power Automate

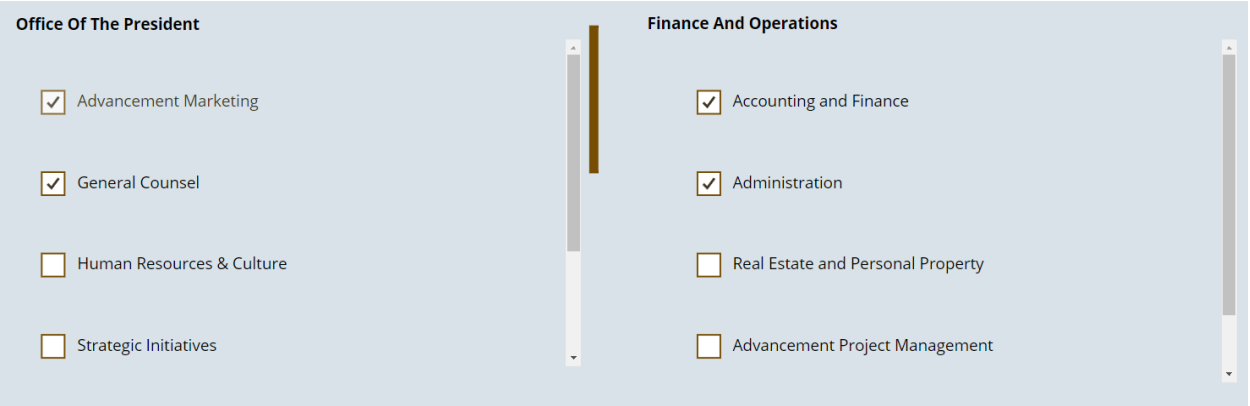


We chose power automate because it is far superior to Ms forms in terms of customizability, because it is possible in power apps to write code to make the application basically perform any data collection task, following are the improvements we gained over Ms forms once we developed the power app:

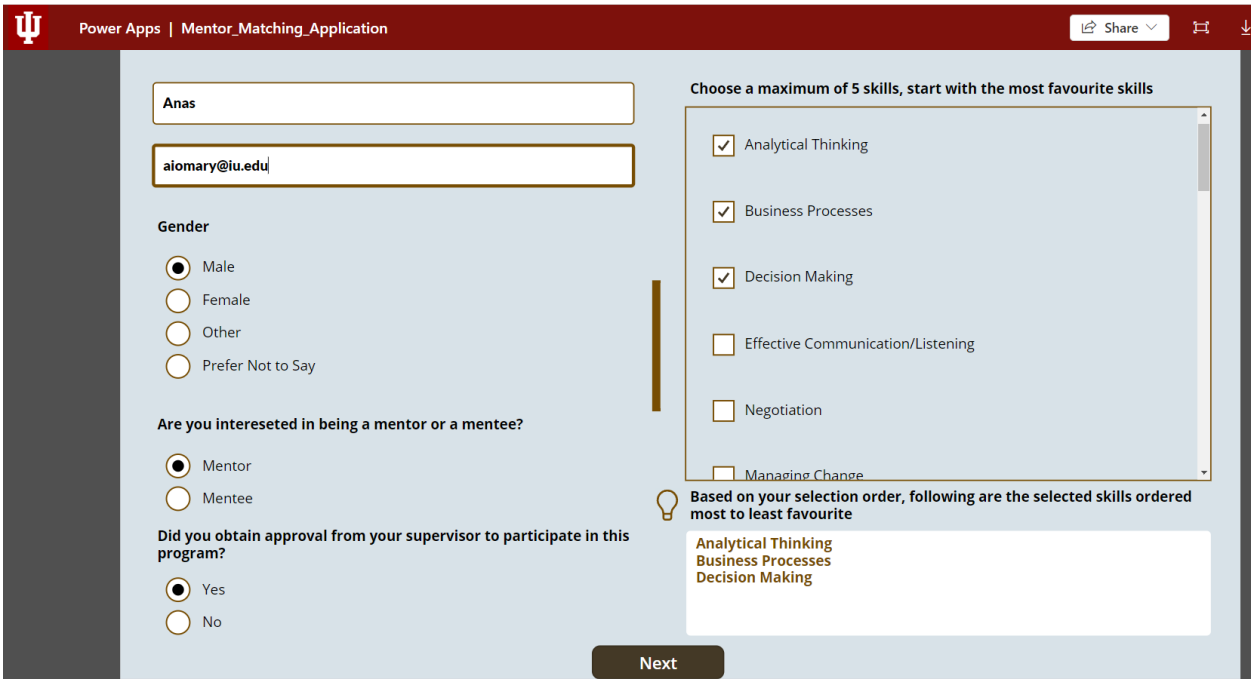
- All checklists became dynamic, there were 5 checklists that were static in Ms forms, that meant if a new item is to be the list, someone has to edit the user interface, using power apps, we created an excel sheet with all checklists and made the power app read checklist items from the sheet dynamically, now if the user wanted to change any checklist, it is just a matter of opening that excel sheet and change any list they wish.

	A	B	C	D
1	Office Of The President	Finance And Operations	Development	Investment
2	Advancement Marketing	Accounting and Finance	Advancement Services	Investment Operations
3	General Counsel	Administration	Affinity Giving	Investment Research
4	Human Resources & Culture	Real Estate and Personal Property	Development (Bloomington)	
5	Strategic Initiatives	Advancement Project Management	Development (Indianapolis)	
6	Donor Relations & Stewardship	Strategy and Analytics	Gift Planning Services	
7	Advancement Events		Principal Gifts & Transformational Giving	
8				

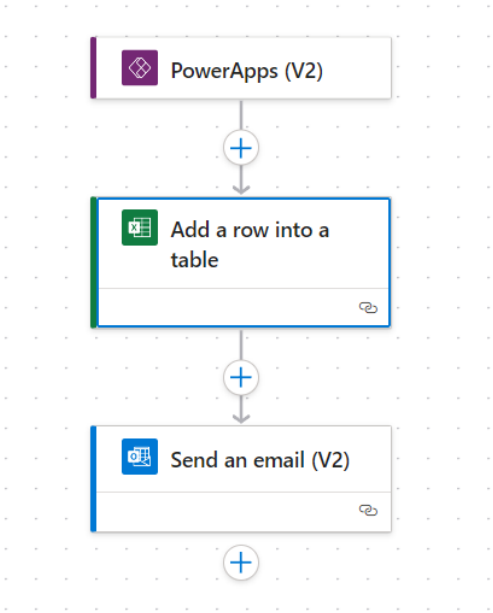
-



- No more manual entry and ordering of selected skills, whenever the user selects a skill, that skill is added to a text box automatically and in the same order in which the skill was selected, the text box is not editable, this way we eliminate any possibility of typos.



- Automated user notification, using power automate we created a workflow that will send an automated confirmation email to the user containing the user input once the user clicks submit.



Once we developed the power app and power automate flow and started collecting user input in an excel sheet on one drive, we started automating the rest of the process.

Feeding the python..data

Most of manual work of the sponsor's team starts after the receiving applications, to automate all manual activities, we created two python programs, the first one reads all user input and transforms it over multiple stages and outputs an exact replica of the matching grid which used to be a cumbersome task that took big team efforts.

Application Type	Submissions	Email	Date	Analytical Thinking	Business Processes	Decision Making	Effective Communication / Listening	Negotiation	Managing Change	Data Analytics / Literacy	Problem Solving	Managing Resources	Project Management	Conflict Management	Using Financials	Presentations	Collaborating with Others	Compliance Practices	Legal Considerations	Fundraising Principles	Real Estate Practices
Mentee	My name	myemail@skjfh.sdfsdf	2023-11	2				3		4	5										
Mentee	Julia Test	julinguy@iu.edu	2023-11		2		3	4		5											
Mentee	Sravya Vujjini	svujjin@iu.edu	2023-11	1	2			3			5										
Mentee	Julia Test 2	julinguy@iu.edu	2023-11	1	3		5														
Mentee	testing test	testingtest@xyz.com	2023-11	2	1							3						4			
Mentee	Julia Test 3	julinguy@iu.edu	2023-11	2																	
Mentee	Sravya test	sravya.vujjini07@gmail.com	2023-11	1				3		4											
Mentor	Anas new	anasnew@sdfsdf.com	2023-11	2	3		5														
Mentor	Dodo	Fff@hyu.vgh	2023-11	2			5	4													
Mentor	Multi Screen	lkjsldkf@sdfsdf.com	2023-11	2	3		5														
Mentor	testing test2	testingtest2@gmail.com	2023-11	1				3		4											
Mentor	Anas Omary	anas@iu.edu	2023-11	2	3		5														

After automating the matching grid we continued our automation journey by automating the process of matching mentees with mentors, which was strictly a human dependent process where the sponsor's team were using the order of skills selected by each mentor\mentee to match which meant if we have 10 mentors each selected 3 skills and

10 mentees each selected 3 skills, that meant 10*3*10*3=900 possible matching options!.

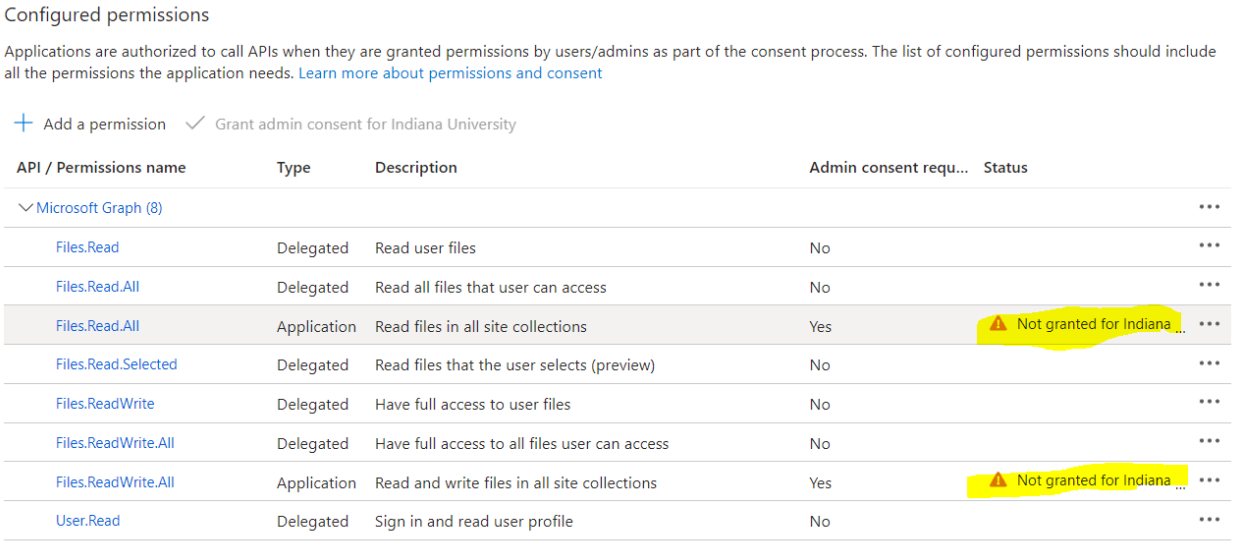
We created a python program that reads all ordered skills for all mentors and mentees and outputs the percentage of match between each mentor and each mentee, in the following screenshot we can see mentors are listed in the first column and every column after that represents a mentee.

Submissions	Mentee1	Julia Test	ravya Vujji	Julia Test 2	esting test	Julia Test 3	Sravya test
Anas new	25.0%	0.0%	0.0%	66.666666	25.0%	100.0%	0.0%
Dodo	25.0%	25.0%	0.0%	33.333333	25.0%	100.0%	0.0%
Mentor 11	25.0%	0.0%	0.0%	66.666666	25.0%	100.0%	0.0%
Mentor 9	50.0%	0.0%	50.0%	33.333333	0.0%	0.0%	100.0%

Limitations:

We faced some limitations that prevented us from presenting an end-to-end ready for production solution, following are those limitations:

- Working with a free version of power automate meant we can't add python script to the power automate flow, this is why we had to improvise and create python script that runs outside the power automate flow.
- Ideally, python program should read mentor\mentee user input from its source, the user input excel sheet located in onedrive, but due to not having access rights to one drive, we had to work with the sheet offline:



4. MATCHING ALGORITHM

Approach 1:

We wrote a Python program that matches mentors with mentees based on their skills and shared interests. The algorithm calculates a compatibility score for each mentor-mentee pair by considering the skills they have and the similarity in personality (SDS summary codes) they share. The higher the compatibility score, the better the match. The program then ranks these pairs and suggests the top matches, making sure each mentor and mentee is used only once. The codes are attached here for more details.

Approach 2:

This algorithm compares the skills of a mentor and a mentee by first extracting the non-null skills for each individual. It then identifies the common skills shared between the mentor and mentee, and calculates a matching score based on the number of common skills in the same order. The matching percentage is determined by dividing the score by the total number of skills possessed by the mentee, providing a quantitative measure (percentage of similarity, the higher the better) of the similarity in skills between the mentor and mentee.

5. SUGGESTIONS FOR FUTURE WORKS

We were limited by the time we had for this project, there were some areas we knew can be improved further but we were focused on automating manual work and decided to complete that first, following are some of the areas that we suggest you working with future DSIP teams on:

- Retrieving permission access to PowerApps from GitHub which Paul is the owner.
- Continuing setting up PowerApps to minimize the technical works for sponsors.
- Excluding mentees that were matched with mentors from the matching grid to have the ability of storing this can be done by adding a status column or by removing matched mentees into another sheet as historical data that can be used for further analysis.
- Adding the automated workflow to send follow-up emails to applicants based on a predefined time period.
- When you have sufficient data on past mentor-mentee relationships and their

matches, consider using machine learning techniques to optimize the pairing algorithm. Train the model on historical data to predict successful pairings.