Technical specifications (TS)

Project name: Implementation of artificial intelligence in the educational process
Language of communication: Uzbek
Work mode: Offline

---

1. Project objective

Automation of the educational process using artificial intelligence, increasing the effectiveness of education and reducing the workload of teachers. All communication and interfaces will be in Uzbek, and the system will operate offline.

---

2. Functional requirements

2.1. Readers recognition

Apply face recognition in offline mode through a camera installed in the class gate.

Automatically calculate the number of students who entered the class.

Store student data from the list in a local database. Registration of

students' classroom attendance.

2.2. Conducting the lesson

After the number of students is counted, artificial intelligence will automatically start the lesson.

Presentations or abstracts related to the lesson topic are uploaded to the local database.

Voice presentation of content in Uzbek (Text-to-Speech technology).

The presentation is visually displayed on the screen.

2.3. Q&A section

After the lesson ends, the artificial intelligence asks: "Do you have any questions?"

Students can ask questions in the form of voice or text (Speech-to-Text technology).

Questions are analyzed through the local NLP (natural language processing) system and searched for answers in the database

If the answer is found, artificial intelligence will respond with a voice or in the form of text.

If the answer is not found, the question is automatically directed to the teacher (through the local communication system).

---

3. Technical requirements

3.1. Equipment

Device	Requirements	Notes
Camera	- Full HD (1080p) - 30+ frames/sec - Wide viewing angle - Support offline face friend	Hikvision DS-2CD2085FWD-I or similar
Server/Personal Computer	- Intel Core i7 or Ryzen 7 (8+ cores) - RAM 16-32 GB - NVIDIA RTX 3060 or high - SSD 512 GB+	For local data processing
Microphones	- Capacitor or directed - USB or XLR - Noise reduction	Blue Yeti USB or Shure SM58
Columns	- Power 20-50 W - Clear voice delivery - Excitement type: wired	JBL Control 25, Bose FreeSpace
Monitor/Projector	Board 100 inches	
Local Area Network	Gigabit Ethernet for equipment connection	Doesn't require stable and fast internet

3.2. Software

Module	Definition	Suggestions
Face recognition module	Open source that works offline, like OpenCV, Dlib, or similar	Easy to work offline

Module	Definition	Suggestions
Text-to-voice (TTS)	Local or cloud solutions supporting the Uzbek language	eSpeak, Festival, or similar
Voice-to-Text (STT)	Vosk or others to convert voice to text in Uzbek	Special preparation is required for the Uzbek language
Natural language processing (NLP)	Analysis of questions, search for answers	Hugging Face models, based on BERT or GPT
Database	Local database for students, lesson materials, and Q&A	PostgreSQL or MySQL
Admin Panel	Upload materials, view statistics, manage system	React/Vue frontend, Node.js or Django backend
Teacher notification system	Send questions, alerts	Telegram Bot, Email or local app

---

4. Security & privacy

All data is stored on the local server, not transmitted to the Internet.

Data is encrypted (TLS or other methods).

Authentication and role-based access permissions will be implemented

Personal data is protected in accordance with the legislation of the Republic of Uzbekistan.

---

5. System language

All voice and text communication will be conducted in Uzbek.

The interface and alerts are also in Uzbek

Launch Face Recognition - Write code for the offline face recognition system with OpenCV and Dlib;

Voice-to-Text (VTT) - Adapt or use Vosk or Coqui VTT to Uzbek;

Text-to-voice (TTS) - Voice in Uzbek with eSpeak or Mozilla TTS;

Preparation of the NLP module - Use and integration of Hugging Face models for Q&A in the Uzbek language;

---

1. Face Recognition

Required libraries:

pip install opencv-python dlib face_recognition

Code example (offline face recognition):

```
import cv2import face_recognition
# Close camera
video_capture = cv2.VideoCapture (0)
# Upload database for friendship (e.g., student photos) known_face_encodings = []
known_face_names = []
# Add one friend info for example:
image = face_recognition.load_image_file ("student1.jpg")
face_encoding = face_recognition.face_encodings (image) [0]
known_face_encodings.append (face_encoding)
known_face_names.append ("Islam")
while True:
   ret, frame = video_capture.read
   () rgb_frame = frame[:,:,::-1]

   face_locations = face_recognition.face_locations (rgb_frame)
   face_encodings = face_recognition.face_encodings (rgb_frame, face_locations)

   for (top, right, bottom, left), face_encoding in zip (face_locations, face_encodings):
      matches = face_recognition.compare_faces (known_face_encodings,
face_encoding)

      name = "Unknown"

      if True in matches:
         first_match_index = matches.index (True)
```

```
            name = known_face_names[first_match_index]

        cv2.rectangle (frame, (left, top), (right, bottom), (0, 255, 0), 2) cv2.putText
        (frame, name, (left, top-10), cv2.FONT_HERSHEY_SIMPLEX, 0.9,
(255, 255, 255), 2)

    cv2.imshow ('Video, frame)

    if cv2.waitKey (1) & 0xFF == ord ('q'):
        break

video_capture.release ()
cv2.destroyAllWindows ()
```

---

2. Speech-to-Text (STT)

Install wax:

pipe install wax

Model for the Uzbek language (imaginary):

https://alphacephei.com/vosk/models - you can download the Uzbek language model here (if available), or work with another language model will begin.

Code example:

```
import osimport sysimport wavefrom vosk import Model, KaldiRecognizer
if not os.path.exists ("model"):
    print ("Model folder not found!") sys.exit
    ()

wf = wave.open ("test.wav," "rb")
model = Model ("model")
rec = KaldiRecognizer (model, wf.getframerate ())
while True:
    data = wf.readframes (4000)
    if len (data) = 0:
        break
    if rec.AcceptWaveform
        (data): print (rec.Result ())
print (rec.FinalResult ())
```

---

3. Text-to-Speech (TTS)

Install and use eSpeak:

In Linux:

sudo apt-get install espeak in

Python:

```
import os

text = "Hello, this is text in Uzbek."
os.system (f'espeak -v uz "{text}"')
```

Mozilla TTS is a larger project, but with high-quality sound.

---

4. Natural language processing (NLP)

Install Hugging Face models:

pip install transformers torch

Code example (using the BERT or XLM-R model in Uzbek):

```
from transformers import AutoModelForQuestionAnswering, AutoTokenizerimport
torch

tokenizer = AutoTokenizer.from_pretrained ("uzbek-bert/bert-base-uzbek-cased")
model = AutoModelForQuestionAnswering.from_pretrained ("uzbek-bert/bert-
base-uzbek-cased")

context = "Tashkent is the capital of Uzbekistan."
question = "Where is the capital of Uzbekistan?"

inputs = tokenizer (question, context, return_tensors="pt") with torch.no_grad
   (): outputs = model (**inputs)

answer_start = torch.argmax (outputs.start_logits)
answer_end = torch.argmax (outputs.end_logits) + 1

answer = tokenizer.convert_tokens_to_string (tokenizer.convert_ids_to_tokens
(inputs['input_ids '][0][answer_start:answer_end]))
print ("Answer:," answer)
```

(Uzbek models are required for this example, or NLP models of other queries may be used.)