QuizBuilder

Yashika Tirkey

Abstract

Project simplifies the creation of educational questions from selected PDFs. Users can choose a PDF, and our tool automatically generates fill-in-the-blank, true or false, and question-and-answer questions, saving time and enhancing the learning experience for students and educators.

Problem Statement

"Insufficient Availability of Educational Practice Questions"

Description: In the realm of education, one of the key challenges faced by students and educators is the inadequate availability of practice questions and problems to support effective learning. This problem encompasses various educational levels, from primary to higher education, and can impact a wide range of subjects and disciplines.

Challenges:

- Limited Learning Opportunities: Inadequate access to practice questions hinders students' ability to reinforce their understanding of concepts and improve their problem-solving skills.
- Skill Development: Practicing questions is essential for skill development and academic success. When students lack a variety of questions to work on, their proficiency and confidence may suffer.
- Diverse Learning Needs: Students have diverse learning needs and paces. Some may need more questions to grasp a concept, while others require different types of questions to challenge themselves.
- Teacher Workload: Educators and teachers often spend a significant amount of time creating or searching for quality practice questions to provide to their students, taking time away from other teaching activities.
- Impact: The shortage of practice questions can result in reduced educational effectiveness, lower student performance, and increased stress and frustration among students and educators.

Solution: Developing a solution that addresses this problem might involve creating a platform or tool for generating, curating, or sharing a diverse range of practice questions across subjects and grade levels. This platform could serve both students and educators, offering a comprehensive collection of questions and customizable features to cater to individual learning needs.

By addressing this problem, students and educators can access a wider variety of practice questions, leading to more effective learning, improved academic outcomes, and reduced teaching workload.

Market/Customer/Business Need Assessment

It's common for education systems to offer courses and test series, but the availability and affordability of these resources can vary. Here's an overview:

Course Availability

Many educational platforms provide a wide range of courses, covering various subjects and skill levels. These courses can be designed for different purposes, from academic learning to professional development.

Challenges

- Limited Test Series: While courses are abundant, test series may be more limited in number. Not all courses come with associated test series, and not all subjects or topics may have dedicated test materials.
- Affordability: Test series, when available, may come at an additional cost. Some students may find these costs to be a barrier, especially if they are already paying for the courses.
- Accessibility: The accessibility of test series can vary based on the platform's pricing structure. Some platforms may offer bundled access to both courses and test series, while others might charge separately for each.

Impact

These limitations can affect the completeness of a student's educational experience. The lack of affordable test series can hinder students' ability to practice and evaluate their understanding, which is crucial for effective learning and exam preparation.

Addressing the Issue

To address these challenges, education platforms can consider the following:

- Affordability: Offering test series at lower costs or providing affordable subscription models can make them accessible to a wider range of students.
- Diverse Test Series: Expanding the range of available test series to cover a broader spectrum of subjects and topics can help cater to the diverse needs of students.
- Integration: Integrating test series as part of the course package or offering them as bundled options can provide a more comprehensive learning experience.
- Inclusivity: Providing options for free or discounted test series for students with limited financial means can make education more inclusive.

In conclusion, the availability and affordability of test series play a significant role in students' learning and exam preparation. Addressing these challenges can help educational platforms enhance the overall learning experience and support students in achieving their educational goals.

Target Specifications

QuizBuilder, offers a unique and empowering solution for both students and teachers in the realm of education.

- Customized Learning: QuizBuilder allows students and teachers to create questions and answers based on their own choice of notes or PDFs.
  This customization empowers learners to tailor their learning experiences to match their individual preferences and needs.

- Affordability: Unlike some education apps that charge high fees for courses and content, QuizBuilder utilizes existing materials, making it an affordable and accessible resource for students.

- Content Relevance: By generating questions from user-provided notes or PDFs, QuizBuilder ensures that the content is up-to-date and directly relevant to the materials students are studying.

- Active Learning: Learners actively engage with the content by creating questions and answers. This active learning approach enhances comprehension, retention, and application of knowledge.

- Flexibility: QuizBuilder offers flexibility in question types, allowing users to choose between multiple formats, including Question and Answer or True or False, accommodating various learning preferences.

- Enhanced Focus and Practice: Users can focus on areas they find most challenging or relevant, thereby optimizing their learning and practice efforts.

- Empowerment and Enjoyment: The ability to generate personalized questions not only enhances the learning experience but also makes it enjoyable and fulfilling.

Success-Oriented Learning: The self-directed approach encourages students to take ownership of their education, fostering a mindset that supports continuous learning and achievement.

QuizBuilder's approach aligns with contemporary educational principles that emphasize personalized learning, active engagement, and adaptability. By providing an accessible and affordable way for students to customize their learning experience, your project contributes to the enhancement of education, enabling learners to excel in their studies.

External Search (online information sources/references/links)

PKE API – https://github.com/boudinfl/pke

pke is an open source python-based keyphrase extraction toolkit. It provides an end-to-end keyphrase extraction pipeline in which each component can be easily modified or extended to develop new models. pke also allows for easy benchmarking of state-of-the-art keyphrase extraction models, and ships with supervised models trained on the SemEval-2010 dataset.

Transformer - https://huggingface.co/docs/transformers/index

Transformer architecture is known for its ability to model dependencies in sequences, such as sentences or words in a text, by utilizing a mechanism called "self-attention." This self-attention mechanism allows the model to weigh the importance of different elements in the input sequence when making predictions for a particular element. This makes it especially powerful for tasks like language translation, sentiment analysis, text generation, and more.

Hugging Face, a popular NLP library and platform, provides pre-trained Transformer-based models that can be fine-tuned for various NLP tasks, making it easier for developers and researchers to leverage the power of the Transformer architecture for natural language understanding and generation. Hugging Face's Transformers library is widely used for tasks like text classification, language generation, and text summarization, among others. It offers a wide range of pre-trained models and tools to work with Transformer-based architectures effectively.

Used T5, BERT Model.

Books

- Natural Language Processing in Action" by Lane, Howard, and Hapke
- Applied Natural Language Processing with Python" by Benjamin Bengfort and Rebecca
- Applied Natural Language Processing with Python" by Benjamin Bengfort and Rebecca

Kaggle NLP tutorial

Google and Chatgpt

Udemy Courses

- https://www.udemy.com/course/question-generation-using-natural-language-processing
- https://www.udemy.com/course/modern-natural-language-processingnlp-using-deep-learning/
- https://www.udemy.com/course/data-science-transformers-nlp/

Youtube Videos

- https://www.youtube.com/watch?v=SMZQrJ_L1vo&pp=ygUUdHJhbnNmb3JtZXIgdHV0b3JpYWw%3D

- https://www.youtube.com/watch?v=acxqoltilME&pp=ygUUdHJhbnNmb3JtZXIgdHV0b3JpYWw%3D
- https://www.youtube.com/watch?v=wxnTj-7yTt8&list=PLMfAHlb5o1xFXFiuBnpv4PZ_4GiQ3O70I
- https://www.youtube.com/watch?v=dvf9xuK5g5A&pp=ygULUHJldHR5dGFibGU%3D

Bench marking alternate products

Existing Products:

- Often provide fixed content, limiting the ability to adapt to individual learning preferences.
- Some education apps charge high fees for access to courses and question sets, making them less accessible to budget-conscious learners.
- Occasionally provide outdated content, which can be a hindrance to effective learning.
- Some Education app offer generic content that may not align perfectly with an individual's learning goals.
- Often apply a one-size-fits-all approach, which may not suit every individual's needs.

QuizBuilder:

- Offers unparalleled customization, allowing users to create questions based on their own notes or PDFs. This flexibility empowers learners to tailor their learning experience to their specific needs.
- Prioritizes affordability by allowing users to generate questions from their existing materials, eliminating additional costs for predefined courses.
- Encourages active learning by enabling users to generate their own questions and answers, fostering a sense of ownership over the learning process.
- Supports learners in focusing on the areas that matter most to them, as they can select the specific content they want to transform into questions.
- Promotes success-oriented learning by giving users the tools to create questions and answers that support their unique learning journeys.
- Users are empowered to create their own path to success, making QuizBuilder a valuable addition to the world of education technology.

Applicable Patents (Patent of Tech/Software/Framework etc you are going to use in your Product/Service idea)

QuizBuilder is a project or concept that allows users to generate questions and answers from PDF documents. While it's not a specific education app or platform, you can potentially integrate or implement QuizBuilder into various existing education apps or use it as a standalone tool. Here are some scenarios in which QuizBuilder could be used:

- Custom Learning Platforms: Customized learning platforms, particularly those used in schools and universities, can integrate QuizBuilder to enhance the assessment and interactive features of their systems.

- Educational Apps: Standalone educational apps, especially those focused on personalized learning, can include QuizBuilder functionality to allow users to generate their own quizzes.

Applicable Regulations

QuizBuilder is used by individuals of all ages and user privacy information needs to be protected, it's essential to implement robust security measures. Here are some technologies and strategies to safeguard user information:

- Education is universal, and QuizBuilder is designed for users everywhere, without any geographical limitations. It's accessible to anyone, regardless of their location.

- Robust Encryption: User data is secured through strong encryption protocols. This means that all data transferred to and from QuizBuilder is shielded using HTTPS, ensuring that sensitive information remains confidential during transmission.

- Data Minimization: We practice data minimization, meaning that we only collect and retain the minimum amount of user information required for the proper functioning of QuizBuilder. Superfluous data is avoided to reduce any potential privacy risks.

- Secure User Authentication: We mandate secure user authentication methods, including two-factor authentication (2FA), to verify user identities and prevent unauthorized access to accounts and private data.

Applicable Constraints

Budgets:

- Server Memory: amount of memory required on server depends on the server's capacity and the number of concurrent users.
- User information: database management system
- Server Cost: Cloud hosting costs can vary based on the chosen provider (e.g., AWS, Azure, Google Cloud) and the server configuration (CPU, RAM, storage).
- Security: Investing in security measures, such as firewalls, DDoS protection, and security audits, is crucial and should be factored into the budget.
- Coding: Cost for Algorithms to speed up the process.

Business Model

- API Access: A premium API for integration with other educational apps and services.
- Advertising and Sponsorships: Collaborate with relevant educational companies for sponsorships and partnerships to promote their products or services.

- Pay-Per-Use Model: Allow users to generate a limited number of questions for free. Implement a pay-per-use model where users can purchase question generation credits or tokens. Users pay only for the number of questions they generate
- Feedback: Regularly gather feedback from your user base to refine and enhance QuizBuilder.

Concept Generation

During my time in school and college, I often faced a common problem: I had a lot of study material to go through, but not enough practice questions or affordable test series to help me prepare. On top of that, when I learned something from a page, creating questions from that material was a challenge.

That's when I had an idea: What if there was a tool that could generate questions based on the text I was studying? This tool would make it easier for me to practice and solidify my understanding of a subject or topic efficiently. This is how the concept of QuizBuilder was born.

QuizBuilder aims to provide a solution to these common educational hurdles by enabling users to generate questions from their study material. It's all about making learning more accessible and effective, helping students like me succeed in their studies without the added burden of expensive test series or the frustration of not having enough practice questions. With QuizBuilder, learning becomes more affordable and tailored to individual needs.

Brief summary of Product

<u>Project Overview</u>:

The project is designed to assist both students and teachers in generating questions and answers from PDF documents. It simplifies the process of creating quizzes by allowing users to upload a PDF file and customize the types of questions they want. The key features are as follows:

<u>User-Friendly Process:</u>

- PDF Upload: Users start by uploading a PDF document containing the study material or content they want to create questions from.

- Question Type Selection: After uploading the PDF, the system presents users with a selection of question types to choose from. These options include Multiple Choice Questions (MCQs), True or False questions, and open-ended Questions and Answers.

Customized Question Generation:

- Based on the user's selection of question type, the system processes the PDF document to generate questions accordingly.
- If the user chooses MCQs, the system will extract content and create multiple-choice questions with answer choices.
- If the user prefers True or False questions, the system will formulate statements and generate True or False questions.

Flexibility and Ease of Use:

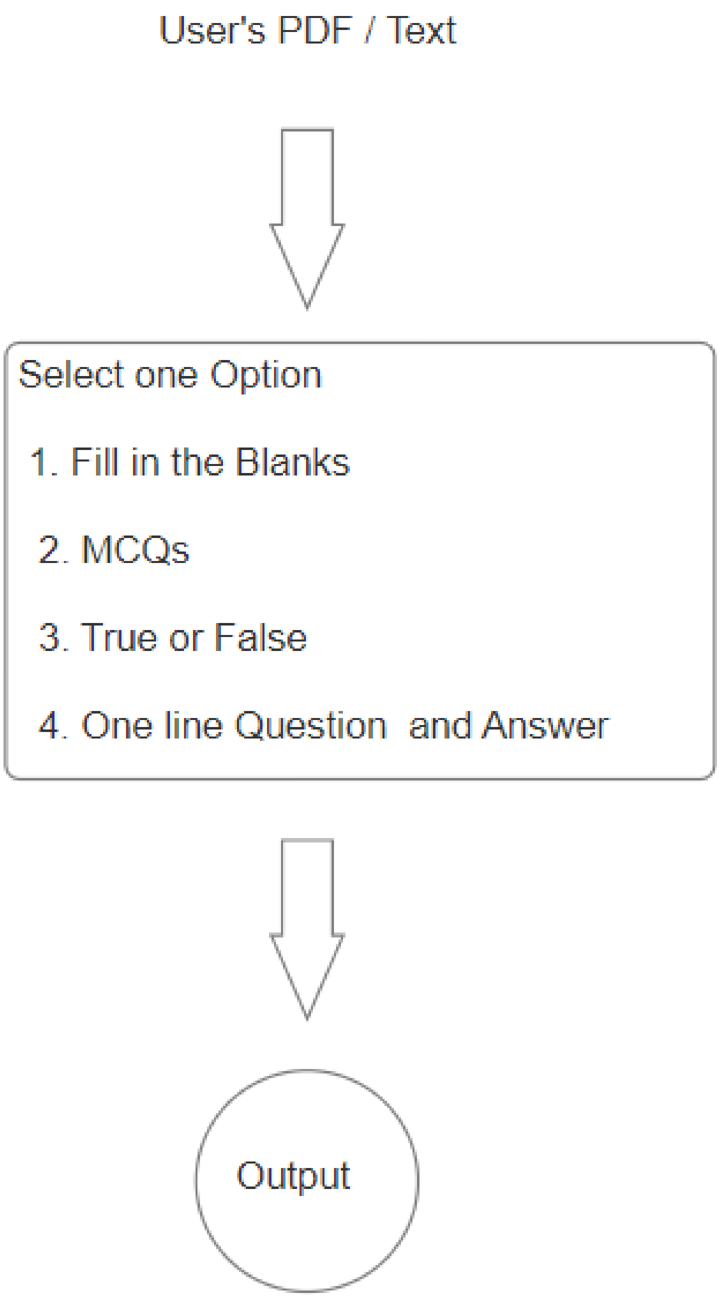
This project offers flexibility, allowing users to generate questions that best suit their learning or teaching needs. It simplifies the process of creating quizzes and assessments, saving users time and effort.

Benefits:

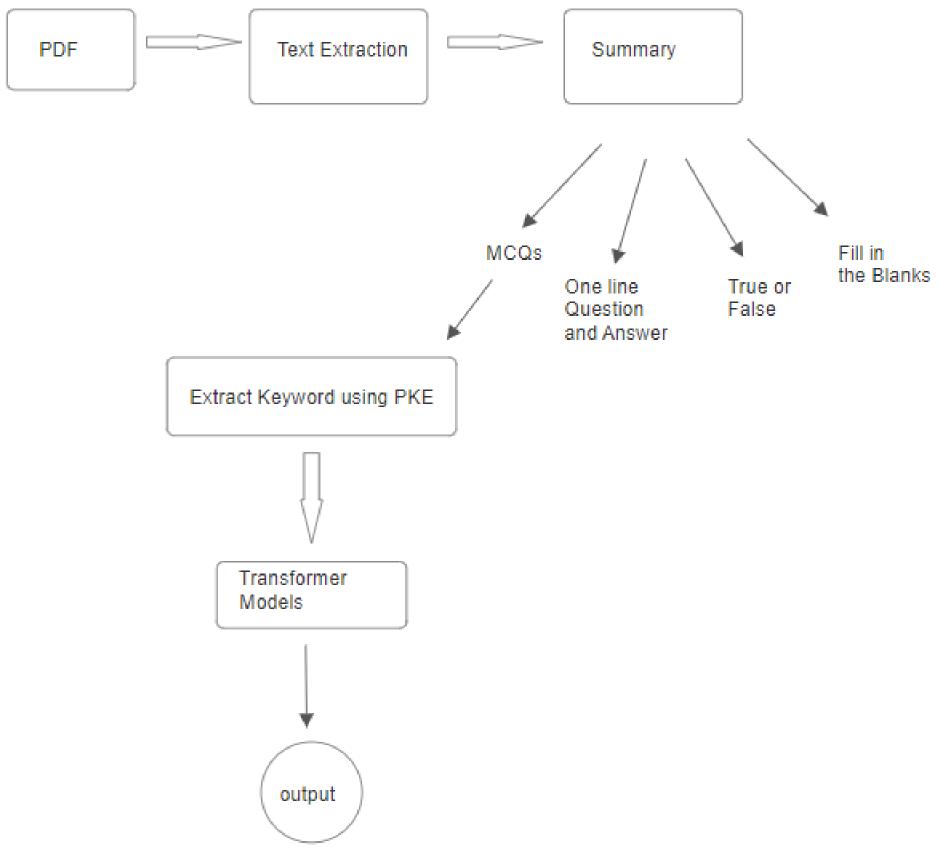
- Students can create practice quizzes from their study materials, enhancing their understanding and retention of the subject matter.
- Teachers can efficiently generate assessments, saving time on question creation, and ensuring a variety of question types in their tests.
- Overall, this project streamlines the process of generating questions and answers from PDF documents, making it a valuable resource for both students and teachers in the educational context. It offers a user-friendly and customizable approach to question generation.

Schematic Diagram

Overview of QuizBuilder Project



QuizBuilder



QuizBuilder uses transformer models like T5 and BERT, as well as sense2vec:

<u>T5 (Text-to-Text Transfer Transformer):</u>

T5 is a transformer-based model developed by Google AI that is specifically designed for a wide range of natural language processing tasks. It excels at text generation, translation, summarization, and more. Here's how QuizBuilder leverages T5:

- Input Processing: When a user uploads a PDF document or text, QuizBuilder preprocesses and sends the text as input to the T5 model.

- Text-to-Text Conversion: T5 converts this input into a text-to-text format, which means that it formulates both the input and the desired output as text. For example, if the user wants to generate multiple-choice questions (MCQs), QuizBuilder transforms this requirement into a text-to-text format.

- Question Generation: T5 generates questions based on the text-to-text format. If the user selected the "MCQ" option, T5 creates multiple-choice questions from the provided content.

- Variety of Tasks: T5's versatility allows QuizBuilder to handle different types of questions, including MCQs, true or false questions, and open-ended questions, providing a comprehensive question generation solution.

2. BERT (Bidirectional Encoder Representations from Transformers):

BERT, also developed by Google, is known for its bidirectional contextual understanding of language. QuizBuilder uses BERT for various purposes:

- Contextual Understanding: When a user uploads content, BERT helps QuizBuilder understand the context and content of the text, which is crucial for generating contextually relevant questions.
- Contextual Question Generation: BERT's contextual understanding enables QuizBuilder to generate questions that are contextually accurate, ensuring that they make sense within the provided content.
- Contextual Variations: BERT's ability to capture contextual variations in language guarantees that the generated questions and answers are contextually precise and relevant to the content.

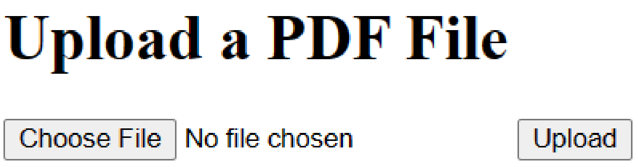
3. sense2vec:

sense2vec is a distributional semantic model that helps understand word meanings and relationships based on their usage in context. In QuizBuilder, sense2vec can be beneficial in several ways:

- Word Embeddings: sense2vec provides word embeddings or vectors that represent the meanings and usage of words. These embeddings are useful for understanding the context of words and phrases in the uploaded text.
- Contextual Relevance: sense2vec assists in ensuring that questions and answers generated by QuizBuilder are contextually relevant and meaningful within the given content.
- Enhancing Question Quality: By using sense2vec, QuizBuilder can improve the quality of generated questions by considering the nuanced meaning of words and phrases in the context of the text.

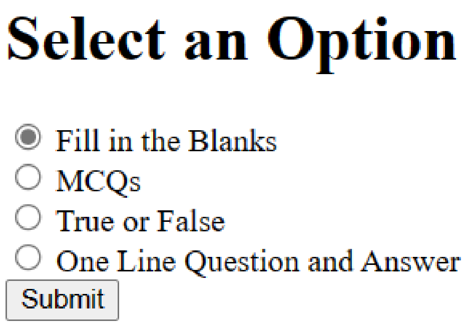
In summary, QuizBuilder leverages T5 and BERT for text-to-text conversion, question generation, answer validation, and contextual understanding. sense2vec contributes to enhancing the contextual relevance and quality of the questions and answers generated by considering the nuances of word usage in the provided text. These advanced natural language processing models enhance the educational value and effectiveness of QuizBuilder's question generation capabilities.

Steps-1


Upload a PDF File

Choose File No file chosen Upload

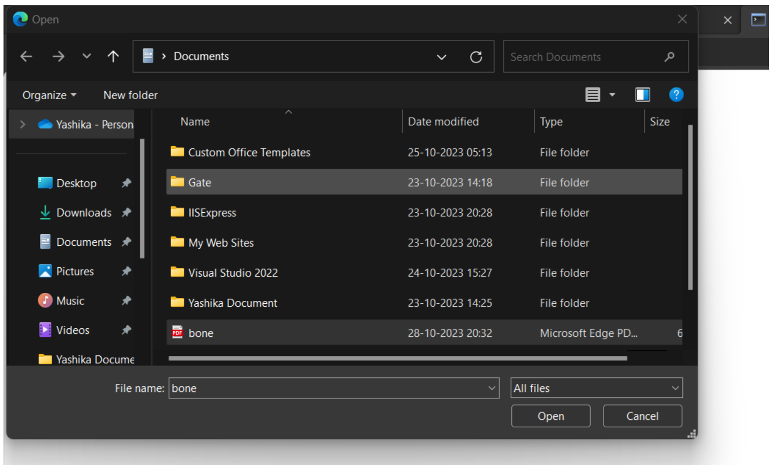
Steps -2


Select an Option

- [✓ Fill in the Blanks]
- MCQs
- True or False
- One Line Question and Answer

Submit

Step -3



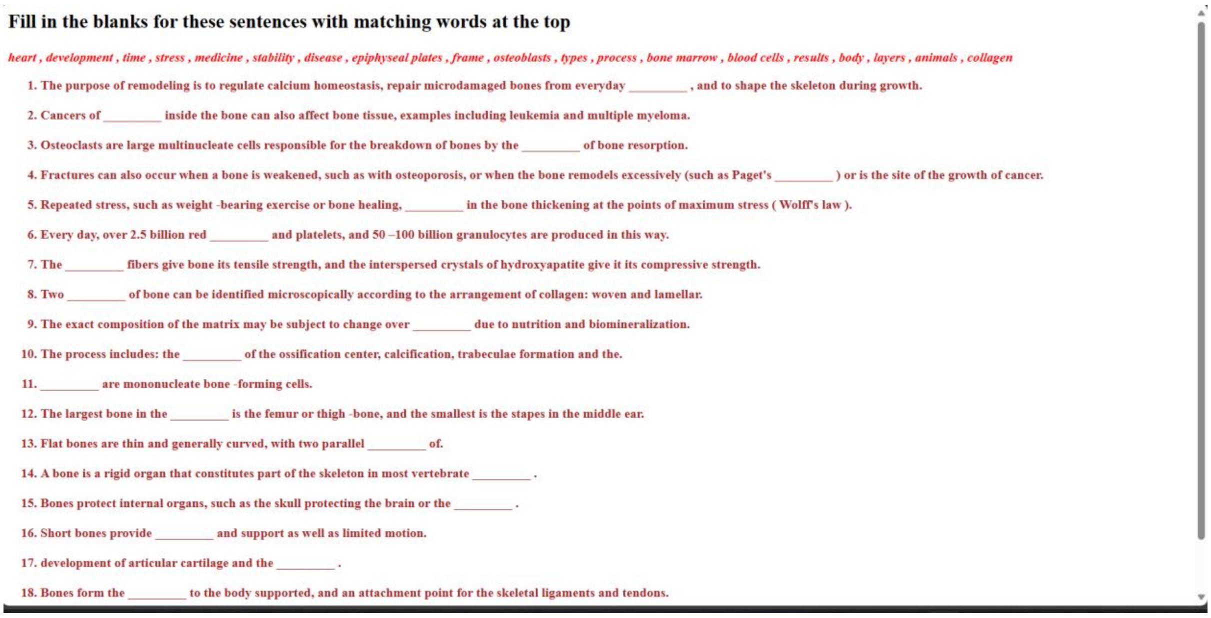
Name	Date modified	Type	Size
Custom Office Templates	25-10-2023 05:13	File folder	
Gate	23-10-2023 14:18	File folder	
IISExpress	23-10-2023 20:28	File folder	
My Web Sites	23-10-2023 20:28	File folder	
Visual Studio 2022	24-10-2023 15:27	File folder	
Yashika Document	23-10-2023 14:25	File folder	
bone	28-10-2023 20:32	Microsoft Edge PD...	6

File name: bone

All files

Open Cancel

Output



```
Fill in the blanks for these sentences with matching words at the top

heart , development , time , stress , medicine , stability , disease , epiphyseal plates , frame , osteoblasts , types , process , bone marrow , blood cells , results , body , layers , animals , collagen

 1. The purpose of remodeling is to regulate calcium homeostasis, repair microdamaged bones from everyday _________ , and to shape the skeleton during growth.
 2. Cancers of _________ inside the bone can also affect bone tissue, examples including leukemia and multiple myeloma.
 3. Osteoclasts are large multinucleate cells responsible for the breakdown of bones by the _________ of bone resorption.
 4. Fractures can also occur when a bone is weakened, such as with osteoporosis, or when the bone remodels excessively (such as Paget's _________ ) or is the site of the growth of cancer.
 5. Repeated stress, such as weight -bearing exercise or bone healing, _________ in the bone thickening at the points of maximum stress ( Wolff's law ).
 6. Every day, over 2.5 billion red _________ and platelets, and 50 –100 billion granulocytes are produced in this way.
 7. The _________ fibers give bone its tensile strength, and the interspersed crystals of hydroxyapatite give it its compressive strength.
 8. Two _________ of bone can be identified microscopically according to the arrangement of collagen: woven and lamellar.
 9. The exact composition of the matrix may be subject to change over _________ due to nutrition and biomineralization.
10. The process includes: the _________ of the ossification center, calcification, trabeculae formation and the.
11. _________ are mononucleate bone -forming cells.
12. The largest bone in the _________ is the femur or thigh -bone, and the smallest is the stapes in the middle ear.
13. Flat bones are thin and generally curved, with two parallel _________ of.
14. A bone is a rigid organ that constitutes part of the skeleton in most vertebrate _________ .
15. Bones protect internal organs, such as the skull protecting the brain or the _________ .
16. Short bones provide _________ and support as well as limited motion.
17. development of articular cartilage and the _________ .
18. Bones form the _________ to the body supported, and an attachment point for the skeletal ligaments and tendons.
```

Code for Fill in the Blanks

```python
import traceback

import xml.etree.ElementTree as et
import random
import re
import pke


from xml.dom import minidom
from flashtext import KeywordProcessor
from pprint import pprint

def Fill_blanks_question(summary):
    out = []
    try:
        extractor = pke.unsupervised.MultipartiteRank()
        extractor.load_document(input=summary,language='en')
        pos = {'NOUN'}
        extractor.candiate_selection(pos=pos)
        extractor.candiate_weighting(alpha=1.1,
                                     threshold=0.01,
                                     method='average')
        keyphrases = extractor.get_n_best(n=50)
        for val in keyphrases:
            out.append(val)
    except:
        out = []
        traceback.print_exc()
        print(out)

    return out
```

```
def get_noun(text):
    out=[]
    try:
        extractor = pke.unsupervised.MultipartiteRank()
        extractor.load_document(input=text,language='en')
        #    not contain punctuation marks or stopwords as candidates.
        pos = {'NOUN'}
        extractor.candidate_selection(pos=pos)
        # 4. build the Multipartite graph and rank candidates using random
walk,
        #    alpha controls the weight adjustment mechanism, see TopicRank
for
        #    threshold/method parameters.
        extractor.candidate_weighting(alpha=1.1,
                                      threshold=0.01,
                                      method='average')
        keyphrases = extractor.get_n_best(n=50)


        for val in keyphrases:
            out.append(val[0])
    except:
        out = []
        traceback.print_exc()

    return out


def get_sentences_for_keyword(keywords, sentences):
    keyword_processor = KeywordProcessor()
    keyword_sentences = {}
    for word in keywords:
        keyword_sentences[word] = []
        keyword_processor.add_keyword(word)
    for sentence in sentences:
        keywords_found = keyword_processor.extract_keywords(sentence)
        for key in keywords_found:
            keyword_sentences[key].append(sentence)

    for key in keyword_sentences.keys():
        values = keyword_sentences[key]
        values = sorted(values, key=len, reverse=True)
        keyword_sentences[key] = values
    return keyword_sentences



def get_fill_in_the_blanks(sentence_mapping):
    out={"title":"Fill in the blanks for these sentences with matching
words at the top"}
    blank_sentences = []
    processed = []
    keys=[]
    for key in sentence_mapping:
        if len(sentence_mapping[key])>0:
            sent = sentence_mapping[key][0]
            # Compile a regular expression pattern into a regular
expression object, which can be used for matching and other methods
            insensitive_sent = re.compile(re.escape(key), re.IGNORECASE)
            no_of_replacements =
len(re.findall(re.escape(key),sent,re.IGNORECASE))
            line = insensitive_sent.sub(' _________ ', sent)
            if (sentence_mapping[key][0] not in processed) and
no_of_replacements<2:
                blank_sentences.append(line)
                processed.append(sentence_mapping[key][0])
                keys.append(key)
    out["sentences"]=blank_sentences[:20]
    out["keys"]=keys[:20]
    return out


def placing_elemnts_fill_blanks(fill_in_the_blanks):

    root = et.Element("div")

    heading = et.Element("h2")
    heading.text = fill_in_the_blanks['title']

    keywords = et.Element("var")
    keywords.set('style', 'color: red;')

    all_keys = fill_in_the_blanks['keys']
    random.shuffle(all_keys)

    for index, blank in enumerate(all_keys):
        child = et.Element("span")
        child.text = blank
        keywords.append(child)
    # Add a comma after each element except the last one
        if index < len(all_keys) - 1:
            comma = et.Element("span")
            comma.text = ', '
            keywords.append(comma)



    sentences = et.Element("ol")
    sentences.set('style', 'color:brown;')
    for sentence in fill_in_the_blanks['sentences']:
        child=et.Element("li")
        child.text = sentence
        sentences.append(child)
        sentences.append(et.Element("br"))

    heading_content = et.Element("h4")

    root.append(heading)
    heading_content.append(keywords)
    heading_content.append(sentences)
    root.append(heading_content)

    xmlstr = et.tostring(root)
    xmlstr = xmlstr.decode("utf-8")
    prettyxmlstr =
minidom.parseString(et.tostring(root)).toprettyxml(indent="   ")
    return prettyxmlstr
```

Github link - https://github.com/yashikart/QuizBuilder