**Education PlanIt**

Gary D. Angdoko, Eugene A. Babu, Chia H. Hou, Saahas Hari, Sarah L. Luongo, Bharathi K. Ravikumar, Karthik Talluri

Tepper School of Business, Carnegie Mellon University

46-991: MSBA Experiential Learning

Dr. Ganesh Mani

November 4, 2023

# Table of Contents

## 

[**Table of Contents 0**](#_oscl7oa7xvu)

[**2. Project Background: 2**](#_5gm7txwrux5v)

[**3. Project Description 3**](#_q6kp02b30ro6)

[**4. Methodology 6**](#_2ngffocdtc1z)

[4.1 Data Collection: 6](#_oyt33i8lbplq)

[4.2 Google Palm API 7](#_7urnubs7nds8)

[4.3 User Interface 8](#_9focdal438fo)

[**5. Results 9**](#_mmni4mioiqu4)

[**6. Issues and learnings from the Project 14**](#_mgv3emm82voz)

[6.1 What we learned: 14](#_ipo7m9t4y7ea)

[6.2 Issues our team face (Negative results / findings) 14](#_pgkjzlrlyczd)

[**7. References 16**](#_tox0ur4a3vnt)

## 

# 2. Project Background:

Emerging from an inspired brainstorming session at the distinguished Google Palm2 API Hackathon hosted by Carnegie Mellon University, Education PlanIt epitomizes the fusion of innovative thinking and the essence of teamwork. What sets our model apart is the unique training on our specially curated dataset, ensuring it delivers insights that are both relevant and tailored to its users. By leveraging the cutting-edge capabilities of Google's Palm2 API, our dedicated team has crafted a tool poised to redefine the way middle school students envision their future careers.

Education PlanIt is a pioneering career guidance tool, meticulously designed for young students uncertain about their future endeavors. By simply entering statements reflecting their strengths, habits, and interests, students are guided towards potential career paths. More than just a career exploration tool, Education PlanIt makes the journey engaging and relatable, ensuring students are empowered with well-informed decisions, setting them on a trajectory towards successful future careers.

Driven by a profound commitment to student success, our team's mission is to offer a clear lens into potential career alignments based on individual strengths and areas of growth. We aim not just to suggest careers but to illuminate the path forward. This includes providing students with comprehensive step-by-step guidance, highlighting essential resources, and ensuring they are well-equipped with the knowledge needed to excel in their chosen fields.

# 3. Project Description

Education PlanIt is a dynamic model designed to gauge students' current interests. Utilizing the capabilities of the Google Palm2 API, our model interprets inputs—whether they're bullet points, sentences, or paragraphs—into a skill set that the LLM can comprehend. For enhanced accuracy, we've aggregated matching qualities from various online sources for 21 distinct job titles and have trained the Palm2 API using Makersuite. Our primary aim is to offer users a detailed guide, providing step-by-step recommendations on courses to pursue at their universities and suggesting the most suitable universities based on their school rankings.

Once a user provides their input, our system springs into action, segmenting that singular input into five distinct processes.

1. **Job Role Suggestions**: The initial process ingests the user's input and, drawing from our trained dataset, recommends five job roles that align with the qualities mentioned by the user. Currently, our database comprehensively defines 21 job roles, while the remaining are sourced from the Google Palm2 API.
2. **Job Role Description**: The second process is designed to illuminate the nature of the suggested job roles. It articulates the roles using simple, relatable language and provides illustrative examples of tasks associated with each job. Given our target audience of middle school students, we prioritize clarity and simplicity over technical jargon to ensure easy comprehension.
3. **Course Recommendations**: The third process directs users towards the foundational courses that would benefit them in their prospective job roles. Instead of delving into specific course numbers or titles, our model offers a generalized overview of the typical subjects or areas of study one should pursue to excel in the suggested profession.
4. **School Recommendations**: The fourth step provides users with a list of the top three schools, complete with direct links to their admission programs. We aim to simplify the search for our users, ensuring they navigate directly to the desired program and avoid any confusion due to similar program names.
5. **Salary Range**: The final step presents the estimated salary range for the suggested job title based on current market data. This range is sourced from online databases and represents reported salaries. It's important to note that actual salaries can vary, and the provided range serves as a general guide. Our intent is to give users an idea of potential earnings, as this can be a determining factor in their career choices.

Our current user interface includes a search box for users to write their interests. After selecting search it will result in five potential jobs based on those interests, returning the five prospects listed earlier: job roles, job descriptions, courses, colleges, and salaries. The UI is simple, providing a clear and structured pathway for users to understand the qualifications needed for each job and the financial prospects they offer. A screen recording demonstrates how a user might interact with our tool [turned in with this assignment].

Our project is currently in a developmental phase, continually evolving towards its envisioned version. Our primary aspiration is to provide unfettered access to premier free online courses and resources, enabling students to effortlessly connect with esteemed resources, such as Harvard's renowned CS50 course—a staple in the programmer's academic journey. Understanding the profound implications of our recommendations, we're committed to monthly updates, ensuring our users remain abreast of the latest free educational opportunities.

Exploring into the realm of university admissions, our current framework showcases the top five institutions for each profession, detailing its top programs. However, our ambition extends to providing more intricate insights. We aim to elucidate aspects like essay prompts, standardized test score benchmarks (e.g., SAT or TOEFL), and specific recommendation letter requirements. Moreover, by analyzing the profile of an average admitted student, we hope to gauge and project an aspirant's likelihood of gaining admission, based on their provided credentials and information given to the model.

In terms of adaptability, we're endeavoring to augment the model's flexibility. Our vision is for users to refine their queries based on their targeted educational tier—whether that's a master's program, an undergraduate stage, or a vocational college degree. Recognizing the unique nuances and prerequisites of each educational level, our objective is to curate a guide tailored to every individual's trajectory. This adaptability extends to the model's learning capabilities. If a user, for instance, identifies as a business analyst when the model suggests a data analytics role, our system should be receptive, learning and recalibrating based on this feedback.

We aim to refine our user interface in the future to be more intuitive and engaging, particularly catering to our target audience of middle school students. Our goal is to develop a straightforward, visually appealing platform where students can input their interests and receive tailored, easy-to-understand results. We plan to integrate specific design elements, for example medical icons or imagery for interests that may result in healthcare careers. Ultimately, we envision a versatile UI that is usable across multiple devices, setting the stage potentially for future app development that students can access from anywhere, simplifying their journey of exploration and discovery.

Lastly, in this era where companies are continuously innovating job titles to mirror specialized roles and responsibilities, our model aims to stay ahead of the curve. We're working towards enabling it to sift through novel job listings, discerning and extracting crucial data such as required skills and qualifications. Our end goal is to pave a clear pathway for students, equipping them with the precise education and resources essential to emerge as formidable contenders in their chosen field.

# 4. Methodology

Our team has adopted a two-pronged approach to complete this project. Firstly, we engage in comprehensive data collection from various sources. Secondly, we leverage the capabilities of Google's latest Palm2 API to generate the desired output for our users. This output encompasses five key components: Job Titles, Job Descriptions, Course Recommendations, School Recommendations, and Salary Range.

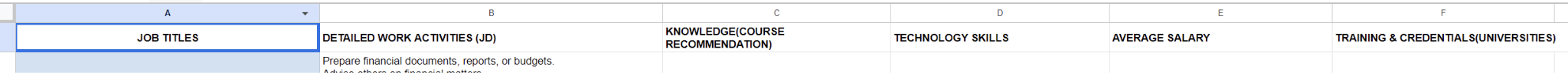
## 4.1 Data Collection:

Facing the pressing challenges of limited time, our team strategically chose to delve deep into 21 specific job titles, meticulously identifying the core qualities and attributes that are in high demand by companies today. Our mission was to cultivate a robust and comprehensive dataset, an endeavor that led us to cast our net across a diverse array of esteemed websites. Each of these platforms was selected for its authoritative stance in the realm of career information and job insights:

1. Onetonline(<https://www.onetonline.org/>)
2. Bureau of Labor Statistics (BLS) Occupational Outlook Handbook (OOH) (<https://www.bls.gov/ooh/>)
3. My Next Move(<https://www.mynextmove.org/>)
4. CareerOneStop(<https://www.careeronestop.org/>)
5. JobHeros(<https://www.jobheros.com/>)
6. Vault (<https://www.vault.com/>)

Going forward on this data collection journey, we committed to rigorous manual data entry for each job title spread across these six platforms. Initially, we wanted to do web scraping, with its promise of efficiency. But, it didn’t work even with help. We sought to harness the capabilities of the Palm API for this. However, the real-world intricacies of web scraping — from the seemingly ever-present CAPTCHAs, erratic page responses,and often frustratingly inconsistent data formats — compelled us to adopt a hands-on, manual approach.

Beyond merely cataloging job titles, we immersed ourselves in each site, extracting crucial attributes that paint a holistic picture of every job. This meant cataloging abilities, preferences, unique working styles, essential skills, nuances of the work environment, and several other vital datasets that would be indispensable for our end goal. Once gathered, we meticulously structured our expansive findings into an organized CSV file, ensuring each piece of data found its rightful place in clearly defined columns. We are fueled by the conviction that this diligently assembled dataset will not only enhance the training process of the API but will significantly elevate the final model's efficacy and accuracy for our project.

At the conclusion of our meticulous data entry, our CSV file was structured into five distinct sections:

1. **Detailed Work Activities**: This column outlines the core activities associated with each job role. While specific responsibilities might vary across companies, this compilation captures the foundational tasks typically expected on a day-to-day basis.
2. **Knowledge / Course Recommendations**: This section delineates the essential knowledge base required for the job. It offers a curated list of topics and areas one should be well-versed in to thrive in the given role.
3. **Technology Skills**: In this section, we detail the specific software or tools crucial for the job. It highlights proprietary applications and platforms that one should master to excel in the role.
4. **Average Salary**: Based on our research from reliable sources, this column provides an estimated average salary for each position. It's important to note that it doesn't present a salary range or standard deviation, making it a general figure that might align closely with certain individuals' experiences.
5. **Training & Credential (Universities)**: This section indicates the educational and training prerequisites for the role. It sheds light on whether a degree or specific training is essential to perform competently in the job.

## 4.2 Google Palm API

After meticulously preparing our CSV file, our team transitioned to the next pivotal phase: training the model. For this, we utilized Google's Makersuite, a powerful tool designed for such advanced applications and used the text-bison model.

Upon uploading the file to Makersuite, we were presented with several customizable parameters critical for model training: tuning epochs, learning rate, and batch size. These parameters dictate how our model learns from the data.

1. **Batch Size**: Given the relatively modest size of our dataset, we opted for a batch size of 8. It's crucial to understand that batch size impacts the granularity of model updates. With a larger dataset, we'd have considered increasing the batch size to 32. It's conventional to use batch sizes that are powers of 2, typically ranging between 16 and 512, to optimize computational efficiency.
2. **Learning Rate**: Set at 0.002, our chosen learning rate ensures the model updates its understanding in measured increments. A suitable learning rate is vital as it influences the speed and quality of learning. Too high, and the model might overshoot optimal solutions; too low, and it might get stuck or learn extremely slowly.
3. **Tuning Epochs**: We established our tuning epochs at 10, striking a balance between sufficient model training and computational efficiency. An epoch represents a complete presentation of the dataset to the model.

Incorporating these parameters, we set our API to learn on the dataset. This structured approach is designed to harness the full potential of our data, fine-tuning the model for optimal performance.

After fine tuning our model, our team plunged into a meticulous phase of experimentation. Our objective was to identify the optimal prompts that would resonate with our refined model and yield the precise results we envisioned. By engaging in a series of tests, using varied prompts for each distinct function, we discerned a pattern. Basic, simple with well-worded sentences and clear parameters emerged as the most effective prompts for eliciting the desired outputs from our model. Through this iterative process, we ensured that our platform would provide results that closely align with our aspirations and the needs of our users.

## 4.3 User Interface

For the user interface, we have chosen to use the Flask framework, which can easily integrate with JavaScript and HTML. Given that our backend code is all written in Python, this made for a logical choice. Our team wanted to provide a frontend interface that would be easy for users to view given the nature of blue-light sensitivity, Therefore, we decided to make the entire website dark-mode, which also provided contrast to the text boxes we used. We used the following overarching elements in our frontend infrastructure construction:

1. **Flask Routing:** For the Flask routing, we determined a main method that we could use as the crux of our data collection and retrieval. Centralizing the routing allowed for an easy depiction of all the information we collected in a table format on the user interface.
2. **HTML Document:** The HTML Document is created with multiple containers and elements to accurately represent and depict all the information collected from the PaLM 2 API.
3. **CSS Styling:** The CSS styling mainly focused on the color of the page and the overall design of what we want the page to look like. We believed that the table organization allowed for a strong organizational scheme that can be easy to view. By using the contrasting colors, it provides more attraction to using the platform, and the platform’s design elements are also more scalable.
4. **JavaScript AJAX controller:** The JavaScript AJAX controller is mainly to create the table infrastructure and how we load the data through functions. It is also built to be scalable and with an expandable table depending on the number of elements we decide to use.

# 5. Results

To see the results of our engine, there should be a prompt typed into the engine. This is the result after typing the prompt “I love sleeping and working out” into our engine.

*'5 Job Roles for People Who Love Sleeping and Working Out', '', '1. Personal trainer. Help people achieve their fitness goals while getting paid to work out yourself.', '2. Yoga instructor. Teach people how to relax and improve their flexibility while getting some exercise yourself.', "3. Sleep coach. Help people get a better night's sleep so they can be more productive during the day.", '4. Fitness model. Get paid to stay in shape and show off your muscles.', '5. Night shift worker. Have your days free to sleep and work out, while still earning a paycheck.'] ['\*\*5 Job Roles for People Who Love Sleeping and Working Out\*\*\n\n\* Personal trainer: help people get fit while working out yourself\n\* Sleep researcher: study how people sleep and how to improve sleep\n\* Athletic trainer: help athletes recover from injuries and stay in shape\n\* Fitness instructor: teach people how to exercise and stay healthy\n\* Night shift worker: get paid to sleep during the day and work at night', ' Natural language processing is a type of artificial intelligence that allows computers to understand and generate human language. It\'s used in things like Google Translate, chatbots, and spam filters. For example, if you ask Siri "What\'s the weather today?", she uses natural language processing to understand your question and give you an answer.', "\*\*Personal trainer:\*\* A personal trainer is someone who helps people get fit by working out with them and teaching them how to exercise.\n\n\*\*Example:\*\* If you want to get in shape but don't know where to start, a personal trainer can help you create a workout plan and motivate you to stick with it.``, 'A yoga instructor is like a personal trainer for your flexibility and relaxation. They teach you how to do yoga poses that will help you feel good and get in shape.\n\nExample: If you want to be able to touch your toes or do a headstand, a yoga instructor can teach you how.', 'A sleep coach helps people get more sleep so they can be more active during the day. For example, a sleep coach might help you create a bedtime routine and avoid caffeine and alcohol before bed.', 'A fitness model is someone who gets paid to stay in shape and show off their muscles. They work out a lot, eat healthy foods, and take care of their bodies. They might be featured in magazines or on social media. For example, Kylie Jenner is a fitness model.', ' Night shift workers work at night and sleep during the day. This means you can have your days free to do other things like sleep, work out, or spend time with your family.\n\nExample: A teacher who works the night shift can sleep in the morning and teach classes in the afternoon.'] ['Exercise Science\n Nutrition\n Personal Training\n Sports Medicine\n Sleep Science', 'Calculus\n Physics\n Chemistry\n Biology\n Programming', 'Anatomy and Physiology\n Exercise Physiology\n Nutrition\n Personal Training\n Business', 'Anatomy\n Physiology\n Yoga\n Yoga Philosophy\n Yoga Therapy', 'Sleep Science\n Sleep Disorders\n Sleep Medicine\n Sleep Psychology\n Sleep Hygiene', 'Nutrition\n Exercise science\n Personal training\n Bodybuilding\n Health and fitness management', 'Nursing\n Hospitality\n Security\n Warehouse\n Retail'] ['Stanford University|Business Administration (BA)(*[*https://gsb.stanford.edu/programs/undergraduate/ba-program)\n*](https://gsb.stanford.edu/programs/undergraduate/ba-program)/n) *University of California, Berkeley|Economics (BA)(*[*https://economics.berkeley.edu/undergraduate-program/)\n*](https://economics.berkeley.edu/undergraduate-program/)/n) *University of Michigan, Ann Arbor|Business Administration (BA)(*[*https://lsa.umich.edu/business/undergrad/*](https://lsa.umich.edu/business/undergrad/)*)', 'Stanford University | Computer Science\n Carnegie Mellon University | Computer Science\n University of California, Berkeley | Computer Science', 'Keiser University|Exercise Science\n University of San Diego|Health Sciences\n Life University|Exercise Science', 'College of Marin | Associate of Arts in Liberal Arts\n Soka University | Bachelor of Arts in Buddhist Studies\n University of California, Los Angeles | Bachelor of Arts in Kinesiology', 'Stanford University|Psychology\n University of California, Berkeley|Psychology\n Harvard University|Human Development', 'University of Southern California | Business Administration\n Arizona State University | Business Administration\n California State University, Long Beach | Business Administration', 'College: University of Pittsburgh\n Degree: Nursing\n College: University of California, San Francisco\n Degree: Nursing\n College: Yale University\n Degree: Nursing'] ['$63,952', '$75,000 - $100,000', '$56,000 - $74,000', '$60,000 - $75,000', '$65,000 - $75,000', '$50,000-85,000', '$15.00 - $20.00']*

To know how extensive our engines are, our team went to compare the output with GPT-4, another leading LLM model. The prompt that our team input was “I like sleeping and working out. Please give me some job titles, salary, and school recommendations. Limit to 5 jobs”. We changed the output to ensure that each LLM will try to give us the same information as our engine. This was the output GPT-4 gives:

1. *Sleep Technician or Polysomnographic Technician*
   * *Salary: $40,000 - $60,000 annually.*
   * *School Recommendations:*
     + *Many community colleges and technical schools offer programs specifically for polysomnography.*
     + *Accreditation by the Commission on Accreditation of Allied Health Education Programs (CAAHEP) or the Commission on Accreditation for Respiratory Care (CoARC) is recommended.*
2. *Personal Trainer*
   * *Salary: $30,000 - $70,000 annually (can be higher with private clients or specialized training).*
   * *School Recommendations:*
     + *A degree isn't always necessary, but programs in Kinesiology, Exercise Science, or Physical Education can be beneficial.*
     + *Certifications from organizations like the American Council on Exercise (ACE), National Academy of Sports Medicine (NASM), or International Sports Sciences Association (ISSA) can enhance employability.*
3. *Physical Therapist*
   * *Salary: $60,000 - $90,000 annually.*
   * *School Recommendations:*
     + *Pursue a Doctor of Physical Therapy (DPT) degree.*
     + *Top-rated programs can be found at institutions like the University of Southern California, University of Pittsburgh, and Washington University in St. Louis.*
4. *Sleep Researcher*
   * *Salary: $50,000 - $90,000+ annually (can increase with advanced degrees and specialized roles).*
   * *School Recommendations:*
     + *Study neuroscience, biology, psychology, or a related field.*
     + *Leading institutions for sleep research or neuroscience include Stanford University, Harvard University, and the University of Pennsylvania.*
5. *Fitness Consultant for Corporate Wellness Programs*
   * *Salary: $40,000 - $80,000 annually.*
   * *School Recommendations:*
     + *A degree in Health Promotion, Exercise Science, or a similar field is beneficial.*
     + *Consider certifications in corporate wellness from organizations like the Wellness Council of America (WELCOA).*

After a thorough comparison, it's evident that our engine outperforms GPT-4 in executing the tasks. The distinctions between them are both clear and significant, highlighting our engine's superior capabilities. These are the differences:

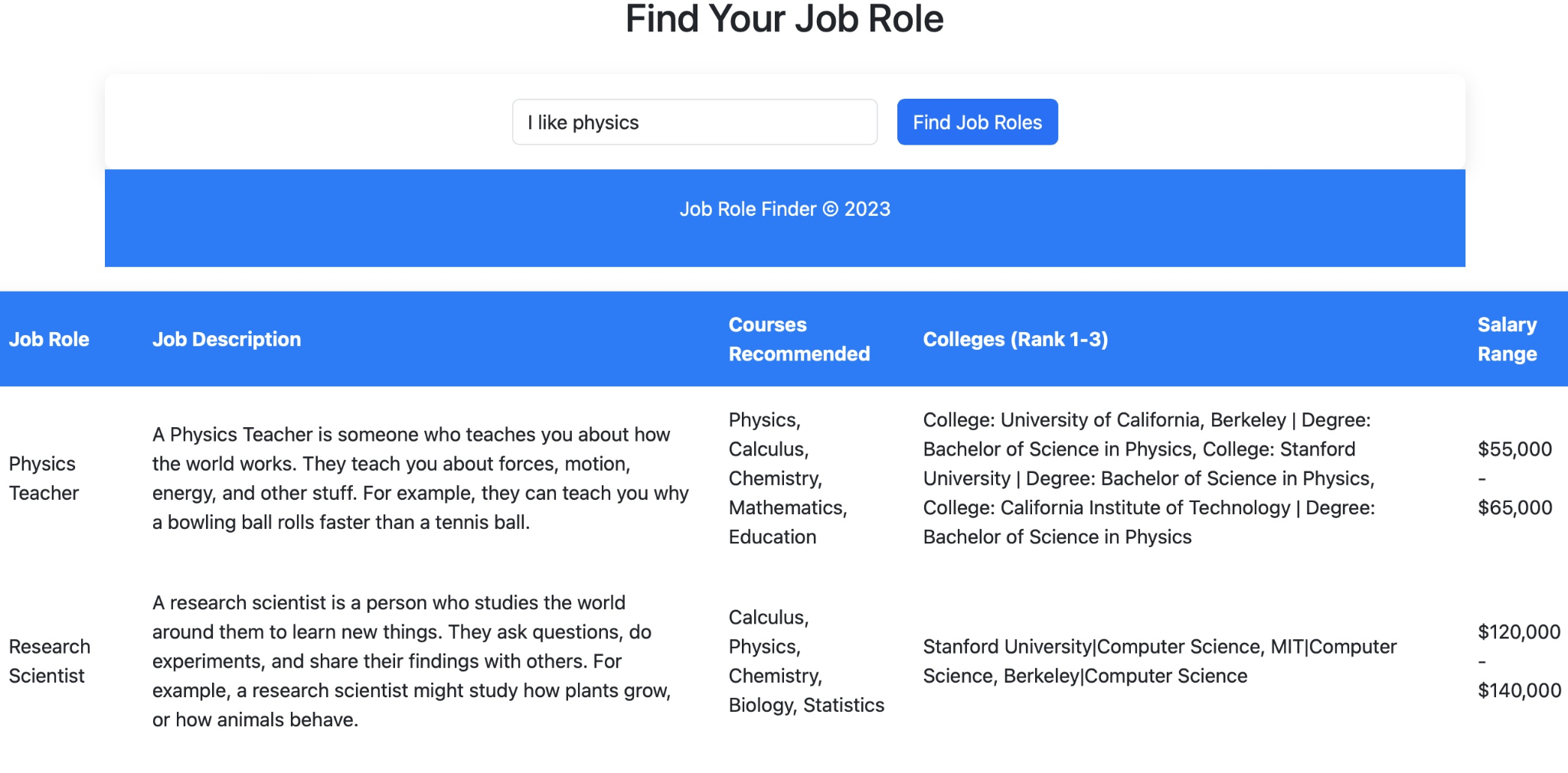
1. The provided job titles lack detailed explanations. Since our target audience is middle school students who might be unfamiliar with these roles, it's crucial to offer them insights into the day-to-day responsibilities and tasks associated with each job. Without such context, it becomes challenging for them to make informed career choices.
2. While our engine offers more comprehensive data than GPT-4, we acknowledge its current presentation can be improved, especially with all of the “\n” annotation that PALM Api added. We're actively working on enhancing the user interface to ensure the output is both informative and user-friendly, especially given the depth of information we provide.

Our team also tried to compare the output with Google Palm Api that was not trained nor coded with more codes. To keep the test similar, our team kept the input the same, “I like sleeping and working out. Please give me some job titles, salary, and school recommendations. Limit to 5 jobs”. This was the output that palm api shows.

*﻿ | Job Title | Salary | School | |---|---|---| | Personal Trainer | $40,000 - $60,000 | A bachelor's degree in exercise science or a related field is typically required. | | Sleep Coach | $40,000 - $60,000 | A bachelor's degree in psychology or a related field is typically required. | | Night Shift Worker | $25,000 - $35,000 | No formal education is required. | | Uber Driver | $25,000 - $40,000 | No formal education is required. | | Dog Walker | $15,000 - $25,000 | No formal education is required. |*

Comparing trained and untrained palm api, the contrast in their outputs is strikingly evident. The untrained model often suggests a disparate set of five job titles compared to its trained counterpart. More notably, these positions typically do not need any formal education. This could inadvertently send a misleading message to our target users. Many of our users turn to our engine with aspirations and curiosity about potential future careers. They seek guidance and a glimpse into what their academic and professional journeys might entail. Offering them roles without the context of educational pathways might inadvertently diminish their enthusiasm and motivation. It underscores the importance of a finely tuned and well-informed system, one that not only provides accurate job titles but also paints a holistic picture of the journey toward those professions.

While there are notable distinctions between the threeLLM models, our model offers unique insights. Our model, tailored to our target users, presents job titles along with easily comprehensible descriptions. It further enriches the user experience by detailing some of the daily job responsibilities and highlighting schools that can pave the way to success. To assist students even further, we sometimes provide direct links to program admission websites, ensuring a safe and accurate online navigation experience. Additionally, users receive an overview of the potential salary ranges for each profession. It's worth noting that our engine is in its early stages, currently drawing from the information we've trained it on and data available through the Google API.



# 

Within the constraints of our project timeline, our team successfully crafted a user interface (UI) that presents the engine's output in a more user-friendly manner. Instead of overwhelming users with a jumble of 15 schools, numerous job titles, and their respective descriptions all mashed together, we've streamlined the presentation. Now, each row is dedicated to a specific job title, complete with its relevant information. This design ensures clarity and coherence, allowing readers to easily navigate and extract the information they seek. We believe this revamped UI will significantly enhance the user experience, making data retrieval seamless and straightforward.

# 

# 6. Issues and learnings from the Project

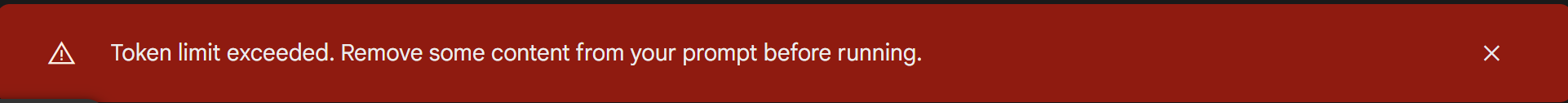
Embarking on our project with the Google Palm API was both a lesson in technological nuance and a journey through uncharted territories. Our initial excitement was grounded in the promise of what the API offered and the innovative solutions it could potentially deliver. Yet, as we navigated through its functionalities, we encountered a series of hurdles. Some stemmed from the inherent limitations of the API, while others were tied to the intricate nature of user feedback and expectations.

## 6.1 What we learned:

During doing the project, we learnt two important things with the palm api, especially on the output and the prompting.

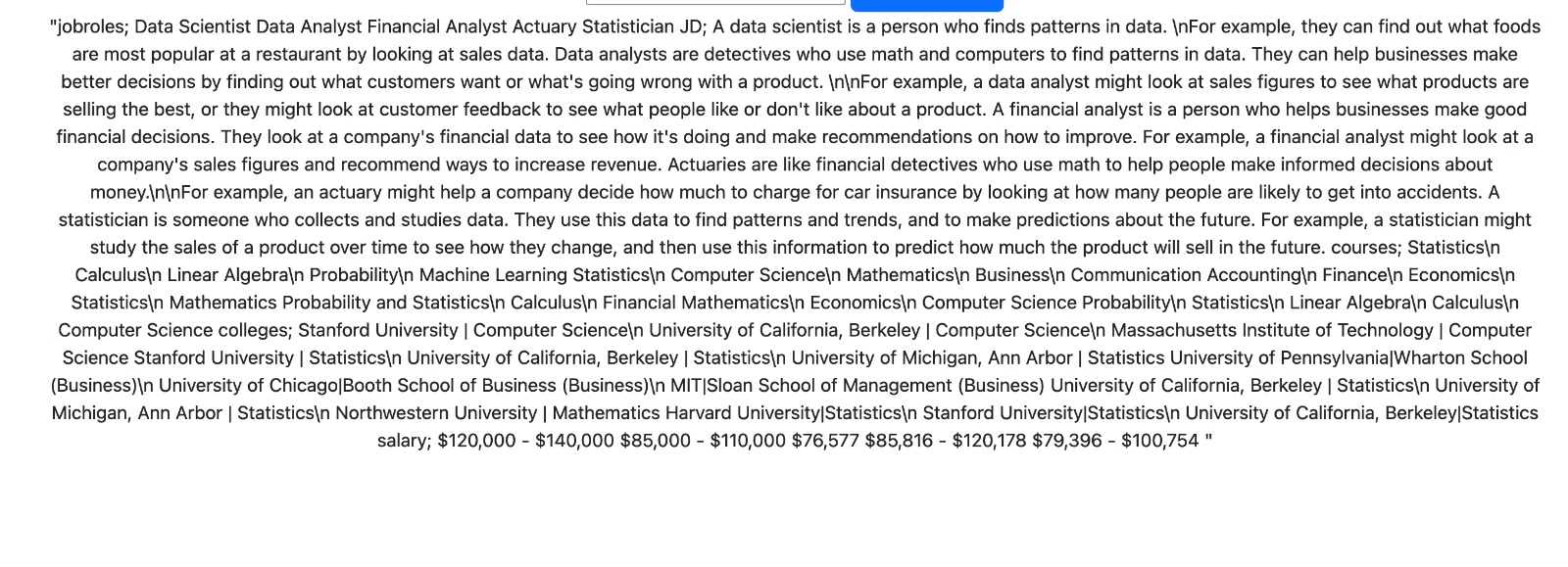
1. **Output** : The Palm API doesn't produce plain text outputs; it generates code. One specific issue we encountered was with the code's use of "\n". In coding terms, "\n" is essential; it introduces a new line, ensuring the code is organized and readable for developers. However, the challenge arose when we integrated this with Flask. Instead of interpreting "\n" as a line break, Flask viewed it as regular text, outputted by the engine. This misinterpretation meant that our desired formatting was compromised. To address this, our team delved into finding a solution, ensuring that our outputs would be displayed as intended.
2. **Prompt** : During our experimentation with the Palm API, our team made an intriguing observation: the system thrives on simplicity. Inputs that were concise and straightforward gave the most accurate and effective results. On the contrary, when we fed the system with intricate, wordy prompts, it seemed to interpret our request as a desire for more complex outputs. This added complexity, unfortunately, was counterproductive, especially when we considered our target audience: middle school students. Young minds often benefit most from clarity and directness, and our goal was to provide information that resonated with them without overwhelming them. Recognizing this, we adjusted our approach to prioritize simplicity, ensuring that the insights generated would be both accessible and impactful for our younger users.

## 6.2 Issues our team face (Negative results / findings)



Our team found this issue when training with our dataset without formatting them and when we try to prompt our palm api. We found that the limit of the free google palm api is 8k for input and 1k for output. At this point we found 2 different causes and solve them appropriately:

1. **CSV File Structure**: Our meticulously crafted CSV file, while not a lot with characters, boasts a multitude of entries. This is primarily because we aimed to comprehensively list key details. However, our experience with Google's Palm API revealed its preference against excessive newline functions. As a solution, we optimized the file by condensing our extensive list and distributing it across four distinct rows. Some rows remained null or empty, reflecting the occasional unavailability of certain data points even after looking on 6 different websites. Rather than filling these gaps with placeholders like '0' or 'N/A', we opted to retain the integrity of the dataset by leaving them as they were.
2. **Prompt Output Dynamics**: The Google Palm API comes with its constraints, one of which is its 1k token limit for outputs. This presented a challenge, especially given our expectation of a rich, detailed output covering areas like Job Titles, Descriptions, Course and School Recommendations, and Salary Ranges. To navigate this limitation, we innovatively split a single prompt into multiple ones. This approach allowed us to generate the requisite data in parts. We then seamlessly merged these segmented outputs into a comprehensive, singular display. In our ongoing pursuit of user-centric design, we're also spearheading efforts to develop an intuitive user interface. This will enhance readability and user experience, ensuring our audience can easily interpret and benefit from the results.



One of the primary challenges our team grappled with was creating a user-friendly interface to display the output. As illustrated in the earlier snippet, the current presentation is far from being easily digestible. Given that our engine amalgamates the results from five distinct functions, the resulting output can come across as a jumbled mix. Unintended separators like “\n” and “|” further muddle the clarity. Moreover, the intended format – where each job title would suggest five unique schools – seems lost in the output, which haphazardly lists 25 schools and their respective programs. Our immediate mission became clear: to craft an interface that would bring coherence to this abundance of information. Overcoming this hurdle was undoubtedly among the most formidable challenges we faced during our hackathon endeavor.

# 7. References

OpenAI. (2023). ChatGPT (Nov 04 version) [Large language model]. <https://chat.openai.com/chat>

GooglePalmApi2. (2023). PalmApi (Nov 04 version) [Large language model]. <https://developers.generativeai.google/>