Final Year Project Report on Automatic Quiz Generator with python

BY

Md. Shakaut Hassain
ID: 201-15-13740

This Report Presented in Partial Fulfilment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

Dr. Md Zahid Hasan
Associate Professor & Program Director MIS
Department of CSE
Daffodil International University

Co-Supervised By

Mr. Dewan Mamun Raza
Lecturer (Senior Scale)
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
January 2023

APPROVAL

This Project/internship titled **"Automatic Quiz generator with python"**, submitted by Md Shakaut hassain, ID No: 201-15-13740 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 19th January 2023.

BOARD OF EXAMINERS

Dr. Touhid Bhuiyan **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Narayan Ranjan Chakraborty **Internal Examiner**
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Raja Tariqul Hasan Tusher **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Md. Safaet Hossain **External Examiner**
Associate Professor & Head
Department of Computer Science and Engineering
City University

DECLARATION

We hereby declare that this project has been done by us under the supervision of Dr. Md Zahid Hasan, Associate Professor & Program Director MIS, Department of CSE in Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised By

Dr. Md Zahid Hasan
Associate Professor & Program Director MIS
Department of CSE
Daffodil International University

Co-Supervised By

Name Mr. Dewan Mamun Raza
Designation Lecturer (Senior Scale)
Department of CSE
Daffodil International University

Submitted By

Shakaut

Md Shakaut Hassain
ID: 201-15-13740
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, I express my heartiest thanks and gratefulness to Almighty God, for His divine blessing makes me possible to complete the final year project successfully.

I am really grateful and wish my profound and indebtedness to Dr. Md Zahid Hasan, Associate Professor & Program Director MIS, Department of Computer Science and Engineering in Daffodil International University, Dhaka. Deep knowledge & keen interest of my supervisor in **the field of** **“**Web application development**” to carry out** this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to Dr. Touhid Bhuiyan, Professor and Head, Department of Computer Science and Engineering in Daffodil International University, Dhaka, for his kind help to finish this project and also to other faculty members of the CSE department, Daffodil International University.

Finally, I must acknowledge with due respect the continuous support and patience of my parents.

ABSTRACT

Currently, this project is being worked upon a Web based development project titled **“Automatic Quiz Generator with python”.** This web based application will allow anyone to generate a bunch of quiz or question from a paragraph of text. This will help our teacher’s as well as any users like student or any person. This application will generate question automatically from any text. Sometimes teacher’s have to take quiz test or class test. And they can easily upload text from lecture slide and can generate quiz automatically. My system will help teacher’s a lot and also help to reduce time in conducting class test.

Additionally, it has the ability to pull fun details from Wikipedia articles on significant entities in the input text. This system depends on since it is topic independent Wikipedia, the greatest general reference work on the Web, is the source of DBpedia, a library of structured content.

Both our teachers and other users, such as students or other people, will benefit from this. This program will automatically create questions from any text. Teachers occasionally have to take quizzes or class tests. Additionally, they can quickly upload content from class slides and automatically create quizzes. My system will be of great assistance to teachers and will also shorten the time needed to administer class tests.

It can extract entertaining information from Wikipedia pages about important subjects in the input text. Since it is not topic-specific, this system is reliant. DBpedia, a collection of structured content, is derived from Wikipedia, the most comprehensive general reference resource on the Internet.

TABLE OF CONTENTS

CONTENTS **PAGE**

Approval i

Board of Examiners i

Declaration ii

Acknowledgement iii

Abstract iv

CHAPTER 1: INTRODUCTION 1-4

1.1 Introduction 1

1.2 Motivation 1

1.3 Objectives 2

1.4 Expected Outcome 2

1.5 Project Management 3

1.6 Report Layout 4

CHAPTER 2: BACKGROUND 5-8

2.1 Terminologies 5

2.2 Related Works 5

2.2 Related Works 6

2.3 Comparative Studies 7

2.4 Scope of the Problem 7

2.5 Challenges 8

CHAPTER 3: REQUIREMENT SPECIFICATION 9-12

3.1 Use Case Modeling and Description 9

3.2 Logical Data Model 10
3.3 Design Requirements 11
3.4 Implementation Requirement 12

CHAPTER 4: DESIGN SPECIFICATION 13-17

4.1 Front-End Design 13
4.1 Front-End Design 14
4.1 Front-End Design 15
4.1 Front-End Design 16
4.2 Back-End Design 17

CHAPTER 5: IMPLEMENTATION AND TESTING 18-24

5.1 Implementation 18
5.2 Implementation of Front-end Design 19
5.2 Implementation of Front-end Design 20
5.2 Implementation of Front-end Design 21
5.2 Implementation of Front-end Design 22
5.3 Implementation of Interactions 23
5.4 Testing Implementation 23
5.5 Test results and reports 24

CHAPTER 6: CONCLUSION AND FUTURE SCOPE 25

6.1 Discussion and Conclusion 25
6.2 Scope for Future Developments 25

REFERENCES 26

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1: Use Case Model	9
Figure 3.2: Logical data model	10
Figure 4.1.1: Input text/context field	13
Figure 4.1.2: Distractor buttons	14
Figure 4.1.3: Question level	15
Figure 4.1.4: Number of question	16
Figure 4.1.5: Export question button	16
Figure 5.1: Full overview of the work and generated quiz	18
Figure 5.2: Implementation of front end design	19
Figure 5.2.1: User Section for input	20
Figure 5.2.2: User Section for output	21
Figure 5.2.3: User Section after generating quiz	22

LIST OF TABLES

TABLES	PAGE NO
Table 5.1: Testing implementation of the application	23
Table 5.2: Result of testing implementation	24

CHAPTER 1
INTRODUCTION

1.1 Introduction

I’m going to make a automatic quiz generation application. Here anyone can generate quiz from simple text. With this application our teacher’s will get help a lot. They will be able to generate question from their lecture slide automatically. This application I designed for only the generate quiz. Teacher also can download all the question with answer in a text file. This application will help a lot and reduce time to the teacher. Teacher can be conduct quiz test or class test easily.

1.2 Motivation

Personally I highly motivated with a story of my own, that I will tell shortly now. Recently when I was in my class. Our one of my teacher was declared that she will conduct a class test in her next class. And the class date arrived our teacher also in our class in time. But she forgot that today was our class test. In that moment we missed the class test for that because of our teacher didn’t able to make question paper for us. And then I thought what if I can make such a system

that will generate quiz or question for us instantly from our lecture slide. It was just a thought but after couple of day’s I started to work with this project. I search different existing project, I read 5/8 research based paper’s to get knowledge. And those helped me a lot. That was a strong motivation for me I think.

1.3 Objectives

My main objective with this project is I will make a easier way to make automatic question generation from text of paragraph. This project will reduce our time a lot. I think it will be able to generate question from text perfectly than a human. I will add answer with the distractors that a teacher's is usually do in the time of exam. Teacher's will be able to conduct quiz test or class test easily by using my application.

1.4 Expected Outcome

My system will generate quiz as well as answer with distractors. By giving a bunch of text the output will be a specific number of quiz question. That quiz question will be generally like our actual quiz and question which teacher's makes a lot manually. My system also provide some advantage which is in below.

- The user interface will be user friendly.
- Anyone can download the quiz file.
- The application will keep the actual equivalent.
- The whole outcome will be a web app that will help us to make automatic quiz from text.
- Their will be an option to download the questions as quiz format into text file.

1.5 Project Management

The procedure for completing a job is through project management. It ensures that all experts involved will work to deliver outcomes that indicate a successful return on investment. Without formal organization, a few issues may arise. They combine a disconnect, unclear goals, inadequate resources, poor quality, and accepted assumptions.

Innovation: This application is reliable for making automatic question on the response of any text. This system can make above 90% of accurate question with response to the answer. I created with python library for backend as well as frontend design.

Time: This impossible evaluation quantifies the designs perspective in respect to the time, which is probably the most important factor to take into account. Time is a crucial issue in this project because this structure must be introduced on schedule. This structure is not very large, therefore it should be possible to complete it in the time allotted for its shellfish.

Resources: This estimate accounts for the materials that will be needed to build the system. The resources required for this structure are essential hardware and a working system, and they are both readily evident and attainable. I may get to the opinion that this system is workable from several estimating views and that it is possible to accomplish the project by looking at the estimations.

1.6 Report Layout

Report layout shows a type of overview of every item. All things considered, the below is a brief:

- **Chapter 1: S**hows a direction for the **“**Automatic Quiz Generation**”**, Motivation, Objectives, Expected outcome, Project management and the report Layout.
- **Chapter 2:** Described the background, related works, comparative studies, severity of the problem, and application challenges.
- **Chapter 3:** Describes the System requirements, Use case modeling, Logical data model and design requirement, implementation requirements.
- **Chapter 4:** Shows the design (Front and Back end), connection and UX and UI. Implementation.
- **Chapter 5:** Implementation of data storage, Detail the execute of front-end design, back-end design, testing technique, Functional testing, Unit test, Compatibility test, result and report.
- **Chapter 6:** End Discussion and extension for future possibilities of events.

CHAPTER 2
BACKGROUND

2.1 Terminologies

The system I worked is a automatic quiz generator. This application is for making question as well as answer from any text input. Automatically my system will store quizzes.

2.2 Related Works

Here I will describe recent projects and related studies that have been done in this field.

Jotform: Which online quiz creation method is best? The free quiz maker on Jotform! You can create interactive, multiple-choice tests for your class without writing a line of code, and they can even be scored automatically. Test your kids knowledge with customized online quizzes, or amuse your friends with Jotform powered trivia quizzes. Without using the Jotform site, you can access to your form data via the Jotform API.

The Jotform API was created with users and outside developers who wish to assist Jotform customers in heart. For instance, Zapier connects reply data with the other services using the Jotform API. Would you mind explaining how you want to use the Jotform API? You may freely complete this form. We'd be happy to assist you, and we'll even give you a designer to work with on your project!

Customers of Jotform have access to a slide form builder where they can alter any aspect of the form. One can also set codes. Several other critical software tools are also integrated into the application.

Quiz-maker: An online course can be created by linking various quizzes. Give out awards based on good marks in load classes, along with other criteria. To get started for free, just create your own quiz with in area above.

Students sign into the learning site to finish their tasks, monitor their progress, and get new quizzes. There will be no marking, multiple link delivery to students, progress tracking, or other boring tasks.

Working with your peers is possible with our premium and platinum editions of our online quiz builder. For access to and management of shared content, each team member can log in to the Dashboard. The 'Account' component of the Dashboard can be used to easily add more users.

To ensure that your respondents are properly authenticated and unable to alter results, make your quiz require logging in. Import an existing user list or enable registration for your respondents during the first quiz. To begin, select the "Quiz Login" option under the settings tab.

The task of giving feedback on tests and quizzes is one that no instructor enjoys. When a quiz asks for feedback, our Intelligence feedback system notifies you and, over time, starts to recommend remarks based on the input you've already provided. An automatic process can reduce an hours-long task to 15 minutes.

SurveyMonky: Want to rapidly create quizzes with a polished, professional look? Everything you want is given by SurveyMonkey, including analysis tools that make quiz scoring simple and enable you to analyze your results quickly.

To gain a baseline assessment of employees, students, or anyone you wish to test, turn your surveys into quizzes. Making your own quiz only requires a few easy steps: Open SurveyMonkey and select "Start from scratch" or one of our free online quiz templates. Your survey must include quiz questions. For each quiz question, click "Point this question (allow quiz mode)".

Use the plus or minus signs to provide points to your answer choices. Your quizzes can be shared by a web link, email, mobile chat, social media, and more. Utilize our quiz feedback tool, which enables you to provide personal notes on specific responses or the entire quiz. For instance:

Say "Great job!" while getting high marks.
After a wrong response, offer advice (such as "Better luck next time!").

2.3 Comparative Studies

I looked at some very well-known projects in the "Related Works" section that are great and is well for quiz maker delivery but not in our system. All the application is discussed is best in the field of quiz generation. Comparing to their system my system also able to give a good accuracy in quiz generating. Their system working well I watched all of them. And I'm working those features to make my system more interactive. Also they used some cloud based featured for analyzing and also cloud database. And these kind of features are my next target to implement.

2.4 Scope of the Problem

I am making this application with a good background which I used to generate. Their will be a system by which I will be able to get all matching outcomes. Teacher's can't find all the question manually. But this will give with a lot of question on same text paragraph. Thus it's a automatic process there is no problem of getting a specific amount of question.

Describes the issue's relevance, its growth over time, and the context of criminal justice management. More crucially, one needs to describe the policy problem and identify its underlying causes in order to do policy analysis.

Economic/market failure, legal/policy failure, governance/administrative failure, or social failure are some of the main causes. • Offer short- or long-term forecasts to show how the issue is expanding. This can be done by providing some facts together with graphs or charts to convey the issue's extent visually.

2.5 Challenges

Like every starting project I also found challenges in my project. From several challenges First I found sometimes my generate question as simple as it can. Because if anyone give input simple text obviously they will get simple question. And teacher for that can't conduct class test perfectly. Manually making question is more easy according to my problem of my project I think. And it's big challenge.

Second I challenge I found that when we making quiz we also making distraction with answer. And the problem is sometimes we get two or three distraction instance of getting 4. This is also a hard challenge for me to over come.
And then thirdly my challenge is I can't make quiz level. Like medium level, hard level etc.

In my next improvement I will work with these problem. And I will try to make quiz level thus student can get into various level. And the next challenge is to make my site a attractive look. For that many things have to work with like javascript, nodejs, flask etc. These kind of library are a big things to work. They are not easy to work. So I think I will try to learn this things and I will implement in my system

CHAPTER 3
REQUIREMENT SPECIFICATION

3.1 Use Case Modeling and Description

All the system of the work is showed in the Figure 3.1. On the left, the use case model is for user who will going to be use my application, which is the main part in this work. The admin application is also the user who will use my application. So user and admin are both the same in my project. Here user is directly can make quiz question. He/she can upload text or lecture slide and generate a specific amount of question and the use case is shown in Figure 3.1.

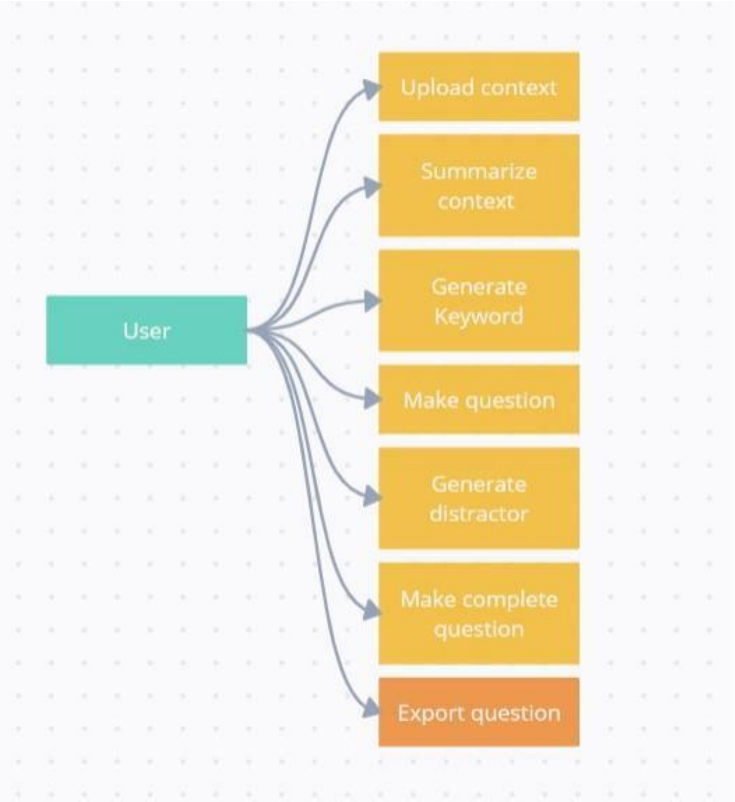


Figure 3.1: Use Case Model

3.2 Logical Data Model

The organization of collected data and their relationships are specified by a logical data model. The actual database that controls how data will be processed is missing from it. A structure for using data is provided by the logical data model. By adding more data, the logical data model enhances the aspects of applied data modeling. The Logical Data Model for the Application is given below in the Figure

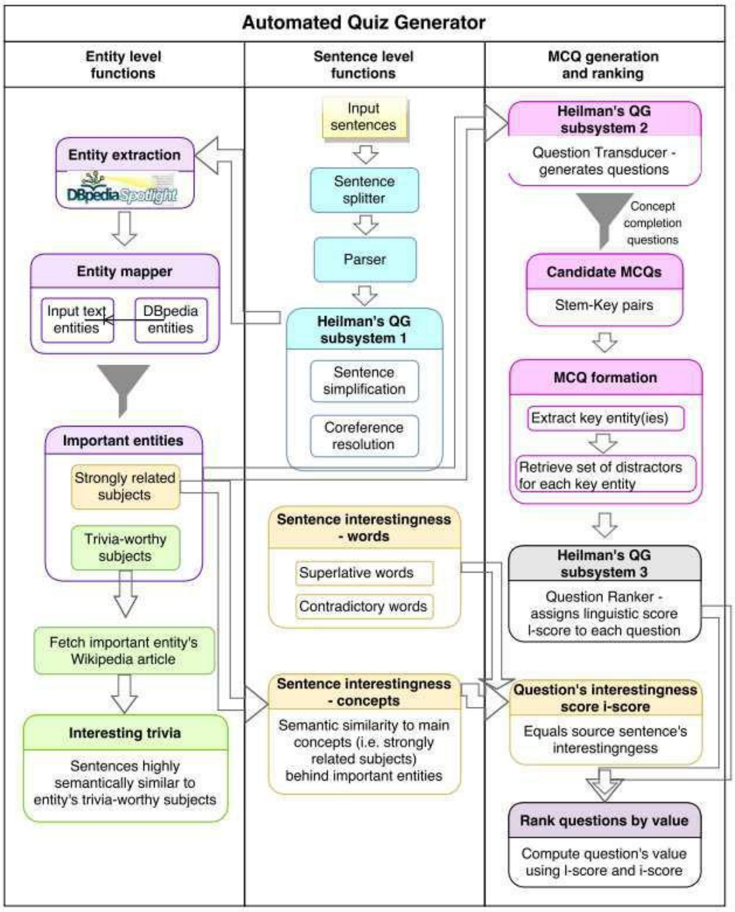

Automated Quiz Generator
Entity level functions
Sentence level functions
MCQ generation and ranking
Input sentences
Sentence splitter
Parser
Heilman's QG subsystem 1
Sentence simplification
Coreference resolution
Entity extraction
DBpedia Spotlight
Entity mapper
Input text entities
DBpedia entities
Important entities
Strongly related subjects
Trivia-worthy subjects
Fetch important entity's Wikipedia article
Interesting trivia
Sentences highly semantically similar to entity's trivia-worthy subjects
Sentence interestingness - words
Superlative words
Contradictory words
Sentence interestingness - concepts
Semantic similarity to main concepts (i.e. strongly related subjects) behind important entities
Heilman's QG subsystem 2
Question Transducer - generates questions
Concept completion questions
Candidate MCQs
Stem-Key pairs
MCQ formation
Extract key entity(ies)
Retrieve set of distractors for each key entity
Heilman's QG subsystem 3
Question Ranker - assigns linguistic score l-score to each question
Question's interestingness score i-score
Equals source sentence's interestingngess
Rank questions by value
Compute question's value using l-score and i-score

Figure 3.2: Logical Data Model

3.3 Design Requirements

Our primary objective is to provide clients with an application that is clear and simple to understand. Because of how consumer our task's setup is, we put an attempt to create our application similarly.

For admin section I made a input text field. Where users or teacher can give text input. Then I have two select button. Where admin can select the distractor type. One is wordnet and other is sens2vec. Then I have a submit library to generate quiz.

Designing requirements for User:

- Welcome screen
- Input field
- Question count
- Select button
- Submit button
- Question display
- Export question

3.4 Implementation Requirement

We utilized Visual Studio code to develop our web based apps for this project. Additionally for my interface I used one of python library that is gradio IO. Where I made all the things for user interface like buttons, input field. Display field. For this assignment, i constructed own web-based apps using Visual Studio Code. Additionally, I used the Gradio IO Python module for my interface. where I created the buttons and input fields for the user interface. display area

Software Requirements for Our Application:

- Visual studio code
- Python 3.6 or more
- Machine learning library for summarization (nltk library for summarization)
- And a lot of machine learning library like T5, wordnet, some text preprocessor library.

Hardware Requirements for Our Application:

- Any computer with minimum 4 gb ram.
- And any mobile device with any specification.

CHAPTER 4

DESIGN SPECIFICATION

4.1 Front-End Design

Input text field

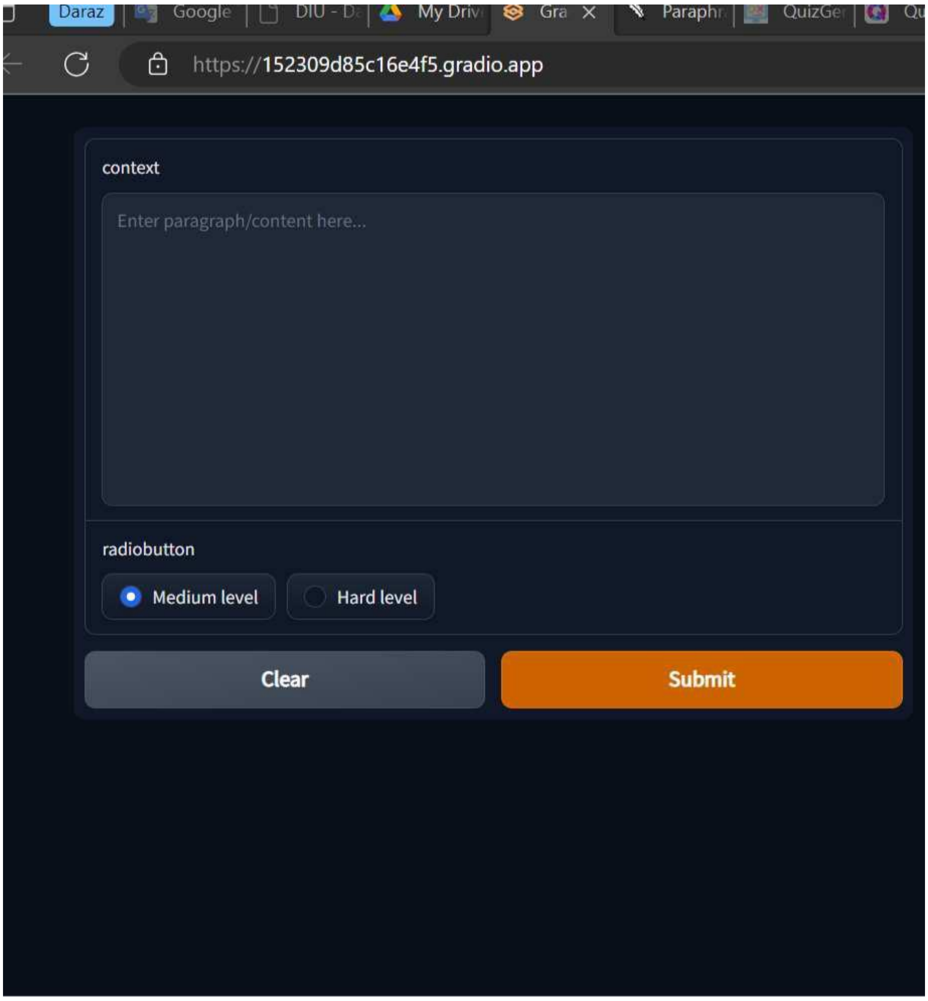
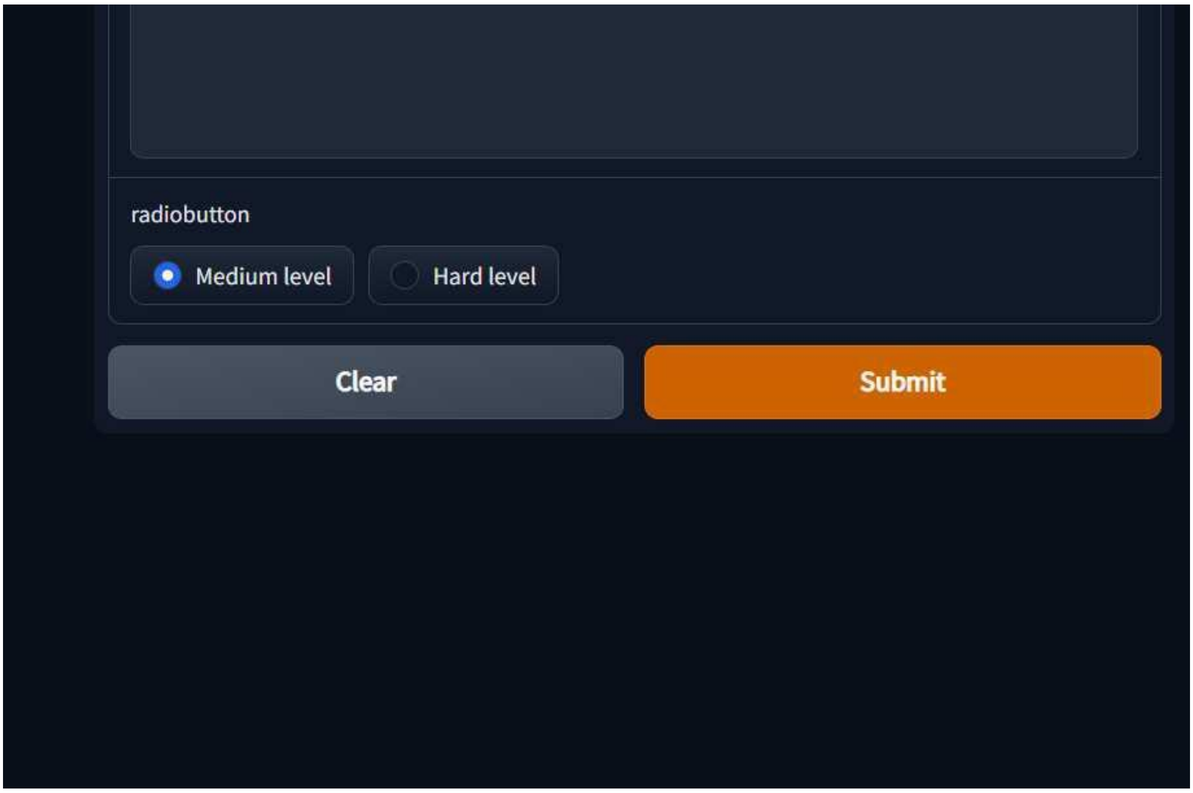


Figure 4.1.1: Input text/context field

Here we can see the input text/context field. Admin simply can paste text here. Any text from website or lecture slide can be paste here. Then my system will extract keyword and generate question from this text.

Button



radiobutton

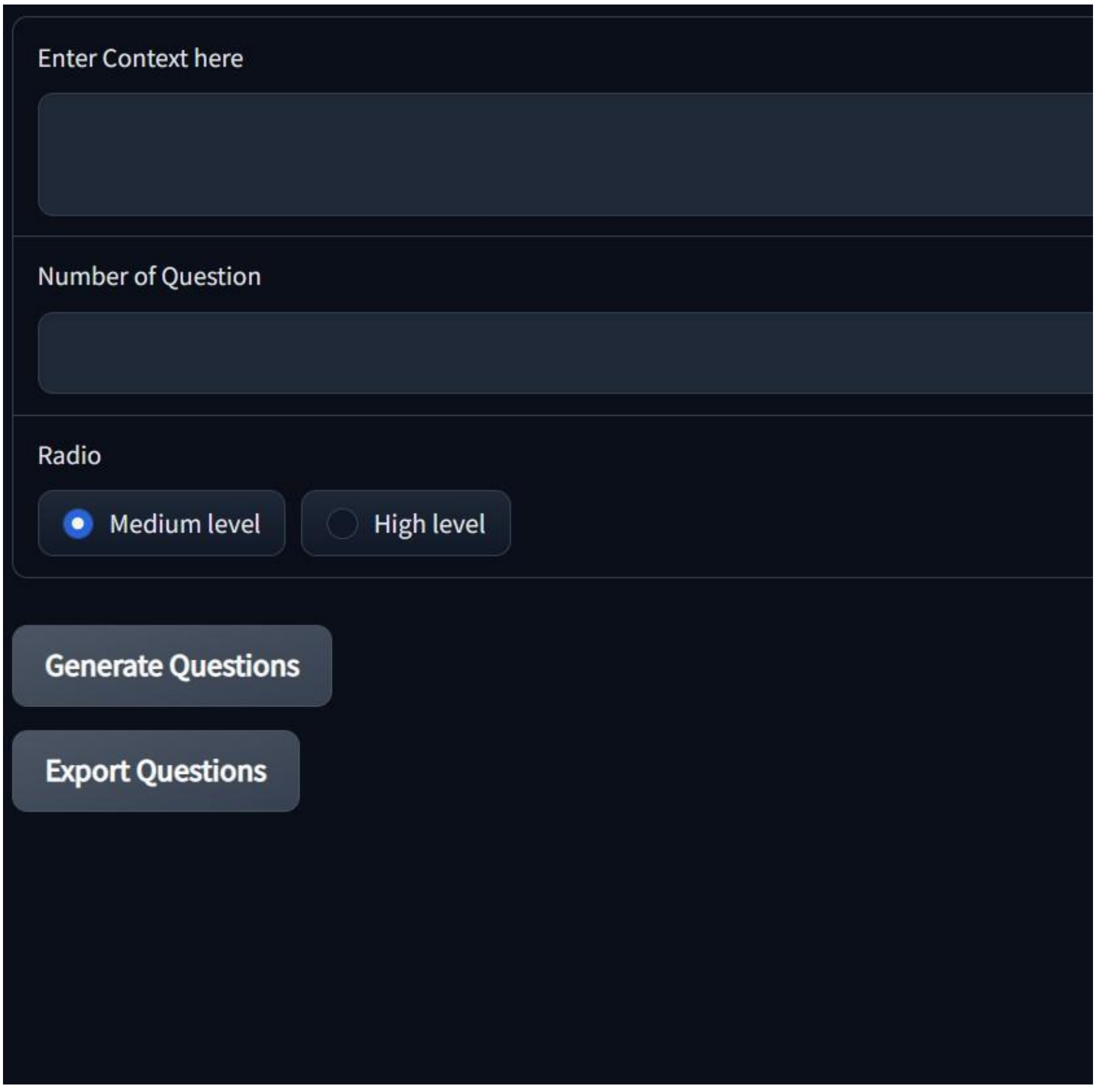
● Medium level ○ Hard level

Figure 4.1.2: Distractor buttons

Here we can see 2 button are added. This 2 button are responsible for distractor in quiz question. We all know there are 3 distractor always stay in the quiz with one correct answer.

First button is medium level is responsible for wordnet vector, which is a simple library for distractor and the second button is hard level which is for sen2vec also for generating distractor. Then we can see there are two more button in the bottom. One is clear and another is submit. Clear button will clear the input field and submit button is responsible for doing the quiz generator task. Submit button submit the input text to the system then all the work done.

Question Level



Enter Context here

Number of Question

Radio

● Medium level ○ High level

Figure 4.1.3: Question level

Here we can see question level radio button. These button will define the level of question's distractor, medium or hard. If we select medium radio button our system will generate distractors with wordnet model. And if we select hard level our system will generate distractors with ses2vec model.

Number of question

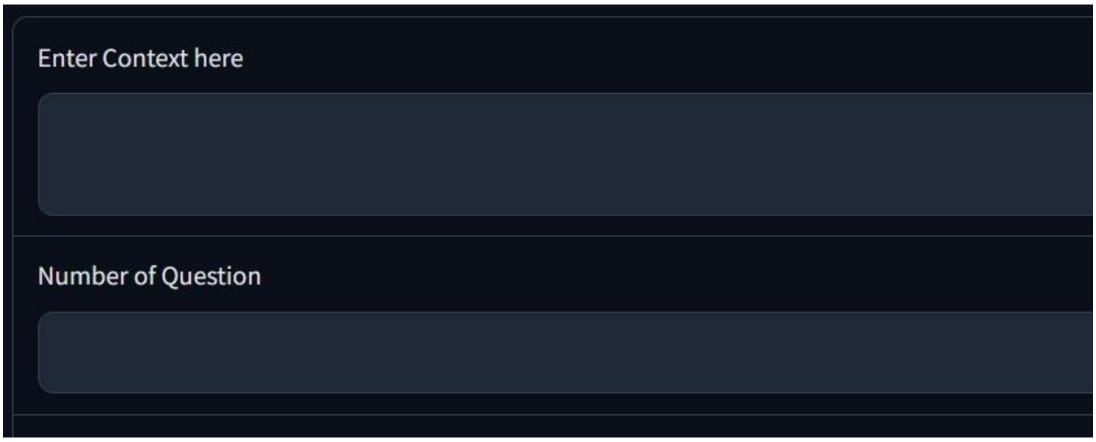
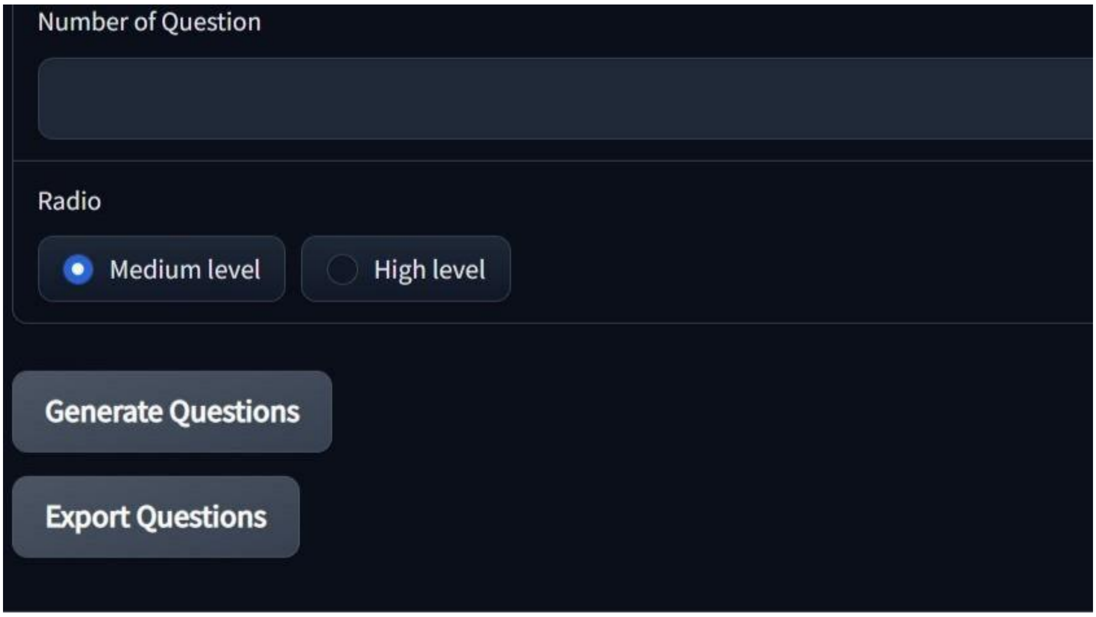


Figure 4.1.4: Number of question

That field is for the total number of question. If we give 10 here that will generate 10 question for use.

Export question button



Radio

● Medium level ○ High level

Figure 4.1.5: Export question button

We can export questions here by this button.

4.2 Back-End Development

For the back-end section I simply stores the text which user give input in our system. And then I summarize the whole text. Then I extracted key word with T5 model and give those keyword to my whole text. Then I used hugging face model to generate only question. After generation only question is used word2vec and sens2vec library to generating distractor.

And thus I store a complete quiz with answer and distractor. And whole things is done in back section of this project. We can first see the full UI for my project. The input context box, pick button, clear button, and submit button are all visible in the left section. Also, the created question for students is visible on the right side. The text is entered by the admin and sent to the system via the input area. There are various tasks that are built in order. The system begins by collecting the entire text, which is then reduced or reduced into a smaller section.

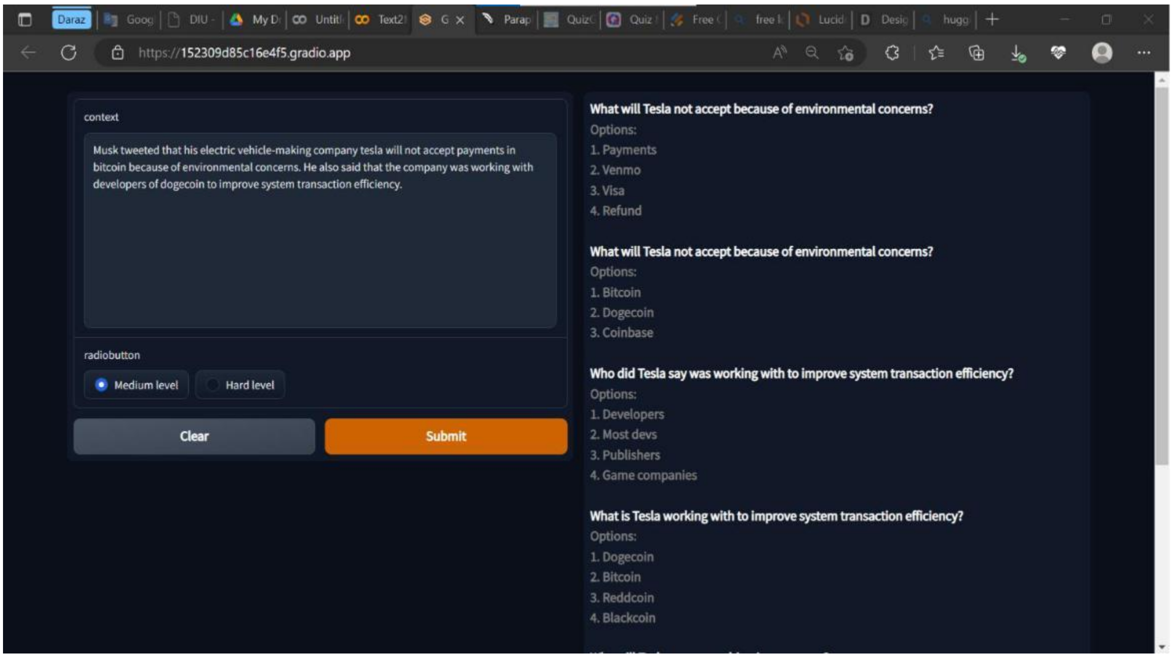
then extract the primary keyword for the query using the T5 model system. The entire text is then given this keyword so that it can existential. then extract the main keywords for the query using the T5 model system. The entire text is then given this keyword so that it can generate questions.

In response to the keyword, it 's different. I therefore need a break to focus on my quiz. It won't be a perfect quiz or question without a distractor. Sens2vec and the Wordnet library are our two options for doing it. According to the response, I can generate a distractor using both. Sens2vec is for complex words, and Wordnet is for simple words.

Gradio IO, a free Python library, is utilized here. I simply created such an interactive interface by utilizing this library. I added 2 choice buttons after taking the input text into account. After that, I used the Gradio library to activate 2 extra buttons: clear and submit. This library can be used to develop web apps for free.

CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation



Musk tweeted that his electric vehicle-making company tesla will not accept payments in bitcoin because of environmental concerns. He also said that the company was working with developers of dogecoin to improve system transaction efficiency.

● Medium level ○ Hard level

What will Tesla not accept because of environmental concerns?
Options:
1. Payments
2. Venmo
3. Visa
4. Refund

What will Tesla not accept because of environmental concerns?
Options:
1. Bitcoin
2. Dogecoin
3. Coinbase

Who did Tesla say was working with to improve system transaction efficiency?
Options:
1. Developers
2. Most devs
3. Publishers
4. Game companies

What is Tesla working with to improve system transaction efficiency?
Options:
1. Dogecoin
2. Bitcoin
3. Reddcoin
4. Blackcoin

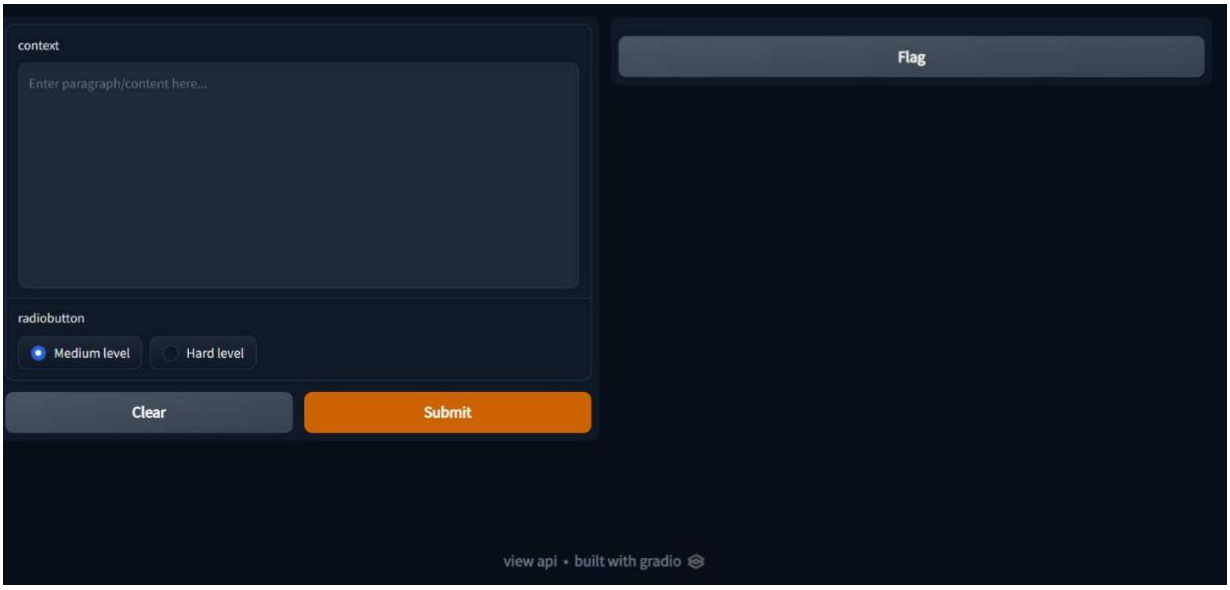
Figure 5.1: Full overview of the work and generated quiz

At first we can see the whole interface of my project. In the left section we can see the input context field, select button and clear, submit button. And in the right side we can see the generated question for students. Input field takes input from admin and pass the whole text to the system. There a lot of task is done sequentially. First system takes all the text and then it shorten or summarize the whole text into a small portion. Then using T5 model system extract the main keyword for making question. Then this keyword is provided to the whole text to generate question from those keyword. It generate question response to the keyword. And then I need some distractor for my quiz. Without distractor it won't a perfect quiz or question. For that we

have to method first is wordnet library and second is sens2vec library. With both I can generation distractor according to the answer. Wordnet is for simple word and sens2vec is for complex word.

5.2 Implementation of Front-end Design

Here is used a free python library named gradio IO. By using this library I simply designed such a interactive interface. First I took input text gradio then I added 2 select button. After that I used two more button one is clear and another is submit by using gradio library. This library is free web application development library. I created two pick buttons after taking the input text into Gradio. Then, using the Gradio library, I used two additional buttons: clear and submit. This library can be used for the free construction of web applications.

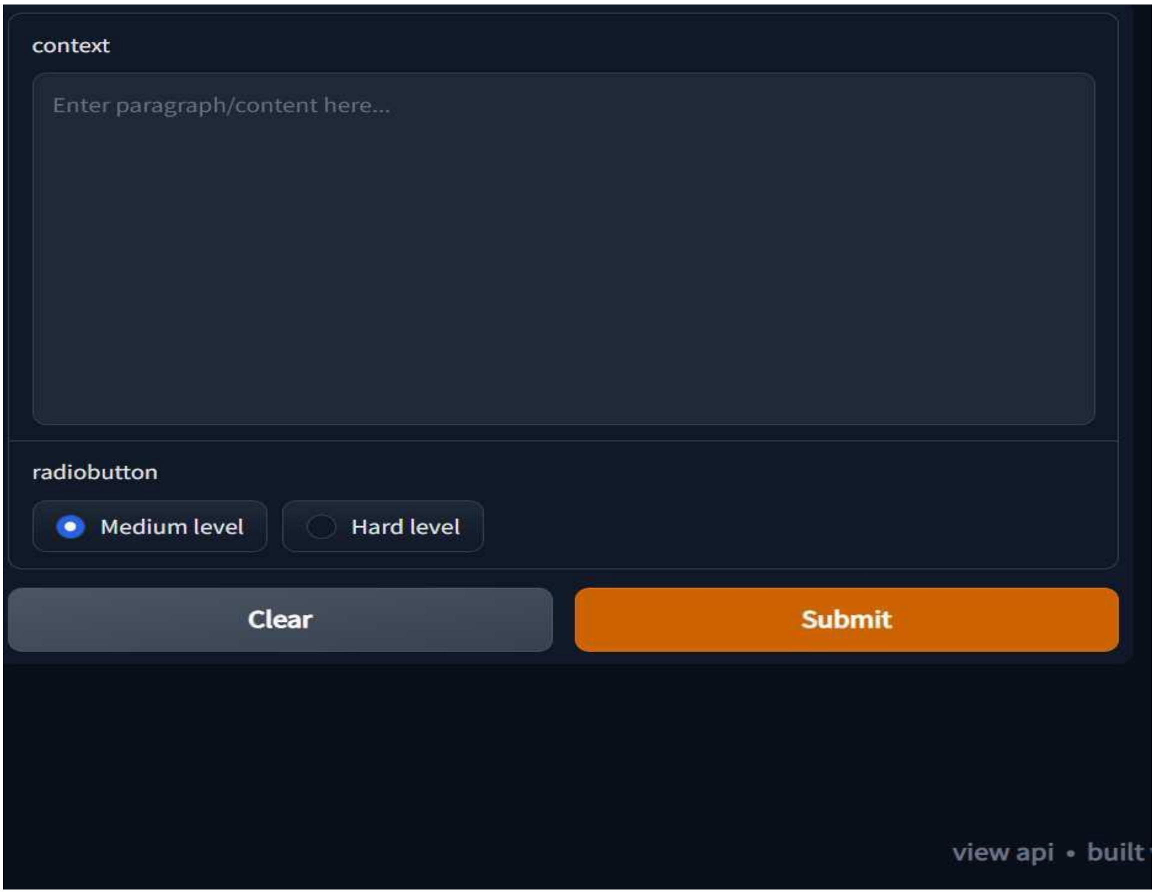


radiobutton

● Medium level ○ Hard level

Figure 5.2: Implementation of front end design

User Section for giving input



radiobutton

● Medium level ○ Hard level

Figure 5.2.1: User Section for input

This area is for administrators. Where admin can input text and generate quiz from text. Admin will get a input section here. And also quiz level button by this admin will define the quiz level. Is it hard or medium. where the administrator can enter content and create a quiz from it. Admin will receive a section for input. Additionally, this admin's quiz level button will specify the quiz level.

User’s Section for displaying questions

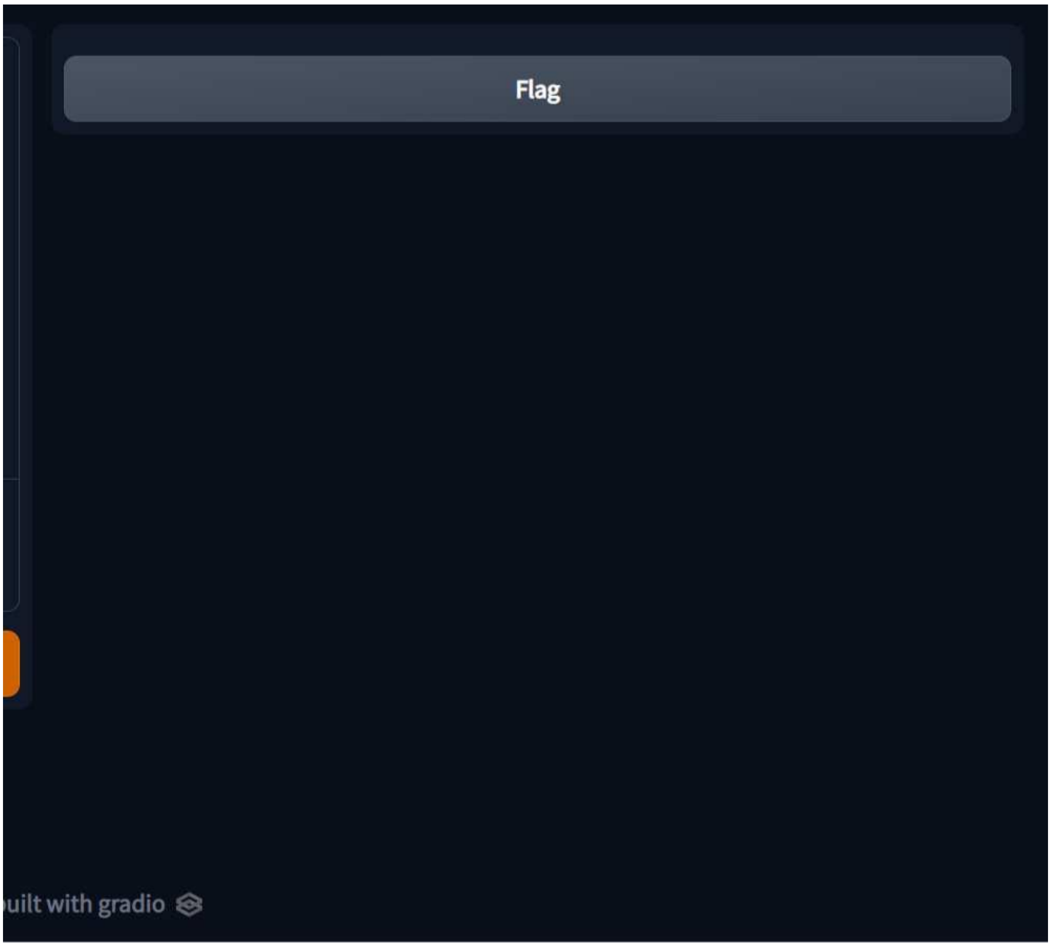
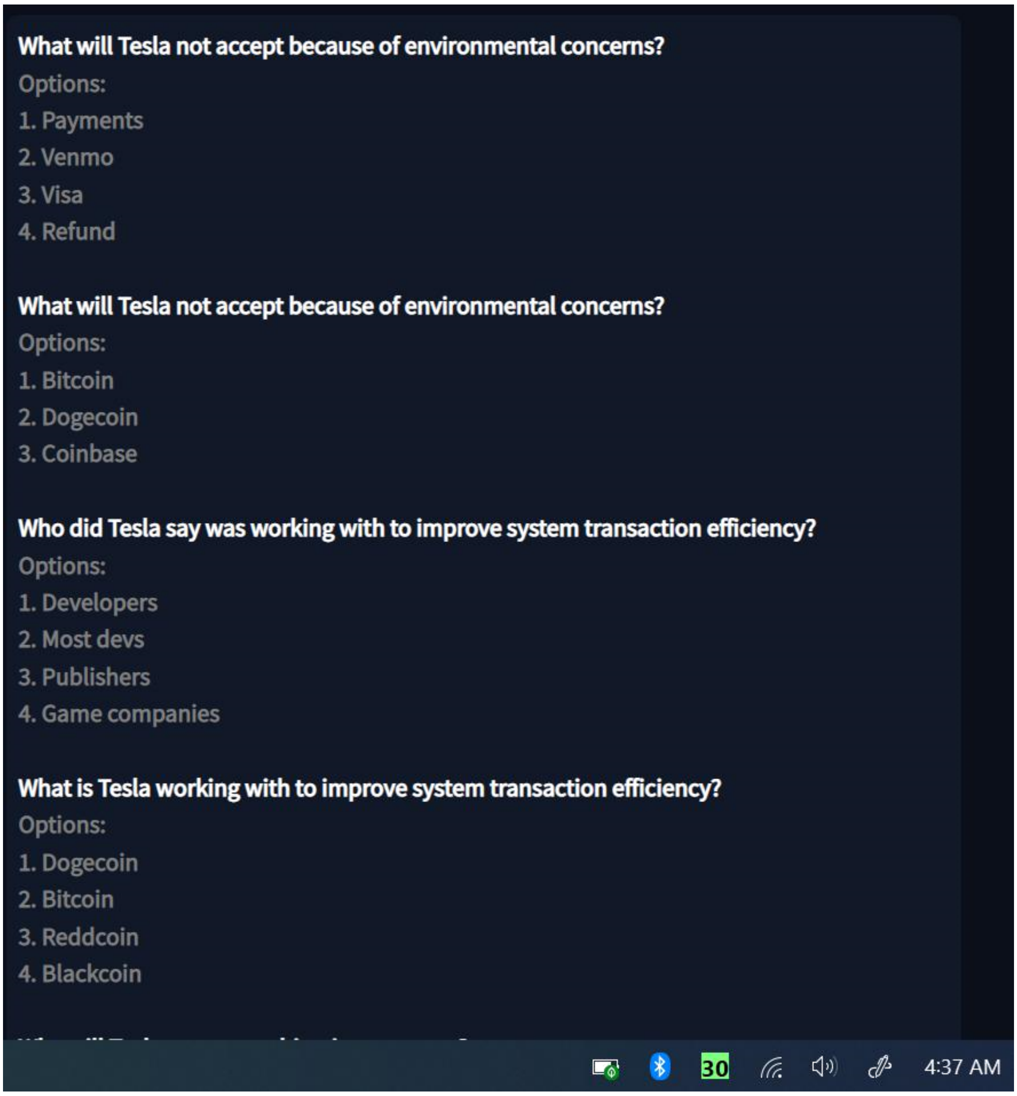


Figure 5.2.2: User Section for output

Initially user section is empty. But quiz will display over here with specific number of count.



What will Tesla not accept because of environmental concerns?

Options:

1. Payments
2. Venmo
3. Visa
4. Refund

What will Tesla not accept because of environmental concerns?

Options:

1. Bitcoin
2. Dogecoin
3. Coinbase

Who did Tesla say was working with to improve system transaction efficiency?

Options:

1. Developers
2. Most devs
3. Publishers
4. Game companies

What is Tesla working with to improve system transaction efficiency?

Options:

1. Dogecoin
2. Bitcoin
3. Reddcoin
4. Blackcoin

Figure 5.2.3: User Section after generating quiz

We can see here our quiz are displayed. First the question and then all the option are here in the quiz display section. Here one option is answer and other three option is distractor. We can see all the distractor are similar to actual answer. I used here wordnet and sens2vec to create distractor.

5.3 Implementation of Interactions

We created an app with the client's wish to use a light, user-friendly, and user-interactive as our main focus. The method we used enabled us to establish a clear relation with the big, diverse, and diverse consumer base. For this, our users can interact much more with our app. Admin can automatically change text and also select question manually.

5.4 Testing Implementation

I put my work to the test when several of our initiatives were still in the early phases of development. **I'm** quite familiar with customer requirements because I lack experienced testers and have learned about them from our company's knowledge

Table 5.4.1: Testing implementation of the application

01	Checking if the program runs
02	Checking for crash on startup
03	Checking the input section taking input or not
04	Checking the select button working or not
05	Checking the clear button and submit button working or not
06	Checking the model can generate question properly
07	Checking the distractor and answer properly showing or not
08	Checking the display section display quiz or not

5.5 Test Results and Reports

Table 5.5.1 showing the outcomes of the tests, as declared by the test cases presented in the earlier part of this chapter.

Table 5.5.2: Result of testing implementation

01	Checking if the program runs	Success
02	Checking for crash on startup	Success
03	Checking the input section taking input or not	Success
04	Checking the select button working or not	Success
05	Checking the clear button and submit button working or not	Success
06	Checking the model can generate question properly	Success
07	Checking the distractor and answer properly showing or not	Success
08	Checking the display section display quiz or not	Success

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Discussion and Conclusion

Deep learning and natural language processing have already seen significant technological improvement in recent years. One of the uses is the creation of tests directly from text. NLP techniques have recently advanced, and this holds a lot of promise. The suggested solution extracts keywords and data from the input text using an NLP pipeline that includes Bert and T5 transformers. Different varieties of questions, including fill-in-the-blank, true/false, Wh-type, and multiple choice questions, are created from the extracted keywords. Modern models have shown high results throughout our workflow. These models' outputs have showed a great deal of potential. In a survey intended to evaluate the model, over 60% of the questions

6.2 Scope for future developments

- I will do a lot of improvement with this project in future. Where student alone can test them self with a test exam.
- Making more interactive and full of fresh article.
- Adding more extra features as per customer choice.

REFERENCES

[1] QuizGenerator, Abailable at: QuizGenerator | Free quiz and question generator, Last Accessed on 03 December, 2022

[2] Jotform, Abailable at: Free Online Quiz Maker | Multiple Choice Quiz | Jotform, Last Accessed on 03 December, 2022

[3] Survey-munkey, Abailable at: **SurveyMonkey: The World's Most Popular Free Online** Survey Tool, Last Accessed on 03 December, 2022

[4] Riddle (auto-quiz), Abailable at: Quiz generator (auto-quiz): show new questions to each quiz taker (riddle.com), Last Accessed on 03 December, 2022

[5] Flexudy, Abailable at: https://www.flexudy.com/ , Last Accessed on 03 December, 2022

[6] Typeform, Abailable at: https://www.typeform.com/ , Last Accessed on 03 December, 2022

Automatic Quiz Generator with python

ORIGINALITY REPORT

11%	11%	1%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	6%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	3%
3	www.coursehero.com Internet Source	<1%
4	George Tillmann. "Usage-Driven Database Design", Springer Science and Business Media LLC, 2017 Publication	<1%
5	repository.president.ac.id Internet Source	<1%
6	kitschmag.com Internet Source	<1%
7	www.slideshare.net Internet Source	<1%
8	dspace.library.daffodilvarsity.edu.bd:8080 Internet Source	<1%
	ec.mediatagtw.com	

9	Internet Source	<1%
10	**www.riddle.com** Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		