AUTOMATIC QUESTION PAPER GENERATOR

Internship Report Submitted In Fulfillment Of The Requirements For The Degree Of

B.Tech

In

Computer Science and Engineering

By

Stuti Sharma

20103043

Department Of Computer Science Engineering And Information Technology,

Jaypee Institute Of Information Technology,

A-10, Sector-62, Noida, Uttar Pradesh

June/2023 - July/2023

INDEX

ABOUT THE ORGANIZATION

Jaypee Institute of Information Technology (JIIT) is a prestigious private technical university located in Noida, India. Established in 2001, it is a part of the Jaypee Group and has gained recognition for its high-quality education and research in engineering, computer science, and allied disciplines. JIIT offers undergraduate, postgraduate, and doctoral programs in various fields, including computer science, electronics, biotechnology, and management.

The institute is known for its modern infrastructure, well-equipped labs, and a strong emphasis on research and innovation. It fosters a vibrant academic environment that encourages students to explore emerging technologies and contribute to advancements in their respective fields. JIIT has collaborations with leading international universities and research organizations, providing students with exposure to global perspectives.

With a commitment to excellence in education and a focus on holistic development, JIIT has produced skilled professionals and researchers who contribute to the industry and academia. The university's emphasis on practical learning, industry interactions, and interdisciplinary education has contributed to its reputation as a hub for technical education and innovation.

Prof. (Dr.) Vikas Saxena, SMIEEE
Ph.D. (CSE), M.E. (CE), B.Tech. (CS&IT)
Head-Department of CSE&IT
Jaypee Institute of Information Technology
A-10, Sector-62, Noida, India, 201307
+91-120-2594273, vikas.saxena@jiit.ac.in

E-Certificate Number: CoE-AIE\SIT-2023\28

To whomsoever it may concern

This is to certify that Mr./Ms. Stuti Sharma (Enrollment no. 20103043) a student of BTech (CSE) from Jaypee Institute of Information Technology has successfully completed the summer training under the supervision of Dr. Indu Chawla from 12th June 2023 to 31st July 2023 (7 weeks) working in a project under Center of Excellence on AI for Education.

We wish him all success in the future.

Prof. Vikas Saxena

PROJECT OBJECTIVE

A software program known as an automated question paper generator (AQPG) may create question papers on demand for many different courses. Typically, it operates by accepting a database of questions as input along with some restrictions on the exam, such as the quantity of questions, the level of difficulty, and the subjects to be covered. The AQPG then creates a test that complies with all the requirements using a randomization mechanism.

An AQPG's goal is to automate the time-consuming and prone to mistake process of creating questions for exams. AQPGs can significantly reduce the amount of time and work required from instructors and examiners by automating this procedure. As the randomization process employed by AQPGs assures that all questions have an equal probability of being picked, they can also aid in ensuring that question papers are impartial and fair.

AQPGs can be used to create more difficult and interesting question papers in addition to saving time and assuring fairness. AQPGs can aid in ensuring that students are suitably challenged by letting teachers to set the difficulty level of the questions. Additionally, they may be utilized to create question papers that are more closely related to the course's unique learning objectives.

In general, AQPGs are a useful tool for instructors and test takers. They can help create more difficult and interesting test questions while also ensuring fairness and saving time.

The project's goal is to create a Regular Expression (Regex)-based Automatic Question Paper Generator. The system's main goal is to automate and streamline the process of writing exams for various educational reasons. The project seeks to provide a versatile and effective mechanism to construct question patterns, kinds, and constraints by utilizing the power of Regex expressions. This will make it possible for teachers and content producers to enter particular criteria for question generation, such as question forms, degree of difficulty, and subject-specific factors. The project's ultimate goals are to increase the effectiveness of question paper development, minimize manual labor, and guarantee the production of varied and well-structured question papers specifically designed for educational contexts.

PROJECT DESCRIPTION

A) DESCRIPTION OF THE WORK CARRIED OUT:

The Automatic Question Paper Generator using Multiple Choice Questions (MCQs) in Python, with the utilization of regular expressions (regex) for question generation, is a project that involves automating the creation of assessment materials. This system is particularly useful for educators and instructors who need to generate question papers quickly and efficiently.

The work that went into this project is broken down as follows:

1) **Collection and preparation of data sets:**
   a) Assembled a MCQ dataset in a csv file.
   b) Every multiple-choice question (MCQ) consisted of a question, difficulty level, a list of possible answers, and an indication of the right answer.

2) **Data Preprocessing:**
   a) The dataset was parsed to extract the question stems, answer choices, and correct answers.
   b) The text was cleaned by removing unnecessary characters, special symbols, and formatting inconsistencies.

3) **Regex-based Question Generation:**
   a) Defined a set of regex patterns that can capture different types of question structures. For example, patterns to identify questions that ask for definitions, comparisons, calculations, etc. Used these patterns to generate new questions.

4) **Generation of Answer Options:**
   a) Created logical answer options based on the created questions. These options were produced by changing the right response, including widespread misconceptions, or by utilizing obfuscators from the initial dataset.

5) **The proper response is:**
   a) Made sure that the right response is properly matched to the created answer choices using the data from the original dataset or the generated question.

6) **Randomness and shuffle:**
   a) Used shuffling algorithm to ensure no question is repeated therefore making the system

7) **Generation of Exam Papers:**
   a) Compiled the selected questions into a structured question paper format, including question numbers, questions, answer choices, and spaces for students to mark their answers.

8) **User Interface:**

a) Developed a graphical user interface (GUI) that allows users (educators) to input parameters such as the number of questions, topic preferences, difficulty levels and number of papers required.
b) The GUI can then display the generated question paper, which can be saved or printed.
c) Established the necessary number of questions for a question paper.
d) A structured question paper is being created using the chosen questions, complete with question numbers, new generated questions, answer options, and spaces for students to record their responses.

By combining a dataset of MCQs, regex patterns for question generation, and systematic processes for handling answer choices and question paper assembly, this project aims to automate the time-consuming task of creating question papers while maintaining the quality and diversity of questions.

B) DETAILS OF DATA COLLECTED:

Prepared a dataset of around 100 questions on the topic Strings in python. Many sites were referred for the preparation of unique questions. The dataset consists of 8 columns named as S.No., question, difficulty, answer, distractor1, distractor2, distractor3, distractor4.

The dataset is well-structured for training an automatic MCQ generator. Here's a breakdown of the columns and their purposes:

1) S.No. (Serial Number):

a) This column represents a unique identifier for each question in the dataset.
b) It helps maintain order and reference to specific questions.

2) Question

a) This column holds the text of the question stem for each MCQ.
b) It presents the problem or scenario that students need to respond to.

3) Difficulty Level:

a) This column indicates the level of difficulty associated with the question.
b) Difficulty levels could be categorized as easy, moderate, or hard, or you could use numerical values for more granularity.

4) Answer:

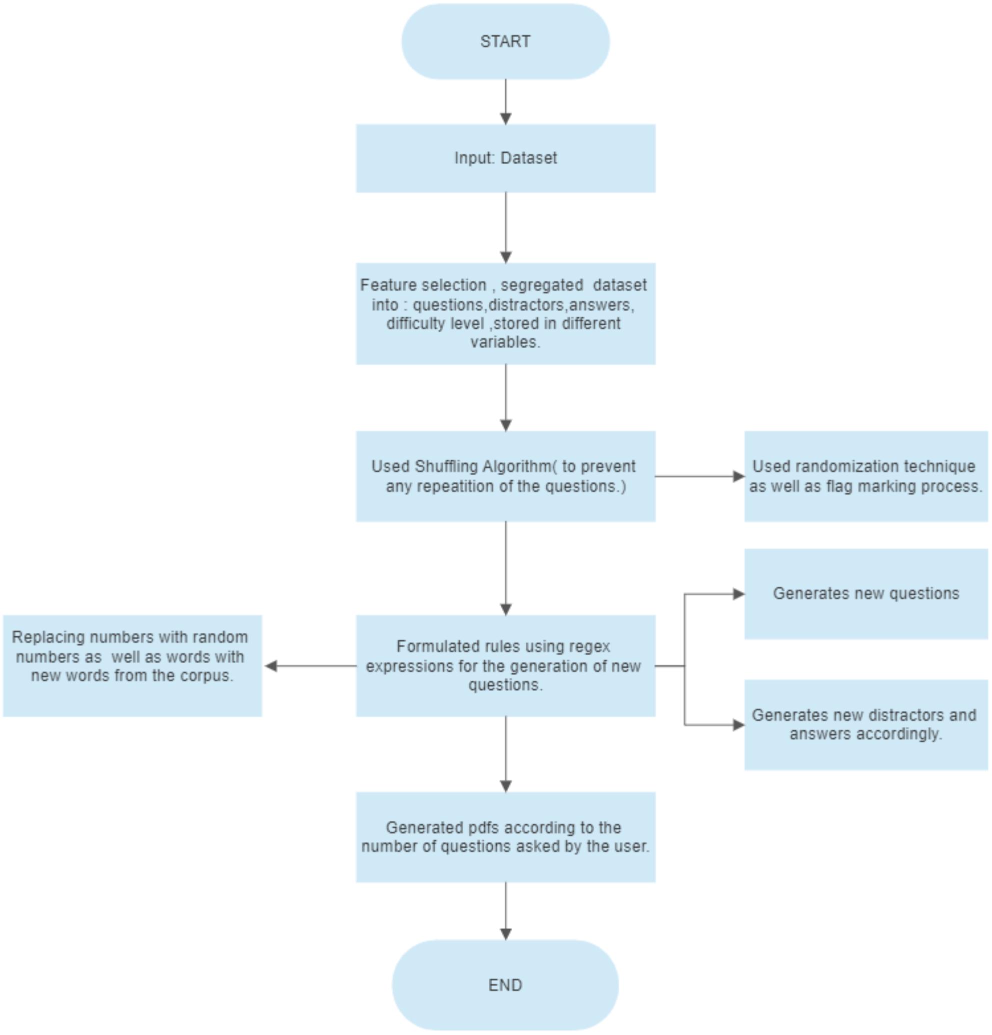
a) This column contains the correct answer for each question.
b) It's the choice that students should ideally select as their response.

5) Distractor 1 to Distractor 4:

a) These columns hold the incorrect answer choices, also known as distractors.
b) Distractors are designed to confuse students who haven't fully grasped the concept or are guessing.
c) Each distractor should be plausible and relevant to the question, making the options challenging.

Having this structured dataset is beneficial for the MCQ generator because it provides a clear separation between the various components of a question. This makes it easier to manipulate and generate new questions while ensuring that the generated questions remain coherent and relevant.

C) DESIGN CARRIED OUT:


START
Input: Dataset
Feature selection , segregated dataset into : questions,distractors,answers, difficulty level ,stored in different variables.
Used Shuffling Algorithm(to prevent any repeatition of the questions.)
Used randomization technique as well as flag marking process.
Generates new questions
Replacing numbers with random numbers as well as words with new words from the corpus.
Formulated rules using regex expressions for the generation of new questions.
Generates new distractors and answers accordingly.
Generated pdfs according to the number of questions asked by the user.
END

This Python script generates Multiple Choice Questions (MCQs) based on a dataset stored in a CSV file. It uses the pandas library to read the data from the CSV file and generates random variations of the questions and options to create a question paper with shuffled MCQs.

D) USAGE/SKILLS AND TECHNOLOGIES USED:

- Python:
  Python is a popular choice for text manipulation tasks due to its simplicity and strong support for regular expressions.

- Text Processing libraries :
  pandas: provides versatile data structures and data analysis tools, with its core data frame function offering a powerful way to manipulate and analyse tabular data.
  re (regex module): Python's built-in re module provides support for working with regular expressions. It allows to search, match, and manipulate strings based on complex patterns.
  random: for generating random numbers, making it useful for tasks like simulations, shuffling, and random selections.
  nltk: provides tools for working with human language data, including text processing, tokenization, stemming, tagging, parsing, and more.
  string: provides a set of operations and methods for manipulating and working with strings in programming languages like Python.
  fpdf: used to create PDF documents by providing a high-level interface for adding text, images, and other elements to the PDF pages.

- File Handling:
  To read and manipulate files containing the source material, and the generated questions.

- Data Structures:
  Data structures like lists, dictionaries, or maps are used to hold the extracted information and the generated questions.

- User Interface
  A simple user interface to input the source material and receive the generated question papers using html, css and javascript and using flask (a micro web framework for Python used to build web applications)

RESULTS AND CONCLUSION

Examples of questions generated according to the rules:

Example 1

Ques. str = "My salary is 7000";

print(str.isalnum())

Difficulty Level: Easy

A. TRUE

B. FALSE

C. Error

D. No output

Ans. FALSE

->Rules Used: rule 2, rule 10

Result:

Question: str = "epidialogue";

print(str.split())

Difficulty Level: Easy

A. ['epidialogue']

B. epidialogue

C. Error

D. No output

ANS. ['epidialogue']

Example 2:

Ques. str1 = 'Welcome'

print(str1*2)

Difficulty Level: Easy

A. WelcomeWelcome

B. TypeError unsupported operand type(s)

C. Welcome

D. No Output

Ans. WelcomeWelcome

->**Rules Used**: rule 2, rule 4

Result:

Question: str1 = 'Lingulella'

print(str1*3)

Difficulty Level: Easy

A. LingulellaLingulellaLingulella

B. Lingulella

C. No Output

D. Error

ANS. LingulellaLingulellaLingulella

Example 3

Ques. animal = "dog"

print("animal" + animal)

Difficulty Level: Medium

A. dogdog

B. dog dog

C. animal dog

D. animaldog

Ans. Animaldog

->**Rules Used**: rule 2, rule 13

Result:

Question: hydrophylliaceous= "hemisaprophytic"

print("hydrophylliaceous" + "demonkind")

Difficulty Level: Easy

A.   hemisaprophytic hydrophylliaceous demonkind

B. demonkindhydrophylliaceoushemisaprophytic

C. hemisaprophytichydrophylliaceousdemonkind

D. demonkind hydrophylliaceous hemisaprophytic

ANS. hemisaprophytichydrophylliaceousdemonkind

Example 4

Ques. print(ord('foo'))

Difficulty Level: Medium

A. 102 111 111

B. It raises an exception

C. 102

D. 324

Ans. It raises an exception

->**Rules Used**: rule 2

Result:

Question: print(ord('untortured'))

Difficulty Level: Medium

A. 102 111 111

B. It raises an exception

C. 102

D. 324

ANS. It raises an exception

Example 5:

Ques. list((b'abcde' + 'fghi')[3:6])

Difficulty Level: Medium

A. b'def'

B. [100, 101, 102]

C. [b'd', b'e', b'f']

D. It raises an exception

Ans. It raises an exception

->**Rules Used:** rule2, rule8

Result

Question: list((b'diffluence' + 'bacilluria')[5:7])

Difficulty Level: Medium

A. en

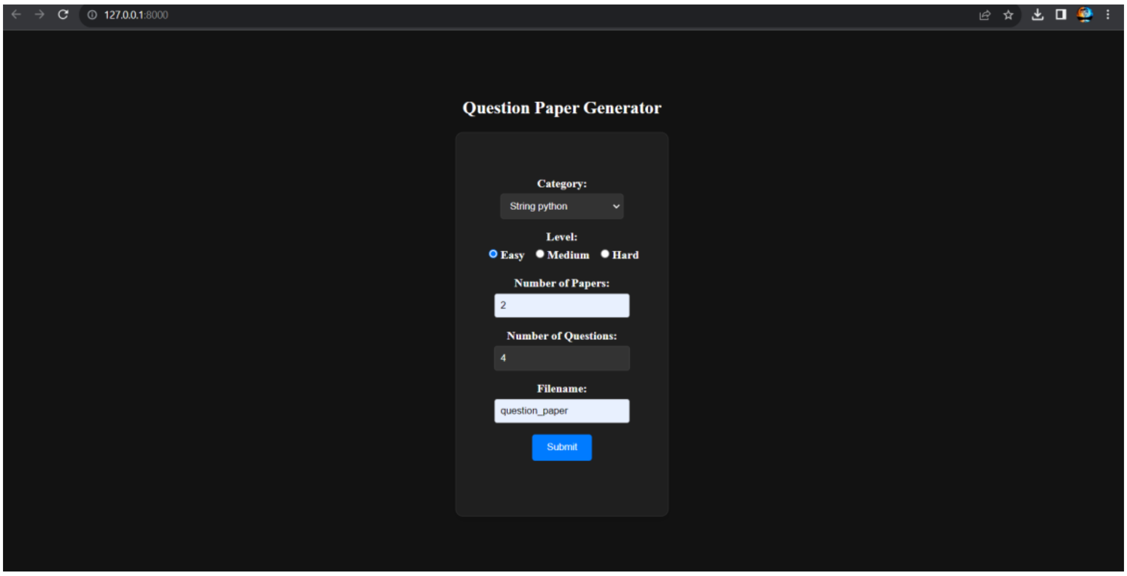
B. ue

C. luen

D. Error

ANS. Ue

SCREENSHOTS:



127.0.0.1:8000

Question Paper Generator

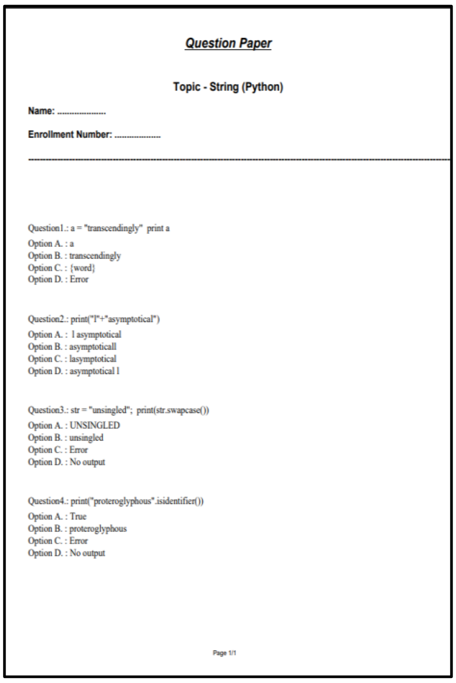
Category: String python

Level: ● Easy ○ Medium ○ Hard

Number of Papers: 2

Number of Questions: 4

Filename: question_paper



Question Paper

Topic - String (Python)

Name: ....

Enrollment Number: ....

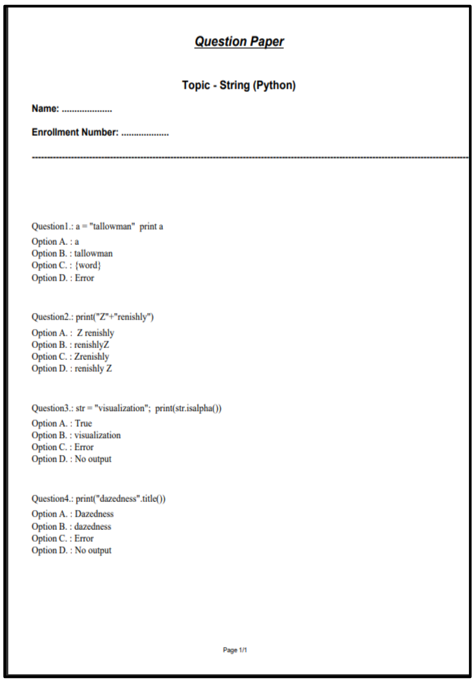
Question1.: a = "transcendingly" print a
Option A. : a
Option B. : transcendingly
Option C. : {word}
Option D. : Error

Question2.: print("l"+"asymptotical")
Option A. : l asymptotical
Option B. : asymptoticall
Option C. : lasymptotical
Option D. : asymptotical l

Question3.: str = "unsingled"; print(str.swapcase())
Option A. : UNSINGLED
Option B. : unsingled
Option C. : Error
Option D. : No output

Question4.: print("proteroglyphous".isidentifier())
Option A. : True
Option B. : proteroglyphous
Option C. : Error
Option D. : No output

Page 1/1



Question Paper

Topic - String (Python)

Name: ....

Enrollment Number: ....

Question1.: a = "tallowman" print a
Option A. : a
Option B. : tallowman
Option C. : {word}
Option D. : Error

Question2.: print("Z"+"renishly")
Option A. : Z renishly
Option B. : renishlyZ
Option C. : Zrenishly
Option D. : renishly Z

Question3.: str = "visualization"; print(str.isalpha())
Option A. : True
Option B. : visualization
Option C. : Error
Option D. : No output

Question4.: print("dazedness".title())
Option A. : Dazedness
Option B. : dazedness
Option C. : Error
Option D. : No output

Page 1/1

Regular expressions have been used to create an Automatic Question Paper Generator, which is a significant development in educational technology. This system automates the selection and layout of questions based on established patterns, streamlining the process of creating test questions. In addition to saving educators important time, this innovation makes guarantee that questions are chosen consistently and fairly. Regex expressions are used to produce question papers, showcasing the potential of technology to improve educational processes and giving teachers a strong tool to design individualized and thorough examinations for pupils.

REFERENCES:

- https://www.w3schools.com/python/python_regex.asp
- https://www.youtube.com/watch?v=q70xzDG6nls
- https://www.geeksforgeeks.org/flask-tutorial/

Research papers:

- Kapil Naik, Shreyas Sule, Shruti Jadhav, Surya Pandey (2014): Automatic Question Paper Generation System using Randomization Algorithm, International Journal of Engineering and Technical Research (IJETR) ISSN: 2321-0869, Volume-2, Issue-12
- Shubham Aaghade,Manan Parikh,Adwait Gudekar, Shweta Sharma(2021): Automatic question paper generator.International Research Journal of Engineering and Technology (IRJET)
- Amruta Umardand,Ashwini Gaikwad(2017):A Survey on Automatic Question Paper Generation System. International Advanced Research Journal in Science, Engineering and Technology.