QUESTGENIUS: AN INTELLIGENT QUESTIONNAIRE GENERATION AND RESULT ANALYZING SYSTEM FOR INVESTIGATION AND RESEARCH USING ARTIFICIAL INTELLIGENCE AND NATURE LANGUAGE

Tianchen Du[1], Ang Li[2]

[1]Crean Lutheran High School, 12500 Sand Canyon Ave, Irvine, CA 92618
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Addressing the complexities of survey creation, response collection, and accurate data analysis, this project aims to enhance the accessibility and quality of survey-based insights [1]. The background centers on the challenges faced in generating effective surveys, avoiding biases, and employing comprehensive analytics [2]. The proposal involves a web-based platform equipped with AI-driven survey generation, unbiased question design, and professional statistical analysis.

Key program components encompass user-friendly interfaces built with HTML, CSS, and JavaScript, seamlessly integrated with AI models for prompt generation and response synthesis. Challenges were met by refining AI prompts for robustness and implementing stringent user input validation.

Experimental scenarios included evaluating AI response resilience to user-crafted prompts, leading to insights on potential vulnerabilities and AI's adaptability [3]. Results demonstrated that well-crafted user prompts influenced AI outputs without compromising core functionality.

Ultimately, this innovative solution offers beginners and users unfamiliar with statistics a powerful survey tool empowered by AI [4]. The ability to create comprehensive surveys, mitigate biases, and access professional insights positions this platform as a valuable resource for informed decision-making across diverse fields.

KEYWORDS

AI Generate survey, Sample selection, Analyze results, Online Survey link

1. INTRODUCTION

During my final project for AP Statistics, I became acutely aware of the complexities involved in crafting effective surveys, selecting appropriate samples, and accurately analyzing survey results. These tasks proved to be intricate and demanding, prompting me to conceive an innovative

David C. Wyld et al. (Eds): SEAS, NLPI, SP, BDBS, CMCA, DMML, CSITEC -2023
pp. 153-162, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131714

solution – a user-friendly survey tool tailored for individuals unfamiliar with statistical methodologies.

As I delved further into statistics, I recognized a prevalent issue: many human-designed survey questions harbored unintentional or deliberate biases, compromising the integrity of the data collected. The integration of artificial intelligence into this process held the potential for remarkable enhancements in survey quality. Moreover, the critical task of sample selection often introduced biases that could significantly distort the final outcomes. Access to professional statistical insights at this juncture would undoubtedly prove invaluable.

Three methodologies in the survey domain were explored: SurveyMonkey, Survey Junkie/Branded Surveys, and Typeform [5].

SurveyMonkey primarily concentrates on collecting responses and offers diverse survey templates and functions. It provides basic result analysis through graphs but has limitations in advanced analytics. Survey Junkie and Branded Surveys pay respondents, distinct from the user-centric approach. Typeform specializes in creating surveys with multimedia elements and integration capabilities with other tools.

SurveyMonkey's shortcomings lie in limited analytics depth, while SurveyJunkie/Branded Surveys solely focus on incentivized responses, missing comprehensive survey creation. Typeform lacks AI-generated prompts and professional statistical guidance.

In response, the presented project combines user-friendly survey creation with AI-driven improvements in question generation, unbiased survey design, and offers insights through statistical analysis. It bridges gaps by catering to both beginners and those seeking sophisticated analyses. Unlike Survey Junkie/Branded Surveys, it offers diverse functionalities beyond mere response collection, and in contrast to Typeform, it integrates AI-based responses and offers advanced statistical insights [6].

The answer to these challenges materialized in the form of a comprehensive website. This online platform would encompass a suite of functionalities required to create, distribute, and analyze surveys effectively. In its initial iteration, my focus was on catering to novices in statistics, as well as those who lacked prior exposure to the subject.

A distinguishing aspect of my website emerged during its development – a departure from the conventional offerings available online. This uniqueness positions my platform to meet the precise needs of specific user groups, setting it apart from existing alternatives. Notably, the innovation doesn't stop at survey creation and distribution; my design empowers users to pose custom questions and subsequently synthesize their responses using the collected data, a feature unparalleled in other survey platforms.

In essence, my project encapsulates a response to the challenges I encountered during my statistical journey, presenting a solution that combines user-friendly survey creation with AI-driven refinement and expert statistical guidance [7]. Through this endeavor, I aspire to democratize the survey process, enabling individuals of diverse backgrounds to harness its power for more accurate insights and informed decision-making.

In both experiments, the focus was on assessing the impact of user-inputted instruction prompts for the AI. The objective was to determine whether unconventional or misleading questions, as well as additional instructions, could potentially disrupt the website's functionality. The primary goal was to uncover any vulnerabilities or bugs arising from user-generated prompts.

Results from the experiments demonstrated that well-crafted questions containing supplementary instructions did exert an influence on the AI-generated responses. However, the foundational structure of the AI's output remained relatively unaffected, indicating a resilient response framework resistant to major disruptions. Thus, while the introduction of new instructions could produce alterations in AI outputs, the integrity of the program's core functionalities remained intact, preventing critical errors [8].

Furthermore, a noteworthy observation emerged regarding the distinction between instruction prompts within the question-asking phase and those utilized during survey design. Notably, the prompt for asking questions exhibited enhanced robustness, primarily due to its more stringent instructional framework. This finding underscored the effectiveness of a well-defined and controlled prompt design, contributing to a more reliable AI response mechanism within the context of the experiment.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Analyzing

When I tried to analyze the first survey I created using the website, the analysis did not fit into my expectation, because in my mind, there are different kinds of questions and they cannot be analyzed in the same way, such as the corresponding questions, they cannot be analyzed independently, but that would be too hard to ask AI to determine which questions are corresponding to each other.

So, my final solution to this question is to give up analyzing every question on the survey and to make the user ask the questions that they want to be analyzed according to the database of the survey.

2.2. The Database

Out of my expectation, the database of the survey is also hard to build. Not only because of the combination of different kinds of questions and answers, but also because of the relationships between the answers from the same people in different questions. The answers to different questions from the same responder should be collected and put into a set for proper analysis of the correlation between events. And the answers from different types of questions can only be collected in specific, different ways. As the solution, I created a database and survey structure that includes the type of each question, such as multiple choice or short answer to make the whole database more organized.

2.3. Make the Content Easy to Understand

At first, I limited the number of questions in a survey to two because I just wanted to make surveys about the correlations between events at that time. But then I found that some correlation would be much more complex to be determined by only two questions or in some conditions the responder to the survey may not explain their thoughts or situations in one or two questions. But when there are more questions in a survey, the relationships between the questions may be complex. I have once thought about dividing the website into two separate functions for correlation questions and normal questions, but that would also be confusing for the users. So I finally choose the solution described in Challenge A.

3. SOLUTION

The program operates through three distinct and interconnected components, forming the backbone of the survey website: Survey Creation, Data Collection, and Analysis with Suggestions [9]. The seamless flow of the program starts with users arriving at the homepage, where they encounter a "Generate a Survey" button alongside a roster of existing surveys.

Initiating the survey creation process, users click the button and are directed to a survey generation page. This interface features two text input fields – one dedicated to specifying the survey's subject matter, and the other for determining the number of questions. Upon submitting these inputs, the website triggers the generation of a tailored prompt, submitted to an AI module for responses. The prompt confines AI-generated answers within a JSON-format survey list, which the website archives and displays on the user's homepage [10].

Following this, users can access their newly created survey by clicking on the provided link. The survey questions are presented on a dedicated page, accompanied by two pivotal buttons. The "Show Results" button aggregates and displays the collected survey responses, affording users a comprehensive overview of participant input. Simultaneously, the "Survey Link" button generates a shareable link for distributing the survey to potential respondents.

For comprehensive analysis, the "Show Results" page incorporates an additional element – an input field where users can submit a specific question for evaluation. The submitted question undergoes analysis using the amassed survey data, yielding insights and suggestions based on the compiled responses.

The entirety of this dynamic program is built upon a foundation of web-based technologies and integrated tools. The user interface is designed with HTML and CSS, while JavaScript facilitates interactive functionalities [15]. The AI-powered survey generation and data analysis utilize machine learning libraries and APIs, ensuring accuracy and sophistication in response generation and result interpretation.

In summary, the survey website seamlessly links three pivotal stages – survey creation, data collection, and analysis with suggestions. This journey begins with user-initiated survey generation, evolves through comprehensive data collection and visualization, and culminates in insightful analysis enriched by AI-generated suggestions. The fusion of user-friendly interfaces, advanced AI capabilities, and web development technologies crafts an innovative platform empowering users to effortlessly craft, conduct, and glean insights from surveys.

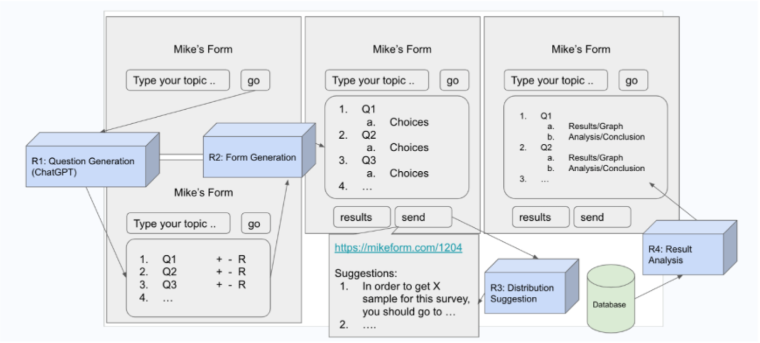


Figure 1. Overview of the solution

The main purpose of the first function, the survey generating, which is the main function of the website, is helping people prevent some bias and difficulties that may appear when creating a survey by themselves.

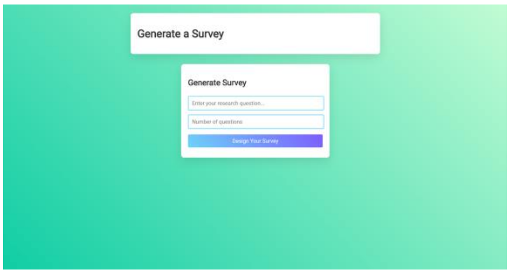
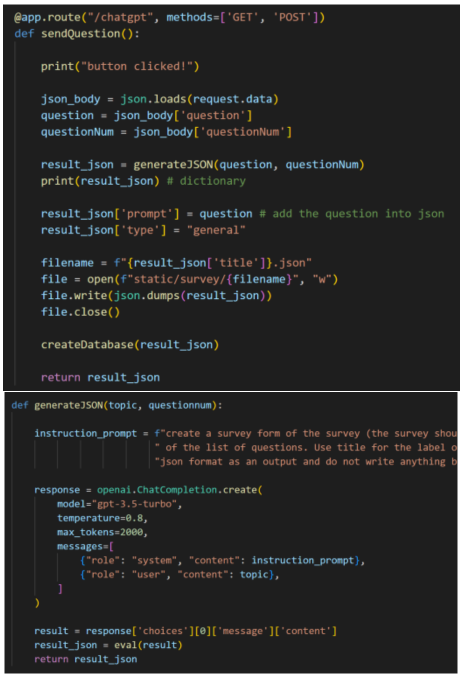


Figure 2. Generate Survey



```
@app.route("/chatgpt", methods=['GET', 'POST'])
def sendQuestion():

    print("button clicked!")

    json_body = json.loads(request.data)
    question = json_body['question']
    questionNum = json_body['questionNum']

    result_json = generateJSON(question, questionNum)
    print(result_json) # dictionary

    result_json['prompt'] = question # add the question into json
    result_json['type'] = "general"

    filename = f"{result_json['title']}.json"
    file = open(f"static/survey/{filename}", "w")
    file.write(json.dumps(result_json))
    file.close()

    createDatabase(result_json)

    return result_json
```

```
def generateJSON(topic, questionnum):

    instruction_prompt = f"create a survey form of the survey (the survey shou
                          " of the list of questions. Use title for the label o
                          "json format as an output and do not write anything b

    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        temperature=0.8,
        max_tokens=2000,
        messages=[
            {"role": "system", "content": instruction_prompt},
            {"role": "user", "content": topic},
        ]
    )

    result = response['choices'][0]['message']['content']
    result_json = eval(result)
    return result_json
```

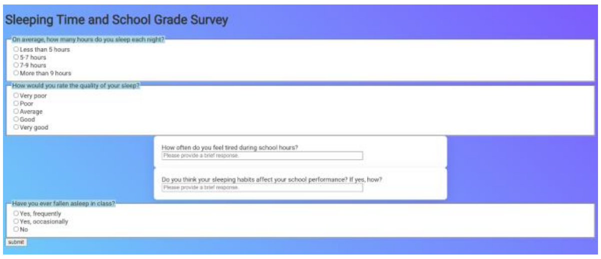
Figure 3. Screenshot of code 1

On the UI Screenshot, there is the Generate Survey page of the website, which is one of the core functions of it. And it will show up after the user clicks the Generate a Survey button on the homepage of the website. The user can use this page to create a survey with the topic and a number of questions they want.

In the first code sample, there is a route with a function that can send the request data from user to AI through the function generateJSON. After getting the survey generated from AI, it will open a new JSON file and write the survey in to store that [14].

And the second code sample is the function generateJSON that is used in the first code sample, which includes a prompt template and the template of the request that will be sent to the AI, after receiving the response from the AI, it will return the result.

Collect data: this section creates the survey link that the user can send to others for completing the survey. And they can answer the questions and submit their responses.



Sleeping Time and School Grade Survey

On average, how many hours do you sleep each night?

○ Less than 5 hours
○ 5-7 hours
○ 7-9 hours
○ More than 9 hours

How would you rate the quality of your sleep?

○ Very poor
○ Poor
○ Average
○ Good
○ Very good

How often do you feel tired during school hours?

Please provide a brief response.

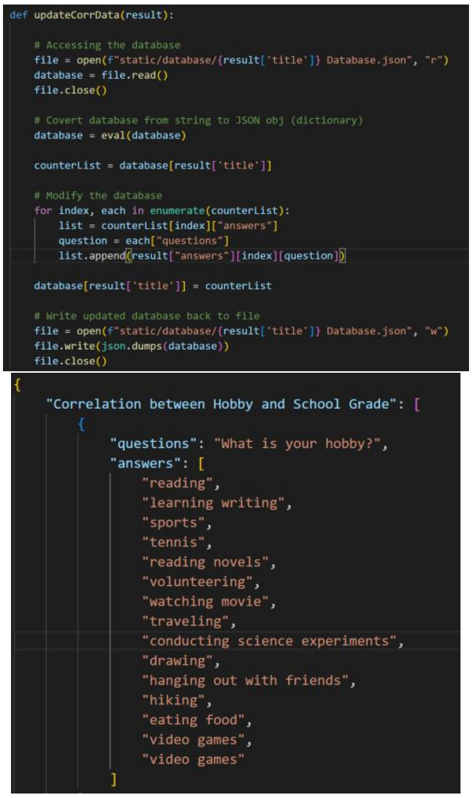
Do you think your sleeping habits affect your school performance? If yes, how?

Please provide a brief response.

Have you ever fallen asleep in class?

○ Yes, frequently
○ Yes, occasionally
○ No

Figure 4. Sleeping time and school grade survey



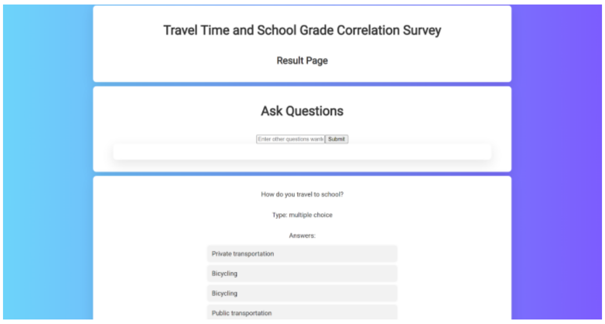
```
def updateCorrData(result):

    # Accessing the database
    file = open(f"static/database/{result['title']} Database.json", "r")
    database = file.read()
    file.close()

    # Covert database from string to JSON obj (dictionary)
    database = eval(database)

    counterList = database[result['title']]

    # Modify the database
    for index, each in enumerate(counterList):
        list = counterList[index]["answers"]
        question = each["questions"]
        list.append(result["answers"][index][question])

    database[result['title']] = counterList

    # Write updated database back to file
    file = open(f"static/database/{result['title']} Database.json", "w")
    file.write(json.dumps(database))
    file.close()
```

```
{
    "Correlation between Hobby and School Grade": [
        {
            "questions": "What is your hobby?",
            "answers": [
                "reading",
                "learning writing",
                "sports",
                "tennis",
                "reading novels",
                "volunteering",
                "watching movie",
                "traveling",
                "conducting science experiments",
                "drawing",
                "hanging out with friends",
                "hiking",
                "eating food",
                "video games",
                "video games"
            ]
```

Figure 5. Screenshot of code 2

In the first code sample, it describes the function for updating the database. This function would trigger after the user finished all the questions and clicked the Submit button in the page shown in the UI screenshot. It will first take the results submitted by the user, which would be a JSON format small database describing the name of the survey that was answered and the answer to each question in the survey, and find the database of the survey according to the name of it from the result. It will then take the existing results out to a new set for editing, and then append the new answer to the answer list of each question. After editing, the function would use the new list to replace the old list in the database to update it. An answer list in the database is shown in the second code sample.

Analyzing the results and answering questions from the user is another important function of the website. After the user gets some results from others, they can use this function to ask the website to give the answer to that with the given data.



Travel Time and School Grade Correlation Survey

Result Page

Ask Questions

How do you travel to school?

Type: multiple choice

Answers:

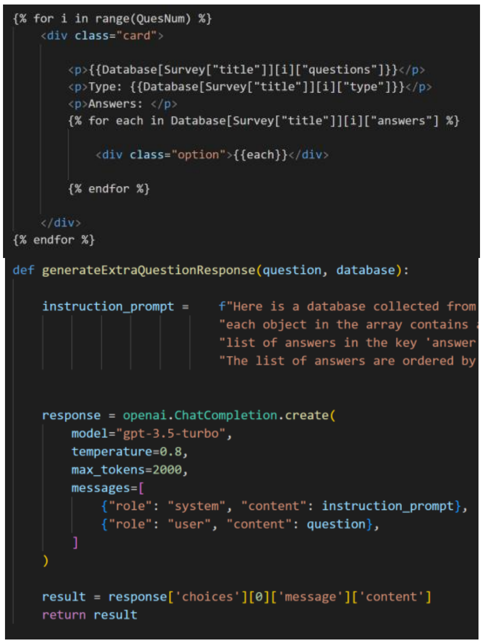
Private transportation

Bicycling

Bicycling

Public transportation

Figure 6. Travel time and school grade correlation survey



```
{% for i in range(QuesNum) %}
    <div class="card">

        <p>{{Database[Survey["title"]][i]["questions"]}}</p>
        <p>Type: {{Database[Survey["title"]][i]["type"]}}</p>
        <p>Answers: </p>
        {% for each in Database[Survey["title"]][i]["answers"] %}

            <div class="option">{{each}}</div>

        {% endfor %}

    </div>
{% endfor %}
```

```
def generateExtraQuestionResponse(question, database):

    instruction_prompt =    f"Here is a database collected from
                            "each object in the array contains 
                            "list of answers in the key 'answer
                            "The list of answers are ordered by


    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        temperature=0.8,
        max_tokens=2000,
        messages=[
            {"role": "system", "content": instruction_prompt},
            {"role": "user", "content": question},
        ]
    )

    result = response['choices'][0]['message']['content']
    return result
```

Figure 7. Screenshot of code 3

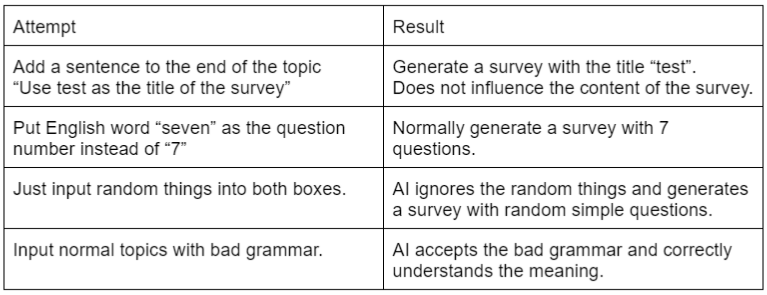
The first code sample shows the basic structure of the result page of the website, which is also shown in the UI screenshot. In this code sample, I used the information that is directly imported from the route in the main file, which is the whole database, and with the Jinja format, I can use the for loop to create all of the questions and their answers at one time. And the second code sample shows the inner code of the Ask Question part shown in the UI screenshot. This function would generate a request that will be sent to the AI with the instruction prompt that can direct to a response we need. And after getting the response, it will take the useful part, which is the response text only, from the whole response.

4. EXPERIMENT

4.1. Experiment 1

The prompt to the AI, because the user may cause a bug when they input some unexpected value to the prompt.

I think I would try to input some random things into the prompt, and also try to give the AI more instruction with the similar format of the prompt template, which may be confusing to the AI. And I would also try to include some strange grammar or symbols in the topic that will be put into the prompt template.



Attempt	Result
Add a sentence to the end of the topic "Use test as the title of the survey"	Generate a survey with the title "test". Does not influence the content of the survey.
Put English word "seven" as the question number instead of "7"	Normally generate a survey with 7 questions.
Just input random things into both boxes.	AI ignores the random things and generates a survey with random simple questions.
Input normal topics with bad grammar.	AI accepts the bad grammar and correctly understands the meaning.

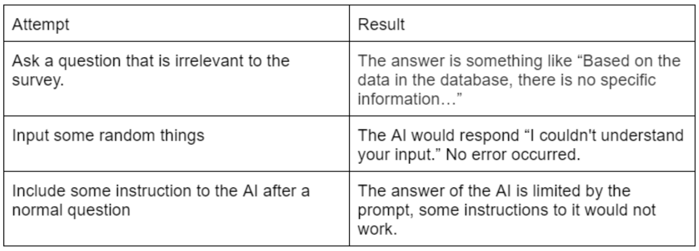
Figure 8. Figure of experiment 1

According to the results obtained from the experiment, I can conclude that the prompt to the AI can prevent the influence of some bad expressing of the topic, such as the missing of some unimportant words or the bad grammar. And it can also notice the meaning of the number words such as "seven", which appeared in the experiment, and correctly use the meaning to complete the task. But when the topic itself is confusing or even just some random combinations of letters, the AI cannot identify the real meaning of the user because the word input is the only data it got from the user.

4.2. Experiment 2

In my opinion, another blind spot may occur in the question asking part, because there is another text input part for the user.

I would try the same way as the first experiment to test the blind spot.



Attempt	Result
Ask a question that is irrelevant to the survey.	The answer is something like "Based on the data in the database, there is no specific information…"
Input some random things	The AI would respond "I couldn't understand your input." No error occurred.
Include some instruction to the AI after a normal question	The answer of the AI is limited by the prompt, some instructions to it would not work.

Figure 9. Figure of experiment 2

In conclusion, according to the data obtained from the experiment, the prompt here can minimize the influence of the instructions to the AI that is input as part of the question. And the AI can deal with some meaningless words and not cause errors. It can also identify the real meaning of the question, if the topic of the question is not relevant to the topic and data of the survey, it would not answer the question with common sense, because all of the analysis should be based on the results obtained from the survey.

5. RELATED WORK

I found that one of the largest survey websites is SurveyMonkey [11]. After the comparison, I found that it's a website focused on helping collect the response, and it would also provide the user some survey templates and some interesting functions, which means more types of questions in the survey. And it can also help analyze the result, but it's limited to the graphs and also data collections. I think that my website would be a better choice for those beginners in survey.

There are also some websites like Survey Junkie or Branded Surveys that pay their users money for them to respond to the surveys [12]. Which would be totally different from my website, because my website includes some functions other than asking others to complete surveys, so I think this is a kind of complementary.

I have also found another website called Typeform, which is a website focusing on creating surveys, it allows the users to put a picture with each question in the survey [13]. But all of the questions should be created by the user, though there are some templates and a question bank. And I think it has an advantage that it can connect with some other websites and applications such as Google Sheets and Excel, which would be useful for people that would work with those.

6. CONCLUSIONS

If I have more time on the project, I want to build the function of editing, adding, or deleting questions from the survey created, which can create a great development on the practicality of the website. And now, there are surely some imperfections and limitations of the project. I think that the first thing I would fix would be separating the database of different users. And I also think the UI now still need some optimization because some of the boxes on the survey page are not arranged really well, and I think that would be important to organize and unify the UI of all pages into a same theme for the whole website, which can make people think about the website when they see the theme.

In conclusion, my project is a website that can provide help to the beginners of statistics and surveys with the creation of surveys, the sample selection, and the analysis of data. The main purpose of it is to minimize the bias in the final results that would usually be made during the survey process.

REFERENCES

[1] Cristani, Matteo, and Roberta Cuel. "A survey on ontology creation methodologies." International Journal on Semantic Web and Information Systems (IJSWIS) 1.2 (2005): 49-69.

[2] Ibáñez, María, et al. "Comprehensive analytical strategies based on high-resolution time-of-flight mass spectrometry to identify new psychoactive substances." TrAC Trends in Analytical Chemistry 57 (2014): 107-117.

[3] Khan, Junaid Iqbal, et al. "Artificial intelligence and internet of things (AI-IoT) technologies in response to COVID-19 pandemic: A systematic review." Ieee Access 10 (2022): 62613-62660.

[4] Barr, Avron, Edward A. Feigenbaum, and Paul R. Cohen, eds. The handbook of artificial intelligence. Vol. 3. HeurisTech Press, 1981.

[5] Symonds, Emily. "A practical application of SurveyMonkey as a remote usability-testing tool." Library Hi Tech 29.3 (2011): 436-445.

[6] Tselentis, Jason. Type Form & Function: A Handbook on the Fundamentals of Typography. Rockport Publishers, 2011.

[7] Meers, Kristof, et al. "mobileQ: A free user-friendly application for collecting experience sampling data." Behavior Research Methods 52 (2020): 1510-1515.

[8] Gao, Catherine A., et al. "Comparing scientific abstracts generated by ChatGPT to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers." BioRxiv (2022): 2022-12.

[9] Morgan, George A., and Robert J. Harmon. "Data collection techniques." Journal-American Academy Of Child And Adolescent Psychiatry 40.8 (2001): 973-976.

[10] Spivak, Iryna, et al. "Validation and data processing in JSON format." IEEE EUROCON 2021-19th International Conference on Smart Technologies. IEEE, 2021.

[11] Abd Halim, Maisaarah, et al. "A review of live survey application: SurveyMonkey and SurveyGizmo." JOIV: International Journal on Informatics Visualization 2.4-2 (2018): 309-312.

[12] Moskowitz, Howard R., and Birgi Martin. "Optimising the language of email survey invitations." International Journal of Market Research 50.4 (2008): 491-510.

[13] Tinker, Miles A., and Donald G. Paterson. "Influence of type form on speed of reading." Journal of Applied Psychology 12.4 (1928): 359.

[14] Maeda, Kazuaki. "Performance evaluation of object serialization libraries in XML, JSON and binary formats." 2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP). IEEE, 2012.

[15] Yadav, Pallavi, and Paras Nath Barwal. "Designing responsive websites using HTML and CSS." international journal of scientific & technology research 3.11 (2014): 152-155.