ENGAGEMENT AND ASSESSMENT OPTIMIZATION SYSTEM FOR ENHANCING LEARNING EXPERIENCES IN ONLINE EDUCATION

24-25J-320

EDUSMART

Our Team

Ms. Wishalya Tissera

[SUPERVISOR]

Ms. Chathushki Chathumali

[CO-SUPERVISOR]

Prematilaka

S.P.D.M

Wijenandana

S.D

Ekanayaka

H.E.M.P.L

Karunarathna

J.H.H.N

SLIIT FACULTY OF COMPUTING

12/8/2024

Content

1 Introduction

2 Research Question

3 Main Objective

4 Sub Objectives

5 System Overview Diagram

6 Individual Components Section

7 Requirements

8 Gantt Chart

9 Work breakdown Chart

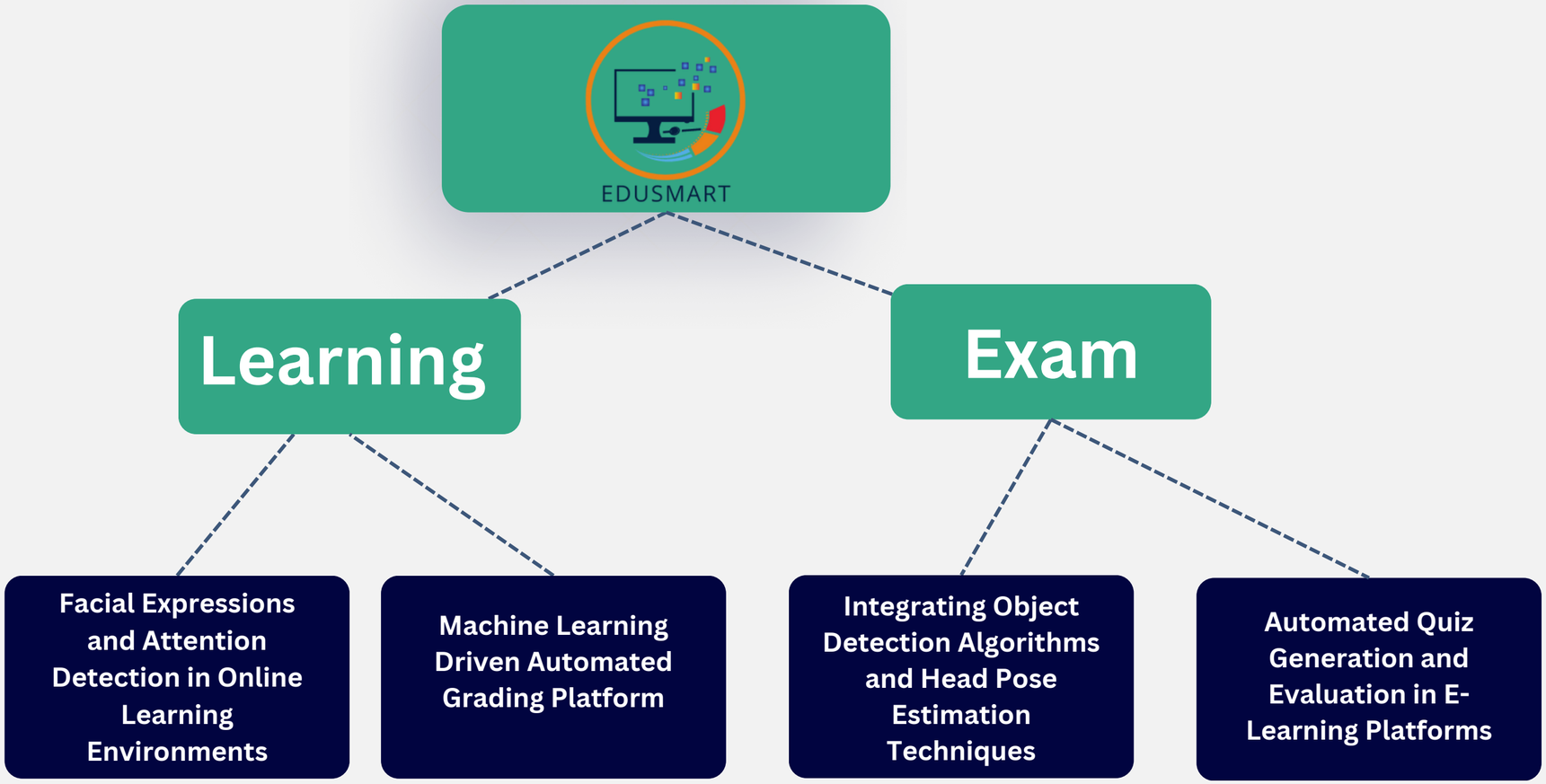
EDUSMART

INTRODUCTION

EDUSMART

- The rise of online learning has transformed education, making it more accessible and flexible for students worldwide
- Online education faces challenges in maintaining student engagement and ensuring the integrity and accuracy of assessments,
- To address these challenges, our research focuses on integrating advanced technologies like facial expression detection, object detection, and machine learning-driven assessments into e-learning platforms, aiming to enhance both engagement and evaluation processes.

INTRODUCTION CON.



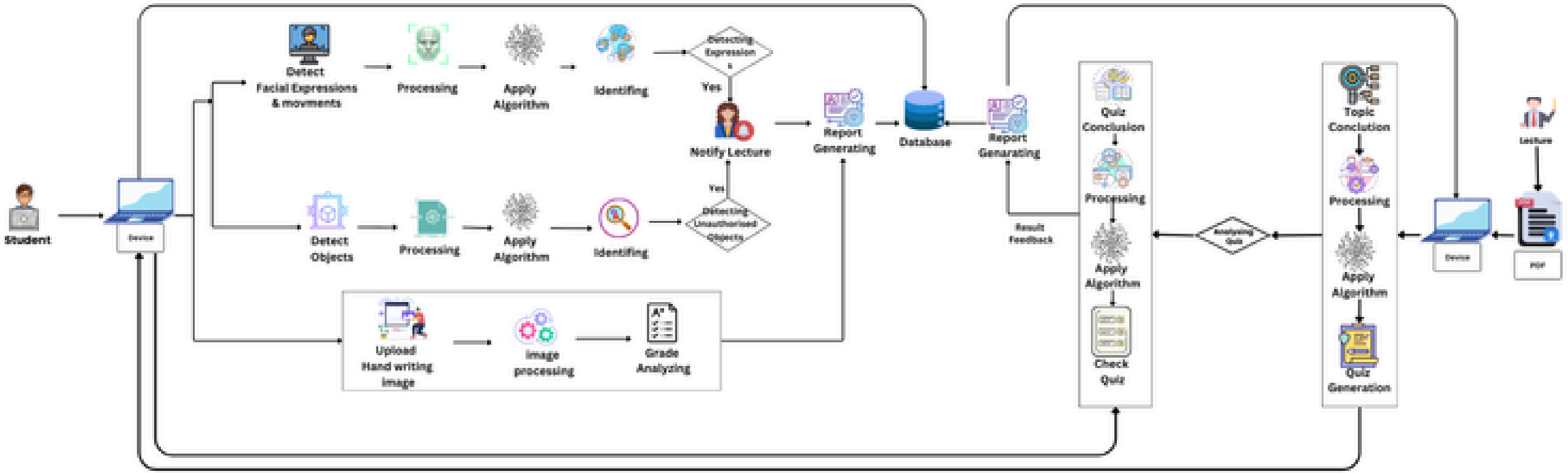
SUB OBJECTIVES

EDUSMART

- Facial Expressions and Attention Detection in Online Learning Environments: Enhancing Student Engagement and Real-Time Feedback
- Integrating Object Detection Algorithms and Head Pose Estimation Techniques Into E-Learning Platforms
- Automated Quiz Generation and Evaluation
- Machine Learning-Driven Automated Grading Platform Enhancing Fairness and Scalability in Educational Assessment

SYSTEM OVERVIEW

EDUSMART



INDIVIDUAL COMPONENTS

EDUSMART

IT21185298 | PREMATHILAKA S.P.D.M

EDUSMART

COMPONENT 01

Facial Expressions and Attention Detection in Online Learning Environments

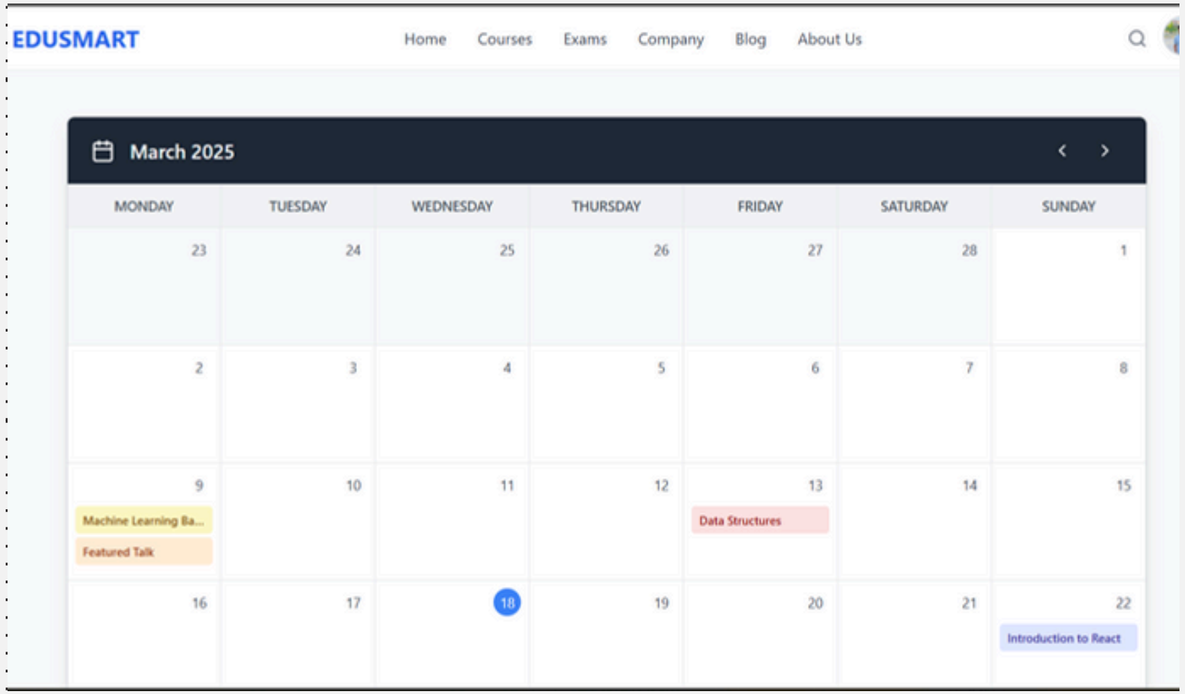
SUB OBJECTIVE

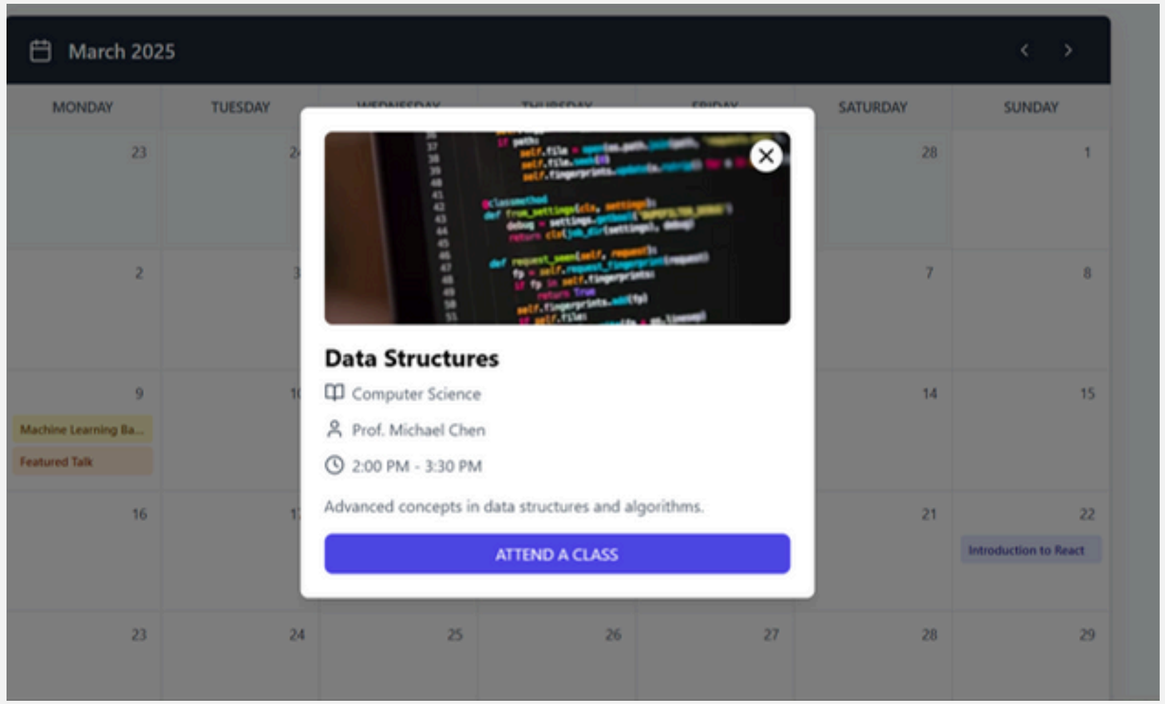
- Add Real Time Notification system
- Develop a teacher interface for real-time monitoring and analysis of students' facial expressions and emotional states.
- Design and Development of Detection Algorithms
- Integration with Learning Management Systems (LMS)

EDUSMART

SYSTEM UI

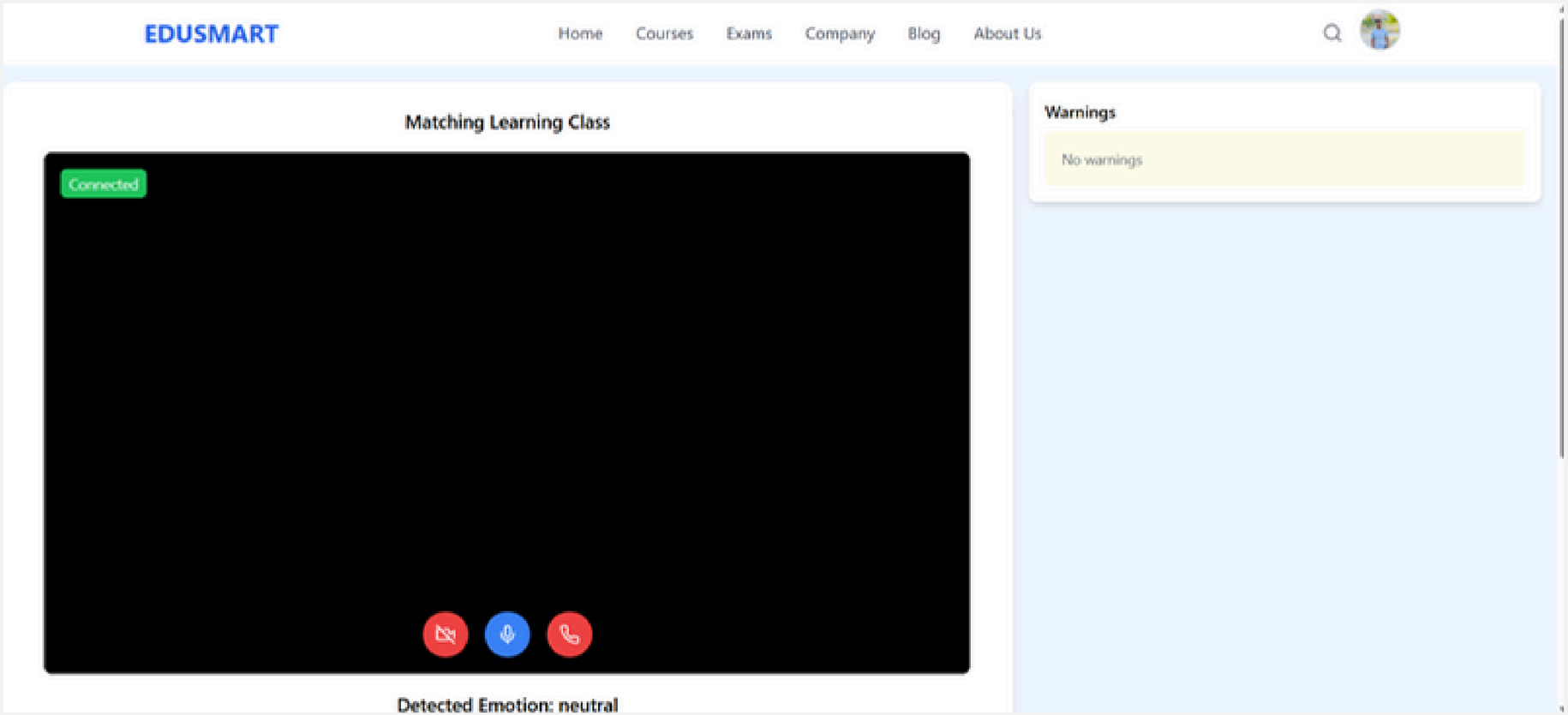
EDUSMART





Student calender

SYSTEM UI



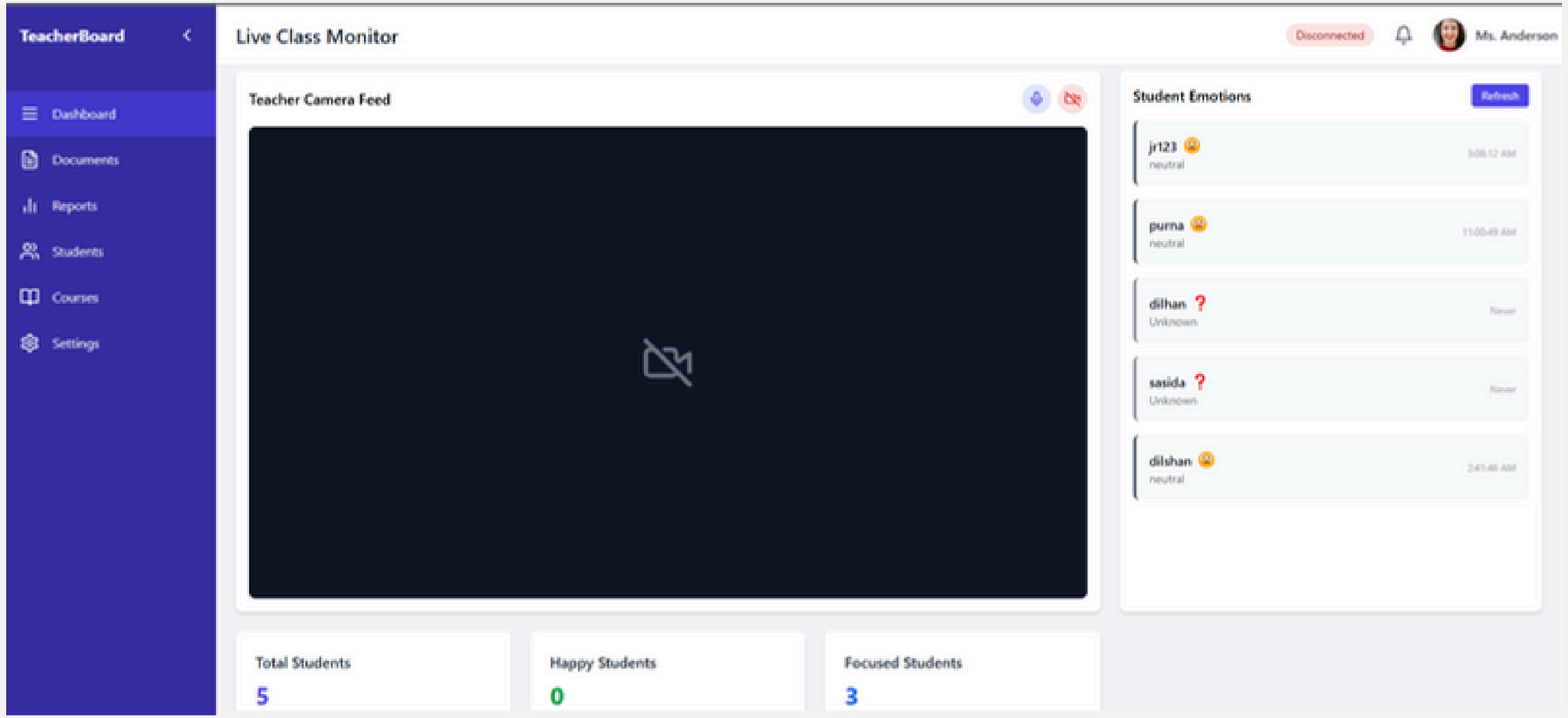
Classroom

SYSTEM UI

EDUSMART

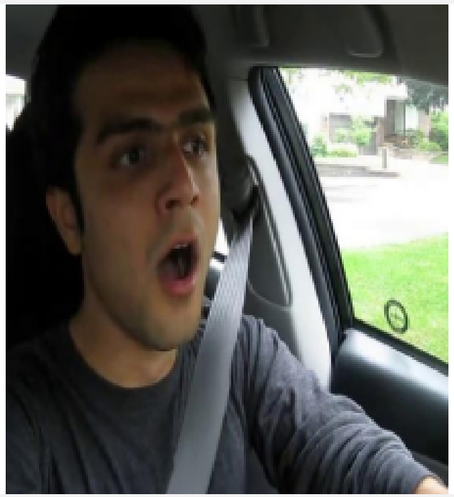
Teacher Home Page

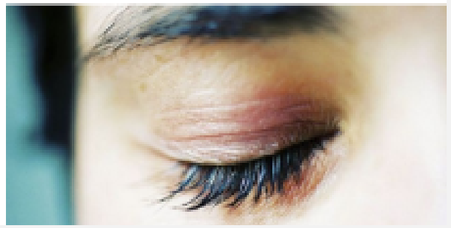
SYSTEM UI



Teacher Dashbord

Data Collection and Pre-processing





Emotional recognition model

Classes	Test	Train
angry	793	3965
disgust	104	524
fear	813	4067
happiness	204	1024
sadness	248	1240
suprices	268	1340
neutral	200	1254

Drownsiness ditection model

Classes	Test	Train
Closed eyes	139	696
open eyes	142	710
yawn	136	680
no_yawn	238	693

Data Collection and Pre-processing

EDUSMART

```
RESTED_LABELS = [1,2,3, 4,5,6,7]
 df[df.emotion.isin(INTERESTED_LABELS)]

nvert pixel data into 48x48 grayscale image arrays
array = df.pixels.apply(lambda x: np.array(x.split(' ')).reshape(48, 48, 1).astype('float
array = np.stack(img_array, axis=0)

code labels using LabelEncoder and convert them to categorical format
 LabelEncoder()
labels = le.fit_transform(df.emotion)
labels = np_utils.to_categorical(img_labels)

p labels to their corresponding classes
ame_mapping = dict(zip(le.classes_, le.transform(le.classes_)))
t(le_name_mapping)

lit data into training and validation sets
ain, X_valid, y_train, y_valid = train_test_split(img_array, img_labels,
                                                  shuffle=True, stratify=img_labels,
                                                  test_size=0.1, random_state=42)

rmalize pixel values to [0, 1] range
ain = X_train / 255.
lid = X_valid / 255.

splay shapes of the training and validation sets
t(X_train.shape, X_valid.shape, y_train.shape, y_valid.shape)
```

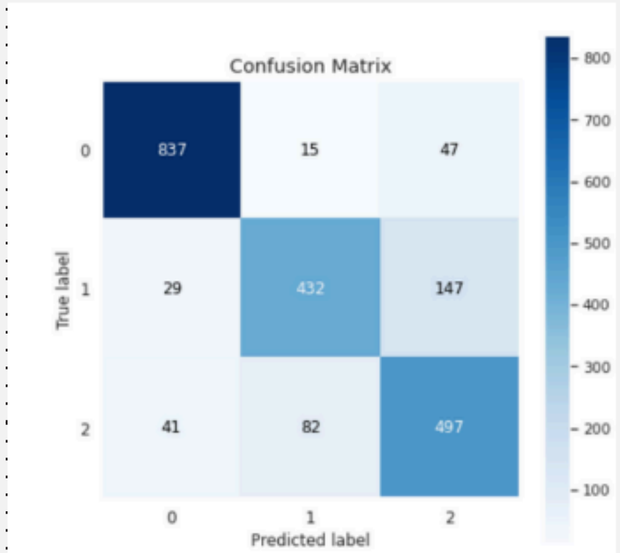
- Data pre-processing implemetation

```
from tensorflow.keras.preprocessing.image import Im

# Data Augmentation: Apply random transformations t
train_datagen = ImageDataGenerator(
    rotation_range=15,              # Random rotation
    width_shift_range=0.15,         # Random horizonta
    height_shift_range=0.15,        # Random vertical
    shear_range=0.15,               # Random shear tra
    zoom_range=0.15,                # Random zoom with
    horizontal_flip=True,           # Random horizonta
)

# Fit the ImageDataGenerator on the training data
train_datagen.fit(X_train)
```

- Data augmentation

Data Collection and Pre-processing

EDUSMART




```
ax = pyplot.subplot(1, 2, 1)
sns.lineplot(history.epoch, history.history['accuracy'], label='train')
sns.lineplot(history.epoch, history.history['val_accuracy'], label='valid')
pyplot.title('Accuracy')
pyplot.tight_layout()

ax = pyplot.subplot(1, 2, 2)
sns.lineplot(history.epoch, history.history['loss'], label='train')
sns.lineplot(history.epoch, history.history['val_loss'], label='valid')
pyplot.title('Loss')
pyplot.tight_layout()
```

Confusion Matrix

```
from sklearn.metrics import classification_report
print(classification_report(np.argmax(y_test, axis=1), pr
```

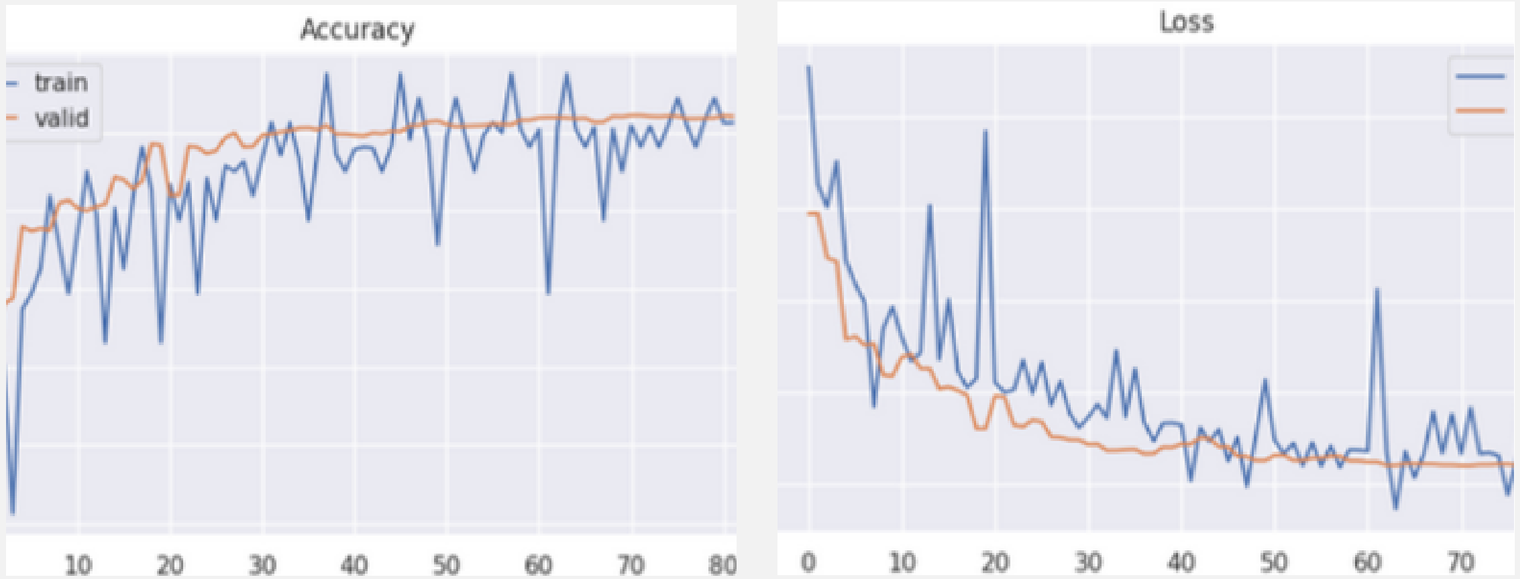
	precision	recall	f1-score	support
yawn	0.89	0.86	0.87	63
no_yawn	0.88	0.93	0.91	74
Closed	1.00	0.90	0.95	215
Open	0.92	1.00	0.96	226
accuracy			0.94	578
macro avg	0.92	0.92	0.92	578
weighted avg	0.94	0.94	0.94	578

Classification Report

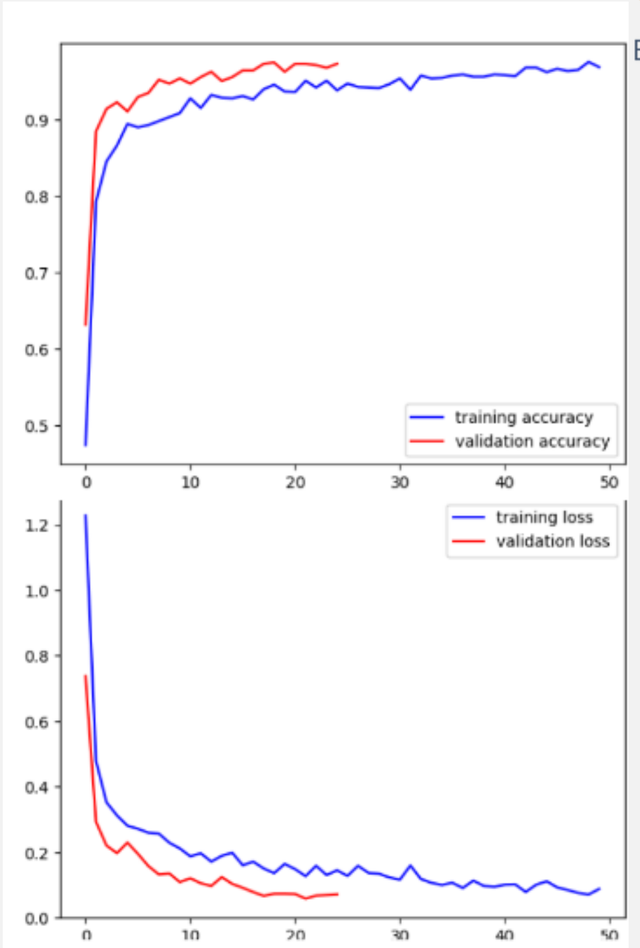
Feature map

Data Collection and Pre-processing

EDUSMART



Emotional ditection accuracy chart



Drownsiness ditection accuracy chart

Completion and Future works

EDUSMART

Completion of the components

- Identification of best architecture for transfer learnin
- Identification of students drownsiness using yawning and eye_sleep classes
- Identification of students facial emotional reactions using seven classes
- Develop a teacher interface for real-time monitoring
- Integration with Learning Management Systems (LMS)

Future Implementations

- Generae full Srudent Report
- Enhance Result Accuracy

IT21158254 || WIJENANDANA S.D

EDUSMART

COMPONENT 02

Automated Quiz Generation and Evaluation : Enhancing Assessment Accuracy and Adaptive Feedback

INTRODUCTION

BACKGROUND

01. Growth of Online Learning:

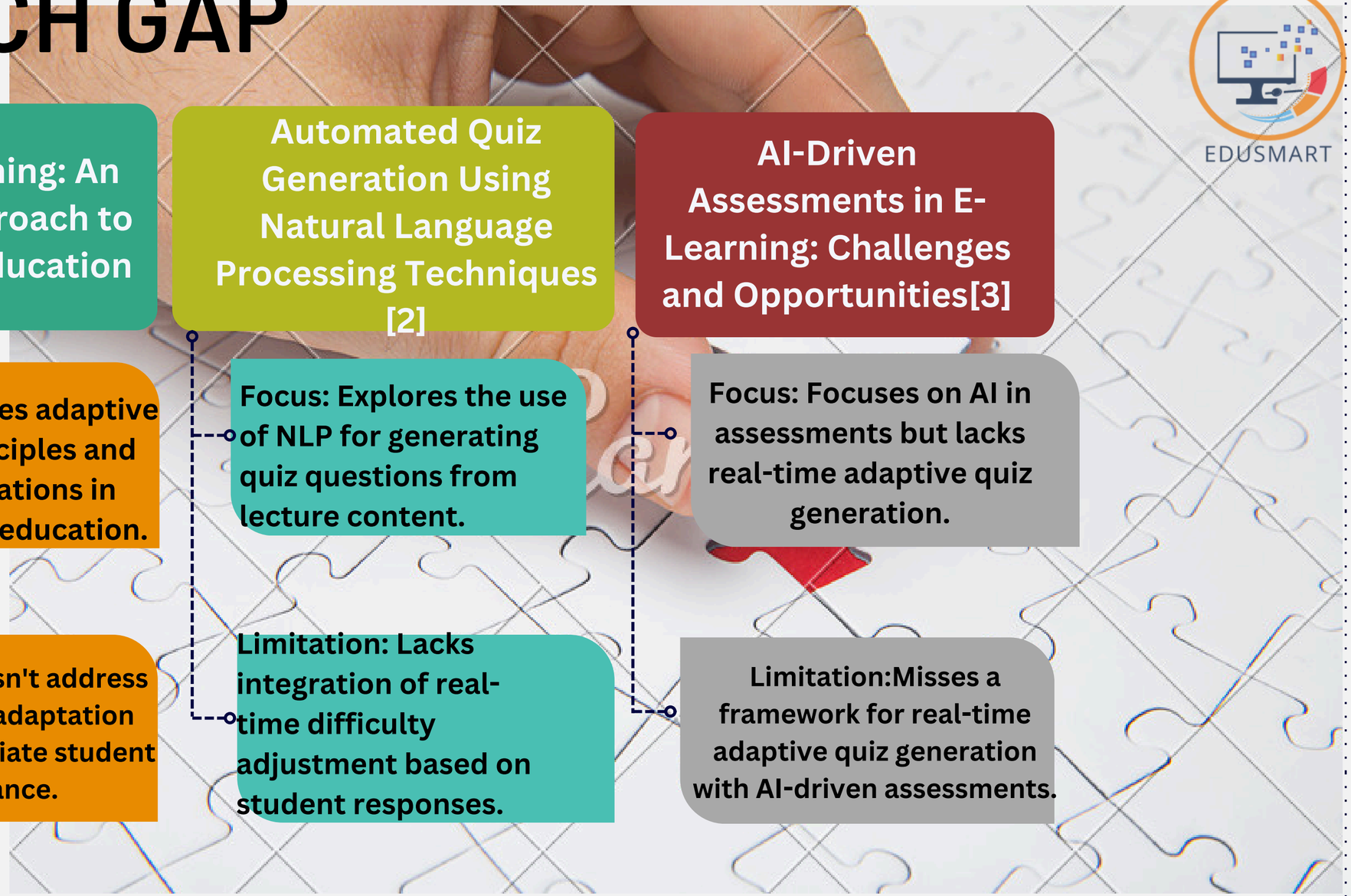
02. Limitations of Current Assessment Methods:

03. Advancements in AI and NLP:

04. Importance of Adaptive Learning:

EDUSMART

RESEARCH GAP



Adaptive Learning: An Innovative Approach to Personalized Education [1]

- Focus: Discusses adaptive learning principles and their applications in personalizing education.
- Limitation: Doesn't address real-time quiz adaptation based on immediate student performance.

Automated Quiz Generation Using Natural Language Processing Techniques [2]

- Focus: Explores the use of NLP for generating quiz questions from lecture content.
- Limitation: Lacks integration of real-time difficulty adjustment based on student responses.

AI-Driven Assessments in E-Learning: Challenges and Opportunities[3]

- Focus: Focuses on AI in assessments but lacks real-time adaptive quiz generation.
- Limitation:Misses a framework for real-time adaptive quiz generation with AI-driven assessments.

RESEARCH GAP

EDUSMART

FEATURES	[1]	[2]	[3]	[4]	EDU-SMART
Adaptability	✓	✗	✗	✓	✓
Dynamic Question Adjustments	✗	✓	✓	✗	✓
Data Privacy	✗	✗	✓	✗	✓
Integration with LMS	✗	✗	✗	✗	✓

RESEARCH PROBLEM

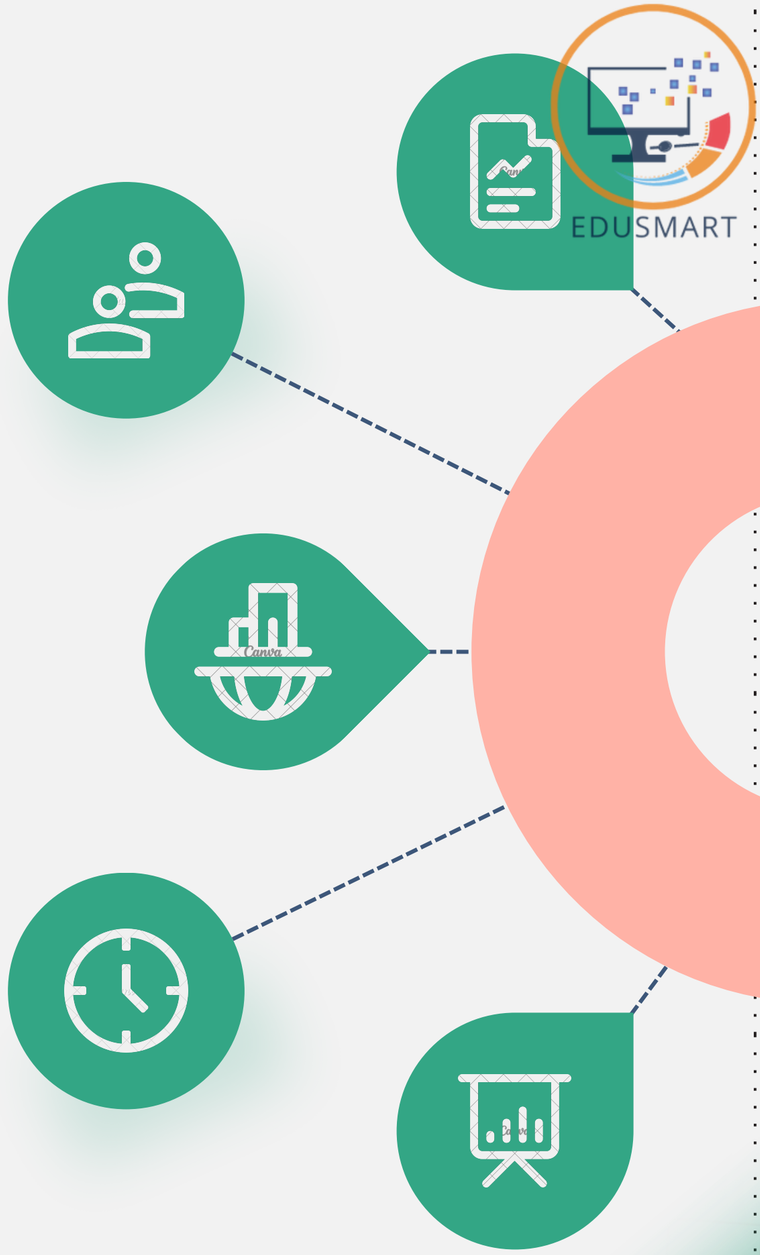
EDUSMART

- How accurately can real-time automated quiz generation assess student understanding from lecture content?
- How effectively can the system adapt question difficulty based on student performance?
- How can personalized reports generated by the system enhance learning outcomes?

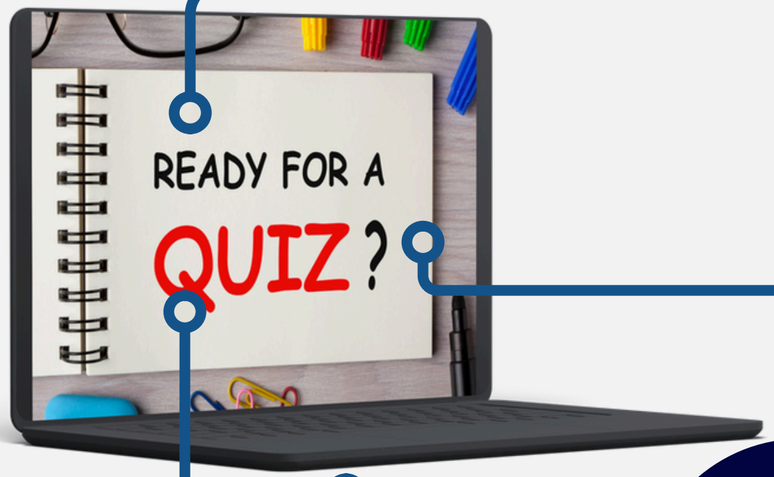
OBJECTIVES

MAIN OBJECTIVE

To develop and evaluate a real-time automated quiz generation system that accurately creates and adapts questions based on lecture content and student performance



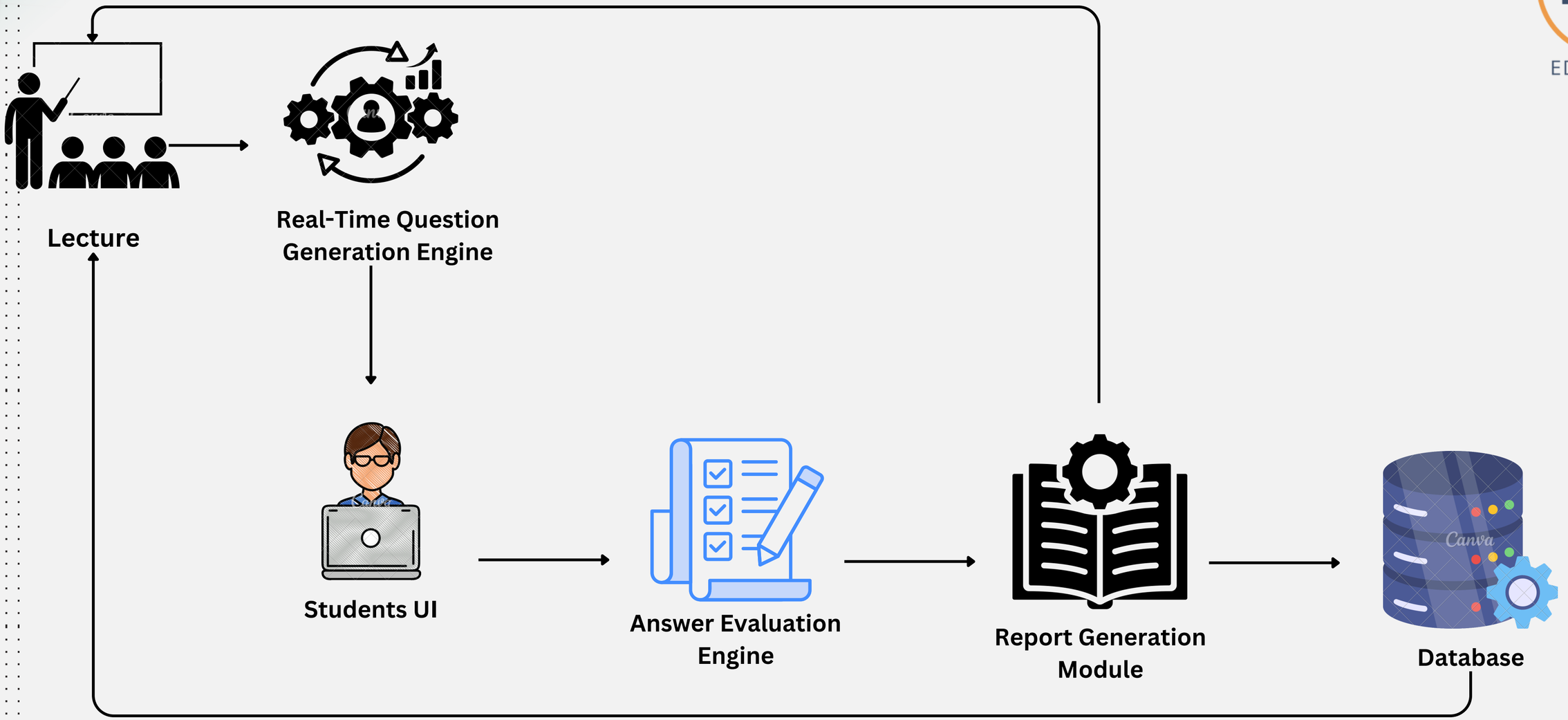
SUB OBJECTIVE



- Develop Classification Algorithms
- Enhance Question Relevance
- Optimize Performance
- Ensure Seamless Integration

METHODOLOGY

Component Overview Diagram



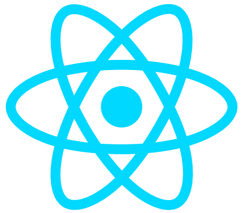
Methodology Evidence of Completion

EDUSMART

- Data collection
- Data pre-processing
- Fine Tune T5 model using question and answer dataset
- Normalized PDF to text and predict question and answer based on context
- Data Visualization
- Output validation
- Implementation Tools

Technologies

1. Transformers
2. Python
3. TensorFlow
4. Jupyter Notebook
5. Google Colab
6. VSCode
7. React



Visual Studio Code

Techniques

1. Normalization
2. Model finetuning
3. Data preprocessing

Data preprocessing, model loading and model training

EDUSMART

```
!pip install datasets

from transformers import T5Tokenizer, T5ForConditionalGeneration, Trainer, TrainingArguments
from datasets import load_dataset, DatasetDict

model_name = "t5-small"
tokenizer = T5Tokenizer.from_pretrained(model_name)
model = T5ForConditionalGeneration.from_pretrained(model_name)

dataset = load_dataset("squad")
```

```
def preprocess_data(examples):
    inputs = ["generate question: " + context for context in examples["context"]]
    targets = [question for question in examples["question"]]
    model_inputs = tokenizer(inputs, max_length=512, truncation=True, padding="max_length")
    labels = tokenizer(targets, max_length=128, truncation=True, padding="max_length").input_ids
    model_inputs["labels"] = labels
    return model_inputs
```

```
[ ] tokenized_datasets = dataset.map(preprocess_data, batched=True)
```

```
[ ] train_size = int(len(tokenized_datasets["train"]))
    validation_size = int(len(tokenized_datasets["validation"]))

    train_dataset = tokenized_datasets["train"].select(range(train_size))
    validation_dataset = tokenized_datasets["validation"].select(range(validation_size))
```

```
training_args = TrainingArguments(
    output_dir="./results",
    evaluation_strategy="epoch",
    save_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=8,
    num_train_epochs=3,
    save_steps=10_000,
    save_total_limit=2,
    logging_dir="./logs",
    logging_steps=50,
    load_best_model_at_end=True,
    metric_for_best_model="eval_loss",
    report_to="none"
)

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=validation_dataset,
    tokenizer=tokenizer,
)


train_output = trainer.train()
```

Text normalizing and PDF Extracting

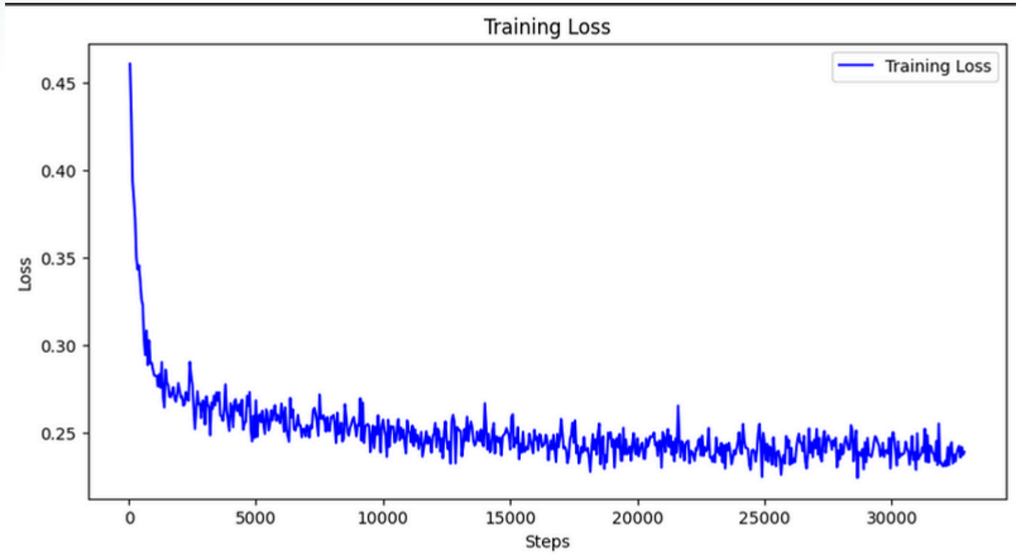
EDUSMART

```python
def extract_text_from_pdf(pdf_path):
    pdf_reader = PdfReader(pdf_path)
    text = ""
    for page in pdf_reader.pages:
        text += page.extract_text()
    return text
```

```python
def generate_questions(chunk, model, tokenizer):
    input_text = f"generate question: {chunk}"

    inputs = tokenizer(input_text, return_tensors='pt', padding=True, truncation=True)

    outputs = model.generate(inputs['input_ids'], max_length=150, num_beams=5, early_stopping=True, temperature=0

    question = tokenizer.decode(outputs[0], skip_special_tokens=True)
    return question
```

```python
def predict(context, query):
    input_text = f"question: {query} context: {context}"

    inputs = tokenizer(input_text, return_tensors='pt', padding="max_length", truncation=True)

    outputs = model.generate(inputs['input_ids'], max_length=1024, num_beams=8, early_stopping=True, temperature=

    answer = tokenizer.decode(outputs[0], skip_special_tokens=True)

    if answer == query:
        answer = "The answer is unclear, please try again."

    return answer
```

```python
def normalize_text(s):
    def remove_articles(text):
        regex = re.compile(r"\b(a|an|the)\b", re.UNICODE)
        return re.sub(regex, " ", text)

    def white_space_fix(text):
        return " ".join(text.split())

    def remove_punc(text):
        exclude = set(string.punctuation)
        return "".join(ch for ch in text if ch not in exclude)

    def lower(text):
        return text.lower()

    return white_space_fix(remove_articles(remove_punc(lower(s))))
```


Training Loss
Training Loss
0.45
0.40
0.35
0.30
0.25
Loss
0
5000
10000
15000
20000
25000
30000
Steps


Validation Loss
Validation Loss
0.229
0.228
0.227
0.226
0.225
0.224
Loss
10000
15000
20000
25000
30000
Steps

EDUSMART

Completion of the components

1. Finetune T5 model
2. Extract PDF
3. Normalize PDF to text
4. Train model using sample dataset

60%

Future Works

1. Train model with specified dataset
2. Generate Report
3. Confirming Data Privacy

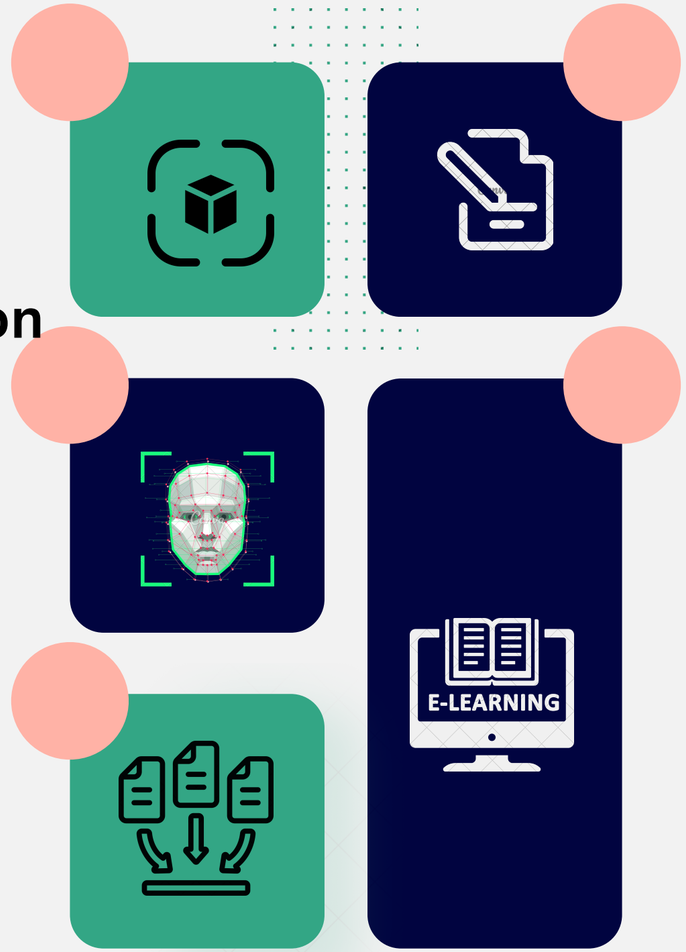
40%

COMPONET SPECIFIC REQUIREMENTS

EDUSMART

FUNCTIONAL REQUIREMENT

- Lecture Upload
- Real-Time Question Generation
- Student Interface
- Answer Evaluation
- Report Generation



NON-FUNCTIONAL REQUIREMENT

- Performance
- Scalability
- Usability
- Reliability
- Security

EDUSMART

REFERENCE

1. Brown, E., & Glassner, A. (2020). Adaptive Learning: An Innovative Approach to Personalized Education. Journal of Educational Technology, 12(3), 45-59.
"Discusses the principles of adaptive learning and its applications in modern education"

2. Chen, X., & Cheng, G. (2021). Automated Quiz Generation Using Natural Language Processing Techniques. International Journal of Artificial Intelligence in Education, 31(2), 203-220.
"Explores the use of NLP for generating quiz questions based on lecture content."

3. Li, Y., & Zhang, H. (2019). AI-Driven Assessments in E-Learning: Challenges and Opportunities. Computers & Education, 145, 103-112.
"Provides insights into the role of AI in enhancing the assessment process in e-learning environments."

REFERENCE

4. Johnson, D., & Ramirez, S. (2020). Personalized Learning Paths: The Future of Online Education. Journal of Learning Analytics, 8(4), 134-150.
"Discusses the importance of personalized learning paths in online education and how they can improve student outcomes."

IT21185502 | EKANAYAKA H.E.M.P.L

Information Technology

EDUSMART

COMPONENT 03

Unauthorized Object Detection and Head Pose Estimation Techniques and Image Verification Into E-Learning Platform

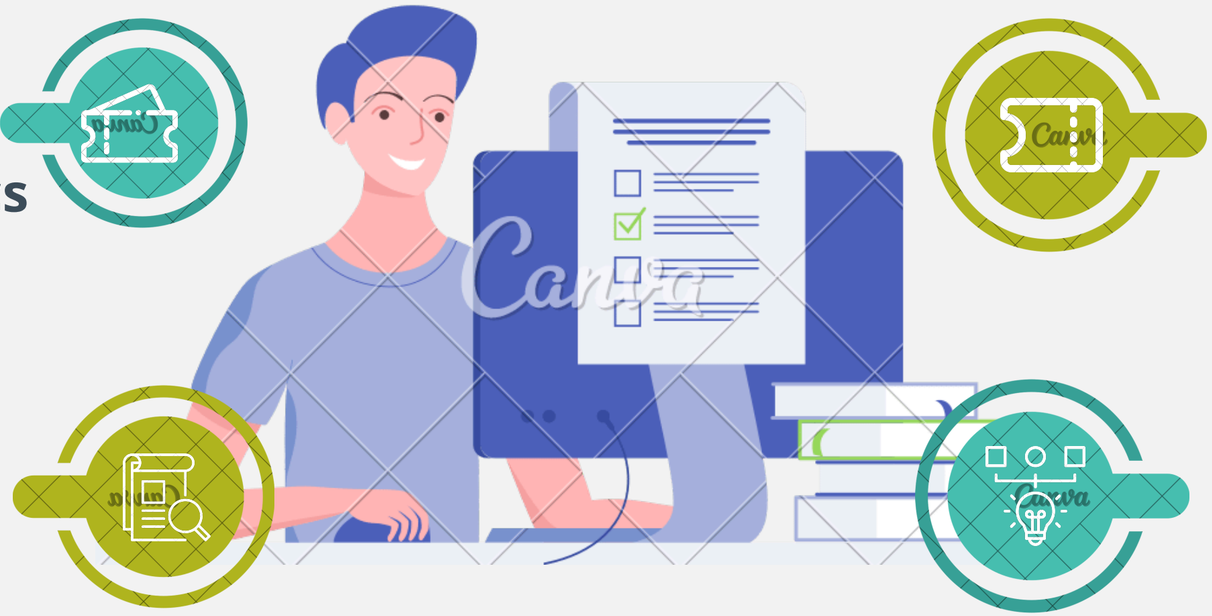
INTRODUCTION

BACKGROUND

EDUSMART

01. Traditional Classroom Dynamics

02. Challenges in Online Exam



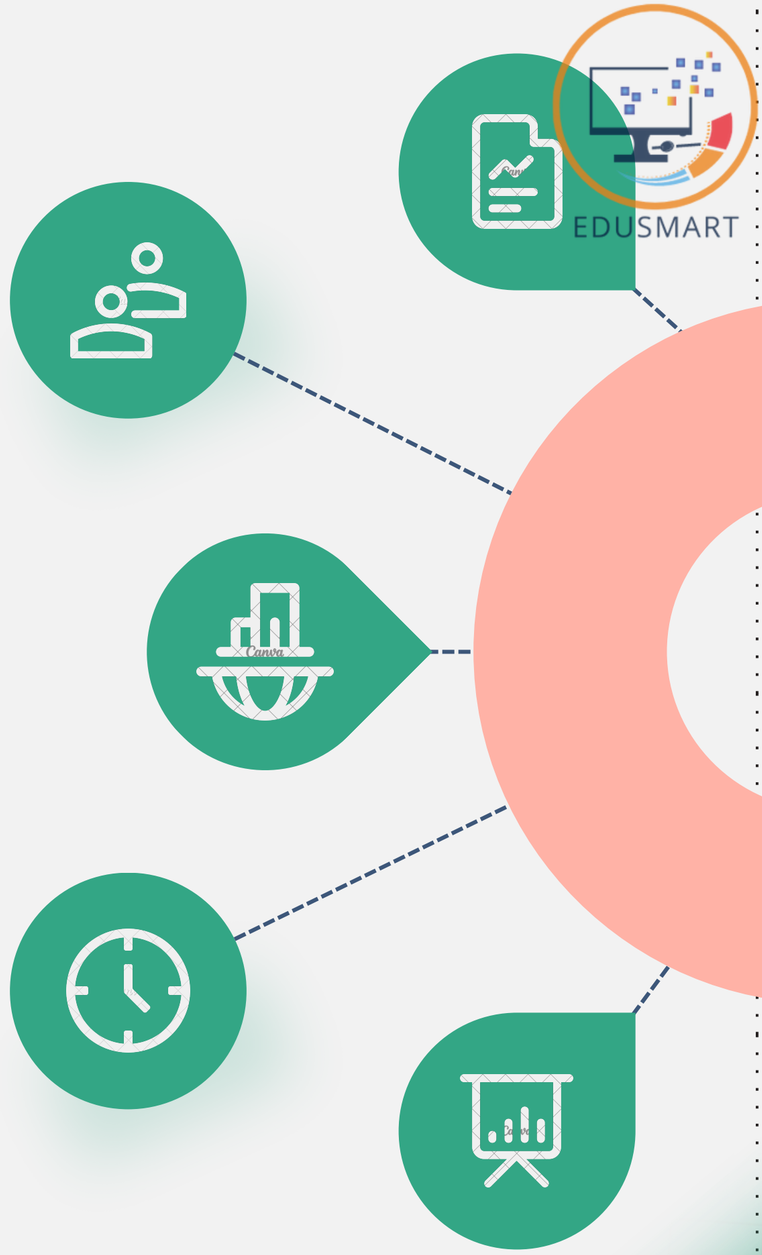
03. Infeasibility of Traditional Proctoring

04. Technological Solutions

OBJECTIVES

MAIN OBJECTIVE

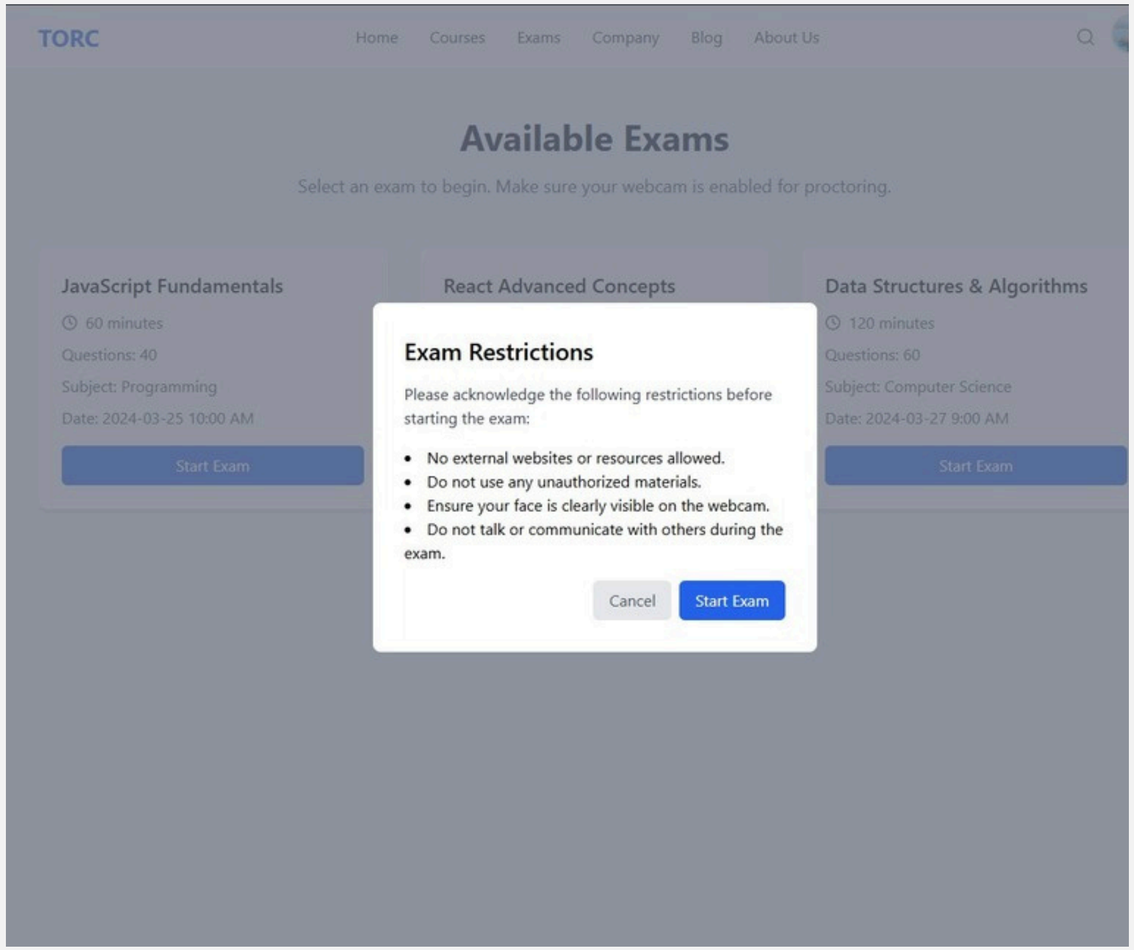
Develop a cutting-edge real-time monitoring system that utilizes advanced computer vision and machine learning techniques to accurately detect and analyze head poses and unauthorized objects during online examinations, ensuring academic integrity and enhancing assessment reliability.

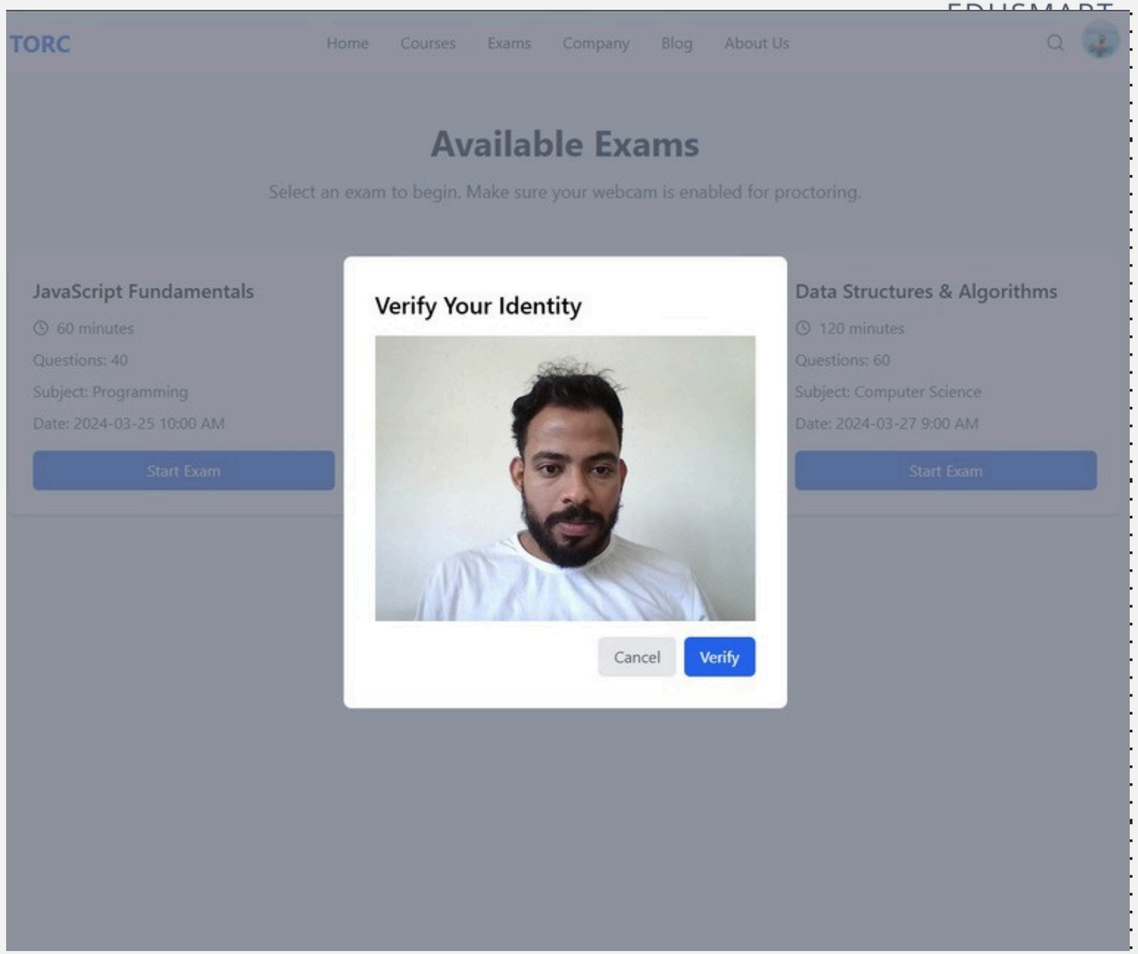


SUB OBJECTIVE

- Integration with Learning Management Systems (LMS)
- Real Time Notify to Lecturer unauthorized object and unusual Head movement
- Design and Development of Detection Algorithms
- System's Final Reporting on Student Exam Performance

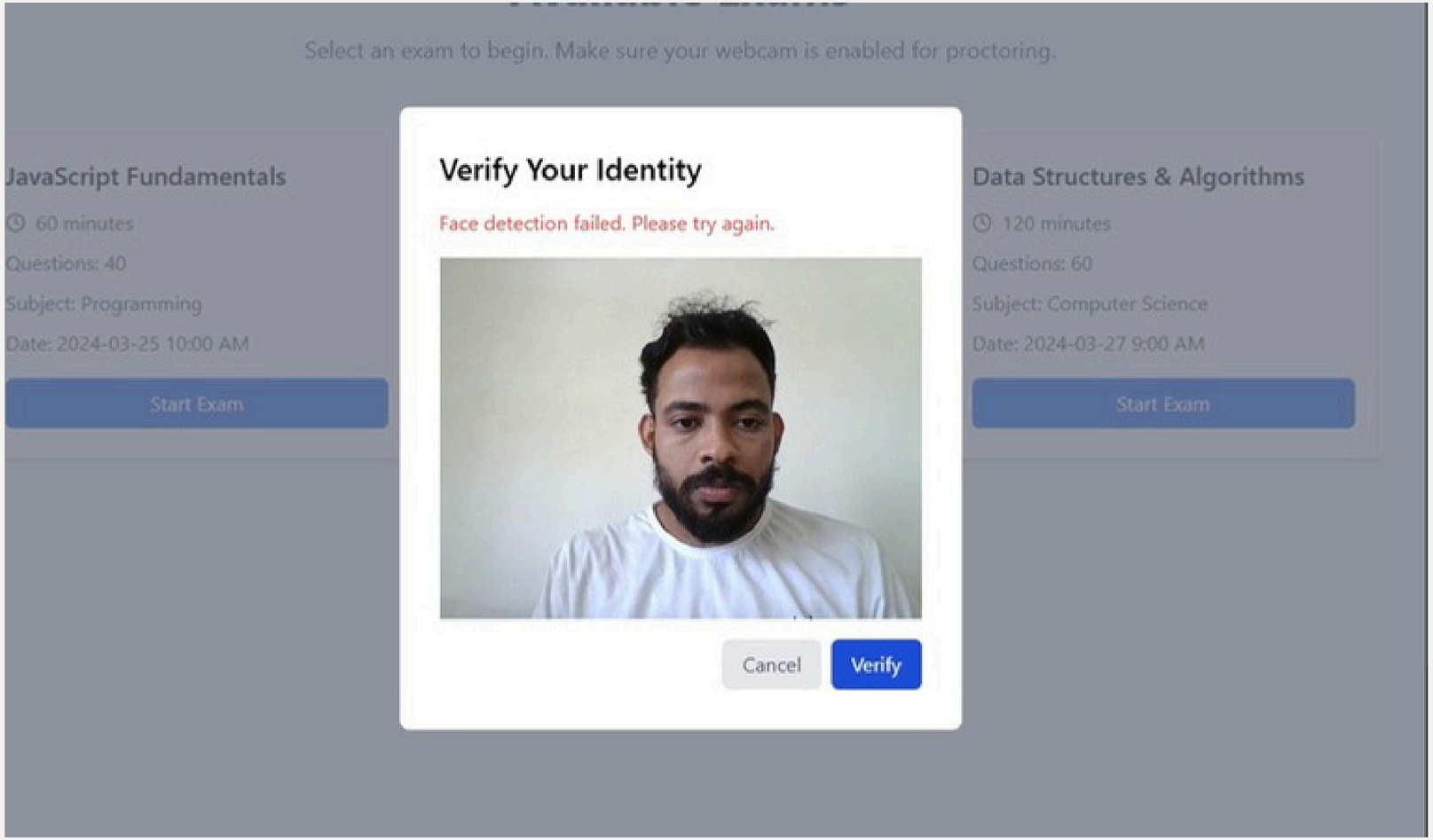
SYSTEM UI


TORC
Home
Courses
Exams
Company
Blog
About Us
Available Exams
Select an exam to begin. Make sure your webcam is enabled for proctoring.
JavaScript Fundamentals
60 minutes
Questions: 40
Subject: Programming
Date: 2024-03-25 10:00 AM
Start Exam
React Advanced Concepts
Data Structures & Algorithms
120 minutes
Questions: 60
Subject: Computer Science
Date: 2024-03-27 9:00 AM
Start Exam
Exam Restrictions
Please acknowledge the following restrictions before starting the exam:
No external websites or resources allowed.
Do not use any unauthorized materials.
Ensure your face is clearly visible on the webcam.
Do not talk or communicate with others during the exam.
Cancel
Start Exam

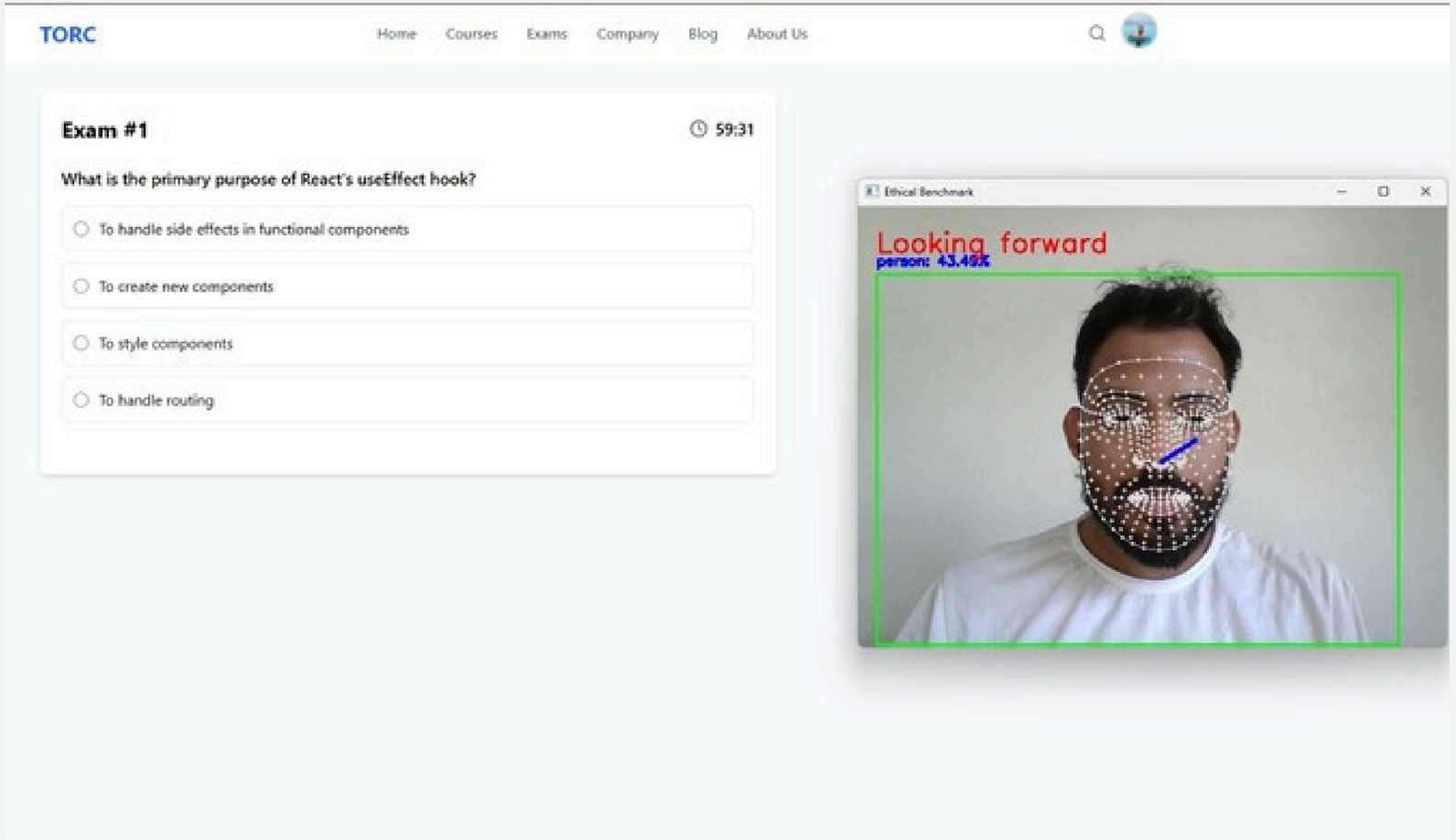

TORC
Home
Courses
Exams
Company
Blog
About Us
Available Exams
Select an exam to begin. Make sure your webcam is enabled for proctoring.
JavaScript Fundamentals
60 minutes
Questions: 40
Subject: Programming
Date: 2024-03-25 10:00 AM
Start Exam
Verify Your Identity
Cancel
Verify
Data Structures & Algorithms
120 minutes
Questions: 60
Subject: Computer Science
Date: 2024-03-27 9:00 AM
Start Exam

SYSTEM UI

EDUSMART

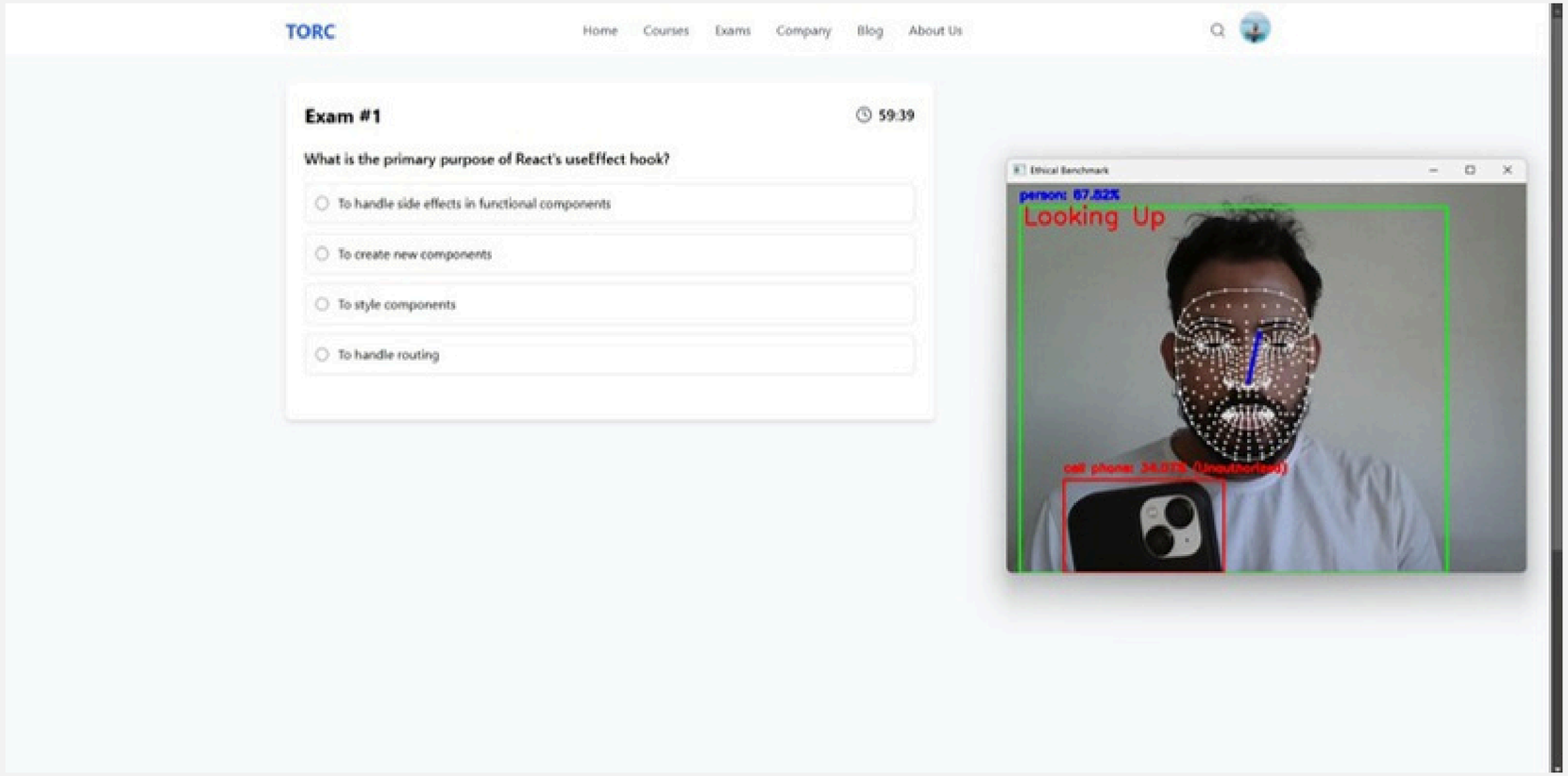


SYSTEM UI

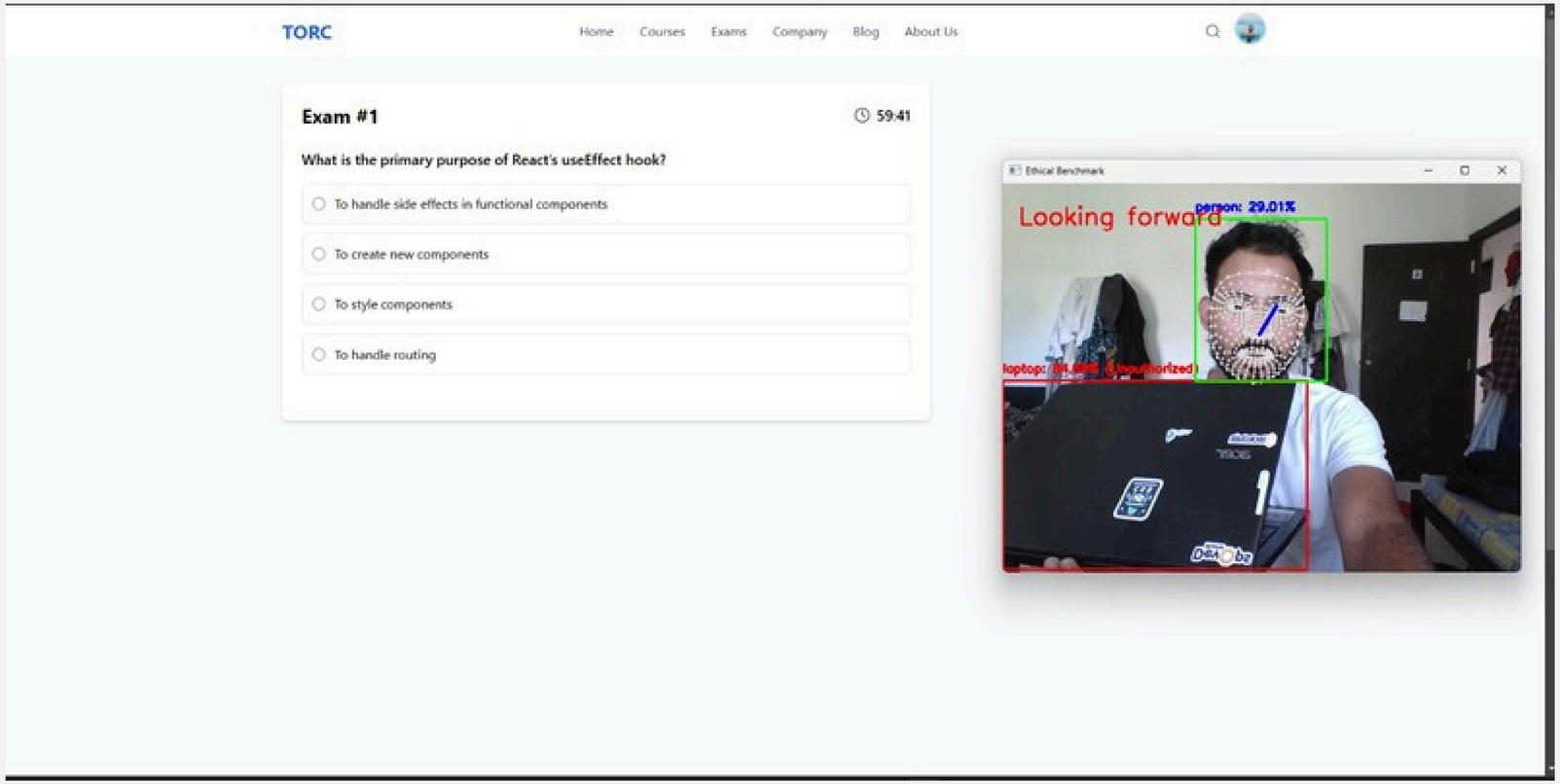

TORC
Home
Courses
Exams
Company
Blog
About Us
Exam #1
59:31
What is the primary purpose of React's useEffect hook?
To handle side effects in functional components
To create new components
To style components
To handle routing
Ethical Benchmark
Looking forward
person: 43.49%

SYSTEM UI

EDUSMART



SYSTEM UI

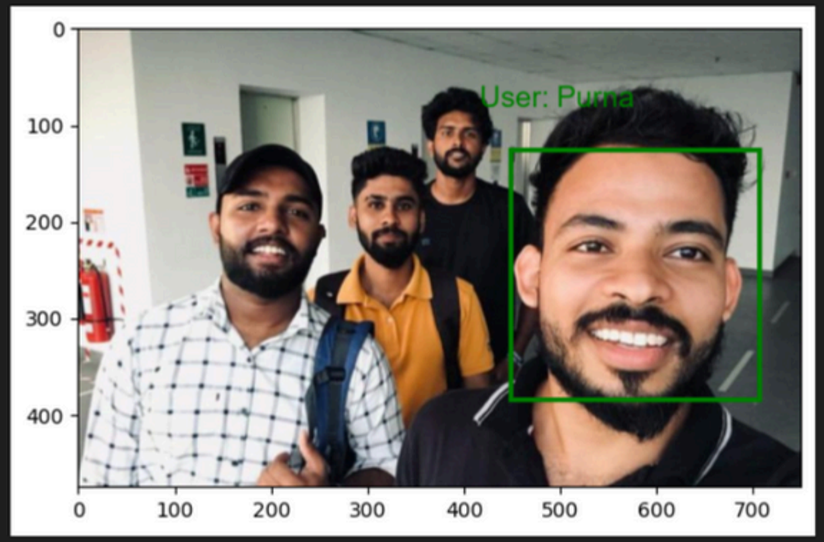

TORC
Home
Courses
Exams
Company
Blog
About Us
Exam #1
59:41
What is the primary purpose of React's useEffect hook?
To handle side effects in functional components
To create new components
To style components
To handle routing
Ethical Benchmark
Looking forward
person: 29.01%

❖ Data Visualization

EDUSMART

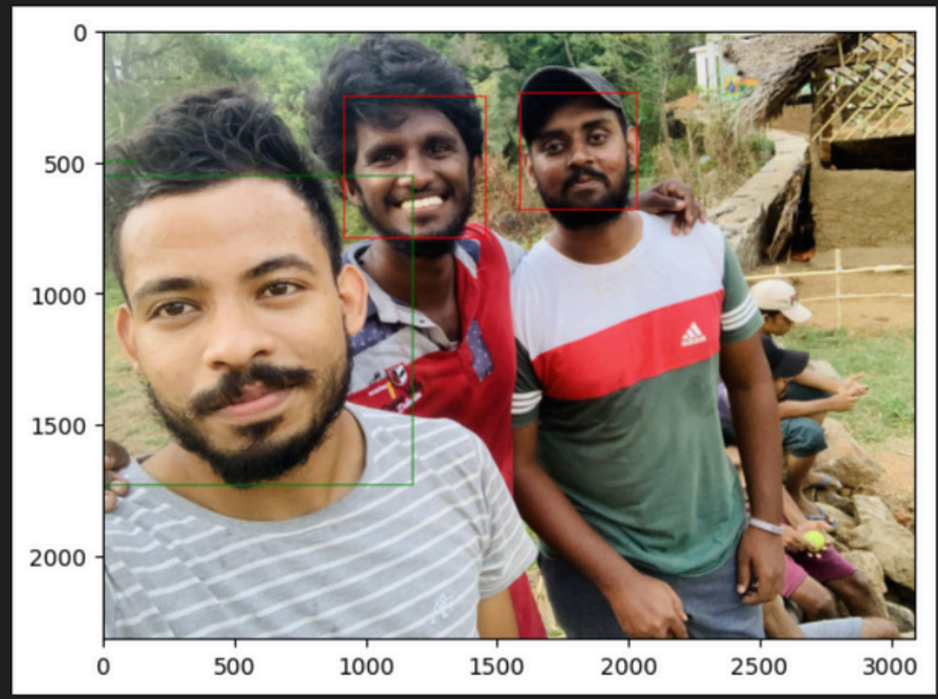
face varification accuracy

```
verify_and_visualize_user("Purna", 'data/test_images/purna14.jpg')
[22] ✓ 1.7s
```



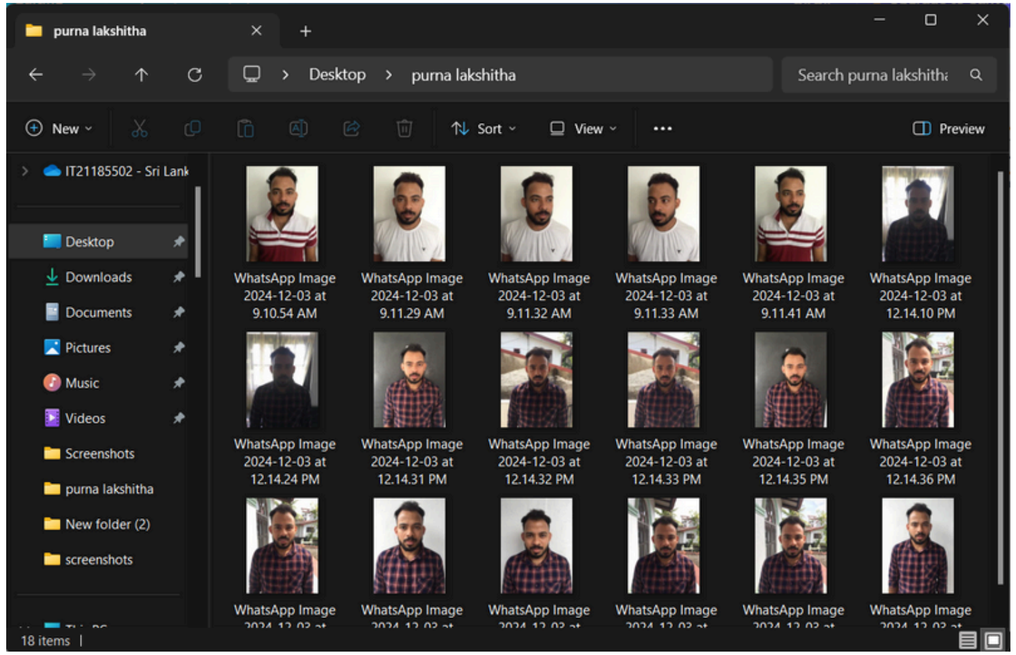

```
verify_and_visualize_user_realtime("Purna")
[20] ✓ 1m 2.5s
```

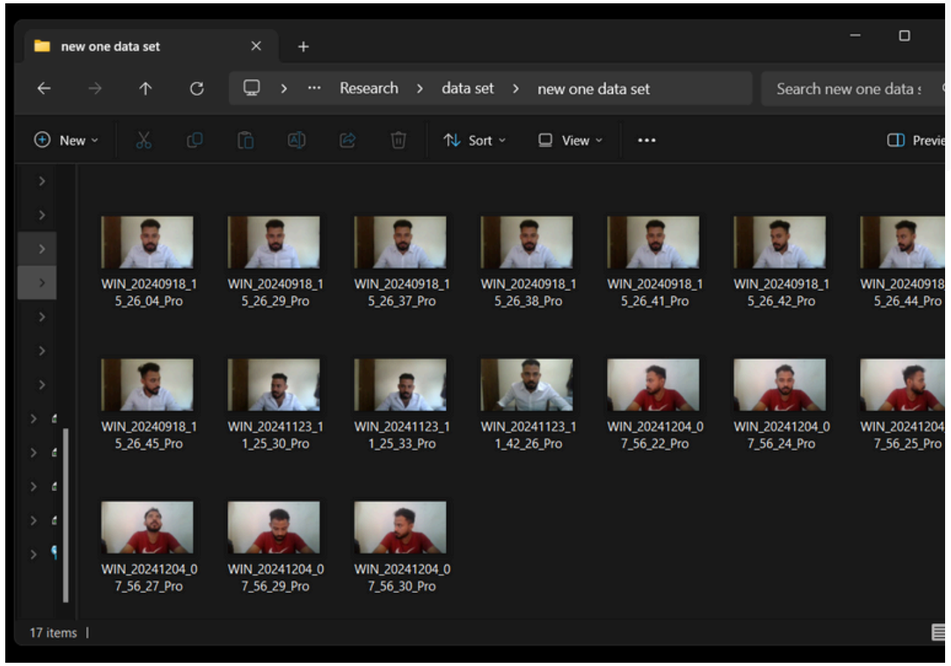
```
verify_and_visualize_user("Purna", 'data/test_images/purna33.jpg')
[25] ✓ 6.6s
```

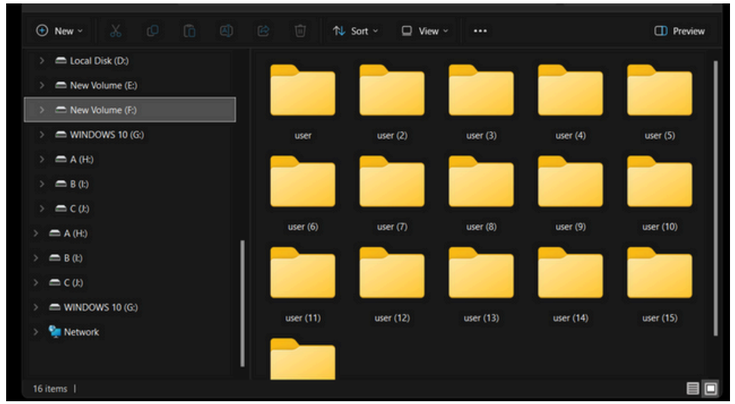



```
verify_and_visualize_user_realtime("Purna")
```

❖ Data Collection







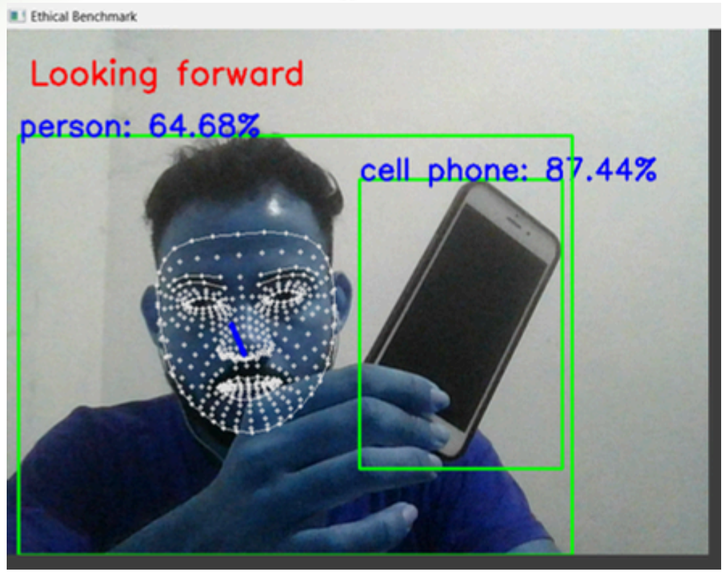
Create Dataset

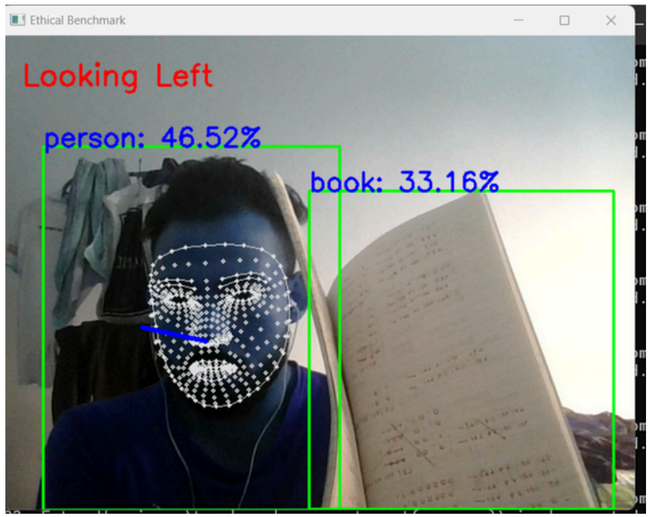
```python
def create_dataset(
                data_dir = "data/facedb/*/*.*",
                json_dir = "data/facedb_jsons/{}/{}",
                ):
    for img_path in glob.glob(data_dir):
        person_objs = []
        img_path = img_path.replace("\\", "/")
        person_name = img_path.split("/")[-2]
        file_name = img_path.split("/")[-1]
        face_objs = DeepFace.represent(
                                img_path = img_path,
                                model_name = "Facenet512",
                                enforce_detection = False
                                )
        if len(face_objs) > 0:
            for i in range(len(face_objs)):
                person_json = {}
                facial_area = face_objs[i]['facial_area']
                x, y, w, h = facial_area['x'], facial_area['y'], facial_area['w'], facial_area['h']
                person_json['person_name'] = person_name
                person_json['facial_area'] = facial_area
                person_json['file_name'] = file_name
                person_json['coor'] = (x, y, w, h)
                person_objs.append(person_json)

        if len(person_objs) > 0:
```

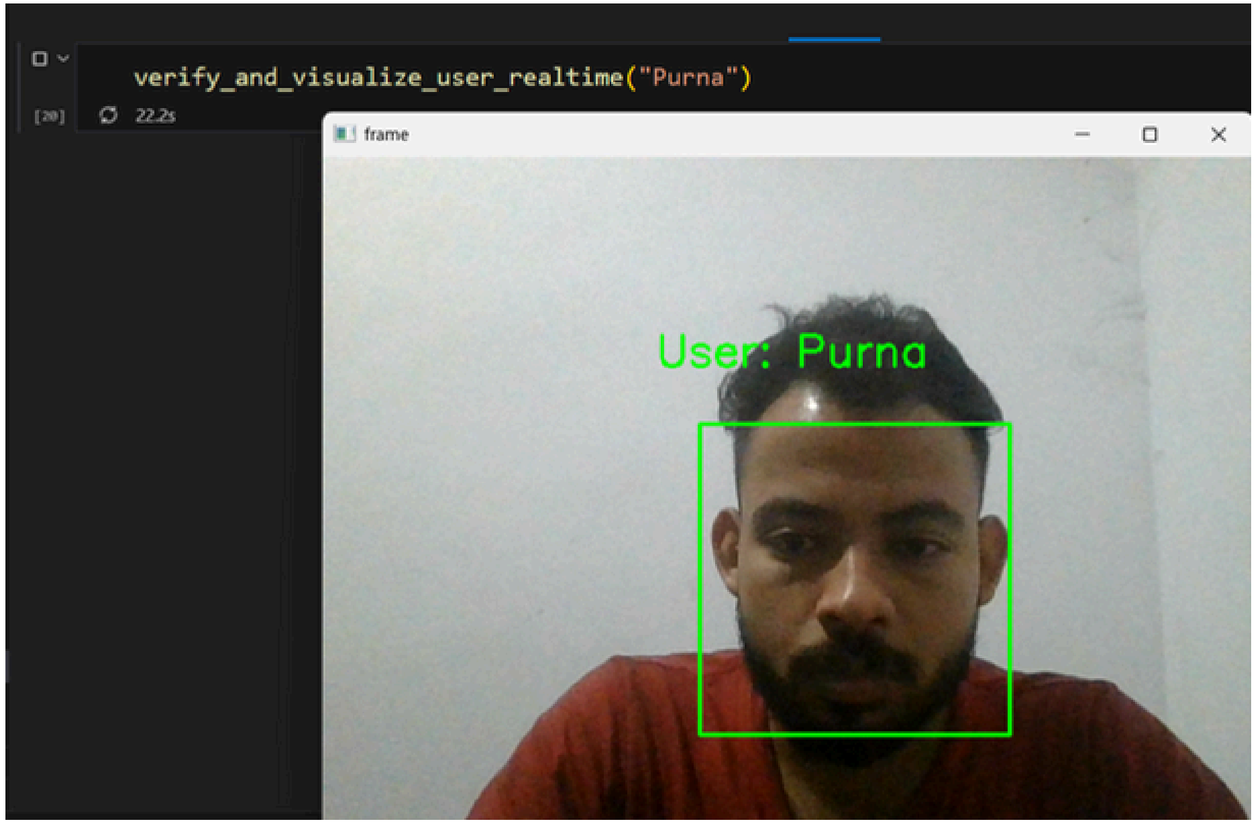
```
create_dataset()
```

```
24-12-03 17:29:55 - facenet512_weights.h5 will be downloaded...
Downloading...
From: https://github.com/serengil/deepface_models/releases/download/v1.0/facenet512_weights.h5
To: C:\Users\purna\.deepface\weights\facenet512_weights.h5
100%|██████████| 95.0M/95.0M [00:04<00:00, 20.0MB/s]
```

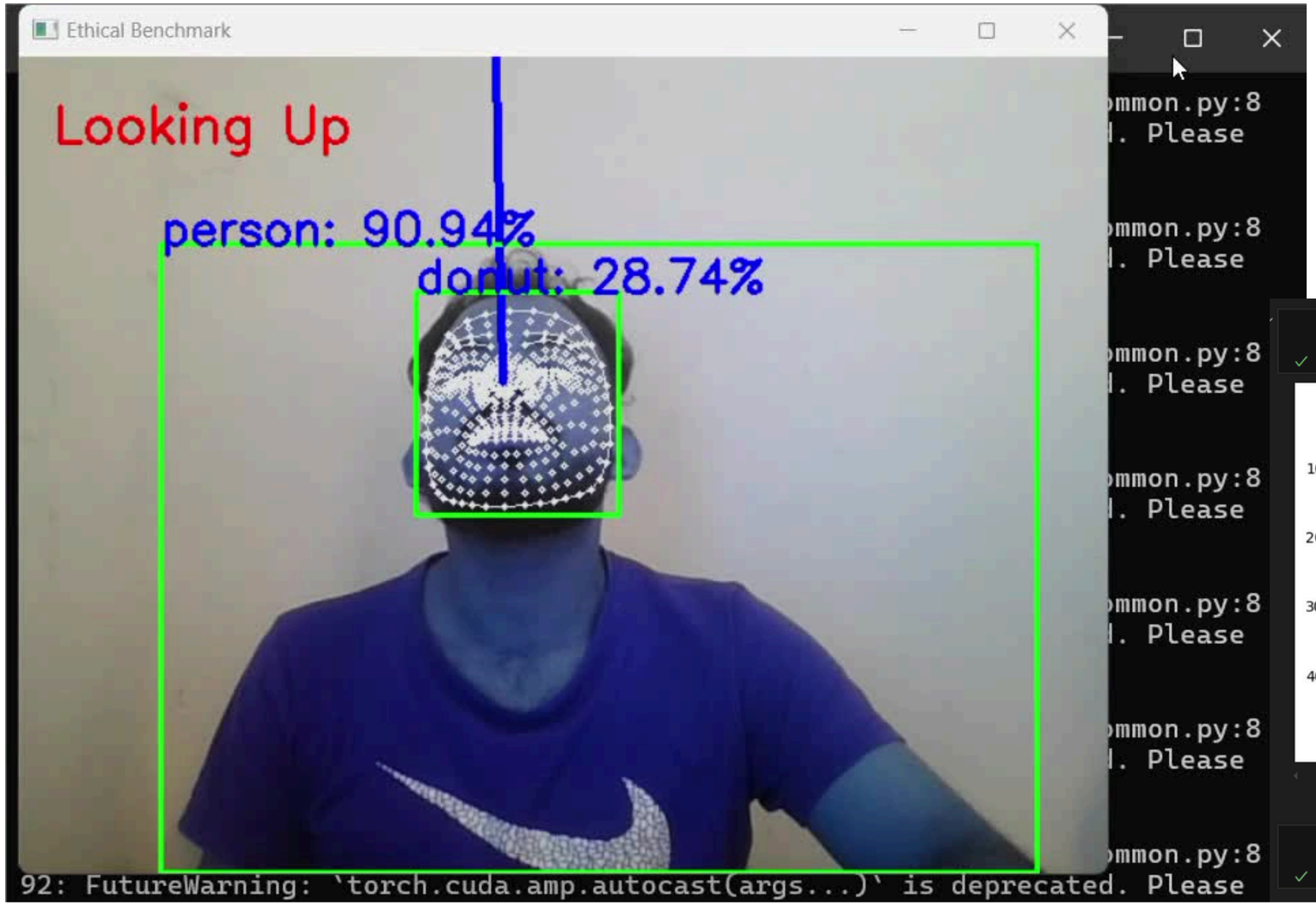
Display the results

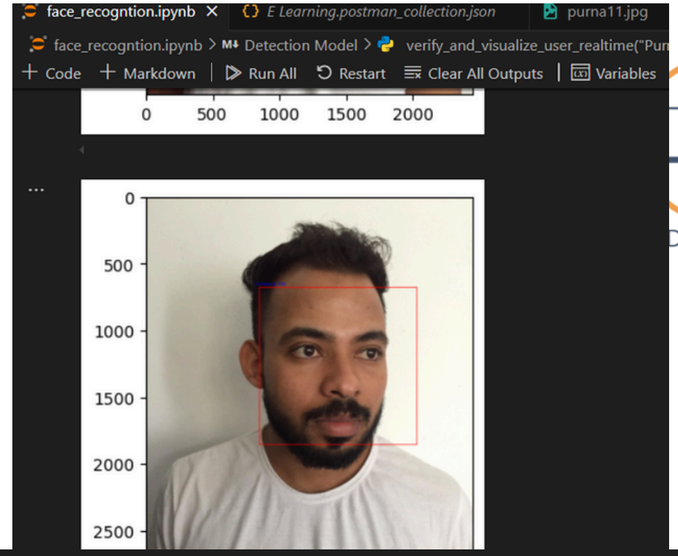


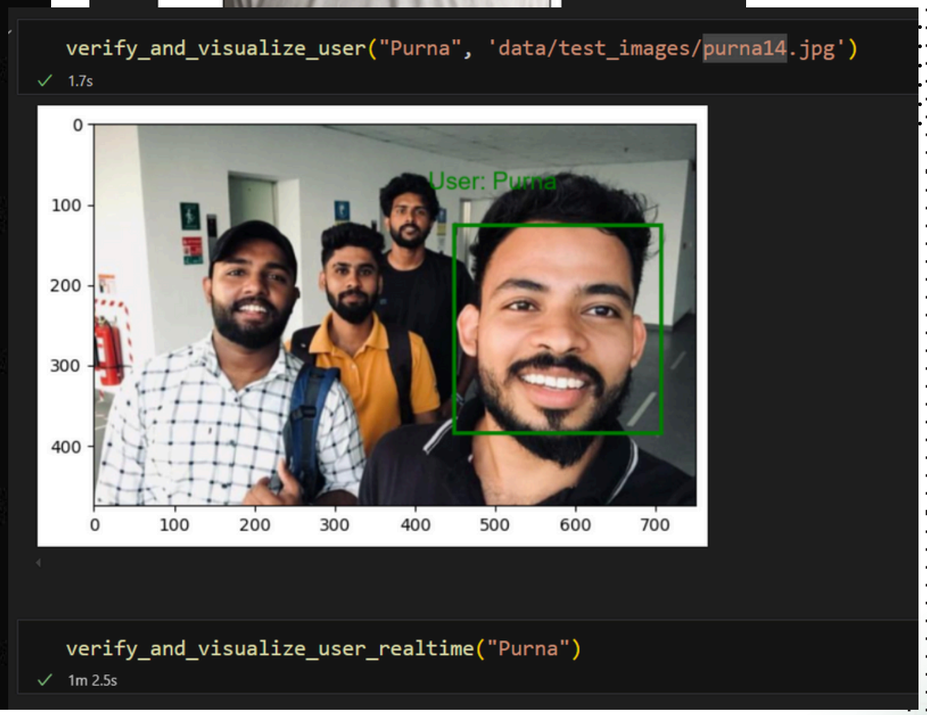




Display Result







Completion and Future works

EDUSMART

Completion of the components

- Identification of Objects in the frames
- Identification of students faces and save details in to a json file
- Identification of students various head poses

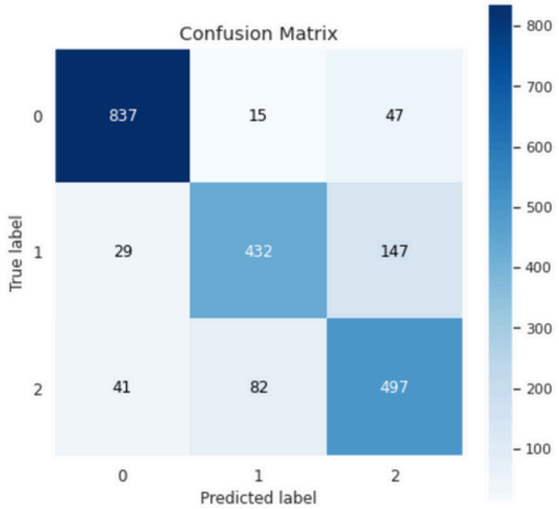
Future Implementations

- Categorize Unauthorized objects and Authorized Objects.
- Real time Alert sent to a lecturer
- Integration with Learning Management Systems (LMS)

❖ Data Visualization

EDUSMART

Confusion Matrix




```
ax = pyplot.subplot(1, 2, 1)
sns.lineplot(history.epoch, history.history['accuracy'], label='train')
sns.lineplot(history.epoch, history.history['val_accuracy'], label='valid')
pyplot.title('Accuracy')
pyplot.tight_layout()

ax = pyplot.subplot(1, 2, 2)
sns.lineplot(history.epoch, history.history['loss'], label='train')
sns.lineplot(history.epoch, history.history['val_loss'], label='valid')
pyplot.title('Loss')
pyplot.tight_layout()
```

Classification Report

```
from sklearn.metrics import classification_report
print(classification_report(np.argmax(y_test, axis=1), pre
```

	precision	recall	f1-score	support
yawn	0.89	0.86	0.87	63
no_yawn	0.88	0.93	0.91	74
Closed	1.00	0.90	0.95	215
Open	0.92	1.00	0.96	226
accuracy			0.94	578
macro avg	0.92	0.92	0.92	578
weighted avg	0.94	0.94	0.94	578

Feature map

REFERENCE

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580-587

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," Advances in Neural Information Processing Systems, vol. 28, 2015

[3] A. Sridhar and J. S. Rajshekhar, “AI-integrated Proctoring system for online exams,” J. Artif. Intell. Capsul. Networks, vol. 4, no. 2, pp. 139–148, 2022.

[4] M. Saleem Durai and N. Sreenu, "Scalability Challenges in Remote Exam Monitoring," Journal of Educational Technology Systems, vol. 45, no. 4, pp. 421-439, 2017.

IT21157950|| KARUANARATHNA J H H N

12/8/2024

EDUSMART

COMPONENT 04

Machine Learning Driven Automated Grading Platform Enhancing Fairness and Scalability in Educational Assessment

IT21157950 | Karunarathna J H H N | 24-25J-320

12/8/2024

INTRODUCTION

BACKGROUND

01 Technological Advancements

02 Challenges of Manual Grading

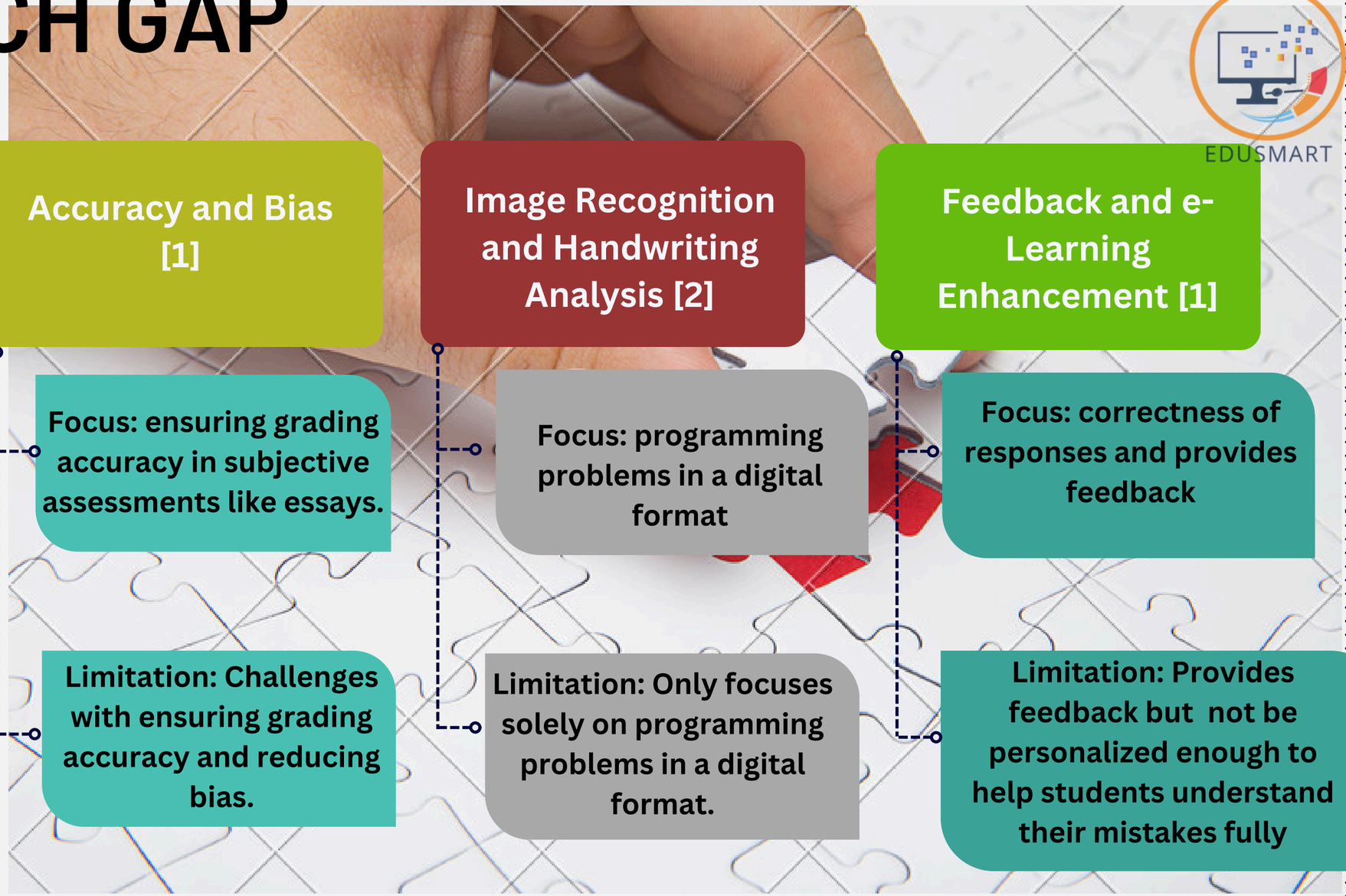
03 Challenges and Considerations

04 Data-Driven Insights

SLIIT FACULTY OF COMPUTING

EDUSMART

RESEARCH GAP



EDUSMART

Handling Synonyms and Contextual Variations [3]

- Focus: addresses synonyms In students handwritten assignments
- Limitation : not fully explore how contextual variations in student answers can be effectively graded.

Accuracy and Bias [1]

- Focus: ensuring grading accuracy in subjective assessments like essays.
- Limitation: Challenges with ensuring grading accuracy and reducing bias.

Image Recognition and Handwriting Analysis [2]

- Focus: programming problems in a digital format
- Limitation: Only focuses solely on programming problems in a digital format.

Feedback and e-Learning Enhancement [1]

- Focus: correctness of responses and provides feedback
- Limitation: Provides feedback but not be personalized enough to help students understand their mistakes fully

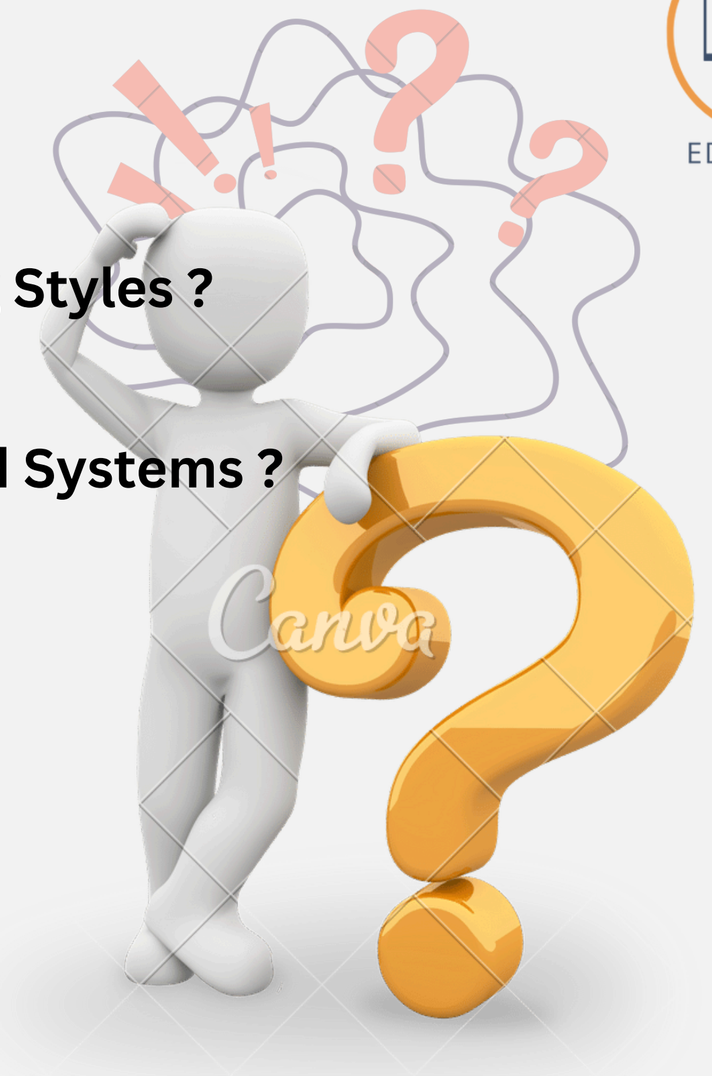
RESEARCH GAP

FEATURES	[1]	[2]	[3]	EDUSMART
Automated grading	✓	✓	✓	✓
Bias mitigation	✗	✓	✓	✓
Personalized feedback generation	✗	✓	✗	✓
OCR	✓	✗	✓	✓
handling contextual and synonyms variations	✓	✗	✗	✓

RESEARCH PROBLEM

EDUSMART

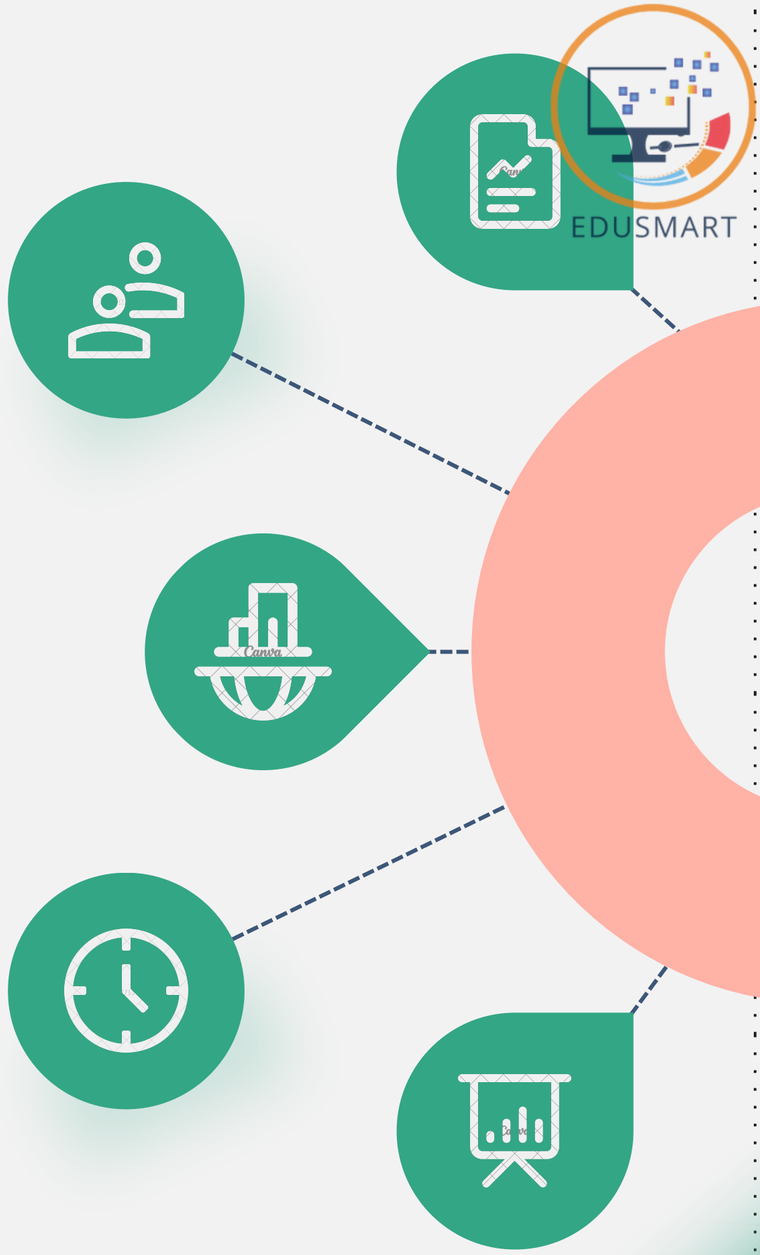
- Enhancing Model Accuracy for Diverse Writing Styles ?
- Ensuring Transparency and Trust in Automated Systems ?
- Impact on Student Learning Outcomes ?



OBJECTIVES

MAIN OBJECTIVE

Develop and implement a system that automates the grading of student assignments using machine learning techniques. This system aims to address the limitations of traditional grading methods by ensuring consistent, unbiased, and efficient assessment across a wide range of educational contexts.

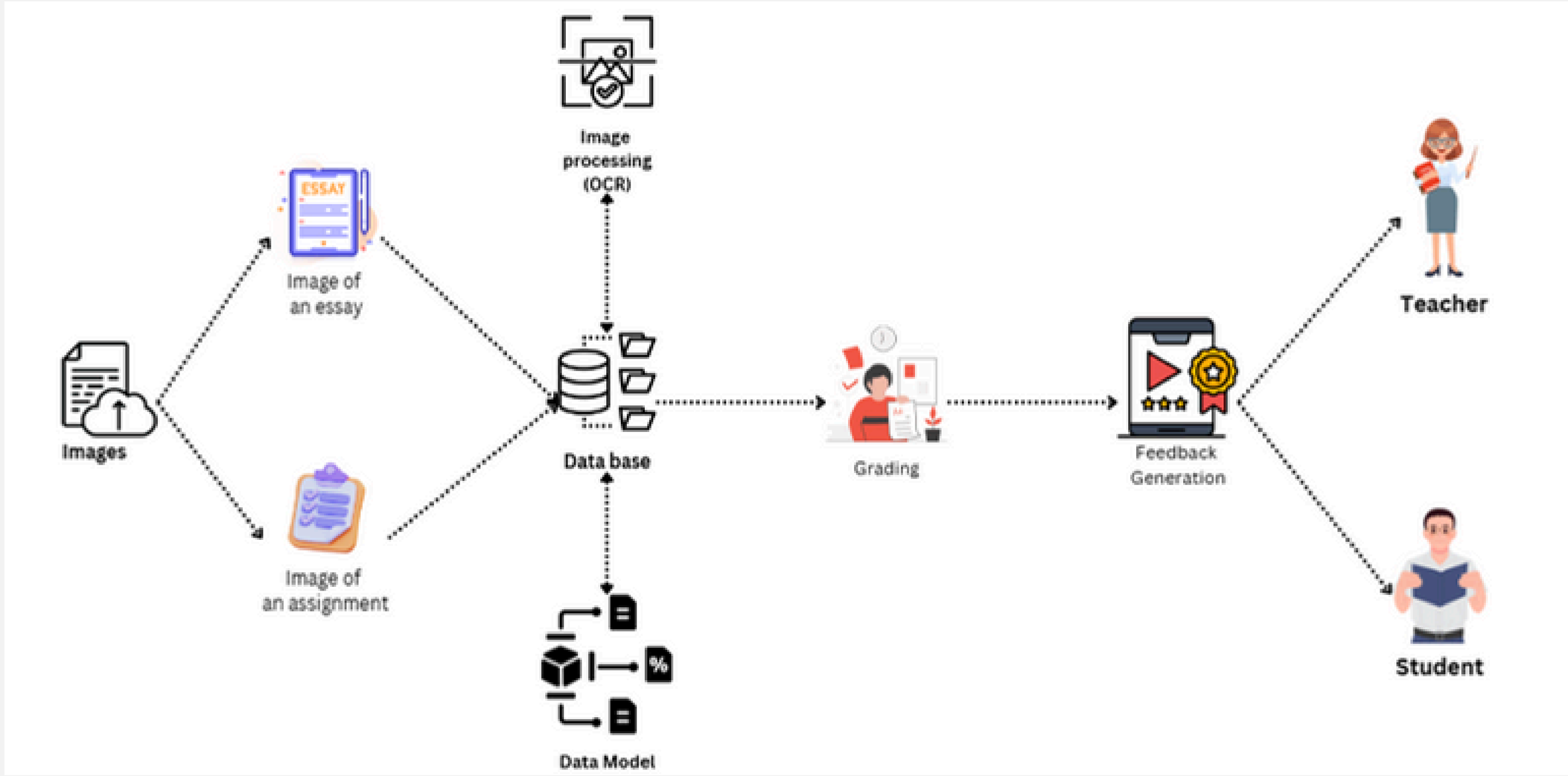


SUB OBJECTIVE

- Optimize OCR and Image Recognition for Handwritten Content
- Implement the bias detection and mitigate
- Promote transparency and trust
- Design a User-Friendly Interface

EDUSMART

METHODOLOGY

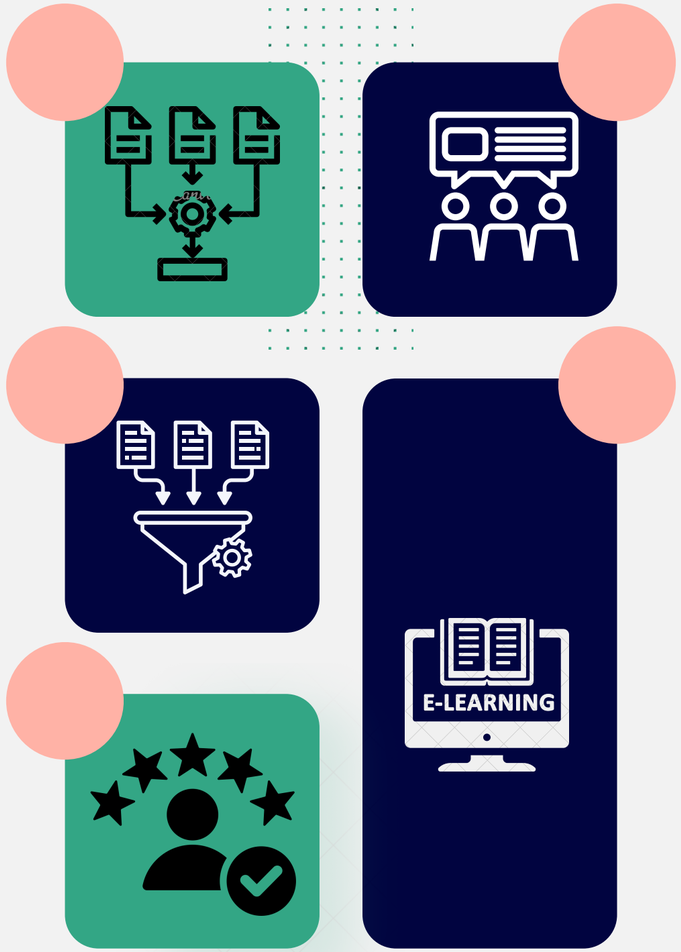


Component Overview Diagram

COMPONET SPECIFIC REQUIREMENTS

FUNCTIONAL REQUIREMENT

- Assignment Submission.
- Data Pre-processing.
- Feedback Generation.
- Automated Grading.
- Grade Compilation and Reporting.



NON-FUNCTIONAL REQUIREMENT

- Performance
- Scalability
- Usability
- Reliability
- Maintainability

Methodology Evidence of Completion

EDUSMART

- Data pre-processing
- Model testing
- Model training
- Data Visualization

❖Data Pre-processing

EDUSMART

```python
def mean_pooling(model_output, attention_mask):
    token_embeddings = model_output[0] #First element of model_output contains all token embeddings
    input_mask_expanded = attention_mask.unsqueeze(-1).expand(token_embeddings.size()).float()
    return torch.sum(token_embeddings * input_mask_expanded, 1) / torch.clamp(input_mask_expanded.sum(1), min=1e-9)

def get_sentence_embeddings(sentence):
    encoded_input = tokenizer([sentence], padding=True, truncation=True, return_tensors='pt')
    with torch.no_grad():
        model_output = model(**encoded_input)
    sentence_embeddings = mean_pooling(model_output, encoded_input['attention_mask'])
    sentence_embeddings = F.normalize(sentence_embeddings, p=2, dim=1)
    return sentence_embeddings
```

```python
    encoded_input_01 = self.tokenizer(
                                [text_pair[0]],
                                padding='max_length',
                                truncation=True,
                                max_length=self.max_length,
                                return_tensors='pt'
                                )

    encoded_input_02 = self.tokenizer(
                                [text_pair[1]],
                                padding='max_length',
                                truncation=True,
                                max_length=self.max_length,
                                return_tensors='pt'
                                )
```

❖Model training

EDUSMART

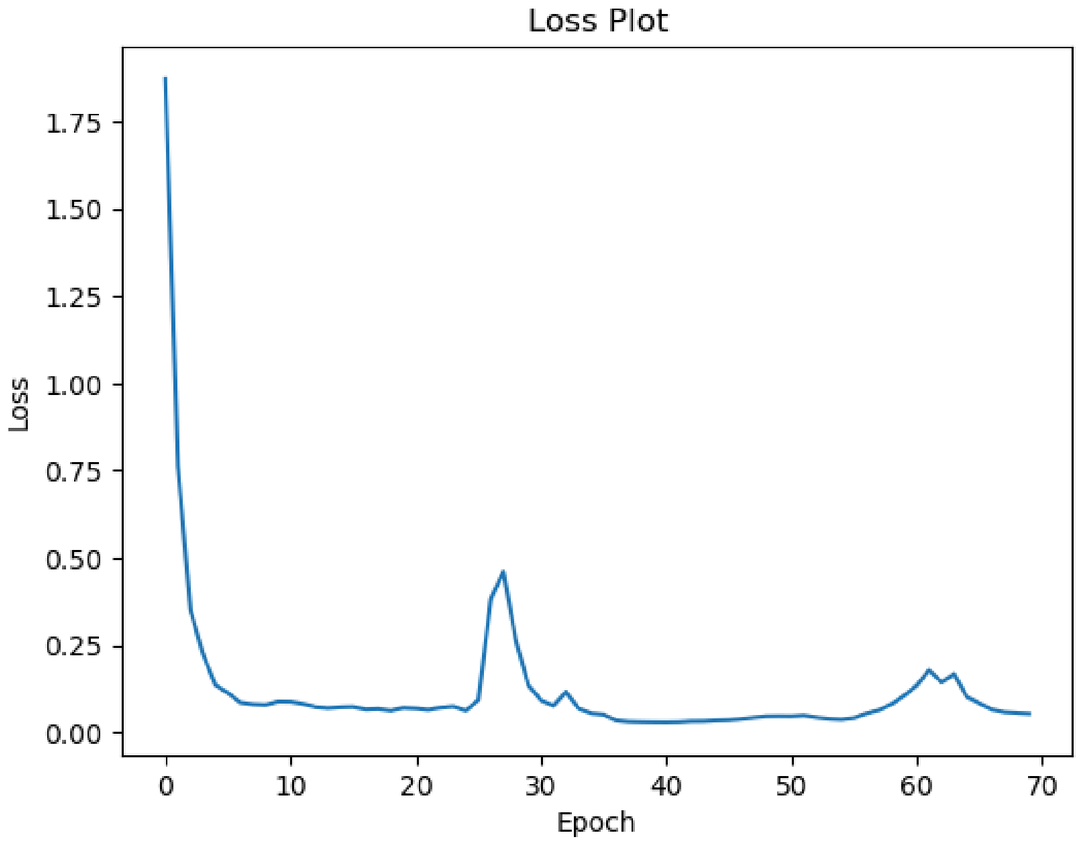
```python
optim = torch.optim.AdamW(model.parameters(), lr=5e-5)
model.to(device)

all_losses = []
for epoch in range(70):
    epoch_loss = 0
    for batch in data_loader:
        optim.zero_grad()
        batch_encoded_input_01 = batch['encoded_input_01'].to(device)
        batch_encoded_input_02 = batch['encoded_input_02'].to(device)
        batch_similarity = batch['similarity'].to('cpu')

        model_output_01 = model(**batch_encoded_input_01)
        model_output_02 = model(**batch_encoded_input_02)

        sentence_embeddings_01 = mean_pooling(model_output_01, batch_encoded_input_01['attention_mask'])
        sentence_embeddings_01 = F.normalize(sentence_embeddings_01, p=2, dim=1)

        sentence_embeddings_02 = mean_pooling(model_output_02, batch_encoded_input_02['attention_mask'])
        sentence_embeddings_02 = F.normalize(sentence_embeddings_02, p=2, dim=1)

        cosine_scores = F.cosine_similarity(x1=sentence_embeddings_01, x2=sentence_embeddings_02)
        rating_scores = cosine_scores * 5

        loss = F.mse_loss(rating_scores, batch_similarity)
        loss.backward()
        optim.step()

        epoch_loss += loss.item()
    epoch_loss = epoch_loss / len(data_loader)
    all_losses.append(epoch_loss)

    print('Epoch: {}, Loss: {}'.format(epoch, epoch_loss))
```

```
C:\Users\isuru\.cache\huggingface\modules\transformers_modules\Alibaba-NLP\new-impl
  attn_output = torch.nn.functional.scaled_dot_product_attention(
Epoch: 0, Loss: 1.8707346485058467
Epoch: 1, Loss: 0.7634115571280321
Epoch: 2, Loss: 0.35205947096149126
Epoch: 3, Loss: 0.22527688572804133
Epoch: 4, Loss: 0.13597518651435772
Epoch: 5, Loss: 0.11338813555737337
Epoch: 6, Loss: 0.08606662598749001
Epoch: 7, Loss: 0.08103099579612415
Epoch: 8, Loss: 0.07934459892722467
Epoch: 9, Loss: 0.08906140224076807
Epoch: 10, Loss: 0.08867461481442054
Epoch: 11, Loss: 0.08230964821142454
Epoch: 12, Loss: 0.07352609477937222
Epoch: 13, Loss: 0.07040445092444618
Epoch: 14, Loss: 0.07325003951787949
Epoch: 15, Loss: 0.07428249281831086
Epoch: 16, Loss: 0.06637087048962713
Epoch: 17, Loss: 0.06799084108943741
Epoch: 18, Loss: 0.06289987468005469
Epoch: 19, Loss: 0.07101501033641398
Epoch: 20, Loss: 0.06922808965978523
Epoch: 21, Loss: 0.06538456826470793
Epoch: 22, Loss: 0.07187004467161993
Epoch: 23, Loss: 0.07525507293641567
Epoch: 24, Loss: 0.06289290042904516
Epoch: 25, Loss: 0.0935435133241117
Epoch: 26, Loss: 0.3832490480815371
```

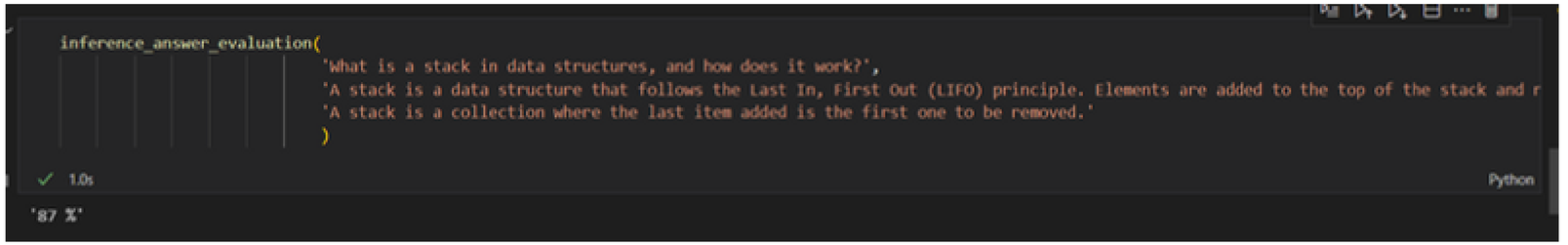
❖ Data Visualization

EDUSMART

Data training Loss plot



❖Model testing



EDUSMART

Completion of the components

1. Model Selection & Training
2. Answer Grading
3. Input Processing
4. Train model using sample dataset

60%

Future Works

1. Adding Feedback Mechanism
2. Image uploading

40%

REFERENCE

[1]Wang, Zining, Jianli Liu, and Ruihai Dong. "Intelligent auto-grading system." 2018 5th IEEE international conference on cloud computing and intelligence systems (CCIS). IEEE, 2018.

[2] Joundy Hazar, Manar, Zinah Hussein Toman, and Sarah Hussein Toman. "Automated scoring for essay questions in e-learning." Journal of Physics: Conference Series. Vol. 1294. No. 4. IOP Publishing, 2019.

[3] Rokade, Amit, et al. "Automated grading system using natural language processing." 2018 Second international conference on inventive communication and computational technologies (ICICCT). IEEE, 2018.

TOOLS & TECHNOLOGIES

EDUSMART

Database

- MongoDb

Middle Ware Technologies

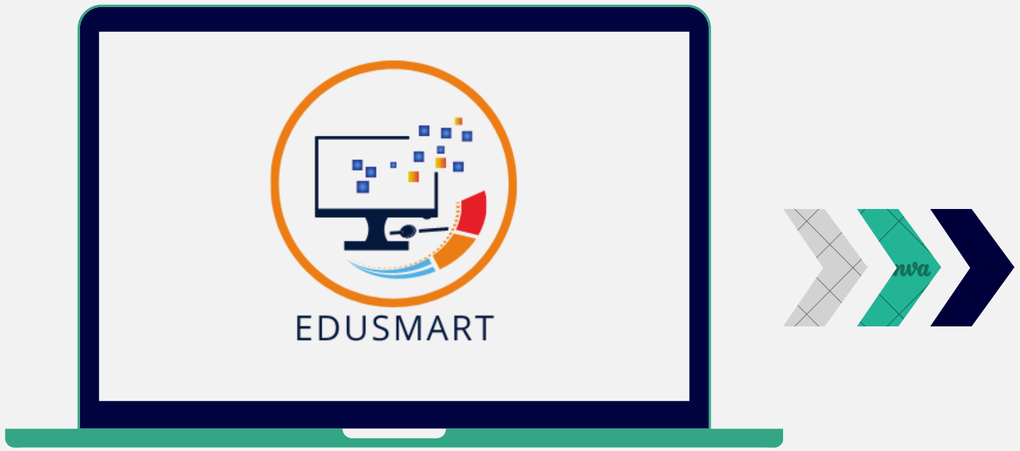
- Python
- TensorFlow
- REST API - Flask

Technical Concepts

- Machine Learning
- Convolutional Neural Network (CNN)
- Image Processing

COMMERCIALIZATION ASPECTS

EDUSMART

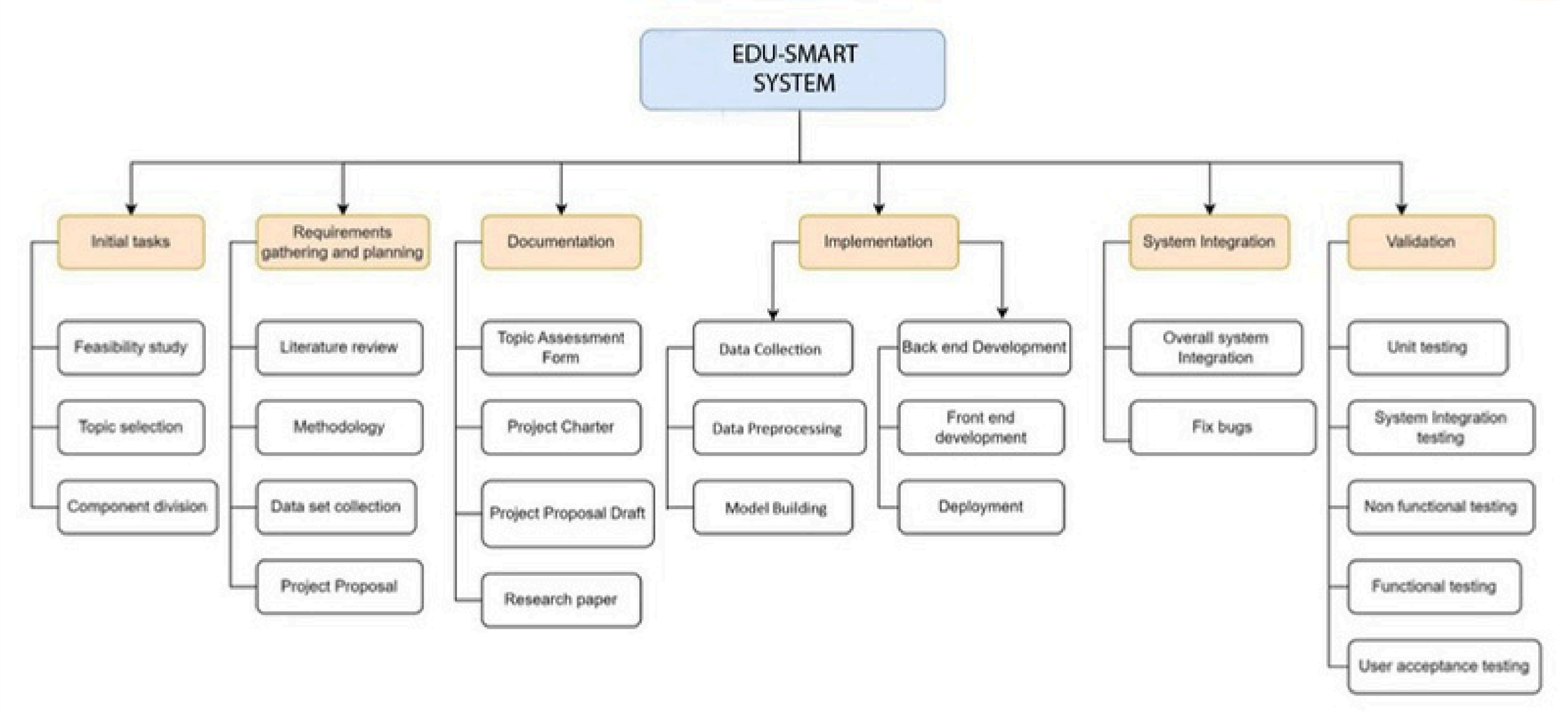


Using the proposed application, we suppose to provide subscription plans.

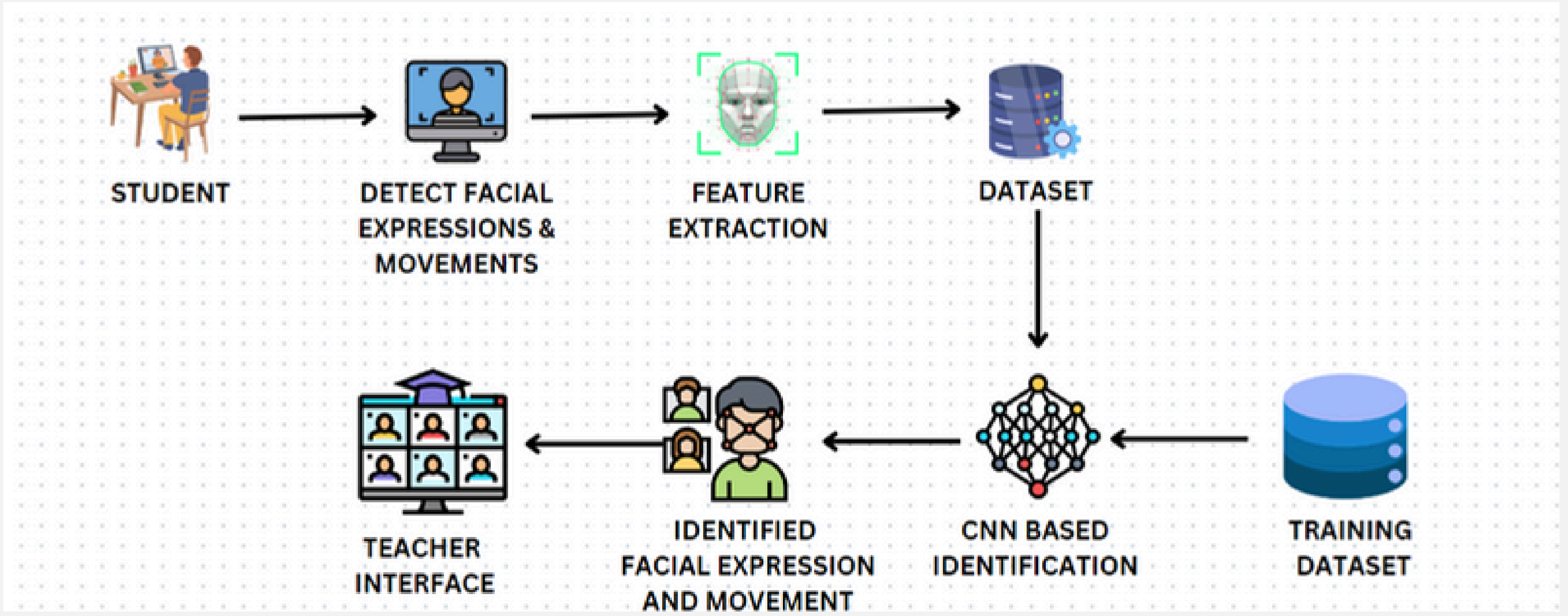
GANTT CHART

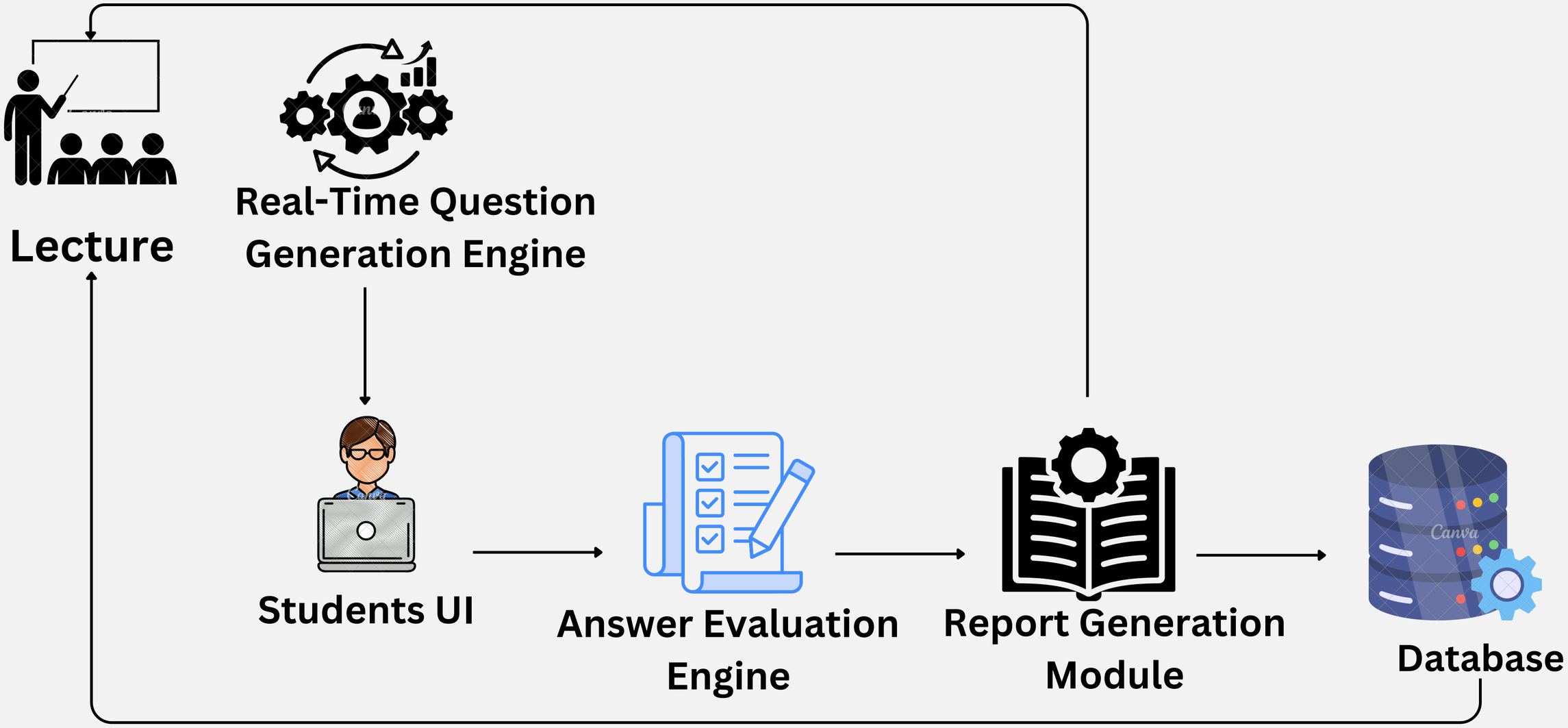
No	Assessment / Milestone	2024-2025													
		4	5	6	7	8	9	10	11	12	1	2	3	4	5
1	Project discussion workshop														
2	Topic evaluation														
3	Project proposal report														
4	Develop the system														
5	Progress Presentation - I														
6	Research Paper														
7	Progress Presentation - II														
8	Final Report Submission														
9	Final Presentation & Viva														

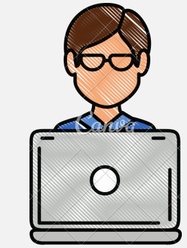
WORK BREAKDOWN CHART



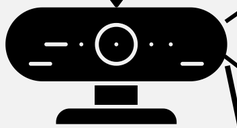
THANK YOU


STUDENT
DETECT FACIAL EXPRESSIONS & MOVEMENTS
FEATURE EXTRACTION
DATASET
TRAINING DATASET
CNN BASED IDENTIFICATION
IDENTIFIED FACIAL EXPRESSION AND MOVEMENT
TEACHER INTERFACE


Lecture
Real-Time Question Generation Engine
Students UI
Answer Evaluation Engine
Report Generation Module
Database



STUDENT



WEB CAM

HEAD POSE AND
OBJECT DETECTION

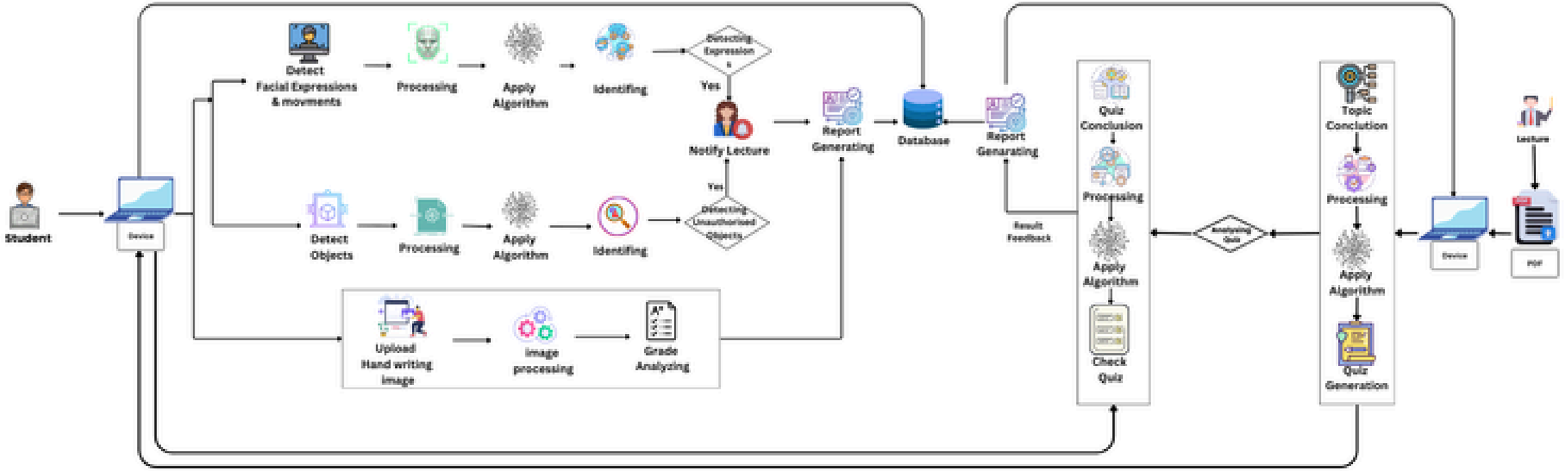
User Verification

USE THE DATA SET &
MACHINE LEARN MODEL

IDENTIFIED OBJECT
DETECTION
&
HEAD POSE
DETECTION

REAL TIME
NOTIFY TO
LECTURER

Report
Generating
HOW TO
PERFORM EXAM


Student
Device
Detect
Facial Expressions
& movements
Processing
Apply
Algorithm
Identifing
Detecting
Expression
Yes
Notify Lecture
Detect
Objects
Processing
Apply
Algorithm
Identifing
Yes
Detecting
Unauthorized
Objects
Upload
Hand writing
image
Image
processing
Grade
Analyzing
Report
Generating
Database
Report
Genarating
Result
Feedback
Quiz
Conclusion
Processing
Apply
Algorithm
Check
Quiz
Analyzing
Quiz
Topic
Conclusion
Processing
Apply
Algorithm
Quiz
Generation
Device
Lecture
PDF

Detect Facial Expressions

Detecting Expressions

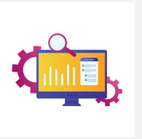
Analyze topic and quiz genaration

Student

Device

Notify Lecture

Database



Report Generating

Answer Evaluation

Detect Objects

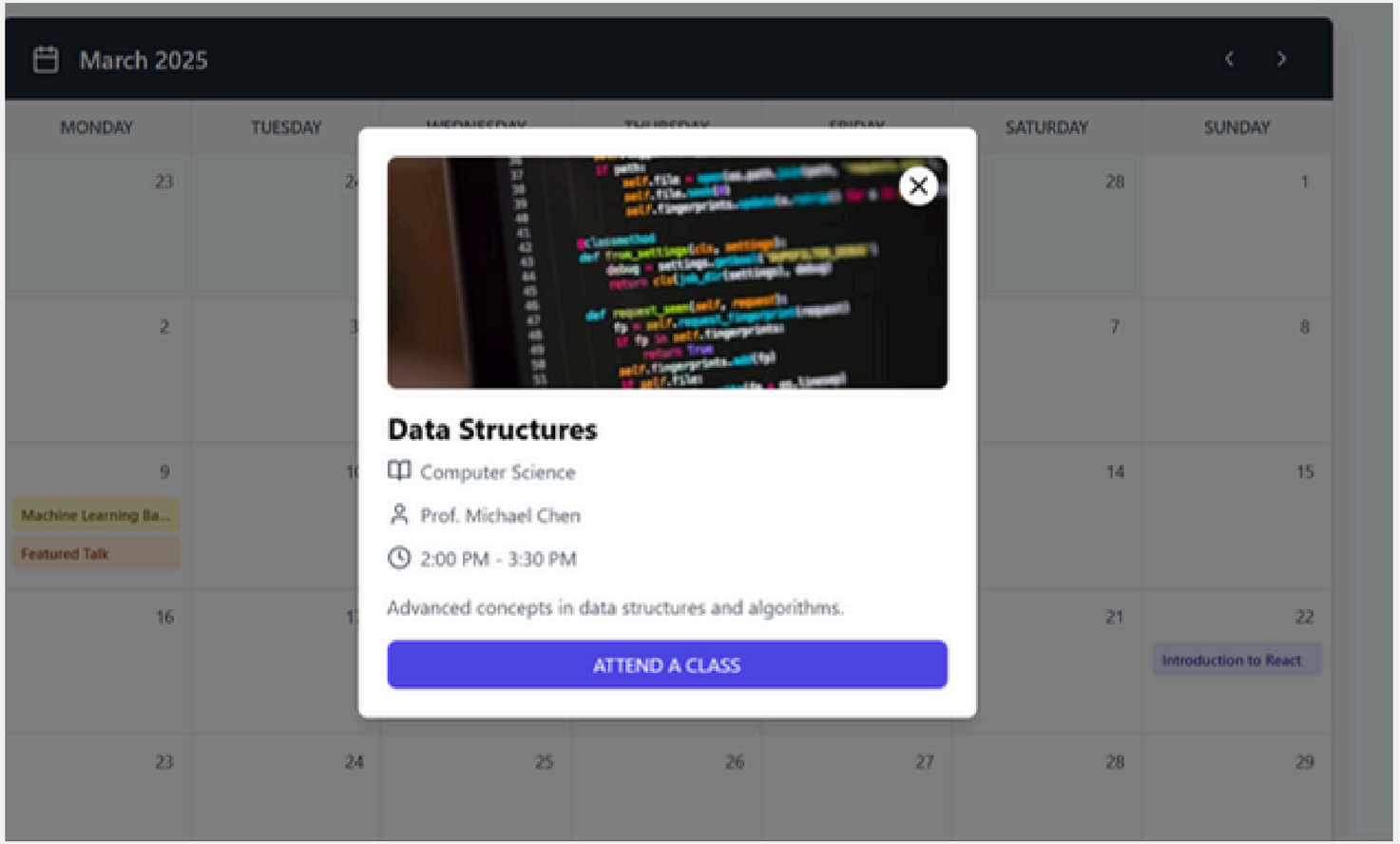
Detecting Unauthorized Objects

Answer the questions

Upload Hand Writing Image

Grade Analyzing

Analyze Question and answer


March 2025
MONDAY
TUESDAY
SATURDAY
SUNDAY
23
28
1
2
7
8
9
14
15
Machine Learning Ba...
Featured Talk
16
21
22
Introduction to React
23
24
25
26
27
28
29
Data Structures
Computer Science
Prof. Michael Chen
2:00 PM - 3:30 PM
Advanced concepts in data structures and algorithms.
ATTEND A CLASS