Data Science Challenge Library Report

Problem Statement

The Data Science Challenge Library project endeavors to fulfill a crucial need within the realm of education: the absence of a comprehensive platform that caters specifically to students in the data science field. This platform aims to create an environment where students can not only practice and refine their skills but also engage deeply in their learning journey through a carefully curated collection of challenges. These challenges span a wide spectrum, encompassing essential concepts in Artificial Intelligence (AI), Machine Learning (ML), data visualizations, data analytics, and real-world case studies.

In the modern landscape of education, the value of experiential learning is undeniable. However, for students navigating the complex terrain of data science, opportunities to apply theoretical knowledge to real-world scenarios can be scarce. This is where the Data Science Challenge Library steps in. By providing an extensive repository of problems that mimic industry challenges, the platform bridges the gap between theory and application. It allows students to roll up their sleeves and dive into practical problem-solving, thereby enhancing their comprehension and mastery of complex concepts.

The challenge at the core of this project is multifaceted: to create an interactive, user-centric website that seamlessly combines education and gamification. One of the cornerstones of this endeavor is the creation of a user-friendly interface that invites students to explore, select, and engage with challenges effortlessly. This interface not only facilitates easy navigation but also encourages curiosity-driven learning as students immerse themselves in a world of intriguing challenges.

However, the project's ambitions go beyond presenting challenges. It addresses a common concern in the world of online learning: the verification of student submissions. To ensure the integrity and accuracy of the learning process, the platform incorporates advanced verification mechanisms. These mechanisms include text verification, algorithm verification, and data visualization verification. Through these tools, students receive immediate feedback on their solutions, enabling them to learn from their mistakes and refine their approaches.

Additionally, the gamified elements embedded within the platform work to enhance user engagement. The introduction of leaderboards and progress tracking introduces an element of healthy competition and a sense of accomplishment. This fosters motivation among students, encouraging them to persistently explore challenges, refine their skills, and elevate their problem-solving abilities.

In essence, the Data Science Challenge Library project seeks to offer a holistic and enriching learning experience. It acknowledges the need for students to not just passively consume information but to actively immerse themselves in practical applications. By creating a platform that is accessible, engaging, and equipped with sophisticated verification mechanisms, the project empowers students to excel in the field of data science.

General Objectives

At the heart of the Data Science Challenge Library project lies the overarching objective to craft a dynamic and robust website that serves as an immersive platform for students to both practice and elevate their skills across the multifaceted domains of Artificial Intelligence (AI) and Machine Learning (ML), data visualizations, and data analytics. By integrating hands-on coding challenges, advanced verification mechanisms, and user-centric gamification elements, the primary aim is to provide a transformative learning experience.

Specific Objectives

1. Building a Comprehensive Problem Repository:

A pivotal stride towards achieving the project's goal is the creation of an extensive and varied repository of problem sets. These problem sets will span the entire spectrum of AI/ML, data visualizations, data analytics, and real-world case studies. This repository will serve as a reservoir of practical challenges that empower students to delve deep into these subjects and gain practical insights, bridging the gap between theoretical knowledge and practical application.

2. Crafting an Intuitive User Interface:

The second specific objective centers around designing and developing a user-friendly interface that becomes a seamless gateway for students to embark on their learning journey. The interface should facilitate easy navigation, enabling students to effortlessly discover, select, and undertake challenges that pique their interest. This user-centric design ensures that the platform becomes a space where curiosity thrives, and learning becomes an engaging and fluid process.

3. Implementing Robust Verification Mechanisms:

A cornerstone of the project is the implementation of advanced verification mechanisms. These mechanisms encompass text verification, algorithm verification, data visualization verification, and plagiarism detection. Their purpose is not only to evaluate the correctness of students' solutions but also to provide constructive feedback that enables students to comprehend their mistakes, learn from them, and refine their coding practices.

4. Infusing Gamification Elements:

The fourth objective seeks to infuse the platform with gamification elements, adding a layer of motivation and engagement. By introducing a dynamic leaderboard and progress tracking

features, the platform becomes more than just a learning space; it evolves into a virtual arena where students can challenge themselves and each other, fostering healthy competition and driving continuous improvement.

5. Continuously Improving Through User Feedback:

Finally, the project aims to create a symbiotic relationship with its users. By actively encouraging users to provide feedback, the platform can undergo iterative enhancements. User suggestions and insights will be pivotal in refining the user experience, rectifying any potential pain points, and ensuring that the platform's evolution remains aligned with the needs and aspirations of its user base.

Functionalities of the Data Science Challenge Library Website

User-Friendly Website Layout and Interface:

Create an inviting and easy-to-navigate environment for users. A well-designed layout enhances user experience, encourages engagement, and communicates professionalism.

User Registration and Login Functionality:

Enable users to create accounts and access personalized features. User registration fosters community engagement, while secure login ensures data privacy and easy access to their accounts.

Database for Challenge Data and User Progress:

Store and manage crucial data efficiently. The database facilitates organizing challenge content and user information, enabling accurate tracking of user progress and achievements.

Intuitive Challenge Submission Form:

Provide a seamless process for users to submit their solutions to challenges. An intuitive form streamlines user interaction, captures necessary information, and ensures accurate submission.

Solution Verification System:

Automate the process of assessing user-submitted solutions. This functionality verifies the correctness of solutions, delivers prompt feedback, and aids users in understanding their progress.

Leaderboard to Track User Performance:

Motivate user engagement and competitiveness. A leaderboard showcases user achievements, encourages active participation, and fosters a sense of accomplishment within the community.

Features of the Data Science Challenge Library Website

- **Text Verification**

Text matching mechanism to evaluate the answers submitted case study and short answer questions.

- **Algorithm Verification**

The module executes the students' code and compares the output with a predefined gold standard. This provides a reliable means of assessing the correctness and efficiency of the students' code.

- **Data Visualization Verification**

Introduced an image-matching module using OpenCV. This allows comparison and correction of students' visualizations. The module aids in evaluating the quality and correctness of data visualizations in the students' submissions.

- **Plagiarism Verification**

This could help in identifying potential plagiarism. It could also be used to identify solutions that are structurally similar but use different variables or function names, which could be useful in comparing and ranking solutions to a challenge.

- **Review Verification**

To gain valuable insights into the sentiment and opinions expressed by users in their reviews.

Methodology

a) Project Planning:

In the initial phase of the project, meticulous planning serves as the foundation for its successful execution.

Project Scope Definition: A comprehensive outline of the project's boundaries, objectives, and expected outcomes is established. This includes clarifying what functionalities and features the Data Science Challenge Library website will encompass.

Defining Objectives and Deliverables: The general and specific objectives are articulated, ensuring that the project's purpose and goals are clearly defined. The project deliverables, such as the fully functional website, verification mechanisms, and gamification features, are outlined.

Team Roles and Allocation: Each team member's role and responsibilities are clearly defined. This ensures that the project's workload is distributed optimally, utilizing each member's expertise. The Scrum Master oversees coordination and progress tracking.

Budget Estimation: A realistic budget estimation is crafted, encompassing costs related to tools, resources, and any potential expenses. This budget serves as a guiding framework throughout the project's lifecycle.

Gantt chart Creation: A detailed Gantt chart is developed, mapping out the project timeline, milestones, and dependencies. This chart aids in task scheduling, resource allocation, and monitoring progress.

b) System Analysis:

This phase focuses on understanding the project's context, audience, and key requirements.

Target Audience Identification: The specific demographics and preferences of the intended audience are meticulously studied. This includes their age, education level, technological proficiency, and learning preferences.

Learning Goals Assessment: The project team delves into the educational aspirations and objectives of the target audience. This understanding informs the creation of challenges and the integration of gamified elements that align with the audience's learning goals.

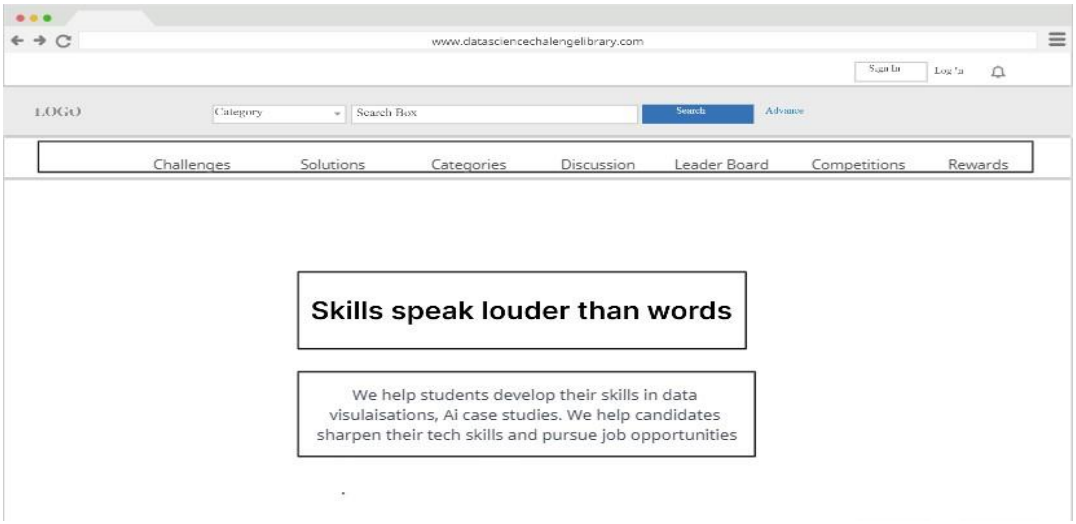
Platform Comparison: Existing platforms within the same domain are analyzed. This examination involves identifying successful features and functionalities that can be incorporated to enhance the user experience.

c) System Design:

This phase involves translating the project's conceptual vision into tangible design elements.

Wireframes and Mock-ups: The user interface's blueprint takes shape through the creation of wireframes and mock-ups. These visual representations provide an early glimpse into the platform's layout, ensuring alignment with user needs and preferences.

Database Design: The database structure is meticulously designed to efficiently store challenge data, user profiles, verification results, and other essential components. This design accounts for scalability and data integrity.



d) System Implementation:

With the groundwork laid, the project moves into the development phase.

Front-End Development: The user interface is developed using HTML, Bootstrap, CSS, and JavaScript, Angular JS. This phase focuses on creating an intuitive and visually appealing interface that resonates with the target audience.

Back-End Development: The back-end is implemented using Flask, enabling seamless communication between the front-end and the database. The back-end handles user interactions, challenge submissions, verification processes, and gamification features.

SQLite is chosen as the database management system. It efficiently stores user profiles, challenge details, and progress tracking. Its lightweight nature and smooth integration with the back-end ensure reliable data storage and retrieval.

Integration of Verification Mechanisms: Advanced verification mechanisms, including text, algorithm, and data visualization verification, are integrated into the system. These mechanisms ensure the accuracy and quality of user submissions.

Gamification Elements: Leaderboards and progress tracking features are seamlessly woven into the platform, enhancing user engagement and motivation.

e) System Testing:

Thorough testing ensures that the platform functions flawlessly and provides an optimal user experience.

Different Testing	**Explanation**
Functional Testing	Each component, including verification mechanisms and gamification elements, undergoes rigorous testing to ensure that it performs as intended.
User Interaction Testing	User interactions are simulated and analyzed to identify any usability issues, glitches, or unexpected behaviors.

f) Acceptance, Installation, and Deployment:

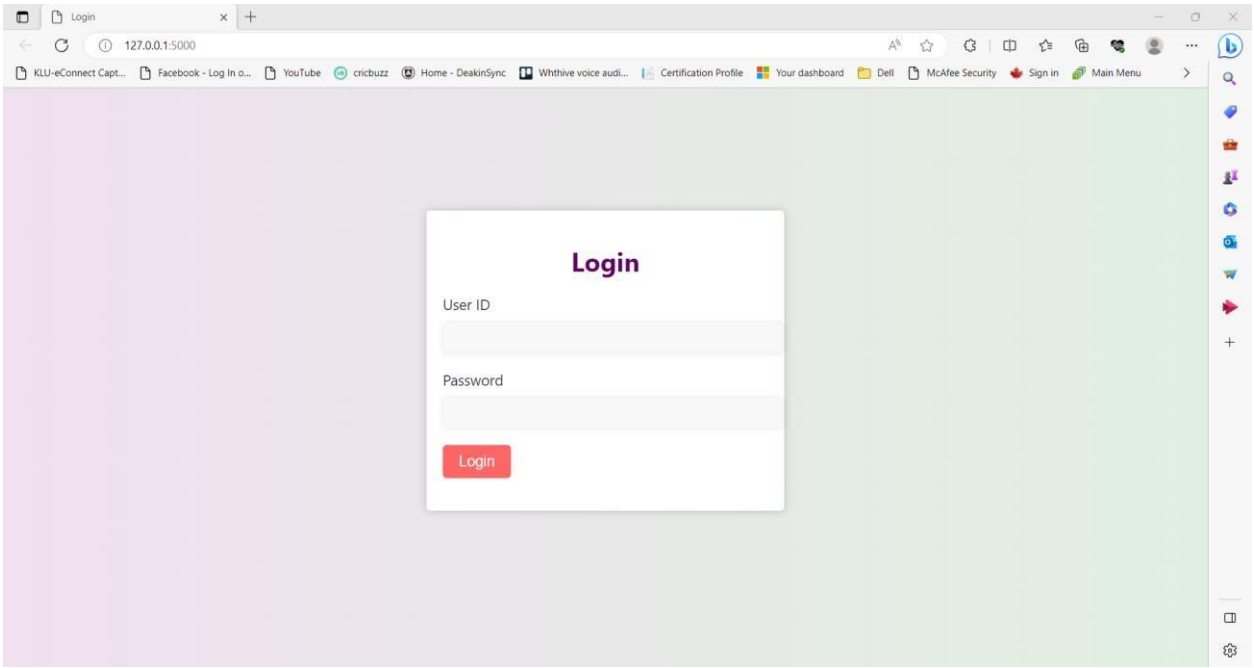
With thorough testing complete, the project moves towards deployment.

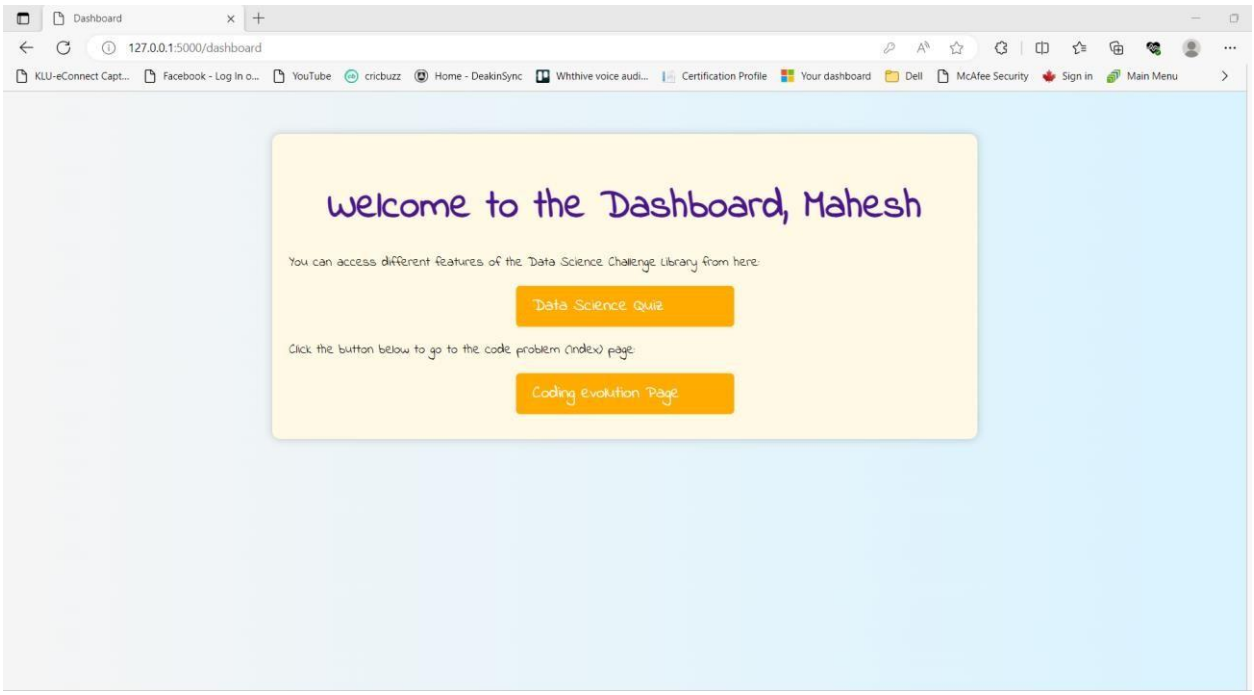
Deployment Preparation: The platform is prepared for deployment, including configuring hosting services and domain setup.

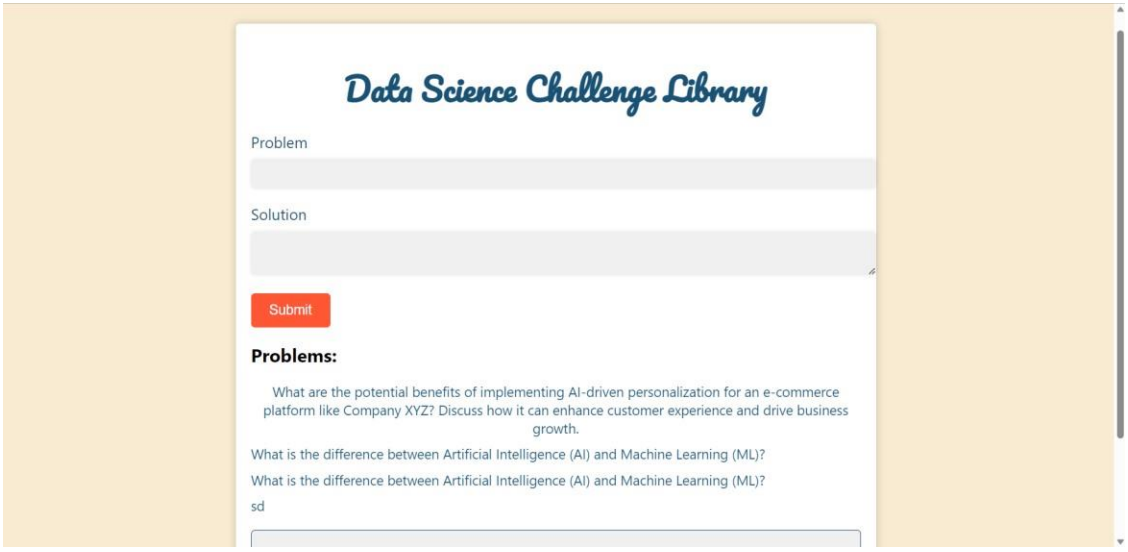
User Acceptance Testing: A final round of testing is conducted by involving actual users. Their feedback and insights are invaluable in ensuring the platform's readiness for public use.

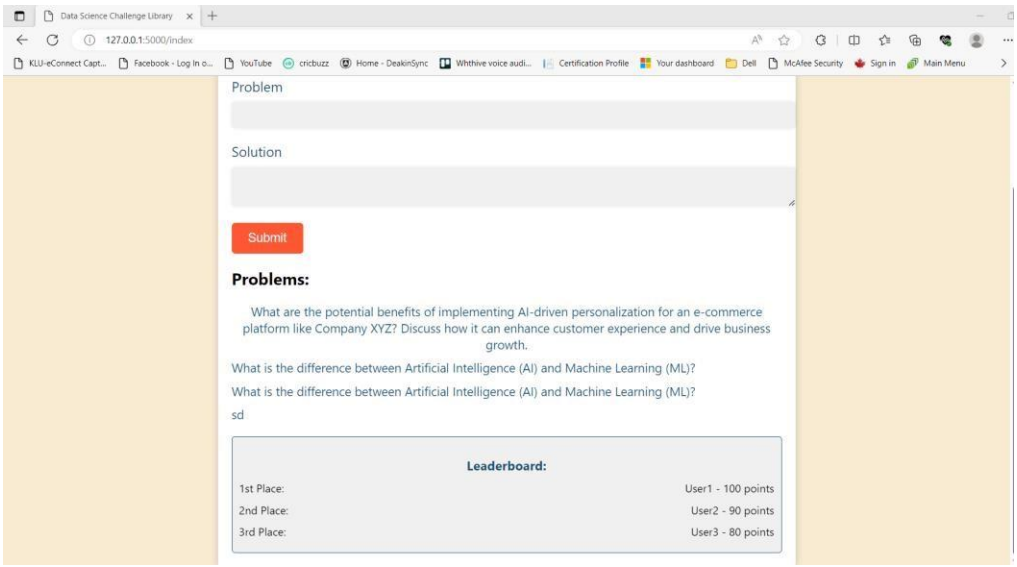
Deployment: The website is deployed on a suitable hosting platform, ensuring it is accessible to users. This phase includes transferring all components and functionalities to the live environment.

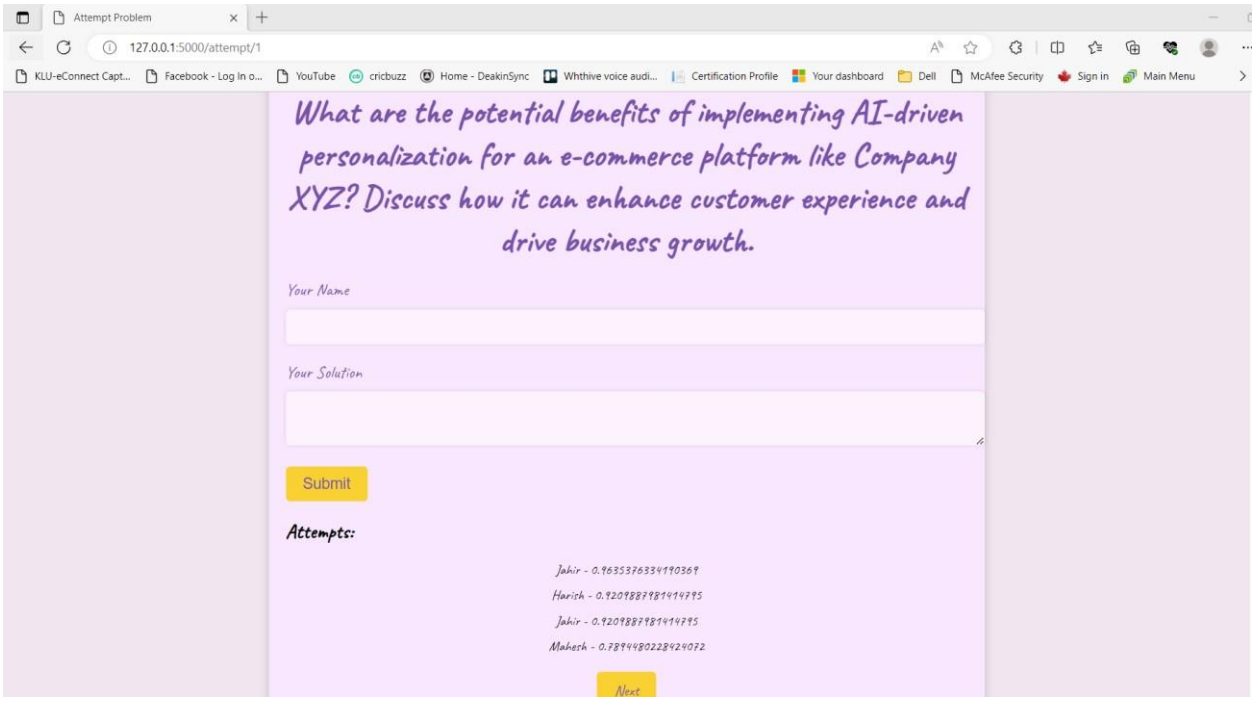
Web Pages Screenshots:

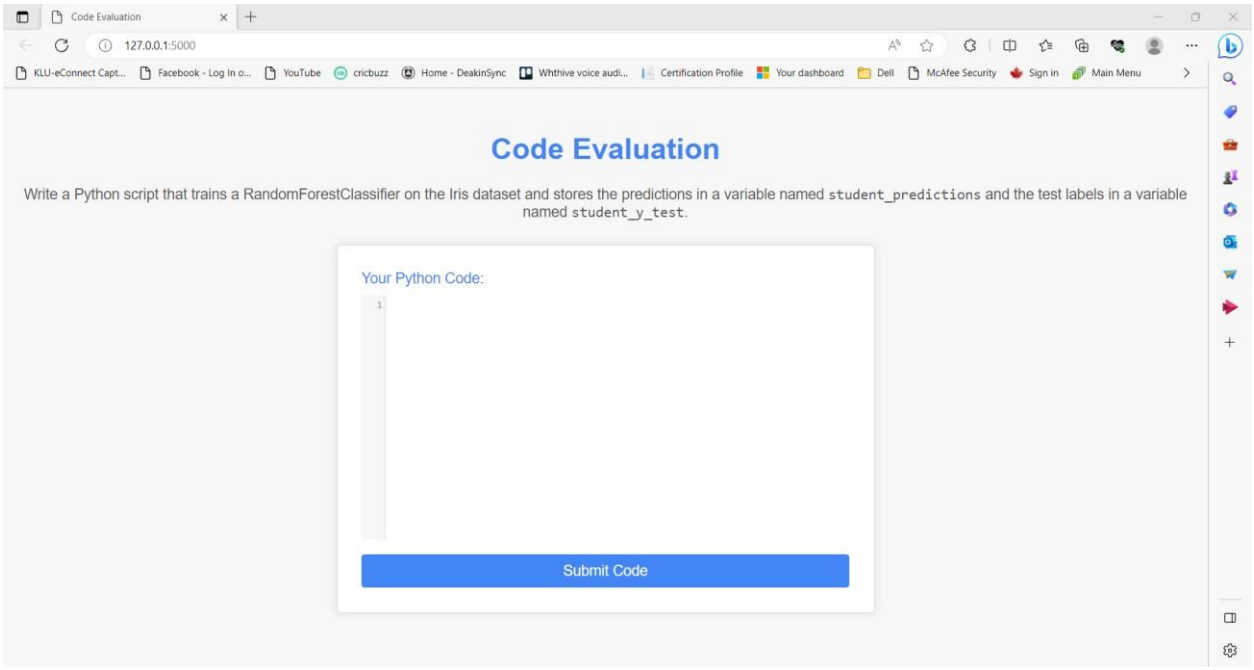


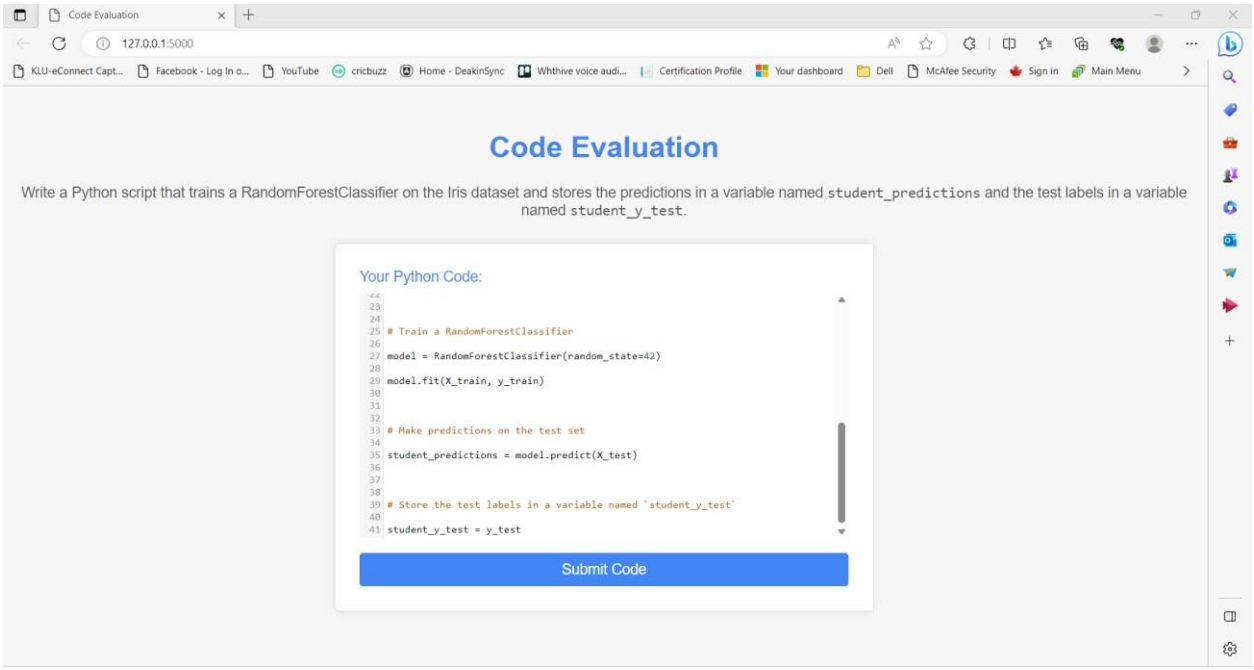

Dashboard
127.0.0.1:5000/dashboard
KLU-eConnect Capt...
Facebook - Log In o...
YouTube
cricbuzz
Home - DeakinSync
Whthive voice audi...
Certification Profile
Your dashboard
Dell
McAfee Security
Sign in
Main Menu
Welcome to the Dashboard, Mahesh
You can access different features of the Data Science Challenge Library from here.
Data Science Quiz
Click the button below to go to the code problem (index) page.
Coding evolution Page


Data Science Challenge Library
Problem
Solution
Submit
Problems:
What are the potential benefits of implementing AI-driven personalization for an e-commerce platform like Company XYZ? Discuss how it can enhance customer experience and drive business growth.
What is the difference between Artificial Intelligence (AI) and Machine Learning (ML)?
What is the difference between Artificial Intelligence (AI) and Machine Learning (ML)?
sd


Data Science Challenge Library
127.0.0.1:5000/index
Problem
Solution
Submit
Problems:
What are the potential benefits of implementing AI-driven personalization for an e-commerce platform like Company XYZ? Discuss how it can enhance customer experience and drive business growth.
What is the difference between Artificial Intelligence (AI) and Machine Learning (ML)?
What is the difference between Artificial Intelligence (AI) and Machine Learning (ML)?
sd
Leaderboard:
1st Place: User1 - 100 points
2nd Place: User2 - 90 points
3rd Place: User3 - 80 points


Attempt Problem
127.0.0.1:5000/attempt/1
What are the potential benefits of implementing AI-driven personalization for an e-commerce platform like Company XYZ? Discuss how it can enhance customer experience and drive business growth.
Your Name
Your Solution
Submit
Attempts:
Jahir - 0.9635376334190369
Harish - 0.9209888981414795
Jahir - 0.9209888981414795
Mahesh - 0.7894480228424072
Next


Code Evaluation
127.0.0.1:5000
KLU-eConnect Capt...
Facebook - Log In o...
YouTube
cricbuzz
Home - DeakinSync
Whthive voice audi...
Certification Profile
Your dashboard
Dell
McAfee Security
Sign in
Main Menu
Code Evaluation
Write a Python script that trains a RandomForestClassifier on the Iris dataset and stores the predictions in a variable named student_predictions and the test labels in a variable named student_y_test.
Your Python Code:
1
Submit Code


Code Evaluation
127.0.0.1:5000
KLU-eConnect Capt...
Facebook - Log In o...
YouTube
cricbuzz
Home - DeakinSync
Whthive voice audi...
Certification Profile
Your dashboard
Dell
McAfee Security
Sign in
Main Menu
Code Evaluation
Write a Python script that trains a RandomForestClassifier on the Iris dataset and stores the predictions in a variable named student_predictions and the test labels in a variable named student_y_test.
Your Python Code:
# Train a RandomForestClassifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
# Make predictions on the test set
student_predictions = model.predict(X_test)
# Store the test labels in a variable named `student_y_test`
student_y_test = y_test
Submit Code

Code Evaluation

Write a Python script that trains a RandomForestClassifier on the Iris dataset and stores the predictions in a variable named `student_predictions` and the test labels in a variable named `student_y_test`.

Your Python Code:

```
1
```

Submit Code

Results:

```
Your model's accuracy is: 1.0
```

Code Evaluation

Write a Python script that trains a RandomForestClassifier on the Iris dataset and stores the predictions in a variable named `student_predictions` and the test labels in a variable named `student_y_test`.

Your Python Code:

```
1
```

Submit Code

An error occurred:

Found input variables with inconsistent numbers of samples: [30, 150]

```
Traceback (most recent call last):
  File "C:\Users\User\Downloads\code evalution\code evalution\app.py", line 37, in index
    student_accuracy = accuracy_score(student_y_test, student_predictions)
  File "C:\Users\User\Downloads\virtualEnv\lib\site-packages\sklearn\utils\_param_validation.py", line 211, in wrapper
    return func(*args, **kwargs)
  File "C:\Users\User\Downloads\virtualEnv\lib\site-packages\sklearn\metrics\_classification.py", line 220, in accuracy_score
    y_type, y_true, y_pred = _check_targets(y_true, y_pred)
  File "C:\Users\User\Downloads\virtualEnv\lib\site-packages\sklearn\metrics\_classification.py", line 84, in _check_targets
    check_consistent_length(y_true, y_pred)
  File "C:\Users\User\Downloads\virtualEnv\lib\site-packages\sklearn\utils\validation.py", line 409, in check_consistent_length
    raise ValueError(
ValueError: Found input variables with inconsistent numbers of samples: [30, 150]
```

Required Tools and Technology Selection

In the development of the Data Science Challenge Library, careful consideration was given to selecting the appropriate tools and technologies that would collectively contribute to the project's success. The chosen tools were aligned with the project's objectives, user requirements, and the technical expertise of the team. Here's a detailed review of the selected tools and the reasons behind their choice:

Front-end Development: HTML, Bootstrap, CSS, JavaScript

HTML (HyperText Markup Language): HTML is the foundational language for structuring content on the web. It's a standard choice for creating the structure of webpages, defining elements, and establishing relationships between different parts of the page.

Bootstrap: Bootstrap is a widely used front-end framework that provides a responsive and consistent grid system, pre-styled components, and a plethora of CSS styles. Its modular and customizable nature allows for rapid development while ensuring a seamless user experience across various devices.

CSS (Cascading Style Sheets): CSS is essential for designing the visual presentation of web content. It enables the customization of fonts, colors, layouts, and responsive behaviors to create an attractive and user-friendly interface.

JavaScript: JavaScript adds interactivity and dynamic behavior to web applications. It enables client-side scripting, facilitating real-time updates, interactive forms, and enhanced user engagement.

Back-end Development: Flask

Flask: Flask is a lightweight and versatile web framework for Python. It simplifies the process of building web applications by providing tools for routing, templating, and handling requestresponse cycles. Its modular nature allows developers to choose components according to project needs, making it an ideal choice for this project.

Database: SQLite

SQLite: SQLite is a lightweight, self-contained, and serverless database management system. It is an ideal choice for projects like the coding challenge library that require simplicity and efficiency. SQLite's relational model allows for organized storage and retrieval of data. It is particularly well-suited for managing challenge information, user profiles, verification outcomes, and other related content in a structured manner.

Reasons for Choosing These Tools

Alignment with Project Scope: HTML, Bootstrap, CSS, JavaScript, Flask, and SQLite are wellaligned with the project's goal of creating an interactive website for coding challenges. These technologies support the development of a user-friendly and efficient platform.

User Experience and Interface: Bootstrap facilitates the creation of responsive and visually appealing user interfaces, ensuring a consistent experience across various devices. CSS customization further enhances the platform's aesthetics to match the project's design vision.

Speed of Development: Flask's simplicity and modular approach accelerate back-end development, providing essential tools without unnecessary complexity. This streamlined development process allows the team to prioritize core functionalities.

Database Flexibility: SQLite's relational structure provides the necessary flexibility to adapt to evolving challenge structures and user-generated content. It accommodates a diverse range of data types while maintaining data integrity.

Python Integration: Flask's Python-based development and SQLite's compatibility with Python's ecosystem seamlessly integrate the front-end and back-end, facilitating smooth data flow and interaction.

Scalability: SQLite's efficiency and ability to handle growing datasets ensure the platform's scalability as the number of challenges and users increases over time.

Team Expertise: Leveraging the team's proficiency with these technologies minimizes the learning curve, promoting efficient collaboration, development, and effective problem-solving throughout the project's lifecycle.

Literature Review

Literature Review: "Towards an Online Programming Platform Complementing Software Engineering Education"

The paper titled "Towards an Online Programming Platform Complementing Software Engineering Education" by Niels Gandraß, Torge Hinrichs, and Axel Schmolitzky offers a profound exploration into the realm of software engineering education. Through the lens of an innovative online programming platform, the authors embark on a journey to transform traditional pedagogical methods into dynamic, interactive, and enriched learning experiences. This literature review delves into the key contributions and insights this paper brings to the forefront.

Literature Review: "The Effectiveness of Gamification in Programming Education: Evidence from a Meta-Analysis"

The study titled "The Effectiveness of Gamification in Programming Education: Evidence from a Meta-Analysis," authored by Zehui Zhan, Luyao He, Yao Tong, Xinya Liang, Shihao Guo, and Xixin Lan, delves into the intersection of gamification and programming education. This literature review seeks to unveil the key insights and contributions of this research endeavor.

Literature Review: "Prospects and Challenges of Using Machine Learning for Academic Forecasting"

The paper titled "Prospects and Challenges of Using Machine Learning for Academic Forecasting" authored by Edeh Michael Onyema, Khalid K. Almuzaini, Fergus Uchenna Onu, Devvret Verma, Ugboaja Samuel Gregory, Monika Puttaramaiah, and Rockson Kwasi Afriyie delves into the intersection of machine learning and academic forecasting. This literature review seeks to unravel the key insights and contributions of this research endeavor.

Literature Review: "An AI-Based System for Formative and Summative Assessment in Data Science Courses"

The article titled "An AI-Based System for Formative and Summative Assessment in Data Science Courses" authored by Pierpaolo Vittorini, Stefano Menini, and Sara Tonelli, published in the International Journal of Artificial Intelligence in Education, delves into the integration of artificial intelligence (AI) into the assessment process within data science courses. This literature review endeavors to elucidate the key insights and contributions of this scholarly work.

Literature Review: "AI-Based Adaptive Personalized Content Presentation and Exercises Navigation for an Effective and Engaging E-Learning Platform"

The research article titled "AI-Based Adaptive Personalized Content Presentation and Exercises Navigation for an Effective and Engaging E-Learning Platform," authored by Wafaa S. Sayed, Ahmed M. Noeman, Abdelrahman Abdellatif, Moemen Abdelrazek, Mostafa G. Badawy, Ahmed Hamed, and Samah El-Tantawy, encapsulates a profound exploration of integrating artificial intelligence (AI) into the realm of e-learning. This literature review endeavors to unveil the pivotal insights and contributions of this scholarly work.

Deliverables with Code- Screenshots

Data Cleansing

```
[2] import matplotlib.pyplot as plt
    import numpy as np
    import pandas as pd
    import seaborn as sns

    import warnings
    from sklearn.exceptions import ConvergenceWarning
    warnings.filterwarnings('ignore', category=FutureWarning)
    warnings.filterwarnings('ignore', category=DeprecationWarning)
    warnings.filterwarnings('ignore', category=ConvergenceWarning)

    %matplotlib inline
```

```
[3] q_data = pd.read_excel('data.xlsx')
    q_data.head()
```

	ID	Type	Questions	Answers
0	1	case study	You are given a train data set having 1000 col...	Processing a high dimensional data on a limite...
1	2	case study	Is rotation necessary in PCA? If yes, Why? Wha...	Yes, rotation (orthogonal) is necessary becaus...
2	3	case study	You are given a data set. The data set has mi...	This question has enough hints for you to star...
3	4	case study	You are given a data set on cancer detection. ...	If you have worked on enough data sets, you sh...
4	5	case study	You are working on a time series data set. You...	Time series data is known to posses linearity....

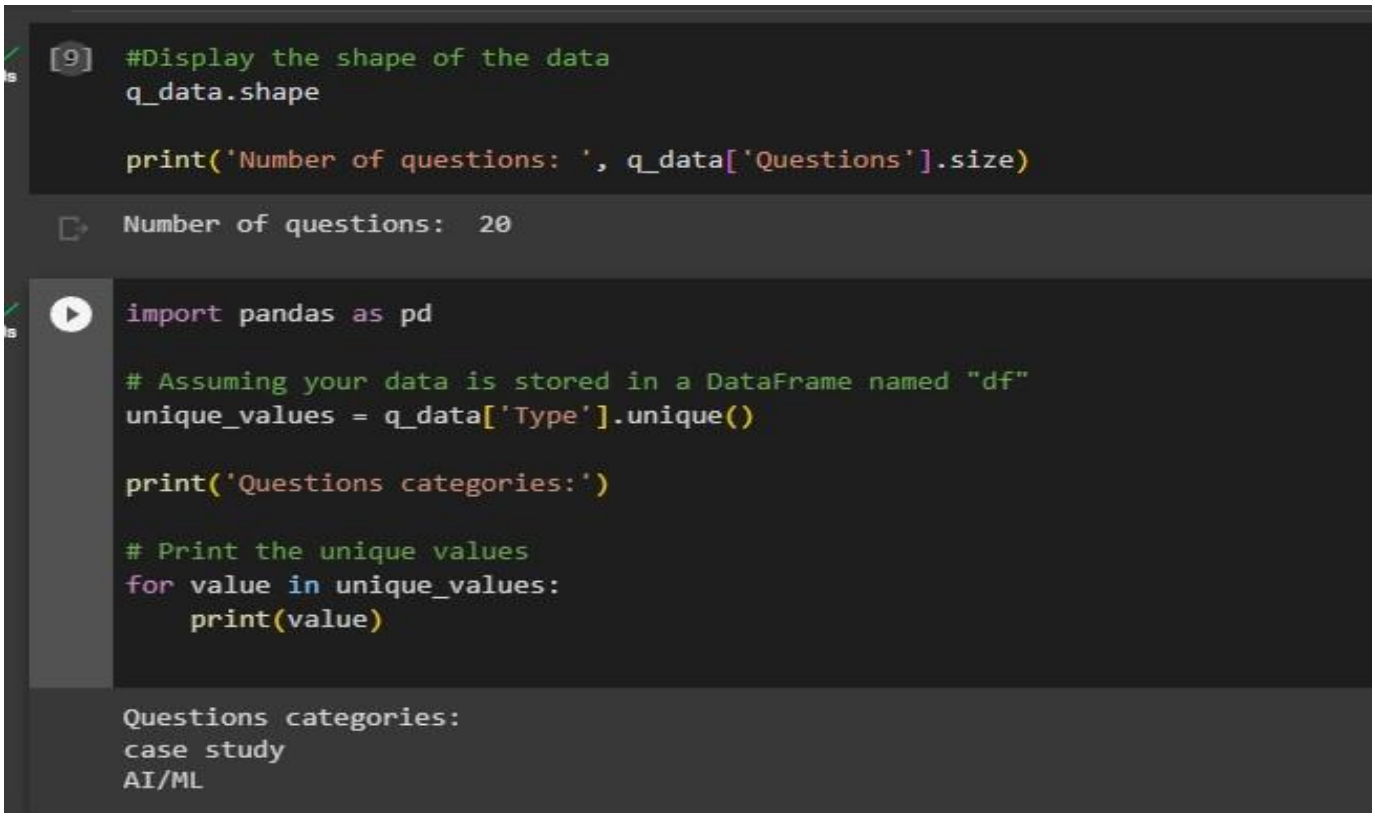
```
[4] #Drop ID Column
    q_data = q_data.drop('ID', axis=1)
```

```
[5] #Dsiplay the first 5 rows
    q_data.head()
```

	Type	Questions	Answers
0	case study	You are given a train data set having 1000 col...	Processing a high dimensional data on a limite...
1	case study	Is rotation necessary in PCA? If yes, Why? Wha...	Yes, rotation (orthogonal) is necessary becaus...
2	case study	You are given a data set. The data set has mi...	This question has enough hints for you to star...

✓ 0s completed at 8:45 PM

Data Visualization



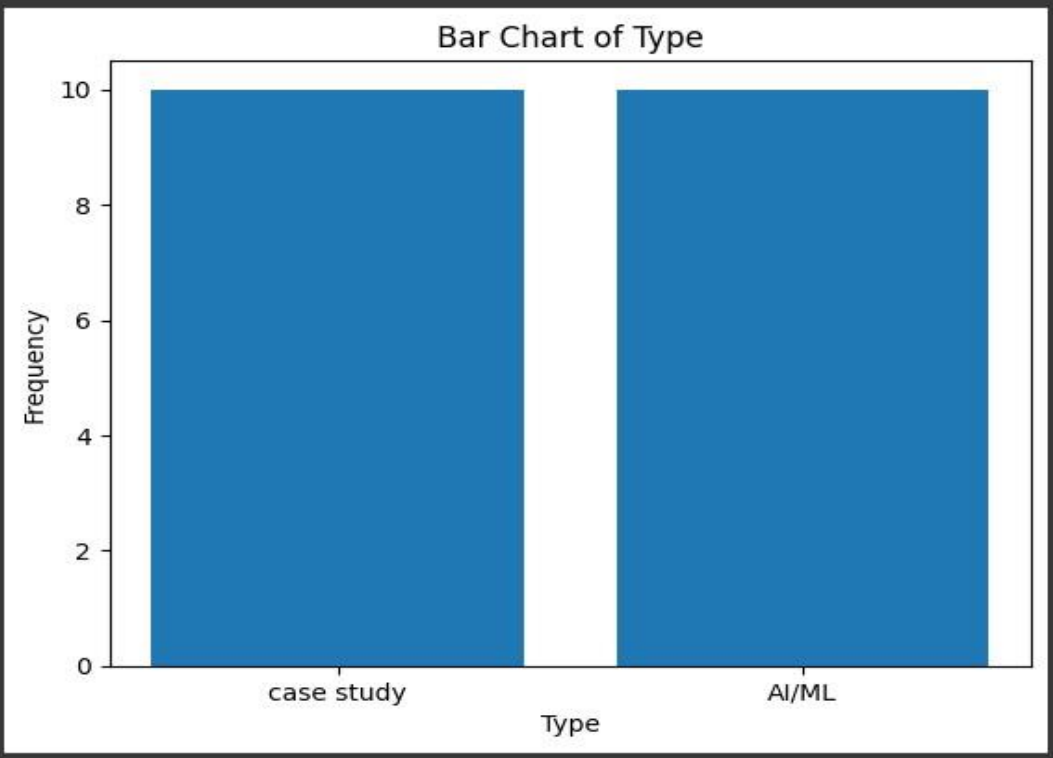
```
[9] #Display the shape of the data
    q_data.shape

    print('Number of questions: ', q_data['Questions'].size)
```

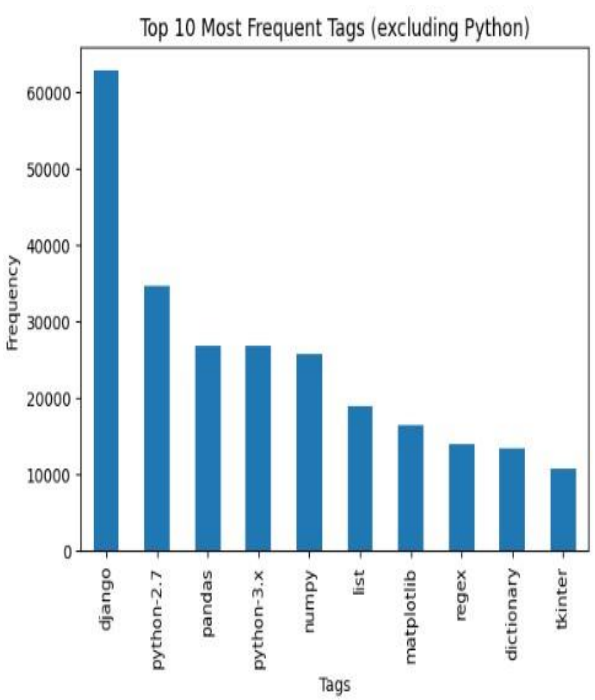
```
Number of questions:  20
```

```
import pandas as pd

# Assuming your data is stored in a DataFrame named "df"
unique_values = q_data['Type'].unique()

print('Questions categories:')

# Print the unique values
for value in unique_values:
    print(value)
```

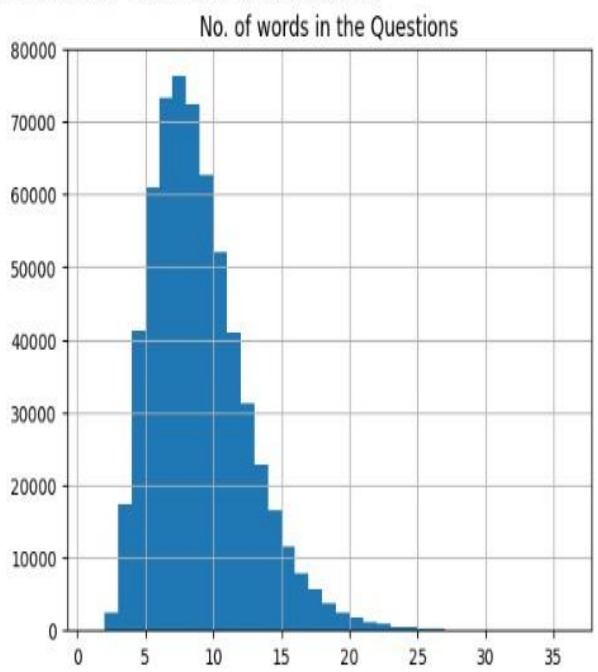
```
Questions categories:
case study
AI/ML
```

```
[ ]  # Count the frequency of each unique value in the "Type" column
     value_counts = q_data['Type'].value_counts()

     # Create a bar chart
     plt.bar(value_counts.index, value_counts.values)

     # Add labels and title
     plt.xlabel('Type')
     plt.ylabel('Frequency')
     plt.title('Bar Chart of Type')

     # Display the chart
     plt.show()
```

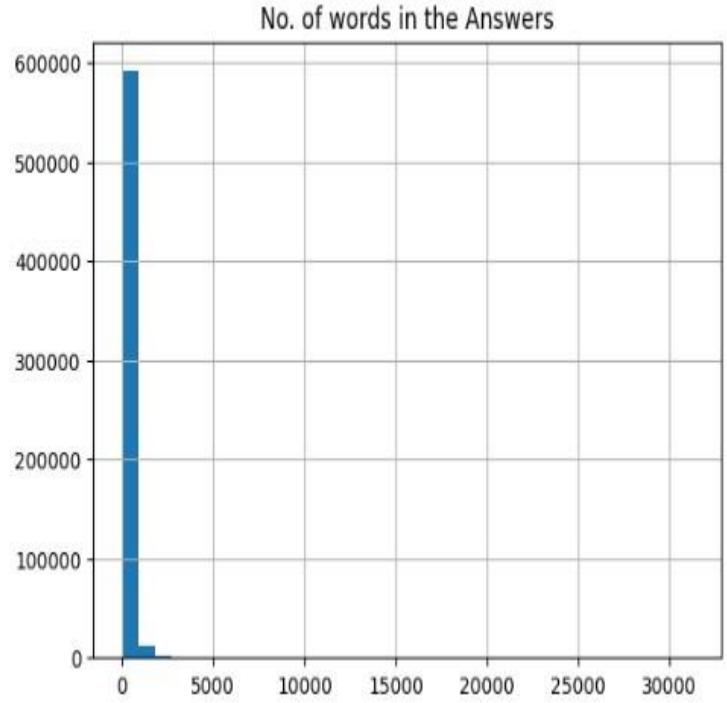


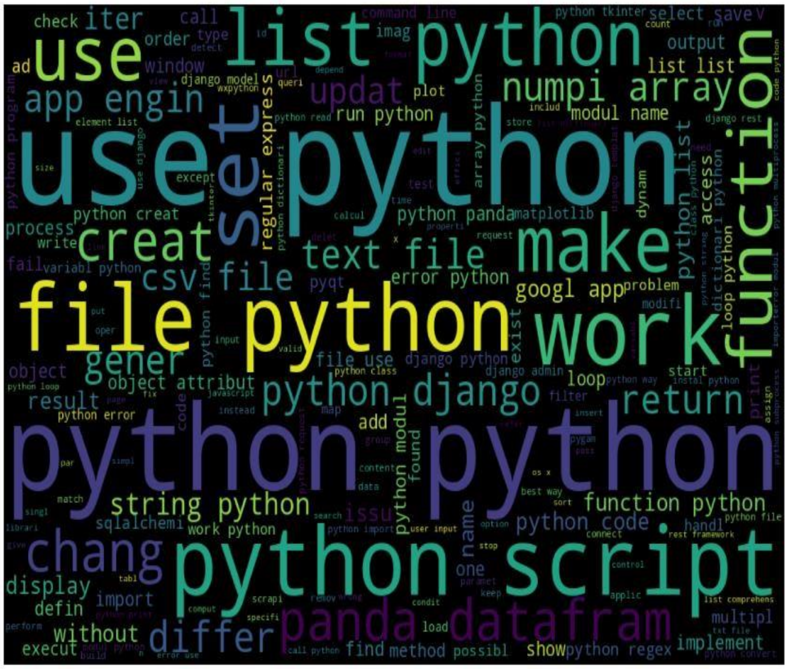


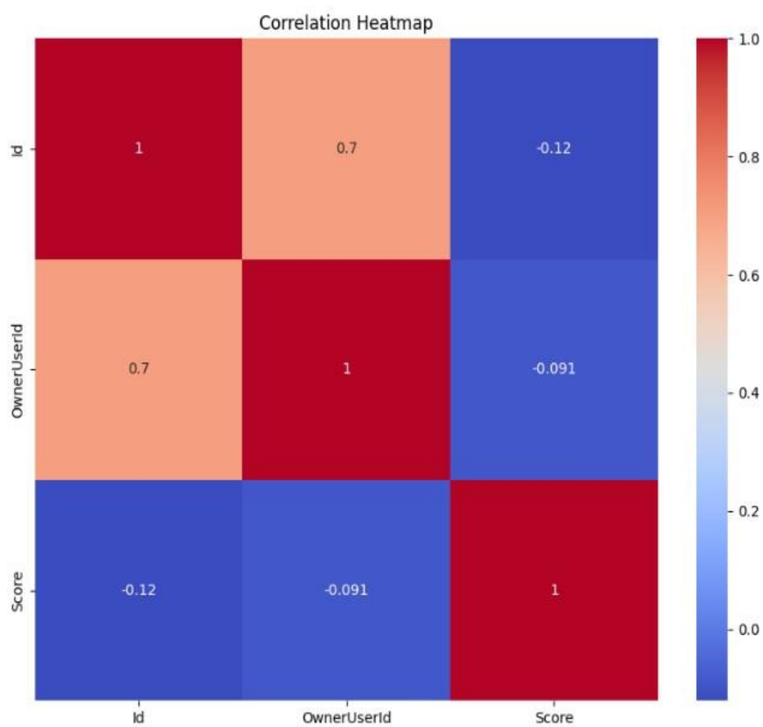

```
Text(0.5, 1.0, 'No. of words in the Questions')
```



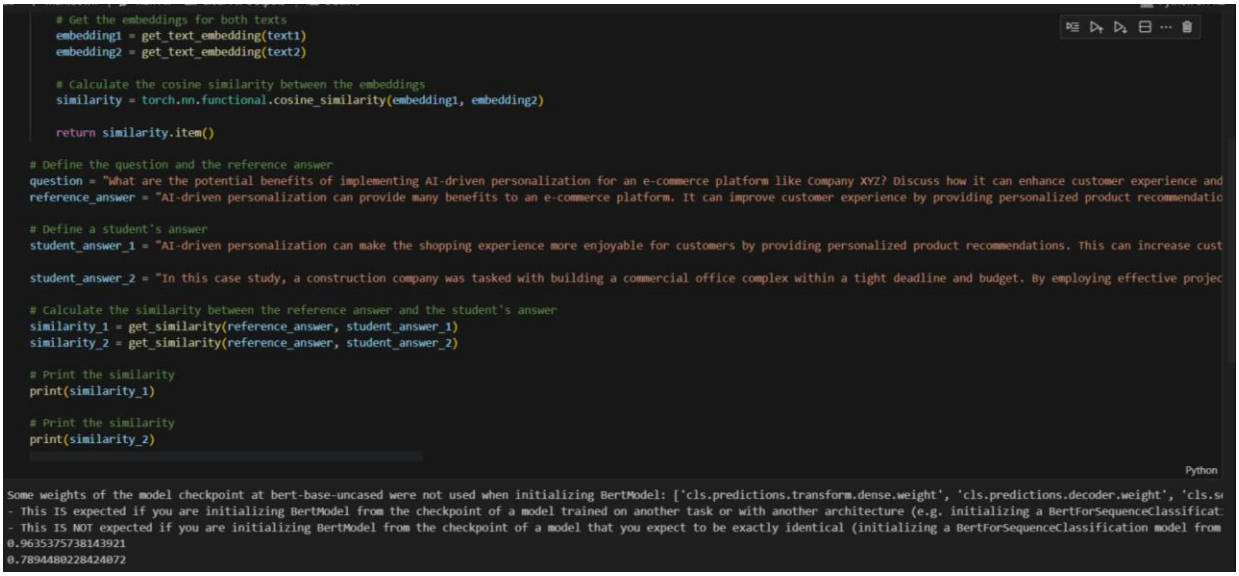

```
Text(0.5, 1.0, 'No. of words in the Answers')
```




use python
python python
file python
list python
python script
work
function
make
set
use
panda datafram
python django
creat
app engin
numpi array
updat
text file
chang
differ
return
string python
function python
python code
gener
csv
file
googl app
issu
name
check
iter
call
order
window
output
list list
modul name
run python
plot
regular express
python find
python list
dynam
access
error python
python panda
matplotlib
process
write
python creat
fail
variabl python
result
object
object attribut
file use
django python
django admin
loop
start
filter
add
map
python modul
found
exist
display
defin
import
without
execut
find
method
possibl
show
python regex
implement
multipl
handl
sqlalchemi
work python
one
select
save
imag
type
url
problem
modifi
print
code
test
pyqt
array python
dictionari python
loop python


Correlation Heatmap
Id
OwnerUserId
Score
1
0.7
-0.12
0.7
1
-0.091
-0.12
-0.091
1
Id
OwnerUserId
Score
1.0
0.8
0.6
0.4
0.2
0.0

Text Verification

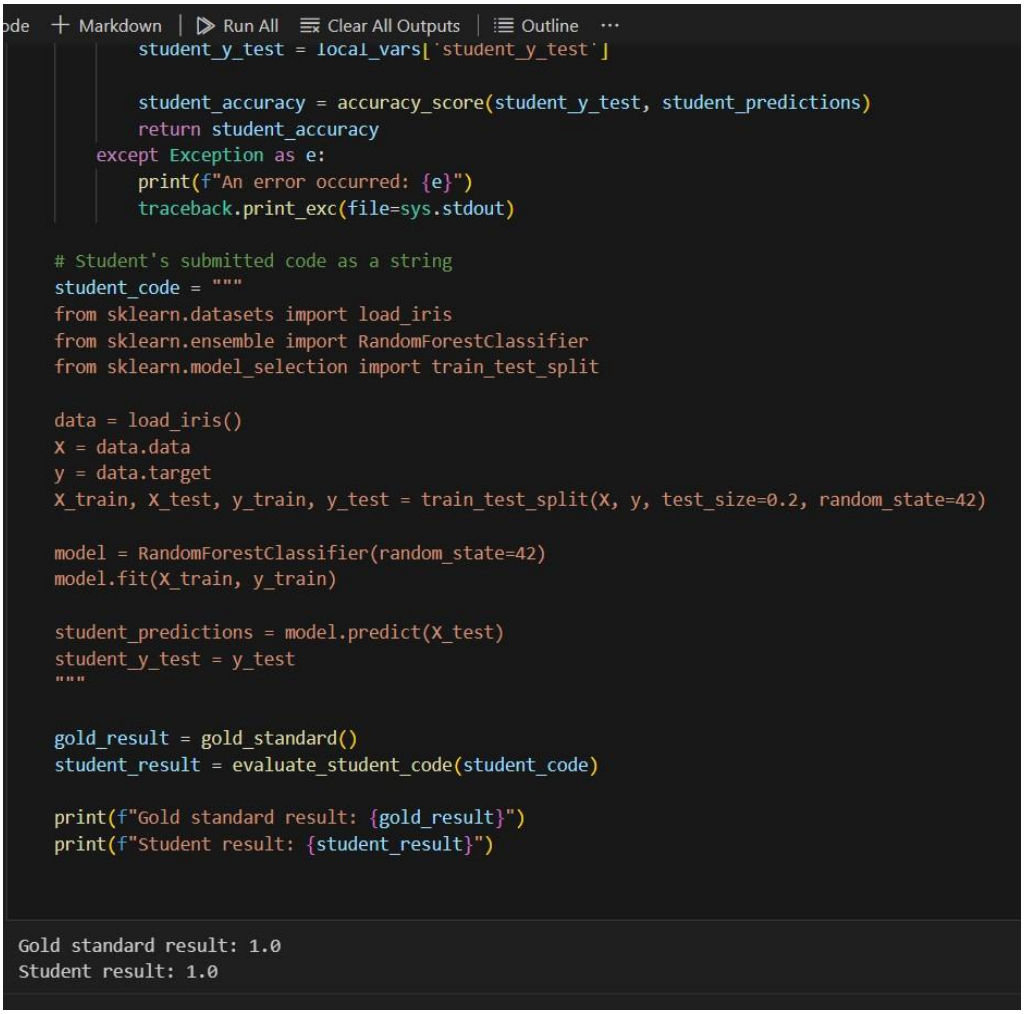


```
    # Get the embeddings for both texts
    embedding1 = get_text_embedding(text1)
    embedding2 = get_text_embedding(text2)

    # Calculate the cosine similarity between the embeddings
    similarity = torch.nn.functional.cosine_similarity(embedding1, embedding2)

    return similarity.item()

# Define the question and the reference answer
question = "What are the potential benefits of implementing AI-driven personalization for an e-commerce platform like Company XYZ? Discuss how it can enhance customer experience and
reference_answer = "AI-driven personalization can provide many benefits to an e-commerce platform. It can improve customer experience by providing personalized product recommendatio

# Define a student's answer
student_answer_1 = "AI-driven personalization can make the shopping experience more enjoyable for customers by providing personalized product recommendations. This can increase cust

student_answer_2 = "In this case study, a construction company was tasked with building a commercial office complex within a tight deadline and budget. By employing effective projec

# Calculate the similarity between the reference answer and the student's answer
similarity_1 = get_similarity(reference_answer, student_answer_1)
similarity_2 = get_similarity(reference_answer, student_answer_2)

# Print the similarity
print(similarity_1)

# Print the similarity
print(similarity_2)
```

Python

```
Some weights of the model checkpoint at bert-base-uncased were not used when initializing BertModel: ['cls.predictions.transform.dense.weight', 'cls.predictions.decoder.weight', 'cls.s
- This IS expected if you are initializing BertModel from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassificat
- This IS NOT expected if you are initializing BertModel from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from
0.9635375738143921
0.7894480228424072
```

Algorithm verification

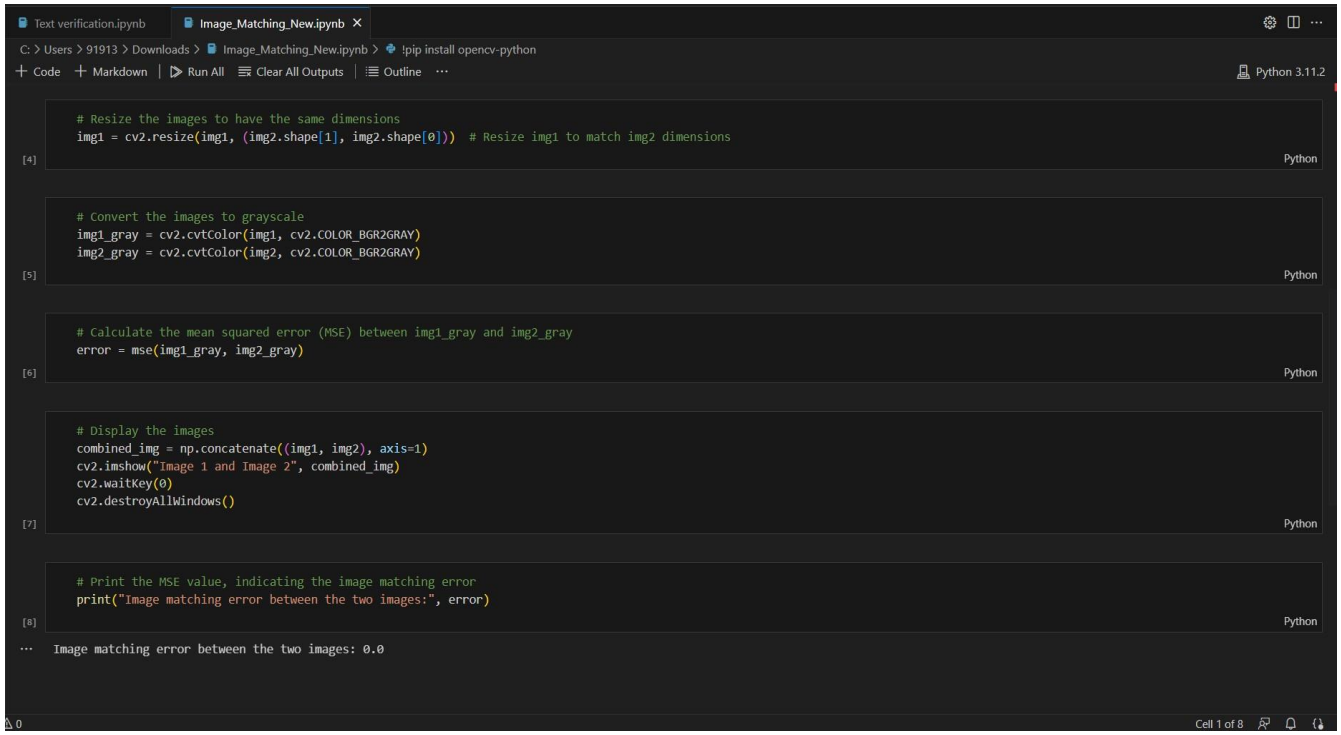


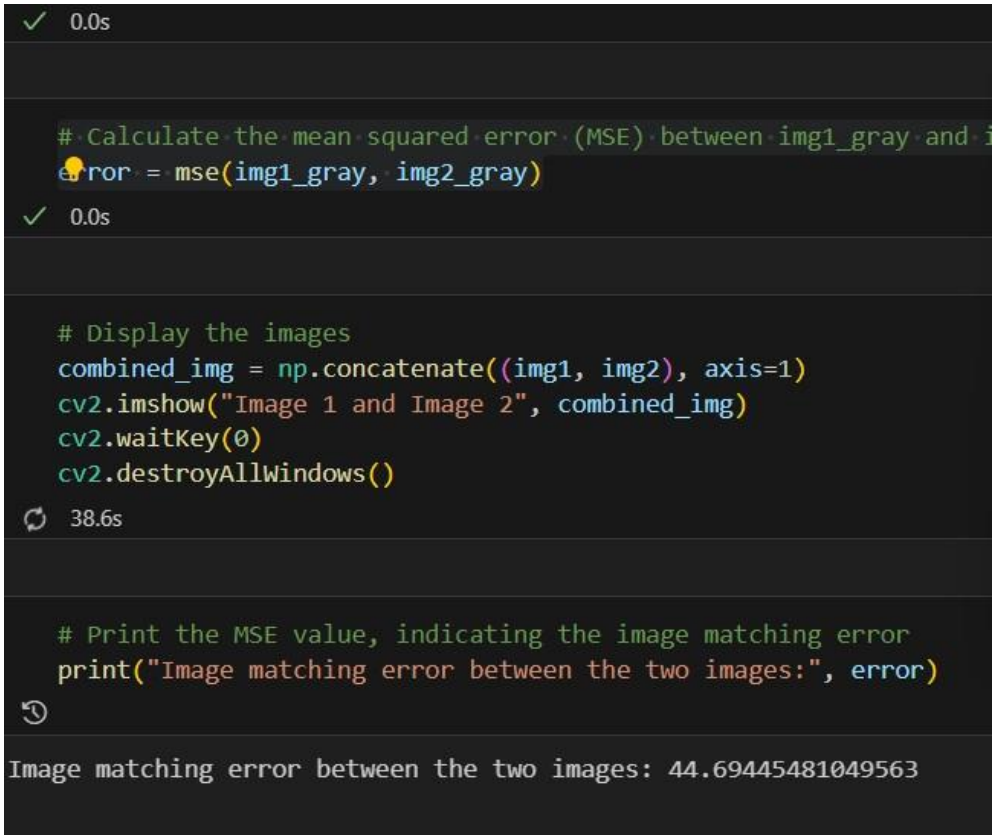
ode + Markdown | Run All | Clear All Outputs | Outline ···

```
            student_y_test = local_vars['student_y_test']

            student_accuracy = accuracy_score(student_y_test, student_predictions)
            return student_accuracy
        except Exception as e:
            print(f"An error occurred: {e}")
            traceback.print_exc(file=sys.stdout)

    # Student's submitted code as a string
    student_code = """
    from sklearn.datasets import load_iris
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.model_selection import train_test_split

    data = load_iris()
    X = data.data
    y = data.target
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    model = RandomForestClassifier(random_state=42)
    model.fit(X_train, y_train)

    student_predictions = model.predict(X_test)
    student_y_test = y_test
    """

    gold_result = gold_standard()
    student_result = evaluate_student_code(student_code)

    print(f"Gold standard result: {gold_result}")
    print(f"Student result: {student_result}")
```

```
Gold standard result: 1.0
Student result: 1.0
```

Data Visualization Verification





Plagiarism Verification

```
    tagged_data = [TaggedDocument(words=word_tokenize(_d.lower()), tags=[str(i)]) for i, _d in enumerate(training_data)]
    model = Doc2Vec(tagged_data, vector_size=20, min_count=1, epochs=100)
    return model

def compute_similarity(model, source_code1, source_code2):
    test_data = word_tokenize(source_code1.lower())
    v1 = model.infer_vector(test_data)

    test_data = word_tokenize(source_code2.lower())
    v2 = model.infer_vector(test_data)

    return model.wv.cosine_similarities(v1, [v2])

# List of source code scripts for training
source_codes = [
    "def add(a, b): return a + b",
    "def subtract(a, b): return a - b",
    "def multiply(a, b): return a * b",
    "def divide(a, b): return a / b if b != 0 else None",
    "def power(a, b): return a ** b"
]

# Source code to compare
source_code1 = "def add_numbers(a, b): return a + b"
source_code2 = "def subtract_numbers(a, b): return a - b"

model = train_model(source_codes)
similarity = compute_similarity(model, source_code1, source_code2)

print(f"The similarity between the two source codes is: {similarity}")
```

```
he similarity between the two source codes is: [0.9736019]
```

Review Verification

```
No model was supplied, defaulted to distilbert-base-uncased-finetuned-sst-2-english and revision af0f99b (https://huggingface.co/distilbert
Using a pipeline without specifying a model name and revision in production is not recommended.

Downloading (…)lve/main/config.json:   0%|          | 0.00/629 [00:00<?, ?B/s]

c:\Users\91913\AppData\Local\Programs\Python\Python311\Lib\site-packages\huggingface_hub\file_download.py:133: UserWarning: `huggingface_hu
To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to see activate d
  warnings.warn(message)

Downloading pytorch_model.bin:   0%|          | 0.00/268M [00:00<?, ?B/s]

Downloading (…)okenizer_config.json:   0%|          | 0.00/48.0 [00:00<?, ?B/s]

Downloading (…)solve/main/vocab.txt:   0%|          | 0.00/232k [00:00<?, ?B/s]

Sentiment: POSITIVE
Sentiment Score: 0.9998866319656372
```

Flask Application

```
flask = Flask(__name__)
flask.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///test.db'
flask.config['SECRET_KEY'] = 'your_secret_key'
db = SQLAlchemy(flask)

class Problem(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    problem = db.Column(db.String(200), nullable=False)
    solution = db.Column(db.String(200), nullable=False)

class Attempt(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    user = db.Column(db.String(80), nullable=False)
    problem_id = db.Column(db.Integer, db.ForeignKey('problem.id'), nullable=False)
    attempt = db.Column(db.String(200), nullable=False)
    score = db.Column(db.Float, nullable=False)

def get_similarity(text1, text2):
    vectorizer = TfidfVectorizer()
    tfidf_matrix = vectorizer.fit_transform([text1, text2])
    similarity_matrix = cosine_similarity(tfidf_matrix[0:1], tfidf_matrix)
    return similarity_matrix[0, 1]

# ************************************ root
@flask.route('/', methods=['GET', 'POST'])
def login():
    if 'user_id' in session:
        return redirect(url_for('dashboard'))

    if request.method == 'POST':
        user_id = request.form['user_id']
        password = request.form['password']

        session['user_id'] = user_id
        return redirect(url_for('dashboard'))
```

```python
# ************************************** dashboard
@flask.route('/dashboard')
def dashboard():
    if 'user_id' not in session:
        return redirect(url_for('login'))
    return render_template('dashboard.html')

@flask.route('/data_science_quiz')
def data_science_quiz():
    if 'user_id' not in session:
        return redirect(url_for('login'))
    return redirect(url_for('index'))

# ************************************** home
@flask.route('/index', methods=['GET', 'POST'])
def index():
    if 'user_id' not in session:
        return redirect(url_for('login'))

    if request.method == 'POST':
        problem = request.form['problem']
        new_problem = Problem(problem=problem, solution="")
        db.session.add(new_problem)
        db.session.commit()
        return redirect(url_for('index'))

    problems = Problem.query.all()
    return render_template('index.html', problems=problems)

# ************************************** upload questions
@flask.route('/submit_solution', methods=['POST'])
def submit_solution():
    if 'user_id' not in session:
        return redirect(url_for('login'))

    problem_id = request.form['problem_id']
    solution = request.form['solution']
```

Conclusion

The Data Science Challenge Library project, driven by a profound commitment to enhancing education through practical engagement and innovation, stands poised to revolutionize the learning landscape. With a meticulously planned approach, the project aims to address the challenges faced by data science students, bridging the gap between theoretical knowledge and real-world application.

The fusion of technology, pedagogy, and user-centric design propels this initiative forward. By offering an expansive repository of challenges spanning AI/ML, data visualizations, and analytics, the project caters to diverse learning styles and aspirations. The creation of an intuitive user interface beckons learners to embark on an enriching journey, where challenges become gateways to deeper understanding and mastery.

The project's emphasis on verification mechanisms underscores its commitment to academic integrity and constructive learning. The incorporation of text, algorithm, and data visualization verification mechanisms ensures that learners not only arrive at accurate solutions but also cultivate a deep comprehension of their craft.

Gamification elements further infuse the platform with motivation and engagement, cultivating a culture of continuous improvement and collaborative learning. Leaderboards and progress tracking transform learning into a dynamic journey of growth and achievement.

The Data Science Challenge Library's future holds promising prospects. As advanced services like plagiarism verification and anomaly detection are integrated, the platform's analytical prowess will amplify. The collection of user feedback, combined with iterative refinement, ensures that the platform's evolution remains in lockstep with learners' needs and aspirations.

In the grand tapestry of education, the Data Science Challenge Library project is a testament to the power of merging technology, education, and innovation. As it continues to shape the landscape of data science education, it is poised to empower learners, inspire educators, and usher in a new era of dynamic and immersive learning experiences.

Future Improvements

The Data Science Challenge Library project's commitment to continuous enhancement is a driving force in shaping the future of data science education. As the platform gains traction and evolves, several avenues for improvement present themselves, ensuring an even more impactful learning experience for students:

Expanded Challenge Categories: While the initial focus may be on AI/ML, data visualizations, and analytics, the project can diversify its challenge categories to encompass broader areas of data science, such as natural language processing, image recognition, and deep learning. This expansion will cater to a wider range of learners' interests and career aspirations.

Real-Time Collaboration: Introducing real-time collaboration features would enable students to work together on challenges, fostering a sense of community and shared learning. Collaborative coding, pair programming, and virtual study groups could become integral components of the platform.

Interactive Learning Paths: Building curated learning paths that guide students from foundational challenges to more advanced topics can provide a structured learning journey. These paths could be tailored to different skill levels and learning objectives, ensuring a progressive and comprehensive learning experience.

Enhanced Feedback Mechanisms: Beyond verification, providing detailed and personalized feedback on students' solutions can be immensely valuable. Offering explanations, alternative approaches, and constructive criticism will facilitate deeper understanding and improvement.

Integration with Data Sources: Incorporating real-world datasets from various industries and domains would enable students to tackle challenges that closely mirror industry scenarios. This integration could foster a deeper connection between academic learning and practical applications.

Interactive Tutorials: Embedding interactive tutorials and walkthroughs within challenges can guide students step by step, making complex concepts more approachable. These tutorials could be accompanied by hints and tips for students to refer to as needed.

Certifications and Badges: Introducing certifications and badges for completing challenges or mastering specific skill sets would incentivize learners and provide a tangible recognition of their achievements. These credentials could hold value in academic settings and job applications.

Integration with Learning Management Systems: Seamless integration with popular learning management systems used in educational institutions would facilitate easy adoption and access for students and educators alike.

Advanced Analytics Dashboard: Building an advanced analytics dashboard could provide insights into learners' progress, performance trends, and areas of improvement. Educators and students could benefit from data-driven insights to tailor learning strategies.

Personalized Recommendations: Implementing AI-driven algorithms to recommend challenges based on a student's skill level, preferences, and learning history would create a tailored learning experience.

Steps to host the website in Cloud

Hosting a Data Science Challenge Library website in the cloud involves several steps to ensure a smooth deployment. Below are the general steps you can follow to host your website in the cloud:

1. Choose a Cloud Provider:

Select a cloud platform that suits your requirements and preferences. Common options include Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and others.

2. Register and Set Up an Account:

Create an account on the chosen cloud platform and set up your billing and payment preferences.

3. Domain and SSL Certificate:

If you have a custom domain, configure DNS settings to point to your cloud resources. Set up an SSL certificate to ensure secure communication over HTTPS.

4. Configure Virtual Machine (VM) or Container:

Set up a virtual machine instance or a containerized environment (using services like AWS EC2, Azure Virtual Machines, or Google Compute Engine) to host your web application.

5. Install Required Software:

Install the necessary software components, including your web server (e.g., Nginx, Apache), database server (e.g., SQLite), and any other dependencies.

6. Upload Your Application Code:

Upload your Data Science Challenge Library website's code to the cloud VM or container.

7. Configure Security Groups and Firewalls:

Configure security groups or firewalls to restrict access to your VM or container and ensure that only authorized traffic is allowed.

8. Database Setup:

Set up your SQLite database and configure the connection details in your application.

9. Environment Configuration:

Set environment variables and configuration files for your application, including database connection strings, API keys, and any other sensitive information.

10. Web Server Configuration:

Configure your web server (e.g., Nginx, Apache) to serve your application, including setting up virtual hosts and SSL settings.

11. Application Deployment:

Start your web application and ensure it's running without errors. Test its functionality and access it using the provided domain or IP address.

12. Backup and Monitoring:

Set up regular backups of your application and database to ensure data safety. Configure monitoring tools to keep track of the application's health and performance.

13. Scaling and Load Balancing (Optional):

If your application experiences high traffic, consider implementing auto-scaling and load balancing to ensure stability and optimal performance.

14. Domain Configuration:

Configure your domain's DNS settings to point to the cloud resources and ensure the proper functioning of your website.

15. Testing:

Thoroughly test your hosted website to ensure all features work as expected in the cloud environment.

16. Documentation and Monitoring:

Document the deployment process, configurations, and any necessary information. Set up monitoring and alerts to promptly address any issues that may arise.

References

Gandraß, N., Hinrichs, T., & Schmolitzky, A. (n.d.). *Towards an Online Programming Platform Complementing Software Engineering Education*. Retrieved August 14, 2023, from https://ceur-ws.org/Vol-2531/paper05.pdf

Zhan, Z., He, L., Tong, Y., Liang, X., Guo, S., & Lan, X. (2022). The effectiveness of gamification in programming education: Evidence from a meta-analysis. *Computers and Education: Artificial Intelligence*, *3*, 100096. https://doi.org/10.1016/j.caeai.2022.100096

Onyema, E. M., Almuzaini, K. K., Onu, F. U., Verma, D., Gregory, U. S., Puttaramaiah, M., & Afriyie, R. K. (2022). Prospects and Challenges of Using Machine Learning for Academic Forecasting. *Computational Intelligence and Neuroscience*, *2022*, 1–7. https://doi.org/10.1155/2022/5624475

Vittorini, P., Menini, S., & Tonelli, S. (2020). An AI-Based System for Formative and Summative Assessment in Data Science Courses. *International Journal of Artificial Intelligence in Education*. https://doi.org/10.1007/s40593-020-00230-2

Sayed, W. S., Noeman, A. M., Abdellatif, A., Abdelrazek, M., Badawy, M. G., Hamed, A., & ElTantawy, S. (2022). AI-based adaptive personalized content presentation and exercises navigation for an effective and engaging E-learning platform. *Multimedia Tools and Applications*, 1–31. https://doi.org/10.1007/s11042-022-13076-8