# AN INTELLIGENT AND AI-DRIVEN MOBILE PLATFORM FOR HIGH SCHOOL STUDENTS IN THE COLLEGE APPLICATION PROCESS USING OPENAI AND GOOGLE MAPS API

Claire Yu[1], Joshua Wu[2]

[1]Germantown Friends School, Philadelphia, PA 19144
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

## *ABSTRACT*

*Students feel overwhelmed in the college process, whether it be due to the inability to find college tours or feeling lost on what they want to pursue in education and where to pursue it. Through our application, CollegeMatcher, we aim to resolve such feelings of aimlessness by providing accessible resources of information and conveying the campus experience for a multitude of universities in America, as well as guiding the student to a higher education home based on their interests and capabilities. We utilize web scraping and API connections (including Google Maps and OpenAI) to retrieve data for each university, including its buildings and images. The application page that provides personalized recommendations of universities integrates with an OpenAI trained chatbot (through its API) in addition to user-inputted data. This application will help students find the university and path of education that best suits them, allowing for the future discovery of passions and pursuits.*

## *KEYWORDS*

*Flutter, Dart, Python, OpenAI, ChatGPT, Google Maps, College Application, Virtual Tour, University, Campus*

## 1. INTRODUCTION

Struggling to navigate the college application process and finding the right university are oftentimes ominous issues that high school students face. It is found that "the stress experienced by high school students in preparing for college [and the college application process] is well known" to the point where "many students become discouraged or intimidated by the stress of [it all]" [1]. Meanwhile, studies show that "the intensity of stress is largely dependent on the availability of external resources for support". With the smooth adoption of technology in society, the convenience and efficiency of the mobile application have become prevalent in young people. In fact, "the number and diversity of apps are increasing at an unprecedented rate" [3]. Students have often found physical and mental support and comfort from online resources, and the ongoing issue of the detrimental effects of college-induced pressure and stress on mental health is no different.

In addition, high school students often feel that "choosing the 'right' college is pressure-filled," whether it be due to feeling without direction or having too little or too many options [4]. However, discovering and exploring the right university is important because it establishes a

David C. Wyld et al. (Eds): NWCOM, ASOFT, ITCCMA, CSITY, BDIoT, MLNLP, SIGPRO, AIFZ– 2024
pp. 15-30, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141902

foundation for excellence and personal growth. With the support of a guide or counselor, building confidence and connection between the student and their potential university will be much more simplified. The guidance counselor's time and devotion to each student is nonetheless limited, therefore taking away from necessary resources and tools that a student may need, requiring an alternative solution. It is also important to understand that a counselor's energy and skills are not always the highest of quality or what a particular student is fit for. Thus, we can observe that the mobile application has the potential to be something more accessible to the student as "[they] provide anyone anywhere instant access to information once limited to the powerful and wealthy" [5].

We combined the popular and accessible vehicle of the mobile application with the guidance and resources that students are in need of. Comprehensive overviews and images of numerous universities in America bring the experience of the university to each student. This convenient interaction fosters simplicity and community in the college application process by deconstructing the physical, mental, or financial barriers that campus exploration may entail. Similarly, our mobile application CollegeMatcher invites students to use their academic and personal interests to receive a customized list of twenty recommended universities, their overviews, and the option to explore deeper through an AI-powered recommendations page. This online AI guidance counselor nurtures the student's passions while enriching them with their best-suited pathway of higher education, enhancing the confidence and support that a student needs.

The content of our mobile application is extracted through web scraping and multiple APIs. Through web scraping in Python, we gathered the descriptions for each university and its building names. We also utilized the Google Maps API to retrieve the address, nearby places names, and photo urls for each image [9]. All of the data collected is put into a Firebase database to build a connection between the backend Python and the frontend framework Dart. To create the recommendations page, this recommendation uses an OpenAI trained chatbot alongside the user-inputted data to generate accurate matches for the user. In Flutter (the language within Dart), we integrated all of the data from the database to create a UI that presents the descriptions, buildings, and images of each building of the universities, a search page, and an interactive page that provides the user with a customized list of universities [10]. Using a connection through the OpenAI API, this page integrates the user-inputted data and an OpenAI chatbot to generate accurate and personalized universities, including reasons why the school may be a good fit for the user.

## 2. CHALLENGES

In the development of our application, we experienced some challenges that affected multiple components of our program. As we had to think critically about the structure, accuracy, and specifics of our application, resolving such problems became an important part of development.

### 2.1. The Backend

In the web scraping part of our application, a major part of the backend that involves extracting online information, we encountered a few challenges. To ensure the accuracy of information, the code for each university to extract its data, specifically the building names, was manually programmed and edited. Oftentimes, we would come across empty values for a school's data due to the program's inability to find information on the particular school. To resolve this, we found data for that university without the program's assistance and manually incorporated it to the application afterwards. In addition, we wanted to maintain the accuracy of information that the

program gathered and worked with. We confirmed the reliability and accuracy by sifting through the university's data to ensure the data was up to date and extracted correctly.

### 2.2. Extracting Images and Data

When extracting images and additional data for each university, we utilized the Google Maps API that was smoothly integrated with the backend language Python but required some troubleshooting in maneuvering different programs. Our original programs to extract data on universities and integrate the data into our database neglected the capability of adding additional universities and scraping new data. We resolved this issue by creating a separate program using the Google Maps API designated strictly for the purpose of updating our list of universities, capturing its data, and connecting it with the database. Although this new file was potentially difficult to integrate into the original programs, we observed the importance of its place and its significance, particularly to be able to continuously update our application's content.

### 2.3. Organizational Issues

We encountered a few organizational issues as well, specifically pertaining to our Firebase database. Because there was a large amount of data that was constantly pushed to the database, we originally designed a format and placement of the content within the database that stored the data in a way that was efficient for the programs to read and access. However, as we continued to develop, the complexity of the database also enlarged, leading to our redevelopment of the organization of the database, including reformatting folders and storing information in a more.

## 3. SOLUTION

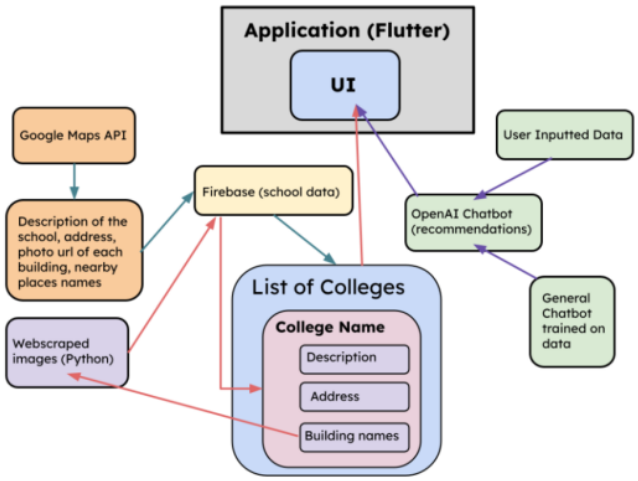


Figure 1. The CollegeMatcher flowchart

When the user opens the application, the UI of our program directs them to a list of universities. This list individually contains the description, address, and images of the school's buildings and nearby places. This information is extracted from our Firebase database that contains all of the data gathered through Python script. This database allows a seamless connection between Python and the frontend language Dart [11]. Within the Python backend code, we utilized the Google Maps API to extract the description of each school, its address, photo urls of each building, and the names of its nearby places. Through Python, we also collected images for the buildings through web scraping. After all of this backend information is sent to the database, it is integrated

with the frontend through the framework Flutter and the language Dart. In Flutter, we created a home page that leads to the list of universities and each individual school, a search page for the user to explore all universities or find a specific university, and a recommendations page. The recommendations page allows the user to input their academic information such as grades, extracurriculars, and awards as well as school preferences. It then returns them a personalized list of twenty universities and their overviews, with the option to click on a school and explore more in its detailed page of our app. This recommendation uses an OpenAI trained chatbot alongside the user-inputted data to generate accurate matches for the user.

The list of buildings and nearby places presents to the user their options in a clear and accessible format. We extracted information about the university through Python with the Google Maps API and organized it in Flutter. The buttons of the buildings are a key part of the application as it leads to one of the main focuses of the app, the images of each building for each university.

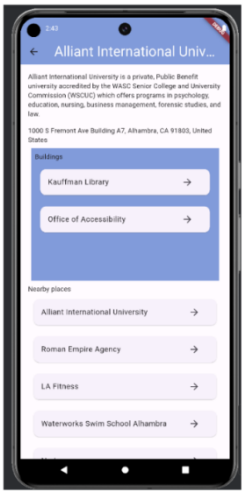


Figure 2. A UI page that presents the details of a university

```
def getSchoolInfo(university):
    #uses google apis to put all of the school information into our own dictionary called location

    url=("https://maps.googleapis.com/maps/api/place/findplacefromtext/json?key={api_key}&input={university}&inputtype=textquery&fields=name,formatted_address,geometry"

    result=requests.get(url)
    result=json.loads(result.text)

    print(result)

    result=result["candidates"]

    correctCandidate=""
    #for loop and if is checking if it exists and finding the right location
    for candidate in result:
        #print(candidate.keys())
        if candidate["name"]==university:
            correctCandidate=candidate

    if correctCandidate=="":
        #print(result)
        correctCandidate=result[0]
    location={}
    location["name"]=correctCandidate["name"] #school name
    location["formatted_address"]=correctCandidate["formatted_address"] #school address
    #print(correctCandidate.keys())
    location["coordinates"]=correctCandidate["geometry"]["location"]
    location["buildings"]=nearbyPlaces(location)
    return(location)
```

Figure 3. A snippet of the backend code that uses the Google Maps API

The function above, called getSchoolInfo, receives a passed-in university and returns a Python dictionary of the school's extracted information. We start by creating an integration with the Google Maps API using our API key and the specific university [12]. Then, all of the information gathered through the API is stored and filtered using the variables result and correctCandidate. Finally, we create a dictionary called location that stores the data that we have extracted: name, address, coordinates, and building data which is gathered through the next function nearbyPlaces. The images of each university in our application allows the user to experience the environment of universities across the US and explore campuses in a way that is accessible and comprehensive. The multiple images for each individual building of a university are organized such that the user

can get to know a particular building of a school while also gaining a general feel for the university. The images are extracted and organized using web scraping in Python, transferred and stored into our Firebase database, and extrapolated to the frontend in Flutter.


Figure 4. A UI page that shows the images of a university

```
def extractFullImages(scriptTag, maximages):
    matched_images_data = "".join(
        re.findall(r"AF_initDataCallback\({[^<]+}\);", str(scriptTag)))
    matched_google_image_data = re.findall(r'\"b-GRID_STATE0\"(.*)sideChannel:\s?{}', matched_images_data)

    removed_matched_google_images_thumbnails = re.sub(
        r'\[\"(https\:\/\/encrypted-tbn0\.gstatic\.com\/images\?.*?)\",\d+,\d+\]',
        "", str(matched_google_image_data))

    matched_google_full_resolution_images = re.findall(
        r"(?:'|,),\[\"(https:|http.*?)\",\d+,\d+\]",
        removed_matched_google_images_thumbnails)

    full_res_images = [
        bytes(bytes(img, "ascii").decode("unicode-escape"),
              "ascii").decode("unicode-escape")
        for img in matched_google_full_resolution_images
    ]
    return (full_res_images)

def saveImages(buildings,university):
    listUrl=[]
    for i in buildings:
        images=getOriginalImages(query=f"{i} building - {university}", maximages=4)
        print(images)
        for local_path in images:
            url=data.upload_file(local_path,firebase_path=f'{university}/building_pictures/{local_path[7:]}')
            listUrl.append(url)
    return listUrl
```

Figure 5. A snippet of the backend code that web scrapes images

The Python code above highlights a portion of the process of extracting and organizing images of the universities. The function extractFullImages occurs after the initial part of web scraping and requesting the images and utilizes regular expressions to sift through the web-scraped images to obtain the images of higher resolution and quality. Additionally, the function saveImages calls the previous function getOriginalImages to obtain the local path of each image in order to save and send that data to our Firebase database.

The recommendations page is a core part of our application as it provides additional and valuable information for students. This page requires the user to input their academic, athletic, and other passions and school preferences in order to give the user a personalized list of 20 recommended universities including the universities' overview, highlights, and the option to explore the school more within our application. The recommendation uses a connection with an OpenAI general trained chatbot, ChatGPT, to generate accurate and customized responses through the OpenAI API.

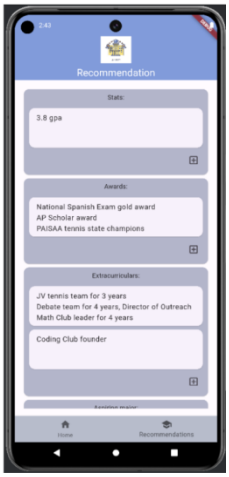


Figure 6. A UI page that shows the recommendations page

```
Future<dynamic> showDialogueError(){
  return showDialog(
      context: context,
      builder: (context)=>AlertDialog(
        title: Text("Error"),
        content:Text("An error occurred while we were generating your schools. Please try again."),
        actions: [
          TextButton(
              onPressed: (){
                Navigator.of(context).pop();
                Navigator.of(context).pop();
                },
              child: Text("OK"))  // TextButton
        ],
      ) // AlertDialog
  );
}
void chatGPTCall() async{
    String instructionPrompt =
        "Please give me a recommendation of 20 universities based on "
        "${widget.info} with some description of each university  ";
    final request=ChatCompleteText(
      messages: [Map.of({"role": "user", "content": instructionPrompt})],
      maxToken: 2500,
      model: Gpt3urbo0631Model()
    );  // ChatCompleteText
    ChatCTResponse? response=await _openAI.onChatCompletion(request:request);
    setState(() {
      result=response!.choices.first.message!.content.trim().replaceAll('"', '');
      isLoading=false;
    });
}
```

Figure 7. A snippet of frontend code that uses the OpenAI API

The recommendations page uses the OpenAI API key to access ChatGPT in a program that is written in the frontend framework of Flutter and language of Dart. The function showDialogueError considers the case when there is a technical issue with the chatbot and thus returns a general error message to the user on the page. In the function chatGPTCall, we crafted a prompt for the chatbot that was designed to provide the accurate responses and later integrated that with the user's inputted data to load recommended universities onto the page. In the process, the program filters and reviews the response to ensure that it is not empty before providing it to the user. Because the user's inputted data and the chatbot's returned list are not stored locally or stored server-side, the user can easily update their responses to match their current state.

## 4. EXPERIMENT

### 4.1. Experiment 1

In our first experiment, we tested the accuracy of the recommendations page by using the user-inputted data to match the user with universities through a general OpenAI ChatGPT chatbot. This involved creating 10 student profiles with randomly generated information. For each question/parameter within the page such as their GPA, awards, extracurriculars, aspiring major,

and personal preferences, we used a random number generator that corresponded to one or more choices in a list of common student pursuits in that category. Specifically, the pool of student data that was randomly selected from was compiled through our research of the most popular and existing majors and common choices for extracurriculars and awards. For the parameter asking about the student's preference on the range of schools, we concluded that some important factors that often impact decisions are location within the United States (such as the west coast and east coast), the level of prestige, and whether it is private or public. We created and compiled the data for each student profile with the goal that these conditions and requirements are universal and holistic for prospective university students, reflecting their demands and needs in such a way that this experiment can be applied to students and their situations in real life.

This randomly generated profile and data is then inputted into the recommendations page five times for each profile to ensure consistency of ChatGPT's returned universities. We then examine each of the five results for each of the 10 student profiles to verify if our application returns universities that match the student's statistics and fit the student's requirements. We recorded the accuracy of each trial per student as a fraction of the number of generated schools that matched their requirements and the total number of generated schools.

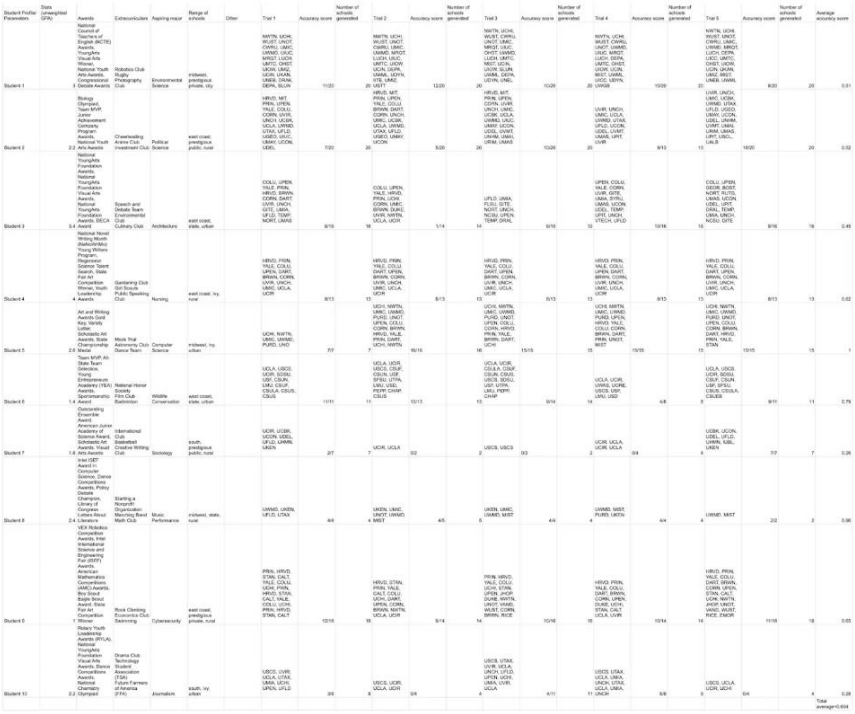

Figure 8. Figure of experiment 1

To help the user brainstorm more suggestions for parameters, it would be beneficial to provide examples of preferences to include for their recommendation in the "other" section of the page. Suggestions for the user to include in their input page are the political direction of their universities, the gender that their universities enroll, or if the university is a minority-serving institution.

Through experimenting on the chatbot in the recommendations page, we found that ChatGPT is not always consistent and has some inaccuracies with its responses. The recommended

universities provided a large scope and breadth of schools that catered to the student's data and overall were good matches for the student.

However, we found some inconsistencies across the trials testing ChatGPT's list of 20 recommended universities. In this experiment, we provided the chatbot with the same student data five times, repeated for 10 students. However, we found that the list the chatbot returned was slightly different each time. The differences in the recommendations included the universities' placement in the list, missing or additional universities, and its fluctuating ability to meet the student's preferences. Specifically, although we requested a list of 20 universities, ChatGPT was often not able to fulfill that requirement, fluctuating in the number of recommended universities within student profiles and trials. In addition, for some student profiles, ChatGPT's response became more fit to the student's preferences the more we ran the student's data.

With average accuracy scores per student profile ranging from a minimum of 0.26 and a maximum of 1.00, the total average accuracy scores of all student profiles is 60.4%, shown in Figure [x]. Though this percent is not ideal, the range of the inconsistencies demonstrate how variable the program's response is. Despite an overall average accuracy score of 60.4%, the recommended schools across student profiles give both a holistic and specific perspective of appropriate universities for students. This current state means that this recommendation service and ChatGPT program has room for improvement whether it be in the chatbot itself, working with another AI model, or creating a more comprehensive user experience
.
It is important that the user is recommended the most fitted list of universities; therefore, we must look into what is driving these inconsistencies and how we can improve the recommended results. One potential reason for these discrepancies is the wide scope of student preferences and diverse universities across America. The many priorities and preferences of students may affect ChatGPT's ability to filter through and guarantee each university fits such requirements.

Large language models such as ChatGPT should be used with caution as there are potential concerns about ChatGPT's accuracy and integrity [x]. In fact, in a recent study researching the authenticity of references in medical articles generated by ChatGPT, it was found that of the 115 generated references, 47% were fabricated, 46% were authentic but inaccurate, and only 7% were authentic and accurate [x]. The severity and significance of these findings have led to the characterization of this behavior as artificial hallucinations, in which chatbots generate "seemingly realistic information that does not correspond to real-world input." [x].

Furthermore, ChatGPT lacks experience with actual students and the college application process. A human college counselor may have an approach that can more consistently provide students with schools based on their specific preferences and capabilities. Despite the convenience and accessibility of this online college counselor, the consistency is an issue that we will continue to address and improve.

We observe that the number of universities generated for each trial differed from each other both slightly and drastically. For instance, Figure [X] shows that while each of the five trials for Student 1 generated 20 schools, for trial 3 of Student 7, only two schools were generated. This inconsistent situation raises the question of whether the AI is experiencing glitches or whether the AI is simply unable to find sufficient schools that exactly match the student's requirements. These two cases suggest that either the student did not receive the most comprehensive recommendations by the chatbot or that the student was given just as many schools that fit their needs. It is important that we understand and uncover whether this implies that ChatGPT is inaccurate or unreliable and whether ChatGPT's deficiencies are inherent or specific to this

scenario. With this in mind, we are led to question if all AI chatbots and models would respond to our requests in the same inconsistent way: would another AI model more strictly follow our request of 20 schools that match the parameters, instead of ChatGPT's more creative and independent response? This question is a crucial one to ask as further experimentation of other AI models could lead to more effective and reliable recommendations for our application.

## 4.2. Experiment 2

In our second experiment, we take into consideration the user's experience and satisfaction of the recommendations page by allowing them to use their information to receive a personalized list of 20 universities and their overviews. To ensure an unbiased user feedback that does not factor aesthetics into their satisfaction but only the recommended results, we chose the Google form as a method to conduct a large number of surveys in an efficient and accessible way. While keeping in mind that the age of a participant may affect their efforts in participating in this experiment, we reached out to students between the ages of 15 and 22.Through a Google form survey, we asked the user to fill out the same questions as the boxes on the recommendations page. After sending them the text result of what ChatGPT creates, we then conducted another round of surveys with the same group of participants to delve into their feedback and satisfaction with the recommended schools. In this second survey, we gave the user the opportunity to answer and provide thoughts on questions such as how sufficient the parameters were and their preference between a human college counselor and the recommendations page.

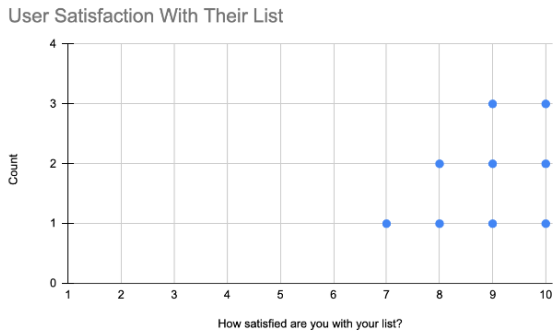


Figure 9. A scatter plot of the first experiment that illustrates experiment participants' responses to the question "How satisfied are you with your list?"

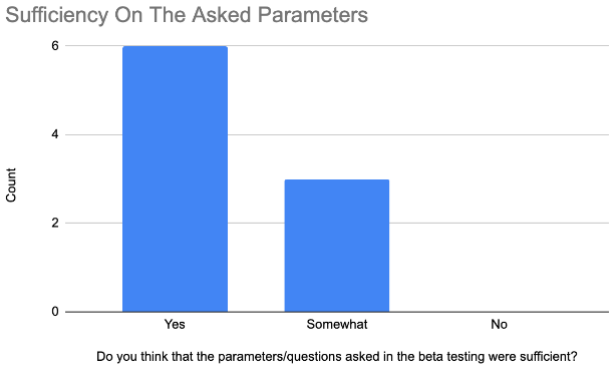


Figure 10. A bar chart of the first experiment that illustrates experiment participants' responses to the question "Do you think that the parameters/questions asked in the beta testing were sufficient

The human data collected through our surveys reflect concern for the specificity and uniqueness of our current approach to university recommendations. When asked about their satisfaction of their recommended list, participants generally expressed satisfaction, with a mean rating of 8.88 out of 10, with 10 being the most satisfied. However, some stressed the importance of the addition of more comprehensive parameters to ask the user.

Multiple responses were concerned about how well a school fit them, specifically regarding top programs at universities. While some emphasized how the inclusion of programs into our recommendations page would allow our application to serve a more "unique" role for a student, others expressed that research into specific programs is a "major part of choosing a university," and therefore a recommendations page without consideration into programs would be not as helpful. We recognize the significance and emphasis that students put on programs, and will implement a list of programs for each college to work with the recommendations page. One participant specified that that program's aspect can be incorporated "both in the survey and in the results," highlighting the different areas to include information on programs whether it be as a parameter for users to specify their preferences or as a part of the recommended list.

Specificity in the asked parameters and the breakdown of recommended universities are crucial in ensuring that the user receives the most individualized information. To this end, the cost of a university and financial aid play a significant part in a student's attendance of universities. Similarly, there was a large number of responses that requested cost to be a parameter. One response reports that "Perhaps financial aid… and [the] cost of the college would be helpful to add (both in the survey and the result)." Another participant phrased, "I recommend including a question about how much [students] are willing to spend on fees." We found that it is important to students that there is a designated parameter for cost, instead of being enveloped inside the "other" section.

After determining the student's needs and preferences, we turn to the organization and content of the recommended universities. When survey participants were asked about their preference between this virtual approach and a human college counselor for recommendations, there were mixed responses that supported either side. While one participant enjoyed "being able to see them all at once rather than having someone list them off one by one," another stated that "you may be able to have a more productive conversation of which schools are a good choice to apply for through a human college counselor." Personalization was a common theme that surfaced with this question, whether it was phrased as a "genuine" recommendation or as humans "know[ing] your traits and values more than ChatGPT." Specifically, many voiced the benefits of the physical and professional experience of a human college counselor. In particular, one student voiced that, "I like this virtual approach, but it takes away personal experience. A human college counselor might have more to share than a virtual one, such as sharing past students' experiences in those schools." Despite these inclinations towards the human college counselor, we also observed factors of convenience that supported participants' favoring of our recommendations page, with one valuing that this method "takes less time".

A handful of participants view this approach as a great "starting place" but lacking in the specifics and thus requiring further research on the student's part. One participant clarifies that "I think it's a great way to filter out schools based on programs and location as a starting place but I feel like after that it isn't giving very unique information about the schools that makes one more appealing than the other." With the aim of providing more in-depth and personalized knowledge, we hope to be able to implement a higher-complexity generative AI. With a higher-quality AI model, we can then focus on providing the user with the most reliable and efficient information. One participant suggests the use of a different AI bot instead of ChatGPT in order to prioritize the amount of training data that an AI receives. This intricate topic requires the time and effort of

separate research dedicated to investigating the efficiency and accuracy of different AI models for this task.

Within our application, to enhance the user's experience and to "make the responses [in the results] more descriptive," as one participant put it, we plan to designate a portion of the recommended list to boost the user's confidence and further strengthen their understanding of these personalized universities. One way to bolster the student's confidence, as numerous participants expressed, is the incorporation of the idea of safety, match, and reach universities into the recommended result. A participant voiced that, "It would be nice to be recommended a specific number of safety, match, and reach schools. Typically, it's good to have at minimum 2 safeties, 3 matches, and 2+ reaches, but it could be up to the user to decide how many safeties/reaches/matches they would like to get in their personalized college list." We observe that it is a priority for students to thoroughly understand their capabilities in the college application process, and the allocation of universities into the safety, match, and reach categories is a direct way to address this need. Additionally, students have requested the addition of acceptance rates into our application to gain a better understanding of their chances in admission. A student stated, "I think it would be good to display the in-state / out of state acceptance rates for these schools, and if possible even the program/major-specific acceptance rates for these schools," combining the idea of acceptance rates and university programs. These overlapping categories reflect how interdependent these features are, and thus, we found that the incorporation of multiple new parameters and information rely on and build off of each other.

The implementation of acceptance rates both as a parameter for the user to request and as additional information given would prevent potential concerns that students may have about the program favoring Ivy League schools as the first few recommendations. We see this concern in the response of this participant who stated that although "this gave a great starting place, … it feels very slanted towards [Ivy League schools] and other… number one ranked colleges." Although it is generally recognized as important to have an ambitious mindset, ChatGPT's overall inclination towards Ivy League schools may suggest the biased nature of the chatbot. Generally, ChatGPT has demonstrated its "systemic" and "implicit or unconscious" bias relating to race, gender, religious affiliation and more [x]. For instance, Singh revealed that when ChatGPT was asked to verify if an individual would make a good scientist, the chatbot responded affirmatively "if the person is a 'White male' and negatively for every other subgroups" [x]. These recurring biases of ChatGPT caution us to utilize the chatbot and its responses with care and with awareness. As developers of our application, CollegeMatcher, and users of this AI tool, we ensure that the purpose of our application's use of ChatGPT avoids biases through the careful crafting of our requests to the chatbot and our care to guide ChatGPT away from situations that may spark bias.

Users may also display worry and uneasiness with their data security, as the integration with AI models like ChatGPT often spark anxiety and controversy over information security and integrity. For instance, a participant expressed concern about the data that was inputted into ChatGPT: "I have some concern about the data you're collecting [through] this form because as you've said, it's integrated with ChatGPT. Your questions don't ask for directly identifiable [information], but it asks for [some] information that I'm worried about ChatGPT having access to [some] of it… [Though], those questions are perfectly relevant to the app's purpose,... I just wanted to suggest adding something to ensure nothing too personal is getting leaked to the AI so you can ensure the safety of the user's data."

Other parameters that participants suggested included a separate section dedicated to the student's leadership and school size, preferences on course rigor, student minors, and requests for special needs and accommodations. One participant vocalized that we "can include a question to

specify if the user has any special needs/accommodation[s]… [such as] for disabled students. A question about accommodation[s] (dorms, etc) could be included as well." Furthermore, we received other fascinating suggestions such as school culture, general weather, and campus and neighborhood safety that can only accurately be implemented with more advancements in artificial intelligence and information technology. As an example, suggesting more "descriptive" responses, a student expresses interest in having a section about "the school culture" but understands that a subjective and variable task like that would likely require "a whole knowledge base of alumni opinions/experiences." Similarly, to ensure the accuracy and consistency of generated information, the day-to-day or general weather of an area would benefit from weather forecast specific technologies or niche applications with user-generated content. Such developments in AI and information technology could also produce voices on campus safety that can be implemented to CollegeMatcher. We hope that the personalization of our application in future updates will create an even more accessible and user-friendly user interface, serving a larger and more diverse audience of students.

## 5. RELATED WORK

In Stephen Hansell's journal article, Student, Parent, and School Effects on the Stress of College Application, Hansell examines the multiple causes of the stress high school students suffer from [6]. We see that the stress of college applications may stem from the social environment of a student (i.e. peer pressure and the priority of higher education in a school), parent values and education level, or family affluence and economic status. Despite the fact that this stress may come from different roots, it is clear that the college application process has a negative impact on student mental health. Our application therefore aims to alleviate this pressure and emphasize support and accessibility while breaking down barriers that may cause such stress. Particularly, the reach and influence of the mobile application and the development of technology can eliminate certain concerns that may stem from social status by promoting the accessibility of student resources and college experiences.

Such resources are the focus of Michael Carnes' study, A phenomenological investigation of high school graduates' experiences in a career pathways model, which delves into a similar field and issue of the college application and its impacts [7]. Carnes focuses on the experiences and stories of high school students and graduates and how participation in a career pathways program, an educational program that illustrates the necessary skills and credentials for career fields, impacted their future in regards to higher education and self-efficacy. He suggests that student engagement in career pathways programs along with the support of a school-provided college counselor improve career readiness and help students navigate the college application process. This solution garnered positive thoughts from students as they report back on the immense support that the programs and counselors have brought them. However, it should also be observed that career pathways programs are not a universal solution for all students due to its limited accessibility, further restricting its influence on students' careers. On the other hand, the rising influence of the mobile application could play a pivotal role in supporting students. Our application uses the efficiency of the mobile application to combine the aid of college counselors with the continuous and necessary online support for students during the college application process, while delivering the resources and experiences of higher education to the student in a way that is familiar and accessible.

The study conducted in Chapter Five of Ensuring That Students Attend and Succeed in College: Analysis of Trends in College Enrollment, Promise Eligibility, and Students′ Perspectives on College Readiness from Transforming an Urban School System: Progress of New Haven School Change and New Haven Promise Education Reforms (2010–2013) discusses the impacts that New Haven's educational program, Promise, has on high school students, which may have

functions and benefits that are similar to the ones our application provides [8]. The Promise program partners with students and families to provide guidance in the high school journey, college application process, and overall career readiness. Using quantitative analysis, the study conveys the ways in which the reforms have increased the college enrollment rate among scholars. While Promise was successful, it is also revealed that many of these scholars still felt lost, particularly during the college application process. Some families of Promise scholars suggested the possibility for Promise to include support in college visits, even if it was simply financial support in gas, food, or lodging. Furthermore, despite the benefits that Promise has provided to students, we must first note its limited accessibility to New Haven. In addition, the fact that the study was published in 2014 indicates the possibility that, with the modernization and development of technology in the current age, that Promise can be improved upon. Utilizing the growth in both mobile applications and AI, our application creates another layer to mentorship and support for students by making the powerful tool of artificial intelligence accessible and catered specifically to generate personalized information in a swift manner. With CollegeMatcher, our application not only addresses the guidance aspect of the college application process as Promise does but also eliminates certain concerns that come with physical college visits. We instead present the campus and holistic experience of universities to the student through the efficiency of the mobile application in a way that boosts students' college experiences without the potential worries that physical visits may bring.

## 6. CONCLUSIONS

Even though resources that students can access may be limited, we have found that there is a way to openly provide the necessary resources to high school students in the college application process through our app. Even so, we must address the limitations of it. First, in order to ensure accuracy of information to the user, the content of our application must be regularly checked and updated. In addition, parts of the extraction of data through web scraping were manual and tedious, which in the future will require a more efficient and seamless method. With the extraction of data from multiple APIs, we must regather and rerun programs involving them in order to provide the user with relevant information. In regards to the accuracy of information, it is important to note that data from the APIs and from web scraping has the possibility of being inaccurate or misleading. Therefore, as developers, we must be cautious about the data provided in our application and be sure to quality check and filter through our programs. Additionally, there are inevitable bugs and technical errors that may occur within the APIs that may cause confusion among users. The database, the connection between the backend and the frontend, required experimentation and reorganization that cost time. There were also some issues that had to be troubleshooted due to the inherent drawbacks of the database platform. Therefore, transitioning to a more efficient and reliable database would be a priority as we gain a larger user population.

Initially, we had aimed to create and incorporate a 360 degree virtual campus tour into our application to provide users with an opportunity to explore their dream college to their hearts' content. However, as we developed our application, we encountered unexpected challenges with this element concerning restrictions on APIs, specifically Google Earth, and thus decided to rely on a multitude of images to capture and convey the experience for users. Nevertheless, there is great potential and impact for this 360 degree feature for future versions of our app, and we are working to develop and improve such a simulation of a campus tour. The possibilities include creating a 3D model of the campus, turning it into an interactive game for users, or manually photographing 360 degree images of university campuses in person. The many avenues to explore this feature are important to research due to the beneficial influence and support that this tool can provide students. The combined convenience of both the 360 degree virtual campus tour and the mobile application makes this feature and its platform valuable for students, families, and

educators. Additionally, for a high school student, understanding the settings of their future and education is crucial in order to better evaluate if an environment and its values align with their needs, preferences, and beliefs. Numerous obstacles, such as personal circumstances or financial status, hinder a student's access to experience a university completely; hence, this accessible and trustworthy resource is part of a solution that gives students a way to explore where they will find themselves and who they may become.

Although we extracted the essential information and highlights about universities through web scraping and delivered it to our users, there is data that was fetched but not used in the application. For instance, we gathered data on the exact geographic coordinates of each university and stored it in our database. However, because this piece of information seemed to have little value for users, we decided to leave it out of our frontend application. This type of discrepancy between collected data and presented data could lead to a potential waste of valuable data. We also hope to gather more information about universities through a more in-depth and efficient web-scraping method, researching and identifying what type of information students are looking for recently and finding necessary for their college career.

A key part of any product is the performance of its aesthetics, as it allows for easy navigation, user satisfaction and more. We plan to enhance the user interface of our application by creating a more cohesive color scheme, researching the most intuitive approach to navigation, and implementing accessibility options for users. The user's experience directly leads to their satisfaction with our application, which is a core part of a trustworthy and resourceful product. Because the experiments in this paper address our user satisfaction in a broad and shallow way, we hope to consistently analyze the thoughts and feedback of our users to ensure a pleasurable experience. Participants of our second experiment were mostly eager about this new service but expressed interest in a more niche and specific role of recommendations. Raising points like providing information on unique programs at universities and the cost/financial assistance of a school, real students felt as if they could benefit from a tool like the recommendations page but could aid from more advanced and in-depth information. Some experiments that we hope to conduct in the future include reaching out to the same group of students to reevaluate their opinions of a new version of our application and conducting our first experiment under the same conditions with another chatbot/AI model instead of ChatGPT. Chatbots that we believe could pair well with our application are Google Gemini, Microsoft Copilot, and Meta AI [14]. This could test not only the integrity of the chatbot but also could provide a statement on chatbot complexity across different models.

There are many paths that we can take in refining the connection between AI chatbots and our recommendations page [15]. To explore more accurate and detailed responses, region-specific chatbots would benefit the diverse location and range of schools across America. For instance, separate chatbots designed for and focused on locations in the US would allow for students to gain a thorough perspective on the areas of universities. University databases could also be a valuable tool in providing users with reliable and unique information. Currently, while our application is US-specific in its universities, we would love for an international expansion of our application. This would make ChatGPT or other chatbot's multilingual services necessary [13].
In regards to the experiments conducted in this paper, generalizability is an issue that needs to be considered whether it be in the user satisfaction results or ratings of a specific page. The group involved in the experiments have the possibility of being affected by personal issues and opinions or by a limited scope. Nonetheless, further experimentation and development can help resolve some of these limitations.

Furthermore, the future for CollegeMatcher, improvements, and applications of the app are vast. With the addition of user accounts within our application, we can implement more complex and personal features such as storing the user's history, being flexible with multiple devices, and connecting with friends. Additional components that give the application more value include the ability to star and share universities and store the user's data for efficiency. Improvements can also be made to the recommendations page specifically to the accuracy and reliability of the chatbot that generates personalized universities. It would be powerful to develop and train our own chatbot that takes in our prompt and more efficiently generates reliable and customized responses. As developers with the interest of delivering a trustworthy and reliable application to our users, we are currently working on these enhancements and plan to regularly update our application to its most refined version.

The applications and influence of CollegeMatcher are potentially extensive especially when thinking about its relevance and impact on young people today. In a society that increasingly prioritizes and favors young adults with higher education yet overlooks its effects on mental health and the disadvantages of underprivileged students, young people and future generations must create a dependable solution. With the goal of improving student mental health and disintegrating the barriers that students may face, our part of this solution starts with students in high school and uses the familiar and efficient platform of the mobile application to provide necessary resources for students during the college application.

This application was created due to my many experiences as a student, struggling finding opportunities to experience school campuses and being unable to attend in-person tours and school events. I hope and will continue to work on how CollegeMatcher efficiently reaches students who face similar issues and other barriers alike, building a community where we can rely on each other and construct a more equitable path of education.

## REFERENCES


[1] Hansell, Stephen. "Student, parent, and school effects on the stress of college application." Journal of Health and Social Behavior (1982): 38-51.

[2] Ireh, Maduakolam. "Stress: Helping students understand and manage it." The High School Journal 77.3 (1994): 233-239.

[3] Han, S. P., S. Park, and W. Oh. "Mobile app analytics: A multiple discrete-continuous choice framework—Management Information Systems Quarterly (MISQ), 2015." URL: https://papers. ssrn. com/sol3/papers. cfm.

[4] McDonough, Patricia M. "Buying and selling higher education: The social construction of the college applicant." The Journal of Higher Education 65.4 (1994): 427-446.

[5] Etzo, Sebastiana, and Guy Collender. "The mobile phone 'revolution'in Africa: rhetoric or reality?." African affairs 109.437 (2010): 659-668.

[6] Goldman, Cristin S., and Eugene H. Wong. "Stress and the college student." Education 117.4 (1997): 604-611.

[7] Carnes, Michael. "A phenomenological investigation of high school graduates' experiences in a career pathways model." (2021).

[8] Gladieux, Lawrence E., and Watson Scott Swail. "Beyond access: Improving the odds of college success." The Phi Delta Kappan 81.9 (2000): 688-692.

[9] Hu, Shunfu, and Ting Dai. "Online map application development using Google Maps API, SQL database, and ASP .NET." International Journal of Information and Communication Technology Research 3.3 (2013).

[10] Walrath, Kathy, and Seth Ladd. Dart: Up and Running: A New, Tool-Friendly Language for Structured Web Apps. " O'Reilly Media, Inc.", 2012.

[11] Weisser, Martin. "DART–The dialogue annotation and research tool." Corpus Linguistics and Linguistic Theory 12.2 (2016): 355-388.

[12] Lu, Hongqian Karen. "Keeping your API keys in a safe." 2014 IEEE 7th International Conference on Cloud Computing. IEEE, 2014.
[13] Biswas, Som S. "Potential use of chat gpt in global warming." Annals of biomedical engineering 51.6 (2023): 1126-1127.
[14] Khan, Adeel. "Microsoft Copilot Studio." Introducing Microsoft Copilot for Managers: Enhance Your Team's Productivity and Creativity with Generative AI-Powered Assistant. Berkeley, CA: Apress, 2024. 621-694.
[15] Gupta, Aishwarya, Divya Hathwar, and Anupama Vijayakumar. "Introduction to AI chatbots." International Journal of Engineering Research and Technology 9.7 (2020): 255-258.